Temporalité et causalité en argumentation abstraite

Yann Munro, Camilo Sarmiento, Isabelle Bloch, Gauvain Bourgne, Marie-Jeanne Lesot

Temporalité et causalité en argumentation abstraite

Yann Munro* **Camilo Sarmiento***
Isabelle Bloch **Gauvain Bourgne** **Marie-Jeanne Lesot**
Sorbonne Université, CNRS, LIP6, Paris, France
{prenom.nom}@lip6.fr

Résumé

Dans le cadre de l'argumentation abstraite, nous présentons les bénéfices de prendre en compte la temporalité, c'est-à-dire l'ordre d'énonciation des arguments, ainsi que la causalité. Nous proposons une réécriture des graphes d'argumentation abstraits acycliques dans un langage d'action permettant de modéliser l'évolution du monde et d'établir des relations causales entre l'énonciation des arguments et leurs conséquences directes comme indirectes. Une implémentation en *Answer Set Programming* est également proposée ainsi que des perspectives pour aller vers des explications.

Abstract

In the context of abstract argumentation, we present the benefits of considering temporality, i.e. the order in which arguments are enunciated, as well as causality. We propose a formal method to rewrite the concepts of acyclic abstract argumentation frameworks into an action language, that allows us to model the evolution of the world, and to establish causal relationships between the enunciation of arguments and their consequences, whether direct or indirect. An Answer Set Programming implementation is also proposed, as well as perspectives towards explanations.

1 Introduction

Un système d'argumentation abstrait (AAF) offre un cadre propice pour représenter et raisonner sur des informations contradictoires par l'intermédiaire d'arguments. Ce cadre permet de trouver des ensembles d'arguments pouvant être acceptés et fournit des explications sur les raisons pour lesquelles ces ensembles ont été acceptés ou non. Les AAF proposent donc des outils appropriés pour modéliser et raisonner sur des débats. Cependant, il s'agit d'un cadre statique qui n'inclut pas de notion de temporalité qui semble cruciale pour modéliser des dialogues. Pour résoudre ce problème, plusieurs types d'approche ont été proposées. Une première catégorie modifie le graphe d'argumentation en ajoutant ou supprimant des attaques et des arguments à l'aide d'opérateurs spécifiques [3], et revient à considérer un AAF à chaque pas de temps. Une autre propose de transformer un système d'argumentation vers un formalisme logique pour ensuite utiliser des opérateurs de révision ou de changement de croyances afin de mettre à jour le système d'argumentation [15]. Nous proposons d'utiliser un autre formalisme logique, les langages d'action, afin de pouvoir modéliser la dynamique d'un dialogue.

En effet, les langages d'action, comme celui proposé dans [16], ont été naturellement conçus pour inclure cette notion dans le modèle. Ce dernier vise à déterminer l'évolution du monde étant donné un ensemble d'actions choisies délibérément par des agents et dont l'occurrence peut entraîner une réaction en chaîne d'évènements dit exogènes. Nous avons choisi le langage d'action de [16] pour trois raisons principales. Tout d'abord, il permet de gérer la concurrence d'évènements. C'est également le cas des langages comme *C* [7] ou PDDL+ [6], mais leur sémantique est adaptée respectivement aux actions non déterministes ou aux actions duratives, ce qui augmente la complexité et n'est pas utile dans notre cadre. Ensuite, ce langage comporte une définition de la notion de causalité effective. Enfin, une traduction complète et correcte en ASP est proposée dans [17].

Cet article est organisé comme suit. La section 2 présente brièvement le formalisme des AAF de Dung [4]. La section 3 fournit une description du langage d'action choisi et une définition de la notion de causalité effective. Dans la section 4, nous détaillons les principales contributions de cet article : une réécriture des AAF acycliques dans le langage d'action, l'implémentation associée et quelques propriétés de cette transformation. Elles concernent principalement la correction et la complétude de notre transformation, ainsi que la pertinence de l'inclusion de la temporalité. La section 5 est une discussion autour d'un exemple sur les

* Ces auteurs ont contribué de façon égale.

apports de notre transformation pour obtenir des informations enrichies sous forme de représentation graphique et de relations causales.

2 Système d'argumentation abstrait

Cette section rappelle les principes de base des AAF [4].

Un système abstrait d'argumentation est un couple (A, R) où A est un ensemble fini d'arguments et R est une relation binaire sur $A \times A$. On appelle R la relation d'attaque et on dit qu'un argument $a \in A$ attaque $b \in A$ si $(a, b) \in R$, ce qui s'écrit $R(a, b)$. Comme R est une relation binaire à support fini, on peut naturellement représenter un système abstrait d'argumentation sous la forme d'un graphe.

Exemple 1 *Pour illustrer ces notions, on introduit ici un scénario argumentatif modélisant l'interaction entre un médecin demandeur, D, et un radiologue, R, à propos d'un examen d'un bébé de n mois pour la pathologie X.*
D : Peux-tu me faire un scanner pour ce bébé ? (a)
R : Il vaut mieux éviter les radiations ionisantes pour les jeunes bébés. (b)
R : Je peux te proposer une IRM dans deux jours. (c)
D : On peut voir X sur une IRM ? (d)
R : Oui bien sûr ! Si tu veux une confirmation, regarde le guide des bonnes pratiques en radiologie. (e)
D : Mais puisqu'il s'agit d'un bébé, il risque de bouger et donc on pourrait manquer l'information que l'on cherche car l'image ne sera pas très nette. (f)
R : Ne t'inquiète pas, j'ai l'habitude de faire ce genre d'examen pour des bébés. (g)
D : Est-ce que cela ne coûte pas beaucoup plus cher à l'hôpital de faire une IRM ? (h) Il faut aussi que je voie avec la famille du patient car ça pourrait leur revenir plus cher (i).
R : Aucun problème dans ces cas là. Ce coût élevé englobe l'expérience acquise par mon équipe, de sorte qu'à l'avenir, elle puisse réaliser ce type d'examen délicat sans moi. (j)
D : Je viens de discuter avec la famille, aucun problème avec l'IRM elle est couverte pour ça. (k)
D : Cependant, la famille n'est pas rassurée de devoir attendre deux jours, peux-tu faire l'IRM dans la journée ? (l)
R : Non je n'ai vraiment plus de place. Mon prochain créneau est dans deux jours comme je te l'ai dit. (m)

