



HAL
open science

Abstract Argumentation with Qualitative Uncertainty: An Analysis in Dynamic Logic

Andreas Herzig, Antonio Yuste-Ginel

► **To cite this version:**

Andreas Herzig, Antonio Yuste-Ginel. Abstract Argumentation with Qualitative Uncertainty: An Analysis in Dynamic Logic. Logic and Argumentation: 4th International Conference, CLAR 2021, Hangzhou, China, October 20–22, 2021, Proceedings, 13040, Springer International Publishing, pp.190-208, 2021, Lecture Notes in Computer Science book series (LNCS), 978-3-030-89390-3. 10.1007/978-3-030-89391-0_11 . hal-03573580

HAL Id: hal-03573580

<https://hal.science/hal-03573580>

Submitted on 15 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Abstract Argumentation with Qualitative Uncertainty: An Analysis in Dynamic Logic

Andreas Herzig¹  and Antonio Yuste-Ginel²  

¹ IRIT, Toulouse, France
herzig@irit.fr

² Departamento de Filosofía, Universidad de Málaga, Málaga, Spain

Abstract. We extend the existing encoding of abstract argumentation frameworks in DL-PA (Dynamic Logic of Propositional Assignments) in order to capture different formalisms for arguing with qualitative forms of uncertainty. More in particular, we encode the main reasoning tasks of (rich) incomplete argumentation frameworks and control argumentation frameworks. After that, and inspired by our encoding, we define and study a new class of structures that are shown to be maximally expressive: *constrained incomplete argumentation frameworks*.

Keywords: Incomplete argumentation frameworks · Dynamic logic of propositional assignments · Control argumentation frameworks

1 Introduction

Formal argumentation has been proved to be a successful approach to non-monotonic reasoning (see e.g. [15]), among many other applications [2, 12]. Within the studies directed to provide a formal model for argument-based inference, abstract models of argumentation play a crucial role, as they answer a rather fundamental question: how should a rational agent choose among a conflicting set of arguments those that are better justified? The adjective *abstract* stresses that these models disregard the nature and structure of arguments, in order to focus on the different semantics through which one could give a precise answer to the question above. The foremost abstract model of argumentation is the use of directed graphs, first proposed by Dung in [23] under the name of *argumentation frameworks* (AFs), where nodes stand for arguments and arrows stand for attacks among arguments.

Andreas Herzig is partially supported by the EU ICT-48 2020 project TAILOR (No. 952215). Antonio Yuste-Ginel gratefully acknowledges funding received from the PhD grant No. MECDFPU 2016/04113.

Despite being an elegant and powerful tool, AFs have limited modelling capabilities for many purposes. Consequently, many extensions of Dung’s model have been proposed in the literature since its publication. Examples of such extensions are the addition of a support relation [16], of recursive forms of attacks [6] and of preferences among arguments [1]. An essential limitation that AFs come equipped with is the assumption that the formalized agent has perfect knowledge about the relevant arguments and attacks of the debate (that is, about the structure of the AF). This turns out to be an important shortcoming in adversarial contexts, where usually one wants to model the information (i.e., the part of an AF) that an agent thinks that her opponent entertains, and thus uncertainty naturally pops up into the picture. However, the assumption of perfect knowledge has been relaxed through the study of extensions of AFs that include different forms of uncertainty, either through the use of probability [28] or through qualitative methods. Among the second group of approaches, *incomplete argumentation frameworks* (IAFs) [8–11, 24] and *control argumentation frameworks* (CAFs) [18, 31] have recently received a lot of attention, resulting in a precise complexity map of the different associated reasoning tasks as well as some applications [19].

Concurrently, a large number of works within formal argumentation have focused on building a suitable logical theory for reasoning about argumentation formalisms, with a special focus on AFs and their dynamics (see [13] for a recent survey on the topic). The *dynamic logic of propositional assignments* (DL-PA) [4] has been shown to be a useful tool for this enterprise [20–22]. DL-PA is a well-behaved variant of *propositional dynamic logic* (PDL) [26], where atomic programs are restricted to assignments of propositional variables to either Truth or Falsity. It is expressive enough to capture all standard argumentation semantics. When compared to encodings in propositional logic, DL-PA can capture semantics that incorporate minimality or maximality criteria more succinctly. Moreover, its advantages over other encodings of AFs in equally succinct languages (e.g. *quantified Boolean formulas*) have also been highlighted [21].

Contribution and Structure. In this paper, we explain how to extend the machinery of [20–22] so as to use DL-PA for reasoning about different formalisms for arguing with qualitative uncertainty. In particular, and after introducing the basic tools (Sects. 2 and 3) we encode in DL-PA programs the main reasoning tasks concerning *incomplete argumentation frameworks* (Sect. 4), their enriched version (called *rich incomplete argumentation frameworks* [29]) (Sect. 5), and *control argumentation frameworks* (Sect. 6). After that, and inspired by our encoding, we define and study the expressive power of a new class of structures for arguing with qualitative uncertainty: *constrained incomplete argumentation frameworks*, whose naturally associated reasoning tasks are also encodable in DL-PA (Sect. 7). We close the paper by discussing related work and pointing out paths for future research in Sect. 8. Most of the proofs are merely sketched for space reasons, details are left to the reader.

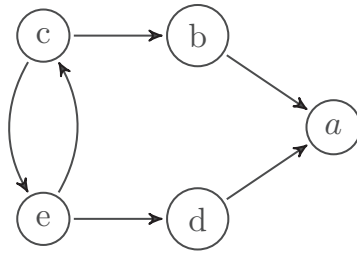
2 Background

General Notation for Sets of Arguments and Attack Relations. We assume a finite, non-empty set of arguments \mathcal{U} (the *universe*) as fixed from now on. We moreover assume that \mathcal{U} is big enough to accommodate our examples. Sets of arguments (denoted A , sometimes with a superscript) are supposed to be subsets of \mathcal{U} ; and all conflict relations (denoted R , sometimes with a superscript) are supposed to be defined over \mathcal{U} (i.e., $R \subseteq \mathcal{U} \times \mathcal{U}$). Given $A \subseteq \mathcal{U}$ and $R \subseteq \mathcal{U} \times \mathcal{U}$, we use $R|_A$ to abbreviate $R \cap (A \times A)$ (the restriction of R to A).

2.1 Abstract Argumentation Frameworks and Their Stable Semantics

An **argumentation framework** (AF) is a directed graph (A, R) [23], where A stands for a set of arguments and R stands for a conflict-based relation among them (typically, an attack relation).¹ Argumentation semantics are meant to capture the informal notion of a reasonable position in a debate (i.e. in an AF). There is a large number of available semantics studied in the literature (see [5]). For the sake of presentation we stick to stable semantics, but our approach can be straightforwardly extended to the rest of standard semantics defined by Dung [23], namely admissible, complete, grounded and preferred: it suffices to combine our results with those of [20]. A set of arguments $E \subseteq A$ is a **stable extension** if (i) $(E \times E) \cap R = \emptyset$ (E is conflict-free') and (ii) $x \in A \setminus E$ implies that there is a $y \in E$ such that $(y, x) \in R$ (E attacks every argument outside itself'). We note $\text{st}(A, R)$ the set of all stable extensions of (A, R) . An argument $x \in E$ is said to be credulously (resp. sceptically) **accepted** if it belongs to at least one (resp. every) extension.

As an example, for the AF (A_0, R_0) represented in the picture below we have $\text{st}(A_0, R_0) = \{\{b, e\}, \{c, d\}\}$.



