Actes des 13es Journées d'Intelligence Artificielle Fondamentale

Sylvie Doutre, Tiago de Lima

AfIA

Association française
pour l'Intelligence Artificielle

JIAF

Journées de l'Intelligence Artificielle Fondamentale

PFIA 2019

Table des matières

Éditorial

Les Journées d'Intelligence Artificielle Fondamentale (JIAF) constituent un rendez-vous annuel de la communauté francophone travaillant sur l'Intelligence Artificielle Fondamentale. Les thématiques de recherche sont relatives aux méthodes et outils fondamentaux de l'Intelligence Artificielle, tel que :

- Définition de modèles de représentation des informations (croyances, connaissances, préférences, obligations et permissions, actions, incertitude, confiance, réputation) : langages des logiques classiques ou non classiques, modèles possibilistes, ontologies, langages à base de contraintes, représentations graphiques, etc.
- Définition et automatisation de raisonnements sur ces informations : raisonnement spatio-temporel, dynamique des informations, révision de croyances, fusion d'informations symboliques, raisonnement par argumentation, raisonnement causal, raisonnement abductif, raisonnement à partir de cas, etc.
- Mise au point de méthodes de codage des informations et d'algorithmes de traitement efficaces : compilation de connaissances, SAT, contraintes, ASP, etc.
- Modélisation formelle de l'interaction : entre utilisateurs et systèmes informatiques, entre entités informatiques autonomes (agents), intégration de ces deux aspects dans les divers agents conversationnels, agents de recherche, assistants personnels.
- Choix social, théorie des jeux, algorithmes pour les jeux.
- Pour des objectifs de décision, planification, ordonnancement, diagnostic, apprentissage et dans différents contextes d'application, comme par exemple le Web sémantique.

Les journées sont composées d'exposés de synthèse, permettant à la communauté de découvrir des thématiques connexes au travers d'exposés de spécialistes, et de communications sélectionnées par le comité de programme.

Les éditions précédentes se sont déroulées à Amiens (2018), Caen (2017), Montpellier (2016), Rennes (2015), Angers (2014), Aix-en-Provence (2013), Toulouse (2012), Lyon (2011), Strasbourg (2010), Marseille (2009), Paris (2008) et Grenoble (2007).

Ces 13èmes Journées d'Intelligence Artificielle Fondamentale (JIAF 2019) ont eu lieu du 1er au 3 juillet 2019, dans le cadre de la Plate-Forme Intelligence Artificielle (PFIA).

Sylvie DOUTRE, Tiago DE LIMA

Comité de programme

Présidents

— Tiago de Lima (CRIL, Université d'Artois)
— Sylvie Doutre (IRIT, Université Toulouse 1 Capitole)

Membres

— Francesco Belardinelli (IBISC, Université d'Évry)
— Elise Bonzon (LIPADE, Université Paris Decartes)
— Tristan Cazenave (LAMSADE, Université Paris Dauphine)
— Nadia Creignou (LIS, Aix-Marseille Université)
— Jérôme Euzenat (LIG, INRIA)
— George Katsirelos (MIAT, INRA)
— Sébastien Konieczny (CRIL, CNRS)
— Jérôme Lang (LAMSADE, Université Paris Dauphine)
— Jean Lieber (LORIA, INRIA)
— Pierre Marquis (CRIL, Université d'Artois)
— Amedeo Napoli (LORIA, CNRS)
— Odile Papini (LIS, Aix-Marseille Université)
— Laurent Perrussel (IRIT, Université Toulouse 1 Capitole)
— Sophie Pinchinat (IRISA, INRIA)
— Stéphanie Roussel (ONERA)
— Serena Villata (I3S, CNRS)
— Christel Vrain (LIFO, Université d'Orléans)
— Bruno Zanuttini (GREYC, UNICAEN)

Actes JIAF 2019

A gentle introduction to unification in modal logics

P. Balbiani[a *] **Ç. Gencer**[a,b] **M. Mojtahedi**[c] **M. Rostamigiv**[a] **T. Tinchev**[d]

[a]Institut de recherche en informatique de Toulouse
CNRS — Toulouse University, Toulouse, France
[b]Faculty of Arts and Sciences
Istanbul Aydın University, Istanbul, Turkey
[c]College of Sciences
Tehran University, Tehran, Iran
[d]Department of Mathematical Logic and its Applications
Sofia University St. Kliment Ohridski, Sofia, Bulgaria

Abstract

Unification in propositional logics is an active research area. In this paper, we introduce the results we have obtained within the context of modal logics and epistemic logics and we present some of the open problems whose solution will have an important impact on the future of the area.

Résumé

L'unification dans les logiques propositionnelles est un domaine de recherche actif. Dans cet article, nous présentons les résultats que nous avons obtenus dans le cadre des logiques modales et des logiques épistémiques et nous introduisons quelques uns des problèmes ouverts dont la résolution aura un impact important sur l'avenir du domaine.

1 Introduction

The problem of solving equations was at the heart of the algebra of logic created by Boole. In modern terms, owing to the fact that given $n \in \mathbb{N}$ and pairs $(\varphi_1, \psi_1), \ldots, (\varphi_n, \psi_n)$ of Boolean formulas, the system consisting of the n equivalences $\varphi_1 \leftrightarrow \psi_1, \ldots, \varphi_n \leftrightarrow \psi_n$ can be readily transformed into an equivalent system $\chi \leftrightarrow \top$ consisting of only one equivalence, the problem of solving equations can be interpreted as a satisfiability problem, or as a unification problem. The satisfiability problem asks whether the variables of χ can be uniformly replaced by constant formulas (i.e. the formulas $\bot$ and $\top$) in such a way that the resulting formula evaluates to 1 in the two-element Boolean algebra. The unification problem asks whether the variables of χ can be uniformly replaced by formulas (i.e. arbitrary formulas) in such a way that the resulting formula evaluates to 1 under all truth assignments in the two-element Boolean algebra.

Of relevance in many applications of Artificial Intelligence, the satisfiability problem has given rise to a significant corpus of results in automated reasoning. This is borne out by the numerous problems having translations to the satisfiability problem and by the multifarious tools available to solve these translations. In the case of most applied logics, a similar remark also applies. Witness, the systematic interest falling on optimal procedures deciding the satisfiability problem of freshly designed applied logics rather than on elegant axiomatizations explaining the meaning of their logical connectives. In comparison, the unification problem has attracted less attention. The truth is that it has been examined from a different angle : a desirable output of the satisfiability problem is a model satisfying the given formula whereas a desirable output of the unification problem is a substitution making the

*Corresponding author. The addresses of the authors are philippe.balbiani@irit.fr, cigdemgencer@aydin.edu.tr, mojtaba.mojtahedi@ut.ac.ir, maryam.rostamigiv@irit.fr and tinko@fmi.uni-sofia.bg.

given formula valid.

From a mathematical point of view, the unification problem is strongly related to the admissibility problem which asks with respect to some predetermined propositional logics **L** whether given a rule $\frac{\varphi_1,\ldots,\varphi_n}{\chi}$, every substitution turning $\varphi_1,\ldots,\varphi_n$ into members of **L** also turns χ into a member of **L**. Firstly, since a given formula φ is unifiable if and only if, when **L** is consistent, the associated rule $\frac{\varphi}{\bot}$ is non-admissible, we can turn any algorithm deciding the admissibility problem into an algorithm deciding the unification problem. Secondly, since a given rule $\frac{\varphi_1,\ldots,\varphi_n}{\chi}$ is admissible if and only if, when **L** is finitary [1], every maximal unifier of $\varphi_1,\ldots,\varphi_n$ is also a unifier of χ, we can turn, when **L** is decidable, any algorithm producing minimal complete sets of unifiers into an algorithm deciding the admissibility problem [2]. See Ghilardi [18, 19] for illustrations within the context of intuitionistic logic and modal logics like **K**4 and **S**4. See also [17, 21].

From the point of view of Artificial Intelligence, unification in propositional logics is an active research area and several applications of the unification problem in the maintenance of knowledge bases have been considered. In this respect, within the context of a terminology of concepts, we may ask whether given $n \in \mathbb{N}$ and pairs $(C_1, D_1), \ldots, (C_n, D_n)$ of concept descriptions, a substitution can make these pairs equivalent by replacing some of their variables by appropriate concept descriptions. Moreover, we may be interested to obtain, if possible, the most general substitution that can make the pairs equivalent. See the related unification algorithms presented in [3, 4] for the description languages $\mathcal{EL}$ and $\mathcal{FL}_0$.

The thing is that there is a wide variety of situations where the unification problem arises. Let us explain our motivation for considering them within the context of Public Announcement Logic [3]. Suppose the epistemic formula $\varphi(\bar{p})$ describes a given initial situation in terms of the knowledge of a group of agents about the list of parameter facts $\bar{p} = (p_1, \ldots, p_m)$ and the epistemic formula $\langle\langle\chi(\bar{p}\bar{x})\rangle\rangle\psi(\bar{p})$ represents the knowledge of the group about $\bar{p}$ in a desirable final situation (the ψ-part) after an executable public announcement concerning $\bar{p}$ and the list of variable facts $\bar{x} = (x_1, \ldots, x_n)$ (the χ-part) has been performed. It may happen that $\varphi(\bar{p}) \rightarrow \langle\langle\chi(\bar{p}\bar{x})\rangle\rangle\psi(\bar{p})$ is not valid in Public Announcement Logic. Hence, we may ask whether there are tuples $\bar{\phi} = (\phi_1, \ldots, \phi_n)$ of formulas such that $\varphi(\bar{p}) \rightarrow \langle\langle\chi(\bar{p}\bar{\phi})\rangle\rangle\psi(\bar{p})$ is valid. Moreover, we may be interested to obtain the most general tuple $\bar{\phi}$ of formulas such that $\varphi(\bar{p}) \rightarrow \langle\langle\chi(\bar{p}\bar{\phi})\rangle\rangle\psi(\bar{p})$ is valid. In Section 5, we will prove that some unifiable epistemic formulas as above have always most general unifiers.

In this paper, we introduce the results we have obtained within the context of modal logics and epistemic logics and we present some of the open problems whose solution will have an important impact on the future of the area.

2 Unification : preliminary definitions

Syntax We consider a propositional language with a countably infinite set $\mathcal{PAR}$ of *propositional parameters* (p, q, etc), a countably infinite set $\mathcal{VAR}$ of *propositional variables* (x, y, etc), a countable set O of *operators* and an *arity function* $\rho : O \longrightarrow \mathbb{N}$. *Atoms* ($\alpha$, β, etc) are parameters, or variables. We denote this language by $\mathcal{PL}$. Its *formulas* are defined by the rule $\varphi ::= \alpha \mid o(\varphi_1, \ldots, \varphi_n)$ where o ranges over O and $n = \rho(o)$. For all formulas φ, its *set of variables* (in symbols, $\mathtt{var}(\varphi)$) is defined as usual. Let $(p_1, p_2, \ldots)$ be an enumeration of $\mathcal{PAR}$ without repetitions and $(x_1, x_2, \ldots)$ be an enumeration of $\mathcal{VAR}$ without repetitions. We write $\varphi(\bar{p}\bar{x})$ to denote a formula whose parameters form a sublist of $\bar{p} = (p_1, \ldots, p_m)$ and whose variables form a sublist of $\bar{x} = (x_1, \ldots, x_n)$. We use the standard definition for the notion of *subformula.*

Semantics Formulas of $\mathcal{PL}$ are interpreted in *algebraic structures,* or in *relational structures :* nondegenerate Boolean algebras in the case of Boolean logic, Kripke frames in the case of modal logics, etc. A set of *values* and a set of *designated values* are coming with each of these structures : A and $\{1_A\}$ in the case of the nondegenerate Boolean algebra $(A, 0_A, \star_A, +_A)$, $\mathcal{P}(W)$ and $\{W\}$ in the case of the Kripke frame (W, R), etc. *Models* are pairs of the form (S, V) where S is a structure and V is a *truth assignment* associating each formula φ with a value $V(\varphi)$ in S. We shall say that a formula φ is *true in* the model (S, V) if $V(\varphi)$ is a designated value in S. Let $\mathcal{C}$ be a class of structures. We shall say that a formula is $\mathcal{C}$*-valid* if it is true in all models based on a structure in $\mathcal{C}$. The *propositional logic* determined by $\mathcal{C}$ is the set of all $\mathcal{C}$-valid formulas : **BL** in the case of Boolean logic, **K**4, **S**4, etc in the case of modal logics, etc.

Substitutions A *substitution* is a homomorphism $\sigma : \mathcal{PL} \longrightarrow \mathcal{PL}$ which moves at most finitely many va-

1. See Section 2 for a definition.

2. It is well-known that owing to its structural completeness, Boolean logic has a decidable admissibility problem. It was only after the results of Rybakov [29, 30] that it has been known that intuitionistic logic and modal logics like **K**4 and **S**4 have a decidable admissibility problem too.

3. In a multi-agent system, public announcements are atomic actions performed by an outside observer consisting of publicly announcing a formula [12, Chapter 4]. Perceived by all agents, it is common knowledge that these public announcements are truthful.

riables. It is *parameter-free* if for all variables x, $\sigma(x)$ is parameter-free. It is *ground* if for all variables x, $\sigma(x)$ is variable-free. The *composition* of the substitutions σ and τ is the substitution $\sigma \circ \tau$ such that for all variables x, $(\sigma \circ \tau)(x) = \tau(\sigma(x))$.

We shall say that a substitution σ is $\mathcal{C}$*-equivalent to* a substitution τ (in symbols $\sigma \simeq \tau$) if for all variables x, each model associate $\sigma(x)$ and $\tau(x)$ with the same value. We shall say that a substitution σ is *more general than* a substitution τ (in symbols $\sigma \preceq \tau$) if there exists a substitution υ such that $\sigma \circ \upsilon \simeq \tau$. Obviously, $\preceq$ contains $\simeq$. Moreover, on the set of all substitutions, $\simeq$ is reflexive, symmetric and transitive and $\preceq$ is reflexive and transitive. We shall say that a set Σ of substitutions is *minimal* if for all $\sigma, \tau \in \Sigma$, if $\sigma \preceq \tau$ then $\sigma = \tau$.

Unifiers We shall say that a formula φ is $\mathcal{C}$*-unifiable* if there exists a substitution σ such that $\sigma(\varphi)$ is $\mathcal{C}$-valid. In that case, σ is a unifier of φ. We traditionally distinguish between *elementary unification* (*ELU*) and *unification with parameters* (*UWP*). *ELU* is the problem of asking whether a given parameter-free formula is $\mathcal{C}$-unifiable. *UWP* is the problem of asking whether a given formula is $\mathcal{C}$-unifiable. It goes without saying that the computability of the unification problem in a propositional logic may vary according to whether one considers *ELU,* or considers *UWP.*

We shall say that a set Σ of unifiers of a $\mathcal{C}$-unifiable formula φ is *complete* if for all unifiers σ of φ, there exists $\tau \in \Sigma$ such that $\tau \preceq \sigma$. It can be easily proved that if a $\mathcal{C}$-unifiable formula has minimal complete sets of unifiers then these sets have the same cardinality. Hence, an important question is the following [14] : when a formula is $\mathcal{C}$-unifiable, has it a minimal complete set of unifiers ? When the answer is "yes", how large is this set ?

Let φ be a $\mathcal{C}$-unifiable formula. φ is said to be *nullary* if there exists no minimal complete set of unifiers of φ, *infinitary* if there exists a minimal complete set of unifiers of φ but there exists no finite one, *finitary* if there exists a finite minimal complete set of unifiers of φ but there exists no with cardinality 1 and *unitary* if there exists a minimal complete set of unifiers of φ with cardinality 1. Obviously, the types "nullary", "infinitary", "finitary" and "unitary" constitute a set of jointly exhaustive and pairwise distinct situations for each unifiable formula.

$\mathcal{C}$ — or the propositional logic determined by $\mathcal{C}$ — is said to be *nullary* if there exists a nullary unifiable formula, *infinitary* if every unifiable formula possesses a minimal complete set of unifiers and there is an infinitary unifiable formula, *finitary* if every unifiable formula possesses a finite minimal complete set of unifiers and there is a finitary unifiable formula and *unitary* if every unifiable formula possesses a minimal complete set of unifiers with cardinality 1. Obviously, the types "nullary", "infinitary", "finitary" and "unitary" constitute a set of jointly exhaustive and pairwise distinct situations for each propositional logic. Let us remark that the type of a propositional logic may vary according to whether one considers *ELU*, or considers *UWP.*

3 Case of Boolean logic

Syntax and semantics In the case of Boolean logic, O consists of the usual Boolean connectives $\bot$, $\neg$ and $\vee$ together with their usual arities. We denote the associated language by $\mathcal{BPL}$. The Boolean connectives $\top$, $\wedge$, $\rightarrow$ and $\leftrightarrow$ are defined by the usual abbreviations.

Formulas of $\mathcal{BPL}$ are interpreted in nondegenerate Boolean algebras $(A, 0_A, \star_A, +_A)$ — with A as set of values and $\{1_A\}$ as set of designated values — by means of truth assignments V such that $V(\bot) = 0_A$, $V(\neg\varphi) = V(\varphi)^{\star_A}$ and $V(\varphi \vee \psi) = V(\varphi) +_A V(\psi)$.

Computability of unification The unification problem in Boolean logic has been firstly investigated by Boole, his method involving successive elimination of variables [28]. In Boolean logic, both *ELU* and *UWP* are decidable. To be more precise, *ELU* and the satisfiability problem are inter-reducible whereas *UWP* and the validity problem of $\forall\exists$-**QBF**-formulas are inter-reducible. Hence, in Boolean logic, *ELU* is **NP**-complete and *UWP* is $\Pi_2^{\mathbf{P}}$-complete. See [1] for a comprehensive analysis.

Unification types Löwenheim [24] has given a formula for the most general solution of a unification problem expressed in terms of a particular solution. Given a unifier σ of a unifiable formula φ, let ϵ_φ^σ be the substitution such that for all $x \in \mathcal{VAR}$, if $x \in \mathtt{var}(\varphi)$ then $\epsilon_\varphi^\sigma(x) = (\varphi \wedge x) \vee (\neg\varphi \wedge \sigma(x))$ else $\epsilon_\varphi^\sigma(x) = x$. By induction on the formula ψ, the reader may prove that if $\mathtt{var}(\psi) \subseteq \mathtt{var}(\varphi)$ then $\varphi \rightarrow (\epsilon_\varphi^\sigma(\psi) \leftrightarrow \psi)$ and $\neg\varphi \rightarrow (\epsilon_\varphi^\sigma(\psi) \leftrightarrow \sigma(\psi))$ are valid. Since σ is a unifier of φ, we obtain the validity of $\varphi \rightarrow \epsilon_\varphi^\sigma(\varphi)$ and $\neg\varphi \rightarrow \epsilon_\varphi^\sigma(\varphi)$. Moreover, for all unifers τ of φ and for all variables x, $\tau(\epsilon_\varphi^\sigma(x)) \leftrightarrow \tau(x)$ is valid. Hence, ϵ_φ^σ is a unifier of φ and for all unifiers τ of φ, $\epsilon_\varphi^\sigma \preceq \tau$. Thus, ϵ_φ^σ is a most general unifier of φ : it constitutes by its own a complete set of unifiers of φ. Consequently, φ is unitary.

In the case of *ELU,* given a unifiable parameter-free formula φ, we can define a most general unifier of it by firstly considering a truth assignment V in the two-element Boolean algebra such that $V(\varphi) = 1$ and by

secondly defining the substitution ϵ' such that for all $x \in \mathcal{VAR}$, if $x \in \mathtt{var}(\varphi)$ then $\epsilon'(x) =$ "if $V(x) = 0$ then $\varphi \wedge x$ else $\varphi \rightarrow x$" else $\epsilon'(x) = x$. Obviously, the substitution ϵ' is equivalent to the substitution $\epsilon^{\sigma_V}_{\varphi}$ defined in the previous paragraph when σ_V is the parameter-free ground substitution such that for all $x \in \mathcal{VAR}$, if $x \in \mathtt{var}(\varphi)$ then $\sigma_V(x) =$ "if $V(x) = 0$ then $\bot$ else $\top$" else $\sigma_V(x) = x$.

Additional comments Restricting $\mathcal{BPL}$ to a language (denoted $\mathcal{BPL}^{\rightarrow}$) where $\rightarrow$ is the sole Boolean connective makes the satisfiability problem trivial seeing that every formula then becomes satisfiable. Nevertheless, the validity problem still remains co**NP**-hard. So, all in all, $\mathcal{BPL}^{\rightarrow}$ is at least easier than $\mathcal{BPL}$ for what concerns the satisfiability problem, or the validity problem. Surprisingly, the effect of restricting $\mathcal{BPL}$ to $\mathcal{BPL}^{\rightarrow}$ is opposite for what concerns the unification problem. This opposite effect is visible for example, in the formula $x \rightarrow (p \vee q)$ which is unifiable and non-unitary [4]. It is unifiable : $\mathcal{BPL}^{\rightarrow}$-substitutions like $\sigma_p(x) = p \vee (q \wedge x)$ and $\sigma_q(x) = (p \wedge x) \vee q$ are unifiers of it [5]. However, it is non-unitary. Let us see why. Indeed, suppose it is unitary. Let τ be a $\mathcal{BPL}^{\rightarrow}$-unifier of it such that $\tau \preceq \sigma_p$ and $\tau \preceq \sigma_q$. As can be proved by induction on $\varphi \in \mathcal{BPL}^{\rightarrow}$, if $\varphi \rightarrow (p \vee q)$ is valid then $p \rightarrow \varphi$ is valid, or $q \rightarrow \varphi$ is valid. Since τ is a unifier of $x \rightarrow (p \vee q)$, therefore $p \rightarrow \tau(x)$ is valid, or $q \rightarrow \tau(x)$ is valid. Without loss of generality, suppose $p \rightarrow \tau(x)$ is valid. Since $\tau \preceq \sigma_q$, therefore $\lambda(\tau(x)) \leftrightarrow \sigma_q(x)$ is valid for some substitution λ. Since $p \rightarrow \tau(x)$ is valid, therefore $p \rightarrow ((p \wedge x) \vee q)$ is valid : a contradiction.

This increased difficulty in the unification problem has to be related to the fact that in $\mathcal{BPL}^{\rightarrow}$, Boolean logic is losing its structural completeness. In order to understand why, let us consider the rule $\frac{x \rightarrow p}{p \rightarrow x}$. As the reader can prove, for all $\mathcal{BPL}^{\rightarrow}$-formulas φ, if $\varphi \rightarrow p$ is valid then $p \rightarrow \varphi$ is valid. Hence, the considered rule is admissible. Nevertheless, it is not derivable. Let us see why. Indeed, suppose the rule is derivable. Thus, by using the derivation theorem which still holds in $\mathcal{BPL}^{\rightarrow}$ [13, Chapter 9], the formula $(x \rightarrow p) \rightarrow (p \rightarrow x)$ is valid : a contradiction.

The type of $\mathcal{BPL}^{\rightarrow}$ for *UWP* is unknown and we conjecture that it is finitary. See Balbiani and Mojtahedi [8] for preliminary results. For *$\mathcal{BPL}^{\rightarrow}$-ELU,* it is relatively easy to prove that every formula is unifiable and unitary. In order to understand why, let $\varphi \in \mathcal{BPL}^{\rightarrow}$ be a parameter-free formula. Let $x \in \mathcal{VAR}$ and $\psi \in \mathcal{BPL}$ be such that x does not occur in ψ and $\varphi \leftrightarrow (\psi \vee x)$ is valid [6]. Let ϵ be the $\mathcal{BPL}^{\rightarrow}$-substitution such that for all $y \in \mathcal{VAR}$, if $y = x$ then $\epsilon(y) = \varphi \rightarrow x$ else $\epsilon(y) = y$. Since x does not occur in ψ and $\varphi \leftrightarrow (\psi \vee x)$ is valid, therefore $\epsilon(\varphi) \leftrightarrow (\psi \vee ((\psi \vee x) \rightarrow x))$ is valid. Moreover, for all $\mathcal{BPL}^{\rightarrow}$-unifiers τ of φ and for all $y \in \mathcal{VAR}$, $\tau(\epsilon(y)) \leftrightarrow \tau(y)$ is valid. Hence, ϵ is a $\mathcal{BPL}^{\rightarrow}$-unifier of φ and for all $\mathcal{BPL}^{\rightarrow}$-unifiers τ of φ, $\epsilon \preceq \tau$. Thus, ϵ is a most general unifier of φ : it constitutes by its own a complete set of unifiers of φ. Consequently, φ is unitary.

4 Case of modal logics

Syntax and semantics In the case of modal logics, on top of the Boolean connectives considered in Section 3, O contains the modality $\Box \ldots$ ("necessarily, ..."") of arity 1. We denote the associated language by $\mathcal{MPL}$. The modality $\Diamond \ldots$ ("possibly, ..."") of arity 1 is defined by the abbreviation : $\Diamond\varphi ::= \neg\Box\neg\varphi$. For all $n \geq 0$, we write $\underline{\varphi_1 \ldots \varphi_n}$ to mean $\varphi_1 \wedge \ldots \wedge \varphi_n$. We write φ^0 to mean $\neg\varphi$ and we write φ^1 to mean φ. For all $d \geq 0$, we write $\Box^{<d}\varphi$ to mean $\top$ when $d = 0$, $(\varphi \wedge \Box\Box^{<d-1}\varphi)$ otherwise and we write $\Box^d\varphi$ to mean φ when $d = 0$, $\Box\Box^{d-1}\varphi$ otherwise. For all formulas ψ, we write $[\psi]\varphi$ to mean $\Box(\psi \rightarrow \varphi)$. For all formulas ψ and for all $d \geq 0$, we write $[\psi]^{<d}\varphi$ to mean $\top$ when $d = 0$, $(\varphi \wedge [\psi][\psi]^{<d-1}\varphi)$ otherwise and we write $[\psi]^d\varphi$ to mean φ when $d = 0$, $[\psi][\psi]^{d-1}\varphi$ otherwise. For all formulas φ, its *modal degree* (in symbols, $\deg(\varphi)$) is defined as usual.

Formulas of $\mathcal{MPL}$ are interpreted in Kripke frames (W, R) — with $\mathcal{P}(W)$ as set of values and $\{W\}$ as set of designated values — by means of truth assignments V such that $V(\bot) = \emptyset$, $V(\neg\varphi) = W \setminus V(\varphi)$, $V(\varphi \vee \psi) = V(\varphi) \cup V(\psi)$ and $V(\Box\varphi) = \{s \in W :$ for all $t \in W$, if $R(s,t)$ then $t \in V(\varphi)\}$. In this section, we will consider the modal logics enumerated in Table 1.

Computability of unification In some popular modal logics, to solve the unification problem is not a mere formality. The truth is that contrary to the case of Boolean logic, there exists parameter-free formulas that are satisfiable without being unifiable. To see why, take the parameter-free formula $\Diamond x \wedge \Diamond\neg x$. In many modal logics such as **K**4, **S**4, etc, this formula is satisfiable. Nevertheless, it is known that if this formula is unifiable in a modal logic **L** then **L** is inconsistent [15, 27]. The thing is that, while attacking the unification problem, little, if anything, from the standard tools in modal logics (canonical models,

4. In $\mathcal{BPL}^{\rightarrow}$, the Boolean connective $\vee$ is definable by $(\varphi \vee \psi) ::= ((\varphi \rightarrow \psi) \rightarrow \psi)$.

5. As the reader can prove, formulas like $p \vee (q \wedge x)$ and $(p \wedge x) \vee q$ are definable in $\mathcal{BPL}^{\rightarrow}$.

6. Given a parameter-free formula $\varphi \in \mathcal{BPL}^{\rightarrow}$, the existence of $x \in \mathcal{VAR}$ and $\psi \in \mathcal{BPL}$ such that x does not occur in ψ and $\varphi \leftrightarrow (\psi \vee x)$ is valid can be proved by induction on φ.

Modal logics	Class of Kripke frames
K	All
KD	Serial
KT	Reflexive
KB	Symmetric
KDB	Serial symmetric
KTB	Reflexive symmetric
KG	Church-Rosser
KDG	Serial Church-Rosser
KTG	Reflexive Church-Rosser
K4	Transitive
S4	Reflexive transitive
S5	Reflexive transitive Euclidean
$\mathbf{Alt}_1$	Deterministic

TABLE 1 – Some modal logics together with the classes of Kripke frames that determine them.

filtrations, etc) is helpful — to such an extent that **the computability of the following problems remains open : *ELU* in K and KB and *UWP* in KD, KT, KDB, KTB and $\mathbf{Alt}_1$.**

The case of $\mathbf{Alt}_1$ constitutes a good example of what are the difficulties of the unification problem in modal logics. Within the context of this modal logic, the modality $\Box$ corresponds in Kripke frames to a deterministic binary relation similar to the binary relation corresponding to the "next" modality of linear temporal logics. It is well-known that $\mathbf{Alt}_1$ gives rise to an **NP**-complete satisfiability problem. Recently, by reducing $\mathbf{Alt}_1$-*ELU* to the problem of determining whether a graph contains an Hamiltonian path, Balbiani and Tinchev [9] have proved that $\mathbf{Alt}_1$-*ELU* is in **PSPACE**. Nevertheless, the argument they have put forward does not seem to be adaptable to the case of $\mathbf{Alt}_1$-*UWP* and **the computability of $\mathbf{Alt}_1$-*UWP* remains open.**

Luckily, in some other popular modal logics, to solve the unification problem is relatively easy. For instance, in modal logics **L** containing **KD**, *ELU* is in **NP**. This can be proved by reducing **L**-*ELU* to the problem of determining whether, given a parameter-free formula $\varphi(\bar{x})$, there are tuples $\bar{\phi}$ of atom-free formulas uniformly replacing the variables in the tuple $\bar{x}$ such that $\varphi(\bar{\phi})$ is in **KD**. Owing to the fact that every atom-free formula is **KD**-equivalent to $\bot$, or **KD**-equivalent to $\top$, one readily observes that the atom-free formulas in $\bar{\phi}$ uniformly replacing the variables in $\bar{x}$ can be restricted to the formulas $\bot$ and $\top$. This gives rise to a nondeterministic algorithm able to solve **L**-*ELU* in polynomial time.

If one considers a modal logic not containing **KD**, or one interests in *UWP* then the complexity of the unification problem may dramatically increase. Investigated as a subproblem of the non-admissibility problem, the unification problem has been proved by Rybakov [29] to be decidable in modal logics such as **K**4, **S**4, etc. Nevertheless, apart from the work of Jeřábek [22] who has studied the computability of the admissibility problem in modal logics such as **K**4, **S**4, etc, **the computability of *ELU* and *UWP* is still largely *terra incognita* in most modal logics.**

Unification types We have seen in Section 3 that in Boolean logic, every unifiable formulas can be proved to be unitary. In some modal logics, that is quite a different matter. Consider for example, the formula $\Box x \vee \Box \neg x$. It is **K**-unifiable : substitutions like $\sigma_\top(x) = \top$ and $\sigma_\bot(x) = \bot$ are unifiers of it. However, it is finitary. Let us see why. Firstly, the above-mentioned unifiers $\sigma_\top$ and $\sigma_\bot$ constitute a complete set of unifiers of it. This is a consequence of the fact that **K** satisfies the modal disjunction property : for all formulas φ, ψ, if $\Box\varphi \vee \Box\psi$ is in **K** then φ is in **K**, or ψ is in **K**. Secondly, there exists no unifier τ of $\Box x \vee \Box \neg x$ such that $\tau \preceq \sigma_\top$ and $\tau \preceq \sigma_\bot$. Indeed, suppose τ is a unifier such that $\tau \preceq \sigma_\top$ and $\tau \preceq \sigma_\bot$. Since τ is a **K**-unifier of $\Box x \vee \Box \neg x$, therefore by the modal disjunction property, $\tau(x)$ is in **K**, or $\neg\tau(x)$ is in **K**. In the former case, since $\tau \preceq \sigma_\bot$, therefore $\top \leftrightarrow \bot$ is in **K** : a contradiction. In the latter case, since $\tau \preceq \sigma_\top$, therefore $\bot \leftrightarrow \top$ is in **K** : a contradiction.

The existence of a finitary **K**-unifiable formula does not imply that all **K**-unifiable formulas are finitary. Witness, the formula $x \rightarrow \Box x$ put forward by Jeřábek [23]. It is **K**-unifiable : substitutions like $\sigma_\top(x) = \top$ and $\sigma_d(x) = \Box^{<d} x \wedge \Box^d \bot$ for each $d \geq 0$ are unifiers of it. However, it is nullary. Let us see why. Firstly, for all unifiers τ of $x \rightarrow \Box x$, $\tau(x)$ is in **K**, or $\tau(x) \rightarrow \Box^{\deg(\tau(x))}\bot$ is in **K**. This is a consequence of the fact that **K** satisfies the following variant of the rule of margins : for all formulas φ, if $\varphi \rightarrow \Box\varphi$ is in **K** then φ is in **K**, or $\varphi \rightarrow \Box^{\deg(\varphi)}\bot$ is in **K**. Secondly, as the reader can prove, for all substitutions τ, $\tau(x)$ is in **K** iff $\sigma_\top \preceq \tau$ and for all unifiers τ of $x \rightarrow \Box x$, $\tau(x) \rightarrow \Box^{\deg(\tau(x))}\bot$ is in **K** iff $\sigma_{\deg(\tau(x))} \preceq \tau$. Thirdly, there exists no minimal complete set of unifiers of $x \rightarrow \Box x$. Indeed, suppose Σ is a minimal complete set of unifiers of $x \rightarrow \Box x$. Since Σ is complete, therefore let $\tau \in \Sigma$ be such that $\tau \preceq \sigma_0$. Since Σ is a set of unifiers of $x \rightarrow \Box x$, therefore $\tau(x)$ is in **K**, or $\tau(x) \rightarrow \Box^{\deg(\tau(x))}\bot$ is in **K**. In the former case, $\sigma_\top \preceq \tau$. Since $\tau \preceq \sigma_0$, therefore $\sigma_\top \preceq \sigma_0$. Hence, $\top \leftrightarrow \bot$ is in **K** : a contradiction. In the latter case, since Σ is a set of unifiers of $x \rightarrow \Box x$, therefore $\sigma_{\deg(\tau(x))} \preceq \tau$. Since Σ is complete, therefore let $\mu \in \Sigma$ be such that $\mu \preceq \sigma_{\deg(\tau(x))+1}$. As the reader can

prove, $\sigma_{\deg(\tau(x))+1} \preceq \sigma_{\deg(\tau(x))}$. Since $\sigma_{\deg(\tau(x))} \preceq \tau$ and $\mu \preceq \sigma_{\deg(\tau(x))+1}$, therefore $\mu \preceq \tau$. Since Γ is minimal, therefore $\mu = \tau$. Since $\sigma_{\deg(\tau(x))} \preceq \tau$ and $\mu \preceq \sigma_{\deg(\tau(x))+1}$, therefore $\sigma_{\deg(\tau(x))} \preceq \sigma_{\deg(\tau(x))+1}$. Thus, $\lambda(\sigma_{\deg(\tau(x))}(x)) \leftrightarrow \sigma_{\deg(\tau(x))+1}(x)$ is in **K** for some substitution λ. Consequently, $\Box^{\deg(\tau(x))+1}\bot \rightarrow \Box^{\deg(\tau(x))}\bot$ is in **K** : a contradiction.

Because of the strong proximity between the modal logics **K**, **KD**, **KT**, **KB**, **KDB** and **KTB** in terms of axiomatization/completeness and decidability/complexity, the reader may wonder whether Jeřábek's line of reasoning can be used as it is for **KD**, **KT**, **KB**, **KDB** and **KTB**. Obviously, in this line of reasoning, the formulas $\Box^d\bot$ for each $d \geq 0$ play an important role [7]. Unfortunately, when $d \geq 1$, $\Box^d\bot$ is equivalent to $\bot$ in **KD**, **KT**, **KDB** and **KTB** and is equivalent to $\Box\bot$ in **KB**. It follows that Jeřábek's line of reasoning has to be seriously adapted if one wants to apply it to **KD**, **KT**, **KB**, **KDB** and **KTB**.

Using a parameter p, Balbiani and Gencer [6] have considered the formula $x \rightarrow (p \wedge [p]x)$ within the context of **KD**. This formula is **KD**-unifiable : substitutions like $\sigma_p(x) = p$ and $\sigma_d(x) = p \wedge [p]^{<d}x \wedge [p]^d\bot$ for each $d \geq 0$ are unifiers of it. However, it is nullary. To see why, it suffices to firstly prove that for all unifiers τ of $x \rightarrow (p \wedge [p]x)$, $\tau(x) \leftrightarrow p$ is in **KD**, or $\tau(x) \rightarrow [p]^{\deg(\tau(x))}\bot$ is in **KD**, to secondly prove that for all substitutions τ, $\tau(x) \leftrightarrow p$ is in **KD** iff $\sigma_p \preceq \tau$ and for all unifiers τ of $x \rightarrow (p \wedge [p]x)$, $\tau(x) \rightarrow [p]^{\deg(\tau(x))}\bot$ is in **KD** iff $\sigma_{\deg(\tau(x))} \preceq \tau$ and to thirdly prove that there exists no minimal complete set of unifiers of $x \rightarrow (p \wedge [p]x)$.

Using distinct parameters p, q, Balbiani [5] and Balbiani and Gencer [7] have respectively considered the formula $(x \rightarrow (p \wedge [q]y)) \wedge (y \rightarrow (q \wedge [p]x))$ within the context of **KT** and the formula $(\underline{p^0q^0} \wedge x) \rightarrow [\underline{p^0q^1}][\underline{p^1q^0}][\underline{p^0q^0}]x)$ within the context of **KB**, **KDB** and **KTB**. Following a line of reasoning similar to the ones presented above, they have proved that the former formula is unifiable and nullary in **KT** and the latter formula is unifiable and nullary in **KB**, **KDB** and **KTB**.

The nullariness character of the modal logics **KD**, **KT**, **KB**, **KDB** and **KTB** constitutes an answer to a question put forward by Dzik [14, Chapter 5]. Nevertheless, this answer only concerns *UWP,* **the types of KD, KT, KB, KDB and KTB for *ELU* remaining unknown.** Moreover, much remains to be done, seeing that, for instance, **the types of simple Church-Rosser modal logics like KG, KDG and KTG are unknown.** As well, **the type of the least modal logic containing $\Box^d\bot$ is unknown for each $d \geq 2$.**

In a sense, the existence of a nullary unifiable formula is the worst thing one can imagine about the unification type of a modal logic. Luckily, there exists modal logics where every unifiable formulas can be proved to be unitary. This is the case of the modal logic **S**5. In order to understand why, it suffices given a unifier σ of a unifiable formula φ, to consider the substitution ϵ^σ_φ such that for all $x \in \mathcal{VAR}$, if $x \in \mathtt{var}(\varphi)$ then $\epsilon^\sigma_\varphi(x) = (\Box\varphi \wedge x) \vee (\neg\Box\varphi \wedge \sigma(x))$ else $\epsilon^\sigma_\varphi(x) = x$ [8]. By induction on the formula ψ, the reader may prove that if $\mathtt{var}(\psi) \subseteq \mathtt{var}(\varphi)$ then $\Box\varphi \rightarrow (\epsilon^\sigma_\varphi(\psi) \leftrightarrow \psi)$ and $\neg\Box\varphi \rightarrow (\epsilon^\sigma_\varphi(\psi) \leftrightarrow \sigma(\psi))$ are valid. Since σ is a unifier of φ, we obtain the validity of $\Box\varphi \rightarrow \epsilon^\sigma_\varphi(\varphi)$ and $\neg\Box\varphi \rightarrow \epsilon^\sigma_\varphi(\varphi)$. Moreover, for all unifers τ of φ and for all variable x, $\tau(\epsilon^\sigma_\varphi(x)) \leftrightarrow \tau(x)$ is valid. Hence, ϵ^σ_φ is a unifier of φ and for all unifiers τ of φ, $\epsilon^\sigma_\varphi \preceq \tau$. Thus, ϵ^σ_φ is a most general unifier of φ : it constitutes by its own a complete set of unifiers of φ. Consequently, φ is unitary.

Additional comments The truth is that the above line of reasoning proving the unitariness of every **S**5-unifiable formula can be adapted to each modal logic where a modality $\forall$ ("everywhere, ...") of arity 1 playing the role of a universal modality is definable [9]. The unitariness of every unifiable formula is a remarkable property of a modal logic seeing that it guarantees the existence of a most general unifier for all its unifiable formulas. Nevertheless, it does not constitute a sufficient condition for the decidability of its unification problem and its admissibility problem. After the first results of Rybakov [29, 30] about the decidability of the unification problem and the admissibility problem in modal logics such as **K**4, **S**4, etc, it was an open question to determine whether the decidability of a modal logic ensures the decidability of its unification problem and its admissibility problem. This question has been negatively answered by Chagrov [11] who has constructed a — rather artificial — decidable modal logic with an undecidable admissibility problem. Later on, Wolter and Zakharyaschev [32] has proved that the unification problem and the admissibility problem are undecidable in **K**, **K**4 and every modal logic between **K** and **K**4 when the above-mentioned modality $\forall$ is definable.

Today, it is possible, without using the universal modality, to construct a very simple decidable modal logic with an undecidable admissibility problem. Let us see

7. As well as the fact that for all $d \geq 0$, $\Box^{d+1}\bot \rightarrow \Box^d\bot$ is not valid.

8. The reader will remark how this substitution looks like the Löwenheim substitution considered in Section 3.

9. Enriching modal logics with the universal modality gives rise to an **EXP**-complete satisfiability problem in the case of **K** and a **PSPACE**-complete satisfiability problem in the case of some simple extensions of **K**4 and **S**4 [20, 31].

how. In the case of the modal logic $\mathbf{Alt}_1 \times \mathbf{Alt}_1$ — the *product* of $\mathbf{Alt}_1$ with itself —, on top of the Boolean connectives considered in Section 3, O contains the modalities $\Box_1 \ldots$ and $\Box_2 \ldots$ of arity 1. The modalities $\Diamond_1 \ldots$ and $\Diamond_2 \ldots$ of arity 1 are defined by the abbreviations : $\Diamond_1\varphi ::= \neg\Box_1\neg\varphi$ and $\Diamond_2\varphi ::= \neg\Box_2\neg\varphi$. Formulas are interpreted in Kripke frames (W, R_1, R_2) where R_1 and R_2 are deterministic binary relations on W such that $R_1 \circ R_2 \subseteq R_2 \circ R_1$, $R_2 \circ R_1 \subseteq R_1 \circ R_2$ and $R_1^{-1} \circ R_2 \subseteq R_2 \circ R_1^{-1}$ — with $\mathcal{P}(W)$ as set of values and $\{W\}$ as set of designated values — by means of truth assignments V such that $V(\Box_1\varphi) = \{s \in W :$ for all $t \in W$, if $R_1(s,t)$ then $t \in V(\varphi)\}$ and $V(\Box_2\varphi) = \{s \in W :$ for all $t \in W$, if $R_2(s,t)$ then $t \in V(\varphi)\}$. It is well-known that in $\mathbf{Alt}_1 \times \mathbf{Alt}_1$, the satisfiability problem is NP-complete [16, Theorem 8.53]. What is maybe less known is that in $\mathbf{Alt}_1 \times \mathbf{Alt}_1$, the validity problem of rules is undecidable [10] [16, Theorem 8.54]. What is new is that in $\mathbf{Alt}_1 \times \mathbf{Alt}_1$, the admissibility problem is undecidable. This can be proved by a reduction of the *domino-tiling problem* (Π) an instance of which consists of a 7-tuple $(D, S_1, S_2, D_u, D_d, D_l, D_r)$ where D is a finite set of domino-types, S_1 and S_2 are binary relations on D and D_u, D_d, D_l and D_r are subsets of D. A tiling of such 7-tuple is a triple (a_1, a_2, f) where a_1, a_2 are positive integers and f is a function associating an element $f(i_1, i_2) \in D$ to each $(i_1, i_2) \in \{1, \ldots, a_1\} \times \{1, \ldots, a_2\}$. We shall say that (a_1, a_2, f) is *correct* if the following conditions hold for all $i_1 \in \{1, \ldots, a_1\}$ and for all $i_2 \in \{1, \ldots, a_2\}$:

— if $i_1 < a_1$ then $f(i_1, i_2) S_1 f(i_1 + 1, i_2)$,
— if $i_2 < a_2$ then $f(i_1, i_2) S_2 f(i_1, i_2 + 1)$,
— $f(a_1, i_2) \in D_u$,
— $f(1, i_2) \in D_d$,
— $f(i_1, 1) \in D_l$,
— $f(i_1, a_2) \in D_r$.

Let $(D, S_1, S_2, D_u, D_d, D_l, D_r)$ be an instance of (Π) with $\delta_1, \ldots, \delta_n$ a list of its domino-types. Let y, z and $x_1, \ldots, x_n$ be pairwise distinct variables. Let ϕ be the conjunction of the following formulas :

— $\Box_1\Box_2\neg(x_k \wedge x_l)$ where $1 \leq k, l \leq n$ and $k \neq l$,
— $\Box_1\Box_2(x_k \rightarrow \Box_1 \bigvee\{x_l : 1 \leq l \leq n$ and $(\delta_k, \delta_l) \in S_1\})$ where $1 \leq k \leq n$,
— $\Box_1\Box_2(x_k \rightarrow \Box_2 \bigvee\{x_l : 1 \leq l \leq n$ and $(\delta_k, \delta_l) \in S_2\})$ where $1 \leq k \leq n$,
— $y \rightarrow (\Box_1 y \wedge \Box_2 y)$,
— $z \rightarrow (\Box_1 z \wedge \Box_2 z)$,
— $\neg y \rightarrow \Box_1\neg y$,
— $\neg z \rightarrow \Box_2\neg z$,
— $\Box_1\Box_2((y \wedge \Box_1\bot) \rightarrow \bigvee\{x_k : 1 \leq k \leq n$ and $\delta_k \in D_u\})$,
— $\Box_2((y \wedge \neg z) \rightarrow \Box_1(z \rightarrow \bigvee\{x_k : 1 \leq k \leq n$ and $\delta_k \in D_d\}))$,
— $\Box_1((\neg y \wedge z) \rightarrow \Box_2(y \rightarrow \bigvee\{x_k : 1 \leq k \leq n$ and $\delta_k \in D_l\}))$,
— $\Box_1\Box_2((z \wedge \Box_2\bot) \rightarrow \bigvee\{x_k : 1 \leq k \leq n$ and $\delta_k \in D_r\})$.

Let ψ be the formula

— $(\Diamond_2 y \wedge \Diamond_1 z \wedge \Box_1\Box_2 \bigvee\{x_k : 1 \leq k \leq n\}) \rightarrow (y \vee z)$.

What is remarkable is that there exists a correct tiling iff the rule $\frac{\phi}{\psi}$ is not admissible. Seeing that the domino-tiling problem (Π) is undecidable [26], this proves that in $\mathbf{Alt}_1 \times \mathbf{Alt}_1$, the admissibility problem is undecidable. **As for the computability of $\mathbf{Alt}_1 \times \mathbf{Alt}_1$-*ELU* and $\mathbf{Alt}_1 \times \mathbf{Alt}_1$-*UWP*, it is open.**

5 Case of multi-agent epistemic logics

Syntax and semantics In the case of multi-agent epistemic logics, on top of the Boolean connectives considered in Section 3, O consists of the modalities $\Box_a \ldots$ ("agent a knows that ...") of arity 1 where a ranges over a countable set $\mathcal{AGT}$ of agents. We denote the associated language by $\mathcal{MAEPL}$. The modalities $\Diamond_a \ldots$ ("it is compatible with the knowledge of agent a that ...") of arity 1 are defined by the abbreviations : $\Diamond_a\varphi ::= \neg\Box_a\neg\varphi$. We shall say that a formula φ is *Boolean* if for all $a \in \mathcal{AGT}$, φ contains no occurrence of the modality $\Box_a$. For all $a \in \mathcal{AGT}$, we shall say that a formula φ is *a-restricted* if for all $b \in \mathcal{AGT}$, if $a \neq b$ then φ contains no occurrence of the modality $\Box_b$. For all $a \in \mathcal{AGT}$, we shall say that a formula φ is *a-monic* if for all $b \in \mathcal{AGT}$ and for all formulas ψ, if $\Box_b\psi$ is a subformula of φ then $a = b$, or ψ is variable-free. Obviously, for all $a \in \mathcal{AGT}$, every a-restricted formula is a-monic.

Formulas of $\mathcal{MAEPL}$ are interpreted in multi-agent Kripke frames $(W, \{R_a : a \in \mathcal{AGT}\})$ by means of truth assignments V such that $V(\Box_a\varphi) = \{s \in W :$ for all $t \in W$, if $R_a(s,t)$ then $t \in V(\varphi)\}$. In this section, we will not reopen the debate about the reasonableness of the notion of knowledge corresponding to such-and-such class of multi-agent Kripke frames. We will rather content ourselves with analyzing the effects of the considered classes of multi-agent Kripke frames on the computability of unification and the unification types. In most cases, it is assumed that for all $a \in \mathcal{AGT}$, the accessibility relation R_a is an equivalence relation [11]. The associated modal logic is denoted $\mathbf{S5}_{\mathcal{AGT}}$. In few cases, it is moreover assumed that for all $a, b \in \mathcal{AGT}$, the accessibility relations R_a and R_b are in local agreement, i.e. for all $s \in W$, $R_a(s) \subseteq R_b(s)$, or

10. The rule $\frac{\varphi_1, \ldots, \varphi_n}{\chi}$ is *valid* if the validity of $\varphi_1, \ldots, \varphi_n$ implies the validity of χ.

11. In that case, for all $a \in \mathcal{AGT}$ and for all $s \in W$, we will denote by $R_a(s)$ the equivalence class modulo R_a with s as its representative.

$R_b(s) \subseteq R_a(s)$. The associated modal logic is denoted $\mathbf{S5}^{la}_{\mathcal{AGT}}$.

Computability of unification $\mathbf{S5}_{\mathcal{AGT}}$-$ELU$ and $\mathbf{S5}^{la}_{\mathcal{AGT}}$-$ELU$ are **NP**-complete, seeing that the corresponding classes of multi-agent Kripke frames are such that for all $a \in \mathcal{AGT}$, R_a is serial [12]. **Regarding $\mathbf{S5}_{\mathcal{AGT}}$-$UWP$ and $\mathbf{S5}^{la}_{\mathcal{AGT}}$-$UWP$, their computability is a mystery.**

Unification types In this paragraph, we consider the class of multi-agent Kripke frames where for all $a \in \mathcal{AGT}$, R_a is an equivalence relation. If $\mathcal{AGT}$ is finite then the modality $\forall$ ("every agent knows that ...") of arity 1 defined by the abbreviation $\forall\varphi ::= \bigwedge\{\Box_a\varphi : a \in \mathcal{AGT}\}$ plays the role of a universal modality when we restrict the discussion to the class of multi-agent Kripke frames where for all $a, b \in \mathcal{AGT}$, R_a and R_b are in local agreement. In this case, following the argument used in Section 4 for modal logics where a modality playing the role of a universal modality is definable, every unifiable formulas can be proved to be unitary.

Otherwise, that is to say if $\mathcal{AGT}$ is not finite, or when we do not restrict the discussion to the class of multi-agent Kripke frames where for all $a, b \in \mathcal{AGT}$, R_a and R_b are in local agreement, the situation is worse. To see why, use pairwise distinct parameters p, q, r and take the formula $(x \to \Box x) \wedge (\neg x \to \blacksquare\neg x)$ where for all formulas φ, we write $\Box\varphi$ to mean $\underline{[p^0q^0r^0]}_1\underline{[p^0q^0r^1]}_2\underline{[p^0q^1r^0]}_1\underline{[p^0q^1r^1]}_2\underline{[p^1q^0r^0]}_1\underline{[p^1q^0r^1]}_2\varphi$ and write $\blacksquare\varphi$ to mean $\underline{p^1q^0r^1} \to \underline{[p^1q^0r^0]}_2\underline{[p^0q^1r^1]}_1$ $\underline{[p^0q^1r^0]}_2\underline{[p^0q^0r^1]}_1\underline{[p^0q^0r^0]}_2\Box_1\varphi$. It is $\mathbf{S5}_{\mathcal{AGT}}$-unifiable : substitutions like $\sigma_d(x) = \Box^{<d}x \wedge \Box^d\bot$ for each $d \geq 0$ and $\tau_d(x) = \neg(\blacksquare^{<d}\neg x \wedge \blacksquare^d\bot)$ for each $d \geq 0$ are unifiers of it. However, following a line of reasoning similar to the ones developed above for **K** and **KD**, the reader may prove that it is nullary.

Additional comments The existence of a nullary $\mathbf{S5}_{\mathcal{AGT}}$-unifiable formula does not imply that all $\mathbf{S5}_{\mathcal{AGT}}$-unifiable formulas are nullary. Witness, the monic formulas. Indeed, let $a \in \mathcal{AGT}$ and φ be an a-monic formula. Suppose it is $\mathbf{S5}_{\mathcal{AGT}}$-unifiable. Since φ is a-monic, therefore for all $b \in \mathcal{AGT}$ and for all formulas ψ, if $\Box_b\psi$ is a subformula of φ then $a = b$, or ψ is variable-free. Given an $\mathbf{S5}_{\mathcal{AGT}}$-unifier σ of a φ, let ϵ^σ_φ be the substitution such that for all $x \in \mathcal{VAR}$, if $x \in \mathtt{var}(\varphi)$ then $\epsilon^\sigma_\varphi(x) = (\Box_a\varphi \wedge x) \vee (\neg\Box_a\varphi \wedge \sigma(x))$ else $\epsilon^\sigma_\varphi(x) = x$. By induction on the a-monic formula ψ, the reader may prove that if $\mathtt{var}(\psi) \subseteq \mathtt{var}(\varphi)$ then $\Box_a\varphi \to (\epsilon^\sigma_\varphi(\psi) \leftrightarrow \psi)$ and $\neg\Box_a\varphi \to (\epsilon^\sigma_\varphi(\psi) \leftrightarrow \sigma(\psi))$ are valid. Since σ is a unifier of φ, we obtain the validity of $\Box_a\varphi \to \epsilon^\sigma_\varphi(\varphi)$ and $\neg\Box_a\varphi \to \epsilon^\sigma_\varphi(\varphi)$. Moreover, for all unifers τ of φ and for all variable x, $\tau(\epsilon^\sigma_\varphi(x)) \leftrightarrow \tau(x)$ is valid. Hence, ϵ^σ_φ is a unifier of φ and for all unifiers τ of φ, $\epsilon^\sigma_\varphi \preceq \tau$. Thus, ϵ^σ_φ is a most general unifier of φ : it constitutes by its own a complete set of unifiers of φ. Consequently, φ is unitary.

In next section, we will apply this line of reasoning to the monic $\mathcal{MAEPL}$-formulas associated to some simple epistemic planning problems.

6 Application to epistemic planning

On top of the Boolean connectives $\bot$, $\neg$ and $\vee$ considered in Section 3 and the modalities $\Box_a$ considered in Section 5, let us define the modalities $[[\varphi]]\ldots$ of arity 1 for each $\varphi \in \mathcal{MAEPL}$ by the inductive abbreviations $[[\varphi]]p ::= (\varphi \to p)$, $[[\varphi]]\bot ::= \neg\varphi$, $[[\varphi]]\neg\psi ::= (\varphi \to \neg[[\varphi]]\psi)$, $[[\varphi]](\psi \vee \chi) ::= ([[\varphi]]\psi \vee [[\varphi]]\chi)$ and $[[\varphi]]\Box_a\psi ::= (\varphi \to \Box_a[[\varphi]]\psi)$. Let us define the modalities $\langle\langle\varphi\rangle\rangle\ldots$ of arity 1 for each $\varphi \in \mathcal{MAEPL}$ by the abbreviations : $\langle\langle\varphi\rangle\rangle\psi ::= \neg[[\varphi]]\neg\psi$. Within the setting of Public Announcement Logic [12, Chapter 4], for all $\varphi \in \mathcal{MAEPL}$, the modalities $[[\varphi]]\ldots$ ("if φ holds then after it is announced, ... holds") and $\langle\langle\varphi\rangle\rangle\ldots$ ("φ holds and after it is announced, ... holds") have been used to formalize the notion of the public announcement of φ which is the atomic action performed by an outside observer, perceived by all agents and consisting of publicly announcing φ. By means of these modalities together with the modality of common knowledge, one can faithfully represent all of the reasoning in examples such as the muddy children puzzle.

The reader will remark that the formulas written by using the modalities $[[\varphi]]\ldots$ and $\langle\langle\varphi\rangle\rangle\ldots$ where φ ranges over the set of all $\mathcal{MAEPL}$-formulas are exponentially more succinct than the $\mathcal{MAEPL}$-formulas they are the abbreviations of. Nevertheless, as proved by Lutz [25], the membership in $\mathbf{S5}_{\mathcal{AGT}}$ of formulas written by using these modalities has the same complexity as the membership in $\mathbf{S5}_{\mathcal{AGT}}$ of $\mathcal{MAEPL}$-formulas : **PSPACE**.

Suppose the formula $\varphi(\bar{p})$ describes a given initial situation in terms of the knowledge of $\mathcal{AGT}$-agents about the list of parameter facts $\bar{p} = (p_1, \ldots, p_m)$ and the formula $\langle\langle\chi(\bar{p}\bar{x})\rangle\rangle\psi(\bar{p})$ represents the knowledge of these agents about $\bar{p}$ in a desirable final situation (ψ-part) after an executable public announcement concerning $\bar{p}$ and the list of variable facts $\bar{x} = (x_1, \ldots, x_n)$ (χ-part) has been performed. It may happen that $\varphi(\bar{p}) \to \langle\langle\chi(\bar{p}\bar{x})\rangle\rangle\psi(\bar{p})$ is not $\mathbf{S5}_{\mathcal{AGT}}$-valid. Hence, we may ask whether there are tuples $\bar{\phi} = (\phi_1, \ldots, \phi_n)$ of formulas such that $\varphi(\bar{p}) \to$

12. The line of reasoning we have developed in Section 4 to prove that ELU is in **NP** for each modal logic containing **KD** can be applied in the multi-agent setting as well.

$\langle\langle\chi(\bar{p}\bar{\phi})\rangle\rangle\psi(\bar{p})$ is valid. Moreover, we may be interested to obtain the most general tuple $\bar{\phi}$ of formulas such that $\varphi(\bar{p}) \rightarrow \langle\langle\chi(\bar{p}\bar{\phi})\rangle\rangle\psi(\bar{p})$ is valid. Introduced in this way, the formula $\varphi(\bar{p}) \rightarrow \langle\langle\chi(\bar{p}\bar{x})\rangle\rangle\psi(\bar{p})$ constitutes a special instance of what is now called an epistemic planning problem. See [10] for a general introduction.

Consider for example the formula $\psi \rightarrow \langle\langle\Box_a x\rangle\rangle\Box_b\chi$ where ψ is a variable-free $\mathcal{MAEPL}$-formula, x is a variable, a and b are distinct agents and χ is a Boolean variable-free $\mathcal{MAEPL}$-formula. Considered as an epistemic planning problem, to solve this formula means to ask whether there exists an $\mathcal{MAEPL}$-formula ϕ such that $\psi \rightarrow \langle\langle\Box_a\phi\rangle\rangle\Box_b\chi$ — "in every situation where ψ holds, a knows that ϕ holds and after it is announced that a knows that ϕ holds, b knows that χ holds" — is $\mathbf{S5}_{\mathcal{AGT}}$-valid. Since χ is Boolean, therefore taking into account the definition of the modalities $[[\varphi]]\ldots$ and $\langle\langle\varphi\rangle\rangle\ldots$ for each $\varphi \in \mathcal{MAEPL}$, this is equivalent to solve the formula $\psi \rightarrow (\Box_a x \wedge \Box_b\chi')$ where χ' is the $\mathcal{MAEPL}$-formula $\Box_a x \rightarrow \chi$. Since $\Box_b$ corresponds in Kripke frames to an equivalence relation, therefore this is equivalent to solve the conjunction κ of the formulas $\psi \rightarrow \Box_a x$ and $\Diamond_b\psi \rightarrow \chi'$. Since ψ and χ are variable-free, therefore κ is a-monic. Moreover, as the reader can prove, κ is $\mathbf{S5}_{\mathcal{AGT}}$-unifiable iff $\Diamond_a\psi \wedge \Diamond_b\psi \rightarrow \chi$ is $\mathbf{S5}_{\mathcal{AGT}}$-valid. In that case, any substitution σ such that $\sigma(x) = \Diamond_a\psi$ is a unifier of it and by applying the line of reasoning developed in the paragraph "Additional comments" of Section 5, one can construct a most general unifier of it and of the given formula $\psi \rightarrow \langle\langle\Box_a x\rangle\rangle\Box_b\chi$ too.

More generally, one may be interested by solving formulas like $\psi \rightarrow \langle\langle\Box_a x \wedge \Diamond_a y_1 \wedge \ldots \wedge \Diamond_a y_m\rangle\rangle(\Box_{b_1}\chi_1 \wedge \ldots \wedge \Box_{b_n}\chi_n)$ where ψ is a variable-free $\mathcal{MAEPL}$-formula, $m, n \geq 0$, x and $y_1, \ldots, y_m$ are pairwise distinct variables, a and $b_1, \ldots, b_n$ are pairwise distinct agents and $\chi_1, \ldots, \chi_n$ are a-restricted variable-free $\mathcal{MAEPL}$-formulas. Considered as an epistemic planning problem, to solve this formula means to ask whether there exists $\mathcal{MAEPL}$-formulas ϕ and $\theta_1, \ldots, \theta_m$ such that $\psi \rightarrow \langle\langle\Box_a\phi \wedge \Diamond_a\theta_1 \wedge \ldots \wedge \Diamond_a\theta_m\rangle\rangle(\Box_{b_1}\chi_1 \wedge \ldots \wedge \Box_{b_n}\chi_n)$ — "in every situation where ψ holds, a knows that ϕ holds, it is compatible with the knowledge of a that θ_i holds for each $i \in \{1, \ldots, m\}$ and after it is announced that a knows that ϕ holds and it is compatible with the knowledge of a that θ_i holds for each $i \in \{1, \ldots, m\}$, b_j knows that χ_j holds for each $j \in \{1, \ldots, n\}$" — is $\mathbf{S5}_{\mathcal{AGT}}$-valid. Since $\chi_1, \ldots, \chi_n$ are a-restricted, therefore taking into account the definition of the modalities $[[\varphi]]\ldots$ and $\langle\langle\varphi\rangle\rangle\ldots$ for each $\varphi \in \mathcal{MAEPL}$, this is equivalent to solve the formula $\psi \rightarrow (\Box_a x \wedge \Diamond_a y_1 \wedge \ldots \wedge \Diamond_a y_m \wedge \Box_{b_1}\chi'_1 \wedge \ldots \wedge \Box_{b_n}\chi'_n)$ where χ'_j is the $\mathcal{MAEPL}$-formula $(\Box_a x \wedge \Diamond_a y_1 \wedge \ldots \wedge \Diamond_a y_m) \rightarrow \chi_j$ for each $j \in \{1, \ldots, n\}$. Since $\Box_{b_1}, \ldots, \Box_{b_n}$ correspond in Kripke frames to equivalence relations, therefore this is equivalent to solve the conjunction κ of the formulas $\psi \rightarrow (\Box_a x \wedge \Diamond_a y_1 \wedge \ldots \wedge \Diamond_a y_m)$ and $\Diamond_{b_j}\psi \rightarrow \chi'_j$ where j ranges over $\{1, \ldots, n\}$. Since ψ and $\chi_1, \ldots, \chi_n$ are variable-free, therefore κ is a-monic. Moreover, as the reader can prove, κ is $\mathbf{S5}_{\mathcal{AGT}}$-unifiable iff $\Diamond_a\psi \wedge \Diamond_{b_j}\psi \rightarrow \chi_j$ is $\mathbf{S5}_{\mathcal{AGT}}$-valid for each $j \in \{1, \ldots, n\}$. In that case, any substitution σ such that $\sigma(x) = \Diamond_a\psi$ and $\sigma(y_j) = \psi$ for each $j \in \{1, \ldots, n\}$ is a unifier of it and by applying the line of reasoning developed in the paragraph "Additional comments" of Section 5, one can construct a most general unifier of it and of the given formula $\psi \rightarrow \langle\langle\Box_a x \wedge \Diamond_a y_1 \wedge \ldots \wedge \Diamond_a y_m\rangle\rangle(\Box_{b_1}\chi_1 \wedge \ldots \wedge \Box_{b_n}\chi_n)$ too.

7 Conclusion

The modal logics like the ones considered in this paper (**K**, **KD**, etc) have now limited mathematical interest for what concerns axiomatization/completeness and decidability/complexity. Nevertheless, with respect to some predetermined propositional logic **L**, considering the question of the determination of its unification type and the question of the computability of the unification problem it gives rise to is justified from the following perspectives : methods for deciding the **L**-unifiability of formulas can be used to understand what is the overlap between the properties formulas correspond to in **L** [2] ; in case **L** is a description logic, unification algorithms can be used to detect redundancies in **L**-based systems [3, 4] ; methods for deciding the **L**-unifiability of formulas can be used to improve the efficiency of theorem provers devoted to solve the membership problem in **L**. One readily observes that, while attacking the above-mentioned problems, little, if anything, from the standard tools in modal logics (canonical models, filtrations, etc) is helpful. In order to successfully solve them, new techniques in modal logics must be developed and much remains to be done. The study of unification in modal logics has still many secrets to reveal.

Acknowledgements

The preparation of this paper has been supported by *French Minister of Europe and Foreign Affairs, French Minister of Higher Education, Research and Innovation* and *Iranian Minister of Research, Science and Technology* (Project *40903VF*), *Bulgarian Science Fund* (Project *DN02/15/19.12.2016*) and *Université Paul Sabatier* (Programme *Professeurs invités 2018*). We also make a point of thanking the referees for their feedback.

Références

[1] Baader, F., 'On the complexity of Boolean unification', *Information Processing Letters* **67** (1998) 215–220.

[2] Baader, F., and S. Ghilardi, 'Unification in modal and description logics', *Logic Journal of the IGPL* **19** (2011) 705–730.

[3] Baader, F., and B. Morawska, 'Unification in the description logic $\mathcal{EL}$', In : *Rewriting Techniques and Applications,* Springer (2009) 350–364.

[4] Baader, F., and P. Narendran, 'Unification of concept terms in description logics', *Journal of Symbolic Computation* **31** (2001) 277–305.

[5] Balbiani, P., 'Remarks about the unification type of some non-symmetric non-transitive modal logics', *Logic Journal of the IGPL* (to appear).

[6] Balbiani, P., and Ç. Gencer, 'KD is nullary', *Journal of Applied Non-Classical Logics* **27** (2017) 196–205.

[7] Balbiani, P., and Ç. Gencer, 'About the unification type of simple symmetric modal logics', (submitted).

[8] Balbiani, P., and M. Mojtahedi, 'Unification in the implication fragment of Boolean Logic', (submitted).

[9] Balbiani, P., and T. Tinchev, 'Unification in modal logic Alt_1', In : *Advances in Modal Logic,* College Publications (2016) 117–134.

[10] Bolander, T., and M. Andersen, 'Epistemic planning for single- and multi-agent systems', *Journal of Applied Non-Classical Logics* **21** (2011) 9–34.

[11] Chagrov, A., 'Decidable modal logic with undecidable admissibility problem', *Algebra and Logic* **31** (1992) 53–61.

[12] Van Ditmarsch, H., W. van der Hoek, and B. Kooi, *Dynamic Epistemic Logic,* Springer (2008).

[13] Dunn, M., and G. Hardegree, *Algebraic Methods in Philosophical Logic,* Oxford University Press (2001).

[14] Dzik, W., *Unification Types in Logic,* Wydawnicto Uniwersytetu Slaskiego (2007).

[15] Fagin, R., J. Halpern, and M. Vardi, 'What is an inference rule ?', *Journal of Symbolic Logic* **57** (1992) 1018–1045.

[16] Gabbay, D., A. Kurucz, F. Wolter, and M. Zakharyaschev, *Many-Dimensional Modal Logics : Theory and Applications,* Elsevier Science (2003).

[17] Gencer, Ç., and D. de Jongh, 'Unifiability in extensions of $K4$', *Logic Journal of the IGPL* **17** (2009) 159–172.

[18] Ghilardi, S., 'Unification in intuitionistic logic', *Journal of Symbolic Logic* **64** (1999) 859–880.

[19] Ghilardi, S., 'Best solving modal equations', *Annals of Pure and Applied Logic* **102** (2000) 183–198.

[20] Hemaspaandra, E., 'The price of universality', *Notre Dame Journal of Formal Logic* **37** (1996) 174–203.

[21] Iemhoff, R., 'On the admissible rules of intuitionistic propositional logic', *Journal of Symbolic Computation* **66** (2001) 281–294.

[22] Jeřábek, E., 'Complexity of admissible rules', *Archive for Mathematical Logic* **46** (2007) 73–92.

[23] Jeřábek, E., 'Blending margins : the modal logic **K** has nullary unification type', *Journal of Logic and Computation* **25** (2015) 1231–1240.

[24] Löwenheim, L., 'Über das Auflösungsproblem im logischen Klassenkalkül', *Sitzungsberichte der Berliner mathematischen Gesellschaft* **7** (1908) 89–94.

[25] Lutz, C., 'Complexity and succinctness of Public Announcement Logic', In : *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems,* ACM (2006) 137–143.

[26] Lutz, C., D. Walther, and F. Wolter, 'Conservative extensions in expressive description logics', In : *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence,* AAAI Press (2007) 453–458.

[27] Porte, J., 'The deducibilities of **S**5', *Journal of Philosophical Logic* **10** (1981) 409–422.

[28] Rudeanu, S., *Boolean Functions and Equations,* Elsevier (1974).

[29] Rybakov, V., 'A criterion for admissibility of rules in the model system $S4$ and the intuitionistic logic. *Algebra and Logic* **23** (1984) 369–384.

[30] Rybakov, V., *Admissibility of Logical Inference Rules,* Elsevier (1997).

[31] Shapirovsky, I., 'Downward-directed transitive frames with universal relations', In : *Advances in Modal Logic,* College Publications (2006) 413–428.

[32] Wolter, F., and M. Zakharyaschev, 'Undecidability of the unification and admissibility problems for modal and description logics', *ACM Transactions on Computational Logic* **9** (2008) 25 :1–25 :20.

Actes JIAF 2019

Elect : Une méthode de gestion des incohérences dans des ontologies légères partiellement préordonnées *

Sihem Belabbes[1] **Salem Benferhat**[1] **Jan Chomicki**[2]

[1] CRIL, Université d'Artois, CNRS-UMR8188, Lens, France
[2] SUNY at Buffalo, Buffalo, NY, USA
{belabbes,benferhat}@cril.fr chomicki@buffalo.edu

Résumé

Nous nous intéressons au problème de la gestion des incohérences dans des ontologies légères. Nous supposons que les bases de connaissances terminologiques (TBox) sont spécifiées dans les logiques de description DL-Lite. Nous supposons aussi que les bases assertionnelles (ABox) sont partiellement préordonnées et peuvent être incohérentes avec la base terminologique TBox. Une contribution importante de cet article est la proposition d'une méthode efficace et saine, appelée Elect, pour restaurer la cohérence de la base ABox par rapport à la base TBox. Notre méthode permet de retrouver la sémantique dite IAR (*Intersection ABox Repair*) lorsque la base ABox est non-ordonnée. Elle permet également de retrouver la sémantique dite non-contestée lorsque la base ABox est totalement préordonnée. La justification sémantique de la méthode Elect est basée d'abord sur le fait qu'une base ABox partiellement préordonnée est interprétée comme une famille de bases ABox totalement préordonnées. Ensuite, la méthode Elect applique l'inférence non-contestée à chacune de ces ABox totalement préordonnées. Dans la deuxième partie de l'article, nous introduisons le concept d'assertions, dites élues, qui permet de donner une caractérisation équivalente de la méthode Elect sans avoir à générer explicitement toutes les bases ABox totalement préordonnées. Nous montrons alors que le calcul des assertions élues se fait en temps polynomial.

Abstract

We focus on the problem of handling inconsistency in lightweight ontologies. We assume that terminological knowledge bases (TBoxes) are specified in DL-Lite and that assertional facts (ABoxes) are partially preordered and may be inconsistent with respect to TBoxes. One of the main contributions of this paper is the provision of an efficient and safe method, called Elect, to restore consistency of the ABox with respect to the TBox. In the case where the ABox is flat (no priorities are associated with the assertions) or totally preordered, our method collapses with the well-known IAR semantics and non-defeated semantics, respectively. The semantic justification of Elect is obtained by first viewing a partially preordered ABox as a family of totally preordered ABoxes, and then applying non-defeated inference to each of the totally preordered ABoxes. We introduce the concept of elected assertions which allows us to provide an equivalent characterisation of Elect without explicitly generating all totally preordered ABoxes. Finally we show that the computation of Elect is done in polynomial time.

1 Introduction

Dans cet article, nous nous intéressons à la gestion des incohérences dans des ontologies spécifiées en DL-Lite [12], une famille de fragments légers des Logiques de Description (DL) ayant de bonnes propriétés calculatoires. Dans le contexte des Logiques de Description, une base de connaissances (KB) comporte deux composantes : une base TBox qui contient la connaissance terminologique et une base ABox qui est une base assertionnelle. Il est raisonnable et d'usage de supposer que le contenu de la base TBox est correct et sans conflits, donc les éléments de la base TBox ne sont pas remis en cause en présence de conflits. Cependant, les assertions de la base ABox peuvent être discutables lorsque la base de connaissances est incohérente. Plusieurs stratégies ont été proposées pour permettre de raisonner avec des bases de connaissances incohérentes [8, 13, 23] (voir également [2] pour un état de l'art). Cela revient souvent à calculer des réparations pour la base ABox, où une réparation est définie comme un sous-ensemble maximal de la base ABox qui est cohérent avec la base TBox.

L'approche dite *ABox Repair* (AR) [17] revient à réparer la base ABox de manière minimale (en termes d'inclusion

*Cet article est paru en anglais dans les actes de *15th International Conference on Logic Programming and Nonmonotonic Reasoning* (LPNMR), Philadelphie, USA, 3–7 Juin 2019, pages 210–223, Springer.

ensembliste) sans modifier la base TBox. La réponse à une requête est basée sur les réponses obtenues de chacune des réparations. L'approche AR est souvent considérée comme un moyen fiable de gérer les conflits. Cependant, le calcul des réponses aux requêtes avec la méthode AR est coûteux, même pour une logique d'ontologie légère telle que DL-Lite. La méthode connue sous le nom *Intersection ABox Repair* (IAR) [17] est plus prudente. Elle interroge une seule sous-base cohérente de la base ABox obtenue à partir de l'intersection de toutes les réparations. L'avantage de la réparation IAR est qu'elle peut être calculée en temps polynomial.

La notion de réparation non-contestée (en anglais *non-defeated*) d'une base ABox incohérente et totalement préordonnée a été proposée dans [4] dans le cadre des logiques DL-Lite et dans [6] dans le cadre de la logique propositionnelle. L'approche suppose que la base ABox est stratifiée du fait de l'application d'un préordre total sur les assertions. Intuitivement, la réparation non-contestée est basée sur l'application itérative de la sémantique IAR à des ensembles formés par un nombre de strates incrémenté à chaque itération. Ce calcul est également effectué en temps polynomial [21] pour DL-Lite.

Dans cet article, nous nous intéressons à la recherche d'un moyen efficace pour calculer les réparations d'une base de connaissances incohérente spécifiée en DL-Lite, et dans laquelle la relation de priorité sur les assertions est un préordre partiel. Pour ce faire, nous proposons une méthode efficace et saine, appelée Elect, pour restaurer la cohérence de la base ABox par rapport à la base TBox. Nous montrons que la méthode Elect généralise à la fois la sémantique IAR lorsque les bases assertionnelles sont non-ordonnées et aussi la sémantique "non-contestée" pour des bases ABox totalement préordonnées.

La justification sémantique de la méthode Elect consiste d'abord à interpréter un préordre partiel associé à une base ABox comme une famille de préordres totaux, puis d'appliquer dans un deuxième temps la sémantique "non-contestée" à chacune des bases ABox totalement préordonnées, et finalement de calculer leur intersection pour produire une réparation unique. La méthode Elect est saine puisqu'elle ne fait pas de choix arbitraire entre les préordres totaux associés à un préordre partiel. De ce fait, tous les préordres totaux sont pris en compte pour définir la méthode Elect. Nous proposons le concept d'assertion élue comme étant une assertion strictement préférée à toute autre assertion avec laquelle elle est en conflit. Cela nous permet de proposer une caractérisation équivalente de la méthode Elect, et donc d'obtenir la réparation sans avoir à calculer explicitement l'ensemble des préordores totaux. Enfin, nous montrons que le calcul de la réparation avec la méthode Elect se fait en temps polynomial. Ainsi, Elect préserve la calculabilité de la sémantique IAR et de la sémantique "non-contestée" pour des bases ABox partiellement préordonnées.

Cet article est structuré comme suit. La section 2 contient des rappels sur la logique DL-Lite. La section 3 présente la sémantique IAR pour des bases ABox non-ordonnées. La section 4 aborde la réparation non-contestée pour des bases ABox totalement préordonnées. La section 5 concerne les bases ABox partiellement préordonnées. Nous introduisons notre méthode appelée Elect et en proposons une caractérisation. La section 6 contient une discussion sur trois extensions possibles de la méthode Elect, avant de conclure l'article dans la section 7.

2 La Logique de Description DL-Lite

Les Logiques de Description (DL) [1] sont une famille de formalismes de représentation des connaissances ayant de nombreuses applications, notamment pour la formalisation d'ontologies. Les fragments légers de DL tels que DL-Lite [12] sont particulièrement intéressants car ils offrent un bon compromis entre le pouvoir expressif et la complexité calculatoire. En effet, répondre à des requêtes en DL-Lite s'effectue de manière efficace. Il existe quelques variantes de DL-Lite, telles que DL-Lite$_R$ que nous utilisons dans le présent article.

Le langage DL-Lite$_R$ est construit sur un ensemble fini de *noms de concepts* C, un ensemble fini de *noms de rôles* R et un ensemble fini de *noms d'individus* I, tels que C, R et I sont mutuellement disjoints. Le langage DL-Lite$_R$ est défini selon les règles suivantes :

$R \longrightarrow P \mid P^{-} \qquad E \longrightarrow R \mid \neg R$

$B \longrightarrow A \mid \exists R \qquad C \longrightarrow B \mid \neg B$

Dans ces règles, $A \in \mathsf{C}$ dénote un nom de concept, $P \in \mathsf{R}$ dénote un nom de rôle, et P^{-} est la relation *inverse* associée à P. De plus, R dénote un *rôle de base*, alors que E représente un *rôle complexe*. Par ailleurs, B dénote un *concept de base* et C est un *concept complexe*.

Exemple 1 *Considérons l'exemple suivant :*
– $\mathsf{C} = \{Danse, Mdanse, Tdanse, DanseA, DanseS, Acc\}$, *représentant respectivement les noms de concept : danse, danse moderne, danse traditionnelle, danse avec des accessoires, danse sans accessoires ainsi que les accessoires utilisés dans certaines danses.*
– $\mathsf{R} = \{UtA\}$, *représentant les accessoires utilisés dans certaines danses et qui peuvent être des fleurs (*fl *en abrégé), un chapeau (*ch *en abrégé), ou des écharpes (*ec *en abrégé).*
– $\mathsf{I} = \{d_1, d_2, d_3, d_4, d_5\} \cup \{fl, ch, ec\}$, *où chaque d_i, pour $i = 1, \ldots, 5$, représente une danse et le reste représente des accessoires.*
Des exemples de concepts complexes sont : $\neg DanseA$ *et* $\neg \exists UtA$. □

Un *axiome d'inclusion* sur des concepts (respectivement des rôles) est un énoncé de la forme $B \sqsubseteq C$ (respectivement, $R \sqsubseteq E$). Une inclusion de concept ayant le symbole ¬ du

côté droit de l'inclusion est appelée *axiome négatif d'inclusion*, sinon elle est appelée *axiome positif d'inclusion*. Des exemples d'axiomes d'inclusion sur des concepts sont : $DanseS \sqsubseteq \neg DanseA$ et $\exists UtA^{-} \sqsubseteq Acc$.

Une base *TBox* $\mathcal{T}$ en DL-Lite$_R$ est un ensemble fini d'axiomes d'inclusion (positifs et négatifs). Une *assertion* est un énoncé de la forme $A(a)$ ou $P(a, b)$, avec $a, b \in \mathsf{I}$. Des exemples d'assertions sont $Mdanse(d_1)$ et $UtA(d_3, ch)$. Une base *ABox* $\mathcal{A}$ en DL-Lite$_R$ est un ensemble fini d'assertions. Pour des bases $\mathcal{T}$ et $\mathcal{A}$, nous notons une *base de connaissances* (KB) par $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$.

Dans cet article, nous utilisons l'exemple suivant.

Exemple 1 *(suite) Soit la base TBox suivante :*

$$\mathcal{T} = \left\{ \begin{array}{l} 1.\ Mdanse \sqsubseteq Danse, \\ 2.\ Tdanse \sqsubseteq Danse, \\ 3.\ Tdanse \sqsubseteq DanseA, \\ 4.\ Mdanse \sqsubseteq DanseS, \\ 5.\ DanseS \sqsubseteq \neg DanseA, \\ 6.\ DanseS \sqsubseteq \neg \exists UtA, \\ 7.\ \exists UtA^{-} \sqsubseteq Acc, \\ 8.\ \exists UtA \sqsubseteq DanseA \end{array} \right\}$$

Les deux premiers axiomes indiquent seulement que les danses modernes et les danses traditionnelles sont des danses. L'axiome 3 signifie que les danses traditionnelles sont des danses qui utilisent des accessoires. L'axiome 4 énonce que les danses modernes n'utilisent pas d'accessoires. L'axiome 5 indique que l'ensemble de danses modernes et l'ensemble de danses traditionnelles sont disjoints. L'axiome 6 énonce qu'une danse moderne n'utilise pas d'accessoires. L'axiome 7 indique que les éléments utilisés par les danses, donnés par le nom de rôle "UtA", doivent appartenir à l'ensemble des éléments spécifiés par le nom de concept "Acc". L'axiome 8 spécifie que tout ce qui utilise des accessoires doit être une danse avec accessoires.

Soit la base ABox donnée par les assertions suivantes :

$$\mathcal{A} = \left\{ \begin{array}{l} Mdanse(d_1), Mdanse(d_2), Tdanse(d_2), \\ Tdanse(d_3), Tdanse(d_4), DanseA(d_3), \\ DanseA(d_5), DanseS(d_5), UtA(d_2, fl), \\ UtA(d_3, ch), UtA(d_4, ec) \end{array} \right\}$$

□

Une base de connaissances $\mathcal{K}$ est dite *cohérente* si elle admet au moins un modèle, sinon elle est *incohérente*. Une base TBox $\mathcal{T}$ est *incohérente* s'il existe un nom de concept $A \in \mathsf{C}$ tel que A est vide dans chaque modèle de $\mathcal{T}$, sinon elle est *cohérente*. Notons que la base KB donnée par $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ dans l'Exemple 1 est incohérente.

Pour de plus amples détails sur la famille DL-Lite de Logiques de Description, nous invitons le lecteur à consulter les travaux de Calvanese et al. [12]. Dans la suite de cet article, nous notons DL-Lite$_R$ simplement par DL-Lite.

3 Sémantique IAR pour des Bases Assertionnelles Non-Ordonnées

Dans cette section, nous considérons une base KB spécifiée en DL-Lite $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ qui peut être incohérente. Nous supposons que la base TBox $\mathcal{T}$ est cohérente et fiable (c-à-d. validée par les concepteurs de l'ontologie). De ce fait, les éléments de $\mathcal{T}$ ne sont pas remis en cause en présence de conflits, contrairement aux assertions de $\mathcal{A}$ qui peuvent être discutables. De plus, nous supposons que la base ABox $\mathcal{A}$ est non-ordonnée, donc toutes les assertions ont le même degré de priorité. Une manière standard de gérer l'incohérence consiste à d'abord calculer l'ensemble des sous-ensembles maximaux cohérents de $\mathcal{A}$, appelés réparations, puis de les utiliser pour l'inférence (c'est-à-dire pour répondre à des requêtes). Formellement, une réparation est définie comme suit [17] :

Définition 1 *Soit $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base KB non-ordonnée et incohérente. Une sous-base $\mathcal{R} \subseteq \mathcal{A}$ est une réparation si $\langle \mathcal{T}, \mathcal{R} \rangle$ est cohérente, et $\forall \mathcal{R}' \subseteq \mathcal{A}$ t.q. $\mathcal{R} \subsetneq \mathcal{R}'$, $\langle \mathcal{T}, \mathcal{R}' \rangle$ est incohérente. De plus, si $\langle \mathcal{T}, \mathcal{A} \rangle$ est cohérente, alors il existe une seule réparation $\mathcal{R} = \mathcal{A}$.*

De ce fait, lorsque $\mathcal{K}$ est incohérente, ajouter une assertion f de $(\mathcal{A} \setminus \mathcal{R})$ à $\mathcal{R}$ entraîne l'incohérence de $\langle \mathcal{T}, \mathcal{R} \cup \{f\} \rangle$.

Nous notons par $MAR(\mathcal{A})$ (de l'anglais *Maximal Assertional-based Repair*) l'ensemble des réparations de $\mathcal{A}$ par rapport à $\mathcal{T}$. Grâce à la notion de réparation, l'incohérence d'une base KB non-ordonnée peut être gérée par l'application d'un mécanisme standard de réponse aux requêtes, soit à l'ensemble des réparations (conséquence universelle ou conséquence AR [17]), soit à une seule réparation (conséquance "brave" [10]). Il est bien connu que la sémantique dite brave est aventureuse et peut entraîner des conclusions contestables, tandis que la sémantique AR est saine mais coûteuse sur le plan calculatoire.

Une alternative à cela est d'utiliser la sémantique IAR [17] qui sélectionne une seule sous-base cohérente de $\mathcal{A}$, dénotée $IAR(\mathcal{A})$. Avant d'introduire la sémantique IAR, nous présentons d'abord la notion de conflit assertionnel. Il s'agit d'un sous-ensemble d'assertions minimal incohérent qui contredit la base TBox. Formellement :

Définition 2 *Soit $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base KB incohérente. Une sous-base $C \subseteq \mathcal{A}$ est un conflit assertionnel dans $\mathcal{K}$ ssi $\langle \mathcal{T}, C \rangle$ est incohérente et $\forall f \in C$, $\langle \mathcal{T}, C \setminus \{f\} \rangle$ est cohérente.*

Nous notons par $C(\mathcal{A})$ l'ensemble des conflits de $\mathcal{A}$. D'après la Définition 2, nous constatons qu'enlever une assertion f de C restaure la cohérence de $\langle \mathcal{T}, C \rangle$. Une propriété intéressante de DL-Lite est que le calcul de l'ensemble des conflits se fait en temps polynomial [13]. Par ailleurs, un conflit C concerne exactement deux assertions [13]. Dans ce cas, si f et g sont deux assertions qui

appartiennent à un conflit, nous notons le conflit par une paire $\{f, g\}$ et nous disons que f et g sont en conflit.

Nous introduisons à présent la notion d'éléments libres.

Définition 3 *Soit $\mathcal{K}$=$\langle \mathcal{T}, \mathcal{A} \rangle$ une base KB incohérente. Une assertion $f \in \mathcal{A}$ est dite libre ssi $\forall C \in C(\mathcal{A})$: $f \notin C$.*

Intuitivement, les assertions libres correspondent aux éléments qui n'apparaissent dans aucun conflict. A l'origine, la notion d'éléments libres avait été proposée dans le cadre de la logique propositionelle [5].

Dans la suite de cet article, nous notons par $IAR(\mathcal{A})$ (de l'anglais *Intersection ABox Repair*) l'ensemble des éléments libres de $\mathcal{A}$. Formellement :

Définition 4 $IAR(\mathcal{A}) = \{f : f \in \mathcal{A}$ *t.q.* f *est libre*$\}$.

La Définition 4 est une réécriture équivalente de la définition standard de $IAR(\mathcal{A})$ donnée par l'intersection de toutes les réparations : $IAR(\mathcal{A}) = \bigcap\{\mathcal{R} \mid \mathcal{R} \in MAR(\mathcal{A})\}$ [5, 17].

Répondre à des requêtes avec la sémantique IAR revient à appliquer un mécanisme standard de réponse aux requêtes à la base $\langle \mathcal{T}, IAR(\mathcal{A}) \rangle$ (du fait que $\langle \mathcal{T}, IAR(\mathcal{A}) \rangle$ est cohérente).

Exemple 2 *L'ensemble des conflits dans $\langle \mathcal{T}, \mathcal{A} \rangle$ est :*

$$C(\mathcal{A}) = \left\{ \begin{array}{l} \{Mdanse(d_2), Tdanse(d_2)\}, \\ \{Mdanse(d_2), UtA(d_2, fl)\}, \\ \{DanseA(d_5), DanseS(d_5)\} \end{array} \right\}$$

Afin d'obtenir $IAR(\mathcal{A})$, il suffit d'enlever de $\mathcal{A}$ toutes les assertions de $C(\mathcal{A})$. Cela donne :

$$IAR(\mathcal{A}) = \left\{ \begin{array}{l} Mdanse(d_1), Tdanse(d_3), Tdanse(d_4), \\ DanseA(d_3), UtA(d_3, ch), UtA(d_4, ec) \end{array} \right\}$$

4 Réparation Non-Contestée pour des Bases Assertionnelles Totalement Préordonnées

Dans cette section, nous considérons des bases KB spécifiées en DL-Lite, incohérentes et totalement préordonnées. Donc une relation de préordre total $\geq$ est appliquée seulement à la composante ABox que nous notons par $(\mathcal{A}, \geq)$. La relation $\geq$ est réflexive, transitive et vérifie : $\forall f \in \mathcal{A}, \forall g \in \mathcal{A}$, soit $f \geq g$ ou bien $g \geq f$. Soient $>$ la relation de préférence stricte et $\equiv$ la relation d'équivalence associées à la relation $\geq$. De plus, nous représentons $(\mathcal{A}, \geq)$ par la partition bien-ordonnée de $\mathcal{A}$ induite par $\geq$. Ainsi, étant donné $(\mathcal{A}, \geq)$, nous considérons que $\mathcal{A}$ est partitionnée en n strates de la forme $\mathcal{A} = (\mathcal{S}_1, \ldots, \mathcal{S}_n)$, tel que :
– $\mathcal{S}_1 = \{f \in \mathcal{A} : \forall g \in \mathcal{A}, f \geq g\}$, et
– $\mathcal{S}_i = \{f \in \mathcal{A} : \forall g \in \mathcal{A} \setminus (\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_{i-1}), f \geq g\}$, pour $i = 2, \ldots, n$.
En d'autres termes, les assertions d'une strate $\mathcal{S}_i$ ont le même degré de priorité i, et elles sont plus fiables que celles d'une strate $\mathcal{S}_j$ pour $j > i$. Donc $\mathcal{S}_1$ contient les assertions les plus importantes, alors que $\mathcal{S}_n$ contient les assertions les moins importantes, et $\mathcal{A} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n$.

Plusieurs travaux prennent en compte la notion de priorité pour interroger des bases de données incohérentes (comme [18, 20]) ou des bases de connaissances DL (comme [9, 15]). La plupart de ces formalismes se basent sur les notions de réparation et de sémantique AR, ils sont donc coûteux sur le plan calculatoire. En particulier, la sémantique "des réparations préférées" a été introduite dans [9] (dans l'esprit de ce qui avait été proposé en logique propositionnelle pondérée [19, 11]). Cette sémantique revisite les sémantiques AR et IAR en remplaçant la notion de réparation par différents types de réparations préférées basées sur : le cardinal d'un ensemble, des degrés de priorité sur la base ABox et des poids sur les assertions. Cependant, ce formalisme entraîne souvent une augmentation de la complexité pour les sémantiques proposées. Notamment, la complexité de la sémantique IAR qui est polynomiale dans un cadre sans priorités explose lorsqu'un préordre total est appliqué à la base ABox.

Dans [4], une attention particulière a été accordée aux approches qui sélectionnent une seule réparation. L'une de ces approches est la réparation non-contestée qui a une complexité polynomiale sans être aventureuse (c'est-à-dire qui génère des conclusions contestables). Intuitivement, la réparation non-contestée consiste à récupérer itérativement, strate par strate, l'ensemble des assertions libres.

Définition 5 *Soit $\mathcal{K}$ une base KB incohérente dont la base ABox $(\mathcal{A}, \geq)$ est totalement préordonnée. Soit $\mathcal{A} = (\mathcal{S}_1, \ldots, \mathcal{S}_n)$ la partition bien-ordonnée associée à $\geq$.*
La réparation non-contestée $nd(\mathcal{A}, \geq) = \mathcal{S}'_1 \cup \ldots \cup \mathcal{S}'_n$ t.q. :
$$\forall i = 1, \ldots, n : \mathcal{S}'_i = IAR(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i)$$
où $\forall i$: $IAR(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i)$ est la base IAR de $(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i)$, donnée par la Définition 4.

La définition d'une sous-base non-contestée est une adaptation de celle proposée dans le cadre de la logique propositionnelle [6]. Cependant, la réparation non-contestée est calculée en temps polynomial dans le cadre de DL-Lite, alors que son calcul est difficile en logique propositionnelle.

Notons qu'une réécriture (similaire à celle de $IAR(\mathcal{A})$) est donnée pour $nd(\mathcal{A}, \geq)$ dans [3]. En fait, une assertion $f \in \mathcal{S}_i$ est dite contestée s'il existe une assertion $g \in \mathcal{S}_j$ telle que $j \leq i$, et $\{f, g\}$ est un conflit. Il a été démontré que $nd(\mathcal{A}, \geq)$ contient toutes les assertions non-contestées [3].

Exemple 3 *Reprenons notre exemple et considérons un préordre total $\geq$ sur les assertions de la base ABox représenté par la Figure 1, où :*
– $f \equiv g$ signifie que les deux assertions ont le même degré de priorité,
– la flèche $f \rightarrow g$ signifie que f est plus prioritaire que g (c-à-d. $f > g$).

$$Tdanse(d_2) \equiv Mdanse(d_1) \equiv Tdanse(d_3) \equiv UtA(d_3, ch)$$
$$\downarrow$$
$$UtA(d_2, fl) \equiv DanseA(d_5) \equiv DanseS(d_5) \equiv DanseA(d_3)$$
$$\downarrow$$
$$Mdanse(d_2) \equiv Tdanse(d_4) \equiv UtA(d_4, ec)$$

FIGURE 1 – Un préordre total sur la base ABox

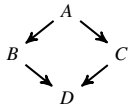


FIGURE 2 – Un préordre partiel sur la base ABox

A partir de cette ABox totalement préordonnée, il est possible de calculer la sous-classe non-contestée de $\mathcal{A}$. La partition bien-ordonnée correspondante est :
– $\mathcal{S}_1 = \{Mdanse(d_1), Tdanse(d_2), Tdanse(d_3), UtA(d_3, ch)\}$.
– $\mathcal{S}_2 = \{UtA(d_2, fl), DanseA(d_3), DanseA(d_5), DanseS(d_5)\}$.
– $\mathcal{S}_3 = \{Mdanse(d_2), Tdanse(d_4), UtA(d_4, ec)\}$.
Nous avons :
$nd(\mathcal{A}, \geq) = IAR(\mathcal{S}_1) \cup IAR(\mathcal{S}_1 \cup \mathcal{S}_2) \cup IAR(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3)$, tel que :
– $IAR(\mathcal{S}_1) = \{Tdanse(d_2), Mdanse(d_1), Tdanse(d_3), UtA(d_3, ch)\}$.
– $IAR(\mathcal{S}_1 \cup \mathcal{S}_2) = \{Tdanse(d_2), Mdanse(d_1), Tdanse(d_3), UtA(d_3, ch), UtA(d_2, fl), DanseA(d_3)\}$.
– $IAR(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3) = IAR(\mathcal{A})$ (donné dans l'exemple 2).
Par conséquent :
$nd(\mathcal{A}, \geq) = \{Mdanse(d_1), Tdanse(d_2), Tdanse(d_3), Tdanse(d_4), DanseA(d_3), UtA(d_2, fl), UtA(d_3, ch), UtA(d_4, ec)\}$.

5 Bases Assertionnelles Partiellement Préordonnées

La sémantique IAR (ABox non-ordonnée) et la sémantique "non-contestée" (ABox totalement préordonnée) sont connues pour leur efficacité pour gérer l'incohérence. En effet, elles calculent une réparation unique pour la base ABox et le font en temps polynomial. Dans cette section, nous proposons une méthode pour calculer une réparation unique lorsqu'un préordre partiel noté $\trianglerighteq$ est appliqué aux assertions de la base ABox notée $(\mathcal{A}, \trianglerighteq)$. Soient $\triangleright$ l'ordre strict (irréflexif et transitif) et $\overset{\trianglerighteq}{\equiv}$ la relation d'équivalence associés à la relation $\trianglerighteq$.

Nous appelons notre méthode "Elect" et notons par $Elect(\mathcal{A}, \trianglerighteq)$ la réparation qu'elle produit. Comme nous le verrons par la suite, la méthode Elect généralise la sémantique IAR dans le cas où la relation $\trianglerighteq$ est non-ordonnée, et aussi la sémantique "non-contestée" lorsque la relation $\trianglerighteq$ est totalement préordonnée.

5.1 Une famille de préordres totaux à partir d'un préordre partiel

Dans cette section, nous considérons qu'un préordre partiel $\trianglerighteq$ est interprété comme une famille de préordres totaux, de sorte que chaque préordre total est une extension totale de la relation $\trianglerighteq$ définie comme suit :

Définition 6 *Un préordre total $\geq$ sur $\mathcal{A}$ est une extension totale de $\trianglerighteq$ sur $\mathcal{A}$ ssi $\forall f, g \in \mathcal{A}$, si $f \trianglerighteq g$ alors $f \geq g$.*

Notons que l'interprétation d'une base KB partiellement préordonnée comme une famille de bases KB totalement préordonnées est une représentation naturelle qui a été utilisée dans d'autres cadres tels que la logique possibiliste partiellement ordonnée [7, 22] et les réseaux probabilistes crédaux [14]. Cette démarche n'a de sens que si les éléments peuvent être comparés (le fait de dire que deux assertions f et g sont incomparables signifie que soit f est strictement préférée à g, soit f est aussi préférée que g, soit g est strictement préférée à f, mais on ignore lequel).

Exemple 4 *Soit un préordre partiel $\trianglerighteq$ sur les assertions de la base ABox qui permet de les répartir en quatre sous-ensembles comme suit :*
– $A = \{Mdanse(d_1) \overset{\trianglerighteq}{\equiv} Tdanse(d_2) \overset{\trianglerighteq}{\equiv} Tdanse(d_3) \overset{\trianglerighteq}{\equiv} UtA(d_3, ch)\}$
– $B = \{UtA(d_2, fl) \overset{\trianglerighteq}{\equiv} DanseA(d_3) \overset{\trianglerighteq}{\equiv} DanseS(d_5) \overset{\trianglerighteq}{\equiv} DanseA(d_5)\}$
– $C = \{Mdanse(d_2)\}$
– $D = \{Tdanse(d_4) \overset{\trianglerighteq}{\equiv} UtA(d_4, ec)\}$
où $f \overset{\trianglerighteq}{\equiv} g$ signifie que les assertions f et g ont le même degré de priorité.
Ce préordre partiel est représenté par la Figure 2, où la flèche $A \rightarrow B$ (par exemple) signifie $\forall f \in A$, $\forall g \in B$, l'assertion f est plus prioritaire que g (c-à-d. $f \triangleright g$).
De ce fait, l'ensemble A (resp. D) contient les assertions les plus (resp. les moins) prioritaires. Les assertions des ensembles B et C sont incomparables.
Il s'en suit que le préordre partiel $\trianglerighteq$ peut être interprété comme une famille de trois préordres totaux $\geq_1, \geq_2$ et $\geq_3$ représentés par la Figure 3, de sorte que :
– Selon $\geq_1$, les assertions de l'ensemble B sont strictement préférées à celles de l'ensemble C.
– Selon $\geq_2$, les assertions des ensembles B et C sont préférées de manière égale.
– Selon $\geq_3$, les assertions de l'ensemble C sont strictement préférées à celles de l'ensemble B. □

Une question qui se pose est comment gérer cette famille de ABox totalement préordonnées. Nous voulons éviter le

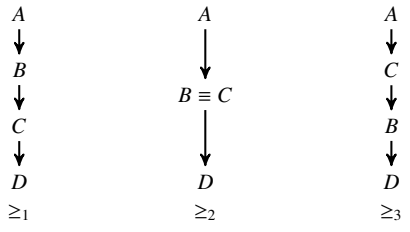


FIGURE 3 – Les extensions totales du préordre partiel $\trianglerighteq$

choix arbitraire qui consiste à sélectionner un seul préordre total parmi tous les autres. Il faudrait donc prendre en compte tous les préordres totaux. Une méthode prudente pour obtenir une sous-base assertionnelle cohérente unique est de considérer l'intersection de toutes les réparations non-contestées associées à tous les préordres totaux.

Définition 7 *Soit $\mathcal{K}$ une base KB spécifiée en DL-Lite, incohérente et dont la base ABox $(\mathcal{A}, \trianglerighteq)$ est partiellement préordonnée.*
– $Elect(\mathcal{A}, \trianglerighteq) = \bigcap_{\geq}\{nd(\mathcal{A}, \geq)$ t.q. $\geq$ est une extension totale de $\trianglerighteq\}$, où $nd(\mathcal{A}, \geq)$ est donné par la Définition 5.
– Soit q une requête. Alors q est une conséquence de $\mathcal{K}$ par la méthode Elect si q s'ensuit de $Elect(\mathcal{A}, \trianglerighteq)$ (en utilisant l'inférence standard de DL-lite).

Nous illustrons cette idée sur notre exemple.

Exemple 5 *Les réparations non-contestées des ABox totalement préordonnées $(\mathcal{A}, \geq_1)$, $(\mathcal{A}, \geq_2)$ et $(\mathcal{A}, \geq_3)$ sont :*
– $nd(\mathcal{A}, \geq_1) = A \cup \{UtA(d_2, fl), DanseA(d_3)\} \cup D$.
– $nd(\mathcal{A}, \geq_2) = A \cup \{DanseA(d_3)\} \cup D$.
– $nd(\mathcal{A}, \geq_3) = A \cup \{DanseA(d_3)\} \cup D$.
Par conséquent :
$Elect(\mathcal{A}, \trianglerighteq) = nd(\mathcal{A}, \geq_1) \cap nd(\mathcal{A}, \geq_2) \cap nd(\mathcal{A}, \geq_3)$
$= A \cup \{DanseA(d_3)\} \cup D.$ □

Dans ce qui suit, nous étudions quelques caractéristiques de notre méthode.

Un résultat important énoncé dans la Proposition 1 est que le calcul de la réparation $\text{Elect}(\mathcal{A}, \trianglerighteq)$ peut s'effectuer en temps polynomial. En effet, il n'est pas nécessaire d'exhiber toutes les extensions totales de la relation $\trianglerighteq$.

Proposition 1 *Le calcul de la réparation $Elect(\mathcal{A}, \trianglerighteq)$ se fait en temps polynomial (par rapport à la taille de $\mathcal{A}$).*

La proposition suivante établit qu'une base KB ayant $Elect(\mathcal{A}, \trianglerighteq)$ comme base ABox est cohérente.

Proposition 2 *$\langle \mathcal{T}, Elect(\mathcal{A}, \trianglerighteq)\rangle$ est cohérente.*

Une autre caractéristique de la méthode Elect est qu'elle correspond à la sémantique IAR lorsque la base ABox est non-ordonnée, et à la sémantique "non-contestée" lorsque la base ABox est totalement préordonnée.

Proposition 3
– Si le préordre partiel $\trianglerighteq$ est non-ordonné, alors $Elect(\mathcal{A}, \trianglerighteq) = IAR(\mathcal{A})$.
– Si le préordre partiel $\trianglerighteq$ est un préordre total, alors $Elect(\mathcal{A}, \trianglerighteq) = nd(\mathcal{A}, \trianglerighteq)$.

Les preuves des Propositions 1 à 3 sont établies grâce à la définition d'une caractérisation équivalente de la réparation $Elect(\mathcal{A}, \trianglerighteq)$, présentée dans la prochaine section.

5.2 Caractérisation de la réparation Elect$(\mathcal{A}, \trianglerighteq)$

Dans cette section, nous proposons une caractérisation de la réparation $Elect(\mathcal{A}, \trianglerighteq)$ qui permet d'éviter de calculer l'ensemble des extensions totales de la relation $\trianglerighteq$. Pour cela, nous introduisons le concept d'assertions élues. Intuitivement, une assertion est élue dans $(\mathcal{A}, \trianglerighteq)$ si elle est strictement préférée à toutes les assertions avec lesquelles elle est en conflit. Formellement :

Définition 8 *Une assertion $f \in \mathcal{A}$ est élue ssi $\forall g \in \mathcal{A}$, si $\{f, g\}$ est un conflit, alors $f \rhd g$ (c-à-d. f est strictement préférée à g).*

La Définition 8 généralise le concept d'assertions libres donné dans la Définition 3. En effet, si la relation $\trianglerighteq$ est non-prioritaire (c-à-d. $\forall f \in \mathcal{A}, \forall g \in \mathcal{A}, f \trianglerighteq g$ et $g \trianglerighteq f$), alors f est élue dans $(\mathcal{A}, \trianglerighteq)$ ssi f est libre. Evidemment, le contraire n'est en général pas vrai (lorsque la relation $\trianglerighteq$ est prioritaire). En effet, une assertion élue peut ne pas être une assertion libre, mais son degré de fiabilité est strictement plus important que celui de ses opposants. Cette définition généralise également la notion d'assertions acceptées donnée pour des réparations non-contestées dans des bases KB totalement préordonnées [3]. Enfin, le concept d'assertions élues est dans le même esprit que la notion de croyances acceptées proposée en théorie de l'incertitude [16].

La Proposition 4 énonce que les assertions élues sont exactement celles de la réparation $Elect(\mathcal{A}, \trianglerighteq)$.

Proposition 4 *Une assertion $f \in \mathcal{A}$ est élue dans $(\mathcal{A}, \trianglerighteq)$ ssi $f \in Elect(\mathcal{A}, \trianglerighteq)$.*

Nous démontrons cette proposition comme suit.

Preuve 1 *Soit $(\mathcal{A}, \trianglerighteq)$ une base assertionnelle partiellement préordonnée.*
1. Soit $f \in \mathcal{A}$ une assertion élue. Montrons que pour chaque extension totale $(\mathcal{A}, \geq)$ de $(\mathcal{A}, \trianglerighteq)$, nous avons $f \in nd(\mathcal{A}, \geq)$. Soit $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ la partition bien-ordonnée associée à $\geq$. Soit i la première strate où $f \in \mathcal{S}_i$.
Rappelons que f est élue dans $(\mathcal{A}, \trianglerighteq)$ signifie $\forall g \in \mathcal{A}$, si $\{f, g\}$ est un conflit, alors $f \rhd g$ (c-à-d. f est strictement préférée à g selon $\trianglerighteq$). Et puisque $\geq$ est une extension totale de $\trianglerighteq$, alors cela signifie que $f > g$. Cela signifie aussi $\forall g \in \mathcal{A}$ tel que $\{f, g\}$ est un conflit, $g \in \mathcal{S}_j$ avec $j > i$. Donc, $f \in IAR(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i)$. Il s'en suit que $f \in nd(\mathcal{A}, \geq)$.

2. Montrons à présent l'inverse. Supposons que $f \in \mathcal{A}$ n'est pas élue et construisons une extension totale $(\mathcal{A}, \geq)$ de $(\mathcal{A}, \trianglerighteq)$ telle que $f \notin nd(\mathcal{A}, \geq)$.
L'assertion f n'est pas élue signifie $\exists g \in \mathcal{A}$ tel que $\{f, g\}$ est un conflit et $f \rhd g$ n'est pas vrai. Donc il existe une extension totale $\geq$ de $\trianglerighteq$ où $g \geq f$. Si $\{f, g\}$ est un conflit et $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ est la partition bien-ordonnée associée à $\geq$, alors si $f \in \mathcal{S}_i$, il s'en suit que $g \in \mathcal{S}_j$ où $j \leq i$. Par conséquent, $\forall k \in \{1, \ldots, n\}$, $f \notin IAR(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_k)$, donc $f \notin nd(\mathcal{A}, \geq)$. □

A partir de ce résultat, nous pouvons démontrer les Propositions 1, 2 et 3 données dans la section précédente.

Preuve 2
1. En ce qui concerne la complexité calculatoire, nous rappelons que le calcul de l'ensemble des conflits $C(\mathcal{A})$ se fait en temps polynomial par rapport à la taille de $\mathcal{A}$. Donc, calculer $Elect(\mathcal{A}, \trianglerighteq)$ se fait aussi en temps polynomial. En effet, vérifier si une assertion $f \in \mathcal{A}$ est élue revient à parcourir tous les conflits assertionnels dans $C(\mathcal{A})$. Ceci se fait en temps linéaire par rapport à la taille de $C(\mathcal{A})$ (cette taille étant bornée par $O(|\mathcal{A}|^2)$).

2. Montrons que la base $Elect(\mathcal{A}, \trianglerighteq)$ est cohérente par rapport à $\mathcal{T}$. Supposons que ce n'est pas le cas. Donc $\exists f \in Elect(\mathcal{A}, \trianglerighteq)$, $\exists g \in Elect(\mathcal{A}, \trianglerighteq), g \neq f$, tel que $\{f, g\}$ est un conflit. Puisque f et g sont dans $Elect(\mathcal{A}, \trianglerighteq)$, alors cela signifie que $f \rhd g$ et $g \rhd f$, ce qui est impossible.

3. Finalement, par construction de la base $Elect(\mathcal{A}, \trianglerighteq)$, il est facile de vérifier que lorsque $\trianglerighteq$ est un préordre total, alors $Elect(\mathcal{A}, \trianglerighteq)$ est la réparation non-contestée de $\trianglerighteq$. Et lorsque $\trianglerighteq$ est non-ordonnée (donc $\forall f \in \mathcal{A}, \forall g \in \mathcal{A}$, $f \trianglerighteq g$ et $g \trianglerighteq f$), alors $Elect(\mathcal{A}, \trianglerighteq) = IAR(\mathcal{A}) = \{f \in \mathcal{A} : \nexists g \in \mathcal{A}$, t.q. $\{f, g\}$ est un conflit$\}$. □

Exemple 6 *Nous utilisons la notion d'assertions élues de la Définition 8 pour recalculer la réparation $Elect(\mathcal{A}, \trianglerighteq)$. Il est aisé de vérifier que :*
– L'assertion $Mdanse(d_2)$ n'est pas élue car $\{Mdanse(d_2), Tdanse(d_2)\}$ est un conflit et la préférence stricte $Mdanse(d_2) \rhd Tdanse(d_2)$ n'est pas vérifiée.
– Les assertions $DanseS(d_5)$ et $DanseA(d_5)$ ne sont pas élues car elles sont en conflit et ont le même degré de priorité.
– L'assertion $UtA(d_2, fl)$ n'est pas élue car $\{Mdanse(d_2), UtA(d_2, fl)\}$ est un conflit et la préférence stricte $UtA(d_2, fl) \rhd Mdanse(d_2)$ n'a pas lieu.
– Les assertions restantes sont toutes élues. A savoir :
$Elect(\mathcal{A}, \trianglerighteq) = \{Mdanse(d_1), Tdanse(d_2), Tdanse(d_3), Tdanse(d_4), DanseA(d_3), UtA(d_3, ch), UtA(d_4, ec)\}$.

Ce résultat correspond à la réparation calculée dans l'Exemple 5 en considérant toutes les extensions totales du préordre partiel $\trianglerighteq$.

6 Extensions de la Méthode Elect

Dans cette section, nous proposons une discussion brève de trois extensions possibles de la méthode Elect. Leur étude approfondie fera l'objet de travaux futurs.

6.1 Au-delà de la méthode Elect

Une question qui se pose est comment obtenir une base plus grande (plus productive) qu'une base calculée par la méthode Elect, tout en produisant une réparation saine. Une solution immédiate est d'utiliser la notion de fermeture déductive positive, selon laquelle la fermeture de la base ABox est définie en termes des axiomes positifs de la base $\mathcal{T}$. Nous introduisons l'opérateur de fermeture [12, 4] comme suit :

Définition 9 *Soit $\mathcal{T}_p$ l'ensemble de tous les axiomes d'inclusions positifs de $\mathcal{T}$. La fermeture déductive de $\mathcal{A}$ par rapport à $\mathcal{T}$ est :*
$cl(\mathcal{A}) = \{B(a) : \langle \mathcal{T}_p, \mathcal{A} \rangle \models B(a)$ t.q. B est un concept de $\mathcal{T}$ et a est un individu de $\mathcal{A}\} \bigcup \{R(a, b) : \langle \mathcal{T}_p, \mathcal{A} \rangle \models R(a, b)$ t.q. R est un rôle de $\mathcal{T}$, et a et b sont des individus de $\mathcal{A}\}$.
Ici, $\models$ est une relation d'inférence standard en DL-Lite.

Il y a deux façons d'appliquer la fermeture déductive positive, soit sur la base ABox initiale, soit sur l'ensemble des réparations non-contestées calculées pour toutes les extensions totales du préordre partiel.

Dans la première option, l'application de la fermeture positive à la base ABox initiale, dans l'esprit de la sémantique ICAR pour des bases ABox non-ordonnées [17], soulève deux questions. Premièrement, en termes de sémantique, l'approche ICAR peut ne pas être fiable puisqu'elle permet de dériver des conséquences à partir d'assertions discutables. Deuxièmement, il peut y avoir plusieurs façons de définir la fiabilité des éléments dérivés. Par exemple, supposons que la base TBox contient $\{A \sqsubseteq B, E \sqsubseteq B\}$ et que la base ABox contient $\{A(x), E(x)\}$. Supposons que $A(x)$ et $E(x)$ sont incomparables. Donc $B(x)$ peut être dérivée de $A(x)$ mais aussi de $E(x)$. La question est alors où positionner $B(x)$. L'intuition est de considérer $B(x)$ comme *au moins aussi plausible que* $A(x)$ et $E(x)$, mais cela n'est pas évident à définir de manière générale (en particulier pour des logiques de description expressives).

La deuxième option consiste à définir la fermeture de toutes les réparations non-contestées, ce qui est plus sain. En effet, les conclusions supplémentaires sont obtenues uniquement d'assertions appartenant à des réparations non-contestées, qui sont connues pour ne contenir que des assertions saines. Ainsi, la réparation associée à un préordre partiel est calculée comme l'intersection des réparations non-contestées fermées [4], une méthode que nous appelons CElect.

Définition 10 *Soit* $(\mathcal{A}, \trianglerighteq)$ *une base ABox partiellement préordonnée,* $cl(.)$ *un opérateur de fermeture donné par la Définition 9, et* $nd(\mathcal{A}, \geq)$ *est donné par la Définition 5.*
$CElect(\mathcal{A}, \trianglerighteq) = \bigcap_{\geq}\{cl(nd(\mathcal{A}, \geq))$ *t.q.* $\geq$ *est une extension totale de* $\trianglerighteq\}$.

Une base calculée avec la méthode CElect est alors plus grande qu'une base calculée avec la méthode Elect. De plus, pour des bases ABox non-ordonnées, une réparation calculée avec la méthode CElect est équivalente à la fermeture des réparations IAR (ce qui est différent d'une réparation calculée avec la méthode ICAR). Et pour des bases ABox totalement préordonnées, une réparation calculée avec la méthode CElect est équivalente à la fermeture des réparations non-contestées. Dans des travaux futurs, nous envisageons d'exhiber des cas spécifiques où la complexité de la méthode CElect serait également polynomiale en DL-Lite.

6.2 Au-delà de la réparation non-contestée

La question traitée dans cette partie concerne la possibilité ou non d'utiliser une sémantique autre que la sémantique "non-contestée" pour définir la méthode Elect. D'un point de vue sémantique, la réponse est positive. Par exemple, il est possible d'utiliser l'une des réparations préférées définies dans [9] au lieu de la réparation non-contestée dans notre définition de $Elect(\mathcal{A}, \trianglerighteq)$. Rappelons d'abord la notion de réparation préférée pour des bases ABox totalement préordonnées.

Définition 11 *Soit* $\mathcal{A} = (\mathcal{S}_1, \ldots, \mathcal{S}_n)$ *une base ABox totalement préordonnée. Soient* $\mathcal{R}_1$ *et* $\mathcal{R}_2$ *deux sous-bases cohérentes de* $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_n$.
– $\mathcal{R}_1$ *est aussi préférée que* $\mathcal{R}_2$ *ssi* $\forall i,\ 1 \leq i \leq n,\ \mathcal{R}_1 \cap \mathcal{S}_i = \mathcal{R}_2 \cap \mathcal{S}_i$.
– $\mathcal{R}_1$ *est strictement préférée à* $\mathcal{R}_2$ *ssi* $\exists i,\ 1 \leq i \leq n,$ *t.q.* $\mathcal{R}_2 \cap \mathcal{S}_i \subsetneq \mathcal{R}_1 \cap \mathcal{S}_i$ *et* $\forall j,\ 1 \leq j < i,\ \mathcal{R}_1 \cap \mathcal{S}_j = \mathcal{R}_2 \cap \mathcal{S}_j$.

Donc une base $\mathcal{R} \subseteq \mathcal{A}$ *est une réparation préférée ssi* $\nexists \mathcal{R}' \subseteq \mathcal{A}$ *t.q.* $\mathcal{R}'$ *est strictement préférée à* $\mathcal{R}$.

(Se référer à [9] et aussi [6] pour de plus amples détails.)

La notion de réparation préférée définie pour un préordre total peut alors servir de base pour définir une réparation en présence d'un préordre partiel $\trianglerighteq$.

Nous appelons ce nouveau formalisme $Partial_{PR}(\mathcal{A}, \trianglerighteq)$ (où *PR* désigne en anglais *Preferred Repairs*). Tout comme une réparation $Elect(\mathcal{A}, \trianglerighteq)$, une réparation $Partial_{PR}(\mathcal{A}, \trianglerighteq)$ considère toutes les extensions totales $\geq$ du préordre partiel $\trianglerighteq$. Cependant, au lieu de considérer l'intersection des réparations non-contestées comme pour calculer $Elect(\mathcal{A}, \trianglerighteq)$, nous considérons l'intersection des réparations préférées, notée *IPR* (de l'anglais *Intersection of Preferred Repairs*), comme suit :

Définition 12 *Soit* $(\mathcal{A}, \trianglerighteq)$ *une base ABox partiellement préordonnée. Soit* $\geq$ *une extension totale de* $\trianglerighteq$, *et* $(\mathcal{A}, \geq)$ *la base ABox totalement préordonnée correspondante.*
– L'intersection des réparations préférées (au sens de la Définition 11) associées à $(\mathcal{A}, \geq)$ *est :*
$IPR(\mathcal{A}, \geq) = \bigcap\{\mathcal{R} \mid \mathcal{R}$ *est une réparation préférée de* $\geq\}$.
– La réparation préférée associée à $(\mathcal{A}, \trianglerighteq)$ *est :*
$Partial_{PR}(\mathcal{A}, \trianglerighteq) = \bigcap_{\geq}\{IPR(\mathcal{A}, \geq)$ *t.q.* $\geq$ *est une extension totale de* $\trianglerighteq\}$.

La réparation $Partial_{PR}(\mathcal{A}, \trianglerighteq)$ est plus grande que la réparation $Elect(\mathcal{A}, \trianglerighteq)$. Cependant, la base $Partial_{PR}(\mathcal{A}, \trianglerighteq)$ n'est pas calculable en temps raisonnable puisque la complexité de $IPR(\mathcal{A}, \trianglerighteq)$ est coNP [9] si la relation $\trianglerighteq$ est un simple préordre total. Dans ce cas, la Proposition 1 n'est pas vérifiée.

6.3 Au-delà de DL-Lite

Une autre question concerne le fait de pouvoir ou non généraliser la méthode Elect à des bases ABox partiellement préordonnées exprimées dans des logiques autres que DL-Lite. D'un point de vue sémantique, il n'y a pas de limitations et les résultats obtenus correspondent également à la réparation IAR (base ABox non-ordonnée) et à la réparation non-contestée (base ABox totalement préordonnée). En effet, les sémantiques IAR et "non-contestée" sont définies indépendamment de la taille des conflits. En fait, la réparation IAR est simplement l'intersection de toutes les réparations maximales, et la réparation non-contestée est exprimée en termes de réparations IAR.

Cependant, d'un point de vue calculatoire, il est impératif d'avoir un moyen efficace de gérer les conflits afin que la Proposition 1 soit vérifiée. En particulier, il n'est pas nécessaire que les conflits assertionnels $C \in \mathcal{C}(\mathcal{A})$ soient binaires (c-à-d. impliquant deux assertions), à condition qu'ils puissent être calculés en temps polynomial.

Dans le cas de conflits impliquant plus de deux assertions, nous redéfinissons la notion d'assertion élue comme étant celle qui est strictement préférée à au moins une assertion de chaque conflit où elle apparaît. Nous utilisons le terme d'assertion élue y compris pour des conflits non-binaires.

Définition 13 *Soit* $\mathcal{K}$ *une base KB spécifiée en Logique de Description, incohérente et dont la base ABox* $(\mathcal{A}, \trianglerighteq)$ *est partiellement préordonnée. Une assertion* $f \in \mathcal{A}$ *est élue ssi* $\forall C \in \mathcal{C}(\mathcal{A})$ *où* $f \in C$, $\exists g \in C, g \neq f$, *tel que* $f \rhd g$ *(c-à-d. l'assertion* f *est strictement préférée à* g*).*

Notons que lorsque C est un conflit binaire, la Définition 13 correspond à la Définition 8.

Pour des langages plus expressifs que DL-Lite, nous définissons formellement la réparation basée sur cette notion d'assertions élues. Nous la nommons "réparation dl-Elect".

Définition 14 *Soit $\mathcal{K}$ une base KB spécifiée en Logique de Description, incohérente et dont la base ABox $(\mathcal{A}, \trianglerighteq)$ est partiellement préordonnée. Alors :*
dl-Elect$(\mathcal{A}, \trianglerighteq) = \bigcap_{\geq}\{nd(\mathcal{A}, \geq)$ t.q. $\geq$ est une extension totale de $\trianglerighteq\}$, où $nd(\mathcal{A}, \geq)$ est donné par la Définition 5.

Nous proposons une caractérisation (sans avoir à spécifier tous les préordres totaux) de la réparation dl-Elect$(\mathcal{A}, \trianglerighteq)$.

Proposition 5
1. Une assertion f est élue dans $(\mathcal{A}, \trianglerighteq)$ ssi $f \in$ dl-Elect$(\mathcal{A}, \trianglerighteq)$ (où une assertion élue est donnée par la Définition 13, et dl-Elect$(\mathcal{A}, \trianglerighteq)$ est donné par la Définition 14).

2. dl-Elect$(\mathcal{A}, \trianglerighteq)$ est cohérent par rapport à $\mathcal{T}$.

Preuve 3 *Soit $(\mathcal{A}, \trianglerighteq)$ une base assertionnelle partiellement préordonnée.*

1.i) Soit $f \in \mathcal{A}$ une assertion élue. Montrons que pour chaque extension totale $\geq$ de $\trianglerighteq$, nous avons $f \in nd(\mathcal{A}, \geq)$. Soit $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ la partition bien-ordonnée associée à $\geq$. Supposons que $f \in \mathcal{S}_i$ pour un certain $i \in \{1, \ldots, n\}$.
Puisque f est élue dans $(\mathcal{A}, \trianglerighteq)$ et $\geq$ est une extension totale de $\trianglerighteq$, appliquer la Définition 13 entraîne $\forall C \in C(\mathcal{A})$ où $f \in C, \exists g \in C, g \neq f$, t.q. $f \rhd g$ et également $f > g$. Cela signifie aussi $\forall C \in C(\mathcal{A})$ t.q. $f \in C$, $\exists g \in C$ t.q. $g \neq f$ et $g \in \mathcal{S}_j$ avec $j > i$. Donc, il n'y a aucun conflit C dans $\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i$ t.q. $f \in C$. (Rappelons qu'un conflit est un ensemble d'assertions minimal incohérent par rapport à $\mathcal{T}$. Donc, enlever un élément de C produit un ensemble d'assertions cohérent par rapport à $\mathcal{T}$.)
Ainsi, $f \in IAR(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_i)$. Donc, $f \in nd(\mathcal{A}, \geq)$. Par conséquent, $f \in$ dl-Elect$(\mathcal{A}, \trianglerighteq)$.

1.ii) Montrons à présent l'inverse. Supposons que $f \in \mathcal{A}$ n'est pas élue et construisons une extension totale $\geq$ de $\trianglerighteq$ tel que $f \notin nd(\mathcal{A}, \geq)$. L'assertion f n'est pas élue signifie $\exists C \in C(\mathcal{A})$ t.q. $f \in C$ et $\forall g \in C$, $f \rhd g$ n'est pas vrai. Cela veut dire qu'il existe une extension totale $\geq$ de $\trianglerighteq$ t.q. $\forall g \in C$, $g \geq f$. En effet, il suffit de définir l'ordre sur les éléments de C par rapport à f comme suit : $\forall g \in C$, $g \neq f$, $g \geq f$, puis de compléter le reste de la relation de sorte à étendre $\trianglerighteq$. Soit $(\mathcal{S}_1, \ldots, \mathcal{S}_n)$ la partition bien-ordonnée associée à $\geq$, et soit $f \in \mathcal{S}_i$ pour un certain $i \in \{1, \ldots, n\}$. Puisque $\forall g \in C, g \geq f$, il s'ensuit $\forall g \in C$, si $f \in \mathcal{S}_i$ alors $g \in \mathcal{S}_j$ pour un certain $j \leq i$. Donc, $\forall k \in \{1, \ldots, n\}$, $f \notin IAR(\mathcal{S}_1 \cup \ldots \cup \mathcal{S}_k)$, ce qui signifie que $f \notin nd(\mathcal{A}, \geq)$.

2. Pour montrer la cohérence de dl-Elect$(\mathcal{A}, \trianglerighteq)$ par rapport à $\mathcal{T}$, supposons le contraire. Donc il existe un conflit $C \subseteq$ dl-Elect$(\mathcal{A}, \trianglerighteq)$. Etant donné que chaque élément de C est élu, alors $\forall f \in C, \exists g \in C, g \neq f$, t.q. $f \rhd g$, ce qui est impossible. □

Considérons la version modifiée suivante de notre exemple.