À la suite de cet échange, la décision est donc arrêtée sur une IRM programmée dans deux jours. Mais plus tard dans la journée, le médecin reçoit un appel de la famille pour prévenir que le bébé ne va vraiment pas bien et insister sur l'urgence de l'examen. Le médecin recontacte donc le radiologue pour ajouter un dernier argument :
D : C'est vraiment urgent pour le bébé, il faut une place aujourd'hui ! (n)

A partir de ce dialogue, on peut extraire manuellement des arguments et les relations entre eux afin d'obtenir un AAF représenté en figure 1 avec les arguments suivants : {***a*** *: Scanner, b : Radiations ionisantes,* ***c*** *: IRM dans deux jours, d : X non visible par IRM, e : X visible par IRM, f : Conditions difficiles, g : Expérience, h : Coût élevé pour l'hôpital, i : Coût élevé pour le patient, j : Pas problématique pour l'hôpital, k : Famille couverte pour une IRM,* ***l*** *: IRM aujourd'hui, m : Pas de disponibilité aujourd'hui, n : C'est une urgence !*}*. Les arguments* a, c, l *sont appelés les variables de décision, leur acceptation étant le critère déclencheur d'une décision : scanner, IRM dans deux jours, ou IRM aujourd'hui.*

*Le système d'argumentation obtenu est un graphe que l'on peut associer à ce dialogue. Ce processus d'extraction peut également être effectué automatiquement, en utilisant des méthodes dites d'*argument mining *[9].*

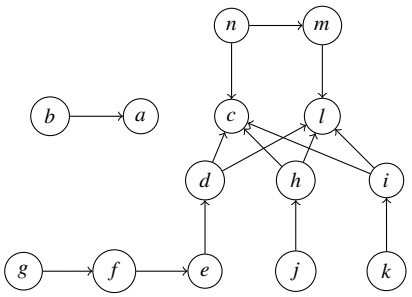

FIGURE 1 – Graphe d'argumentation associé à l'exemple 1.

Remarque – Il s'agit d'une représentation statique du dialogue dont toute notion de temporalité a été effacée. Ainsi, si les arguments avaient été énoncés dans un ordre différent, cela ne changerait pas pour autant le graphe. Cela a de l'importance quand on s'intéresse aux notions de causalité, cf section 5.2.

Une fois le graphe d'argumentation construit, il est possible de raisonner sur ce graphe afin de déterminer les ensembles d'arguments qui peuvent être acceptés. Pour cela, on rappelle quelques définitions supplémentaires :

• On note Att_a **l'ensemble des attaquants directs** de a pour la relation R : $Att_a = \{b \in A \mid R(b, a)\}$.

• Un ensemble S est **sans conflit** s'il n'y pas d'arguments $(a, b) \in S^2$ qui s'attaquent l'un l'autre.

• Un argument $a \in A$ est **acceptable** par un ensemble S si S attaque tous les attaquants de a.

• Un ensemble S sans conflit est dit **admissible** si tous ses éléments sont acceptables par S.

On peut également définir des sémantiques à base d'extension. Ce sont des propriétés qui doivent être respectées par un ensemble d'arguments afin qu'il soit accepté. Dans le cas des graphes acycliques, toutes ces sémantiques coïncident et ne forment qu'une unique extension, admissible, et ne seront donc pas évoquées ici [14].

Exemple 1 *(suite) – Le modèle obtenu pour modéliser le dialogue entre le radiologue et le médecin est acy-*

clique. Pour déterminer l'ensemble des arguments acceptables, il suffit de partir des arguments non attaqués, ici $\{b, g, j, k, n\}$*. Ces derniers sont par défaut acceptés. Ensuite, un argument attaqué par au moins un argument accepté ne peut être accepté. En appliquant ce principe, on obtient que l'argument* l *est accepté à l'inverse de* a *et* c*. La décision finale est donc de réaliser une IRM en urgence dans la journée.*

3 Langage d'action et causalité

Cette section présente d'abord la notion de langage d'action telle que définie dans [16]. Elle introduit ensuite brièvement ce qui y est défini comme causalité effective.

3.1 Sémantique et syntaxe

Le langage d'action introduit dans [16] a été conçu dans l'optique de déterminer l'évolution du monde étant donné un ensemble d'actions choisies délibérément par des agents. L'occurrence de ces actions pouvant entraîner une réaction en chaîne d'évènements dits exogènes, il est nécessaire pour avoir une connaissance complète de l'évolution du monde de s'intéresser aussi bien à l'évolution des états du monde qu'à l'occurrence des évènements. Le formalisme utilisé s'appuie sur une décomposition du monde en deux ensembles : $\mathbb{F}$ contient les variables décrivant l'état dans lequel se trouve le monde, plus précisément il s'agit de fluents instanciés représentant des propriétés du monde pouvant varier dans le temps ; $\mathbb{E}$ contient des variables décrivant des transitions dont l'occurrence modifie les fluents.

Un *littéral de fluent* est soit un fluent $f \in \mathbb{F}$, ou sa négation $\neg f$. L'ensemble des littéraux de fluents dans $\mathbb{F}$ est noté $Lit_{\mathbb{F}}$, défini par $Lit_{\mathbb{F}} = \mathbb{F} \cup \{\neg f \mid f \in \mathbb{F}\}$. Le *complément* d'un littéral de fluent l est défini comme $\bar{l} = \neg f$ si $l = f$, ou $\bar{l} = f$ si $l = \neg f$.