2.2 Dynamic Logic of Propositional Assignments (DL-PA)

We shall use DL-PA as a the general logical framework of this paper. We start by associating several kinds of propositional variables to arguments. To every

¹ As $A \subseteq \mathcal{U}$, we actually focus on *finite* AFs, as most of the literature does. This is an essential limitation of our approach, as our encodings use formulas parametrised by \mathcal{U} , which makes finiteness of \mathcal{U} necessary. Capturing some argumentation semantics for the general case has been shown to require powerful logical languages, such as modal μ -calculus for the grounded semantics [25].

set of arguments $A \subseteq \mathcal{U}$ we associate the set of awareness variables $\text{AW}_A = \{\text{aw}_x \mid x \in A\}$, and the set of acceptance variables $\text{IN}_A = \{\text{in}_x \mid x \in A\}$. Furthermore, to every relation $R \subseteq \mathcal{U} \times \mathcal{U}$ we associate the set of attack variables $\text{ATT}_R = \{r_{x,y} \mid (x,y) \in R\}$. The set of propositional variables of our logic is then

$$\begin{aligned} \text{Prp}_{\mathcal{U}} &= \text{AW}_{\mathcal{U}} \cup \text{IN}_{\mathcal{U}} \cup \text{ATT}_{\mathcal{U} \times \mathcal{U}} \\ &= \{\text{aw}_x \mid x \in \mathcal{U}\} \cup \{\text{in}_x \mid x \in \mathcal{U}\} \cup \{r_{x,y} \mid (x,y) \in \mathcal{U}^2\}. \end{aligned}$$

Then formulas and programs of DL-PA are defined by mutual recursion:

$$\begin{aligned} \varphi &::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid [\pi]\varphi \\ \pi &::= +p \mid -p \mid \varphi? \mid (\pi; \pi) \mid (\pi \cup \pi) \mid \pi^\sim \end{aligned}$$

where p ranges over $\text{Prp}_{\mathcal{U}}$. The formula $[\pi]\varphi$ reads “ φ is true after every possible execution of π ”. The program $+p$ makes p true and $-p$ makes p false. The program $\varphi?$ tests that φ is true (and fails when it is false). The program $\pi_1; \pi_2$ is the sequential composition of π_1 and π_2 , and $\pi_1 \cup \pi_2$ is their nondeterministic composition. Finally, π^\sim is the execution of π ‘the other way round’. As usual, skip abbreviates the program \top .

Formulas of DL-PA are interpreted over classical propositional valuations, i.e., subsets of $\text{Prp}_{\mathcal{U}}$. Programs are interpreted as binary relations on the set of all valuations. We use v, v', v'' to denote valuations. Again by mutual recursion, the interpretation of modal formulas is:

$$v \models [\pi]\varphi \text{ if } (v, v') \in \|\pi\| \text{ implies } v' \models \varphi,$$

and the interpretation of programs $\|\pi\| \subseteq 2^{\text{Prp}_{\mathcal{U}}} \times 2^{\text{Prp}_{\mathcal{U}}}$ is:

$$\begin{aligned} \|\!+p\|\! &= \{(v, v') \mid v' = v \cup \{p\}\} \\ \|\!-p\|\! &= \{(v, v') \mid v' = v \setminus \{p\}\} \\ \|\!\varphi?\|\! &= \{(v, v) \mid v \models \varphi\} \\ \|\!\pi; \pi'\|\! &= \|\pi\| \circ \|\pi'\| \\ \|\!\pi \cup \pi'\|\! &= \|\pi\| \cup \|\pi'\| \\ \|\!\pi^\sim\|\! &= \|\pi\|^{-1} \end{aligned}$$

A formula φ is DL-PA satisfiable if $v \models \varphi$ for some v , and it is valid if $v \models \varphi$ for every v . It is known that satisfiability, validity, and model-checking are all PSPACE complete decision problems [3].

From Valuations to AFs and Backward. Each propositional valuation $v \subseteq \text{Prp}_{\mathcal{U}}$ represents an AF (A_v, R_v) , where $A_v = \{x \in \mathcal{U} \mid \text{aw}_x \in v\}$ and $R_v = \{(x, y) \in \mathcal{U}^2 \mid r_{x,y} \in v\}_{|A_v}$. The other way round, each AF (A, R) can be represented as a propositional valuation $v_{(A,R)} = \{\text{aw}_x \mid x \in A\} \cup \{r_{x,y} \mid (x,y) \in R\}$. Note that if we start with a valuation v' we have that $v_{(A_{v'}, R_{v'})} = v'$ does not generally hold (because a valuation can contain an attack variable $r_{a,b}$ with neither aw_a nor aw_b being members of it). If we, however, start with an AF (A', R') we have that $(A_{v_{(A',R')}}, R_{v_{(A',R')}}) = (A', R')$ is always the case.

3 Formalisms for Arguing with Qualitative Uncertainty

We now review three formalisms for representing qualitative uncertainty about abstract argumentation frameworks. We start by presenting *control argumentation frameworks* (CAFs) [18], which besides uncertainty, also include a dynamic component. After that, we introduce *rich incomplete argumentation frameworks* (rIAFs) [29] and *incomplete argumentation frameworks* (IAFs) [8] as special cases of CAFs.

A **control argumentation framework** is a triple $\text{CAF} = (F, C, U)$ where:

- $F = (A^F, R^F)$ is the *fixed part*, with $R^F \subseteq (A^F \cup A^?) \times (A^F \cup A^?)$, and both A^F and $A^?$ being two finite sets of arguments;
- $U = (A^?, (R^? \cup R^{\leftrightarrow}))$ is the *uncertain part*, where

$$R^?, R^{\leftrightarrow} \subseteq (A^F \cup A^?) \times (A^F \cup A^?)$$

and R^{\leftrightarrow} is symmetric and irreflexive;²

- $C = (A^C, R^C)$ is the *control part* where A^C is yet another finite set of arguments and

$$R^C \subseteq (A^C \times (A^F \cup A^? \cup A^C)) \cup ((A^F \cup A^? \cup A^C) \times A^C);$$

- $A^F, A^?,$ and A^C are pairwise disjoint; and
- $R^F, R^?, R^{\leftrightarrow},$ and R^C are pairwise disjoint.

Standard AFs can be viewed as CAFs with empty uncertain and control parts: CAFs where $A^?, R^?, R^{\leftrightarrow},$ and A^C are empty (and therefore R^C is empty by definition too).

Given a $\text{CAF} = (F, C, U)$, a **control configuration** is a subset of control arguments $\text{CFG} \subseteq A^C$. The *CAF associated to CFG* is $\text{CAF}_{\text{CFG}} = (F, C_{\text{CFG}}, U)$ where $C_{\text{CFG}} = (\text{CFG}, R^C \upharpoonright_{A^F \cup A^? \cup A_{\text{CFG}}})$.

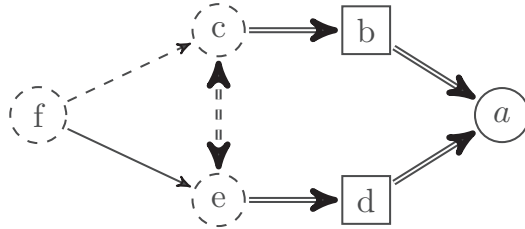
Epistemic Interpretation of CAFs. In order to throw some intuition, let us briefly recall the epistemic interpretation of CAFs provided in [32]. A CAF can be thought as modelling an agent (the proponent) who is trying to convince another agent (the opponent) to accept certain argument(s). Under this interpretation, F represents the arguments and attacks that the proponent knows the opponent knows. U represents the argument and attacks such that the proponent is not sure about how the opponent perceives them. In particular, R^{\leftrightarrow} is an conflict relation such that the proponent knows that the opponent knows that these attacks hold, but the direction of the attack according to the opponent's perception is unknown to the proponent. This makes perfect sense if we understand conflict relations as defeat relations (as done in the field of structured argumentation [14]). In this picture, the proponent can be sure about the opponent's perception of at least one of the attacks between a and b (for instance, because

² Symmetry and irreflexivity of R^{\leftrightarrow} are not assumed in the original paper [18], but as pointed out by [30,31], both assumptions can be made without loss of generality.

they have contradictory conclusions), but the proponent still lacks information about the opponent's knowledge so as to know how he (the opponent) perceives the relative strength of a and b , and hence the direction of the defeat. As for $C = (A^C, R^C)$, it is supposed to be the part of the framework that depends on the actions of the proponent. More precisely, it can be interpreted as private, communicable knowledge of the proponent, i.e., the arguments and attacks such that (i) they are known to the proponent, (ii) they are unknown to the opponent (and the proponent knows this). Moreover, CAFs make a strong assumption about control arguments: (iii) the proponent is completely sure about the effects of communicating each of them.