Exemple 7 *Soient Reg_1, Reg_2 et Reg_3 des régions d'origine de danses.*
Soit $\mathcal{T}' = \mathcal{T} \bigcup \{Mdanse \sqsubseteq Reg_1, Tdanse \sqsubseteq Reg_2,$
$Reg_1 \sqcap Reg_2 \sqcap Reg_3 \sqsubseteq \bot\}$.
Le premier (resp. second) axiome indique que les danses modernes (resp. traditionnelles) proviennent de la région 1 (resp. région 2). Le troisième axiome énonce qu'une danse peut être originaire de deux régions mais pas de trois. Cet axiome représente un conflit ternaire, donc il ne peut pas être exprimé en DL-Lite.
Soit $\mathcal{A}' = \mathcal{A} \bigcup \{Reg_1(d_6), Reg_2(d_6), Reg_3(d_6), Mdanse(d_6),$ $Tdanse(d_6)\}$, où d_6 représente une danse.
Supposons que $\trianglerighteq$ sur $\mathcal{A}'$ définit les préférences strictes :
– $Reg_1(d_6) \rhd Reg_2(d_6) \rhd Reg_3(d_6)$, et
– $Mdanse(d_6) \rhd Tdanse(d_6)$.
L'ensemble des conflits est :

$$C(\mathcal{A}') = C(\mathcal{A}) \cup \left\{ \begin{array}{l} \{Reg_1(d_6), Reg_2(d_6), Reg_3(d_6)\}, \\ \{Mdanse(d_6), Tdanse(d_6)\} \end{array} \right\}$$

Il est aisé de vérifier que : dl-Elect$(\mathcal{A}', \trianglerighteq)$ = Elect$(\mathcal{A}, \trianglerighteq)$ $\bigcup \{Reg_1(d_6), Reg_2(d_6), Mdanse(d_6)\}$. □

D'un point de vue calculatoire, la tractabilité peut être préservée dans le contexte de langages plus expressifs que DL-Lite à condition que le calcul des conflits se fasse de manière efficace. En effet, la complexité pour calculer la réparation dl-Elect$(\mathcal{A}, \trianglerighteq)$ dépend de la complexité de calculer les conflits. Si cette dernière est polynomiale, alors tout le processus est polynomial. Notons que vérifier si une assertion $f \in \mathcal{A}$ est élue revient simplement à parcourir tous les conflits assertionnels, et vérifier pour chaque conflit $C \in C(\mathcal{A})$ s'il existe une assertion $g \in C$ qui est strictement moins préférée que f. Ceci se fait en temps polynomial par rapport à la taille et au nombre de conflits.

7 Conclusion

Nous avons étudié le problème de la restauration de la cohérence d'une base ABox partiellement préordonnée et incohérente par rapport à la base TBox pour des ontologies DL-Lite. Nous avons proposé une méthode appelée Elect qui généralise la sémantique IAR (ABox non-ordonnée) et la sémantique dite non-contestée (ABox totalement préordonnée). Dans la méthode Elect, un préordre partiel est interprété comme une famille de préordres totaux auxquels l'inférence non-contestée est appliquée, produisant des réparations non-contestées. Nous avons introduit le concept d'assertions élues et proposé une caractérisation équivalente de la méthode Elect. En particulier, nous avons montré que la complexité de la méthode Elect est polynomiale.

Dans le cadre du projet AniAge, nous envisageons d'appliquer la méthode Elect à des ontologies représentant des danses de l'Asie du Sud-Est. Des experts annotent des vidéos de danses par rapport à la base TBox pour capturer les

connaissances culturelles représentées par certaines postures de danses, des mouvements, des tenues ou des accessoires. Différents experts peuvent annoter une même vidéo, donnant lieu à des conflits que nous proposons de résoudre avec la méthode Elect.

Remerciements : Les auteurs remercient les relecteurs pour leurs commentaires utiles à l'amélioration de cet article. Ce travail a été financé par le projet européen H2020-MSCA-RISE : AniAge (High Dimensional Heterogeneous Data based Animation Techniques for Southeast Asian Intangible Cultural Heritage).

Références

[1] Baader, F., D. Calvanese, D. Mcguinness, D. Nardi et P. Patel-Schneider: *The Description Logic Handbook : Theory, Implementation, and Applications*. 2007.

[2] Baget, J-F., S. Benferhat, Z. Bouraoui, M. Croitoru, M-L. Mugnier, O. Papini, S. Rocher et K. Tabia: *A General Modifier-Based Framework for Inconsistency-Tolerant Query Answering*. Dans *KR, Le Cap, Afrique du Sud*, pages 513–516, 2016.

[3] Benferhat, S., Z. Bouraoui, H. Chadhry, M. Shafry Bin Mohd Rahim Fc, K. Tabia et A. Telli: *Characterizing Non-Defeated Repairs in Inconsistent Lightweight Ontologies*. Dans *SITIS, Naples, Italie*, pages 282–287, 2016.

[4] Benferhat, S., Z. Bouraoui et K. Tabia: *How to Select One Preferred Assertional-Based Repair from Inconsistent and Prioritized DL-Lite Knowledge Bases ?* Dans *IJCAI, Buenos Aires, Argentine*, pages 1450–1456, 2015.

[5] Benferhat, S., D. Dubois et H. Prade: *Representing Default Rules in Possibilistic Logic*. Dans *Knowledge Representation and Reasoning*, pages 673–684, 1992.

[6] Benferhat, S., D. Dubois et H. Prade: *Some syntactic approaches to the handling of inconsistent knowledge bases : A comparative study. Part 2 : the prioritized case*, tome 24, pages 473–511. 1998.

[7] Benferhat, S., S. Lagrue et O. Papini: *Reasoning with partially ordered information in a possibilistic logic framework*. Fuzzy Sets and Systems, 144(1) :25–41, 2004.

[8] Bienvenu, M. et C. Bourgaux: *Inconsistency-Tolerant Querying of Description Logic Knowledge Bases*. Dans *Reasoning Web : Logical Foundation of Knowledge Graph Construction and Query Answering*, tome 9885, pages 156–202. LNCS. Springer, 2016.

[9] Bienvenu, M., C. Bourgaux et F. Goasdoué: *Querying Inconsistent Description Logic Knowledge Bases under Preferred Repair Semantics*. Dans *AAAI, Québec, Canada*, pages 996–1002, 2014.

[10] Bienvenu, M. et R. Rosati: *Tractable Approximations of Consistent Query Answering for Robust Ontology-based Data Access*. Dans *IJCAI, Pékin, Chine*, pages 775–781, 2013.

[11] Brewka, G.: *Preferred Subtheories : An Extended Logical Framework for Default Reasoning*. Dans *IJCAI, Detroit, USA*, pages 1043–1048, 1989.

[12] Calvanese, D., G. De Giacomo, D. Lembo, M. Lenzerini et R. Rosati: *Tractable Reasoning and Efficient Query Answering in Description Logics : The DL-Lite Family*. Journal of Automated Reasoning, 39(3) :385–429, 2007.

[13] Calvanese, D., E. Kharlamov, W. Nutt et D. Zheleznyakov: *Evolution of DL-Lite Knowledge Bases*. Dans *International Semantic Web Conference (1), Shanghai, Chine*, pages 112–128, 2010.

[14] Cozman, F.G.: *Credal networks*. Artificial Intelligence Journal, 120 :199–233, 2000.

[15] Du, J., G. Qi et Y. Shen: *Weight-based consistent query answering over inconsistent SHIQ knowledge bases*. Knowledge and Information Systems, 34(2) :335–371, 2013.

[16] Dubois, D., H. Fargier et H. Prade: *Ordinal and Probabilistic Representations of Acceptance*. Journal of Artificial Intelligence Research, 22 :23–56, 2004.

[17] Lembo, D., M. Lenzerini, R. Rosati, M. Ruzzi et D. Fabio Savo: *Inconsistency-Tolerant Semantics for Description Logics*. Dans *Web Reasoning and Rule Systems*, tome 6333 de *LNCS*, pages 103–117, 2010.

[18] Martinez, M. V., F. Parisi, A. Pugliese, G. I. Simari et V. S. Subrahmanian: *Inconsistency Management Policies*. Dans *Knowledge Representation and Reasoning*, pages 367–377. AAAI Press, 2008.

[19] Rescher, N. et R. Manor: *On inference from inconsistent premisses*. Theory and Decision, 1(2) :179–217, 1970.

[20] Staworko, S., J. Chomicki et J. Marcinkowski: *Prioritized repairing and consistent query answering in relational databases*. Annals of Mathematics and Artificial Intelligence, 64(2-3) :209–246, 2012.

[21] Telli, A., S. Benferhat, M. Bourahla, Z. Bouraoui et K. Tabia: *Polynomial Algorithms for Computing a Single Preferred Assertional-Based Repair*. Künstliche Intelligenz, 31(1) :15–30, 2017.

[22] Touazi, F., C. Cayrol et D. Dubois: *Possibilistic reasoning with partially ordered beliefs*. Journal of Applied Logic, 13(4) :770–798, 2015.

[23] Trivela, D., G. Stoilos et V. Vassalos: *Querying Expressive DL Ontologies under the ICAR Semantics*. Dans *DL workshop. Tempe, USA*, 2018.

Actes JIAF 2019

An Abstraction-based Method for Verifying Strategic Properties in Multi-agent Systems with Imperfect Information *

Francesco Belardinelli[1,2] **Alessio Lomuscio**[1] **Vadim Malvone**[2]

[1] Imperial College London, United Kingdom
[2] Laboratoire IBISC, Université d'Evry, France
francesco.belardinelli@imperial.ac.uk a.lomuscio@imperial.ac.uk
vadim.malvone@univ-evry.fr

Abstract

We investigate the verification of Multi-agent Systems against strategic properties expressed in Alternating-time Temporal Logic under the assumptions of imperfect information and perfect recall. To this end, we develop a three-valued semantics for concurrent game structures upon which we define an abstraction method. We prove that concurrent game structures with imperfect information admit perfect information abstractions that preserve three-valued satisfaction. Further, we present a refinement procedure to deal with cases where the value of a specification is undefined. We illustrate the overall procedure in a variant of the Train Gate Controller scenario under imperfect information and perfect recall.

1 Introduction

Alternating-time Temporal Logic (ATL) and its extension ATL^* are well-known formalisms for reasoning about strategic behaviours in Multi-agent Systems [2]. An attractive feature of ATL is the computational complexity of its model checking problem, which is PTIME-complete under the assumption of perfect information. Multi-agent systems (MAS), however, typically exhibit imperfect information, and model checking MAS against ATL specifications under imperfect information and perfect recall is known to be undecidable [14]. Given the practical and theoretical importance of the imperfect information setting, even partial solutions to the problem can be useful. Previous approaches (see related work below) have either focused on how the information is shared amongst the agents in the system [10, 9], or developed notions of bounded recall [8].

Instead, at the heart of the present contribution is the idea that, under a three-valued semantics, MAS with imperfect information can be approximated (or *abstracted*) by perfect information variants. This enables us to derive a sound, albeit incomplete, verification procedure for ATL and ATL^* under imperfect information and perfect recall. In more detail, given a concurrent game structure with imperfect information (iCGS) representing a MAS, we build a perfect information abstraction that preserves satisfaction for a three-valued variant of ATL^*. As we show, if the ATL^* specification is true (resp. false) in the (perfect information) abstraction, then it is also true (resp. false) in the original iCGS with imperfect information. On the other hand, if the specification is undefined, we can proceed to refining the abstraction in an attempt to give a defined truth value to the specification. The original problem is undecidable; so no guarantee can be given that by succesive refinements, the property's truth or falsity can ever be established. However, the procedure provides a constructive method to partially model check ATL^* under imperfect information and perfect recall.

Related work. Several approaches for the verification of specifications in ATL and ATL^* under imperfect information and perfect recall have been recently put forward. In one line, restrictions are made on how information is shared amongst the agents, so as to retain decidability [11]. In a related line, interactions amongst agents are limited to public actions only [10, 9]. These approaches are markedly different from ours as they seek to identify classes for which verification is decidable. Instead, we consider the whole class of iCGS and define a general verification procedure. In this sense, our approach is closely related to [8] where a bounded recall method, also incomplete, is defined. However, while in that work perfect recall is approximated, here abstraction is carried out on the levels of information.

*This work was accepted and published in the proceedings of Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19).

At the heart of the method we describe is the notion of abstraction and refinement of MAS models, as well as three-valued semantics in modal languages. An abstraction-refinement framework for CTL over the 3-valued semantics was studied in [26, 27] and the case of hierarchical systems is considered in [3]. Moreover, in [17] an abstraction-refinement technique for full μ-calculus is introduced. An abstraction-refinement procedure for network games with perfect information was introduced in [4] and a symbolic abstraction-refinement approach to the solution of two-player games with reachability or safety goals is shown in [1]. Games with incomplete information are studied in [15] by considering only safety goals and, as we do in this paper, abstraction and refinement are used to generate from a game with imperfect information a new one with perfect information. Model checking MAS by abstraction in an epistemic context was originally investigated in [13, 7]. Three-valued abstractions for the verification of ATL properties have also been put forward in [6, 21, 22, 23]. There are, however, considerable differences between these approaches and the one here pursued. In fact, the methods above focus on decidable settings. In [6, 26] ATL^* is interpreted under perfect information; while [21, 22, 23] considers *non-uniform* strategies [25]. In both cases the corresponding model checking problem is decidable. Their aim, therefore, is to speed-up the verification task and not, as we do here, to provide a sound procedure for an undecidable problem. In [20] is shown a multi-valued semantics for ATL^* that is a conservative extension of the classical 2-valued variant. Mainly, they consider the model checking problem for perfect information games but they also refer at the imperfect information case by giving an undecidable result in general and an exponential-time result for singleton coalitions. Finally, in [19] is shown an abstraction method over formulas to make decidable the model checking problem for ATL with imperfect information and perfect recall.

2 Classic Imperfect Information

In this section we introduce a two-valued semantics for the Alternating-time Temporal Logic ATL^* under imperfect information and perfect recall. To fix the notation, we assume that $Ag = \{1, \ldots, m\}$ is the set of agents and AP the set of atomic propositions. Given a set U, $\overline{U}$ denotes its complement. We denote the length of a tuple v as $|v|$, and its i-th element either as v_i or $v.i$. Let $last(v) = v_{|v|}$ be the last element in v. For $i \leq |v|$, let $v_{\geq i}$ be the suffix $v_i, \ldots, v_{|v|}$ of v starting at v_i and $v_{\leq i}$ the prefix $v_1, \ldots, v_i$ of v.

Models for MAS. We begin by giving a formal model for Multi-agent Systems by means of concurrent game structures with imperfect information [2, 18].

Definition 1 (iCGS) *Given sets Ag of agents and AP of atoms, a* concurrent game structure with imperfect information (iCGS) *is a tuple $M = \langle Ag, AP, S, s_0, \{Act_i\}_{i\in Ag}, d, \delta, \{\sim_i\}_{i\in Ag}, V\rangle$ such that :*

- *$S \neq \emptyset$ is a finite set of* states*, with* initial state *$s_0 \in S$.*
- *For every $i \in Ag$, Act_i is a nonempty finite set of* actions*. Let $Act = \bigcup_{i\in Ag} Act_i$ be the set of all actions, and $ACT = \prod_{i\in Ag} Act_i$ the set of all joint actions.*
- *The* protocol function *$d : Ag \times S \to (2^{Act} \setminus \emptyset)$ defines the availability of actions so that for every $i \in Ag$, $s \in S$, (i) $d(i, s) \subseteq Act_i$ and (ii) $s \sim_i s'$ implies $d(i, s) = d(i, s')$.*
- *The* (deterministic) transition function *$\delta : S \times ACT \to S$ assigns a successor state $s' = \delta(s, \vec{a})$ to each state $s \in S$, for every joint action $\vec{a} \in ACT$ such that $a_i \in d(i, s)$ for every $i \in Ag$, that is, $\vec{a}$ is* enabled *at s.*
- *For every $i \in Ag$, $\sim_i$ is a relation of* indistinguishability *between states. That is, given states $s, s' \in S$, $s \sim_i s'$ iff s and s' are observationally indistinguishable for agent i.*
- *$V : S \times AP \to \{\mathrm{tt}, \mathrm{ff}\}$ is the* two-valued labelling function*.*

By Def. 1 an iCGS describes the interactions of a group Ag of agents, starting from the initial state $s_0 \in S$, according to the transition function δ. The latter is constrained by the availability of actions to agents, as specified by the protocol function d. Further, we assume that every agent i has imperfect information of the exact state of the system; so in any state s, i considers epistemically possible all states s' that are i-indistinguishable from s [16]. When every $\sim_i$ is the identity relation, i.e., $s \sim_i s'$ iff $s = s'$, we obtain a standard CGS with perfect information [2]. Hereafter we consider both the class $iCGS$ of all iCGS, and its subclass CGS of all CGS with perfect information.

Given a set $\Gamma \subseteq Ag$ of agents and a joint action $\vec{a} \in ACT$, let $\vec{a}_\Gamma$ and $\vec{a}_{\overline{\Gamma}}$ be two tuples comprising only of actions for the agents in Γ, resp. $\overline{\Gamma}$. We also write $\vec{a}_i$ and $\vec{a}_{\overline{i}}$ for $\vec{a}_{\{i\}}$ and $\vec{a}_{\overline{\{i\}}}$ respectively. Finally, for $\vec{a}$ and $\vec{b}$ in ACT, $(\vec{a}_\Gamma, \vec{b}_{\overline{\Gamma}})$ denotes the joint action where the actions for the agents in Γ (resp. $\overline{\Gamma}$) are taken from $\vec{a}$ (resp. $\vec{b}$).

A history $h \in S^+$ is a finite (non-empty) sequence of states. The indistinguishability relations are extended to histories in a synchronous, pointwise way, i.e., histories $h, h' \in S^+$ are *indistinguishable* for agent $i \in Ag$, or $h \sim_i h'$, iff (i) $|h| = |h'|$ and (ii) for all $j \leq |h|$, $h_j \sim_i h'_j$.

Syntax. To reason about the strategic abilities of agents in iCGS with imperfect information, we use the Alternating-time Temporal Logic ATL^* [2].

Definition 2 (ATL^*) *State (φ) and path (ψ) formulas in ATL^* are defined as follows, where $q \in AP$ and $\Gamma \subseteq Ag$:*

$$\varphi ::= q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle \Gamma \rangle\!\rangle \psi$$
$$\psi ::= \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid (\psi U \psi)$$

Formulas in ATL^* are all and only the state formulas.

As customary, a formula $\langle\langle\Gamma\rangle\rangle\Phi$ is read as "the agents in coalition Γ have a strategy to achieve Φ". The meaning of linear-time operators *next* X and *until* U is standard [5]. Operators $[\![\Gamma]\!]$, *release* R, *finally* F, and *globally* G can be introduced as usual.

Formulas in the ATL fragment of ATL^* are obtained from Def. 2 by restricting path formulas ψ as follows, where φ is a state formula and R is the *release* operator :

$$\psi \quad ::= \quad X\varphi \mid (\varphi U \varphi) \mid (\varphi R \varphi)$$

Hereafter we also consider the fragment of Γ-formulas, i.e., formulas in which the strategic operator $\langle\langle\Gamma\rangle\rangle$ ranges only over some coalition $\Gamma \subseteq Ag$.

Semantics. When giving a semantics to ATL^* formulas we assume that agents are endowed with *uniform strategies* [18], i.e., they perform the same action whenever they have the same information.

Definition 3 (Uniform Strategy with Perfect Recall) *A* uniform strategy with perfect recall *for agent $i \in Ag$ is a function $f_i : S^+ \to Act_i$ such that for all histories $h, h' \in S^+$, (i) $f_i(h) \in d(i, last(h))$; and (ii) if $h \sim_i h'$ then $f_i(h) = f_i(h')$.*

By Def. 3 any strategy for agent i has to return actions that are enabled for i. Also, whenever two histories are indistinguishable for i, then the same action is returned. Notice that, for the case of (perfect information) CGS, condition (ii) is satisfied by any strategy $f_i : S^+ \to Act_i$.

Given an iCGS M, a *path* $p \in S^\omega$ is an infinite sequence $s_1 s_2 \ldots$ of states. Given a joint strategy $F_\Gamma = \{f_i \mid i \in \Gamma\}$, comprising of one strategy for each agent in coalition Γ, a path p is F_Γ-*compatible* iff for every $j \geq 1$, $p_{j+1} = \delta(p_j, \vec{a})$ for some joint action $\vec{a}$ such that for every $i \in \Gamma$, $a_i = f_i(p_{\leq j})$, and for every $i \in \overline{\Gamma}$, $a_i \in d(i, p_j)$. Let $out(s, F_\Gamma)$ be the set of all F_Γ-compatible paths from s.

We can now assign a meaning to ATL^* formulas on iCGS based on a semantics with two truth values : ff and tt.

Definition 4 (Satisfaction) *The two-valued satisfaction relation $\models^2$ for an iCGS M, state $s \in S$, path $p \in S^\omega$, atom $q \in AP$, and ATL^* formula ϕ is defined as follows (clauses for Boolean connectives are immediate and thus omitted) :*

$(M, s) \models^2 q$ *iff* $V(s, q) = \text{tt}$
$(M, s) \models^2 \langle\langle\Gamma\rangle\rangle\psi$ *iff for some F_Γ, for all $p \in out(s, F_\Gamma)$, $(M, p) \models^2 \psi$*
$(M, p) \models^2 \varphi$ *iff* $(M, p_1) \models^2 \varphi$
$(M, p) \models^2 X\psi$ *iff* $(M, p_{\geq 2}) \models^2 \psi$
$(M, p) \models^2 \psi U \psi'$ *iff for some $k \geq 1$, $(M, p_{\geq k}) \models^2 \psi'$, and for all j, $1 \leq j < k \Rightarrow (M, p_{\geq j}) \models^2 \psi$*

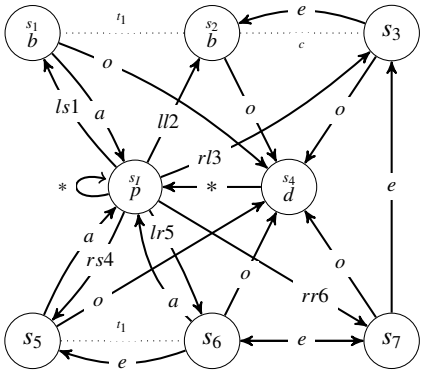


FIGURE 1 – The iCGS M for Example 1. Notice that the transitions are generated with triples of actions. To improve readability every occurrence of the action idle (i) is omitted. Moreover, $*$ denotes any tuple of actions for which a transition is not given explicitly.

We say that formula φ is *true* in an iCGS M, or $M \models^2 \varphi$, iff $(M, s_0) \models^2 \varphi$.

We now state the model checking problem within the two-valued semantics.

Definition 5 (Model Checking) *Given an iCGS M and a formula ϕ, the model checking problem amounts to determine whether $M \models^2 \phi$.*

Since the semantics provided in Def. 4 is the standard interpretation of ATL^* [2, 18], it is well known that model checking ATL, *a fortiori* ATL^*, against iCGS with imperfect information and perfect recall is undecidable [14]. In the rest of the paper we develop methods to obtain partial solutions to this ; but first we illustrate the formal machine above with a toy example.

Example 1 *The iCGS M depicted in Fig. 1 describes a variant of the Train Gate Controller scenario [2]. Two trains t_1 and t_2 pass through a crossroad. Due to agreements between the railway companies, train t_1 can choose between the right (r) or left (l) track, while t_2 can choose between the right (r), left (l) or straight (s) track. At the same time, controller c has to select the right combination of tracks. For example, if t_1 and t_2 choose the joint action rs, then c has to select action* 1 *to proceed to the next step. Moreover, train t_1 has partial observability on the choices of t_2. For instance, if t_1 chooses l, then she cannot distinguish whether t_2 selects r or s, but she would observe if t_2 chose l as well.*

After this first step, c can still change her mind. Specifically, she can change arbitrarily the selection of tracks (e), request a new choice to the trains (a), or execute their selection (o). The controller c has partial observability, she cannot distinguish between s_2 and s_3, i.e. she

does not distinguish r and l of t_1 when t_2 selects l. Finally, we use three atoms, one to denote the initial state (p), one for the preferred selections for t_1 (b), and one to mark that an agreement has been reached amongst the players (d). More formally, the iCGS M is comprised of the agents in $Ag = \{t_1, t_2, c\}$, atoms in $AP = \{p, b, d\}$, states in $S = \{s_I, s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ with initial state s_I, actions in $Ac_{t_1} = \{r, l, i\}$, $Ac_{t_2} = \{r, l, s, i\}$, $Ac_c = \{1, 2, 3, 4, 5, 6, a, e, o, i\}$. Transitions are given as in Fig. 1, and we have the following indistinguishability between different states (indistinguishability is reflexive as well) : $s_1 \sim_{t_1} s_2$, $s_5 \sim_{t_1} s_6$, and $s_2 \sim_c s_3$.

As an example of specifications in ATL, consider the formula $\varphi = \langle\langle\Gamma\rangle\rangle F(b \wedge \neg pUd)$, for $\Gamma = \{t_1, c\}$. This formula can be read as : controller c and train t_1 have a joint strategy such that eventually one of the preferred selections for t_1 is visited, and then an agreement has to be reached before visiting the initial state again. Notice that φ is true in M, while for $\Gamma = \{c\}$, it is false as, whenever t_1 always chooses r and t_2 always chooses s, then controller c cannot make b true before d holds. Finally, consider the ATL formula $\langle\langle Ag\rangle\rangle Fd$, whereby all agents aim at reaching an agreement, thus making the railway work, which can be seen to be true in M. However, given the undecidability of the corresponding model checking problem, there is no general method to verify specifications like these on any given iCGS. Hereafter we provide a sound, albeit partial, method to tackle this problem.*

3 Three-valued Imperfect Information

In this section we introduce a novel generalisation of iCGS in terms of over- and under-approximations. Then, we develop a three-valued semantics for ATL^*, and show that it conservatively extends the two-valued semantics of the previous section. In what follows, for $x = may$ (resp. *must*) we have that $\overline{x} = must$ (resp. *may*).

Definition 6 (Generalized iCGS) *Given sets Ag of agents and AP of atoms, a* generalized iCGS (with imperfect information) *is a tuple $M = \langle Ag, AP, S, s_0, \{Act_i\}_{i\in Ag}, d^{may}, d^{must}, \delta^{may}, \delta^{must}, \{\sim_i\}_{i\in Ag}, V\rangle$ such that :*

1. *S, s_0, $\{Act_i\}_{i\in Ag}$, $\{\sim_i\}_{i\in Ag}$ are defined as in Def. 1.*
2. *d^{may} and d^{must} are* protocol functions *from $Ag \times S$ to $2^{Act} \setminus \emptyset$ such that for every $i \in Ag$ and $s \in S$, (i) $d^{must}(i, s) \subseteq d^{may}(i, s) \subseteq Act_i$ and (ii) $s \sim_i s'$ implies $d^x(i, s) = d^x(i, s')$.*
3. *δ^{may} and δ^{must} are* transition relations *on $S \times ACT \times S$ such that $s' \in \delta^x(s, \vec{a})$ is defined for some $s' \in S$ only if $a_i \in d^x(i, s)$ for every $i \in Ag$. Moreover, $\delta^{must}(s, \vec{a}) \subseteq \delta^{may}(s, \vec{a})$.*
4. *$V : S \times AP \to \{\text{tt}, \text{ff}, \text{uu}\}$ is the* three-valued labelling function.

Intuitively, *must*-components (i.e., d^{must} and δ^{must}) are more restrictive than *may*-components (i.e., d^{may} and δ^{may}) : *must*-transitions can be interpreted as under-approximations of the actual transitions in the iCGS, while *may*-transitions can be thought of as over-approximations. The undefined value uu can be interpreted in various ways, for instance, unknown, unspecified, or inconsistent, depending on the application in hand. This is standard in multi-valued abstraction based methods [26, 6] and we do not discuss this further. We say that the truth value τ is *defined* whenever $\tau \neq$ uu. In the case that under- and over-approximations coincide, i.e., $d^{may} = d^{must}$ and $\delta^{may} = \delta^{must}$, and the truth value of every atom is defined, then we have a standard iCGS as per Def. 1. On the other hand, if each equivalence relation $\sim_i$ is the identity, then we have a generalized CGS (with perfect information).

Next, we introduce *must*- and *may*-strategies.

Definition 7 (Uniform x-Strategy with Perfect Recall) *For $x \in \{may, must\}$, a* uniform *x*-strategy with perfect recall *for agent $i \in Ag$ is a function $f_i^x : S^+ \to Act_i$ such that for every history $h, h' \in S^+$, (i) $f_i^x(h) \in d^x(i, last(h))$; and (ii) if $h \sim_i h'$ then $f_i^x(h) = f_i^x(h')$.*

Here we distinguish between *may* and *must* strategies to over- and under-approximate the strategic abilities of agents. Again, the distinction collapses in the case of standard (two-valued) iCGS.

For $x \in \{may, must\}$ and a joint strategy $F_\Gamma^x = \{f_i^x \mid i \in \Gamma\}$, a path $p \in S^\omega$ is *F_Γ^x-compatible* iff for every $j \geq 1$, $p_{j+1} = \delta^{\overline{x}}(p_j, \vec{a})$ for some joint action $\vec{a}$ such that for every $i \in \Gamma$, $a_i = f_i^x(p_{\leq j})$, and for every $i \notin \Gamma$, $a_i \in d^{\overline{x}}(i, p_j)$. Then, let $out(s, F_\Gamma^x)$ be the set of all F_Γ^x-compatible paths starting from s. We report full definitions in Table 1.

Intuitively, when computing the outcomes of a joint strategy F_Γ^{must} from state s, we adopt a "conservative" stance with respect to the abilities of agents in Γ, by considering only actions enabled according to the under-approximated protocol d^{must}, as well as an "optimistic" stance about the capabilities of agents in $\overline{\Gamma}$, as given by the over-approximated protocol d^{may} and transition δ^{may}. For $out(s, F_\Gamma^{may})$ the reasoning is symmetric (notice that it might be empty in general). This modelling choice is in line with similar three-valued semantics for logics of strategies [6, 23].

Formally we define the three-valued semantics for ATL^* as follows.

Definition 8 (Satisfaction) *The three-valued satisfaction relation $\models^3$ for an iCGS M, state $s \in S$, path $p \in S^\omega$, atom $q \in AP$, $v \in \{\text{tt}, \text{ff}\}$, and ATL^* formula ϕ is defined as in Table 2. In all other cases the value of ϕ is* uu.

Observe that, in the clauses for ATL^* operators *must*-strategies are used to check the truth of formulas, while

$$out(s, F^{must}_{\Gamma}) = \{p \in S^{\omega} \mid for\ all\ j \geq 0, p_{j+1} \in \delta^{may}(p_j, (F^{must}_{\Gamma}(p_{\leq j}), \vec{a}_{\overline{\Gamma}}))\ and\ for\ all\ i \in \overline{\Gamma}, a_i \in d^{may}(i, p_j)\}$$
$$out(s, F^{may}_{\Gamma}) = \{p \in S^{\omega} \mid for\ all\ j \geq 0, p_{j+1} \in \delta^{must}(p_j, (F^{may}_{\Gamma}(p_{\leq j}), \vec{a}_{\overline{\Gamma}}))\ and\ for\ all\ i \in \overline{\Gamma}, a_i \in d^{must}(i, p_j)\}$$

Table 1 – The definitions of $out(s, F^{must}_{\Gamma})$ and $out(s, F^{may}_{\Gamma})$.

$((M,s) \models^3 q) = v$	iff $V(s,q) = v$
$((M,s) \models^3 \neg\varphi) = v$	iff $((M,s) \models^3 \varphi) = \neg v$
$((M,s) \models^3 \varphi \wedge \varphi') = \mathrm{tt}$	iff $((M,s) \models^3 \varphi) = \mathrm{tt}$ and $((M,s) \models^3 \varphi') = \mathrm{tt}$
$((M,s) \models^3 \varphi \wedge \varphi') = \mathrm{ff}$	iff $((M,s) \models^3 \varphi) = \mathrm{ff}$ or $((M,s) \models^3 \varphi') = \mathrm{ff}$
$((M,s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \mathrm{tt}$	iff for some F^{must}_{Γ}, for all $p \in out(s, F^{must}_{\Gamma})$, $((M,p) \models^3 \psi) = \mathrm{tt}$
$((M,s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \mathrm{ff}$	iff for every F^{may}_{Γ}, for some $p \in out(s, F^{may}_{\Gamma})$, $((M,p) \models^3 \psi) = \mathrm{ff}$
$((M,p) \models^3 \varphi) = v$	iff $((M,p_1) \models^3 \varphi) = v$
$((M,p) \models^3 \neg\psi) = v$	iff $((M,p) \models^3 \psi) = \neg v$
$((M,p) \models^3 \psi \wedge \psi') = \mathrm{tt}$	iff $((M,p) \models^3 \psi) = \mathrm{tt}$ and $((M,p) \models^3 \psi') = \mathrm{tt}$
$((M,p) \models^3 \psi \wedge \psi') = \mathrm{ff}$	iff $((M,p) \models^3 \psi) = \mathrm{ff}$ or $((M,p) \models^3 \psi') = \mathrm{ff}$
$((M,p) \models^3 X\psi) = v$	iff $((M,p_{\geq 2}) \models^3 \psi) = v$
$((M,p) \models^3 \psi U \psi') = \mathrm{tt}$	iff for some $k \geq 1$, $((M,p_{\geq k}) \models^3 \psi') = \mathrm{tt}$, and for all j, $1 \leq j < k \Rightarrow ((M,p_{\geq j}) \models^3 \psi) = \mathrm{tt}$
$((M,p) \models^3 \psi U \psi') = \mathrm{ff}$	iff for all $k \geq 1$, $((M,p_{\geq k}) \models^3 \psi') = \mathrm{ff}$, or for some $j \geq 1$, $((M,p_{\geq j}) \models^3 \psi) = \mathrm{ff}$, and for all j', $1 \leq j' \leq j \Rightarrow ((M,p_{\geq j'}) \models^3 \psi') = \mathrm{ff}$

Table 2 – The three-valued satisfaction relation for ATL^*.

may-strategies appear in the clauses for falsehood. Specifically, to check whether $((M,s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \mathrm{tt}$ we consider all paths in $out(s, F^{must}_{\Gamma})$, which are defined by δ^{may}-transitions. This restricts the choices available to coalition Γ, while increasing the number of paths in which the formula needs to be satisfied. Similarly, to verify whether $((M,s) \models^3 \langle\langle\Gamma\rangle\rangle\psi) = \mathrm{ff}$ we need to use δ^{must}-transitions over the paths in $out(s, F^{may}_{\Gamma})$, so as to reduce the number of candidates witnessing the falsehood of the formula. Notice also that, as regards Boolean operators, our semantics correspond to Kleene's three-valued logic.

Finally, $(M \models^3 \varphi) = \mathrm{tt}$ (resp. ff) iff $((M,s_0) \models^3 \varphi) = \mathrm{tt}$ (resp. ff). Otherwise, $(M \models^3 \varphi) = \mathrm{uu}$.

We conclude this section by proving some auxiliary results on conservative extensions and the model checking problem.

Lemma 1 (Conservativeness) *Let M be a standard iCGS, that is, $d^{may} = d^{must}$, $\delta^{may} = \delta^{must}$ are functions, and the truth value of every atom is defined. Then, for every formula ϕ in ATL^*,*

$$((M,s) \models^3 \phi) = \mathrm{tt} \quad \Leftrightarrow \quad (M,s) \models^2 \phi \tag{1}$$
$$((M,s) \models^3 \phi) = \mathrm{ff} \quad \Leftrightarrow \quad (M,s) \not\models^2 \phi \tag{2}$$

By Lemma 1 the three-valued semantics for ATL^* is a conservative extension of its two-valued semantics, as the two coincide whenever we consider standard iCGS. Thus, from the results in the previous section it immediately follows that model checking ATL^* formulas under the three-valued semantics, with imperfect information and perfect recall is also undecidable. However, for perfect information we can show the following.

Theorem 1 *The model checking problem for generalized CGS (with perfect information) is 2EXPTIME-complete for ATL^* and PTIME-complete for ATL.*

In the following section we leverage on the decidable model checking problem for the three-valued semantics under perfect information to develop a sound, albeit incomplete, abstraction-based method to verify imperfect information.

4 Abstraction

We now define perfect information, three-valued abstractions for iCGS. Then, we show that defined truth values for ATL^* formulas transfer from such abstractions to the original iCGS with imperfect information. Since the model checking problem on the former is decidable (as per Theorem 1), this preservation result can be used to define a sound, albeit partial, verification procedure under imperfect information and perfect recall.

To begin with, given a coalition $\Gamma \subseteq Ag$ of agents, define the *common knowledge relation* $\sim^C_{\Gamma}$ as the reflexive and transitive closure $(\bigcup_{i \in \Gamma} \sim_i)^*$ of the union of indistinguishability relations $\sim_i$ for $i \in \Gamma$ [16]. That is, $s \sim^C_{\Gamma} s'$ iff s' is

reachable from s by a sequence $s_1, \ldots, s_n$ of states such that (i) $s_1 = s$, (ii) $s_n = s'$, and (iii) for every $j < n$, $s_j \sim_i s_{j+1}$ for some $i \in \Gamma$. Clearly, $\sim^C_\Gamma$ is an equivalence relation. Now, let $[s]_\Gamma = \{s' \in S \mid s' \sim_\Gamma s\}$ be the equivalence class of s according to $\sim_\Gamma$. The relation $\sim_\Gamma$ is extended to histories in a synchronous, pointwise way, i.e., given $h, h' \in S^+$, $h \sim_\Gamma h'$ iff (i) $|h| = |h'|$ and (ii) for all $j \leq |h|$, $h_j \sim_\Gamma h'_j$. So, we introduce the notation $[h]_\Gamma = \{h' \in S^+ \mid h' \sim_\Gamma h\}$.

Now, we introduce abstractions for iCGS.

Definition 9 (Abstract CGS) *Given an iCGS $M = \langle Ag, AP, S, s_0, \{Act_i\}_{i\in Ag}, d, \delta, \{\sim_i\}_{i\in Ag}, V\rangle$ and a coalition $\Gamma \subseteq Ag$, the* abstract (generalized) CGS *$M_\Gamma = \langle Ag, AP, S_\Gamma, [s_0]_\Gamma, \{Act_i\}_{i\in Ag}, d^{may}_\Gamma, d^{must}_\Gamma, \delta^{may}_\Gamma, \delta^{must}_\Gamma, V_\Gamma\rangle$ is defined such that :*

1. *$S_\Gamma = \{[s]_\Gamma \mid s \in S\}$ is the set of equivalence classes for all states $s \in S$, with initial state $[s_0]_\Gamma$;*
2. *for every $t, t' \in S_\Gamma$ and joint action $\vec{a}$, $t' \in \delta^{may}_\Gamma(t, \vec{a})$ iff for some $s \in t$ and $s' \in t'$, $\delta(s, \vec{a}) = s'$;*
3. *for every $t, t' \in S_\Gamma$ and joint action $\vec{a}$, $t' \in \delta^{must}_\Gamma(t, \vec{a})$ iff for all $s \in t$ there is $s' \in t'$ such that $\delta(s, \vec{a}) = s'$;*
4. *for $x \in \{may, must\}$, $t \in S_\Gamma$, and $i \in Ag$, $d^x_\Gamma(i, t) = \{a_i \in Act_i \mid \delta^x_\Gamma(t, (a_i, \vec{a_i}))$ is defined for some $\vec{a_i}\}$;*
5. *for $v \in \{\mathrm{tt}, \mathrm{ff}\}$, $p \in AP$, and $t \in S_\Gamma$, $V_\Gamma(t, p) = v$ iff $V(s, p) = v$ for all $s \in t$; otherwise, $V_\Gamma(t, p) = \mathrm{uu}$.*

We now show that the abstraction of an iCGS is indeed a generalized CGS (with perfect information) as defined in Def. 6. In particular, the indistinguishability relation for every $i \in Ag$ is assumed to be the identity relation.

Lemma 2 *For every coalition $\Gamma \subseteq Ag$, any abstraction M_Γ of an iCGS M is a generalized CGS.*

We can now state the main theoretical result in this section, namely if a Γ-formula has a defined truth value in an abstract CGS M_Γ, built on an iCGS M, then the Γ-formula has the same truth value in M.

Theorem 2 *Given an iCGS M, state s, and coalition $\Gamma \subseteq Ag$, for every Γ-formula ϕ in ATL^*, we have that*

$$((M_\Gamma, [s]_\Gamma) \models^3 \phi) = \mathrm{tt} \quad \Rightarrow \quad (M, s) \models^2 \phi \qquad (3)$$

$$((M_\Gamma, [s]_\Gamma) \models^3 \phi) = \mathrm{ff} \quad \Rightarrow \quad (M, s) \not\models^2 \phi \qquad (4)$$

By Theorem 2, a defined answer to the model checking problem w.r.t. abstract, generalized CGS (with perfect information), which is decidable, can be transferred to the concrete, two-valued iCGS (with imperfect information), whose model checking problem is undecidable in general. Obviously, if the returned value is undefined (uu), then no conclusive answer can be drawn.

We illustrate the abstraction procedure with our Train Gate Controller scenario in Example 1.

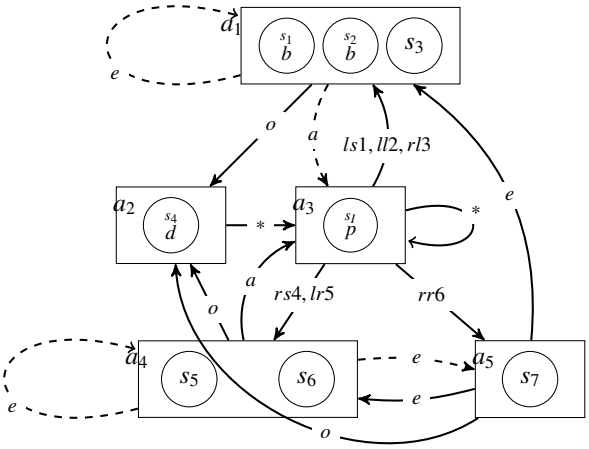


FIGURE 2 – The abstract CGS for the iCGS in Example 1, where *must*-transitions are depicted with continuous lines while *may*-transitions are both the continuous and dashed lines.

Example 2 *In Fig. 2 we show the abstract CGS obtained from the iCGS for the Train Gate Controller scenario in Example 1 by considering the formula $\varphi = \langle\!\langle\Gamma\rangle\!\rangle F(b \wedge \neg pUd)$ for $\Gamma = \{t_1, c\}$. Specifically, the abstraction M_Γ includes five abstract states according to the equivalence relation $\sim^C_{\{t_1,c\}}$. Notice that formula φ is undefined in M_Γ due to the undefined value of atom b in the abstract state a_1.*

5 Refinement

By Theorem 2 if a formula is undefined on abstraction M_Γ, then no conclusion can be drawn on the model checking problem for M. In this section we provide a refinement procedure taking as input a "failure" state s_f in M_Γ and a Γ-formula φ such that φ is undefined in s_f, and returning a refined CGS M^r_Γ, whose state space is smaller than M in general, and for which we are able to prove Theorem 3, a preservation result similar to Theorem 2. In what follows we assume that failure states are identified manually. We leave their automatic generation for further work.

The algorithm *Refinement*(M_Γ, M, s_f) is described in Fig. 3a. Intuitively, we look at incoming transitions into s_f. For concrete states s and s' in s_f, if the Γ-component of actions ending respectively in s and s' are different, any uniform strategy for Γ will visit either s or s'. As a result, the abstract state s_f can be split "safely" into an s- and an s'-component. More precisely, the procedure *Refinement*() begins by initializing as true the values of a matrix m that stores the relation outlined above between the concrete

states in s_f (line 1). Then, the algorithm calls the subroutine $Check_1(M_\Gamma, M, s_f, m)$ in Fig. 3b, which updates the values in m by considering the concrete transition function δ in M. In particular, at each iteration $Check_1()$ considers one predecessor t_f of s_f (line 1). Then, two other loops consider pairs of states s and s' in the abstract state s_f and pairs of states t and t' in the predecessor t_f (lines 2-3). If s and s' are indistinguishable for some agent $i \in \Gamma$ and i performs the same action in the transitions from t and t' to s and s' respectively (lines 4-6), then we update the value of the corresponding cell in m to false (line 6). The subroutine reported in $Check_1()$ carries out the first round of updates on m. Further updates in the *Refinement*() algorithm are performed by the subroutine $Check_2(M_\Gamma, s_f, m, update)$ reported in Fig. 3c, which considers the "indirect" binding that some concrete states may have in an abstract state. Specifically, given the states s and s' in the abstract state s_f that have *true* as value in m (lines 2-3), we need to consider the relation that s and s' have with the other states in s_f (lines 4-6) : if the values in m for both states related with some other state t are *false*, then we update the value of cell $m[s, s']$ to *false* as well. Subroutine $Check_2()$ is called repeatedly in algorithm *Refinement*() as long as guard *update* remains *true*. When *update* becomes $false$, we proceed to check whether there is at least an element *true* in m (line 8). If this is the case, we assign the related concrete states s and s' to two different, new abstract states v and w (line 10). Finally, we populate the new abstract states v and w with the other concrete states in the old abstract state s_f (which is removed) according to matrix m (lines 12-14).

Hereafter we present the formal definition of the refined CGS M^r_Γ as obtained by the application of the *Refinement*() algorithm.

Definition 10 (Refined CGS) *Given an abstract CGS* $M_\Gamma = \langle Ag, AP, S_\Gamma, s_0, \{Act_i\}_{i\in Ag}, d^{may}_\Gamma, d^{must}_\Gamma, \delta^{may}_\Gamma, \delta^{must}_\Gamma, V_\Gamma\rangle$, *its refinement* $M^r_\Gamma = \langle Ag, AP, S^r_\Gamma, s^r_0, \{Act_i\}_{i\in Ag}, d^{may}_\Gamma, d^{must}_\Gamma, \delta^{may}_\Gamma, \delta^{must}_\Gamma, V^r_\Gamma\rangle$ *as obtained by an application of algorithm* $Refinement(M_\Gamma, M, s_f)$ *is defined as follows :*

1. S^r_Γ *is the set* S_Γ *of states in* M_Γ*, possibly without the "failure" state* s_f*, but with the new states added by Refinement(). Then,* s^r_0 *is the state in* S^r_Γ *such that* $s_0 \in s^r_0$*, for* $s_0 \in M$.
2. *For* $x \in \{may, must\}$*, the transitions relations* δ^x_Γ *and the protocol functions* d^x_Γ *are defined as in Def. 9. In particular,*
   (a) *for every* $t, t' \in S^r_\Gamma$ *and joint action* $\vec{a}$, $t' \in \delta^{may}_\Gamma(t, \vec{a})$ *iff for some* $s \in t$ *and* $s' \in t'$, $\delta(s, \vec{a}) = s'$ *;*
   (b) *for every* $t, t' \in S^r_\Gamma$ *and joint action* $\vec{a}$, $t' \in \delta^{must}_\Gamma(t, \vec{a})$ *iff for all* $s \in t$ *there is* $s' \in t'$ *such that* $\delta(s, \vec{a}) = s'$ *;*
   (c) *for every* $t \in S^r_\Gamma$*, and* $i \in Ag$, $d^x_\Gamma(i, t) = \{a_i \in Act_i \mid \delta^x_\Gamma(t, (a_i, \vec{a}_{\bar{i}}))$ *is defined for some* $\vec{a}_{\bar{i}}\}$.
3. *For* $v \in \{\text{tt}, \text{ff}\}$, $p \in AP$*, and* $t \in S^r_\Gamma$, $V^r_\Gamma(t, p) = v$ *iff* $V(s, p) = v$ *for all* $s \in t$ *; otherwise,* $V_\Gamma(s, p) = \text{uu}$.

Algorithm *Refinement*(M_Γ, M, s_f) :

```
1  for s, s' ∈ s_f, m[s, s'] = true;
2  Check_1(M_Γ, M, s_f, m);
3  update = true;
4  while update = true
5  Check_2(M_Γ, s_f, m, update);
6  split = false;
7  while s, s' ∈ s_f and split = false
8  if m[s, s'] = true then
9  remove(s_f, S_Γ);
10 add(v, S_Γ); add(w, S_Γ); add(s, v); add(s', w);
11 split = true;
12 for t ∈ s_f
13 if m[s, t] = true then add(t, w);
14 else add(t, v);
```

(a)

Algorithm $Check_1(M_\Gamma, M, s_f, m)$:

```
1  for t_f ∈ Pre(s_f)
2  for s, s' ∈ s_f
3  for t, t' ∈ t_f
4  if δ(t, a⃗) = s and δ(t', b⃗) = s' then
5  for i ∈ Γ
6  if s ∼_i s' and a⃗_i = b⃗_i then m[s, s'] = false;
```

(b)

Algorithm $Check_2(M_\Gamma, s_f, m, update)$:

```
1  update = false;
2  for s, s' ∈ s_f
3  if m[s, s'] = true then
4  for t ∈ s_f
5  if m[s, t] = false and m[s', t] = false then
6  m[s, s'] = false;
7  update = true;
```

(c)

FIGURE 3 – The *Refinement procedure (3a) with its auxiliary subroutines* $Check_1$ *and* $Check_2$ *(3b and 3c respectively).*

By Def. 10 the components of the refined CGS M^r_Γ coincide with those in abstraction M_Γ, except possibly as regards the "failure" state s_f and new states introduced by *Refinement*(). On the new states, the transition relations and protocol functions are defined in analogy with M_Γ.

We now show a property of the refined CGS M^r_Γ, which will be useful to prove the main preservation result Theorem 3. Intuitively, must strategies in M^r_Γ respect uniformity *on the set of their outcomes*.

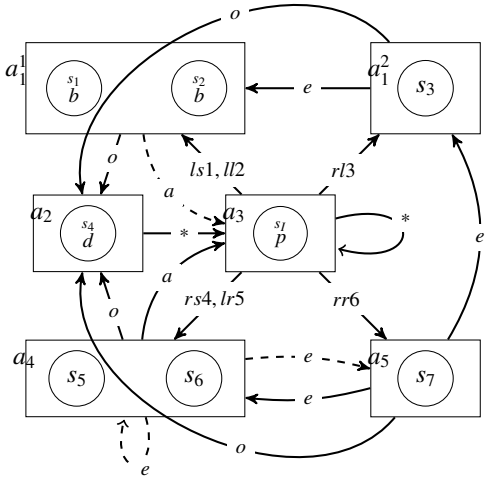


FIGURE 4 – Example of split of abstract CGS in Fig. 2.

Lemma 3 *In M^r_Γ for every joint strategy F^{must}_Γ, for all $p, \hat{p} \in out(t, F^{must}_\Gamma)$, all $p' \in p$, $\hat{p}' \in \hat{p}$, and all $i \in \Gamma$, $j \in \mathbb{N}$, if $p'_{\leq j} \sim_i \hat{p}'_{\leq j}$ then $f^{must}_i(p_{\leq j}) = f^{must}_i(\hat{p}_{\leq j})$.*

By Lemma 3 we can prove the main preservation result of this section. In particular, the lemma is used in the inductive step for strategy operators.

Theorem 3 *Given an iCGS M, state s, coalition Γ, its abstract CGS M_Γ with refinement M^r_Γ, and state $s^r_\Gamma \ni s$, for every Γ-formula ϕ in ATL^*,*

$$((M^r_\Gamma, s^r_\Gamma) \models^3 \phi) = \mathrm{tt} \quad \Rightarrow \quad (M, s) \models^2 \phi \qquad (5)$$

$$((M^r_\Gamma, s^r_\Gamma) \models^3 \phi) = \mathrm{ff} \quad \Rightarrow \quad (M, s) \not\models^2 \phi \qquad (6)$$

By Theorem 3 defined truth values are preserved from the refined CGS to the original iCGS, similarly to Theorem 2.

Example 3 *In Fig. 4 we present a refinement of the abstract CGS in Fig. 2. In this new model we split the state a_1 in two new abstract states a^1_1 and a^2_1 according to the Refinement() algorithm in Fig. 3a. By doing so, formula $\varphi = \langle\!\langle \{t_1, c\} \rangle\!\rangle F(b \wedge \neg p U d)$ becomes true as atom b becomes defined in a^1_1 and a^2_1. So, by Theorem 3 formula φ is true in the original iCGS M as well.*

By combining the results in Section 3, 4, and 5 we can outline a method to verify strategic properties of Multi-Agent Systems under the assumptions of imperfect information and perfect recall. Given an iCGS M and a Γ-formula ϕ in ATL^*, we first build the abstract, three-valued CGS M_Γ as per Def. 9. We can model check ϕ on M_Γ, as the corresponding decision problem is decidable by Theorem 1, and then tranfer any defined answer to the original iCGS M in virtue of Theorem 2. In case of an undefined answer, we can apply the refinement procedure in Section 5 iteratively : if the value of ϕ or any of its subformulas is undefined at some state s_f in M_Γ, we can apply the refinement algorithm so as to obtain a refined CGS M^r_Γ : any defined value for ϕ on M^r_Γ transfers to M by Theorem 3. The refinement step can be iterated as long as ϕ stays undefined. Since the verification of ATL^* under imperfect information and perfect recall is undecidable in general [14], the procedure here outlined is obviously partial and there is no guarantee of termination with a defined answer. However, partial results can be useful in cases of interest, like the Train Gate Controller scenario illustrated in Example 1, 2, and 3.

6 Conclusions

As we discussed in the introduction one of the key issues in employing logics for strategic reasoning, such as ATL and ATL^*, in the context of Multi-agent Systems is that their model checking problem is undecidable under perfect recall and incomplete information. Yet, this is one of the most natural and compelling setup in applications. Finding appropriate approximations remains an open problem at present.

In this paper we have put forward a notion of abstraction between different classes of systems to overcome this difficulty. Specifically, we showed that iCGS with imperfect information admit a (perfect information) abstraction which preserves satisfaction back to the original model, when checked under a three-valued semantics. This enabled us to give an incomplete but sound procedure for the original model checking problem, which is undecidable in general.

In future work we intend to build a toolkit to generate abstractions and refinements automatically, perhaps in combination with refinement techniques built on interpolants [6]. Moreover, we plan to extend the abstraction and refinement techniques here developed to more expressive languages for strategic reasoning including Strategy Logic [12, 24].

Références

[1] Alfaro, Luca de et Pritam Roy: *Solving games via three-valued abstraction refinement*. Inf. Comput., 208(6) :666–676, 2010.

[2] Alur, R., T.A. Henzinger et O. Kupferman: *Alternating-time temporal logic*. J. ACM, 49(5) :672–713, 2002.

[3] Aminof, Benjamin, Orna Kupferman et Aniello Murano: *Improved model checking of hierarchical systems*. Inf. Comput., 210 :68–86, 2012.

[4] Avni, Guy, Shibashis Guha et Orna Kupferman: *An Abstraction-Refinement Methodology for Reasoning about Network Games*. Dans *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 70–76, 2017.

[5] Baier, C. et J. P. Katoen: *Principles of Model Checking (Representation and Mind Series)*. 2008, ISBN 026202649X, 9780262026499.

[6] Ball, Thomas et Orna Kupferman: *An abstraction-refinement framework for multi-agent systems*. Dans *Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science (LICS06)*, pages 379–388. IEEE, 2006.

[7] Belardinelli, F. et A. Lomuscio: *A Three-value Abstraction Technique for the Verification of Epistemic Properties in Multi-agent Systems*. Dans *Proc. of the 15th European Conference on Logics in Artificial Intelligence (JELIA16)*, 2016.

[8] Belardinelli, F., A. Lomuscio et V. Malvone: *Approximating Perfect Recall when Model Checking Strategic Abilities*. Dans *Proceedings of the 16th International Conference on Principles of Knowledge Representation and Reasoning (KR2018)*, 2018.

[9] Belardinelli, F., A. Lomuscio, A. Murano et S. Rubin: *Verification of Broadcasting Multi-Agent Systems against an Epistemic Strategy Logic*. Dans *IJCAI'17*, pages 91–97, 2017.

[10] Belardinelli, F., A. Lomuscio, A. Murano et S. Rubin: *Verification of Multi-agent Systems with Imperfect Information and Public Actions*. Dans *AAMAS 2017*, pages 1268–1276, 2017.

[11] Berthon, Raphaël, Bastien Maubert, Aniello Murano, Sasha Rubin et Moshe Y. Vardi: *Strategy logic with imperfect information*. Dans *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017, ISBN 978-1-5090-3018-7.

[12] Chatterjee, K., T. Henzinger et N. Piterman: *Strategy Logic*. Dans *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR07)*, tome 4703, pages 59–73, 2007.

[13] Cohen, M., M. Dam, A. Lomuscio et F. Russo: *Abstraction in model checking multi-agent systems*. Dans *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS09)*, pages 945–952. IFAAMAS Press, 2009.

[14] Dima, C. et F.L. Tiplea: *Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable*. CoRR, abs/1102.4225, 2011.

[15] Dimitrova, Rayna et Bernd Finkbeiner: *Abstraction Refinement for Games with Incomplete Information*. Dans *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2008, December 9-11, 2008, Bangalore, India*, pages 175–186, 2008.

[16] Fagin, R., J.Y. Halpern, Y. Moses et M.Y. Vardi: *Reasoning about Knowledge.* MIT, 1995.

[17] Grumberg, Orna, Martin Lange, Martin Leucker et Sharon Shoham: *When not losing is better than winning : Abstraction and refinement for the full mu-calculus*. Inf. Comput., 205(8) :1130–1148, 2007.

[18] Jamroga, W. et W. van der Hoek: *Agents that Know How to Play*. Fund. Inf., 62 :1–35, 2004.

[19] Jamroga, Wojciech, Michał Knapik et Damian Kurpiewski: *Fixpoint Approximation of Strategic Abilities under Imperfect Information*. Dans *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 1241–1249, 2017.

[20] Jamroga, Wojciech, Beata Konikowska et Wojciech Penczek: *Multi-Valued Verification of Strategic Ability*. Dans *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, pages 1180–1189, 2016.

[21] Lomuscio, A. et J. Michaliszyn: *An abstraction technique for the verification of multi-agent systems against ATL specifications*. Dans *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR14)*, pages 428–437. AAAI Press, 2014.

[22] Lomuscio, A. et J. Michaliszyn: *Verifying Multi-Agent Systems by Model Checking Three-valued Abstractions*. Dans *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS15)*, pages 189–198, 2015.

[23] Lomuscio, A. et J. Michaliszyn: *Verification of Multi-Agent Systems via Predicate Abstraction against ATLK specifications*. Dans *Proc. of the 15th Int. Conference on Autonomous Agents and Multiagent Systems (AAMAS16)*, pages 662–670, 2016.

[24] Mogavero, F., A. Murano, G. Perelli et M.Y. Vardi: *Reasoning About Strategies : On the Model-Checking Problem*. ACM Trans. Comp. Log., 15(4) :34 :1–34 :47, 2014.

[25] Raimondi, F. et A. Lomuscio: *The complexity of symbolic model checking temporal-epistemic logics*.

Dans *Proceedings of Concurrency, Specification & Programming (CS&P)*, pages 421–432. Warsaw University, 2005.

[26] Shoham, S. et O. Grumberg: *Monotonic Abstraction-Refinement for CTL*. Dans *Proceedings of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS04)*, tome 2988 de *Lecture Notes in Computer Science*, pages 546–560. Springer, 2004.

[27] Shoham, Sharon et Orna Grumberg: *A game-based framework for CTL counterexamples and 3-valued abstraction-refinement*. ACM Trans. Comput. Log., 9(1) :1, 2007.

Actes JIAF 2019

Consistency Measures, Inconsistency Measures, and Mix Measures (Preliminary Report)

Philippe Besnard[1] **Vincent Risch**[2]

[1] IRIT, CNRS, Univ. Toulouse, France
[2] LIS, Université Aix-Marseille, France
besnard@irit.fr vincent.risch@univ-amu.fr

Résumé

Nous proposons une tentative d'exploration du concept de mesures de cohérence. Par celles-ci, il s'agit d'attribuer un degré de cohérence à des ensembles finis de formules logiques, comme un pendant au concept bien connu de mesures d'incohérence qui attribuent un degré d'incohérence à des ensembles finis de formules logiques. Nous introduisons un ensemble primitif de postulats pour des mesures de cohérence. Nous nous penchons sur quelques correspondances avec les mesures d'incohérence. Nous posons également les bases d'une dualité entre les deux univers. Finalement, nous examinons de façon préliminaire ce que pourrait être une mesure mixte, à savoir, une mesure qui détermine un degré, sur un même référentiel, pour la cohérence (valeur positive) ainsi que pour l'incohérence (valeur négative). Nous abordons aussi, en comparaison, la question des super-modèles de Ginsberg et col., ainsi que ce qui peut en être considéré comme une généralisation, les morpho-logiques.

Abstract

We give some insight into a preliminary attempt at investigating a notion of consistency measures. These would provide a consistency degree for any finite collection of logical formulas, on a par with the well-known notion of inconsistency measures, that aim at assigning degrees of inconsistency to finite sets of logical formulas. We first propose a basic set of postulates for consistency measures. We look at a couple of relationships with inconsistency measures. We even lay grounds for a formal duality between these two domains. Lastly, we have a look at what would be a mix measure, that is, a measure that gives a degree, on the same scale, for consistency (positive values) and inconsistency (negative values). We also mention supermodels, as defined by Ginsberg et al., as well as a theory that can be regarded as a generalization of super-models, namely morpho-logics.

1 Introduction

In some sense, a formula can be viewed as more consistent than another. An illustration on such an idea is as follows. For a formula φ and a propositional variable a, define

$$\nu a.\varphi \stackrel{\text{def}}{=} \varphi[\top \leftarrow a] \wedge \varphi[\bot \leftarrow a].$$

Then, assuming $\mathsf{Var}(\varphi) = \{a_1, \ldots, a_n\}$, and ρ a permutation (for the a_i's), an example of a "consistency measure" is

$$\boldsymbol{C}(\varphi) = \inf\{k \mid \exists \rho \text{ s.t. } \nu a_{\rho(1)} \cdots a_{\rho(k)}.\varphi \vdash \bot\}.$$

In a more explicit way, $\min\{k \mid \exists \rho \text{ s.t. } \nu a_{\rho(1)} \cdots a_{\rho(k)}.\varphi \vdash \bot\}$ is the value of $\boldsymbol{C}(\varphi)$ in the case that there exist k and ρ such that $\nu a_{\rho(1)} \cdots a_{\rho(k)}.\varphi$ is inconsistent. Otherwise, $\boldsymbol{C}(\varphi) = \infty$.

Example 1 *In view of* $\nu a.\neg a = (\neg\top \wedge \neg\bot)$*, it holds that* $\boldsymbol{C}(\neg a) = 1$*. Similarly,* $\boldsymbol{C}(a) = 1$*. However,* $\nu a.(a \vee \neg a) = (\top \vee \neg\top) \wedge (\bot \vee \neg\bot)$ *so that* $\boldsymbol{C}(a \vee \neg a) = \infty$.

Generalizing the above idea, a consistency measure is a total function that maps every finite set of formulas K to a value in $\mathbb{R}^+ \cup \{\infty\}$.

2 Preliminaries

All the formal matters will refer to propositional logic $\vdash$ with a language $\mathscr{L}$ based on a set of propositional variables denoted $\mathsf{Var}(\mathscr{L})$ as well as the propositional constants $\bot$ and $\top$. The symbols for the connectives are $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (material implication). Logical equivalence (not the equivalence connective) is denoted by means of $\equiv$. For the sake of clarity, we write & (and), $\Rightarrow$ (if ... then), $\Leftrightarrow$ (if and only if) in meta-level statements where $\varphi, \psi, \ldots$ denote formulas of $\mathscr{L}$ while K, K',

... are called belief bases and denote finite sets of formulas of $\mathscr{L}$. As a final word about notation, $\mathcal{K}_{\mathscr{L}}$ is comprised of all belief bases over $\mathscr{L}$.

For our purposes, a consistency measure is a map

$$\boldsymbol{C} : \mathcal{K}_{\mathscr{L}} \to \mathbb{R}^+ \cup \{\infty\}.$$

The paper is organized as follows: in section 3 below, basic postulates for consistency measures are proposed, and the duality with inconsistency measures is investigated. In section 4, the notion of mixed measure is introduced. Section 5.1 exhibits a link with supermodels such as defined by Ginsberg et al.

3 Postulates for consistency measures

Not all such functions $\boldsymbol{C}$ can do! A list of requirements over $\boldsymbol{C}$ is needed. To this end, postulates can ensure $\boldsymbol{C}$ *to make sense* for the purpose of consistency measuring.

This section is an investigation into such requirements as postulates formulated on the following grounds:

— The context is classical logic $\vdash$ over a language $\mathscr{L}$.

— Belief bases are finite sets of formulas of $\mathscr{L}$.

— $\boldsymbol{C}$ maps all finite sets of formulas of $\mathscr{L}$ to values in $\mathbb{R}^+ \cup \{\infty\}$.

The first postulates that come to mind are those specifying what cases must be ascribed the lowest, respectively highest, consistency degree. Thus arise the following postulates.

Inconsistency Null

$$\boldsymbol{C}(K) = 0 \quad \Leftrightarrow \quad K \vdash \bot$$

Inconsistent bases, and only them, have the lowest consistency degree: 0.

Tautology Top

$$\vdash K \quad \Rightarrow \quad \boldsymbol{C}(K) \geq \boldsymbol{C}(K')$$

Tautologies have the highest consistency degree.

Equivalence

$$K \equiv K' \quad \Rightarrow \quad \boldsymbol{C}(K) = \boldsymbol{C}(K')$$

Logically equivalent bases have the same consistency degree. [1]

Variant Equality

$$\sigma K = K' \ \ \& \ \ \sigma' K = K \quad \Rightarrow \quad \boldsymbol{C}(K) = \boldsymbol{C}(K')$$

Renaming does not change the consistency degree.

1. Due to Inconsistency Null, introducing the extra condition $K \nvdash \bot$ (or, equivalently, $K' \nvdash \bot$) would be otiose.

Proposition 1 *Inconsistency Null is equivalent to the conjunction of the conditions below.*

$$(\forall K' \ \boldsymbol{C}(K) \leq \boldsymbol{C}(K')) \quad \Rightarrow \quad K \vdash \bot \tag{1}$$

$$K \vdash \bot \quad \Rightarrow \quad \boldsymbol{C}(K) \leq \boldsymbol{C}(K') \tag{2}$$

$$\exists K \ \boldsymbol{C}(K) = 0 \tag{3}$$

Two (equivalent in the context of (1)–(3)) consequences are:

$$K \vdash \bot \ \ \& \ \ K' \nvdash \bot \quad \Rightarrow \quad \boldsymbol{C}(K) < \boldsymbol{C}(K') \tag{4}$$

$$K \nvdash \bot \quad \Rightarrow \quad 0 < \boldsymbol{C}(K) \tag{5}$$

Proof Let us start by showing that Inconsistency Null entails (1)–(3). To start with, (3) is a trivial consequence of Inconsistency Null. As regards (2), it is a direct consequence of Inconsistency Null because the codomain of $\boldsymbol{C}$ is $\mathbb{R}^+ \cup \{\infty\}$. Now, in order to prove (1), assume $\forall K' \ \boldsymbol{C}(K) \leq \boldsymbol{C}(K')$. By (3), $\boldsymbol{C}(K_0) = 0$ for some K_0. Due to Inconsistency Null, $K_0 \vdash \bot$. An instance of (2) is $K \vdash \bot \Rightarrow \boldsymbol{C}(K) \leq \boldsymbol{C}(K_0)$ which gives $K \vdash \bot \Rightarrow \boldsymbol{C}(K) \leq 0$ hence $K \vdash \bot \Rightarrow \boldsymbol{C}(K) = 0$ (the codomain of $\boldsymbol{C}$ is $\mathbb{R}^+ \cup \{\infty\}$). Applying Inconsistency Null, $K \vdash \bot$ and (1) is proven. Let us now show that (1)–(3) entail Inconsistency Null. An obvious consequence of (1)–(2) is $\big(\forall K' \ \boldsymbol{C}(K) \leq \boldsymbol{C}(K')\big) \Leftrightarrow K \vdash \bot$ i.e. $\boldsymbol{C}(K) = \min\{\boldsymbol{C}(K') \mid K' \subseteq_f \mathscr{L}\} \Leftrightarrow K \vdash \bot$. [2] Now, (3) means that $\min\{\boldsymbol{C}(K') \mid K' \subseteq_f \mathscr{L}\} = 0$ since the codomain of $\boldsymbol{C}$ is $\mathbb{R}^+ \cup \{\infty\}$. Therefore, $\big(\forall K' \ \boldsymbol{C}(K) \leq \boldsymbol{C}(K')\big) \Leftrightarrow K \vdash \bot$ is equivalent with Inconsistency Null.

We now show that (1) and (2) imply (4). Due to (1), $\big(\forall K'' \ \boldsymbol{C}(K') \leq \boldsymbol{C}(K'')\big) \Rightarrow K' \vdash \bot$. In contrapositive form, $K' \nvdash \bot \Rightarrow \exists K'' \ \boldsymbol{C}(K') > \boldsymbol{C}(K'')$. Since an immediate instance of (2) is $K \vdash \bot \Rightarrow \boldsymbol{C}(K) \leq \boldsymbol{C}(K'')$, it ensues that $K \vdash \bot \ \& \ K' \nvdash \bot$ gives $\exists K'' \ \boldsymbol{C}(K'') < \boldsymbol{C}(K')$ as well as $\forall K'' \ \boldsymbol{C}(K) \leq \boldsymbol{C}(K'')$ hence $\exists K'' \ \boldsymbol{C}(K) \leq \boldsymbol{C}(K'') < \boldsymbol{C}(K')$. Summing it up, $K \vdash \bot \ \& \ K' \nvdash \bot \Rightarrow \boldsymbol{C}(K) < \boldsymbol{C}(K')$.

Lastly, we show that (4) and (5) are equivalent in the context of (1)–(3). Assume first that $\boldsymbol{C}$ satisfies $K \vdash \bot \ \& \ K' \nvdash \bot \Rightarrow \boldsymbol{C}(K) < \boldsymbol{C}(K')$. However, $\{\bot\} \vdash \bot$. Accordingly, $K' \nvdash \bot \Rightarrow \boldsymbol{C}(\{\bot\}) < \boldsymbol{C}(K')$. Since the codomain of $\boldsymbol{C}$ is $\mathbb{R}^+ \cup \{\infty\}$, it then follows that $K' \nvdash \bot \Rightarrow 0 < \boldsymbol{C}(K')$. As to the converse, assume that $\boldsymbol{C}$ satisfies $K' \nvdash \bot \Rightarrow 0 < \boldsymbol{C}(K')$. A consequence of (2) and (3) (together with the fact that the codomain of $\boldsymbol{C}$ is $\mathbb{R}^+ \cup \{\infty\}$) is $K \vdash \bot \Rightarrow \boldsymbol{C}(K) = 0$. It follows that $K \vdash \bot \ \& \ K' \nvdash \bot \Rightarrow \boldsymbol{C}(K) < \boldsymbol{C}(K')$. ■

Conditions (1)–(2) express that inconsistent formulas get the lowest consistency degree and induce that consistent formulas are ascribed a strictly greater consistency degree.

2. We write $X \subseteq_f Y$ to denote that X is a finite subset of Y.

3.1 Postulate chopping

Do we wish to endorse the general principle that $K' \vdash K$ entails $\boldsymbol{C}(K) \geq \boldsymbol{C}(K')$? Under the reading of "being more consistent" as "having more models", the answer must be positive.

Adopting a positive answer actually amounts to supplementing Inconsistency Null with the following postulate.

Entailment Decrease

$$K \vdash K' \;\; \Rightarrow \;\; \boldsymbol{C}(K) \leq \boldsymbol{C}(K')$$

Proposition 2 *Entailment Decrease entails Tautology Top, Equivalence, Variant Equality, condition (2) from Proposition 1 and the following property.*

$$\sigma K = K' \;\; \Rightarrow \;\; \boldsymbol{C}(K) \geq \boldsymbol{C}(K') \tag{6}$$

Proof The case of Equivalence is trivial. As regards Tautology Top, if $\vdash K$ then $K' \vdash K$. By Entailment Decrease, $\boldsymbol{C}(K') \leq \boldsymbol{C}(K)$. As to Variant Equality, it is a special case of (6) that is taken care of as the last item in this proof. As regards (2), assume $K \vdash \bot$. Thus, $K \vdash K'$. Applying Entailment Decrease, $\boldsymbol{C}(K) \leq \boldsymbol{C}(K')$. As regards (6), assume $\sigma K = K'$. Now, $K' = \sigma K$ entails $K' \equiv K \cup \{a \rightarrow b \mid [b \leftarrow a] \in \sigma\}$. Obviously, $K' \vdash K$. In view of Entailment Decrease, $\boldsymbol{C}(K') \leq \boldsymbol{C}(K)$ ensues. ■

In (6), the significant case is K' consistent (otherwise, $\boldsymbol{C}(K') = 0$ by Inconsistency Null and $\boldsymbol{C}(K) \geq \boldsymbol{C}(K')$ trivially ensues). Thus, (6) expresses that K has a higher (or equal) consistency degree than any instance of K.

Example 2 *Let $K = \{p \vee q\}$ and $K' = \{p \vee p\}$. Thus, $\sigma K = K'$ for $\sigma = [p \leftarrow q]$. In view of (6), $\boldsymbol{C}(K) \geq \boldsymbol{C}(K')$ i.e. $\boldsymbol{C}(p \vee q) \geq \boldsymbol{C}(p \vee p)$.*

Obviously, Entailment Decrease realizes the idea that "having more models" does imply "being more consistent": If $\mathrm{Mod}(K) \subseteq \mathrm{Mod}(K')$ then $\boldsymbol{C}(K) \leq \boldsymbol{C}(K')$. The converse is untrue because $\leq$ is a total order whereas $\subseteq$ (as ranging over models of finite subsets of $\mathscr{L}$) is only a partial order. In particular, for $\mathsf{Var}(K) \cap \mathsf{Var}(K') = \emptyset$, it must still be the case that either $\boldsymbol{C}(K) \leq \boldsymbol{C}(K')$ or $\boldsymbol{C}(K') \leq \boldsymbol{C}(K)$ but neither $\mathrm{Mod}(K) \subseteq \mathrm{Mod}(K')$ nor $\mathrm{Mod}(K') \subseteq \mathrm{Mod}(K)$ hold.

Note. A strict version of Entailment Decrease is of interest, in the form of condition (4) from Proposition 1, that is:

Strict Entailment Decrease

$$K \vdash K' \;\; \& \;\; K' \nvdash K \;\; \Rightarrow \;\; \boldsymbol{C}(K) < \boldsymbol{C}(K')$$

The codomain of a consistency measure might be restricted to $[0, 1]$. Equivalently, such a consistency measure can be viewed as satisfying the following postulate.

Normalization

$$0 \leq \boldsymbol{C}(K) \leq 1$$

Trivially, Normalization suggests alternative postulates, e.g., Tautology Top could be replaced by the following (stronger even in the case that the codomain of $\boldsymbol{C}$ is $[0, 1]$) postulate.

Tautology 1-Top

$$\vdash K \;\; \Rightarrow \;\; \boldsymbol{C}(K) = 1$$

3.2 Relationship with inconsistency measures

Inconsistency measures (they have received a great deal of attention [9]) are meant to indicate to what extent a finite set of formulas K is inconsistent. Formally, an inconsistency measure $\boldsymbol{I}$ maps every finite set of formulas K to a value in $\mathbb{R}^+ \cup \{\infty\}$. Interestingly, various postulates have been proposed for inconsistency measuring, here is the most important one (see the survey [18]).

Consistency Null

$$\boldsymbol{I}(K) = 0 \;\; \Leftrightarrow \;\; K \nvdash \bot$$

No more is needed to show that every pair $(\boldsymbol{C}, \boldsymbol{I})$, *no matter how arbitrary*, conveys some duality between $\boldsymbol{C}$ and $\boldsymbol{I}$:

Proposition 3 *Let $\boldsymbol{C}$ enjoy Inconsistency Null and $\boldsymbol{I}$ enjoy Consistency Null. Then,*

$$\boldsymbol{C}(K) = 0 \;\; \Leftrightarrow \;\; K \vdash \bot \;\; \Leftrightarrow \;\; \boldsymbol{I}(K) > 0$$
$$\boldsymbol{C}(K) > 0 \;\; \Leftrightarrow \;\; K \nvdash \bot \;\; \Leftrightarrow \;\; \boldsymbol{I}(K) = 0$$

Proof ($\boldsymbol{C}(K) = 0 \Rightarrow K \vdash \bot$) Assuming $\boldsymbol{C}(K) = 0$, Inconsistency Null then gives $K \vdash \bot$. ($K \vdash \bot \Rightarrow \boldsymbol{I}(K) > 0$) Assuming $K \vdash \bot$, Consistency Null gives $\boldsymbol{I}(K) \neq 0$. As the codomain of $\boldsymbol{I}$ is $\mathbb{R}^+ \cup \{\infty\}$, it follows that $\boldsymbol{I}(K) > 0$. ($\boldsymbol{I}(K) > 0 \Rightarrow \boldsymbol{C}(K) = 0$) Assuming $\boldsymbol{I}(K) > 0$, Consistency Null gives $K \vdash \bot$. Applying Inconsistency Null, $\boldsymbol{C}(K) = 0$.

$\boldsymbol{C}(K) > 0 \Leftrightarrow K \nvdash \bot \Leftrightarrow \boldsymbol{I}(K) = 0$ is proven in a similar way. ■

The *Tautology Top* postulate is to be put in regard with the following property (induced from Consistency Null and the fact that $\boldsymbol{I}$ has codomain $\mathbb{R}^+ \cup \{\infty\}$):

$$K \nvdash \bot \;\; \Rightarrow \;\; \boldsymbol{I}(K) \leq \boldsymbol{I}(K')$$

That is, the lowest value for $\boldsymbol{I}$ holds for the case that K is consistent. Duality appears as *Tautology Top* actually means that the highest value for $\boldsymbol{C}$ holds for K being tautological, intuitively the most extreme [3] form of consistency:

$$\underbrace{K \in \max_{\geq_{cons}}\{K \mid K \nvdash \bot\}}_{\vdash K} \;\; \Rightarrow \;\; \boldsymbol{C}(K) \geq \boldsymbol{C}(K')$$

3. We use $\geq_{cons}$ to intuitively denote a pre-order for "is at least as consistent as".

According to this scheme, it is expected that the following consequence of Inconsistency Null

$$K \vdash \bot \;\Rightarrow\; \boldsymbol{C}(K) \leq \boldsymbol{C}(K')$$

gets a dual so that the highest value for $\boldsymbol{I}$ holds for the most extreme [4] form of inconsistency:

$$\underbrace{K \in \max_{\geq_{inc}}\{K \mid K \vdash \bot\}}_{???} \;\Rightarrow\; \boldsymbol{I}(K) \geq \boldsymbol{I}(K')$$

Duality breaks down as no such notion of "maximal inconsistent" arises in classical logic.

3.2.1 Formal duality

The aim here is to set a duality formally between postulates (and more generally properties) about consistency measures and inconsistency measures.

Definition 1 *Recursively set* $[A \mathcal{R} B]^\star = A^\star \mathcal{R}^\star B^\star$ *where the relation* $\mathcal{R}$ *as well as the expressions A and B have the following primitive cases:*

$$\begin{array}{ccc}
\boldsymbol{C}(X) & \overset{\star}{\longleftrightarrow} & \boldsymbol{I}(X) \\
\textit{is inconsistent} & \overset{\star}{\longleftrightarrow} & \textit{is consistent} \\
\Rightarrow & \overset{\star}{\longleftrightarrow} & \Rightarrow \\
\leq & \overset{\star}{\longleftrightarrow} & \leq \\
> & \overset{\star}{\longleftrightarrow} & > \\
= & \overset{\star}{\longleftrightarrow} & = \\
\bot & \overset{\star}{\longleftrightarrow} & \bot \\
0 & \overset{\star}{\longleftrightarrow} & 0 \\
K & \overset{\star}{\longleftrightarrow} & K \\
K' & \overset{\star}{\longleftrightarrow} & K' \\
\vdots & &
\end{array}$$

Example 3 *Here are a few examples of dual properties (we write* $K \vdash \bot$ *to stand for "K is inconsistent" but the latter expression is what is used for determining the dual expression, and similarly for* $K \nvdash \bot$ *standing for "K is consistent").*

$$\begin{aligned}
[\boldsymbol{C}(K) = 0 \Rightarrow K \vdash \bot]^\star &= [\boldsymbol{C}(K) = 0]^\star \Rightarrow [K \vdash \bot]^\star \\
&= \boldsymbol{I}(K) = 0 \Rightarrow K \nvdash \bot \\
[K \vdash \bot \Rightarrow \boldsymbol{I}(K) > 0]^\star &= [K \vdash \bot]^\star \Rightarrow [\boldsymbol{I}(K) > 0]^\star \\
&= K \nvdash \bot \Rightarrow \boldsymbol{C}(K) > 0 \\
[K \vdash \bot \Rightarrow \boldsymbol{C}(K) \leq \boldsymbol{C}(K')]^\star &= [K \vdash \bot]^\star \Rightarrow [\boldsymbol{C}(K) \leq \boldsymbol{C}(K')]^\star \\
&= K \nvdash \bot \Rightarrow \boldsymbol{I}(K) \leq \boldsymbol{I}(K')
\end{aligned}$$

The duality fails with

$$\vdash K \;\Rightarrow\; \boldsymbol{C}(K) \geq \boldsymbol{C}(K')$$

because $\vdash K$ is not mapped to $\nvdash K$ (there is no "is maximally unsatisfiable" —which is expected as dual to "is valid" understood as "is maximally satisfiable").

The duality also fails as Definition 1 has no entry for $\vdash^\star$. In fact, the idea that "having more models" implies "being more consistent" has no counterpart in the universe of inconsistency measures: by strong completeness of propositional logic, having more models is equivalent with propositional entailment but this is useless for inconsistent sets of formulas because they all entail each other. In symbols, $[K \vdash K' \Rightarrow \boldsymbol{C}(K) \leq \boldsymbol{C}(K')]^\star$ is undefined. The corresponding item would be $K \vdash K' \Rightarrow \boldsymbol{I}(K) \leq \boldsymbol{I}(K')$ which would make $\boldsymbol{I}$ to collapse because $K \vdash K'$ holds for every inconsistent K and K'.

4 Mixed measures

Consider extending $\boldsymbol{C}$ so that not all inconsistent bases get the same consistency degree. An idea is to extend the codomain to $\mathbb{R} \cup \{-\infty, +\infty\}$. The main postulate would become

Inconsistency below zero

$$\boldsymbol{C}(K) \leq 0 \;\Leftrightarrow\; K \vdash \bot$$

The negative values can as well come from a given inconsistency measure $\boldsymbol{I}$, by letting $\boldsymbol{C}(K) = -\boldsymbol{I}(K)$ whenever $K \vdash \bot$. [5]

The original (i.e., non-extended) $\boldsymbol{C}$ ascribes 0 to every inconsistent K whereas $\boldsymbol{I}$ ascribes 0 to every consistent K. Since no K is both consistent and inconsistent, the extended $\boldsymbol{C}$ is to ascribe 0 to no K at all: the codomain of the extended $\boldsymbol{C}$ is not to include 0.

Definition 2 *A mixed measure M assigns any finite K a value in* $\mathbb{R}^* \cup \{-\infty, +\infty\}$.

For mixed measures, the fundamental postulate therefore is

Inconsistency

$$\boldsymbol{M}(K) < 0 \;\Leftrightarrow\; K \vdash \bot$$

Lemma 1 *The Inconsistency postulate is equivalent to*

$$\boldsymbol{M}(K) > 0 \;\Leftrightarrow\; K \nvdash \bot$$

Proof Trivial. ∎

4. We use $\geq_{inc}$ to intuitively denote a pre-order for "is at least as inconsistent as".

5. Careful: Equivalently, $\boldsymbol{I}(K) = -\boldsymbol{C}(K)$ but keep in mind that $K \vdash \bot$ must hold hence $\boldsymbol{I}$ actually fails to be defined from $\boldsymbol{C}$.

Lemma 2 *The Inconsistency postulate implies that* $\boldsymbol{M}(K) > \boldsymbol{M}(K')$ *whenever* $K \nvdash \bot$ *and* $K' \vdash \bot$.

Proof Trivial. ∎

Another major postulate is

Tautology Top

$$\vdash K \quad \Leftrightarrow \quad \forall K' \; \boldsymbol{M}(K) \geq \boldsymbol{M}(K')$$

That is, tautologies are ascribed the highest degree (they are regarded as more consistent than any non-tautologies).

5 Consistency Measures based on Operations over Models

Entailment Decrease (or the weaker Equivalence postulate) allows us to take advantage of operations over the set of models of a belief base K to ascribe a consistency degree to K using the set of models of K. The next two sections present such examples: supermodels and morpho-logics.

5.1 Supermodels

Ginsberg et al. [6] introduced the notion of supermodels. In a nutshell, these act as would be models for a formula φ as follows: if the truth-value of n atoms (*not* occurrences of atoms) are switched, does switching the truth-value of m other atoms yield a model of φ? This can be viewed as an assessment of consistency, indicative of the extent to which the model can be distorted and still remain a model. To be self-contained, here is a short presentation of supermodels.

Definition 3 *An* (n, m)*-supermodel of* φ *is a model* $\mathfrak{M}$ *of* φ *such that whenever the truth-value of at most n variables is switched in* $\mathfrak{M}$ *(yielding an interpretation* $\mathfrak{M}'$*), we can [still] obtain a model of* φ *by changing the truth-value of at most m other variables in* $\mathfrak{M}'$.

Remark. Should there be less than n or m variables occurring in φ, the definition means that the extra variables whose value may happen to be varied are taken in the rest of the infinite supply (remember that an interpretation is defined over the set of all variables).

Notation. The set of all (n, m)-supermodels of φ is denoted $\mathrm{Supmod}_{\varphi}(n, m)$.

Obviously, the classical models of φ always are the $(0, 0)$-supermodels of φ, i.e. $\mathrm{Supmod}_{\varphi}(0, 0) = \mathrm{Mod}(\varphi)$.

Example 4 *The interpretation pq is a* $(1, 0)$*-supermodel of* $p \vee q$ *as well as a classical model of* $p \vee q$.

Making a restricted use of supermodels is enough to induce a consistency measure as follows.

$$\boldsymbol{C}(K) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \mathrm{Supmod}_K(0, 0) = \emptyset \\ 1 + \sup\{n \mid \mathrm{Supmod}_K(n, 0) \neq \emptyset\} & \text{otherwise} \end{cases}$$

Detailing, $\boldsymbol{C}(K) = 1 + \max\{n \mid \mathrm{Supmod}_K(n, 0) \neq \emptyset\}$ when K is neither tautologous nor inconsistent, while $\boldsymbol{C}(K) = \infty$ for K tautologous.

Please observe that $\boldsymbol{C}$ satisfies Inconsistency Null and Entailment Decrease hence Tautology Top, Equivalence, Variant Equality. Actually, $\boldsymbol{C}$ satisfies Strict Entailment Decrease. Trivially, $\boldsymbol{C}$ fails Tautology 1-Top and Normalization.

5.2 Morpho-Logics

Mathematical morphology, which is based on set theory, deals with shapes and transformations. It has been introduced in logic by Bloch and Lang [4]. Two basic operations in mathematical morphology are dilution (meant to enlarge a shape) and erosion (meant to contract a shape). They are based on a auxiliary structure, captured by a function that maps each point (of the shape) to a set of points (meant to serve as a neighborhood but other options are possible).

This gives us, for Ω a set of interpretations on $\mathsf{Var}(\mathscr{L})$, and ω varying over the set of all interpretations on $\mathsf{Var}(\mathscr{L})$,

$$D_f(\Omega) = \{\omega \mid f(\omega) \cap \Omega \neq \emptyset\}$$

and

$$E_f(\Omega) = \{\omega \mid f(\omega) \subseteq \Omega\}$$

Clearly, properties of f are crucial for D_f and E_f to be faithful to the idea of *actually* enlarging and contracting: e.g., f must be extensive in the sense that $\omega \in f(\omega), \ldots$

Both D_f and E_f are monotone wrt set inclusion of sets of interpretations, e.g.,

$$\text{if } \Gamma \subseteq \Omega \text{ then } E_f(\Gamma) \subseteq E_f(\Omega)$$

With respect to the auxiliary structure, however, dilution is monotone but erosion is anti-monotone: [6]

$$\text{if } f \leq g \text{ then } E_g(\Omega) \subseteq E_f(\Omega)$$

A host of consistency measures can be cast by resorting to distance-based functions. For example, a family $(f_i)_{i \in \mathbf{N}}$ can be defined by

$$f_i(\omega) = \{\omega' \mid \delta(\omega, \omega') \leq i\}$$

where δ is an integer-valued distance over the set of all interpretations on $\mathsf{Var}(\mathscr{L})$.

6. $\leq$ is the usual order over functions with codomain a set of sets: $f \leq f'$ iff $\forall \omega, \; f(\omega) \subseteq f'(\omega)$.

A collection of consistency measures then arises from

$$\boldsymbol{C}(K) \stackrel{\text{def}}{=} \sup\left\{n \mid E_{f_n}(\text{Mod}(K)) \neq \emptyset\right\}$$

Please observe that the consistency measure previously defined with supermodels is the same as the consistency measure obtained here for the case that δ is the Hamming [7] distance d_H defined as

$$d_H(\omega, \omega') = \sum_{a \in \text{Var}(K)} |\omega(a) - \omega'(a)|$$

6 Interim Conclusion

Indeed, there is an extensive body of literature on inconsistency measures: From the seminal article [7], with milestones such as [13, 11, 17] up to more recent contributions in a special issue [14] and an anniversary book [9], as well as, in the meantime, specific measures [19, 15, 1, 5, 12] or analysis of measures through properties and postulates [10, 2, 3, 18] and even measures defined with positive/negative values [16] (also, [8] by a measure comparing inconsistency in two sets of formulas). From a dual perspective, a similar development would seem natural. It certainly makes sense to rank logical bases from more consistent to less consistent in the context of getting conclusions on firmer grounds.

We have shown that a study of consistent measures could reflect work on inconsistency measures. A number of correspondences show up. Though, even with such a preliminary investigation, a caveat has been exhibited: While a notion of "maximally satisfiable" makes sense, there is no corresponding notion of "maximally unsatisfiable". On the more positive side, a finding of interest is that very little (actually, the pair of Null postulates) is needed to establish the correspondence between consistency/inconsistency and positive/negative values for consistency measures and inconsistency measures (Proposition 3).

Also, we have initiated a process of merging a consistency and inconsistency measure into a single measure. Here, a perhaps unexpected byproduct is that the null value disappears… Hopefully, further such surprises are afoot in the area of consistency measures.

References

[1] Ammoura, Meriem, Badran Raddaoui, Yakoub Salhi et Brahim Oukacha: *On an MCS-based inconsistency measure*. Journal of Approximate Reasoning, 80:443–459, 2017.

[2] Besnard, Philippe: *Revisiting postulates for inconsistency measures*. Dans Fermé, Eduardo et João Leite (rédacteurs): *14th European Conference on Logics in Artificial Intelligence (JELIA'14)*, tome 8761 de *Lecture Notes in Computer Science*, pages 383–396, Funchal, Madeira, Portugal, September 24-26, 2014. Springer.

[3] Besnard, Philippe: *Basic postulates for inconsistency measures*. Dans Hameurlain, Abdelkader, Josef Küng, Roland Wagner et Hendrik Decker (rédacteurs): *Transactions on Large-Scale Data- and Knowledge-Centered Systems* XXXIV, tome 10620 de *Lecture Notes in Computer Science*, pages 1–12. Springer, 2017.

[4] Bloch, Isabelle et Jérôme Lang: *Towards mathematical morpho-logics*. Dans Bouchon-Meunier, Bernadette, Julio Gutiérrez-Ríos, Luis Magdalena et Ronald R. Yager (rédacteurs): *Technologies for Constructing Intelligent Systems. 2 – Tools*, tome 90 de *Studies in Fuzziness and Soft Computing*, pages 367–380. Springer, 2002.

[5] De Bona, Glauber et Anthony Hunter: *Localising iceberg inconsistencies*. Artificial Intelligence, 246:118–151, 2017.

[6] Ginsberg, Matthew L., Andrew J. Parkes et Amitabha Roy: *Supermodels and robustness*. Dans *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-1998)*, pages 334–339. AAAI Press / The MIT Press, 1998.

[7] Grant, John: *Classifications for inconsistent theories*. Notre Dame Journal of Formal Logic, 19(3):435–444, 1978.

[8] Grant, John et Anthony Hunter: *Measuring the good and the bad in inconsistent information*. Dans Walsh, Toby (rédacteur): *22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, pages 2632–2637, Barcelona, Catalonia, Spain, July 16-22, 2011. AAAI Press.

[9] Grant, John et Maria Vanina Martinez (rédacteurs): *Measuring Inconsistency in Information*. College Publications, 2018.

[10] Hunter, Anthony et Sébastien Konieczny: *Measuring inconsistency through minimal inconsistent sets*. Dans Brewka, Gerhard et Jérôme Lang (rédacteurs): *11th Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 358–366, Sydney, Australia, September 16-19, 2008. AAAI Press.

[11] Hunter, Anthony et Sébastien Konieczny: *On the measure of conflicts: Shapley inconsistency values*. Artificial Intelligence, 174(14):1007–1026, 2010.

[12] Jabbour, Saïd, Yue Ma, Badran Raddaoui et Lakhdar Saïs: *Quantifying conflicts in propositional logic*

7. Intuitively, the Hamming distance counts how many atoms have a different value (0 or 1) in ω and ω'.

through prime implicates. Journal of Approximate Reasoning, 89:27–40, 2017.

[13] Knight, Kevin: *Measuring inconsistency*. Journal of Philosophical Logic, 31(1):77–98, 2002.

[14] Liu, Weiru et Kedian Mu (rédacteurs): *Special Issue on Theories of Inconsistency Measures and their Applications*, tome 89 de *Journal of Approximate Reasoning*. Elsevier, 2017.

[15] McAreavey, Kevin, Weiru Liu et Paul C. Miller: *Computational approaches to finding and measuring inconsistency in arbitrary knowledge bases*. Journal of Approximate Reasoning, 55(8):1659–1693, 2014.

[16] Mu, Kedian, Jun Hong, Zhi Jin et Weiru Liu: *From inconsistency handling to non-canonical requirements management*. Journal of Approximate Reasoning, 54(1):109–131, 2013.

[17] Thimm, Matthias: *Inconsistency measures for probabilistic logics*. Artificial Intelligence, 197:1–24, 2013.

[18] Thimm, Matthias: *On the evaluation of inconsistency measures*. Dans Grant, John et Maria Vanina Martinez (rédacteurs): *Measuring Inconsistency in Information*, pages 19–60. College Publications, 2018.

[19] Xiao, Guohui et Yue Ma: *Inconsistency measurement based on variables in minimal unsatisfiable subsets*. Dans De Raedt, Luc, Christian Bessière, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz et Peter J. F. Lucas (rédacteurs): *20th European Conference on Artificial Intelligence (ECAI'12)*, tome 242 de *Frontiers in Artificial Intelligence and Applications*, pages 864–869, Montpellier, France, August 27-31, 2012. IOS Press.

Actes JIAF 2019

Combiner les sémantiques à base d'extensions et les sémantiques à base de classement en argumentation abstraite *

Elise Bonzon[1] **Jérôme Delobelle**[2] **Sébastien Konieczny**[3] **Nicolas Maudet**[4]

[1] Université de Paris, LIPADE, F-75006 Paris, France
[2] Université Côte d'Azur, Inria, CNRS, I3S, Sophia-Antipolis, France
[3] CRIL, CNRS - Université d'Artois, Lens, France
[4] Sorbonne Universités, UPMC Univ Paris 06, CNRS - LIP6, UMR 7606, Paris, France
elise.bonzon@mi.parisdescartes.fr jerome.delobelle@inria.fr konieczny@cril.fr nicolas.maudet@lip6.fr

Résumé

Les sémantiques à base d'extensions évaluent l'acceptabilité d'ensembles d'arguments, tandis que les sémantiques à base de classement évaluent la force de chaque argument pour ensuite les classer. Ces deux types de sémantiques se concentrent sur différents aspects de l'information véhiculée par les systèmes d'argumentation. Après avoir discuté du pour et du contre de ces deux approches, nous étudions comment les combiner afin de tirer profit des deux. Nous définissons six nouvelles sémantiques combinant ces deux types de sémantiques. Plus précisément, nous proposons d'affiner les sémantiques à base de classement en utilisant des informations provenant de l'acceptabilité des arguments issue des sémantiques à base d'extensions, et de modifier les extensions provenant des sémantiques à base d'extensions à l'aide d'informations préférentielles issues des sémantiques à base de classement.

Abstract

Extension-based semantics evaluate the acceptability of sets of arguments, while ranking-based semantics evaluate the strength of each argument. These two kinds of semantics focus on different aspects of the information conveyed by argumentation systems. After discussing pros and cons of both approaches, we study how to combine them, in order to take benefits from both. We propose six new families of semantics combining extension-based and ranking-based semantics. More precisely we propose to refine the ranking-based semantics using information coming from extension-based semantics acceptability of arguments, and to modify the extensions chosen by extension-based semantics using preferential information coming from ranking-based semantics.

*Cet article est la version française d'un article publié à KR'18 [9].

1 Introduction

L'argumentation consiste à raisonner à partir d'informations conflictuelles basées sur l'interaction entre arguments. Dans le cadre de l'argumentation abstraite [15], les sémantiques à base d'extensions ont été les premières à être introduites dans l'optique de déterminer les ensembles d'arguments pouvant être conjointement acceptés. Ces extensions doivent généralement être sans-conflits (deux arguments dans une même extension ne peuvent pas s'attaquer) et être capable de défendre chacun de ces arguments. Chaque ensemble d'arguments est donc évalué de manière binaire (un ensemble d'arguments est ou n'est pas une extension) par ces sémantiques.

Les sémantiques à base de labellings [10] ont été introduites par la suite afin d'associer un label à chaque argument d'un système d'argumentation. Une fonction est utilisée pour associer à chaque argument un label $\{in, out, undec\}$, où *in* signifie que l'argument est accepté, *out* signifie que l'argument est rejeté et *undec* signifie que le statut de l'argument est indécis. Tout comme les sémantiques à base d'extensions, l'évaluation proposée par ces sémantiques s'effectuent sur les ensembles d'arguments. Il a d'ailleurs été prouvé [10] que les principales sémantiques à base d'extensions correspondent exactement à certaines sémantiques à base de labellings.

Plus récemment, il a été avancé que cette évaluation binaire ou ternaire est parfois trop rude pour certaines applications (comme les plateformes de débat en ligne [19]), justifiant ainsi la nécessité d'une évaluation plus ciblée de chaque argument. Les sémantiques à base de classement ont alors été proposées (e.g. [11, 18, 8]), avec comme objectif d'évaluer (comparativement) chaque argument. Une

sémantique à base de classement est une fonction qui associe à chaque système d'argumentation un classement (un pré-ordre) entre les arguments. Ce classement représente la force comparative de chaque argument. Ainsi, contrairement aux sémantiques à base d'extensions (et de labellings), cette approche n'évalue pas des ensembles d'arguments mais chaque argument individuellement, en fonction de sa situation dans le graphe d'argumentation. Un type apparenté de sémantique sont les sémantiques graduées (e.g. [6, 19, 12]), où une valeur numérique est associée à chaque argument. L'évaluation ici est donc numérique au lieu d'être ordinale, mais l'objectif reste d'évaluer chaque argument individuellement. Clairement, si une sémantique graduée est définie, alors cela induit directement une sémantique à base de classement correspondante [14].

Deux types d'évaluations existent donc : individuellement, au niveau des arguments (avec les sémantiques à base de classement ou graduées) ou au niveau des ensembles d'arguments (avec les sémantiques à base d'extensions ou de labellings). Ces deux manières d'évaluer les informations codées dans un système d'argumentation sont complémentaires, et utiles pour différentes applications. La première approche est beaucoup plus récente et des travaux restent nécessaires pour mieux comprendre leur comportement et définir de nouvelles sémantiques significatives. Le second type d'évaluation, bien qu'étudié depuis longtemps, nécessite toujours une étude approfondie pour comprendre leurs principes sous-jacents (il convient de mentionner que, contrairement à d'autres tâches de raisonnement telles que l'inférence [20] ou la révision [1, 16], aucun théorème de représentation ou postulat permettant de caractériser les sémantiques de l'argumentation rationnelles n'existe).

Ce travail est basé sur l'observation que ces deux types d'évaluation sont, en un sens, orthogonaux puisqu'ils peuvent tous deux être utilisés pour extraire des informations sur le statut, la force, la situation (des ensembles) des arguments. Au lieu de considérer ces approches comme mutuellement exclusives, notre idée est de combiner le meilleur des deux approches. Nous pensons qu'étudier le potentiel d'une telle combinaison pourra être utile dans le développement des sémantiques de l'argumentation.

Dans ce travail, nous proposons six nouvelles familles de sémantiques, combinant les sémantiques à base d'extensions et les sémantiques à base de classement. Plus précisément, nous proposons d'affiner les sémantiques à base de classement avec des informations provenant de l'acceptabilité d'arguments basée sur les sémantiques à base d'extensions, et de modifier les extensions provenant des sémantiques à base d'extensions à l'aide d'informations préférentielles issues des sémantiques à base de classement.

Dans la section suivante, nous discuterons des différences entre l'évaluation des arguments faite par les sémantiques à base d'extensions et celle faite par les sémantiques à base de classement. Nous rappellerons ensuite certaines notions de l'argumentation abstraite avant de proposer quatre méthodes pour modifier les sémantiques à base de classement en tenant compte des informations provenant des sémantiques à base d'extensions. La première se focalise sur le statut d'acceptabilité de chaque argument (issu des sémantiques à base d'extensions). La seconde se base sur une évaluation plus précise du statut d'acceptabilité de chaque argument [22]. Les deux dernières modifient les sémantiques de propagation [8] pour permettre une distinction plus fine des arguments grâce à ces statuts d'acceptabilité. A l'inverse, afin de modifier les sémantiques à base d'extensions en utilisant les sémantiques à base de classement, nous montrons d'abord que ces dernières peuvent être utilisées pour évaluer les extensions et ne sélectionner que les meilleures. Nous discutons ensuite de la possibilité de prendre le classement retourné par une sémantique à base de classement en tant qu'information préférentielle dans un cadre d'argumentation basé sur les préférences [2] pour ne sélectionner que les attaques les plus convaincantes.

2 Confrontation des sémantiques

Les sémantiques à base d'extensions [15] sont étroitement liées aux modèles de programmes logiques car elles présentent une évaluation « tout ou rien » des ensembles d'arguments. Amgoud et Ben-Naim [18] soulignent certaines caractéristiques propres à ces sémantiques : *Élimination* : l'impact d'une attaque d'un argument y sur un argument x est radical, *i.e.* si y appartient à une extension, alors x est automatiquement exclu de cette extension ; *Existence* : une attaque réussie contre un argument x a le même effet que plusieurs attaques réussies (*i.e.* une attaque est suffisante pour « éliminer » x et plusieurs attaques ne peuvent pas éliminer x dans une plus grande mesure) ; *Absolutisme* : les trois statuts d'acceptabilité (accepté, rejeté ou indécis) ont un sens même sans comparaison entre eux ; *Planéité* : tous les arguments ayant le même statut d'acceptabilité ne peuvent pas être distingués. Ce type d'évaluation peut être utile pour définir des arguments à partir de formules logiques. Ici, les principes d'Élimination et d'Existence semblent essentiels pour capturer le fait qu'une attaque est létale et ainsi éviter toute contradiction entre les arguments afin d'obtenir un ensemble cohérent de formules.

Cependant, dans d'autres applications, certaines de ces propriétés sont discutables. Les plateformes de débat en ligne ont récemment fait leur apparition pour prendre de plus en plus d'importance. Sur ces plateformes, les utilisateurs peuvent argumenter pour ou contre un sujet donné (sous forme de question ou d'affirmation) ou un des arguments déjà existants. Comme le soulignent Leite et Martins [19], l'objectif n'est pas ici de déterminer les arguments pouvant être acceptés ensembles, mais plutôt d'évaluer à quel point la question/affirmation est acceptée. De

plus, face à de nombreux arguments, il est préférable de disposer d'une évaluation des arguments plus détaillée que celle retournée par les sémantiques à base d'extensions. En outre, pour représenter avec exactitude les opinions de milliers d'utilisateurs, il semble plus approprié d'évaluer les arguments en utilisant plusieurs degrés d'acceptabilité ou une acceptabilité graduée. Les sémantiques à base de classement permettent justement de précisément obtenir une évaluation plus détaillée de la force de chaque argument. Cela peut être utile aussi bien pour ces plates-formes de débat que pour sélectionner les meilleurs arguments dans tous les types de débats (persuasion, délibération, etc.).

Cependant, un inconvénient pourrait être que l'évaluation de chaque argument n'est pas du tout liée à son statut d'acceptabilité : être un argument avec une bonne évaluation ne signifie pas que cet argument doit être accepté (étant donné une sémantique à base d'extensions), et même le terme «acceptation» étant défini par rapport au classement, il n'existe aucun seuil naturel permettant de faire la distinction entre les arguments acceptés et non acceptés. Définir une sémantique à base de classement compatible avec le statut d'acceptabilité d'une sémantique à base d'extensions serait alors une solution possible. C'est pourquoi nous proposons, dans cet article, de construire une nouvelle famille de sémantiques qui affinent les sémantiques à base de classement en utilisant les sémantiques à base d'extensions.

A l'inverse, les sémantiques à base d'extensions ont l'inconvénient de proposer une évaluation trop stricte des arguments. Il est, par exemple, impossible de donner une meilleure évaluation à un argument non-attaqué qu'à tous les arguments qu'il défend (voir principe de Planéité), alors que l'acceptabilité de ces derniers dépendent de l'acceptabilité de l'argument non-attaqué. Une évaluation plus détaillée des arguments peut être utilisée pour modifier les sémantiques à base d'extensions en sélectionnant, par exemple, uniquement les meilleures extensions. Ces deux approches seront détaillées par la suite.

3 Préliminaires

Dans cette section, nous rappelons brièvement certains éléments clés des systèmes d'argumentation abstraits.

Définition 1 *Un* ***système d'argumentation (AF)*** *est un couple $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ où $\mathcal{A}$ est un ensemble fini d'arguments et $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ est la* ***relation d'attaque*** *entre arguments. Un ensemble d'arguments $\mathcal{S} \subseteq \mathcal{A}$ attaque un argument $y \in \mathcal{A}$, si $\exists x \in \mathcal{S}$, tel que $(x, y) \in \mathcal{R}$. $\mathcal{S}$ défend $z \in \mathcal{A}$ contre son attaquant y si $\mathcal{S}$ attaque y.*

Un système d'argumentation abstrait peut être représenté par un graphe orienté, où les nœuds représentent les arguments et les arêtes représentent la relation d'attaque entre deux arguments.
Rappelons certaines notions connues sur les graphes.

Définition 2 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ et $x, y \in \mathcal{A}$. Un* **chemin** *P de y vers x, noté $P(y, x)$, est une séquence $\langle x_0, \ldots, x_n \rangle$ d'arguments telle que $x_0 = x$, $x_n = y$ et $\forall i < n, (x_{i+1}, x_i) \in \mathcal{R}$. La longueur du chemin P est n (le nombre d'attaques qui le compose) et est noté $l_P = n$.*

En fonction de la longueur d'un chemin entre deux arguments, l'argument au début de ce chemin peut être un attaquant et/ou un défenseur (*i.e.*, un argument qui attaque un attaquant) de l'argument au bout du chemin.

Définition 3 *Un* **défenseur** *(resp.* **attaquant***) de a est un argument situé au début d'un chemin de longueur paire (resp. impaire). Soit $\mathcal{R}_n(x) = \{y \mid \exists P(y, x)$ with $l_P = n\}$ le multi-ensemble des arguments situés au début d'un chemin de longueur n vers l'argument x. Un argument $y \in \mathcal{R}_n(x)$ est un attaquant (resp. défenseur) direct de x si $n = 1$ (resp. $n = 2$). Une* ***branche défensive*** *(resp.* ***attaquante****) est un chemin de longueur paire (resp. impaire) commençant par un argument non-attaqué.*

3.1 Sémantique à base d'extensions/de labellings

Dans le cadre de Dung [15], plusieurs sémantiques d'acceptabilité ont été définies pour sélectionner les ensembles d'arguments, appelés extensions, qui peuvent être conjointement acceptés étant donné un système d'argumentation.

Définition 4 *Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$. Un ensemble d'arguments $\mathcal{S} \subseteq \mathcal{A}$ est* ***sans-conflit*** *dans AF si $\forall x, y \in \mathcal{S}, (x, y) \notin \mathcal{R}$. Un ensemble sans-conflit $\mathcal{S}$ est* ***admissible*** *s'il défend tous ces arguments contre chacun de leur attaquants directs. Un ensemble admissible $\mathcal{S}$ est une* ***extension complète*** *si chaque argument défendu par $\mathcal{S}$ appartient à $\mathcal{S}$; une* ***extension préférée*** *si c'est un ensemble admissible maximal pour $\subseteq$; une* ***extension stable*** *s'il attaque chaque argument dans $\mathcal{A} \backslash \mathcal{S}$; l'unique* ***extension de base*** *(ou* grounded*) si c'est l'extension complète minimale pour $\subseteq$.*
*Notons $\mathcal{E}_\sigma(AF)$ l'ensemble des extensions de AF pour $\sigma \in$ {**co**(mplète), **pr**(eférée), **st**(able), **gr**(ounded)}.*

Une autre façon de représenter les concepts d'admissibilité consiste à utiliser les labellings [10].

Définition 5 *Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$. Un* ***labelling*** *de AF est une fonction $\mathcal{L} : \mathcal{A} \to \{in, out, undec\}$. Soit $l \in \{in, out, undec\}$, on note $l(\mathcal{L}) = \{x \in \mathcal{A} \mid \mathcal{L}(x) = l\}$.*
Un labelling $\mathcal{L}$ est un ***reinstatement labelling*** *de AF ssi*

- *$\forall x \in \mathcal{A}$, $\mathcal{L}(x) = in$ ssi $\forall y \in \mathcal{R}_1(x)$, $\mathcal{L}(y) = out$;*
- *$\forall x \in \mathcal{A}$, $\mathcal{L}(x) = out$ ssi $\exists y \in \mathcal{R}_1(x)$, $\mathcal{L}(y) = in$;*
- *$\forall x \in \mathcal{A}$, $\mathcal{L}(x) = undec$ ssi $\nexists y \in \mathcal{R}_1(x)$, $\mathcal{L}(y) = in$ et $\exists z \in \mathcal{R}_1(x)$, $\mathcal{L}(z) = undec$.*

Un reinstatement labelling $\mathcal{L}$ est :

- *un* ***labelling complet*** *;*
- *un* ***labelling de base*** *si $in(\mathcal{L})$ est minimal pour $\subseteq$;*
- *un* ***labelling préféré*** *si $in(\mathcal{L})$ est maximal pour $\subseteq$;*

— *un* ***labelling stable*** *si* $undec(\mathcal{L}) = \emptyset$.
Notons $\mathcal{L}_\sigma(AF)$ *l'ensemble des reinstatement labellings de* AF *pour* $\sigma \in \{co, pr, st, gr\}$.

Concernant le **statut d'acceptabilité**, pour un AF avec au moins une extension (resp. reinstatement labelling), un argument est dit **sceptiquement** accepté s'il appartient à toutes les extensions (resp. il est *in* dans tous les reinstatement labellings) de AF. Un argument est **crédulement** accepté s'il appartient à au moins une des extensions (resp. il est *in* dans au moins un des reinstatement labellings) de AF. Pour une sémantique σ donnée, $sa_\sigma(AF)$ (resp. $ca_\sigma(AF)$) représente l'ensemble des arguments sceptiquement (resp. crédulement) acceptés dans AF.

Une notion plus fine du statut de justification a été introduite dans [22] où ce statut est cette fois-ci basé sur le labelling des arguments. Concrètement, le statut de justification d'un argument est constitué de l'ensemble des labels qui lui a été attribués en utilisant la sémantique complète.

Définition 6 *Soit* $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ *et* $x \in \mathcal{A}$. *Le* ***statut de justification*** *de* x *est le résultat de la fonction* $\mathcal{JS} : \mathcal{A} \rightarrow 2^{\{in,out,undec\}}$ *avec* $\mathcal{JS}(x) = \{\mathcal{L}(x) \mid \mathcal{L} \in \mathcal{L}_{co}(AF)\}$.

Par exemple, si un argument est étiqueté soit *in* ou *undec* dans tous les labellings complets alors le statut de justification de cet argument sera $\{in, undec\}$. Six statuts peuvent être considérés avec la sémantique complète : $\{in\}$, $\{out\}$, $\{undec\}$, $\{in, undec\}$, $\{out, undec\}$ et $\{in, out, undec\}$.

3.2 Sémantique à base de classement

Une sémantique à base de classement permet de classer les arguments des plus acceptables aux moins acceptables.

Définition 7 *Une* **sémantique à base de classement** σ *associe à chaque système d'argumentation* $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ *un classement* $\succeq^\sigma_{AF}$ *sur* $\mathcal{A}$, *où* $\succeq^\sigma_{AF}$ *est un pré-ordre (une relation reflexive et transitive) sur* $\mathcal{A}$. $x \succeq^\sigma_{AF} y$ *signifie que* x *est au moins aussi acceptable que* y ; $x \simeq^\sigma_{AF} y$ *équivaut à* $x \succeq^\sigma_{AF} y$ *et* $y \succeq^\sigma_{AF} x$; *et* $x \succ^\sigma_{AF} y$ *équivaut à* $x \succeq^\sigma_{AF} y$ *et* $y \not\succeq^\sigma_{AF} x$.

Un grand nombre de ces sémantiques ont été proposées (voir [14] pour un aperçu) avec, pour chacune d'elles, un comportement et des propriétés logiques différents. Dans cet article, nous nous focaliserons sur la sémantique *h-categoriser* pour illustrer nos méthodes. Cette sémantique a été d'abord introduite dans [6] en tant que sémantique graduée avant d'être définie comme une sémantique à base de classement dans [21].

Définition 8 *Soit* $AF = \langle \mathcal{A}, \mathcal{R} \rangle$. *La fonction* ***h-categoriser*** $Cat : \mathcal{A} \rightarrow]0, 1]$ *est définie t.q.* $\forall x \in \mathcal{A}$,

$$Cat(x) = \begin{cases} 1 & si\ \mathcal{R}_1(x) = \emptyset \\ \frac{1}{1+\sum_{y \in \mathcal{R}_1(x)} Cat(y)} & sinon \end{cases}$$

Définition 9 *La sémantique* ***h-categoriser (Cat)*** *associe à chaque* $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ *un classement* $\succeq^{Cat}_{AF}$ *sur* $\mathcal{A}$ *tel que* $\forall x, y \in \mathcal{A}$, $x \succeq^{Cat}_{AF} y$ *ssi* $Cat(x) \geq Cat(y)$.

Exemple 1 *Calculons l'ensemble des extensions, des reinstatement labellings pour* $\sigma \in \{co, pr, st, gr\}$ *ainsi que le classement retourné par la sémantique h-categoriser sur l'AF de la Figure 1.*

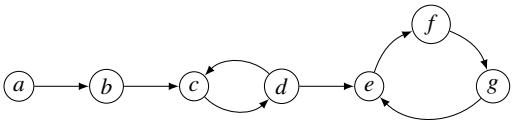


Figure 1 – Un système d'argumentation AF

$\mathcal{E}_{gr}(AF) = \{a\}$, $\mathcal{E}_{pr}(AF) = \{\{a, c\}, \{a, d, f\}\}$,
$\mathcal{E}_{st}(AF) = \{\{a, d, f\}\}$ *et* $\mathcal{E}_{co}(AF) = \{\{a\}, \{a, c\}, \{a, d, f\}\}$.

AF possède 3 reinstatement labellings $\mathcal{L}_1$, $\mathcal{L}_2$ *et* $\mathcal{L}_3$ *avec :*
$in(\mathcal{L}_1) = \{a\}, out(\mathcal{L}_1) = \{b\}, undec(\mathcal{L}_1) = \{c, d, e, f, g\}$
$in(\mathcal{L}_2) = \{a, c\}, out(\mathcal{L}_2) = \{b, d\}, undec(\mathcal{L}_2) = \{e, f, g\}$
$in(\mathcal{L}_3) = \{a, d, f\}, out(\mathcal{L}_3) = \{b, c, e, g\}, undec(\mathcal{L}_3) = \emptyset$

$$Cat(AF) = a \succ^{Cat}_{AF} f \succ^{Cat}_{AF} d \succ^{Cat}_{AF} g \succ^{Cat}_{AF} b \succ^{Cat}_{AF} c \succ^{Cat}_{AF} e$$

De nombreuses propriétés ont été introduites [7, 5] pour mieux comprendre le comportement de ces sémantiques à base de classement dans diverses situations. Nous étudions, par la suite, comment certaines de nos méthodes tiennent compte de ces propriétés. Nous nous contentons de rappeler leur définition informelle mais orientons le lecteur vers [7] pour les versions formelles. Les *propriétés générales de base* disent qu'un classement sur un ensemble d'arguments doit être défini uniquement sur la base de la relation d'attaque (*Abstraction*, **Abs**) ; que le classement entre deux arguments doit être indépendant des arguments qui ne sont connectés ni à l'un ni à l'autre (*Independence*, **In**) ; que tous les arguments sont comparables (*Total*, **Tot**) ; et que tous les arguments non-attaqués doivent avoir le même niveau d'acceptabilité (*Non-attacked Equivalence*, **NaE**).
Les *propriétés locales* se focalisent sur les attaquants/défenseurs directs en disant qu'un argument non-attaqué doit être strictement plus acceptable que tout argument attaqué (*Void Precedence*, **VP**) ; qu'un argument qui s'auto-attaque doit être strictement moins acceptable qu'un argument qui ne s'auto-attaque pas (*Self-Contradiction*, **SC**) ; que si un argument x a strictement plus d'attaquants directs qu'un autre argument y, alors y doit être strictement plus acceptable que x (*Cardinality Precedence*, **CP**) ; que si x possède un attaquant direct qui est strictement plus acceptable que tous les attaquants directs de y, alors x doit être strictement plus acceptable que y (*Quality Precedence*, **QP**) ; que pour deux arguments avec le même

nombre d'attaquants directs, un argument défendu doit être strictement plus acceptable qu'un argument non-défendu (*Defense Precedence*, **DP**) ; qu'une défense où chaque défenseur attaque un attaquant différent est la meilleure (*Distributed-Defense Precedence*, **DDP**) ; que si les attaquants directs de y sont (i) au moins aussi nombreux et (ii) acceptables que ceux de x, alors x doit être au moins aussi acceptable que y (*Counter-Transitivity*, **CT**). Pour sa version stricte (**SCT**), soit (i) ou (ii) doit être strict, impliquant une comparaison stricte entre x et y.

Les *propriétés globales* précisent comment le classement devrait être affecté sur la base de la comparaison des branches attaquantes et défensives de chaque argument. Plus précisément, ajouter une branche défensive à n'importe quel argument doit augmenter son acceptabilité (*Strict Addition of Defense Branch*, $\oplus$**DB**) ; la même propriété a été définie en ajoutant la restriction que l'argument ciblé doit être attaqué (*Addition of Defense Branch*, **+DB**) ; augmenter la longueur d'une branche attaquante d'un argument doit augmenter son acceptabilité (*Augmentation de la branche d'attaque*, $\uparrow$**AB**) ; ajouter une branche attaquante à un argument doit diminuer son acceptabilité (*Addition of Attack Branch*, **+AB**) ; et augmenter la longueur d'une branche défensive d'un argument doit diminuer son acceptabilité (*Increase of Defense Branch*, $\uparrow$**DB**). Dans le même esprit, (*Attack vs Full Defense*, **AvsFD**) requiert qu'un argument possédant uniquement des branches défensives (donc aucune branche attaquante) doit être strictement plus acceptable qu'un argument attaqué uniquement par un argument non-attaqué.

Nous insistons sur le fait qu'il est impossible de satisfaire toutes ces propriétés simultanément [7]. Cependant, vérifier quelles sont celles satisfaites par une sémantique donnée permet de mieux caractériser son comportement.

4 Améliorer les sémantiques à base de classement en utilisant les sémantiques à base d'extensions

4.1 Raffiner avec le statut d'acceptabilité

Cette approche vise à contraindre les classements à être compatibles avec le statut d'acceptabilité des arguments. Pour cela, nous combinons lexicographiquement le classement représentant le statut d'acceptabilité des arguments donné par une sémantique à base d'extensions avec le classement fourni par une sémantique à base de classement.

Définition 10 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ et $\succeq^1_{AF}$, $\succeq^2_{AF}$ deux classements sur $\mathcal{A}$. Le* ***raffinement*** *(lexicographique) de $\succeq^2_{AF}$ par $\succeq^1_{AF}$ donne un nouveau classement $\succeq^{1,2}_{AF}$ tel que $\forall x, y \in \mathcal{A}$,*

$$x \succeq^{1,2}_{AF} y \text{ ssi } (x \succ^2_{AF} y) \text{ ou } (x \simeq^2_{AF} y \text{ et } x \succeq^1_{AF} y)$$

La définition suivante permet de construire un classement à partir du statut d'acceptabilité donné par une sémantique à base d'extensions [1] : un argument sceptiquement accepté est plus acceptable qu'un argument crédulement accepté, qui est plus acceptable qu'un argument rejeté.

Définition 11 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ et $\sigma \in \{co, pr, st, gr\}$. $\succeq^\sigma_{AF}$ est un classement sur $\mathcal{A}$ tel que $\forall x, y \in \mathcal{A}$, $x \succeq^\sigma_{AF} y$ ssi une des ces conditions est satisfaite :* **i)** $x \in sa_\sigma(AF)$, **ii)** $x \in ca_\sigma(AF) \backslash sa_\sigma(AF)$ *et* $y \notin sa_\sigma(AF)$, **iii)** $x, y \notin ca_\sigma(AF)$.

Définition 12 *Soient σ_1 une sémantique à base de classement et $\sigma_2 \in \{co, pr, st, gr\}$. La* ***sémantique à base de classement basée sur l'acceptabilité*** $\texttt{ARS}_{\sigma_1,\sigma_2}$ *associe à chaque $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un classement $\succeq^{\sigma_1,\sigma_2}_{AF}$ sur $\mathcal{A}$ qui est le raffinement de $\succeq^{\sigma_2}_{AF}$ par $\succeq^{\sigma_1}_{AF}$.*

Exemple 2 *Les arguments sceptiquement et crédulement acceptés selon la sémantique complète sur l'AF de la Figure 1 sont respectivement $sa_{co}(AF) = \{a\}$ et $ca_{co}(AF) = \{a, c, d, f\}$. On obtient donc le classement suivant :*

$$a \succ^{co}_{AF} c \simeq^{co}_{AF} d \simeq^{co}_{AF} f \succ^{co}_{AF} b \simeq^{co}_{AF} e \simeq^{co}_{AF} g$$

Rappelons que le classement retourné par la sémantique h-categoriser est le suivant :

$$a \succ^{Cat}_{AF} f \succ^{Cat}_{AF} d \succ^{Cat}_{AF} g \succ^{Cat}_{AF} b \succ^{Cat}_{AF} c \succ^{Cat}_{AF} e$$

Donc en combinant ces deux classements, la nouvelle sémantique retourne le classement suivant :

$$a \succ^{Cat,co}_{AF} f \succ^{Cat,co}_{AF} d \succ^{Cat,co}_{AF} c \succ^{Cat,co}_{AF} g \succ^{Cat,co}_{AF} b \succ^{Cat,co}_{AF} e$$

Cet exemple montre que g possède une assez bonne évaluation avec la sémantique h-categoriser, alors qu'il s'agit d'un argument rejeté avec la sémantique complète. Son évaluation est même meilleure que celle de c qui est crédulement accepté. Cette nouvelle sémantique $\succeq^{Cat,co}$ permet donc de forcer c à être plus acceptable que g.

La question est maintenant de savoir si ces modifications changent la « rationalité » de la sémantique initiale.

Proposition 1 *Soient σ_1 une sémantique à base de classement et $\sigma_2 \in \{co, pr, st, gr\}$. Soit α une propriété parmi Abs, In, VP, DP, DDP, SC, $\oplus$DB, +DB, +AB, $\uparrow$AB, $\uparrow$DB, Tot, NaE. Si σ_1 satisfait la propriété α, alors la sémantique $\texttt{ARS}_{\sigma_1,\sigma_2}$ satisfait la propriété α. Les sémantiques $\texttt{ARS}_{\sigma_1,gr}$ et $\texttt{ARS}_{\sigma_1,st}$ satisfait QP, CT et SCT. La sémantique $\texttt{ARS}_{\sigma_1,\sigma_2}$ satisfait AvsFD et ne satisfait pas CP.*

Il est intéressant de noter que, à part pour AvsFD et CP, cette nouvelle sémantique satisfait les mêmes propriétés que celles satisfaites par la sémantique à base de classement d'origine. Ainsi, la conformité de la sémantique à base de classement vis-à-vis de ces propriétés est préservée même lorsque celle-ci est raffinée par une sémantique à base d'extension. Mieux que cela, elle garantie AvsFD qui est satisfaite par peu de sémantiques existantes [7]. C'est donc un moyen simple d'obtenir de nouvelles sémantiques satisfaisant AvsFD à partir des sémantiques de Dung.

1. Notons que, à priori, toute sémantique à base d'extensions peut être utilisée ici, mais nous nous focalisons uniquement sur les quatre sémantiques de Dung dans cet article (pour les propriétés notamment).

4.2 Raffiner avec le statut de justification

Au lieu de prendre en compte le statut d'acceptabilité des arguments, il est également possible d'établir un classement à partir du statut de justification des arguments qui est basé sur les labellings, offrant une distinction plus fine des arguments. Cependant, la définition d'origine [22] (voir Définition 6) concerne uniquement la sémantique complète. C'est pourquoi nous proposons d'étendre cette définition à toutes les sémantiques de Dung.

Définition 13 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, $\sigma \in \{co, pr, st, gr\}$ et $x \in \mathcal{A}$. Le **statut de justification étendu** de x est le résultat de la fonction $\mathcal{JS} : \mathcal{A} \rightarrow 2^{\{in,out,undec\}}$ t.q. $\mathcal{JS}_\sigma(x) = \{\mathcal{L}_\sigma(x) \mid \mathcal{L} \in \mathcal{L}_\sigma(AF)\}$.*

En plus des 6 statuts $\{in\}$, $\{out\}$, $\{undec\}$, $\{in, undec\}$, $\{out, undec\}$ et $\{in, out, undec\}$, nous devons ajouter le statut $\{in, out\}$, qui ne peut pas apparaître avec la sémantique complète [2], mais qui peut être obtenu avec les autres sémantiques. Les statuts peuvent donc être classés de la façon suivante : $\{in\} \succ_{js} \{in, undec\} \succ_{js} \{undec\} \simeq \{in, out, undec\} \simeq \{in, out\} \succ_{js} \{out, undec\} \succ_{js} \{out\}$. En se basant sur cette classification, il est possible de dire qu'un argument est plus acceptable qu'un autre s'il possède un meilleur statut.