Définition 1 (État S**)** *L'ensemble* $L \subseteq Lit_{\mathbb{F}}$ *est un* état *si :*
- *il est cohérent :* $\forall l \in L, \bar{l} \notin L$ *;*
- *il est complet :* $\forall f \in \mathbb{F}, f \in L$ *ou* $\neg f \in L$*.*

Un état est donc un ensemble $L \subseteq Lit_{\mathbb{F}}$ donnant la valeur de chaque fluent décrivant le monde. Le temps est modélisé de façon linéaire de sorte à obtenir un état $S(t)$ pour chaque pas de temps t de l'ensemble $\mathbb{T} = \{-1, 0, \ldots, N\}$, avec $S(0)$ l'*état initial*. Il s'agit d'une formalisation bornée dans le passé d'un problème réel qui lui n'est pas borné. Pour avoir la formalisation la plus fidèle possible, tous les états précédant $t = 0$ sont recueillis dans un état $S(-1) = \mathbb{F} \backslash S(0)$.

Un évènement $e \in \mathbb{E}$ est une formule atomique caractérisée par trois composantes : des *préconditions* indiquant les conditions devant être satisfaites par l'état S pour que l'évènement puisse se déclencher ; des *conditions de déclenchement* donnant toutes les conditions devant être satisfaites au temps t pour que l'évènement puisse se déclencher, conditions dont la singularité par rapport aux préconditions est détaillée ci-dessous ; des *effets* précisant les changements de l'état du monde attendus si l'évènement se produit. Il faut en effet noter qu'un évènement peut avoir moins d'effets que ceux formalisés lorsqu'il se produit dans certains contextes.

Les préconditions et les effets sont respectivement représentés par des formules des langages $\mathcal{P} ::= l|\psi_1 \wedge \psi_2|\psi_1 \vee \psi_2$ et $\mathcal{E} ::= l|\varphi_1 \wedge \varphi_2$. Les fonctions associant à chaque évènement préconditions, conditions de déclenchement et effets sont respectivement notées pre, tri et eff, et sont définies comme : $pre : \mathbb{E} \to \mathcal{P}$, $tri : \mathbb{E} \to \mathcal{P}$, et $eff : \mathbb{E} \to \mathcal{E}$. Deux ensembles disjoints $\mathbb{A}$ et $\mathbb{U}$ forment une partition de $\mathbb{E}$ avec : $\mathbb{A}$ contient les actions réalisées par des agents et donc soumises à leur volition ; $\mathbb{U}$ contient les évènements exogènes se déclenchant aussitôt que leurs préconditions pre sont satisfaites, sans qu'un agent n'ait besoin de les réaliser. Pour les évènements exogènes, il n'y a pas de différence entre pre et tri. À l'opposé, les conditions de déclenchement des actions ne se limitent pas aux préconditions, il faut en plus la volonté de réaliser l'action de la part de l'agent, ou une sorte de manipulation d'un agent tiers qui s'y substituerait.

L'ensemble contenant tous les évènements se produisant au pas de temps t est noté $E(t)$. Le fait de gérer la concurrence d'évènements (plus d'un évènement peut avoir lieu à chaque pas de temps) est l'un des avantages principaux de ce langage d'action.

Le langage d'action décrit peut être résumé comme étant un système de transition classique, où $E(t)$ génère la transition entre les états $S(t)$ et $S(t+1)$. De ce fait, les états s'enchaînent au fur et à mesure que les évènements se produisent, simulant ainsi l'évolution du monde.

Afin d'être en mesure d'obtenir des relations causales en accord avec la conception communément admise dans la communauté de philosophes qui s'intéressent à la causalité, et cela malgré le fait d'avoir une formalisation bornée dans le passé, il est nécessaire que les évènements s'étant produits avant $t = 0$ soient représentés. Pour chaque littéral $l \in S(0)$ on introduit un évènement $ini_l \in \mathbb{E}$ tel que $eff(ini_l) = l$. On note alors $E(-1) = \{ini_l, l \in S(0)\}$ qui vérifie $eff(E(-1)) = S(0)$.

Pour résoudre des conflits potentiels ou établir des priorités entre les événements, un ordre partiel strict $\succ_{\mathbb{E}}$ est introduit, qui garantit la priorité de déclenchement d'un événement par rapport à un autre.

Définition 2 (Contexte κ**)** *Le* contexte *noté* κ *est l'octuple* $(\mathbb{E}, \mathbb{F}, pre, tri, eff, S(0), \succ_{\mathbb{E}}, \mathbb{T})$*, où* $\mathbb{E}$*,* $\mathbb{F}$*,* pre*,* tri*,* eff*,* $S(0)$*,* $\succ_{\mathbb{E}}$*, et* $\mathbb{T}$ *ont été définis précédemment.*

Définition 3 (Exécution valide) *Une* exécution *est une séquence* $E(-1), S(0), E(0), \ldots, E(N), S(N+1)$*. Elle est* va-

lide *étant donné un contexte* κ *si elle vérifie* $\forall t \in \mathbb{T}$ *:*

1. $S(t) \subseteq Lit_{\mathbb{F}}$ *est un état au sens de la définition 1.*
2. $E(t) \subseteq \mathbb{E}$ *vérifie :*
   - 2.a $\forall e \in E(t),\ S(t) \models pre(e)$;
   - 2.b $\nexists (e, e') \in E(t)^2,\ e \succ_{\mathbb{E}} e'$;
   - 2.c $\forall e \in \mathbb{E}$ *tel que* $S(t) \models tri(e)$, $e \in E(t)$ *ou* $\exists e' \in E(t),\ e' \succ_{\mathbb{E}} e$;
3. $S(t+1) = \left\{ l \in S(t), \forall e \in E(t), \bar{l} \notin \mathit{eff}(e) \right\} \cup \{ l \in Lit_{\mathbb{F}}, \exists e \in E(t), l \in \mathit{eff}(e) \}$.