Example 1. With the above interpretation in mind, consider the CAF $\text{CAF}_0 = (F_0, C_0, U_0)$ where $A_0^F = \{a\}$, $R_0^F = \{(f, e)\}$, $A_0^U = \{c, e, f\}$, $R^U = \{(f, c)\}$, $R^{\leftrightarrow} = \{(c, e), (e, c)\}$, $A_0^C = \{b, d\}$ and $R_0^C = \{(b, a), (d, a), (c, b), (e, d)\}$.

We represent CAF_0 graphically as follows:



where solid circles stand for fixed arguments, normal arrows stand for fixed attacks, dashed circles stand for uncertain arguments, dashed arrows stand for uncertain attacks, dashed double arrows stand for symmetric attacks, squares stand for control arguments, and double arrows for control attacks.

A fundamental notion for reasoning about CAFs (and the rest of formalisms for qualitative uncertainty that will be studied here) is that of completion.

A **completion** of $\text{CAF} = (F, C, U)$ is any AF (A^*, R^*) such that:

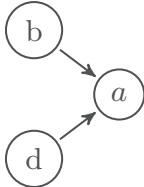
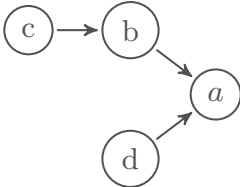
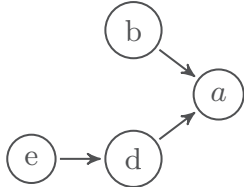
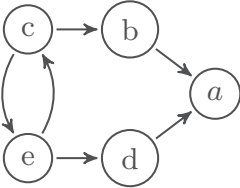
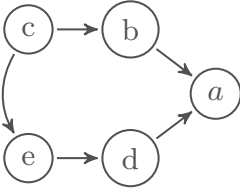
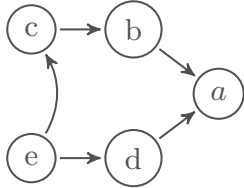
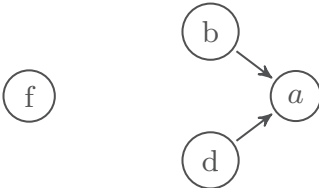
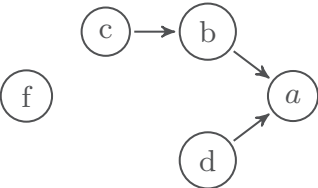
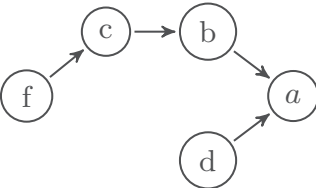
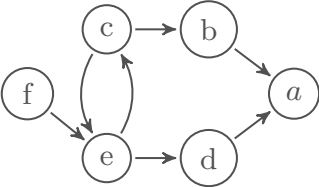
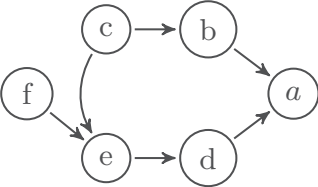
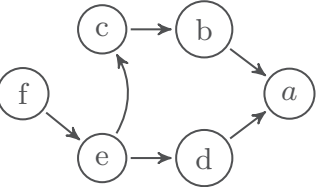
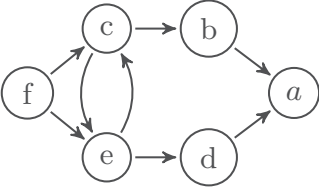
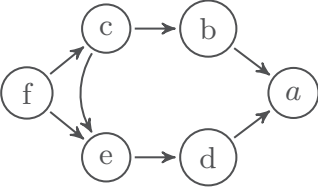
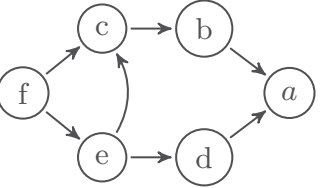
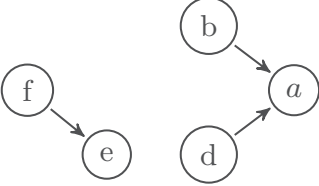
- $(A^F \cup A^C) \subseteq A^* \subseteq (A^F \cup A^C \cup A^?)$;
- $(R^F \cup R^C)_{|A^*} \subseteq R^* \subseteq (R^F \cup R^C \cup R^? \cup R^{\leftrightarrow})_{|A^*}$; and
- for every x, y : $(x, y) \in R^{\leftrightarrow}$ and $x, y \in A^*$ implies $(x, y) \in R^*$ or $(y, x) \in R^*$.

A completion can be seen as a provisional removal of uncertainty or, in epistemic terms, as a possible world (cf. [27, 32]). This removal lets the proponent reason under the assumption that the opponent's AF is such-and-such. If we identify standard AFs (A, R) with CAFs with empty uncertain and control parts then (A, R) is the unique completion of itself.

The completions of CAF_0 are depicted in Table 1.

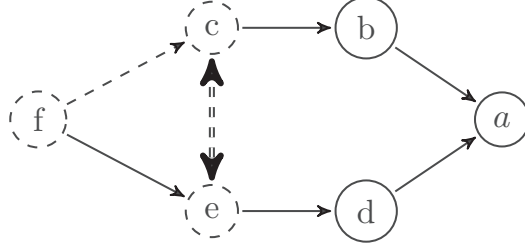
A **rich incomplete AF** [29] is a pair $\text{rIAF} = (F, U)$ where F and U are exactly as in a CAF. A rich incomplete AF can be informally understood as a CAF with empty A^C and R^C , i.e., where we abstract away from the dynamics. We sometimes unravel F and U and represent rich incomplete AFs as tuples of the form $(A^F, A^?, R^F, R^?, R^{\leftrightarrow})$. The notion of completion is easily adapted to rIAFs.

Table 1. Completions of CAF_0 . The column [1, 2, ..., 6] and the row [A, B, C] are just included for numbering purposes. Empty cells do *not* represent the empty completion (\emptyset, \emptyset) .

	A	B	C
1			
2			
3			
4			
5			
6			

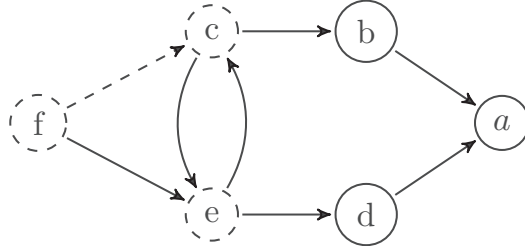
Example 2. Let $\text{rIAF}_0 = (A_0^F, A_0^U, R_0^F, R_0^U, R_0^{\leftrightarrow})$ where $A_0^F = \{a, b, d\}$, $A_0^U = \{c, e, f\}$, $R_0^F = \{(b, a), (d, a), (c, b), (e, d), (f, e)\}$, $R_0^U = \{(f, c)\}$, and $R_0^{\leftrightarrow} = \{(c, e), (e, c)\}$. Note that rIAF_0 has exactly the same set of completions as CAF_0 (from Example 1), that is, all those depicted in Table 1. Actually rIAF_0 can be seen as the full development of CAF_0 , i.e., CAF_0 where all the control arguments have been used by the proponent.

We represent rIAF_0 graphically as follows:



An **incomplete AF** [8] (IAF), is a rich IAF with empty R^{\leftrightarrow} . We represent IAFs as tuples of the form $(A, A^?, R, R^?)$. There are some notable subclasses of IAFs, well-studied in the literature, namely **attack-incomplete AFs** (att-IAFs, for short), which are IAFs with empty $A^?$; and **argument-incomplete AFs** (arg-IAFs, for short), which are IAFs with empty $R^?$. The notion of completion is again straightforwardly relativised to IAFs.