Définition 14 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ et $\sigma \in \{co, pr, st, gr\}$. $\succeq_{AF}^{\mathcal{JS}_\sigma}$ est un classement sur $\mathcal{A}$ t.q. $\forall x, y \in \mathcal{A}$,*

$$x \succeq_{AF}^{\mathcal{JS}_\sigma} y \text{ ssi } \mathcal{JS}_\sigma(x) \succeq_{js} \mathcal{JS}_\sigma(y)$$

Définition 15 *Soient σ_1 une sémantique à base de classement et $\sigma \in \{co, pr, st, gr\}$. La **sémantique à base de classement basée sur le statut de justification** $\mathtt{JRS}_{\sigma_1,\sigma}$ associe à chaque $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un classement $\succeq_{AF}^{\sigma_1,\mathcal{JS}_\sigma}$ sur $\mathcal{A}$ qui est le raffinement de $\succeq_{AF}^{\mathcal{JS}_\sigma}$ par $\succeq_{AF}^{\sigma_1}$.*

Exemple 3 *Rappelons que l'AF de la Figure 1 possède 3 reinstatement labellings $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$. Le(s) statut(s) de justification de chaque argument est $\mathcal{JS}_{co}(a) = \{in\}$, $\mathcal{JS}_{co}(b) = \{out\}$, $\mathcal{JS}_{co}(c) = \mathcal{JS}_{co}(d) = \{in, out, undec\}$, $\mathcal{JS}_{co}(e) = \mathcal{JS}_{co}(g) = \{undec, out\}$ et $\mathcal{JS}_{co}(f) = \{in, undec\}$. Ce qui nous donne le classement suivant :*

$$a \succ_{AF}^{JS_{co}} f \succ_{AF}^{JS_{co}} c \simeq_{AF}^{JS_{co}} d \succ_{AF}^{JS_{co}} e \simeq_{AF}^{JS_{co}} g \succ_{AF}^{JS_{co}} b$$

En le combinant avec le classement retourné par la sémantique h-categorizer, on obtient le classement suivant :

$$a \succ_{AF}^{Cat,JS_{co}} f \succ_{AF}^{Cat,JS_{co}} d \succ_{AF}^{Cat,JS_{co}} c \succ_{AF}^{Cat,JS_{co}} g \succ_{AF}^{Cat,JS_{co}} e \succ_{AF}^{Cat,JS_{co}} b$$

Proposition 2 *Soient σ_1 une sémantique à base de classement $\sigma_2 \in \{co, pr, st, gr\}$. Soit α une propriété parmi Abs, In, VP, DP, DDP, $\oplus$DB, +DB, +AB, $\uparrow$AB, $\uparrow$DB, Tot, NaE. Si σ_1 satisfait la propriété α, alors la sémantique $\mathtt{JRS}_{\sigma_1,JS_{\sigma_2}}$ satisfait la propriété α. Les sémantiques $\mathtt{JRS}_{\sigma_1,JS_{gr}}$ et $\mathtt{JRS}_{\sigma_1,JS_{st}}$ satisfait QP, CT et SCT. La sémantique $\mathtt{JRS}_{\sigma_1,JS_{\sigma_2}}$ satisfait AvsFD et ne satisfait pas CP, SC.*

2. Pour la sémantique complète, il est a été démontré [22, Théorème 2] que si un argument est *in* dans un labelling et *out* dans un autre alors il existe obligatoirement un autre labelling où cet argument est *undec*.

Remarquons que la différence entre les propriétés satisfaites par cette sémantique et celle de base est mineure. En effet, la seule différence concerne Self-Contradiction (SC) et peut s'expliquer par le fait qu'un argument qui s'auto-attaque possède le label *undec* s'il n'est pas attaqué par d'autres arguments. Ainsi, cet argument est plus acceptable qu'un argument directement attaqué par un argument non-attaqué. Cela explique la différence avec la Proposition 1 où les fonctions d'inférence sceptique et crédule considèrent toujours ces deux arguments comme rejetés.

4.3 Raffiner les sémantiques de propagation avec les statuts d'acceptabilité et de justification

Dans cette section, nous proposons d'adapter les sémantiques de propagation introduites dans [8]. L'idée de ces sémantiques est d'attribuer, dans un premier temps, une valeur initiale à chaque argument (cette valeur est plus élevée pour les arguments non-attaqués que pour les arguments attaqués, afin d'améliorer leur impact sur l'évaluation des arguments). Ces valeurs sont ensuite propagées dans l'AF et combinées avant d'être comparées.
Afin de permettre un évaluation initiale plus fine (et non plus binaire), nous proposons d'utiliser le statut d'acceptabilité et le statut de justification.

Définition 16 *Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$. La fonction de valuation $v : \mathcal{A} \rightarrow [0, 1]$ assigne un poids initial à chaque argument. L'évaluation P de $x \in \mathcal{A}$, à l'étape $i \in \mathbb{N}$, est donnée par :*

$$P_i^v(x) = \begin{cases} v(x) & \text{si } i = 0 \\ P_{i-1}^v(x) + (-1)^i \sum\limits_{y \in \mathcal{R}_i(x)} v(y) & \text{sinon} \end{cases}$$

*Le **vecteur de propagation** de x est noté $P^v(x) = \langle P_0^v(x), P_1^v(x), \ldots \rangle$.*

Comme dans [8], nous utilisons l'ordre lexicographique pour comparer les différents vecteurs de propagation.

Définition 17 *L'**ordre lexicographique** entre deux vecteurs de réels $V = \langle V_1, \ldots, V_n \rangle$ and $V' = \langle V'_1, \ldots, V'_n \rangle$ est défini par $V \succeq_{lex} V'$ ssi $\exists i \leq n$ t.q. $V_i \geq V'_i$ et $\forall j < i, V_j = V'_j$.*

Définition 18 *La sémantique à base de classement $Propa^v$ associe à chaque système d'argumentation $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un classement $\succeq_{AF}^{P_v}$ sur $\mathcal{A}$, où v est une fonction d'évaluation t.q. $\forall x, y \in \mathcal{A}$, $x \succeq_{AF}^{P_v} y$ ssi $P^v(x) \succeq_{lex} P^v(y)$.*

Le classement retourné par cette sémantique dépend clairement de la fonction de valuation choisie v. Dans [8], cette fonction ne prend que deux valeurs (une pour les arguments non-attaqués et une pour les arguments attaqués). Considérons maintenant des fonctions de valuations plus complexes en commençant par une fonction de valuation prenant en compte le statut d'acceptabilité des arguments.

Définition 19 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ et $\vec{z}_\sigma = \langle \alpha, \beta, \gamma, \delta \rangle$ un vecteur de réels lié à la sémantique $\sigma \in \{co, pr, st, gr\}$. La fonction de valuation $v_\sigma : \mathcal{A} \rightarrow [0, 1]$ est définie $\forall x \in \mathcal{A}$,*

$$v_{\vec{z}_\sigma}(x) = \begin{cases} \alpha & si\ \mathcal{R}_1(x) = \emptyset \\ \beta & si\ x \in sa_\sigma(AF)\ et\ \mathcal{R}_1(x) \neq \emptyset \\ \gamma & si\ x \in ca_\sigma(AF) \backslash sa_\sigma(AF) \\ \delta & si\ x \notin ca_\sigma(AF) \end{cases}$$

avec $1 \geq \alpha > \beta > \gamma > \delta \geq 0$.

L'importance donnée aux arguments non-attaqués est préservée puisqu'ils ont la plus grande valeur initiale, suivis des arguments sceptiquement acceptés, puis des arguments crédulement acceptés, pour finir avec les arguments rejetés.

Exemple 4 *Avec la sémantique complète, nous observons sur l'AF de la Figure 1 que a est non-attaqué, c, d et f sont crédulement (mais pas sceptiquement) acceptés et b, e, g sont rejetés. La table suivante résume les évaluations de chaque argument avec $\vec{z}_{co} = \langle 1, 0.7, 0.3, 0 \rangle$:*

$P_i^{\vec{z}_{co}}$	a	b	c	d	e	f	g
0	1	0	0.3	0.3	0	0.3	0
1	1	-1	0	0	-0.3	0.3	-0.3
2	1	-1	1.3	0.3	0.3	0.6	-0.3

En comparant lexicographiquement chaque vecteur de propagation, nous obtenons le classement suivant :

$$a \succ_{AF}^{P_{\vec{z}_{co}}} f \succ_{AF}^{P_{\vec{z}_{co}}} c \succ_{AF}^{P_{\vec{z}_{co}}} d \succ_{AF}^{P_{\vec{z}_{co}}} e \succ_{AF}^{P_{\vec{z}_{co}}} g \succ_{AF}^{P_{\vec{z}_{co}}} b$$

Vérifions maintenant les propriétés que cette sémantique satisfait :

Proposition 3 *Soit $\sigma \in \{co, pr, st, gr\}$. La sémantique $Propa^{\vec{z}_\sigma}$ satisfait Abs, In, VP, DP, ↑AB, ↑DB, +AB, Tot, NaE et AvsFD. Les autres propriétés ne sont pas satisfaites.*

Les propriétés satisfaites par $Propa^{\vec{z}_\sigma}$ sont à peu près les mêmes que celles satisfaites par la sémantique $Propa_\epsilon$ introduite dans [8]. Les différences concernent AvsFD qui est maintenant satisfaite par $Propa^{\vec{z}_\sigma}$ grâce à la distinction faite entre les arguments attaqués par la fonction de valuation, par contre SCT et CT qui ne sont plus satisfaites.

La fonction de valuation suivante prend en compte le statut de justification étendue (voir Définition 13).

Définition 20 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ et $\vec{js}_\sigma = \langle \alpha, \beta, \gamma, \delta, \epsilon, \omega \rangle$ un vecteur lié à la sémantique $\sigma \in \{co, pr, st, gr\}$. La fonction de valuation $v_{\vec{JS}_\sigma} : \mathcal{A} \rightarrow [0, 1]$ est définie $\forall x \in \mathcal{A}$,*

$$v_{\vec{JS}_\sigma}(x) = \begin{cases} \alpha & si\ \mathcal{R}_1(x) = \emptyset \\ \beta & si\ \mathcal{JS}_\sigma(x) = \{in\}\ et\ \mathcal{R}_1(x) \neq \emptyset \\ \gamma & si\ \mathcal{JS}_\sigma(x) = \{in, undec\} \\ \delta & si\ \mathcal{JS}_\sigma(x) \in \{\{undec\}, \{in, out\}, \\ & \qquad \{in, undec, out\}\} \\ \epsilon & si\ \mathcal{JS}_\sigma(x) = \{undec, out\} \\ \omega & si\ \mathcal{JS}_\sigma(x) = \{out\} \end{cases}$$

avec $1 \geq \alpha > \beta > \gamma > \delta > \epsilon > \omega \geq 0$.

Exemple 5 *En appliquant la sémantique complète à l'AF de la Figure 1 et avec $\vec{js}_{co} = \langle 1, 0.8, 0.6, 0.4, 0.2, 0 \rangle$, nous obtenons les évaluations suivantes :*

$P_i^{\vec{JS}_{co}}$	a	b	c	d	e	f	g
0	1	0	0.4	0.4	0.2	0.6	0.2
1	1	-1	0	0	-0.4	0.4	-0.4
2	1	-1	1.4	0.4	0.6	1	-0.2

Ce qui donne le classement suivant :

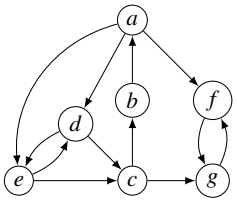
$$a \succ_{AF}^{P_{\vec{JS}_{co}}} f \succ_{AF}^{P_{\vec{JS}_{co}}} c \succ_{AF}^{P_{\vec{JS}_{co}}} d \succ_{AF}^{P_{\vec{JS}_{co}}} e \succ_{AF}^{P_{\vec{JS}_{co}}} g \succ_{AF}^{P_{\vec{JS}_{co}}} b$$

Comme le montre la proposition suivante, augmenter le nombre de distinctions entre les arguments attaqués permet de conserver l'idée des sémantiques de propagation de base tout en proposant une distinction plus fine.

Proposition 4 *Soit $\sigma \in \{co, pr, st, gr\}$. La sémantique $Propa^{\vec{z}_{JS_\sigma}}$ satisfait Abs, In, VP, DP, ↑AB, ↑DB, +AB, Tot, NaE et AvsFD. Les autres propriétés ne sont pas satisfaites.*

5 Améliorer les sémantiques à base d'extensions en utilisant les sémantiques à base de classement

Dans cette section, nous proposons trois méthodes visant à utiliser les sémantiques à base de classement pour modifier les résultats retournés par les sémantiques à base d'extensions. L'objectif des deux premières méthodes est de réduire le nombre d'extensions, afin de faciliter l'inférence, en utilisant les sémantiques à base de classement. La dernière méthode considère le classement retourné par les sémantiques à base de classement en tant qu'information préférentielle dans un cadre d'argumentation basé sur les préférences pour ne sélectionner que les attaques les plus convaincantes.



$\mathcal{E}_{gr}(AF) = \{\}$
$\mathcal{E}_{pr}(AF) = \{\{a, c\}, \{b, d, f\}, \{b, e, f\}, \{b, d, g\}, \{b, e, g\}\}$
$\mathcal{E}_{st}(AF) = \{\{a, c\}, \{b, d, f\}, \{b, e, f\}, \{b, d, g\}, \{b, e, g\}\}$
$\mathcal{E}_{co}(AF) = \{\emptyset, \{a, c\}, \{b, d, f\}, \{b, e, f\}, \{b, d, g\}, \{b, e, g\}\}$

$$Cat(AF) = b \succ_{AF}^{Cat} a \succ_{AF}^{Cat} c \succ_{AF}^{Cat} g \succ_{AF}^{Cat} d \simeq_{AF}^{Cat} e \succ_{AF}^{Cat} f$$

FIGURE 2 – Un AF possédant de nombreuses extensions

5.1 Sélectionner les meilleurs extensions

Comme le montre l'AF de la Figure 2, lorsque plusieurs cycles de longueur paire existent, les sémantiques à base d'extensions de Dung renvoient plusieurs extensions (à l'exception de la sémantique de base qui retourne toujours une extension unique). Comme discuté dans [17], sélectionner les arguments sceptiquement et crédulement acceptés peut être problématique dans ce cas de figure. En effet, l'utilisation de l'inférence sceptique peut ne fournir presque aucune information alors que l'inférence crédule peut retourner jusqu'à l'ensemble de tous les arguments. Par exemple, sur l'AF de la Figure 2, avec la sémantique préférée (la remarque s'applique également aux sémantiques stable et complète), l'ensemble des arguments sceptiquement acceptés est vide, alors que l'ensemble des arguments crédulement acceptés contient tous les arguments de AF. Certains travaux existants (*e.g.* [13, 3]) considèrent certains éléments supplémentaires (quand ils sont disponibles), comme les poids sur les attaques ou les préférences sur les arguments, pour réduire le nombre d'extensions. Cependant, notre objectif est de le faire sans aucune information supplémentaire. Alors que, dans [17], la relation d'attaque est prise en compte pour discriminer certaines extensions, nous proposons ici de considérer le classement renvoyé par une sémantique à base de classement pour sélectionner les "meilleures" extensions. Pour cela, nous proposons deux approches.

5.1.1 Comparer le rang des arguments

Le premier critère que nous considérons est le rang que possède un argument dans un classement d'arguments renvoyé par une sémantique à base de classement. Supposons qu'un agent veuille sélectionner les arguments les plus convaincants à mettre en avant dans un débat afin de convaincre un auditoire : l'agent souhaite utiliser les meilleurs arguments tout en restant cohérent et en étant capable de se défendre contre d'éventuelles attaques.

Définition 21 *Soit $AF = \langle \mathcal{A}, \mathcal{R} \rangle$. Étant donné une sémantique à base de classement σ, le **rang** de $x \in \mathcal{A}$, noté $r_\sigma(x)$, est le niveau auquel il appartient dans la séquence ordonnée des classes d'équivalence de $\mathcal{A}$ étant donné $\succeq^\sigma_{AF}$. Ainsi $r_\sigma(x) = i$ où i est le plus long chemin $x_1 \succ^\sigma_{AF} \ldots \succ^\sigma_{AF} x_i \succ^\sigma_{AF} x$; et $r_\sigma(x) = 0$ si $\nexists y \in \mathcal{A}$ t.q. $y \succ^\sigma_{AF} x$.*

Exemple 6 *En considérant le classement retourné par la sémantique h-categoriser sur AF (Figure 2), le rang de chaque argument est $r_{Cat}(a) = 1$, $r_{Cat}(b) = 0$, $r_{Cat}(c) = 2$, $r_{Cat}(d) = 4$, $r_{Cat}(e) = 4$, $r_{Cat}(f) = 5$ et $r_{Cat}(g) = 3$.*

Étant donné une sémantique à base de classement, le multi-ensemble de rangs d'une extension obtenu à partir d'une sémantique à base d'extensions est composé du rang de chacun de ces arguments.

Définition 22 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, σ_1 une sémantique à base d'extensions et σ_2 une sémantique à base de classement. Pour $\mathcal{E} = \{x_1, \ldots, x_n\} \in \mathcal{E}_{\sigma_1}(AF)$, son **multi-ensemble de rangs** est $rv_{\sigma_2}(\mathcal{E}) = (r_{\sigma_2}(x_1), \ldots, r_{\sigma_2}(x_n))$.*

Exemple 6 (cont.) *En se focalisant sur l'extension préférée $\{b, d, f\}$ et la sémantique h-categorizer, nous obtenons $rv_{Cat}(\{b, d, f\}) = (0, 4, 5)$.*

Nous utilisons une fonction d'agrégation afin d'agréger les valeurs appartenant au même multi-ensemble de rangs.

Définition 23 *Une fonction $\otimes$ est une fonction d'agrégation si $\forall n \in \mathbb{N}$, $\otimes$ est une fonction de $\mathbb{N}^n \rightarrow \mathbb{N}$ t.q.*

- *$x_i \geq x'_i \Rightarrow \otimes(x_1, \ldots, x_i, \ldots, x_n) \geq \otimes(x_1, \ldots, x'_i, \ldots, x_n)$*
- *$\otimes(x_1, \ldots, x_n) = 0$ ssi pour tout i, $x_i = 0$*
- *$\otimes(x) = x$.*

De nombreuses fonctions d'agrégation existent comme *sum*, *avg*, *max*, *min*, *leximax*, *leximin*, etc.

L'objectif est maintenant de comparer le score attribué à chaque extension afin de sélectionner les meilleures.

Définition 24 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, σ_1 une sémantique à base d'extensions, σ_2 une sémantique à base de classement et $\otimes$ une fonction d'agrégation. L'ensemble des **extensions basées sur le rang** (*RBE*) est défini comme suit :*

$$\mathtt{RBE}^{\otimes}_{\sigma_1,\sigma_2}(AF) = \mathrm{argmin}_{\mathcal{E} \in \mathcal{E}_{\sigma_1}(AF)} \otimes(rv_{\sigma_2}(\mathcal{E}))$$

De toute évidence, l'ensemble d'extensions résultant dépend de la fonction d'agrégation choisie. En effet, l'utilisation de la moyenne favorise les extensions avec peu d'arguments mais qui ont un bon rang (même si ce ne sont pas les meilleurs rangs) alors que quand le leximin est utilisé, le nombre d'arguments n'a pas autant d'impact car le rang du meilleur argument de chaque extension est d'abord comparée et en cas d'égalité, les arguments avec le deuxième meilleur rang sont comparés, etc. Ainsi, un agent peut privilégier une fonction d'agrégation soit lexicographique ou basée sur la moyenne, en fonction de la manière dont il pense que le public percevra les arguments (se focaliser sur les plus significatifs ou évaluer le débat dans sa globalité).

Exemple 7 *Sélectionnons les meilleures extensions parmi celles retournées par la sémantique préférée dans l'AF de la Figure 2. Le rang de chaque argument est calculé en utilisant la sémantique h-categoriser. Nous nous focalisons sur la moyenne et le leximin comme fonction d'agrégation.*

	$\mathcal{E}_{pr}(AF)$	$\mathcal{L}eximin$	avg
$\mathcal{E}_1$	$\{a, c\}$	$(1, 2)$	1.5
$\mathcal{E}_2$	$\{b, d, f\}$	$(0, 4, 5)$	3
$\mathcal{E}_3$	$\{b, e, f\}$	$(0, 4, 5)$	3
$\mathcal{E}_4$	$\{b, d, g\}$	$(0, 3, 4)$	2.33
$\mathcal{E}_5$	$\{b, e, g\}$	$(0, 3, 4)$	2.33

Donnant $\mathtt{RBE}^{\mathcal{L}eximin}_{pr,Cat}(AF) = \{\mathcal{E}_4, \mathcal{E}_5\}$ *et* $\mathtt{RBE}^{avg}_{pr,Cat}(AF) = \{\mathcal{E}_1\}$.

Proposition 5 *Pour tout ⊗, pour toutes sémantiques σ_1 et σ_2, pour tout $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, pour tout $x \in \mathcal{A}$,*
- $\texttt{RBE}^{\otimes}_{\sigma_1,\sigma_2}(AF) \subseteq \mathcal{E}_{\sigma_1}(AF)$
- $sa_{\sigma_1}(AF) \subseteq \bigcup_{\mathcal{E} \in \texttt{RBE}^{\otimes}_{\sigma_1,\sigma_2}(AF)} \mathcal{E} \subseteq ca_{\sigma_1}(AF)$

Baroni et Giacomin [4] ont défini un ensemble de propriétés pour les sémantiques à base d'extensions. Vérifions lesquelles sont satisfaites par RBE.

Proposition 6 *Soient ⊗ une fonction d'agrégation et σ_2 une sémantique à base de classement. Soit α une propriété parmi I-maximality, Admissibility, Strong Admissibility, Reinstatement, Weak Reinstatement et CF-Reinstatement [4]. Si la sémantique à base d'extensions σ_1 satisfait α, alors* $\texttt{RBE}^{\otimes}_{\sigma_1,\sigma_2}$ *satisfait α.*

Ainsi, RBE satisfait les mêmes propriétés que la sémantique à base d'extensions (ou de labellings) initiale, à l'exception de la propriété Directionality, comme dans [17].

5.1.2 Comparaison deux à deux

Notre seconde approche vise à comparer toutes les extensions deux à deux en fonction du nombre d'arguments d'une extension qui sont plus acceptables que les arguments d'une autre extension. Un tel choix de comparaison pourrait être intéressant lorsque l'utilisateur aura la possibilité de proposer lui-même des extensions alternatives et de demander une justification quant à la raison pour laquelle cette autre extension n'a pas été sélectionnée.

Définition 25 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, σ_1 une sémantique à base d'extensions, σ_2 une sémantique à base de classement et $\mathcal{E}, \mathcal{E}' \in \mathcal{E}_{\sigma_1}(AF)$.*
$\mathcal{N}_{\sigma_2}(\mathcal{E}, \mathcal{E}') = |\{(x, y) \text{ t.q. } x \succ^{\sigma_2}_{AF} y \text{ avec } x \in \mathcal{E} \text{ et } y \in \mathcal{E}'\}|$

Exemple 8 *Considérons l'ensemble des extensions préférées de AF (Figure 2) et le classement retourné par h-categoriser. On a $\mathcal{N}_{Cat}(\mathcal{E}_1, \mathcal{E}_4) = 4$ car $c \succ^{Cat} d$, $c \succ^{Cat} g$, $a \succ^{Cat} d$ et $a \succ^{Cat} g$. La table suivante contient les résultats des comparaisons de toutes les paires d'extensions :*

$\mathcal{N}_{Cat}$	$\mathcal{E}_1$	$\mathcal{E}_2$	$\mathcal{E}_3$	$\mathcal{E}_4$	$\mathcal{E}_5$
$\mathcal{E}_1 = \{a, c\}$	×	4	4	4	4
$\mathcal{E}_2 = \{b, d, f\}$	2	×	0	0	0
$\mathcal{E}_3 = \{b, e, f\}$	2	0	×	0	0
$\mathcal{E}_4 = \{b, d, g\}$	2	1	1	×	0
$\mathcal{E}_5 = \{b, e, g\}$	2	1	1	0	×

L'approche consiste à compter le nombre d'extensions « vaincues » par chaque extension et sélectionner celle(s) obtenant le meilleur score.

Définition 26 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, σ_1 une sémantique à base d'extensions et σ_2 une sémantique à base de classement. L'ensemble des* ***extensions basées sur l'acceptabilité*** *(*ABE*) est défini comme suit :* $\texttt{ABE}_{\sigma_1,\sigma_2}(AF) =$
$\underset{\mathcal{E} \in \mathcal{E}_{\sigma_1}(AF)}{\text{argmax}} |\{\mathcal{E}' \in \mathcal{E}_{\sigma_1}(AF) : \mathcal{N}_{\sigma_2}(\mathcal{E}, \mathcal{E}') > \mathcal{N}_{\sigma_2}(\mathcal{E}', \mathcal{E})\}|$

Exemple 8 (cont.) *L'extension $\mathcal{E}_1$ bat toutes les autres extensions donc* $\texttt{ABE}_{pr,Cat}(AF) = \{\mathcal{E}_1\}$.

Proposition 7 *Pour toutes sémantiques σ_1 et σ_2, pour tout $AF = \langle \mathcal{A}, \mathcal{R} \rangle$, pour tout $x \in \mathcal{A}$,*
- $\texttt{ABE}_{\sigma_1,\sigma_2}(AF) \subseteq \mathcal{E}_{\sigma_1}(AF)$
- $sa_{\sigma_1}(AF) \subseteq \bigcup_{\mathcal{E} \in \texttt{ABE}_{\sigma_1,\sigma_2}(AF)} \mathcal{E} \subseteq ca_{\sigma_1}(AF)$

Tout comme RBE, la sémantique ABE satisfait globalement les mêmes propriétés que la sémantique initiale.

Proposition 8 *Soit σ_2 une sémantique à base de classement. Soit α une propriété parmi I-maximality, Admissibility, Strong Admissibility, Reinstatement, Weak Reinstatement et CF-Reinstatement [4]. Si la sémantique à base d'extensions σ_1 satisfait α, alors* $\texttt{ABE}_{\sigma_1,\sigma_2}$ *satisfait α.*

5.2 Suppression d'attaques

La dernière modification des sémantiques à base d'extensions que nous proposons est une modification plus radicale puisqu'elle prend beaucoup plus en considération le classement retourné par les sémantiques à base de classement. L'idée ici est de donner une plus grande priorité aux arguments les plus acceptables selon une sémantique à base de classement, en ne considérant que les attaques provenant des arguments plus acceptables que leurs cibles. Nous avons choisi d'utiliser le cadre d'argumentation basé sur les préférences de Amgoud et Cayrol [2] en considérant le classement retourné par une sémantique à base de classement comme relation de préférence entre les arguments.

Amgoud et Cayrol [2] redéfinissent la relation d'attaque en disant qu'un argument x attaque un argument y si et seulement s'il existe bien une attaque de x vers y mais avec la condition supplémentaire que y ne soit pas préféré à x selon le relation de préférence. Cette nouvelle relation d'attaque est ensuite utilisée pour définir les extensions et l'acceptabilité des arguments de manière standard.

Définition 27 *Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ et σ une sémantique à base de classement. Un AF_σ est un triplet $\langle \mathcal{A}', \mathcal{R}', \succeq^{\sigma}_{AF} \rangle$ où :*
- $\mathcal{A}' = \mathcal{A}$
- $\mathcal{R}' = \{(a, b) \mid (a, b) \in \mathcal{R} \text{ et } a \succeq^{\sigma}_{AF} b\}$
- $\succeq^{\sigma}_{AF}$ *est le classement sur $\mathcal{A}$ retourné par σ sur AF*

Exemple 9 *Illustrons cela sur l'AF de la Figure 2 avec la sémantique h-categoriser. Les attaques suivantes doivent être retirées :* (c, b) *(car* $b \succ^{Cat}_{AF} c$*),* (d, c) *(car* $c \succ^{Cat}_{AF} d$*),* (e, c) *(car* $c \succ^{Cat}_{AF} e$*) and* (f, g) *(car* $g \succ^{Cat}_{AF} f$*). Les extensions de AF_{Cat} sont donc : $\mathcal{E}_{gr}(AF_{Cat}) = \{b, c, f\}$, $\mathcal{E}_{co}(AF_{Cat}) = \{\{b, c, f\}, \{b, c, e, f\}, \{b, c, d, f\}\}$, $\mathcal{E}_{pr}(AF_{Cat}) = \mathcal{E}_{st}(AF_{Cat}) = \{\{b, c, e, f\}, \{b, c, d, f\}\}$.*

Notons que ces nouvelles extensions ne sont pas nécessairement sans-conflits par rapport au système d'argumentation original à cause de la suppression de certaines attaques. Mais cela n'est pas surprenant car ces attaques sont

considérées comme illégitimes selon la nouvelle relation d'attaque. Dans [9, Definition 30], une méthode est donnée pour rendre ces extensions sans-conflits.

6 Conclusion

Les sémantiques à base d'extensions et celles à base de classement offrent des évaluations différentes pour les systèmes d'argumentation. Bien qu'elles puissent être utilisées séparément pour différentes applications, il est également intéressant de combiner ces deux approches afin de tirer le meilleur de ces deux types d'évaluation. Dans ce travail, plusieurs approches sont proposées pour combiner ces sémantiques. Plus précisément, nous avons affiné les sémantiques à base de classement en utilisant des informations provenant du statut d'acceptabilité (ou de justification) des arguments issues des sémantiques à base d'extensions. Puis, nous avons procédé au processus inverse en modifiant les extensions pour ne sélectionner que les meilleures d'entre elles, soit en utilisant directement le classement entre arguments ou à l'aide d'informations préférentielles issues des sémantiques à base de classement. Pour chacune d'entre elles, nous avons montré qu'elles gardent globalement les mêmes propriétés logiques.

Références

[1] Alchourrón, Carlos E., Peter Gärdenfors et David Makinson: *On The Logic Of Theory Change : Partial Meet Contraction And Revision Functions*. Journal of Symbolic Logic, 50 :510–530, 1985.

[2] Amgoud, Leila et Claudette Cayrol: *Inferring from Inconsistency in Preference-Based Argumentation Frameworks*. International Journal of Automated Reasoning, 29(2) :125–169, 2002.

[3] Amgoud, Leila et Srdjan Vesic: *Rich preference-based argumentation frameworks*. International Journal of Approximate Reasoning, 55(2) :585–606, 2014.

[4] Baroni, Pietro et Massimiliano Giacomin: *On principle-based evaluation of extension-based argumentation semantics*. Artificial Intelligence, 171(10-15) :675–700, 2007.

[5] Baroni, Pietro, Antonio Rago et Francesca Toni: *How Many Properties Do We Need for Gradual Argumentation ?* Dans *AAAI'18*, 2018.

[6] Besnard, Philippe et Anthony Hunter: *A logic-based theory of deductive arguments*. Artificial Intelligence, 128(1-2) :203–235, 2001.

[7] Bonzon, Elise, Jérôme Delobelle, Sébastien Konieczny et Nicolas Maudet: *A Comparative Study of Ranking-based Semantics for Abstract Argumentation*. Dans *AAAI'16*, pages 914–920, 2016.

[8] Bonzon, Elise, Jérôme Delobelle, Sébastien Konieczny et Nicolas Maudet: *Argumentation Ranking Semantics Based on Propagation*. Dans *COMMA'16*, pages 139–150, 2016.

[9] Bonzon, Elise, Jérôme Delobelle, Sébastien Konieczny et Nicolas Maudet: *Combining Extension-Based Semantics and Ranking-Based Semantics for Abstract Argumentation*. Dans *KR'18*, pages 118–127, 2018.

[10] Caminada, Martin: *On the Issue of Reinstatement in Argumentation*. Dans *JELIA'06*, pages 111–123, 2006.

[11] Cayrol, Claudette et Marie-Christine Lagasquie-Schiex: *Graduality in Argumentation*. Journal of Artificial Intelligence Research, 23 :245–297, 2005.

[12] Costa Pereira, Célia da, Andrea Tettamanzi et Serena Villata: *Changing One's Mind : Erase or Rewind ?* Dans *IJCAI'11*, pages 164–171, 2011.

[13] Coste-Marquis, Sylvie, Sébastien Konieczny, Pierre Marquis et Mohand Akli Ouali: *Selecting Extensions in Weighted Argumentation Frameworks*. Dans *COMMA'12*, pages 342–349, 2012.

[14] Delobelle, Jérôme: *Ranking-based Semantics for Abstract Argumentation.* Thèse de doctorat, 2017.

[15] Dung, Phan Minh: *On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games*. Artificial Intelligence, 77(2) :321–358, 1995.

[16] Katsuno, Hirofumi et Alberto O. Mendelzon: *Propositional Knowledge Base Revision and Minimal Change*. Artificial Intelligence, 52 :263–294, 1991.

[17] Konieczny, Sébastien, Pierre Marquis et Srdjan Vesic: *On Supported Inference and Extension Selection in Abstract Argumentation Frameworks*. Dans *ECSQARU'15*, pages 49–59, 2015.

[18] Leila, Amgoud et Jonathan Ben-Naim: *Ranking-Based Semantics for Argumentation Frameworks*. Dans *SUM'13*, pages 134–147, 2013.

[19] Leite, João et João Martins: *Social Abstract Argumentation*. Dans *IJCAI'11*, pages 2287–2292, 2011.

[20] Makinson, David: *General Patterns in Nonmonotonic Reasoning*. Dans *Handbook of Logic in Artificial Intelligence and Logic Programming (Vol. 3)*, pages 35–110. Oxford University Press, 1994.

[21] Pu, Fuan, Jian Luo, Yulai Zhang et Guiming Luo: *Argument Ranking with Categoriser Function*. Dans *KSEM'14*, pages 290–301, 2014.

[22] Wu, Yining et Martin Caminada: *A Labelling-Based Justification Status of Arguments*. Studies in Logic, 3(4) :12–29, 2010.

Actes JIAF 2019

Estimabilité à état unique des systèmes à évènements discrets

Valentin Bouziat[1] **Xavier Pucel**[1] **Stéphanie Roussel**[1] **Louise Travé-Massuyès**[2]

[1] ONERA / DTIS, Université de Toulouse, F-31055 Toulouse – France
[2] LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

prenom.nom@onera.fr louise@laas.fr

Résumé

Les systèmes autonomes prennent une place de plus en plus importante dans nos environnements quotidiens et imposent des spécifications particulières. Il est donc important de définir et vérifier des propriétés liées à la sécurité, la sûreté ou encore la fiabilité de tels systèmes. Dans cet article, nous adoptons un formalisme de modélisation basé sur les systèmes à évènement discrets (SED) et nous intéressons à des propriétés relatives au diagnostic. Une nouvelle propriété appelée *estimabilité à état unique* est introduite. Lorsque les observations produites par un système ne permettent pas d'être certain de l'état dans lequel il se trouve (plusieurs états sont compatibles avec les observations) l'estimation de son état est ambiguë. Cette propriété exprime alors la possibilité de limiter l'estimation à un seul état sans rencontrer d'incohérence dans la suite de l'exécution. Limiter l'estimation à un seul état facilite entre autres la décision notamment par l'utilisation éventuelle d'un planificateur déterministe dans une architecture autonome complète. Une condition nécessaire et suffisante pour l' *estimabilité à état unique* des SED est fournie ainsi qu'un algorithme récursif permettant de la vérifier et, testé sur différents jeux de données.

Abstract

Specific requirements must guide the design of autonomous systems as they are increasingly present in our everyday environment. Their properties must be carefully defined and checked to guarantee safety, security and dependability. In this paper, we adopt a discrete event system modelling framework and focus on properties that are related to diagnosis. A new property called *single state trackability* is introduced. When available observations may lead to an ambiguous estimate, i.e. several admissible state candidates, this property assesses the possibility of reducing the estimate to a single state without this leading to a dead-end in the continuation of the execution. A single state estimate advantageously facilitates decision making and allows the use of a deterministic planner in the autonomous architecture. We provide a necessary and sufficient condition for *single state trackability* of a discrete event system and we propose a recursive algorithm to check this property. The algorithm is validated with a set of benchmarks.

1 Introduction

La capacité d'un système autonome à réagir à des événements imprévus dépend de sa capacité à estimer son état interne. Ceci influe fortement sur la prise de décision et peut être nécessaire à sa survie. Nous nous concentrons sur des systèmes dont la dynamique peut être représentée par un formalisme de modélisation à événements discrets à temps échantillonné.

Dans ce contexte, l'estimation d'état et le diagnostic ont fait l'objet de nombreux travaux [17, 18, 6, 5]. En général, les observations issues du système ne suffisent pas à garantir l'observabilité de l'état [12, 10], ce qui signifie que l'estimation d'état est ambiguë : plusieurs estimations sont possibles à chaque pas de temps. Par conséquent, le nombre d'états candidats augmente de façon exponentielle au fil de l'exécution du système. La plupart des travaux s'attaquent à ce problème en sélectionnant un nombre limité de meilleurs candidats en fonction d'un critère de préférence, par exemple des probabilités comme dans [16, 8]. Cependant, lorsque l'état réel ne fait pas partie du sous-ensemble sélectionné, cela peut conduire à une impasse dans la suite de l'exécution. La solution proposée par [8] pour ce problème est de revenir en arrière et de récupérer la trajectoire de l'état qui permet à l'estimation de reprendre.

Dans ce travail, nous réduisons à l'extrême le nombre d'estimations et nous proposons de n'en conserver qu'une seule, comme dans [1]. Cela découle de plusieurs raisons. Premièrement, les architectures autonomes ont une quantité très limitée de mémoire et il n'est pas souhaitable de stocker l'historique complet de l'exécution du système puisque celle-ci croît au fil du temps. Deuxièmement,

l'estimation doit être incrémentale et fondée uniquement sur l'estimation précédente et sur l'observation actuelle à chaque étape. Enfin, l'estimation d'un état unique facilite avantageusement la prise de décision et permet l'utilisation de planificateurs déterministes dans l'architecture autonome.

Nous pensons qu'un retour en arrière en cas d'impasse n'est pas une solution viable compte tenu des contraintes de temps réel. C'est pourquoi nous introduisons une nouvelle propriété appelée *estimabilité à état unique* qui évalue la possibilité de réduire l'estimation à un seul état sans que cela n'aboutisse à une impasse dans la poursuite de l'exécution. Des observations ultérieures peuvent prouver que l'état estimé était erroné, invalidant ainsi les décisions précédemment prises. Bien qu'il soit souvent impossible d'assurer l'exactitude de l'état estimé, la propriété présentée dans cet article évalue s'il est possible de sélectionner une estimation unique de l'état du système de manière à ce qu'elle ne soit jamais remise en cause par une séquence d'observations ultérieure. Notre principale contribution est de fournir une condition nécessaire et suffisante pour l'*estimabilité à état unique* d'un système à événements discrets et nous proposons un algorithme récursif pour vérifier cette propriété à partir du modèle du système et de son estimateur.

Cet article est structuré comme suit. Les travaux connexes sont présentés en Section 2. La Section 3 présente le formalisme de modélisation pour les systèmes à événements discrets et les estimateurs incrémentaux. Ensuite, la propriété d'*estimabilité à état unique* est présentée en Section 4. Les concepts utiles à la vérification de cette propriété sont introduits en Section 5. Enfin, un algorithme de vérification est proposé en Section 6 et validé sur des systèmes expérimentaux en Section 7.

2 Travaux connexes

Bien que la propriété d'*estimabilité à état unique* soit nouvelle, les notions connexes d'observabilité et de diagnosticabilité des SED ont fait l'objet de plusieurs travaux. [12] traite de l'observabilité dans le cadre de l'observation partielle de l'état et de l'observation totale des événements. La notion d'états (faiblement) indiscernables tient compte de toutes les observations futures, et l'observabilité (faible) est obtenue lorsque toutes les paires d'états sont (faiblement) indiscernables. La notion de coobservabilité est définie de la même façon pour des observations antérieures. Une forte coobservabilité implique que l'état actuel peut être déterminé de façon certaine à partir de l'historique des observations, ce qui est beaucoup plus fort que l' *estimabilité à état unique* car celle-ci n'impose pas à l'état estimé de correspondre parfaitement à l'état du système.

Dans [10], seuls quelques événements sont observés, et un système est dit observable lorsqu'un observateur qui suit tous les états possibles visite régulièrement un état connu avec certitude. La notion d'observabilité avec délai est similaire, mais permet à l'état unique connu de se situer dans le passé. La définition des observateurs résilients reflète le principe de robustesse aux observations perturbées. L'observabilité exige la connaissance de l'état exact du système, ce qui n'est pas nécessaire pour l'*estimabilité à état unique*. Inversement, l'*estimabilité à état unique* exige que, parmi deux états indiscernables, l'un explique toutes les observations produites par l'autre, ce qui n'est pas nécessaire dans les définitions de l'observabilité.

La diagnosticabilité, telle que définie dans [13], diffère de notre approche par plusieurs aspects. Tout d'abord, elle cible les fautes permanentes, tandis que notre approche peut également être appliquée pour estimer la présence de fautes intermittentes. Deuxièmement, elle nécessite la construction complète d'un diagnostiqueur, ce qui pose un problème de passage à l'échelle ([14] atténue ce problème). Enfin, elle tient compte d'un délai limité entre l'apparition d'une faute et son diagnostic. Bien que cette idée soit intéressante et constitue une exigence réaliste, il n'existe aucun moyen universel de l'appliquer aux phénomènes intermittents. Les méthodes issues de [3] et [7], ont été difficile à appliquer à notre contexte de systèmes autonomes.

3 Estimation d'état dans les SED

Dans cet article, nous adoptons une approche de modélisation similaire à celle du *model-checking*, dans laquelle les états sont définis par des affectations sur un ensemble de variables booléennes $\mathtt{S}$. Certaines variables sont observées (toujours les même à chaque pas de temps) et forment un ensemble $\mathtt{O}$ tel que $\mathtt{O} \subseteq \mathtt{S}$. Les variables qui ne sont pas observés sont alors estimées. On appelle observation une affectation aux variables d'états observés. Nous supposons que le système suit une dynamique discrète et que tous les pas de temps sont de même durée.

définition 1 (Système à évènements discrets) *Un système à évènements discrets (SED) est représenté par un tuple (S, Δ, s_0) où S est l'espace d'état, $\Delta \subseteq S \times S$ est la relation de transition et $s_0 \in S$ est l'état initial.*

On suppose qu'il n'existe aucune contrainte à propos de la vivacité du système, cet hypothèse n'influant pas le problème traité et les définitions associées.

définition 2 (Langage d'exécution) *Pour un SED (S, Δ, s_0), le langage d'exécution $\mathcal{L}(\Delta) \subseteq S^*$ est l'ensemble des séquences finis d'états cohérentes :*

$$\mathcal{L}(\Delta) = \{(s_0, \ldots, s_n) | n \in \mathbb{N}^+, \forall i \in [0, n-1], (s_i, s_{i+1}) \in \Delta\}$$

Notations : Soit $seq \in \mathcal{L}(\Delta)$ une séquence d'états, $seq[i]$ dénote le $i^{\text{ème}}$ état de la séquence commençant à l'état initial $seq[0]$, la taille de la séquence est notée $|seq|$.

Une affectation sur les variables de 0 est appelé une observation. On note O l'ensemble des observations possibles et on note *obs* la fonction d'observation de S vers O qui projette un état sur son observation et nous l'étendons naturellement de S^* vers O^* comme suit : $|obs(seq)| = |seq|$ et $obs(seq)[i] = obs(seq[i])$ pour $i \in [0, |seq|[$.

Nous définissons ensuite le langage des observations pour un SED.

définition 3 (Langage des observations) *Pour un SED, le langage des observations $\mathcal{L}_{obs}(\Delta) \subseteq O^*$ est l'ensemble des séquences d'observations cohérentes.*

$$\mathcal{L}_{obs}(\Delta) = \{obs(seq) | seq \in \mathcal{L}(\Delta)\}$$

Pour une séquence d'observations *sobs* de $\mathcal{L}_{obs}(\Delta)$ et une observation o de O, on note *sobs.o* la concaténation de *sobs* et o.

Nous définissons maintenant le problème de l'estimation d'état dans les SED. L'estimation est un processus incrémental qui base son raisonnement uniquement sur l'état précédent (estimé) et l'observation actuelle.

définition 4 (Candidats à l'estimation) *Étant donnés un SED $(S, \Delta, s_0,)$, un état $s \in S$ et une observation $o \in O$, nous définissons l'ensemble des candidats à l'estimation $cands(s, o)$ comme l'ensemble des états dans lesquels le système pourrait se trouver, en supposant qu'il était dans l'état s à l'instant précédent, et produit l'observation o à l'instant courant. Formellement :*

$$cands(s, o) = \{\hat{s} \in S | (s, \hat{s}) \in \Delta \text{ et } obs(\hat{s}) = o\}$$

Pour une paire donnée (s, o), il existe souvent plusieurs estimations possibles. Dans notre approche, un seul candidat est sélectionné à chaque étape. Ainsi, l'estimation d'état peut être modélisée comme suit.

définition 5 (Fonction d'estimation) *Soit (S, Δ, s_0) un SED. Une fonction d'estimation est une fonction $estim : (S \times O) \to S$ qui sélectionne un état unique parmi les candidats, c.a.d. pour chaque $s \in S$ et $o \in O$, si $cands(s, o) \neq \emptyset$ alors $estim(s, o) \in cands(s, o)$.*

Un estimateur est complètement défini par sa fonction d'estimation. Il reçoit les observations en entrée, et l'état estimé est réutilisé à l'étape suivante pour calculer l'estimation.

définition 6 (Séquence d'estimations) *Soient (S, Δ, s_0) un SED, $estim : (S \times O) \to S$ une fonction d'estimation, et sobs une séquence d'observations de $\mathcal{L}_{obs}(\Delta)$. La séquence d'estimations pour sobs est l'unique séquence d'états sest telle que :*

- *$sest[0] = s_0$ et*
- *pour $i \in [1, |sobs|[$, si $cands(sest[i-1], sobs[i]) \neq \emptyset$ alors $sest[i] = estim(sest[i-1], sobs[i])$ sinon sest est indéfinie.*

4 Estimabilité à état unique

Le simple fait qu'un estimateur sélectionne une estimation unique crée des scénarios dans lesquels l'estimation peut différer de l'état réel du système, le système peut par la suite produire une observation incompatible avec l'état précédent estimé. Dans de tels scénarios, l'estimateur est incapable de produire une estimation, car l'ensemble des candidats à l'estimation est vide et la fonction d'estimation est alors indéfinie [1]. Formellement, si nous notons s l'état réel du système et $\hat{s}$ l'estimation de l'état faite par l'estimateur, si $s \neq \hat{s}$, le système peut évoluer dans un état s' et produire une observation $obs(s') = o$ telle que $cands(\hat{s}, o) = \emptyset$. Nous appelons une telle situation une impasse, et le chemin observable est appelé un chemin sans issue.

définition 7 (Chemin sans issue) *Soient (S, Δ, s_0) un SED, $estim : (S \times O) \to S$ une fonction d'estimation, $k \in \mathbb{N}$, sobs une séquence d'observations de $\mathcal{L}_{obs}(\Delta)$ telle que $k = |sobs|$ et o une continuation de sobs telles que $sobs.o \in \mathcal{L}_{obs}(\Delta)$. sobs.o est un chemin sans issue si et seulement si :*

- *il existe une séquence d'estimations $sest = (\hat{s_0}, \ldots, \hat{s_k})$ pour sobs ;*
- *il n'existe aucun candidat à l'estimation pour o, c.a.d. $cands(\hat{s_k}, o) = \emptyset$.*

Les chemins sans issue illustrent le problème dans lequel l'estimateur fait une hypothèse au sujet de l'état réel du système, et découvre plus tard que celle-ci était fausse. Cela peut poser problème si une décision importante a été prise à la suite de cette hypothèse. La plupart du temps, il est impossible (et pas nécessaire) de connaître avec certitude l'état réel du système pour le faire fonctionner. C'est pourquoi nous introduisons le concept d'*estimabilité à état unique*, c.a.d. la capacité d'estimer de manière unique l'état du système et de ne jamais rencontrer un chemin sans issue.

définition 8 (Estimabilité à état unique) *Un SED est* estimable à état unique [2] *si et seulement s'il existe une fonction d'estimation $estim : (S \times O) \to S$ telle qu'aucune séquence d'observations $sobs \in \mathcal{L}_{obs}(\Delta)$ n'est un chemin sans issue.*

exemple 1 *Considérons le SED représenté en figure 1, décrit comme un automate de Moore [9] sans alphabet d'entrée.*

Au début de l'exécution, le système produit la séquence d'observations (a, a, b). Pour les 3 premiers pas de temps,

1. Notons que de tels scénarios peuvent se produire puisque l'estimateur ne garde pas en mémoire tous les candidats à l'estimation.

2. Dans cet article, « estimable » et « estimabilité » font respectivement référence à estimable à état unique et estimabilité à état unique

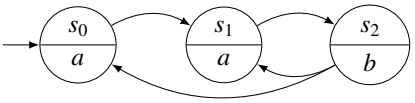


FIGURE 1 – Un SED non estimable avec comme états $S = \{s_0, s_1, s_2\}$ et observations $O = \{a, b\}$. Δ et *obs* sont représentés comme dans une machine de Moore.

l'estimation est triviale puisque le système ne peut qu'emprunter la séquence d'états (s_0, s_1, s_2)*. Mais pour la séquence d'observations* (a, a, b, a)*, l'ensemble des candidats est alors* $cands(s_2, a) = \{s_0, s_1\}$*. Un choix doit alors être fait. Considérons donc les deux fonctions d'estimation possibles* $estim_0(s_2, a) = s_0$ *et* $estim_1(s_2, a) = s_1$*, ainsi que les deux séquences d'observations* (a, a, b, a, a) *et* (a, a, b, a, b) *respectivement produites par les séquences d'états* $(s_0, s_1, s_2, s_0, s_1)$ *et* $(s_0, s_1, s_2, s_1, s_2)$*.*

Avec $estim_0$*, la séquence d'observations* (a, a, b, a) *est estimée comme* (s_0, s_1, s_2, s_0)*, mais du fait que* $cands(s_0, b) = \emptyset$*, la séquence d'observations* (a, a, b, a, b) *est une impasse.*

Avec $estim_1$*, la séquence d'observations* (a, a, b, a) *est estimée comme* (s_0, s_1, s_2, s_1)*, mais du fait que* $cands(s_1, a) = \emptyset$*, la séquence d'observations* (a, a, b, a, a) *est une impasse.*

Puisque toutes les fonctions d'estimation possibles rencontrent des impasses, le système n'est pas estimable.

5 Vérification de l'estimabilité à état unique

Vérifier l'estimabilité d'un système est une tâche difficile. Nous introduisons dans cette section certaines conditions nécessaires et une condition d'équivalence.

définition 9 (État atteignable) *Un état* $\hat{s}$ *est atteignable par la séquence d'observations* $sobs \in \mathcal{L}_{obs}(\Delta)$ *si et seulement si il existe une séquence d'états* $seq \in \mathcal{L}(\Delta)$ *telle que* $obs(seq) = sobs$*, et* seq *se termine par* $\hat{s}$*. L'ensemble des états atteignables via sobs est noté* $reach(sobs)$*.*

Notons que si l'ensemble des séquences d'observations est infini, l'ensemble des états atteignables est toujours fini car c'est un sous-ensemble de 2^S. Cet ensemble peut être énuméré en utilisant une construction par sous-ensembles cohérents avec *obs*.

définition 10 (État non-bloquant) *Soient* (S, Δ, s_0) *un SED, et* $sobs \in \mathcal{L}_{obs}(\Delta)$ *une séquence d'observations. Un état* $\hat{s}$ *est* non-bloquant *pour sobs si celui-ci est atteignable par sobs et pour chaque observation ultérieure o, il existe au moins un candidat. Formellement,* $\hat{s} \in reach(sobs)$ *est* non-bloquant *si et seulement si :*

$$\forall o \in O \text{ si } sobs.o \in \mathcal{L}_{obs}(\Delta), \text{ alors } cands(\hat{s}, o) \neq \emptyset$$

Un état atteignable via *sobs* mais qui n'est pas non-bloquant pour *sobs* est appelé *état bloquant*. Notons qu'un état $\hat{s}$ peut être non-bloquant pour une séquence d'observations $sobs_1$ et bloquant pour une autre séquence $sobs_2$. Les états non-bloquants sont importants car les estimateurs doivent toujours les sélectionner, sinon ils pourraient rencontrer un chemin sans issue.

proposition 1 (Condition sur les états non-bloquants) *S'il existe une séquence d'observations* $sobs \in \mathcal{L}_{obs}(\Delta)$ *telle que* $reach(sobs)$ *ne contient que des états bloquants, alors le système n'est pas estimable.*

Intuitivement, la proposition 1 signifie que si l'ensemble des états atteignables $reach(sobs)$ pour une séquence d'observations *sobs* de $\mathcal{L}_{obs}(\Delta)$ ne contient que des états bloquants, alors quel que soit l'état $\hat{s} \in reach(sobs)$ choisi, il existe une continuation $sobs.o \in \mathcal{L}_{obs}(\Delta)$ qui est une impasse. Il est donc impossible, dans cette configuration, de construire un estimateur qui ne rencontrerai pas d'impasse.

proposition 2 (Transitions entre états non bloquants) *Soit* $sobs \in \mathcal{L}_{obs}(\Delta)$ *une séquence d'observations et* $o \in O$ *une observation telle que* $sobs.o \in \mathcal{L}_{obs}(\Delta)$*. Si le système est estimable, alors il existe une paire d'états* $(s_1, s_2) \in \Delta$ *telle que* s_1 *est non-bloquant pour sobs et* s_2 *est non-bloquant pour* $sobs.o$*.*

La proposition 2 étend la proposition 1 aux transitions, et pourrait être étendue à des chemins de longueur arbitraire. D'une part, lors de la construction de l'ensemble de tous les $reach(sobs)$ afin de vérifier la proposition 1, et en particulier lors de la construction d'une transition entre $reach(sobs)$ et $reach(sobs.o)$, il est facile de vérifier la proposition 2 à la volée. La recherche de chemins plus longs nécessiterait une recherche spécifique.

D'autre part, la vérification de cette propriété pour les chemins de n'importe quelle longueur ne fournit pas une condition suffisante pour l'estimabilité. Dans le SED présenté en figure 2, il est toujours possible pour une séquence d'observations de construire des paires d'états respectant la proposition 2 cependant le SED n'est pas estimable. En effet, si nous prenons la séquence d'observations (a, b, a, b, c) produite par la séquence d'états non bloquants $(s_0, s_1, s_0, s_2, s_4)$, il est cependant impossible de construire une fonction $estim : (S \times O) \rightarrow S$ permettant d'emprunter cette séquence puisque qu'a partir de s_0 et en recevant l'observation b, il faudrait choisir tantôt s_1, tantôt s_2.

Afin de fournir une condition nécessaire et suffisante pour assurer l'estimabilité, il est requis de s'assurer que le langage des observations (l'ensemble des séquences d'observations) est supporté par un estimateur.

définition 11 (Langage accepté par l'estimateur) *Soit* (S, Δ, s_0) *un SED,* $sobs \in \mathcal{L}_{obs}(\Delta)$ *une séquence d'observations et* $estim : (S \times O) \rightarrow S$ *une fonction d'estimation.*

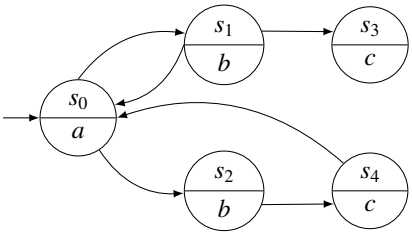


FIGURE 2 – Un SED non estimable mais satisfaisant la condition de la proposition 2.

Le langage accepté par cette fonction d'estimation, noté $\mathcal{L}_{obs}(estim)$*, est l'ensemble de toutes les séquences d'observations possibles pour lesquelles il existe une séquence d'estimations. Formellement :*

$$\mathcal{L}_{obs}(estim) = \{sobs \in \mathcal{L}_{obs}(\Delta) \mid |sobs| > 1 \ \ et \ \exists sest \in \mathcal{L}(\Delta)$$

$$t.q.\ sobs = obs(sest)$$

$$et\ pour\ i \in [1, |sobs|], estim(sest[i-1], sobs[i]) = sest[i]\}$$

Nous définissons ensuite une condition nécessaire et suffisante pour l'estimabilité à état unique.

proposition 3 (Condition d'estimabilité) *Un SED* (S, Δ, s_0) *est* estimable à état unique *si et seulement s'il existe une fonction d'estimation dont le langage accepté est égal au langage des observations du système :*

$$\exists\, estim : (S \times O) \to S \ telle\ que\ \mathcal{L}_{obs}(estim) = \mathcal{L}_{obs}(\Delta)$$

Il devient maintenant évident qu'il est possible de trouver des impasses pour un estimateur donné à l'aide d'un algorithme semblable à celui de la vérification de l'égalité des langages réguliers.

La preuve est assez directe : si la condition de la proposition 3 est satisfaite, alors la fonction d'estimation fournit une séquence d'estimations pour tous les éléments de $\mathcal{L}_{obs}(\Delta)$. Il n'y a donc pas d'impasse. La principale difficulté est de trouver une telle fonction d'estimation ou de prouver qu'elle n'existe pas.

Afin de vérifier efficacement l'estimabilité, notre approche consiste à construire récursivement des fonctions d'estimation partiellement définies et les tester. Nous définissons les impasses pour ces fonctions d'estimation partielles et introduisons une proposition utilisée dans notre algorithme.

définition 12 (Prolongement d'une fonction d'estimation) *Soient estim et pestim deux fonctions d'estimation de* $(S \times O)$ *vers* S*. On dit que estim* prolonge *pestim si et seulement si pour tout couple* (s, o) *de* $S \times O$ *tel que* $pestim(s, o)$ *est définie, alors* $estim(s, o)$ *est également définie et* $pestim(s, o) = estim(s, o)$

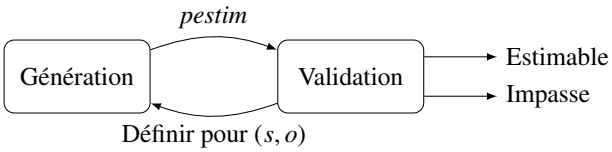


FIGURE 3 – Structure de l'algorithme.

proposition 4 *Soient sobs une séquence d'observations de* $\mathcal{L}_{obs}(\Delta)$*, pestim et estim deux fonctions d'estimation de* $(S \times O)$ *vers* S *telles que estim prolonge pestim. Si sobs est un chemin sans issue pour pestim, alors sobs est également un chemin sans issue pour estim.*

6 Algorithme

Nous décrivons un algorithme pour vérifier l'estimabilité d'un système. Le principe est de trouver des impasses pour des fonctions d'estimation partiellement définies. Notre algorithme est structuré en deux composantes : la première produit des fonctions d'estimation partiellement définies et la seconde vérifie si une fonction d'estimation partielle donnée satisfait la proposition 3. La structure de l'algorithme est illustré en figure 3.

La composante de génération produit récursivement des fonctions d'estimation partielles *pestim* et les envoie pour validation. La composante de validation a trois types de retours : « Estimable » signifie que le système peut être surveillé avec la *pestim* courante ; « Impasse » signifie que *pestim* rencontre une impasse ; « Définir pour (s, o) » signifie que *pestim* doit être prolongée pour inclure une estimation pour (s, o). Selon le résultat de la validation, la composante de génération peut terminer ou produire récursivement d'autres fonctions d'estimation partielles.

6.1 Génération de fonction d'estimation

La composante de génération passe par une fonction GENESTIM décrite dans l'algorithme 1 qui lance le processus de vérification et appelle la composante de validation. Lors de la première itération, la composante de génération produit la fonction d'estimation partielle vide (c.a.d. définie sur $\emptyset$), et l'envoie à la composante de validation.

Si la composante de validation retourne « Estimable », cela signifie que la fonction d'estimation partielle courante *pestim* définit un estimateur qui accepte $\mathcal{L}_{obs}(\Delta)$, et que les paires d'états pour lesquelles *pestim* n'est pas définie ne sont pas nécessaires pour l'estimation. De ce fait, toute extension de *pestim* satisfait la proposition 3, et le système est estimable.

Si la composante de validation retourne « Impasse », cela signifie qu'il existe un chemin sans issue pour *pestim*. Par la proposition 4, aucune extension de cette fonction ne peut satisfaire la proposition 3. La récursion se termine.

Algorithme 1 Composante de génération : fonction GENESTIM

```
1:  Entrees
2:    Σ = (S, Δ, s_0) : un SED
3:  Sorties
4:    pestim : une fonction d'estimation partielle
5:  function GENESTIM(Σ, pestim)
6:    switch CHECKESTIM(Σ, pestim, s_0, {s_0}) :
7:      case Estimable : return true
8:      case Impasse : return false
9:      case Définir pour (s,o) :
10:       for c in cands(s, o) do
11:         extension ← pestim ∪ ((s, o), c)
12:         if GENESTIM(Σ, extension) then
13:           return true
14:       end for
15:       return false
```

Si la composante de validation retourne « Définir pour (s, o) », cela signifie qu'il existe une paire (s, o) telle que s est atteignable, $cands(s, o)$ contient plusieurs candidats, et *pestim* n'est pas définie pour (s, o). Dans ce cas, nous devons vérifier s'il existe une option d'estimation pour (s, o) qui satisfait la proposition 3, et les extensions de *pestim* sont alors générées récursivement.

L'algorithme explore récursivement toutes les options d'estimation, Ainsi s'il termine sans qu'aucune fonction d'estimation satisfaisant la proposition 3 n'ait été trouvée, alors le système n'est pas estimable.

6.2 Validation de fonction d'estimation

La composante de validation contient une fonction CHECKESTIM qui vérifie si le langage accepté par une fonction d'estimation partielle est égal au langage des observations du système. L'algorithme est basé sur une légère modification de l'algorithme classique pour tester l'égalité des langages réguliers [2] afin de tenir compte des cas où plusieurs candidats à l'estimation existent et pour lesquelles la fonction d'estimation partielle est indéfinie.

L'approche consiste à simuler l'exécution de l'estimateur et du système, tout en s'assurant qu'ils sont synchronisés sur les mêmes séquences d'observations. Puisque pour chaque séquence d'observations *sobs*, il existe (au plus) une séquence d'estimations unique, nous n'avons besoin de garder trace que d'un seul état dans l'estimateur. Ainsi, l'état de l'estimateur atteint via une séquence d'observations *sobs* est associé à l'ensemble des états systèmes $reach(sobs)$ (voir définition 9).

L'algorithme explore récursivement des paires $(estSt, sysSts)$ où $sysSts = reach(sobs)$ pour une $sobs \in \mathcal{L}_{obs}(\Delta)$, et où $estSt \in sysSts$. Pour s'assurer que l'algorithme termine, une variable globale « *visited* » stocke les paires qui ont déjà été explorées. L'algorithme recherche les séquences *sobs* pour lesquelles l'estimateur n'est pas défini, de sorte que la condition de terminaison n'est remplie que si une telle séquence n'existe pas. Dans ce cas, les langages sont égaux, et nous retournons « Estimable » (lignes 11 à 12).

Algorithme 2 Composante de validation : fonction CHECKESTIM

```
1:  Entrees
2:    Σ = (S, Δ, s_0) : un SED
3:    estim : une fonction d'estimation partielle
4:    estSt ∈ S : état de l'estimateur
5:    sysSts ⊆ S : états du système
6:  Sorties
7:    « Estimable » , « Impasse » ou « Définir pour (s, o) »
8:  Global
9:    visited ∈ S × 2^S ← ∅
10: function CHECKESTIM(Σ, estim, estSt, sysSts)
11:   if (estSt, sysSts) ∈ visited then
12:     return Estimable
13:   visited ← visited ∪ (estSt, sysSts)
14:   nextObs ← {obs(s')|∃s ∈ sysSts, (s, s') ∈ Δ}
15:   for o in nextObs do
16:     estNxts ← NEXTESTSTATES(estSt, o)
17:     sysNxts ← {s' | obs(s') = o ∧
18:                                  ∃s ∈ sysSts, (s, s') ∈ Δ}
19:     if estNxts = ∅ then
20:       return Impasse
21:     else if estNxts = {estNxt} then
22:       rec ← CHECKESTIM(Σ, estim, estNxt, sysNxts)
23:       switch rec :
24:         case Impasse : return Impasse
25:         case Définir pour (s, o) :
26:           return Définir pour (s, o)
27:         case Estimable : continue
28:     else
29:       return Définir pour (estSt, o)
30:   end for
31:   return Estimable
```

A chaque itération, CHECKESTIM calcule les observations que le système peut produire (ligne 14). Ensuite, pour chaque observation produite, il appelle une fonction `NextEstStates`$(estSt, o)$ (ligne 16) définie comme suit. Si *pestim* est définie pour $(estSt, o)$, celle-ci retourne $\{pestim(estSt, o)\}$ sinon $cands(estSt, o)$. L'algorithme teste ensuite le nombre d'état possibles pour l'estimateur :

— Si $estNxts = \emptyset$ (ligne 19) : cela signifie que si *sobs* est la séquence d'observations courante (dans l'appel récursif), *sobs.o* est un chemin sans issue (voir définition 7),« Impasse » est retourné.
— si $estNxts = \{estNxt\}$ (ligne 21) : cela signifie que

pour la paire d'état courante $(estSt, o)$, soit il n'y a qu'un unique successeur, soit *pestim* est définie. Dans ce cas la recherche continue avec des appels récursifs.

— si $|estNxts| > 1$ (ligne 28) : cela signifie qu'il y a plusieurs candidats à l'estimation, et que *pestim* n'est pas définie pour $(estSt, o)$. « Définir pour $(estSt, o)$ » est retourné, afin que la composante de génération produise des fonctions d'estimation définies pour cette paire.

6.3 Améliorations

Les performances de l'algorithme décrit ci-dessus peuvent être considérablement améliorées avec quelques mécanismes. Une vérification préliminaire est ajoutée pour vérifier les propositions 1 et 2 pour chaque ensemble d'états qui peut être atteint via une séquence d'observations.

Dans nos expériences, la principale source de complexité est le nombre de fonctions d'estimation partielles à tester. Le seul mécanisme dont nous disposons pour élaguer les fonctions d'estimations partielle est de trouver des impasses le plus tôt possible.

Tout d'abord, dans l'algorithme 1, lors de la détection d'une impasse, les triplets d'estimation qui ne sont pas utilisés dans le chemin sans issue peuvent être supprimés de *pestim*, et nous pouvons mémoriser uniquement cette fonction d'estimation partielle tronquée. De cette façon, par la proposition 4, nous pouvons tester à la ligne 11 si certaines extensions sont compatibles et étendre celle que nous venons de mémoriser afin d'épargner quelques appels à l'algorithme 2.

Deuxièmement, il existe des fonctions d'estimation partielles qui contiennent à la fois un chemin sans issue et des paires non définies (s, o). Dans ce cas, dans l'algorithme 2, soit « Impasse » , soit « Définir pour (s, o) » est retourné selon l'ordre de traversée de la ligne 15. Afin de favoriser « Impasse » , au lieu de retourner « Définir pour (s, o) » , nous stockons la fonction partielle dans une variable globale, et continuons la recherche. Lorsque la recherche récursive globale est terminée, si le résultat est « Impasse » nous le retournons, sinon si la variable globale contient une paire (s, o) nous retournons « Définir pour (s, o) » et sinon nous retournons « Estimable » . De cette façon, nous détectons les impasses beaucoup plus tôt dans la recherche.

Les propriétés sur les états bloquants des propositions 1 et 2 peuvent également être utilisées pour réduire l'espace de recherche : dans l'algorithme 1 à la ligne 10, les états bloquants peuvent être ignorés car ils mèneront forcément à une impasse. Puisque les fonctions d'estimation sont générées en largeur pour un état précédent et une observation, cela permet d'élaguer une branche dans l'espace de recherche pour chaque état bloquant.

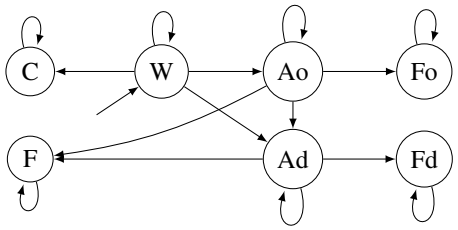


FIGURE 4 – Le workflow des états des actions qui peuvent être (W)aiting, (C)ancelled, (A)ctive (o)k, (F)inished (o)k, (F)ailed, (A)ctive (d)elayed, ou (F)inished (d)elayed.

7 Expérimentations

Nous avons testé notre approche sur un exemple inspiré du langage de robotique autonome PLEXIL. [15]. Ce cadre est organisé autour du concept d'actions, qui ont un workflow hiérarchique complexe. Nous utilisons un workflow d'actions séquentielles simplifié décrit dans l'exemple 2.

exemple 2 *Nous considérons un cadre robotique où les plans des robots sont représentés sous forme de séquence d'actions. La figure 4 illustre le workflow des actions. Nous considérons un robot avec un plan constitué de deux actions séquentielles : move et inspect, dont les états sont représentés par les variables mv et ins. L'état de santé du robot est décrit par trois variables booléennes hnav, hsens et hpow représentant respectivement si les fonctions de navigation, de capteur et d'alimentation électrique fonctionnent normalement. Une autre variable booléenne, pert, indique si le robot est sujet à des perturbations (terrain glissant, obstacle, vent) à chaque pas de temps. Nous notons move* $= W$ *pour exprimer que l'action move passe dans l'état W. Nous notons* $Y(v)$ *la valeur à l'étape précédente pour la variable* $v \in \{mv, ins, hnav, hsens, hpow, pert\}$. *Nous utilisons la fonction* $start(v) = K$ *qui signifie* $Y(v) \neq K \wedge v = K$. *Le comportement du système est décrit par l'automate pour chaque action, associé aux contraintes suivantes :*

$$mv \in \{W, Ao, Ad\} \rightarrow ins = W \quad (1)$$

$$mv \in \{C, F\} \rightarrow ins = C \quad (2)$$

$$start(mv = Fo) \rightarrow ins = Ao \quad (3)$$

$$start(mv = Fd) \rightarrow ins = Ad \quad (4)$$

$$start(mv = Ad) \rightarrow (\neg hnav \vee \neg hpow \vee pert) \quad (5)$$

$$start(mv = F) \rightarrow \neg hpow \quad (6)$$

$$start(ins = Ad) \rightarrow (\neg hsens \vee \neg hpow \vee pert) \quad (7)$$

$$start(ins = F) \rightarrow \neg hpow \quad (8)$$

$$hnav \vee hsens \rightarrow hpow \quad (9)$$

L'action ins doit rester dans W lorsque l'action mv s'exécute (1). *Si mv échoue ou est annulé, ins est annulé*

Modèle	États	Succ.	Résultat	Temps(s)
Exemple 2	112	13.7	oui	66
Exemple 2-v2	112	13.7	non	0.4
Valve Controller	209	15.5	non	0.4
Valve Driver	51	22.3	oui	56

TABLE 1 – Temps de calcul pour vérifier l'estimabilité. La colonne « États »indique le nombre d'états du système, « Succ. »le nombre moyen de transitions pour chaque état, « Résultat »indique si le système est estimable, « Temps »le temps de calcul en secondes.

(2). *ins commence au moment où mv finit* (3), (4). *Un retard dans mv (resp. ins) peut s'expliquer par un problème de navigation (resp. capteur), d'alimentation, ou une perturbation* (5). *(resp.* (7)*). Une défaillance dans mv ou ins ne peut s'expliquer que par un problème d'alimentation* (6), (8). *Un problème d'alimentation électrique se propage au capteur et à la navigation* (9).

Les variables mv et ins sont observables, c'est-à-dire que l'état de chaque action est connu. Les variables hnav, hsens, hpow et pert sont estimées. L'ensemble d'états S est l'ensemble des évaluations pour toutes les variables, la fonction obs limite une évaluation aux variables mv et ins. Par exemple, l'état initial $(mv = W, ins = W, hnav, hsens, hpow, \overline{pert})$ *produit l'observation* $(mv = W, ins = W)$.

Notre algorithme est implémenté en Scala, et exécuté sur un processeur Intel® Xeon(R) W-2123, 3.60GHz 8 coeurs, avec une mémoire limitée à 4 Go. Le système tel que décrit dans l'exemple 2 est estimable, nous avons donc introduit quelques modifications pour le rendre non estimable. Nous avons fait en sorte que les actions produisent la même observation dans leurs états « Ao » et « Ad ». Nous avons aussi modélisé les systèmes d'exemple de [13] (Valve Controller) et [16] (Valve Driver).

Les résultats présentés dans la table 1 montrent que les systèmes non estimables sont détectés très rapidement. Ceci est dû à la vérification préliminaire introduite dans la Section 6.3 qui capte les impasses plus tôt. Bien que les propositions 1 et 2 ne suffisent pas à assurer l'estimabilité, elles sont utiles dans une grande majorité des cas. Dans le cas des systèmes estimables, le temps de calcul est lié au nombre de fonctions d'estimation partielle testées. Notons qu'il peut être améliorer en tenant compte des besoins opérationnels pour limiter l'espace des fonctions d'estimation à examiner.

8 Conclusion

Cet article motive et définit l'*estimabilité à état unique* pour les systèmes d'événements discrets partiellement observables. Cette propriété indique s'il est possible de suivre l'exécution d'un système en ne gardant en mémoire qu'un unique état, sans jamais perdre de cohérence avec sa dynamique. Certaines conditions sont fournies avec un algorithme pour vérifier cette propriété. Les résultats expérimentaux montrent que cet algorithme peut être appliqué à des exemples tels que des plans de robots autonomes. L'algorithme constitu une validation de concept et pourrait être amélioré de nombreuses façons, par exemple en définissant des heuristiques spécifiques pour le rendre plus efficace. Des tests sur des jeux de données plus larges viendront étoffer les expérimentations.

Dans cet article, l'*estimabilité à état unique* est obtenue en trouvant une fonction d'estimation dont le langage des observations correspond à celui du système. En général, comme il peut exister un nombre important de telles fonctions, nous pourrions chercher la meilleure fonction d'estimation par rapport à d'autres propriétés, par exemple la correction de l'estimation pour certaines variables ou à certains moments.

Les travaux sur l'observabilité et la diagnostibilité tiennent souvent compte d'un délai limité possible entre les événements, leur observation ou leur diagnostic. Si nous pouvions tenir compte des retards dans le processus d'estimation, nous pourrions nous pencher sur une définition plus générale de l'*estimabilité à état unique*. L'étude de la complexité théorique du problème de l'estimabilité à état unique fait aussi partie de nos perspectives.

De plus, nous sommes convaincus qu'il existe un lien très fort entre certains travaux issus de la théorie des jeux [4], ainsi que de la synthèse de contrôleur [11] ce qui laisse entrevoir des perspectives d'application et de comparaison avec ces différents domaines.

Enfin, la synthèse automatique ou semi-automatique des fonctions d'estimation est une application directe de ce travail, par exemple en représentant la fonction d'estimation avec des langages compacts comme le font les auteurs dans [1].

Références

[1] Bouziat, V., X. Pucel, S. Roussel et L. Travé-Massuyès: *Preferential discrete model-based diagnosis for intermittent and permanent faults*. Dans *Proceedings of the 29th International Workshop on Principles of Diagnosis DX'18*, Warsow, Poland, Août 2018. CEUR Workshops Proceedings. https://hal.archives-ouvertes.fr/hal-01796310.

[2] Cassandras, C.G. et S. Lafortune: *Introduction to discrete event systems*. Springer Science & Business Media, 2009.

[3] Contant, O., S. Lafortune et D.Teneketzis: *Diagnosis of intermittent faults*. Discrete Event Dynamic Systems, 14(2) :171–202, 2004.

[4] Doyen, L. et J.F. Raskin: *Games with Imperfect Information : Theory and Algorithms*. Dans *Lectures in Game Theory for Computer Scientists*, page 185–212, Cambridge, 2011.

[5] Grastien, A., R. Anbulagan, J. Rintanen et E. Kelareva: *Diagnosis of discrete-event systems using satisfiability algorithms*. Dans *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, tome 22, page 305, Vancouver, British Columbia, 2007.

[6] Grastien, A., M.O. Cordier et C. Largouët: *Incremental diagnosis of discrete-event systems*. Dans *Proceedings of the 29th International Workshop on Principles of Diagnosis DX'05*, Pacific Grove, CA, USA, 2005.

[7] Kleer, J. De: *Diagnosing multiple persistent and intermittent faults*. Dans *Proceedings of the 21th International Joint Conference on Artificial Intelligence IJCAI'19*, Pasadena, CA, USA, Juin 2009.

[8] Kurien, J. et P. Nayak: *Back to the future for consistency-based trajectory tracking*. Dans *Proceedings of the 17th AAAI Conference on Artificial Intelligence*, pages 370–377, Austin, Texas, USA, 2000.

[9] Moore, E.F.: *Gedanken-Experiments on Sequential Machines*. Dans Shannon, Claude et John McCarthy (rédacteurs) : *Automata Studies*, pages 129–153. Princeton University Press, Princeton, NJ, 1956.

[10] Ozveren, C. M. et A. S. Willsky: *Observability of discrete event dynamic systems*. IEEE Transactions on Automatic Control, 35(7) :797–806, Juillet 1990.

[11] Pralet, C., G. Verfaillie, M. Lemaître et G. Infantes: *Constraint-Based Controller Synthesis in Non-Deterministic and Partially Observable Domains*. Dans *Proceedings of the 2010 Conference on ECAI 2010 : 19th European Conference on Artificial Intelligence*, pages 681–686, Amsterdam, The Netherlands, 2010.

[12] Ramadge, P.J.: *Observability of discrete event systems*. Dans *Proceedings of the 25th IEEE Conference on Decision and Control CDC'86*, pages 1108–1112, Athens, Greece, 1986.

[13] Sampath, M., R. Sengupta, S. Lafortune, K. Sinnamohideen et D.Teneketzis: *Diagnosability of discrete-event systems*. IEEE Transactions on automatic control, 40(9) :1555–1575, 1995.

[14] Schumann, A., Y. Pencolé *et al.*: *Scalable Diagnosability Checking of Event-Driven Systems.* Dans *Proceedings of the 20th International Joint Conference on Artificial Intelligence IJCAI'00*, tome 7, pages 575–580, Hyderabad, India, 2007.

[15] V.Verma, T.Estlin, A.Jónsson, C. Pasareanu, R. Simmons et K. Tso: *Plan execution interchange language (PLEXIL) for executable plans and command sequences*. Dans *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space iSAIRAS'05*, Munich, Germany, 2005.

[16] Williams, B.C. et P. Nayak: *A Model-based Approach to Reactive Self-Configuring Systems*. Dans *Proceedings of the 13th AAAI Conference on Artificial Intelligence*, Portland, Oregon, février 1996.

[17] Wonham, W.M., K. Cai et K. Rudie: *Supervisory control of discrete-event systems : A brief history*. Annual Reviews in Control, 45 :250–256, 2018.

[18] Zaytoon, J. et S. Lafortune: *Overview of fault diagnosis methods for discrete event systems*. Annual Reviews in Control, 37(2) :308–320, 2013.

Actes JIAF 2019

Strengthening Neighbourhood Substitutability

Martin C. Cooper
IRIT, Université de Toulouse III, France
cooper@irit.fr

Résumé

La réduction de domaines est un outil essentiel dans la résoluion de problèmes de satisfaction de contraintes (CSP). Pour les CSP binaires, la substitution de voisinage constiste à éliminer une valeur s'il existe une autre valeur qui peut la remplacer dans chaque contrainte. Nous démontrons qu'il est possible de rendre plus forte la notion de substitution de voisinage de deux façons distinctes sans augmenter la complexité temporelle. Nous démontrons que contrairement à ce qui se passe pour la substitution de voisinage, trouver la séquence optimale de ces nouvelles opérations est NP-difficile.

Abstract

Domain reduction is an essential tool for solving the constraint satisfaction problem (CSP). In the binary CSP, neighbourhood substitution consists in eliminating a value if there exists another value which can be substituted for it in each constraint. We show that the notion of neighbourhood substitution can be strengthened in two distinct ways without increasing time complexity. We also show the theoretical result that, unlike neighbourhood substitution, finding an optimal sequence of these new operations is NP-hard.

1 Introduction

Domain reduction is classical in constraint satisfaction. Indeed, eliminating inconsistent values by what is now known as arc consistency [28] predates the first fomulation of the constraint satisfaction problem [23]. Maintaining arc consistency, which consists in eliminating values that can be proved inconsistent by examining a single constraint together with the current domains of the other variables, is ubiquitous in constriant solvers [1]. In binary CSPs, various algorithms have been proposed for enforcing arc consistency in $O(ed^2)$ time, where d denotes maximum domain size and e the number of constraints [25, 3]. Generic constraints on a number of variables which is unbounded are known as global constraints. Arc consistency can be efficiently enforced for many types of global constraints [17]. This has led to the development of efficient solvers providing a rich modelling language. Stronger notions of consistency have been proposed for domain reduction which lead to more eliminations but at greater computational cost [1, 2, 29].

In parallel, other research has explored domain-reduction methods that preserve satisfiability of the CSP instance but do not preserve the set of solutions. When searching for a single solution, all but one branch of the explored search tree leads to a dead-end, and so any method for faster detection of unsatisfiability is clearly useful. One family of satisfiability-preserving domain-reduction operations is value merging. For example, two values can be merged if the so-called broken triangle (BT) pattern does not occur on these two values [10]. Other value-merging rules have been proposed which allow less merging than BT-merging but at a lower cost [22] or more merging at a greater cost [11, 26]. Another family of satisfiability-preserving domain-reduction operations are based on the elimination of values that are not essential to obtain a solution [14]. The basic operation in this family which corresponds most closely to arc consistency is neighbourhood substitution: a value b can be eliminated from a domain if there is another value a in the same domain such that b can be replaced by a in each tuple in each constraint relation (reduced to the current domains of the other variables) [13]. In binary CSPs, neighbourhood substitution can be applied until convergence in $O(ed^3)$ time [6]. In this paper, we study notions of substitutability which are strictly stronger than neighbourhood substitutability but which can be applied in the same $O(ed^3)$ time

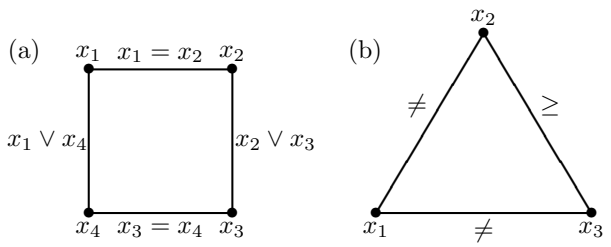


Figure 1: (a) A 4-variable CSP instance over boolean domains; (b) a 3-variable CSP instance over domains $\{0, 1, 2\}$ with constraints $x_1 \neq x_2$, $x_1 \neq x_3$ and $x_2 \geq x_3$.

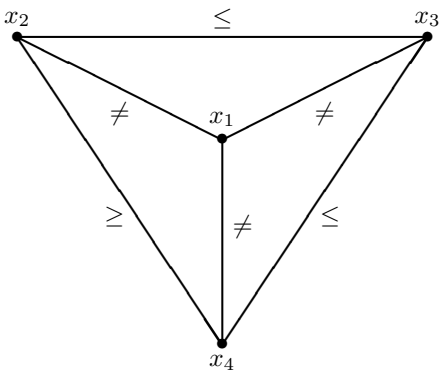


Figure 2: A 4-variable CSP instance over domain $\{0, 1, 2, 3\}$ with constraints $x_1 \neq x_2$, $x_1 \neq x_3$, $x_1 \neq x_4$, $x_2 \leq x_3$, $x_2 \geq x_4$ and $x_4 \leq x_3$.

complexity.

To illustrate the strength of the new notions of substitutability that we introduce in this paper, consider the instances shown in Figure 1 and Figure 2. These instances are all globally consistent (each variable-value assignment occurs in a solution) and neighbourhood subsitution is not powerful enough to eliminate any values. In this paper, we introduce three novel value-elimination rules, defined in Section 2: SS, CNS and SCSS. We will show that snake substitution (SS) allows us to reduce all domains to singletons in the instance in Figure 1(a). Using the notation $\mathcal{D}(x_i)$ for the domain of the variable x_i, conditioned neighbourhood-substitution (CNS), allows us to eliminate value 0 from $\mathcal{D}(x_2)$ and value 2 from $\mathcal{D}(x_3)$ in the instance shown in Figure 1(b), reducing the constraint between x_2 and x_3 to a null constraint (the complete relation $\mathcal{D}(x_2) \times \mathcal{D}(x_3)$). Snake-conditioned snake-substitution (SCSS) subsumes both SS and CNS and allows us to reduce all domains to singletons in the instance in Figure 2 (as well as both instances in Figure 1).

In Section 2 we define the substitution operations SS, CNS and SCSS. In Section 3 we prove the validity of these three substitution operations, in the sense that they define satisfiability-preserving value-elimination rules. In Section 4 we explain in detail the examples in Figures 1 and 2 and we give other examples from the semantic labelling of line drawings. Section 5 discusses the complexity of applying these value-elimination rules until convergence. Unlike NS, finding an optimal sequence of value eliminations by SS or CNS is NP-hard: this is shown in Section 6.

2 Definitions

We study binary constraint satisfaction problems. A *binary CSP instance* $I = (X, \mathcal{D}, R)$ comprises

- n variables $x_1, \ldots, x_n$,
- a domain $\mathcal{D}(x_i)$ for each variable x_i $(i = 1, \ldots, n)$, and
- a binary constraint relation R_{ij} for each pair of distinct variables x_i, x_j $(i, j \in \{1, \ldots, n\})$

For notational convenience, we assume that there is exactly one binary relation R_{ij} for each pair of variables. Thus, if x_i and x_j do not constrain each other, then we consider that there is a *trivial constraint* between them with $R_{ij} = \mathcal{D}(x_i) \times \mathcal{D}(x_j)$. Furthermore, R_{ji} (viewed as a boolean matrix) is always the transpose of R_{ij}. A *solution* to I is an n-tuple $s = \langle s_1, \ldots, s_n \rangle$ such that $\forall i \in \{1, \ldots, n\}$, $s_i \in \mathcal{D}(x_i)$ and for each distinct $i, j \in \{1, \ldots, n\}$, $(s_i, s_j) \in R_{ij}$.

We say that $v_i \in D(x_i)$ has a *support* at variable x_j if $\exists v_j \in D(x_j)$ such that $(v_i, v_j) \in R_{ij}$. A binary CSP instance I is *arc consistent* if for all pairs of distinct variables x_i, x_j, each $v_i \in D(x_i)$ has a support at x_j [20].

In the following we assume that we have a binary CSP instance $I = (X, \mathcal{D}, R)$ over n variables and, for clarity of presentation, we write $j \neq i$ as a shorthand for $i \in \{1, \ldots, n\} \setminus \{i\}$. We use the notation $b \xrightarrow{ij} a$ for

$$\forall c \in \mathcal{D}(x_j),\ (b, c) \in R_{ij} \Rightarrow (a, c) \in R_{ij}$$

(i.e. a can be substituted for b in any tuple $(b, c) \in R_{ij}$).

Definition 1 *[13] Given two values $a, b \in \mathcal{D}(x_i)$, b is* neighbourhood substitutable (NS) *by a if $\forall j \neq i$, $b \xrightarrow{ij} a$.*

It is well known and indeed fairly obvious that eliminating a neighbourhood substitutable value does not change the satisfiability of a binary CSP instance. We will now define stronger notions of substitutability. The proofs that these are indeed valid value-elimination rules are not directly obvious and hence

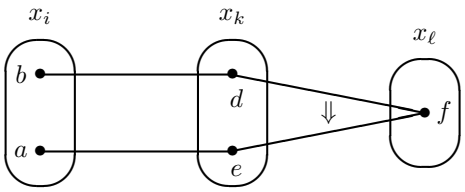


Figure 3: An illustration of the definition of $b \overset{ik}{\rightsquigarrow} a$.

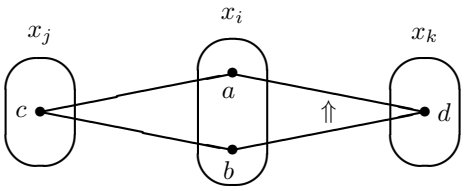


Figure 4: An illustration of the definition of conditioned neighbourhood-substitutability of b by a (conditioned by x_j).

are delayed until Section 3. We use the notation $b \overset{ik}{\rightsquigarrow} a$ for

$$\forall d \in \mathcal{D}(x_k),\ (b,d) \in R_{ik}\ \Rightarrow$$
$$\exists e \in \mathcal{D}(x_k)((a,e) \in R_{ik}\ \wedge\ \forall \ell \notin \{i,k\}, d \xrightarrow{k\ell} e).$$

This is illustrated in Figure 3, in which ovals represent domains, bullets represent values, a line joining two values means that these two values are compatible (so, for example, $(a,e) \in R_{ik}$), and the $\Uparrow$ means that $(e,f) \in R_{k\ell} \Rightarrow (d,f) \in R_{k\ell}$. Since e in this definition is a function of i,k,a and d, if necessary, we will write $e(i,k,a,d)$ instead of e. In other words, the notation $b \overset{ik}{\rightsquigarrow} a$ means that a can be substituted for b in any tuple $(b,d) \in R_{ik}$ provided we also replace d by $e(i,k,a,d)$. It is clear that $b \xrightarrow{ik} a$ implies $b \overset{ik}{\rightsquigarrow} a$ since it suffices to set $e(i,k,a,d) = d$ since, trivially, $d \xrightarrow{k\ell} d$ for all $\ell \notin \{i,k\}$.

Definition 2 *Given two values $a,b \in \mathcal{D}(x_i)$, b is* snake substitutable (SS) *by a if $\forall k \neq i$, $b \overset{ik}{\rightsquigarrow} a$.*

In the following two definitions, b can be eliminated from $\mathcal{D}(x_i)$ because it can be substituted by some other value in $\mathcal{D}(x_i)$, but this value is a function of the value assigned to another variable x_j. Definition 3 is illustrated in Figure 4.

Definition 3 *Given a value $b \in \mathcal{D}(x_i)$, b is* conditioned neighbourhood-substitutable (CNS) *if for some $j \neq i$, $\forall c \in \mathcal{D}(x_j)$ with $(b,c) \in R_{ij}$, $\exists a \in \mathcal{D}(x_i) \setminus \{b\}$ such that $((a,c) \in R_{ij}\ \wedge\ \forall k \notin \{i,j\}, b \xrightarrow{ik} a)$.*

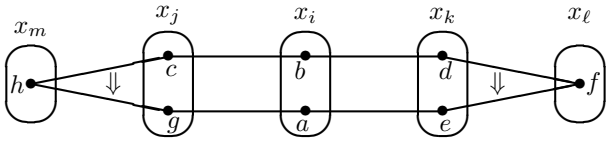


Figure 5: An illustration of the definition of snake-conditioned snake-substitutability of b by a.

A CNS value $b \in \mathcal{D}(x_i)$ is substitutable by a value $a \in \mathcal{D}(x_i)$ where a is a function of the value c assigned to some other variable x_j. Observe that CNS subsumes arc consistency; if a value $b \in \mathcal{D}(x_i)$ has no support c in $\mathcal{D}(x_j)$, then b is trivially CNS (conditioned by the variable x_j). It is easy to see from their definitions that SS and CNS both subsume NS (in instances with more than one variable), but that neither NS nor SS subsume arc consistency.

We now integrate the notion of snake substitutability in two ways in the definition of CNS: the value d (see Figure 4) assigned to a variable $k \notin \{i,j\}$ may be replaced by a value e (as in the definition of $b \overset{ik}{\rightsquigarrow} a$, above), but the value c (see Figure 4) assigned to the conditioning variable x_j may also be replaced by a value g. This is illustrated in Figure 5.

Definition 4 *Given a value $b \in \mathcal{D}(x_i)$, b is* snake-conditioned snake-substitutable (SCSS) *if for some $j \neq i$, $\forall c \in \mathcal{D}(x_j)$ with $(b,c) \in R_{ij}$, $\exists a \in \mathcal{D}(x_i) \setminus \{b\}$ such that $(\forall k \notin \{i,j\}, b \overset{ik}{\rightsquigarrow} a\ \wedge\ (\exists g \in \mathcal{D}(x_j)((a,g) \in R_{ij}\ \wedge\ \forall m \notin \{i,j\}, c \xrightarrow{jm} g)))$.*

We can see that SCSS subsumes CNS by setting $g = c$ in Definition 4 and by recalling that $b \xrightarrow{ik} a$ implies that $b \overset{ik}{\rightsquigarrow} a$. It is a bit more subtle to see that SCSS subsumes SS: if b is snake substititable by some value a, it suffices to choose a in Definition 4 to be this value (which is thus constant i.e. not dependent on the value of c), then the snake substitutability of b by a implies that $b \overset{ik}{\rightsquigarrow} a$ for all $k \neq i,j$ and $b \overset{ij}{\rightsquigarrow} a$, which in turn implies that $(a,g) \in R_{ij}\ \wedge\ \forall m \notin \{i,j\}, c \xrightarrow{jm} g$ for $g = e(i,j,a,c)$; thus b is snake-conditioned snake-substitutable.

3 Value elimination

It is well-known that NS is a valid value-elimination property, in the sense that if $b \in \mathcal{D}(x_i)$ is neighbourhood substitutable by a then b can be eliminated from $\mathcal{D}(x_i)$ without changing the satisfiability of the CSP instance [13]. In this section we show that SCSS is a valid value-elimination property. Since SS and CNS

are subsumed by SCSS, it follows immediately that SS and CNS are also valid value-elimination properties.

Theorem 1 *In a binary CSP instance I, if $b \in \mathcal{D}(x_i)$ is snake-conditioned snake-substitutable then b can be eliminated from $\mathcal{D}(x_i)$ without changing the satisfability of the instance.*

Proof: By Definition 4, for some $j \neq i$, $\forall c \in \mathcal{D}(x_j)$ with $(b,c) \in R_{ij}$, $\exists a \in \mathcal{D}(x_i) \setminus \{b\}$ such that

$$\forall k \notin \{i,j\}, b \overset{ik}{\rightsquigarrow} a \quad \wedge \tag{1}$$

$$\exists g \in \mathcal{D}(x_j)((a,g) \in R_{ij} \ \wedge \ \forall m \notin \{i,j\}, c \xrightarrow{jm} g) \tag{2}$$

We will only apply this definition for fixed i, j, and for fixed values a and c, so we can consider g as a constant (even though it is actually a function of i, j, a, c). Let $s = \langle s_1, \ldots, s_n \rangle$ be a solution to I with $s_i = b$. It suffices to show that there is another solution $t = \langle t_1, \ldots, t_n \rangle$ with $t_i \neq b$. Consider $c = s_j$. Since s is a solution, we know that $(b,c) = (s_i, s_j) \in R_{ij}$. Thus, according to the above definition of SCSS, there is a value $a \in \mathcal{D}(x_i)$ that can replace b (conditioned by the assignment $x_j = c = s_j$) in the sense that (1) and (2) are satisfied. Now, for each $k \notin \{i,j\}$, $b \overset{ik}{\rightsquigarrow} a$, i.e.

$$\forall d \in \mathcal{D}(x_k), \ (b,d) \in R_{ik} \ \Rightarrow$$
$$\exists e \in \mathcal{D}(x_k)((a,e) \in R_{ik} \ \wedge \ \forall \ell \notin \{i,k\}, d \xrightarrow{k\ell} e).$$

Recall that e is a function of i, k, a and d. But we will only consider fixed i, a and a unique value of d dependant on k, so we will write $e(k)$ for brevity. Indeed, setting $d = s_k$ we can deduce from $(b,d) = (s_i, s_k) \in R_{ik}$ (since s is a solution) that $\forall k \neq i, j$,

$$\exists e(k) \in \mathcal{D}(x_k)((a,e(k)) \in R_{ik} \wedge \forall \ell \notin \{i,k\}, s_k \xrightarrow{k\ell} e(k)). \tag{3}$$

Define the n-tuple t as follows:

$$t_r = \begin{cases} a & \text{if } r = i \\ s_r & \text{if } r \neq i \ \wedge \ (a,s_r) \in R_{ir} \\ g & \text{if } r = j \ \wedge \ (a,s_r) \notin R_{ir} \\ e(r) & \text{if } r \neq i,j \ \wedge \ (a,s_r) \notin R_{ir} \end{cases}$$

Clearly $t_i \neq b$ and $t_i \in \mathcal{D}(x_r)$ for all $r \in \{1, \ldots, n\}$. To prove that t is a solution, it remains to show that all binary constraints are satisfied, i.e. that $(t_k, t_r) \in R_{kr}$ for all distinct $k, r \in \{1, \ldots, n\}$. There are three cases: (1) $k = i$, $r \neq i$, (2) $k = j$, $r \neq i, j$, (3) $k, r \neq i, j$.

(1) There are three subcases: (a) $r = j$ and $(a, s_j) \notin R_{ij}$, (b) $r \neq i$ and $(a, s_r) \in R_{ir}$, (c) $r \neq i, j$ and $(a, s_r) \notin R_{ir}$. In case (a), $t_i = a$ and $t_j = g$, so from equation 2, we have $(t_i, t_r) = (a, g) \in R_{ij}$. In case (b), $t_i = a$ and $t_r = s_r$ and so, trivially, $(t_i, t_r) = (a, s_r) \in R_{ir}$. In case (c), $t_i = a$ and $t_r = e(r)$, so from equation 3, we have $(t_i, t_r) = (a, e(r)) \in R_{ir}$.

(2) There are four subcases: (a) $(a, s_r) \in R_{ir}$ and $(a, s_j) \in R_{ij}$, (b) $(a, s_r) \notin R_{ir}$ and $(a, s_j) \in R_{ij}$, (c) $(a, s_r) \in R_{ir}$ and $(a, s_j) \notin R_{ij}$, (d) $(a, s_r) \notin R_{ir}$ and $(a, s_j) \notin R_{ij}$. In case (a), $t_j = s_j$ and $t_r = s_r$, so $(t_j, t_r) \in R_{jr}$ since s is a solution. In case (b), $t_j = s_j$ and $t_r = e(r)$; setting $k = r$, $\ell = j$ in equation 3, we have $(t_j, t_r) = (s_j, e(r)) \in R_{jr}$ since $(s_j, s_r) \in R_{jr}$. In case (c), $t_j = g$ and $t_r = s_r$; setting $c = s_j$ and $m = r$ in equation 2 we can deduce that $(t_j, t_r) = (g, s_r) \in R_{jr}$ since $(s_j, s_r) \in R_{jr}$. In case (d), $t_j = g$ and $t_r = e(r)$. By the same argument as in case 2(b), we know that $(s_j, e(r)) \in R_{jr}$, and then setting $c = s_j$ and $m = r$ in equation 2, we can deduce that $(t_j, t_r) = (g, e(r)) \in R_{jr}$.

(3) There are three essentially distinct subcases: (a) $(a, s_r) \in R_{ir}$ and $(a, s_k) \in R_{ik}$, (b) $(a, s_r) \notin R_{ir}$ and $(a, s_k) \in R_{ik}$, (c) $(a, s_r) \notin R_{ir}$ and $(a, s_k) \notin R_{ik}$. In cases (a) and (b) we can deduce $(t_k, t_r) \in R_{kr}$ by the same arguments as in cases 2(a) and 2(b), above. In case (c), $t_k = e(k)$ and $t_r = e(k)$. Setting $\ell = r$ in equation 3, we have $s_k \xrightarrow{kr} e(k)$ from which we can deduce that $(e(k), s_r) \in R_{kr}$ since $(s_k, s_r) \in R_{kr}$. Reversing the roles of k and r in equation 3 (which is possible since they are distinct and both different to i and j), we also have that $s_r \xrightarrow{rk} e(r)$. We can then deduce that $(t_k, t_r) = (e(k), e(r)) \in R_{kr}$ since we have just shown that $(e(k), s_r) \in R_{kr}$.

We have thus shown that any solution s with $s_i = b$ can be transformed into another solution t that does not assign the value b to x_i and hence that the elimination of b from $\mathcal{D}(x_i)$ preserves satisfiability. ∎

Corollary 1 *In a binary CSP instance I, if $b \in \mathcal{D}(x_i)$ is snake-substitutable or conditioned neighbourhood substitutable, then b can be eliminated from $\mathcal{D}(x_i)$ without changing the satisfability of the instance.*

4 Examples

We illustrate the power of SS, CNS and SCSS using the examples given in Figure 1 and Figure 2. In Figure 1(a), the value $0 \in \mathcal{D}(x_1)$ is snake substitutable by 1: we have $0 \overset{12}{\rightsquigarrow} 1$ by taking $e(1,2,1,0) = 1$ (where the arguments of $e(i,k,a,d)$ are as shown in Figure 3), since $(1,1) \in R_{12}$ and $0 \xrightarrow{23} 1$; and $0 \overset{14}{\rightsquigarrow} 1$ since $0 \xrightarrow{14} 1$. Indeed, by a similar argument, the value 0 is snake substitutable by 1 in each domain. In Figure 1(b),

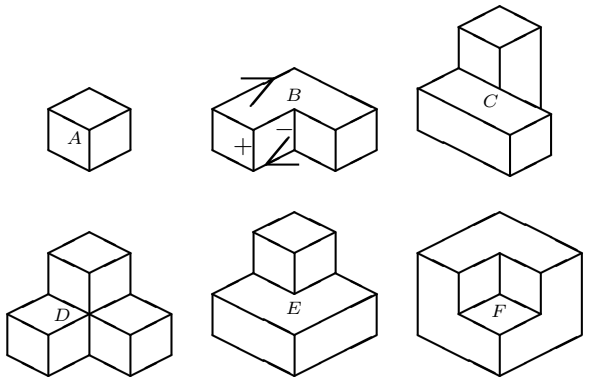


Figure 6: The six different types of trihedral vertices: A, B, C, D, E, F.

the value $0 \in \mathcal{D}(x_2)$ is conditioned neighbourhood-substitutable (CNS) with x_1 as the conditioning variable (i.e. $j = 1$ in Definition 3): for the assignments of 0 or 1 to x_1, we can take $a = 2$ since $0 \xrightarrow{23} 2$, and for the assignment 2 to x_1, we can take $a = 1$ since $0 \xrightarrow{23} 1$. By a symmetrical argument, the value $2 \in \mathcal{D}(x_3)$ is CNS, again with x_1 as the conditioning variable. We can note that in the resulting CSP instance, after eliminating 0 from $\mathcal{D}(x_2)$ and 2 from $\mathcal{D}(x_3)$, all domains can be reduced to singletons by applying snake substitutability.

In Figure 2, the value $3 \in \mathcal{D}(x_1)$ is snake-conditioned snake-substitutable (SCSS) with x_2 as the conditioning variable: for the assignment of 0 or 2 to x_2, we can take $a = 1$ since $3 \overset{13}{\rightsquigarrow} 1$ (taking $e(1,3,1,d) = 3$ for $d = 0,1,2$) and $3 \overset{14}{\rightsquigarrow} 1$ (taking $e(1,4,1,d) = 0$ for $d = 0,1,2$), and for the assignment of 1 to x_2, we can take $a = 2$ since $3 \overset{13}{\rightsquigarrow} 2$ (again taking $e(1,3,2,d) = 3$ for $d = 0,1,2$) and $3 \overset{14}{\rightsquigarrow} 2$ (again taking $e(1,4,2,d) = 0$ for $d = 0,1,2$). By similar arguments, all domains can be reduced to singletons following the SCSS elimination of values in the following order: 0 from $\mathcal{D}(x_1)$, 0, 1 and 2 from $\mathcal{D}(x_3)$, 0, 1 and 2 from $\mathcal{D}(x_2)$, 1, 2 and 3 from $\mathcal{D}(x_4)$ and 2 from $\mathcal{D}(x_1)$.

To give a non-numerical example, we considered the impact of SS and CNS in the classic problem of labelling line-drawings of polyhedral scenes composed of objects with trihedral vertices [4, 18, 28]. There are six types of trihedral vertices: A, B, C, D, E and F, shown in Figure 6. The aim is to assign each line in the drawing a semantic label among four possibilities: convex (+), concave (−) or occluding ($\leftarrow$ or $\rightarrow$ depending whether the occluding surface is above or below the line). Some lines in the top middle drawing in Figure 6 have been labelled to illustrate the meaning of these labels. This problem can be expressed as

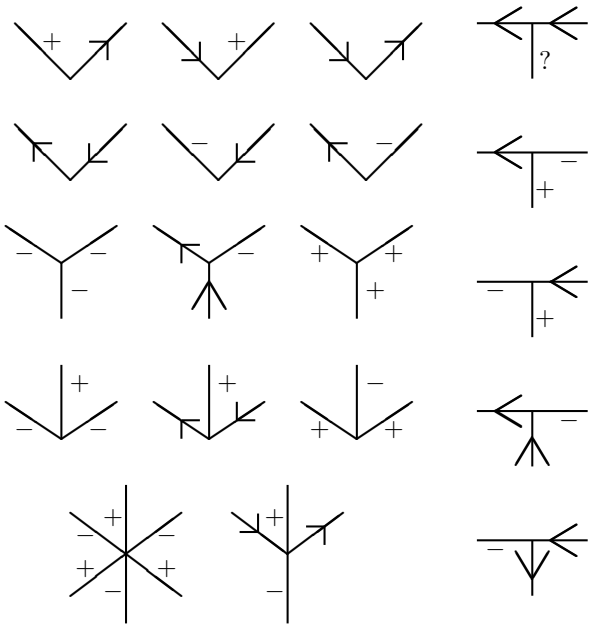


Figure 7: The catalogue of labelled junctions that are projections of trihedral vertices.

a binary CSP by treating the junctions as variables. The domains of variables are given by the catalogue of physically realisable labellings of the corresponding junction according to its type. This catalogue of junction labellings is obtained by considering the six vertex types viewed from all possible viewpoints [4, 18]. For example, there are 6 possible labellings of an L-junction, 8 for a T-junction, 5 for a Y-junction and 3 for a W-junction [9]. The complete catalogue of labelled junctions is shown in Figure 7, where a question mark represents any of the four labels and rotationally symmetric labellings are omitted. There is a constraint between any two junctions joined by a line: this line must have the same semantic label at both ends. We can also apply binary constraints between distant junctions: the 2Reg constraint limits the possible labellings of junctions such as A and D in Figure 8, since two non-colinear lines, such as AB and CD, which separate the same two regions cannot both be concave [8, 9].

The drawing shown in Figure 8 is ambiguous. For example, any of lines AB, BC or CD could be projections of concave edges (meaning that the two blocks on the left side of the figure are part of the same object) or all three could be projections of occluding edges (meaning that these two blocks are, in fact, separate objects). The drawing shown in Figure 8 is an exam-

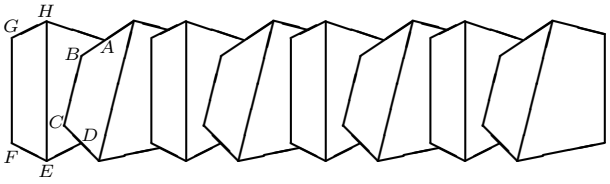


Figure 8: An example from a family of line drawings whose exponential number of labellings is reduced to one by snake substitution.

ple of a family of line drawings. In this figure there are four copies of the basic structure, but there is a clear generalisation to drawings containing n copies of this basic structure. The ambiguity that we have pointed out above gives rise to an exponential number of valid labellings for this family of drawings. However, after applying arc consistency and snake substitution until convergence, each domain is a singleton for this family of line drawings.

Of course, there are line drawings where snake substution is much less effective than in Figure 8. Nevertheless, in the six drawings in Figure 6, which are a representative sample of simple line drawings, 22 of the 73 junctions have their domains reduced to singletons by arc consistency alone and a further 20 junctions have their domains reduced to singletons when both arc consistency and snake substitution are applied. This can be compared with neighbourhood substitution which eliminates no domain values in this sample of six drawings. It should be mentioned that we found no examples where conditioned neighbourhood substitution could lead to the elimination of labellings in the line-drawing labelling problem.

5 Complexity

In a binary CSP instance $(X, \mathcal{D}, R)$, we say that two variables $x_i, x_j \in X$ constrain each other if there is a non-trivial constraint between them (i.e. $R_{ij} \neq \mathcal{D}(x_i) \times \mathcal{D}(x_j)$). Let e denote the number of pairs $\{i, j\}$ such that x_i, x_j constrain each other. We use d to denote the maximum size of the domains $\mathcal{D}(x_i)$.

Using standard data structures and techniques inspired by efficient algorithms for arc consistency and neighbouhood substitution, we obtain the complexities for CNS and SCSS, as given below. Proofs are omitted due to space limitations.

Theorem 2 *Value eliminations by snake substitution can be applied until convergence in $O(ed^3)$ time and $O(ed^2)$ space.*

Theorem 3 *Value eliminations by conditioned neighbourhood substitution can be applied until convergence in $O(ed^3)$ time and $O(ed^2)$ space.*

Thus, the complexity of applying the value-elimination rules CNS and SS is comparable to the $O(ed^3)$ time complexity of applying neighbourhood substitution (NS) [6]. This is interesting because (in instances with more than one variable) CNS and SS both strictly subsume NS.

Theorem 4 *It is possible to verify in $O(ed^3)$ time and $O(ed^2)$ space whether or not any value eliminations by SCSS can be performed on a binary CSP instance. Value eliminations by SCSS can then be applied until convergence in $O(end^5)$ time and $O(ed^2)$ space.*

6 Optimal sequences of eliminations

It is known that applying different sequences of neighbourhood operations until convergence produces isomorphic instances [6]. This is not the case for CNS, SS or SCSS. Indeed, as we show in this section, the problems of maximising the number of value-eliminations by CNS, SS or SCSS are all NP-hard. These intractability results do not detract from the utility of these operations, since any number of value eliminations reduces search-space size regardless of whether or not this number is optimal.

Theorem 5 *Finding the longest sequence of CNS value-eliminations or SCSS value-eliminations is NP-hard.*

Proof: We prove this by givng a polynomial reduction from the set cover problem [19], the well-known NP-complete problem which, given sets $S_1, \ldots, S_m \subseteq U$ and an integer k, consists in determining whether there are k sets $S_{i_1}, \ldots, S_{i_k}$ which cover U (i.e. such that $S_{i_1} \cup \ldots \cup S_{i_k} = U$). We can assume that $S_1 \cup \ldots \cup S_m = U$ and $k < m$, otherwise the problem is trivially solvable. Given sets $S_1, \ldots, S_m \subseteq U$, we create a 2-variable CSP instance with $\mathcal{D}(x_1) = \{1, \ldots, m\}$, $\mathcal{D}(x_2) = U$ and $R_{12} = \{(i, u) \mid u \in S_i\}$. We can eliminate value i from $\mathcal{D}(x_1)$ by CNS (with, of course, x_2 as the conditioning variable) if and only if $S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, m$ cover U. Indeed, we can continue eliminating elements from $\mathcal{D}(x_1)$ by CNS provided the sets S_j ($j \in \mathcal{D}(x_1)$) still cover U. Clearly, maximising the number of eliminations from $\mathcal{D}(x_1)$ by CNS is equivalent to minimising the size of the cover. To prevent any eliminations from the domain of x_2 by CNS, we add variables x_3 and x_4 with domains $\{1, \ldots, m\}$, together with the three equality

constraints $x_2 = x_3$, $x_3 = x_4$ and $x_4 = x_2$. To complete the proof for CNS, it is sufficient to observe that this reduction is polynomial.

It is easily verified that in this instance, CNS and SCSS are equivalent. Hence, this proof also shows that finding the longest sequence of SCSS value-eliminations is NP-hard. ∎

In the proof of the following theorem, we need the following notion: we say that a sequence of value-eliminations by snake-substitution (SS) is *convergent* if no more SS value-eliminations are possible after this sequence of eliminations is applied.

Theorem 6 *Finding a longest sequence of snake-substitution value-eliminations is NP-hard.*

Proof: It suffices to demonstrate a polynomial reduction from the problem Max 2-Sat which is known to be NP-hard [15]. Consider an instance I_{2SAT} of Max 2-Sat with variables $X_1, \ldots, X_N$ and M binary clauses: the goal is to find a truth assignment to these variables which maximises the number of satisfied clauses. We will construct a binary CSP instance I_{CSP} on $O(N + M)$ variables, each with domain of size at most four, such that the convergent sequences S of SS value-eliminations in I_{CSP} correspond to truth assignments to $X_1, \ldots, X_N$ and the length of S is $\alpha N + \beta m$ where α, β are constants and m is the number of clauses of I_{2SAT} satisfied by the corresponding truth assignment.

We require four constructions (which we explain in detail below):

1. the construction in Figure 9 simulates a Max 2-Sat literal X by a path of CSP variables joined by greater-than-or-equal-to constraints.
2. the construction in Figure 10 simulates the relationship between a Max 2-Sat variable X and its negation $\overline{X}$.
3. the construction in Figure 11 allows us to create multiple copies of a Max 2-Sat literal X.
4. the construction in Figure 12 simulates a binary clause $X \vee Y$ where X, Y are Max 2-Sat literals.

In each of these figures, each oval represents a CSP variable with the bullets inside the oval representing the possible values for this variable. If there is a non-trivial constraint between two variables x_i, x_j this is represented by joining up with a line those pairs of values a, b such that $(a, b) \in R_{ij}$. Where the constraint has a compact form, such as $x_1 \geq x_2$ this is written next to the constraint. In the following, we write $b \overset{x_i}{\rightsquigarrow} a$ if $b \in \mathcal{D}(x_i)$ is snake substitutable by $a \in \mathcal{D}(x_i)$. Our constructions are such that the only value that can be eliminated from any domain by SS is the value 2.

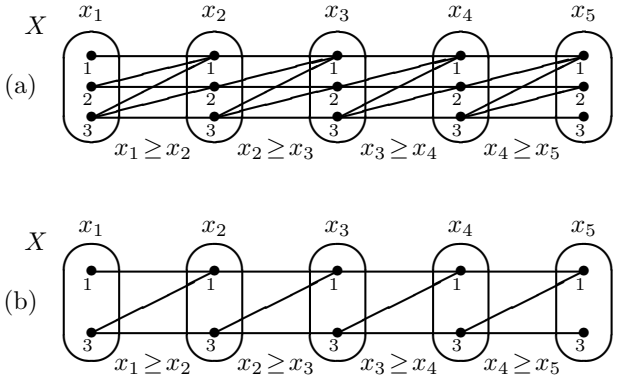


Figure 9: A construction to simulate a Max 2-Sat variable X: (a) $X = 0$, (b) $X = 1$.

Figure 9(a) shows a path of CSP variables constrained by greater-than-or-equal-to constraints. The end variables x_1 and x_5 are constrained by other variables that, for clarity of presentation, are not shown in this figure. If value 2 is eliminated from $\mathcal{D}(x_1)$, then we have $2 \overset{x_2}{\rightsquigarrow} 3$. In fact, 2 is neighbourhood substitutable by 3. Once the value 2 is eliminated from $\mathcal{D}(x_2)$, we have $2 \overset{x_3}{\rightsquigarrow} 3$. Indeed, eliminations of the value 2 propagate so that in the end we have the situation shown in Figure 9(b). By a symmetrical argument, the elimination of the value 2 from $\mathcal{D}(x_5)$ propagates from right to left (this time by neighbourhood substitution by 1) to again produce the situation shown in Figure 9(b). It is easily verified that, without any eliminations from the domains $\mathcal{D}(x_1)$ or $\mathcal{D}(x_5)$, no values for the variables x_2, x_3, x_4 are snake-substitutable. Furthermore, the values 1 and 3 for the variables x_2, x_3, x_4 are not snake-substitutable even after the elimination of the value 2 from all domains. So we either have no eliminations, which we associate with the truth assignment $X = 0$ (where X is the Max 2-Sat literal corresponding to this path of variables in I_{CSP}) or the value 2 is eliminated from all domains, which we associate with the truth assignment $X = 1$.

The construction in Figure 10 joins the two path-of-CSP-variables constructions corresponding to the literals X and $\overline{X}$. This construction ensures that exactly one of X and $\overline{X}$ are assigned the value 1. It is easy (if tedious) to verify that the only snake substitutions that are possible in this construction are $2 \overset{x_0}{\rightsquigarrow} 3$ and $2 \overset{\tilde{x}_0}{\rightsquigarrow} 3$, but that after elimination of the value 2 from either of $\mathcal{D}(x_0)$ or $\mathcal{D}(\tilde{x}_0)$, the other snake substitution is no longer valid. Once, for example, 2 has been eliminated from $\mathcal{D}(x_0)$, then this elimination propagates along the path of CSP variables $(x_1, x_2, x_3, \ldots)$ corresponding to X, as shown in Figure 9(b). By a symmet-

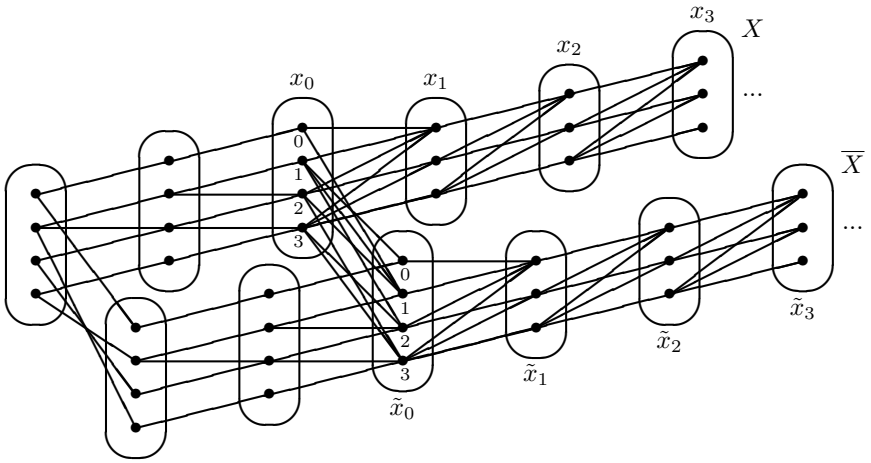


Figure 10: A construction to simulate a Max 2-Sat variable X and its negation $\overline{X}$.

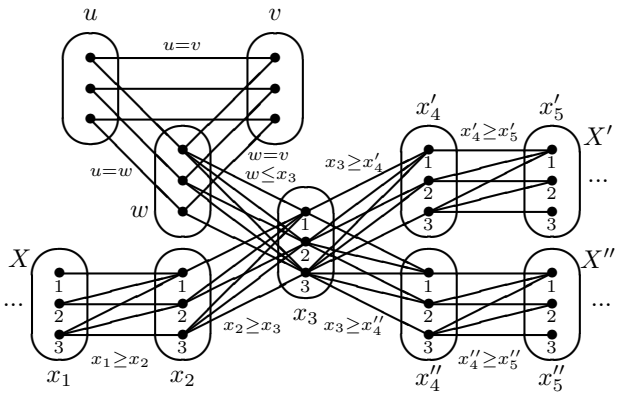


Figure 11: A construction to create two copies X' and X'' of the Max 2-Sat variable X.

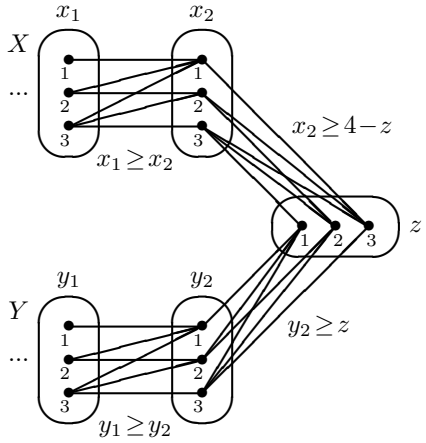


Figure 12: A construction to simulate a Max 2-Sat clause $X \vee Y$.

rical argument, if 2 is eliminated from $\mathcal{D}(\tilde{x}_0)$, then this elimination propagates along the path of CSP variables $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \ldots)$ corresponding to $\overline{X}$. Thus, this construction simulates the assignment of a truth value to X and its complement to $\overline{X}$.

Since any literal of I_{2SAT} may occur in several clauses, we need to be able to make copies of any literal. Figure 11 shows a construction that creates two copies X', X'' of a literal X. This construction can easily be generalised to make k copies of a literal, if required, by having k identical paths of greater-than-equal-to constraints on the right of the figure all starting at the pivot variable x_3. Before any eliminations are performed, no snake substitutions are possible in this construction. However, once the value 2 has been eliminated from $\mathcal{D}(x_1)$, eliminations propagate, as in Figure 9: the value 2 can successively be eliminated from the domains of variables x_2, x_3, x'_4, x'_5, and x''_4, x''_5. Each elimination is in fact by neighbourhood substitution, as in Figure 9. These eliminations mean that we effectively have two copies X', X'' of the literal X. The triangle of equality constraints at the top left of this construction is there simply to prevent propagation in the reverse direction: even if the value 2 is eliminated from the domains of x'_5, x'_4 and x''_5, x''_4 by the propagation of eliminations from the right, this cannot provoke the elimination of the value 2 from the domain of the pivot variable x_3.

Finally, the construction of Figure 12 simulates the clause $X \vee Y$. In fact, this construction simply joins

together the paths of CSP-variables corresponding to the two literals X, Y, via a variable z. It is easily verified that the elimination of the value 2 from the domain of x_1 allows the propagation of eliminations of the value 2 from the domains of x_2, z, y_2, y_1 in exactly the same way as the propagation of eliminations in Figure 9. Similarly, the elimination of the value 2 from the domain of y_1 propagates to all other variables in the opposite order y_2, z, x_2, x_1. Thus, if one or other of the literals X or Y in the clause is assigned 1, then the value 2 is eliminated from all domains of this construction. Eliminations can propagate back up to the pivot variable (x_3 in Figure 11) but no further, as explained in the previous paragraph.

Putting all this together, we can see that there is a one-to-one correspondence between convergent sequences of SS value-eliminations and truth assignments to the variables of the MAX 2-SAT instance. Furthermore, the number of SS value-eliminations is maximised when this truth assignment maximises the number of satisfied clauses, since it is $\alpha N+\beta m$ where α is the number of CSP-variables in each path of greater-than-or-equal-to constraints corresponding to a literal, β is the number of CSP-variables in each clause construction and m is the number of satisfied clauses. This reduction is clearly polynomial. ∎

7 Discussion and Conclusion

We have given two different value-elimination rules, namely snake substitutability (SS) and conditioned neighbourhood substitutability (CNS), which strictly subsume neighbourhood substitution but nevertheless can be applied in the same $O(ed^3)$ time complexity. We have also given a more general notion of substitution (SCSS) subsuming both these rules that can be detected in $O(ed^3)$ time. The examples in Figures 1 and 2 show that these three rules are strictly stronger than neighborhood substitution and that SS and CNS are incomparable. We found snake substitution to be particularly effective when applied to the problem of labelling line-drawings of polyhedral scenes.

Further research is required to investigate generalisations of SS, CNS or SCSS to non-binary or even global constraints. Another obvious avenue of research is the generalisation to valued CSPs (also known as cost-function networks). It is known that the generalisation of neighbourhood substitution to binary valued CSPs [16, 21] can be applied to convergence in $O(ed^4)$ time if the aggregation operator is strictly monotonic or idempotent [7]. The notion of snake substitutability has already been generalised to binary valued CSPs and it has been shown that it is possible to test this notion in $O(ed^4)$ time if the aggregation operator is addition over the non-negative rationals (which is a particular example of a strictly monotonic operator) [12]. However, further research is required to determine the complexity of applying this operation until convergence.

It is known that it is possible to efficiently find all (or a given number of) solutions to a CSP after applying neighbourhood substitution: given the set of all solutions to the reduced instance, it is possible to reconstruct $K \geq 1$ solutions to the original instance I (or to determine that I does not have K solutions) in $O(K(de + n^2))$ time [6]. This also holds for the case of conditioned neighbourhood substitution, since, as for neighbourhood substitution, for each solution s found and for each value b eliminated from some domain $\mathcal{D}(x_i)$, it suffices to test each putative solution obtained by replacing s_i by b. Unfortunately, the extra strength of snake substitution (SS) is here a drawback, since, by exactly the same argument as for the $\exists 2snake$ value-elimination rule (which is a weaker version of SS) [5], we can deduce that determining whether a binary CSP instance has two or more solutions is NP-hard, even given the set of solutions to the reduced instance after applying SS.

This work begs the interesting theoretical question as to the existence of reduction operations which strengthen other known reduction operations without increasing complexity.

References

[1] Bessiere, C.: Constraint propagation. In: Rossi et al. [27], pp. 29–83

[2] Bessière, C., Debruyne, R.: Optimal and suboptimal singleton arc consistency algorithms. In: Kaelbling, L.P., Saffiotti, A. (eds.) IJCAI-05. pp. 54–59. Professional Book Center (2005)

[3] Bessière, C., Régin, J., Yap, R.H.C., Zhang, Y.: An optimal coarse-grained arc consistency algorithm. Artif. Intell. **165**(2), 165–185 (2005)

[4] Clowes, M.B.: On seeing things. Artif. Intell. **2**(1), 79–116 (1971)

[5] Cohen, D.A., Cooper, M.C., Escamocher, G., Zivny, S.: Variable and value elimination in binary constraint satisfaction via forbidden patterns. J. Comput. Syst. Sci. **81**(7), 1127–1143 (2015)

[6] Cooper, M.C.: Fundamental properties of neighbourhood substitution in constraint satisfaction problems. Artif. Intell. **90**(1-2), 1–24 (1997)

[7] Cooper, M.C.: Reduction operations in fuzzy or valued constraint satisfaction. Fuzzy Sets and Systems **134**(3), 311–342 (2003)

[8] Cooper, M.C.: Constraints between distant lines in the labelling of line drawings of polyhedral scenes. International Journal of Computer Vision **73**(2), 195–212 (2007)

[9] Cooper, M.C.: Line Drawing Interpretation. Springer (2008)

[10] Cooper, M.C., Duchein, A., Mouelhi, A.E., Escamocher, G., Terrioux, C., Zanuttini, B.: Broken triangles: From value merging to a tractable class of general-arity constraint satisfaction problems. Artif. Intell. **234**, 196–218 (2016)

[11] Cooper, M.C., El Mouelhi, A., Terrioux, C.: Extending Broken Triangles and Enhanced Value-Merging. In: Rueher, M. (ed.) CP 2016. Lecture Notes in Computer Science, vol. 9892, pp. 173–188. Springer (2016)

[12] Cooper, M.C., Jguirim, W., Cohen, D.A.: Domain reduction for valued constraints by generalising methods from CSP. In: Hooker, J.N. (ed.) CP 2018. Lecture Notes in Computer Science, vol. 11008, pp. 64–80. Springer (2018)

[13] Freuder, E.C.: Eliminating interchangeable values in constraint satisfaction problems. In: Dean, T.L., McKeown, K.R. (eds.) Proc. 9th National Conference on Artificial Intelligence, Vol. 1. pp. 227–233. AAAI Press / The MIT Press (1991)

[14] Freuder, E.C., Wallace, R.J.: Replaceability and the substitutability hierarchy for constraint satisfaction problems. In: Benzmüller, C., Lisetti, C.L., Theobald, M. (eds.) GCAI 2017. EPiC Series in Computing, vol. 50, pp. 51–63. EasyChair (2017)

[15] Garey, M.R., Johnson, D.S., Stockmeyer, L.J.: Some simplified np-complete graph problems. Theor. Comput. Sci. **1**(3), 237–267 (1976)

[16] Goldstein, R.F.: Efficient rotamer elimination applied to protein side-chains and related spin glasses. Biophysical Journal **66**(5), 1335–1340 (1994)

[17] van Hoeve, W., Katriel, I.: Global constraints. In: Rossi et al. [27], pp. 169–208

[18] Huffman, D.A.: Impossible objects as nonsense sentences. In: Meltzer, B., Michie, D. (eds.) Machine Intelligence, vol. 6, pp. 295–323. Edinburgh University Press (1971)

[19] Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Proceedings symposium on the Complexity of Computer Computations. pp. 85–103. Plenum Press, New York (1972)

[20] Lecoutre, C.: Constraint Networks Techniques and Algorithms. ISTE/Wiley (2009)

[21] Lecoutre, C., Roussel, O., Dehani, D.E.: WCSP integration of soft neighborhood substitutability. In: Milano [24], pp. 406–421

[22] Likitvivatanavong, C., Yap, R.H.C.: Many-to-many interchangeable sets of values in CSPs. In: Shin, S.Y., Maldonado, J.C. (eds.) Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13. pp. 86–91. ACM (2013)

[23] Mackworth, A.K.: Consistency in networks of relations. Artif. Intell. **8**(1), 99–118 (1977)

[24] Milano, M. (ed.): CP 2012, Lecture Notes in Computer Science, vol. 7514. Springer (2012)

[25] Mohr, R., Henderson, T.C.: Arc and path consistency revisited. Artif. Intell. **28**(2), 225–233 (1986)

[26] Naanaa, W.: New schemes for simplifying binary constraint satisfaction problems. https://hal.archives-ouvertes.fr/hal-01731250 (2018)

[27] Rossi, F., van Beek, P., Walsh, T. (eds.): Handbook of Constraint Programming, Foundations of Artificial Intelligence, vol. 2. Elsevier (2006)

[28] Waltz, D.: Understanding line drawings of scenes with shadows. In: Winston, P.H. (ed.) The Psychology of Computer Vision, pp. 19–91. Computer Science Series, McGraw-Hill (1975)

[29] Woodward, R.J., Karakashian, S., Choueiry, B.Y., Bessiere, C.: Revisiting neighborhood inverse consistency on binary CSPs. In: Milano [24], pp. 688–703

Actes JIAF 2019

When 'knowing whether' is better than 'knowing that'

M. Cooper[1] **A. Herzig**[1] **F. Maffre**[2] **F. Maris**[1] **E. Perrotin**[1] **P. Régnier**[1]
[1 2] IRIT, CNRS, Univ. Paul Sabatier, France
[1]fistname.surname@irit.fr [2]faustine.maffre@gmail.com

Abstract

We study a simple epistemic logic with a restricted language where formulas are boolean combinations of (epistemic) atoms: sequences of 'knowing whether' operators followed by propositional variables. Our language is strictly more expressive than existing restricted languages, where atoms are sequences of epistemic operators and negations followed by propositional variables (in other words: atoms are epistemic formulas without conjunctions and disjunctions). Going further beyond existing approaches, we also introduce a 'common knowledge of a group whether' operator. We give an axiomatization for this logic and show that the model checking and satisfiability problems can be reduced to their classical counterparts.