Pour un contexte κ donné, il existe potentiellement plus d'une exécution valide. En effet, aucune spécification du moment où les actions sont réalisées n'est inclus dans le contexte. Leurs préconditions peuvent être satisfaites, et donc des exécutions peuvent être valides, mais leurs conditions de déclenchement ne le peuvent pas. L'ajout en entrée d'un ensemble d'actions couplées à un temps $\sigma \subseteq \mathbb{A} \times \mathbb{T}$ qui modélise la volition des agents, appelé *scénario*, permet d'obtenir une unique exécution valide. D'une telle exécution il est possible d'extraire deux types de traces :

Définition 4 (Traces $\tau^e_{\sigma,\kappa}$ et $\tau^s_{\sigma,\kappa}$) *Étant donné un scénario* σ *et un contexte* κ*, la* trace d'évènements $\tau^e_{\sigma,\kappa}$ *de* σ, κ *est la séquence d'évènements* $E(-1), E(0), \ldots, E(N)$ *contenue dans une des exécutions valides étant donné* κ*, telle que :* $\forall t, \forall e \in E(t),\ e \in \mathbb{A} \Leftrightarrow (e,t) \in \sigma$.
La trace d'états $\tau^s_{\sigma,\kappa}$ *est la séquence d'états* $S(0), S(1), \ldots, S(N+1)$ *correspondant à* $\tau^e_{\sigma,\kappa}$.

3.2 Causalité effective

La définition de causalité effective proposée dans [16] est une formalisation adaptée aux langages d'action du « NESS test ». Celui-ci stipule [18] : *«A particular condition was a cause of a specific consequence if and only if it was a necessary element of a set of antecedent actual conditions that was sufficient for the occurrence of the consequence.»*

Une relation causale est un lien entre une cause à un effet. Le fait que les langages d'action représentent le monde comme une succession d'états produits par des occurrences d'évènements introduit des états entre les évènements. De ce fait, en plus de la relation de causalité effective qui relie deux occurrences d'évènements entre elles, comme communément accepté par les philosophes, il est nécessaire de définir des relations causales où les causes sont des occurrences d'évènements, et les effets sont la véracité de formules du langage $\mathcal{P}$ à un temps donné. Le NESS test est utilisé pour définir ces relations intermédiaires. Pour pouvoir fournir la définition de causalité effective adaptée aux langages d'action, trois relations causales sont introduites dans [16] (pour les détails voir [17]). (i) Les NESS-causes directes donnent des informations essentielles en se basant sur les effets que l'occurrence d'un évènement a réellement eus, qui à nouveau ne sont pas nécessairement les mêmes que ceux attendus. Comme mentionné précédemment, il s'agit donc d'une relation entre l'occurrence d'un évènement et la véracité de formules du langage $\mathcal{P}$. Malgré leur aspect indispensable, ces relations ne sont pas toujours les plus intéressantes. En effet, l'ensemble des NESS-causes directes d'une formule peut contenir un certain nombre d'évènements exogènes qui ne sont pas nécessairement pertinents. Il est donc essentiel d'établir une chaîne causale en remontant le temps de sorte à retrouver l'ensemble d'actions qui sont derrière la véracité de la formule de $\mathcal{P}$. (ii) Les NESS-causes permettent de retrouver cette chaîne causale. En notant $\psi \in \mathcal{P}$ la formule qui nous intéresse à l'instant t_ψ et C l'ensemble des NESS-causes directes de (ψ, t_ψ), la NESS-cause s'intéresse à ce qui a causé $(tri(C), t)$, où $t < t_\psi$ nécessairement. Il faut noter que par définition les NESS-causes directes sont un type particulier de NESS-causes. Enfin, l'occurrence d'un premier évènement est considéré comme une cause effective (iii) de l'occurrence d'un second d'évènement si l'occurrence du premier est une NESS-cause des conditions de déclenchement du deuxième. De cela nous pouvons déduire que, si l'occurrence (e', t_2) est une NESS-cause directe de (ψ, t_3) et que l'occurrence (e, t_1) est une cause effective de (e', t_2) avec $t_1 < t_2 < t_3$, alors l'occurrence (e, t_1) est une NESS-cause de (ψ, t_3). Ces trois relations causales sont illustrées à l'aide de l'exemple en section 5.2.

4 Passage des AAF au langage d'action

Dans cette section, nous présentons la contribution principale de cet article, à savoir une réécriture d'un graphe d'argumentation abstrait acyclique dans le langage d'action présenté dans la section précédente. Pour cela, la section 4.1 présente la définition du contexte argumentatif κ, la section 4.2 fournit les définitions modifiées de la sémantique associée au langage d'action, la section 4.3 décrit brièvement l'implémentation ASP. Enfin, la section 4.4 présente les propriétés de la transformation proposée.

Contrairement aux AAF, nous proposons de prendre en compte l'ordre d'énonciation des arguments. Au lieu d'avoir seulement un couple (A, R), l'entrée est un couple (Δ, R), où Δ est un dialogue, c'est-à-dire une séquence d'énoncés en langage naturel :

Définition 5 (Dialogue Δ) *Un* dialogue*,* Δ*, est défini comme* $\Delta = \{(a, o) \mid a \in A, o \in \mathbb{N}\}$*, où chaque argument* a *est associé à son ordre d'énonciation* o.