Example 3. Let us consider $\text{IAF}_0 = (A_0^F, A_0^U, R_0^F, R_0^U)$, where $A_0^F = \{a, b, d\}$, $A_0^U = \{c, e, f\}$, $R_0^F = \{(b, a), (d, a), (c, b), (e, d), (c, e), (e, c), (f, e)\}$ and $R_0^U = \{(f, c)\}$, graphically represented below. The set of completions of IAF_0 is the one depicted in Table 1 except for the cells **B2**, **C2**, **B4**, **C4**, **B5** and **C5** (as the symmetric attack $c \leftrightarrow e$ is now fixed).



Given a control AF, CAF, we note $\text{completions}(\text{CAF})$ its set of completions (and we do the same for IAFs and rIAFs).

Classic reasoning tasks such as extension enumeration or argument acceptance have been generalized from AFs to both IAFs and rIAFs. As an example, let us consider the following one:

stable-Necessary-Credulous-Acceptance (st-NCA)
Given: A rich IAF $\text{rIAF} = (A^F, A^?, R^F, R^?, R^{\leftrightarrow})$ and an argument $a \in A^F$.
Question: Is it true that for every $(A^*, R^*) \in \text{completions}(\text{rIAF})$ there is an $E \in \text{st}(A^*, R^*)$ such that $a \in E$?

We can replace **st** by any other semantics as well as switch quantifiers in the definition above in order to obtain different variants of the problem.

Regarding CAFs, defining relevant reasoning tasks gets slightly more complicated, since we have to take into account their dynamic aspect (the control part). In this context, a natural reasoning task is finding a control configuration (that is, a set of control arguments) such that a certain argument gets accepted by the opponent after the latter learns about them. Just as before, acceptability is then relativised to quantification over completions and extensions. As an example, let us consider:

<p>stable-Necessary-Sceptical-Controllability (st-NSCon)</p> <p>Given: A control argumentation framework $\text{CAF} = (F, C, U)$ and an argument $a \in A^F$.</p> <p>Question: Is it true that there is a configuration $\text{CFG} \subseteq A_C$ such that for every completion (A^*, R^*) of CAF_{CFG} and for every $E \in \text{st}(A^*, R^*), a \in E$?</p>
--

Expressivity via Sets of Completions. Following [29], we can compare the modelling power of each of the previous formalisms for arguing with uncertainty (IAFs, rIAFs and possibly others) using the sets of completions that they can represent. Let $(\text{att-}, \text{arg-})\mathcal{IAF}$ (resp. $\mathcal{RIAF}, \mathcal{CAF}$) denote the class of all (att-, arg-)IAFs (resp. rIAFs, CAFs), and let \mathcal{X} and \mathcal{Y} be metavariables denoting arbitrary classes of the previous list. We say that \mathcal{X} is **at least as expressive as** \mathcal{Y} (in symbols, $\mathcal{X} \succeq \mathcal{Y}$) if, for every $Y \in \mathcal{Y}$ there is a $X \in \mathcal{X}$ such that $\text{completions}(X) = \text{completions}(Y)$. We use \succ to denote the strict counterpart of \succeq , we use \preceq to denote the inverse of \succeq , and we use \equiv to abbreviate $\succeq \cap \preceq$. For instance, in [29], it was proved that $\mathcal{RIAF} \succ \mathcal{IAF}$.

4 Incomplete AFs in DL-PA

Our first aim is to capture incomplete AFs using DL-PA. More precisely, given an incomplete argumentation framework $\text{IAF} = (A, A^?, R, R^?)$, we want to design a program $\text{makeComp}^{\text{IAF}}$ such that every valuation that is $\text{makeComp}^{\text{IAF}}$ -accessible from the valuation v_{IAF} associated to IAF represents a completion of IAF and, vice versa, every completion of IAF is represented by at least one $\text{makeComp}^{\text{IAF}}$ -successor of v_{IAF} .

First of all, we **associate to IAF its valuation**

$$\begin{aligned}
 v_{\text{IAF}} &= v_{(A^F, R^F)} \\
 &= \text{AW}_{A^F} \cup \text{ATT}_{R^F} \\
 &= \{\text{aw}_x \mid x \in A^F\} \cup \{\text{r}_{x,y} \mid (x, y) \in R^F\}.
 \end{aligned}$$

Note that $(A_{v_{\text{IAF}}}, R_{v_{\text{IAF}}})$ is already a completion of IAF: it is the smallest one, where only fixed arguments and fixed attacks between them are considered. What we need to do in order to compute all the completions of IAF is varying the value of propositional variables representing arguments in $A^?$ and attacks in $R^?$. Let

us first define the DL-PA program that computes all possible combinations of variables in a given set. Let $P = \{p_1, \dots, p_n\}$ be a subset of $\text{Prp}_{\mathcal{U}}$ and define

$$\text{vary}(P) = (+p_1 \cup -p_1); \dots; (+p_n \cup -p_n).$$

(Note that the order of the propositional variables does not matter.) With this abbreviation at hand we are able to define the program to compute the completions we are after:

$$\text{makeComp}^{\text{IAF}} = \text{vary}(\text{AW}_{A^?}); \text{vary}(\text{ATT}_{R^?}).$$

The next proposition shows that our original target is reached.

Proposition 1. *Let $\text{IAF} = (A, A^?, R, R^?)$, then*

- *If $(v_{\text{IAF}}, v) \in \|\text{makeComp}^{\text{IAF}}\|$, then $(A_v, R_v) \in \text{completions}(\text{IAF})$.*
- *If $(A^*, R^*) \in \text{completions}(\text{IAF})$, then $(v_{\text{IAF}}, v_{(A^*, R^*)}) \in \|\text{makeComp}^{\text{IAF}}\|$.*

Proof. For the first item, suppose $(v_{\text{IAF}}, v) \in \|\text{makeComp}^{\text{IAF}}\|$. We recall that $\|\text{vary}(P)\| = \{(v', v'') \mid (v' \setminus v'') \cup (v'' \setminus v') \subseteq P\}$ for any set of atoms P [21].³ Hence, by the semantics of the sequential composition operator $;$, we have that $(v_{\text{IAF}}, v) \in \|\text{makeComp}^{\text{IAF}}\|$ amounts to saying that the set of variables whose truth values differs from v_{IAF} to v , formally the set $(v \setminus v_{\text{IAF}}) \cup (v_{\text{IAF}} \setminus v)$, must be a subset of $\text{AW}_{A^?} \cup \text{ATT}_{R^?}$. But, since all variables from $\text{AW}_{A^?} \cup \text{ATT}_{R^?}$ are false in v_{IAF} by definition, we have that $v = v_{\text{IAF}} \cup P$ for some $P \subseteq \text{AW}_{A^?} \cup \text{ATT}_{R^?}$. From this statement, and applying the definition of (A_v, R_v) and the one of completion, we obtain that $(A_v, R_v) \in \text{completions}(\text{IAF})$.

For the second item, suppose that $(A^*, R^*) \in \text{completions}(\text{IAF})$, which amounts to $A^F \subseteq A^* \subseteq A^F \cup A^?$ and $R_{|A^*}^F \subseteq R^* \subseteq (R^F \cup R^?)_{|A^*}$. Now, remember that $v_{(A^*, R^*)} = \text{AW}_{A^*} \cup \text{ATT}_{R^*}$. From the two previous statements and the definition of v_{IAF} , we can deduce that the set of variables whose truth values differ from v to $v_{(A^*, R^*)}$ must be a subset of $\text{AW}_{A^?} \cup \text{ATT}_{R^?}$, which, as argued before, amounts to saying that $(v_{\text{IAF}}, v_{(A^*, R^*)}) \in \|\text{makeComp}^{\text{IAF}}\|$.