1 Introduction

Suppose there are n agents, each of which knows some secret: a piece of information that is not known to the others. They communicate by phone calls, and whenever one person calls another they tell each other all they know at that time. How many calls are required before each item of gossip is known to everyone? This is known as the gossip problem. It is of great interest in the conception of communication networks [15] and in parallel and distributed computing, but there are other less obvious applications like the management of data on storage devices [20], or the computation of the syntenic distance between two genomes (minimum number of fusions, fissions, and translocations required to transform one into the other) [24]. Several variants have been studied in the literature, and a survey of these alternatives and the associated results has been published [15]. The gossip problem can be viewed as a multiagent epistemic planning task where the goal is shared knowledge: everybody knows all secrets. It is a purely epistemic planning task because it is only the agents' knowledge that evolves, while the facts of the world remain unchanged. There are numerous variations of such a planning problem; see e.g. [10] for an overview. The goal may in particular be to achieve higher-order shared knowledge: everybody knows that everybody knows all secrets, and so on. Due to its simplicity and its variations, the gossip problem can be viewed as a paradigmatic epistemic planning task, much in the same way as the blocksworld is a paradigmatic classical planning task.

The aim of the present paper is to follow up on [8], in which we introduce a simple logical approach within which we can account for epistemic planning tasks such as the above. We could have chosen Dynamic Epistemic Logic (DEL) for that purpose: its event models allow us to model telephone calls as private announcements. However, we take inspiration from a simpler framework: the Dynamic Epistemic Logic of Propositional Assignment and Observation (DEL-PAO), introduced in [16] and further developed in [6, 17, 25]. It is based on visibility atoms, that are recursively defined as either propositional variables or atomic pieces of information about whether or not an agent sees the truth value of a visibility atom. DEL-PAO was applied to epistemic planning in [9]. We here concentrate on the epistemic logic underlying DEL-PAO, disregarding the dynamic aspects. The logic is consequently called the Epistemic Logic of Observation, abbre-

viated to EL-O. We improve in [8] over [9] by integrating a form of common knowledge into planning, and polynomially embed the EL-O-based planning we define into classical planning. It follows that deciding the solvability of a simple epistemic planning task is PSPACE complete.

However, the form of common knowledge we have in [8] only accounts for knowledge shared by *all* of the agents. In particular, in the gossip problem, we cannot model the common knowledge between only two agents that we would expect to occur after a phone call between those agents. In many situations we can imagine knowledge becoming common within a subgroup of agents, such as agents in a call or agents present in a room (while some other agents may be outside of that room). In this paper we therefore study a generalization of EL-O which accounts for joint vision within subgroups of agents.

The paper is organized as follows. In Section 2 we provide a detailed motivation of our approach. In Section 3 we introduce our generalization of EL-O. In Section 4 we study its properties. We conclude in Section 5.

2 Background and motivation

Reasoning in epistemic logic as introduced by Hintikka [18] and popularized in AI by Halpern and colleagues [12] is strictly more complex than in classical logic: the satisfiability problem is at least in PSPACE, and is EXPTIME-complete if common knowledge is involved [14]. The complexity gap widens for planning tasks: while the solvability problem is PSPACE complete for classical planning [5], it is undecidable for planning in DEL [3]. Such DEL-based accounts of epistemic planning provide rather expressive models for the agents' perception of actions. These models parallel the standard epistemic logic modelling of uncertainty in terms of indistinguishability relations between possible states: uncertainty in the perception of actions is modelled by so-called event models in terms of indistinguishability relations between possible events. As explored in a series of papers, undecidability is already the case under very weak hypotheses about these event models [1, 4, 7]. Basically, DEL planning tasks are only decidable when the event model is a singleton, i.e., when all actions are public. However, actions in the gossip problem as well as in many real world applications are not public.

We propose in [8] to base epistemic planning on an epistemic logic that is simpler than Hintikka's. Basically, what we do is restrict epistemic information to atomic formulas. We are not the first to do this: several previous approaches have proposed languages where the scope of the epistemic operator K_i is restricted to literals, or literals that are preceded by a sequence of epistemic operators [11, 28, 21], possibly with negations [27]. Such restrictions however exclude formulas such as $K_i(K_j p \vee K_j \neg p)$ expressing that agent i knows that agent j knows whether the propositional variable p is true. This is a major drawback because such formulas are fundamental in communication and more generally in any forms of interaction: a situation where agent i does not know whether p is the case or not ($\neg K_i p \wedge \neg K_i \neg p$) but knows that j knows ($K_i(K_j p \vee K_j \neg p)$) may lead agent i to ask j about p. For example, in the modelling of the gossip problem 'knowing-whether' information is required to describe the initial situation

$$\bigwedge_i ((K_i s_i \vee K_i \neg s_i) \wedge \bigwedge_{j \neq i} (\neg K_i s_j \wedge \neg K_i \neg s_j))$$

where s_i is the secret of agent i, as well as the goal $\bigwedge_{i,j} (K_i s_j \vee K_i \neg s_j)$.

We here make a similar restriction, but move from the primitive 'knowing-that' operator K_i to the less standard 'knowing-whether' or 'knowing-if' operator Kif_i, also considered in [26]. In the unrestricted language these two operators are interdefinable: we have $Kif_i\varphi \leftrightarrow K_i\varphi \vee K_i\neg\varphi$ and $K_i\varphi \leftrightarrow \varphi \wedge Kif_i\varphi$, for arbitrary formulas φ [13]. However, the situation changes when we restrict the language to sequences of 'knowing-whether' operators followed by propositional variables: we can not only define $K_i p$ as $p \wedge Kif_i p$ and $K_i \neg p$ as $\neg p \wedge Kif_i p$, but we can also express formulas such as the above $K_i(K_j p \vee K_j \neg p)$, namely by $Kif_j p \wedge Kif_i Kif_j p$.

We present in [8] the logic EL-O, where Kif_i is written as S_i. We read $S_i\, p$ as "agent i sees whether p is true or not". When we define $K_i p$ as $p \wedge S_i\, p$ we therefore consider that i's knowledge comes from what i sees. Such a primitive was introduced in the model checking literature in order to represent epistemic models in a compact way [29, 19, 2]. All these approaches were based on the hypothesis that who sees what is common knowledge. This is a very strong hypothesis: it follows that whenever i knows that p is true then i knows exactly who else knows that p is true. In terms of the visibility operator, these papers consider $S_i\, S_j\, p$ to be valid for arbitrary i and j. As first proposed in [16], this restriction can be overcome by abandoning the validity of $S_i\, S_j\, p$ and giving the status of first-class citizens to such sequences.

Formally, in EL-O, it is considered that every visibility atom $S_{i_1} \ldots S_{i_n}\, p$ is an atomic formula. Com-

plex formulas are then boolean combinations of such atomic formulas. The reader should keep in mind that S_i is not a modal operator, in opposition to Kif_i (and our move in notation also aims at stressing that). This comes with a move away from Kripke models with accessibility relations: a model is simply a valuation over the set of visibility atoms, alias a state. We identify such valuations with sets of visibility atoms. As argued in [29, 2], such compact models are more attractive than Kripke models when it comes to model checking.

Simplifications of epistemic logic typically lack a modal operator of common knowledge. In [8] we introduce the operator $\mathcal{JS}$, reading $\mathcal{JS}\,p$ as "all agents jointly see the value of p", or "all agents jointly see whether p is true or not". Metaphorically, joint attention about a propositional variable p can be understood as eye contact between the agents when observing p. $\mathcal{JS}$ is a powerful operator that represents common knowledge obtained through copresence. To understand the power of $\mathcal{JS}$, one may think of $\mathcal{JS}\,p$ as implying the infinite set of propositions of the form $S_{i_1} \dots S_{i_n}\, p$ for all $i_1, \dots, i_n$ and all $n \geq 1$. Suppose p stands for 'the door is open'. We can imagine that $\mathcal{JS}\,p$ is true when all agents are present in the same room and not only observe the door, but also mutually observe each other. In this concrete example, if an agent leaves the room and closes the door behind her, $\mathcal{JS}\,p$ becomes false because the agents no longer mutually observe each other (even if the agent who has left can see the closed door from the outside). As joint visibility of p should imply individual visibility of p, we require that states containing $\mathcal{JS}\,p$ should also contain every $S_i\,p$, and more generally every p preceded by any sequence of S_i and $\mathcal{JS}$. Defining $CK\,p$ as $p \wedge \mathcal{JS}\,p$, we obtain a common knowledge operator that satisfies all the standard properties except the induction axiom $CK(p \rightarrow \bigwedge_{i \in Agt} K_i p) \rightarrow (p \rightarrow CK\,p)$; actually the antecedent cannot be directly expressed by means of $\mathcal{JS}$ operators.

This 'joint vision' operator still has expressive limits. In particular, it does not account for joint vision between groups of agents, when those groups are strictly smaller than the set of all agent. We therefore generalize the notion of joint vision to 'joint vision within a group', and introduce the corresponding operator $\mathcal{JS}_G$, where G is a group of agents. The operators $\mathcal{JS}$ and S_i can be expressed as $\mathcal{JS}_{Agt}$ and $\mathcal{JS}_{\{i\}}$ respectively. We can now express by $\mathcal{JS}_{\{1,2\}}\,p$ the fact that agents 1 and 2 jointly see p, for example after 1 has called 2 to share information about p, excluding 3 from the conversation.

It is generally considered in the AI literature that a reasonable epistemic logic should satisfy introspection: the formulas $K_i\varphi \rightarrow K_iK_i\varphi$ and $\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$ should both be valid.[1] In terms of visibility this means that states should contain every $\mathcal{JS}_{G'}\,\mathcal{JS}_G\,p$ for $G' \subseteq G$, and more generally $\mathcal{JS}_{G'}\,\mathcal{JS}_G\,p$ preceded by any sequence $\mathcal{JS}_H \dots \mathcal{JS}_{H'}$.

In the rest of the paper we will work out the details of what we have sketched up to now.

3 EL-O: Epistemic Logic of Observation

We introduce our generalization of the Epistemic Logic of Observation, abbreviated to EL-O.

3.1 Atoms, introspective atoms, and atomic consequence

Let *Prop* be a countable set of propositional variables and let *Agt* be a finite set of agents. We call finite set of agents *groups*, and denote them by G, H, etc.

We use σ, σ', etc. to denote elements of $(2^{Agt})^*$, i.e., words on the set of groups. The set of non-empty words built from elements of G is $(2^G)^+$.

An *atom* is a word of groups followed by a propositional variable. We use α, α', etc. to denote elements of the set of atoms $ATM = (2^{Agt})^* \times Prop$. We treat atoms as words, too, allowing the notation $\sigma\alpha$ for atoms, and identify in atoms any group G with the operator $\mathcal{JS}_G$; for example, we write $\mathcal{JS}_{G_1}\,\mathcal{JS}_{G_2}\,p$ instead of the couple $\langle G_1G_2, p\rangle$. We read $\mathcal{JS}_G\,\alpha$ as "group G jointly sees α". Atoms with an empty sequence of groups are nothing but propositional variables.

Here are some examples: $\mathcal{JS}_{\{1\}}\,p$ reads "1 sees the value of p". It means that 1 knows whether p is true or false. $\mathcal{JS}_{\{1,3\}}\,\mathcal{JS}_{\{2\}}\,q$ reads "1 and 3 jointly see whether agent 2 sees the value of q". In other words, there is *joint attention* between 1 and 3 concerning 2's observation of q: agent 2 may or may not see the value of q, and in both cases this is jointly observed.

We are going to design the semantics in such a way that principles of introspection are valid. An atom is *introspective* if it is of the form $\sigma\mathcal{JS}_H\,\mathcal{JS}_G\,\alpha$ with $H \subseteq G$. We call *I-ATM* the set of all introspective atoms.

1. We are aware that negative introspection was criticised in the literature as being too strong [22, 23]. One may however argue that these criticisms do not apply to visibility-based knowledge.

Atomic consequence is a relation on ATM that is defined inductively by:

$$p \Rightarrow \beta \text{ iff } p = \beta$$
$$J\!S_G\, \alpha \Rightarrow \beta \text{ iff } \beta = \sigma\alpha \text{ and } \sigma \in (2^G)^+$$

For example, $J\!S_{\{1,2,3\}}\, p \Rightarrow J\!S_{\{3\}}\, J\!S_{\{1,2\}}\, p$. The relation $\Rightarrow$ is clearly reflexive and transitive. When $\alpha \Rightarrow \beta$, we say that α is a *cause* of β and that β is a *consequence* of α. We will, of course, ensure that atomic consequences are valid implications.

The *set of atomic consequences* of an atom α is $\alpha^{\Rightarrow} = \{\beta : \alpha \Rightarrow \beta\}$. The *set of atomic causes* of an atom α is $\alpha^{\Leftarrow} = \{\beta : \beta \Rightarrow \alpha\}$. The set of consequences of an atom other than a propositional variables is infinite. The set of causes of an atom α is finite and if we define the length of a visibility operator $J\!S_G$ to be one, then each cause is no longer than α.

We generalise the definition of atomic consequence to sets of atoms $s \subseteq ATM$ in the obvious way: $s^{\Rightarrow} = \bigcup_{\alpha \in s} \alpha^{\Rightarrow}$.

3.2 Language

The language of EL-O is defined by the following grammar:

$$\varphi ::= \alpha \mid \neg\varphi \mid (\varphi \wedge \varphi)$$

where α ranges over ATM. The set of EL-O *formulas* is noted $Fml_{\text{EL-O}}$.

The boolean operators $\top$, $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are defined in the standard way. The set $ATM(\varphi)$ of atoms of formula φ is defined by:

$$ATM(\alpha) = \{\alpha\}$$
$$ATM(\neg\varphi) = ATM(\varphi)$$
$$ATM(\varphi \wedge \varphi') = ATM(\varphi) \cup ATM(\varphi')$$

For example, $ATM(J\!S_{\{1,3\}}\, q \wedge J\!S_{\{2\}}\, p) = \{J\!S_{\{1,3\}}\, q, J\!S_{\{2\}}\, p\}$ and $ATM(J\!S_{\{1\}}\, J\!S_{\{2,3\}}\, p) = \{J\!S_{\{1\}}\, J\!S_{\{2,3\}}\, p\}$. Note that neither p nor $J\!S_{\{2,3\}}\, p$ are atoms of $J\!S_{\{1\}}\, J\!S_{\{2,3\}}\, p$.

3.3 Interpretation of formulas

A state is a subset of the set of atoms ATM. We denote states by s, s', etc. The set of all states is 2^{ATM}.

An obvious way of guaranteeing introspection is to interpret formulas exclusively in *introspectively closed states*: states that contain all introspective atoms and are closed under $\Rightarrow$, i.e., sets of atoms that equal $s^{\Rightarrow} \cup I\text{-}ATM$ for some state $s \subseteq ATM$. This is what was done in [16]. The drawback of such a semantics is that introspective states are infinite, while model checking requires a finite state. This problem can be overcome by defining states that are 'sufficiently introspective', as done in [25]. We here take an easier road by directly working with finite models, interpreting formulas in such a way that introspection is simulated.

$s \models \alpha$	iff $\alpha \in s^{\Rightarrow} \cup I\text{-}ATM$
$s \models \neg\varphi$	iff not $(s \models \varphi)$
$s \models \varphi \wedge \varphi'$	iff $s \models \varphi$ and $s \models \varphi'$

Table 1 – Interpretation of formulas

The truth conditions for EL-O formulas are given in Table 1. The non-standard one is the one for atoms α: α is true in state s if and only if α is introspective or $\beta \Rightarrow \alpha$ for some $\beta \in s$.

Example 1. In the initial state of the gossip problem every agent only knows her own secret. For the sake of simplicity we moreover suppose that all secrets are initially true. Therefore $s_0 = \{J\!S_{\{i\}}\, s_i : i \in Agt\} \cup \{s_i : i \in Agt\}$. Then $s_0 \models J\!S_{\{i\}}\, s_i$ and $s_0 \models \bigwedge_{j \neq i} \neg J\!S_{\{i\}}\, s_j$ for every $i \in Agt$.

When $s \models \varphi$ for every $s \subseteq ATM$ then we say that φ is *valid*; when $s \models \varphi$ for some $s \subseteq ATM$ then we say that φ is *satisfiable*.

Proposition 1. *For every $\alpha \subseteq ATM$, α is valid iff α is introspective.*

Proof. For the left-to-right direction, suppose α is valid. Then in particular $\emptyset \models \alpha$. As $\emptyset^{\Rightarrow}$ is still empty, α must be introspective. □

If $\alpha \Rightarrow \beta$ then $\alpha \rightarrow \beta$ is EL-O valid. The converse is not always the case. However:

Proposition 2. *$\alpha \rightarrow \beta$ is EL-O valid iff β is introspective or $\alpha \Rightarrow \beta$.*

Proof. The right-to-left direction is obvious. For the left-to-right direction, suppose $\alpha \rightarrow \beta$ is EL-O valid. Then in particular $\{\alpha\} \models \beta$. Hence β is either an introspective atom or an atomic consequence of α. □

4 Properties

We now reduce EL-O to Classical Propositional Calculus, then give an axiomatization for it.

4.1 Relation with Classical Propositional Calculus

In our framework there are two options when it comes to defining Classical Propositional Calculus (CPC). The first is to restrict the language to formulas without operators of visibility; classical logic is then the set of validities of that language. The second option is more interesting here: we consider the full language built from ATM, but we modify the interpretation of visibility atoms to

$$s \models^{\mathsf{CPC}} \alpha \text{ iff } \alpha \in s$$

and keep the same interpretation as in EL-O otherwise. Hence we have visibility atoms but do not care about introspection; for example, the formulas $\neg JS_{\{i\}} JS_{\{i\}} p$ and $JS_{\{i,j\}} p \wedge \neg JS_{\{i\}} p$ are classically satisfiable: for the former, $\emptyset \models^{\mathsf{CPC}} \neg JS_{\{i\}} JS_{\{i\}} p$; for the latter, $\{JS_{\{i,j\}} p\} \models^{\mathsf{CPC}} \neg JS_{\{i\}} p$.

In the rest of the section we provide two embeddings from EL-O into CPC: one where we expand states, and one where we expand formulas.

Proposition 3. *For every state $s \subseteq ATM$ and formula $\varphi \in Fml_{\mathsf{EL\text{-}O}}$, $s \models \varphi$ if and only if $s^{\Rightarrow} \cup I\text{-}ATM \models^{\mathsf{CPC}} \varphi$.*

Proof. The proof is by induction on the form of φ. The base case is immediate from the interpretation of atoms in EL-O. □

The above proposition confirms that we simulate introspectively closed states with our interpretation of formulas. It also entails the following properties:

Lemma 1. *For all states $s, s' \subseteq ATM$ and for every formula $\varphi \in Fml_{\mathsf{EL\text{-}O}}$, if s and s' classically agree on atoms of φ then they classically agree on φ. That is, if $s \cap ATM(\varphi) = s' \cap ATM(\varphi)$ then $s \models^{\mathsf{CPC}} \varphi$ iff $s' \models^{\mathsf{CPC}} \varphi$.*

Proposition 4. *For every state $s \subseteq ATM$ and formula $\varphi \in Fml_{\mathsf{EL\text{-}O}}$, $s \models \varphi$ if and only if $(s^{\Rightarrow} \cup I\text{-}ATM) \cap ATM(\varphi) \models^{\mathsf{CPC}} \varphi$.*

Corollary 1. *For every state $s \subseteq ATM$ and formula $\varphi \in Fml_{\mathsf{EL\text{-}O}}$, $s \models \varphi$ if and only if $(s^{\Rightarrow} \cup I\text{-}ATM) \cap ATM(\varphi) \models \varphi$.*

Proof. This stems from the fact that if $s_\varphi = (s^{\Rightarrow} \cup I\text{-}ATM) \cap ATM(\varphi)$, then $(s_\varphi)_\varphi = s_\varphi$. Therefore $s_\varphi \models \varphi$ iff $s_\varphi \models^{\mathsf{CPC}} \varphi$. □

As $ATM(\varphi)$ is finite, we hence have the finite model property both in EL-O and in the classical translation. In particular, the model checking problem and the satisfiability problem have the same complexities as in CPC.

The second embedding is more interesting from a planning point of view, as it does not modify the states. We define the *expansion* of an atom as follows:

$$\mathsf{Exp}(\alpha) = \begin{cases} \top & \text{if } \alpha \in I\text{-}ATM \\ (\bigvee \alpha^{\Leftarrow}) & \text{otherwise} \end{cases}$$

We extend this definition homomorphically to formulas. For example, if $Agt = \{i, j\}$, the expansion of the EL-O unsatisfiable $JS_{\{i,j\}} p \wedge \neg JS_{\{i\}} p$ is $JS_{\{i,j\}} p \wedge \neg(JS_{\{i\}} p \vee JS_{\{i,j\}} p)$, which is classically unsatisfiable; and the expansion of the EL-O satisfiable $JS_{\{i,j\}} JS_{\{i\}} p \wedge \neg JS_{\{i\}} p$ is the classically satisfiable $(JS_{\{i,j\}} JS_{\{i\}} p \vee JS_{\{i,j\}} p) \wedge \neg(JS_{\{i\}} p \vee JS_{\{i,j\}} p)$. This can be generalised:

Proposition 5. *For every state $s \subseteq ATM$ and formula $\varphi \in Fml_{\mathsf{EL\text{-}O}}$, $s \models \varphi$ iff $s \models^{\mathsf{CPC}} \mathsf{Exp}(\varphi)$.*

Proof. The proof is by induction on the form of φ. For the base case we distinguish two subcases. First, if $\alpha \in I\text{-}ATM$ then $\mathsf{Exp}(\alpha) = \top$ and therefore $s \models \alpha$ and $s \models^{\mathsf{CPC}} \mathsf{Exp}(\alpha)$. Second, if $\alpha \notin I\text{-}ATM$ then $s \models \alpha$ iff $\beta \Rightarrow \alpha$ for some $\beta \in s$. The latter is equivalent to $s \models^{\mathsf{CPC}} \bigvee \alpha^{\Leftarrow}$. □

In the version of EL-O with only the JS and S_i operators, the expansion was polynomial and allowed us to go from epistemic to classical planning with no increase in complexity. However, in this more general case, this expansion causes an explosion of the length of the formula, as the number of causes of an atom $JS_G \alpha$ is exponential in the size of G. We will come back to this issue in the conclusion.

4.2 Axiomatization

The valid EL-O formulas are axiomatized by the schemas of Table 2.

Proposition 6. *For every formula $\varphi \in Fml_{\mathsf{EL\text{-}O}}$, φ is* EL-O *valid iff φ is provable in* CPC *from the four axiom schemas Vis_1– Vis_4 of Table 2.*

Proof. We take advantage of Proposition 3 and show that the schemas Vis_1–Vis_4 characterise the set of introspectively closed states $\{s^{\Rightarrow} \cup I\text{-}ATM : s \subseteq ATM\}$.

The right-to-left direction is clear: each of the four axiom schemas is valid in introspectively closed states.

For the left-to-right direction, we show that every s satisfying Vis_1–Vis_4 is introspectively closed. We begin with closure under atomic consequence. Let $s \models \alpha$. The interesting case is when $\alpha = JS_G \alpha'$. Then $s \models JS_G JS_{H_1} \alpha'$ for any $H_1 \subseteq G$ by axiom (Vis_4), and

$JS_H\, JS_G\, \alpha$	for $H \subseteq G$	(Vis_1)
$JS_{Agt}\, JS_H\, JS_G\, \alpha$	for $H \subseteq G$	(Vis_2)
$JS_G\, \alpha \rightarrow JS_H\, \alpha$	for $H \subseteq G$	(Vis_3)
$JS_G\, \alpha \rightarrow JS_G\, JS_H\, \alpha$	for $H \subseteq G$	(Vis_4)

Table 2 – Axioms for introspection

also $s \models JS_G\, JS_{H_2}\, JS_{H_1}\, \alpha'$ for any $H_1, H_2 \subseteq G$, and so on: we can generate any $s \models JS_G\, \sigma\alpha'$ for any $\sigma \in (2^G)^*$ and then, by (Vis_3), we can obtain $s \models JS_H\, \sigma\alpha'$ for any $H \subseteq G$ and any $\sigma \in (2^G)^*$. We can therefore obtain that s satisfies any sequence whose strict postfix is α', that is, every $\sigma\, \alpha'$ for any $\sigma \in (2^G)^+$.

We use the same technique to show that a state s satisfying Vis_1–Vis_4 satisfies every $\sigma\, JS_H\, JS_G\, \alpha$ for $\sigma \in (2^{2^{Agt}})^*$ and $H \subseteq G$, using (Vis_1) if σ is empty, and (Vis_2), (Vis_4) and (Vis_3) otherwise. □

5 Conclusion

In this paper, we have generalized the logic EL-O, subsuming its observability operators by an operator of joint vision within an arbitrary group. We have defined and axiomatized the corresponding logic, and shown that it can be reduced to classical propositional logic.

Based on the visibility information that is contained in the states, an accessibility relation can be defined for every group of agents to in order to interpret a common knowledge operator CK_G, which becomes the standard knowledge operator K_i when $G = \{i\}$ and the common knowledge operator CK when $G = Agt$. The properties of the latter and the relation with standard epistemic logic are discussed in [8] and can be extended to this generalized definition. In particular, $CK_G\alpha$ is equivalent to $\alpha \wedge JS_G\, \alpha$.

We give in [8] a definition of planning in EL-O as well as a reduction to classical planning. The decidability of EL-O-based epistemic planning contrasts with the undecidability of DEL-based epistemic planning, which is the case even for simple fragments.

It is proved in [8] that the complexity of solvability in EL-O-based planning is PSPACE complete. The proof relies on the fact that the length and number of causes of a given atom are not greater than the length of that atom, resulting in polynomial translations to CPC and classical planning. In the more general setting where joint vision within arbitrary groups is allowed, the number of causes of an atom becomes exponential, and the property no longer holds. We can however retain this low complexity while keeping our framework relevant for a number of applications by allowing joint vision only for relevant groups. For example, in the gossip problem, we only need joint vision between pairs of agents. Similarly, it is natural to imagine that there may be a limit on the number of agents that can be present in a room at the same time, or that can join in on a conversation.

We have as of now only presented the generalization of EL-O. The next step is of course to extend this generalization to epistemic planning, and further refine the limits in complexity mentioned above. More generally, it will be interesting to study how our epistemic actions compare to DEL event models.

References

[1] Aucher, Guillaume et Thomas Bolander: *Undecidability in epistemic planning*. Dans Rossi, Francesca (rédacteur) : *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, pages 27–33. AAAI Press, 2013.

[2] Benthem, Johan van, Jan van Eijck, Malvin Gattinger et Kaile Su: *Symbolic model checking for Dynamic Epistemic Logic - S5 and beyond*. J. Log. Comput., 28(2) :367–402, 2018. https://doi.org/10.1093/logcom/exx038.

[3] Bolander, Thomas et Mikkel Birkegaard Andersen: *Epistemic planning for single and multi-agent systems*. Journal of Applied Non-Classical Logics, 21(1) :9–34, 2011. http://www.tandfonline.com/doi/abs/10.3166/jancl.21.9-34.

[4] Bolander, Thomas, Martin Holm Jensen et François Schwarzentruber: *Complexity Results in Epistemic Planning*. Dans Yang, Qiang et Michael Wooldridge (rédacteurs) : *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2791–2797. AAAI Press, 2015, ISBN 978-1-57735-738-4. http://ijcai.org/Abstract/15/395.

[5] Bylander, Tom: *The computational complexity of propositional STRIPS planning*.

Artificial Intelligence, 69 :165–204, 1994, ISSN 00043702.

[6] Charrier, Tristan, Emiliano Lorini, Andreas Herzig, Faustine Maffre et François Schwarzentruber: *Building epistemic logic from observations and public announcements*. Dans *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, 2016.

[7] Cong, Sébastien Lê, Sophie Pinchinat et François Schwarzentruber: *Small Undecidable Problems in Epistemic Planning*. Dans Lang, Jérôme (rédacteur) : *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 4780–4786. ijcai.org, 2018, ISBN 978-0-9992411-2-7. https://doi.org/10.24963/ijcai.2018/664.

[8] Cooper, Martin, Andreas Herzig, Faustine Maffre, Frédéric Maris, Elise Perrotin et Pierre Régnier: *A Lightweight Epistemic Logic and its Application to Planning*. Soumis à Artificial Intelligence, 2019. https://www.irit.fr/~Andreas.Herzig/P/aij19_submitted.pdf.

[9] Cooper, Martin C., Andreas Herzig, Faustine Maffre, Frédéric Maris et Pierre Régnier: *A simple account of multi-agent epistemic planning*. Dans *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*, pages 193–201, 2016.

[10] Cooper, Martin C., Andreas Herzig, Faustine Maffre, Frédéric Maris et Pierre Régnier: *The epistemic gossip problem*. Discrete Mathematics, 342(3) :654–663, 2019. https://doi.org/10.1016/j.disc.2018.10.041.

[11] Demolombe, Robert et Maria del Pilar Pozos Parra: *A Simple and Tractable Extension of Situation Calculus to Epistemic Logic*. Dans Ras, Zbigniew W. et Setsuo Ohsuga (rédacteurs) : *Foundations of Intelligent Systems, 12th International Symposium, ISMIS 2000, Charlotte, NC, USA, October 11-14, 2000, Proceedings*, tome 1932 de *Lecture Notes in Computer Science*, pages 515–524. Springer, 2000, ISBN 3-540-41094-5. https://doi.org/10.1007/3-540-39963-1_54.

[12] Fagin, Ronald, Joseph Y. Halpern, Yoram Moses et Moshe Y. Vardi: *Reasoning about Knowledge*. MIT Press, 1995.

[13] Fan, Jie, Yanjing Wang et Hans van Ditmarsch: *Contingency and Knowing Whether*. Rew. Symb. Logic, 8(1) :75–107, 2015. https://doi.org/10.1017/S1755020314000343.

[14] Halpern, Joseph Y. et Yoram Moses: *A guide to completeness and complexity for modal logics of knowledge and belief*. Artificial Intelligence, 54(3) :319–379, 1992.

[15] Hedetniemi, Sandra M., Stephen T. Hedetniemi et Arthur L. Liestman: *A survey of gossiping and broadcasting in communication networks*. Networks, 18(4) :319–349, 1988, ISSN 00283045.

[16] Herzig, Andreas, Emiliano Lorini et Faustine Maffre: *A poor man's epistemic logic based on propositional assignment and higher-order observation*. Dans Hoek, Wiebe van der, Wesley H. Holliday et Wen fang Wang (rédacteurs) : *Proceedings of the 5th International Conference on Logic, Rationality and Interaction (LORI 2015)*, pages 156–168. Springer Verlag, 2015. http://www.irit.fr/~Andreas.Herzig/P/Lori15.html.

[17] Herzig, Andreas, Emiliano Lorini, Faustine Maffre et François Schwarzentruber: *Epistemic boolean games based on a logic of visibility and control*. Dans Kambhampati, Subbarao (rédacteur) : *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*. AAAI Press, 2016.

[18] Hintikka, Jaakko: *Knowledge and Belief : An Introduction to the Logic of the Two Notions*. Cornell University Press, 1962.

[19] Hoek, Wiebe van der, Petar Iliev et Michael Wooldridge: *A logic of revelation and concealment*. Dans Hoek, Wiebe van der, Lin Padgham, Vincent Conitzer et Michael Winikoff (rédacteurs) : *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 1115–1122. IFAAMAS, 2012. http://dl.acm.org/citation.cfm?id=2343856.

[20] Khuller, Samir, Yoo Ah Kim et Yung-Chun (Justin) Wan: *On Generalized Gossiping and Broadcasting (Extended Abstract)*. Dans Di Battista, Giuseppe et Uri Zwick (rédacteurs) : *Algorithms - ESA 2003, 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003, Proceedings*, tome 2832 de *Lecture Notes in Computer*

Science, pages 373–384. Springer, 2003, ISBN 3-540-20064-9.

[21] Lakemeyer, Gerhard et Yves Lespérance: *Efficient Reasoning in Multiagent Epistemic Logics*. Dans Raedt, Luc De, Christian Bessière, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz et Peter J. F. Lucas (rédacteurs) : *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, tome 242 de *Frontiers in Artificial Intelligence and Applications*, pages 498–503. IOS Press, 2012, ISBN 978-1-61499-097-0. https://doi.org/10.3233/978-1-61499-098-7-498.

[22] Lenzen, Wolfgang: *Recent work in epistemic logic*. North Holland Publishing Company, Amsterdam, 1978.

[23] Lenzen, Wolfgang: *On the semantics and pragmatics of epistemic attitudes*. Dans Laux, Armin et Heinrich Wansing (rédacteurs) : *Knowledge and belief in philosophy and AI*, pages 181–197. Akademie Verlag, Berlin, 1995.

[24] Liben-Nowell, David: *Gossip is synteny : Incomplete gossip and the syntenic distance between genomes*. J. Algorithms, 43(2) :264–283, 2002.

[25] Maffre, Faustine: *Ignorance is bliss : observability-based dynamic epistemic logics and their applications. (Le bonheur est dans l'ignorance : logiques épistémiques dynamiques basées sur l'observabilité et leurs applications)*. Thèse de doctorat, Paul Sabatier University, Toulouse, France, 2016. https://tel.archives-ouvertes.fr/tel-01488408.

[26] Miller, Tim, Paolo Felli, Christian J. Muise, Adrian R. Pearce et Liz Sonenberg: *'Knowing Whether' in Proper Epistemic Knowledge Bases*. Dans Schuurmans, Dale et Michael P. Wellman (rédacteurs) : *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1044–1050. AAAI Press, 2016, ISBN 978-1-57735-760-5. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12291.

[27] Muise, Christian, Vaishak Belle, Paolo Felli, Sheila A. McIlraith, Tim Miller, Adrian R. Pearce et Liz Sonenberg: *Planning over multi-agent epistemic states : A classical planning approach*. Dans *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 3327–3334. AAAI Press, 2015.

[28] Steedman, Mark et Ronald PA Petrick: *Planning dialog actions*. Dans *Proceedings of the 8th SIGDIAL Workshop on Discourse and Dialogue (SIGdial 2007)*, pages 265–272, 2007.

[29] Su, Kaile, Abdul Sattar et Xiangyu Luo: *Model checking temporal logics of knowledge via OBDDs*. The Computer Journal, 50(4) :403–420, 2007.

Actes JIAF 2019

An early guidance system for a general knowledge-based aiding framework using probabilistic interventions

Véronique Delcroix[1] **Pierre-Henri Wuillemin**[2]

[1] Univ. Polytechnique Hauts-de-France, CNRS, UMR 8201 - LAMIH, F-59313 Valenciennes, France
[2] Sorbonne université, CNRS, UMR 7606 - LIP6, F- 75252 Paris cedex 05, France
Veronique.Delcroix@uphf.fr Pierre-Henri.Wuillemin@lip6.fr

Résumé

Un problème de décision courant se répète de nombreuses fois, avec le même type d'alternatives et le même ensemble de critères, mais avec une situation de décision différente à chaque occurence du problème. Dans ce type de problème, le conseil en amont vise à faciliter la sélection d'un sous ensemble d'alternatives satisfaisantes pour le cas de décision considéré, sans demander à l'utilisateur d'avoir des connaissances sur le problème. Cet article propose un système de conseil en amont basé sur un modèle de connaissances du problème de décision courant. Pour commencer, l'article présente la construction d'un réseau bayésien pour embarquer la connaissance dans le système. Ensuite, le concept d'intervention dans un réseau bayésien proposé par Pearl est étendu aux interventions probabilistes pour des variables simples et des ensembles de variables. Enfin, la procédure de conseil en amont pour un problème de décision courant est présentée, sur la base du modèle de connaissance et en utilisant les interventions probabilistes pour fixer l'écosystème de la personne, même lorsque le cas de décision n'est que partiellement observé.

Abstract

A common decision problem repeats a lot of time with the same kind of alternatives and the same set of criteria, but with a different decision case in each occurrence. The objective of early guidance in this kind of problem is to facilitate the selection of a subset of satisfactory alternatives for each new decision case, without asking the user any knowledge of the problem. This article proposes an early guidance system based on a model of knowledge of the common decision problem. It first presents the construction of a Bayesian network for a common decision problem to embed the knowledge in the aiding framework. Second, the concept of intervention proposed by Pearl is extended to probabilistic interventions for a single variable and for a set of variables. Finally the early guidance procedure is presented on the basis of the Bayesian network and using a probabilistic intervention to set a decision case even though it is partially observed.

1 Introduction

A common decision problem occurs repeatedly in different decision cases, corresponding to different actors, needs, constraints and priorities, but including each time the same type of alternatives and the same set of decision criteria. An example is the problem of study and career guidance for students. This question, what to do after the baccalaureate, concerns a high number of young people, but each of them has its own characteristics (location, level, project, ambition, constraints, etc.). Another example of a common decision problem is the choice of a manual wheelchair that has been analyzed in [21].

The problem of early guidance in a common decision problem consists in providing advice at a very early step of a decision process, when the actors of the decision have not yet seriously thought about the problem. Early guidance precedes the whole decision process. At this step, the persons concerned by the consequences of a potential or future decision are considered to be merely passive.

Since the person concerned by the consequences of the choice is not obligatory the user of the general aiding framework, the early guidance system can be used by different persons involved in a decision case of a common decision problem. It can also be used as an automatic system, without any human user.

In a common decision problem, the whole decision process is most often achieved without the aid of any expert, and the actors involved generally gather only a small part of the relevant knowledge to guide the decision. For example, they do not necessary know what kind of alternative would best match their own situation. Because of that, it is important that an early guidance system does not appeal to the

actors knowledge or preference. The only certain available information is a partial description of the personal ecosystem of the person concerned by the consequences of the decision, including some of the constraints and preferences that could influence the decision.

As a consequence, the early guidance system has to embed general knowledge about the common decision problem, in order to provide some relevant guidance. This knowledge is about : (1) decision criteria to be considered, (2) alternative description in terms of characteristics and their evaluation (3) the personal ecosystem, *i.e.* any elements about the person and the context that could influence the choice.

In this paper, we address the new challenge that concern the design of an early guidance system, as part of a general aiding framework. Once defined, it should be possible to use the system easily for any new decision case, with the constraints presented above about the *ignorant* user. The objective of our early guidance system is to facilitate the selection of a small subset of satisfactory alternatives on the sole base of the available information about the decision case. Depending on the objective, this subset can be the final result, such as in the display of targeted car advertisement for a given internet user, or it could be used to choose an alternative among this subset, which can be done directly or by using a classic decision aiding system.

In the following of this paper, we define what is a common decision problem and set out the early guidance problem. Second, we present the knowledge based aiding framework and we explain the construction of the Bayesian network model of knowledge for a common decision problem. Then we describe the early guidance system based on the Bayesian network models. In that section, we explain the satisfactory situation principle, the definition and the implementation of probabilistic intervention, and the early guidance procedure. In the last section, we briefly present and discuss some related works.

2 Common Decision Problem and Early Guidance

In this section, we precisely define a common decision problem and the early guidance problem. We also provide a brief comparison with classical multi-criteria decision systems.

2.1 A Common Decision Problem

A *common decision problem* is a decision problem (such as choice, sorting or ranking) that repeats a lot of times, each time in a specific *ecosystem*, whose observed part is called a *decision case*. It involves alternatives whose evaluation regarding the decision criteria may also depend on the decision case.

The main components of a common decision problem are :

the characteristics of the alternatives ; The set of alternatives is potentially defined through the domain of the set of attributes. An alternative is an assignment of the set of characteristics. Some aberrant combinations of values are excluded due to some inherent constraints between some subsets of characteristics of an alternative.

the ecosystem of the person concerned by the decision ; it regroups any parameters related to the person and his/her context that may influence the decision. A decision case is a partial instantiation of the ecosystem, corresponding to the available observations of that specific case.

the set of decision criteria, each of them being characterized by :

- **the level of importance** of the criterion in the decision case.
- **the criterion evaluation index** that represents the evaluation (or the quality) of an alternative according to the criterion and the decision case.
- **the level of satisfaction** brought by an alternative on the criterion in that decision case. It depends on the importance and the evaluation such that (1) when an alternative presents a very good quality for a criterion, the satisfaction is very good whatever the importance of the criterion ; (2) the higher is the quality of the alternative, the higher is the satisfaction, but the lower is the importance, the easier it is to get a good level of satisfaction ; (3) when the criterion is indifferent according to a decision case, the satisfaction is good whatever the quality of the alternative according to that criterion.

A *common decision problem* is a triple $(\mathbf{X_{Alt}}, \mathbf{X_{eco}}, \mathcal{S})$ where $\mathbf{X_{Alt}}$ and $\mathbf{X_{eco}}$ are finitely-valued variables and $\mathcal{S}$ is a set $\{(Imp_g, Ind_g, S_g), g \in G\}$ with $G = \{1, 2, ...|G|\}$ and Imp_g, Ind_g, S_g being probabilistic functions such that :

Imp_g is defined on $\mathbf{X_g^{imp}} \subset \mathbf{X_{eco}}$ and $Imp_g(\mathbf{x}) = P(\cdot \mid \mathbf{x})$ with $\mathbf{x} \in Dom(\mathbf{X_g^{imp}})$.

Ind_g is defined on $\mathbf{X_g^{ind}} \subset \mathbf{X_{eco}} \cup \mathbf{X_{Alt}}$ and $Ind_g(\mathbf{x}) = P(\cdot \mid \mathbf{x})$ where $\mathbf{x} \in Dom(\mathbf{X_g^{ind}})$.

S_g only depends on Imp_g and Ind_g.

and

$\mathbf{X_{Alt}}$ is the set of characteristics of the alternatives,

$\mathbf{X_{eco}}$ is the set of characteristics of the ecosystem,

$\mathbf{a} \in \mathcal{A}$ is an alternative,

$\mathcal{A} \subset \bigtimes_{X \in \mathbf{X_{Alt}}} Dom(X)$ is the set of alternatives,

Let G a set of criteria and $(\mathbf{X_{Alt}}, \mathbf{X_{eco}}, \mathcal{S})$ a common decision problem for G. A *decision case* for $(\mathbf{X_{Alt}}, \mathbf{X_{eco}}, \mathcal{S})$

is a couple $(\mathbf{X_{DC}}, \mathbf{x_d c})$, where $\mathbf{X_{DC}} \subset \mathbf{X_{eco}}$ and $\mathbf{x_{dc}} \in \bigtimes_{Y \in \mathbf{X_{DC}}} Dom(Y)$.

In order to simplify the notation, we use the same notation to denote the probabilistic functions Imp_g, Ind_g, S_g and the resulting random variables. The variables Imp_g, Ind_g and S_g represent respectively the level of importance, the criterion evaluation index and the level of satisfaction.

2.2 The problem of early guidance

Given the set $\mathcal{A}$ of potential alternatives, and a decision case, the problem of early guidance is to make easier the selection of a subset of satisfactory alternatives in the considered situation. The satisfaction refers to the level of satisfaction S_g provided by an alternative regarding each decision criterion

A naive approach would be to evaluate the satisfaction of each alternative according to each criterion, in the considered situation, and try to exploit this huge matrix to classify the alternatives. this is not relevant since the size of the set of alternatives $\mathcal{A}$ is exponential in the number of attributes of the alternatives $\mathbf{X_{Alt}}$. This approach is possible when an initial small set of relevant alternatives has first been selected. It corresponds to the approach of Multi-criteria Decision Analysis that we discuss further. In our approach we focus on *satisfactory* alternatives, on the basis on the local level of satisfaction for each criterion, without trying to maximize any form of global satisfaction.

Since a common decision problem concerns a wide range of actors, we consider the possibility of "ignorant" actors, or with low involvement ; we also consider decision cases where the selection a subset of satisfactory alternatives for a given decision case is achieved without any participation of the person concerned by the decision, as in targeted advertisement. Those reasons underline the interest of an early guidance system.

The repetitiveness of a common decision problem makes interesting to capitalized general knowledge about the decision problem and identify what could be reused in each new decision case. Thus, we propose a knowledge based system. The modeling of the common decision problem is made once. It embeds general probabilistic knowledge about the parameters that should be taken into account and the way they influence the decision. Each use of the early guidance system concerns a specific decision case of the common decision problem, meaning specific persons, situations, objectives, etc. It takes place before the decision process as usually understood in multi-criteria decision analysis (MCDA).

The next section provides a quick comparison between the early guidance problem and MCDA.

2.3 Position of early guidance with respect to MCDA

In multi-criteria decision analysis (MCDA), each decision problem is usually unique (not repeated). The methods developed for MCDA [9] are based on the hypothesis that the decision maker is strongly involved in the decision process, and that he/she is guided by an analyst. At the opposite, we consider that a case of decision of a common decision problem generally occurs without the presence of an analyst or a specialist, meaning that the actors may not be aware of all the aspects of the decision problem. Consequently, an early guidance system can not be based on preferences, whereas most MDCA methods are based on pairwise comparisons of alternatives regarding each criterion.

Another point concerns the definition of a criterion. In [20], a criterion is a tool allowing to compare alternatives according to a particular significance axis or a point of view. More precisely, in MCDA, a criterion is a real-valued function on the set of alternatives, such that it appears meaningful to compare two alternatives a and b according to a particular point of view on the sole basis of the two numbers $g(a)$ and $g(b)$ [2]. In a common decision problem, the evaluation of an alternative according to a criterion may also depend on the decision case, since what is suitable for a decision case may not be suitable for another one. As a consequence, the evaluation of an alternative may be different depending on the decision case.

Another difference is about the set of alternatives : in our approach, the alternatives are potentially defined through the domain of the set of attributes, whereas in MCDA, a problem is usually defined by the so called decision matrix that gathers the performances of n alternatives a with respect to m criteria, in addition with the weights of the criteria. The decision matrix is the start point in MCDA, whereas the subset of satisfactory alternatives is the objective of our proposed early guidance system.

However, the early guidance problem partially meets the objective of the screening problem in MCDA. A screening procedure aims to reduce a large set of alternatives to a smaller set that most likely contains the best choice. As mentioned in [8], screening “should eliminate alternatives that are unlikely to be chosen, so that later effort can be focused on the more attractive options.” However, screening is based on preference information provided by the decision maker, whereas we only have a partial description of the ecosystem of the person concerned by the consequences of the decision.

3 A knowledge based aiding framework for a common decision problem

In this section, we present a general aiding framework for a common decision problem and we describe the construction of the knowledge models embedded.

3.1 Overview of the framework

Figure 1 shows the structure of an aiding framework for a common decision problem.

The early guidance system is only part of the knowledge based aiding framework

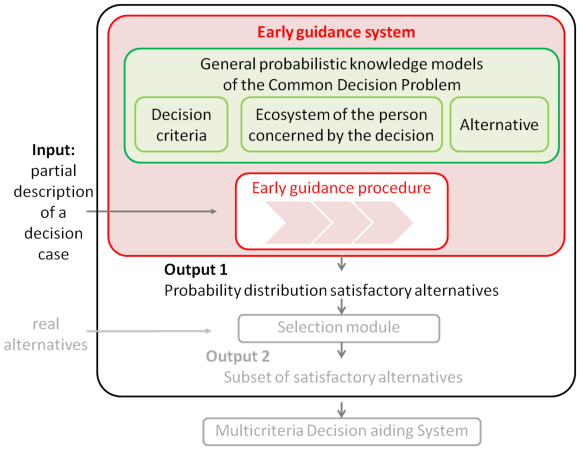


FIGURE 1 – General knowledge based aiding framework for a common decision problem.

The early guidance system is composed of a set of models of probabilistic knowledge, based on Bayesian networks. They concern decision criteria, the ecosystem of the person and the alternatives. Those models are linked together through the evaluation of the alternative regarding the criteria and the ecosystem of a decision case, and the associated satisfaction. Those models are built once for all, since they embed general probabilistic knowledge, which are relatively stable. Based on those models, the early guidance procedure is to be used in each new decision case, meaning in each situation where a decision has to be taken, according to its specific features. The only input is a partial description of the decision case, which can be given by the actors of the choice, or obtained thanks to automatic information collector. The output of the early guidance procedure is a probability distribution of satisfactory alternatives. This output can be exploited through different algorithms in order to get a subset of satisfactory alternatives. In the MCDA literature, this subset is the starting point of the decision process. In common decision problem, we claim that there is often a real need to provide early guidance to get such a subset.

In this article, we focus on the early guidance procedure.

3.2 The Bayesian network model(s) construction

The general knowledge model embedded in the aiding framework is constructed once for all. We propose to embed general probabilistic knowledge about the alternatives (their characteristics) and the ecosystem of the person concerned by a common decision problem in two local Bayesian networks. Those features are next used in the local Bayesian network models that we propose for each criterion.

A *Bayesian network* $\mathcal{B}$ consists in a set of random variables $\mathbf{X} = \{X_1, \ldots, X_n\}$, a simple directed acyclic graph $\mathcal{G} = (\mathbf{X}, \mathbf{E})$, and a set of parameters Θ including the local probability distributions associated with each variable in $\mathbf{X}$. Together, $\mathcal{G}$ and Θ define a probability distribution P over $\mathbf{X}$ which factorizes as $P(\mathbf{X} = \mathbf{x}) = \prod_{X_i \in \mathbf{X}} P(X_i = x \mid pa(X_i))$, where $pa(X_i) \subset \mathbf{X}$ is the set of the parents (immediate predecessors) of X_i in the graph $\mathcal{G}$. Similarly, $ch(X)$ denotes the set of children of X in the graph. Let $Dom(X)$ represents the finite domain of definition of the variable X. Nodes X, Y, and Z form an *unshielded collider* in a DAG if X and Y have a common child Z and there is no arc between X and Y. In this paper, we consider a *causal* Bayesian network, meaning that each directed edge represents a direct causal influence between a cause and an effect.

We propose the construction of the following three Bayesian networks.

The **knowledge model of the alternatives** is a Bayesian network $\mathcal{B}_{Alt}$ whose nodes are the characteristics $\mathbf{X_{Alt}}$ of the alternatives. The graph structure and the local probability distributions embed pieces of knowledge about the alternatives, such as the (in)dependence between the attributes, internal constraints between some features, the state of the market or the actual distribution of some characteristics regarding some parameters. The set of alternatives $\mathcal{A}$ is composed of any configuration of the variables of $\mathbf{X_{Alt}}$ having a non zero prior probability in the Bayesian network $\mathcal{B}_{Alt}$: $\mathcal{A} = \{\mathbf{a} \in \bigtimes_{X \in \mathbf{X_{Alt}}}, P(\mathbf{a}) > 0\}$.

The **knowledge model of the ecosystem of the person** concerned by a common decision problem is a Bayesian network $\mathcal{B}_{eco}$ regrouping all the parameters that may influence the decision process, except the characteristics of the alternatives. The set of nodes is $\mathbf{X_{eco}}$ and the graph reflects statistic and expert knowledge about those variables. A decision case **dc** is a partial instantiation of the ecosystem, $\mathbf{X_{DC}} = \mathbf{dc}$, where $\mathbf{X_{DC}} \subset \mathbf{X_{eco}}$.

We also propose a **knowledge model for each decision criterion**. In the context of common decision problem, a *criterion* is a way to evaluate an alternative according to a decision case and a particular significance axis. A criterion given by the triplet (Imp_g, Ind_g, S_g) is represented in a causal Bayesian network by an unshielded collider as shown in Figure 2. The parents of the unshielded collider are the importance index Imp_g and the evaluation index Ind_g whose ancestors are some characteristics of the alternative in $\mathbf{X_{Alt}}$ and possibly some characteristics of the decision case in $\mathbf{X_{eco}}$. the node S_g associated with the level of satisfaction of the criterion g has exactly two parents since the para-

meters involved in the evaluation of an alternative for the criterion are ancestors of the node Ind_g.

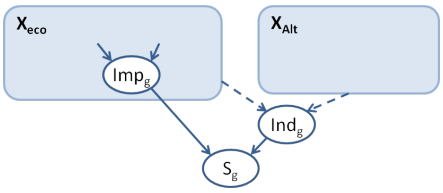


FIGURE 2 – Graph structure of the local Bayesian network of a decision criterion. Dotted arrows represent sets of arrows from any node of the set $\mathbf{X_{Alt}}$ or $\mathbf{X_{eco}}$ to the node Ind_g.

The modeling of a decision criterion by an unshielded collider in a Bayesian network can be found for example in [5, 14] but it was not formalized.

The complete graph of **the Bayesian network of a common decision problem** is obtained by combining the local graphs of each criterion with the graphs of $\mathcal{B}_{Alt}$ and $\mathcal{B}_{eco}$. Combining those graphs consists in merging the nodes that represent the same parameter when several criteria depend on it. This operation does not bring to add a new parent for any node, and thus no conditional probability tables have to be modified.

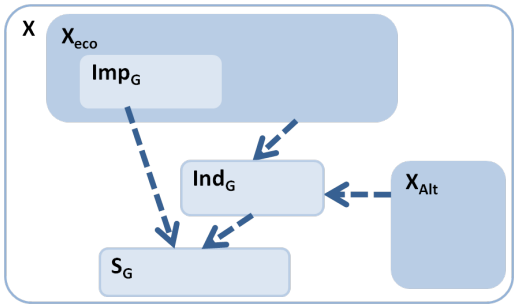


FIGURE 3 – The Bayesian network graph of a common decision problem. Bold dotted arrows represent sets of arrows between any node of the connected sets of nodes.

Figure 3 shows the general structure of the graph of the Bayesian network associated with a common decision problem. The set of nodes is $\mathbf{X} = \mathbf{X_{eco}} \cup \mathbf{X_{Alt}} \cup \mathbf{S_G} \cup \mathbf{Ind_G}$, where :

$\mathbf{S_G} = \{S_g, g \in G\}$ is the set of variables associated with the level of satisfaction of each criterion.

$\mathbf{Imp_G} = \{Imp_g, g \in G\}$ is the set of variables associated with the level of importance of each criterion,

$\mathbf{Ind_G} = \{Ind_g, g \in G\} \cup \mathbf{I}$ is the set of variables associated with the criterion evaluation index of each criterion, augmented with the set $\mathbf{I}$ of additional intermediate variables defined to evaluate a criterion.

$\mathbf{X_{eco}}$ is redefined as the set of the characteristics of the ecosystem, augmented with the set $\mathbf{Imp_G}$, since the level of importance of a criterion is a derived characteristic of the ecosystem.

Once defined, this Bayesian network can be used for early guidance in any new decision case.

4 A Bayesian network based early guidance system

When a new decision case occurs in a common decision problem, it corresponds to a partial instantiation of the ecosystem of the considered situation, because the situation is only partially observed. We thus have a subset $\mathbf{X_{DC}} = \mathbf{dc}$, with $\mathbf{X_{DC}} \subset \mathbf{X_{eco}}$. Remark that the composition of the subset $\mathbf{X_{DC}}$ differs from one situation to another. The uncertainties about the ecosystem can be reduced by updating the unobserved elements of $\mathbf{X_{eco}}$ given $\mathbf{dc}$ in the Bayesian network $\mathcal{B}_{eco}$, meaning computing $P(\mathbf{X_{eco}} \mid \mathbf{dc})$.

The objective of early guidance lead to focus on the characteristics of the alternative for a specific decision case, guided by the desire of a good level of satisfaction of criteria. It's about fixing the ecosystem during the reasoning about the characteristics of the alternative and criteria. We thus make an action that consists in setting the ecosystem as a whole. Indeed, without that action, the reasoning could lead to some belief changes on unobserved variables of the decision case, which is not relevant.

Such an action corresponds to the concept of intervention in a Bayesian network, with the difference that we consider a set of variables, and we do not have a complete instantiation of that set for the intervention. We thus propose definitions for a *probabilistic intervention* on a variable $do(R(X))$and for a set of variables $do(R(\mathbf{X}))$, that extend the definition of an intervention $do(X = x)$.

In the next section, we first explain the principle of *satisfactory situation* that guides the early guidance procedure. Then we define probabilistic interventions and we propose the early guidance procedure based on these two points.

4.1 The satisfactory situation

Given a common decision problem and a decision case, we first define a *satisfactory alternative* in an informal way as an alternative providing a *sufficient* level of satisfaction regarding each criterion, meaning that we have $S_g \geq s$ for each criterion $g \in G$, where $s \in [0, 1]$ represents the minimum level of satisfaction to ba satisfactory. Remark that we consider the level of satisfaction and not the quality index of the criterion, which make relevant to consider the

same value s for all the criterion. Following this, we define a *satisfactory situation* as a situation where a satisfactory alternative has been found.

In order to find a subset of *satisfactory alternatives* for a given decision case we propose the following two steps :

1. Imagine a satisfactory situation in which the person concerned by the consequence of a decision has found a satisfactory alternative ;
2. Compute the probability distribution of the alternative that *explain* that satisfactory situation in the considered decision case.

Remark that we are not interested in the alternative that best explains the satisfactory situation since we do not look for a single alternative, but we look for a subset of satisfactory alternatives.

As introduced above, this result cannot be obtained by a simple inference (updating) in the Bayesian network $\mathcal{B}_{eco}$ because of the unobserved variables in $\mathbf{X_{eco}}$.

We have to compute $P(\mathbf{X_{Alt}} \mid \mathbf{S_G} \geq s, do(R(\mathbf{X_{eco}})))$, where $\mathbf{S_G} \geq s$ represents the observation of a satisfactory situation : $S_g \geq s$, for each $g \in G$ and $do(R(\mathbf{X_{eco}}))$ is a probabilistic intervention that set the ecosystem with the considered decision case. Thanks to this approach, we obtain a probability distribution of the alternatives that *explains* the satisfaction.

With that objective, we know introduce Probabilistic intervention.

4.2 Probabilistic intervention

An intervention $do(X = x)$ in a causal Bayesian network [18] consists in making an action on the variable X that makes the variable being in the state x. An intervention (or manipulation) forces a variable into a certain value or state. As a consequence, incoming arrows have to be removed, since we remove the direct influences by intervening.

Intuitively, a probabilistic intervention $do(R(X = x))$ on a variable X in a causal Bayesian network consists in an action on X that makes $R(X)$ stand fix, such that further reasoning takes place in a restricted world where $R(X)$ stands. This means that the belief on X must not be modified by future observations on other nodes of the Bayesian network. In order to integrate the influence of $R(X)$, the Bayesian network is temporarily modified.

Let consider a a causal Bayesian network $\mathcal{B} = (\mathcal{G}, P)$, with $\mathcal{G} = (\mathbf{X}, \mathcal{E})$.

A *probabilistic intervention* $do(R(X))$ on a variable $X \in \mathbf{X}$ in is an action that integrate $R(X)$ in the probability distribution of a modified Bayesian network $\mathcal{B}' = (\mathcal{G}', P')$ obtained as follows :

1. Remove the node X and the associated arcs,
2. Define the graph $\mathcal{G}'$ as the subgraph of $\mathcal{G}$ induced by the set $\mathbf{X}' \subset \mathbf{X} \setminus \{X\}$ including the children of X and all connected nodes.
3. The probability distribution P' on $\mathbf{X}'$ is defined by :

$$P'(\mathbf{X}') = \prod_{Y \notin ch(X)} P(Y \mid pa(Y)) \prod_{Z \in ch(X)} P'(Z \mid pa_{\mathcal{G}'}(Z))$$

where $P'(Z \mid pa_{\mathcal{G}'}(Z))$ is defined as follows, supposing without loss of generality that Y is the only parent of Z in $\mathcal{G}'$, meaning that X and Y were common parents of Z in $\mathcal{G}$ (see Figure 4) :

$$P'(Z \mid Y) = \sum_x P(Z \mid X = x, Y) R(X = x) \quad (1)$$

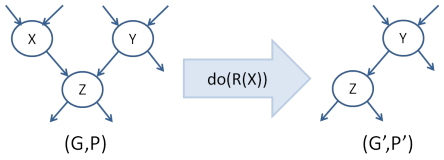


FIGURE 4 – Configuration of the graph around a node X before and after a probabilistic intervention $do(R(X))$.

Figure 4 shows the modification of the graph of the Bayesian network after the intervention $do(R(X))$. The children of X in the initial graph $\mathcal{G}$ are the only nodes whose local conditional probability distribution is modified by the absorption of the node X in $\mathcal{G}'$.

Remark that when the probability distribution $R(X)$ is a dirac on the value x, the definition of $do(R(X))$ and $do(X = x)$ are the same.

We extend the definition of probabilistic intervention for a subset of variables in the configuration of the graph described on the left part of in Figure 5. We consider a Bayesian network $\mathcal{B} = (\mathcal{G}, P)$ whose set of nodes is $\mathbf{X} \cup \mathbf{Z}$, with $\mathbf{X} \cap \mathbf{Z} = \emptyset$ and there is no arrow from a node in $\mathbf{Z}$ toward a node in $\mathbf{X}$.

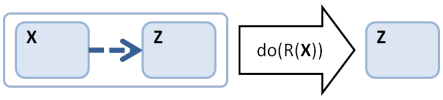


FIGURE 5 – Modification of the graph of a Bayesian network after a probabilistic intervention on a subset of variables. The bold dotted arrow represents the set of arrows from any node of the set $\mathbf{X}$ to any node of the set $\mathbf{Z}$.

A *probabilistic intervention* $do(R(\mathbf{X}))$ on a set $\mathbf{X}$ is an action that integrates $R(\mathbf{X})$ in the probability distribution P'

of the modified Bayesian network $\mathcal{B}' = (\mathcal{G}', P')$ obtained as follows :

1. Define the graph $\mathcal{G}'$ as the subgraph of $\mathcal{G}$ induced by the set $\mathbf{Z}$;
2. Define the subset $\mathbf{X_{out}} \subset \mathbf{X}$ composed of the nodes of $\mathbf{X}$ with an outgoing arc toward $\mathbf{Z}$. By construction the children of a node in $\mathbf{X_{out}}$ are in $\mathbf{Z} \cup \mathbf{X_{out}}$.
3. Define the probability distribution P' on $\mathbf{Z}$:

$$P'(\mathbf{Z}) = \prod_{Y \notin ch(\mathbf{X_{out}})} P(Y \mid pa(Y)) \prod_{Z \in ch(\mathbf{X_{out}})} P'(Z \mid pa_{\mathcal{G}'}(Z))$$

where $P'(Z \mid pa_{\mathcal{G}'}(Z))$ is defined below ; for that, let $\mathbf{X}_{pa(Z)}$ be the set of parents of a node $Z \in ch(\mathbf{X_{out}})$ in $\mathbf{X_{out}}$, meaning that $\mathbf{X}_{pa(Z)} = pa(Z) \cap \mathbf{X_{out}}$. We suppose without loss of generality that Y is the only parent of Z in $\mathcal{G}'$, meaning that Y and $\mathbf{X}_{pa(Z)}$ were common parents of Z in $\mathcal{G}$ (see Figure 6). Remark that it doesn't matter whether $Y \in ch(\mathbf{X_{out}})$ or $Y \notin ch(\mathbf{X_{out}})$, since we compute the new conditional probability distribution of Z to incorporate in it the influence of $R(\mathbf{X})$:

$$P'(Z \mid Y) = \sum_{\mathbf{x}} P(Z \mid \mathbf{X}_{pa(Z)} = \mathbf{x}, Y) R(\mathbf{X}_{pa(Z)} = \mathbf{x}) \tag{2}$$

The probability $R(\mathbf{X}_{pa(Z)} = \mathbf{x})$ is a marginal from $R(\mathbf{X})$.

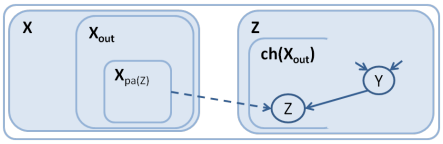


FIGURE 6 – Detail of the parents of a node Z whose probability is re defined in the modified Bayesian network after the probabilistic intervention on a subset of variables $\mathbf{X}$.

A direct consequence of this definition is that there is no distinction between the probabilistic interventions $do(R(\mathbf{X}))$ and $do(R(\mathbf{X_{out}}))$; they produce exactly the same Bayesian network $\mathcal{B}'$.

Remark also that when the set $\mathbf{X}$ is a singleton, the definition of the probabilistic intervention $do(R(\mathbf{X}))$ and $do(R(X))$ are the same.

We can now present the early guidance procedure.

4.3 The early guidance procedure

The early guidance procedure follows the idea of the *satisfactory situation* and it is based on the Bayesian network of the common decision problem. The input of the procedure is a partial description of the ecosystem of the new decision case. The output is the posterior probability distribution of satisfactory alternatives. It can be later used in different ways to provide a list of relevant alternatives.

The three main steps of the early guidance procedure are presented in Figure 7.

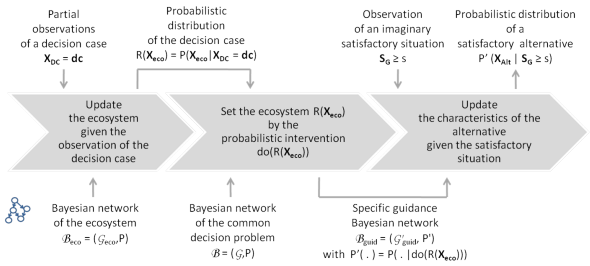


FIGURE 7 – The early guidance procedure.

Here is the description of the early guidance procedure :

Input :

$\mathbf{X_{DC}} = \mathbf{dc}$, with $\mathbf{X_{DC}} \subset \mathbf{X_{eco}}$ a partial description of a decision case,

$\mathcal{B}$, the Bayesian network of the common decision problem,

s, minimum level of satisfaction for the criteria.

Output : $P(\mathbf{X_{Alt}} \mid \mathbf{S_G} \geq s, do(R(\mathbf{X_{eco}})))$ probability distribution of the attributes of the satisfactory alternatives for the given decision case.

step 1 - Compute $R(\mathbf{X_{eco}}) = P(\mathbf{X_{eco}} \mid \mathbf{X_{DC}} = \mathbf{dc})$: use the Bayesian network of the ecosystem $\mathcal{B}_{eco}$ to update the belief about the unobserved variables of the ecosystem from a partial set of observations $\mathbf{X_{DC}} = \mathbf{dc}$ of the decision case.

step 2 - Make the probabilistic intervention $do(R(\mathbf{X_{eco}}))$ in the Bayesian network $\mathcal{B}$ to set the ecosystem with $R(\mathbf{X_{eco}})$. The result is a new Bayesian network $\mathcal{B}' = (\mathcal{G}', P')$ to be used for early guidance in that specific decision case. Note that the condition to make a probabilistic intervention on the set $\mathbf{X_{eco}}$ is fulfilled since there is no arcs from the rest of the graph towards $\mathbf{X_{eco}}$ (see the left part of Figure 8); the Figure shows the graph before and after the probabilistic intervention. The set of nodes of $\mathcal{B}'$ is $\mathbf{X}' = \mathbf{S_G} \cup \mathbf{Ind_G} \cup \mathbf{X_{Alt}}$.

The new probability distribution P' on $\mathbf{X}'$ incorporates $R(\mathbf{X_{eco}})$. It is defined following equation (3), by redefining the local conditional probability distributions of nodes of $\mathbf{S_G}$ and $\mathbf{Ind_G}$.

step 3 - Compute $P'(\mathbf{X_{Alt}} \mid \mathbf{S_G} \geq s)$ to propagate an imaginary satisfactory situation $\mathbf{S_G} \geq s$ in the Bayesian network $\mathcal{B}'$.

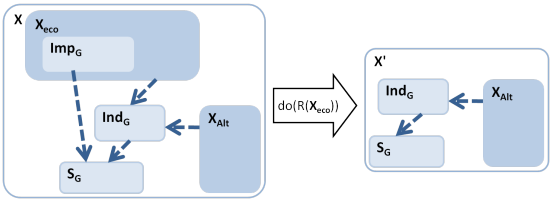


FIGURE 8 – The graph resulting of the probabilistic intervention $do(R(\mathbf{X_{eco}}))$ in the Bayesian network $\mathcal{B}$.

This procedure requires inferences in steps 1 and 3, to compute joint probability distributions on the sets $\mathbf{X_{eco}}$ and $\mathbf{X_{Alt}}$. Since each of these set corresponds to a local Bayesian network, it is sufficient to compute and store the set of local conditional probability distributions associated to each node given its parents, following the partial order induced by the graph. By this way, the required size in memory is decreased. It only depends on the number of parents of the nodes and the size of the domains.

Moreover, thanks to the remark above about equivalent probabilistic interventions, Step 1 can be made easier by considering only the nodes of $\mathbf{X_{eco}}$ that are parents of a node in $\mathbf{X}'$. Let $\mathbf{X_{pa(X')}} \subset \mathbf{X_{eco}}$ be this set : $\mathbf{X_{pa(X')}} = \mathbf{Img_G} \cup (pa(\mathbf{Ind_G}) \cap \mathbf{X_{eco}})$. The probabilistic interventions $do(R(\mathbf{X_{eco}}))$ and $do(R(\mathbf{X_{pa(X')}}))$ are equivalent. So, in Step 1, it is sufficient to compute $R(\mathbf{X_{pa(X')}})$, where $\mathbf{X_{pa(X')}}$ is composed of the nodes of importance of the criteria and the variables of the ecosystem involved in the evaluation of the quality of an alternative regarding a criterion.

The result of this procedure is the posterior probability distribution of satisfactory alternatives for a decision case. As shown in Figure 1, this result is a first step to make easier the selection of a subset of satisfactory alternatives. With this aim, different algorithms could be proposed depending on the common decision problem and the context of use of the aiding framework. For example, in the choice of a manual wheelchair [21], an interesting objective is be to propose a very short list of three real wheelchairs such that all of them are relevant for the considered person and his/her situation, and such that each wheelchair is as different as possible from the other two. Each of the three proposal could then be evaluated in detail according to the most important criteria for this decision case, which would allow the person to understand the advantages and the relative drawback of each wheelchair. For the final selection of a subset of alternatives, several elements can be very different depending on the objective : what is the desired size of the final set of alternatives ? Do we desire all the most satisfactory alternatives, or just a panel ? Can we have a subset of characteristics, and / or an ordered subset of characteristics ? Is it possible (desirable) to involve the actors of the choice in this step ? and what kind of information could be waited ? Finally, the complexity of this final procedure has also to be considered. In this article, we provide a general framework that can suit different common decision problems, and different kind of applications. We let for a future work the selection of a subset of alternatives from the posterior probability distribution that results from the early guidance procedure proposed in this paper.

5 Related works and conclusion

This last part is dedicated to a brief comparison of probabilistic intervention with another proposal that also aim to fix a local conditional probability distribution in a Bayesian network.

5.1 A brief comparison with soft evidence and AEBN

The idea to keep fixed a local probability distribution in a Bayesian network has been approached through *Soft evidence* [1, 22, 15]. It refers to a local probability distribution that defines a constraint on the belief after this information has been propagated ; it describes the state of beliefs “all things considered” and can not be modified by the propagation of other findings. Actually, soft evidence may correspond to the second meaning of the term "uncertain input" in the probabilistic framework [4] : *The input is a partial description of a probability measure ; the uncertainty is part of the input and is taken as a constraint on the final cognitive state. The input is then a correction to the prior cognitive state.* It must be clearly distinguished from the cases where the uncertainty bears on the meaning of the input, which is usually referred as *virtual evidence* or *likelihood evidence* [17] in the context of Bayesian networks. Soft evidence were first introduced in the context of Agent Encapsulated Bayesian networks (AEBN) [1, 22, 13, 11]. In an agent based model using AEBNs, the belief of a receiver agent is updated following the transmission of a soft evidence sent from a publisher agent. Several algorithms have been proposed to manage soft evidence in Bayesian networks. Most of them are based on the Iterative Proportional Fitting Procedure (IPFP), adapted to the factorized form of the joint probability distribution to be used with Bayesian networks [19, 22, 10, 16, 12], including cases with inconsistencies [23]. However, those algorithms are rarely proposed in Bayesian network software tools [1]. This kind of algorithm has been used in Markov networks in a complex application for item planning and capacity management in the automotive industry [6].

Even if both probabilistic intervention and the revision of a probability distribution leads to define a new proba-

1. In 2014, Bayesialab was the only Bayesian network software to propose fixing the probabilities of some selected nodes to the current marginal probabilities [15].

bility distribution, they are deeply different. First because an intervention is an action performed on a *causal* model, and it leads to modify the graph since the action performed move aside any other cause of the value resulting from the action. At the opposite, the revision of a joint probability distribution by a local one has nothing to do with causality, and it keeps unchanged the structure of the graph.

Another point of comparison is worth mentioning : the propagation of several pieces of evidence including at least one piece of soft evidence is not commutative. As a consequence, as long as a *soft evidence* stands, each additional piece of evidence requires to apply again the algorithm to maintain the constraint. Using a probabilistic intervention leads to define temporarily a new Bayesian network in which several pieces of evidence can be easily taken into account.

5.2 Conclusion and perspectives

In this article, we define what is a common decision problem, and introduce the concept of ecosystem of the person concerned by the decision as an explicit set of parameters to be considered. we provide a brief comparison between the early guidance problem and multi-criteria decision analysis. We explain the need of early guidance to facilitate the selection of a a subset of satisfactory alternatives, before the decision process really starts, and without requiring any specific knowledge from the person regarding the problem. We propose an early guidance system that embeds general probabilistic knowledge about a common decision problem and redistributes it appropriately in each decision case as relevant guidance. Our proposition helps the actors of a common decision problem and takes in input only a partial description of their decision case (who, what, where, when, why, how important, etc.).

We propose a new definition of a decision criterion that gathers the importance of the criterion according to the ecosystem of the person, the evaluation of the alternative according to ecosystem, and the satisfaction associated with an alternative in a given decision case. We explain the construction of the Bayesian network for a common decision problem, embedding knowledge about the alternative, the ecosystem and using an unshielded collider in the graph for each decision criterion. We introduce the principle of the satisfactory situation as our starting point in the reasoning towards satisfactory alternatives. In that reasoning, we explain the need of a probabilistic intervention in the Bayesian network to set the ecosystem even though it is partially observed. We extend the concept of intervention proposed by Pearl to the concept of probabilistic intervention. for a single variable, and for a set of variables in a specific configuration of the graph. Finally, we propose the early guidance procedure based on the knowledge model and using probabilistic intervention. The output of the early guidance system is a probability distribution of satisfactory alternatives that is a basis for the selection of a subset of relevant alternatives.

In future works, we plan to complete this proposal with an additional algorithm to provide a subset of satisfactory alternatives based on the posterior probability distribution resulting from the early guidance procedure. With that aim, we will consider a specific common decision problem and a precisely defined context of use of the aiding framework. We also plan to implement and test probabilistic intervention in pyAgrum [2][7], and to implement the early guidance procedure on a specific common decision problem as the one presented in [3].

6 Acknowledgments

This research has been supported by the project ELSAT 2020, the International Campus on Safety and Intermodality in Transportation ; the Hauts-de-France Region ; the European Community ; the Regional Delegation for Research and Technology ; the Ministry of Higher Education and Research ; and the National Center for Scientific Research. The authors gratefully acknowledge the support of these institutions.

7 References

Références

[1] Bloemeke, Mark: *Agent encapsulated Bayesian networks*. Ph.D. thesis, Department of Computer Science, University of South Carolina, 1998.

[2] Bouyssou, Denis: *Readings in Multiple Criteria Decision-Aid*, chapitre Building criteria : a prerequisite for MCDA, pages 58–80. Springer Verlag, Heidelberg, 1990.

[3] Delcroix, Véronique: *Bayesian Network Model of a Criterion in a Repeated Multi-Criteria Decision Problem*. Dans *Actes des Journées de l'Intelligence Artificielle Fondamentale (JIAF)*, Caen, France, july 2017.

[4] Dubois, Didier, Serafin Moral et Henri Prade: *Belief change rules in ordinal and numerical uncertainty theories*. Dans Gabbay, D.M. et Ph. Smets (rédacteurs) : *Belief Change, (D. Dubois, H. Prade, eds.), Vol. 3 of the Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 311–392. Kluwer Academic Publishers, Dordrecht, 1998.

[5] Fenton, Norman et Martin Neil: *Making Decisions : Bayesian Nets and MCDA*. Knowledge-Based Systems, 14(7) :307–325, 2001.

2. http ://agrum.gitlab.io/

[6] Gebhardt, J., C. Borgelt et R. Kruse: *Knowledge Revision in Markov Networks*. Mathware & Soft Computing, 11(1) :93–107, 2004.

[7] Gonzales, Christophe, Lionel Torti et Pierre-Henri Wuillemin: *aGrUM : A Graphical Universal Model Framework*. Dans *Advances in Artificial Intelligence : From Theory to Practice - 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE, Proceedings, Part II*, pages 171–177, Arras, France, June 2017.

[8] Hobbs, Benjamin F. et Peter Meier: *Energy Decisions and the Environment A Guide to the Use of Multicriteria Methods*. Springer US, 2000, ISBN 978-0-7923-7875-4.

[9] Ishizaka, A. et P. Nemery: *Multi-criteria Decision Analysis : Methods and Software*. Wiley, 2013, ISBN 9781118644928.

[10] Kim, Young Gyun, Marco Valtorta et Jirí Vomlel: *A Prototypical System for Soft Evidential Update*. Applied Intelligence, 21(1) :81–97, 2004.

[11] Langevin, Scott: *Knowledge representation, communication, and update in probability-based multiagent systems*. Thèse de doctorat, University of South Carolina, Columbia, SC, USA, 2011, ISBN 978-1-124-64713-5. AAI3454755.

[12] Langevin, Scott et Marco Valtorta: *Performance Evaluation of Algorithms for Soft Evidential Update in Bayesian Networks : First Results*. Dans *SUM*, pages 284–297, 2008.

[13] Langevin, Scott, Marco Valtorta et Mark Bloemeke: *Agent-encapsulated Bayesian networks and the rumor problem*. Dans *AAMAS '10 Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, tome 1, pages 1553–1554, 2010.

[14] Leicester, Philip A., Chris I. Goodier et Paul Rowley: *Using a Bayesian Network to evaluate the social, economic and environmental impacts of community renewable energy*. Dans *Clean Technology for Smart Cities and Buildings (CISBAT)*, 2013.

[15] Mrad, Ali Ben, Véronique Delcroix, Sylvain Piechowiak, Philip Leicester et Mohamed Abid: *An explication of uncertain evidence in Bayesian networks : likelihood evidence and probabilistic evidence - Uncertain evidence in Bayesian networks*. Appl. Intell., 43(4) :802–824, 2015.

[16] Pan, Rong, Yun Peng et Zhongli Ding: *Belief Update in Bayesian Networks Using Uncertain Evidence*. Dans *ICTAI*, pages 441–444, 2006.

[17] Pearl, Judea: *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988, ISBN 0-934613-73-7.

[18] Pearl, Judea: *Causality : Models, Reasoning, and Inference*. University Press, New York, USA, 2000,Second ed., 2009.

[19] Peng, Y., S. Zhang et R. Pan: *Bayesian Network Reasoning with Uncertain Evidences*. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 18(5) :539–564, 2010.

[20] Roy, B.: *Multicriteria Methodology for Decision Aiding*. Kluwer Academic, Dordrecht, 1996.

[21] Sedki, K., V. Delcroix, F. X. Lepoutre, E. Adam, A. P. Maquinghen-Godillon et I. Ville: *Bayesian network model for decision problems*. Dans Klopotek, M.a., M. Marciniak, A. Mykowiecka, W. Penczek et S.t. Wierzchon (Ed.) (rédacteurs) : *Intelligent Information Systems, new approaches*, pages 285–298, Publishing House of University of Podlasie, Siedlce, Poland, June 2010. ISBN 978-83-7051-580-5.

[22] Valtorta, Marco, Young Gyun Kim et Jirí Vomlel: *Soft evidential update for probabilistic multiagent systems*. International Journal of Approximate Reasoning, 29(1) :71–106, 2002.

[23] Vomlel, Jirí: *Integrating Inconsistent Data in a Probabilistic Model*. Journal of Applied Non-Classical Logics, 14(3) :367–386, 2004.

Actes JIAF 2019

Argumentation-based Negotiation with Incomplete Opponent Profiles

Yannis Dimopoulos[1] **Jean-Guy Mailly**[2] **Pavlos Moraitis**[2]

[1] Department of Computer Science, University of Cyprus, Cyprus
[2] LIPADE, Université Paris Descartes, France
yannis@cs.ucy.ac.cy jean-guy.mailly@parisdescartes.fr
pavlos@mi.parisdescartes.fr

Abstract

Computational argumentation has taken a predominant place in the modeling of negotiation dialogues over the last years. A competent agent participating in a negotiation process is expected to decide its next move taking into account an, often incomplete, model of its opponent. This work provides a complete computational account of argumentation-based negotiation under incomplete opponent profiles. After the agent identifies its best option, in any state of a negotiation, it looks for suitable arguments that support this option in the theory of its opponent. As the knowledge on the opponent is uncertain, the challenge is to find arguments that, ideally, support the selected option despite the uncertainty. We present a negotiation framework based on these ideas, along with experimental evidence that highlights the advantages of our approach.

1 Introduction

During the last years computational argumentation has taken a predominant place in the modeling of negotiation dialogues (for a survey see [11], [24]). The goal of a negotiation dialogue is to allow interacting agents to resolve conflicts and reach a mutually accepted agreement, which in this work is a mutually accepted offer (e.g. the price of a product, the mode of payment). In an argumentation-based negotiation (ABN), agents choose offers that are likely to be accepted by the opponent and exchange arguments that support these offers, either based on their own theories (see e.g. [1], [3], [18],[13], [22], [14]), or based on the opponent's profile (e.g. [15], [23], [9]).

The modeling of the opponent profile is an important issue in negotiation dialogues (and more generally other types of dialogue such as persuasion). As explained in [5], although there are important differences between opponent models, there are strong reasons justifying their use, such as the *minimization of negotiation cost*, *the adaptation to the opponent* and *the capacity to reach win-win agreements*, especially in cooperative environments. Learning the opponent profile means learning its acceptance and bidding strategies, the deadlines and its preference profile [5]. In most of the proposed works, the (online) opponent modeling is based on learning techniques (see e.g. [4] for a survey). Apart from the fact that learning the opponent profile with traditional learning techniques is not an easy task, as pointed out in [28], those techniques seem better suited to game-theoretic (or utility-based) negotiations, rather than argumentation-based negotiations. Other works (although they concern persuasion dialogues and legal disputes), have proposed a probabilistic approach for dealing with the uncertainty about the opponent profile. In these works (e.g. [16], [27], [17]), probabilities are used in different ways for finding the arguments that are most likely to be accepted by the opponent. Finally, some works (e.g. [26], [21], [8]) investigate other approaches to modeling the opponent profile in argumentation-based dialogues.

This work advances the state of the art in argumentation-based negotiation by making original contributions to the *opponent modeling*, and the associated *acceptance strategy* (i.e. what offers are most likely to be accepted) as well as *bidding strategy* (i.e. the strategy that an agent applies for choosing the next offer). For opponent modeling, it builds on the work of [10] on control argumentation frameworks (CAFs), a formalism for modeling the uncertainty about the *opponent profile*. More specifically, it borrows the concepts of "on/off" arguments (i.e. arguments we don't know whether they are present or not in a theory), and the three different categories of attacks (i.e. attacks we know their existence and direction, attacks we know the existence but not the direction, attacks we don't know the existence but we

know the direction). This allows generating different profiles modeled as completions of the known part of the opponent's theory, and seeking offers that satisfy all possible profiles (or as many as possible). Regarding the *bidding* and *acceptance strategies*, the originality of this work lies in the assumption that in argumentation-based negotiation, a central challenge for an agent is to lead, by means of appropriate arguments, its counter party to change its theory, and eventually accept the offer it proposes, hence influencing its *acceptance strategy*. Thus, in our approach, we propose a *bidding strategy* that relies on the previous assumption. More precisely, the idea is that a proponent agent uses first its own theory for choosing the best offer to propose, but next, it uses the incomplete theory of its opponent to find the arguments to support it. Then, it seeks and puts forward a set of arguments called *control configuration*, that could reinstate the supporting arguments, if these are rejected in the current state of the argumentative negotiation theories of all (or most) of the generated opponent profiles. Once the arguments of the control configuration are inserted in the opponent theory, they would, ideally, allow it to reach an agreement with the proponent, thus they alter its *acceptance decision*.

2 Background

We assume that the reader is familiar with abstract argumentation frameworks as introduced in [12], presented as a pair $\langle A, R\rangle$, where A is a set of *arguments*, and $R \subseteq A \times A$ is an *attack relation*. The relation *a attacks b* is denoted by $a\ R\ b$ or $(a, b) \in R$. Different acceptability semantics were also introduced in this work. Based on the acceptability semantics, we can define the status of any argument, namely *skeptically accepted, credulously accepted* and *rejected arguments*.

Now we introduce briefly the control argumentation frameworks (CAFs) proposed in [10], and discusses how they capture the knowledge of an agent on its opponents. On a high level, a CAF is an argumentation framework where arguments are divided in three parts, *fixed*, *uncertain* and *control*.

The *fixed* part of the theory concerns the certain knowledge that an agent holds about its opponent. This includes arguments as well as attacks that undoubtedly belong to the argumentation theory of the opponent. For instance, a seller agent knows that the customer agent prefers European cars, that safety is an important issue for it and that it prefers electric or gasoline-powered cars than diesel cars. The *uncertain* part captures the uncertainty about the presence of arguments in a theory (expressed by the "on/off" arguments as shown below), as well as the presence and the direction of attacks between arguments in this theory. It reflects the uncertainty that arises due to lack of complete information on the current state of the world that determines the decisions of the opponent, but also its beliefs and preferences. For example, the seller agent may not know the income of the customer agent, whether a car is a social status symbol for it, the highest price that it is ready to pay, or whether it is willing to pay more if some extras are included, and payment by installments is accepted. Finally, the *control* part contains arguments that can be used against arguments of the fixed or uncertain parts that attack arguments that are in favour of some offer of the proponent. Therefore, the control part serves to ensure that arguments in the fixed part that support some offer of the seller that is not adequate with some certain (i.e. European car) or uncertain (e.g. max price, preferred mode of payment) preferences of the customer, can be accepted under some circumstances. For instance, a control argument could allow a seller agent to propose a car from abroad Europe (which is against the known preference of the customer agent and represented in the fixed part) by proposing some interesting options (e.g. five airbags knowing that safety is an important issue for the customer and also represented in the fixed part) and in a price that is probably higher than the highest price the customer is intended to pay (this is part of the uncertain knowledge) but which allows the seller to accept a payment by installments, if this is the preferred payment mode for the customer (this is also part of the uncertain knowledge). Formally, a CAF is defined as follows :

Definition 1 *Let $\mathcal{L}$ be a language from which we can build arguments, and let* $\texttt{Args}(\mathcal{L})$ *be the set which contains all those arguments. A Control Argumentation Framework (*CAF*) is a triple $\mathcal{CAF} = \langle F, C, U\rangle$ where F is the* fixed part*, U is the* uncertain part *and C is the* control part *of $\mathcal{CAF}$ with :*

1. $F = \langle A_F, \rightarrow\rangle$ where A_F is a set of arguments that we know they belong to the system and $\rightarrow \subseteq (A_F \cup A_U) \times (A_F \cup A_U)$ is an attack relation representing a set of attacks for which we are aware both of their existence and their direction.

2. $U = \langle A_U, (\rightleftarrows \cup \dashrightarrow)\rangle$ where A_U is a set of arguments for which we are not sure that they belong to the system, $\rightleftarrows \subseteq (((A_U \cup A_F) \times (A_U \cup A_F)) \setminus \rightarrow)$ is an attack relation representing a set of attacks for which we are aware of their existence but not of their direction, and $\dashrightarrow \subseteq (((A_U \cup A_F) \times (A_U \cup A_F)) \setminus \rightarrow)$ is an attack relation representing a set of attacks for which we are not aware of their existence but we are aware of their direction, with $\rightleftarrows \cap \dashrightarrow = \emptyset$.

3. $C = \langle A_C, \Rrightarrow\rangle$ where A_C is a set of arguments, called control arguments*, that the agent can choose to use or not, and $\Rrightarrow \subseteq \{(a_i, a_j) \mid a_i \in A_C,\ a_j \in A_F \cup A_C \cup A_U\}$ is an attack relation.*

A_F, A_U and A_C are disjoint subsets of $\texttt{Args}(\mathcal{L})$.

A CAF features a set of distinct attack relations that capture different sorts of information. Its simplest part is $\langle A_F, \rightarrow \cap (A_F \times A_F)\rangle$, which is a classical AF that contains

the indisputable knowledge of the agent on its opponent. The idea of CAFs essentially extends this basic argumentation framework with additional attack relations defined on arguments from the sets A_U and A_C. For instance, there is an attack $(a_i, a_j) \in \rightleftarrows$, with $a_i, a_j \in A_F$ when it is certain that the two arguments exist and are in conflict (*e.g.* because they make mutually exclusive claims), but the direction of the attack(s) is unknown (*e.g* because of lack of information on the intrinsic strength of arguments, or on the preference relation between arguments). An attack $(a_i, a_j) \in \rightarrow$, with $a_i \in A_U$ and $a_j \in A_F$, represents a situation where it is unknown whether a_i is present in the system (*e.g.* some of its premises could be false at the current time), but *if a_i is in the system*, then a_i definitely attacks a_j.

Central to controllability is the notion of *completion* of a CAF. Intuitively, a completion is a classical AF which is built from the CAF, by choosing one of the possible options for each uncertain argument or attack.

Definition 2 *[10] Given a CAF $\mathcal{CAF} = \langle F, C, U\rangle$, a completion of $\mathcal{CAF}$ is an AF $\mathcal{AF} = \langle A, R\rangle$, s.t.*
1. $A = A_F \cup A_C \cup A_{comp}$ where $A_{comp} \subseteq A_U$;
2. if $(a, b) \in R$, then $(a, b) \in \rightarrow \cup \rightleftarrows \cup \dashrightarrow \cup \Rrightarrow$;
3. if $(a, b) \in \rightarrow$, then $(a, b) \in R$;
4. if $(a, b) \in \rightleftarrows$ and $a, b \in A$, then $(a, b) \in R$ or $(b, a) \in R$;
5. if $(a, b) \in \Rrightarrow$ and $a, b \in A$, then $(a, b) \in R$.

Note that the definition of a completion leaves the attacks from $\dashrightarrow$ unspecified, as these attacks may not appear in the theory. For some examples of completions the reader can see [10].

Controllability means that we can select a subset $A_{conf} \subseteq A_C$ and the corresponding attacks $\{(a_i, a_j) \in \Rrightarrow | \ a_i \in A_C, \ a_j \in (A_F \cup A_C \cup A_U)\}$ such that whatever the completion of $\mathcal{CAF}$, a given target is always reached. We focus on two kinds of targets : credulous acceptance of a set of arguments (this is reminiscent of extension enforcement [6]), and skeptical acceptance of a set of arguments.

Definition 3 *[10] A control configuration of a CAF $\mathcal{CAF} = \langle F, C, U\rangle$ is a subset $A_{conf} \subseteq A_C$. Given a set of arguments $T \subseteq A_F$ and a semantics σ, we say that T is skeptically (resp. credulously) reached by the configuration A_{conf} under σ if T is included in every (resp. at least one) σ-extension of every completion of $\mathcal{CAF}' = \langle F, C', U\rangle$, with $C' = \langle A_{conf}, \{(a_i, a_j) \in \Rrightarrow | \ a_i \in A_C, \ a_j \in (A_F \cup A_C \cup A_U)\}\rangle$. We say that $\mathcal{CAF}$ is skeptically (resp. credulously)* controllable *w.r.t. T and σ.*

In a nutshell, CAFs are a powerful enabler of advanced negotiation techniques, that blend together a number of desirable features such as the qualitative representation of uncertainty, simultaneous reasoning with different profiles through completions, simultaneous consideration of both certain and uncertain knowledge of the opponent, the use of control arguments (corresponding to a persuasion phase embedded in negotiation, allowing for the reinstatement of rejected arguments), along with a computational model based on QBFs.

3 The Negotiation Framework

This section presents a new argumentation-based negotiation framework that relies on CAFs [10] for representing the incomplete information that agents have about their opponents. Agents communicate through the exchange of messages (or dialogue moves, see e.g. [11]). We assume that agents play the roles of the proponent and opponent in a turn-taking round-based protocol (e.g. similar to the alternating offers protocol of [14]), where a proponent initiates a round and passes the token to its opponent when it is unable to defend an offer rejected by the opponent. The opponent may accept an offer when one of the supporting arguments is an acceptable argument for it, or reject an offer if it cannot accept any of the different supporting arguments sent by the proponent. We build on the works of [1], [14], and in the following, $\mathcal{L}$ denotes a logical language, and $\equiv$ an equivalence relation associated with it. From $\mathcal{L}$, a set $O = \{o_1, \ldots, o_n\}$ of *n offers* is identified, such that $\nexists o_i, o_j \in O$ such that $o_i \equiv o_j$. This means that the offers are different. Offers correspond to the different alternatives (e.g. prices for a product) that can be exchanged during the negotiation dialogue. We assume that agents share the same set of offers O but those offers can be supported by different practical arguments (although not necessarily) in the theories of the negotiating agents. By argument, we mean a *reason* in believing (called epistemic arguments) or doing something (called practical arguments). The set $Args(\mathcal{L})$ is then divided into two subsets : a subset $Args_p(\mathcal{L})$ of practical arguments supporting offers, and a subset $Args_e(\mathcal{L})$ of epistemic arguments supporting beliefs. Thus, $Args(\mathcal{L}) = Args_p(\mathcal{L}) \cup Args_e(\mathcal{L})$. A negotiation theory is therefore represented as follows :

Definition 4 (Negotiating agent theory) *Let O be a set of n offers. A* negotiating theory *of an agent α is a tuple $\mathcal{T} = \langle O, \mathcal{T}^\alpha, \mathcal{CAF}^{\alpha,\beta}, \mathcal{F}^\alpha\rangle$ with $\mathcal{T}^\alpha = \langle A^\alpha, \rightarrow_\alpha\rangle$ and $\mathcal{CAF}^{\alpha,\beta} = \langle F^{\alpha,\beta}, U^{\alpha,\beta}, C^{\alpha,\beta}\rangle$ and where :*

1. $A^\alpha \subseteq Args(\mathcal{L})$ is a set of arguments s.t. $A^\alpha = A^\alpha_p \cup A^\alpha_e$ where A^α_p is a set of practical arguments, A^α_e a set of epistemic arguments, and $A^\alpha_c \subseteq A^\alpha_e$ is the set of control arguments. For the attack relation it holds $\rightarrow_\alpha = \rightarrow_p \cup \rightarrow_e \cup \rightarrow_m$, with $\rightarrow_p \subseteq A^\alpha_p \times A^\alpha_p$, representing an attack relation for practical arguments, $\rightarrow_e \subseteq A^\alpha_e \times A^\alpha_e$ representing an attack relation for epistemic arguments and $\rightarrow_m \subseteq A^\alpha_e \times A^\alpha_p$ representing an attack relation between epistemic and practical arguments i.e. $(a, \delta) \in \rightarrow_m$, if $a \in A^\alpha_e$ and $\delta \in A^\alpha_p$ (see [2], [14]).

2. $C\mathcal{AF}^{\alpha,\beta}$ is defined by :

— $F^{\alpha,\beta}=\langle A_F^{\alpha,\beta}, \rightarrow_{\alpha,\beta}\rangle$ *with* $A_F^{\alpha,\beta}=A_{F_e}^{\alpha,\beta} \cup A_{F_p}^{\alpha,\beta}$, $\rightarrow_{\alpha,\beta}=\rightarrow_e^{\alpha,\beta} \cup \rightarrow_p^{\alpha,\beta}$ *and* $\langle A_{F_e}^{\alpha,\beta}, \rightarrow_e^{\alpha,\beta}\rangle$ *defining the epistemic arguments subpart s.t.* $\rightarrow_e^{\alpha,\beta} \subseteq (A_{F_e}^{\alpha,\beta} \cup A_{U_e}^{\alpha,\beta}) \times (A_{F_e}^{\alpha,\beta} \cup A_{U_e}^{\alpha,\beta})$. *The above hold also for the practical arguments subpart. It also holds* $A_U^{\alpha,\beta}=A_{U_e}^{\alpha,\beta} \cup A_{U_p}^{\alpha,\beta}$.

— $U^{\alpha,\beta}=\langle A_U^{\alpha,\beta}, \rightleftarrows_{\alpha,\beta} \cup \dashrightarrow_{\alpha,\beta}\rangle$ *with* $\rightleftarrows_{\alpha,\beta}=\rightleftarrows_e \cup \rightleftarrows_p$, $\dashrightarrow_{\alpha,\beta}=\dashrightarrow_e \cup \dashrightarrow_p$ *and* $\langle A_{U_e}^{\alpha,\beta}, \rightleftarrows_e \cup \dashrightarrow_e\rangle\rangle$, $\rightleftarrows_e \subseteq ((A_{U_e}^{\alpha,\beta} \cup A_{F_e}^{\alpha,\beta}) \times (A_{U_e}^{\alpha,\beta} \cup A_{F_e}^{\alpha,\beta}))\setminus \rightarrow_e^{\alpha,\beta}$), $\dashrightarrow_e \subseteq ((A_{U_e}^{\alpha,\beta} \cup A_{F_e}^{\alpha,\beta}) \times (A_{U_e}^{\alpha,\beta} \cup A_{F_e}^{\alpha,\beta}))\setminus \rightarrow_e^{\alpha,\beta}$, *defining the epistemic arguments subpart. The same hold for the practical arguments subpart.* $\rightleftarrows_e \cap \dashrightarrow_e = \emptyset$.

— $C^{\alpha,\beta}=\langle A_c^{\alpha}, \Rrightarrow\rangle$ *where* $\Rrightarrow \subseteq \{(a_i, a_j) \mid a_i \in A_c^{\alpha}$ *and* $a_j \in A_c^{\alpha} \cup A_{F_e}^{\alpha,\beta} \cup A_{U_e}^{\alpha,\beta}\} \setminus (\rightarrow_e^{\alpha,\beta} \cup \rightleftarrows_e \cup \dashrightarrow_e))$.

3. $\mathcal{F}^{\alpha} : O \rightarrow 2^{A_{p^{\alpha}}}$ *s.t* $\forall i, j$ *with* $i \neq j$, $\mathcal{F}^{\alpha}(o_i) \cap \mathcal{F}^{\alpha}(o_j) = \emptyset$. *Let* $A_{pO}^{\alpha} = \bigcup \mathcal{F}^{\alpha}(o_i)$ *with* $i = 1, \ldots, n$. *This function returns the practical arguments supporting offers in* O.

In the following we present the different procedures that implement the new negotiation framework of this paper.

3.1 Best Offers Selection

Algorithm 1 is the procedure invoked by the proponent agent α in order to compute, first, its best offer, based on its own theory, and it is implemented through function $comp\_next\_offer$. This function looks for the best offer supported by an acceptable practical argument by using a ranking on the supporting arguments based on a partial preorder (other methods can be also applied here). Then, based on its $C\mathcal{AF}^{\alpha,\beta}$, it computes the practical arguments that support this offer in its opponent theory and calls a procedure, implemented by algorithm 2, that selects the supporting argument to be sent. If the proponent agent has no (other) offer to propose, the opponent of the agent is informed by a suitable message (i.e. nothing).

Algorithm 1: choose-best-offer($O, \mathcal{T}^{\alpha}, C\mathcal{AF}^{\alpha,\beta}, \mathcal{F}^{\alpha,\beta}(o)$)

```
1 o ← comp_next_offer(O, T^α);
2 if o ≠ ∅ then
3 |   F^{α,β}(o) ← compute_sup_arg(o, A^{α,β}_{F_p} ∪ A^{α,β}_{U_p});
4 |   call choose-support-arg(o, F^{α,β}(o), CAF^{α,β});
5 else
6 |   message(α,β)=nothing;   send(message(α,β));
```

3.2 Supporting Argument Selection

The algorithm described below, selects (through function $choose - arg$, where the choice can be random, as herein, or based on other methods) the argument that the proponent agent α sends to its opponent agent to support its offer. Moreover, another procedure finds the arguments that defend this supporting argument whenever this argument is currently rejected by the opponent. This task is carried out by the procedure implemented by algorithm 3. If there is no other available argument that supports the current offer, the agent abandons this offer and passes the negotiation token to the opponent agent.

Algorithm 2: choose-support-arg($o, \mathcal{F}^{\alpha,\beta}(o), C\mathcal{AF}^{\alpha,\beta}$)

```
1 if F^{α,β}(o) ≠ ∅ then
2 |   θ ← choose-arg(F^{α,β}(o));
3 |   call defend-offer(o, θ, F^{α,β}(o), CAF^{α,β})
4 else
5 |   O=O − {o};   A^α_p = A^α_p − F^α(o);
6 |   message(α,β)=give_token;   send(message(α,β));
```

3.3 The Bidding Strategy

The *bidding strategy* of the proponent agent is implemented by algorithm 3. The main task here is to defend the proposed offer by an argument that (as said before) supports the offer in the opponent's theory. Consider for instance a car seller agent who proposes an expensive luxury SUV of a prestigious brand to a customer who, as the agent understands, seems to afford it. The reason (argument) that the seller agent has chosen this particular car is probably the high sales commission that it brings. However, this is not an argument it can use to convince its customer. The pool of appropriate arguments could include the smooth ride, fast acceleration, high top speed, off-road capabilities, safety features, or even the high social status associated with the brand. In fact, the discovery of those arguments takes place inside algorithms 1 and 2. The role of the bidding strategy algorithm is to determine whether such a supporting argument is already acceptable in the opponent's theory, or to search for a *control configuration* that can defend the selected supporting argument under all possible opponent profiles.

More precisely, acceptance in the context of incomplete theories is based on the notion of *completion* which represents a possible profile (see definition 2). The computation in line 1 of the algorithm relies on reasoning with Quantified Boolean Formulas (QBFs), as described in [10], that is carried out by the `quantom` solver [25]. The credulous controllability wrt the theory $A_F^{\alpha,\beta} \cup A_U^{\alpha,\beta}$ (i.e. arguments in A_c^{α} are not considered in this case) is computed by using the following *Formula* 1 :

$$\forall\{on_{x_i} \mid x_i \in A_U^{\alpha,\beta}\}\forall\{att_{x_i,x_j} \mid (x_i, x_j) \in \dashrightarrow_{\alpha,\beta} \cup \rightleftarrows_{\alpha,\beta}\}$$
$$\exists\{acc_{x_i} \mid x_i \in \mathbf{A}\}[\Phi_{st}^{cr}(C\mathcal{AF}, \theta)$$
$$\vee(\bigvee_{(x_i,x_j)\in\rightleftarrows_{\alpha,\beta}}(\neg att_{a_i,a_j} \wedge \neg att_{a_j,a_i}))]$$

where $\mathbf{A} = A_F^{\alpha,\beta} \cup A_{comp}$ with $A_{comp} \subseteq A_U^{\alpha,\beta}$.

Algorithm 3: defend-offer($o, \theta, \mathcal{F}^{\alpha,\beta}(o), \mathcal{CAF}^{\alpha,\beta}$)

1 **if** θ *is credulously accepted in all completions of the theory* $A_F^{\alpha,\beta} \cup A_U^{\alpha,\beta}$
2 **then**
3 offer$(\alpha,\beta) = \langle o, \theta, \langle \emptyset, \emptyset \rangle \rangle$;
4 $\mathcal{F}^{\alpha,\beta}(o) = \mathcal{F}^{\alpha,\beta}(o) - \{\theta\}$;
5 message(α,β)=offer(α,β); send(message(α,β))
6 **else**
7 $S \leftarrow$ comp_contr_conf($\mathcal{CAF}^{\alpha,\beta}, \theta$);
8 **if** $S \neq \emptyset$ **then**
9 $\mathcal{R} = \{(a_i, a_j) | a_i \in S, a_j \in A_F^{\alpha,\beta} \cup A_U^{\alpha,\beta}\}$; offer$(\alpha,\beta) = \langle o, \langle \theta, \langle S, \mathcal{R} \rangle \rangle \rangle$;
10 message(α,β)=offer(α,β); send(message(α,β))
11 **else**
12 $\mathcal{F}^{\alpha,\beta}(o) = \mathcal{F}^{\alpha,\beta}(o) - \{\theta\}$;
13 call choose-support-arg($o, \mathcal{F}^{\alpha,\beta}(o), \mathcal{CAF}^{\alpha,\beta}$);

The on_{x_i} variable means that the argument x_i currently belongs to the system; it is used for making the differentiation between the completions where x_i is included and those where it is not. Similarly, att_{x_i,x_j} is true when there is an attack from x_i to x_j. This variable has to be true if (x_i, x_j) is a fixed attack of $\mathcal{CAF}$. Otherwise the truth value of this variable allows to make the distinction between the completions where (x_i, x_j) is included and those where it is not. Finally acc_{x_i} is a propositional variable representing the acceptance status of the argument x_i. The propositional matrix $\Phi_{st}^{cr}(\mathcal{CAF}, \theta)$ of the formula is satisfiable when θ belongs to at least one extension of a completion of $\mathcal{CAF}$ (more details about this part are given later). Straightforwardly, the prefix of the formula corresponds to an enumeration of every completion (by the $\forall$ quantifiers); for every such completion, we have to search for at least one extension (represented by the existentially quantified part) such that θ belongs to it.
Now, in case this computation succeeds, θ is acceptable in all possible opponent profiles (completions), and agent α sends to agent β the offer o, along with θ.

In case θ is not acceptable wrt the above theory, agent α reacts as depicted in lines 7-13 of algorithm 3. First, it uses its *CAF* to seek a *control configuration* S, that defends θ. This is again a problem on QBFs that is solved by a call to `quantom` solver (line 7 of the algorithm). However, this time arguments in A_c^α are considered and credulous controllability is computed by using the following *Formula* 2 :

$$\exists\{on_{x_i} \mid x_i \in A_c^\alpha\}\forall\{on_{x_i} \mid x_i \in A_U^{\alpha,\beta}\}\forall\{att_{x_i,x_j} \mid (x_i, x_j) \in \dashrightarrow_{\alpha,\beta} \cup \rightleftarrows_{\alpha,\beta}\}\exists\{acc_{x_i} \mid x_i \in \mathbf{A}\}[\Phi_{st}^{cr}(\mathcal{CAF}, \theta) \vee (\bigvee_{(x_i,x_j)\in\rightleftarrows_{\alpha,\beta}}(\neg att_{a_i,a_j} \wedge \neg att_{a_j,a_i}))]$$

where $\mathbf{A} = A_F^{\alpha,\beta} \cup A_c^\alpha \cup A_{comp}$ with $A_{comp} \subseteq A_U^{\alpha,\beta}$.

Note that this formula is very similar to the previous one. This time, the existential quantifier over the on_{x_i} variables, for $x_i \in A_c^\alpha$, corresponds to the search for one control configuration. So the whole formula corresponds to the definition of credulous controllability : the formula is true if there is a control configuration such that, for every completion, θ belongs to at least one extension.

In both above cases we use the formula $\Phi_{st}^{cr}(\mathcal{CAF}, \theta) = \Phi_{st}(\mathcal{CAF}) \wedge acc_\theta$ which is based on

$$\Phi_{st}(\mathcal{CAF}) = \bigwedge_{x_i\in A_F^{\alpha,\beta}}[acc_{x_i} \Leftrightarrow \bigwedge_{x_j\in\mathbf{A}}(att_{x_j,x_i} \Rightarrow \neg acc_{x_j})] \wedge \bigwedge_{x_i\in A_C^{\alpha,\beta}\cup A_U^{\alpha,\beta}}[acc_{x_i} \Leftrightarrow (on_{x_i} \wedge \bigwedge_{x_j\in\mathbf{A}}(att_{x_j,x_i} \Rightarrow \neg acc_{x_j}))] \wedge \bigwedge_{(x_i,x_j)\in\rightarrow_{\alpha,\beta}\cup\Rrightarrow_{\alpha,\beta}} att_{x_i,x_j} \bigwedge_{(x_i,x_j)\in\rightleftarrows_{\alpha,\beta}} att_{x_i,x_j} \vee att_{x_j,x_i} \bigwedge_{(x_i,x_j)\notin\mathbf{R}} \neg att_{x_i,x_j}$$

where $\mathbf{R} = \rightarrow_{\alpha,\beta} \cup \Rrightarrow_{\alpha,\beta} \cup \dashrightarrow_{\alpha,\beta} \cup \rightleftarrows_{\alpha,\beta}$.

Moreover, in the first case, where the control arguments are not used (in *Formula* 1), $\bigwedge_{x_i\in A_C^{\alpha,\beta}\cup A_U^{\alpha,\beta}}$ becomes $\bigwedge_{x_i\in A_U^{\alpha,\beta}}$.

This formula is a generalization of the encoding of stable semantics defined in [7]. When every *att*-variable and every *on*-variable is assigned a truth value, this assignment corresponds to a completion. Then, the consistent truth assignments of the *acc*-variables correspond to the set of stable extensions of the completion. This means that if $\Phi_{st}(\mathcal{CAF}) \wedge acc_\theta$ is satisfiable, then θ belongs to at least one stable extension of the completion which is represented by the *att* and *on*-variables.

Now if in this second case the call succeeds, agent α sends offer o to agent β, along with the supporting argument θ, the set of arguments S, and the associated attacks R. Otherwise, the agent abandons this argument and picks another from $\mathcal{F}^{\alpha,\beta}(o)$ in order to continue defending o. This is done by function *choose-support-arg*. Recall that our approach looks for sets of arguments that are control configurations, i.e. work for all possible profiles of agent β. However, if there is no such solution, the QBF based techniques of `quantom` [25], can find sets of arguments that work for most of these profiles.

In the following we define an operator $\oplus$ that is used in algorithms 4 and 5.

Definition 5 *Let A_1, A_2, A_3 be sets. We define $(A_1, A_2) \oplus A_3$ as the pair (A_1', A_2') such that $A_1' = A_1 \setminus (A_1 \cap A_3)$ and $A_2' = A_2 \cup (A_1 \cap A_3)$.*

At the beginning of the negotiation each agent has in its theory (i.e. A^α and A^β respectively) only a part of the possible epistemic arguments (wrt a specific application). That means that some arguments are in A^α and not in A^β (and vice-versa). However, when an agent will use arguments (and the associated attacks) that do not belong to the opponent's theory, the opponent agent will add them (as well as the associated attacks) in its own theory, and it will be able to use them from that point onwards in the negotiation. This situation may take place in the algorithms 4 and 5.

3.4 The Acceptance Strategy

This section discusses Algorithm 4, that implements the *acceptance strategy* of an agent. Upon receiving an offer and its supporting arguments (and the associated attacks) sent by a proponent agent, the algorithm updates the theory as well as the $\mathcal{CAF}$ of the receiving agent by integrating the supporting arguments, the defending arguments (i.e. the control configuration), and the associated attacks into both theories (i.e. the receiving agent own theory and its $\mathcal{CAF}$). Then, the receiver agent either accepts the offer (i.e. if the supporting arguments are acceptable) and informs the proponent accordingly, or sends to the proponent the reasons for rejecting its offer.

Algorithm 4: decide-upon-offer($\mathcal{T}^\alpha, \mathcal{CAF}^{\alpha,\beta}$, offer($\beta, \alpha$))

1 $\langle o, \theta, \langle S, \mathcal{R}\rangle\rangle$=offer($\beta, \alpha$);
2 **if** $S \neq \emptyset$ **then**
3 $\quad \mathcal{T}^\alpha$=($A^\alpha \cup S, \rightarrow_\alpha \cup \mathcal{R}$);
4 $\quad (A_U^{\alpha,\beta}, A_F^{\alpha,\beta}) = (A_U^{\alpha,\beta}, A_F^{\alpha,\beta}) \oplus S$;
5 $\quad (\dashrightarrow_{\alpha,\beta}, \rightarrow_{\alpha,\beta}) = (\dashrightarrow_{\alpha,\beta}, \rightarrow_{\alpha,\beta}) \oplus \mathcal{R}$;
6 $\quad (\rightleftarrows_{\alpha,\beta}, \rightarrow_{\alpha,\beta}) = (\rightleftarrows_{\alpha,\beta}, \rightarrow_{\alpha,\beta}) \oplus \mathcal{R}$
7 **if** θ *is a credulous conclusion of theory* $\mathcal{T}^\alpha$ **then**
8 $\quad$ message(α, β)=Accept(o);
9 $\quad$ send(message(α, β))
10 **else**
11 $\quad$ Compute Q $\subseteq \mathcal{E}$ where $\mathcal{E}$ is an extension of $\mathcal{T}^\alpha$ and Q is the set of arguments from which θ is reachable in the attack graph;
12 $\quad$ Reasons=$\{(p, \theta) | (p, \theta) \in \rightarrow_\alpha$ and $p \in Q\}$;
13 $\quad$ message(α, β)=Reject($o, \theta, \langle Q, Reasons\rangle$);
14 $\quad$ send(message(α, β));

3.5 The Negotiation Protocol

The algorithm 5 described below implements the core procedure that drives the overall negotiation between the two negotiating agents through the necessary updates of their negotiation theories and calls to appropriate functions. The first part of algorithm (lines 1-2) implements the behavior of an agent when it is the proposer of the first offer, whereas the second part (lines 3-24) is concerned with its reaction when it receives an answer from another agent (i.e. the opponent). While the first part is straightforward as it concerns the selection of the best offer to propose, the second part is more involved and breaks down to several subcases. Those cases concern different situations that may arise during a negotiation, such as the rejection of an offer by the opponent, the acceptance of an offer (that terminates the negotiation with an agreement), the situation where the opponent informs that it has no other offer to propose, the situation where the opponent responds that it has no offer to propose too in a received similar message by the (proponent) agent (this ends the negotiation without agreement), the situation where an agent informs that it gives the token, and the situation where an offer is received and the receiver agent has to decide upon its acceptance or rejection. The example below explains how the protocol works.

Algorithm 5: Procedure negotiate($\langle O, \mathcal{T}^\alpha, \mathcal{CAF}^{\alpha,\beta}, \mathcal{F}^\alpha\rangle$)

1 **if** *agent* α *proposes first* **then**
2 $\quad$ call choose-best-offer($O, \mathcal{T}^\alpha, \mathcal{CAF}^{\alpha,\beta}, \mathcal{F}^{\alpha,\beta}(o)$);
3 **while** *true* **do**
4 $\quad$ get message(β, α);
5 $\quad$ **switch** *message*(β, α) **do**
6 $\quad\quad$ **case** *Reject*($o, \theta, \langle Q, Reasons\rangle$) **do**
7 $\quad\quad\quad$ $(A_U^{\alpha,\beta}, A_F^{\alpha,\beta}) = (A_U^{\alpha,\beta}, A_F^{\alpha,\beta}) \oplus Q$;
8 $\quad\quad\quad$ $(\dashrightarrow_{\alpha,\beta}, \rightarrow_{\alpha,\beta}) = (\dashrightarrow_{\alpha,\beta}, \rightarrow_{\alpha,\beta}) \oplus Reasons$;
9 $\quad\quad\quad$ $(\rightleftarrows_{\alpha,\beta}, \rightarrow_{\alpha,\beta}) = (\rightleftarrows_{\alpha,\beta}, \rightarrow_{\alpha,\beta}) \oplus Reasons$;
10 $\quad\quad\quad$ call defend-offer($o, \theta, \mathcal{F}^{\alpha,\beta}(o), \mathcal{CAF}^{\alpha,\beta}$);
11 $\quad\quad$ **case** *Accept(o)* **do**
12 $\quad\quad\quad$ End of negotiation with agreement on offer o
13 $\quad\quad$ **case** *nothing* **do**
14 $\quad\quad\quad$ **if** $O \neq \emptyset$ **then**
15 $\quad\quad\quad\quad$ call choose-best-offer($O, \mathcal{T}^\alpha, \mathcal{CAF}^{\alpha,\beta}, \mathcal{F}^{\alpha,\beta}(o)$);
16 $\quad\quad\quad$ **else**
17 $\quad\quad\quad\quad$ answer(α, β)=*nothing_too*;
18 $\quad\quad\quad\quad$ send(answer(α, β))
19 $\quad\quad$ **case** *nothing_too* **do**
20 $\quad\quad\quad$ End of negotiation without agreement
21 $\quad\quad$ **case** *give_token* **do**
22 $\quad\quad\quad$ call choose-best-offer($O, \mathcal{T}^\alpha, \mathcal{CAF}^{\alpha,\beta}, \mathcal{F}^{\alpha,\beta}(o)$);
23 $\quad\quad$ **case** *offer*(β, α)=$\langle o, \langle \theta, \langle S, \mathcal{R}\rangle\rangle\rangle$ **do**
24 $\quad\quad\quad$ call decide-upon-offer($\mathcal{T}^\alpha, \mathcal{CAF}^{\alpha,\beta}$, offer($\beta, \alpha$));

3.6 A Negotiation Example

In the following we run an example of negotiation for illustrating our framework. Figure 1 presents the agents α and β theories (before (a) and after the negotiation (b)) and their associated CAF respectively. Green arguments (resp. attacks) represent certain arguments (resp. attacks), red arguments (resp. attacks) represent uncertain arguments (resp. attacks) and blue arguments (resp. attacks) represent control arguments (resp. attacks). Thus in the current example we have $A_p^\alpha = \{X\}$ and $A_e^\alpha = \{B, E, K\}$ for agent α and $A_p^\beta = \{Y\}$ and $A_e^\beta = \{B, E, D, F\}$ for agent β. The arguments $\{D, F\}$ are ignored by agent α. We have also the common set of offers $O^\alpha = O^\beta = \{o\}$. $\mathcal{F}^\alpha(o) = \{X\}$ and $\mathcal{F}^\beta(o) = \{Y\}$ represent the practical ar-

guments supporting offer o in the agents α and β theories respectively. For their CAF we have $\mathcal{F}(o)^{\alpha,\beta} = \{Y\}$ and $\mathcal{F}(o)^{\beta,\alpha} = \{X\}$ respectively. Regarding the uncertainty, for $\mathcal{CAF}^{\alpha,\beta}$ we have $A^{\alpha,\beta}_{U_e}$=$\{B\}$, $\dashrightarrow_{\alpha,\beta}$=$\{(E,Y)\}$ and for $\mathcal{CAF}^{\beta,\alpha}$ we have $A^{\beta,\alpha}_{U_e}$=$\{E\}$, $\dashrightarrow_{\beta,\alpha}$=$\{(B,X)\}$, $\rightleftarrows_{\beta,\alpha}$=$\{(K,E),(E,K)\}$, $\Rrightarrow_{\beta,\alpha}$=$\{(F,E),(D,B)\}$ and control arguments $A^{\beta}_{c} = \{D,F\}$.

The negotiation starts with agent α as proponent (see Fig. 1 (a)) by invoking algorithm 5. Following line 2 there is a call of algorithm 1. This algorithm computes the next (best) offer (line 1) to propose that is supported by an acceptable argument. In our example there is offer o but the supporting argument X is rejected as it is attacked by arguments B and E that belong into the two stable extensions namely $\{B,K\}$ and $\{B,E\}$. Agent α has no offer to propose to agent β and following line 6 it prepares a $message(\alpha,\beta) = nothing$ and sends it to agent β. Agent β acts now as proponent (see Fig. 1, (a)). By using algorithm 5 (line 13) it checks whether $\mathcal{O}^{\beta} \neq \emptyset$ (line 14) which is the case and calls algorithm 1. This algorithm computes (as previously) the next (best) offer (line 1) that is supported by an acceptable argument. In the current situation we have the offer o which is now supported by the acceptable argument Y as it belongs to the (only) stable extension $\{Y,D,F\}$. Then (line 3) it computes the supporting practical arguments in the uncertain theory of agent α namely $\mathcal{F}(o)^{\beta,\alpha} = \{X\}$ by using its CAF. Then (line 4) there is a call of algorithm 2. This algorithm allows to choose a supporting argument (line 2). In our case there is only one the argument X. Then there is a call (line 3) of algorithm 3. This algorithm allows to check firstly (line 1) whether X is credulously accepted in the uncertain theory of agent α without the use of a control configuration (see *Formula* 1). Argument X is attacked by the uncertain argument E (i.e. see attack (E,X)). That means that there is a completion (or profile) where this argument is present in the theory. Moreover the type of uncertain attack between arguments K and E informs us that an attack is indeed present but the direction is unknown. That means that there are two completions (profiles) (among the three possible ones) where we have $\{(K,E),(E,K)\}$ and $\{(E,K)\}$ as possible attacks. In one of these completions argument E defends itself against the attack from K and in the other it attacks K. Therefore in both cases E will be an acceptable argument and X will be rejected (as there is no defence against this attack). Argument X is also attacked by argument B through the uncertain attack (B,X). That means that there is a completion (profile) where this attack is present in the theory and in that case X will also be rejected as B is an acceptable argument and there is no defence for X against the attack (B,X). Therefore X cannot be accepted without the use of a control configuration. By looking at the real theory (green arguments) of agent α we may observe that the profile with the attacks $\{(K,E),(E,K)\}$ is the right one but agent β ignores this information. Then the algorithm tries to check whether it can find (see *Formula* 2) a control configuration S (line 7). As we may observe such a set exists (see line 9) that can defend X no matter the real profile (i.e. for all the completions) of agent α. More precisely we have $S = \{D,F\}$ and $\mathcal{R}$=$\{(F,E),(D,B)\}$ and an offer(β,α)=$\langle o,\langle X,\langle\{D,F\},\{(F,E),(D,B)\}\rangle\rangle\rangle$ is built. Then, following line 10, a message(β,α)=offer(β,α) is prepared and sent to agent α. Agent α acts as receiver now. By using algorithm 5 (see line 23) it calls algorithm 4 (see line 24). By using algorithm 4 agent α updates its theory and $\mathcal{CAF}$(see lines 3-6), by using $S = \{D,F\}$ and $\mathcal{R}$=$\{(F,E),(D,B)\}$ (see Fig. 1 (b)). Then it checks whether it can accept X (see line 7). As shown in Figure 1 (b), the integration of agent's β control (blue) arguments $\{D,F\}$ (and the associated attacks) in agent's α theory (see green arguments and attacks in Fig. 1 (b)), allows this agent to accept argument X as $\{X,D,F,K\}$ is a stable extension and therefore to accept offer o. Thus, following lines 8-9 it prepares a message(α,β)=accept(o) and sends it to agent β. Agent β acts as receiver by using algorithm 5 (see line 11) and the negotiation ends successfully (line 12) with an agreement on offer o.

4 Experimental evaluation

Our framework has been implemented by using the JADE (http ://jade.tilab.com/) platform and evaluated on negotiations with random argumentation theories.

4.1 Random Theory Generation

The experimental evaluation of the proposed framework is based on a system, implemented in Java, that generates pairs of random negotiation theories and associated CAFs, with different user specified characteristics.

Each negotiation experiment involves a pair of random theories $\mathcal{T}^{\alpha} = \langle A^{\alpha}, \rightarrow_{\alpha}\rangle$ and $\mathcal{T}^{\beta} = \langle A^{\beta}, \rightarrow_{\beta}\rangle$ that share a common part, i.e. there exists $N_{\alpha,\beta} = \langle A^{N_{\alpha,\beta}}, \rightarrow_{N_{\alpha,\beta}}\rangle$, such that $A^{N_{\alpha,\beta}} = A^{\alpha} \cap A^{\beta}$ and $(a,b) \in \rightarrow_{N_{\alpha,\beta}}$ iff $(a,b) \in \rightarrow_{\alpha} \cap \rightarrow_{\beta}$. Moreover, control arguments are only attacked by other control arguments, i.e. $((A^{\alpha} \setminus A^{\alpha}_{c}) \times A^{\alpha}_{c}) \cap \rightarrow_{\alpha} = \emptyset$.
The structure of the generated theories depends on a number of user supplied parameter values that are explained briefly below.

The user inputs the number of epistemic, practical and control arguments of theories $\mathcal{T}^{\alpha}$ and $\mathcal{T}^{\beta}$, as well as their `density`, defined as the ratio of attacks present in the theory to the number of all possible attacks between the arguments of the theory. Moreover, the instance generation system receives as input the number of epistemic, practical and control arguments of the shared part $N_{\alpha,\beta}$.

From theory $\mathcal{T}^{\beta}$, the CAF $\mathcal{CAF}^{\alpha,\beta} = \langle\langle A^{\alpha,\beta}_{F}, \rightarrow_{\alpha,\beta}\rangle, \langle A^{\alpha,\beta}_{U}, \rightleftarrows_{\alpha,\beta} \cup \dashrightarrow_{\alpha,\beta}\rangle, \langle A^{\alpha}_{c}, \Rrightarrow\rangle\rangle$ is built (similarly for $\mathcal{T}^{\alpha}$ and $\mathcal{CAF}^{\beta,\alpha}$), which is the theory that agent α holds about

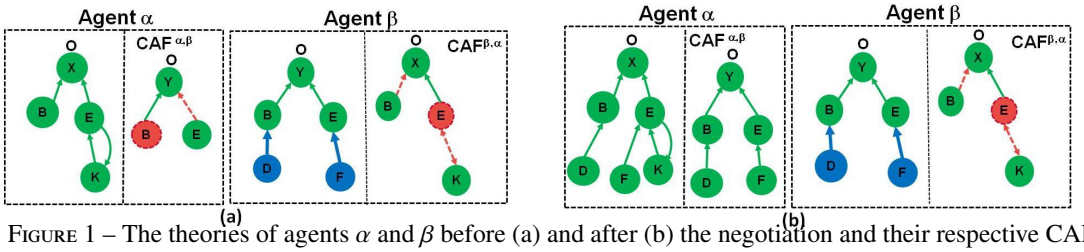


FIGURE 1 – The theories of agents α and β before (a) and after (b) the negotiation and their respective CAFs.

agent β. $\mathcal{CAF}^{\alpha,\beta}$ satisfies the following conditions (a) $A_F^{\alpha,\beta} \cup A_U^{\alpha,\beta} = A^\beta \cup A_p^\alpha$, (b) $A_p^\beta \subseteq A_F^{\alpha,\beta}$.