4.1 Instanciation du contexte κ

Pour pouvoir passer d'un graphe d'argumentation au langage d'action décrit en section 3, il faut d'abord définir les fluents $\mathbb{F}$ c'est-à-dire les variables nécessaires pour décrire

le monde, ici le graphe d'argumentation. Deux éléments doivent être pris en compte : les arguments et les relations d'attaque. Pour décrire un argument x, nous introduisons deux fluents : $p_x \in \mathbb{F}$ qui indique si l'argument est présent ou non dans le graphe et $a_x \in \mathbb{F}$ qui indique l'acceptabilité de l'argument. Pour R, nous utilisons le fluent $cA_{y,x} \in \mathbb{F}$ exprimant le fait que y peut attaquer x. Comme nous ne traitons que des AAF acycliques, $\nexists (x_1, \ldots, x_n) \in A$ tel que $\left(cA_{x_1,x_2}, \ldots, cA_{x_{n-1},x_n}, cA_{x_n,x_1}\right) \in \mathbb{F}$. Nous appelons cette propriété l'acyclicité des fluents cA.

En ce qui concerne les évènements $\mathbb{E}$, dans le cas de l'argumentation abstraite la seule action volontaire possible est d'énoncer un argument x, notée $enunciate_x \in \mathbb{A}$. Pour cela, il faut que l'argument en question n'ait pas déjà été prononcé. Dans ce cas, x devient présent et acceptable par défaut. Ce choix est justifié car on évaluera son acceptabilité à l'état suivant avant qu'il n'ait un impact sur le reste du graphe. Formellement :

$$pre(enunciate_x) \equiv \neg p_x$$
$$eff(enunciate_x) \equiv p_x \wedge a_x$$

Remarque – Aucun des évènements décrits par la suite n'a pour effet de rendre un argument non présent. Cela implique qu'il n'est pas possible de ré-énoncer un argument déjà énoncé. Cette hypothèse n'est pas en contradiction avec le cadre de l'argumentation classique. En effet, un argument répété se manifesterait par un argument identique mais de nom différent dans le graphe ce qu'il est évidemment possible aussi avec notre transformation. Cependant, le langage d'action que nous utilisons offrant des outils pour tenir compte de la temporalité, une meilleure approche pourrait exister mais nécessiterait une étude approfondie. Malgré tout, cet article étant une première étape, il vise à poser des bases solides au prix de quelques hypothèses simplificatrices.

Contrairement au cadre de l'argumentation abstraite, nous prenons ici en compte l'ordre d'énonciation des arguments. Cela implique de mettre à jour l'acceptabilité de tous les autres arguments présents après l'énonciation d'un nouvel argument et avant l'énonciation du suivant. Cela définit des états que nous appelons *états argumentatifs*.

Définition 6 (État argumentatif) *Un état $S(t)$ est dit argumentatif si :*
i) $\forall x, y, \left[S(t) \models a_x \wedge p_y \wedge cA_{y,x} \Rightarrow S(t) \models \neg a_y\right]$;
ii) $\forall x, \left[S(t) \models p_x \wedge \left(\bigwedge_y \neg a_y \vee \neg cA_{y,x}\right) \Rightarrow S(t) \models a_x\right]$.

Après l'énonciation d'un argument, nous souhaitons que des mises à jour soient déclenchées automatiquement. Nous les représentons par deux évènements exogènes : $makesUnacc_{y,x} \in \mathbb{U}$ et $makesAcc_x \in \mathbb{U}$. Pour rappel, un argument n'est acceptable que s'il est non attaqué ou attaqué uniquement par des arguments non acceptables. De fait, il suffit que l'un des attaquants soit acceptable pour rendre l'argument attaqué non acceptable. Cela implique donc au moins deux cas à envisager :

Mise à jour de l'acceptabilité : Supposons que l'argument y venant d'être énoncé peut attaquer l'argument x et que x et y sont acceptables. Alors, x étant attaqué par un argument acceptable y, x devient non acceptable. Formellement, l'évènement exogène $makesUnacc_{y,x}$ peut s'écrire :

$$tri(makesUnacc_{y,x}) \equiv a_x \wedge a_y \wedge cA_{y,x}$$
$$eff(makesUnacc_{y,x}) \equiv \neg a_x$$

Cette écriture permet également de traiter les cas où un nouvel argument z rend un attaquant y de x à nouveau acceptable. Dans ce cas, x devient non acceptable.

Mise à jour de la non-acceptabilité : Supposons que l'argument x est non acceptable et qu'un argument z vient d'être prononcé. Celui-ci n'a pas de lien direct avec l'argument x mais a pu impacter l'acceptabilité de certains attaquants de x. On vérifie donc si tous les arguments pouvant attaquer x sont acceptables ou non. Si aucun d'entre eux n'est effectivement acceptable, alors x le redevient. Dans le langage d'action, cela se traduit par l'évènement exogène $makesAcc_x$ tel que :

$$tri(makesAcc_x) \equiv p_x \wedge \neg a_x \wedge \left(\bigwedge_y \neg cA_{y,x} \vee \neg a_y\right)$$
$$eff(makesAcc_x) \equiv a_x$$

Enfin, lorsqu'un argument x est énoncé, il faut vérifier qu'il n'est pas rendu non acceptable par un argument y déjà présent avant qu'il ne rende non acceptables d'autres arguments, z par exemple. Cela se traduit par la règle de priorité ci-dessous :

$$makesUnacc_{y,x} \succ_{\mathbb{E}} makesUnacc_{x,z}$$

Notons qu'ajouter un argument dans le graphe ne peut impacter les autres arguments de manière directe qu'en les rendant non acceptables. Pour cette raison, il n'est pas nécessaire d'établir une règle de priorité de la forme $makesUnacc_{y,x} \succ_{\mathbb{E}} makesAcc_z$ car cette situation est déjà couverte par la règle précédente.