Using this result together with the general technique to compute extensions provided in [20, 22], we can reduce reasoning problems in IAFs to model-checking problems in DL-PA. Note that we need an encoding of argumentation semantics that takes into account our “awareness” variables. As shown in [22], the following propositional schema characterizes the stable semantics in awareness-relativised AFs:

$$\text{Stable} = \bigwedge_{x \in \mathcal{U}} \left((\text{in}_x \rightarrow \text{aw}_x) \wedge \left(\text{aw}_x \rightarrow (\text{in}_x \leftrightarrow \neg \bigvee_{y \in \mathcal{U}} (\text{in}_y \wedge r_{y,x} \wedge \text{aw}_x \wedge \text{aw}_y)) \right) \right).$$

³ Note that vary is noted flipSome in [21].

The authors show that v is a model of **Stable** if and only if $\{x \in \mathcal{U} \mid \text{in}_x \in v\}$ is a stable extension of (A_v, R_v) . Note that **Stable** is actually parametrised by \mathcal{U} , but we drop it to simplify notation.

The program $\text{makeExt}^{\text{st}} = \text{vary}(\text{IN}_{\mathcal{U}}); \text{Stable?}$ nondeterministically builds all possible stable extensions by first varying the values of the ‘in’ variables and then checking that a stable valuation has been obtained [20, 22]. Our general technique is then illustrated by the following result.

Proposition 2. *Let $\text{IAF} = (A, A^?, R, R^?)$, and $a \in A^F$, then the answer to **st-NCA** with input IAF and a is yes iff $v_{\text{IAF}} \models [\text{makeComp}^{\text{IAF}}] \langle \text{makeExt}^{\text{st}} \rangle \text{in}_a$.*

Proof (Sketched). The result follows from the definition of the reasoning task, the correctness of $\text{makeExt}^{\text{st}}$ ([20, 21]), Proposition 1, and the semantics of DL-PA.

5 Rich Incomplete AFs in DL-PA

Things get slightly more complicated when computing the completions of a rich IAF in DL-PA, since the program vary does not suffice to compute the symmetric attacks of R^{\leftrightarrow} . We can, however, find a specific program for this purpose.

First of all, given $\text{rIAF} = (A^F, A^?, R^F, R^?, R^{\leftrightarrow})$, we **associate to rIAF its valuation**

$$\begin{aligned} v_{\text{rIAF}} &= v_{(A^F, R^F)} \\ &= \text{AW}_{A^F} \cup \text{ATT}_{R^F} \\ &= \{\text{aw}_x \mid x \in A^F\} \cup \{\text{r}_{x,y} \mid (x,y) \in R^F\}. \end{aligned}$$

Note that, contrarily to what happened with IAFs, $(A_{v_{\text{rIAF}}}, R_{v_{\text{rIAF}}})$ is *not* always a completion of rIAF (this is false as soon as $R^{\leftrightarrow} \cap (A^F \times A^F)$ is not empty). Let us now define the program that will be used to compute the presence of elements of R^{\leftrightarrow} in each completion. Let $\text{ATT}_R = \{\text{r}_{x_1, y_1}, \dots, \text{r}_{x_n, y_n}\}$ be a set of attack variables, and define

$$\text{dis}(\text{ATT}_R) = (+\text{r}_{x_1, y_1} \cup +\text{r}_{y_1, x_1}); \dots; (+\text{r}_{x_n, y_n} \cup +\text{r}_{y_n, x_n}).$$

Intuitively, dis makes true at least one of the pairs from the set $\{(x_i, y_i), (y_i, x_i)\}$, for each $1 \leq i \leq n$. Moreover, when applied to symmetric relations, dis makes true either (x_i, y_i) , or (y_i, x_i) , or both. We have now the tools to define the program makeComp in its version for rIAFs. Let $\text{rIAF} = (A^F, A^?, R^F, R^?, R^{\leftrightarrow})$, and define

$$\text{makeComp}^{\text{rIAF}} = \text{vary}(\text{AW}_{A^?}); \text{vary}(\text{ATT}_{R^?}); \text{dis}(\text{ATT}_{R^{\leftrightarrow}}).$$

The following proposition shows that the above program is correct.

Proposition 3. *Let $\text{rIAF} = (A^F, A^?, R^F, R^?, R^{\leftrightarrow})$, then:*

- *If $(v_{\text{rIAF}}, v) \in \|\text{makeComp}^{\text{rIAF}}\|$, then $(A_v, R_v) \in \text{completions}(\text{rIAF})$.*

– If $(A^*, R^*) \in \text{completions}(\text{rIAF})$, then $(v_{\text{rIAF}}, v_{(A^*, R^*)}) \in \|\text{makeComp}^{\text{rIAF}}\|$.

Proof (Sketched). The proof is analogous to the one of Proposition 1, but taking into account the observation that, when applied to the *symmetric* relation $R^{\leftrightarrow} = \{(x_1, y_1), (y_1, x_1), \dots, (x_n, y_n), (y_n, x_n)\}$, every execution of $\text{dis}(\text{ATT}_{R^{\leftrightarrow}})$ makes true either r_{x_i, y_i} , or r_{y_i, x_i} or both, for every $1 \leq i \leq n$.

Again, acceptance problems can be reduced to model-checking problems. As an example, consider the following reduction, where **st-PSA** stands for *stable-Possible-Sceptical Acceptance*.

Proposition 4. *Let $\text{rIAF} = (A^F, A^?, R^F, R^?, R^{\leftrightarrow})$, and let $a \in A^F$, then the answer to **st-PSA** with input rIAF and a is yes iff*

$$v_{\text{rIAF}} \models \langle \text{makeComp}^{\text{rIAF}} \rangle [\text{makeExt}^{\text{st}}] \text{in}_a.$$

Proof (Sketched). The result follows from the definition of the reasoning problem, the correctness of $\text{makeExt}^{\text{st}}$ [20, 21], Proposition 3, and the semantics of DL-PA.

6 Control AFs in DL-PA

We now move to control argumentation frameworks. Regarding uncertainty, control argumentation frameworks are essentially rich incomplete argumentation frameworks; however, the delicate part is their dynamic component, i.e., the control part.

First, given a CAF $\text{CAF} = (F, C, U)$, we define its **associated valuation** as

$$\begin{aligned} v_{\text{CAF}} &= v_{(A^F, R^F \cup R^C)} \\ &= \text{AW}_{A^F} \cup \text{ATT}_{R^F} \cup \text{ATT}_{R^C} \\ &= \{\text{aw}_x \mid x \in A^F\} \cup \{r_{x,y} \mid (x,y) \in R^F\} \cup \{r_{x,y} \mid (x,y) \in R^C\}. \end{aligned}$$

Note that v_{CAF} contains all attack variables corresponding to control attacks, but none of them appear in $(A_{v_{\text{CAF}}}, R_{v_{\text{CAF}}})$, since none of the control arguments has been communicated yet. This highlights the fact that in the epistemic interpretation of CAFs, the proponent knows how the opponent will perceive the attack relations regarding all communicable arguments (a point that might be subject to criticism).

To capture the dynamic component of CAF we define the following program:

$$\text{control}^{\text{CAF}} = \text{vary}(\text{AW}_{A^C}).$$

Intuitively, $\text{control}^{\text{CAF}}$ nondeterministically chooses some of the possible control configurations of CAF (i.e., some subset of control arguments).

What about completions? As mentioned, if we restrict to uncertainty, CAFs are essentially rIAFs [29]. Hence, once we have computed some control configuration, it suffices to use the same program as for rIAFs:

$$\text{makeComp}^{\text{CAF}} = \text{vary}(\text{AW}_{A^?}); \text{vary}(\text{ATT}_{R^?}); \text{dis}(\text{ATT}_{R^{\leftrightarrow}}).$$

We again state a correctness result:

Proposition 5. Let $\text{CAF} = (F, C, U)$.

- If $(v_{\text{CAF}}, v) \in \|\text{control}^{\text{CAF}}; \text{makeComp}^{\text{CAF}}\|$, then there is a control configuration $\text{CFG} \subseteq A^C$ and a completion (A^*, R^*) of CAF_{CFG} such that $(A_v, R_v) = (A^*, R^*)$.
- For every control configuration $\text{CFG} \subseteq A^C$ and every $(A^*, R^*) \in \text{completions}(\text{CAF}_{\text{CFG}})$ there is a valuation $v \in 2^{\text{Prp}_U}$ such that $(v_{\text{CAF}}, v) \in \|\text{control}^{\text{CAF}}; \text{makeComp}^{\text{CAF}}\|$ and $(A_v, R_v) = (A^*, R^*)$.