The attack relation $\rightarrow_{\alpha,\beta} \cup \rightleftarrows_{\alpha,\beta} \cup \dashrightarrow_{\alpha,\beta}$ of $\mathcal{CAF}^{\alpha,\beta}$, is generated so that it satisfies the following conditions :
a) $\rightarrow_{\alpha,\beta} \subseteq \rightarrow_\beta$,
b) $\rightarrow_\beta \cap (A_F^{\alpha,\beta} \times A_F^{\alpha,\beta}) \subseteq \rightarrow_{\alpha,\beta}$,
c) $(\rightleftarrows_{\alpha,\beta} \cup \dashrightarrow_{\alpha,\beta}) \subseteq (\rightarrow_\beta \setminus \rightarrow_{\alpha,\beta})$, and
d) $\rightleftarrows_{\alpha,\beta} \cap \dashrightarrow_{\alpha,\beta} = \emptyset$.

The main consequence of the above requirements is that the attack relation of $\mathcal{CAF}^{\alpha,\beta}$ is a subset of the attack relation of $\mathcal{T}^\beta$. The rationale for this restriction, in this initial experimental evaluation, is to focus on negotiation experiments where agents possess an "accurate" model of their opponent. One way to formalize the model accuracy is via the above relation between individual theories and CAFs. Moreover, it is interesting to study how the framework behaves when this restriction is removed. Indeed, the next section provides initial evidence that the method of this paper can cope with the relaxation of this restriction.

As with the individual agent theories $\mathcal{T}^\alpha$ and $\mathcal{T}^\beta$, the random instance generation software accepts as input a number of parameter values that determine various features of the CAFs of the agents. Most of them concern the uncertainty of an agent profile on its opponent, as captured by the corresponding CAF. The first is parameter `rateUncertArgs` that defines the ratio of uncertain arguments to all (fixed and uncertain) arguments of the theory. That is, `rateUncertArgs`= $|A_U^{\alpha,\beta}|/|A_F^{\alpha,\beta} \cup A_U^{\alpha,\beta}|$ for agent α, and similarly for agent β.

Other parameters of the system include `rateUncertAtt`, that defines the ratio of uncertain attacks over all attacks, as well as `rateUndirAtt` that defines the ratio of undirected attacks to all attacks. That is, `rateUncertAtt`=$|\dashrightarrow_{\alpha,\beta}|/|\rightarrow_{\alpha,\beta} \cup \rightleftarrows_{\alpha,\beta} \cup \dashrightarrow_{\alpha,\beta}|$, and `rateUndirAtt`=$|\rightleftarrows_{\alpha,\beta}|/|\rightarrow_{\alpha,\beta} \cup \rightleftarrows_{\alpha,\beta} \cup \dashrightarrow_{\alpha,\beta}|$.
Moreover, parameter `densContrAtt` defines the ratio of attacks from the control arguments of the agent to the arguments of its opponent that are included in its CAF to all possible such attacks from control arguments. For instance, `densContrAtt`=0.1 for $\mathcal{CAF}^{\alpha,\beta}$, means that 10% of all possible attacks from arguments of A_c^α to arguments in $A_F^{\alpha,\beta} \cup A_U^{\alpha,\beta}$ are included in the particular $\mathcal{CAF}^{\alpha,\beta}$. Finally, the instance generation system receives as input the number of offers, i.e. $|O^\alpha|$ and $|O^\beta|$, as well as the number of practical arguments that support each offer.

4.2 Experimental Results

This section reports on selected results of the experimental evaluation of the framework. As the negotiation theory generation system accepts several parameter values, it is outside the scope of this work to provide exhaustive experimental results for all possible value combinations. Instead, we present results for selected runs that reveal important factors that influence the working of the negotiation algorithm, and highlight its merits and limitations. In all experiments we fix $|A^\alpha| = |A^\beta| = 40$, $|A_p^\alpha| = |A_p^\beta| = 6$, $|O^\alpha| = |O^\beta| = 4$ and $A_c^\alpha \cap A^{N_{\alpha,\beta}} = A_c^\beta \cap A^{N_{\alpha,\beta}} = \emptyset$.

The experimental evaluation is centered around 12 sets of agent theories, and associated CAFs, that differ in the uncertainty of these CAFs and the size of the shared part of agent theories. More specifically, four (4) combinations of parameter values concerning the CAFs are considered, same for both agents. The first combination, abbreviated as `comb1`, is determined by the values `rateUncertArgs`= 0, `rateUndirAtt`= 0, `rateUncertAtt`= 0 which correspond to the case where both agents have complete knowledge of their opponent. Then, `comb2` is defined by the values `rateUncertArgs`= 0.10, `rateUndirAtt`= 0.5, `rateUncertAtt`= 0.5. Moreover, the third combination `comb3` is `rateUncertArgs`= 0.25, `rateUndirAtt`= 0.125, `rateUncertAtt`= 0.125. Finally, the last combination `comb4` is `rateUncertArgs`= 0.50, `rateUndirAtt`= 0.25, `rateUncertAtt`= 0.25 and is the case where the agents have the highest uncertainty about their opponents among all the experiments.
Each of the above set of values for the 3 CAF parameters is combined with one of the three possible values $\{0.25, 0.5, 0.75\}$ for the ratio $|A^{N_{\alpha,\beta}}|/|A^\alpha|$ that capture different degrees of similarity between agent theories.

Each row of Tables 1 and 2 presents the *agreement rate*, (i.e. ratio of the number of negotiations terminated with agreement over the total number of negotiations) of 600 negotiations consisting of 50 randomly generated experiments for each of the 12 parameter values combinations described above. Therefore, each experiment is an amalgamation of negotiation theories of various types as far as the values of the 12 value parameters is concerned. Each row of Table 1 corresponds to an experiment (600 negotia-

tions) where the number of control arguments is shown in the numContrArg column, whereas the value of parameter dens ContrAtt in the corresponding column. The last two columns refer to the agreement rates achieved when the density of the individual theories of the agents participating in the negotiations is fixed to 0.15 (column "Agr 0.15") and 0.2 (column "Agr 0.2") respectively. The first row corresponds to the case where none of the agents has any control arguments.

The first conclusion that can be readily drawn from Table 1 is that the presence of control arguments increases significantly the number of negotiations that terminate with agreement. Indeed, for theories with density 0.15 (column "Agr 0.15"), the agreement rate almost doubles from 0.23 to 0.44 for cases where there are relatively few control arguments and attacks from those arguments, and triples to 0.65 in the experiments with the highest number of control arguments and attacks.

Similar are the results when the density of the individual theories of the participating agents is set to 0.2 (column "Agr 0.2"). Observe that the slight increase of the density leads to a decrease in the rate of agreements in all cases. However, again the presence of control arguments increases the agreement rate from 0.16 to as high as 0.56.

numContrArg	densContrAtt	Agr 0.15	Agr 0.2
0	0	0.23	0.16
3	0.03	0.44	0.39
3	0.05	0.46	0.44
3	0.1	0.57	0.49
3	0.2	0.60	0.52
6	0.03	0.58	0.52
6	0.05	0.59	0.50
6	0.1	0.65	0.56
6	0.2	0.58	0.54

Table 1 – Agreement rate for negotiations with individual theories of density 0.15 and 0.20

Recall that the negotiation experiments are generated so that $A_F^{\alpha,\beta} \cup A_U^{\alpha,\beta} = A^\beta \cup A_p^\alpha$ i.e. agent α CAF about β contains all the arguments of its opponents. In the experiments of Table 2 this assumption is removed by allowing agent β to possess arguments that are not part of the CAF of agent α. The number of these arguments is determined by the value of parameter unknown defined as $|(A^\beta - (A_F^{\alpha,\beta} \cup A_U^{\alpha,\beta}))|/|(A_F^{\alpha,\beta} \cup A_U^{\alpha,\beta})|$. In the experiments of Table 2 this value is set to 0.25 with the effect of a decrease in the agreement rate when compared to the case with no unknown arguments. This decrease was less significant for theories with more control attacks. The experimental evaluation leads to a number of general conclusions. The first is that, not surprisingly, the effectiveness of the approach wrt the rate of agreements depends on a number of parameters including the density of the individual theories, the number of attacks from control arguments, etc. Moreover, other experiments not reported here, have shown that the agreement rate also depends on the size of the shared part $N_{\alpha,\beta}$. In all cases it seems that, for "reasonably good" opponent profiles, the method leads to a significant increase in the number of negotiations that terminate with agreement.

numContrArg	densContrAtt	Agreement
3	0.03	0.32
3	0.05	0.37
3	0.1	0.42
3	0.2	0.43
6	0.03	0.45
6	0.05	0.43
6	0.1	0.55
6	0.2	0.57

Table 2 – Agreement rate for negotiations with individual theories of density 0.15 and unknown= 0.25

5 Related work and Conclusions

In this paper we presented an original argumentation-based negotiation framework that exploits a recent work proposed in [10] on control argumentation frameworks for modeling the uncertainty about the opponent profile and also the acceptance and bidding strategies of the negotiating agents. Compared to previous works proposed in the literature on argumentation-based negotiation (see e.g. [1],[3], [18],[13], [22],[14], [19],[20]) this new framework introduces and combines together a number of original ideas, with most notable a qualitative representation of uncertainty that enables simultaneous consideration of several different profiles, the bidding strategy that allows an agent to use arguments that do not belong to its theory, along with the notion of control arguments that facilitates persuasion and utilizes arguments that defend against all the possible attacks at once, hence minimizing the number of exchanged messages. We consider that our work generalizes the works proposed in [15],[9]. Our work is also different from the work proposed in [23] where the agents have an incomplete theory on the opponent which evolves based on the information contained in the exchanged offers during the negotiation through classical belief revision. The bidding strategy also used in this work is different to ours. Our experimental results have shown that the outcome of an argumentation-based negotiation dialogue depends on different parameters of the argumentation theories of the agents but in all cases the use of control arguments seems to have a positive impact on the number of agreements.

Références

[1] Amgoud, L., Y. Dimopoulos et P. Moraitis: *A unified and general framework for argumentation-based negotiation*. Dans *Proc. AAMAS 2007*, page 158, 2007.

[2] Amgoud, L., Y. Dimopoulos et P. Moraitis: *Making Decisions through Preference-Based Argumentation*. Dans *Proc. KR 2008*, pages 113–123, 2008.

[3] Amgoud, L. et S. Kaci: *On the Study of Negotiation Strategies*. Dans *Proc. Agent Communication II*, pages 150–163, 2006.

[4] Baarslag, T., M. J. C. Hendrikx, K. V. Hindriks et C. M. Jonker: *Learning about the opponent in automated bilateral negotiation : a comprehensive survey of opponent modeling techniques*. Autonomous Agents and Multi-Agent Systems, 30(5) :849–898, 2016.

[5] Baarslag, T., M. J. C. Hendrikx, K. V. Hindriks et C. M. Jonker: *A Survey of Opponent Modeling Techniques in Automated Negotiation*. Dans *Proc. AAMAS 2016*, pages 575–576. ACM, 2016.

[6] Baumann, R. et G. Brewka: *Expanding Argumentation Frameworks : Enforcing and Monotonicity Results*. Dans *Proc. COMMA 2010*, pages 75–86, 2010.

[7] Besnard, P. et S. Doutre: *Checking the acceptability of a set of arguments*. Dans *Proc. NMR 2004*, pages 59–64, 2004.

[8] Black, E. et K. Atkinson: *Choosing persuasive arguments for action*. Dans *Proc. AAMAS 2011*, pages 905–912, 2011.

[9] Bonzon, E., Y. Dimopoulos et P. Moraitis: *Knowing each other in argumentation-based negotiation*. Dans *Proc. AAMAS 2012*, pages 1413–1414, 2012.

[10] Dimopoulos, Y., J.-G. Mailly et P. Moraitis: *Control Argumentation Frameworks*. Dans *Proc. AAAI 2018*, pages 4678–4685, 2018.

[11] Dimopoulos, Y. et P. Moraitis: *Advances in Argumentation-based Negotiation*. Dans *Negotiation and Argumentation in Multi-Agent Systems : Fundamentals, Theories, Systems and Applications*, pages 82–125, 2014.

[12] Dung, P. M.: *On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games*. Artificial Intelligence, 77 :321–357, 1995.

[13] Dung, P. M., P. M. Thang et F. Toni: *Towards argumentation-based contract negotiation*. Dans *Proc. COMMA 2008*, pages 134–146, 2008.

[14] Hadidi, N., Y. Dimopoulos et P. Moraitis: *Argumentative alternating offers*. Dans *Proc. AAMAS 2010*, pages 441–448, 2010.

[15] Hadidi, N., Y. Dimopoulos et P. Moraitis: *Tactics and Concessions for Argumentation-based Negotiation*. Dans *COMMA12*, pages 285–296, 2012.

[16] Hadjinikolis, C., Y. Siantos, S. Modgil, E. Black et P. McBurney: *Opponent Modelling in Persuasion Dialogues*. Dans *Proc. IJCAI 2013*, pages 164–170, 2013.

[17] Hunter, A.: *Modelling the Persuadee in Asymmetric Argumentation Dialogues for Persuasion*. Dans *Proc. IJCAI 2015*, pages 3055–3061, 2015.

[18] Kakas, A. et P. Moraitis: *Adaptive agent negotiation via argumentation*. Dans *Proc. AAMAS 2006*, pages 384–391, 2006.

[19] Marey, O., J. Bentahar, E. K. Asl, M. Mbarki et R. Dssouli: *Agents' Uncertainty in Argumentation-based Negotiation : Classification and Implementation*. Dans *Proc. ANT 2014*, pages 61–68, 2014.

[20] Monteserin, A. et A. Amandi: *A reinforcement learning approach to improve the argument selection effectiveness in argumentation-based negotiation*. Expert Syst. Appl., 40(6) :2182–2188, 2013.

[21] Oren, N. et T. J. Norman: *Arguing Using Opponent Models*. Dans *ArgMAS09*, pages 160–174, 2009.

[22] Parsons, S., C. Sierra et N. R. Jennings: *Agents That Reason and Negotiate by Arguing*. J. Log. Comput., 8(3) :261–292, 1998.

[23] Pilotti, P., A. Casali et C. I. Chesñevar: *A Belief Revision Approach for Argumentation-Based Negotiation Agents*. Applied Mathematics and Computer Science, 25(3) :455–470, 2015.

[24] Rahwan, I., S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons et L. Sonenberg: *Argumentation-based negotiation*. Knowledge Eng. Review, 18(4) :343–375, 2003.

[25] Reimer, S., M. Sauer, P. Marin et B. Becker: *QBF with Soft Variables*. Dans *Proc. AVOCS 2014*, 2014.

[26] Rienstra, T., M. Thimm et N. Oren: *Opponent Models with Uncertainty for Strategic Argumentation*. Dans *Proc. IJCAI 2013*, pages 332–338, 2013.

[27] Riveret, R., A. Rotolo, G. Sartor, H. Prakken et B. Roth: *Success chances in argument games : a probabilistic approach to legal disputes*. Dans Lodder, Arno R. et Laurens Mommers (rédacteurs) : *Proc. JURIX 2007*, tome 165, pages 99–108. IOS Press, 2007.

[28] Zafari, F. et F. N. Mofakham: *POPPONENT : Highly accurate, individually and socially efficient opponent preference model in bilateral multi issue negotiations (Extended Abstract)*. Dans *Proc. IJCAI 2017*, pages 5100–5104, 2017.

Actes JIAF 2019

Adpositional Argumentation (AdArg): A new method for representing linguistic and pragmatic information about argumentative discourse

Federico Gobbo[1] **Jean H.M. Wagemans**[2]

[1] ACLC, University of Amsterdam, The Netherlands / StudiUm, University of Turin, Italy
[2] ACLC, University of Amsterdam, The Netherlands
F.Gobbo@uva.nl J.H.M.Wagemans@uva.nl

Résumé

Cet article décrit et illustre l'utilisation de *Adpositional Argumentation* (AdArg), une méthode formelle nouvelle qui permet à l'analyste du discours argumentatif de représenter des informations linguistiques et pragmatiques de manière très détaillée et quand même flexible. L'article explique d'abord les points de départ théorétiques de la méthode, qui associe le cadre de représentation linguistique *Constructive Adpositional Grammars* (CxAdGrams) et le cadre de classification des arguments *Periodic Table of Arguments* (PTA). Il détaille ensuite les étapes de base de la méthode et les illustre en expliquant comment construire un 'argumentative adpositional tree' (ou 'arg-adtree') d'un exemple concret d'argument.

Abstract

This paper describes and illustrates the use of *Adpositional Argumentation* (AdArg), a new formal method that enables the analyst of argumentative discourse to represent linguistic and pragmatic information in a highly detailed and yet flexible way. It first explains the theoretical starting points of the method, which is a combination of the linguistic representation framework of *Constructive Adpositional Grammars* (CxAdGrams) and the argument classification framework of the *Periodic Table of Arguments* (PTA). It then lays out the basic steps of the method and illustrates them by explaining how to build a so-called 'argumentative adpositional tree' (arg-adtree) of a concrete example of an argument.

1 Introduction

Scholars in the fields of Argumentation Theory and Rhetoric have developed a great many insights concerning the way in which people support their points of view with arguments. These linguistic and pragmatic insights pertain to the nature and constituents of various types of arguments, the structure of different genres of argumentative discourse, as well as the stylistic features of such discourse. In combination with normative standards regarding the validity, reasonableness, and effectiveness of argumentation, this knowledge is used for providing theoretically informed analyses and evaluations of argumentative texts and discussions – see, e.g., [4]

So far, researchers in the fields of Artificial Intelligence and Computational Argumentation have used only a small part of this *cornu copiae* of knowledge. Their computational models of argument and tools for argument mapping, argument mining and computer-aided decision-making usually operate on the abstract level of complete propositions and the interactions between them. This goes, for example, for approaches inspired on Dung's abstract argumentation frameworks, which study sets of atomic arguments and their interrelations. But it also applies to approaches that take Walton's argument schemes as a point of departure, in which an argument scheme is taken to consist of a conclusion and a set of premises – see, e.g., [10].

One of the factors that explains the lack of interaction between the two research areas is that the insights about argumentation and rhetoric developed by scholars from the humanities, although profound and detailed, are mainly informal in nature. Therefore, in order to be used by more formally oriented researchers, these insights need to be translated in such a way that formalizing them does not result in a decrease of the richness of the information that can be represented or in a loss of relevant details.

Adpositional Argumentation (AdArg) is a constructive linguistic approach to argumentation that is aimed at formalizing the insights developed within Argumentation Theory and Rhetoric so as to enable their integration into the models

and frameworks developed within Artificial Intelligence and Computational Argumentation. One of the first achievements of this approach is the development of a method for the formal representation of linguistic and pragmatic information concerning arguments expressed in natural language. The method produces so-called 'argumentative adpositional trees' (or 'arg-adtrees'), which enable the analyst not only to represent sentences on the morphosyntactic level, but also to include information regarding the argumentative function of their constituents. The parsing largely depends on the natural language in use, while the pragmatic information is language-independent to a large extent.

In this paper, we explain the basic characteristics of this representation method and illustrate its use by constructing an arg-adtree of an argument expressed in natural language. For the sake of brevity and clarity, in our example we focus on representing the conclusion and the premise of a single argument on the level of the individual words. As we intend to demonstrate, the method enables the analyst to identify in a very detailed way those linguistic elements that have a pragmatic function in the argumentation. At the same time, it is highly flexible in that the analyst is free to choose which details to show and hide, according to her needs. In particular, the representation of the argument in an arg-adtree may prepare the ground for an assessment of its quality, helping the analyst to identify in a very precise way the 'points of attack' of the argument under scrutiny.

Our method for representing arguments in adtrees results from a combination of two theoretical frameworks. Its basic characteristics are derived from *Constructive Adpositional Grammars* (CxAdGrams), a formal linguistic framework developed by Gobbo and Benini [6] that employs adpositional trees for the purpose of representing natural language. The addition of a layer of pragmatic information to these adtrees is achieved by using the *Periodic Table of Arguments* (PTA), an argument classification framework developed by Wagemans [13, 17] that is especially suitable for formal linguistic and computational approaches to argument. In the following, we explain the theoretical starting points of CxAdGrams (Section 2) and the PTA (Section 3). We then lay out the basic steps of the method and illustrate them by explaining how to construct an arg-adtree of a concrete example of an argument expressed in natural language (Section 4). We conclude the paper with a short reflection on further research and applications (Section 5).

2 The linguistic representation framework

The theoretical framework of *Constructive Adpositional Grammars* (CxAdGrams) is the result of the application of constructive mathematics to the adpositional paradigm in linguistics. We first elucidate this framework by explaining the meaning of the key terms 'constructive' and 'adpositional' in this particular context. Then, we explain the central notion of 'adpositional tree' and illustrate its characteristics by means of an example.

Constructive mathematics is an approach to mathematics that is premised on the idea that regarding the formulas of a theorem, the information content of any statement should be strictly preserved – see Bridges and Richman [2]. There is a tradition of using constructive mathematics to formally represent natural languages, starting from the work of Adjukiewicz [1] and Church [3]. Of the various constructive models in mathematical and computational linguistics developed so far, CxAdGrams specifically are based on topos theory. It thus permits to use Grothendieck's topoi as the mathematical instrument to formalize natural languages and their regularities, both intra a single language and between two or more natural languages in comparison.

The adpositional paradigm in linguistics follows the idea that each pair of linguistic elements can be conveniently described in terms of asymmetrical relations, that is, in such a way that their arrangement cannot be reversed. Thus, given a pair of morphemes, words or expressions, one element 'governs' the other, and consequently, the latter element 'depends' on the former. A very basic example is the phrase *children play*, which has the verb *play* as the governing element (*gov*) and the noun *children* as the dependent element (*dep*). The hierarchical relation between a pair of linguistic terms is conventionally called an 'adposition' and can be pictured in a so-called 'adpositional tree' (or 'adtree').

CxAdGrams, then, is the result of taking a constructive mathematical approach to representing natural language in adpositional trees. More specifically, the formation of an 'adpositional grammar' (or 'adgram'), a set of rules for building adtrees that is admissible within a given natural language, follows certain meta-rules that are described in terms of Grothendieck topoi. As a result, the adtrees produced within CxAdGrams do not only represent natural language expressions in the form of recursive trees but can also be interpreted as formulas – which means that they are suitable for the purpose of natural language processing (for an example, see Figure 4 below). [1]

We now turn to explaining the notion of 'adpositional tree' in more detail. A minimal adtree consists of a pair of linguistic elements and their relation, expressed in terms of their adposition. The governing element (*gov*) is conventionally put on the right leaf at the bottom of the rightmost branch, while the dependent element (*dep*) is put on the left leaf at the bottom of the leftmost branch. The adposition (*adp*), which represents the relation between the governor and the dependent, is depicted as a hook under the bifurcation of the two branches. Under both elements and their adposition, a grammar character (*gc*) is placed. In order to

1. The linguistic and formal rules of CxAdGrams are not discussed here for reasons of conciseness. For a comprehensive presentation of this approach to linguistic analysis, see [6]. The formal model is presented in Appendix B of that work.

illustrate these concepts, we picture in Figure 1 the adtree of the phrase *children play* that was mentioned above.

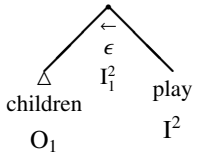


FIGURE 1 – The adtree of the phrase *children play*

In this example, the governor is substantiated by *play*, the dependent by *children*, and the adposition by an epsilon (ϵ), indicating that there is a syntactic relation between the two words. In general, a triangle ($\triangle$) indicates the possibility of recursion, i.e., the fact that another adtree can be appended to each leaf recursively. In this particular case, it indicates the possibility of representing morphological information regarding the word *children*, which is irrelevant for the present purposes but illustrates the fact that the analyst can hide or show details according to her needs.

The theoretical framework of CxAdGrams uses five different grammar characters, which are represented by five vowels (*A*, *E*, *I*, *O*, *U*). Table 1 explains their meaning – adapted from [6, 41].

gc	*name*	*function*	*examples*
A	adjunctive	modifier of *O*	adjectives, articles, determiners
E	circumstantial	modifier of *I*	adverbs, adverbial expressions
I	verbant	valency ruler	verbs, interjections
O	stative	actants	nouns, pronouns, name-entities
U	underspecified	transferer	prepositions, derivational morphemes

TABLE 1 – The meaning of grammar characters in adtrees

The overall shape of an adtree is mainly defined by verbants – typically, verbs (see Table 1). Their grammar characters (*I*) and those of the correlated nominal expressions (*O*) may show additional parameters. In the case of verbants, an apex indicates the verbal valency (*val*), i.e., the number of actants that are potentially involved in the activity described by the verb. [2] The I^2 in our example indicates that the verb *play* is bivalent (*val* = 2), because semantically it implies a player (the first actant) as well as a game or a musical instrument (the second actant). [3]

In the case of the nominal expressions correlated with the verbant, a pedix indicates the number by which they are identified. In our example, *children* acts as the first actant (O_1), while the second actant has remained implicit.

Finally, the information expressed in the complete adtree is summarized by the grammar character I_1^2 under the hook. Here, again, the apex indicates the valency value (*val*) of the verb, while the pedix indicates the number of actants (*act*) present in the sentence. The former is bigger than the latter (*val* = 2 and *act* = 1), which means that the verb is only partially saturated.

We now discussed most of the basic aspects of linguistic adtrees. The meaning of arrows, such as the left arrow ($\leftarrow$) above the epsilon (ϵ) in Figure 1, will be explained in Section 4. Given our current purpose of showing how CxAdGrams can be applied to the pragmatic relation between premise and conclusion in an argument, we first turn to expounding the theoretical framework of the PTA, a formal linguistic approach to argument classification.

3 The argument classification framework

The *Periodic Table of Arguments* (PTA) is a classification of argument that integrates the traditional dialectical accounts of argument schemes and fallacies as well as the rhetorical accounts of logical, ethotic, and pathetic means of persuasion into a systematic and comprehensive whole. [4] The theoretical framework of the table is based on three partial characterizations of argument, namely (1) as first-order or second-order arguments ; (2) as predicate or subject arguments ; and (3) as a specific combination of types of statements. The superposition of these three partial characterizations yields a factorial typology of argument that can be used in order to develop tools for analyzing, evaluating, and producing argumentative discourse.

Every single type of argument described in the PTA consist of exactly one premise and one conclusion, both of which are expressed by means of a statement containing a subject and a predicate. Closely following logical conventions, subjects are indicated with letters *a*, *b*, etc., predicates with letters *X*, *Y*, etc. (predicate $\top$ having the fixed meaning 'true'), and complete propositions with letters *p*, *q*, etc. The identification of the type of argument takes place by following the so-called Argument Type Identification Procedure (ATIP), which is a heuristic device that helps the analyst

2. The concept of valency was introduced by Tesnière [12] within the framework of Structural Syntax. Gobbo and Benini [7] clarify the relation between CxAdGrams and that framework.

3. Please note that if the adposition of the second actant of 'to play' is filled by the preposition 'with', the overall semantics slightly changes. Thus, in CxAdGrams, 'to play' and 'to play with' can be treated as different verbs, if it is convenient. This apparently ad hoc treatment depends on English where a preposition may be used to modify the meaning of the verb, e.g., compare 'to get', 'to get out', 'to get off'. Such decisions lie beyond the scope of the present paper.

4. The present explanation of the PTA draws on [13, 14, 15, 17]. For regularly updated online information on related research projects, analyses of concrete examples, and downloads of relevant papers, see periodic-table-of-arguments.org.

to determine the 'argument form', a notion that comprises the first two partial characteristics mentioned above, as well as the 'argument substance', a notion that covers the third partial characteristic - see [16].

The theoretical framework of the PTA distinguishes between four basic argument forms : first-order predicate arguments, first-order subject arguments, second-order subject arguments, and second-order predicate arguments. In the visual representation of the table, these forms correspond to four different quadrants, which are indicated with Greek letters α, β, γ, and δ respectively. In Table 2, for each quadrant (Q) we list the corresponding argument form and provide a concrete example.

Q	*argument form*	*example*
α	a is X, because a is Y	The suspect (a) was driving fast (X), because he (a) left a long trace of rubber on the road (Y)
β	a is X, because b is X	Cycling on the grass (a) is forbidden (X), because walking on the grass (b) is forbidden (X)
γ	q is $\top$, because r is $\top$	He must have gone to the pub (q), because the interview was cancelled (r)
δ	q is $\top$, because q is Z	We only use 10% of our brain (q), because that (q) was said by Einstein (Z)

TABLE 2 – Argument forms and examples in the PTA

The argument types situated within each of the quadrants are further differentiated on the basis of a determination of the argument substance, i.e., the specific combination of types of statements. For this purpose, the PTA makes use of a tripartite typology consisting of statements of fact (F), statements of value (V), and statements of policy (P). Given these possibilities, the conclusion and premise of the argument substantiate one of the following nine combinations of types of statements : PP, PV, PF, VP, VV, VF, FP, FV, FF. The argument *The government should invest in jobs, because this will lead to economic growth*, for instance, can be characterized as a PF argument since its combines a statement of policy (P) in its conclusion with a statement of fact (F) in its premise.

When taken together, the three partial characterizations constitute a theoretical framework that allows for 2 x 2 x 9 = 36 different argument types. *The suspect was driving fast, because he left a long trace of rubber on the road*, for instance, can be identified as a first-order predicate argument that combines a statement of fact with another statement of fact. The systematic type indicator of this argument is therefore '1 pre FF'.

A final differentiation relates this systematic way of characterizing arguments to the traditional names that can be found in the existing dialectical and rhetorical classifications of argument. Each of the 36 argument types hosts an indefinite number of 'isotopes', which share the same characteristics, but differ as to the linguistic formulation of the connection between premise and conclusion. The key term of this formulation provides the name of the isotope. *The suspect was driving fast, because he left a long trace of rubber on the road*, to use the same example, can be identified as an 'argument from effect', since *leaving a long trace of rubber on the road* is an 'effect' of *driving fast*.

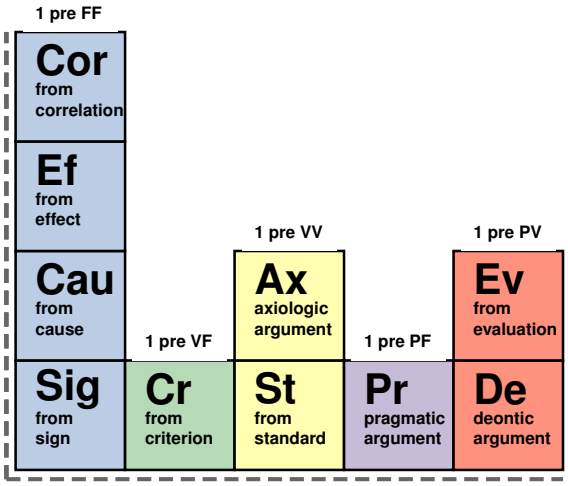


FIGURE 2 – The Alpha Quadrant of the PTA

The various types of argument are visually represented in the table in such a way that it becomes immediately clear what their characteristics are. The placement of an argument type within a particular quadrant provides information about its argument form – see Table 2. Within each of the quadrants, the horizontal placement of the argument types reflects the argument substance as expressed by the specific combination of types of statements – FF, VF, VV, etc. And finally, within every column, the vertical placement depends on the istope, thus reflecting the linguistic formulation of the relation between the premise and the conclusion.

In Figure 2, we showcase the Alpha Quadrant of the current version of the PTA (for the full picture, see its official website periodic-table-of-arguments.org). The argument from effect just mentioned is represented with the symbol 'Ef' and can be found in the leftmost column of this quadrant. It is accompanied by its isotopes, the 'argument from sign', the 'argument from cause', and the 'argument from correlation', which are all first-order predicate arguments based on different relationships between facts.

In the next section we demonstrate how the theoretical framework of the PTA can be used for enriching the linguistic adtrees generated by CxAdGrams with pragmatic information regarding the type of argument.

4 Building argumentative adpositional trees

While CxAdGrams is primarily built for expressing morphology and syntax, the adtrees generated by its theoretical framework are also suitable for expressing pragmatics. [5] In this section, we explain how to combine this formalism with that of the PTA and demonstrate how to insert pragmatic information concerning the characteristics of an argument into a linguistic adtree, thereby transforming it into a 'argumentative adtree' (or 'arg-adtree').

In order to build an arg-adtree, the analyst starts with constructing the two linguistic adtrees that represent the statements that function as the conclusion and the premise of the argument. [6]

Then, step by step, pragmatic information concerning the statements is added to the respective linguistic adtrees. This information includes their argumentative function as a conclusion or a premise. Conclusions are indicated by a sigma (σ), standing for the Greek equivalent $\sigma\upsilon\mu\pi\epsilon\rho\alpha\sigma\mu\alpha$ (sumpérasma), and premises by a pi (π), standing for $\pi\rho o\theta\alpha\sigma\iota\sigma$ (prótasis). It also includes the type of statement (P, V, or F) they substantiate, which is added under the function indicator.

Next, the two adtrees are conjoined, placing the conclusion on the right and the premise on the left. Depending on the order of presentation in the actual discourse, which can be progressive (premise, *therefore* conclusion) or retrogressive (conclusion, *because* premise), the arrow under the topmost hook points to the left or the right.

[7] Under this arrow, the analyst places pragmatic information about the type of argument, which includes the argument form (indicated by a Greek letter representing the corresponding quadrant) and its substance, the combination of types of statements (FF, VF, PF, etc.) Finally, the subjects and predicates of the statements are indicated by the same letters as the ones in use within the theoretical framework of the PTA (a, b, etc. for subjects and X, Y, etc. for predicates).

We will illustrate this process by constructing the arg-adtree of the example of the argument from effect mentioned in the previous section. This argument, *The suspect was driving fast, because he left a long trace of rubber on the road*, has been identified as a first-order predicate argument that supports a statement of fact with another statement of fact ('1 pre FF').

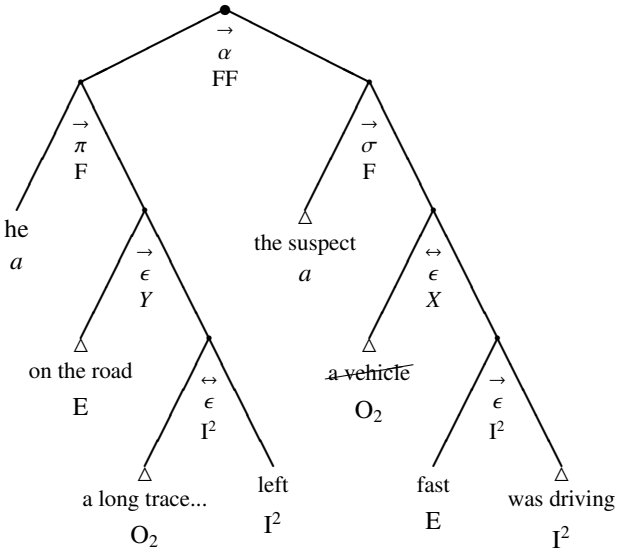


FIGURE 3 – The arg-adtree of the example

As pictured in Figure 3, the linguistic adtrees of the two statements that function as the premise and the conclusion of the argument have been conjoined and infused with pragmatic information so as to create the argumentative adtree. Under the topmost hook, information has been placed about the order of presentiation in the discourse (retrogressive, represented by →) as well as about the type of argument (1 pre FF, abbreviated as α FF). On the subsequent levels, apart from the linguistic information derived from the theoretical framework of CxAdGrams, pragmatic information is given regarding the argumentative function of the statements (conclusion, indicated by σ, and premise, indicated by π) as well as the type of statement (both are statements of fact, indicated by F).

Regarding the placement of the subject and predicate of the statements, the building of an argumentative adtree involves specific transformations of the linguistic adtrees. [8] While the position of the predicate does not change during this transformation process, in the argumentative adtree the subject is emphasized — see Figure 3. In this case, *he*, the first actant (O_1), functions as the subject of the premise (a) and is therefore put in evidence as the topmost left branch of the tree. The remaining linguistic material automatically becomes part of the predicate of the premise (Y).

5. Gobbo and Benini [6, ch. 6] have already provided pragmatic adtrees, representing the relation between illocutionary and locutionary acts as described in [11].

6. For an explanation of how to construct such linguistic adtrees, see [8]. The graphical aspect of an adtree is adapted for human readibility – or machine readability, in the case of linearisation. Such aspect follows some aesthetic and typographical – or coding – conventions, but what matters is how the various pieces are linked together.

7. Regarding the order of presentation of a premise and a conclusion, van Eemeren and Snoeck Henkemans [5, 33] distinguish between a progressive and a retrogressive mode. The former presents first the premise (π) and then the conclusion (σ), connecting them by means of the conjunction *so*, *therefore*, or another one with a similar function. Since there is a multitude of equivalent expressions, in argumentative adtrees of reconstructed arguments the progressive mode is represented only formally, by a left arrow (←). The retrogressive mode starts from the conclusion (σ) and then arrives at the premise (π), connecting them with a conjunction such as *because*, *since*, etc. This mode is represented in argumentative adtrees by a right arrow (→).

8. These transformations are formally justified by the so-called conjugate construction in the formal model of CxAdGrams – see [6, 211].

The reader is invited to note the corresponding tabularized form of this arg-adtree in Figure 4, which is useful for constructing a treebank of arg-adtrees suitable for natural language processing.

$\alpha_{FF}^{\rightarrow}($

$\quad \pi_{F}^{\rightarrow}($

$\quad\quad \text{he}_{a},$

$\quad\quad\quad \epsilon_{X}^{\rightarrow}((\text{on the road})_{E}$

$\quad\quad\quad ,$

$\quad\quad\quad \epsilon_{X}^{\leftrightarrow}((\text{a long trace}\ldots)_{O_2}$

$\quad\quad \text{left}_{I^2}))),$

$\quad \sigma_{F}^{\rightarrow}($

$\quad \ldots$

FIGURE 4 – Tabularization of the arg-adtree

The representation of the argument in an adtree reveals that there is a significant difference between the linguistic and the argumentative analysis : what is peripheral from a linguistic point of view, can be central from an argumentative perspective. In particular, the circumstantial *fast* (E) of the premise (π) is linguistically merely a decoration of the ruling verbal form *was driving* (I^2). In the analysis of the persuasive force of the argument, however, it plays a central role : if the suspect weren't driving *fast* (E), he could have never left a *long* trace (O_2) on the road.

As illustrated by this example, argumentative adtrees are powerful tools that enable the analyst to make explicit where the pragmatic force is placed within the linguistic material. In this way, the representation of the argument prepares the ground for its evaluation, indicating where to find the crucial 'points of attack'.

5 Conclusion

In this paper we demonstrated how the linguistic representation framework of *Constructive Adpositional Grammars* (CxAdGrams) and the argument classification framework of the *Periodic Table of Arguments* (PTA) can be combined so as to represent arguments in natural language. The method we developed for this purpose centers around the notion of 'argumentative adpositional tree' (or 'arg-adtree'). By providing an explanation of how to build an arg-adtree of an argument expressed in natural language, we made it clear how to represent not only the linguistic features of the statements involved, but also how to include pragmatic information about the overall structure of the argument, its type, and the argumentative function of its constituents. The method permits the analyst to operate on the level of the individual words and to freely choose the level of linguistic detail to be shown in the representation.

It is our aim to extend this combined approach, which we have named *Adpositional Argumentation* (AdArg), to the analysis of all the types of argument distinguished in the PTA. We will then apply it to concatenations of arguments, thereby providing a complete analysis of an argumentative text. In fact, since CxAdGrams provide a way to represent punctuation as conjunctions between sentences, they permit to represent a whole text in the terms of a single, comprehensive adtree.

In general, AdArg is aimed at fulfilling the need for a method for representing argumentative discourse that enables a formalization of the rich and detailed insights developed in the fields of Argumentation Theory and Rhetoric. Such a formalization, so we believe, is of use for researchers in the fields of Artificial Intelligence and Computational Argumentation, since it enables the development of tools that automatize to some extent the analysis and evaluation of argumentative discourse.

Regarding the possible application of the method presented in this paper, we think that it could be implemented in a computational model under a form of a tool. Such a tool would assist the analyst in making decisions regarding what linguistic and pragmatic information to include in specific reconstructions of argumentative discourse. Whereas the state-of-the-art in Computational Argumentation has automatized the extraction of complete propositions and their relations, our approach can be used to develop tools for a more fine-grained computer-assisted analysis of argumentative texts. If the linguistic and pragmatic information included in our framework is combined with example-based data extracted from past analyses, it would become possible to partially automatize the whole procedure using Artificial Intelligence techniques.

Acknowledgements

This paper is an amended merger of two earlier papers in which we have explained our method for building argumentative adpositional adtrees for different readerships [8, 9]. The authors thank Marco Benini for his thorough reading of the manuscript, and in particular for checking the parts concerning constructive mathematics.

Références

[1] Adjukiewicz, K.: *Die syntaxische Konnexität*. Studia Philosophica, 1 :1–27, 1935.

[2] Bridges, D. et F. Richman: *Varieties of Constructive Mathematics*. Cambridge University Press, 1987.

[3] Church, A.: *A formulation of the simple theory of types*. Journal of Symbolic Logic, 5:56–58, 1940.

[4] Eemeren, F. H. van, B. J. Garssen, E. C. W. Krabbe, A. F. Snoeck Henkemans, H. B. Verheij et J. H. M. Wagemans: *Handbook of Argumentation Theory*. Springer, 2014.

[5] Eemeren, F. H. van et A. F. Snoeck Henkemans: *Argumentation. Analysis and evaluation*. Routledge, 2e édition, 2016.

[6] Gobbo, F. et M. Benini: *Constructive Adpositional Grammars. Foundations of Constructive Linguistics*. Cambridge Scholars Publishing, 2011.

[7] Gobbo, F. et M. Benini: *Dependency and valency. from structural syntax to constructive adpositional grammars*. Dans *Computational Dependency Theory*, pages 113–135, 2013.

[8] Gobbo, F. et J. H. M. Wagemans: *Building argumentative adpositional trees: Towards a high precision method for reconstructing arguments in natural language*. Dans *Proceedings of the 9th Conference of ISSA*, pages 408–420, 2019.

[9] Gobbo, F. et J. H. M. Wagemans: *A method for reconstructing first-order arguments in natural language*. Dans *Proceedings of AIXIA 2018*, pages 27–23, 2019.

[10] Modgil, S., K. Budzynska et J. Lawrence: *Computational models of argument*. Dans *Proceedings of COMMA 2018*, 2019.

[11] Searle, J. R.: *Making the social world. The structure of human civilization*. Oxford University Press, 2010.

[12] Tesnière, L.: *Éléments de Syntaxe Structurale*. Klincksieck, 1959.

[13] Wagemans, J. H. M.: *Constructing a periodic table of arguments*. Dans *Argumentation, Objectivity, and Bias: Proceedings of the 11th International Conference of OSSA*, 2016.

[14] Wagemans, J. H. M.: *Analogy, similarity, and the periodic table of arguments*. Studies in Logic, Grammar and Rhetoric, 55:63–75, 2018.

[15] Wagemans, J. H. M.: *Assertoric syllogistic and the periodic table of arguments*. Dans *Argumentation and Inference: Proceedings of the 2nd ECA*, 2018.

[16] Wagemans, J. H. M.: *Argument type identification procedure (atip)*. Published online, jan 2019. `periodic-table-of-arguments.org/argument-type-identification-procedure`.

[17] Wagemans, J. H. M.: *Four basic argument forms*. Research in Language, 17:57–69, 2019.

Actes JIAF 2019

Semantic(s) of negative sequential patterns

Thomas Guyet
AGROCAMPUS-OUEST/IRISA – UMR6074
thomas.guyet@irisa.fr

Résumé

Dans le domaine de la fouille de motifs, un motif séquentiel négatif exprime un comportement par une séquence d'événements devant survenir et par des événements qui doivent être absents. Par exemple, le motif $\langle a\ \neg b\ c\rangle$ décrit l'absence d'un événement b entre les occurrences des événements a et c.

Dans cet article, nous mettons la lumière sur l'ambiguité de cette notation et nous identifions huit sémantiques possibles à la relation d'inclusion d'un motif dans une séquence. Ces sémantiques sont illustrées et nous les étudions formellement. Nous proposons ainsi des relations de dominance et d'équivalence entre ces sémantiques, et nous mettons en évidence de nouvelles propriétés d'anti-monotonie. Ces résultats pourraient être utilisés pour développer de nouveaux algorithmes efficaces pour la fouille de motifs séquentiels négatifs fréquents.

Abstract

In the field of pattern mining, a negative sequential pattern specifies a behavior by a sequence of events that must occur and negative events that must be absent. For instance, the pattern $\langle a\ \neg b\ c\rangle$ specifies the absence of event b between occurrences of a and c.

In this article, we shed light on the ambiguity of this notation and we identify eight possible semantics for the containment relation between a pattern and a sequence. These semantics are illustrated and formally studied. We propose dominance and equivalence relations between them and we establish new anti-monotonicity properties. These results may be used to develop new algorithms to extract efficiently frequent negative patterns.

1 Introduction

Pattern mining consists in exploring a set of potential patterns to output all but only the most interesting ones. The set of potential patterns can be seen as a search space and the notion of pattern interestingness can be seen as a set of constraints.

The search space in which lies the potential patterns defines a pattern domain. A large number of pattern domains have been proposed in the pattern mining community. The most studied patterns domains [8] are: itemsets, sequential patterns [12] and graph patterns. But a lot of variants from these patterns domains have been proposed. As we focus our attention on temporal data, we can mention: temporal patterns [10], episodes [11] or chronicles [2, 5].

The notion of interestingness studied in pattern mining often refers to the number of its occurrences in a database. A pattern is said to be interesting if it occurs *frequently* in a dataset of examples (frequent sequential pattern mining task). In practice, a pattern is frequent if its number of occurrences is above a user-defined threshold.

Therefore, counting sequences in which the pattern occurs is of paramount importance. It is strongly related to the *containment relation* which decides whether a pattern occurs in a sequence or not. The support measure denotes the counting of sequences in a dataset $\mathcal{D}$ that *contains* the pattern $\boldsymbol{p}$, denoted $supp_{\mathcal{D}}(\boldsymbol{p})$.

The success of pattern mining techniques comes from an anti-monotonicity property of some support measures [1]. Intuitively, if $\boldsymbol{p}$ is not frequent, no "larger" pattern than $\boldsymbol{p}$ is frequent. The pattern mining trick is to prune the search space as soon as an unfrequent pattern as been found. The "is larger than" relation is a topological structure on the set of patterns. As soon as a support measure is anti-monotonic on this topological structure, the frequent pattern mining trick can be used to efficiently prune the search space. Ideally, this structure is a lattice. In this case, the above strategy is complete and correct.

It is worth noticing that the support measure strongly contributes to the semantic of the interestingness of patterns. In fact, output patterns depend on their supports in the dataset(s) of examples. Different support measures have different outputs. For instance, in sequential pattern mining, counting occurrences considering *gap* constraints

does not result in the same set of patterns. [1]

In this work, we explore the domain of negative sequential patterns. Sequential patterns describe a sequence of items. The pattern occurs in a sequence while its items appear in the same order in a sequence. For instance, a pattern $\langle a\ b\ c\rangle$ is read as "a occurs and then b occurs and finally c occurs". Negative sequential patterns are sequential patterns with the specification of absent events. Intuitively, the syntax of a negative sequential pattern will be the following $\langle a\ \neg b\ c\rangle$. This pattern is read as "a occurs and then c occurs, but b does not occurs in between".

The apparently intuitive notion of absent event appears to be complex. Few approaches explored what seems to be the same pattern domain [3, 4, 6, 9, 13, 14]. They use the similar notation of negation ($\neg$) but this unique syntax is hiding different semantics.

In this article, we bring the light on eight different semantics of negative sequential patterns and we study them formally: we introduce dominance and equivalence relations between these semantics, and we establish anti-monotony results to thwart the idea that negative sequential patterns are not anti-monotonic.

2 Negative sequential patterns

In the sequel, $[n] = \{1, \ldots, n\}$ denotes the set of the first n strictly positive integers. Let $\mathcal{I}$ be the set of items (alphabet). An *itemset* $A = \{a_1\ a_2\ ...\ a_m\} \subseteq \mathcal{I}$ is a set of items. A *sequence* $\boldsymbol{s}$ is a set of sequentially ordered itemsets $\boldsymbol{s} = \langle s_1\ s_2\ ...\ s_n\rangle$: $\forall i, j \in [n],\ i < j$ means that s_i is located before s_j in sequence $\boldsymbol{s}$ which starts by s_1 and finishes by s_n.

Definition 1 (Negative sequential patterns (NSP))**.** *A negative pattern $\boldsymbol{p} = \langle p_1\ \neg q_1\ p_2\ \neg q_2\ \ldots\ p_{n-1}\ \neg q_{n-1}\ p_n\rangle$ is a finite sequence where $p_i \subseteq \mathcal{I} \setminus \emptyset$ for all $i \in [n]$ and $q_i \subseteq \mathcal{I}$ for all $i \in [n-1]$.*

The length *of a NSP, denoted $|\boldsymbol{p}|$ is n, its number of itemsets (negative or positive). $\boldsymbol{p}^+ = \langle p_1\ \ldots\ p_n\rangle$ is so-called the positive part of the NSP.*

We denote by $\mathcal{N}$ the set of negative sequential patterns.

It can be noticed that Definition 1 introduces a syntactic limitation on negative sequential patterns:

— a pattern cannot start neither finish by a negative pattern,
— a pattern cannot have two successive negative itemsets.

Example 1 (Negative sequential pattern)**.** *This example illustrates notations of Definition 1. Let $\mathcal{I} = \{a, b, c, d\}$ and $\boldsymbol{p} = \langle a\ \neg(bc)\ (ad)\ d\ \neg(ab)\ d\rangle$. We have $p_1 = \{a\}$, $p_2 = \{ad\}$, $p_3 = \{d\}$, $p_4 = \{d\}$ and $q_1 = \{bc\}$, $q_2 = \emptyset$, $q_3 = \{ab\}$. The length of $\boldsymbol{p}$ is $|\boldsymbol{p}| = 6$ and $\boldsymbol{p}^+ = \langle a\ (ad)\ d\ d\rangle$.*

3 Semantics of negative sequential patterns

The semantics of negative sequential patterns relies on *negative containment*: a sequence $\boldsymbol{s}$ supports pattern $\boldsymbol{p}$ (or $\boldsymbol{p}$ matches the sequence $\boldsymbol{s}$) iff $\boldsymbol{s}$ contains a sub-sequence $\boldsymbol{s}'$ such that every positive itemset of $\boldsymbol{p}$ is included in some itemset of $\boldsymbol{s}'$ in the same order and for any negative itemset $\neg q_i$ of $\boldsymbol{p}$, q_i is *not included* in any itemset occurring in the sub-sequence of $\boldsymbol{s}'$ located between the occurrence of the positive itemset preceding $\neg q_i$ in $\boldsymbol{p}$ and the occurrence of the positive itemset following $\neg q_i$ in $\boldsymbol{p}$.

Definition 2 (Non inclusion)**.** *We introduce two operators relating two itemsets $P \subseteq \mathcal{I} \setminus \emptyset$ and $I \subseteq \mathcal{I}$:*
— partial non inclusion: $P \not\subseteq I \Leftrightarrow \exists e \in P, e \notin I$
— total non inclusion: $P \not\subseteq I \Leftrightarrow \forall e \in P, e \notin I$
and, by convention, $\emptyset \not\subseteq I$ and $\emptyset \not\subseteq I$ for all $I \subseteq \mathcal{I}$.

In the sequel we will denote the general form of itemset non inclusion by the symbol $\not\subseteq$, meaning either $\not\subseteq$ or $\not\subseteq$.

Intuitively, partial non inclusion considers the itemset as a disjunction of negative constraints, *i.e.* at least one of the item has to be absent, and total non-inclusion consider the itemset as a conjunction of negative constraints: all items have to be absent.

Choosing one non inclusion interpretation or the other has consequences on extracted patterns as well as on pattern search. Let's illustrate this with following sequence dataset:

$$\mathcal{D} = \left\{ \begin{array}{l} \boldsymbol{s}_1 = \langle (bc)\ f\ a\rangle \\ \boldsymbol{s}_2 = \langle (bc)\ (cf)\ a\rangle \\ \boldsymbol{s}_3 = \langle (bc)\ (df)\ a\rangle \\ \boldsymbol{s}_4 = \langle (bc)\ (ef)\ a\rangle \\ \boldsymbol{s}_5 = \langle (bc)\ (cdef)\ a\rangle \end{array} \right\}.$$

Table 1 compares the support of patterns under the two semantics of itemset non-inclusion. Let's consider pattern $\boldsymbol{p}_2$ on sequence $\boldsymbol{s}_2$. Considering that the positive part of $\boldsymbol{p}_2$ is in $\boldsymbol{s}_2$, $\boldsymbol{p}_2$ occurs in the sequence iff $(cd) \not\subseteq (cf)$. In case of total non inclusion, it is false that $(cd) \not\subseteq (cf)$ because of c that occurs in (cf), and thus $\boldsymbol{p}_2$ does not occur in $\boldsymbol{s}_2$. But in case of a partial non inclusion, it is true that $(cd) \not\subseteq (cf)$, because of d that does not occur in (cf), and thus $\boldsymbol{p}_2$ occurs in $\boldsymbol{s}_2$.

Lemma 1. [2] *Let $P, I \subseteq \mathcal{I}$ be two itemsets:*

$$P \not\subseteq I \implies P \not\subseteq I \tag{1}$$

Now, we formulate the notions of sub-sequence, non inclusion and absence by means of the concept of embedding.

1. For instance, a *max*-gap constraint specifies a maximum delay between two successive items of an occurrences.

2. All proofs are provided in Appendix A.

Table 1 – Lists of supported sequences in $\mathcal{D}$ by negative patterns $(\boldsymbol{p}_i)_{i=1..4}$ under the total and partial non inclusion semantics. Each pattern has the shape $\langle a\ \neg q_i\ b\rangle$ where q_i are itemsets such that $q_i \subset q_{i+1}$.

	partial non inclusion $\not\preceq$	total non inclusion $\not\sqsubseteq$
$\boldsymbol{p}_1 = \langle b\ \neg c\ a\rangle$	$\{\boldsymbol{s}_1, \boldsymbol{s}_3, \boldsymbol{s}_4\}$	$\{\boldsymbol{s}_1, \boldsymbol{s}_3, \boldsymbol{s}_4\}$
$\boldsymbol{p}_2 = \langle b\ \neg(cd)\ a\rangle$	$\{\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_4\}$	$\{\boldsymbol{s}_1, \boldsymbol{s}_4\}$
$\boldsymbol{p}_3 = \langle b\ \neg(cde)\ a\rangle$	$\{\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_4\}$	$\{\boldsymbol{s}_1\}$
$\boldsymbol{p}_4 = \langle b\ \neg(cdeg)\ a\rangle$	$\{\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_4, \boldsymbol{s}_5\}$	$\{\boldsymbol{s}_1\}$

Definition 3 (Positive pattern embedding)**.** *Let $\boldsymbol{s} = \langle s_1 \dots s_n\rangle$ be a sequence and $\boldsymbol{p} = \langle p_1 \dots p_m\rangle$ be a (positive) sequential pattern. $\boldsymbol{e} = (e_i)_{i\in[m]} \in [n]^m$ is an* embedding *of pattern $\boldsymbol{p}$ in sequence $\boldsymbol{s}$ iff $\forall i \in [m]$, $p_i \subseteq s_{e_i}$ and $e_i < e_{i+1}$ for all $i \in [m-1]$.*

Definition 4 (Strict and soft embeddings of negative patterns)**.** *Let $\boldsymbol{s} = \langle s_1 \dots s_n\rangle$ be a sequence and $\boldsymbol{p} = \langle p_1\ \neg q_1\ \dots\ \neg q_{m-1}\ p_m\rangle$ be a negative sequential pattern.*

*$\boldsymbol{e} = (e_i)_{i\in[m]} \in [n]^m$ is a **soft-embedding** of pattern $\boldsymbol{p}$ in sequence $\boldsymbol{s}$ iff:*

— *$p_i \subseteq s_{e_i}$, $\forall i \in [m]$*

— *$q_i \not\subseteq s_j$, $\forall j \in [e_i+1, e_{i+1}-1]$ for all $i \in [m-1]$*

*$\boldsymbol{e} = (e_i)_{i\in[m]} \in [n]^m$ is a **strict-embedding** of pattern $\boldsymbol{p}$ in sequence $\boldsymbol{s}$ iff:*

— *$p_i \subseteq s_{e_i}$, $\forall i \in [m]$*

— *$q_i \not\subseteq \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$ for all $i \in [m-1]$*

Intuitively, the constraint of a negative itemset q_i is checked on the sequence's itemsets at positions in interval $[e_i+1, e_{i+1}-1]$, *i.e.* between occurrences of the positive itemset surrounding the negative itemset in the pattern. The soft embedding considers individually each of the sequence's itemsets of $[e_i+1, e_{i+1}-1]$ while strict embedding consider them as a whole.

Notation 1. Soft-*embedding is denoted* $\circ$*-embedding, and* strict-*embedding is denoted* $\bullet$*-embedding.*

Example 2 (Itemset absence semantics)**.** *Let $\boldsymbol{p} = \langle a\ \neg(bc)\ d\rangle$ be a pattern and four sequences:*

Sequence	$\not\sqsubseteq$ $\bullet$	$\not\sqsubseteq$ $\circ$	$\not\preceq$ $\bullet$	$\not\preceq$ $\circ$
$\boldsymbol{s_1} = \langle a\ c\ b\ e\ d\rangle$				✓
$\boldsymbol{s_2} = \langle a\ (bc)\ e\ d\rangle$				
$\boldsymbol{s_3} = \langle a\ b\ e\ d\rangle$			✓	✓
$\boldsymbol{s_4} = \langle a\ e\ d\rangle$	✓	✓	✓	✓

One can notice that each sequence contains a unique occurrence of $\langle a\ d\rangle$, the positive part of pattern $\boldsymbol{p}$. Using soft-embedding and partial non inclusion ($\not\preceq$:=$\not\preceq$), $\boldsymbol{p}$ occurs in $\boldsymbol{s_1}$, $\boldsymbol{s_3}$ and $\boldsymbol{s_4}$ but not in $\boldsymbol{s_2}$. Using strict-embedding and partial non-inclusion, $\boldsymbol{p}$ occurs in sequence $\boldsymbol{s_3}$ and $\boldsymbol{s_4}$. Indeed, items b and c occur between occurrences of a and d in sequences 1 *and* 2*. With total non inclusion ($\not\preceq$:=$\not\sqsubseteq$) and either type of embeddings, the absence of an itemset is satisfied if any of its item is absent. As a consequence, $\boldsymbol{p}$ occurs only in sequence $\boldsymbol{s_4}$.*

Lemma 2. *If $\boldsymbol{e}$ is a $\bullet$-embedding, then $\boldsymbol{e}$ is a $\circ$-embedding, whatever is the itemset non-inclusion ($\not\subseteq$).*

Lemma 3. *$\boldsymbol{e}$ is a $\circ$-embedding iff $\boldsymbol{e}$ is a $\bullet$-embedding when $\not\subseteq$:=$\not\sqsubseteq$.*

Lemma 4. *Let $p = \langle p_1\ \neg q_1\ \dots\ \neg q_{n-1}\ p_n\rangle \in \mathcal{N}$ s.t. $|q_i| \leq 1$ for all $i \in [n-1]$, then $\boldsymbol{e}$ is a $\circ$-embedding iff $\boldsymbol{e}$ is a $\bullet$-embedding.*

Lemma 4 shows that in the simple case of patterns with negative singleton only, strict and soft-embeddings are equivalent.

Lemma 5. *Let $p = \langle p_1\ \neg q_1\ \dots\ \neg q_{n-1}\ p_n\rangle \in \mathcal{N}$, if $\boldsymbol{e}$ is an embedding of pattern $\boldsymbol{p}$ in some sequence $\boldsymbol{s}$, then $\boldsymbol{e}$ is an embedding of the positive sequential pattern $\boldsymbol{p}^+$ in $\boldsymbol{s}$.*

Example 2 illustrates the impact of itemset non-inclusion operator and of embedding type.

Another point that determines the semantics of negative containment concerns the multiple occurrences of some pattern in a sequence: should all or at least one occurrence(s) of the pattern positive part in the sequence satisfy the non inclusion constraints?

Definition 5 (Negative pattern occurrence)**.** *Let $\boldsymbol{s}$ be a sequence, $\boldsymbol{p}$ be a negative sequential pattern, and $\boldsymbol{p}^+$ the positive part of $\boldsymbol{p}$. Let $\not\subseteq \in \{\not\sqsubseteq, \not\preceq\}$ be a itemset non-inclusion operator, and $\bullet \in \{\circ, \bullet\}$ correspond to the embedding strategy ($\circ$: soft-embedding, and $\bullet$: strict-embedding).*

— *Pattern $\boldsymbol{p}$* softly-occurs *in sequence $\boldsymbol{s}$, denoted $\boldsymbol{p} \preceq^{\not\subseteq}_{\bullet} \boldsymbol{s}$, iff there exists at least one embedding of $\boldsymbol{p}$ in $\boldsymbol{s}$.*

— *Pattern $\boldsymbol{p}$* strictly-occurs *in sequence $\boldsymbol{s}$, denoted $\boldsymbol{p} \sqsubseteq^{\not\subseteq}_{\bullet} \boldsymbol{s}$, iff for each embedding $\boldsymbol{e}$ of $\boldsymbol{p}^+$ in $\boldsymbol{s}$, $\boldsymbol{e}$ is also an embedding of $\boldsymbol{p}$ in $\boldsymbol{s}$, and there exists at least one embedding $\boldsymbol{e}$ of $\boldsymbol{p}^+$.*

Definition 5 allows for capturing two semantics for negative sequential patterns depending on the occurrences of the positive part:

— *strict occurrence*: a negative pattern $\boldsymbol{p}$ occurs in a sequence $\boldsymbol{s}$ iff there exists at least one occurrence of the positive part of pattern $\boldsymbol{p}$ in sequence $\boldsymbol{s}$ and **every** such occurrence satisfies the negative constraints,

— *soft occurrence*: a negative pattern $\boldsymbol{p}$ occurs in a sequence $\boldsymbol{s}$ iff there exists at least one occurrence of the positive part of pattern $\boldsymbol{p}$ in sequence $\boldsymbol{s}$ and **at least one** of these occurrences satisfies the negative constraints.

Example 3 (Strict vs soft occurrence semantics)**.** *Let* $\boldsymbol{p} = \langle a\ b\ \neg c\ d\rangle$ *be a pattern,* $\boldsymbol{s_1} = \langle a\ b\ e\ d\rangle$ *and* $\boldsymbol{s_2} = \langle a\,b\,c\,a\,d\,e\,b\,d\rangle$ *be two sequences.* $\boldsymbol{p}^+ = \langle a\,b\,d\rangle$ *occurs once in* $\boldsymbol{s_1}$ *so there is no difference for occurrences under the two semantics. But, it occurs fourth in* $\boldsymbol{s_2}$ *with embeddings* $(1,2,5)$*,* $(1,2,8)$*,* $(1,7,8)$ *and* $(4,7,8)$*. The two first occurrences do not satisfy the negative constraint* $(\neg c)$ *while the two last occurrences do. Under the soft occurrence semantics, pattern* $\boldsymbol{p}$ *occurs in sequence* $\boldsymbol{s_2}$ *whereas it does not under the strict occurrence semantics.*

Lemma 6. *Let* $\boldsymbol{p}$ *be a NSP and* $\boldsymbol{s}$ *a sequence,*

$$\boldsymbol{p} \sqsubseteq^{\not\subseteq}_{\bullet} \boldsymbol{s} \implies \boldsymbol{p} \preceq^{\not\subseteq}_{\bullet} \boldsymbol{s} \tag{2}$$

where $\bullet \in \{\circ, \bullet\}$ *and* $\not\subseteq \in \{\not\preceq, \not\sqsubseteq\}$.

Lemma 7. *Let* $\boldsymbol{p}$ *be a NSP and* $\boldsymbol{s}$ *a sequence,*

$$\boldsymbol{p} \dashv\!\exists^{\not\sqsubseteq}_{\bullet} \boldsymbol{s} \implies \boldsymbol{p} \dashv\!\exists^{\not\preceq}_{\bullet} \boldsymbol{s} \tag{3}$$

where $\dashv\!\exists \in \{\preceq, \sqsubseteq\}$ *and* $\bullet \in \{\circ, \bullet\}$

In this section, we shown that there are several semantics associated to negative patterns. This leads to eight different types of pattern occurrences. We denote Θ the set of considered pattern occurrence operators:

$$\Theta = \left\{ \preceq^{\not\sqsubseteq}_{\circ}, \preceq^{\not\sqsubseteq}_{\bullet}, \preceq^{\not\preceq}_{\circ}, \preceq^{\not\preceq}_{\bullet}, \sqsubseteq^{\not\sqsubseteq}_{\circ}, \sqsubseteq^{\not\sqsubseteq}_{\bullet}, \sqsubseteq^{\not\preceq}_{\circ}, \sqsubseteq^{\not\preceq}_{\bullet} \right\}$$

These operators allows to disambiguate the semantics of negative pattern containment. But, is there no useless distinctions between containment relations? Is there some equivalent containment relations in Θ? The next section answers these questions by introducing the notion of dominance between semantics. Then, we provide some results about the anti-mononicity of these containment relations.

4 Dominance and equivalence between containment relations

Definition 6 (Dominance)**.** *For all* $\theta, \theta' \in \Theta$*,* θ *dominates* θ'*, denoted* $\theta \geqslant \theta'$*, iff* $\boldsymbol{p}\theta\boldsymbol{s} \implies \boldsymbol{p}\theta'\boldsymbol{s}$ *for all* $\boldsymbol{p} \in \mathcal{N}$ *and all sequence* $\boldsymbol{s}$*.*

The idea behind the dominance relation between two containment relations θ and θ' is related to the sequences in which a pattern occurs. By definition, if $\theta \geqslant \theta'$ then for any pattern $\boldsymbol{p} \in \mathcal{N}$, if $\boldsymbol{p}$ occurs in a sequence $\boldsymbol{s}$ according to the θ pattern containment relation, then it also occurs in $\boldsymbol{s}$ according to the θ' pattern containment relation. In the context of pattern mining, such kinds of relation are useful to propose algorithms which could benefit from properties of a dominating containment relation to extract efficiently the patterns according to dominated containment relations.

Notation 2. *We denote by* $\theta \not\geqslant \theta'$ *iff* $\theta \geqslant \theta'$ *is false.*

Lemma 8. *Dominance relation is a pre-order.*

Definition 7 (Equivalent containment relations)**.** *For all* $\theta, \theta' \in \Theta$*,* θ *is equivalent to* θ'*, denoted* $\theta \sim \theta'$ *iff* $\theta \geqslant \theta'$ *and* $\theta' \geqslant \theta$*.*

Lemma 9. $\sim$ *is an equivalence relation on* Θ*.*

Equivalent containment relations have the same semantic. The sets of sequences in which a given pattern occurs are the same and, reciprocally, the sets of negative patterns which occur in a sequence are the same considering these *equivalent* containment relations.

We now study the practical dominance relations we have between the elements of Θ.

Proposition 1. *The following dominances between containment relations hold:*

$$\dashv\!\exists^{\not\subseteq}_{\bullet} \geqslant \dashv\!\exists^{\not\subseteq}_{\circ} \tag{4}$$
$$\dashv\!\exists^{\not\sqsubseteq}_{\circ} \geqslant \dashv\!\exists^{\not\sqsubseteq}_{\bullet} \tag{5}$$
$$\sqsubseteq^{\not\subseteq}_{\bullet} \geqslant \preceq^{\not\subseteq}_{\bullet} \tag{6}$$
$$\dashv\!\exists^{\not\sqsubseteq}_{\bullet} \geqslant \dashv\!\exists^{\not\preceq}_{\bullet} \tag{7}$$

and the following non-dominance statements hold:

$$\dashv\!\exists^{\not\preceq}_{\bullet} \not\geqslant \dashv\!\exists^{\not\sqsubseteq}_{\bullet} \tag{8}$$
$$\sqsubseteq^{\not\subseteq}_{\bullet} \not\geqslant \preceq^{\not\subseteq}_{\bullet} \tag{9}$$
$$\dashv\!\exists^{\not\preceq}_{\bullet} \not\geqslant \dashv\!\exists^{\not\preceq}_{\circ} \tag{10}$$

where $\not\subseteq \in \left\{\not\sqsubseteq, \not\preceq\right\}$*,* $\dashv\!\exists \in \{\preceq, \sqsubseteq\}$ *and* $\bullet \in \{\circ, \bullet\}$*.*

Proposition 1 gathers the results from the previous section. Each line expresses several relationships between pairs of containment relations. Equations 4-7 are dominances deduced from Lemmas 2, 3, 6 and 7. Equations 8-10 states the absence of dominance for which we can exhibit counterexamples. Figure 1 summarizes them. [3]

In addition, many other dominance and non dominance relationships can be deduced from Proposition 1 using the transitivity of dominance (Lemma 8). Nonetheless, in this work, some relationships between pairs of containment relations can not be deduced from Proposition 1. For instance, the relationship between $\sqsubseteq^{\not\preceq}_{\circ}$ and $\preceq^{\not\preceq}_{\bullet}$ is not determined by Proposition 1.

One interesting result of Proposition 1 is that there are two pairs of containment relations, $\left(\sqsubseteq^{\not\sqsubseteq}_{\circ}, \sqsubseteq^{\not\sqsubseteq}_{\bullet}\right)$ and $\left(\sqsubseteq^{\not\preceq}_{\circ}, \sqsubseteq^{\not\preceq}_{\bullet}\right)$, whose elements are equivalent.

3. Assuming that two containment relations are not equals iff they are different, the dominance relation is a pre-order. The property is not central in the following. Figure 1 illustrates clearly that it is not a partial order. Indeed, two pairs of different containment relations are symmetrically dominated.

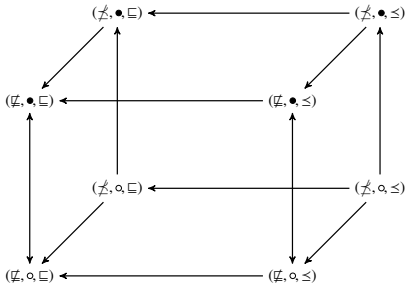


Figure 1 – Summary of the dominance relation between relations of Θ. An arrow shows that the operator at the destination is dominated by the operator at the origin. Unidirectional arrows indicates that the dominance relation holds in one direction but not in the other direction.

It ensues that they are six equivalent classes of containment relations: $\preceq_{\circ}^{\not\sqsubseteq}$, $\preceq_{\bullet}^{\not\sqsubseteq}$, $\preceq_{\circ}^{\not\preceq}$, $\preceq_{\bullet}^{\not\preceq}$, $\left\{\sqsubseteq_{\circ}^{\not\sqsubseteq}, \sqsubseteq_{\bullet}^{\not\sqsubseteq}\right\}$ and $\left\{\sqsubseteq_{\circ}^{\not\preceq}, \sqsubseteq_{\bullet}^{\not\preceq}\right\}$.

We can finally point out that Lemma 4 adds a dominance relation when negative sequential patterns are restricted to have singleton negative itemsets. In this case, the equivalent classes become: $\left\{\preceq_{\circ}^{\not\sqsubseteq}, \preceq_{\bullet}^{\not\sqsubseteq}\right\}$, $\left\{\preceq_{\circ}^{\not\preceq}, \preceq_{\bullet}^{\not\preceq}\right\}$, $\left\{\sqsubseteq_{\circ}^{\not\sqsubseteq}, \sqsubseteq_{\bullet}^{\not\sqsubseteq}\right\}$ and $\left\{\sqsubseteq_{\circ}^{\not\preceq}, \sqsubseteq_{\bullet}^{\not\preceq}\right\}$.

5 Anti-monotonicity

Our question is now to know whether there are containment relations that have more interesting properties. In our original context of mining frequent negative sequential patterns, we investigate the anti-monotonicity properties.

According to Zheng et *al.* [14], "the APriori principle doesn't apply to negative sequential pattern". The "APriori principle" can be understood as the anti-monotonicity property. We will see that assertion is actually only partially true.

The anti-monotonicity makes sense only with a partial order on the set of NSPs. We first introduce different possible partial orders and then we introduce the anti-monotonicity.

For sake of conciseness, the remaining of the section assumes that $\not\subseteq := \not\sqsubseteq$. Thus, we can count on the anti-monotonicity of the non inclusion of itemsets: $q \subseteq q' \implies q' \not\subseteq q$ for all $q, q' \subseteq \mathcal{I}$. The following results can be extended to the case $\not\subseteq := \not\preceq$ by reversing the inclusion relations for negatives in the partial orders.

5.1 Partial orders

Definition 8 introduces several relations between negative sequential patterns that are partial orders (see Proposition 2).

Definition 8 (NSP relations)**.** *Let* $\boldsymbol{p} = \langle p_1\ \neg q_1\ p_2\ \neg q_2\ \cdots\ p_{k-1}\ \neg q_{k-1}\ p_k\rangle$ *and* $\boldsymbol{p'} = \langle p'_1\ \neg q'_1\ p'_2\ \neg q'_2\ \cdots\ p'_{k'-1}\ \neg q'_{k'-1}\ p'_{k'}\rangle$ *be two NSPs.*

By definition, $\boldsymbol{p} \lhd \boldsymbol{p'}$ *iff* $k \leq k'$ *and* $\exists (u_i)_{i\in[k]} \in [k']^k$ *s.t.:*

1. $\forall i \in [k],\ p_i \subseteq p'_{u_i}$
2. $\forall i \in [k-1],\ q_i \subseteq \bigcup_{j\in[u_i, u_{i+1}-1]} q'_j$
3. $k = k' \implies \exists j \in [k],\ p_j \neq p'_j$ *or* $q_j \neq q'_j$
4. $u_i < u_{i+1}$, *for all* $i \in [k-1]$

by definition, $\boldsymbol{p} \lhd \boldsymbol{p'}$ *iff* $k \leq k'$ *s.t.:*

1. $\forall i \in [k],\ p_i \subseteq p'_i$
2. $\forall i \in [k-1],\ q_i \subseteq q'_i$
3. $k = k' \implies p_k \neq p'_k$ *or* $\exists j \in [k-1]$ *s.t.* $q_j \neq q'_j$

and, by definition, $\boldsymbol{p} \lhd \boldsymbol{p'}$ *iff* $k = k'$ *s.t.:*

1. $\forall i \in [k],\ p_i = p'_i$
2. $\forall i \in [k-1],\ q_i \subseteq q'_i$
3. $\exists j \in [k-1]$ *s.t.* $q_j \neq q'_j$

The $\lhd$ relation can be seen as the "classical" inclusion relation between sequential patterns [12]. A NSP $\boldsymbol{p}$ is less specific than $\boldsymbol{p'}$ iff $\boldsymbol{p}^+$ is a subsequence of $\boldsymbol{p'}^+$ and negative constraints are satisfied. The principal difference with $\lhd$ is that $\lhd$ permits to insert new positive itemsets in the middle of the sequence while $\lhd$ permits only insertion of new positive itemsets at the end. [4, 5] Nonetheless, it is still possible to insert items to the positive itemsets. The $\lhd$ does not even permit such differences: each pair of positive itemsets must be equals to have comparable NSP.

Lemma 10. *For all* $\boldsymbol{p}, \boldsymbol{p'} \in \mathcal{N}$,

$$\boldsymbol{p} \lhd \boldsymbol{p'} \implies \boldsymbol{p} \lhd \boldsymbol{p'} \implies \boldsymbol{p} \lhd \boldsymbol{p'} \tag{11}$$

Proposition 2 (Strict partial orders)**.** $\lhd$, $\lhd$ *and* $\lhd$ *are partial orders on* $\mathcal{N}$.

We can notice that the third conditions in Definition 8 enforce the relations to be irreflexive. Removing these conditions enables to define non-strict partial orders.

5.2 Anti-monotonicity

Let us first define the anti-monotonicity property of a containment relation $\theta \in \Theta$ considering a strict partial order $\ltimes \in \{\lhd, \lhd, \lhd\}$.

Definition 9 (Anti-monotonicity on $(\mathcal{N}, \ltimes)$)**.** *Let* $\theta \in \Theta$ *be a containment relation,* θ *is anti-monotonic on* $(\mathcal{N}, \ltimes)$ *iff for all* $\boldsymbol{p}, \boldsymbol{p'} \in \mathcal{N}$ *and all sequence* $\boldsymbol{s}$*:*

$$\boldsymbol{p} \ltimes \boldsymbol{p'} \implies (\boldsymbol{p'}\theta\boldsymbol{s} \implies \boldsymbol{p}\theta\boldsymbol{s})$$

4. In sequential pattern mining, it is called a *backward*-extension of the patterns.

5. We remind that, by Definition 1, $p_i \neq \emptyset$ and that we never have two successive negative itemsets in a NSP.

First of all, we provide an example showing that none of the containment relation is monotonic on $(\mathcal{N}, \lhd)$. Let $\boldsymbol{p} = \langle b\ \neg c\ a\rangle$, $\boldsymbol{p}' = \langle b\ \neg c\ d\ a\rangle$ and $\boldsymbol{s} = \langle b\ e\ d\ c\ a\rangle$. Then, we have $\boldsymbol{p} \lhd \boldsymbol{p}'$.[6] Nonetheless, for each $\theta \in \Theta$, $\boldsymbol{p}'\theta\boldsymbol{s}$ but it is false that $\boldsymbol{p}\theta\boldsymbol{s}$. In fact, the presence of the item d in the sequence changes the scope for checking the absence of c.

This example illustrates the case of Zheng et *al.* [14] to argue for the absence of anti-monotonic property for negative patterns. But, using partial orders that prevent from changing the scope for absent items enables to exhibit anti-monotonicity properties.

Proposition 3. *$\preceq^{\not\sqsubseteq}_{\bullet}$ is anti-monotonic on $(\mathcal{N}, \lhd)$, where $\bullet \in \{\circ, \bullet\}$.*

Proposition 3 shows that using the $\lhd$ order leads to have anti-monotonicity only for containment with soft-occurrence, but not with strict-occurrence. Let us give a counterexample illustrating the problem with strict-occurrence. Let $\boldsymbol{p} = \langle a\ \neg b\ c\rangle$, $\boldsymbol{p}' = \langle a\ \neg b\ c\ d\rangle$ and $\boldsymbol{s} = \langle a\ c\ d\ a\ b\ c\rangle$. Then, we have $\boldsymbol{p} \lhd \boldsymbol{p}'$.[7] Nonetheless, $\boldsymbol{p}' \sqsubseteq^{\not\sqsubseteq}_{\bullet} \boldsymbol{s}$ holds but it is false that $\boldsymbol{p} \sqsubseteq^{\not\sqsubseteq}_{\bullet} \boldsymbol{s}$. In fact, without the presence of the item d in the pattern, there are three possible embeddings of $\boldsymbol{p}$ in $\boldsymbol{s}$. Considering $\sqsubseteq^{\not\sqsubseteq}_{\bullet}$ each embedding must satisfy the negation of b, which is not the case, while it is sufficient to have only one embedding satisfying negations for $\preceq^{\not\sqsubseteq}_{\bullet}$.

The previous example illustrates the problem while extending the pattern with additional itemsets. The same issue is encountered with the following example considering same length patterns but with an extended itemset. Let $\boldsymbol{p} = \langle a\ \neg b\ c\rangle$, $\boldsymbol{p}' = \langle a\ \neg b\ (cd)\rangle$ and $\boldsymbol{s} = \langle a\ (cd)\ a\ b\ c\rangle$. Then, we have $\boldsymbol{p} \lhd \boldsymbol{p}'$. Nonetheless, $\boldsymbol{p}' \sqsubseteq^{\not\sqsubseteq}_{\bullet} \boldsymbol{s}$ holds but it is false that $\boldsymbol{p} \sqsubseteq^{\not\sqsubseteq}_{\bullet} \boldsymbol{s}$.

Proposition 4. *$\dashv\!\exists^{\not\sqsubseteq}_{\bullet}$ is anti-monotonic on $(\mathcal{N}, \lhd)$, where $\dashv\!\exists \in \{\preceq, \sqsubseteq\}$ and $\bullet \in \{\circ, \bullet\}$.*

We remind that this section presented the case of total non-inclusion ($\not\sqsubseteq$:=$\not\sqsubseteq$) but similar results can be obtained with partial non-inclusion.

6 Come back to pattern mining

The definitions of pattern support, frequent pattern and pattern mining derive naturally from the notion of occurrence of a negative sequential pattern, no matter the choices for embedding (soft or strict), non inclusion (partial or total) and occurrences (soft or strict). However, these choices concerning the semantics of NSPs impact directly the number of frequent patterns (under the same minimal threshold constraint) and further the computation time. The stronger the negative constraints, the lesser the number of sequences containing a pattern, and the lesser the number of frequent patterns.

Definition 10 (Pattern supports)**.** *Let $\mathcal{D} = \{\boldsymbol{s}_i\}_{i\in[n]}$ be a set of n sequences and $\boldsymbol{p}$ be a NSP. The support of $\boldsymbol{p}$ in $\mathcal{D}$, denoted $supp^{\mathcal{D}}_{\theta}(p)$ is the number of sequences of $\mathcal{D}$ in which $\boldsymbol{p}$ occurs according to the $\theta \in \Theta$ containment relation.*

Notation 3. *When there is no ambiguity on the dataset of sequences, $supp^{\mathcal{D}}_{\theta}(p)$ is denoted $supp_{\theta}(p)$. And, for sake of readability, the θ operators are represented as triplets.*

It is clear that if a containment relation θ is dominated by another containment relation θ', then the Proposition 1 implies that the support of the pattern evaluated with θ is lower than the support of the pattern evaluated with θ'. Thus, we have the following proposition.

Proposition 5. *For all pattern $\boldsymbol{p} \in \mathcal{N}$:*

$$supp_{\not\preceq,\bullet,\dashv\!\exists}(\boldsymbol{p}) \leq supp_{\not\preceq,\circ,\dashv\!\exists}(\boldsymbol{p}) \tag{12}$$

$$supp_{\not\sqsubseteq,\circ,\dashv\!\exists}(\boldsymbol{p}) \leq supp_{\not\sqsubseteq,\bullet,\dashv\!\exists}(\boldsymbol{p}) \tag{13}$$

$$supp_{\not\preceq,\bullet,\sqsubseteq}(\boldsymbol{p}) \leq supp_{\not\preceq,\bullet,\preceq}(\boldsymbol{p}) \tag{14}$$

$$supp_{\not\sqsubseteq,\bullet,\dashv\!\exists}(\boldsymbol{p}) \leq supp_{\not\preccurlyeq,\bullet,\dashv\!\exists}(\boldsymbol{p}) \tag{15}$$

In addition, we can also deduce anti-monotonicity properties for support measures from the Propositions 3 and 4.

Proposition 6. *For all pairs of NSPs $\boldsymbol{p}, \boldsymbol{p}' \in \mathcal{N}$:*

$$\boldsymbol{p} \lhd \boldsymbol{p}' \implies supp_{\not\sqsubseteq,\bullet,\preceq}(\boldsymbol{p}') \leq supp_{\not\sqsubseteq,\bullet,\preceq}(\boldsymbol{p}) \tag{16}$$

$$\boldsymbol{p} \lhd\!\lhd \boldsymbol{p}' \implies supp_{\not\sqsubseteq,\bullet,\dashv\!\exists}(\boldsymbol{p}') \leq supp_{\not\sqsubseteq,\bullet,\dashv\!\exists}(\boldsymbol{p}) \tag{17}$$

Then, there is two ways to use these results to implement efficient frequent NSP mining algorithms. On the one hand, the results from Proposition 6 can be directly used to implement algorithms with efficient and correct strategies to prune the search space.[8] For $\preceq^{\not\sqsubseteq}_{\bullet}$ containment relation, Equation 16 fully exploits the $\lhd$ partial order to early prune a priori unfrequent patterns. For $\sqsubseteq^{\not\sqsubseteq}_{\bullet}$ containment relation, the $\lhd\!\lhd$ partial order must be used to ensure the correctness of the algorithm (Equation 17). Unfortunalety, this partial order is less interesting than $\lhd$ because it gives a priori information on less patterns than $\lhd$ does. On the other hand, the support evaluated with $\preceq^{\not\sqsubseteq}_{\bullet}$ is an upper bound for the support of $\sqsubseteq^{\not\sqsubseteq}_{\bullet}$ (Equation 14). Thus, it is possible also to prune patterns accessible with $\lhd$ partial order without losing the correctness of the pruning strategy.

7 Conclusion and perspectives

In this article, we explored the semantics of negation in sequential patterns. We gave eight possible semantics

6. In this case, we do not have $\boldsymbol{p} \lhd \boldsymbol{p}'$ nor $\boldsymbol{p} \lhd\!\lhd \boldsymbol{p}'$
7. In this case, we also have $\boldsymbol{p} \lhd\!\lhd \boldsymbol{p}'$ but not $\boldsymbol{p} \lhd \boldsymbol{p}'$
8. The completeness of the algorithms requires to study how to traverse the search space. It is out of the scope of this article.

where the state of the art in sequential pattern mining did not notice these differences. We investigated the formal properties of these semantics: their respective relations (dominance and equivalence) and anti-monotonicity properties. These results may be used to develop new efficient algorithms to extract negative sequential patterns.

It is worth noticing that no semantics is "more" correct or relevant than another one. It depends on the information to be captured. Our objective is to give the opportunity to make an informed choice. Even if, in the context of pattern mining, the choice is constrained by computational considerations. With the three proposed partial orders, we have seen that interesting anti-monotonicity holds only for some semantics. Thus, dominance relations may be used to propose alternative search heuristics.

Hence, the first perspective of this work is to develop negative sequential pattern mining algorithms. This has been done for containment relations $\prec^{\not\sqsubseteq}_{\bullet}$ [7]. [9] In this algorithm, we also introduced *maxgap* and *maxspan* constrains negative sequential patterns. The anti-mononicity properties still hold with these constraints. A step further, the framework presented in this article would make possible to propose a complete and correct algorithm to mine closed NSP.

Our second perspective is to provide a complete results on the dominance between containment relations. There are still few cases that are undetermined to achieve this objective. Further studies aim at having results on the quotient set $\Theta/\sim$.

Our third perspective is to propose new intuitive syntax(es) of containment relations. The main problem was the semantic overload of the negation symbol ($\neg$) in the literature of negative sequential patterns. We are currently working on some proposals and plan to evaluate them. To be intuitive, these proposals are based on the equivalent classes we highlight in this work.

References

[1] Agrawal, Rakesh et Ramakrishnan Srikant: *Fast Algorithms for Mining Association Rules in Large Databases*. Dans *Proc. of the Int. Conf. on Very Large Data Bases (VLDB)*, pages 487–499, 1994.

[2] Besnard, Philippe et Thomas Guyet: *Chronicles*. à paraître, 2019.

[3] Cao, Longbing, Xiangjun Dong et Zhigang Zheng: *e-NSP: Efficient negative sequential pattern mining*. Artificial Intelligence, 235:156–182, 2016.

[4] Chen, Yen-Liang, Mei-Ching Chiang et Ming-Tat Ko: *Discovering time-interval sequential patterns in sequence databases*. Expert System with Applications, 25(3):343–354, 2003.

[5] Dauxais, Yann, Thomas Guyet, David Gross-Amblard et André Happe: *Discriminant chronicles mining - application to care pathways analytics*. Dans *Proc. of Conf. on Artificial Intelligence in Medicine (AIME)*, pages 234–244, 2017.

[6] Gong, Yongshun, Tiantian Xu, Xiangjun Dong et Guohua Lv: *e-NSPFI: Efficient mining negative sequential pattern from both frequent and infrequent positive sequential patterns*. Int. Journal of Pattern Recognition and Artificial Intelligence, 31(02):1750002, 2017.

[7] Guyet, Thomas et René Quiniou: *NegPSpan: efficient extraction of negative sequential patterns with embedding constraints*. CoRR, abs/1804.01256, 2018.

[8] Han, Jiawei, Micheline Kamber et Jian Pei: *Advanced pattern mining*. Dans *Data Mining (Third Edition)*, pages 279–325. Morgan Kaufmann, 2012.

[9] Hsueh, Sue Chen, Ming Yen Lin et Chien Liang Chen: *Mining negative sequential patterns for e-commerce recommendations*. Dans *Proc. of Asia-Pacific Services Computing Conference*, pages 1213–1218, 2008.

[10] Laxman, Srivatsan et P. S. Sastry: *A survey of temporal data mining*. Sadhana, 31(2):173–198, 2006.

[11] Mannila, Heikki, Hannu Toivonen et A. Inkeri Verkamo: *Discovery of frequent episodes in event sequences*. Data Mining and Knowledge Discovery, 1(3):259–289, 1997.

[12] Mooney, Carl H. et John F. Roddick: *Sequential pattern mining – approaches and algorithms*. ACM Computing Survey, 45(2):1–39, 2013.

[13] Xu, Tiantian, Xiangjun Dong, Jianliang Xu et Yongshun Gong: *E-msNSP: Efficient negative sequential patterns mining based on multiple minimum supports*. Int. Journal of Pattern Recognition and Artificial Intelligence, 31(02):1750003, 2017.

[14] Zheng, Zhigang, Yanchang Zhao, Ziye Zuo et Longbing Cao: *Negative-GSP: An efficient method for mining negative sequential patterns*. Dans *Proc. of the Australasian Data Mining Conference*, pages 63–67, 2009.

A Proofs

Proof of Lemma 1. Let $P, I \subseteq \mathcal{I}$ s.t. $P \not\sqsubseteq I$. If $P = \emptyset$, by definition, $P \not\preceq I$. Otherwise, because P is not empty, then there exists $e \in P$ s.t. $e \notin I$, *i.e.* $P \not\preceq I$. □

9. A demo of such an algorithm is accessible online: `http://people.irisa.fr/Thomas.Guyet/negativepatterns/evalnegpat.php`.

Proof of Lemma 2. Let $\boldsymbol{e} = (e_i)_{i\in[m]} \in [n]^m$ be a $\bullet$-embedding of a NSP $\boldsymbol{p} = \langle p_1\ \neg q_1\ \ldots\ \neg q_{m-1}\ p_m\rangle$ in a sequence $\boldsymbol{s} = \langle s_1\ \ldots\ s_n\rangle$. For all positive itemsets p_i, the definition of $\bullet$-embedding matches the one for $\circ$-embedding. For a negative itemset q_i, let us start with $\not\subseteq$:=$\not\subseteq$. By definition 4, $q_i \not\subseteq \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$, and by Definition 2, $\exists\alpha \in q_i,\ \alpha \notin \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$. And then, $\exists\alpha \in q_i,\ \forall j \in [e_i+1, e_{i+1}-1],\ \alpha \notin s_j$. That is $\forall j \in [e_i+1, e_{i+1}-1],\ \exists\alpha \in q_i,\ \alpha \notin s_j$. This shows $\forall j \in [e_i+1, e_{i+1}-1],\ q_i \not\subseteq s_j$ ($\circ$-embedding definition). It remains $\not\sqsubseteq$:=$\not\subseteq$. By definition 4, $q_i \not\sqsubseteq \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$, and by Definition 2, $\forall\alpha \in p_i,\ \alpha \notin \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$. And then, $\forall\alpha \in q_i,\ \forall j \in [e_i+1, e_{i+1}-1],\ \alpha \notin s_j$. That is $\forall j \in [e_i+1, e_{i+1}-1],\ \forall\alpha \in q_i,\ \alpha \notin s_j$. This shows $\forall j \in [e_i+1, e_{i+1}-1],\ q_i \not\subseteq s_j$. □

Proof of Lemma 3. Let $\boldsymbol{s} = \langle s_1\ \ldots\ s_n\rangle$ be a sequence and $\boldsymbol{p} = \langle p_1\ \neg q_1\ \ldots\ \neg q_{m-1}\ p_m\rangle$ be a negative sequential pattern. Lemma 2 shows that $\bullet$-embedding implies $\circ$-embedding. It remains the implication to the left. Let $\boldsymbol{e} = (e_i)_{i\in[m]} \in [n]^m$ be a $\circ$-embedding of pattern $\boldsymbol{p}$ in sequence $\boldsymbol{s}$. Then, the definition matches the one for $\bullet$-embedding for positives, p_i. For negatives, q_i, then $\forall j \in [e_i+1, e_{i+1}-1],\ q_i \not\sqsubseteq s_j$, *i.e.* $\forall j \in [e_i+1, e_{i+1}-1],\ \forall\alpha \in q_i,\ \alpha \notin s_j$ and then $\forall\alpha \in q_i,\ \forall j \in [e_i+1, e_{i+1}-1],\ \alpha \notin s_j$. It thus implies that $\forall\alpha \in p_i,\ \alpha \notin \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$, *i.e.* by definition, $q_i \not\sqsubseteq \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$. □

Proof of Lemma 4. Let $\boldsymbol{s} = \langle s_1\ \ldots\ s_n\rangle$ be a sequence and $\boldsymbol{p} = \langle p_1\ \neg q_1\ \ldots\ \neg q_{m-1}\ p_m\rangle$ be a NSP s.t. $\forall i,\ |q_i| \leq\leq 1$. Let $\boldsymbol{e} = (e_i)_{i\in[m]} \in [n]^m$ be a $\circ$-embedding of $\boldsymbol{p}$ in $\boldsymbol{s}$ then, by definition, 1) $p_i \subseteq s_{e_i}$ for all $i \in [m]$ and 2) $q_i \not\subset s_j$ for all $j \in [e_i+1, e_{i+1}-1]$. In case $|q_i| = 0$, there is no constraint. In case $|q_i| = 1$, and 2) becomes $q_i \notin s_j$ for all $j \in [e_i+1, e_{i+1}-1]$ whatever $\not\subseteq \in \{\not\subset, \not\sqsubset\}$. Hence, $q_i \notin \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$ *i.e.* $q_i \not\subseteq \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$ (no matter $\not\subset$ or $\not\sqsubset$). As a consequence $\boldsymbol{e}$ is a $\bullet$-embedding of $\boldsymbol{p}$. □

Proof of Lemma 5. Let $\boldsymbol{s} = \langle s_1\ \ldots\ s_n\rangle$ be a sequence and $\boldsymbol{p} = \langle p_1\ \neg q_1\ \ldots\ \neg q_{m-1}\ p_m\rangle \in \mathcal{N}$ be a pattern. By definition 4, if $\boldsymbol{e} = (e_i)_{i\in[m]} \in [n]^m$ is an embedding of pattern $\boldsymbol{p}$ in sequence $\boldsymbol{s}$ then $\forall i \in [l]$: $p_i \subseteq s_{e_i}$ because p_i is positive. According to definition 3, $\boldsymbol{e}$ is an embedding of the positive pattern $\boldsymbol{p}^+$. □

Proof of Lemma 6. Let $\boldsymbol{s} = \langle s_1\ \ldots\ s_n\rangle$ be a sequence and $\boldsymbol{p} = \langle p_1\ \neg q_1\ \ldots\ \neg q_{m-1}\ p_m\rangle \in \mathcal{N}$ be a pattern s.t. $p \sqsubseteq^{\not\subseteq}_{\bullet} s$. Then, there exists $\boldsymbol{e}$ an embedding of $\boldsymbol{p}^+$ in $\boldsymbol{s}$ and, by definition, it is also an embedding of $\boldsymbol{p}$ in $\boldsymbol{s}$. This means that $p \preceq^{\not\subseteq}_{\bullet} s$. □

Proof of Lemma 7. Let $\boldsymbol{s} = \langle s_1\ \ldots\ s_n\rangle$ be a sequence and $\boldsymbol{p} = \langle p_1\ \neg q_1\ \ldots\ \neg q_{m-1}\ p_m\rangle \in \mathcal{N}$ be a pattern.

We start by considering relations between semantics at the embedding level, and then we will conclude at the pattern level.

Let's first assume that $\bullet := \bullet$. Then, for all $(\bullet, \not\sqsubseteq)$-embedding $e = (e_i)_{i\in[m]}$ of pattern $\boldsymbol{p}$ in sequence $\boldsymbol{s}$. Hence, $\forall i \in [m]$, $p_i \subseteq s_{e_i}$ and $\forall i \in [m-1], \forall j \in [e_i+1, e_{i+1}-1]$, $q_i \not\sqsubseteq s_j$. According to eq. 1, we have $q_i \not\preceq s_j$. It comes that e is a $(\bullet, \preceq)$-embedding.

Let's now assume that $\bullet := \circ$. Then, let $e = (e_i)_{i\in[m]}$ be a $(\circ, \not\sqsubseteq)$-embedding of pattern $\boldsymbol{p}$ in sequence $\boldsymbol{s}$. Hence, $\forall i \in [m]$, $p_i \subseteq s_{e_i}$ and $\forall i \in [m-1]$, $q_i \not\sqsubseteq \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$. In addition $\bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j \subseteq \mathcal{I}$, then, $\forall i \in [m-1]$, $q_i \not\preceq \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$ according to eq. 1. It comes that e is an $(\circ, \preceq)$-embedding.

Let's come back at the pattern level. if $p \dashv\!\!\exists^{\not\sqsubseteq}_{\bullet} s$, in the two cases ($\dashv\!\!\exists \in \{\preceq, \sqsubseteq\}$). In the first case the existing $(\bullet, \not\sqsubseteq)$-embedding is a $(\bullet, \not\preceq)$-embedding, and in the second case, all $(\bullet, \not\sqsubseteq)$-embeddings are $(\bullet, \not\preceq)$-embeddings. Therefore, we have that $p \dashv\!\!\exists^{\not\preceq}_{\bullet} s$. □

Proof of Lemma 8. A pre-order is a reflexive, transitive binary relation. The reflexivity of the relation comes with Definition 6. Let $\theta, \theta', \theta'' \in \Theta$ be three dominance relations s.t. $\theta \geqslant \theta'$ and $\theta' \geqslant \theta''$. Then, for all $\boldsymbol{p} \in \mathcal{N}$ and sequence $\boldsymbol{s}$: $\boldsymbol{p}\theta\boldsymbol{s} \implies \boldsymbol{p}\theta'\boldsymbol{s}$ and $\boldsymbol{p}\theta'\boldsymbol{s} \implies \boldsymbol{p}\theta''\boldsymbol{s}$. Hence, we have, $\boldsymbol{p}\theta\boldsymbol{s} \implies \boldsymbol{p}\theta''\boldsymbol{s}$, *i.e.* $\theta \geqslant \theta''$. □

Proof of Lemma 9. Let $\theta, \theta' \in \Theta$, by reflexivity of $\geqslant$ we have that $\sim$ is reflexive. By definition ($\theta \geqslant \theta' \wedge \theta' \geqslant \theta$), $\sim$ is symmetric. And , $\sim$ is also transitive. Let $\theta, \theta', \theta'' \in \Theta$ be three dominance relation s.t. $\theta \sim \theta'$ and $\theta' \sim \theta''$ then, $\theta \geqslant \theta'$, $\theta' \geqslant \theta''$, $\theta' \geqslant \theta$ and $\theta'' \geqslant \theta'$. Hence, by transitivity of $\geqslant$, $\theta \geqslant \theta''$ and $\theta'' \geqslant \theta$, $\theta \sim \theta''$. □

Proof of Proposition 1. Let $\boldsymbol{p} \in \mathcal{N}$ and $\boldsymbol{s}$ a sequence.

According to Lemma 6, $\boldsymbol{p} \sqsubseteq^{\not\subseteq}_{\bullet} \boldsymbol{s} \implies \boldsymbol{p} \preceq^{\not\subseteq}_{\bullet} \boldsymbol{s}$. Thus we obtain Equality 6 by Definition 6. is immediately deduced from .

According to Lemma 7, $\boldsymbol{p} \dashv\!\!\exists^{\not\sqsubseteq}_{\bullet} \boldsymbol{s} \implies \boldsymbol{p} \dashv\!\!\exists^{\not\preceq}_{\bullet} \boldsymbol{s}$. Thus we obtain Equality 7 by Definition 6.

According to Lemma 2, a $\bullet$-embedding is a $\circ$-embedding whatever the itemset non-inclusion operator. Then, we can conclude that $\boldsymbol{p} \dashv\!\!\exists^{\not\subseteq}_{\bullet} \boldsymbol{s} \implies \boldsymbol{p} \dashv\!\!\exists^{\not\subseteq}_{\circ} \boldsymbol{s}$ (Equality 4).

In addition, Lemma 3 shows that a $\circ$-embedding is $\bullet$-embedding in case of total itemset non-inclusion. Then, we can conclude that $\boldsymbol{p} \dashv\!\!\exists^{\not\sqsubseteq}_{\circ} \boldsymbol{s} \implies \boldsymbol{p} \dashv\!\!\exists^{\not\sqsubseteq}_{\bullet} \boldsymbol{s}$ (Equality 5).

Let now gives some counterexamples for known non-dominance relationships. A counterexample for Equation 8 is $p = \langle a\ \neg(bc)\ d\rangle$ and $s = \langle a\ b\ d\rangle$. We have that $p \dashv\!\!\exists^{\not\preceq}_{\bullet} s$ but $p \dashv\!\!\exists^{\not\sqsubseteq}_{\bullet} s$ is false.
Similarly, a counterexample for Equation 9 is $p = \langle a\ \neg b\ c\rangle$ and $s = \langle a\ c\ b\ c\rangle$, and a counterexample for Equation 10 is $p = \langle a\ \neg(bc)\ d\rangle$ and $s = \langle a\ b\ c\ d\rangle$. □

Proof of Lemma 10. We start with the implication $\boldsymbol{p} \lhd \boldsymbol{p}' \implies \boldsymbol{p} \lhd \boldsymbol{p}'$. Let $p, p' \in \mathcal{N}$ s.t. $\boldsymbol{p} \lhd \boldsymbol{p}'$. By definition, $k = k'$ and 1. $\forall i \in [k],\ p_i = p'_i$, 2. $\forall i \in [k-1],\ q_i \subseteq q'_i$

and 3. $\exists j \in [k-1]$ s.t. $q_i \neq q'_i$. A particular case of 1. is that $\forall i \in [k],\ p_i \subseteq p'_i$. In addition, third condition of $\lhd$ is obtained easily from 3. adding a disjunctive condition. Hence, $\boldsymbol{p} \lhd \boldsymbol{p'}$.

We now prove the second implication: $\boldsymbol{p} \lhd \boldsymbol{p'} \implies \boldsymbol{p} \lhd \boldsymbol{p'}$. Let $p, p' \in \mathcal{N}$ s.t. $\boldsymbol{p} \lhd \boldsymbol{p'}$. Let's now define the sequence u_i such that $u_i = i$ for all $i \in [k]$. By construction, we have that $u_i < u_{i+1}$, for all $i \in [k-1]$ (4.). In addition, by definition of $\lhd$, we have that $\forall i \in [k],\ p_i \subseteq p'_i = p'_{u_i}$, and $\forall i \in [k-1], q_i \subseteq q'_i = \bigcup_{j\in[i,(i+1)-1]} q'_j = \bigcup_{j\in[u_i,u_{i+1}-1]} q'_j$. Assuming $k = k'$, then $p_k \neq p'_k$ or $\exists j \in [k-1]$ s.t. $q_j \neq q'_j$. If $p_k \neq p'_k$ the third condition of $\lhd$ is satisfied (with $j = k$). Otherwise, it is also satisfied with the j of the definition of $\lhd$. □

Proof of Proposition 2. We begin with the $\lhd$ relation. We first remind that $\lhd$ is a strict partial order iff the three following conditions hold:

1. $\forall p \in \mathcal{N}$, not $p \lhd p$ (irreflexive),
2. $\forall p, p', p'' \in \mathcal{N}$, $p \lhd p'$ and $p' \lhd p'' \implies p \lhd p''$ (transitivity),
3. $\forall p, p' \in \mathcal{N}$, $p \lhd p' \implies$ not $p' \lhd p$ (asymmetry)

Irreflexive. Let's assume that $\exists \in \mathcal{N}$ s.t. $p \lhd p$. Then, because $k = k'$ the third condition implies that $\exists j \in [k-1]$ s.t. $q_j \neq q_j$, which is absurd. Then $\lhd$ is irreflexive.
Transitivity. Let $p, p', p'' \in \mathcal{N}$ s.t. $p \lhd p'$ and $p' \lhd p''$. Then, for all $i \in [k]$, $p_i = p'_i = p''_i$ and for all $i \in [k-1]$, $q_i \subseteq q'_i \subseteq q''_i$ ($k = k' = k''$). Finally, it is not possible to have $q_i = q''_i$ for all $i \in [k-1]$. In fact, if we have these equalities, then we would have $q_i = q'_i$ and $q'_i = q''_i$ for all $i \in [k-1]$ because $q_i \subseteq q'_i \subseteq q''_i$. But, it is not possible according to 3. Therefore, we have that $p \lhd p''$.
Asymmetry. Let $p, p' \in \mathcal{N}$ s.t. $p \lhd p'$. Then, according to 2. and 3., there exists $j \in [k-1]$ s.t. $q_j \subsetneq q'_j$. And then, it is not possible to have $q'_j \subseteq q_j$. As a consequence, we can not have $p' \lhd p$.
We now prove that $\lhd$ is a strict partial order.
Irreflexive. Let's assume that $\exists \in \mathcal{N}$ s.t. $p \lhd p$. Then, because $k = k'$ and that it is not possible that $p_k \neq p_k$, then the third condition implies that $\exists j \in [k-1]$ s.t. $q_j \neq q_j$, which is also absurd. Then $\lhd$ is irreflexive.
Transitivity. Let $p, p', p'' \in \mathcal{N}$ s.t. $p \lhd p'$ and $p' \lhd p''$. Then, for all $i \in [k]$, $p_i \subseteq p'_i \subseteq p''_i$ and for all $i \in [k-1]$, $q_i \subseteq q'_i \subseteq q''_i$ ($k \leq k' \leq k''$). Finally, if $k = k''$, then $k = k'' = k'$. Assuming that $p_k = p'_k$ and $p'_k = p''_k$ then $p_k = p''_{k''}$. Assuming that $p_k \neq p'_k$ or $p'_k \neq p''_k$, then $\exists j \in [k-1]$ s.t. $q_j \neq q'_j$ or $q'_j \neq q''_j$, and hence $q_j \neq q''_j$. Then, we have that $p \lhd p''$.
Asymmetry. Let $p, p' \in \mathcal{N}$ s.t. $p \lhd p'$. Then, if $k < k'$ we can not have $p' \lhd p$. Assuming that $k = k'$ and that $p_k = p'_k$ for the same reason as the asymmetry of $\lhd$, it is not possible to have $q_j \neq q'_j$. If $p_k \neq p'_k$ the, $p_k \subsetneq p'_k$ (according to 1.) and then it is not possible to have $p'_j \subseteq p_j$. As a consequence, we can not have $p' \lhd p$.
We now prove that $\lhd$ is a strict partial order.
Irreflexive. Let's assume that $\exists \in \mathcal{N}$ s.t. $p \lhd p$. Then, the unique strictly incremental (auto-)mapping is $u_i = i$, for all $i \in [k]$. Then, the third condition implies that $\exists j \in [k-1]$ s.t. $q_j \neq q_j$ or $p_j \neq p_j$, which is absurd. Then $\lhd$ is irreflexive.
Transitivity. Let $p, p', p'' \in \mathcal{N}$ s.t. $p \lhd p'$ and $p' \lhd p''$. We denotes by $(u_i)_i \in [k']^k$ and $(v_i)_i \in [k'']^{k'}$ the respective mapping, and we define $(w_i)_i \in [k'']^{[}k]$ such that $w_i = v_{u_i}$ for all $i \in [k]$. Then, for all $i \in [k]$, $p_i \subseteq p'_{u_i} \subseteq p''_{v_{u_i}} = p''_{w_i}$; $q_i \subseteq \bigcup_{j\in[u_i,u_{i+1}-1]} q'_j = \bigcup_{j\in[u_i,u_{i+1}-1]} \bigcup_{l\in[v_j,v_{j+1}-1]} q''_l$. The union of the q''_l in the intervals $[v_j, v_{j+1} - 1]$ for $j \in [u_i, u_{i+1} - 1]$ can be sum up as an union on the interval $[v_{u_i}, v_{(u_{i+1}-1)+1} - 1] = [v_{u_i}, v_{u_{i+1}} - 1] = [w_i, w_{i+1} - 1]$ because intervals are contiguous. Then, $q_i \subseteq \bigcup_{j\in[w_i,w_{i+1}-1]} q''_j$. Finally, if $k = k''$, then $k = k'' = k'$ and then it exists $j \in [k]$, s.t. $p_j \neq p'_j \subseteq p''_j$ or $q_j \neq q'_j \subseteq q''_j$. Thus, $p_j \neq p''_j$ or $q_j \neq q''_j$. As a consequence, we have $p \lhd p''$.
Asymmetry. Let $p, p' \in \mathcal{N}$ s.t. $p \lhd p'$. Then, if $k < k'$ we can not have $p' \lhd p$. Assuming that $k = k'$ (and thus $u_i = i$ for all $i \in [k]$), we have that it exists $j \in [k]$ s.t. $p_j \neq p'_j$ or $q_j \neq q'_j$. If $p_j \neq p'_j$ then, according to 1. $p_j \subsetneq p'_j$, it is not possible to have $p'_j \subseteq p_j$. If $q_j \neq q'_j$, then, according to 2. $q_j \subsetneq \bigcup_{j\in[u_i,u_{i+1}-1]} q'_j = q'_j$. Thus, it is not possible to have $q'_j \subseteq q_j = \bigcup_{j\in[u_i,u_{i+1}-1]} q_j$. As a consequence, we can not have $p' \lhd p$. □

Proof of Proposition 3. We start this proof by a small result about the anti-monotonicity of $\not\sqsubseteq$. Let $P, Q \in \mathcal{I}$ be two itemsets s.t. $P \subseteq Q$, and $I \in \mathcal{I}$ another itemset. Then, $Q \not\sqsubseteq I \implies P \not\sqsubseteq I$. In fact, $Q \not\sqsubseteq I$ implies that for all $e \in Q$, $e \notin I$, and because $P \subseteq Q$, we also have that $e \in P$, $e \notin I$.

Let $\boldsymbol{p} = \langle p_1\ \neg q_1\ \dots\ \neg q_{m-1}\ p_m \rangle \in \mathcal{N}$ and $\boldsymbol{p'} = \langle p'_1\ \neg q'_1\ \dots\ \neg q'_{m'-1}\ p'_{m'} \rangle \in \mathcal{N}$ be two NSP s.t. $\boldsymbol{p} \lhd \boldsymbol{p'}$.

We first show that an $(\circ, \not\sqsubseteq)$-embedding of $\boldsymbol{p'}$ in a sequence $\boldsymbol{s}$, denoted $\boldsymbol{e} = (e_i)_{i\in[m']}$, induces an $(\circ, \not\sqsubseteq)$-embedding of $\boldsymbol{p}$. By Definition 4, we have $p'_i \subseteq s_{e_i}$, $\forall i \in [m']$ and $q'_i \not\sqsubseteq s_j$, for all $j \in [e_i + 1, e_{i+1} - 1]$ and for all $i \in [m' - 1]$.
On the other side, $\boldsymbol{p} \lhd \boldsymbol{p'}$ implies that $p_i \subseteq p'_i$ for all $i \in [m]$. Then, because $m \leq m'$ ($\boldsymbol{p} \lhd \boldsymbol{p'}$), we have that $p_i \subseteq s_{e_i}$ for all $i \in [m]$. In addition, $\boldsymbol{p} \lhd \boldsymbol{p'}$ also implies that $q_i \subseteq q'_i$ for all $i \in [m-1]$ and thus, by anti-monotonicity of $\not\sqsubseteq$ (and $q'_i \not\sqsubseteq s_j$), we have $q_i \not\sqsubseteq s_j$ for all $j \in [e_i + 1, e_{i+1} - 1]$ and for all $i \in [m-1]$. In conclusion, we have that $\boldsymbol{e} = (e_i)_{i\in[m]}$ is an $(\circ, \not\sqsubseteq)$-embedding of $\boldsymbol{p}$.

We now show that an $(\bullet, \not\sqsubseteq)$-embedding of $\boldsymbol{p'}$ in a sequence $\boldsymbol{s}$, denoted $\boldsymbol{e} = (e_i)_{i\in[m']}$, induces an $(\bullet, \not\sqsubseteq)$-embedding of $\boldsymbol{p}$. By Definition 4, we have $p_i \subseteq s_{e_i}$, $\forall i \in [m']$ and $q_i \not\sqsubseteq \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$, for all $i \in [m' - 1]$.
On the other side, $\boldsymbol{p} \lhd \boldsymbol{p'}$ implies that $p_i \subseteq p'_i$ for all $i \in [m]$. Then, because $m \leq m'$ ($\boldsymbol{p} \lhd \boldsymbol{p'}$), we have that $p_i \subseteq s_{e_i}$ for all $i \in [m]$. In addition, $\boldsymbol{p} \lhd \boldsymbol{p'}$ also implies that $q_i \subseteq q'_i$ for all $i \in [m-1]$, by anti-monotonicity of $\not\sqsubseteq$, we have

$q_i \not\sqsubseteq \bigcup_{j\in[e_i+1,e_{i+1}-1]} s_j$, for all $i \in [m-1]$. In conclusion, we have that $\boldsymbol{e} = (e_i)_{i\in[m]}$ is an $(\bullet, \not\sqsubseteq)$-embedding of $\boldsymbol{p}$. □

Proof of Proposition 4. Let $\boldsymbol{p} = \langle p_1\ \neg q_1\ \ldots\ \neg q_{k-1}\ p_k\rangle \in \mathcal{N}$ and $\boldsymbol{p'} = \langle p'_1\ \neg q'_1\ \ldots\ \neg q'_{k'-1}\ p'_{k'}\rangle \in \mathcal{N}$ be two NSP s.t. $\boldsymbol{p} \lhd \boldsymbol{p'}$. Thus, we have that $k = k'$.

Similarly to the proof of Proposition 3, we can show that any $(\mathbf{o}, \not\sqsubseteq)$-embedding of $\boldsymbol{p'}$ in $\boldsymbol{s}$ induces an $(\mathbf{o}, \not\sqsubseteq)$-embedding of $\boldsymbol{p}$ in $\boldsymbol{s}$. This enables to conclude that $\preceq^{\not\sqsubseteq}_{\mathbf{o}}$ is anti-monotonic on $(\mathcal{N}, \lhd)$.

The anti-monotonicity of $\sqsubseteq^{\not\sqsubseteq}_{\circ}$ requires that each embedding of $\boldsymbol{p}^+$ in $\boldsymbol{s}$ satisfies the negations. Let us assume that $p' \sqsubseteq^{\not\sqsubseteq}_{\circ} s$, then there exists an embedding $(e_i)_{i\in[k]}$ of $\boldsymbol{p'}$. $(e_i)_{i\in[k]}$ is also an embedding of $\boldsymbol{p'}^+$ (Lemma 5). According to 1. and because $k = k'$, $\boldsymbol{p'}^+ = \boldsymbol{p}+$, and then $(e_i)_{i\in[k]}$ is an embedding of $\boldsymbol{p}$ in $\boldsymbol{s}$. Thus, we shown that there is at least one embedding of $\boldsymbol{p}^+$ in $\boldsymbol{s}$. If $\sqsubseteq^{\not\sqsubseteq}_{\circ}$ is not anti-monotonic, then there exists an embedding $(e_i)_{i\in[k]}$ of $\boldsymbol{p}^+$ such that $\exists j \in [k]$ and $l \in [e_j + 1, e_{j+1} - 1]$, s.t. it is false that $q_j \not\sqsubseteq s_l$ ($\exists \alpha \in q_j, \alpha \in s_l$). According to 2. $q_j \subseteq q'_j$, and thus it is false $q'_j \not\sqsubseteq s_l$. Nonetheless, $(e_i)_{i\in[k]}$ is also an embedding of $\boldsymbol{p'}^+$. And $p' \sqsubseteq^{\not\sqsubseteq}_{\circ} s$, it implies that $q'_j \not\sqsubseteq s_l$. There is a contradiction, thus $\sqsubseteq^{\not\sqsubseteq}_{\circ}$ is anti-monotonic.

The anti-monotonicity of $\sqsubseteq^{\not\sqsubseteq}_{\bullet}$ requires that each embedding of $\boldsymbol{p}^+$ in $\boldsymbol{s}$ satisfies the negations. Let us assume that $p' \sqsubseteq^{\not\sqsubseteq}_{\bullet} s$, then there exists an embedding $(e_i)_{i\in[k]}$ of $\boldsymbol{p'}$. $(e_i)_{i\in[k]}$ is also an embedding of $\boldsymbol{p'}^+$ (Lemma 5). According to 1. and because $k = k'$, $\boldsymbol{p'}^+ = \boldsymbol{p}+$, and then $(e_i)_{i\in[k]}$ is an embedding of $\boldsymbol{p}$ in $\boldsymbol{s}$. Thus, we shown that there is at least one embedding of $\boldsymbol{p}^+$ in $\boldsymbol{s}$. If $\sqsubseteq^{\not\sqsubseteq}_{\bullet}$ is not anti-monotonic, then there exists an embedding $(e_i)_{i\in[k]}$ of $\boldsymbol{p}^+$ such that $\exists j \in [k]$, s.t. it is false that $q_j \not\sqsubseteq \bigcup_{l\in[e_j+1,e_{j+1}-1]} s_l$. According to 2. $q_j \subseteq q'_j$, and thus it is false $q'_j \not\sqsubseteq \bigcup_{l\in[e_j+1,e_{j+1}-1]} s_l$. Nonetheless, $(e_i)_{i\in[k]}$ is also an embedding of $\boldsymbol{p'}^+$. And $p' \sqsubseteq^{\not\sqsubseteq}_{\bullet} s$, it implies that $q'_j \not\sqsubseteq \bigcup_{l\in[e_j+1,e_{j+1}-1]} s_l$. There is a contradiction, thus $\sqsubseteq^{\not\sqsubseteq}_{\bullet}$ is anti-monotonic. □

Proof of Proposition 5. Let $\theta, theta' \in \Theta$, then $\theta \geqslant \theta' \implies supp_\theta(\boldsymbol{p}) \leq supp_{\theta'}(\boldsymbol{p})$ for all $\boldsymbol{p} \in \mathcal{N}$ (by Definition 6 of the dominance relation). Thus, Proposition 5 comes immediately with Proposition 1. □

Proof of Proposition 6. Let $\boldsymbol{p}, \boldsymbol{p'} \in \mathcal{N}$ be two negative sequential patterns such that $\boldsymbol{p} \lhd \boldsymbol{p'}$. According to Proposition 3, $\boldsymbol{p'} \preceq^{\not\sqsubseteq}_{\mathbf{o}} \boldsymbol{s} \implies \boldsymbol{p} \preceq^{\not\sqsubseteq}_{\mathbf{o}} \boldsymbol{s}$ for all $\boldsymbol{s}$. Thus, $supp_{\not\sqsubseteq,\mathbf{o},\preceq}(\boldsymbol{p}) \leq supp_{\not\sqsubseteq,\mathbf{o},\preceq}(\boldsymbol{p})$.

If $\boldsymbol{p} \lhd \boldsymbol{p'}$. According to Proposition 4, $\boldsymbol{p'} \not\exists^{\not\sqsubseteq}_{\mathbf{o}} \boldsymbol{s} \implies \boldsymbol{p} \not\exists^{\not\sqsubseteq}_{\mathbf{o}} \boldsymbol{s}$ for all $\boldsymbol{s}$. Thus, $supp_{\not\sqsubseteq,\mathbf{o},\not\exists}(\boldsymbol{p}) \leq supp_{\not\sqsubseteq,\mathbf{o},\not\exists}(\boldsymbol{p})$. □

Actes JIAF 2019

Recherche Monte Carlo multi-arbres pour l'exploitation des jeux décomposés

Aline Hufschmitt **Jean-Noël Vittaut** **Nicolas Jouandeau**

LIASD - University of Paris 8, France
{alinehuf,jnv,n}@ai.univ-paris8.fr

Résumé

Dans cet article nous présentons une variation de la recherche arborescente Monte Carlo (MCTS) consistant à réaliser une recherche dans plusieurs arbres dans le but d'exploiter des jeux décomposés. Cette recherche multi-arbres MCTS (MT-MCTS) consiste à construire simultanément plusieurs arbres de recherche MCTS correspondant aux différents sous-jeux et permet, comme tous les algorithmes de la famille MCTS, d'évaluer les actions en cours de jeu. Nous appliquons MT-MCTS sur des jeux décomposés dans le domaine du *General Game Playing*. Nous présentons des résultats encourageants montrant que cette approche est prometteuse et ouvre de nouvelles pistes de recherche dans le domaine de l'exploitation des décompositions. Des jeux composés complexes sont résolus de 2 fois (*Incredible*) jusqu'à 25 fois plus vite (*Nonogramme*).

Abstract

In this paper, we propose a variation of the MCTS framework to perform a search in several trees to exploit game decompositions. Our Multiple Tree MCTS (MT-MCTS) approach builds simultaneously multiple MCTS trees corresponding to the different sub-games and allows, like MCTS algorithms, to evaluate moves while playing. We apply MT-MCTS on decomposed games in the *General Game Playing* framework. We present encouraging results showing that this approach is promising and opens new avenues for further research in the domain of decomposition exploitation. Complex compound games are solved from 2 times faster (*Incredible*) up to 25 times faster (*Nonogramme*).

1 Introduction

Le *General Game Playing* (GGP) est une branche de l'Intelligence Artificielle visant à créer des programmes polyvalents capables de jouer à n'importe quel jeu sans intervention humaine. Comme les règles du jeu ne sont pas connues à l'avance, aucun savoir expert ni algorithme spécialisé ne peut être utilisé. Un aspect important du GGP est le développement de méthodes d'analyse automatique des règles dans le but d'accélérer la recherche de la meilleure stratégie.

Parmi les jeux considérés dans le domaine du GGP, certains sont composés de différents sous-jeux indépendants assemblés séquentiellement, ou joués en parallèle de manière synchrone ou asynchrone [3]. Un programme capable d'identifier ces sous-jeux, de trouver la meilleure stratégie pour chacun et de combiner celles-ci, peut significativement réduire le coût de la résolution du jeu global [4]. Plusieurs méthodes ont été proposées pour décomposer des jeux solitaires [5] ou multi-joueurs [18, 7, 8]. Deux approches distinctes ont été proposées pour exploiter ces décompositions, la première inspirée de la planification hiérarchique et la seconde de la vérification de modèle utilisant ASP.

La première approche, nommée *Concept Decomposition Search* [5], est une recherche par approfondissement itératif permettant de résoudre des jeux solitaires. Chaque itération de la recherche est composée de deux étapes : une recherche locale permettant de collecter l'ensemble des plans locaux à profondeur donnée et une recherche globale visant à combiner les sous-plans des différents sous-jeux pour trouver le meilleur plan global. Cette approche est étendue aux jeux multi-joueurs [18] en utilisant des *turn-move sequences* (TMSeqs) comme résultat de la recherche locale. Les TMSeqs indiquent non seulement les actions, mais également les joueurs qui les ont exécutées. La recherche globale est fondée sur les techniques classiques de recherche arborescente, mais utilise les TMSeqs à la place des actions légales, ce qui réduit considérablement la taille de l'arbre par rapport à une recherche standard.

La seconde approche [3] utilise l'*Answer Set Programming* (ASP) pour résoudre les jeux décomposés. Les plans locaux sont combinés à mesure de leur calcul pour trouver

un plan global. Cette recherche est utilisée uniquement sur des jeux solitaires. La manière dont cette approche pourrait être généralisée aux jeux multi-joueurs n'est pas évidente et reste un problème ouvert.

Dans toutes ces approches précédentes, la recherche retourne un plan global ou rien. Dans le cadre du GGP, un temps limité est alloué pour analyser les règles avant le début du jeu (*startclock*). Un algorithme de recherche doit permettre de déterminer rapidement la première action à jouer puis de raffiner la stratégie pendant le temps de réflexion accordé entre chaque coup (*playclock*). Les recherches locales et globales doivent donc être menées en parallèle et permettre d'améliorer le plan tout en jouant.

Dans cet article, nous proposons une nouvelle approche pour résoudre les jeux décomposés fondée sur la recherche arborescente Monte Carlo (MCTS). MCTS constitue l'état de l'art pour les joueurs programmés dans le domaine du GGP, mais aussi pour des joueurs spécialisés comme Alpha Go [16], pour les jeux vidéos et d'autres domaines en dehors des jeux [2]. Notre recherche multi-arbres (MT-MCTS) construit simultanément plusieurs arbres correspondant aux différents sous-jeux. Plutôt que de produire un plan global, MT-MCTS évalue les actions à chaque étape du jeu. Un joueur programmé utilisant MT-MCTS peut donc commencer à jouer tout en explorant la suite du jeu à la recherche la meilleure action suivante. Nous comparons la performance de notre algorithme MT-MCTS, sur des jeux solitaires, avec celle d'une recherche MCTS utilisant la politique *Upper Confidence Bound applied to Trees* (UCT) et les tables de transposition.

Le reste de cet article est organisé comme suit. Dans la section suivante, nous présentons brièvement la recherche MCTS, la politique UCT et ses optimisations courantes. Dans la section 3, nous présentons l'approche MT-MCTS utilisant plusieurs arbres pour résoudre les jeux décomposés. Nous présentons les résultats expérimentaux sur différents jeux solitaires dans la section 4. Dans la section 5, nous discutons ces résultats, nous présentons les défis soulevés par la recherche simultanée dans plusieurs arbres, les possibles extensions de notre algorithme et les problèmes ouverts. Nous concluons dans la section 6.

2 MCTS et UCT

Un jeu peut être représenté sous forme d'un arbre. Chaque nœud représente un état du jeu et chaque arc représente un mouvement joint des joueurs [1]. Les algorithmes de la famille MCTS permettent de construire un arbre du jeu de manière incrémentale et asymétrique [2].

MCTS démarre à partir d'un unique nœud représentant l'état courant du jeu et répète les quatre étapes suivantes pendant un nombre d'itérations ou un laps de temps donné : la sélection d'un chemin dans l'arbre du jeu selon la *politique de l'arbre* ; l'expansion de l'arbre avec la création d'un nouveau nœud ; une simulation (*playout*) d'après la *politique par défaut* ; la rétro-propagation du résultat du *playout* pour mettre à jour l'évaluation statistique des nœuds (nombre de visites et cumul des récompenses). Le nombre de *playouts* dans MCTS est un facteur clef pour la qualité de l'évaluation des nœuds de l'arbre [10].

La *politique par défaut* consiste généralement à jouer des coups au hasard jusqu'à atteindre un état terminal du jeu. Pour la sélection, UCT [1] est la *politique de l'arbre* la plus courante. Elle permet un compromis entre l'exploitation des meilleurs coups et l'exploration de l'arbre complet. Dans chaque état visité, le coup choisi est celui qui maximise U :

$$U = \frac{w_i}{n_i} + C * \sqrt{\frac{\log N}{n_i}} \quad (1)$$

avec w_i le cumul des récompenses [2] obtenues pendant les playouts en choisissant le coup i, n_i le nombre de visites du nœud enfant correspondant, N le nombre de visites du nœud courant et C une constante équilibrant les termes exploration et exploitation [3].

Une optimisation courante consiste à utiliser une table de transposition : les états identiques du jeu sont représentés par un nœud unique dans l'arbre. Dans le cadre du GGP, pour garantir que tous les jeux se terminent en un nombre fini d'étapes, beaucoup de jeux qui comprennent des cycles utilisent un compteur de coups nommé *stepper*. Les états identiques du jeu, présents à différentes profondeurs dans l'arbre, sont ainsi différenciés par ce *stepper* et les transpositions peuvent uniquement apparaître à une même profondeur. Quand un programme utilise les transpositions, l'évaluation des différentes actions légales est généralement stockée dans les arcs sortants d'un nœud plutôt que dans ses nœuds enfants [15]. Le nombre de visites reste stocké dans le nœud parent.