Remarque – Dans la transformation proposée, on ne fait pas de distinction entre la notion d'attaque potentielle et celle d'attaque réelle car la différence entre ces dernières disparaît dans les équations. En effet, considérons un fluent $attack_{y,x} \in \mathbb{F}$ traduisant le fait que l'argument y attaque effectivement l'argument x. Définissons l'évènement exogène $isAttacking_{y,x} \in \mathbb{U}$ comme :

$$tri(isAttacking_{y,x}) \equiv p_x \wedge p_y \wedge cA_{y,x}$$
$$eff(isAttacking_{y,x}) \equiv attack_{y,x}$$

D'après cette définition, y attaque x si les deux sont présents et y peut attaquer x. Cependant, pour que cette attaque soit prise en compte, il faut toujours que l'attaquant y soit acceptable. On obtient des conditions de la forme $a_y \wedge attack_{y,x}$, c'est-à-dire $a_y \wedge p_y \wedge cA_{y,x}$. Or un argument ne peut pas être acceptable sans être présent donc $a_y \wedge p_y \equiv a_y$. Ainsi, avec ce nouveau fluent on aurait : $tri(makesUnacc_{y,x}) = a_y \wedge a_x \wedge attack_{y,x} = a_y \wedge a_x \wedge cA_{y,x}$. On retrouve la même précondition que sans l'introduction de $isAttacking$ et de $attack$. De fait, nous avons choisi de n'utiliser que $canAttack$.

4.2 Modification de la sémantique

Après avoir défini le contexte κ pour le cadre argumentatif, il faut modifier les définitions associées à la sémantique du langage d'action. Cela va permettre en particulier d'obtenir des traces représentatives de la réalité. Pour cela, les arguments sont énoncés à partir d'états argumentatifs étape par étape dans l'ordre de l'interaction. Comme nous ne considérons que des graphes acycliques, il existe toujours un tel état après l'ajout d'un nouvel argument et il est donc toujours possible de continuer l'interaction.

La définition actuelle du scénario σ n'est pas adaptée à ce cas. En effet, elle demande la connaissance préalable du nombre d'étapes nécessaires pour revenir à un état argumentatif, de sorte à prévoir l'état exact dans lequel l'argument suivant pourra être énoncé. Nous proposons pour résoudre ce problème d'introduire un ensemble d'actions ordonnées $\varsigma \subseteq \mathbb{A} \times \mathbb{N}$, appelé *séquence*. L'unicité de l'exécution valide n'est plus obtenue grâce au scénario σ, mais à la séquence ς. Il faut donc modifier les définitions 3 et 4.

Définition 7 (Configuration argumentative χ) *La* configuration argumentative*, notée χ, est le couple (ς, κ) avec ς une séquence et κ un contexte.*

La définition 8 suivante modifie la définition 3 : la condition 2.d est ajoutée et dans la condition 2.c $\forall e \in \mathbb{E}$ est remplacé par $\forall e \in \mathbb{U}$. Ces modifications expriment le fait qu'une action de la séquence ς ne peut être déclenchée que si aucun évènement exogène ne se déclenche au même pas de temps. Les conditions 1, 2.a, 2.b, et 3 restent identiques, nous ne modifions donc rien vis-à-vis du déclenchement des évènements exogènes.

Définition 8 (Exécution valide dans le cadre des AAF) *Une exécution est une séquence :*
$E(-1), S(0), E(0), \ldots, E(N), S(N+1)$. *Elle est valide étant donné κ si, en plus des conditions 1, 2.a, 2.b et 3 de la définition 3, elle vérifie $\forall t \in \mathbb{T}$:*

2 $E(t) \subseteq \mathbb{E}$ vérifie :

2.c $\forall e \in \mathbb{U}$ tel que $S(t) \models tri(e)$,
$e \in E(t)$ ou $\exists e' \in E(t),\ e' \succ_{\mathbb{E}} e$;

2.d Si $\exists e \in E(t) \cap \mathbb{A}$, alors $\forall e' \in \mathbb{U}$, $S(t) \not\models tri(e')$;

2.e $E(t) \neq \emptyset$.

Dans la définition 4, les traces ont été définies comme des extraits d'une exécution valide compte tenu de κ et de conditions supplémentaires liées à σ. Au lieu de définir directement des traces, la définition 9 suivante correspond à une exécution valide étant donnée $\chi = (\varsigma, \kappa)$. Les traces sont simplement des extraits de ces exécutions valides.

Définition 9 (Exécution valide étant donné χ) *Soit une configuration argumentative $\chi = (\varsigma, \kappa)$. Une exécution valide étant donné κ est* valide étant donné χ *si :*

1. $\forall t \in \mathbb{T}$, $E(t) \subset (\{a, \exists o \in \mathbb{N}, (a, o) \in \varsigma\} \cup \mathbb{U})$;
2. $\forall ((e, o), (e', o')) \in \varsigma^2$ *tel que* $o < o'$, $\exists t, t'$ *tel que* $e \in E(t)$ *et* $e' \in E(t')$ *et* $t < t'$;
3. $\forall ((e, o), (e', o')) \in \varsigma^2$ *tel que* $o = o'$, $\exists t$ *tel que* $(e, e') \in E(t)^2$.

Soit une exécution valide étant donné χ. Sa *trace d'évènements* τ^e_χ est sa séquence d'évènements $E(-1), E(0), \ldots, E(N)$, et sa *trace d'états* τ^s_χ sa séquence d'états $S(0), S(1), \ldots, S(N+1)$.