Proof (Sketched). The proof is analogous to those of Propositions 1 and 3. The essential difference lies in the fact that the previous execution of $\text{control}^{\text{CAF}}$ is needed to nondeterministically choose a control configuration of CAF. Also, note that $\text{ATT}_{RC} \subseteq v_{\text{CAF}}$ is essential for obtaining the needed control attacks in the corresponding completion.

We can then combine the previous programs with makeExt in order to reduce controllability problems to model-checking problems in DL-PA. As an example, consider the following.

Proposition 6. Let $\text{CAF} = (F, C, U)$ and $a \in A^F$. The answer to st-NSCon with input CAF and a is yes if and only if

$$v_{\text{CAF}} \models \langle \text{control}^{\text{CAF}} \rangle [\text{makeComp}^{\text{CAF}}; \text{makeExt}^{\text{st}}] \text{in}_a.$$

Proof (Sketched). The result follows from the definition of the reasoning task, the correctness of $\text{makeExt}^{\text{st}}$ [20, 21], Proposition 5, and the semantics of DL-PA.

7 Constrained Incomplete AFs and Their Encoding in DL-PA

We now move to study a very general class of structures for modelling qualitative uncertainty about AFs: *constrained incomplete AFs*. To the best of our knowledge, these structures have not been studied before in the literature. They are however inspired by the notion of *constrained AF* [17], and by the encoding of other structures in DL-PA as undertaken in this paper.

Let \mathcal{U} be given, a **constrained incomplete AF** (cIAF) is a pair $\text{cIAF} = (\mathcal{U}, \varphi)$ where φ is a Boolean formula built over the set of propositional variables $\text{AW}_{\mathcal{U}} \cup \text{ATT}_{\mathcal{U} \times \mathcal{U}}$. The set of **completions** of a given cIAF is defined as

$$\text{completions}(\mathcal{U}, \varphi) = \{(A_v, R_v) \mid v \subseteq \text{Prp}_{\mathcal{U}} \text{ and } v \models \varphi\}.$$

We note $c\text{-IAF}$ the class of all constrained incomplete argumentations frameworks.

Example 4. Let us consider $\text{cIAF}_1 = (\mathcal{U}, \varphi)$ with $\mathcal{U} = \{a, b\}$ and $\varphi = (\text{aw}_a \wedge \text{aw}_b) \wedge (r_{a,b} \vee r_{b,a}) \wedge \neg(r_{a,b} \wedge r_{b,a}) \wedge \neg r_{a,a} \wedge \neg r_{b,b}$. The completions of cIAF_1 are:

$$a \bullet \longrightarrow \bullet b \qquad a \bullet \longleftarrow \bullet b$$

Proposition 7. *cIAFs are strictly more expressive than IAFs and rIAFs. In other words, for every (r)IAF, there is a cIAF with the same set of completions; but there is a cIAF such that no (r)IAF has the same set of completions.*

Proof. We only have to prove $c\text{-IAF} \succ \mathcal{R}\text{IAF}$ (as $c\text{-IAF} \succ \text{IAF}$ follows from $\mathcal{R}\text{IAF} \succ \text{IAF}$ [29] and the transitivity of \succ).

For the first part of the statement (for every rIAF there is a cIAF with the same set of completions), let rIAF be a rIAF with $\text{completions}(\text{rIAF}) = \{(A_1^*, R_1^*), \dots, (A_n^*, R_n^*)\}$. Note that, for any AF (A, R) defined over \mathcal{U} , we can write its *theory* (see e.g. [22]), that is, the formula

$$\text{Th}(A, R) = \bigwedge_{x \in A} \text{aw}_x \wedge \bigwedge_{x \in \mathcal{U} \setminus A} \neg \text{aw}_x \wedge \bigwedge_{(x,y) \in R} r_{x,y} \wedge \bigwedge_{(x,y) \in \mathcal{U}^2 \setminus R} \neg r_{x,y}.$$

It is then easy to show that for any valuation $v \subseteq \text{Prp}_{\mathcal{U}}$, we have that $v \models \text{Th}(A, R)$ iff $(A_v, R_v) = (A, R)$. Now, letting $\rho = \bigvee_{1 \leq i \leq n} \text{Th}(A_i^*, R_i^*)$, we have that

$$\text{completions}(\mathcal{U}, \rho) = \text{completions}(\text{rIAF}).$$

In order to prove the second half of the proposition, it suffices to use the cIAF of Example 4 (called cIAF_0). Reasoning towards contradiction, suppose that there is a rIAF $\text{rIAF} = (A^F, A^?, R^F, R^?, R^{\leftrightarrow})$ with the same set of completions as cIAF_0 . Then $(a, b) \in R^F \cup R^? \cup R^{\leftrightarrow}$ (since (a, b) appears in a completion of rIAF). We show that the last statement is absurd. If $(a, b) \in R^F$, then (a, b) should appear in all completions of rIAF where a and b are present, but this is not true. If $(a, b) \in R^?$, we reason by cases on $(b, a) \in R^F \cup R^? \cup R^{\leftrightarrow}$: the first one is impossible, since (b, a) would be in every completion where a and b appear, and that is not the case; the second one is absurd because we would have an extension with neither (a, b) nor (b, a) , and this is not the case; the third one is impossible because it would imply $(a, b) \in R^{\leftrightarrow}$, but we have assumed that $(a, b) \in R^?$, and we know that $R^? \cap R^{\leftrightarrow} = \emptyset$ by definition. Finally, suppose that $(a, b) \in R^{\leftrightarrow}$, which implies $(b, a) \in R^{\leftrightarrow}$ (by symmetry of R^{\leftrightarrow}), which is impossible because we would have a completion with both (a, b) and (b, a) , but this is not the case.

Note that in the first part of the proof we have used an argument that works for *any* set of directed graphs with domain \mathcal{U} (and not only for the completions of a given rIAF), hence we can state that:

Corollary 1. *For any set S of directed graphs with domain \mathcal{U} , there is a cIAF cIAF such that $S = \text{completions}(\text{cIAF})$.*

In words, cIAFs are a maximally expressive formalism for representing qualitative uncertainty about AFs. Figure 1 depicts the relative expressivity of the different formalisms adding qualitative uncertainty to abstract argumentation that we have discussed in this paper.

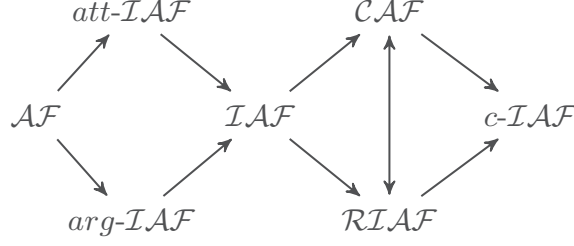


Fig. 1. Relative expressivity of formalisms for qualitative uncertainty in formal argumentation. An arrow from \mathcal{X} to \mathcal{Y} means that $\mathcal{X} \preceq \mathcal{Y}$. Transitive and reflexive arrows are omitted.

The Need of cIAFs. Besides being mathematically interesting, one may wonder why one should use cIAFs. Our main motivation is that, while the computational complexity of reasoning tasks associated to the previously introduced formalisms ((r)IAFs and subclasses) is well-known and relatively low, their modelling power is rather limited. Consider, for instance, a proponent reasoning about the view of her opponent on a very simple debate, containing only two arguments $\{a, b\}$. Suppose that a is an argument about public health policies stated by the right-wing presidential candidate. Similarly, b is an argument stated by the left-wing candidate. Imagine that a and b have contradictory conclusions, so they are mutually incompatible. Let us informally understand R as a *defeat* relation here, that is, a relation based on logical incompatibility plus some kind of epistemic-based assessment of the involved arguments (for instance, regarding the reliability of their premisses), as it is usually done in structured argumentation. Now, suppose our proponent knows that her opponent is polarized, in the sense that he (the opponent) is already inclined towards one side of the political spectrum, but she does not know which one; then the possible AFs that the agent attributes to her opponent are exactly the completions of cIAF_1 (see Example 4). As we have shown in the proof of Proposition 7, there is no rIAF (and therefore no IAF) with that exact set of completions as cIAF_1 .