Une autre optimisation commune, utilisée pour guider la recherche, est l'élagage des branches totalement explorées [12, 17]. Durant l'étape de sélection, plutôt que de retourner dans les branches qui ont été complètement évaluées, le score moyen de la branche est calculé et utilisé pendant l'étape de rétro-propagation.

3 Multiple Tree MCTS (MT-MCTS)

Dans le cadre du GGP, l'état du jeu est décrit par un ensemble fini de fluents instanciés dont certains sont vrais.

1. Dans le cadre du GGP, les joueurs jouent toujours simultanément. Pour simuler un jeu à coups alternés, les joueurs qui n'ont pas la main disposent d'un seul coup légal consistant à passer leur tour.

2. Dans le framework GGP, les récompenses sont représentées par un entier entre 0 et 100. Dans cet article, nous considérons que cette récompense est normalisée entre 0.00 et 1.00

3. Dans le framework GGP, C est généralement fixé à une valeur de 0.4 apportant un bon équilibre entre exploration et exploitation dans une majorité de jeux GGP.

Un ensemble de F fluents permet de décrire 2^F états distincts. Le résultat de la décomposition est l'identification de différents sous-ensembles de ces fluents représentant l'état des *sous-jeux* [6] que nous nommons ici des *sous-états*. Un jeu peut être décomposé en sous-jeux disjoints [8] dans ce cas ces sous-ensembles de fluents sont disjoints et l'ensemble des sous-jeux correspond à un partitionnement des fluents du jeu global. Une décomposition d'un jeu en sous-jeux non-disjoints peut également être envisagée. Dans ce cas les sous-ensembles de fluents peuvent se recouper e.g. quand les fluents représentent les cases d'un plateau de jeu et chaque sous-jeu représente une ligne ou une colonne du jeu global. La manière d'obtenir ces décompositions est hors du cadre du présent article.

Dans cette section, nous présentons les difficultés résultant de la décomposition, le principe général de notre approche multi-arbres MCTS qui permet de résoudre ces difficultés et des aspects de MT-MCTS pour lesquels nous avons empiriquement développé une politique spécifique : la sélection locale des meilleures actions dans un sous-jeu, la sélection d'une action au niveau global d'après les recommandations des sous-jeux et le marquage des branches totalement explorées.

3.1 Difficultés résultant de la décomposition

La décomposition d'un jeu en différents sous-jeux produit des sous-états dans lesquels les actions légales dépendent de la combinaison de ces sous-états, ce qui amène une première difficulté concernant le calcul des coups légaux.

Une autre difficulté apparaît quand un *stepper* est séparé du reste d'un jeu : des cycles peuvent apparaître dans certains sous-jeux. Dans les transpositions d'un sous-état, l'évaluation des actions peut différer en fonction de la profondeur du jeu. Ce problème est évoqué sous les termes de *graph history interaction problem* [13]. Il existe une solution générale pour les jeux à score binaire [9]. Dans le cadre du GGP, les scores sont plus progressifs et cette solution générale n'est donc pas applicable.

La décomposition soulève également un problème pour l'identification des sous-états terminaux. Par exemple, le jeu *Incredible* est décomposé en un labyrinthe (*Maze*), un jeu de cubes (*Blocks World*), un *stepper* et un ensemble d'actions inutiles de contemplation. Le jeu est terminé si le *stepper* atteint 20 ou si le jeu *Maze* est terminé. Dans *Blocks World*, un sous-état n'est jamais terminal par lui-même. Il est possible également d'imaginer un jeu dans lequel deux sous-états seraient tous deux non-terminaux mais dont la combinaison serait terminale et devrait être évitée dans un plan global.

La décomposition amène aussi une difficulté pour l'évaluation des sous-états dont le score peut dépendre d'une combinaison opportune avec d'autres sous-états. De plus, dans le cadre du GGP, le score décrit par le prédicat `goal` est fiable uniquement si l'état courant est terminal [11]. Ces deux faits rendent la fonction d'évaluation moins fiable dans les sous-arbres.

Finalement, le problème de la fiabilité de la décomposition se pose. Si la décomposition est inconsistante, l'évaluation des coups légaux peut être erronée, amenant le joueur programmé à choisir des coups illégaux et à calculer des états incohérents.

Pour éviter tous ces problèmes, nous proposons l'approche suivante : réaliser les simulations dans le jeu global et construire un sous-arbre pour chaque sous-jeu. Les coups légaux, l'état suivant et la fin de jeu peuvent être évalués pour le jeu global dans lequel le score réel est connu. La sélection des actions est réalisée en fonction de l'évaluation des sous-états dans les sous-arbres. Une décomposition inconsistante peut être détectée si durant deux simulations, la même action à partir du même sous-état amène à différents sous-états suivants. Une décomposition partielle [4] mais consistante permet de jouer dans le respect des règles même si son exploitation peut s'avérer moins efficace. Pour prendre en charge les transpositions tout en évitant le *graph history interaction problem*, la version courante de MT-MCTS ne considère les transpositions qu'à la même profondeur dans les sous-arbres. Chaque sous-jeu est donc représenté par un graphe acyclique dirigé avec une racine unique.

3.2 Simulations globales et construction des sous-arbres

Notre MT-MCTS itère les quatre même étapes que MCTS, à la différence que l'étape de sélection alterne différentes phases de sélection locale et globale. La figure 1 représente ces quatre étapes. Prenons le jeu solitaire représenté par l'arbre partiel en haut à gauche de la figure. Considérons que la décomposition de ce jeu aboutit à l'identification de deux sous-jeux parallèles asynchrones. Remarquez que les actions `a` et `c` à partir de la racine ont le même effet sur le premier sous-jeu et amènent toutes deux au même sous-état '+'. De même, les actions `b` et `c` ont le même effet sur le second sous-jeu et amènent au même sous-état '`o`'.

Considérons que les actions `a` et `b` ont été testées chacune une fois. Les sous-arbres des sous-jeux 1 et 2 contiennent les arcs représentant ces actions et les nœuds représentant les sous-états atteints. Les transitions ont été évaluées grâce aux scores remontés par les deux *playouts*. Durant la troisième descente (voir 1a), l'action `c`, non testée, est déjà évaluée : cette action amène à la combinaison des sous-états '+' et '`o`' qui ont déjà été ajoutés dans les sous-arbres et évalués. Les transitions représentées par

4. Une décomposition est *partielle* si un sous-jeu peut être décomposé.

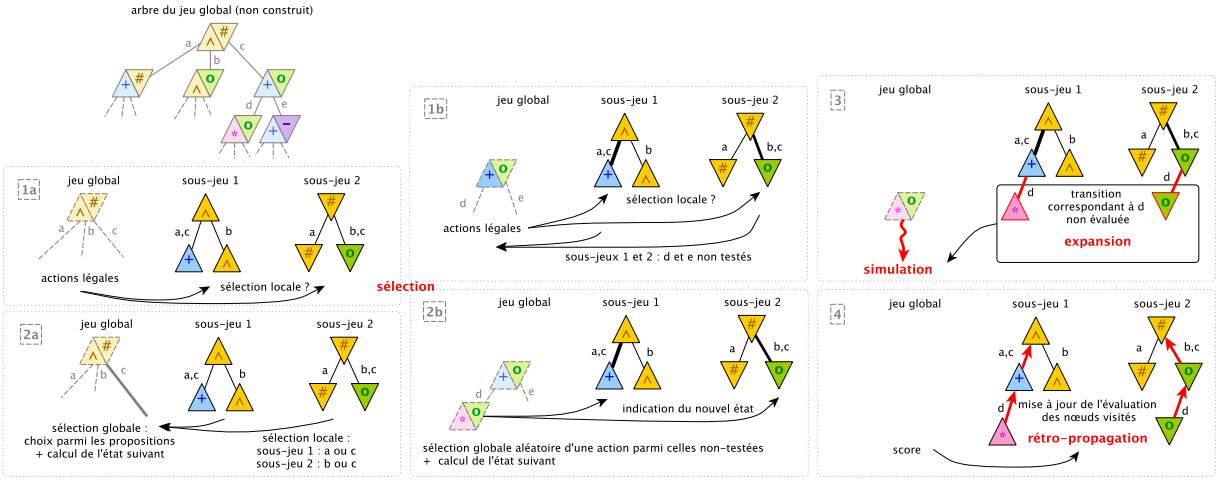


FIGURE 1 – Les quatre étapes de MT-MCTS. Les sous-états identiques sont représentés par un triangle de même orientation, symbole et couleur. La partie de chaque état appartenant au premier sous-jeu est représentée par un triangle pointe en haut, tandis que celle appartenant au second sous-jeu est représentée par un triangle pointe en bas. *Sélection* : 1a Sélection locale dans les sous-jeux, 2a sélection globale en fonction de la sélection locale, 1b sélection locale avec actions non testées, 2b sélection globale aléatoire des actions non testées. *Expansion et simulation* : 3 expansion des sous-arbres et simulation globale. *Rétro-propagation* : 4 rétro-propagation du score global dans chaque sous-jeu.

les arcs de l'arbre, sont étiquetées avec les différentes actions menant au même sous-état : ces dernières forment une *meta-action* [7]. Jouer avec des jeux décomposés permet donc de réduire significativement le facteur de branchement et la taille de l'espace de recherche. Remarquez que les actions b dans le sous-jeu 1 et a dans le sous-jeu 2 ne modifient pas l'état initial du jeu : les transitions qui laissent l'état d'un sous-jeu inchangé permettent d'évaluer l'intérêt de jouer ailleurs que dans ce sous-jeu. Comme les actions légales a, b et c ont été testées une fois, la sélection est effectuée d'après l'évaluation des sous-jeux.

Supposons (voir 2a) que le sous-jeu 1 recommande la transition vers le sous-état '+' (action a ou c) et que le sous-jeu 2 recommande la transition vers le sous-état 'o' (action b ou c). Maintenant, la sélection globale doit faire un choix en fonction de ces recommandations. Ici, l'action c est la plus recommandée et est choisie. Les étapes 1a et 2a sont itérées pour réaliser la descente dans l'arbre global. L'état suivant et les coups légaux sont calculés pour le jeu global ce qui garantit que seules des actions légales sont jouées et que la cohérence de l'état du jeu est préservée.

Quand le sous-état atteint comprend des actions inconnues (voir 1b), la sélection locale retourne la liste des actions non-testées. Une action est sélectionnée au hasard parmi elles (voir 2b), l'état suivant est évalué et les sous-jeux sont interrogés pour savoir si cet état suivant correspond à une transition déjà connue. Si les sous-états atteints sont connus de tous les sous-jeux, l'action est associée aux transitions correspondantes et l'étape 2b est ré-itérée en sélectionnant au hasard une autre action non-testée. S'il ne reste plus d'action non-testée, la descente continue comme décrit à l'étape 1a.

Si le sous-état atteint est nouveau pour au moins un des sous-jeux, une expansion est effectuée (voir 3) pour ajouter la nouvelle transition et le nouveau nœud correspondant dans chaque sous-arbre. Si une transposition existe, le sous-état correspondant est associé à la transition. Un *playout* est exécuté au niveau global à partir de ce nouvel état du jeu.

Dans l'état terminal atteint, le score est évalué (voir 4) puis transmis aux sous-jeux. Chaque sous-jeu remonte le score dans son sous-arbre pour évaluer les transitions et les nœuds visités. L'arbre du jeu global n'est pas construit. Seul l'état courant est conservé en mémoire pour permettre, à chaque étape, d'évaluer les coups légaux, l'état suivant et éventuellement de réaliser un *playout*.

3.3 Sélection locale

A chaque étape de la sélection par MT-MCTS, un sous-jeu reçoit un ensemble d'actions légales à évaluer. Le sous-état courant est associé à un nombre de visites N. Chaque transition i à partir de ce sous-état est évaluée par un nombre de visites n_i et un score cumulé w_i. La sélection locale retourne un ensemble d'actions s'il existe différentes transitions avec la même évaluation U ou si la meilleure transition est étiquetée avec différentes actions.

Nous avons exploré différentes approches pour réaliser la sélection locale. La première est une application standard de la politique UCT. Cependant, cette approche n'est pas satisfaisante car une transition dans un sous-jeu peut alors recevoir une bonne évaluation sans avoir contribué au score global : l'évaluation a été obtenue grâce à une séquence d'actions menant à une évaluation positive dans un autre sous-jeu. Un autre problème plus grave survient dans le cas des jeux à score binaire : le score est toujours zéro jusqu'à ce qu'une solution soit trouvée. Dans ce cas, la recherche se ramène à une exploration en largeur d'abord et ne permet pas la découverte des bonnes combinaisons de sous-états.

Dans le cadre du GGP, un état du jeu est décrit par un ensemble fini de fluents instanciés dont certains sont vrais. La décomposition partitionne ces fluents en différents groupes qui représentent les états des sous-jeux. Dans un état terminal global, il est possible de garder uniquement les fluents correspondant à l'un des sous-jeux, de donner la valeur *indéfini* aux autres fluents et d'évaluer les règles logiques du jeu avec une logique tri-valuée. Les prédicats `goal` vrais ou indéfinis représentent les scores possibles d'après l'état de ce sous-jeu. Le `goal` de score maximum (*lmax*) correspond au score maximum potentiel qui peut être obtenu si le reste de l'état du jeu correspond à la meilleure configuration possible dans les autres sous-jeux.

Le score *lmax* est indicatif, le vrai score maximum peut varier légèrement car une logique tri-valuée ne garantit pas l'information la plus précise [14]. L'évaluation *lmax* est néanmoins une estimation précieuse de la valeur d'un sous-état. Cette information peut être rétro-propagée en plus du score global et cumulée dans une variable w_i^{max}. Pour une transition donnée, w_i/n_i est un estimateur du score global et w_i^{max}/n_i est un estimateur du score local. Ces deux informations peuvent être utilisées dans une politique dérivée d'UCT (voir eq.1). Les actions choisies par le sous-jeu sont celles associées aux transitions dont l'évaluation U est maximum :

$$U = \alpha \frac{w_i}{n_i} + (1-\alpha)\frac{w_i^{max}}{n_i} + C * \sqrt{\frac{\log N}{n_i}} \qquad (2)$$

avec $\alpha \in [0, 1]$ pondérant les estimateurs de score global et local.

Pour éviter de retourner dans les branches déjà complètement explorées, les transitions correspondant à ces branches sont exclues de la sélection locale aussi longtemps que des transitions non totalement explorées existent.

3.4 Sélection globale

Dans un jeu comme *Incredible*, le sous-jeu *Maze* est rapidement complètement exploré. Ce sous-jeu recommande alors systématiquement la séquence d'actions menant au gain maximum dans ce sous-jeu (45% du score total). Cependant, terminer ce sous-jeu met fin prématurément au jeu global. Pour un algorithme fondé sur une balance entre exploration et exploitation, il faut un très grand nombre de visites de l'action terminale de *Maze* pour orienter la recherche vers l'exploration des autres actions possibles (jouer dans le sous-jeu *Blocks World*). Pour éviter ce problème, les coups légaux sont étiquetés en fonction de leur statut *terminal* ou non. Si certaines actions sont terminales, le score correspondant est évalué. Si une action terminale retourne le score maximum possible pour le joueur courant, elle est directement sélectionnée. Dans le cas contraire, la sélection locale est appliquée sur les actions légales non-terminales.

La sélection globale de la meilleure action est réalisée d'après les actions recommandées par les sous-jeux. Dans les jeux en série ou les jeux parallèles synchrones, l'intersection des ensembles d'actions recommandées est toujours non-vide. La sélection globale est alors évidente : une action est choisie au hasard dans cette intersection. En revanche, dans un jeu parallèle asynchrone, différents sous-jeux peuvent proposer des ensembles d'actions disjoints. Il est alors nécessaire de définir une politique pour le choix de l'action au niveau global. Pour définir une politique compatible avec tous les jeux, nous proposons une politique de vote. Chaque sous-jeu recommandant une action lui apporte une voix. Dans le cas des jeux en série ou parallèles synchrones, la meilleure action obtient autant de voix qu'il y a de sous-jeux. Dans le cas des jeux parallèles asynchrones, il est possible que chaque action n'obtienne qu'une seule voix.

Parmi les actions ayant reçu le plus de voix, celles apportant la plus grande espérance de gain sont sélectionnées pour diriger la recherche sur les actions menant à la meilleure combinaison de sous-états. Quand le but du jeu correspond à la conjonction des buts des sous-jeux, ce gain maximum espéré pour une action m parmi T sous-jeux est le produit des probabilités de gain dans chaque sous-jeu s :

$$\prod_{s=1}^{T} \frac{w_m^s}{n_m^s} \qquad (3)$$

Quand le but du jeu correspond à la disjonction des buts des sous-jeux, ce gain maximum espéré est la somme des probabilités de gain dans chaque sous-jeu s :

$$\sum_{s=1}^{T} \frac{w_m^s}{n_m^s} \qquad (4)$$

w_m^s est le cumul des récompenses gagnées durant les *playouts* et n_m^s le nombre de visites associés à la transition étiquetée par cette action m dans le sous-jeu s. Si plusieurs actions offrent la même espérance de gain maximum, une d'elles est sélectionnée aléatoirement.

3.5 Marquage des branches totalement explorées

Différentes séquences d'actions peuvent mener à visiter de nombreuses fois la même branche d'un sous-arbre. Pour éviter de re-visiter inutilement les branches totalement explorées, dans le cas d'UCT, il est courant de les étiqueter pour stopper la descente ou encourager la visite des branches voisines. Cependant, dans le cas des sous-jeux, un sous-état n'est pas toujours terminal en fonction du reste de l'état global. Il est donc nécessaire de développer une approche spécifique pour étiqueter les branches totalement explorées dans les sous-jeux. Nous proposons de résoudre ce problème par une simple révision du marquage durant les descentes successives.

Durant l'étape de sélection, la liste des coups légaux est calculée. Nous vérifions à cette occasion si les sous-états suivants connus comme terminaux sont bien terminaux dans la situation courante. Dans le cas contraire, l'étiquetage est révisé et cette révision est rétro-propagée le long du chemin visité pendant la descente. Lorsque toutes les transitions à partir d'un sous-état sont signalées comme totalement explorées, la transition menant à ce sous-état est également étiquetée comme "totalement explorée". Dans le cas de MT-MCTS, ce marquage est utilisé non pas pour stopper définitivement la recherche mais pour encourager la visite des branches voisines en priorité.

4 Expérimentations

Nous présentons ici des expérimentations menées avec MT-MCTS [5]. Premièrement nous évaluons différentes pondérations de notre politique de sélection locale et deuxièmement nous comparons la performance de MT-MCTS face à UCT. Nous montrons que notre approche peut réduire significativement le nombre de simulations et le temps de résolution des jeux. Nous menons nos expérimentations sur différents jeux solitaires : *Incredible*, différentes grilles de *Nonogramme* de taille 5×5 et 6×6 (fig.2) et *Queens08lg*.

Incredible est un jeu intéressant car il est possible de mettre prématurément fin au jeu avec un score sous-optimal. Il est couramment utilisé pour évaluer les joueurs programmés capables d'exploiter les décompositions. Le joueur met fin à la partie et obtient 45 points s'il rapporte un trésor à l'entrée du labyrinthe (*Maze*), additionné à $25 + 30$ points s'il réalise au préalable deux piles de cubes (*Blocks World*).

Les *Nonogrammes* sont des puzzles logiques dans lesquels les cases d'une grille doivent être colorées ou laissées blanches en fonction de nombres placés en bout des colonnes et des lignes. Le score est binaire et UCT n'apporte pas d'amélioration par rapport à une recherche en profondeur ou en largeur d'abord sur ce jeu.

Queens08lg est au contraire rapidement résolu par UCT. Il s'agit d'un *puzzle des huit reines* dans lequel il est illégal de placer une reine dans une position où elle pourrait capturer une autre reine en un coup. Le jeu est terminé quand aucune reine ne peut être ajoutée sur l'échiquier ou si le joueur déclare forfait. Le joueur reçoit une partie des points pour chacune des huit reines à placer.

Nous avons décomposé ces jeux en utilisant une méthode de décomposition statistique [8]. Pour chaque jeu et chaque configuration, nous avons réalisé 10 tests et nous mesurons le nombre moyen de *playouts* et le temps moyen nécessaires pour résoudre le jeu. Un jeu est considéré résolu quand une feuille de score maximum est trouvée.

4.1 Pondération de la politique de sélection locale

Le but de notre première expérimentation est de comparer différentes valeurs de α dans la politique de sélection locale (eq.2). Nous utilisons $C = 0.4$ qui permet un bon équilibre entre exploration et exploitation dans une majorité de jeux GGP. Nous présentons les résultats de nos tests sur deux jeux : *Incredible* et *Nonogramme "damier"*. Les résultats sont présentés sur la figure 3.

En utilisant uniquement l'estimateur de score global ($\alpha = 1$) dans la politique de sélection locale, il n'est pas possible de résoudre le *Nonogramme* à cause de son score binaire. Le score estimé est toujours zéro et les actions choisies au hasard ont très peu de chance de mener à la bonne combinaison de sous-états. Au contraire, l'utilisation exclusive de l'estimateur de score local ($\alpha = 0$) permet de guider la recherche dans les sous-arbres et de résoudre le *Nonogramme* en moins de 5 secondes. Cependant, l'utilisation de $\alpha = 1$ donne de meilleurs résultats dans *Incredible* tandis que $\alpha = 0$ nécessite presque deux fois plus de temps pour résoudre le jeu. En faisant varier α, nous remarquons qu'une faible participation de l'estimateur de score local ($\alpha = 0.75$) permet d'obtenir un résultat encore meilleur dans *Incredible* et une faible participation de l'estimateur de score global ($\alpha = 0.25$) permet d'obtenir la résolution la plus rapide du *Nonogramme*. Cependant, l'importance des écarts types, du même ordre de grandeur que le temps de résolution ou un ordre en dessous, ne permettent pas d'identifier un effet vraiment significatif de la variation de la pondération. En considérant l'écart type, les temps de résolution sont similaires dans les trois tests mêlant les estimateurs de score local et global. Plus d'expérimentation seront nécessaires pour vérifier l'influence d'une pondération inégale de ces deux informations.

Toutefois, l'association de ces deux estimateurs semble souhaitable pour constituer une politique polyvalente. Dans les expérimentations suivantes, nous utilisons la pondération $\alpha = 0.5$ dans la politique de sélection locale car elle

5. Les expérimentations ont été réalisées sur un cœur d'un Intel Core i7 2,7GHz avec 8Go de DDR3 à 1.6GHz.

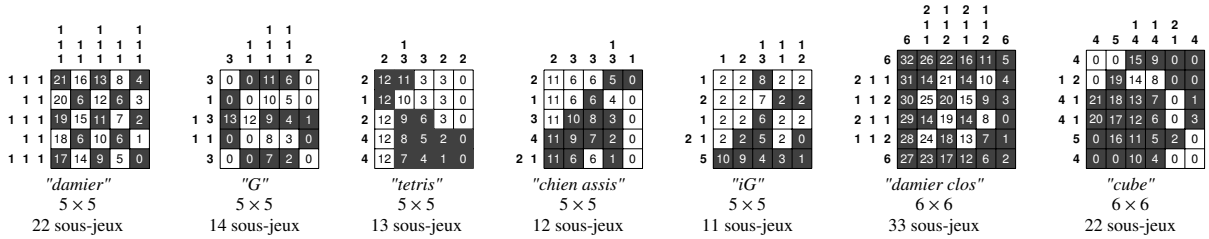


FIGURE 2 – Différentes grilles de Nonogramme (résolues). Le numéro à l'intérieur de chaque case indique le sous-jeu auquel elle appartient. Chaque case dont le statut peut être déterminé indépendamment des autres constitue un sous-jeu.

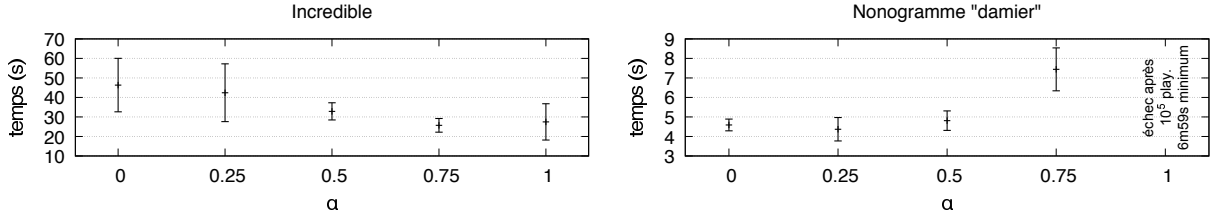


FIGURE 3 – Temps moyen sur 10 tests pour résoudre *Incredible* et *Nonogramme "damier"* avec différentes valeurs de α pour la politique de sélection locale de MT-MCTS (eq.2).

donne globalement le meilleur résultat.

4.2 MT-MCTS vs. UCT

Dans une seconde expérimentation (fig.4), nous comparons l'efficacité de MT-MCTS face à UCT en termes de nombre de *playouts* et de temps utilisés pour la résolution des jeux.

MT-MCTS est significativement meilleur que UCT sur le jeu *Incredible* en terme de *playouts* nécessaires pour résoudre le jeu. MT-MCTS utilise vingt fois moins de *playouts* mais la sélection des actions est significativement plus longue. Au final, le jeu est résolu deux fois plus vite.

La comparaison de nos résultats avec ceux obtenus avec la *Concept Decomposition Search* (*Fluxplayer*) [5] ou en encodant le problème en ASP [3] est difficile car ces approches sont totalement différentes de UCT. *Fluxplayer* met 2 heures pour résoudre *Incredible* en calculant 41 millions d'états (à comparer aux 280 milles *playouts* pour UCT [6]). L'exploitation du jeu décomposé réduit ce temps à 45 secondes et 3212 états. Ce temps de résolution est supérieur au nôtre bien que moins d'états soient calculés par leur approche. L'encodage de *Incredible* en ASP permet la résolution du jeu en 6.11 secondes. Ce temps est réduit à 1.94 secondes en exploitant la décomposition, soit un facteur de 3. Il faut noter que cette approche est optimisée pour les jeux solitaires. Comme notre approche requiert 21 fois moins de *playouts* pour la résolution du jeu décomposé, en optimisant l'étape de sélection de MT-MCTS, nous espérons obtenir une performance similaire voire meilleure.

6. Comme chaque *playout* donne lieu à une expansion de l'arbre, nous pouvons comparer le nombre de *playouts* avec le nombre d'états calculés.

Pour un puzzle simple comme *Queens08lg*, même si la résolution du jeu décomposé nécessite 3 fois moins de *playouts*, le temps de décomposition est trop important comparé au gain qui peut être espéré dans le temps de résolution.

Sur les grilles de *Nonogramme* (5×5 et 6×6), MT-MCTS est significativement meilleur que UCT. La résolution est 25 fois plus rapide pour *"tetris"*. Ce gain n'est pas directement lié au nombre de sous-jeux identités : pour *"damier"*, qui a un nombre plus important de sous-jeux, le temps de résolution n'est que 6 fois plus rapide. Le gain observé dans le temps de résolution est directement lié au nombre de simulations nécessaires : de 300 fois moins pour *"iG"* jusqu'à 2400 fois moins pour *"damier clos"* (sans mentionner les 4 tests où UCT a été interrompu après 10^7 *playouts* sans trouver de solution). L'étape de sélection plus lourde est largement compensée ici par le gain significatif dans le nombre de simulations. Le temps nécessaire pour résoudre *"cube"* avec MT-MCTS est de 2 heures en moyenne. UCT trouve parfois une solution en moins de 2 heures, mais la majorité des tests (8/10) échoue à trouver une solution après 3 heures en moyenne.

5 Discussion

Comme certaines combinaisons de sous-états peuvent ne jamais être testées, le partitionnement de la recherche sur plusieurs arbres n'offre en théorie aucune garantie de trouver la solution avec un nombre infini de *playouts*. En pratique, une bonne politique de sélection permet de guider la recherche vers la séquence d'actions adéquate pour at-

Jeu	UCT / MT-MCTS	# playouts	temps (décomp.)	σ	# échec
Incredible	UCT	280158	1m	14.3s	-
	MT-MCTS	13199	32.86s (2.50s)	4.4s	-
Queens08lg	UCT	67	0.01s	<0.1s	-
	MT-MCTS	22	1.30s (1.29s)	<0.1s	-
Nonogramme "damier"	UCT	432019	31.31s	16.5s	-
	MT-MCTS	776	4.81s (0.69s)	0.5s	-
Nonogramne "G"	UCT	2988921	4m24s	4m8s	-
	MT-MCTS	9344	36.61s (0.77s)	16.45s	-
Nonogramne "tetris"	UCT	4969111	7m20s	3m53s	1 *(après 10^7 play. = 15m17s)*
	MT-MCTS	3767	16.36s (1.01s)	6.5s	-
Nonogramne "chien assis"	UCT	2552082	3m43s	2m50s	-
	MT-MCTS	5476	26.27s (0.98s)	14.92s	-
Nonogramne "iG"	UCT	3086172	4m34s	3m9s	-
	MT-MCTS	10232	28.60s (0.85s)	12.40s	-
Nonogramne "damier clos"	UCT	4284872	13m7s	6m41s	4 *(après 10^7 play. = 30m25s minimum)*
	MT-MCTS	1762	49.26s (2.40s)	3.72s	-
Nonogramne "cube"	UCT	21438731	1h17m23s	19m16s	8 *(après 5×10^7 play. = 3h57s minimum)*
	MT-MCTS	358608	1h53m8s (2.75s)	45m7s	-

FIGURE 4 – Comparaison des temps de résolution des différents jeux avec UCT et MT-MCTS. Les colonnes présentent le nombre moyen de *playouts* et le temps moyen pour résoudre les puzzles (échecs exclus) sur 10 tests. La partie du temps utilisé pour la décomposition est présentée entre parenthèses. La colonne σ indique l'écart type. La colonne "échec" indique le nombre de recherches stoppées avant d'avoir trouvé une solution avec, entre parenthèses, le nombre de playouts exécutés et le temps minimum utilisé pour les réalisés.

teindre la bonne combinaison de sous-états.

Le problème, dans le cadre du GGP, est que la politique optimale dépend de la structure du jeu. Par exemple, exclure les branches totalement explorées de la sélection locale peut rapidement guider la recherche vers une solution dans *Incredible* car cela permet d'éviter de ré-explorer le chemin menant à un score sous-optimal. Cependant, cela peut ralentir la résolution d'un jeu comme *Nonogramme* dans lequel jouer les actions déjà identifiées comme bonnes permet de colorer certaines cases et de guider la découverte des bonnes actions suivantes.

Malgré ce ralentissement de la résolution pour les *Nonogrammes*, MT-MCTS est plus efficace que UCT sur ces puzzles. Cependant, comme une intéressante combinaison de sous-états peut rester inexplorée pendant un long moment, nous supposons qu'une valeur fixe de α dans notre politique de sélection locale peut ne pas être aussi efficace sur tous les jeux GGP. Des recherches complémentaires sont donc nécessaires. On peut envisager de nombreuses politiques distinctes qui permettraient de descendre rarement dans les branches totalement explorées et d'améliorer l'étape de sélection de MT-MCTS. Trouver une politique pour MT-MCTS dont l'efficacité serait prouvée sur tous les jeux est un problème ouvert intéressant.

Les contraintes chiffrées indiquées en bout des lignes et colonnes font que *Nonogramme* présente naturellement une composition en lignes et en colonnes. La structure de MT-MCTS permet d'exploiter un jeu décomposé de cette manière. Une autre piste de recherche à considérer est donc l'exploitation de différents sous-jeux qui se chevauchent et, plus généralement, de sous-jeux non disjoints.

Notre version de MT-MCTS ne considère pas toutes les transpositions pour éviter les jeux contenant des cycles. Cinquante cinq pour cent des jeux GGP utilisent un *stepper*. Le développement d'une politique de sélection spécifique pour tirer avantage des transpositions dans les jeux contenant des cycles est donc une piste intéressante qui pourrait significativement améliorer le niveau des joueurs programmés.

6 Conclusion

Dans cet article nous avons proposé une extension de MCTS pour effectuer une recherche dans plusieurs arbres représentant les différentes parties d'un problème décomposé. Nous avons testé cette idée sur différents jeux solitaires dans le cadre du *General Game Playing*. Jouer avec des jeux décomposés permet d'espérer un réel changement d'échelle dans leur vitesse de résolution. Nos tests avec une politique de sélection pondérée donnent des résultats prometteurs : les jeux sont résolus de 2 fois plus vite (*Incredible*) jusqu'à 25 fois plus vite (*Nonogramme*). La recherche multi-arbres (MT-MCTS) peut être étendue aux jeux multijoueurs comme les approches MCTS conventionnelles et permet également l'exploitation de sous-jeux non-indépendants.

La nouvelle approche MT-MCTS ouvre différentes pistes de recherche : le développement d'une politique de sélection efficace sur les différents types de jeux composés, la prise en charge du cas particulier des jeux contenant des cycles et utilisant un *stepper*, l'exploitation des sous-jeux se chevauchant et l'exploitation de décompositions incomplètes voire imparfaites.

Références

[1] Auer, Peter, Nicolò Cesa-Bianchi et Paul Fischer: *Finite-time Analysis of the Multiarmed Bandit Problem*. Mach. Learn., 47(2-3) :235–256, mai 2002, ISSN 0885-6125.

[2] Browne, Cameron B., Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis et Simon Colton: *A survey of Monte Carlo Tree Search methods*. Computational Intelligence and AI in Games, IEEE Transactions on, 4(1) :1–43, 2012.

[3] Cerexhe, Timothy, David Rajaratnam, Abdallah Saffidine et Michael Thielscher: *A Systematic Solution to the (De-)Composition Problem in General Game Playing*. Dans *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 2014.

[4] Cox, Evan, Eric Schkufza, Ryan Madsen et Michael Genesereth: *Factoring General Games using Propositional Automata*. Dans *Proceedings of the IJCAI-09 Workshop on General Game Playing (GIGA'09)*, pages 13–20, 2009.

[5] Günther, Martin, Stephan Schiffel et Michael Thielscher: *Factoring General Games*. Dans *Proceedings of the IJCAI-09 Workshop on General Game Playing (GIGA'09)*, pages 27–33, 2009.

[6] Hufschmitt, Aline: *Décomposition des jeux dans le domaine du General Game Playing*. Thèse de doctorat, LIASD, Université Paris 8, 2018.

[7] Hufschmitt, Aline, Jean Méhat et Jean Noël Vittaut: *A General Approach of Game Description Decomposition for General Game Playing*. Dans *Proceedings of the IJCAI-16 Workshop on General Game Playing (GIGA'16)*, pages 23–29, juillet 2016.

[8] Hufschmitt, Aline, Jean Noël Vittaut et Nicolas Jouandeau: *Statistical GGP Games Decomposition*. Dans *Proceedings of the IJCAI-18 Workshop on Computer Games (CGW 2018)*, page 19p., juillet 2018.

[9] Kishimoto, Akihiro et Martin Müller: *A General Solution to the Graph History Interaction Problem*. Dans *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*, pages 644–649, 2004.

[10] Kocsis, Levente et Csaba Szepesvári: *Bandit Based Monte-Carlo Planning*. Dans *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, 2006.

[11] Love, Nathaniel, Timothy Hinrichs, David Haley, Eric Schkufza et Michael Genesereth: *General Game Playing : Game Description Language Specification*. Rapport technique LG-2006-01, Stanford University, janvier 2006.

[12] Mehat, Jean et Tristan Cazenave: *Combining UCT and nested Monte Carlo search for single-player general game playing*. IEEE Transactions on Computational Intelligence and AI in Games, 2(4) :271–277, 2011.

[13] Palay, A.J.: *Searching With Probabilities*.

[14] Reps, Thomas W., Alexey Loginov et Shmuel Sagiv: *Semantic Minimization of 3-Valued Propositional Formulae*. Dans *17th IEEE Symposium on Logic in Computer Science (LICS 2002), 22-25 July 2002, Copenhagen, Denmark, Proceedings*, page 40, 2002.

[15] Saffidine, Abdallah, Jean Méhat et Tristan Cazenave: *UCD : Upper Confidence Bound for Rooted Directed Acyclic Graphs*. Dans *TAAI 2010*, pages 467–473, Piscataway, NJ -, 2010. TAAI 2010, IEEE.

[16] Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel et Demis Hassabis: *Mastering the game of Go without human knowledge*. Nature, 550 :354–359, octobre 2017.

[17] Winands, Mark H., Yngvi Björnsson et Jahn Takeshi Saito: *Monte-Carlo Tree Search Solver*. Dans *Proceedings of the 6th International Conference on Computers and Games*, CG '08, 2008.

[18] Zhao, Dengji, Stephan Schiffel et Michael Thielscher: *Decomposition of Multi-Player Games*. Dans *Proceedings of the Australasian Joint Conference onArtificial Intelligence*, tome 5866, pages 475–484. Springer, 2009.

Actes JIAF 2019

Une approche SAT sensible à la mémoire pour les logiques modales PSPACE

Jean-Marie Lagniez[1], Daniel Le Berre[1], Tiago de Lima[1] et Valentin Montmirail[2]

[1]CRIL, Univ Artois et CNRS, France
[2]I3S, Univ Côte-d'Azur et CNRS, France
{lagniez,leberre,delima}@cril.fr vmontmirail@i3s.unice.fr

Résumé

Le solveur SAT est devenu un oracle NP efficace pour résoudre des problèmes NP-complet (voir au-delà). En général, ces problèmes sont résolus soit par une traduction directe vers SAT soit en résolvant itérativement des problèmes SAT dans une procédure comme CEGAR. Récemment, une nouvelle boucle CEGAR récursive travaillant sur deux niveaux d'abstractions, appelée RECAR, a été proposée et instanciée pour la logique modale K. Nous visons à compléter ce travail pour les logiques modales utilisant les axiomes (B), (D), (T), (4) et (5). De plus, pour le rendre performant en pratique, nous généralisons le framework RECAR pour utiliser des compositions de fonctions d'abstraction. Les résultats expérimentaux montrent l'efficacité de cette approche et qu'elle surpasse les solveurs de l'état de l'art pour les logiques modales K, KT et S4 sur les benchmarks considérés.

Abstract

SAT technology has become an efficient running NP-oracle for solving NP-complete problems (and beyond). Usually those problems are solved by direct translation to SAT or by solving iteratively SAT problems in a procedure like CEGAR. Recently, a new recursive CEGAR loop working with two abstraction levels, called RECAR, was proposed and instantiated for modal logic K. We aim to complete this work for modal logics based on axioms (B), (D), (T), (4) and (5). Moreover, to make it efficient in practice, we generalize the RECAR framework to deal with compositions of abstraction functions. Experimental results show that the approach is efficient and outperforms state-of-the-art modal logic solvers for modal logics K, KT and S4 on considered benchmarks.

1 Introduction

La technologie SAT s'est avérée être une approche pratique très efficace pour résoudre des problèmes NP-complet [4]. L'un des principaux problèmes est de trouver le "bon" encodage pour le problème, c'est-à-dire, de trouver une réduction polynomiale du problème original vers une formule de la logique propositionnelle en forme normal conjonctive (CNF) qui peut être efficacement résolue par un solveur SAT [24]. En raison de leur efficacité, les solveurs SAT sont utilisés pour résoudre des problèmes au-delà de NP tels que la planification [25], la logique modale K [31] et QBF [14, 28] qui sont tous des problèmes PSPACE-complets ou encore le comptage de modèles en logique propositionnelle qui est un problème #P-complet et où les meilleures approches utilisent un solveur SAT [20, 23, 29, 33].

Malheureusement, en raison de la taille de la traduction, il n'est généralement pas possible de les encoder directement dans CNF. Pour pallier à ce problème, des procédures plus complexes utilisant des solveurs SAT comme oracles ont été conçues. Un exemple d'une telle procédure est la procédure CEGAR (*Counter-Example-Guided Abstraction Refinement*) [7]. Cette procédure peut être instanciée de deux manières différentes, à savoir CEGAR-over et CEGAR-under. CEGAR-over (resp. CEGAR-under) utilise des sur-abstractions (resp. des sous-abstractions). L'oracle est alimenté par une abstraction du problème original permettant moins (resp. plus) de modèles. S'il trouve un modèle (resp. s'il prouve qu'il n'y a pas de modèle), le problème d'origine est résolu. Sinon une nouvelle abstraction doit être définie, tenant compte de la réponse de l'oracle.

Récemment, Lagniez et al. [18] ont proposé une version récursive de CEGAR dans une procédure appelée RECAR (*Recursive Explore and Check Abstraction Refinement*). Ils ont instancié leur framework pour la logique modale K et ont démontré expérimentalement les avantages de basculer entre les deux types d'abstractions de manière récursive, en particulier pour repérer de petites sous-formules insatisfiables. Cependant, certaines applications pratiques nécessitent des logiques modales différentes de K, comme

KT en économie [21] ou S4 dans le domaine des logiques de descriptions [34]. Même si ces logiques modales sont PSPACE-complètes, en pratique, les réductions ne sont pas simples et des informations structurelles peuvent être perdues pendant la traduction. Afin d'étendre la portée de leur framework à d'autres logiques modales, nous proposons d'exploiter la correspondance entre les axiomes de la logique modale et les contraintes structurelles, mises en évidence dans la litterature [27], en codant les axiomes de la logique modale (D), (B), (4) et (5) en CNF. Pour ce faire, nous complétons la fonction de sur-abstraction initiale en ajoutant, pour chaque axiome, une contrainte structurelle qui force la structure de Kripke en construction à satisfaire cet axiome.

La taille de la CNF produite par [18] dépend fortement du nombre de mondes considérés pour la structure de Kripke en construction. Même si ce nombre est théoriquement exponentiel dans la taille de la formule d'entrée, ils ont démontré expérimentalement que le nombre de mondes nécessaires sur les instances considérées était souvent suffisamment petit pour faire qu'une approche basée sur SAT soit efficace en pratique pour la logique modale K. Quand ce n'est pas le cas, considérer judicieusement les sous-parties de la formule peut aider à décider de l'insatisfiabilité.

Cependant, comme démontré dans nos expériences, ceci n'est pas nécessairement vrai lorsque l'on considère des instances spécifiques dans d'autres logiques modales. Pour surmonter cette difficulté, nous proposons donc une généralisation des principes de RECAR. L'idée est de réduire agressivement la formule originale tout en préservant la satisfiabilité. Une telle idée a déjà été développée dans [18] lors de la suppression de certaines conjonctions de la formule d'entrée. Ici, nous allons encore plus loin en supprimant les disjonctions et/ou les modalités, en fonction de la logique modale considérée, en considérant des sous-parties de la formule originale à la fois pour la sur-abstraction et la sous-abstraction (seul la sous-abstraction a été considérée dans les travaux antérieurs).

Nous présentons également des simplifications supplémentaires, spécifiques aux logiques, pour les chaînes de modalités préservant la satisfiabilité. Et enfin, nous évaluons expérimentalement l'approche proposée pour les logiques modales K, KT et S4.

2 Préliminaires

Avant de passer à la logique modale et à son axiomatisation, rappelons quelques concepts fondamentaux de la logique propositionnelle.

Définition 1 (Langage de la logique propositionnelle)**.** *Soit $\mathbb{P} = \{p_0, \ldots, p_{n-1}\}$ un ensemble fini et non-vide de n variables propositionnelles. Le langage de la logique propositionnelle (noté CPL) est l'ensemble des formules contenant $\mathbb{P}$ et fermé sous l'ensemble des connecteurs propositionnels $\{\neg, \wedge\}$. Nous utilisons également les abréviations standard pour $\top$, $\bot$, $\vee$, $\rightarrow$ dt $\leftrightarrow$. Par exemple, $(\phi_1 \rightarrow \phi_2) \stackrel{def}{=} \neg(\phi_1 \wedge \neg\phi_2)$.*

Définition 2 (CNF - Forme Normale Conjonctive)**.** *Un littéral est une variable propositionnelle dans $\mathbb{P}$ ou sa négation. Une clause est une disjonction de littéraux. Une formule en forme normale conjonctive (CNF) est une conjonction de clauses.*

On sait que toute formule de *CPL* peut être traduite en une formule logiquement équivalente en CNF. Cependant, en pratique, des traductions efficaces telles que celle de Tseitin [36] exige des variables supplémentaires. Ainsi, il ne conserve que la satisfiabilité et parfois le nombre de modèles. Ici, lorsque nous utilisons le terme CNF, ou simplement formule, nous entendons une formule en CNF.

De récents solveurs SAT (logiciels capables de décider de la satisfiabilité d'une CNF) sont capables de vérifier la satisfiabilité d'une formule "sous hypothèses" [8] et de sortir un noyau inconsistant (une "raison" pour son insatisfiabilité). Le noyau insatisfiable est défini comme suit :

Définition 3 (Noyau inconsistant sous hypothèses)**.** *Soit ϕ une formule satisfiable en Forme Normal Conjonctive (CNF) construite en utilisant les variables booléennes de $\mathbb{P}$. Soit A un ensemble cohérent de littéraux construits en utilisant des variables booléennes à partir de $\mathbb{P}$, de telle sorte que $(\phi \wedge \bigwedge_{a \in A} a)$ soit inconsistant. $C \subseteq A$ est un noyau inconsistant (UNSAT core) de ϕ sous hypothèses A si et seulement si $(\phi \wedge \bigwedge_{c \in C} c)$ est inconsistant.*

Maintenant, nous pouvons introduire les notions requises pour comprendre la logique modale. De plus amples détails suer la logique modale ayant déjà été décrits dans la littérature [5, 6], passons maintenant aux notions utilisant la sémantique de Kripke [17].

Définition 4 (Langage de la logique modale)**.** *Soit $\mathbb{P} = \{p_0, \ldots, p_{n-1}\}$ un ensemble fini et non-vide de n variables propositionnelles et $\mathbb{M} = \{\Box_1, \ldots, \Box_m\}$ un ensemble fini et non-vide de m opérateur modaux unaires. Le langage de la logique modale (noté $\mathcal{L}$) est l'ensemble des formules contenant $\mathbb{P}$ et fermé sous l'ensemble des connecteurs propositionnels $\{\neg, \wedge\}$ et sous l'ensemble des opérateurs modaux dans $\mathbb{M}$. Nous utilisons également l'abréviation standard $\Diamond_a\phi \stackrel{def}{=} \neg\Box_a\neg\phi$.*

La profondeur modale d'une formule ϕ dans $\mathcal{L}$, noté depth(ϕ), est le nombre le plus élevé de modalités imbriquées. Le nombre d'atomes dans une formule ϕ dans $\mathcal{L}$ est noté $Atom(\phi)$.

Définition 5 (Structure de Kripke)**.** *Soit $\mathbb{P}$ un ensemble fini et non-vide de n variables propositionnelles et $\mathbb{M}$ un ensemble fini et non-vide de m opérateurs modaux unaires.*

TABLE 1 – Axiomes et leurs propriétés structurelles

Axiome	Contrainte du premier ordre	Schéma
Réflexivité (T)	$R_a(w,w)$	$\Box_a\phi \rightarrow \phi$
Symétrie (B)	$R_a(w_1,w_2) \rightarrow R_a(w_2,w_1)$	$\phi \rightarrow \Box_a\Diamond_a\phi$
Sérialité (D)	$R_a(w_1,w_2)$	$\Box_a\phi \rightarrow \Diamond_a\phi$
Transitivité (4)	$(R_a(w_1,w_2)\wedge R_a(w_2,w_3)) \rightarrow R_a(w_1,w_3)$	$\Box_a\phi \rightarrow \Box_a\Box_a\phi$
Euclidianité (5)	$(R_a(w_1,w_2)\wedge R_a(w_1,w_3)) \rightarrow R_a(w_2,w_3)$	$\Box_a\phi \rightarrow \Diamond_a\Box_a\phi$

Une structure de Kripke est un triplet $\mathcal{K} = \langle W, \{R_a \mid \Box_a \in \mathbb{M}\}, V\rangle$, où : W est en ensemble non-vide de mondes possibles, chaque $R_a \subseteq W \times W$ est une relation d'accessibilité binaire sur W et $V : \mathbb{P} \rightarrow 2^W$ est une fonction d'évaluation qui associe, à chaque $p \in \mathbb{P}$, l'ensemble des mondes possibles de W où p est vraie. Une structure de Kripke pointée est une paire $\langle \mathcal{K}, w\rangle$, où $\mathcal{K}$ est une structure de Kripke et w est un monde possible dans W. Par la suite, chaque fois que nous utilisons le terme "structure de Kripke", nous entendons "structure de Kripke pointée".

La taille d'une structure de Kripke $\langle \mathcal{K}, w\rangle$, qui est le nombre d'éléments dans W, est notée dans la suite $|\mathcal{K}|$. Sans perte de généralité, nous considérons uniquement les formules logiques modales sous forme normale négative, notées NNF (les négations apparaissent seulement avant les variables propositionnelles) [26, p. 204].

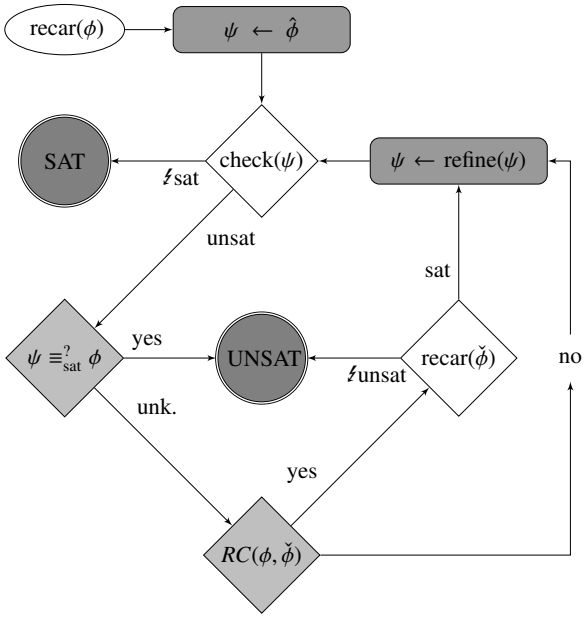
Définition 6 (Relation de satisfiabilité)**.** *La relation de satisfiabilité $\vDash$ entre les formules de logique modale et les structures est récursivement définie comme suit :*

$$
\begin{array}{lll}
\langle M,w\rangle \vDash \top & & \\
\langle M,w\rangle \vDash p & ssi & w \in \mathcal{I}(p) \\
\langle M,w\rangle \vDash \neg\phi & ssi & \langle M,w\rangle \nvDash \phi \\
\langle M,w\rangle \vDash \phi \wedge \psi & ssi & \langle M,w\rangle \vDash \phi \text{ et } \langle M,w\rangle \vDash \psi \\
\langle M,w\rangle \vDash \phi \vee \psi & ssi & \langle M,w\rangle \vDash \phi \text{ ou } \langle M,w\rangle \vDash \psi \\
\langle \mathcal{K},w\rangle \models \Box_a\phi & ssi & (w,w') \in R_a \text{ implique } \langle \mathcal{K},w'\rangle \models \phi \\
\langle \mathcal{K},w\rangle \models \Diamond_a\phi & ssi & (w,w') \in R_a \text{ et } \langle \mathcal{K},w'\rangle \models \phi
\end{array}
$$

Définition 7 (Validité)**.** *Une formule $\phi \in \mathcal{L}$ est valide (notée $\models \phi$) si et seulement si elle est satisfaite par toutes les structures de Kripke $\langle \mathcal{K}, w\rangle$. Une formule $\phi \in \mathcal{L}$ est satisfiable si et seulement si $\not\models \neg\phi$. Une structure de Kripke qui satisfait ϕ sera appelé un "modèle de ϕ".*

Par exemple, $\Box_a(\phi \rightarrow \psi) \rightarrow (\Box_a\phi \rightarrow \Box_a\psi)$ est valable pour tous $1 \leq a \leq m$ et toutes formules $\phi, \psi \in \mathcal{L}$. Cet ensemble particulier de formules est important dans la logique modale, il s'appelle "Schéma K ". Toutes les structures de Kripke satisfont K, et c'est pourquoi nous les appellerons ici "structures K". Une structure K peut satisfaire d'autres schémas comme ceux listés dans la Table 1. Toutes les combinaisons de ces schémas donnent lieu à 15 types de structures différentes [31, Table 25.2]. Ceci est dû au fait que certains schémas en impliquent d'autres (comme satisfaire (T) implique satisfaire (D), ou satisfaire (T) et (5) implique de satisfaire tous les schémas). Formellement nous avons :

FIGURE 1 – Le framework RECAR



Définition 8 ($K\star$-Validité)**.** *Soit $\star$ qui varie sur les 15 différents types de structures. Une formule $\phi \in \mathcal{L}$ est $K\star$-valide (notée $\models_{K\star} \phi$) si et seulement si elle est satisfaite par toutes les structures $K\star$ $\langle \mathcal{K}, w\rangle$. Une formule $\phi \in \mathcal{L}$ est $K\star$-satisfiable si et seulement si $\not\models_{K\star} \neg\phi$. Une structure $K\star$ qui $K\star$-satisfait une formule ϕ sera appelé un "modèle $K\star$ de ϕ'.*

Dans cet article, nous considérons seulement 3 logiques modales : K dont les modèles correspondent à toutes les structures K, KT dont les modèles correspondent à toutes les structures KT (c'est-à-dire, les structures K qui satisfont également le schéma (T)), et S4 dont les modèles correspondent à toutes les structures KT4 (c'est-à-dire, les structures K qui satisfont également les deux schémas (T) et (4)). En théorie, nous pouvons traiter toutes les logiques modales basées sur K. En pratique, nous avons testé expérimentalement seulement K, KT et S4 pour lesquels nous avons pu trouver des benchmarks.

3 Une approche RECAR pour K, KT et S4

Le framework RECAR, représenté sur la Fig.1, travaille avec deux niveaux d'abstraction. Il commence par considérer une sur-abstraction ψ de la formule d'entrée ϕ et vérifier sa satisfiabilité. Si ψ est satisfiable alors ϕ est satisfiable, la procédure s'arrête et renvoie SAT. Dans le cas contraire, il vérifie si ψ est trivialement equisatisfiable à ϕ. Si tel est le cas, alors ϕ est insatisfiable, la procédure s'arrête et renvoie

UNSAT. Sinon, il vérifie s'il est possible de construire une sous-abstraction $\check{\phi}$ de ϕ. Si une telle sous-abstraction ne peut être trouvée, la sur-abstraction est raffinée et le processus boucle. Sinon, $\check{\phi}$ est considérée comme à résoudre par un appel récursif à RECAR. Parce que $\check{\phi}$ est une sous-abstraction, si l'appel récursif prouve que $\check{\phi}$ est insatisfiable alors ϕ est également insatisfiable et la procédure s'arrête. Sinon, il ne peut pas conclure et donc il raffine ψ et répète tout le processus.

Pour être correctement instancié, certaines conditions doivent être vérifiées par les différents blocs utilisés dans RECAR [18] :

Hypothèses sur RECAR :

1. L'oracle 'check' est correct, complet et se termine.
2. $\hat{\phi}$ est satisfiable implique refine($\hat{\phi}$) est satisfiable.
3. Il existe $n \in \mathbb{N}$ tel que refine$^n(\hat{\phi}) \equiv^?_{\text{sat}} \phi$.
4. $\check{\phi}$ est insatisfiable implique ϕ est insatisfiable.
5. Let under(ϕ) = $\check{\phi}$. Il existe $n \in \mathbb{N}$ tel que $RC(\text{under}^n(\phi), \text{under}^{n+1}(\phi))$ renvoie faux, où RC représente une fonction booléenne déterminant si un appel récursif doit avoir lieu.

Puisque '$\hat{\phi}$' est une CNF, il suffit d'utiliser un solveur SAT pour respecter Hypo. 1. Pour respecter Hypo. 2 et 3, [18] propose une fonction $\hat{\phi}$ = over(ϕ, n) ce qui traduit la formule modale ϕ en une modélisation CNF demandant "est-ce que ϕ est satisfiable par un modèle de taille n ?". Lorsque la procédure répond positivement, ϕ est satisfiable. Sinon, il peut être raffiné avec $m > n$ jusqu'à ce que m atteint une borne supérieure théorique, qui est notée $UB(\phi)$. La borne supérieure théorique garantit également que le CNF généré est équisatisfiable à ϕ (RECAR Hypo. 3). La traduction ajoute de nouvelles variables p_i et $r^a_{i,j}$ à la formule : p_i dénote que la variable p est vraie dans le monde w_i alors que $r^a_{i,j}$ correspond à w_j étant accessible depuis w_i par la relation a.

Parce que ce sera important plus tard, nous rappelons ici que l'Hypo. 3 correspond aux lemmes suivants.

Such upper-bound is specific for each modal logic, as we can see below.

Lemme 1 ([30]). *$UB(\phi) = Atom(\phi)^{\text{depth}(\phi)}$ en logique modale K.*

Lemme 2 ([9]). *$UB(\phi) = 2^{|\phi|}$ en logique modales KT et S4.*

Dans ce qui suit, ces borne supérieures sont utilisées pour les logiques modales K, KT et S4. On sait que, s'il n'y a pas de modèle de taille $n \leq UB(\phi)$ qui satisfait ϕ alors il n'y a pas de modèle pour ϕ. Pour simplifier la lecture, bien que RECAR soit générique, dans la suite, nous utilisons RECAR pour se référer à une instanciation du framework pour la logique modale.

3.1 Comment encoder les axiomes

Il est connu que certains axiomes correspondent à des contraintes sur les structures de Kripke [27]. Par exemple, toute structure K réflexive satisfait (T). Même s'il existe aussi des structures K non-réflexives qui satisfont (T), il est toujours possible de trouver une structure K réflexive "équivalente". Deux structures K sont équivalentes si et seulement si elles sont bi-similaires [5]. Par conséquent, si l'on veut trouver un modèle KT fini, il est suffisant de ne rechercher que parmi les structures K réflexives. Un raisonnement analogue peut également être utilisé pour les autres propriétés. Par conséquent, en suivant la Table 1, nous appelons structure KT une structure K réflexive et nous appelons structure S4 (ou structure KT4) une structure K réflexive et transitive.

Par conséquent, pour traiter différentes logiques modales, nous ajoutons à la traduction en CNF les contraintes suivantes correspondant aux différents axiomes (avec m le nombre d'opérateurs modaux et n le nombre de mondes courants) :

Définition 9 (Traduction des axiomes)**.**

$$\text{over}((T), n) = \bigwedge_{a=0}^{m} \bigwedge_{i=0}^{n} (r^a_{i,i}) \quad \text{over}((D), n) = \bigwedge_{a=0}^{m} \bigwedge_{i=0}^{n} \bigvee_{j=0}^{n} (r^a_{i,j})$$

$$\text{over}((B), n) = \bigwedge_{a=0}^{m} \bigwedge_{i=0}^{n} \bigwedge_{j=0}^{n} (r^a_{i,j} \rightarrow r^a_{j,i})$$

$$\text{over}((4), n) = \bigwedge_{a=0}^{m} \bigwedge_{i=0}^{n} \bigwedge_{j=0}^{n} \bigwedge_{k=0}^{n} ((r^a_{i,j} \wedge r^a_{j,k}) \rightarrow r^a_{i,k})$$

$$\text{over}((5), n) = \bigwedge_{a=0}^{m} \bigwedge_{i=0}^{n} \bigwedge_{j=0}^{n} \bigwedge_{k=0}^{n} ((r^a_{i,j} \wedge r^a_{i,k}) \rightarrow r^a_{j,k})$$

La traduction de chaque axiome vient des relations avec la logique du premier ordre, présenté par [27]. Ainsi, lorsque l'axiome (T) est considéré (logique modal KT), la fonction de sur-abstraction est (over(ϕ, n) $\wedge$ over($(T), n$)). Quand les deux axiomes (T) et (4) (logique modale S4) sont considérés, la fonction de sur-abstraction est (over(ϕ, n) $\wedge$ over($(T), n$) $\wedge$ over($(4), n$)).

4 Abstractions agressives au niveau modal

Lorsque l'on considère des axiomes de logique modale autres que K, les formules ont généralement besoin de plus de mondes à satisfaire. Par conséquent, la traduction SAT devient parfois trop grande pour être manipulée (certains CNF ont des centaines de millions de clauses). Ainsi, nous proposons une nouvelle fonction de sur-abstraction sensible à l'espace et une nouvelle fonction de sous-abstraction consciente des axiomes, à utiliser dans le framework RECAR.

4.1 Sous-abstraction sensible aux axiomes

Jusqu'à présent, lorsque la formule était insatisfiable, la seule façon de prouver son insatisfiabilité était de couper une branche enracinée dans des nœuds ET afin de produire une sous-formule insatisfiable qui peut être traduite en CNF. En traitant avec l'axiome (T), nous avons $\Box_a\phi \rightarrow \phi$ et donc, comme démontré dans la propriété suivante, il est possible de construire une sous-abstraction de la formule qui supprime également certaines modalités.

Propriété 1. *Considérons une formule $\phi \in \mathcal{L}$ en NNF dans une logique modale satisfaisant (T). Si nous remplaçons une sous-formule $\Box_a\psi$ par ψ, alors la formule résultante ϕ' est une sous-abstraction de ϕ.*

Démonstration. Pour prouver que cette propriété est valide, il suffit de vérifier que si ϕ' est unsatisfiable alors ϕ l'est aussi. Ou, avec la contraposée, il suffit de vérifier que si ϕ est satisfiable alors ϕ' l'est aussi. Sans perte de généralité, dans ce qui suit nous supposons que tous les nœuds OU et ET sont binaires et nous rappelons que les opérateurs booléens sont commutatifs.

Considérons une formule logique modale ϕ en NNF, et ϕ' une copie de ϕ qui diffère d'une seule sous-formule enracinée sur un nœud de box $\Box_a\psi \in \phi$ où $\Box_a\psi$ a été remplacée par ψdans ϕ'.

Nous montrons que, s'il existe un modèle de Kripke $\mathcal{K} = \langle W, \{R_a \mid \Box_a \in \mathbb{M}\}, V\rangle$ et un monde possible $w \in W$, t.q. $\langle\mathcal{K}, w\rangle \models \phi$ alors $\langle\mathcal{K}, w\rangle \models \phi'$ par induction sur la structure de ϕ. Base d'induction : $\phi = \Box_a\psi$ et $\phi' = \psi$. Car ϕ et ϕ' sont en NNF, si $\exists\langle\mathcal{K}, w\rangle \models \Box_a\psi$, alors nous savons que nous avons $(w, w) \in R_a$ en raison de l'axiome de réflexivité (T), nous avons donc $\exists\langle\mathcal{K}, w\rangle \models \psi$. Montrons maintenant les différents cas sur l'étape d'induction :

(1) $\phi = (\chi_1 \wedge \chi_2)$ et χ_1 contient $(\Box_a\psi)$ et $\phi' = (\chi'_1 \wedge \chi_2)$ où χ'_1 est χ_1 où $(\Box_a\psi)$ a été remplacée par ψ. Le cas où χ_2 contient $(\Box_a\psi)$ est analogue à cause de la commutativité. Nous avons $\langle\mathcal{K}, w\rangle \models (\chi'_1 \wedge \chi_2)$ si et seulement si $\langle\mathcal{K}, w\rangle \models \chi'_1$ et $\langle\mathcal{K}, w\rangle \models \chi_2$. Par l'hypothèse d'induction, $\langle\mathcal{K}, w\rangle \models \chi_1$ et $\langle\mathcal{K}, w\rangle \models \chi_2$, si et seulement si $\langle\mathcal{K}, w\rangle \models (\chi_1 \wedge \chi_2)$. Ainsi $\langle\mathcal{K}, w\rangle \models \phi$.

(2) $\phi = (\chi_1 \vee \chi_2)$ et χ_1 contient $(\Box\psi)$ et $\phi' = (\chi'_1 \vee \chi_2)$ où χ'_1 est χ_1 où $(\Box_a\psi)$ a été remplacée par ψ. Analogue à (1).

(3) $\phi = \Box_a\chi$ et χ contient $(\Box_a\psi)$ et $\phi' = \Box_a\chi'$ où χ' est χ où $(\Box_a\psi)$ a été remplacée par ψ. Nous avons $\langle\mathcal{K}, w\rangle \models \Box_a\chi'$ si et seulement si $\forall w'$ t.q. si $(w, w') \in R_a$ alors nous avons $\langle\mathcal{K}, w'\rangle \models \chi'$. Par l'hypothèse d'induction, $\langle\mathcal{K}, w'\rangle \models \chi$, alors $\forall w'$ t.q. si $(w, w') \in R_a$ alors nous avons $\langle\mathcal{K}, w'\rangle \models \Box_a\chi$. Ainsi $\langle\mathcal{K}, w\rangle \models \phi$.

(4) $\phi = \Diamond_a\chi$ et χ contient $(\Box_a\psi)$ et $\phi' = \Diamond_a\chi'$ où χ' est χ où $(\Box_a\psi)$ a été remplacée par ψ. Analogue à (3).

Par conséquent pour toute formule $\phi \in \mathcal{L}$ en NNF dans une logique modale satisfaisant (T). Si nous remplaçons une sous-formule $\Box_a\psi$ par ψ, alors la formule résultante ϕ' est une sous-abstraction de ϕ. □

Notez que cette propriété ne s'applique pas à la logique modale K : $\Box_a\bot$ est valide en logique modale K mais incohérent en logique modale KT. Afin de sélectionner les cases qui seront remplacées, nous proposons d'améliorer la sous-abstraction présentée dans [18], qui combine sélecteurs et noyaux inconsistant pour rechercher des sous-formules insatisfiables.

Là, les auteurs ont proposé d'ajouter un sélecteur à chaque branche enracinée dans un nœud ET afin de pouvoir activer/désactiver certaines parties de la formule. Lorsque le solveur est appelé pour vérifier la satisfiabilité de la formule, il est appelé avec l'ensemble des sélecteurs comme hypothèses. Si le solveur renvoie UNSAT, le noyau inconsistant renvoyé est utilisé pour supprimer les parties de la formule qui ne font pas parti de la raison de son incohérence. Ce que nous proposons ici consiste également à remplacer chaque $\Box_a\psi$ par $((\neg s_k \vee \Box_a\psi) \wedge \psi)$. D'une certaine manière "quand nous activons le sélecteur s_k, alors nous traduisons la modalité entière, sinon nous traduisons simplement ψ dans le monde courant". Maintenant, nous pouvons à nouveau utiliser le noyau retourné par le solveur pour extraire une sous-formule insatisfiable. La sous-abstraction supprimera certaines cases qui ne sont pas satisfiables en raison de l'incohérence de la formule. À noter, quand nous traduisons une modalité □ en SAT, il n'est pas utile de traduire deux fois la formule ψ dans le monde courant.

4.2 Sur-abstraction sensible à la mémoire

Comme indiqué dans l'introduction, le goulot d'étranglement des approches CEGAR utilisant un oracle SAT est la taille des formules CNF générées. Pour le cas qui nous intéresse, ce goulot d'étranglement est atteint lorsque la fonction de sur-abstraction est appelée avec un trop grand nombre de mondes. Cependant, il est possible d'estimer la taille de la formule CNF avant de la calculer et ainsi prédire que la traduction épuisera la mémoire autorisée pour résoudre l'instance.

Dans ce qui suit, nous proposons une fonction de sur-abstraction spatiale qui est utilisée à la place de la fonction originale lorsque l'espace occupé par le CNF atteint un seuil donné. Cette nouvelle sur-abstraction est effectuée en désactivant certaines disjonctions de la formule d'origine. Pour que cette fonction existe et respecte les hypothèses de RECAR, nous devons vérifier que les branches enracinées dans un nœud OU qui ont été coupées produisent une formule plus faible, c'est-à-dire que chaque modèle de la formule résultante est également un modèle de la formule initiale.

Propriété 2. *Considérons une formule logique modale $\phi \in \mathcal{L}$ en NNF. Si nous coupons une branche enracinée dans un nœud OU, alors la formule résultante ϕ' est une sur-abstraction de ϕ.*

Démonstration. Pour prouver que cette propriété est valide, il suffit de vérifier que chaque modèle de ϕ' est aussi un modèle de ϕ. Sans perte de généralité, dans ce qui suit nous supposons que tous les nœuds OU et ET sont binaires. En effet, $(\phi_1 \oplus \phi_2 \oplus \cdots \oplus \phi_n)$ peut être réécrit comme $(\phi_1 \oplus (\phi_2 \oplus (\cdots \oplus (\phi_{n-1} \oplus \phi_n))))$, où $\oplus \in \{\wedge, \vee\}$. Rappelons aussi que les opérateurs booléens sont commutatifs, donc nous n'avons pas besoin de prouver le cas où nous ajoutons une nouvelle sous-formule à gauche. Soit ϕ une formule en NNF $\mathcal{L}$ contenant $(\psi_1 \vee \psi_2)$ et ϕ' est égale à ϕ mais avec $(\psi_1 \vee \psi_2)$ remplacée par ψ_1. Nous montrons que, s'il existe un modèle de Kripke $\mathcal{K} = \langle W, \{R_a \mid \Box_a \in \mathbb{M}\}, V\rangle$ et un monde possible $w \in W$, t.q. $\langle \mathcal{K}, w\rangle \models \phi'$ alors $\langle \mathcal{K}, w\rangle \models \phi$ par induction sur la structure de ϕ. Base d'induction : $\phi' = \psi_1$ et $\phi = (\psi_1 \vee \psi_2)$. Puisque ϕ et a fortiori ϕ' sont en NNF, la propriété est clairement valide. Laissez-nous maintenant prouver les différentes étapes :

(1) $\phi = (\chi_1 \wedge \chi_2)$ et χ_1 contient $(\psi_1 \vee \psi_2)$ et $\phi' = (\chi' \wedge \chi_2)$ où χ' est χ où $(\psi_1 \vee \psi_2)$ a été remplacée ψ_1. Le cas où χ_2 contient $(\psi_1 \vee \psi_2)$ est analogue à cause de la commutativité. Nous avons $\langle \mathcal{K}, w\rangle \models (\chi'_1 \wedge \chi_2)$ si et seulement si $\langle \mathcal{K}, w\rangle \models \chi'_1$ et $\langle \mathcal{K}, w\rangle \models \chi_2$. Par l'hypothèse d'induction, $\langle \mathcal{K}, w\rangle \models \chi_1$ et $\langle \mathcal{K}, w\rangle \models \chi_2$, si et seulement si $\langle \mathcal{K}, w\rangle \models (\chi_1 \wedge \chi_2)$. Ainsi $\langle \mathcal{K}, w\rangle \models \phi$.

(2) $\phi = (\chi_1 \vee \chi_2)$ et χ_1 contient $(\psi_1 \vee \psi_2)$ et $\phi' = (\chi' \vee \chi_2)$ où χ' est χ où $(\psi_1 \vee \psi_2)$ a été remplacée ψ_1. Analogue à (1).

(3) $\phi = \Box_a \chi$ et χ contient $(\psi_1 \vee \psi_2)$ et $\phi' = \Box_a \chi'$ où χ' est χ où $(\psi_1 \vee \psi_2)$ a été remplacée ψ_1. Nous avons $\langle \mathcal{K}, w\rangle \models \Box_a \chi'$ iff $\forall w'$ s.t. if $(w, w') \in R_a$ alors nous avons $\langle \mathcal{K}, w'\rangle \models \chi'$. Par l'hypothèse d'induction, $\langle \mathcal{K}, w'\rangle \models \chi$, alors $\forall w'$ t.q. si $(w, w') \in R_a$ alors nous avons $\langle \mathcal{K}, w'\rangle \models \Box_a \chi$. Ainsi $\langle \mathcal{K}, w\rangle \models \phi$.

(4) $\phi = \Diamond_a \chi$ et χ contient $(\psi_1 \vee \psi_2)$ et $\phi' = \Diamond_a \chi'$ où χ' est χ où $(\psi_1 \vee \psi_2)$ a été remplacée ψ_1. Analogue à (3).

Ainsi, $\forall \phi$ en NNF, si $\langle \mathcal{K}, w\rangle \models \phi'$ alors $\langle \mathcal{K}, w\rangle \models \phi$. □

Remarquons que la Propriété 2 peut être étendue au cas où un ensemble de branches enracinées dans des nœuds OU sont coupés. Par conséquent, il est possible d'envisager un nouveau type de sur-abstraction qui coupe les bords enracinés dans les nœuds OU. Plus précisément, nous créons une telle sous-formule de sur-abstraction en coupant via une heuristique un ensemble de branches enracinées dans des nœuds OU qui ont le plus grand impact sur la limite supérieure et que nous traduisons en une formule CNF. Nous appelons cette fonction cut-or(ϕ, b), où le paramètre b est le nombre de nœuds OU coupés. Évidemment, la fonction de raffinement est liée à cette sur-abstraction (il n'est pas possible de couper une branche et d'augmenter le nombre de mondes ensuite sans violer les hypothèse de RECAR). Ainsi, nous considérons une nouvelle fonction de raffinement refine$_{or}(\hat{\phi}, b)$ qui est défini de manière récursive comme suit :

$$\text{refine}_{or}(\hat{\phi}, b) = \begin{cases} \hat{\phi} & \text{si } b = 0 \\ \phi' & \forall b > 0 \text{ et } \text{refine}_{or}(\hat{\phi}, b-1) \neq \phi \\ \phi & \text{sinon} \end{cases}$$

où ϕ' est défini de telle sorte que refine$_{or}(\hat{\phi}, b-1) \subsetneq \phi' \subseteq \phi$. Intuitivement, refine$_{or}(\hat{\phi}, b)$ rétablit b nœuds OU à $\hat{\phi}$ si possible.

Fondamentalement, la fonction de raffinement consiste à restaurer au moins une branche coupée à chaque étape d'induction. Maintenant, montrons que le couple proposé sur-abstraction cut-or(ϕ, b) et fonction de raffinement refine$_o^n(\psi)$ satisfait les conditions requises pour RECAR rappelées dans les préliminaires.

Théorème 1. *ψ = cut-or(ϕ, n) est satisfiable implique* refine$_{or}(\psi, 1)$ *est satisfiable (Hypo. 2) et* refine$_{or}(\psi, n) \equiv_{\text{sat}}$ *ϕ (Hypo. 3).*

Démonstration. Par la Propriété 2 nous savons qu'ajouter une branche à un nœud enraciné en un OU préserve la satisfiabilité. Puisque refine$_{or}(\psi, b)$ peut seulement rajouter des branches, alors nous avons directement que ψ = cut-or(ϕ, n) est satisfiable implique refine$_{or}(\psi, 1)$ est satisfiable. Pour l'hypothèse 3, le résultat vient directement de la définition de refine$_{or}(\psi, n)$ et le fait que nous ne pouvons ajouter qu'un nombre fini de branches. □

4.3 Simplifications des chaînes de modalités

Une fois que toutes les simplifications ci-dessus sont effectuées, la formule "abstraite" qui en résulte peut contenir des chaînes de modalités. C'est une information critique qui peut être exploitée si nous voulons améliorer encore les performances du solveur.

Simplifications pour la logique modale S4 Les simplifications des chaîne des modalités dans la logique modale S4 sont connues et elles préservent l'équivalence logique [37, Sec. 5.5].

$$\Box_a \Box_a \phi \leftrightarrow \Box_a \phi \quad (1)$$

$$\Diamond_a \Diamond_a \phi \leftrightarrow \Diamond_a \phi \quad (2)$$

$$\Box_a \Diamond_a \Box_a \Diamond_a \phi \leftrightarrow \Box_a \Diamond_a \phi \quad (3)$$

Ainsi, cela signifie que toute chaîne d'opérateurs modaux (impliquant la même modalité a) peut être réduite à une chaîne d'au maximum 3 opérateurs modaux dans la logique modale S4, ce qui est tolérable pour la traduction en CNF faite par la fonction de sur-abstraction.

Puisque la logique modale K et la logique modale KT ont une infinité de modalités non équivalentes (*infinitely many non-equivalent modalities*) [37], il n'y a pas de tel résultat au meilleur de nos connaissances pour K et KT. Pour traiter des chaînes de modalités dans ces logiques, nous proposons les simplifications suivantes qui préservent la satisfiabilité.

Simplifications pour la logique modale K Considérons un cas simple où le préfixe modal ne contient que des diamants. Dans ce cas, il est suffisant de tester uniquement la fin de la chaîne sans tenir compte des modalités. La sous-formule résultante est équisatisfiable à la formule originale.

Théorème 2. $\Diamond_a \Diamond_a \dots \Diamond_a \psi \equiv_{\text{sat}} \psi$.

Démonstration. ($\Rightarrow$) Si $\langle \mathcal{K}, w \rangle \models \Diamond_a \dots \Diamond_a \psi$ alors il y a $w' t.q. \langle \mathcal{K}, w' \rangle \models \psi$. ($\Leftarrow$) Si $\langle \mathcal{K}, w \rangle \models \psi$ alors nous pouvons ajouter un monde w' à $\mathcal{K}$ t.q. $w \in R_a(w')$. Dans ce cas, $\langle \mathcal{K}, w' \rangle \models \Diamond_a \psi$. En continuant ainsi, nous pouvons montrer que $\Diamond_a \dots \Diamond_a \psi$ est satisfiable. □

Théorème 3. $\Diamond_a \dots \Diamond_a \Box_a \circ \dots \circ \psi$ *est satisfiable, pour* $\circ \in \{\Box_a, \Diamond_a\}$.

Démonstration. $\Box_a \psi$ est satisfiable pour tous $\psi \in \mathcal{L}$. Par conséquent, par Theo 2, $\Diamond_a \dots \Diamond_a \Box_a \chi$ est satisfiable pour tous $\chi \in \mathcal{L}$. □

Simplifications pour la logique modale KT La logique modale KT est différente. La réflexivité (Axiome (T)) implique que $\Box_a \bot$ est unsatisfiable, ce qui signifie que nous ne pouvons pas appliquer la même technique qu'en logique modale K. De plus, en raison de l'absence de transitivité (Axiome (4)), nous ne pouvons pas simplifier $(\Box_a \Diamond_a \Box_a \Diamond_a \phi)$ en $(\Box_a \Diamond_a \phi)$, comme c'est le cas en S4. Nous proposons donc de nouvelles simplifications qui préservent la satisfiabilité (mais pas l'équivalence) dans KT. Cependant, il est toujours possible de récupérer un modèle pour la formule d'origine en trouvant un modèle pour la formule simplifiée.

Théorème 4. $\Box_a \circ \psi \equiv_{\text{sat}} \circ \psi$, *for* $\circ \in \{\Box_a, \Diamond_a\}$.

Démonstration. Cas 1 : $\circ = \Diamond_a$. ($\Rightarrow$) If $\Box_a \Diamond_a \psi$ est satisfiable alors, par l'axiome (T), $\Diamond_a \psi$ est satisfiable. ($\Leftarrow$) Supposons $\langle \mathcal{K}, w \rangle \models \Diamond_a \psi$. Nous pouvons créer un nouveau modèle $\langle \mathcal{K}', w \rangle$ qui est le déroulement de $\langle \mathcal{K}, w \rangle$. Ceci est un arbre infini avec w comme racine, qui est bisimilaire à $\langle \mathcal{K}, w \rangle$. Par hypothèse, il y a au moins un monde $w' \in R_a(w)$ t.q. $\langle \mathcal{K}', w' \rangle \models \psi$. Notez également que la propriété de réflexivité de la structure d'origine $\mathcal{K}$ implique $\langle \mathcal{K}', w' \rangle \models \Diamond_a \psi$. Il peut aussi y avoir des mondes $w'' \in R_a(w)$ t.q. $\langle \mathcal{K}, w'' \rangle \not\models \psi$. Nous pouvons supprimer toutes ces branches de la racine. Dans le modèle obtenu, tous les mondes accessibles depuis w satisfont ψ. Pour faire de ce modèle un modèle KT, nous ajoutons une flèche réflexive uniquement sur la racine w. Ce modèle final satisfait $\Box_a \Diamond_a \psi$. Cas 2 est similaire. La seule différence est qu'il n'y a pas de branches w'' à supprimer. □

Théorème 5. $\Diamond_a \psi \equiv_{\text{sat}} \psi$

Démonstration. ($\Rightarrow$) S'il y a $\langle \mathcal{K}, w \rangle \models \Diamond_a \psi$, alors il y a $w' \in R_a(w)$ t.q. $\langle \mathcal{K}, w' \rangle \models \psi$. ($\Leftarrow$) If $\langle \mathcal{K}, w \rangle \models \psi$, par la contraposée de l'axiome (T), $\langle \mathcal{K}, w \rangle \models \Diamond_a \psi$. □

5 Résultats expérimentaux

Nous avons implémenté les abstractions et les simplifications proposées en plus de MoSaiC 1.0 [18, 19], dans un solveur appelé MoSaiC 2.0. Nous avons choisi de comparer les solveurs sur les benchmarks LWB classiques pour les logiques modales K, KT et S4 [3]. Ces benchmarks sont générés en utilisant le script de [22] en utilisant 56 formules avec 18 paramètres, pour un total de 1008 formules, 504 satisfiables, 504 insatisfiables.

La Table 2 synthétise les résultats. Elle montre le nombre de benchmarks résolus, ou une cellule grise si le solveur ne gère pas la logique considérée. En gras, les meilleurs résultats d'une ligne donnée et, entre parenthèses, le nombre de cas-tests qui ne peuvent être résolus par manque de mémoire.

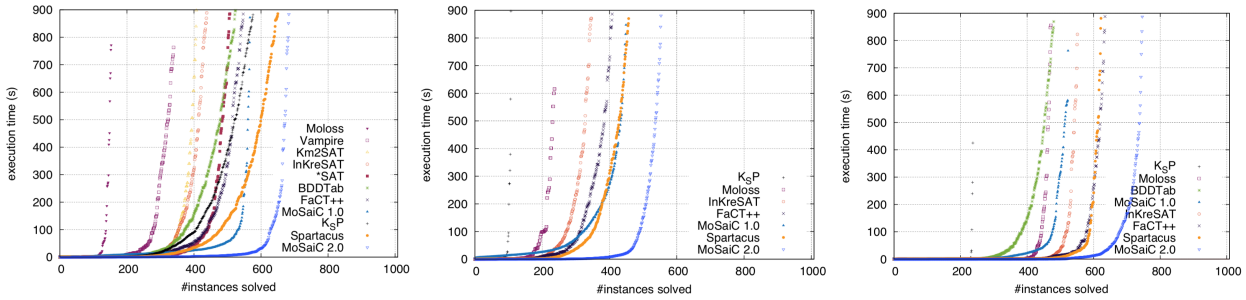
La ligne du VBS représente le *Virtual Best Solver* (une limite supérieure pratique sur la performance réalisable en choisissant le meilleur solveur pour chaque benchmark). Les expériences ont été exécutées sur un cluster de Xeon, 4 cœurs, 3.3 GHz avec CentOS 6.4 avec une limite de mémoire de 32Go et une limite de temps de 900 secondes par solveur par instance, quel que soit la logique considérée. Toutes les réponses aux solveurs ont été vérifiées car la satisfiabilité des instances est connu par construction. Aucune différence n'a été trouvée.

Nous avons comparé MoSaiC 2.0, avec les solveurs de l'état de l'art pour les logiques modales K, KT et S4, à savoir :

Moloss 0.9 [1],
$K_S P$ 0.1.2 [22],
BDDTab 1.0 [11],
FaCT++ 1.6.4 [35],
InKreSAT 1.0 [15],
*SAT 1.3 [10],
Km2SAT 1.0 [32] combiné avec le même solveur SAT Glucose (4.1) utilisé dans MoSaiC (1.0 et 2.0) [2],
Spartacus 1.0 [12],
MoSaiC 1.0 [19],
Vampire 4.0 [16] avec une combinaison de la traduction fonctionnelle optimisée [13].

Solver #Instances	**LWB$_K$ SAT** 504	**LWB$_K$ UNSAT** 504	**Total$_K$** 1008	**LWB$_{KT}$ SAT** 504	**LWB$_{KT}$ UNSAT** 504	**Total$_{KT}$** 1008	**LWB$_{S4}$ SAT** 504	**LWB$_{S4}$ UNSAT** 504	**Total$_{S4}$** 1008
Moloss	71 (0)	83 (0)	154 (0)	68 (0)	170 (0)	238 (0)	269 (0)	203 (0)	472 (0)
InKreSAT	192 (24)	247 (0)	439 (24)	155 (9)	193 (0)	348 (9)	248 (0)	304 (0)	552 (0)
BDDTab	248 (5)	277 (4)	525 (9)	–	–	–	211 (0)	270 (0)	481 (0)
FaCT++	264 (10)	284 (19)	548 (29)	184 (30)	226 (59)	410 (89)	298 (42)	338 (25)	636 (67)
MoSaiC 1.0	263 (241)	306 (198)	569 (439)	230 (251)	222 (253)	452 (504)	277 (229)	247 (255)	524 (484)
$K_S P$	249 (4)	**328** (3)	577 (7)	130 (2)	93 (0)	223 (2)	223 (0)	205 (0)	428 (0)
Spartacus	331 (33)	320 (10)	651 (43)	207 (74)	251 (59)	458 (133)	273 (17)	350 (13)	623 (30)
MoSaiC 2.0	**362** (142)	321 (73)	**684** (215)	**304** (167)	**249** (219)	**553** (386)	**360** (8)	**390** (31)	**750** (39)
VBS	362	342	704	304	249	553	364	390	750

TABLE 2 – Number of LWB instances solved in K, KT and S4



(a) en LWB K (b) en LWB KT (c) en LWB S4

FIGURE 2 – Distribution des temps d'exécution

Pour le basculement entre les deux fonctions de sur-abstraction dans MoSaiC 2.0, nous avons déterminé expérimentalement que le seuil ($|\phi| \times borneCourante > 5000$) a bien fonctionné pour ces instances sur notre ordinateur avec une limite de mémoire de 32 Go.

Les résultats obtenus Il est important de remarquer que $K_S P$ (aimablement fourni par ses auteurs) est encore en développement pour les logiques modales KT et S4. Ses résultats doivent être considérés comme préliminaires. Nous pouvons voir que le nombre de *Memory-Out* entre MoSaiC 1.0 et MoSaiC 2.0 réduit drastiquement dans les trois logiques grâce à la nouvelle sur-abstraction, qui réduit la taille de la formule qui doit être traduite.

En logique modale S4, on peut voir que l'écart entre MoSaiC 1.0 et MoSaiC 2.0 est plus important. L'approche basée sur SAT pour S4 est extrêmement efficace en raison des simplifications des chaînes de modalité, qui réduisent considérablement le temps de traduction. De plus, le temps de traduction étant réduit en raison de la nouvelle sur-abstraction et de la nouvelle sous-abstraction, MoSaiC 2.0 est beaucoup plus rapide que MoSaiC 1.0 (2s vs 28s en temps médian). Une autre manière d'observer l'impact de ces nouvelles abstractions est de regarder la distribution des temps d'exécution entre MoSaiC 1.0 et 2.0, visible sur la Figure 2(a) pour la logique modale K, sur la Figure 2(b) pour la logique modale KT et sur la Figure 2(c) pour la logique modale S4. Le coût de la traduction était très visible pour MoSaiC 1.0 si on compare sa distribution de temps d'exécution par rapport aux autres solveurs de l'état de l'art (il résout très peu d'instance en moins de 20 secondes par exemple), alors que MoSaiC 2.0 a pallié à ce coût, d'où le temps de résolution médian beaucoup plus bas.

Afin de comprendre la différence d'efficacité de la résolution de ces logiques, nous avons collecté des informations sur la taille des modèles Kripke calculés.

TABLE 3 – Tailles des structures de Kripke pour des benchmarks satisfiable pour chaque logique modale

	min	**Q_1**	**med**	**mean**	**Q_3**	**max**
K	1	2	22	174	279	903
KT	1	52	207	364	613	1505
S4	1	33	113	256	399	1217

Comme représenté dans la Table 3, l'ajout d'axiomes tend à augmenter la taille des modèles trouvés. Ceci peut s'expliquer en partie par le fait que ces modèles doivent satisfaire plus de contraintes. Cela arrive moins souvent en S4 car nous pouvons réduire considérablement le nombre de modalités.

6 Conclusion

Dans cet article, nous présentons un nouveau solveur de logique modale MoSaiC 2.0 qui étend MoSaiC 1.0 en considérant de nouvelles fonctions d'abstraction et en ciblant de nouvelles logiques modales (à savoir KT et S4). Les nouvelles fonctions d'abstraction visent à extraire une sous-formule insatisfiable, lorsque le problème est insatisfiable, ou de trouver un modèle pour une sous-formule satisfiable qui peut être étendue à toute la formule.

Les nouvelles fonctions d'abstraction ont été conçues pour traiter les cas où MoSaiC 1.0 sature la mémoire. Nous avons montré que ces fonctions pouvaient être combinées avec celles d'origine afin d'améliorer l'efficacité de l'approche. À cet égard, nous avons étendu le principe de simplification de la formule au niveau du domaine afin d'obtenir de meilleurs résultats au niveau SAT. Des simplifications logiques supplémentaires ont été proposées pour traiter les chaînes de modalités générées par les nouvelles fonctions d'abstraction.

MoSaiC 2.0 surpasse les autres solveurs sur les benchmarks considérés. Alors que le système de sélection actuel pour les abstractions à appliquer est simple et empirique, l'étape suivante serait d'adapter les abstractions à utiliser en fonction de la mémoire disponible, à chaque étape de la procédure.

Remerciements

Les auteurs remercient Ullrich Hustadt pour ses scripts en Perl pour générer les benchmarks LWB, Cláudia Nalon pour son aide sur comment faire pour que $K_S P$ résolvent des instances en KT et S4 et Mark Kaminski pour son aide sur FaCT++ et comment lui faire résoudre des instances en KT et S4.

Une partie de ce travail a été supporté par le Ministère de l'Enseignement Supérieur et la Recherche, par le projet ANR Investissement d'Avenir UCA$^{\text{JEDI}}$ (ANR-15-IDEX-01) et par le Conseil Régional des Haut-de-France au travers du "Contrat de Plan État Région (CPER) DATA".

Références

[1] Areces, Carlos, Pascal Fontaine et Stephan Merz: *Modal Satisfiability via SMT Solving*. Dans *Software, Services, and Systems*, pages 30–45. Springer, 2015, ISBN 978-3-319-15545-6. https://doi.org/10.1007/978-3-319-15545-6_5.

[2] Audemard, Gilles, Jean-Marie Lagniez et Laurent Simon: *Improving Glucose for Incremental SAT Solving with Assumptions: Application to MUS Extraction*. Dans *Proc. of SAT'13*, pages 309–317, 2013. http://dx.doi.org/10.1007/978-3-642-39071-5_23.

[3] Balsiger, Peter, Alain Heuerding et Stefan Schwendimann: *A Benchmark Method for the Propositional Modal Logics K, KT, S4*. JAR, 24(3):297–317, 2000. http://dx.doi.org/10.1023/A:1006249507577.

[4] Biere, Armin, Marijn Heule, Hans van Maaren et Toby Walsh (rédacteurs): *Handbook of Satisfiability*, tome 185 de *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009, ISBN 978-1-58603-929-5.

[5] Blackburn, Patrick, Johan van Benthem et Frank Wolter: *Handbook of Modal Logic*, tome 3. Elsevier, 2006, ISBN 978-0444516909.

[6] Chellas, Brian F.: *Modal Logic: An Introduction*. Cambridge University Press, 1980, ISBN 978-0521295154.

[7] Clarke, Edmund M., Orna Grumberg, Somesh Jha, Yuan Lu et Helmut Veith: *CounterExample-Guided Abstraction Refinement For Symbolic Model Checking*. J. of the ACM, 50(5):752–794, 2003. http://doi.acm.org/10.1145/876638.876643.

[8] Eén, Niklas et Niklas Sörensson: *An Extensible SAT-solver*. Dans *Proc. of SAT'03*, pages 502–518, 2003. http://dx.doi.org/10.1007/978-3-540-24605-3_37.

[9] Gabbay, Dov M.: *A General Filtration Method For Modal Logics*. Journal of Philosophical Logic, 1(1) :29–34, 1972. https://doi.org/10.1007/BF00649988.

[10] Giunchiglia, Enrico, Armando Tacchella et Fausto Giunchiglia: *SAT-Based Decision Procedures for Classical Modal Logics*. JAR, 28(2):143–171, 2002. http://dx.doi.org/10.1023/A:1015071400913.

[11] Goré, Rajeev, Kerry Olesen et Jimmy Thomson: *Implementing Tableau Calculi Using BDDs: BDDTab System Description*. Dans *Proc. of IJCAR'14*, pages 337–343, 2014. http://dx.doi.org/10.1007/978-3-319-08587-6_25.

[12] Götzmann, Daniel, Mark Kaminski et Gert Smolka: *Spartacus: A Tableau Prover for Hybrid Logic*. ENTCS, 262:127–139, 2010. http://dx.doi.org/10.1016/j.entcs.2010.04.010.

[13] Horrocks, Ian, Ullrich Hustadt, Ulrike Sattler et Renate A. Schmidt: *Computational Modal Logic*. Studies in Logic and Practical Reasoning, 3:181–245, 2007.

[14] Janota, Mikolás, William Klieber, Joao Marques-Silva et Edmund M. Clarke: *Solving QBF with CounterExample Guided Refinement*. Art. Int., 234:1–25, 2016.

[15] Kaminski, Mark et Tobias Tebbi: *InKreSAT: Modal Reasoning via Incremental Reduction to SAT*. Dans *Proc. of CADE'13*, pages 436–442, 2013. http://dx.doi.org/10.1007/978-3-642-38574-2_31.

[16] Kovács, Laura et Andrei Voronkov: *First-Order Theorem Proving and Vampire*. Dans *Proc. of CAV'13*, pages 1–35, 2013. http://dx.doi.org/10.1007/978-3-642-39799-8_1.

[17] Kripke, Saul: *A Completeness Theorem in Modal Logic*. J. Symb. Log., 24(1):1–14, 1959. http://dx.doi.org/10.2307/2964568.

[18] Lagniez, Jean Marie, Daniel Le Berre, Tiago de Lima et Valentin Montmirail: *A Recursive Shortcut for CEGAR: Application To The Modal Logic K Satisfiability Problem*. Dans *Proc. of IJCAI'17*, 2017. https://doi.org/10.24963/ijcai.2017/94.

[19] Lagniez, Jean Marie, Daniel Le Berre, Tiago de Lima et Valentin Montmirail: *A SAT-Based Approach For PSPACE Modal Logics*. Dans *Proc. of KR'18*, pages 651–652. AAAI Press, 2018. https://aaai.org/ocs/index.php/KR/KR18/paper/view/17997.

[20] Lagniez, Jean-Marie et Pierre Marquis: *An Improved Decision-DNNF Compiler*. Dans *Proc. of IJCAI'17*, pages 667–673, 2017. https://doi.org/10.24963/ijcai.2017/93.

[21] Matsuhisa, Takashi: *Core Equivalence in Economy for Modal Logic*. Dans *Proc. of ICCS'03, Part II*, pages 74–83, 2003. https://doi.org/10.1007/3-540-44862-4_9.

[22] Nalon, Cláudia, Ullrich Hustadt et Clare Dixon: *$K_S P$: A Resolution-Based Prover for Multimodal K*. Dans *Proc. of IJCAR'16*, pages 406–415, 2016. http://dx.doi.org/10.1007/978-3-319-40229-1_28.

[23] Oztok, Umut et Adnan Darwiche: *On Compiling CNF into Decision-DNNF*. Dans *Proc. of CP'14*, pages 42–57, 2014. https://doi.org/10.1007/978-3-319-10428-7_7.

[24] Prestwich, Steven David: *CNF Encodings*. Dans Biere, Armin *et al.* [4], pages 75–97, ISBN 978-1-58603-929-5. http://dx.doi.org/10.3233/978-1-58603-929-5-75.

[25] Rintanen, Jussi: *Planning and SAT*. Dans Biere, Armin *et al.* [4], pages 483–504, ISBN 978-1-58603-929-5. https://doi.org/10.3233/978-1-58603-929-5-483.

[26] Robinson, John Alan et Andrei Voronkov (rédacteurs): *Handbook of Automated Reasoning*, tome 1. Elsevier and MIT Press, 2001, ISBN 0-444-50813-9.

[27] Sahlqvist, Henrik: *Completeness and correspondence in the first and second order semantics for modal logic*. Dans *Proc. of the 3rd Scandinavian Logic Sumposium*, 1973.

[28] Samulowitz, Horst et Fahiem Bacchus: *Using SAT in QBF*. Dans *Proc. of CP'05*, pages 578–592, 2005. https://doi.org/10.1007/11564751_43.

[29] Sang, Tian, Fahiem Bacchus, Paul Beame, Henry A. Kautz et Toniann Pitassi: *Combining Component Caching and Clause Learning for Effective Model Counting*. Dans *Proc. of SAT'04*, 2004. http://www.satisfiability.org/SAT04/programme/21.pdf.

[30] Sebastiani, Roberto et David McAllester: *New Upper Bounds for Satisfiability in Modal Logics the Case-study of Modal K*. Technical Report 9710-15, IRST, Trento, Italy, October 1997.

[31] Sebastiani, Roberto et Armando Tacchella: *SAT Techniques for Modal and Description Logics*. Dans Biere, Armin *et al.* [4], pages 781–824, ISBN 978-1-58603-929-5. http://dx.doi.org/10.3233/978-1-58603-929-5-781.

[32] Sebastiani, Roberto et Michele Vescovi: *Automated Reasoning in Modal and Description Logics via SAT Encoding: the Case Study of K(m)/ALC-Satisfiability*. JAIR, 35:343–389, 2009.

[33] Thurley, Marc: *sharpSAT - Counting Models with Advanced Component Caching and Implicit BCP*. Dans *Proc. of SAT'06*, pages 424–429, 2006. https://doi.org/10.1007/11814948_38.

[34] Toriz, Juan Pablo Munoz, Iván Martínez Ruiz et José Ramón Enrique Arrazola-Ramírez: *On Automatic Theorem Proving with ML*. Dans *Proc. of MICAI'14*, pages 231–236, 2014. https://doi.org/10.1109/MICAI.2014.42.

[35] Tsarkov, Dmitry et Ian Horrocks: *FaCT++ Description Logic Reasoner: System Description*. Dans *Proc. of IJCAR'06*, pages 292–297, 2006. http://dx.doi.org/10.1007/11814771_26.

[36] Tseitin, G. S: *On the Complexity of Derivation in Propositional Calculus*, pages 466–483. Springer, 1983.

[37] Van Benthem, Johan: *Modal Logic For Open Minds*, tome 1. Center for the Study of Language and Inf, 2010, ISBN 978-1575865980.

Actes JIAF 2019

Decentralized Reasoning on a Network of Aligned Ontologies with Link Keys

Jérémy Lhez[1] **Chan Le Duc**[1] **Thinh Dong**[2] **Myriam Lamolle**[1]

[1] LIASD, IUT de Montreuil, Université Paris 8, France
[2] Université de Danang, Vietnam
{lhez, leduc, lamolle}@iut.univ-paris8.fr, dnnthinh@kontum.udn.vn

Résumé

Les clés de liage ont été récemment introduites pour la formalisation de données interconnectées entre des sources de données. Elles sont considérées comme un nouveau type de correspondances d'alignement d'ontologie. Nous proposons une procédure de raisonnement décentralisé sur un réseau d'ontologies avec alignements comprenant des clés de liage. Dans cet article, les ontologies incluses dans un tel réseau sont exprimées en logique de description $\mathcal{ALC}$ alors que les alignements peuvent contenir des correspondences d'individu, de clé de liage et de concept. Ces dernières sont munies d'une sémantique affaiblie. L'aspect décentralisé de notre procédure est fondé sur un processus de propagation de connaissances à travers le réseau via les correspondances. Ce processus permet de réduire polynomialement le raisonnement global au raisonnement local.

Abstract

Link keys are recently introduced to formalize data interlinking between data sources. They are considered as a new kind of correspondences included in ontology alignments. We propose a procedure for reasoning in a decentralized manner on a network of ontologies with alignments containing link keys. In this paper, the ontologies involved in such a network are expressed in the logic $\mathcal{ALC}$ while the alignments can contain concept, individual and link key correspondences equipped with a loose semantics. The decentralized aspect of our procedure is based on a process of knowledge propagation through the network via correspondences. This process allows to reduce polynomially global reasoning to local reasoning.

1 Introduction

Reasoning on a network of aligned ontologies has been investigated in different contexts where the semantics given to correspondences differs from one to another. To be able to develop a procedure for reasoning on a network of aligned ontologies, it is needed to equip the correspondences of the alignment with a semantics compatible with those defined in the ontologies. A simple approach to this issue consists in considering the correspondences as logical axioms expressed in the ontology language and merging all involved ontologies and the alignments into a unique ontology. In this case, the reasoning problem on such a network of aligned ontologies can be expressed as the following usual entailment :

$$\bigcup_{1\leq i\leq n} O_i \cup \bigcup_{1\leq i<j\leq n} A_{ij} \models \alpha \qquad (1)$$

where O_i is an $\mathcal{ALC}$ ontology, A_{ij} is an alignment between O_i and O_j, and α [1] is a link key or a concept assertion/axiom.

This approach is characterized by the following two main aspects : (i) the correspondences of the alignments are semantically handled as ontology assertion/axioms, and (ii) reasoning is performed on the unique ontology in a centralized manner, *i.e.* all reasoning tasks are carried out on a single location with a reasoner. Such an approach is quite unexploitable in the context of the Web where numerous ontologies and alignments are located in different sites. There have been researches [4, 6, 3, 13, 14, 1], which aimed at distributing reasoning over several locations. However, these approaches usually lead to an exponential blow-up of message passing between local reasoners associated with different locations. The main reason for this exponential blow-up is due to the strong semantics of the correspondences involved in the alignments.

In this paper, we introduce a new semantic of correspondences which are weaker than the usual ones and propose a

1. Consistency of the network can be reduced to the entailment (1) with $\alpha = \bot(x)$

procedure for reasoning on a network of aligned ontologies in a decentralized manner–that means–reasoning can be independently performed on different sites following a process of knowledge propagation through the network of the ontologies via the alignments with link keys. Usefulness of link keys in Semantic Web applications and the problem of reasoning with them in the centralized context have been investigated by Atencia and Gmati [2, 5].

To illustrate our settings, we consider the following example in which knowledge is modelled in description logics. This formalism is used to encode the semantics of web languages such as OWL2.

Example 1. *Consider two ontologies, denoted O_1 and O_2, where O_1 describes a terminology used by conference organizers, and O_2 stores information about researchers and conferences they have attended. In O_1, there are classes* Participant, Presenter, DemoPaperPresenter*; and a property* present*. In O_2, we can find classes* Researcher, PhDStudent, Developer*; and a property* registerTo *(i.e. someone registers to present a paper).*

An alignment A_{12} tells us that DemoPaperPresenter *is simultaneously aligned with* Researcher *and* Developer*.*

$$\mathsf{DemoPaperPresenter} \rightarrow \mathsf{Researcher} \quad (2)$$

$$\mathsf{DemoPaperPresenter} \rightarrow \mathsf{Developer} \quad (3)$$

In addition, A_{12} contains a link key which says that if a participant presents in the conference the same paper as that to which a researcher registers the conference then the participant and the researcher would be the same person.

$$\{\langle \mathsf{present}, \mathsf{registerTo}\rangle\} \text{ linkkey } \langle \mathsf{Participant}, \mathsf{Researcher}\rangle \quad (4)$$

If we now add to O_1 and O_2 the following axioms/assertion

$$O_1 : \mathsf{DemoPaperPresenter}(\mathsf{Anna}) \quad (5)$$

$$O_1 : \mathsf{DemoPaperPresenter} \sqsubseteq \mathsf{Participant} \quad (6)$$

$$O_2 : \mathsf{PhDStudent} \sqsubseteq \mathsf{Researcher} \quad (7)$$

$$O_2 : \mathsf{Researcher} \sqsubseteq \neg\mathsf{Developer} \quad (8)$$

then a reasoner can find the entailment :

$$O_1 \cup O_2 \cup A_{12} \models \{\langle \mathsf{present}, \mathsf{registerTo}\rangle\} \text{ linkkey } \langle \mathsf{DemoPaperPresenter}, \mathsf{PhDStudent}\rangle \quad (9)$$

This entailment holds because of the axioms (6), (7) and the link key (4). If we now interpret the correspondences (2) and (3) as subsumption in the standard semantics then the network $O_1 \cup O_2 \cup A_{12}$ is inconsistent because of the assertion/axiom (5) and (8). However, if we interpret these correspondences as a means for propagating concept unsatisfiability, i.e. unsatisfiability of the "subsumer" implies unsatisfiability of the "subsumee", then the network is consistent. In the following sections, we show that the weakened semantics corresponding to the latter interpretation of concept correspondences leads to a substantial change of the computational complexity of algorithms for reasoning.

In addition, the weakened semantics would not be really interesting for the correspondences (2) and (3). However, it would be more relevant for correspondences between ontologies of different nature. Given two ontologies about `equipment` *and* `staff` *and a correspondence* Computer $\rightarrow$ Developer *between them. With this correspondence, the weakened semantics tells us that if there is no developer then there is no computer. The standard semantics is irrelevant in this case.* □

Based on the weakened semantics of alignments, we introduce in this paper the notion of consistency for a network of ontologies with alignments containing link keys (or an *ontology network* for short). Then, we propose an algorithm for checking consistency of an ontology network by reducing this task to checking consistency of each ontology which is polynomially extended. This consists in (i) propagating individual equalities of the form $a \approx b$ through all ontologies of the network via individual correspondences of the same form $a \approx b$, (ii) applying link keys in the alignments, which may lead to add new individual correspondences, (iii) propagating concept unsatisfiabilities through all ontologies of the network via concept correspondences of the form $C \rightarrow D$. We show that the complexity of the process of knowledge propagation is polynomial in the size of the network. In addition, we also prove that consistency of the ontologies and alignments extended by this process of knowledge propagation is equivalent to consistency of the network.

The remainder of the paper is organised as follows. Section 2 positions our work with respect to works on distributed reasoning in description logics. Section 3 describes the logic $\mathcal{ALC}$ with individuals, alignments, a new semantics of alignments and inference services. Section 4 provides the algorithms for propagating individual equalities, applying link keys and propagating concept satisfiabilities. We also prove that reasoning on the ontology network is reducible to reasoning on each ontology extended by the algorithms, and this reduction is polynomial in the size of the ontology network. Section 5 presents examples of the use of the algorithms. Section 6 describes the architecture of Draon in which the algorithms are implemented in a decentralized manner. We also report some experimental results. Section 7 concludes the paper and presents future work.

2 Related Work

In the literature, there have been several reasoning approaches which either (i) merge all ontologies and alignments into a unique ontology and perform reasoning

over that unique ontology, or (ii) use a distributed semantics such as DDL (Distributed Description Logics) [4], E-connection [6], IDDL (Integrated Distributed Description Logics) [13] , Package-based Description Logics [3] and design a distributed algorithm for reasoning. The second option consists in defining new formalisms which allow reasoning with multiple domains in a distributed way. The new semantics of these formalisms reconcile conflicts between ontologies, but they do not adequately formalize the quite common case of ontologies related with ontology alignments produced by third party ontology matchers. Indeed, these formalisms assert cross-ontology correspondences (bridge rules, links or imports) from one ontology's point of view, while often, such correspondences are expressed from a point of view that encompasses both aligned ontologies. Another issue of these non-standard semantics is that reasoners such as Drago [12], Pellet [10], an early version of Draon [8] using the distributed algorithms resulting from the corresponding semantics require an exponential number of message exchanges over network. This exponential blow-up results from exchanging model portions (the so-called distributed tableau) between modules of the reasoner located on different sites.

Recenty, Atencia and Gmati [2, 5] have proposed a tableau algorithm for reasoning in the centralized context on an $\mathcal{ALC}$ ontology with link keys. They have showed that adding link keys to $\mathcal{ALC}$ does not augment the complexity of the tableau algorithm.

3 Preliminaries

The syntax and semantics of the logic $\mathcal{ALC}$ are defined below.

Definition 1 (Syntax of $\mathcal{ALC}$)**.** *Let* **C***,* **R** *and* **I** *be non-empty sets of* concept names*,* role names *and* individuals*, respectively. The set of* $\mathcal{ALC}$-concepts *(or simply concepts) is the smallest set such that every concept name in* **C***,* $\top$ *and* $\bot$ *are concepts, and if* C, D *are concepts and* R *is a role name in* **R** *then* $C \sqcap D$*,* $C \sqcup D$*,* $\neg C$*,* $\forall R.C$ *and* $\exists R.C$ *are concepts. A* general concept inclusion *(GCI) is an expression of the form* $C \sqsubseteq D$ *where* C, D *are concepts. A* terminology *or* TBox *is a finite set of GCIs. An* ABox assertion *is an expression of the form* $C(a)$*,* $R(a, b)$*,* $a \approx b$ *or* $a \not\approx b$ *where* C *is a concept,* R *is a role name in* **R** *and* a, b *are individuals in* **I***. An* ABox *is a finite set of ABox assertions. A pair* $O = (\mathcal{A}, \mathcal{T})$*, where* $\mathcal{T}$ *is a TBox and* $\mathcal{A}$ *is an ABox, is called an* $\mathcal{ALC}$ ontology*. We use* $\mathsf{Voc}_I(O)$*,* $\mathsf{Voc}_C(O)$ *and* $\mathsf{Voc}_R(O)$ *to denote the sets of individuals, concept names and role names occurring in* O*.*

Definition 2 (Semantics of $\mathcal{ALC}$)**.** *An* interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *is composed of a non-empty set* $\Delta^{\mathcal{I}}$*, called the* domain *of* $\mathcal{I}$*, and a* valuation $\cdot^{\mathcal{I}}$ *which maps every concept name to a subset of* $\Delta^{\mathcal{I}}$*, every role name to a subset of* $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ *and each individual to an element of* $\Delta^{\mathcal{I}}$*. The valuation is extended to constructed concepts such that, for all concepts* C, D *and role name* R*, the following is satisfied :*

$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}, \bot^{\mathcal{I}} = \emptyset, (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$

$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}, (C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$

$(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \forall y.\langle x, y\rangle \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$

$(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y.\langle x, y\rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$

An interpretation $\mathcal{I}$ satisfies *a* GCI $C \sqsubseteq D$*, denoted by* $\mathcal{I} \models C \sqsubseteq D$*, if* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$*.* $\mathcal{I}$ *is a* model *of a* TBox $\mathcal{T}$ *if* $\mathcal{I}$ *satisfies every GCI in* $\mathcal{T}$*. An interpretation* $\mathcal{I}$ *satisfies the ABox assertions :* $C(a)$ *if* $a^{\mathcal{I}} \in C^{\mathcal{I}}$*;* $R(a, b)$ *if* $\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$*;* $a \approx b$ *if* $a^{\mathcal{I}} = b^{\mathcal{I}}$*;* $a \not\approx b$ *if* $a^{\mathcal{I}} \neq b^{\mathcal{I}}$*. Given an ABox assertion* α*,* $\mathcal{I} \models \alpha$ *denotes that* $\mathcal{I}$ *satisfies* α*.* $\mathcal{I}$ *is a model of an ABox* $\mathcal{A}$ *if it satisfies every ABox assertion in* $\mathcal{A}$*. An interpretation* $\mathcal{I}$ *is a model of an* $\mathcal{ALC}$ *ontology* $O = (\mathcal{A}, \mathcal{T})$ *if* $\mathcal{I}$ *is a model of* $\mathcal{T}$ *and* $\mathcal{A}$*. An ontology* O *is* consistent *if there exists a model of* O*.* O *entails alpha, written* $O \models \alpha$*, where* α *is a GCI or an assertion, if every model of* O *satisfies* α*.*

We need notations and definitions that will be used in the paper. We use $|S|$ to denote the cardinality of a set S. Given an $\mathcal{ALC}$ ontology $O = \langle \mathcal{A}, \mathcal{T}\rangle$, we denote by $\mathsf{sub}(O) = \mathsf{sub}(\mathcal{A}, \mathcal{T})$ the set of all sub-concepts occurring in $\mathcal{A}, \mathcal{T}$. The size of an ontology O is denoted by $|O| = |\mathcal{A}| + |\mathcal{T}|$ where $|\mathcal{A}|$ is the size (number) of all assertions, $|\mathcal{T}|$ the size of all GCIs. It holds that $|\mathsf{sub}(O)|$ is polynomially bounded by $|O|$ since if a concept is represented by a string then a sub-concept is a substring.

To be able to define a network of aligned ontologies, we need alignments which represent semantic links between ontology entities such as individuals, concepts or roles.

Definition 3 (network of aligned ontologies)**.** *An* $\mathcal{ALC}$ *network of aligned ontologies is a tuple* $\langle \{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ *where* O_i *is an* $\mathcal{ALC}$ *ontology with* $1 \leq i \leq n$*, and* A_{ij} *with* $1 \leq i < j \leq n$ *is an alignment containing correspondences of the following forms :*

- $C \rightarrow D$ *or* $C \leftarrow D$ *where* $C \in \mathsf{sub}(O_i)$ *and* $D \in \mathsf{sub}(O_j)$*. Such a correspondence is called concept correspondence.*
- $a \approx b$ *(*$a \not\approx b$*) where* $a \in \mathsf{Voc}_I(O_i)$ *and* $b \in \mathsf{Voc}_I(O_j)$*. Such a correspondence is called individual correspondence.*
- *a link key* $\{\langle P_k, Q_k\rangle\}_{k=1}^n \mathsf{linkkey}\langle C, D\rangle$ *where* $P_k \in \mathsf{Voc}_R(O_i)$*,* $Q_k \in \mathsf{Voc}_R(O_j)$ *for* $1 \leq k \leq n$*,* $C \in \mathsf{sub}(O_i)$ *and* $D \in \mathsf{sub}(O_j)$*. Such a correspondence is called link key correspondence.*

The following definition formalizes the semantics of correspondences in an alignment so that it is compatible with that of ontologies. We retain the standard semantics for individual and link key correspondences while the semantics of concept correspondences is weakened.

Definition 4 (semantics of alignments)**.** *An $\mathcal{ALC}$ network of aligned ontologies is a tuple $\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ where O_i is an $\mathcal{ALC}$ ontologies with $1 \leq i \leq n$, and A_{ij} is an alignment with $1 \leq i < j \leq n$. Let $\mathcal{I}$ and $\mathcal{J}$ be models of O_i and O_j respectively.*

- *If $C \rightarrow D$ is in A_{ij} then $D^{\mathcal{J}} = \emptyset$ implies $C^{\mathcal{I}} = \emptyset$.*
- *If $a \approx b$ is in A_{ij} then $a^{\mathcal{I}} = a^{\mathcal{J}}$.*
- *If $a \not\approx b$ is in A_{ij} then $a^{\mathcal{I}} \neq a^{\mathcal{J}}$.*
- *If $\{\langle P_k, Q_k\rangle\}_{k=1}^n$ linkkey$\langle C, D\rangle$ is in A_{ij} then $(a_k^i)^{\mathcal{I}} = (a_k^j)^{\mathcal{J}}$, $\langle a^{\mathcal{I}}, (a_k^i)^{\mathcal{I}}\rangle \in P_k^{\mathcal{I}}$, $\langle b^{\mathcal{J}}, (a_k^j)^{\mathcal{J}}\rangle \in Q_k^{\mathcal{J}}$ for all $1 \leq k \leq n$, $a^{\mathcal{I}} \in C^{\mathcal{I}}$, $b^{\mathcal{J}} \in D^{\mathcal{J}}$ imply $a^{\mathcal{I}} = b^{\mathcal{J}}$.*

The notion of consistency for a network of aligned ontologies can be naturally introduced thanks to the semantics of ontologies and alignments involved in the network.

Definition 5 (network consistency)**.** *Let $\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ be a network of aligned ontologies in $\mathcal{ALC}$. The network is consistent if there is a model $\mathcal{I} = \{\mathcal{I}_i\}_{i=1}^n$ of O_i for all $1 \leq i \leq n$ such that*

1. *For each correspondence $a \approx b$ in A_{ij} with $1 \leq i < j \leq n$, $a^{\mathcal{I}_i} = b^{\mathcal{I}_j}$. For each correspondence $a \not\approx b$ in A_{ij} with $1 \leq i < j \leq n$, $a^{\mathcal{I}_i} \neq b^{\mathcal{I}_j}$.*
2. *There are no pair of correspondences $a \approx b, a \not\approx b$ in A_{ij}. In this case, we say that A_{ij} is clash-free.*
3. *For each correspondence $C \rightarrow D$ in A_{ij} with $1 \leq i < j \leq n$, if $D^{\mathcal{I}_j} = \emptyset$ then $C^{\mathcal{I}_i} = \emptyset$.*
4. *For each correspondence $\{\langle P_k, Q_k\rangle\}_{k=1}^n$ linkkey$\langle C, D\rangle$ in A_{ij} with $1 \leq i < j \leq n$, if $(a_k^i)^{\mathcal{I}_i} = (a_k^j)^{\mathcal{I}_j}$, $\langle a^{\mathcal{I}_i}, (a_k^i)^{\mathcal{I}_i}\rangle \in P_k^{\mathcal{I}_i}$, $\langle b^{\mathcal{I}_j}, (a_k^j)^{\mathcal{I}_j}\rangle \in Q_k^{\mathcal{I}_j}$ for all $1 \leq k \leq n$, $a^{\mathcal{I}_i} \in C^{\mathcal{I}_i}$, $b^{\mathcal{I}_j} \in D^{\mathcal{I}_j}$ then $a^{\mathcal{I}_i} = b^{\mathcal{I}_j}$.*

A network $N = \langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ entails *a link key α, written $N \models \alpha$, if every model of N satisfies α.*

We finish this section by proving the following lemma which allows to reduce link key entailment to consistency of the network of aligned ontologies.

Lemma 1 (Reduction of link key entailment to consistency)**.** *Let $\langle\{O_1, O_2\}, A_{12}\rangle$ be a network of aligned ontologies in $\mathcal{ALC}$. It holds that*

$$\langle\{O_1, O_2\}, A_{12}\rangle \models (\{\langle P_i, Q_i\rangle\}_{i=1}^m \text{ linkkey } \langle C, D\rangle) \text{ iff } \langle\{O_1', O_2'\}, A_{12}'\rangle \text{ is inconsistent}$$

with $O_1' = O_1 \cup \{C(x)\} \cup \{P_i(x, z_i)\}_{i=1}^n$, $O_2' = O_2 \cup \{D(y)\} \cup \{Q_i(y, z_i')\}_{i=1}^n$, $A_{12}' = A_{12} \cup \{z_i \approx z_i'\}_{i=1}^n \cup \{x \not\approx y\}$, $x, z_1, \cdots, z_n$ are new individuals in O_1 and $y, z_1', \cdots, z_n'$ are new individuals in O_2.

Proof. Let $\lambda = \{\langle P_i, Q_i\rangle\}_{i=1}^n$ linkkey $\langle C, D\rangle$. Assume that $\langle\{O_1, O_2\}, A_{12}\rangle \models \lambda$. Let us show that $\langle\{O_1', O_2'\}, A_{12}'\rangle$ is inconsistent. By contradiction, assume that $\langle\{O_1', O_2'\}, A_{12}'\rangle$ has a model $\mathcal{I} = \langle\mathcal{I}_1, \mathcal{I}_2\rangle$, i.e. O_1' and O_2' have models $\mathcal{I}_1$ and $\mathcal{I}_2$ satisfying Definition 5. This implies that $\mathcal{I}_1$ and $\mathcal{I}_2$ are models of O_1 and O_2. That means that $x^{\mathcal{I}_1} \in C^{\mathcal{I}_1}$, $y^{\mathcal{I}_2} \in D^{\mathcal{I}_2}$, $\langle x^{\mathcal{I}_1}, z_i^{\mathcal{I}_1}\rangle \in P_i^{\mathcal{I}_1}$, $\langle y^{\mathcal{I}_2}, z'^{\mathcal{I}_2}_i\rangle \in Q_i^{\mathcal{I}_2}$, $z_i^{\mathcal{I}_1} = z'^{\mathcal{I}_2}_i$ and $x^{\mathcal{I}_1} \neq y^{\mathcal{I}_2}$. This implies that $\mathcal{I} \not\models \lambda$. Thus, we have a model $\mathcal{I}$ of $\langle\{O_1', O_2'\}, A_{12}'\rangle$ such that $\mathcal{I} \not\models \lambda$. Therefore, $\langle\{O_1', O_2'\}, A_{12}'\rangle \not\models \lambda$, which contradicts the assumption.

Assume now that $\langle\{O_1', O_2'\}, A_{12}'\rangle \not\models \lambda$. Let us show that $\langle\{O_1', O_2'\}, A_{12}'\rangle$ is consistent. Since $\langle\{O_1', O_2'\}, A_{12}'\rangle \not\models \lambda$, then there exists an interpretation $\mathcal{I} = \langle\mathcal{I}_1, \mathcal{I}_2\rangle$ such that $\mathcal{I} \models \langle\{O_1', O_2'\}, A_{12}'\rangle$ and $\mathcal{I} \not\models \lambda$.

Since $\mathcal{I} \not\models \lambda$, by the semantics of link keys, there exist $\delta, \delta_1, \ldots, \delta_n \in \Delta_1^{\mathcal{I}_1}$ and $\delta', \delta_1', \ldots, \delta_n' \in \Delta_2^{\mathcal{I}_2}$ such that $\delta \in C^{\mathcal{I}_1}$, $\delta' \in D^{\mathcal{I}_2}$, $(\delta, \delta_1) \in P_1^{\mathcal{I}_1}, (\delta', \delta_1') \in Q_1^{\mathcal{I}_2}, \ldots, (\delta, \delta_n) \in P_n^{\mathcal{I}_1}, (\delta', \delta_n) \in Q_n^{\mathcal{I}_2}$, $\delta_1 = \delta_1', \ldots, \delta_n = \delta_n'$ and $\delta \neq \delta'$. Let us extend $\mathcal{I}$ by defining $x^{\mathcal{I}_1} = \delta$, $y^{\mathcal{I}_2} = \delta'$, $z_1^{\mathcal{I}_1} = \delta_1, \ldots, z_n^{\mathcal{I}_1} = \delta_n$, $z'^{\mathcal{I}_2}_1 = \delta_1', \ldots, z'^{\mathcal{I}_2}_n = \delta_n'$. Then, $\mathcal{I}$ is a model of $\langle\{O_1', O_2'\}, A_{12}'\rangle$. Therefore, $\langle\{O_1', O_2'\}, A_{12}'\rangle$ is consistent. □

This lemma can be extended to a general network of aligned ontologies containing more than two ontologies.

4 An algorithm for a network of aligned ontologies

The algorithm for deciding consistency of a network of aligned ontologies deals with pair by pair of ontologies in the network. For each pair of ontologies and an alignment between them, the algorithm repeats the following three tasks : propagating individual equalities from one ontology to the other via individual correspondences ; applying link key correspondences which may lead to the addition of new individual correspondences ; and propagating concept unsatisfiabilities from an ontology to the other via concept correspondences. The execution of a task may trigger the execution of another task. The execution of these tasks may lead to a change of ontologies and alignments in the network. The algorithm terminates on the pair of ontologies when the ontologies and the alignment reach stationarity. The first and second tasks are described in Algorithm 1 while the third one is outlined in Algorithm 2.

The following lemma establishes that the propagation performed by Algorithms 1 and 2 and the consistency of the pair of the extended ontologies suffice to decide consistency of the network composed of the initial ontologies and the alignment.

Lemma 2 (reduction for a pair)**.** *Let O_1, O_2 be two consistent ontologies and A_{12} be an alignment. We use $\widehat{O_1}$ and $\widehat{O_2}$ to denote the resulting consistent ontologies obtained by calling* propagatePair*(O_1, O_2, A_{12}). It holds that $\widehat{O_1}, \widehat{O_2}$ are consistent and $\widehat{A_{12}}$ is clash-free iff the network $\langle\{O_1, O_2\}, \{A_{12}\}\rangle$ is consistent.*

Before providing a complete proof of the lemma, we summarize the main arguments. The soundness of the if-

Algorithm 1 Propagating individual equalities

1: **function** PROPAGATEEQUAL(O_i, O_j, A_{ij})
2: **while** A_{ij} or O_i or O_j is unstationary **do**
3: **if** O_i or O_j is inconsistent or A_{ij} is not clash-free **then**
4: **return false**
5: **end if**
6: **for** $a_i^1 \approx a_j^1 \in A_{ij}, a_i^2 \approx a_j^2 \in A_{ij}$ **do**
7: **for** $O_k \models a_k^m \approx a_k^h, k \in \{i, j\}, m, h \in \{1, 2\}, m \neq h$ **do**
8: $A_{ij} \leftarrow A_{ij} \cup \{a_i^h \approx a_j^m, a_i^m \approx a_j^h\}$
9: $O_k \leftarrow O_k \cup \{a_k^1 \approx a_k^2\}$
10: **end for**
11: **end for**
12: **for each** $\{\langle P_k, Q_k\rangle\}_{k=1}^n \mathsf{linkkey}\langle C, D\rangle \in A_{ij}$ **do**
13: **for** $a_k^i \approx a_k^j \in A_{ij}, a \in \mathsf{Voc}_I(O_i), b \in \mathsf{Voc}_I(O_j), P_k(a', a'^i_k) \in O_i,$
14: $Q_k(b', a'^j_k) \in O_j, O_i \models a \approx a', O_i \models a_k^i \approx a'^i_k, O_j \models b \approx b'$
15: $O_j \models a_k^j \approx a'^j_k$ for all $1 \leq k \leq n$ **do**
16: **if** $O_i \cap \{C(a), \sim C(a)\} = \emptyset$ **then**
17: $O_i \leftarrow O_i \cup \{(C \sqcup \sim C)(a)\}$
18: **end if**
19: **if** $O_j \cap \{D(b), \sim D(b)\} = \emptyset$ **then**
20: $O_j \leftarrow O_j \cup \{(D \sqcup \sim D)(b)\}$
21: **end if**
22: **end for**
23: **for** $a_k^i \approx a_k^j \in A_{ij}, O_i \models C(a), O_j \models D(b), P_k(a', a'^i_k) \in O_i,$
24: $Q_k(b', a'^j_k) \in O_j, O_i \models a \approx a', O_i \models a_k^i \approx a'^i_k, O_j \models b \approx b'$
25: $O_j \models a_k^j \approx a'^j_k$ for all $1 \leq k \leq n$ **do**
26: $A_{ij} \leftarrow A_{ij} \cup \{a \approx b\}$
27: **end for**
28: **end for**
29: **end while**
30: **return true**
31: **end function**

direction of Lemma 2 is straightforward since Algorithms 1 and 2 add only logical consequences of the network to the ontologies and alignments. The soundness of the only-if-direction of the lemma is based on the following elements : (i) consistency of the extended ontologies and clash-freeness of the extended alignments imply consistency of the initial ontologies and clash-freeness of the initial alignements ; (ii) Algorithms 1 and 2 make implicit all individual equalities, and thus potential clashes of the kind $a \approx b, a \not\approx b$ must be discovered. This ensures that two models of the extended ontologies satisfy individual correspondences ; (iii) Algorithms 1 and 2 apply link keys until they are not applicable over the initial individuals in the ontologies. Since models of an $\mathcal{ALC}$ ontology are tree-shaped and $\mathcal{ALC}$ does not allow for inverse roles, satisfaction of the link keys over the initial individuals is sufficient ; and (iv) Algorithms 1 and 2 propagate concept unsatisfiabilities. If the “subsumer" of a concept correspondence is satisfiable then a model of the ontology can be extended such that the interpretation of the subsumer in this model is not empty. This implies that the concept correspondence is satisfied.