4.3 Implémentation en ASP

Nous proposons une implémentation en ASP, sur la base de celle décrite dans [17]. Les programmes ASP $\pi_{con}(\kappa)$ et $\pi_{seq}(\varsigma)$ sont obtenus par la traduction respectivement du contexte κ et de la séquence ς, $\pi_{\mathbb{A}}$ est obtenu par la traduction de la sémantique du langage d'action introduite dans la section 3.1 et modifiée dans la section 4.2, et $\pi_{\mathbb{C}}$ est obtenu par la traduction des définitions des relations causales introduites par [16]. Une traduction complète et correcte est proposée dans [17]. Le programme complet $\Pi(\chi) = \pi_{sce}(\varsigma) \cup \pi_{con}(\kappa) \cup \pi_{\mathbb{A}} \cup \pi_{\mathbb{C}}$ est disponible [1].

4.4 Quelques propriétés formelles

Cette section donne les propriétés formelles de la transformation proposée. Tout d'abord, nous établissons que la notion de temporalité est bien prise en compte par la transformation. Ensuite, nous établissons sa correction et sa complétude. Enfin, nous introduisons une proposition qui ouvre la voie à la discussion de la section 5. Toutes les preuves sont omises pour cause de place, elles sont détaillées dans [13].

Le premier résultat montre que, bien que les exécutions valides étant donné κ ne soient pas uniques, les exécutions valides étant donné χ le sont, ainsi que les traces correspondantes τ^e_χ et τ^s_χ.

Proposition 1 *Soit une configuration argumentative $\chi = (\varsigma, \kappa)$, les traces τ^e_χ et τ^s_χ sont uniques.*

1. https://gitlab.lip6.fr/sarmiento/kr_2023.git

Dorénavant, lorsqu'il sera question d'évènements et d'états, il s'agira de ceux étant donné les traces uniques τ^e_χ et τ^s_χ. Ainsi, l'ensemble des événements qui se sont effectivement produits à l'instant t est $E^\chi(t) = \tau^e_\chi(t)$. De même, l'état réel à l'instant t est $S^\chi(t) = \tau^s_\chi(t)$.

Nous établissons à présent l'aspect complet et correct de notre transformation. Pour cela, nous introduisons d'abord la notion de graphe associé.

Définition 10 *Soit $S^\chi(t)$ un état. On appelle graphe associé le graphe $AF' = (A', R')$, tel que $A' = \{x \mid S^\chi(t) \models p_x\}$ et $R' = \{(y, x) \mid S^\chi(t) \models cA_{y,x}\}$.*

Un état argumentatif est considéré comme un état où rien ne se passe tant qu'une action volontaire n'est pas effectuée. Nous montrons maintenant qu'il est toujours possible d'atteindre un tel état à partir d'un état argumentatif dans lequel un argument $x \in A$ est énoncé.

Proposition 2 *Soit $S^\chi(t)$ un état argumentatif et $x \in A$ un argument. Si $enunciate_x \in E^\chi(t)$, alors $\exists t' \in \mathbb{T}$, $t < t'$ tel que $S^\chi(t')$ est un état argumentatif.*

Enfin, la proposition suivante permet de prouver qu'un argument acceptable dans l'état argumentatif est acceptable dans le graphe associé et vice-versa.

Proposition 3 *Soit $S^\chi(t)$ un état argumentatif et $AF = (A, R)$ son graphe associé. Alors, pour tout x, $x \in A$ est acceptable par A si et seulement si $S^\chi(t) \models a_x$.*

Nous avons établi qu'il existe une équivalence entre un état argumentatif et son graphe associé. Maintenant, à partir d'un dialogue et de la relation d'attaque, les traces sont générées ainsi qu'un AAF. Nous établissons alors l'existence d'un état dont le graphe associé est égal à l'AAF initial. Un tel état est appelé *état argumentatif final* et est défini comme un état argumentatif $S^\chi(t)$ tel que $\forall x \in A$, $\exists t' \in \mathbb{T}$ tel que $t' < t$ et $enunciate_x \in E^\chi(t')$.

Théorème 1 (Correction et complétude) *Soit un dialogue Δ et R un ensemble de relations d'attaque. Étant donné une configuration argumentative χ, le graphe argumentatif associé AF' à un état argumentatif final $S^\chi(t)$, et $AF = (A, R)$ le graphe d'argumentation construit à partir de (Δ, R), on a $AF' = AF$.*

Les résultats précédents permettent de montrer la cohérence de l'état final du langage d'action avec l'argumentation. En particulier, le théorème 1 est essentiel car il établit la correction et la complétude de notre approche avec l'argumentation, et permet ainsi d'assurer qu'aucune information n'est perdue.

À l'inverse, intégrer la temporalité permet d'ajouter des informations supplémentaires grâce aux états intermédiaires, comme illustré dans la section suivante, et aux relations causales qui peuvent en être déduites. Ainsi, on a le résultat suivant :

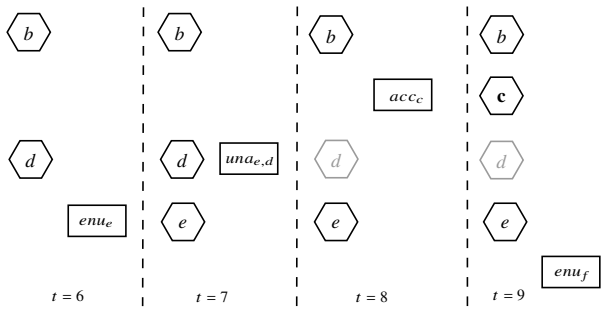

FIGURE 2 – Représentation possible d'un extrait des traces d'évènements $\tau^e_{\varsigma,\kappa}$ et d'états $\tau^s_{\varsigma,\kappa}$. Les fluents sont représentés par des hexagones et les évènements par des rectangles.

Proposition 4 *Les relations causales sont dépendantes de la séquence ς.*

Cette proposition montre que les relations causales sont dépendantes de l'ordre d'énonciation des arguments. Ainsi, même si, comme dans le cadre classique de l'argumentation, l'acceptabilité d'un argument dans l'état argumentatif final n'en dépend pas (cf théorème 1), il est tout de même essentiel de tenir compte de la temporalité lorsque l'on s'intéresse à des notions proches de la causalité notamment pour l'explicabilité. Ce résultat peut être visualisé à partir de l'exemple 2 et illustre l'enrichissement qu'apporte le passage à un langage d'action. L'utilisation de ces relations causales est discutée dans la section suivante.