Let us now show how cIAFs can be captured using DL-PA. Let $\text{cIAF} = (\mathcal{U}, \varphi)$, and define its associated valuation simply as the empty set, that is, $v_{\text{cIAF}} = \emptyset$. (Actually any valuation over $\text{Prp}_{\mathcal{U}}$ will do the job.) The program that generates all completions of cIAF is defined as

$$\text{makeComp}^{\text{cIAF}} = \text{vary}(\text{AW}_{\mathcal{U}}); \text{vary}(\text{ATT}_{\mathcal{U} \times \mathcal{U}}); \varphi?.$$

Proposition 8. *Let $\text{cIAF} = (\mathcal{U}, \varphi)$, then:*

- *If $(v_{\text{cIAF}}, v) \in \|\text{makeComp}^{\text{cIAF}}\|$, then $(A_v, R_v) \in \text{completions}(\text{cIAF})$.*
- *If $(A^*, R^*) \in \text{completions}(\text{cIAF})$, then $(v_{\text{cIAF}}, v_{(A^*, R^*)}) \in \|\text{makeComp}^{\text{cIAF}}\|$.*

Proof (Sketched). Note that the interpretation of $\text{vary}(\text{AW}_{\mathcal{U}}); \text{vary}(\text{ATT}_{\mathcal{U} \times \mathcal{U}})$, when restricted to $2^{\text{Prp}_{\mathcal{U}} \setminus \text{IN}_{\mathcal{U}}}$, is actually the total relation $2^{\text{Prp}_{\mathcal{U}} \setminus \text{IN}_{\mathcal{U}}} \times 2^{\text{Prp}_{\mathcal{U}} \setminus \text{IN}_{\mathcal{U}}}$. Hence from $v_{\text{cIAF}} = \emptyset$ we have an execution of $\text{vary}(\text{AW}_{\mathcal{U}}); \text{vary}(\text{ATT}_{\mathcal{U} \times \mathcal{U}})$ that goes to *any* valuation in $2^{\text{Prp}_{\mathcal{U}} \setminus \text{IN}_{\mathcal{U}}}$. Then, the execution of $\varphi?$ filters those valuations of $2^{\text{Prp}_{\mathcal{U}} \setminus \text{IN}_{\mathcal{U}}}$ that satisfy the constraint of cIAF , i.e. the set of valuations

of $2^{\text{Prp}_U} \setminus \text{In}_U$ representing the set of completions of cIAF. As an illustration of the proof, consider Fig. 2.

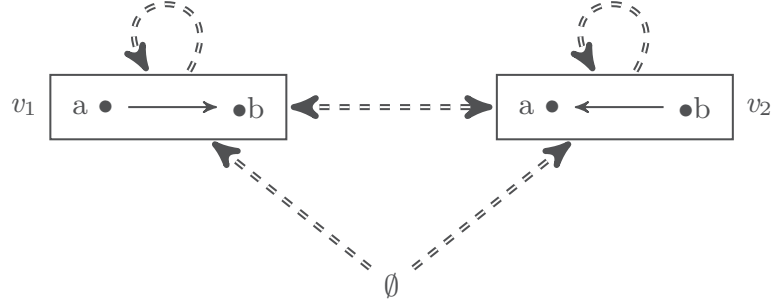


Fig. 2. Completions of cIAF_1 seen as valuations over $\text{Prp}_{\{a,b\}}$. Dashed double arrows represent the interpretation of $\text{makeComp}^{\text{cIAF}}$ (the other valuations over $\text{Prp}_{\{a,b\}}$ are omitted).

Reasoning problems for IAFs can be easily adapted to cIAFs: we just have to ensure that the argument about which we formulate the query belongs to all completions. As an example, consider:

<p>stable-Necessary-Credulous-Acceptance (st-NCA)</p> <p>Given: A constrained IAF $\text{cIAF} = (\mathcal{U}, \varphi)$ and an argument $a \in \mathcal{U}$ such that $\models \varphi \rightarrow \text{aw}_a$.</p> <p>Question: Is it true that for every $(A^*, R^*) \in \text{completions}(\text{cIAF})$ there is an $E \in \text{st}(A^*, R^*)$ such that $a \in E$?</p>
--

Note that requiring $\models \varphi \rightarrow \text{aw}_a$ amounts to requiring $a \in A$ for all $(A, R) \in \text{completions}(\mathcal{U}, \varphi)$. Once again, we can reduce acceptability problems in cIAFs to DL-PA model-checking problems. As an example, we have the following:

Proposition 9. *Let $\text{cIAF} = (\mathcal{U}, \varphi)$ and let $a \in \mathcal{U}$ such that $\models \varphi \rightarrow \text{aw}_a$, then the answer to st-PSA with input cIAF and a is yes if and only if*

$$v_{\text{cIAF}} \models \langle \text{makeComp}^{\text{cIAF}} \rangle [\text{makeExt}^{\text{st}}] \text{in}_a.$$

Proof (Sketched). The result follows from the definition of the reasoning task, the correctness of $\text{makeExt}^{\text{st}}$ ([20, 21]), Proposition 8, and the semantics of DL-PA.

8 Discussion and Future Work

Getting Closer to the Model-Checking Approach. Our encoding of formalisms for arguing with qualitative uncertainty can be qualified as *hybrid*, since it combines

some previous semantic reasoning with reasoning inside DL-PA. For instance, in order to compute the completions of an IAF, one first needs to find its associated valuation (reasoning outside the logic, using semantic objects), then has to write down the `makeComp` program, and finally reasoning in DL-PA to find the `makeComp`-successors of the associated valuation. We followed this hybrid method because we found intuitive the identification of directed graphs with propositional valuations over $\text{Prp}_{\mathcal{U}}$. However, we can adopt results from [20–22] to get a more homogeneous method here. For instance, let $\text{IAF} = (A, A^?, R, R^?)$ be an IAF, instead of computing its associated valuation, we can write down a propositional formula that characterizes its fixed elements (similarly to what is done in [20] for standard AFs):

$$\text{Th}(\text{IAF}) = \bigwedge_{x \in A^F} \text{aw}_x \wedge \bigwedge_{x \in \mathcal{U} \setminus A^F} \neg \text{aw}_x \wedge \bigwedge_{(x,y) \in R^F} r_{x,y} \wedge \bigwedge_{(x,y) \in \mathcal{U}^2 \setminus R^F} \neg r_{x,y}.$$

If we combine this formula with the `makeComp` program and the inverse operator we obtain a formula whose models completely characterize the set of completions of IAF:

$$\text{completions}(\text{IAF}) = \{(A_v, R_v) \mid v \in \|\langle (\text{Th}(\text{IAF})?; \text{makeComp}^{\text{IAF}})^{\sim} \rangle \top\|\}.$$

Comparison to QBF Encodings. As mentioned before, all we have done in DL-PA can as well be done in equally expressive logical frameworks like propositional logic or *quantified Boolean formulas* (QBF). The advantage over the former is that (1) some semantics can be expressed more compactly in DL-PA, and (2) the reasoning problems can be expressed directly as DL-PA programs. The advantage over QBFs is that the DL-PA encoding of reasoning problems by means of programs is more natural than the rather complex QBF encodings that one can find in the literature. Actually, most of the works on arguing with qualitative uncertainty use QBF encodings and algorithms for determining the complexity of associated reasoning tasks (see e.g. [8] or [31]). All advantages already pointed out by [21] of using DL-PA instead of QBF for encoding argumentative semantics are preserved by our encodings. In particular, “extension construction programs such as `makeExt` ^{σ} capture things in a more general, flexible and natural way than a QBF encoding”. This enables a straightforward extension of our results to all semantics that have been encoded in DL-PA (admissible, complete, grounded, preferred) and potentially others.