Proof. “If-direction". Assume that the network $\langle\{O_1, O_2\}, \{A_{12}\}\rangle$ is consistent. By definition, O_i has a model $\mathcal{I}_i$ with $1 \leq i \leq 2$ such that they satisfy all correspondences $\alpha \in A_{12}$. We use $\widehat{A_{ij}}$ to denote the resulting alignment obtained by calling propagateEqualities(O_1, O_2, A_{12}) and propagateUnsat(O_1, O_2, A_{12}). We show that $\mathcal{I}_1$ is a model of $\widehat{O_1}$. For this, we have to prove that :

• $a_0^{\mathcal{I}_1} = a_n^{\mathcal{I}_1}$ if $a_0 \approx a_n$ is added to O_1 by Line 9 in Algorithm 1 (this implies that there is no clash of the kind $a^{\mathcal{I}_1} = b^{\mathcal{I}_1}$, $a^{\mathcal{I}_1} \not\approx b^{\mathcal{I}_1}$). We have $a_0 \approx a_n$ is added to O_1 if there is a sequence of equalities $a_0 \approx a_1, \cdots, a_{n-1} \approx a_n$ such that $a_i \approx a_{i+1} \in \widehat{O_1} \cup \widehat{O_2} \cup \widehat{A_{12}}$ for $0 \leq i \leq n-1$. This sequence of equalities implies $a_0^{\mathcal{I}_1} = a_n^{\mathcal{I}_1}$. By using the same argument, we can show $a_0^{\mathcal{I}_2} = a_n^{\mathcal{I}_2}$ if $a_0 \approx a_n$ is added to O_2 by Line 9 in Algorithm 1.

Algorithm 2 Propagating concept unsatisfiability

1: **function** PROPAGATEUNSAT(O_i, O_j, A_{ij})
2: **while** A_{ij} or O_i or O_j is unstationary **do**
3: **if** O_i or O_j is inconsistent or A_{ij} is not clash-free **then**
4: **return false**
5: **end if**
6: **for** $C_i^1 \rightarrow C_j^1 \in A_{ij}, C_i^2 \leftarrow C_j^2 \in A_{ij}$ **do**
7: **for** $O_j \models C_j^1 \sqsubseteq C_j^2$ **do**
8: $A_{ij} \leftarrow A_{ij} \cup \{C_i^1 \rightarrow C_j^2, C_i^2 \leftarrow C_j^1\}$
9: **end for**
10: **end for**
11: **for** $C_i^1 \leftarrow C_j^1 \in A_{ij}, C_i^2 \rightarrow C_j^2 \in A_{ij}$ **do**
12: **for** $O_i \models C_i^1 \sqsubseteq C_i^2$ **do**
13: $A_{ij} \leftarrow A_{ij} \cup \{C_i^1 \rightarrow C_j^2, C_i^2 \leftarrow C_j^1\}$
14: **end for**
15: **end for**
16: **for each** $D \rightarrow C \in A_{ij}$ **do**
17: **if** $O_j \models C \sqsubseteq \bot$ **then**
18: $O_i \leftarrow O_i \cup \{D \sqsubseteq \bot\}$
19: **end if**
20: **end for**
21: **for each** $D \leftarrow C \in A_{ij}$ **do**
22: **if** $O_i \models D \sqsubseteq \bot$ **then**
23: $O_j \leftarrow O_j \cup \{C \sqsubseteq \bot\}$
24: **end if**
25: **end for**
26: **end while**
27: **return true**
28: **end function**

• $C_0^{\mathcal{I}_1} = \emptyset$ if $C_0 \sqsubseteq \bot$ is added to O_1 by Line 18 in Algorithm 2 (this implies that there is no clash of the kind $a^{\mathcal{I}_1} \in C_0^{\mathcal{I}_1}$, $C_0^{\mathcal{I}_1} = \emptyset$). We have $C_0 \sqsubseteq \bot$ is added to O_1 if there is a sequence $C_0 \Rightarrow C_1, \cdots, C_{n-1} \Rightarrow C_n$ such that $\widehat{O_1} \models C_i \Rightarrow C_{i+1}$ or $\widehat{O_2} \models C_i \Rightarrow C_{i+1}$ or $C_i \Rightarrow C_{i+1} \in \widehat{A_{12}}$ for $0 \leq i \leq n-1$, and $\widehat{O_i} \models C_n^{\mathcal{I}_i} \sqsubseteq \bot$ ($i \in \{1, 2\}$) where "$\Rightarrow$" represents "$\rightarrow$" or "$\sqsubseteq$" and $C \leftarrow D = D \Rightarrow C$, $C \sqsupseteq D = D \Rightarrow C$. This implies $C_i^{\mathcal{I}_i} = \emptyset$ for $1 \leq i \leq n$. By using the same argument, we can show $C_0^{\mathcal{I}_2} = \emptyset$ if $C_0 \sqsubseteq \bot$ is added to O_2 by Line 23 in Algorithm 2.

• The concepts $(C \sqcup \sim C)(a)$ and $(D \sqcup \sim D)(b)$ added by Lines 17 and 20 in Algorithm 2 do not change consistency since they are tautologies.

"Only-If-direction". Since $\widehat{O_i}$ is consistent, according to [7], $\widehat{O_i}$ has a tree-shaped model $\mathcal{I}_i$ where each interpretation domain Δ_i of $\mathcal{I}_i$ is composed of a set of initial individuals I_{old}^i and a set of new individuals I_{new}^i for $1 \leq i \leq 2$. Since $O_i \subseteq \widehat{O_i}$, $\mathcal{I}_i$ is a model of O_i with $1 \leq i \leq 2$. We will extend $\mathcal{I}_1$ and $\mathcal{I}_2$ so that they satisfy the correspondences in A_{12}.

• For each $a \approx b \in \widehat{A_{12}}$, we define $a^{\mathcal{I}_1} = a^{\mathcal{I}_2}$. Thus, $a^{\mathcal{I}_1} = a^{\mathcal{I}_2}$ for each $a \approx b \in A_{12}$ since $A_{12} \subseteq \widehat{A_{12}}$. By construction, $\mathcal{I}_1$ and $\mathcal{I}_2$ satisfy all of the individual correspondences in A_{12} according to Definition 4.

• If $a \not\approx b \in \widehat{A_{12}}$ then $a \approx b \notin \widehat{A_{12}}$ since $\widehat{A_{12}}$ is clash-free.

• Let $C_h \rightarrow D_h \in A_{12}$. If $\widehat{O_2} \models D_h \sqsubseteq \bot$ then $C_h \sqsubseteq \bot$ is added to $\widehat{O_1}$ by Algorithm 2. Hence, $D_h^{\mathcal{I}_2} = \emptyset$ implies $C_h^{\mathcal{I}_1} = \emptyset$. Note that if $\widehat{O_2} \models D_h \sqsubseteq \bot$ then $\widehat{O_2'} \models D_h \sqsubseteq \bot$ for all $\widehat{O_2} \subseteq \widehat{O_2'}$.

Assume that $\widehat{O_2} \not\models D_h \sqsubseteq \bot$. Thus, $\widehat{O_2} \cup \{D_h(x_h)\}$ is consistent where x_h is a new individual. According to [7], $\widehat{O_2} \cup \{D_h(x_h)\}$ has a tree-shaped model $\mathcal{I}_2'$ of $\widehat{O_2} \cup \{D_h(x_h)\}$. We show that if $\widehat{O_2} \cup \{D_1(x_1)\}$ and $\widehat{O_2} \cup \{D_2(x_2)\}$ are consistent with new individuals x_1, x_2 then $\widehat{O_2} \cup \{D_1(x_1), D_2(x_2)\}$ is consistent. Indeed, running the standard tabaleau algorithm in [7] on $\widehat{O_2} \cup \{D_1(x_1)\}$ can build a set $\mathbf{T}$ of completion trees rooted at the initial individuals in $\widehat{O_2}$ and a completion tree T_{x_1} rooted at x_1. Analogously, if the standard tableau algorithm runs on $\widehat{O_2} \cup \{D_2(x_2)\}$, it can build a set $\mathbf{T}$' of completion trees rooted at the initial individuals in $\widehat{O_2}$ and a completion tree T_{x_2} rooted at x_2. All trees are clash-free and complete. Hence, the set of trees $\mathbf{T} \cup \{T_{x_1}, T_{x_2}\}$ would be built by the standard tableau algorithm when it runs on $\widehat{O_2} \cup \{D_1(x_1), D_2(x_2)\}$.

Therefore, we can run the standard tableau algorithm in [7] on $\widehat{O_2} \cup \{D_i(x_i)\}_{i=1}^m$ to obtain a tree-shaped model $\mathcal{J}_2$ of $\widehat{O_2} \cup \{D_i(x_i)\}_{i=1}^m$ where x_h is a new individual and $\widehat{O_2} \not\models D_h \sqsubseteq \bot$ for $1 \leq h \leq m$.

By using the same argument, we can obtain a tree-shaped model $\mathcal{J}_1$ of $\widehat{O_1} \cup \{D'_1(x'_1), \cdots, D'_{m'}(x'_{m'})\}$. By construction, $\mathcal{J}_1$ and $\mathcal{J}_2$ satisfy all of the concept correspondences in A_{12} according to Definition 4. In addition, they remain to satisfy all of the individual correspondences in A_{12}. For the sake of the simplicity, we rename $\mathcal{J}_1$ and $\mathcal{J}_2$ to $\mathcal{I}_1$ and $\mathcal{I}_2$.

• Assume that $\{\langle P_k, Q_k\rangle\}_{k=1}^n$ linkkey$\langle C, D\rangle$ is a link key in A_{12} and $(a_k^1)^{\mathcal{I}_1} = (a_k^2)^{\mathcal{I}_2}, \langle a^{\mathcal{I}_1}, (a_k^1)^{\mathcal{I}_1}\rangle \in P_k^{\mathcal{I}_1}, \langle b^{\mathcal{I}_2}, (a_k^2)^{\mathcal{I}_2}\rangle \in Q_k^{\mathcal{I}_2}$ for all $1 \leq k \leq n$, $a^{\mathcal{I}_1} \in C^{\mathcal{I}_1}$, $b^{\mathcal{I}_2} \in D^{\mathcal{I}_2}$.

1. If $(a_k^1)^{\mathcal{I}_1} = (a_k^2)^{\mathcal{I}_2}$ then there is a sequence $a_0 \approx a_1, \cdots, a_{m-1} \approx a_m$ such that $a_i \approx a_{i+1} \in \widehat{O_1} \cup \widehat{O_2} \cup \widehat{A_{12}}$ for $0 \leq i \leq m-1$ with $a_k^1 = a_0, a_k^2 = a_m$. This implies that $a_k^1 \approx a_k^2 \in \widehat{A_{12}}$ for $1 \leq k \leq n$.
2. Since $\mathcal{I}_1$ and $\mathcal{I}_2$ are tree-shaped whose roots are the old individuals, the condition of the link key holds only if all individuals a_k^1, a_k^2 for $1 \leq k \leq n$, and a, b are contained $I_{old}^1 \cup I_{old}^2$. Hence, $\langle a^{\mathcal{I}_1}, (a_k^1)^{\mathcal{I}_1}\rangle \in P_k^{\mathcal{I}_1}$ iff $P_k(a', a'^1_k) \in O_1$ with $O_i \models a \approx a'$, $O_i \models a_k^1 \approx a'^1_k$ for $1 \leq k \leq n$ where a, a' and a_k^1, a'^1_k are old individuals.
3. Since $a^{\mathcal{I}_1} \in C^{\mathcal{I}_1}$ and $(C \sqcup {\sim}C)(a) \in O_1$ (Line 17, Algorithm 1), we have $\widehat{O_1} \models C(a)$. Analogously, from $b^{\mathcal{I}_2} \in D^{\mathcal{I}_2}$ and $(D \sqcup {\sim}D)(b) \in O_2$ (Line 20, Algorithm 1), we obtain $\widehat{O_2} \models D(b)$.

Therefore, the 3 items above trigger Line 26 in Algorithm 1 which adds to $\widehat{A_{12}}$ the assertion $a \approx b$. Thus, we obtain $a^{\mathcal{I}_1} \approx b^{\mathcal{I}_2}$.

We have proven that $\mathcal{I}_1$ and $\mathcal{I}_2$ are models of O_1 and O_2 which satisfy all of the correspondences in A_{12}. □

We can observe that Algorithms 1 and 2 can be implemented in a decentralized manner since each call for checking ontology entailment or consistency can be sent to a local reasoner associated with the ontology located on a different site.

To check consistency of a network of aligned ontologies, it is needed to run Algorithms 1 and 2 on each pair of ontologies with the alignment between them until all ontologies and alignments are stationary. Note that saturating a pair of ontologies with the alignment can make a saturated pair of ontologies unsaturated. This is due to the fact that an ontology can be shared by several pairs of ontologies.

The following theorem is a consequence of Lemma 2.

Theorem 1 (reduction for network)**.** *Let $\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ be a network of aligned ontologies. We use $\widehat{O_i}$ ($1 \leq i \leq n$) to denote the resulting consistent ontologies obtained by calling* propagateOverNetwork$(\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle)$*. It holds that $\widehat{O_i}$ is consistent for all $1 \leq i \leq n$ and $\widehat{A_{ij}}$ is clash-free for all $1 \leq i < j \leq n$ iff the network $\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ is consistent.*

We now investigate the complexity of the algorithms. Under the hypothesis in which a call to reasoners associated with ontologies is considered as an oracle, *i.e.* an elementary operation, our algorithms are tractable.

Theorem 2. *Let $\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle$ be a network of aligned ontologies. The algorithm* propagateOverNetwork$(\langle\{O_i\}_{i=1}^n, \{A_{ij}\}_{i,j=1,i\neq j}^n\rangle)$ *runs in polynomial time in the size of the network if each check of entailment or consistency occurring in the algorithms is considered as an oracle.*

Proof. The complexity of propagateOverNetwork depends on the complexity of propagateEqual, propagateUnsat. When running these algorithms, each ontology can be monotonically extended. It is straightforward to obtain that the number of axioms of the form $C \sqsubseteq \bot$ added to ontologies O_i and O_j is bounded by a polynomial function in the size of initial alignments since C must occur in an initial correspondence. Analogously, the number of individuals correspondences $a \approx b$ added to alignments A_{ij} is bounded by a polynomial function in the size of initial alignments since a, b must occur in an initial correspondence. This implies that the number of iterations of the while loops in Algorithms 1, 2 and 3 is bound by a polynomial function in the size of initial alignments.

In addition, the number of iterations of the for loops in Algorithms 1, 2 and 3 is bounded by a polynomial function in the size of initial alignments, the size of ontologies and the number of ontologies and alignments included in the network. This observation completes the proof. □

5 Examples

This section provides some examples for showing how to use the algorithms presented in Section 4.

Example 2. *The ontologies and alignment in Example 1 can be rewritten as follows :*

O_1={$DP \sqsubseteq P, DP(a)$}, O_2={$PS \sqsubseteq R, R \sqsubseteq \neg D$},

A_{12}={$DP \rightarrow R, DP \rightarrow D, \langle pr, re\rangle$linkkey$\langle P, R\rangle$}

If the correspondences are considered as standard subsumptions then the ontology $O_1 \cup O_2 \cup A_{12}$ is inconsistent. Indeed, assume that there is a model $\mathcal{I} = \langle\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$ of the ontology. This implies that $a^{\mathcal{I}} \in DP^{\mathcal{I}}$, $DP^{\mathcal{I}} \subseteq R^{\mathcal{I}}$ and $DP^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Thus, $a^{\mathcal{I}} \in R^{\mathcal{I}} \cap D^{\mathcal{I}}$. However, we have $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \setminus D^{\mathcal{I}}$, which is a contradiction.

If we now interpret the correspondences under the semantics given in Definition 4 then there is no propagation needed according to Algorithms 1 and 2. It is obvious that

Algorithm 3 Complete propagation over the whole network

```
1: function propagateOverNetwork(⟨{O_i}_{i=1}^n, {A_ij}_{i,j=1,i≠j}^n⟩)
2:     while O_i, O_j, A_ij are unstationary for all 1 ≤ i < j ≤ n do
3:         for 1 ≤ i < j ≤ n do
4:             while O_i, O_j, A_ij are unstationary do
5:                 if propagateEqual(O_i, O_j, A_ij) returns false then
6:                     return false
7:                 end if
8:                 if propagateUnsat(O_i, O_j, A_ij) returns false then
9:                     return false
10:                end if
11:            end while
12:        end for
13:    end while
14:    return true
15: end function
```

O_1 and O_2 are consistent, and the network $\langle\{O_1, O_2\}, A_{12}\rangle$ is consistent under the semantics given in Definition 4.

Example 3. *In this example, we reduce two correspondences in Example 2 to one as follows.*

O_1=$\{DP \sqsubseteq P, DP(a)\}$, O_2=$\{PS \sqsubseteq R, R \sqsubseteq \neg D\}$,

A_{12}=$\{DP \rightarrow R \sqcap D, \langle pr, re\rangle\mathsf{linkkey}\langle P, R\rangle\}$

We now interpret the correspondence under the semantics given in Definition 4. Since $O_2 \models R \sqcap D \sqsubseteq \bot$, Algorithm 2 propagates unsatisfiability of $R \sqcap D$ to O_1 via the correspondence $DP \rightarrow R \sqcap D$. Hence, it adds $DP \sqsubseteq \bot$ to O_1. This leads to inconsistency of $\widehat{O_1}$. Therefore, the network $\langle\{O_1, O_2\}, A_{12}\rangle$ is not consistent.

Example 4. *The ontologies and alignment in Example 1 can be rewritten as follows :*

O_1=$\{DP \sqsubseteq P, DP(a)\}$, O_2=$\{PS \sqsubseteq R, R \sqsubseteq \neg D\}$,

A_{12}=$\{DP \rightarrow R, DP \rightarrow D, \langle pr, re\rangle\mathsf{linkkey}\langle P, R\rangle\}$

We consider whether $\langle\{O_1, O_2\}, A_{12}\rangle \models \lambda$ where $\lambda = \langle pr, re\rangle\mathsf{linkkey}\langle P, R\rangle$. Due to Lemma 1, we extend O_1, O_2 and A_{12} by adding to O_1 assertions $DP(x), pr(x, x_1)$, to O_2 assertions $PS(y), re(y, y_1)$, and to A_{12} assertions $x_1 \approx y_1, x \not\approx y$. Let $\widehat{O_1}, \widehat{O_2}$ and $\widehat{A_{12}}$ be the extended ontologies and alignment. If there are models $\mathcal{I}_1$ and $\mathcal{I}_2$ of $\widehat{O_1}, \widehat{O_2}$ then, we have $x \in DP^{\mathcal{I}_1}$ and $y \in PS^{\mathcal{I}_2}$, and $DP^{\mathcal{I}_1} \subseteq P^{\mathcal{I}_1}$ and $PS^{\mathcal{I}_2} \subseteq R^{\mathcal{I}_2}$.Thus, the link key $\langle pr, re\rangle\mathsf{linkkey}\langle P, R\rangle$ is applicable, and Algorithm 1 adds $x \approx y$ to $\widehat{A_{12}}$. This leads to a clash in $\widehat{A_{12}}$ and thus the network $\langle\{\widehat{O_1}, \widehat{O_2}\}, \widehat{A_{12}}\rangle$ is not consistent. Therefore, $\langle\{O_1, O_2\}, A_{12}\rangle \models \lambda$ holds.

6 Implementation and Experimental Results

An implementation of the proposed algorithms has been integrated within a reasoner written in Java, called Draon [8], which already allowed to reason in a decentralized manner on a network of aligned ontologies under the IDDL semantics [14]. Algorithms 1, 2 and 3 can be naturally implemented such that reasoning tasks on ontologies can be independently performed by different reasoners located on different sites.

The architecture of Draon is despicted in Figure 1. A global reasoner implements Algorithm 3. This global reasoner loads alignments and executes Algorithm 3. It sends assertions/axioms different to local reasoners located on different sites. Then it asks local reasoners to check entailment and consistency of the ontology associated with each local reasoner. The global reasoner and each local reasoner use HermiT [11] as OWL reasoner. The communication between the global reasoner and all local reasoner is based on OWLLink [9]. When connecting to a local reasoner, the global reasoner creates a Java thread which deals with the communication between them. Data shared by the threads are synchronized and protected by using semaphores. Note that we can replace HermiT with any OWL reasoner since OWLLink supports a generic OWL reasoner.

Table 1 provides information on the ontologies and alignments used for the experiments. These datasets are taken from OAEI2012 [2] and OAEI2018 [3] Campaigns. We have chosen small ontologies and alignments such as `iasted.owl`, `sigkdd.owl`, `iasted-sigkdd.rdf` to test our algorithm on alignments with link keys since they are well understood and manually checkable. This allows us to create manually relevant link keys (to our best knowledge, there is no system which can generate link keys expressed in the alignment syntax). In addition, we have selected large ontologies and alignment such as SNOMED, FMA, FMA-SNOMED in order that the difference between the reasoning complexities of the two semantics IDDL (implemented in Draon) and APPROX (the new semantics pre-

2. `cs.ox.ac.uk/isg/projects/SEALS/oaei/2012/`
3. `oaei.ontologymatching.org/2018/conference`

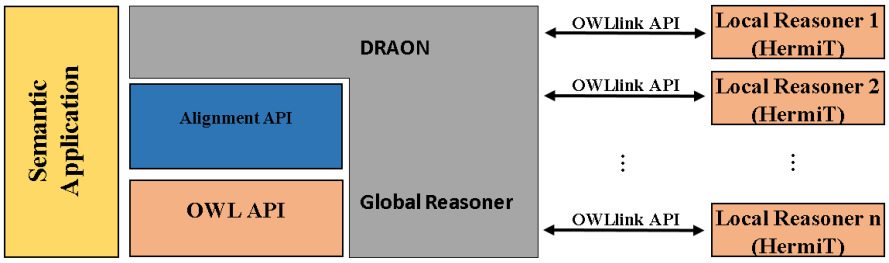


FIGURE 1 – Architecture of Draon

	Concepts	Roles	Individuals	Axioms/Correspondences
Iasted	141	38	6	551
Sigkdd	50	18	5	210
iast-sigkdd (without link keys)				15
Conference	60	46	2	414
Ekaw	74	33	4	351
conference-ekaw (without link keys)				27
Cmt	30	49	3	327
Edas	104	30	117	1025
cmt-edas (without link keys)				14
FMA	10157	0	0	47467
SNOMED	13412	18	0	47104
FMA-SNOMED (without link keys)				9139
NCI	25591	87	0	135556
FMA-NCI (without link keys)				3038

TABLE 1 – Ontologies and aligments without link keys and their characteristics

Ontology 1	Ontology 2	Alignment	IDDL	APPROX
Iasted	Sigkdd	iasted-sigkdd (without link keys)	3.5s	9 ms
Conference	Ekaw	conference-ekaw (without link keys)	7.5s	11 ms
Cmt	Edas	cmt-edas (without link keys)	7.5s	16 ms
FMA	SNOMED	FMA-SNOMED (without link keys)	> 15 minutes	81s
FMA	NCI	FMA-NCI (without link keys)	> 15 minutes	10s

TABLE 2 – Execution time for checking consistency of ontology networks according to different semantics

Ontology 1	Ontology 2	Alignment	Consistency in APPROX
Iasted	Sigkdd	iast-sigkdd (with link keys)	9 ms
Conference	Ekaw	conference-ekaw (with link keys)	11 ms
Cmt	Edas	cmt-edas (with link keys)	17 ms

TABLE 3 – Execution time (in milliseconds) for checking consistency of ontology networks with link keys

sented in the paper) is more noticeable.

We use two remote DELL servers with Intel 3.4GHz Processor 8 cores and 32Gb RAM on which two HermiT-based local reasoners are running. The global reasoner is also launched on a third computer with the same configuration.

We run Draon to check consistency of several networks of ontologies each of which is composed of ontologies and alignment described in Table 1. The results are put in Table 2 which shows execution times of Draon under the two different semantics IDDL and APPROX. The difference of the performances in time results from the fact that reasoning under IDDL may require in the worst case an exponential number of message exchanges between the global reasoner

and the local reasoners while reasoning under APPROX needs at most a polynomial number of message exchanges.

Table 3 provides first experimental results when running Draon to check consistency of networks containing small ontologies and alignment with link keys. The alignments in this table are obtained by adding to the corresponding alignments in Table 2 some link keys manually created.

7 Conclusion and Future Work

We have presented a new semantic of alignments which is weaker than the standard semantics. This weakened semantics of alignments allows us to express correspondences between ontologies of different nature on the one hand and to propose an efficient algorithm for reasoning on a network of ontologies with alignments containing link keys on the other hand. This new kind of correspondences is useful for establishing data links between heterogeneous datasets. The complexity of the proposed algorithm is polynomial in the size of the network if each call for checking ontology entailment or consistency is considered as an oracle. We have integrated an implementation of our algorithm within a distributed reasoner, called Draon, and reported some experimental results.

Our algorithm can be extended to deal with ontologies expressed in a more expressive Description Logic than $\mathcal{ALC}$ in condition that the new logic does not allow for inverse roles. This restriction on expressiveness prevents the current algorithm from merging individuals which are initially not in the ontology. Another extension of the current work aims at adding role correspondences to alignments. This may require the algorithm to support ontologies allowing for hierarchy of roles and the negation of roles. We plan to carry out experiments of Draon on ontologies and alignments located on a large number of nodes equipped with a local reasoner. New evaluations of Draon on alignments with a large number of link keys are also expected.

Acknowledgements

This work has been partially supported by the ANR project Elker (ANR-17-CE23-0007-01).

Références

[1] Adjiman, Philippe, Philippe Chatalic, François Goasdoué, Marie-Christine Rousset et Laurent Simon: *Distributed Reasoning in a Peer-to-Peer Setting : Application to the Semantic Web*. J. Artif. Intell. Res., 25 :269–314, 2006.

[2] Atencia, Manuel, Jérôme David et Jérôme Euzenat: *Data interlinking through robust linkkey extraction*. Dans *Proc. 21st european conference on artificial intelligence (ECAI)*, pages 15–20. IOS press, 2014.

[3] Bao, Jie, Doina Caragea et Vasant G Honavar: *A distributed Tableau Algorithm for Package-based Description Logics*. Dans *Proceedings of the ECAI Workshop on Context Representation and Reasoning*, 2006.

[4] Borgida, Alex et Luciano Serafini: *Distributed Description Logics : Assimilating information from peer sources*. Journal Of Data Semantics, (1) :153–184, 2003.

[5] Gmati, Maroua, Manuel Atencia et Jérôme Euzenat: *Tableau extensions for reasoning with link keys*. Dans *Proceedings of the 11th International Workshop on Ontology Matching*, pages 37–48, 2016.

[6] Grau, Bernardo Cuenca, Bijan Parsia et Evren Sirin: *Combining OWL Ontologies Using $\mathcal{E}$-connections*. Journal Of Web Semantics, 4(1), 2006.

[7] Horrocks, Ian, Ulrike Sattler et Stephan Tobies: *Reasoning with Individuals for the Description Logic SHIQ*. Dans *Proc. of the 17th Int. Conf. on Automated Deduction (CADE 2000)*, pages 482–496. Springer, 2000.

[8] Le Duc, Chan, Myriam Lamolle, Antoine Zimmermann et Olivier Curé: *DRAOn : A Distributed Reasoner for Aligned Ontologies*. Dans *Informal Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE-2013)*, pages 81–86, 2013.

[9] Liebig, Thorsten, Marko Luther, Olaf Noppens et Michael Wessel: *OWLlink*. Semantic Web, 2(1) :23–32, 2011.

[10] Serafini, Luciano et Andrei Tamilin: *DRAGO : Distributed Reasoning Architecture for the Semantic Web*. Dans *Proceedings of the Europeen Semantic Web Conference*, pages 361–376, 2005.

[11] Shearer, Rob, Boris Motik et Ian Horrocks: *HermiT : A Highly-Efficient OWL Reasoner*. Dans *Proc. of the 5th Int. Workshop on OWL : Experiences and Directions (OWLED 2008 EU)*, 2008.

[12] Sirin, Evren, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur et Yarden Katz: *Pellet : a pratical OWL-DL reasoner*. Journal of Web Semantics, 5(2) :51–53, 2007.

[13] Zimmermann, Antoine et Jérôme Euzenat: *Three semantics for distributed systems and their relations with alignment composition*. Dans *Proc. 5th International Semantic Web Conference (ISWC)*, pages 16–29, 2006.

[14] Zimmermann, Antoine et Chan Le Duc: *Reasoning with a Network of Aligned Ontologies*. Dans *RR*, pages 43–57, 2008.

Actes JIAF 2019

Exploiter la compétence de couples de cas pour améliorer le raisonnement à partir de cas par analogie

Jean Lieber[1] **Emmanuel Nauer**[1] **Henri Prade**[2]

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
[2] IRIT, CNRS & Université de Toulouse, 31062 Toulouse cedex 9, France
Jean.Lieber@loria.fr Emmanuel.Nauer@loria.fr Henri.Prade@irit.fr

Résumé

Une proportion analogique est une relation quaternaire qui se lit « a est à b ce que c est à d » et qui vérifie certaines propriétés de symétrie et de permutation. Cette relation, qui met en œuvre deux couples, est le fondement de l'approche de raisonnement à partir de cas appelée extrapolation analogique, qui consiste à remémorer trois cas formant une proportion analogique avec le problème à résoudre dans l'espace des problèmes, puis à trouver une solution à ce problème en résolvant une équation analogique dans l'espace des solutions. Ce travail étudie comment la notion de compétence des couples de cas sources peut être évaluée et utilisée pour améliorer l'extrapolation. Un prétraitement de la base de cas associe à chaque couple de cas une compétence sous la forme de deux scores : le support et la confiance du couple de cas, calculé à partir d'autres couples avec qui il forme une proportion analogique. Une évaluation dans un cadre booléen montre que l'utilisation des compétences des couples de cas améliore significativement les résultats d'un système de raisonnement à partir de cas par analogie.

Mots clefs : compétence, extrapolation, inférence analogique, proportion analogique, raisonnement à partir de cas

Abstract

An analogical proportion is a quaternary relation that is to be read "a is to b as c is to d", verifying some symmetry and permutation properties. As can be seen, it involves a pair of pairs. Such a relation is at the basis of an approach to case-based reasoning called analogical extrapolation, which consists in retrieving three cases forming a proportional analogy with the target problem in the problem space and then in finding a solution to this problem by solving an analogical equation in the solution space. This paper studies how the notion of competence of pairs of source cases can be estimated and used in order to improve extrapolation. A preprocessing of the case base associates to each case pair a competence given by two scores: the support and the confidence of the case pair, computed on the basis of other case pairs forming a proportional analogy with it. An evaluation in a Boolean setting shows that using case pair competences improves significantly the result of the analogical extrapolation process.

Keywords: analogical inference, analogical proportion, case-based reasoning, competence, extrapolation

1 Introduction

Dans un travail récent [13], les auteurs ont montré que le raisonnement à partir de cas (RàPC [16]) peut ne pas être fondé uniquement sur un raisonnement utilisant la similarité (à la recherche des cas résolus les plus proches) mais peut également utiliser des proportions analogiques à des fins d'extrapolation. L'extrapolation repose sur l'inférence analogique qui utilise des triplets de cas pour construire la solution d'un quatrième (nouveau) cas par le biais d'un mécanisme d'adaptation.

Habituellement, plusieurs triplets de la base de cas peuvent être utilisés pour prédire la solution du quatrième cas et les prédictions peuvent diverger. En réalité, il a été démontré que dans un contexte booléen une inférence de ce type ne produit pas d'erreurs (i.e. tous les triplets conduisent vers la même prédiction) si et seulement si la fonction qui associe une solution à la description du cas est une fonction booléenne affine [4]. C'est pourquoi, lorsque la fonction n'est pas supposée affine, une procédure de vote est mise en œuvre sur les prédictions des triplets.

Une telle procédure est assez basique et n'exploite pas pleinement la base de cas. En effet, il se peut que certains triplets dans la base échouent dans la prédiction de la bonne réponse d'un autre cas déjà présent dans la base, traité comme un nouveau cas à résoudre. Dans ce travail, nous proposons de prendre en compte ce type d'informa-

tion pour restreindre le nombre de triplets utilisés pour effectuer une prédiction.

Ce document est organisé de la façon suivante. La section 2 fournit les prérequis nécessaires aux proportions analogiques et aux notations sur le RàPC utilisées dans ce document. La section 3 explique comment restreindre les triplets autorisés à participer à une prédiction donnée. La section 4 présente une expérimentation qui montre le gain en précision de la nouvelle procédure d'inférence. Enfin, la section 5 situe ce travail au regard de l'existant.

2 Préliminaires

Cette section présente dans un premier temps le cadre formel de ce travail : celui des représentations nominales et, plus particulièrement, les représentations par n-uplets de booléens. Puis, elle rappelle quelques notions et notations, d'abord sur les proportions analogiques puis sur le raisonnement à partir de cas.

2.1 Représentation nominale et contexte booléen

Les représentations de type attribut-valeur sont souvent utilisées en RàPC (voir, e.g., [11]). Une représentation nominale est une représentation de type attribut-valeur où le co-domaine de l'attribut est fini (et, typiquement, petit). Plus formellement, soit $\mathcal{U}_1$, $\mathcal{U}_2$, …, $\mathcal{U}_p$ p ensembles finis et $\mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2 \times \ldots \times \mathcal{U}_p$. Un attribut sur $\mathcal{U}$ est une des p projections $(x_1, x_2, \ldots, x_p) \in \mathcal{U} \mapsto x_i \in \mathcal{U}_i$ ($i \in \{1, 2, \ldots, p\}$).

Une représentation booléenne est une représentation nominale où $\mathcal{U}_1 = \mathcal{U}_2 = \ldots = \mathcal{U}_p = \mathbb{B}$, avec $\mathbb{B} = \{0, 1\}$ l'ensemble des valeurs booléennes : la valeur « faux » est assimilée à l'entier 0, et la valeur « vrai » est assimilée à 1. Les opérateurs booléens $\neg$, $\wedge$ et $\vee$ sont définis, pour $\mathtt{a}, \mathtt{b} \in \mathbb{B}$, par $\neg\mathtt{a} = 1 - \mathtt{a}$, $\mathtt{a} \wedge \mathtt{b} = \begin{cases} 1 & \text{if } \mathtt{a} = \mathtt{b} = 1 \\ 0 & \text{sinon} \end{cases}$, $\mathtt{a} \vee \mathtt{b} = \neg(\neg\mathtt{a} \wedge \neg\mathtt{b})$, $\mathtt{a} \equiv \mathtt{b} = (\neg\mathtt{a} \vee \mathtt{b}) \wedge (\neg\mathtt{b} \vee \mathtt{a})$. Un élément de $\mathbb{B}^p$ est noté sans virgules ni parenthèses, par exemple 01101 correspond à $(0, 1, 1, 0, 1)$.

2.2 Proportions analogiques

Étant donné un ensemble $\mathcal{U}$, une proportion analogique sur $\mathcal{U}$ est une relation quaternaire sur $\mathcal{U}$, notée $a:b::c:d$ pour $(a, b, c, d) \in \mathcal{U}^4$, et satisfiant les postulats suivants (pour $a, b, c, d \in \mathcal{U}$) :

(Réflexivité) $a:b::a:b$.

(Symétrie) Si $a:b::c:d$ alors $c:d::a:b$.

(Échange des moyens) Si $a:b::c:d$ alors $a:c::b:d$.

Étant donné un ensemble fini $\mathcal{U}_i$, la relation définie ci-dessous est une proportion analogique [15] :

$$a:b::c:d \stackrel{\text{déf}}{=} ((a \equiv b) \wedge (c \equiv d)) \vee ((a \equiv c) \wedge (b \equiv d))$$

Par conséquent, dans le cas nominal, les quadruplets (a, b, c, d) en analogie sont d'une des trois formes suivantes : (s, s, s, s), (s, t, s, t) et (s, s, t, t) pour $s, t \in \mathcal{U}_i$. En particulier, si $\mathcal{U}_i = \mathbb{B}$, l'ensemble des $(\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}) \in \mathbb{B}^4$ tels que $\mathtt{a}:\mathtt{b}::\mathtt{c}:\mathtt{d}$ est donc $\{0000, 1111, 0011, 1100, 0101, 1010\}$.

Pour un $\mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2 \times \ldots \times \mathcal{U}_p$ fini, la proportion analogique suivante peut être définie :

$$\begin{aligned} a:b::c:d = {} & a_1:b_1::c_1:d_1 \\ & \wedge\, a_2:b_2::c_2:d_2 \\ & \wedge\, \ldots \\ & \wedge\, a_p:b_p::c_p:d_p \end{aligned}$$

Étant donné $a, b, c \in \mathcal{U}$, résoudre l'*équation analogique* $a:b::c:y$ consiste à trouver les $y \in \mathcal{U}$ tels que cette relation soit vérifiée. Dans une représentation nominale, une telle équation a 0 ou 1 solution. Plus précisément :

— Si $a = b$, la solution est $y = c$.
— Si $a = c$, la solution est $y = b$.
— Sinon, $a:b::c:y$ n'a pas de solution.

2.3 Notations et hypothèses du RàPC

Soient $\mathcal{P}$ et $\mathcal{S}$ deux ensembles. Un *problème* $\mathtt{x}$ est par définition un élément de $\mathcal{P}$ et une *solution* $\mathtt{y}$, un élément de $\mathcal{S}$. Si $a \in \mathcal{P} \times \mathcal{S}$, alors $\mathtt{x}^a$ et $\mathtt{y}^a$ sont respectivement sa partie problème et sa partie solution : $a = (\mathtt{x}^a, \mathtt{y}^a)$. Soit $\rightsquigarrow$ une relation sur $\mathcal{P} \times \mathcal{S}$. Pour $(\mathtt{x}, \mathtt{y}) \in \mathcal{P} \times \mathcal{S}$, $\mathtt{x} \rightsquigarrow \mathtt{y}$ exprime « $\mathtt{x}$ a pour solution $\mathtt{y}$ » ou « $\mathtt{y}$ résout $\mathtt{x}$ ». Un *cas* est un couple $(\mathtt{x}, \mathtt{y})$ tel que $\mathtt{x} \rightsquigarrow \mathtt{y}$. Le but d'un système de RàPC est de résoudre des problèmes, i.e., il doit approcher la relation $\rightsquigarrow$: étant donné $\mathtt{x}^{\mathtt{cible}} \in \mathcal{P}$ (le problème *cible*), il doit proposer $\mathtt{y}^{\mathtt{cible}} \in \mathcal{S}$ tel qu'il est plausible que $\mathtt{x}^{\mathtt{cible}} \rightsquigarrow \mathtt{y}^{\mathtt{cible}}$. Pour cela, un ensemble fini de cas, appelé la base de cas, notée $\mathtt{BC}$, est utilisée. Un élément de $\mathtt{BC}$ est appelé un *cas source*. Hormis la base de cas, d'autres *conteneurs* de connaissances sont souvent utilisés [16], mais ne sont pas considérés dans ce travail.

La façon habituelle de définir le processus de RàPC consiste à sélectionner un ensemble de k cas sources en relation avec $\mathtt{x}^{\mathtt{cible}}$ (étape de remémoration) et puis à résoudre $\mathtt{x}^{\mathtt{cible}}$ à l'aide des cas remémorés (étape d'adaptation). D'autres étapes sont considérées dans le modèle classique de RàPC [1], mais pas dans ce travail. Dans [13], trois approches sont présentées pour $k \in \{1, 2, 3\}$, correspondant à trois relations entre k-uplets de cas sources et le problème à résoudre. Dans ce travail, l'approche avec $k = 3$ est rappelée dans la section suivante.

3 Améliorer l'extrapolation grâce à la compétence des couples de cas

Cette section présente l'approche proposée. Dans un premier temps, on montrera comment une notion de compétence associée aux couples de cas peut être utilisée pour améliorer l'extrapolation. Puis, la notion de compétence est définie formellement. Enfin, des stratégies utilisant cette notion en vue d'améliorer l'extrapolation sont décrites.

3.1 Principes

Le principe de l'inférence par proportion analogique [18] peut être énoncé comme suit (avec l'utilisation des notations de RàPC introduites précédemment ; $a = (\mathtt{x}^a, \mathtt{y}^a)$, $b = (\mathtt{x}^b, \mathtt{y}^b)$, $c = (\mathtt{x}^c, \mathtt{y}^c)$ et $d = (\mathtt{x}^d, \mathtt{y}^d)$ sont quatre cas) :

$$\frac{\mathtt{x}^a : \mathtt{x}^b :: \mathtt{x}^c : \mathtt{x}^d \text{ est vérifiée}}{\mathtt{y}^a : \mathtt{y}^b :: \mathtt{y}^c : \mathtt{y}^d \text{ est vérifiée}}$$

Résoudre un nouveau problème $\mathtt{x}^{\mathtt{cible}}$, consiste à considérer tous les triplets de cas sources (a, b, c) tels que $\mathtt{x}^a : \mathtt{x}^b :: \mathtt{x}^c : \mathtt{x}^{\mathtt{cible}}$ est vérifiée et tels que l'équation $\mathtt{y}^a : \mathtt{y}^b :: \mathtt{y}^c : y$ est résoluble. Soit $\mathcal{T}$ l'ensemble de tous ces triplets. L'implantation de cette inférence utilise un vote parmi tous les triplets de $\mathcal{T}$ et choisit la solution y trouvée sur le plus grand ensemble de triplets. Ceci est le principe de l'extrapolation analogique (ou, simplement, extrapolation [13]).

Dans la suite, nous supposons pour simplifier que tous les attributs sont nominaux (p. ex., booléens). Lorsqu'il y a un seul attribut pour les solutions, la tâche de résolution de problème est une tâche de classification (consistant à trouver la classe $\mathtt{y}^{\mathtt{cible}} \in \mathcal{S}$ devant être associée à $\mathtt{x}^{\mathtt{cible}}$). Lorsqu'il y a plusieurs attributs, il est possible de les traiter un par un si et seulement si ils sont logiquement indépendants. Sinon, le vote doit être organisé entre les vecteurs complets décrivant les différentes solutions. Dans la suite, nous supposons l'indépendance et nous considérons une des composantes $\mathtt{y}_i$ d'une solution $\mathtt{y}$ (ainsi, l'index i est inutile : $\mathtt{y}_i$ est noté par $\mathtt{y}$).

Cependant, on peut se demander si tous les triplets de $\mathcal{T}$ impliqués dans un vote pour faire une prédiction particulière ont la même légitimité. En effet, des leçons peuvent être tirées de $\mathcal{T}$ en observant lors d'une prédiction de solution pour un problème déjà présent dans $\mathcal{T}$ s'il existe des triplets produisant une prédiction erronée, comme suggéré dans [14]. La situation peut être mieux analysée en termes de couples, comme on le montre à présent.

La table 1 donne un exemple avec trois couples de booléens tels que $\mathtt{a:b::c:d}$ et $\mathtt{a':b'::c:d}$ sont vérifiées dans toutes les colonnes, exceptée la dernière colonne, 'S', colonne solution. Les deux premières colonnes 'D' (comme désaccord) montrent les motifs possibles exprimant que a et b diffèrent de la même façon que c et d et que a′ et b′([1]). Les colonnes 'A' (comme accord) montrent toutes les possibilités où a et b sont en accord, avec c et d également en accord, ainsi que a′ et b′, mais éventuellement d'une façon différente [2]. Si on regarde la valeur de d dans la colonne 'S' et que l'on cherche à la prédire à partir des autres valeurs de cette colonne, l'équation $\mathtt{a:b::c:}y$ donne le bon résultat, tandis que l'équation $\mathtt{a':b'::c:}y$ donne le mauvais.

Ainsi, pour chaque couple comme $(\mathtt{a}, \mathtt{b})$ ou $(\mathtt{a}', \mathtt{b}')$ dans la table, il est possible de compter le nombre de fois où les couples conduisent à une bonne ou à une mauvaise prédiction pour un exemple pris dans BC (via des c appropriés également présents dans BC). Cela fournit un moyen de favoriser des triplets contenant des couples menant à de bonnes prédictions, dans la procédure de vote.

L'idée ci-dessus, de regarder les couples de cas, peut être mise en relation avec l'interprétation d'un couple de cas (a, b) comme une règle potentielle exprimant soit que le changement de $\mathtt{x}^a$ à $\mathtt{x}^b$ explique le changement de $\mathtt{y}^a$ à $\mathtt{y}^b$, indépendamment du contexte (encodé par les attributs pour lesquelles les exemples sont en accord), soit que le changement de $\mathtt{x}^a$ à $\mathtt{x}^b$ ne modifie pas la solution (si $\mathtt{y}^a = \mathtt{y}^b$). Cette vision des couples de cas comme des règles a déjà été proposée en RàPC pour trouver des règles d'adaptation [9, 5, 6] et par la suite dans une perspective d'inférence fondée sur les proportions analogiques [3, 2].

Par conséquent, nous sommes intéressés par un prétraitement, afin de découvrir des *ruptures d'analogie* dans $\mathcal{T}$. Par rupture d'analogie, nous entendons l'existence d'un quadruplet de cas (a, b, c, d) tel que (i) $\mathtt{x}^a : \mathtt{x}^b :: \mathtt{x}^c : \mathtt{x}^d$ est vérifié, tandis que (ii) $\mathtt{y}^a : \mathtt{y}^b :: \mathtt{y}^c : \mathtt{y}^d$ *n'est pas* vérifié. Si des ruptures d'analogie peuvent être trouvées dans $\mathcal{T}$, cela signifie que la fonction booléenne partiellement inconnue qui associe à un problème sa solution (ou sa classe) ne peut pas être affine [4]. Dans cette situation, une inférence analogique ne peut pas être appliquée aveuglément avec n'importe quel triplet, et nous devrions tenir compte de ces ruptures d'analogie, en introduisant des restrictions supplémentaires pour le choix de triplets appropriés.

Plus précisément, l'idée est de faire un prétraitement des couples $(a, b) \in \mathtt{BC}^2$, en associant à chacun d'eux une *compétence*. L'intuition derrière cette notion est que plus un couple de cas est compétent pour résoudre des problèmes, plus il peut jouer un rôle dans le processus de vote. Pour évaluer la compétence d'un couple $(a, b) \in \mathtt{BC}^2$, il doit être mis en proportion analogique avec d'autres couples $(c, d) \in \mathtt{BC}^2$ tels que le triplet (a, b, c) peut être utilisé pour résoudre le problème $\mathtt{x}^d$ par extrapolation. Si le résultat de l'extrapolation y est égal à $\mathtt{y}^d$, la compétence du couple de cas (a, b) augmente, sinon elle diminue. La définition de la

1. $D(0/1)$ indique le désaccord entre a et b (resp., entre c et d et entre a′ et b′) où le premier vaut 0 et le second vaut 1. $D(1/0)$ est le désaccord inverse.

2. Dans la table 1, $A(u, v, w)$ signifie que $\mathtt{a} = \mathtt{b} = u$, $\mathtt{c} = \mathtt{d} = v$ et $\mathtt{a}' = \mathtt{b}' = w$.

	$D(0/1)$	$D(1/0)$	$A(0,0,0)$	$A(0,0,1)$	$A(0,1,0)$	$A(0,1,1)$	$A(1,0,0)$	$A(1,0,1)$	$A(1,1,0)$	$A(1,1,1)$	S
a	0	1	0	0	0	0	1	1	1	1	0
b	1	0	0	0	0	0	1	1	1	1	1
c	0	1	0	0	1	1	0	0	1	1	0
d	1	0	0	0	1	1	0	0	1	1	1
a′	0	1	0	1	0	1	0	1	0	1	0
b′	1	0	0	1	0	1	0	1	0	1	0

TABLE 1 – Double appariement de couples (a, b), (c, d) et (a′, b′) : rupture d'analogie en S.

compétence est détaillée dans la section suivante.

La compétence d'un couple de cas peut être utilisée au moment de la résolution d'un nouveau problème, avec différentes stratégies. La section 3.3 présente certaines de ces stratégies qui sont évaluées expérimentalement dans un cadre booléen en section 4.

3.2 Acquisition des compétences des couples de cas

Soit (a, b) un couple de cas sources : $a = (\mathtt{x}^a, \mathtt{y}^a) \in \mathtt{BC}$ et $b = (\mathtt{x}^b, \mathtt{y}^b) \in \mathtt{BC}$. La *compétence* du couple (a, b) est définie par deux scores : le support et la confiance de (a, b), définis ci-dessous à partir du principe présenté précédemment.

Tout d'abord, soit $\mathtt{RésolublesPar}(a, b)$ l'ensemble des couples de cas sources $(c, d) \neq (a, b)$ tels que le triplet (a, b, c) peut être utilisé pour résoudre $\mathtt{x}^d$ par extrapolation : $\mathtt{x}^a : \mathtt{x}^b :: \mathtt{x}^c : \mathtt{x}^d$ avec l'équation $\mathtt{y}^a : \mathtt{y}^b :: \mathtt{y}^c : y$ résoluble (et ainsi, sa solution y est unique dans une représentation nominale). Formellement :

$$\mathtt{RésolublesPar}(a, b) = \left\{ (c, d) \in \mathtt{BC}^2 \;\middle|\; \begin{array}{l} (c, d) \neq (a, b), \\ \mathtt{x}^a : \mathtt{x}^b :: \mathtt{x}^c : \mathtt{x}^d \\ \text{et l'équation} \\ \mathtt{y}^a : \mathtt{y}^b :: \mathtt{y}^c : y \\ \text{a une solution} \end{array} \right\}$$

En d'autres termes, $\mathtt{RésolublesPar}(a, b)$ est l'ensemble des couples de cas sources (c, d) tels que c peut être adapté en une solution de $\mathtt{x}^d$ en utilisant (a, b) comme une règle d'adaptation (sans considérer le cas trivial $(a, b) = (c, d)$). Le support de (a, b), $\mathtt{supp}(a, b)$, est simplement le nombre de tels couples :

$$\mathtt{supp}(a, b) = |\mathtt{RésolublesPar}(a, b)|$$

Parmi les $(c, d) \in \mathtt{RésolublesPar}(a, b)$ certains conduisent à la bonne solution ($y = \mathtt{y}^d$) et d'autres non. Les premiers constituent l'ensemble suivant :

$$\mathtt{RésolublesCorrectementPar}(a, b) = \left\{ (c, d) \in \mathtt{RésolublesPar}(a, b) \mid \mathtt{y}^a : \mathtt{y}^b :: \mathtt{y}^c : \mathtt{y}^d \right\}$$

Par exemple, si $\mathtt{supp}(a, b) = 6$ et $|\mathtt{RésolublesCorrectementPar}(a, b)| = 4$, cela signifie que (a, b), considéré comme une règle, a été testé sur la base de cas 6 fois et a donné 4 bonnes réponses. La proportion de bonnes réponses est donc $4/6 = 2/3$. Cette proportion est appelée confiance de (a, b). Un cas particulier doit être considéré quand $\mathtt{supp}(a, b) = 0$. Cela signifie que « la règle d'adaptation » (a, b) ne peut pas être testée sur la base de cas. Dans ce cas, la valeur de la confiance est fixé à 0,5 (meilleure qu'une confiance de, par exemple, 3/7 pour laquelle la règle échoue plus souvent qu'elle ne réussit et moins bien qu'une confiance de 4/7 pour laquelle la règle donne de bonnes réponses plus souvent que de mauvaises). En résumé, la confiance d'un couple (a, b) est :

$$\mathtt{conf}(a, b) = \begin{cases} \frac{|\mathtt{RésolublesCorrectementPar}(a,b)|}{\mathtt{supp}(a,b)} & \text{si } \mathtt{supp}(a, b) \neq 0 \\ 0{,}5 & \text{sinon} \end{cases}$$

3.3 Utilisation des compétences de couples de cas dans des stratégies de sélection et de vote

Étant donné un problème cible $\mathtt{x}^{\mathtt{cible}}$, l'extrapolation consiste à remémorer des triplets $(a, b, c) \in \mathtt{BC}^3$ tels que $\mathtt{x}^a : \mathtt{x}^b :: \mathtt{x}^c : \mathtt{x}^{\mathtt{cible}}$ et à adapter ces triplets en résolvant l'équation $\mathtt{y}^a : \mathtt{y}^b :: \mathtt{y}^c : y$ pour chacun de ces triplets (évidemment, les triplets (a, b, c) pour lesquelles l'équation n'a pas de solution ne sont pas considérés). Ainsi, le résultat de l'extrapolation est l'ensemble $\mathcal{R}$ des $((a, b, c), y) \in \mathtt{BC}^3 \times \mathcal{S}$, y étant le résultat de l'extrapolation de (a, b, c) afin de résoudre $\mathtt{x}^{\mathtt{cible}}$. Maintenant, la question est comment considérer toutes ces solutions y pour proposer une unique solution $\mathtt{y}^{\mathtt{cible}}$ pour $\mathtt{x}^{\mathtt{cible}}$. Pour cela, quatre stratégies sont considérées.

La première, appelée `sansComp`, effectue simplement un vote avec toutes les valeurs de y, indépendamment des compétences. La solution proposée est ainsi

$$\mathtt{y}^{\mathtt{cible}} = \underset{\widehat{y}}{\operatorname{argmax}} \left| \{ ((a, b, c), y) \in \mathcal{R} \mid y = \widehat{y} \} \right|$$

C'est la stratégie utilisée dans [13] et elle constitue une référence pour l'évaluation.

La deuxième stratégie, appelée `toutesConf`, considère tous les $((a, b, c), y) \in \mathcal{R}$ et effectue un vote pondéré par la confiance :

$$\mathtt{y}^{\mathtt{cible}} = \underset{\widehat{y}}{\operatorname{argmax}} \sum_{((a,b,c),y) \in \mathcal{R}, y = \widehat{y}} \mathtt{conf}(a, b)$$

La troisième stratégie, appelée `maxConf`, considère uniquement les $((a,b,c),y) \in \mathcal{R}$ ayant la plus haute confiance, et effectue un vote à partir de ces dernières. Formellement :

$$\begin{aligned} \text{avec } \mathtt{conf}_{\max} &= \max\{\mathtt{conf}(a,b) \mid ((a,b,c),y) \in \mathcal{R}\} \\ \text{et } \mathcal{R}^* &= \{((a,b,c),y) \in \mathcal{R} \mid \mathtt{conf}(a,b) = \mathtt{conf}_{\max}\} \\ \mathtt{y}^{\mathtt{cible}} &= \operatorname*{argmax}_{\widehat{y}} \left|\{((a,b,c),y) \in \mathcal{R}^* \mid y = \widehat{y}\}\right| \end{aligned} \quad (1)$$

La quatrième stratégie, appelée `maxConfSupp`, est similaire à la troisième, excepté qu'elle utilise à la fois la confiance et le support pour établir une préférence. Plus précisément, cette stratégie s'appuie sur un ordre de préférence des couples de cas défini ci-dessous (pour $(a,b),(a',b') \in \mathtt{BC}^2$:

$$(a,b) \succcurlyeq (a',b') \quad \text{si} \quad \left| \begin{array}{l} \mathtt{conf}(a,b) > \mathtt{conf}(a',b') \textbf{ ou} \\ (\mathtt{conf}(a,b) = \mathtt{conf}(a',b') \\ \quad \textbf{et } \mathtt{supp}(a,b) \geqslant \mathtt{supp}(a',b')) \end{array} \right.$$

En d'autres termes, la confiance est le critère principal mais, en cas d'égalité, plus le support est élevé, plus le couple de cas (a,b) est considéré comme étant compétent. Par exemple, si $\mathtt{conf}(a,b) = \mathtt{conf}(a',b') = 0{,}75$, $\mathtt{supp}(a,b) = 8$ et $\mathtt{supp}(a',b') = 4$, alors (a,b) donne la bonne réponse dans 6 situations sur 8, tandis que (a',b') le fait dans 3 situations sur 4. Soit alors $\mathcal{R}^*$ l'ensemble des $((a,b,c),y) \in \mathcal{R}$ tels que (a,b) est maximal pour $\succcurlyeq$. Alors, $\mathtt{y}^{\mathtt{cible}}$ résulte d'un vote, comme décrit précédemment dans l'équation (1).

L'intérêt de considérer un triplet (a,b,c) dans la procédure de vote à la fin du processus d'inférence est évalué en terme de compétence du couple (a,b). Comme les proportions analogiques sont stables pour la permutation centrale, on peut également penser à considérer les couples (a,c). Des investigations préliminaires, utilisant différentes combinaisons (minimum, maximum, somme et produit) des confiances de (a,b) et (a,c) n'ont pas montré d'apport significatif par rapport à l'utilisation seule de la confiance de (a,b) ; c'est pourquoi nous nous sommes restreints, dans ce travail, à ce dernier type de prise en compte de la compétence. Cependant, la combinaison n'a porté que sur l'utilisation de la confiance brute de tous les triplets (comme dans la méthode `toutesConf`) sans considérer de stratégie de restriction des triplets participant au vote (comme dans les stratégies `maxConf` et `maxConfSupp`). L'étude plus approfondie d'une stratégie de combinaison reposant sur la confiance mais également sur le support constitue une perspective de ce travail.

4 Évaluation

L'objectif de l'évaluation est d'étudier l'impact des stratégies pour la sélection des couples de cas participant au vote sur différents types de fonctions booléennes.

4.1 Contexte expérimental

Dans nos expériences, $\mathcal{P} = \mathbb{B}^8$ et $\mathcal{S} = \mathbb{B}$. Ici $\rightsquigarrow$ est supposé être fonctionnelle : $\rightsquigarrow = \mathtt{f}$.

La fonction `f` est générée de manière aléatoire à l'aide des générateurs suivants qui sont fondées sur deux des formes normales les plus classiques dans le but d'avoir une diversité de fonctions générées :

`DNF` `f` est générée en forme normale disjonctive, c.-à-d., que $\mathtt{f}(\mathtt{x})$ est une disjonction de n_{disj} clauses (conjonctions de littéraux), par exemple :
$\mathtt{f}(\mathtt{x}) = (\mathtt{x}_1 \wedge \neg\mathtt{x}_7) \vee (\neg\mathtt{x}_3 \wedge \mathtt{x}_7 \wedge \mathtt{x}_8) \vee \mathtt{x}_4$.
La valeur de n_{disj} est choisie au hasard uniformément dans $\{3,4,5\}$. Chaque disjonction est générée sur la base de deux paramètres, $p^+ > 0$ et $p^- > 0$, avec $p^+ + p^- < 1$: dans chaque clause, pour chaque variable x_i, elle apparaît sous forme de littéral positif (resp., négatif) avec une probabilité p^+ (resp., p^-). Dans nos expériences, nous avons choisi $p^+ = p^- = 0{,}1$([3]).

`Pol` `f` est générée sous forme polynomiale : `f` a la même forme que si elle était générée par `DNF`, sauf que les disjonctions ($\vee$) sont remplacées par des ou exclusifs ($\oplus$). Comme seuls les littéraux positifs apparaissent dans la forme normale polynomiale, le paramètre $p^- = 0$.

La base de cas `BC` est générée aléatoirement, avec des valeurs de taille $|\mathtt{BC}| \in \{32, 64, 96, 128\}$, i.e. $|\mathtt{BC}|$ est compris entre $\frac{1}{8}$ et $\frac{1}{2}$ de $|\mathcal{P}| = 2^8 = 256$. Chaque cas source $(\mathtt{x},\mathtt{y})$ est généré comme suit : $\mathtt{x}$ est choisi au hasard dans $\mathcal{P}$ avec une distribution uniforme et $\mathtt{y} = \mathtt{f}(\mathtt{x})$.

Soient `#pb_cibles` le nombre de problèmes cibles soumis au système, `#rep` le nombre de réponses (correctes ou incorrectes) (`#pb_cibles` − `#rep` est le nombre de problèmes cibles pour lesquels le système échoue à proposer une solution), et `#rep_ok` le nombre de réponses correctes. Pour chaque stratégie de sélection et vote, les scores suivants sont calculés :

Le taux d'erreur `%err` est la moyenne des $\left(1 - \frac{\#\mathtt{rep\_ok}}{\#\mathtt{rep}}\right) \times 100 \in [0,100]$.

Le taux de réponse `%rep` est la moyenne des rapports $\frac{\#\mathtt{rep}}{\#\mathtt{pb\_cibles}} \times 100 \in [0,100]$. Si le système retourne toujours une réponse (correcte ou non) alors $\mathtt{\%rep} = 100$.

3. Un générateur `CNF`, générant des formules en CNF (forme normale conjonctive : conjonction de disjonctions de littéraux) peut également être considéré. Cependant, ceci n'apporterait rien car ce serait dual avec le générateur `DNF` pour deux raisons. Premièrement, les inférences effectuées sont indépendantes du codage, signifiant que remplacer des attributs par leur négations ne changent pas le résultat de l'inférence, en particulier, pour $\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d} \in \mathbb{B}$, $\mathtt{a{:}b{::}c{:}d}$ si et seulement si $\neg\mathtt{a{:}}\neg\mathtt{b{::}}\neg\mathtt{c{:}}\neg\mathtt{d}$. Deuxièmement, si `f` est obtenu à partir d'un générateur `DNF` alors $\neg\mathtt{f}$ peut facilement être transformé en une fonction g écrite en CNF en utilisant les lois de De Morgan, et la distribution de g obtenue de cette façon serait la même que la distribution d'un générateur `CNF` avec les mêmes paramètres.

La moyenne est calculée sur 1 million de résolutions de problème pour chaque générateur, nécessitant la génération de 1420 `f` pour chacun d'eux. La moyenne du temps de calcul d'une session de RàPC (remémoration et adaptation pour résoudre un problème) est d'environ 2 ms sur un ordinateur portable courant.

Par souci de reproductibilité, le code des expérimentations est disponible sur `https://tinyurl.com/analogyCBRTests`, avec les résultats détaillés (fonctions générées et détails des évaluations).

4.2 Résultats

La table 2 présente les taux d'erreur et de réponse pour les différentes stratégies de sélection et de vote pour les deux générateurs, avec une application à différentes tailles de la base de cas. Les courbes du taux d'erreur sont données en Figure 1.

Étant donnée un générateur de fonction et une taille de base de cas, le taux de réponse est le même pour chacune des quatre stratégies car chacune des stratégies de sélection fournit des résultats pour un problème qui peut être résolu par l'approche sans sélection `sansComp` (i.e. si un triplet a été trouvé pour résoudre un cas par `sansComp`, ce triplet sera considéré par les trois stratégies de sélection et soit il participera à la résolution de $\mathtt{x}^{\mathtt{cible}}$, soit il existe un autre triplet qui est « meilleur » vis-à-vis de la procédure de sélection). Le taux de réponse est élevé pour toutes les méthodes : plus de 96% pour $|\mathtt{BC}| = 32$ et 100% pour $|\mathtt{BC}| \geqslant 64$.

Excepté pour $|\mathtt{BC}| = 32$, qui semble être un ensemble d'apprentissage trop petit pour calculer les compétences, le taux d'erreur montre que l'hypothèse de sélection des couples de cas améliore la précision. Pour les deux générateurs, toutes les stratégies de sélection donnent de meilleurs résultats que la stratégie de référence (`sansComp`). Cependant, l'amélioration est assez différente en fonction de la stratégie de sélection : plus la sélection des couples de cas est contrainte, plus le taux d'erreur décroit. `toutesConf` diminue légèrement le taux d'erreur, `maxConf` diminue un peu plus le taux d'erreur, et les meilleurs résultats sont donnés par `maxConfSupp`.

Le bénéfice de chaque stratégie est lié à la taille de la base de cas : plus la base de cas contient de cas pour l'acquisition des compétences, meilleurs sont les résultats. Comparé à la stratégie de référence, le bénéfice pour la meilleure stratégie pour la sélection `maxConfSupp` est remarquable. Même si le taux de réponse est déjà très bon, avec la stratégie de référence et d'autant plus avec un taux de réponse de 100%, `maxConfSupp` l'améliore encore en se rapprochant significativement de 100% de réponses correctes. Pour `DNF`, selon la taille de la base de cas (64, 96 et 128), le taux d'erreur `%err` passe de 10,1 à 6,9 (diminution de 32%), de 8,4 à 3,3 (diminution de 61%) et de 7,7 à 1,7 (diminution de 78%). Pour `Pol`, les résultats sont encore plus impressionnant, selon la taille de la base de cas (64, 96 et 128), le taux d'erreur `%err` passe de 13,7 à 6,3 (diminution de 54%), de 10,5 à 1,6 (diminution de 859%) et de 8,8 à 0,5 (diminution de 94%).

Ainsi, les premiers résultats expérimentaux montrent que, à partir d'une certaine taille de la base de cas, la stratégie `maxConfSupp` surpasse toutes les autres méthodes et diminue nettement le taux d'erreur, tout en utilisant un nombre de triplets plus restreint.

5 Discussion et travaux proches

Dans cette section, l'approche présentée dans ce papier est comparée à la littérature sur le RàPC selon deux angles : la notion de compétence et l'apprentissage de connaissances d'adaptation.

Les compétences en RàPC. Dans [13], trois types de processus de RàPC sont distingués, en particulier l'extrapolation, qui remémore et réutilise les cas par triplets, et l'approximation, qui remémore et réutilise les cas par singletons. Par rapport aux recherches précédentes sur la compétence qui sont liées à l'approximation, les travaux présentés dans cet article considère une notion de compétence liée à l'extrapolation.

La notion de compétence en RàPC est utilisée en général pour la maintenance de la base de cas, soit pour supprimer les cas les moins compétents [17] (dans l'optique de minimiser la perte de compétence globale), soit ajouter des cas compétents [19] (dans l'optique de maximiser le gain de compétence globale). Dans ces précédentes études, la compétence est liée au cas sources, individuellement, par rapport aux autres cas de la base de cas. En particulier, dans l'article fondateur [17], la compétence des cas est évaluée en mettant les cas sources dans des catégories (des cas centraux — *pivotal cases* —, qui sont les plus compétents, aux cas auxiliaires), ces catégories étant définies avec l'aide d'une notion binaire de l'adaptabilité entre cas et problème. Ainsi, la notion de compétence est liée au processus d'approximation, considérant les cas sources individuellement.

En revanche, cet article aborde la compétence liée au processus d'extrapolation : les cas sont remémorés par triplets. La compétence d'un triplet $(a, b, c) \in \mathtt{BC}^3$ est réduite à la compétence du couple $(a, b) \in \mathtt{BC}^2$, qui est liée à l'ensemble des autres couples $(c, d) \in \mathtt{BC}^2$. Un point commun à ces deux notions de compétence est que la compétence d'un objet (l'objet étant un cas pour l'approximation et un couple de cas pour l'extrapolation) n'est pas une propriété intrinsèque à l'objet, mais est liée aux autres objets (de $\mathtt{BC}$ ou $\mathtt{BC}^2$).

Une différence mineure entre les études précédentes sur la compétence et celle-ci est liée à l'utilisation de la com-

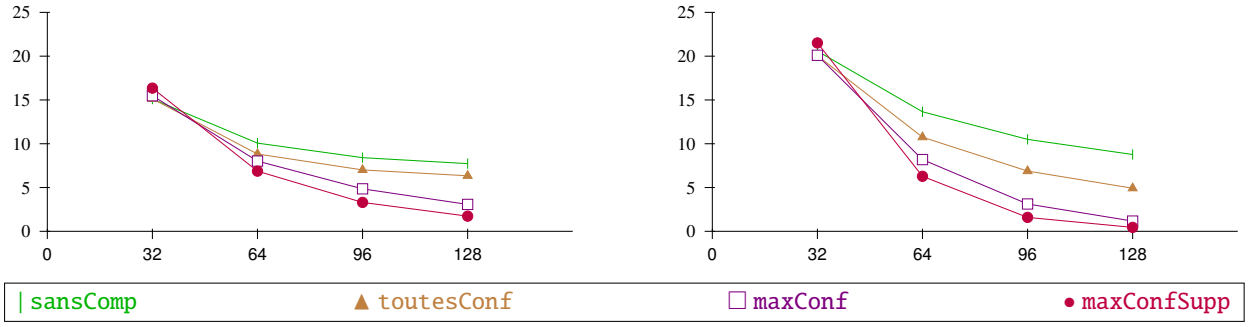


FIGURE 1 – Taux d'erreur en fonction de la taille de BC, pour chaque générateur (à gauche : DNF, à droite : Pol).

		|BC| = 32		|BC| = 64		|BC| = 96		|BC| = 128	
		%err	%rep	%err	%rep	%err	%rep	%err	%rep
DNF	sansComp	**15,1**	97,5	10,1	100,0	8,4	100,0	7,7	100,0
	toutesConf	15,1		8,8		7,0		6,3	
	maxConf	15,5		8,0		4,9		3,1	
	maxConfSupp	16,4		**6,9**		**3,3**		**1,7**	
POL	sansComp	20,6	95,8	13,7	100,0	10,5	100,0	8,8	100,0
	toutesConf	**20,1**		10,8		6,9		4,9	
	maxConf	20,1		8,2		3,1		1,2	
	maxConfSupp	21,5		**6,3**		**1,6**		**0,5**	

TABLE 2 – %err et %rep pour les différentes stratégies de sélection et vote pour les différents générateurs.

pétence : la maintenance de la base de case pour les précédentes et la résolution de problème pour celle-ci.

Liens avec l'apprentissage de connaissances d'adaptation. Le travail présenté dans cet article a des liens forts avec la majorité des travaux sur l'apprentissage de connaissances d'adaptation (ACA). L'adaptation dont il est question ici est celle qui suit la remémoration d'un seul cas (c'est une partie d'un processus d'approximation intégrant une adaptation). Elle s'appuie en particulier sur les connaissances d'adaptation AK qui peuvent se définir informellement par :

$$\texttt{AK} = \text{« Comment varie la solution quand varie le problème. »}$$

La démarche très généralement suivie en ACA, initiée par le travail de Kathleen Hanney et Mark T. Keane [9], s'appuie sur la base de cas pour extraire des connaissances d'adaptation selon le principe suivant. Elle considère un ensemble EA de couples de cas sources (a, b), avec $a \neq b$, soit en prenant tous les couples distincts de BC, soit en se restreignant sur la base d'une similarité minimale entre a et b. Puis, EA est utilisé comme ensemble d'apprentissage d'un processus d'apprentissage supervisé : pour tout couple (a, b), l'entrée est le couple $(\mathbf{x}^a, \mathbf{x}^b)$ et la sortie est le couple $(\mathbf{y}^a, \mathbf{y}^b)$. Le processus d'apprentissage supervisé donne alors un modèle de cette connaissance AK, utilisée par l'adaptation.

Plusieurs travaux s'inscrivent dans ce schéma général. Dans [10], AK consiste en une représentation de « cas d'adaptation ». Dans [5], différentes techniques sont utilisées, en particulier l'induction par arbre de décisions et des techniques d'apprentissage ensembliste/ Dans [6], l'extraction de motifs fermés fréquents est utilisée. L'interprétation par des experts du domaine d'application permet de produire des règles d'adaptation à ajouter à AK. Dans [8], des techniques similaires à celles de [6] sont utilisées (l'analyse formelle de concepts et l'extraction de motifs fermés fréquents étant des techniques de fouille de données très proches), mais, dans ce travail, des cas négatifs (i.e., des couples $(\mathbf{x}, \mathbf{y}) \in \mathcal{P} \times \mathcal{S}$ pour lesquels $\mathbf{y}$ n'est pas une solution correcte de $\mathbf{x}$) sont utilisés, ce qui améliore nettement les résultats de l'apprentissage.

On peut considérer que le travail présenté dans cet article relève également de l'ACA. En fait, nous avons déjà utilisé la notion de règle d'adaptation pour considérer un couple (a, b) de cas ; précisons cette idée. Soit la relation $\sim$ définie, pour (a, b) et (a', b'), deux couples de cas, par :

$$(a, b) \sim (a', b') \quad \text{si} \quad \mathbf{x}^a : \mathbf{x}^b :: \mathbf{x}^{a'} : \mathbf{x}^{b'} \text{ et } \mathbf{y}^a : \mathbf{y}^b :: \mathbf{y}^{a'} : \mathbf{y}^{b'}$$

Pour les proportions analogiques sur les représentations nominales définies en section 2.2, $\sim$ est une relation d'équivalence [4]. Donc, résoudre un problème $\mathtt{x}^{\mathtt{cible}}$ par extrapolation à partir d'un triplet $(a, b, c) \in \mathtt{BC}^3$ ou d'un triplet $(a', b', c) \in \mathtt{BC}^3$ (avec le même c) tels que $(a, b) \sim (a', b')$, donnera le même résultat : le choix du représentant de la classe d'équivalence de (a, b) pour $\sim$ est indifférent pour l'extrapolation. On peut utiliser une telle classe d'équivalence C comme une règle d'adaptation (où c est le cas remémoré et $\mathtt{x}^{\mathtt{cible}}$ est le problème à résoudre) :

avec (a, b) choisi arbitrairement dans C

si $\mathtt{x}^a : \mathtt{x}^b :: \mathtt{x}^c : \mathtt{x}^{\mathtt{cible}}$ et $\mathtt{y}^a : \mathtt{y}^b :: \mathtt{y}^c : y$ a une solution

alors cette solution est une solution plausible de $\mathtt{x}^{\mathtt{cible}}$

L'ensemble des classes d'équivalences de la restriction de $\sim$ à $\mathtt{BC}^2$ donne donc un ensemble de règles d'adaptation candidates, mais toutes les règles ne se valent pas : certaines sont plus plausibles que d'autres. Un critère doit donc être défini pour effectuer une préférence entre ces règles ou, si on décide de les appliquer toutes, de le faire en pondérant dans un vote certaines règles.

Une façon simple de faire cela (utilisée, par exemple, dans [6]) consiste à utiliser le cardinal de C. On rejoint là l'idée de compétence d'un couple de cas : si $(a, b) \in C$, alors, $|C| = \mathtt{supp}(a, b) \times \mathtt{conf}(a, b)$. La limite de cette approche est qu'elle ne compte que les exemples (à l'appui de la règle), pas les contre-exemples (qui pénalisent la règle). En revanche, l'approche considérée dans ce papier tient compte des contre-exemples : si on sait, par exemple, que $\mathtt{conf}(a, b) = 1/3$, alors on sait que pour chaque exemple, il y a deux contre-exemples, donc même si $\mathtt{supp}(a, b)$ est important, la règle associée à (a, b) sera douteuse.

Une autre différence avec le travail de [6] est que dans [6], on considérait des motifs de variations qui peuvent être approximatifs. Par exemple, si (a, b) et (a', b') sont deux couples de cas sources tels que, pour *la plupart* des attributs i, $\mathtt{x}_i^a : \mathtt{x}_i^b :: \mathtt{x}_i^{a'} : \mathtt{x}_i^{b'}$, on considérera la règle construite sur ces attributs communs, en négligeant le reste. Dans un cadre où les analogies sont rares (par exemple, quand on a des attributs à valeurs réelles), remplacer la proportion analogique exacte par une proportion analogique graduelle [7] dans l'approche décrite dans cet article pourrait se justifier, ce qui constitue une perspective de recherche potentielle de ce travail. En effet, la démarche présentée dans cet article pourrait s'appliquer avec une analogie graduelle, en particulier avec des attributs numériques.

Cette discussion montre que certaines idées d'ACA à partir de la base de cas se trouvent naturellement reformulées dans le cadre des proportions analogiques : ce lien établi entre les deux champs est donc potentiellement fécond.

4. La réflexivité et la symétrie découlent des postulats éponymes, en revanche, la transitivité n'est pas réalisée par toutes les proportions analogiques [12].

6 Conclusion

Le RàPC classique s'appuie sur les similarités individuelles du problème à résoudre avec chacun des problèmes résolus déjà connus. Nous avons montré qu'il pouvait également être intéressant d'envisager des triplets de cas (a, b, c) pour rendre égaux le changement de a à b avec le changement de c au problème à résoudre associée à sa solution potentielle. C'est le fondement de l'extrapolation analogique fondée sur les proportions analogiques. Cependant, il a été observé que certains triplets peuvent conduire à de mauvaises inférences.

Dans cet article, nous avons proposé de discriminer les triplets par une évaluation de la « compétence » de couples participant aux triplets. En effet, une proportion analogique « a est à b ce que c est à d » peut être vue comme l'établissement d'un parallèle entre ces deux couples. Les différences entre les attributs d'un couple de problèmes sont naturellement liées aux différences entre leurs solutions, mais cette relation dépend du contexte exprimé par les valeurs des attributs qui ne changent pas. Nous avons montré qu'il était possible, au moins dans une certaine mesure, d'évaluer la compétence de couples pour sélectionner les « bons » triplets et améliorer les résultats de l'inférence analogique. Ceci contribue à confirmer l'intérêt de l'extrapolation analogique pour le RàPC.

Une première perspective de ce travail a été évoquée à la fin de la section 3. Elle consiste, lors du choix d'un triplet (a, b, c), à considérer non seulement la compétence de (a, b) mais également celle de (a, c).

Une autre perspective de ce travail consiste à s'intéresser à l'utilisation d'idées dans le champ du RàPC consacré à l'acquisition de connaissances d'adaptation (cf. section 5). En particulier, on pourrait envisager d'améliorer l'approche décrite dans ce papier en s'appuyant sur l'utilisation d'une base de cas négatifs, comme c'est le cas dans [8].

Enfin, on peut également s'intéresser à la compétence pour associer à la solution proposée par le système de RàPC une indication sur sa plausibilité, qui sera d'autant meilleure que la compétence sera grande. L'intérêt de cela serait de voir comment combiner l'extrapolation utilisant la compétence avec d'autres approches du RàPC qui donne également une indication de plausibilité.

Références

[1] Aamodt, A. et E. Plaza: *Case-based Reasoning : Foundational Issues, Methodological Variations, and System Approaches*. AI Communications, 7(1) :39–59, 1994.

[2] Bounhas, M., H. Prade et G. Richard: *Analogical classification : A rule-based view*. Dans *Proc. of the International Conference on Information Processing*

and Management of Uncertainty in Knowledge-Based Systems, pages 485–495. Springer, 2014.

[3] Correa, W. F., H. Prade et G. Richard: *Trying to understand how analogical classifiers work*. Dans *Proc. of the International Conference on Scalable Uncertainty Management*, pages 582–589. Springer, 2012.

[4] Couceiro, M., N. Hug, H. Prade et G. Richard: *Analogy-preserving functions : a way to extend Boolean samples*. Dans *Proc. of the 26th Int. Joint Conf. on Artificial Intelligence (IJCAI'17)*, pages 1575–1581. Morgan Kaufmann, Inc., 2017.

[5] Craw, S., N. Wiratunga et R. C. Rowe: *Learning adaptation knowledge to improve case-based reasoning*. Artificial Intelligence, 170(16-17) :1175–1192, 2006.

[6] d'Aquin, M., F. Badra, S. Lafrogne, J. Lieber, A. Napoli et L. Szathmary: *Case base mining for adaptation knowledge acquisition*. Dans Veloso, M. M. (rédacteur) : *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI'07)*, pages 750–755. Morgan Kaufmann, Inc., 2007.

[7] Dubois, D., H. Prade et G. Richard: *Multiple-valued extensions of analogical proportions*. Fuzzy Sets and Systems, 292 :193–202, 2016. https://doi.org/10.1016/j.fss.2015.03.019.

[8] Gillard, T., J. Lieber et E. Nauer: *Improving Adaptation Knowledge Discovery by Exploiting Negative Cases : First Experiment in a Boolean Setting*. Dans *Proc. of ICCBR 2018 - 26th International Conference on Case-Based Reasoning*, Stockholm, Sweden, juillet 2018. https://hal.inria.fr/hal-01905077.

[9] Hanney, K. et M. T. Keane: *Learning adaptation rules from a case-base*. Dans Smith, I. et B. Faltings (rédacteurs) : *Advances in Case-Based Reasoning – Proc. of the Third Eur. Workshop, EWCBR'96*, LNAI 1168, pages 179–192. Springer Verlag, Berlin, 1996.

[10] Jarmulak, J., S. Craw et R. Rowe: *Using case-base data to learn adaptation knowledge for design*. Dans *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI'01)*, pages 1011–1016. Morgan Kaufmann, Inc., 2001.

[11] Kolodner, J.: *Case-Based Reasoning*. Morgan Kaufmann, Inc., 1993.

[12] Lepage, Y.: *Proportional analogy in written language data*. Dans Gala, N., R. Rapp et G. Bel-Enguix (rédacteurs) : *Language, Production, Cognition and the Lexicon*, Text, Speech and Language Technology 48, pages 151–173. Springer International Publishing Switzerland, 2014. https://link.springer.com/chapter/10.1007/978-3-319-08043-7_10.

[13] Lieber, J., E. Nauer, H. Prade et G. Richard: *Making the Best of Cases by Approximation, Interpolation and Extrapolation*. Dans *Proc. of ICCBR 2018 - 26th International Conference on Case-Based Reasoning*, tome 11156, pages 580–596, Stockholm, Sweden, juillet 2018. Springer. https://hal.inria.fr/hal-01905058.

[14] Prade, H. et G. Richard: *A discussion of analogical-proportion based inference*. Dans Sánchez-Ruiz, A. A. et A. Kofod-Petersen (rédacteurs) : *Proc. ICCBR'17 Workshops (CAW, CBRDL, PO-CBR), Doctoral Consortium, and Competitions co-located with the 25th Int. Conf. on Case-Based Reasoning (ICCBR'17), Trondheim, June 26-28*, tome 2028 de *CEUR Workshop Proceedings*, pages 73–82, 2017.

[15] Prade, H. et G. Richard: *Analogical proportions : From equality to inequality*. Int. J. Approx. Reasoning, 101 :234–254, 2018.

[16] Richter, M. M. et R. O. Weber: *Case-based reasoning, a textbook*. Springer, 2013.

[17] Smyth, B. et M. T. Keane: *Remembering to forget*. Dans *Proc. of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, Montréal, 1995.

[18] Stroppa, N. et F. Yvon: *Analogical learning and formal proportions : Definitions and methodological issues*. Technical Report D004, ENST-Paris, 2005.

[19] Zhu, J. et Q. Yang: *Remembering to add : competence-preserving case-addition policies for case base maintenance*. Dans *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99)*, pages 234–241, 1999.

Actes JIAF 2019

Une distance dans un ensemble fini ordonné et le cas du treillis de concepts

Laurent Miclet[1] **Nelly Barbot**[1] **Henri Prade**[2]

[1] IRISA ENSSAT Lannion Université Rennes 1, France
[2] IRIT, Université Toulouse 1 Capitole, France
laurent.miclet@gmail.com barbot@irisa.fr prade@irit.fr

Résumé

Cet article définit une distance dans un ensemble ordonné fini en s'appuyant sur l'analyse formelle des concepts (FCA). En particulier, son application aux treillis de concepts permet de munir ce treillis d'une distance cohérente avec sa structure. On étudie aussi les rapports entre cette distance et la notion de proportion analogique dans un treillis de concepts.

Abstract

We show in this paper how to associate a distance to a finite POS (partially ordered set), using the tools of Formal Concept Analysis (FCA). In particular, its application to concept lattices allows us to equip this lattice with a distance depending only on its structure. We also study the links between this distance and the notion of analogical proportion in a concept lattice.

1 Motivation et introduction

Le but de ce travail était au départ d'étendre aux treillis de concepts la notion de *dissimilarité analogique*, un outil qui s'est révélé utile dans l'utilisation de la proportion analogique à des fins de classification [13, 12] et de découverte d'analogies [14]. Pour cela, on a défini une distance entre concepts formels (la notion de proportion analogique entre concepts avait été étudiée auparavant, [15]). Cette distance, grâce à la complétion de Dedekind-Mac Neille et au théorème de Birkoff, s'est révélée pouvoir s'étendre aux ensembles finis partiellement ordonnés (POS) (sections 3 et 4).

La bibliographie sur la « métrisation » des POS est à notre connaissance assez mince, comme nous le verrons dans la dernière section, et n'avait jamais été abordée sous cet angle. L'intérêt est théorique : relier les notions fondamentales d'ordre et de distance est en soi un problème intéressant. Il est aussi pratique : il intéresse les questions de classification, supervisée ou surtout non-supervisée, les problèmes de préférences, etc. Nous n'avons pas exploré ces aspects appliqués pour le moment.

Pour poursuivre, nous avons analysé plus particulièrement les propriétés de cette distance sous l'angle des proportions analogiques (section 5) et de la dissimilarité analogique entre quadruplets de concepts (section 6).

Ce travail vise donc d'un côté à consolider les outils d'exploration analogique des treillis de concepts. D'un autre côté, il ouvre une perspective originale en proposant une distance calculée sur la structure même d'un POS.

2 Généralités

2.1 Ensembles finis ordonnés

Une relation binaire $\leq$ sur un ensemble T est un ordre (partiel) sur T si elle vérifie les trois propriétés suivantes :

1. Réflexivité : pour tout $x \in T$, $x \leq x$,
2. Antisymétrie : pour tous $x, y \in T$, ($x \leq y$ et $y \leq x$) impliquent $x = y$,
3. Transitivité : pour tous $x, y, z \in T$, ($x \leq y$ et $y \leq z$) impliquent $x \leq z$.

Nous ne traiterons ici que le cas où l'ensemble ordonné T est fini. Un élément y tel que $y \leq x$ est appelé minorant de x. L'ensemble des minorants de x est noté Tx. L'ensemble des minorants stricts $Tx\backslash\{x\}$ de x se note $(x[$. L'ensemble des majorants de x (éléments y tels que $x \leq y$) est noté xT et l'ensemble de ses majorants stricts $xT\backslash\{x\}$ se note $]x)$.

Une partie Y de T est minorée (resp. majorée) par x si x minore (resp. majore) tous les éléments de Y. Par convention, si la partie Y est vide, elle admet T comme ensemble de minorants et de majorants.

Un *père* de x est un élément y tel que [1] $x < y$ et qu'il n'existe pas d'élément z tel que $x < z < y$; on dit aussi que y *couvre* x. On définit un *fils* de façon duale.

Une *chaîne* est un sous-ensemble totalement ordonné de T, i.e. tout couple (x, y) d'éléments de cette chaîne vérifie $x \leq y$ ou $y \leq x$. Une chaîne de cardinal n a une *longueur* égale à $n-1$. Une *antichaîne* est un sous-ensemble dont les éléments sont incomparables deux à deux.

Une *extension* de la relation d'ordre $(T, \leq)$ est une relation d'ordre $(T', \leq)$, avec $T \subset T'$ et $x \leq y$ dans $(T, \leq)$ implique $x \leq y$ dans $(T', \leq)$.

Soit Y une partie d'un ensemble ordonné T, on dit que $r \in T$ est l'*infimum* de Y, noté $inf(Y)$, si l'ensemble de ses minorants est non vide et admet r pour maximum [2]. De même, Y a un *supremum* t si Y si l'ensemble de ses majorants est non vide et admet t pour minimum.

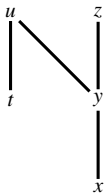
L'élément $x \in T$ est *sup-irréductible* s'il n'est pas supremum de la partie $(x[$, et $x \in T$ est *inf-irréductible* s'il n'est infimum de la partie $]x)$.

On note S_x l'ensemble des éléments sup-irréductibles qui minorent x et I_x l'ensemble des éléments inf-irréductibles qui majorent x.

Comme on le verra plus loin, la définition des éléments irréductibles est beaucoup plus simple dans le cas particulier où la relation d'ordre est un treillis.

Exemple

Soit la relation d'ordre $(T, \leq)$ donnée par le diagramme de Hasse suivant :



Par exemple : pour z, on a $(z[= \{y, x\}$ et $(z[$ est majorée par z et y. Le suprémum de z est y et z n'est pas suprémum de $(z[$, donc sup-irréductible.

De même, $(x[= \emptyset$ admet tous les éléments de T comme majorants, en particulier t et x. Or T n'admet pas de minorant, donc il n'y a pas de suprémum à $(x[$. Et donc x ne peut pas être suprémum de $]x)$. On en déduit que x est sup-irreductible.

Plus généralement, on a $S_T = \{x, y, z, t\}$ et $I_T = \{x, z, t, u\}$, et à titre d'exemple, on a $Tu = \{u, t, x, y\}$, $yT = \{y, z, u\}$, $S_u = \{u\}$ et $I_u = \{x, y\}$.

2.2 Graphes associés

On associe le graphe orienté acyclique (DAG) $G = (T, E)$ appelé *fermeture transitive* de $(T, \leq)$ en construisant pour tout $(x, y) \in T^2$ tel que $x < y$ un arc orienté de x vers y.

La *réduction transitive* de G est un graphe $G' = (V, E')$ avec un arc orienté de x vers y si et seulement si x est fils de y. Le diagramme de Hasse d'une relation d'ordre est le graphe de sa réduction transitive, dans lequel un fils est toujours à une hauteur inférieure à celle de son père.

2.3 Treillis

Un treillis est un ensemble T muni de deux lois internes notées $\vee$ (*join*) et $\wedge$ (*meet*) telles que :

— les deux lois sont commutatives et associatives ;

— pour tous x et y de T : $x \wedge (x \vee y) = x = x \vee (x \wedge y)$
$x \wedge (x \vee y) = x = x \vee (x \wedge y)$ (lois d'absorption).

La loi d'absorption entraîne l'idempotence de tout élément x de T pour les deux lois : $a \vee a = a$ et $a \wedge a = a$.

À partir d'une telle structure on peut définir sur T une relation d'ordre notée $\leq$ par l'équivalence $(a \leq b) \Leftrightarrow (a \vee b = b)$.

Un treillis est donc aussi un ensemble partiellement ordonné dans lequel chaque paire d'éléments admet une borne supérieure et une borne inférieure. Un treillis est dit *complet* si toute partie admet un supremum et un infimum.

Quand un treillis $(T, \wedge, \vee, \leq)$ est fini, il est complet, et il possède en particulier un plus grand élément $\top$ et un plus petit élément $\bot$

Un élément d'un treillis fini est *sup-irréductible* si et seulement s'il possède un unique fils [3]. L'ensemble des sup-irréductibles est noté S_T. Dualement, un $x \in T$ est inf-irréductible si et seulement s'il admet un unique père. Les éléments *inf-irreductibles* sont définis dualement et leur ensemble est noté I_T.

2.4 Treillis de concepts

L'analyse formelle de concepts (FCA) étudie les relations binaires R définies entre un ensemble O d'objets et un ensemble $\mathcal{A}$ d'attributs. Le triplet $(O, \mathcal{A}, R)$ est appelé *contexte (formel)*. On note $(o, a) \in R$ ou oRa pour signifier que l'objet o possède l'attribut a. On note $o^\uparrow = \{a \in \mathcal{A} \,|\, (o, a) \in R\}$ l'ensemble des attributs de l'objet o et $a^\downarrow = \{o \in O \,|\, (o, a) \in R\}$ l'ensemble des objets qui possèdent l'attribut a.

De même, pour un sous-ensemble $\mathbf{o}$ d'objets, $\mathbf{o}^\uparrow$ est défini comme $\{a \in \mathcal{A} \,|\, a^\downarrow \supseteq \mathbf{o}\}$. Un *concept (formel)* est un couple $(\mathbf{o}, \mathbf{a})$ tel que $\mathbf{a}^\downarrow = \mathbf{o}$ and $\mathbf{o}^\uparrow = \mathbf{a}$. On nomme $\mathbf{o}$ l'*extension* du concept et $\mathbf{a}$ son *intension*.

On note un concept $x = (\mathbf{o}_x, \mathbf{a}_x)$ ou $x = \boxed{\begin{matrix}\mathbf{o}_x \\ \mathbf{a}_x\end{matrix}}$.

1. Par définition, $(x < y) \Leftrightarrow (x \leq y$ et $x \neq y)$.

2. Un élément maximal dans une partie $X \subset T$ est un élément de X tel qu'il n'existe aucun autre élément de X qui lui soit supérieur

3. Un élément a d'un treillis T non nécessairement fini est dit *sup-irréductible* s'il n'est pas l'élément minimal (s'il existe) de T et si $x < a$ et $y < a$ implique $x \vee y < a$. Les *inf-irreductibles* sont définis dualement. Cette définition est équivalente à celle donnée pour une relation d'ordre et à celle que l'on vient de donner pour un treillis fini.

L'ensemble des concepts associés à un contexte possède une relation d'ordre définie comme : $(\mathbf{o}_1, \mathbf{a}_1) \leq (\mathbf{o}_2, \mathbf{a}_2)$ ssi $\mathbf{o}_1 \subseteq \mathbf{o}_2$ (ou, de manière équivalente, $\mathbf{a}_2 \subseteq \mathbf{a}_1$). Cet ensemble est un treillis fini, dit *le treillis de concepts* (*concept lattice*) of R. On le note $\mathcal{B}(O, \mathcal{A}, R)$.

La notion de *Weak Analogical Proportion* ou *WAP* a été introduite dans [15] pour décrire des relations d'analogie (plus exactement, de proportion analogique) entre quatre concepts formels.

Definition 1. *Dans un treillis, la WAP* $(x : y \overset{WAP}{::} z : t)$ *est caractérisée par les égalités :* $x \wedge t = y \wedge z$ *et* $x \vee t = y \vee z$.

Dans un treillis de concepts, on a la propriété suivante.

Proposition 1. *Quatre concepts* $x = (\mathbf{o}_x, \mathbf{a}_x)$, $y = (\mathbf{o}_y, \mathbf{a}_y)$, $z = (\mathbf{o}_z, \mathbf{a}_z)$ *et* $t = (\mathbf{o}_t, \mathbf{a}_t)$ *vérifient* $(x : y \overset{WAP}{::} z : t)$ *ssi*

$$\mathbf{o}_x \cap \mathbf{o}_t = \mathbf{o}_y \cap \mathbf{o}_z \quad et \quad \mathbf{a}_x \cap \mathbf{a}_t = \mathbf{a}_y \cap \mathbf{a}_z\,.$$

Cette propriété se déduit du « Main Theorem of Formal Concepts » [11].

2.5 Tout treillis est un treillis de concepts

Un résultat fondamental dû à G. Birkhoff [5, 2] établit que tout treillis fini T est isomorphe au treillis de concepts du contexte $(J_T, M_T, \leq)$, où J_T est l'ensemble des sup-irréductibles de T et M_T celui de ses inf-irréductibles. De plus, ce contexte est le contexte *réduit* (voir ci-dessous).

On peut calculer les concepts associés aux éléments d'un treillis en utilisant la construction de Birkhoff associée au résultat indiqué ci-dessus, mais aussi avec la technique décrite à la section suivante qui évite la construction complète du treillis de concepts.

Le contexte minimal en termes d'objets et d'attributs produisant le même treillis de concepts qu'un contexte donné est appelé contexte *réduit* ou *standard*. Il est unique à un isomorphisme près. Quand on a un contexte quelconque, on peut le réduire en éliminant d'abord les lignes et les colonnes en double, puis celles qui sont l'intersection de deux autres, voir [11], c'est-à-dire en enlevant les éléments non irréductibles de la relation d'ordre ; la structure du treillis est évidemment inchangée. Cette opération permet d'assurer la bijection pour tout élément (concept) x du treillis entre $\mathbf{o}_x$ et S_x, comme entre $\mathbf{a}_x$ et I_x (voir le Théorème 5.1 (6) de [7]).

Pour résumer :

— À tout treillis on peut associer un treillis de concepts comportant des ensembles d'objets et d'attributs minimaux, correspondant à un contexte réduit.

— Tout contexte peut être réduit sans changer la structure du treillis de concepts associé et en minimisant les ensembles d'objets et d'attributs associés à ce treillis.

— Dans un treillis pour lequel les ensembles d'objets et d'attributs sont minimaux, il y a bijection entre $\mathbf{o}_x$ et S_x, comme entre $\mathbf{a}_x$ et I_x.

2.6 Construction directe des concepts minimaux d'un treillis

Entrée : un treillis fini.

Sortie : le même treillis où chaque élément est détaillé comme un concept.

Algorithme

Partir du plus petit élément $\perp$ et lui affecter l'ensemble vide. Pour chaque père de $\perp$, créer un nouvel objet et lui affecter. Remonter de la sorte dans le treillis en respectant la règle suivante :

1. Si l'élement traité n'est pas sup-irréductible, lui affecter l'union des objets de ses fils.
2. S'il est sup-irréductible, lui affecter les objets de son fils, plus un nouvel objet.

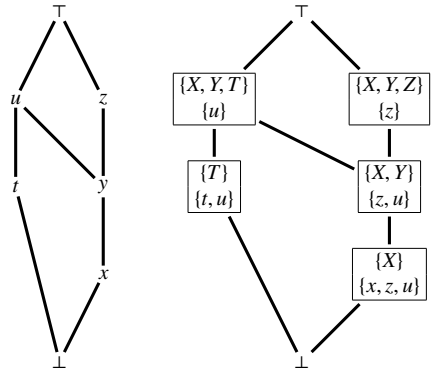
La construction des ensembles d'attributs se fait de façon duale. En notant S_x l'ensemble des sup-irréductibles inférieurs ou égaux à un élément x et I_x l'ensemble des inf-irréductibles supérieurs ou égaux à x, cet algorithme assure que pour tout x on a $\mathbf{o}_x$ et S_x en bijection, comme $\mathbf{a}_x$ et I_x.

2.7 Exemple

Soit le treillis $(T, \wedge, \vee, \leq)$ décrit par le diagramme de Hasse à gauche dans la figure ci-dessous. Comme on le verra à la section 4.1, ce treillis est la *complétion* en treillis de la relation d'ordre donnée en exemple à la section 2.1.

Les irréductibles de ce treillis sont les mêmes que ceux de la relation d'ordre dont il est la complétion : $S_T = \{x, y, z, t\}$ et $I_T = \{x, z, t, u\}$.

Le treillis de concepts construit par la méthode présentée ci-dessus est décrit à droite dans la figure ci-dessous. Afin d'y distinguer objets et attributs, on pose $O = \{X, Y, Z, T, U\}$ et $\mathcal{A} = \{x, y, z, t, u\}$ et on note $T_{O,\mathcal{A}}$ le treillis de concepts. Pour tout élément s de T (treillis décrit à gauche), les objets du concept associé dans $T_{O,\mathcal{A}}$ forment maintenant l'ensemble S_s des sup-irréductibles minorants de s dans T ; ses attributs forment l'ensemble I_s des inf-irréductibles majorants de s dans T.



Par abus de langage, grâce à ce parallèle entre le treillis T et le treillis de concepts $T_{O,\mathcal{A}}$, la lettre x peut désigner un élément de T, un concept de $T_{O,\mathcal{A}}$, voire, dans cet exemple, un attribut. Ici, on a donc $x = (\mathbf{o}_x, \mathbf{a}_x) = (\{X\}, \{x, z, u\})$.

3 Distance dans un treillis

3.1 Définition

Une distance sur un ensemble T est une application $d : T^2 \to \mathbb{R}$ positive, telle que, pour tout $(x, y, z) \in T^3$:

1. $d(x, y) = 0$ si et seulement si $x = y$,
2. $d(x, y) = d(y, x)$,
3. $d(x, y) \leq d(x, z) + d(z, y)$.

Comme on l'a vu, chaque élément x d'un treillis peut être décrit comme un concept, c'est-à-dire à l'aide d'un sous-ensemble d'objets $\mathbf{o}_x$ et d'un sous-ensemble minimal d'attributs $\mathbf{a}_x$. Nous nous appuyons sur ces ensembles pour définir une distance de la façon suivante :

Proposition 2. *Soit δ_H la distance de Hamming entre sous-ensembles d'un ensemble fini, c'est-à-dire le cardinal de leur différence symétrique Δ, les formules suivantes définissent une distance dans un treillis :*

$$\delta(x, y) = \delta_H(\mathbf{o}_x, \mathbf{o}_y) + \delta_H(\mathbf{a}_x, \mathbf{a}_y) = |\mathbf{o}_x \Delta \mathbf{o}_y| + |\mathbf{a}_x \Delta \mathbf{a}_y|$$

Démonstration. δ est bien une distance puisque, comme somme de deux distances. □

3.2 Contexte réduit ou non ?

Il est à noter que si le treillis a été construit à partir d'un contexte, deux cas peuvent se présenter. Le premier est celui d'un contexte réduit. La distance δ peut alors s'écrire

$$\delta(x, y) = |I_x \ \Delta \ I_y| + |S_x \ \Delta \ S_y|$$

$\mathbf{o}_x$ et S_x étant en bijection, tout comme $\mathbf{a}_x$ et I_x.

Dans le cas inverse, et si le contexte a été obtenu par complétion (voir plus loin la section 4.1) d'une relation d'ordre T, $\mathbf{o}_x$ et xT sont en bijection, comme $\mathbf{a}_x$ et Tx et l'on a alors

$$\delta(x, y) = |xT \ \Delta \ yT| + |Tx \ \Delta \ Ty|.$$

Dans la suite, nous adopterons l'hypothèse que les ensembles d'objets et d'attributs sont toujours minimaux, ce qui signifie en particulier que si le treillis a été construit à partir d'un contexte, ce dernier a été réduit.

Notons cependant que toutes les propriétés suivantes sont également valables si on adopte l'hypothèse inverse.

Si on est dans le cadre de la FCA, la réduction du contexte enlève de l'information : on peut alors préférer calculer la distance à partir du contexte original.

3.3 Propriété fondamentale

Proposition 3. *Pour tous x, y et z tels que $x \leq y \leq z$:*

$$\delta(x, z) = \delta(x, y) + \delta(y, z) \tag{1}$$

Démonstration. Les cas d'égalité entre x, y ou z sont triviaux. Supposons $x < y < z$: $\mathbf{o}_x$ est donc strictement inclus dans $\mathbf{o}_y$ et l'on écrit $\mathbf{o}_y = \mathbf{o}_x \cup O_1$ où $O_1 \cap \mathbf{o}_x = \emptyset$. De même, $\mathbf{o}_z = \mathbf{o}_y \cup O_2$ où O_2 est un ensemble d'objets en intersection vide avec $\mathbf{o}_y$ et donc avec $\mathbf{o}_x$. On a : $\delta_H(\mathbf{o}_x, \mathbf{o}_y) = |O_1|$, $\delta_H(\mathbf{o}_y, \mathbf{o}_z) = |O_2|$ et $\delta_H(\mathbf{o}_x, \mathbf{o}_z) = |O_1| + |O_2|$. En raisonnant de même avec les attributs, la formule est établie. □

Cette propriété assure que toute chaîne $(x = x_1 \leq \ldots \leq x_n = y)$ entre deux éléments x et y cumule la même distance ; ceci permet de munir le treillis de l'application v de T dans $\mathbb{R}^+$ définie par $v(x) = \delta(x, \bot)$. La distance entre deux éléments ordonnés s'exprime alors comme :

$$x \leq y \ \Leftrightarrow \ \delta(x, y) = v(y) - v(x) \tag{2}$$

avec v monotone isotone, puisque $x \leq y \Leftrightarrow v(x) \leq v(y)$.

3.4 Remarques

— On a aussi :

$$\delta(x, y) = |\mathbf{o}_x| + |\mathbf{o}_y| + |\mathbf{a}_x| + |\mathbf{a}_y| - 2 \cdot |\mathbf{o}_{x \wedge y}| - 2 \cdot |\mathbf{a}_{x \vee y}|$$

— Quand on dispose d'un contexte, on peut le réduire. Cette opération permet d'assurer la bijection pour tout élément x du treillis entre $\mathbf{o}_x$ et S_x, comme entre $\mathbf{a}_x$ et I_x.

— Caspard, Leclerc et Monjardet citent une distance dans les inf-demi-treillis [4] distributifs, dite de la *différence symétrique* qu'ils utilisent pour montrer qu'un tel treillis est *rangé*.

> Soit T un inf-demi treillis distributif et S_t l'ensemble des sup-irréductibles inférieurs ou égaux à un élément t de T. La distance de la différence symétrique sur T est définie par : $\forall t$ et $t' \in T : \Delta(t, t') = |S_t \Delta S'_t|$. ([7], page 231)

La distance que nous proposons généralise donc la leur, puisque $\mathbf{o}_x$ et S_x sont en bijection sous l'hypothèse (que nous faisons) que les ensembles d'objets et d'attributs sont minimaux.

— Une autre distance envisageable serait la longueur du plus court chemin entre x et y dans le diagramme de Hasse. Mais ce chemin ne passe pas nécessairement par $x \vee y$ ni par $x \wedge y$; sa longueur ne se calcule pas non plus directement à partir des ensembles d'objets

4. Un inf-demi-treillis est un ensemble ordonné dans lequel deux éléments quelconques admettent toujours une borne inférieure.

et d'attributs associés à x et y ; enfin, il a la même longueur pour des positions relatives pourtant bien différentes (voir la section 4.2 pour un exemple).

— Dans un treillis distributif, la distance δ et celle du plus court chemin sont identiques ; c'est une conséquence du théorème 5.1 (6) de [7].

4 Distance dans un ensemble ordonné fini quelconque

4.1 Construction d'un contexte et calcul de la distance

La complétion de Dedekind-MacNeille d'un ensemble ordonné $(T, \leq)$ est le plus petit treillis complet qui contient cette relation d'ordre. Une façon de le construire, donnée par exemple dans [11], consiste à construire le contexte $(O, \mathcal{A}, \leq)$ avec $O = \mathcal{A} = T$ puis à calculer le treillis de concepts $\mathcal{B}(O, \mathcal{A}, \leq)$ associé.

Le contexte $(O, \mathcal{A}, \leq)$ n'est pas obligatoirement réduit. Notre choix est, rappelons-le, de le réduire systématiquement (voir la section 3.2).

Tout élément $x \in T$ joue le rôle d'un objet dans le contexte $(O, \mathcal{A}, \leq)$ (où $O = \mathcal{A} = T$) et on peut lui associer le concept $\left(x^{\uparrow}, (x^{\uparrow})^{\downarrow}\right)$. En particulier, $x^{\uparrow}$ correspond à xT et $(x^{\uparrow})^{\downarrow}$ à Tx. La distance entre deux éléments x et y de T s'écrit donc :

$$\delta(x, y) = \left|(x^{\uparrow})^{\downarrow} \,\Delta\, (y^{\uparrow})^{\downarrow}\right| + \left|x^{\uparrow} \,\Delta\, y^{\uparrow}\right|$$

δ correspond bien à une distance sur T^2 : pour tout $(x, y) \in T^2$, $\delta(x, y) = 0$ signifie que $x^{\uparrow} = y^{\uparrow}$, d'où $x \leq y$ et $y \leq x$, et donc $x = y$. La symétrie de δ et l'inégalité triangulaire découlent de la proposition 2 car

$$\delta(x, y) = \delta\left(\left(x^{\uparrow}, (x^{\uparrow})^{\downarrow}\right), \left(y^{\uparrow}, (y^{\uparrow})^{\downarrow}\right)\right).$$

Le calcul de la distance entre deux éléments de T peut donc se faire directement dans le contexte, sans construire le treillis, ni calculer les irréductibles de la relation.

La figure 1 récapitule le calcul de la distance à partir d'un ensemble ordonné, d'un contexte et d'un treillis.

4.2 Exemple

Considérons de nouveau l'exemple d'ensemble ordonné $(T, \leq)$ présenté à la section 2.1. Le contexte obtenu par complétion de Dedekind-Mac Neille est donné par

	x	y	z	t	u
X	×	×	×		×
Y		×	×		×
Z			×		
T				×	×
U					×

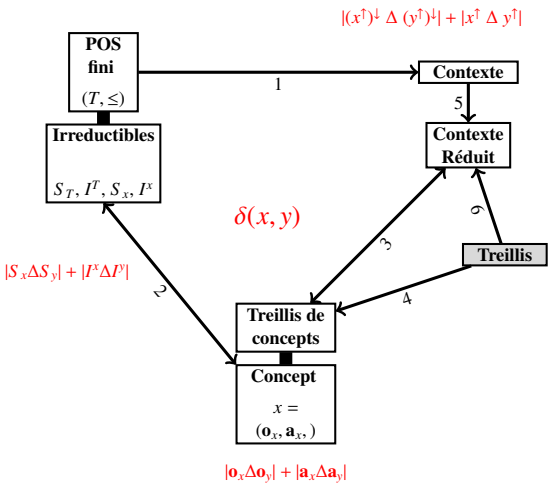
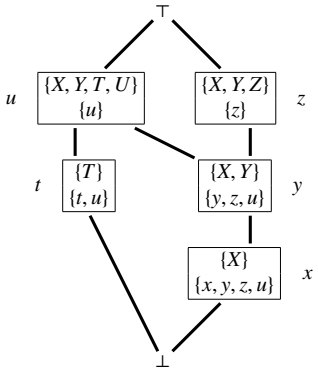


FIGURE 1 – Trois façons équivalentes de définir une distance dans un *POS* fini et une façon de le faire dans un treillis (en grisé). Les étiquettes s'interprètent ainsi. (1) Complétion de Dedekind- Mc Neille. (2) Isomorphisme. (3) Construction du treillis de concepts à partir du contexte et vice-versa. (4) Tout treillis est un treillis de concepts. (5) Réduction du contexte. (6) Construction d'un contexte réduit à partir d'un treillis.

Les objets de ce contexte sont notés en majuscule et les attributs en minuscule, mais il s'agit des mêmes éléments de T. La complétion a créé dans cet exemple les concepts $\top$ et $\bot$ pour augmenter T en treillis.



La réduction de ce contexte enlève l'objet U et l'attribut y, qui ne sont pas irréductibles et produit le treillis déjà construit à la section 2.7. Le tableau de distances δ dans $(T, \leq)$ est détaillé ci-dessous. Pour comparaison, on a ajouté le tableau des distances en utilisant le double de la longueur du plus court chemin, qui est moins « précise » que δ : elle donne par exemple la même distance entre y et t que entre z et u, ou entre entre x et t et entre z et t, alors que δ marque une différence justifiée par la structure du treillis. Cette remarque est justifiée expérimentalement à la fin de la section 6.4.

δ	x	y	z	t	u
x	0	2	4	5	4
y		0	2	5	2
z			0	7	4
t				0	3
u					0

$2pcc$	x	y	z	t	u
x	0	2	4	4	4
y		0	2	4	2
z			0	6	4
t				0	2
u					0

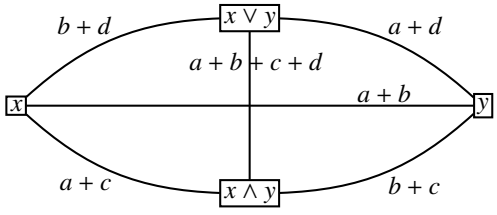
5 Quelques propriétés de la distance δ

Dans cette seconde moitié de l'article, nous appliquons la distance introduite dans un treillis de concepts à l'étude des proportions analogiques entre concepts [15]. Une proportion analogique dans un treillis est une relation quaternaire, dénotée $x : y :: z : t$, qui se lit « x est à y comme z est à t », et qui est définie dans sa forme faible par les conditions $x \wedge t = y \wedge z$ et $x \vee t = y \vee z$. Voir [15] pour la forme forte et des explications. Ceci garantit le rôle symétrique des paires (x, y) et (z, t) et la permutabilité de y et z d'une part et de x et t d'autre part. Quelques propriétés de la distance de Hamming entre les éléments d'une proportion avaient été établies dans le cadre de treillis booléens [8].

5.1 Proportions canoniques

On s'intéresse d'abord au cas de ce qui est appelé proportion canonique dans [15], c'est-à-dire aux WAP de la forme $(x : x \vee y \overset{WAP}{::} x \wedge y : y)$.

Proposition 4. *Soit x et y deux éléments distincts d'un treillis de concepts, il existe 4 nombres positifs a, b, c et d tels que les distances deux à deux entre x, y, $x \vee y$ et $x \wedge y$ se décomposent de la façon suivante :*



Démonstration. Il suffit de poser
$a = |\mathbf{o}_x \backslash \mathbf{o}_y| + |\mathbf{a}_y \backslash \mathbf{a}_x| \quad b = |\mathbf{o}_y \backslash \mathbf{o}_x| + |\mathbf{a}_x \backslash \mathbf{a}_y|$
$c = |\mathbf{a}_{x \wedge y} \backslash (\mathbf{a}_x \cup \mathbf{a}_y)| \quad d = |\mathbf{o}_{x \vee y} \backslash (\mathbf{o}_x \cup \mathbf{o}_y)|$ □

Ceci permet de représenter la répartition des distances entre les éléments d'une proportion canonique comme sur la figure 2, où sont ajoutés deux éléments (en gris) notés $x \sqcup y$ et $x \sqcap y$ qui ne sont pas en général des concepts. On peut remarquer que, dans cette figure, c et d peuvent être nuls, mais pas a ni b (car $x \neq y$).

Proposition 5. *Si $v(x) \geq v(y)$ alors*
$\delta(x, y) = v(x) - v(y) + 2 \cdot \left(|\mathbf{o}_y \backslash \mathbf{o}_{x \wedge y}| + |\mathbf{a}_x \backslash \mathbf{a}_{x \vee y}| \right)$

Démonstration. Cela découle de la proposition 4. □

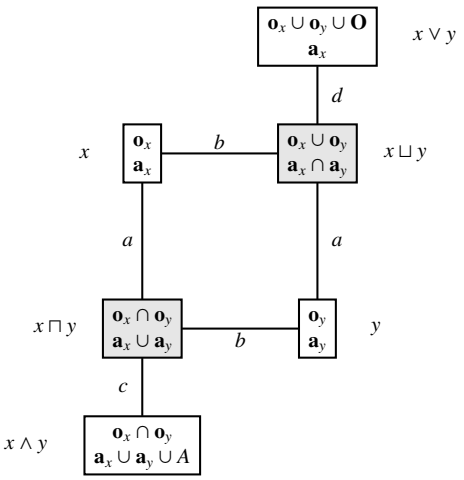


FIGURE 2 – Distances dans une proportion canonique, avec les notations $A = \mathbf{a}_{x \wedge y} \backslash (\mathbf{a}_x \cup \mathbf{a}_y)$ et $O = \mathbf{o}_{x \vee y} \backslash (\mathbf{o}_x \cup \mathbf{o}_y)$.

Proposition 6. *Pour tous concepts x et y*

$$|v(x) - v(y)| \leq \delta(x, y) \leq \delta(x \wedge y, x \vee y) = v(x \vee y) - v(x \wedge y)$$

Démonstration. Cela se déduit des propositions 4 et 5 et de l'égalité (2). □

Proposition 7. *Pour tous concepts x, y et z*

$$\delta(x, y) \leq \delta(x \vee z, y \wedge z) + \delta(x \wedge z, y \vee z) \qquad (3)$$

Démonstration. C'est une conséquence de l'inégalité triangulaire, la proposition 6 et l'égalité (2). □

5.2 Inégalités dans une *WAP*

Quand le treillis est muni de la distance δ, un certain nombre de propriétés métriques permettent de mieux cerner la notion de *WAP*.

Proposition 8. *Dans la WAP $(x : y \overset{WAP}{::} z : t)$, on a :*
$\delta(x, t) + \delta(y, z) \leq \delta(x, y) + \delta(x, z) + \delta(y, t) + \delta(z, t)$
$\leq 2(\delta(x, t) + \delta(y, z))$

Démonstration. La première inégalité est vraie pour tout quadruplet (la preuve est triviale), mais pas la seconde. Par définition de la *WAP*, on a $x \vee t = y \vee z$ et $x \wedge t = y \wedge z$, d'où en particulier $\mathbf{o}_x \cap \mathbf{o}_t = \mathbf{o}_z \cap \mathbf{o}_y$. Les quatre ensembles d'objets, sous cette contrainte, se divisent en 9 sous-ensembles disjoints comme l'illustre la figure 3.

Afin de simplifier les notations, on notera par exemple : $\mathbf{o}_x = AFHI$. Notons $\delta_o(x, y) = \mathbf{o}_x \Delta \mathbf{o}_y$ et $\delta_a(x, y) = \mathbf{a}_x \Delta \mathbf{a}_y$. On a donc

$\delta_o(x, t) = |ADEFGH| \quad \delta_o(x, y) = |ABFG|$
$\delta_o(x, z) = |ACEH| \quad \delta_o(y, z) = |BCEFGH|$
$\delta_o(y, t) = |BDEH| \quad \delta_o(z, t) = |CDFG|$

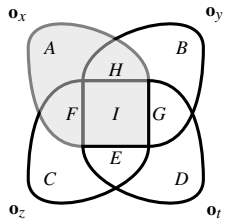


FIGURE 3 – Diagramme de Venn de $\mathbf{o}_x$, $\mathbf{o}_y$, $\mathbf{o}_z$ et $\mathbf{o}_t$ lorsque les concepts associés vérifient $x \wedge t = y \wedge z$. On a par exemple $\mathbf{o}_x = A \cup F \cup H \cup I$.

Ainsi, $\delta_o(x,t) + \delta_o(y,z) = |ABCD| + 2 \cdot |EFGH|$ et $\delta_o(x,y) + \delta_o(x,z) + \delta_o(y,z) + \delta_o(z,t) = 2 \cdot |ABCD| + 2 \cdot |EFGH|$. D'où les inégalités de la proposition 8 pour δ_o, les 9 sous-ensembles étant disjoints.
Par symétrie, on établit les mêmes inégalités pour δ_a et donc pour leur somme δ. □

Remarques

— Les inégalités de la proposition 8 sont vraies en géométrie du plan pour un quadrilatère (x, y, z, t) de sommets opposés x et t, à condition qu'il soit convexe.
On peut remarquer aussi que si (x, y, z, t) est un parallélogramme, le quadrilatère $(\overrightarrow{Ox} + \overrightarrow{Oy}, \overrightarrow{Ox} + \overrightarrow{Oz}, \overrightarrow{Oy} + \overrightarrow{Ot}, \overrightarrow{Oz} + \overrightarrow{Ot})$ est aussi un parallélogramme, dont l'intersection des diagonales est située en $\overrightarrow{Ox} + \overrightarrow{Ot} = \overrightarrow{Oy} + \overrightarrow{Oz}$.

— Si (x, y, z, t) n'est pas une *WAP* dans cet ordre, la seconde inégalité est fausse en général.

5.3 Caractérisation d'une *WAP*

Soient x, y, z et t quatre concepts, on note $\overset{\star}{\mathbf{o}}_x^{xyzt}$ (ou $\overset{\star}{\mathbf{o}}_x$ si aucune confusion n'est possible) l'ensemble $(\mathbf{o}_x \cup \mathbf{o}_y \cup \mathbf{o}_z \cup \mathbf{o}_t) \setminus (\mathbf{o}_y \cup \mathbf{o}_z \cup \mathbf{o}_t)$ des objets que x possède et que ni y, ni z, ni t ne possèdent.

Proposition 9. $(x : y \overset{WAP}{::} z : t)$ *implique :*

1. $\delta(x,y) - |\overset{\star}{\mathbf{o}}_x| - |\overset{\star}{\mathbf{o}}_y| - |\overset{\star}{\mathbf{a}}_x| - |\overset{\star}{\mathbf{a}}_y| = \delta(z,t) - |\overset{\star}{\mathbf{o}}_z| - |\overset{\star}{\mathbf{o}}_t| - |\overset{\star}{\mathbf{a}}_z| - |\overset{\star}{\mathbf{a}}_t|$ *et*
2. $\delta(x,z) - |\overset{\star}{\mathbf{o}}_x| - |\overset{\star}{\mathbf{o}}_z| - |\overset{\star}{\mathbf{a}}_x| - |\overset{\star}{\mathbf{a}}_z| = \delta(y,t) - |\overset{\star}{\mathbf{o}}_y| - |\overset{\star}{\mathbf{o}}_t| - |\overset{\star}{\mathbf{a}}_y| - |\overset{\star}{\mathbf{a}}_t|$ *et*
3. $\mathbf{o}_x \setminus (\overset{\star}{\mathbf{o}}_x \cup \mathbf{o}_y \cup \mathbf{o}_z) = \mathbf{o}_y \setminus (\overset{\star}{\mathbf{o}}_y \cup \mathbf{o}_x \cup \mathbf{o}_t) = \emptyset$ *et*
4. $\mathbf{a}_x \setminus (\overset{\star}{\mathbf{a}}_x \cup \mathbf{a}_y \cup \mathbf{a}_z) = \mathbf{a}_y \setminus (\overset{\star}{\mathbf{a}}_y \cup \mathbf{a}_x \cup \mathbf{a}_t) = \emptyset$

Démonstration. Par définition de la *WAP*. □

En notant $\alpha = \delta(x,y) - |\overset{\star}{\mathbf{o}}_x| - |\overset{\star}{\mathbf{o}}_y| - |\overset{\star}{\mathbf{a}}_x| - |\overset{\star}{\mathbf{a}}_y|$ et $\beta = \delta(y,t) - |\overset{\star}{\mathbf{o}}_y| - |\overset{\star}{\mathbf{o}}_t| - |\overset{\star}{\mathbf{a}}_y| - |\overset{\star}{\mathbf{a}}_t|$, on a, dans la WAP $(x : y \overset{WAP}{::} z : t)$

$\delta(x,t) - |\overset{\star}{\mathbf{o}}_x| - |\overset{\star}{\mathbf{o}}_t| - |\overset{\star}{\mathbf{a}}_x| - |\overset{\star}{\mathbf{a}}_t| =$
$\delta(y,z) - |\overset{\star}{\mathbf{o}}_y| - |\overset{\star}{\mathbf{o}}_z| - |\overset{\star}{\mathbf{a}}_y| - |\overset{\star}{\mathbf{a}}_z| = \alpha + \beta$

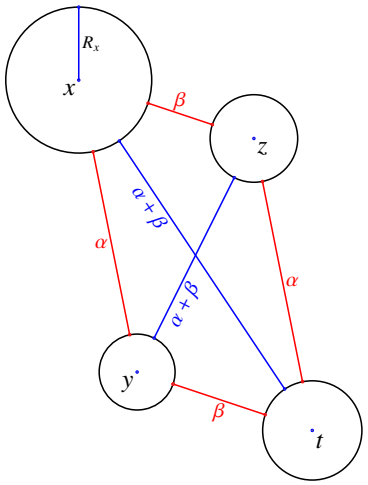


FIGURE 4 – La distance δ dans une *WAP*. Le rayon de cercle du centre x vaut $R_x = |\overset{\star}{\mathbf{o}}_x| + |\overset{\star}{\mathbf{a}}_x|$. La distance entre le cercle de centre x et celui de centre y est égale à α, comme celle entre le cercle de centre z et celui de centre t.

6 Dissimilarité Analogique

6.1 Principe

La *Dissimilarité Analogique* $DA(x,y,z,t)$ relative à la *WAP* entre quatre éléments d'un treillis pris dans cet ordre cherche à mesurer de combien ces quatre éléments « manquent » la *WAP*. Cette notion a été utilisée en particulier dans [13] pour des problèmes de classification supervisée basée sur les proportions analogiques, avec des données binaires. Elle est également utile pour découvrir des proportions analogiques dans des données, par des techniques de type « descente de gradient » (voir [14]).
Entre concepts formels, on peut donc la définir par :

$$\begin{aligned} DA(x,y,z,t) &= \delta(x \vee t, y \vee z) + \delta(x \wedge t, y \wedge z) \\ &= \delta_H(\mathbf{o}_{\mathbf{x}\vee\mathbf{t}}, \mathbf{o}_{\mathbf{y}\vee\mathbf{z}}) + \delta_H(\mathbf{o}_{\mathbf{x}\wedge\mathbf{t}}, \mathbf{o}_{\mathbf{y}\wedge\mathbf{z}}) \\ &\quad + \delta_H(\mathbf{a}_{\mathbf{x}\vee\mathbf{t}}, \mathbf{a}_{\mathbf{y}\vee\mathbf{z}}) + \delta_H(\mathbf{a}_{\mathbf{x}\wedge\mathbf{t}}, \mathbf{a}_{\mathbf{y}\wedge\mathbf{z}}) \end{aligned}$$

Une version simplifiée est :

$$\begin{aligned} DA_s(x,y,z,t) &= \delta_H(\mathbf{o}_{\mathbf{x}\wedge\mathbf{t}}, \mathbf{o}_{\mathbf{y}\wedge\mathbf{z}}) + \delta_H(\mathbf{a}_{\mathbf{x}\vee\mathbf{t}}, \mathbf{a}_{\mathbf{y}\vee\mathbf{z}}) \\ &= \delta_H(\mathbf{o}_{\mathbf{x}} \cap \mathbf{o}_{\mathbf{t}}, \mathbf{o}_{\mathbf{y}} \cap \mathbf{o}_{\mathbf{z}}) + \delta_H(\mathbf{a}_{\mathbf{x}} \cap \mathbf{a}_{\mathbf{t}}, \mathbf{a}_{\mathbf{y}} \cap \mathbf{a}_{\mathbf{z}}) \\ &= |(\mathbf{o}_{\mathbf{x}} \cap \mathbf{o}_{\mathbf{t}}) \Delta (\mathbf{o}_{\mathbf{y}} \cap \mathbf{o}_{\mathbf{z}})| + |(\mathbf{a}_{\mathbf{x}} \cap \mathbf{a}_{\mathbf{t}}) \Delta (\mathbf{a}_{\mathbf{y}} \cap \mathbf{a}_{\mathbf{z}})| \end{aligned}$$

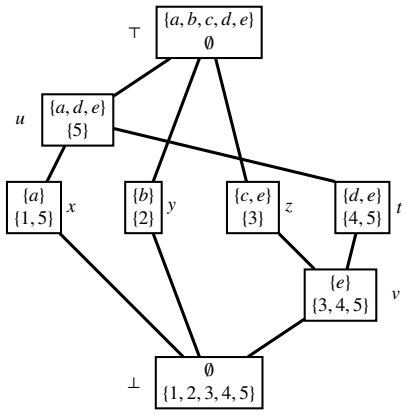
Comme on va le démontrer plus loin, $DA(x,y,z,t)$ et $DA_s(x,y,z,t)$ sont nulles si et seulement si on a une *WAP* de la forme $(x : y \overset{WAP}{::} z : t)$ et sont invariantes pour les 8

permutations de la proportion analogique ($x : y :: z : t$ a 8 formes équivalentes à cause des propriétés requises de symétrie et de permutation centrale).

Par exemple, dans le contexte suivant et le treillis correspondant, on obtient les valeurs suivantes :

	1	2	3	4	5
a	×				×
b		×			
c			×		
d				×	×
e			×	×	×

	DA	DA_s
(x, y, z, t)	3	1
(x, y, z, u)	7	2
(x, y, z, v)	3	1



6.2 Définition et propriétés

Dans la suite, de préférence à DA, on utilisera $DA_s(x, y, z, t) = |(\mathbf{o}_x \cap \mathbf{o}_t)\Delta(\mathbf{o}_y \cap \mathbf{o}_z)| + |(\mathbf{a}_x \cap \mathbf{a}_t)\Delta(\mathbf{a}_y \cap \mathbf{a}_z)|$, en particulier pour la proposition 11.

Proposition 10. *1.* $DA_s(x, y, z, t) = 0 \Leftrightarrow (x : y \overset{WAP}{::} z : t)$.

2. *$DA_s(x, y, z, t)$ est invariante pour les 8 permutations de la proportion analogique.*
3. *En général, $DA_s(x, y, z, t) \neq DA_s(y, x, z, t)$.*

Démonstration. Conséquence de la proposition 1. □

Proposition 11. *Soit cinq éléments x, y, z, t et t′ d'un treillis de concepts. On a :*

$$|DA_s(x, y, z, t) - DA_s(x, y, z, t')| \leq |\mathbf{o}_x \cap (\mathbf{o}_t \Delta \mathbf{o}_{t'})| + |\mathbf{a}_x \cap (\mathbf{a}_t \Delta \mathbf{a}_{t'})|$$

Démonstration. Par la partition en 15 sous-ensembles de l'union de $\mathbf{o}_{y \wedge z}$, $\mathbf{o}_x$, $\mathbf{o}_t$ et $\mathbf{o}_{t'}$. □

On peut remarquer que cette propriété n'a pas d'analogue pour DA. On note aussi que ces valeurs peuvent être nulles pour $t \neq t'$.

6.3 Inégalités triangulaires

Proposition 12. *Pour six éléments quelconques d'un treillis de concepts on a :*

$$|DA_s(x, y, z, t) - DA_s(x, y, z, t')| \leq |DA_s(x, y, z, t) - DA_s(x, y, z, t'')| + |DA_s(x, y, z, t'') - DA_s(x, y, z, t')|$$

Démonstration. Sur $\mathbb{N}$, la valeur absolue de la différence est une distance. □

Proposition 13. *Pour six éléments quelconques d'un treillis de concepts on a :*

$$DA_s(x, y, z, t) \leq DA_s(x, u, v, t) + DA_s(y, u, v, z)$$

On le démontre facilement, à partir de l'inégalité triangulaire sur δ_H. On peut noter que cette propriété est vraie aussi pour DA. En revanche, les inégalités suivantes sont en général fausses pour DA_s et DA :

$$DA_s(x, y, z, t) \leq DA_s(x, y, u, v) + DA_s(u, v, z, t) \quad \text{et}$$
$$DA_s(x, y, z, t) \leq DA_s(x, u, z, v) + DA_s(u, y, v, t)$$

6.4 Exemple

On donne ici un sous-contexte du contexte Zoo [10], composé de mammifères et d'oiseaux. Il n'est pas réduit, ce qui fait que la distance utilisée est la seconde donnée en section 3.2. La figure 5 détaille le treillis et les concepts irréductibles, ainsi que la valeur ν de chaque concept.

SmallZoo		hair	feathers	eggs	milk	airborne	aquatic	predator	toothed
		a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
o_0	aardvark	×			×			×	×
o_1	chicken		×	×		×			
o_2	crow		×	×		×		×	
o_3	dolphin				×		×	×	×
o_4	duck		×	×		×	×		
o_5	fruitbat	×			×	×			×
o_6	kiwi		×	×				×	
o_7	mink	×			×		×	×	×
o_8	penguin		×	×			×	×	
o_9	platypus	×		×	×		×	×	

Le tableau ci-dessous donne les distances δ entre les concepts où les objets (les animaux, rangés comme « oiseaux » ou « mammifères ») apparaissent pour la première fois en montant dans le diagramme de Hasse du treillis de concepts. Par exemple, « aadvark » correspond au concept $C(25)$ de la figure 5 et « duck » à $C(21)$. L'utilisation d'un algorithme de classification hiérarchique standard [5] produit pour deux classes une séparation parfaite des mammifères et des oiseaux. Le même algorithme sur le tableau des distances « plus court chemin » commet des confusions.

5. `AgglomerativeClustering` du paquetage `scikit-learn` Python, avec l'indice du saut maximum comme avec celui de la moyenne. Merci à Cédric Fayet.

Ceci tend à confirmer que δ est par construction cohérente avec la sémantique des données

δ	aardvark	dolphin	fruitbat	mink	platypus	chicken	crow	duck	kiwi	penguin
aard.	0	4	5	2	6	12	9	11	10	9
dolph.	4	0	7	2	6	12	9	9	10	7
frui.	5	7	0	5	7	9	8	8	11	10
mink	2	2	5	0	4	12	9	9	10	7
plat.	6	6	7	4	0	10	7	7	8	5
chic.	12	12	9	12	10	0	3	3	6	7
crow	9	9	8	9	7	3	0	4	3	4
duck	11	9	8	9	7	3	4	0	7	4
kiwi	10	10	11	10	8	6	3	7	0	3
peng.	9	7	10	7	5	7	4	4	3	0

7 Situation bibliographique et conclusion

Cette section bibliographique ne traite pas de la définition et des propriétés de la proportion analogique dans les treillis de concepts. Un article des mêmes auteurs est en soumission, qui développera le sujet et ses sources.

Beaucoup de travaux ont été réalisés sur l'agrégation de préférences, où l'on définit une distance entre deux ordres sur le même ensemble. En revanche, il est intéressant de constater que le mariage d'une seule relation d'ordre avec une distance a été jusqu'ici peu fécond. Dans [3], Barthelemy propose une distance calculée à partir d'une application strictement croissante de l'ensemble ordonné dans $\mathbb{R}$. Montjardet [16] a écrit un article de revue sur le sujet en 1981, qui présente surtout des distances dans des semi-treillis distributifs, basées sur le *rang* qu'on peut y définir. Dans [17], le même auteur analyse certains travaux de M. Barbut qui utilisent la même idée (attribuée à [6]) tournés vers la notion de *médiane*. Dans le livre de référence [7] publié en 2008 sur les relations d'ordre, aucune allusion n'y est faite, exceptée celle citée à la section 3.4.

Il est possible que les auteurs de cet article, non experts dans l'étude avancée des relations d'ordre, soient passés à côté de travaux récents dans le domaine ; cependant, une recherche par mots-clé dans la revue de référence *Order* s'est (entre autres) avérée infructueuse.

La prolongation de ces travaux visera d'abord à légitimer cette distance comme un outil d'analyse des données ordonnées et des treillis de concepts. Des questions théoriques se posent aussi, par exemple : peut-on construire un ordre partiel à partir d'une distance sur un ensemble fini ?

Par ailleurs, les treillis de concepts sont induits à partir de contextes formels qui peuvent correspondre à n'importe quel type de relations. Différents exemples de concepts formels abstraits peuvent se rencontrer en *granular computing* [9] et en argumentation abstraite [1]. La distance introduite dans cet article pourrait aussi trouver des applications dans ces domaines.

Références

[1] L. Amgoud and H. Prade. *A formal concept view of abstract argumentation.* ECSQARU'13, Utrecht. LNCS 7958, Springer. 2013.

[2] M. Barbut et B. Monjardet, Eds. *L'ordre et la classification*. Hachette, 1970.

[3] J.-P. Barthelemy *Remarques sur les propriétés métriques des ensembles ordonnés* Mathématiques et sciences humaines, tome 61 (1978), p. 39-60

[4] M. Besson. *À propos des distances entre ensembles de parties.* Mathématiques et Sciences Humaines 42 (1973) : 17-35. <http://eudml.org/doc/94122>.

[5] G. Birkhoff. *Lattice Theory*. 3rd ed. American Mathematical Society. 1967 (1st ed. 1940).

[6] G. Birkhoff, S. Kiss (1947). *A ternary operation in distributive lattices.* Bulletin of the American Mathematical Society 53, pp. 749-752.

[7] N. Caspard, B. Leclerc et B. Monjardet. *Ensembles ordonnés finis : concepts, résultats et usages.* Springer, 2007.

[8] M. Couceiro, N. Hug, H. Prade and G. Richard. *Behavior of Analogical Inference w.r.t. Boolean Functions*, IJCAI'18, July 13-19, Stockholm, 2018.

[9] D. Dubois and H. Prade. *Bridging gaps between several forms of granular computing.* Granular Computing, vol 1, number 2, 2016.

[10] A. Frank and A. Asuncion, *UCI Machine Learning Repository*, 2010, University of California, Irvine, School of Information and Computer Sciences, http://archive.ics.uci.edu/ml.

[11] B. Ganter and R. Wille. *Formal Concept Analysis. Mathematical Foundations.* Springer, 1998.

[12] N. Hug, H. Prade, G. Richard, M. Serrurier. *Analogical Classifiers : A Theoretical Perspective*. ECAI 2016, The Hague.

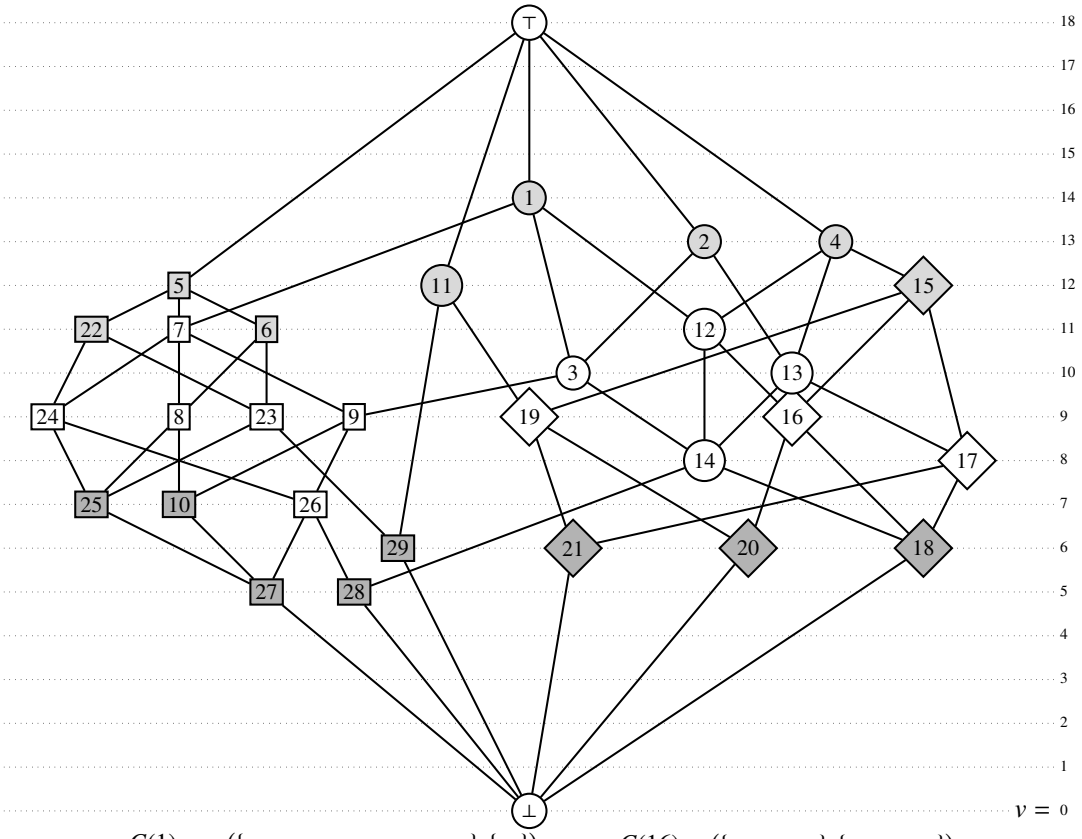
[13] L. Miclet, S. Bayoudh and A. Delhay. *Analogical dissimilarity : definition, algorithms and two experiments in machine learning*. JAIR, 32, 2008

[14] L. Miclet, H. Prade and D. Guennec, *Looking for analogical proportions in a formal concept analysis setting.* Proc. 8th IC on Concept Latices and Their Applications (CLA'11) , 2011.

[15] L. Miclet, N. Barbot and H. Prade, *From analogical proportions to proportional analogies in formal concepts.* ECAI 2014.

[16] B. Montjardet. *Metrics on partially ordered sets – a survey.* Discrete Mathematics, Volume 35, 1981.

[17] B. Monjardet. *Marc Barbut au pays des médianes.* Documents de travail du Centre d'Economie de la Sorbonne 2013.39 - ISSN : 1955-611X. 2013. halshs-00825005. Page 7



$C(1)$	$(\{o_0, o_2, o_3, o_6, o_7, o_8, o_9\}, \{a_6\})$	$C(16)$	$(\{o_2, o_6, o_8\}, \{a_1, a_2, a_6\})$
$C(2)$	$(\{o_3, o_4, o_7, o_8, o_9\}, \{a_5\})$	$C(17)$	$(\{o_4, o_8\}, \{a_1, a_2, a_5\})$
$C(3)$	$(\{o_3, o_7, o_8, o_9\}, \{a_5, a_6\})$	$C(18)$	$(\{o_8\}, \{a_1, a_2, a_5, a_6\})$
$C(4)$	$(\{o_1, o_2, o_4, o_6, o_8, o_9\}, \{a_2\})$	$C(19)$	$(\{o_1, o_2, o_4\}, \{a_1, a_2, a_4\})$
$C(5)$	$(\{o_0, o_3, o_5, o_7, o_9\}, \{a_3\})$	$C(20)$	$(\{o_2\}, \{a_1, a_2, a_4, a_6\})$
$C(6)$	$(\{o_0, o_3, o_5, o_7\}, \{a_3, a_7\})$	$C(21)$	$(\{o_4\}, \{a_1, a_2, a_4, a_5\})$
$C(7)$	$(\{o_0, o_3, o_7, o_9\}, \{a_3, a_6\})$	$C(22)$	$(\{o_0, o_5, o_7, o_9\}, \{a_0, a_3\})$
$C(8)$	$(\{o_0, o_3, o_7\}, \{a_3, a_6, a_7\})$	$C(23)$	$(\{o_0, o_5, o_7\}, \{a_0, a_3, a_7\})$
$C(9)$	$(\{o_3, o_7, o_9\}, \{a_3, a_5, a_6\})$	$C(24)$	$(\{o_0, o_7; o_9\}, \{a_0, a_3, a_6\})$
$C(10)$	$(\{o_3, o_7\}, \{a_3, a_5, a_6, a_7\})$	$C(25)$	$(\{o_0, o_7\}, \{a_0, a_3, a_6, a_7\})$
$C(11)$	$(\{o_1, o_2, o_4, o_5\}, \{a_4\})$	$C(26)$	$(\{o_7, o_9\}, \{a_0, a_3, a_5, a_6\})$
$C(12)$	$(\{o_2, o_6, o_8, o_9\}, \{a_2, a_6\})$	$C(27)$	$(\{o_7\}, \{a_0, a_3, a_5, a_6, a_7\})$
$C(13)$	$(\{o_4, o_8, o_9\}, \{a_2, a_5\})$	$C(28)$	$(\{o_9\}, \{a_0, a_2, a_3, a_5, a_6\})$
$C(14)$	$(\{o_8, o_9\}, \{a_2, a_5, a_6\})$	$C(29)$	$(\{o_5\}, \{a_0, a_3, a_4, a_7\})$
$C(15)$	$(\{o_1, o_2, o_4, o_6, o_8\}, \{a_1, a_2\})$		

FIGURE 5 – Treillis des 31 concepts du contexte SmallZoo.Les éléments en forme de rectangle (resp. losange) ont tous leurs objets dans la classe « oiseau » (resp. « mammifère »). Les éléments en forme de cercle possèdent à la fois des objets de classe 1 et de classe 2. Les éléments sup-irréductibles sont en gris foncé, les inf-irréductibles en gris clair.
À titre d'exemple, on remarque que, puisque $C(19) \wedge C(26) = C(29) \wedge C(17) = \bot$ et que $C(19) \vee C(26) = C(29) \vee C(17) = \top$, on a la proportion analogique $C(19) : C(29) \overset{WAP}{::} C(17) : C(26)$. Avec les notations de la figure 4, on a $R_x = R_{C(19)} = 2$, $R_y = R_{C(29)} = 2$, $R_z = R_{C(17)} = 2$ et $R_t = R_{C(26)} = 3$, avec $\alpha = 5$ et $\beta = 2$.

Actes JIAF 2019

Manipulation in Majoritarian Goal-based Voting

Arianna Novaro[1] **Umberto Grandi**[1]
Dominique Longin[2] **Emiliano Lorini**[2]
[1] IRIT, University of Toulouse
[2] IRIT, CNRS, Toulouse
[arianna.novaro, umberto.grandi, longin, lorini]@irit.fr

Résumé

Dans le vote par buts les agents s'expriment sur des questions binaires grâce à des formules de logique propositionnelle. Les buts individuels sont agrégés par une fonction qui calcule la décision collective comme un ensemble d'évaluations. Avoir des agents motivés par des buts individuels amène naturellement à des situations de vote stratégique, où un agent peut obtenir un meilleur résultat en déclarant un but insincère. La majorité étant une des règles les plus connues utilisées pour prendre des décisions collectives, nous étudions trois de ses variantes dans le cadre du vote par buts. Nous étudions la manipulation pour ces règles en général, ainsi que pour un ensemble limité d'actions stratégiques ou des restrictions sur le langage des buts. Nous établissons aussi la complexité computationnelle pour qu'un agent puisse trouver une manipulation.

1 Introduction

A key aspect of agent-based architectures is endowing agents with goals [25], and propositional goals in particular are common in models of strategic reasoning. When taking collective decisions in a multi-issue domain, agents share the control over the variables at stake while still holding individual goals, as in the following example :

Example 1.1. *Three automated personal assistants need to arrange a business meal for their owners. They have to decide whether the restaurant should be fancy (F), if it should be in the center (C), and if they should meet for lunch (L) instead of dinner. Each owner gives to their assistant a propositional goal with respect to these issues. The goal of the first agent is that if they go to a restaurant in the suburbs, then they should have a casual lunch : $\gamma_1 = \neg C \rightarrow (\neg F \wedge L)$. The second agent wants that the meeting is either in the suburbs or casual, but not both : $\gamma_2 = \neg C \oplus \neg F$. The third agent wants a fancy dinner in the center : $\gamma_3 = F \wedge \neg L \wedge C$. The assistants want to ensure that the final decision satisfies their owners.*

In Example 1.1 we need a procedure to make the autonomous agents reach a collective decision on every issue as precisely as possible : if the assistants returned a large set of possible options, the owners would ultimately have to make the choice they wanted to avoid. Secondly, strategic behavior needs to be taken into account, as agents are goal-oriented and nothing bounds them to be truthful. Two frameworks have been proposed in the literature on artificial intelligence to solve analogous problems : belief merging (see, e.g., Konieczny and Pino Pérez [17]) and goal-based voting [22]. Given our primary concern of resoluteness of the voting outcome we choose the latter.

A variety of functions could be studied to handle individual goals : our focus is on majoritarian rules. The appeal of majority lies not only in its intuitive definition and extensive application in real-world scenarios, but also on having been widely studied in the related fields of voting theory and judgment aggregation [20, 3]. However, when moving to goal-based voting many definitions of majority are possible. The three adaptations studied here strike a balance between different needs : that of providing a resolute result, and that of treating each issue independently while still considering the complex structure of propositional goals.

Each of these majoritarian goal-based voting rules will be analyzed with respect to their resistance to

several manipulation strategies. Negative results, i.e., finding that a rule *can* be manipulated, lead us to study the computational complexity of manipulation, as well as restricting the language of individual goals in the hope of discovering niches of strategy-proofness.

Related work. Our starting point is the work on voting in multi-issue domains with compactly represented preferences by Lang [18]. Propositional goals are such an example, linked to the literature on social choice with dichotomous preferences [7, 8]. A related preference language is that of CP-nets, in which preferences that are not necessarily dichotomous can be expressed [1]. The literature on combinatorial voting (see, e.g., the chapter by Lang and Xia [19]) provides solutions to tackle the combinatorial explosion entailed by the structure of the alternatives, such as voting sequentially over issues using tractable voting rules.

Closely related work is the study of strategy-proofness in judgment aggregation [4, 10], where the input is a complete binary choice over all issues rather than a propositional goal, as well as in belief merging [11], which focuses on a specific set of rules defined by axioms inspired from belief revision. The latter is the closest setting from a technical point of view, and in Section 2.5 we clarify the differences between the two models. Manipulation of voting rules has been amply studied in voting theory, starting from the seminal result of Gibbard and Satterthwaite [13, 24] to more recent studies aimed at finding barriers to manipulation (see, e.g., the survey by Conitzer and Walsh [2]).

Propositional goals in a strategic setting have been extensively studied in the literature on boolean games [15, 26]. Here, however, issues are not exclusively controlled by agents, since they express their goals using a common set of variables, a closer model being that of aggregation games [14].

Paper structure. In Section 2 we present the framework of majoritarian goal-based voting for different notions of resoluteness. Section 3 introduces manipulation, providing both theoretical and computational complexity results. Section 4 studies syntactic restrictions on the goal language and analyses strategy-proofness. Section 5 concludes.

2 Formal Framework

We start by presenting goal-based voting, as first introduced by Novaro *et al.* [22], focussing on three variants of issue-wise majority rule with varying degrees of resoluteness. We also include a detailed comparison with the related framework of belief merging [12].

2.1 Goal-Based Voting

A group of *agents*, represented by the finite set $\mathcal{N} = \{1, \dots, n\}$, has to take a collective decision over a number of *issues*, represented by the finite set $\mathcal{I} = \{1, \dots, m\}$ of propositional variables. We call *literal* L any atom $j \in \mathcal{I}$ or its negation $\neg j$. We let $\mathcal{L}_{\mathcal{I}}$ be the propositional language over the atoms in $\mathcal{I}$, with the usual boolean connectives. Each agent i expresses her *individual goal* by a consistent formula γ_i of $\mathcal{L}_{\mathcal{I}}$, as in Example 1.1. A *goal-profile* $\mathbf{\Gamma} = (\gamma_1, \dots, \gamma_n)$ collects the n agents' goals.

In Section 4 we will study restrictions on the language of goals, i.e., languages $\mathcal{L}^{\star}$ for $\star \in \{\wedge, \vee, \oplus\}$, defined by the following BNF grammars :

$$\varphi := p \mid \neg p \mid \varphi \star \varphi$$

An *interpretation* (or *alternative*) is a function $v : \mathcal{I} \to \{0, 1\}$ associating a binary value to each variable in $\mathcal{I}$, where 0 means the issue is rejected and 1 that it is accepted. We assume that there is no integrity constraint, allowing all possible interpretations over the issues. We write $v \models \varphi$ if interpretation v makes φ true (i.e., v is a *model* of φ). The set $\mathrm{Mod}(\varphi) = \{v \mid v \models \varphi\}$ contains all the models of φ. We denote agent i's choices for issue j in the models of her goal as $v_i(j) = (m^1_{ij}, m^0_{ij})$, with $m^x_{ij} = |\{v \in \mathrm{Mod}(\gamma_i) \mid v(j) = x\}|$ for $x \in \{0, 1\}$. Abusing notation, we write $v_i(j) = x$ if $|\mathrm{Mod}(\gamma_i)| = 1$ and $m^x_{ij} = 1$.

Numerous procedures can be used to turn individual goals into a group decision. For instance, in Example 1.1 a goal-based voting rule should provide a decision over type and location of the restaurant and the timing of the meal. A *goal-based voting rule* is formally defined as a function $F : (\mathcal{L}_{\mathcal{I}})^n \to \mathcal{P}(\{0, 1\}^m) \setminus \emptyset$ for any n and m. The input is a profile of n formulas, and the output is a non-empty set of alternatives. The number of acceptances and rejections of issue j in the outcome $F(\mathbf{\Gamma})$ is defined as $F(\mathbf{\Gamma})_j = (F(\mathbf{\Gamma})^0_j, F(\mathbf{\Gamma})^1_j)$, where $F(\mathbf{\Gamma})^x_j = |\{v \in F(\mathbf{\Gamma}) \mid v_j = x\}|$ for $x \in \{0, 1\}$. If $F(\mathbf{\Gamma})^x_j = 0$, we just write $F(\mathbf{\Gamma})_j = 1 - x$.

In this paper we study three generalizations of *majority*. Its definition in judgment aggregation [4, 9], where each agent i express a complete binary ballot B_i over all issues, is $Maj(\boldsymbol{B})_j = 1$ *iff* $\sum_{i \in \mathcal{N}} b_{ij} \geq \lceil \frac{n+1}{2} \rceil$ where $\boldsymbol{B} = (B_1, \dots, B_n)$.

2.2 Resolute Rules

We begin by presenting two definitions of *resolute* rules, always returning a unique model as their output. The first resolute variant of majority is *EMaj*, a quota rule accepting an issue if and only if more than half of

the total votes are in its favor :

$$\mathit{EMaj}(\mathbf{\Gamma})_j = 1 \;\; \mathit{iff} \;\; \sum_{i\in\mathcal{N}}\Big(\sum_{v\in \mathrm{Mod}(\gamma_i)} \frac{v(j)}{|\mathrm{Mod}(\gamma_i)|}\Big) \geq \left\lceil \frac{n+1}{2} \right\rceil$$

As the goal formulas of the agents may have a varying number of models satisfying them, to guarantee their *equal* treatment *EMaj* gives a weight to each model of an agent's goal inversely proportional to the total number of models of her goal.

The second resolute variant of majority *2sMaj* proceeds in *two steps* by first applying *Maj* to the models of the agents' goals, and then to the result of the first aggregation step. We write $\mathit{Maj}(\mathrm{Mod}(\gamma_i))$ for $\mathit{Maj}(v_1,\ldots,v_k)$ where $\mathrm{Mod}(\gamma_i)=\{v_1,\ldots,v_k\}$:

$$\mathit{2sMaj}(\mathbf{\Gamma}) = \mathit{Maj}(\mathit{Maj}(\mathrm{Mod}(\gamma_1)),\ldots,\mathit{Maj}(\mathrm{Mod}(\gamma_n)))$$

Both *EMaj* and *2sMaj* are generalizations of *Maj* : they coincide with it when agents have goals in the form of complete conjunctions of literals.

While resolute rules help agents come to a unique decision, other desiderata of unbiasedness cannot be guaranteed at the same time. Consider the following two axioms, as defined by Novaro *et al.* [22].

Let $\varphi[j \mapsto k]$ for $j,k \in \mathcal{I}$ be the replacement of each occurrence of k by j in φ. A goal-based voting rule is *dual* if for all profiles $\mathbf{\Gamma}$, $F(\overline{\gamma}_1,\ldots,\overline{\gamma}_n) = \{(1-v(1),\ldots,1-v(m)) \mid v \in F(\mathbf{\Gamma})\}$ where $\overline{\gamma} = \gamma[\neg 1 \mapsto 1,\ldots,\neg m \mapsto m]$. A rule F is *anonymous* if for any profile $\mathbf{\Gamma}$ and permutation $\sigma : \mathcal{N} \to \mathcal{N}$, we have that $F(\gamma_1,\ldots,\gamma_n) = F(\gamma_{\sigma(1)},\ldots,\gamma_{\sigma(n)})$.

Unfortunately, these three desirable properties cannot be simultaneously satisfied, as shown by the following theorem : [1]

Theorem 2.1. *There is no resolute rule F satisfying both anonymity and duality.*

Démonstration. Consider a rule F and suppose towards a contradiction that F is resolute, anonymous and dual. Take profile $\mathbf{\Gamma}$ for $\mathcal{N}=\{1,2\}$ and $\mathcal{I}=\{1,2\}$ where $\gamma_1 = 1 \wedge \neg 2$ and $\gamma_2 = \neg 1 \wedge 2$. By anonymity of F, for profile $\mathbf{\Gamma}' = (\gamma_2,\gamma_1)$ we have $F(\mathbf{\Gamma}) = F(\mathbf{\Gamma}')$. Since F is resolute, $F(\mathbf{\Gamma}) = \{(x,y)\}$, for $x,y \in \{0,1\}$, and thus $F(\mathbf{\Gamma}') = \{(x,y)\}$. However, note that $\gamma_1 = \overline{\gamma_2}$ and $\gamma_2 = \overline{\gamma_1}$. Hence, $\mathbf{\Gamma}' = \overline{\mathbf{\Gamma}}$ and by duality we must have $F(\mathbf{\Gamma}') = \{(1-x,1-y)\}$. Contradiction. □

In the next section we will thus define a weaker but more attainable notion of resoluteness, which can be satisfied by anonymous and dual majoritarian rules.

1. A related result in social choice theory states that there exists no resolute, anonymous, and neutral voting procedure for 2 alternatives and an even number of voters (see, e.g., Moulin [21]).

2.3 Weakly Resolute Rules

We call a rule *weakly resolute* if the alternatives in its outcome either accept, reject or abstain on any of the issues. Formally, a rule F is weakly resolute if on every profile $\mathbf{\Gamma}$, $F(\mathbf{\Gamma}) = \mathrm{Mod}(\varphi)$ for some conjunction $\varphi \in \mathcal{L}^{\wedge}$. We begin by showing that each goal-based voting rule that satisfies an axiom called independence by Novaro *et. al.* [22] is weakly resolute. A rule F is *independent* if there are functions $f_j : \mathcal{D}_m^n \to \mathcal{C}$ for $j \in \mathcal{I}$ such that for all $\mathbf{\Gamma}$ we have $F(\mathbf{\Gamma}) = \Pi_{j\in\mathcal{I}} f(m_1(j),\ldots,m_n(j))$, for $\mathcal{D}_m=\{(a,b) \mid a,b \in \mathbb{N} \text{ and } a+b \leq 2^m\}$ and $\mathcal{C}=\{\{0\},\{1\},\{0,1\}\}$.

Theorem 2.2. *Each independent goal-based voting rule is weakly resolute.*

Démonstration. Consider an arbitrary $\mathbf{\Gamma}$ and the outcome of an independent rule $F(\mathbf{\Gamma})$. As F is independent, we have $F(\mathbf{\Gamma}) = \Pi_{j\in\mathcal{I}} f(m_1(j),\ldots,m_n(j))$, where each $m_x(j) \in \{\{0\},\{1\},\{0,1\}\}$. We want to show that F is weakly resolute. We construct a conjunction ψ as follows : for all $j \in \mathcal{I}$, if $f(m_1(j),\ldots,m_n(j)) = \{0\}$ add conjunct $\neg j$ to ψ; if $f(m_1(j),\ldots,m_n(j)) = \{1\}$ add conjunct j to ψ; if $f(m_1(j),\ldots,m_n(j)) = \{0,1\}$ skip. For all $v \in \mathrm{Mod}(\psi)$ and for all $j \in \mathcal{I}$ appearing as conjuncts in ψ, we have $v(j) = 1$ for a positive literal j, and $v(j) = 0$ for a negative literal $\neg j$. Moreover, for all $k \in \mathcal{I}$ which did not appear in ψ we have any possible combination of truth values. Therefore, $\mathrm{Mod}(\psi) = F(\mathbf{\Gamma})$. □

The other direction of Theorem 2.2 does not hold : consider a rule F that returns $\{(1,1,\ldots,1)\}$ if in at least one $v \in \mathrm{Mod}(\gamma_1)$ issue 1 is true, and else it returns $\{(0,0,\ldots,0)\}$.

Novaro *et. al.* [22] showed that the only rule satisfying independence, as well as a number of other desirable properties, is the following :

$$\mathit{TrueMaj}(\mathbf{\Gamma}) = \Pi_{j\in\mathcal{I}} M(\mathbf{\Gamma})_j$$

where for all $j \in \mathcal{I}$:

$$M(\mathbf{\Gamma})_j = \begin{cases} \{x\} & \text{if } \sum_{i\in\mathcal{N}} \frac{m_{ij}^x}{|\mathrm{Mod}(\gamma_i)|} > \sum_{i\in\mathcal{N}} \frac{m_{ij}^{1-x}}{|\mathrm{Mod}(\gamma_i)|} \\ \{0,1\} & \text{otherwise} \end{cases}$$

TrueMaj compares issue-by-issue the total acceptances with the total rejections, setting the result to 1 (respectively, 0) if higher (respectively, lower), and to both 0 and 1 if tied.

2.4 Implementation of Majoritarian Rules

To compare the three definitions of majoritarian goal-based voting, we wrote a program to compute their outcome over all 16581375 profiles for 3 agents

and 3 issues : i.e., for any combination of consistent goals and by considering γ_1 and γ_2 equivalent if $\text{Mod}(\gamma_1) = \text{Mod}(\gamma_2)$. Results are shown in Table 1. *TrueMajR* is a resolute version of *TrueMaj* picking a single random alternative in the outcome if multiple are present (we show an average over 10 executions). First, we evaluated the maximization of *social welfare* : i.e., the percentage of profiles on which the outcome satisfies all agents' goals. For *TrueMaj* is on almost 50% of the profiles (40% for the random tie-breaking version), going down to 30% for *2sMaj* and 21% for *EMaj*. Then, we checked the percentage of profiles for which the rules return an outcome that does not satisfy *any* goal. This happens for less than 5% of profiles for all rules, and for less than 0,5% for *TrueMaj*. Finally, we analyzed the percentage of profiles on which agent 1 is satisfied with the result — if they are not satisfied they may have a strategy to manipulate. Agent 1 is satisfied on more than 61% of profiles for *EMaj*, almost 70% for *2sMaj* and almost 80% for *TrueMaj* (75 % for its resolute version). While preliminary, these results provide an overall picture on the performance of majoritarian rules, with *TrueMaj* standing out even when coupled with random tie-breaking.

2.5 Goal-based Voting and Belief Merging

Belief merging was proposed and widely studied as a framework to combine the beliefs of multiple agents [17]. While belief merging rules and axioms differ from those proposed in goal-based voting, both settings are concerned with the problem of combining formulas into sets of interpretations. In what follows we refer to the formulation of postulates by Everaere *et al.* [12].

We begin by observing that the (IC2) postulate, which states that the outcome of a rule should coincide with the conjunction of the goals if they are consistent, is incompatible with both resoluteness, in case $|\text{Mod}(\bigwedge_{i\in\mathcal{N}}\gamma_i)| > 1$, and weak resoluteness, e.g., when $\bigwedge_{i\in\mathcal{N}}\gamma_i = 1 \vee 2$. Our primary concern in resoluteness lead us to choose goal-based voting instead.

As for the other postulates, since in goal-based voting the integrity constraint is absent and $F(\mathbf{\Gamma}) \neq \emptyset$ on all $\mathbf{\Gamma}$, (IC0) and (IC1) are satisfied by default. The principle of irrelevance of syntax (IC3) is implicitly satisfied by goal-based voting rules, as propositional logic is simply used for the compact representation of goals. Postulate (IC4), defined for two agents, is not satisfied by neither *EMaj*, *TrueMaj* nor *2sMaj* : consider a profile $\mathbf{\Gamma}$ for two agents and three issues such that $\gamma_1 = \neg 1 \wedge \neg 2 \wedge \neg 3$ and $\gamma_2 = (1\wedge\neg 2\wedge\neg 3)\vee(\neg 1\wedge 2\wedge\neg 3)\vee(\neg 1\wedge\neg 2\wedge 3)$. We have that $\mathit{EMaj}(\mathbf{\Gamma}) = \mathit{TrueMaj}(\mathbf{\Gamma}) = \mathit{2sMaj}(\mathbf{\Gamma}) = \{(000)\}$: the outcome is thus only consistent with the goal of agent 1 and not with that of agent 2. Postulates (IC5) and (IC6) are known in the literature on social choice theory as *reinforcement* [27], which is satisfied by all three majoritarian rules proposed (where $\mathbf{\Gamma} \sqcup \mathbf{\Gamma}' = (\gamma_1, \ldots, \gamma_n, \gamma'_1, \ldots, \gamma'_n)$ indicates the union of profiles $\mathbf{\Gamma}$ and $\mathbf{\Gamma}'$) :

Theorem 2.3. *For any $\mathbf{\Gamma}$ and $\mathbf{\Gamma}'$, EMaj, 2sMaj and TrueMaj satisfy $F(\mathbf{\Gamma}) \cap F(\mathbf{\Gamma}') = S \neq \emptyset$ if and only if $F(\mathbf{\Gamma} \sqcup \mathbf{\Gamma}') = S$.*

Démonstration. Consider two arbitrary profiles $\mathbf{\Gamma}$ and $\mathbf{\Gamma}'$. Let $\mathit{EMaj}(\mathbf{\Gamma}) = \mathit{EMaj}(\mathbf{\Gamma}') = \{w\}$. For all $j \in \mathcal{I}$: if $w(j) = 1$, then there were more than $\lceil \frac{n+1}{2} \rceil$ votes for j in both $\mathbf{\Gamma}$ and $\mathbf{\Gamma}'$ (and consequently in $\mathbf{\Gamma} \sqcup \mathbf{\Gamma}'$) ; if $w(j) = 0$, then in $\mathbf{\Gamma}$ and $\mathbf{\Gamma}'$ either there was a tie for j or there were less than $\lceil \frac{n+1}{2} \rceil$ votes for j. Any combination of ties or $< \lceil \frac{n+1}{2} \rceil$ votes for j in $\mathbf{\Gamma}$ and $\mathbf{\Gamma}'$ still leads to $\mathit{EMaj}(\mathbf{\Gamma} \sqcup \mathbf{\Gamma}')_j = 0$. For *2sMaj* is as for *EMaj*, focusing on the second step only. Let $\mathit{TrueMaj}(\mathbf{\Gamma}) \cap \mathit{TrueMaj}(\mathbf{\Gamma}') = S$. For all $j \in \mathcal{I}$: if there are $w, w' \in S$ such that $w(j) = 1$ and $w'(j) = 0$, then $\mathbf{\Gamma}$ and $\mathbf{\Gamma}'$ had a tie in the votes for j and thus a tie will be in $\mathbf{\Gamma} \sqcup \mathbf{\Gamma}'$ (hence in the outcome). If $w(j) = 1$ for all $w \in S$ (analogously for 0), there may have been a tie in either $\mathbf{\Gamma}'$ or $\mathbf{\Gamma}$ for j, but not both, and so $\mathbf{\Gamma} \sqcup \mathbf{\Gamma}'$ will have no tie for j. □

Both (IC7) and (IC8) are not applicable to goal-based voting as there is no integrity constraint. Finally, the belief merging postulate (Maj), is not satisfied by *TrueMaj*. Consider goals $\gamma_1 = 1 \wedge 2$ and $\gamma_2 = 1 \oplus 2$: no matter how many times γ_2 is repeated in a profile, in the presence of γ_1 the result will always be $\{(11)\}$.

3 Manipulation : Theory and Complexity

Propositional goals lead to a dichotomous preference relation on alternatives : agents equally prefer any model of their goal to any counter-model. For resolute rules, the *unique* result satisfies an agent if and only if it is a model of their goal. Otherwise, different notions of satisfaction arise depending on how an agent compares two *sets* of interpretations.

Let $\mathit{sat} : \mathcal{L}_\mathcal{I} \times (\mathcal{P}(\{0,1\}^m) \setminus \emptyset) \to [0,1]$ be a function expressing the *satisfaction* of agent i towards the outcome of a rule F on profile $\mathbf{\Gamma}$. We simply write $\mathit{sat}(i, F(\mathbf{\Gamma}))$ instead of $\mathit{sat}(\gamma_i, F(\mathbf{\Gamma}))$. The *optimistic*, *pessimistic* and *expected utility maximizer* are three

	EMaj	*TrueMaj*	*TrueMajR*	*2sMaj*
% all agents sat	21,47	48,32	39,77	30,38
% no agent sat	4,91	0,47	0,76	2,18
% agent 1 sat	61,13	79,30	75,19	69,03

Table 1 : *For each rule we show the percentage of profiles where all agents/no agent/agent 1 are satisfied with the result. Satisfaction is optimistic.*

notions of satisfaction an agent may hold :

$$opt(i, F(\mathbf{\Gamma})) = \begin{cases} 1 & \text{if } F(\mathbf{\Gamma}) \cap \text{Mod}(\gamma_i) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

$$pess(i, F(\mathbf{\Gamma})) = \begin{cases} 1 & \text{if } F(\mathbf{\Gamma}) \subseteq \text{Mod}(\gamma_i) \\ 0 & \text{otherwise} \end{cases}$$

$$eum(i, F(\mathbf{\Gamma})) = \frac{|\text{Mod}(\gamma_i) \cap F(\mathbf{\Gamma})|}{|F(\mathbf{\Gamma})|}$$

Optimists are satisfied if in the outcome there is at least one model of their goal. Pessimists want all the interpretations in the outcome to be models of their goal (this notion was introduced by Jimeno *et al.* [16]). Expected utility maximizers assume that a unique interpretation will be chosen at random among those tied in the outcome, and the higher the proportion of models of their goal in $F(\mathbf{\Gamma})$ over the total number of interpretations in $F(\mathbf{\Gamma})$, the better. [2]

Agent i's preference on outcomes is a complete and transitive relation $\succcurlyeq_i$, whose strict part is $\succ_i$:

$$F(\mathbf{\Gamma}) \succcurlyeq_i F(\mathbf{\Gamma}') \;\; \textit{iff} \;\; sat(i, F(\mathbf{\Gamma})) \geq sat(i, F(\mathbf{\Gamma}')).$$

For $\mathbf{\Gamma} = (\gamma_i)_{i \in \mathcal{N}}$, let $(\mathbf{\Gamma}_{-i}, \gamma_i') = (\gamma_1, \ldots, \gamma_i', \ldots, \gamma_n)$ be the profile where only agent i changed her goal from γ_i to γ_i'. Agent i has an *incentive* to manipulate by submitting goal γ_i' instead of γ_i if and only if $F(\mathbf{\Gamma}_{-i}, \gamma_i') \succ_i F(\mathbf{\Gamma})$. A rule F is *strategy-proof* if and only if for all profiles $\mathbf{\Gamma}$ there is no agent i who has an incentive to manipulate.

We focus on three kind of manipulation strategies, following previous work by Everaere *et al.* [11] :

— *Unrestricted* : i can send any γ_i' instead of γ_i

— *Erosion* : i can send γ_i' s.t. $\text{Mod}(\gamma_i') \subseteq \text{Mod}(\gamma_i)$

— *Dilatation* : i can send γ_i' s.t. $\text{Mod}(\gamma_i) \subseteq \text{Mod}(\gamma_i')$

3.1 Manipulability of Majority Rules

In judgment aggregation, the issue-by-issue majority rule has been proven to be single-agent strategy-proof by Dietrich and List [4]. Surprisingly, when moving to propositional goals strategy-proofness is not guaranteed anymore for the three adaptations of the majority rule, as shown by the following result :

Theorem 3.1. *EMaj, TrueMaj and 2sMaj can be manipulated by both erosion and dilatation.*

Démonstration. We provide goal-profiles where an agent can get a better result by submitting an untruthful goal. For ease of presentation we display the models of the agents' goals, but the input of a rule F consists of propositional formulas. Consider the profiles $\mathbf{\Gamma}$, $\mathbf{\Gamma}'$ and $\mathbf{\Gamma}''$ for three agents and three issues, together with the results of *EMaj*, *TrueMaj* and *2sMaj* :

	$\mathbf{\Gamma}$	$\mathbf{\Gamma}'$	$\mathbf{\Gamma}''$	$\mathbf{\Gamma}^\star$	$\mathbf{\Gamma}^{\star\star}$
$\text{Mod}(\gamma_1)$	(111)	(111)	(111)	(111)	(111)
$\text{Mod}(\gamma_2)$	(001)	(001)	(001)	(111) (011) (100)	(111) (011) (100)
$\text{Mod}(\gamma_3)$	(101) (010) (000)	(101)	(101) (010) (000) (100) (110)	(011) (010) (101)	(011) (010) (101) (001)
EMaj	(001)	(101)	(001)	(111)	(011)
TrueMaj	(001)	(101)	(101)	-	-
2sMaj	(001)	(101)	(101)	-	-

Let $\mathbf{\Gamma}$ be the profile where agents submit their truthful goal : $\gamma_1 = 1 \wedge 2 \wedge 3$, $\gamma_2 = \neg 1 \wedge \neg 2 \wedge 3$, $\gamma_3 = (\neg 1 \wedge \neg 3) \vee (1 \wedge \neg 2 \wedge 3)$. For erosion manipulation, agent 3 prefers the result of *EMaj*, *2sMaj* and *TrueMaj* (which they happen to coincide) when applied to $\mathbf{\Gamma}'$ rather than when applied to $\mathbf{\Gamma}$. For dilatation manipulation, agent 3 prefers the result of *TrueMaj* and *2sMaj* when applied to $\mathbf{\Gamma}''$ rather than to $\mathbf{\Gamma}$. For *EMaj* and dilatation manipulation, agent 3 can get a better result by manipulating $\mathbf{\Gamma}^\star$ and moving to $\mathbf{\Gamma}^{\star\star}$. □

Theorem 3.1 is thus in sharp contrast with the result of judgment aggregation. Since the profiles used in the proof give singleton outcomes, the theorem holds for expected utility maximizers, optimists and pessimists.

2. Expected utility maximizers, optimists and pessimists, correspond to the *probabilistic*, *weak drastic* and *strong drastic* satisfaction indexes in the work of Evaraere *et al.* [11].

3.2 Computational Complexity

Majoritarian goal-based voting rules are manipulable, as shown by Theorem 3.1, but how difficult it is for an agent to find another goal allowing her to get a better outcome ? If we restrict to resolute rules, the problem definition is analogous to existing work in judgment aggregation [10] :

MANIP(F)
Input : Profile $\mathbf{\Gamma} = (\gamma_1, \ldots, \gamma_n)$, agent i
Question : If $\text{Mod}(\gamma_i) \cap F(\mathbf{\Gamma}) = \emptyset$, is there γ_i' such that $\text{Mod}(\gamma_i) \cap F(\gamma_1, \ldots, \gamma_i', \ldots, \gamma_n) \neq \emptyset$?

Let Probabilistic Polynomial Time (PP) be the class of problems that can be solved in nondeterministic polynomial time with acceptance condition that more than half of the computations accept [23]. In what follows we refer to the PP-complete problem MAJ-SAT-p, asking whether $|\text{Mod}(\varphi \wedge p)| > |\text{Mod}(\varphi \wedge \neg p)|$ for propositional formula φ and one of its variables p [22].

Theorem 3.2. MANIP*(2sMaj) is* PP*-hard.*

Démonstration. We reduce from MAJ-SAT-p. Take an instance of MAJ-SAT-p with a formula $\varphi[p_1, \ldots, p_k]$ and p_1 one of its variables. Construct an instance of MANIP(*2sMaj*) [3] where $\mathcal{I} = \{p_1, \ldots, p_k, q, r\}$, and a profile $\mathbf{\Gamma}$ with $\gamma_1 = (p_2 \wedge \cdots \wedge p_k) \wedge p_1 \wedge q \wedge r$, and $\gamma_2 = \varphi \wedge \neg q \wedge \neg r$, and $\gamma_3 = (p_2 \wedge \cdots \wedge p_k) \wedge (p_1 \oplus q) \wedge (r \rightarrow q)$.

We show that $|\text{Mod}(\varphi \wedge p_1)| > |\text{Mod}(\varphi \wedge \neg p_1)|$ if and only if agent 3 can manipulate *2sMaj* on $\mathbf{\Gamma}$. The following table represents some features of $\mathbf{\Gamma}$, where question marks represent the possibly many models of φ over $p_1, \ldots, p_k$:

	$p_2 \ldots p_k$	p_1	q	r
$\text{Mod}(\gamma_1)$	$1 \ldots 1$	1	1	1
$\text{Mod}(\gamma_2)$	?	?	0	0
	$\vdots$	$\vdots$	0	0
	?	?	0	0
$\text{Mod}(\gamma_3)$	$1 \ldots 1$	0	1	1
	$1 \ldots 1$	0	1	0
	$1 \ldots 1$	1	0	0

The result on $p_2, \ldots, p_k$ is decided by agents 1 and 3 : all issues will be accepted regardless of the vote of agent 2.

Let us now focus on p_1, q and r. Applying (strict) majority to the models of γ_3 leads to the first-step result (010). Agent 3 is pivotal on issues q and r (since agents 1 and 2 will give one vote for and one vote against them after the first step). There are now two cases to consider :

3. For ease of presentation and to avoid confusion we write the issues as $\mathcal{I} = \{p, q, r, \ldots\}$ instead of $\mathcal{I} = \{1, 2, 3, \ldots\}$.

a) If $|\text{Mod}(\varphi \wedge p_1)| > |\text{Mod}(\varphi \wedge \neg p_1)|$, the result of *2sMaj*($\mathbf{\Gamma}$) is $(1 \ldots 1, 110)$, that is not a model of γ_3. However, by submitting $\gamma_3' = (p_2 \wedge p_k) \wedge p_1 \wedge \neg q \wedge \neg r$, we have *2sMaj*($\mathbf{\Gamma}'$) $= (1 \ldots 1, 100)$ which is a model of γ_3. Hence, agent 3 has an incentive to manipulate.

b) If $|\text{Mod}(\varphi \wedge p_1)| \leq |\text{Mod}(\varphi \wedge \neg p_1)|$, we have that *2sMaj*($\mathbf{\Gamma}$) $= (1 \ldots 1, 010)$, which is a model of γ_3. Agent 3 has thus no incentive to manipulate.

This completes the reduction from MAJ-SAT-p, showing that MANIP(*2sMaj*) is PP-hard. □

A similar reduction allows us to show the following :

Theorem 3.3. MANIP*(EMaj) is* PP*-hard.*

Démonstration. We construct an instance of MANIP(*EMaj*) from a given instance (φ, p_1) of MAJ-SAT-p. Let $\mathcal{I} = \{p_1, \ldots, p_k, q, r\}$, and profile $\mathbf{\Gamma} = (\gamma_1, \gamma_2, \gamma_3)$ with $\gamma_1 = (p_2 \wedge \cdots \wedge p_k) \wedge p_1 \wedge q \wedge r$, $\gamma_2 = \varphi(\overline{p_1}) \wedge \neg q \wedge r$ and $\gamma_3 = (p_2 \wedge \cdots \wedge p_k) \wedge (p_1 \oplus q)$, where $\varphi(\overline{p_1})$ is a formula where each occurrence of p_1 in φ has been replaced by $\neg p_1$ and vice-versa. The final part of the proof, showing that $|\text{Mod}(\varphi \wedge p_1)| > |\text{Mod}(\varphi \wedge \neg p_1)|$ if and only if agent 3 can manipulate *EMaj* on $\mathbf{\Gamma}$, is analogous to that of Theorem 3.2 and thus omitted. □

We leave the extension of the above results to *TrueMaj* for each attitude of the agent (optimist, utility maximizer, or pessimist) for future work, conjecturing it to be PP-hard as well.

4 Language Restrictions

We study three restrictions on the goal-language : conjunctions, disjunctions and exclusive disjunctions, as defined by the corresponding languages $\mathcal{L}^\star$ defined in Section 2.1. Results are presented in Table 2.

4.1 Conjunctions

The formulas of the *language of conjunctions* $\mathcal{L}^\wedge$ are conjunctions of literals over issues in $\mathcal{I}$. $\mathcal{L}^\wedge$ captures the framework of judgment aggregation with abstentions [5, 6], as agents have definite opinions over the issues appearing as literals in their goal and they do not care about the other issues. We find positive results of strategy-proofness :

Theorem 4.1. *An agent i with $\gamma_i \in \mathcal{L}^\wedge$ has no incentive to manipulate unrestrictedly 2sMaj and EMaj.*

Démonstration. Take $\mathbf{\Gamma}$ with $\gamma_i \in \mathcal{L}^\wedge$. Since *2sMaj* and *EMaj* are resolute, we have a unique outcome $\{w\}$ on $\mathbf{\Gamma}$. Suppose that $w \notin \text{Mod}(\gamma_i)$. As $\gamma_i = L_1 \wedge \cdots \wedge L_k$ for $k \leq m$, we have for all $j \in \mathcal{I}$:

— $v_i(j) = (\frac{|\text{Mod}(\gamma_i)|}{2}, \frac{|\text{Mod}(\gamma_i)|}{2})$ if γ_i has no L_j.
— If $L_j = j$ appears in γ_i, then $v_i(j) = (|\text{Mod}(\gamma_i)|, 0)$, and if $L_j = \neg j$, then $v_i(j) = (0, |\text{Mod}(\gamma_i)|)$.

Therefore, if $w \notin \text{Mod}(\gamma_i)$ there must be ℓ literals with $\ell \leq k$ such that $w \models \neg L_1 \wedge \cdots \wedge \neg L_\ell$. Take an arbitrary such L_x.

2sMaj. Let $\mathit{Maj}(\gamma_i) = \{w_i\}$ be the result of the first step of majority applied to γ_i. We have that $w_i(x) = 1 - w(x)$, and therefore agent i cannot influence the outcome towards $w_i(x)$, and more generally towards her goal.

EMaj. If $w(x) = 1$ (similarly for 0), then $L_x = \neg x$. Since $v_i(x) = (0, |\text{Mod}(\gamma_i)|)$, we have $\sum_{v \in \text{Mod}(\gamma_i)} \frac{v(x)}{|\text{Mod}(\gamma_i)|} = 0$ and thus $\sum_{k \in \mathcal{N} \setminus \{i\}} \sum_{v \in \text{Mod}(\gamma_k)} \frac{v_k(x)}{|\text{Mod}(\gamma_k)|} \geq \lceil \frac{n+1}{2} \rceil$. Agent i is already giving no support to x and yet x is accepted in the outcome. Therefore, *EMaj* cannot be manipulated. □

The result for *TrueMaj* has a similar proof, which is omitted for space constraints, also considering optimists, pessimists and expected utility maximizers.

Theorem 4.2. *If agent i has $\gamma_i \in \mathcal{L}^\wedge$ she has no incentive to manipulate unrestrictedly the rule TrueMaj.*

A consequence of Theorem 4.1 and 4.2 is that goals in $\mathcal{L}^\wedge$ make the three majorities strategy-proof :

Corollary 4.1. *For any $\mathcal{I}$ and $\mathcal{N}$, if $\gamma_i \in \mathcal{L}^\wedge$ for all $i \in \mathcal{N}$ then EMaj, TrueMaj and 2sMaj are strategy-proof for unrestricted manipulation.*

4.2 Disjunctions

In the *language of clauses* $\mathcal{L}^\vee$, formulas are disjunctions of literals. Unfortunately, this restriction does not guarantee strategy-proofness for two of our rules :

Theorem 4.3. *There exists $\mathbf{\Gamma}$ and $i \in \mathcal{N}$ with $\gamma_i \in \mathcal{L}^\vee$ such that agent i has an incentive to manipulate EMaj and TrueMaj by erosion.*

Démonstration. Profiles are for three agents and two issues. For *EMaj*, consider $\mathbf{\Gamma}$ with $\gamma_1 = 1$, $\gamma_2 = \neg 1$ and $\gamma_3 = 1 \vee 2$. By submitting $\gamma_3' = 1 \wedge 2$ agent 3 can change $\mathit{EMaj}(\mathbf{\Gamma}) = \{(00)\}$ into $\{(11)\}$: hence, they have an incentive to lie. For *TrueMaj*, consider $\mathbf{\Gamma}$ with $\gamma_1 = \neg 1 \wedge \neg 2$ and $\gamma_2 = \gamma_3 = 1 \vee 2$. The result is $\mathit{TrueMaj}(\mathbf{\Gamma}) = \{(00)\}$, but if agent 3 submits $\gamma_3' = 1 \wedge 2$ we get $\mathit{TrueMaj}(\mathbf{\Gamma}') = \{(11)\}$ and thus agent 3 has an incentive to manipulate. □

We can obtain positive results if we restrict the set of available manipulation strategies to dilatation :

Theorem 4.4. *For any $\mathbf{\Gamma}$ with $\gamma_i \in \mathcal{L}^\vee$ for $i \in \mathcal{N}$, agent i has no incentive to manipulate EMaj and TrueMaj by dilatation.*

Proof sketch. Take $\mathbf{\Gamma}$ with $\gamma_i \in \mathcal{L}^\vee$ for some $i \in \mathcal{N}$, i.e., $\gamma_i = L_1 \vee \cdots \vee L_k$. Consider $\mathit{EMaj}(\mathbf{\Gamma}) = \{w\}$, such that $w \notin \text{Mod}(\gamma_i)$. As the agent is restricted to dilatation strategies, the goals γ_i' such that $\text{Mod}(\gamma_i) \subseteq \text{Mod}(\gamma_i')$ agent i can use are those whose models would lower i's support for each literal L_ℓ in γ_i. Thus she cannot manipulate.

Let now $\mathit{TrueMaj}(\mathbf{\Gamma}) = \{w_1, \ldots, w_k\}$. Reasoning as above, an optimist agent i does not have any dilatation strategy to include one of the models of γ_i into the outcome. An expected utility maximizer i however, may want to remove some $w_k \in \mathit{TrueMaj}(\mathbf{\Gamma})$ such that $w_k \models \neg L_j$ for all L_j in γ_i. But this is only possible when $\{w_1, \ldots, w_\ell\} \cap \text{Mod}(\gamma_i) = \emptyset$, since the agent is restricted to dilatation strategies, and other reported goal would have more models increasing the votes against the literals present in her sincere goal γ_i. Therefore, this would not constitute a profitable deviation. A similar reasoning applies to pessimists. □

Theorem 4.5. *For any profile where $\gamma_i \in \mathcal{L}^\vee$ for $i \in \mathcal{N}$, agent i has no incentive to manipulate unrestrictedly 2sMaj.*

Démonstration. Since $\gamma_i \in \mathcal{L}^\vee$, the result w_i of $\mathit{Maj}(\text{Mod}(\gamma_i))$ is such that $w_i(x) = 1$ if $L_x = x$ appears in γ_i, and $w_i(x) = 0$ if $L_x = \neg x$. Hence, it coincides with the result of $\mathit{Maj}(\text{Mod}(\gamma_i'))$ for $\gamma_i' \in \mathcal{L}^\wedge$ where every occurrence of $\vee$ in γ_i has been replaced by $\wedge$ in γ_i'. The proof of Theorem 4.1 can thus be applied, since agent i is already maximizing her chances of getting γ_i satisfied by submitting γ_i. □

By combining the results of Theorem 4.4 and Theorem 4.5 we get the following corollary :

Corollary 4.2. *If $\gamma_i \in \mathcal{L}^\vee$ for all $i \in \mathcal{N}$ then EMaj and TrueMaj are strategy-proof for dilatation manipulation and 2sMaj is strategy-proof for unrestricted manipulation.*

4.3 Exclusive Disjunctions

In the *language of exclusive disjunctions* $\mathcal{L}^\oplus$, each formula is an exclusive disjunction of literals (cf. agent 2's goal in Example 1.1). We prove the following :

Theorem 4.6. *There exists profile $\mathbf{\Gamma}^0, \mathbf{\Gamma}^1, \mathbf{\Gamma}^2, \mathbf{\Gamma}^3$, and $i \in \mathcal{N}$ with $\gamma_i \in \mathcal{L}^\oplus$ such that agent i has an incentive to manipulate rules 2sMaj, EMaj and TrueMaj, by erosion and dilatation.*

	$\mathcal{L}^{\wedge}$		$\mathcal{L}^{\vee}$		$\mathcal{L}^{\oplus}$	
	E	D	E	D	E	D
EMaj	SP	SP	M	SP	M	M
TrueMaj	SP	SP	M	SP	M	M
2sMaj	SP	SP	SP	SP	M	M

Table 2 : *E stands for erosion, D for dilatation, SP for strategy-proof and M for manipulable.*

Démonstration. All profiles are for three agents and two issues.

2sMaj. For erosion, consider profile $\mathbf{\Gamma}^0$ where $\gamma_1 = 1 \wedge 2$, $\gamma_2 = \neg 1 \wedge \neg 2$ and $\gamma_3 = 1 \oplus 2$. We have that $\mathit{2sMaj}(\mathbf{\Gamma}) = \{(00)\}$. Consider now $\gamma_3' = \neg 1 \wedge 2$. The result is $\{(01)\}$, and thus agent 3 has an incentive to manipulate. For dilatation, consider the profile $\mathbf{\Gamma}^1$ where $\gamma_1 = \neg 1 \wedge 2$, $\gamma_2 = \neg 1 \wedge \neg 2$ and $\gamma_3 = 1 \oplus 2$. We have that $\mathit{2sMaj}(\mathbf{\Gamma}^1) = \{(00)\}$. If we consider $\gamma_3^\star = 1 \vee 2$, the result is $\mathit{2sMaj}(\mathbf{\Gamma}^{1\star}) = \{(01)\}$, and thus agent 3 has again an incentive to manipulate.

EMaj. For erosion, the results of *2sMaj* and *EMaj* coincide on $\mathbf{\Gamma}^0$ and $\mathbf{\Gamma}'$. For dilatation, take $\mathbf{\Gamma}^2$ with $\gamma_1 = \neg 1 \wedge 2$, $\gamma_2 = \neg 1 \vee \neg 2$ and $\gamma_3 = 1 \oplus 2$. We have $\mathit{EMaj}(\mathbf{\Gamma}^2) = \{(00)\}$. Agent 3 can submit $\gamma_3^\star = 1 \vee 2$ to obtain $\{(01)\}$.

TrueMaj. Take $\mathbf{\Gamma}^3$ with $\gamma_1 = 1 \wedge 2$, $\gamma_2 = 1 \wedge \neg 2$ and $\gamma_3 = 1 \oplus 2$. We have $\mathit{TrueMaj}(\mathbf{\Gamma}^3) = \{(10), (11)\}$. Agent 3 can manipulate by erosion with $\gamma_3^\star = 1 \wedge \neg 2$, and by dilatation with $\gamma_3^{\star\star} = \neg 1 \vee \neg 2$. In both cases the result is $\{(10)\}$. □

5 Conclusions

In this paper we studied the strategic component of the framework of goal-based voting [22], related (yet different) to both judgment aggregation and belief merging. Our focus was on three rules that have been proposed as adaptations of the issue-wise majority rule in this setting, with varying degrees of resoluteness. We find that all the majoritarian rules are not immune from manipulation, even when the manipulator can only apply limited strategies on their truthful goal (i.e., erosion and dilatation). We also find that, although not strategy-proof in general, *EMaj* and *2sMaj* are PP-hard for an agent to manipulate, as hard as their winner determination problem. Moreover, restricting the language of an agent's goal to conjunctions makes manipulation impossible, as well as dilatation manipulation for the language of disjunctions, suggesting promising directions for further research on minimal restrictions to the goal language to guarantee strategy-proofness of majoritarian rules.

Références

[1] Boutilier, Craig, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos et David Poole: *CP-nets : A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements.* Journal of Artificial Intelligence Research, 21 :135–191, 2004.

[2] Conitzer, Vincent et Toby Walsh: *Barriers to Manipulation in Voting.* Dans Brandt, F., V. Conitzer, U. Endriss, J. Lang et A. D. Procaccia (rédacteurs) : *Handbook of Computational Social Choice*, chapitre 6. Cambridge University Press, 2016.

[3] Dietrich, Franz et Christian List: *Judgment aggregation by quota rules : Majority voting generalized.* Journal of Theoretical Politics, 19(4) :391–424, 2007.

[4] Dietrich, Franz et Christian List: *Strategy-proof judgment aggregation.* Economics & Philosophy, 23(3) :269–300, 2007.

[5] Dietrich, Franz et Christian List: *Judgment Aggregation Without Full Rationality.* Social Choice and Welfare, 31(1) :15–39, 2008.

[6] Dokow, Elad et Ron Holzman: *Aggregation of Binary Evaluations with Abstentions.* Journal of Economic Theory, 145(2) :544–561, 2010.

[7] Elkind, Edith et Martin Lackner: *Structure in Dichotomous Preferences.* Dans *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[8] Elkind, Edith, Martin Lackner et Dominik Peters: *Structured Preferences.* Dans Endriss, Ulle (rédacteur) : *Trends in Computational Social Choice*, chapitre 10, pages 187–207. AI Access, 2017.

[9] Endriss, Ulle: *Judgment Aggregation.* Dans Brandt, F., V. Conitzer, U. Endriss, J. Lang et A. D. Procaccia (rédacteurs) : *Handbook of Computational Social Choice*, chapitre 17. Cambridge University Press, 2016.

[10] Endriss, Ulle, Umberto Grandi et Daniele Porello: *Complexity of Judgment Aggregation.* J. of Artificial Intelligence Research, 45 :481–514, 2012.

[11] Everaere, Patricia, Sébastien Konieczny et Pierre Marquis: *The Strategy-Proofness Landscape of Merging.* Journal of Artificial Intelligence Research, 28 :49–105, 2007.

[12] Everaere, Patricia, Sébastien Konieczny et Pierre Marquis: *An Introduction to Belief Merging and its Links with Judgment Aggregation.* Dans Endriss, Ulle (rédacteur) : *Trends in Computational Social Choice*, chapitre 7, pages 123–143. AI Access, 2017.

[13] Gibbard, Allan: *Manipulation of voting schemes : a general result.* Econometrica, pages 587–601, 1973.

[14] Grandi, Umberto, Davide Grossi et Paolo Turrini: *Equilibrium Refinement through Negotiation in Binary Voting.* Dans *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 540–546, 2015.

[15] Harrenstein, Paul, Wiebe van der Hoek, John-Jules Ch. Meyer et Cees Witteveen: *Boolean games.* Dans *Proceeding of the 8th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2001.

[16] Jimeno, José L., Joaquín Pérez et Estefanía García: *An extension of the Moulin No Show Paradox for voting correspondences.* Social Choice and Welfare, 33(3) :343–359, 2009.

[17] Konieczny, Sébastien et Ramón Pino Pérez: *Merging Information Under Constraints : A Logical Framework.* Journal of Logic and Computation, 12(5) :773–808, 2002.

[18] Lang, Jérôme: *Logical Preference Representation and Combinatorial Vote.* Annals of Mathematics and Artificial Intelligence, 42(1-3) :37—71, 2004.

[19] Lang, Jérôme et Lirong Xia: *Voting in Combinatorial Domains.* Dans Brandt, F., V. Conitzer, U. Endriss, J. Lang et A. D. Procaccia (rédacteurs) : *Handbook of Computational Social Choice*, chapitre 9. Cambridge University Press, 2016.

[20] May, Kenneth O: *A set of independent necessary and sufficient conditions for simple majority decision.* Econometrica : Journal of the Econometric Society, pages 680–684, 1952.

[21] Moulin, H.: *The strategy of social choice.* North-Holland, 1983.

[22] Novaro, Arianna, Umberto Grandi, Dominique Longin et Emiliano Lorini: *Goal-based Voting with Propositional Goals.* Dans *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[23] Papadimitriou, Christos H.: *Computational Complexity.* John Wiley and Sons, 2003.

[24] Satterthwaite, Mark Allen: *Strategy-proofness and Arrow's conditions : Existence and correspondence theorems for voting procedures and social welfare functions.* Journal of Economic Theory, 10(2) :187–217, 1975.

[25] Wooldridge, Michael: *An introduction to multiagent systems.* John Wiley and Sons, 2009.

[26] Wooldridge, Michael, Ulle Endriss, Sarit Kraus et Jérôme Lang: *Incentive Engineering in Boolean Games.* Artificial Intelligence, 195 :418–439, 2013.

[27] Young, H.P: *An axiomatization of Borda's rule.* Journal of Economic Theory, 9(1) :43 – 52, 1974.

Actes JIAF 2019

Minimizing and balancing envy among agents using Ordered Weighted Average

Parham Shams[1] **Aurélie Beynier**[1] **Sylvain Bouveret**[2] **Nicolas Maudet**[1]

[1] LIP6, Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
[2] LIG, Univ. Grenoble Alpes, CNRS, Grenoble INP, Grenoble, France
{aurelie.beynier,nicolas.maudet,shamsp}@lip6.fr sylvain.bouveret@imag.fr

Abstract

In the problem of fair resource allocation, envy freeness is one of the most interesting fairness criterion as it ensures that no agent prefers the bundle of another agent. However, when considering indivisible goods, an envy-free allocation may not exist. In this paper, we investigate a new relaxation of envy freeness consisting in minimizing the Ordered Weighted Average (OWA) of the envy vector. The idea is to choose the allocation that is fair in the sense of the distribution of the envy among agents. The OWA aggregator is a well-known tool to express fairness in multiagent optimization. In this paper, we focus on fair OWA operators where the weights of the OWA are decreasing. When an envy-free allocation exists, minimizing OWA will return this allocation. However, when no envy-free allocation exists, one may wonder how fair min OWA allocations are.

After some definitions and description of the model, we show how to formulate the computation of such a min OWA allocation as a Mixed Integer Program. Then, we investigate the link between the min OWA allocation and other well-known fairness measures such as max min share and envy freeness up to one good or to any good.

1 Introduction

In this paper, we investigate fair division of indivisible goods. In this context, several approaches have been proposed to model fairness. Amongst these models, one prominent solution concept is to look for *envy-free* allocations [9]. These allocations are such that no agent would like to swap her own bundle with the bundle of any other agent.

Envy-freeness is a very attractive criterion : the fact that each agent is better off with her own share than with any other share is a guarantee of social stability. Besides, since this criterion is only based on personal comparisons, it does not require any interpersonal comparability. Unfortunately, envy-freeness is a very demanding notion, and it is a well-known fact that in many situations, no such allocation exists (consider for instance the simple situation where the number of items to allocate is strictly less than the number of agents at stake). Hence several relaxations of the envy-freeness notion have been studied in recent years. Two orthogonal approaches have been considered. A first possibility is to "forget" some items when comparing the agents' shares. This leads to the definition of envy-freeness up to one good [10] and envy-freeness up to any good [6]. Recently, Amanatidis *et al.* [1] explored how different relaxations of envy-freeness relate to each other. Another possible approach is to relax the Boolean notion of envy and to introduce a quantity of envy that we seek to minimize. This is the path followed by Lipton *et al.* [10] or Endriss *et al.* [7] for instance. Several approximation algorithms dedicated to minimize these measures were subsequently designed, see e.g. [11].

In this paper, we elaborate on the idea of minimizing the degree of envy. More precisely, we explore the idea of finding allocations where envy is "fairly balanced" amongst agents. For that purpose, we start from the notion of individual degree of envy and use a *fair* Ordered Weighted Average operator [1] to aggregate these individual envies into a collective one, that we try to minimize. After giving some preliminary definitions in Section 2, we formally introduce our fairness MinOWA envy criterion (Section 3) and we show of OWA minimization problem can be formulated as a linear program. We then relate this criterion to other fairness notions and study properties of the allocations obtained by minimizing the OWA of the envy vector (Section 4). Finally, we present some experimental results investigating the fairness of min OWA solutions (Section 5).

1. By "fair", we mean an OWA where weights are non-increasing.

2 Model and Definitions

We will consider a classic multiagent resource allocation setting, where a finite set of *objects* $O = \{o_1, \ldots, o_m\}$ has to be allocated to a finite set of *agents* $\mathcal{N} = \{1, \ldots, n\}$. In this setting, an *allocation* is a vector $\overrightarrow{\pi} = \langle \pi_1, \ldots, \pi_n \rangle$ of *bundles* of objects, such that $\forall i, \forall j$ with $i \neq j : \pi_i \cap \pi_j = \emptyset$ (exclusion : a given object cannot be allocated to more than one agent) and $\bigcup_{i \in \mathcal{N}} \pi_i = O$ (no free-disposal : all the objects are allocated). $\pi_i \subseteq O$ is called agent i's *share*.

Any satisfactory allocation must take into account the agents' preferences on the objects. Here, we will make the assumption that these preferences are *numerically additive*. Each agent i has a *utility function* $u_i : 2^O \rightarrow \mathbb{R}^+$ measuring her satisfaction $u_i(\pi)$ when she obtains share π, which is defined as follows :

$$u_i(\pi) \stackrel{def}{=} \sum_{o_k \in \pi} w(i, o_k),$$

where $w(i, o_k)$ is the weight given by agent i to object o_k. This assumption, as restrictive as it may seem, is made by a lot of authors [10, 2, for instance] and is considered a good compromise between expressivity and conciseness.

Definition 1 *An instance of the* additive multiagent resource allocation problem *(add-MARA instance for short) $I = \langle \mathcal{N}, O, w \rangle$ is a tuple with $\mathcal{N}$ and O as defined above and $w : \mathcal{N} \times O \rightarrow \mathbb{R}^+$ is a mapping with $w(i, o_k)$ being the weight given by agent i to object o_k. We will denote by $\mathcal{P}(I)$ the set of allocations for I.*

2.1 Envy-free allocations

A prominent fairness notion in multiagent resource allocation is *envy-freeness*. Envy-freeness (EF) can be defined as follows :

Definition 2 *Let $I = \langle \mathcal{N}, O, w \rangle$ be an add-MARA instance and $\overrightarrow{\pi}$ be an allocation of I. $\overrightarrow{\pi}$ is* envy-free *if and only if $\forall i, j \in \mathcal{N}$, $u_i(\pi_i) \geq u_i(\pi_j)$.*

In other words, every agent i weakly prefers her own share to the share of any other agent j.

In the context of fair division of indivisible goods, this notion is very demanding and there exists a lot of add-MARA instances for which no envy-free allocation exists. To relax envy-freeness, a possibility is to introduce a notion of degree of envy based on pairwise envy [10].

Definition 3 *Let $I = \langle \mathcal{N}, O, w \rangle$ be an add-MARA instance and $\overrightarrow{\pi}$ be an allocation of I. The* pairwise envy *between agents i and j is defined as follows :*

$$pe(i, j, \overrightarrow{\pi}) \stackrel{def}{=} \max\{0, u_i(\pi_j) - u_i(\pi_i)\}.$$

In other words, the pairwise envy between i and j is 0 if i does not envy j, and is equal to the difference between the utility for agent i if she had agent j's bundle and her actual utility in the allocation $\overrightarrow{\pi}$. It can be interpreted as how much agent i envies agent j's bundle.

From that notion of pairwise envy, we can derive a notion of total envy of an agent, that we define as the maximal pairwise envy that this agent experiences for another agent :

Definition 4 *Let $I = \langle \mathcal{N}, O, w \rangle$ be an add-MARA instance and $\overrightarrow{\pi}$ be an allocation of I. The* envy *of agent i is :*

$$e(i, \overrightarrow{\pi}) \stackrel{def}{=} \max_{j \in \mathcal{N}} pe(i, j, \overrightarrow{\pi}).$$

The vector $\overrightarrow{e}(\overrightarrow{\pi}) = \langle e(1, \overrightarrow{\pi}), ..., e(n, \overrightarrow{\pi}) \rangle$ will be called vector of envy *of allocation $\overrightarrow{\pi}$.*

Note that an allocation $\overrightarrow{\pi}$ is envy-free if and only if $\overrightarrow{e}(\overrightarrow{\pi}) = \langle 0, ..., 0 \rangle$.

2.2 Relaxations of envy-freeness

Different relaxations of the envy-freness notions have been proposed to measure the fairness of an allocation when there is no envy-free solution. They correspond to weaker solution concepts that are easier to satisfy. *Envy-freness up to one good* (EF1) [10, 4] is one of the most studied relaxations. An allocation is said to be envy-free up to one good if, for each envious agent i, the envy of i towards an agent j can be eliminated by removing an item from the bundle of j.

Definition 5 *Let $I = \langle \mathcal{N}, O, w \rangle$ be an add-MARA instance $\overrightarrow{\pi}$ be an allocation of I. $\overrightarrow{\pi}$ is* envy-free up to one good *if and only if $\forall i, j \in \mathcal{N}$, either $u_i(\pi_i) \geq u_i(\pi_j)$ or $\exists o_k \in \pi_j$ such that $u_i(\pi_i) \geq u_i(\pi_j) \backslash \{o_k\}$.*

It has been proved that an EF1 allocation always exists and, in the additive case, an EF1 allocation can be obtained using a round-robin protocol.

Caragiannis *et al.* [6] proposed another relaxation of envy-freeness that is closer to the original notion. An allocation is said to be *envy-free up to any good* (EFX) if for all envious agents i, the envy of i towards j can be eliminated by removing *any* item from the bundle of j.

Definition 6 *Let $I = \langle \mathcal{N}, O, w \rangle$ be an add-MARA instance and $\overrightarrow{\pi}$ be an allocation of I. $\overrightarrow{\pi}$ is* envy-free up to one good *if and only if $\forall i, j \in \mathcal{N}$, either $u_i(\pi_i) \geq u_i(\pi_j)$ or $\forall o_k \in \pi_j$ $u_i(\pi_i) \geq u_i(\pi_j \backslash \{o_k\})$.*

Clearly, we have $EF \implies EFX \implies EF1$. While an EF1 allocation can be computed in polynomial time, the guarantee of existence of an EFX allocation (where the full set of objects is allocated) remains an open issue [5].

2.3 Other fairness notions

Other notions of fairness have been introduced in the literature. Bouveret *et al.* [3] showed that some connections can be drawn between some widely used notions among which the min-max fair share, the proportional fair share and the max-min fair share.

Definition 7 *For a MARA instance $\mathcal{I}$ we define the min-max share (mMS) of agent i as follows :*

$$u_i^{mMS} \stackrel{def}{=} \min_{\overrightarrow{\pi} \in \mathcal{P}(I)} \max_{j \in \mathcal{N}} u_i(\pi_j)$$

Besides, we say that an allocation $\overrightarrow{\pi}$ is min-max fair share if every agent gets at least her min-max share ; formally : $u_i(\pi_i) \geq u_i^{mMS}$

Definition 8 *For a MARA instance $\mathcal{I}$ we define the proportional share (PS) of agent i as follows :*

$$u_i^{PS} \stackrel{def}{=} \frac{1}{n} u_i(\mathcal{N})$$

Besides, we say that an allocation $\overrightarrow{\pi}$ is proportional fair share if every agent gets at least her proportional share ; formally : $u_i(\pi_i) \geq u_i^{PS}$.

Definition 9 *For a MARA instance $\mathcal{I}$ we define the max-min share (MMS) of agent i as follows :*

$$u_i^{MMS} \stackrel{def}{=} \max_{\overrightarrow{\pi} \in \mathcal{P}(I)} \min_{j \in \mathcal{N}} u_i(\pi_j)$$

Besides, we say that an allocation $\overrightarrow{\pi}$ is max-min fair share if every agent gets at least her max-min share ; formally : $u_i(\pi_i) \geq u_i^{MMS}$.

Bouveret et al. [3] showed that these notions form a linear scale of increasing requirements where :

$$EF \implies mMS \implies PS \implies MMS$$

This scale can be used to characterize the level of fairness of a given allocation. In this hierarchy, EF is the most demanding criterion while the max-min fair share impose few restrictions.

3 MinOWA Envy

Our approach elaborates on minimizing the degree of envy of the agents while balancing the envy among the agents as suggested by [10]. The general idea would be to look for allocations that minimize this vector of envy in some sense : the lower this vector is, the less envious the agents are. This corresponds to a multiobjective optimization problem where each component of the envy vector is a different objective to minimize.

3.1 Fair OWA

There are different ways to tackle this minimization problem, each approach conveying a different definition of minimization. Our approach, guided by the egalitarian notion of fairness [13], is to ensure that the envy is as equally distributed as possible amongst agents. To this end, we will use a prominent aggregation operator that can convey fairness requirements : order weighted averages.

Ordered Weighted Averages (OWA) have been introduced by Yager [14] with the idea to build a family of aggregators that can weight the importance of objectives (or agents) according to their relative utilities, instead of their identities. In this way, we can explicitly choose to favour the poorest (or richest) agents, or to concentrate the importance of the criterion on the middle-class agents. Formally, the OWA operator is defined as follows :

Definition 10 *Let $\overrightarrow{\alpha} = \langle \alpha_1, \ldots, \alpha_n \rangle$ be a vector of weights. In the context of minimization, the* ordered weighted average *parameterized by $\overrightarrow{\alpha}$ is the function :*

$$owa^{\overrightarrow{\alpha}} : \overrightarrow{x} \mapsto \sum_{i=1}^{n} \alpha_i \times x_i^{\downarrow},$$

where $\overrightarrow{x}^{\downarrow}$ denotes a permutation of $\overrightarrow{x}$ where $x_1^{\downarrow} \geq x_2^{\downarrow} \geq \ldots \geq x_n^{\downarrow}$.

If we want to convey some notion of fairness, we have to give more weights to the unhappiest agents. Intuitively, it means that the weights in $\overrightarrow{\alpha}$ should be decreasing. This notion can be formalized by the following property. Let $\overrightarrow{x}$ be a vector such that $x_j \geq x_i$ (i is better off than j) and let ε be such that $0 \leq \varepsilon \leq 2(x_j - x_i)$. Then, for any non-increasing vector $\overrightarrow{\alpha}$:

$$owa^{\overrightarrow{\alpha}}(\overrightarrow{x}) \leq owa^{\overrightarrow{\alpha}}(\langle x_1, \ldots, x_i + \varepsilon, \ldots, x_j - \varepsilon, \ldots, x_n \rangle)$$

In other words, such an OWA favours any transfer of wealth from an happier agent to an unhappier agent. Such a transfer is called a *Pigou-Dalton* transfer, and the OWA with non-increasing weight vectors $\overrightarrow{\alpha}$ are called *fair OWAs*.

Since our motivation is to minimize the envy while equally distributing it between the agents, we propose to minimise the fair OWAs of the envy vector. Our solution concept can then be defined as follows :

Definition 11 *Let $I = \langle \mathcal{N}, O, w \rangle$ be an add-MARA instance and $\overrightarrow{\alpha}$ be a non-increasing vector. An allocation $\overrightarrow{\pi}$ is an* $\overrightarrow{\alpha}$-minOWA Envy allocation *if :*

$$\overrightarrow{\pi} \in \text{argmin}_{\overrightarrow{\pi} \in \mathcal{P}(I)}(owa^{\overrightarrow{\alpha}}(\overrightarrow{e}(\overrightarrow{\pi}))).$$

It is important to note that a major advantage of this solution is that it always exists as it is the result of an optimization process. Moreover, as we will see, this optimization problem can be modeled as an Integer Linear Program,

which will enable efficient computation of optimal allocations.

3.2 Linearization of OWA minimization

By using a linearization introduced by Ogryczak in [12] we can model our problem (minimization of the OWA of the envy vector) as a linear program. This linearization smartly uses the definition of OWA with its Lorenz components. Let $e_{(k)}$ denote the k^{th} bigger envy of an agent and $L_{(k)}$ the k^{th} Lorenz component of the Lorenz vector L. Moreover we consider decreasing OWA weights (because we consider here a fair OWA) so $\alpha_1 \geq \alpha_2 .... \geq \alpha_n$ and we denote $\overrightarrow{\alpha'} = (\alpha_1 - \alpha_2, \alpha_2 - \alpha_3, ..., \alpha_n)$. Finally, we recall the definition of the Lorenz vector L= $(e_{(1)}, e_{(1)}+e_{(2)}, ..., \sum_{i=1}^{n} e_{(i)})$. The OWA of a vector can be written with its Lorenz components :

$$\begin{aligned} \min \mathrm{OWA}(\overrightarrow{e}(\overrightarrow{\pi})) &= \min \sum_{k=1}^{n} \alpha_k e_{(k)} \\ &= \min \sum_{k=1}^{n} \alpha'_k \mathrm{L}_k(\overrightarrow{e}) \end{aligned}$$

Besides, there is a known LP to compute $L_k(\overrightarrow{e})$:

$$\mathrm{L}_k(\overrightarrow{e}) = \max \sum_{i=1}^{n} a_i^k e_i$$

$$s.t. \begin{cases} \sum_{i=1}^{m} a_i^k = k & *r_k \\ a_i^k \in [0,1] \quad \forall i \in [\![1,n]\!] & *b_i^k \end{cases}$$

However, we cannot inject this LP in the previous one as their optimization directions are not similar. This is why we use the dual (with the dual variables shown in red in the previous LP) of the LP :

$$\mathrm{L}_k(\overrightarrow{e}) = \min \; kr_k - \sum_{i=1}^{n} b_i^k$$

$$s.t. \begin{cases} r_k + b_i^k \geq e_i & \forall i \in [\![1,n]\!] \\ b_i^k \geq 0 & \forall i \in [\![1,n]\!] \end{cases}$$

The linearization of OWA is over but we still have to write three types of constraints directly related to our problem. First, we want to express the fact that any item j can be held by at most one agent $\sum_{i=1}^{n} z_i^j = 1$. We also should write the constraint expressing completeness (every item has to be allocated) $\sum_{i=1}^{n} \sum_{j=1}^{m} z_i^j = m$ but we note that adding this constraint is redundant as it is the sum over the items of the previous constraint. Finally, we have to make the link between the envy of agent a_i denoted by e_i and the utilities of the agents. e_i is the maximum of the envies between the agents, hence we linearize that maximum easily as we are in a minimization problem. For e_i we write n constraints expressing that e_i is greater than or equal to how much a_i envies another agent. We can finally write the final Mixed Integer Linear Program :

$$\min \mathrm{OWA}(\overrightarrow{e}(\overrightarrow{\pi})) = \min \sum_{k=1}^{n} \alpha'_k (kr_k + \sum_{i=1}^{n} b_i^k)$$

$$\begin{cases} r_k + b_i^k \geq e_i & \forall i,k \in [\![1,n]\!] \\ e_i \leq r_k + b_i^k & \forall i,k \in [\![1,n]\!] \\ e_i \geq \sum_{j=1}^{m} w(i,o_j)(z_h^j - z_i^j) & \forall i,h \in [\![1,n]\!] \\ \sum_{i=1}^{n} z_i^j = 1 & \forall j \in [\![1,m]\!] \\ z_i^j \in \{0,1\} \quad \forall j \in [\![1,m]\!] & \forall i \in [\![1,n]\!] \\ b_i^k \geq 0 & \forall i,k \in [\![1,n]\!] \\ e_i \geq 0 & \forall i \in [\![1,n]\!] \end{cases}$$

4 Link with other fairness measures

We focus here on the possible links between the min OWA allocation and other fairness measures such as maxmin fair share, envy-freeness up to one good and envy-freeness up to any good. We recall that if an envy-free allocation exists, it will be returned by the min OWA optimization.

4.1 Commensurability issues

We first show that if the utilities of the agents are not commensurable, we can find an example for which the min OWA allocation violates EF1 and MMS. Let's consider the following MARA instance involving 3 agents and 4 objects :

Π_1	o_1	o_2	o_3	o_4
a_1	6	**3**	2	**1**
a_2	0.07	0.06	**0.04**	0.03
a_3	**20**	0	0	0

Π_2	o_1	o_2	o_3	o_4
a_1	6	**3**	**2**	**1**
a_2	0.07	0.06	0.04	0.03
a_3	**20**	0	0	0

Π_3	o_1	o_2	o_3	o_4
a_1	6	**3**	2	1
a_2	0.07	0.06	**0.04**	**0.03**
a_3	**20**	0	0	0

The allocation Π_1 is obviously EF1. However, the allocation Π_2 is the one returned by the OWA minimization and it is not EF1. Indeed, a_2 gets no item in this allocation and a_1 gets three items that a_2 values strictly positively.

Hence, removing one item from a_1's bundle is not enough for a_2 not to envy a_1 anymore. Besides, we can notice that $u_1^{\text{MMS}} = 3$, $u_2^{\text{MMS}} = 0.06$ and $u_3^{\text{MMS}} = 0$ so the allocation Π_3 is MMS. So, we have found that the min OWA allocation violates EF1 and MMS.
Finally, by going up in the hierarchy introduced by Bouveret et al., we would like to check whether the absence of commensurability prevents the min OWA allocation from being PS or mMS. This time, we cannot conclude with the same instance as the previous example has no PS allocation.

We thus consider the following instance :

Π_1	o_1	o_2	o_3	o_4
a_1	10	6	6	1
a_1	0.1	0.06	0.06	0.01
a_1	1	6	6	10

Π_2	o_1	o_2	o_3	o_4
a_1	10	6	6	1
a_1	0.1	0.06	0.06	0.01
a_1	1	6	6	10

The allocation Π_1 is mMS as $u_1^{\text{mMS}} = u_3^{\text{mMS}} = 100 * u_2^{\text{mMS}} = 10$. It is then also PS. However, the allocation Π_2 is the one returned by the minimization of the envy vector and it is obviously not PS nor mMS. This is why from now on we will only consider MARA instances with commensurable agents (with K constant) :

$$\sum_{j=1}^{m} w(i, o_j) = K \quad \forall i \in [\![1, n]\!]$$

4.2 Two-agent settings

In the special case where the allocation problem involves only two agents, we highlight strong connections between the min OWA allocation and other fairness measures.

Proposition 1 *When the MARA instance involves only 2 agents then, the min OWA allocation is a max-min fair share allocation (thus also EFX).*

Proof 1 *For MARA instances where an envy-free allocation exists, our proof is straightforward as min OWA returns the EF allocation. It is thus also MMS, EF1 and EFX.*

We now focus on MARA instances for which there is no EF allocation. In the presence of only 2 agents any min OWA allocation $\overrightarrow{\pi}$ is such that only one of the two agents is envious. Indeed, if no agent is envious then it means the MARA instance has an envy-free allocation (which is a contradiction). Similarly, if both agent are envious it means there is an envy-free allocation (which is again a contradiction) as the agents would just have to exchange their bundles to obtain that allocation. Consequently, the sorted envy vector will be of this form $(e, 0)$. So, let us consider that such an allocation is not MMS. The agent that is envy-free obviously has her max-min share. So, under the assumption that the allocation is not MMS, the envious agent does not have her max-min share. But, in this case, the envious agent could obtain a better share and hence would have a lower envy. This leads to a contradiction because we consider the min OWA allocation. Indeed, in this case just, the min OWA allocation minimizes the envy of the envious agent as the other was not envious. It is known [6] in the pairwise setting that MMS implies EFX. □

4.3 EF1 for the general case : $n \geq 3$

We now turn to more general settings involving at least 3 agents. Since an EF1 allocation is guaranteed to exist, we more specifically focus on the possible links between min OWA and EF1. Unfortunately, we mainly obtain negative results.

Proposition 2 *In the general case, the min OWA allocation is not necessarily EF1.*

Proof 2 *We will prove this proposition through two examples with different OWA weights :*

Example 1 *We consider a MARA instance involving 3 agents and 7 objects, with the OWA weights $\alpha = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Minimizing OWA of the envy vector using this set of weight consists in minimizing the sum of the envies of the agents.*

Π_1	o_1	o_2	o_3	o_4	o_5	o_6	o_7
a_1	5	1	1	1	1	1	1
a_2	5	1	1	1	1	1	1
a_3	5	1	1	1	1	1	1

Π_2	o_1	o_2	o_3	o_4	o_5	o_6	o_7
a_1	5	1	1	1	1	1	1
a_2	5	1	1	1	1	1	1
a_3	5	1	1	1	1	1	1

The allocation Π_1 is the one returned by the minimization of the OWA of the envy vector. a_1 is envy free whereas a_2 and a_3 both envy a_1 by respectively 3 and 1. Hence, the sorted envy vector is $(3, 1, 0)$ and the value of OWA is $\frac{4}{3}$. Moreover, this allocation is not EF1 as a_2 would still envy a_3 by 1 even if one item is removed from the latter.

The allocation Π_2 has the same OWA value ($\frac{4}{3}$) but it is EF1. In fact, in this allocation, a_2 does not envy a_3. We could then argue that one of the min OWA allocations was EF1. This is why we show a stronger example for which the unique min OWA allocation is not EF1.

Example 2 *We consider a MARA instance involving 3 agents and 4 objects, with the OWA weights $\alpha = (1, 0, 0)$.*

Minimizing OWA of the envy vector using this set of weight consists in minimizing the maximum envy of the agents.

Π_1	o_1	o_2	o_3	o_4
a_1	14	*3*	*2*	1
a_2	7	6	4	*3*
a_3	*20*	0	0	0

Π_2	o_1	o_2	o_3	o_4
a_1	14	*3*	2	1
a_2	7	6	*4*	*3*
a_3	*20*	0	0	0

The allocation Π_1 corresponds to the min OWA one (this is the only one min OWA allocation). The OWA value of the envy vector is 9 (the envy of a_1 towards a_3). However, the allocation is not EF1. In fact, regardless of the item we remove from a_1, a_2 will still envy her.

The allocation Π_2 is EF1 but its OWA value is 11 (it still stands for the envy of a_1 towards a_3) which is quite far from the min OWA solution. Moreover, contrary to the previous example, a small change in the OWA weight vector does not change the situation. For instance, even for $\alpha = (0.9, 0.1, 0)$ the ordering stays the same.

From the previous examples, we can conclude that the minimization of the OWA of the envy vector does not necessarily return an EF1 allocation in the general case. □

However, we can wonder if we could always find a set of weights that returns an EF1 by minimizing the OWA. We leave this question open for the moment.

4.4 Continuity of EF1

We showed that the weights of the OWA influences the EF1 property of the min OWA solution. We can thus wonder whether there exists, among all the allocations, a kind of continuity of the EF1 solutions. It would be the case if by sorting all the allocations by their OWA value, all the EF1 allocations would be contained in a single interval of OWA value. We show that it is not the case through the following example.

Example 3 *We consider a MARA instance involving 3 agents and 3 objects, with the OWA weights $\alpha = (0.44, 0.36, 0.2)$:*

Π_1	o_1	o_2	o_3
a_1	*2*	1	3
a_2	2	*3*	1
a_3	1	0	*5*

Π_2	o_1	o_2	o_3
a_1	2	1	3
a_2	*2*	*3*	1
a_3	1	0	*5*

Π_3	o_1	o_2	o_3
a_1	2	1	*3*
a_2	2	*3*	1
a_3	*1*	0	5

Π_4	o_1	o_2	o_3
a_1	2	1	*3*
a_2	*2*	*3*	1
a_3	1	0	5

Π_5	o_1	o_2	o_3
a_1	*2*	*1*	3
a_2	2	3	1
a_3	1	0	*5*

For the purpose of the proof, we extract 5 allocations among all possible allocations. Allocations Π_1 to Π_5 are sorted by increasing values of the OWA of their envy vector. Π_1 is the best solution and Π_5 is the worst among the five selected allocations. The following table summarizes the OWA values of these allocations and specifies whether the allocations are EF1 or not :

	Π_1	Π_2	Π_3	Π_4	Π_5
OWA	0.44	1.32	1.76	2.20	2.20
$EF1$	$True$	$False$	$True$	$True$	$False$

We can easily see that there is no continuity of EF1 regarding the OWA values as Π_2 (which is between Π_1 and Π_3) is not EF1 whereas Π_1 and Π_3 are both EF1. Interestingly, it can be noticed that Π_4 and Π_5 have the same OWA values even if the allocations totally differ (not a single object has been given to the same agent). Moreover, Π_4 is EF1 whereas Π_5 is not.

5 Experimental results

We drew some experiments to compare the performances of the allocations obtained by min OWA envy with other allocation methods. The first method is a succession of sincere choices where the picking sequence is a concatenation of round robins. From all the possible allocations (for every possible round robin) generated by this method we keep the minRR (maxRR) that (respectly) is the allocation with the minimum OWA envy (maximum) while meanRR is the average of all these allocations. The max OWA util is the allocation that maximizes the OWA of the valuations of the agents. For the OWA allocation method, we used the following fair weight $(\frac{1}{2}, ..., \frac{1}{2^n})$.

We generated 100 random add-MARA instances for each method and for each couple $(|\mathcal{N}|,|\mathcal{O}|)$ from $(3, 4)$ to $(10, 12)$. We considered such couples of values in order to produce settings where few EF allocations exist as recommended in [8]. We did not go beyond 10 agents because of

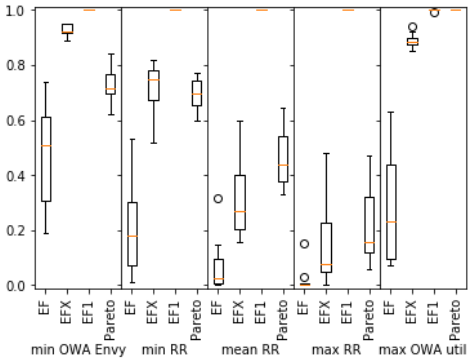


FIGURE 1 – Percentage of EF,EFX,EF1,PO for each method averaged on all ($|\mathcal{N}|$,$|\mathcal{O}|$)

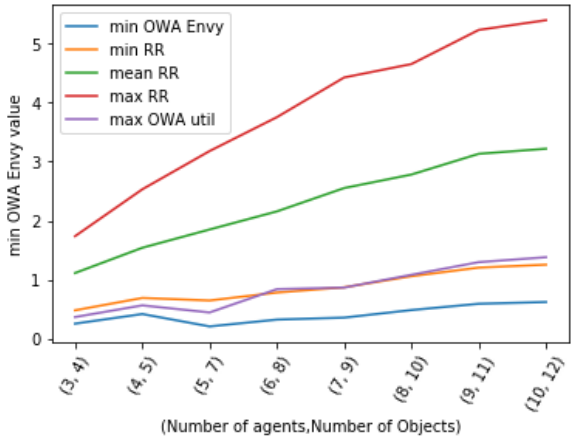


FIGURE 2 – Value of the OWA of the vector of envy for each allocation protocol

the computation time of the RR methods as we had to generate every possible allocation from every possible round robin. Indeed, even for 10 agents both the OWA methods were computed in less than two seconds (via Gurobi solver).

We evaluated the performances of the different methods by measuring the percentage allocations that are EF, EFX, EF1 and Pareto optimal. The results are aggregated in Figure 1. It can first be observed that our method obtains the highest rate of EF and EFX allocations : it is not that surprising for EF as our method is a relaxation of EF but the result about EFX is very promising. Every method has returned an EF1. Regarding the number of Pareto optimal allocations, our methods places second as maximizing OWA of the utilities necessarily produces a PO allocation.

Figure 2 shows how far the other methods are in terms of OWA envy values. We notice that even the closest method (max OWA util) is relatively far from our method, none of them has similar performances. Figure 3 shows that our

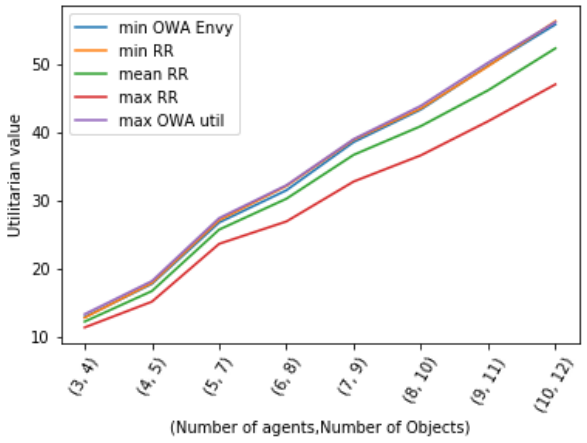


FIGURE 3 – Value of the sum of the vector of utilities for each allocation protocol

method has also good performances in terms of utilitarian criteria.

We performed same experiments for the OWA weights $(1, 0, 0, 0)$ and obtained similar results. For these weights the max OWA util does not always return a PO allocation and our method returns a lower rate of EFX (but still more than the other methods). This reinforces the idea that considering the envy of all agents and not only the maximum leads to fairer allocations.

6 Conclusion

In this paper, we introduced a new fairness concept following the idea of minimizing envy. More particularly, we used an OWA to express fairness in the distribution of envy between agents. After implementing a MIP to compute min OWA allocations, we showed the connections between the min OWA allocation and other famous fairness measures. We ran some experiments to test the performances of our method and compared it with other allocation protocols.

We also left the question of the performance of our method by considering the weights of the OWA as variables and not as a fix entry of the problem. We could focus on this matter for future works and try for example to learn them.

Finally, it could also be interesting to change the definition of the envy of an agent. Indeed, instead of defining it as the maximum of the pairwise envies we could define it as other aggregations of the pairwise envies such as their sum or even their OWA.

Références

[1] Amanatidis, Georgios, Georgios Birmpas et Vange-

lis Markakis: *Comparing approximate relaxations of envy-freeness*. Dans *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 42–48, 2018.

[2] Bansal, Nikhil et Maxim Sviridenko: *The santa claus problem*. Dans *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, STOC '06, pages 31–40, New York, NY, USA, 2006. ACM.

[3] Bouveret, Sylvain et Michel Lemaître: *Characterizing conflicts in fair division of indivisible goods using a scale of criteria*. Autonomous Agents and Multi-Agent Systems, 30(2):259–290, 2016, ISSN 1573-7454.

[4] Budish, Eric: *The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes*. Journal of Political Economy, 119(6):1061–1103, dec 2011.

[5] Caragiannis, I., N. Gravin et Xin Huang: *Envy-freeness up to any item with high nash welfare: The virtue of donating items*. note, 2019.

[6] Caragiannis, Ioannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah et Junxing Wang: *The unreasonable fairness of maximum nash welfare*. Dans *Proceedings of the 2016 ACM Conference on Economics and Computation*, EC '16, pages 305–322, New York, NY, USA, 2016. ACM, ISBN 978-1-4503-3936-0.

[7] Chevaleyre, Yann, Ulle Endriss et Nicolas Maudet: *Allocating goods on a graph to eliminate envy*. Dans *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07)*, Vancouver, British Columbia, Canada, juillet 2007.

[8] Dickerson, John P., Jonathan Goldman, Jeremy Karp, Ariel D. Procaccia et Tuomas Sandholm: *The computational rise and fall of fairness*. Dans *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI-14)*, pages 1405–1411, Québec City, Québec, Canada, juillet 2014. AAAI Press.

[9] Foley, Duncan K.: *Resource allocation and the public sector*. Yale Economic Essays, 7(1):45–98, 1967.

[10] Lipton, Richard, Evangelos Markakis, Elchanan Mossel et Amin Saberi: *On approximately fair allocations of divisible goods*. Dans *Proceedings of the 5th ACM Conference on Electronic Commerce (EC-04)*, pages 125–131, New York, NY, mai 2004. ACM.

[11] Nguyen, Trung Thanh et Jörg Rothe: *How to decrease the degree of envy in allocations of indivisible goods*. Dans Perny, Patrice, Marc Pirlot et Alexis Tsoukiàs (rédacteurs): *Algorithmic Decision Theory*, pages 271–284, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg, ISBN 978-3-642-41575-3.

[12] Ogryczak, Wlodzimierz et Tomasz Śliwiński: *On solving linear programs with the ordered weighted averaging objective*. European Journal of Operational Research, 148:80–91, 2003.

[13] Rawls, John: *A Theory of Justice*. Harvard University Press, Cambridge, Mass., 1971. Traduction française disponible aux éditions du Seuil.

[14] Yager, Ronald R.: *On ordered weighted averaging aggregation operators in multicriteria decision making*. IEEE Transactions on Systems, Man, and Cybernetics, 18:183–190, 1988.

Actes JIAF 2019

FUNN: Flexible Unsupervised Neural Network

David Vigouroux Sylvain Picard

IRT Saint Exupery, Toulouse, France
david.vigouroux@irt-saintexupery.com
sylvain.picard@irt-saintexupery.com

Résumé

Les réseaux neuronaux profonds ont démontré une grande précision dans les tâches de vision par ordinateur. Cependant, ils se sont avérés non robustes face aux exemples adversariaux. Une petite perturbation dans l'image peut totalement modifier le résultat d'une classification. Ces dernières années, plusieurs moyens de défense ont été proposés pour tenter de résoudre ce problème dans le cadre de tâches de classification supervisées, sans arriver à des résultats satisfaisants. Dans cet article nous proposons une méthode permettant d'obtenir des features robustes contre les attaques adversariales dans le cadre de tâches d'apprentissage non supervisées. Notre méthode se distingue des solutions existantes par l'apprentissage direct des features robustes sans qu'il soit nécessaire de projeter les exemples adversariaux dans l'espace de distribution des images d'origine. Un premier auto-encodeur, l'attaquant, est chargé de perturber l'image d'entrée afin de tromper un second auto-encodeur, le defenseur, qui lui est chargé de régénérer l'image d'origine. L'attaquant essaie de trouver l'image la moins perturbée sous la contrainte que l'erreur dans la sortie du defenseur soit au moins égale à un seuil. Grâce à cette formation, l'encodeur du defenseur sera robuste contre les attaques adversariales et pourra être utilisé dans différentes tâches comme la classification. En utilisant une architecture de réseau de l'état de l'art, nous démontrons la robustesse des fonctionnalités obtenues grâce à cette méthode dans les tâches de classification.

Abstract

Deep neural networks have shown high accuracy in computer vision tasks. However, they are known to be weak against adversarial examples. A small perturbation in the image can change the classification dramatically. In recent years, several defences methods have been proposed to solve the issue in the context of supervised classification tasks. We propose a method to find robust features against adversarial attacks in the context of unsupervised learning. Our method differs from existing solutions by directly learning the robust features without projecting the adversarial examples in the original distribution space.A first auto-encoder, called attacker, perturbs the input image in order to fool a second auto-encoder, called defender, which tries to regenerate the original image.The goal of the attacker is to perturb images as little as possible while guaranteeing that the reconstructed images will be at a given distance from the original images.After such training, we extract from the defender an encoder that should be robust against adversarial attacks. Using state-of-art network architectures, we demonstrate the robustness of the features obtained by this method in classification tasks.

1 Introduction

Neural networks and especially convolutional neural networks have shown impressive results in many different tasks in computer vision such as object detection, image recognition and segmentation. A downside of these techniques is their lack of robustness [3, 6, 9, 22] : imperceptible perturbations of the input (image, sound, ...) can disturb the networks and drastically change the output. Such perturbed inputs are called *"adversarial examples"* [6] , and it is possible to design algorithms that can generate such examples. Those algorithms are known as *"adversarial attacks"* [6]. We commonly distinguish two types of attacks:

— **white box** attacks, when the network architecture and the weights are known;
— **black box** attacks, when they are not.

This work focus on non-targeted white-box attacks computed using different attack methods.

The field of adversarial generation is an active research field. Various methods have been developed in order to produce adversarial example [25]. In this work, we consider five of the most common attack methods from the literature: the Fast Gradient Sign Method (FGSM) [6], Iterative FGSM [10], Single Pixel attack and LocalSearch [15] and Deepfool [14]. Those methods

are *gradient-based* or *score-based* approaches that try to find minimal perturbations that will change the model prediction.

Different methods have been proposed in order to deal with these adversarial attacks. Some approaches try to project the adversarial examples into the original distribution [12, 18, 19] while others propose to add adversarial examples into the training dataset [6, 14, 22], similarly to data augmentation. Finally, recent methods propose to transform the input data to make it less sensitive to perturbations [8]. All these methods are attack-specific, and thus not efficient against new or different attacks. Recently, new methods have been proposed to learn the attacking and defending concept concurrently in order to be independent of the attacking method. Some of those methods use Generative Adversarial Networks (GANs) to generate adversarial examples as in [11, 18] or Auto-Encoders as in [1, 4, 20].

In this paper, we attempt to learn an encoder with robust features in an unsupervised manner. For this, we train concurrently an *attacker* and a *defender*. The attacker generates perturbed images from the original image in order to fool the defender. The defender try to reconstruct the original image from the perturbed one in order to produce robust features.

2 Related Work

2.1 Attack strategies

Several attack strategies have been proposed, which can be split in two categories: **black-box** and **white-box** attacks. White-box attacks have access to the weights and gradients of the classifier while, black-box strategies only have access to the predictions of the network.In this work, we concentrate on two kind of white-box attacks: gradient-based and score-based. This section describes the method we consider.

Fast Gradient Sign Method (FGSM) [6] Given an image X and its corresponding label Y, the FGSM attack sets the perturbation δ to:

$$\delta(X, Y) = \epsilon \cdot sign(\nabla_X L(X, Y)) \tag{1}$$

where ϵ is a small number and ∇ is the gradient of the loss function L with respect to the input X. For each pixel, FGSM slightly increases or decreases its value depending on the sign of the gradient with respect to the pixel.

DeepFool [14] The DeepFool method performs an iterative attack using a linear approximation in order to find the shortest distance between the original input and the decision boundary of adversarial examples. If f is a binary differentiable classifier, an iterative method is used to approximate the perturbation. The minimal perturbation is computed as:

$$\begin{aligned} \arg\min_{\eta_i} \quad & \|\eta_i\|_2 \\ s.t. \quad & f(x_i) + \nabla f(x_i)^T \eta_i = 0 \end{aligned} \tag{2}$$

η_i is the noise level and x_i is the perturbed image at iteration i, f is the binary classifier.

The method can be extended to multi-class classifiers by finding the closest hyperplanes. DeepFool provides less perturbation and reduces intensity compared to FGSM.

Single Pixel [21] Single Pixel is an iterative methods that tries to create an adversarial example by setting the value of a **single** pixel in the image to the minimum or maximum value of any pixel in the image. The method randomly selects a set of candidate pixel to modify and then iterate over them until an adversarial example is generated. The method can fail if none of the candidate pixel can be modified to generate an adversarial example.

Local Search [15] This local search procedure tries to find a pixel (or a group of pixels) that is critical for the classifier robustness and then modify its value to generate adversarial images. The Local Search algorithm finds pixel locations to modify and then applies a fixed transformation function to the selected pixels in order to generate an adversarial image.

2.2 Defence strategies

Several studies have assumed that the lack of robustness comes from the fact that adversarial examples are outside of the dataset's distribution, near the border of decision as shown in [5, 24]. Multiple defence methods have been proposed to increase the robustness of deep neural networks against adversarial attacks. This section describes the most common strategies.

Adversarial training Adversarial training is an intuitive method that consists in augmenting the training dataset with adversarial examples [23]. While this method is efficient to increase robustness against adversarial examples similar to the ones added to the training dataset, it is attack-specific and thus poorly generalize to adversarial examples generated differently from the ones in the training dataset.

Defensive distillation Defensive distillation [16] is a method that train the classifier in two steps using a variation of the distillation [7] method. This creates a smoother network, thus reducing the amplitude of gradients around input points, and consequently increasing the robustness

against adversarial examples. However, this method has proven to be inefficient against recent black-box attacks [2].

Adversarial detectors Another method of defence is to detect adversarial examples [13] and exclude them. For each attack method considered, a classifier (detector) is trained to tell whether an input is normal or adversarial. The detector is directly trained on both normal and adversarial examples. This method shows good performance when the training and testing attack examples are generated from the same process and the perturbations are large enough, but does not generalize well across different attack parameters and attack generation processes.

MagNet MagNet [12] is a method based on adversarial detector strategy. It trains a reformer network (which is an auto-encoder or a collection of auto-encoders) to move adversarial examples closer to the manifold of legitimate, or natural, examples. It is an effective strategy against grey-box attacks where the attacker is aware of the network architecture and defences, but does not know its weights.

3 Proposed Method

3.1 Motivation

The previously described defence methods provide some intuition on what can make a network robust to adversarial examples. Our strategy differs from these by performing adversarial training with data generated by a separated network during training. Unlike adversarial detectors strategies, our generating network is only used during training, and not when doing inference. This paper focus only on unsupervised learning strategies, which is why we use auto-encoders. An auto-encoder is a deep neural network composed by an encoder, responsible for extracting deep features from input images, and a decoder, responsible for reconstructing images from the extracted features. During training, the network is optimized in order to generate images similar to input images by minimizing the distance between the original and generated images.

We propose a defence strategy to increase robustness against white box attacks using Euclidean norm (L_2) as training objective. We then demonstrate its efficiency against attacks that use L_2 or infinity norm (L_∞). The approach is to train two auto-encoders, an **attacker** and a **defender**. The attacker generates adversarial examples by perturbing the original image while the defender tries to reconstruct the original image from adversarial examples given by the attacker. The two models are trained concurrently and adapt themselves during training until convergence much like how Generative Adversarial Networks (GAN) are trained. At the end of the training, we expect that the encoder of the defender will be able to extract deep features which are robust against the perturbations generated by the attacker. We expect this training method to generate deep features representing a larger distribution of input images than the original one, and thus increase the robustness of the defending auto-encoder.

The attacker model is similar to a style transfer model, it must reconstruct the input image while adding noise in order to induce errors in the defender output. Thus the attacker has two adversarial training objectives:

— generate an image close to the input image;
— add perturbation to the generated image in order to disrupt the defender with a minimum noise level β.

The objective of the defender is to produce an image as close as possible to the original image (input of the attacker) using the image generated by the attacker.

The distance between the original and reconstructed image, L_α, is the loss function of the problem, and can be any distance. In our experiences, we choose the Euclidean distance. The corresponding optimization problem is described by equation (3–5). This optimization problem represents a non zero-sum game.

$$\min_{\theta_{att}} \quad E_{x\sim\chi}\left(L_\alpha\left(Att\left(x,\theta_{att}\right),x\right)\right) \tag{3}$$

$$\min_{\theta_{def}} \quad E_{x\sim\chi}(L_\alpha(x, Def(Att(x,\theta_{att}),\theta_{def}))) \tag{4}$$

$$E_{x\sim\chi}(L_\alpha(x, Def(Att(x,\theta_{att}),\theta_{def}))) \geq \beta \tag{5}$$

where:

— χ is the ensemble of distribution examples,
— θ_{att} is the weight of the *attacker* auto-encoder,
— θ_{def} is the weight of the *defender* auto-encoder,
— $L_\alpha : \chi, \chi \rightarrow R$ is the loss function,
— $Att(x,\theta_{att}) : \chi \rightarrow \chi$ is a function with parameters θ_{att},
— $Def(x,\theta_{def}) : \chi \rightarrow \chi$ is a function with parameters θ_{def}.

Objective (3) is to minimize the distance between the original image and the perturbed one (attacker). Objective (4) is to minimize the distance between the original image and the reconstructed image (defender). Constraint (5) enforces a given minimum distance between the original image and the reconstructed one.

To solve this under constraint optimization problem, we choose to relax Constraint (5) by penalizing it in the objective function:

$$\min_{\theta_{att}} E_{x\sim\chi}(L_\alpha(Att(x,\theta_{att}),x)) + \gamma * \max(E_{x\sim\chi}(L_\alpha(x, Def(Att(x,\theta_{att}),\theta_{def}))) - \beta, 0.0)$$

where γ is an hyper-parameter that controls noise generation and must be large enough to satisfy Constraint (5).

3.2 Model and training procedure

The attacker is implemented as an auto-encoder where an uniform random vector is concatenated to the encoder's features and then passed through the generator. Adding an uniform random vector allows the attacker to generate noise without modifying the real features. The defender is implemented as a standard auto-encoder.

We use a two-stage approach, with an *initialization stage* that trains the encoder of attacker and the decoder of the defender, and an *optimization stage* that deals with the real optimization problem.

During the initialization stage, both auto-encoders are trained to generate realistic images, without constraints (Equation (3, 4)). This first phase facilitates the training procedure and guarantees that the optimization stage goes in the correct direction.

During the optimization stage, the weights of the attacker's encoder and the defender's generator are not updated anymore. This choice is justified since:

— The purpose of the attacker is to produce adversarial examples from given non-perturbed examples. After the initialization stage, the encoder of the attacker is already able to compress input images into their features and does not need to be trained anymore. During the optimization stage the attacker can be seen as an adversarial example generator taking as input previously learned features concatenated with random noise.

— The goal of the defender is to be robust against adversarial attacks. After the initialization stage, the generator of the defender is able to reproduce correct images from good features. We want the features of adversarial examples to be identical to the features of non-perturbed examples. By fixing the generator weights, we want the defender to extract features from perturbed images that are very close to the features of non-perturbed images.

3.3 Testing procedure

Since auto-encoders are architectures built for unsupervised learning, no classifiers are used during training. After training, in order to be able to use adversarial attacks and estimate the robustness of the network, we train a classifier using the features produced by the encoder of the defender on images coming from the original dataset. The robustness of this classifier is evaluated against the following attacks: DeepFool (L_2-norm and L_∞-norm), Single Pixel, LocalSearch and FGSM.

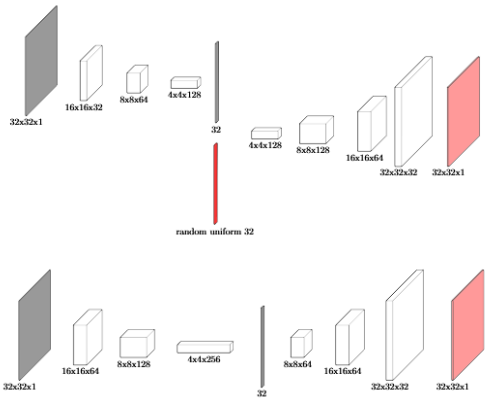


Figure 1 – Network architectures.

We evaluate robustness using two criterion:

— the *attack success rate*, i.e., the percentage of times the attack was able to fool the classifier;

— the *noise level* of adversarial examples defined as the L_2-norm between the original image and the generated adversarial example.

4 MNIST Experiments

4.1 Configuration

We evaluate our method on the MNIST dataset with the networks shown in Figure 1.

The neural networks use convolutionnal layers with kernel size 3x3, strides 2x2 and ReLU activation functions except for the last layer. The last layers have $(1+\tanh)/2$ as activation function. We use a batch size of 128 and Adam optimizers with learning rates described in Table 1.

	Initialisation	Optimisation
Defender	5.10^{-4}	10^{-5}
Attacker	10^{-3}	10^{-5}

Table 1 – Learning rates for the Adam optimizer.

We use $\gamma = 5.0$, $\beta = 0.01$ and the L_2 Euclidean norm as the loss function. We run the initialization stage for 9 epochs and the optimization stage for 31 epochs.

The classifier is trained during 20 epochs using an Adam optimizer with a learning rate of 10^{-3}, and a softmax-cross-entropy loss. The classifier uses the defender's encoder (whose weights are fixed) and a hidden dense layer of 64 units with ReLU activation functions.

The network architectures and hyper parameters (γ and β) have not been optimized for this task. However, several learning rates have been tested to achieve the

presented results: a bad choice of learning rate can create a non-robust network.

4.2 Results

Adversarial attacks tools Tools have been implemented to facilitate adversarial attacks implementation such as Cleverhans [**?**] or Foolbox [17]. Those tools are open-source frameworks that propose state-of-art algorithms used to generate adversarial examples on any model.

In order to compare our method against a traditional network, we trained a classic auto-encoder with an architecture similar to the defender architecture. We then train a classifier as previously described on the defender's encoder network. After 20 epochs, this classifier achieves an accuracy close to 97% on the MNIST test set. We then attack this classifier with several methods on the whole test set using Foolbox [17].

The evaluation is only done on the first 1000 images of the dataset due to computation time. Results are reported in Table 2

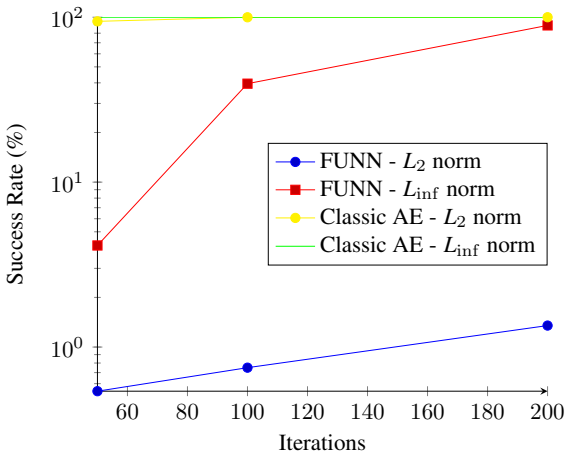


Figure 2 – DeepFool attacks

4.3 Analysis

DeepFool with L_2-norm In Table 2, we see that only 0.5% of attack succeed when DeepFool is used with a maximum of 50 iterations, generating a mean noise level around 0.0075. This is expected as, during learning, the attacker tries to find noise level greater than $\beta = 0.01$ (in fact, very close to β as explained above) which can fool the defender. If the defender manages to defend against this level of noise, we expect it to be robust against levels of noise lower than this threshold. The classifier seems to be robust against L_2 DeepFool attacks with at most 50 iterations. When we increase the DeepFool maximum number of iterations, the success rate of attacks slowly increases. With 200 iterations, 1.35% of attack succeeds, and with 1000 iterations, the percentage of successful attacks starts to becomes significant (15.25%), but the noise level increases significantly at the same time to reach 0.73 which is far from the threshold value of 0.01 used in training. One may think that increasing the threshold β during training would increase the robustness of our method, but using the MNIST dataset, it is impossible because doing so, the attacker would generate images that are almost entirely black. Indeed, the average L_2 distance between an MNIST image and a black image is around 0.07, it is thus not possible to increase the threshold that much.

DeepFool with L_∞-norm With the default maximum number of steps for DeepFool (50), the classifier seems to be robust (4.13% of success rate) for the L_∞ norm. However, the classifier quickly fails as the number of steps increase, with a 89.20% success rate of attack for 200 steps.Usually, such number of iteration steps are not tested in the literature, but this shows that even if some networks seem to be robust against few iterations of DeepFool, larger number of iterations often manage to fool most of these networks. However, even with 50 steps, the noise level is already quite high (1.13), which is why it does not necessarily mean that the network is not robust against this attack.

Single Pixel and LocalSearch Even if the attack success rate is relatively low (4.72%), this level of success rate is higher than what we may expect since the L_2 distances between input images and adversarial examples generated by single pixel are quite small. For our methods, this attack looks more powerful than DeepFool which is designed precisely to minimize the L_2-norm.

Gradient Methods FGSM and Gradient have a high attack success rate, near 100% for Gradient Sign and Gradient. This is expected since the defender is not trained against the high levels of noise generated by all these methods.

MagNet Comparaison MagNet [12] and our method show similar performances. However, MagNet uses an additional network in order to detect adversarial examples upstream to the main network. This detector network is also used during inference, inducing more computation and thus slowing down the inference process. Our method use two auto encoders during the training process, but only the encoder of the defender is used during inference, coupled with, e.g., a small classifier for computer vision tasks such as classification. The global architecture of the final network is thus much smaller than the one used during

	Classic Auto Encoder		MagNet	Our Method	
Attack Name	Success Rate	Noise Level	Success Rate	Success Rate	Noise Level
DeepFool (max-iter=50 – L_2)	94,41%	1,25	—	**0.54%**	**0.0075**
DeepFool (max-iter=50 – L_∞)	100%	1,81	**0.6%**	4.13%	**1.13**
FGSM (ϵ=0.005)	0.41%	0.07	3.2%	**1.61%**	**0.04**
Iterative (ϵ=0.005)	1.04%	0.08	4.8%	**1.25%**	**0.04**
Single Pixel	9.17%	1.0	—	**4.72%**	**1.0**
LocalSearch	44.41%	5.22	—	**27.51%**	**9.71**

Table 2 – Attack success rates and noise levels for various attacks.

training, and the computation time required for inference is reduced without loss of robustness.

5 Conclusion

Without adding extra computational complexity, such as image filtering or adversarial image detection, we demonstrated that a classical network can be robust against L_2 attacks by using an appropriate unsupervised learning procedure. This procedure shows promising results for learning robust unsupervised networks. However, even if it was expected due to the defence strategy, the trained network is not robust against all types of attacks, in particular it is weak against gradient attacks. Our future work will focus on creating a more general defence learning procedure, by using different distance methods. We also plan to test this method on more complex datasets, such as CIFAR-10 and ImageNet, with different network architectures.

References

[1] Baluja, Shumeet and Ian Fischer: *Learning to Attack: Adversarial Transformation Networks*. page 9.

[2] Carlini, Nicholas and David Wagner: *Defensive Distillation is Not Robust to Adversarial Examples*. arXiv:1607.04311 [cs], July 2016. http://arxiv.org/abs/1607.04311, visited on 2019-05-24, arXiv: 1607.04311.

[3] Chakraborty, Anirban, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay: *Adversarial Attacks and Defences: A Survey*. arXiv:1810.00069 [cs, stat], September 2018. http://arxiv.org/abs/1810.00069, visited on 2019-06-06, arXiv: 1810.00069.

[4] Folz, Joachim, Sebastian Palacio, Joern Hees, Damian Borth, and Andreas Dengel: *Adversarial Defense based on Structure-to-Signal Autoencoders*. arXiv:1803.07994 [cs, stat], March 2018. http://arxiv.org/abs/1803.07994, visited on 2019-05-24, arXiv: 1803.07994.

[5] Gilmer, Justin, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow: *Adversarial Spheres*. http://arxiv.org/abs/1801.02774, visited on 2019-05-24, arXiv: 1801.02774, January 2018.

[6] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy: *Explaining and Harnessing Adversarial Examples*. arXiv:1412.6572 [cs, stat], December 2014. http://arxiv.org/abs/1412.6572, visited on 2019-05-24, arXiv: 1412.6572.

[7] Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean: *Distilling the Knowledge in a Neural Network*. arXiv:1503.02531 [cs, stat], March 2015. http://arxiv.org/abs/1503.02531, visited on 2019-05-24, arXiv: 1503.02531.

[8] Kabilan, Vishaal Munusamy, Brandon Morris, and Anh Nguyen: *VectorDefense: Vectorization as a Defense to Adversarial Examples*. arXiv:1804.08529 [cs], April 2018. http://arxiv.org/abs/1804.08529, visited on 2019-05-24, arXiv: 1804.08529.

[9] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio: *Adversarial examples in the physical world*. arXiv:1607.02533 [cs, stat], July 2016. http://arxiv.org/abs/1607.02533, visited on 2019-05-24, arXiv: 1607.02533.

[10] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio: *Adversarial Machine Learning at Scale*. arXiv:1611.01236 [cs, stat], November 2016. http://arxiv.org/abs/1611.01236, visited on 2019-05-24, arXiv: 1611.01236.

[11] Lee, Hyeungill, Sungyeob Han, and Jungwoo Lee: *Generative Adversarial Trainer: Defense to Adversarial Perturbations with GAN*. arXiv:1705.03387 [cs, stat], May 2017. http://arxiv.org/abs/1705.03387, visited on 2019-05-24, arXiv: 1705.03387.

[12] Meng, Dongyu and Hao Chen: *MagNet: A Two-Pronged Defense against Adversarial Examples*. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* -

CCS '17, pages 135–147, Dallas, Texas, USA, 2017. ACM Press, ISBN 978-1-4503-4946-8. http://dl.acm.org/citation.cfm?doid=3133956.3134057, visited on 2019-05-24.

[13] Metzen, Jan Hendrik, Tim Genewein, Volker Fischer, and Bastian Bischoff: *On Detecting Adversarial Perturbations*. arXiv:1702.04267 [cs, stat], February 2017. http://arxiv.org/abs/1702.04267, visited on 2019-05-24, arXiv: 1702.04267.

[14] Moosavi-Dezfooli, Seyed Mohsen, Alhussein Fawzi, and Pascal Frossard: *DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks*. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, Las Vegas, NV, USA, June 2016. IEEE, ISBN 978-1-4673-8851-1. http://ieeexplore.ieee.org/document/7780651/, visited on 2019-05-24.

[15] Narodytska, Nina and Shiva Prasad Kasiviswanathan: *Simple Black-Box Adversarial Perturbations for Deep Networks*. arXiv:1612.06299 [cs, stat], December 2016. http://arxiv.org/abs/1612.06299, visited on 2019-05-24, arXiv: 1612.06299.

[16] Papernot, N., P. McDaniel, X. Wu, S. Jha, and A. Swami: *Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks*. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597, May 2016.

[17] Rauber, Jonas, Wieland Brendel, and Matthias Bethge: *Foolbox: A Python toolbox to benchmark the robustness of machine learning models*. arXiv:1707.04131 [cs, stat], July 2017. http://arxiv.org/abs/1707.04131, visited on 2019-05-24, arXiv: 1707.04131.

[18] Samangouei, Pouya, Maya Kabkab, and Rama Chellappa: *Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models*. arXiv:1805.06605 [cs, stat], May 2018. http://arxiv.org/abs/1805.06605, visited on 2019-05-24, arXiv: 1805.06605.

[19] Santhanam, Gokula Krishnan and Paulina Grnarova: *Defending Against Adversarial Attacks by Leveraging an Entire GAN*. arXiv:1805.10652 [cs, stat], May 2018. http://arxiv.org/abs/1805.10652, visited on 2019-05-24, arXiv: 1805.10652.

[20] Srinivasan, Vignesh, Arturo Marban, Klaus Robert Müller, Wojciech Samek, and Shinichi Nakajima: *Counterstrike: Defending Deep Learning Architectures Against Adversarial Samples by Langevin Dynamics with Supervised Denoising Autoencoder*. arXiv:1805.12017 [cs, stat], May 2018. http://arxiv.org/abs/1805.12017, visited on 2019-05-24, arXiv: 1805.12017.

[21] Su, J., D. V. Vargas, and K. Sakurai: *One Pixel Attack for Fooling Deep Neural Networks*. IEEE Transactions on Evolutionary Computation, pages 1–1, 2019, ISSN 1089-778X.

[22] Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus: *Intriguing properties of neural networks*. arXiv:1312.6199 [cs], December 2013. http://arxiv.org/abs/1312.6199, visited on 2019-05-24, arXiv: 1312.6199.

[23] Tramèr, Florian, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel: *Ensemble Adversarial Training: Attacks and Defenses*. arXiv:1705.07204 [cs, stat], May 2017. http://arxiv.org/abs/1705.07204, visited on 2019-05-24, arXiv: 1705.07204.

[24] Tramèr, Florian, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel: *The Space of Transferable Adversarial Examples*. arXiv:1704.03453 [cs, stat], April 2017. http://arxiv.org/abs/1704.03453, visited on 2019-05-24, arXiv: 1704.03453.

[25] Yuan, X., P. He, Q. Zhu, and X. Li: *Adversarial Examples: Attacks and Defenses for Deep Learning*. IEEE Transactions on Neural Networks and Learning Systems, pages 1–20, 2019, ISSN 2162-237X.