Dynamics and Uncertainty. The dynamic nature of our approach also paves the way for a systematic study of the different dynamic extensions of IAFs (e.g. in order to enforce arguments, as done in [7]) and the rest of formalisms studied here, which we leave for future work.

References

1. Amgoud, L., Vesic, S.: A new approach for preference-based argumentation frameworks. *Ann. Math. Artif. Intell.* **63**(2), 149–183 (2011). <https://doi.org/10.1007/s10472-011-9271-9>

2. Atkinson, K., et al.: Towards artificial argumentation. *AI Mag.* **38**(3), 25–36 (2017). <https://doi.org/10.1609/aimag.v38i3.2704>
3. Balbiani, P., Herzig, A., Schwarzentruher, F., Troquard, N.: DL-PA and DCL-PC: model checking and satisfiability problem are indeed in PSPACE. *CoRR* abs/1411.7825 (2014). <http://arxiv.org/abs/1411.7825>
4. Balbiani, P., Herzig, A., Troquard, N.: Dynamic logic of propositional assignments: a well-behaved variant of PDL. In: 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science, pp. 143–152. IEEE (2013). <https://doi.org/10.1109/LICS.2013.20>
5. Baroni, P., Caminada, M., Giacomin, M.: Abstract argumentation frameworks and their semantics. In: *Handbook of Formal Argumentation*, pp. 159–236. College Publications (2018)
6. Baroni, P., Cerutti, F., Giacomin, M., Guida, G.: Encompassing attacks to attacks in abstract argumentation frameworks. In: Sossai, C., Chemello, G. (eds.) *ECSQARU 2009*. LNCS (LNAI), vol. 5590, pp. 83–94. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02906-6_9
7. Baumann, R., Brewka, G.: Expanding argumentation frameworks: enforcing and monotonicity results. In: Baroni, P., Cerutti, F., Giacomin, M., Simari, G.R. (eds.) *Proceedings of the COMMA 2010*, vol. 216, pp. 75–86. IOS Press (2010). <https://doi.org/10.3233/978-1-60750-619-5-75>
8. Baumeister, D., Järvisalo, M., Neugebauer, D., Niskanen, A., Rothe, J.: Acceptance in incomplete argumentation frameworks. *Artif. Intell.* **295**, 103470 (2021). <https://doi.org/10.1016/j.artint.2021.103470>
9. Baumeister, D., Neugebauer, D., Rothe, J.: Credulous and skeptical acceptance in incomplete argumentation frameworks. In: *Proceedings of the COMMA 2018*. *Frontiers in AI and Applications*, vol. 305, pp. 181–192. IOS Press (2018). <https://doi.org/10.3233/978-1-61499-906-5-181>
10. Baumeister, D., Neugebauer, D., Rothe, J., Schadrack, H.: Complexity of verification in incomplete argumentation frameworks. In: McIlraith, S.A., Weinberger, K.Q. (eds.) *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, (AAAI 2018), pp. 1753–1760. AAAI Press (2018)
11. Baumeister, D., Neugebauer, D., Rothe, J., Schadrack, H.: Verification in incomplete argumentation frameworks. *Artif. Intell.* **264**, 1–26 (2018). <https://doi.org/10.1016/j.artint.2018.08.001>
12. Bench-Capon, T.J., Dunne, P.E.: Argumentation in artificial intelligence. *Artif. Intell.* **171**(10–15), 619–641 (2007). <https://doi.org/10.1016/j.artint.2007.05.001>
13. Besnard, P., Cayrol, C., Lagasquie-Schiex, M.C.: Logical theories and abstract argumentation: a survey of existing works. *Argument Comput.* **11**(1–2), 41–102 (2020). <https://doi.org/10.3233/AAC-190476>
14. Besnard, P., et al.: Introduction to structured argumentation. *Argument Comput.* **5**(1), 1–4 (2014). <https://doi.org/10.1080/19462166.2013.869764>
15. Caminada, M.: Rationality postulates: applying argumentation theory for non-monotonic reasoning. *J. Appl. Log.* **4**(8), 2707–2734 (2017)
16. Cayrol, C., Lagasquie-Schiex, M.C.: On the acceptability of arguments in bipolar argumentation frameworks. In: Godo, L. (ed.) *ECSQARU 2005*. LNCS (LNAI), vol. 3571, pp. 378–389. Springer, Heidelberg (2005). https://doi.org/10.1007/11518655_33
17. Coste-Marquis, S., Devred, C., Marquis, P.: Constrained argumentation frameworks. In: *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning*, pp. 112–122. AAAI Press (2006)

18. Dimopoulos, Y., Maily, J., Moraitis, P.: Control argumentation frameworks. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI 2018), The 30th innovative Applications of Artificial Intelligence (IAAI 2018), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI 2018), New Orleans, Louisiana, USA, 2–7 February 2018, pp. 4678–4685. AAAI Press (2018). <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16639>
19. Dimopoulos, Y., Maily, J.G., Moraitis, P.: Argumentation-based negotiation with incomplete opponent profiles. In: 13èmes Journées d’Intelligence Artificielle Fondamentale (JIAF 2019), pp. 91–100 (2019)
20. Doutre, S., Herzig, A., Perrussel, L.: A dynamic logic framework for abstract argumentation. In: Baral, C., De Giacomo, G., Eiter, T. (eds.) Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning. AAAI Press (2014)
21. Doutre, S., Herzig, A., Perrussel, L.: Abstract argumentation in dynamic logic: representation, reasoning and change. In: Liao, B., Ågotnes, T., Wang, Y.N. (eds.) CLAR 2018. LASLL, pp. 153–185. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-7791-4_8
22. Doutre, S., Maffre, F., McBurney, P.: A dynamic logic framework for abstract argumentation: adding and removing arguments. In: Benferhat, S., Tabia, K., Ali, M. (eds.) IEA/AIE 2017. LNCS (LNAI), vol. 10351, pp. 295–305. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60045-1_32
23. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artif. Intell.* **77**(2), 321–357 (1995). [https://doi.org/10.1016/0004-3702\(94\)00041-X](https://doi.org/10.1016/0004-3702(94)00041-X)
24. Fazzinga, B., Flesca, S., Furfaro, F.: Revisiting the notion of extension over incomplete abstract argumentation frameworks. In: Proceedings of IJCAI 2020, pp. 1712–1718. IJCAI Organization, July 2020. <https://doi.org/10.24963/ijcai.2020/237>
25. Grossi, D.: On the logic of argumentation theory. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, pp. 409–416. IFAMA (2010)
26. Harel, D., Kozen, D., Tiuryn, J.: *Dynamic Logic*. MIT Press, Cambridge (2000)
27. Herzig, A., Yuste-Ginel, A.: On the epistemic logic of incomplete argumentation frameworks. In: Proceedings of International Conference on Principles of Knowledge Representation and Reasoning. AAAI Press (2021)
28. Li, H., Oren, N., Norman, T.J.: Probabilistic argumentation frameworks. In: Modgil, S., Oren, N., Toni, F. (eds.) TAFAs 2011. LNCS (LNAI), vol. 7132, pp. 1–16. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29184-5_1
29. Maily, J.G.: A note on rich incomplete argumentation frameworks. arXiv preprint [arXiv:2009.04869](https://arxiv.org/abs/2009.04869) (2020)
30. Niskanen, A.: Computational approaches to dynamics and uncertainty in abstract argumentation. Ph.D. thesis, Helsingin yliopisto (2020)
31. Niskanen, A., Neugebauer, D., Jarvisalo, M., et al.: Controllability of control argumentation frameworks. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020). IJCAI Organization (2021). <https://doi.org/10.24963/ijcai.2020/257>
32. Proietti, C., Yuste-Ginel, A.: Dynamic epistemic logics for abstract argumentation. *Synthese* 1–60 (2021). <https://doi.org/10.1007/s11229-021-03178-5>