5 Application à l'exemple et discussion

Dans cette section, nous appliquons le programme $\Pi(\chi)$ à l'exemple 1. Les traces d'évènements $\tau^e_{\varsigma,\kappa}$ et d'états $\tau^s_{\varsigma,\kappa}$, ainsi que les relations causales permettent de construire les figures 2, 3 présentées ci-dessous.

5.1 Représentation graphique et explication

En argumentation, Cyras et al. [19] proposent une classification des méthodes permettant de générer des explications. Parmi elles, une catégorie se concentre sur l'extraction de sous-graphes argumentatifs pour justifier l'acceptation ou le rejet d'un argument pour une certaine sémantique, produisant une représentation graphique du processus d'acceptation ou de rejet d'un argument.

Grâce à la transformation décrite dans la section précédente, nous proposons également des représentations graphiques du processus argumentatif. En effet, les traces d'évènements et d'états permettent d'obtenir une narration de l'interaction représentable graphiquement. Une première forme que peut prendre cette visualisation est présentée pour l'exemple 1 avec le schéma simplifié en figure 2.

Exemple 1 *(suite) – Comme nous nous concentrons principalement sur l'acceptabilité des arguments, nous avons*

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h,i*	*j*	*k*	*l*	*m*	*n*
a	•	◦	◦	◦	◦	◦	◦	◦	◦	◦	◦	◦	◦
b		•	•	•	•	•	•	•	•	•	•	•	•
c			•	◦	•	◦	•	◦	◦	•	•	•	◦
d				•	◦	•	◦	◦	◦	◦	◦	◦	◦
e					•	◦	•	•	•	•	•	•	•
f						•	◦	◦	◦	◦	◦	◦	◦
g							•	•	•	•	•	•	•
h								•	◦	◦	◦	◦	◦
i								•	•	◦	◦	◦	◦
j									•	•	•	•	•
k										•	•	•	•
l											•	◦	•
m												•	◦
n													•

TABLE 1 – Représentation graphique de l'interaction.

décidé de représenter uniquement les fluents a_x de $\tau^s_{\varsigma,\kappa}$. Pour rester concis, nous n'utilisons que le nom des arguments pour représenter leur acceptabilité. Qui plus est, pour des raisons de lisibilité, nous ne faisons pas apparaître ce fluent lorsque c'est sa négation qui est vraie dans l'état. Nous faisons une exception lorsque c'est l'occurrence d'un évènement représenté qui a pour effet la négation du fluent. Dans ce cas, la négation est représentée par une nuance plus claire. Les évènements $enunciate_x$, $makesUnacc_{y,x}$, et $makesAcc_x$ sont représentés respectivement par les noms plus courts enu_x, $una_{y,x}$, et acc_x.

Le premier état représenté correspond à $S(6)$, état argumentatif au sens de la définition 6 permettant l'énonciation de l'argument suivant. Tous les arguments précédant e ayant déjà été énoncés, l'action $enunciate_e$ peut être faite. L'occurrence de cet évènement est la transition vers l'état suivant $S(7)$ où, comme le présente la figure 2, l'argument e est acceptable. Contrairement à $S(6)$, $S(7)$ n'est pas un état argumentatif. En effet, la condition (i) de la définition 6 n'est pas respectée car $(a_d \wedge cA_{e,d}) \in S(7)$ et $a_e \in S(7)$. L'état n'étant pas argumentatif, l'argument qui suit ne peut pas être énoncé. Toutefois, les conditions de déclenchement de $makesUnacc_{e,d}$ étant satisfaites, cet évènement exogène est déclenché ce qui entraîne une nouvelle transition d'état. L'argument d n'étant plus acceptable dans $S(8)$, la condition (i) est à nouveau satisfaite. Cela ne suffit pas pour rendre l'état argumentatif. En effet, la condition (ii) n'est pas satisfaite par $S(8)$ empêchant l'argument suivant d'être énoncé. $makesAcc_c$ est déclenché conduisant à l'état suivant $S(9)$. Ici, comme le représente la figure 2, l'argument c est acceptable. Ce nouvel état étant argumentatif, l'argument suivant, f, peut être énoncé. Le dialogue se poursuit ainsi pas à pas et se termine à l'état $S(31)$.

Pour des raisons de place, cette figure ne représente qu'une partie de l'interaction.

Une deuxième forme possible, plus compacte, est proposée dans le tableau 1 : la première colonne représente les arguments du graphe et la première ligne l'ordre des actions

ς_1	c	d	e	f	g	h,i	j	k	l	m	n
c	•	◦	•	◦	•	◦	◦	•	•	•	◦
ς_2	c	l	m	n	d	e	f	g	h,i	j	k
c	•	•	•	◦	◦	◦	◦	◦	◦	◦	◦

TABLE 2 – Impact de l'ordre d'énonciation des arguments (lignes 1, 3) sur l'acceptabilité des arguments (lignes 2, 4).

réalisées. Par souci de lisibilité, l'action $enunciate_x$ est résumée en x. Enfin, • signifie que l'argument est acceptable tandis que ◦ signifie qu'il ne l'est pas. Si un argument n'a pas encore été énoncé, la notion d'acceptabilité n'a pas de sens, ce qui a été représenté par les cases grisées. Contrairement à la représentation précédente où l'on faisait apparaître les étapes de mises à jour, cette deuxième forme a l'avantage d'être plus compacte et donc de permettre de mieux visualiser l'échange dans sa globalité. Ainsi, en regardant la ligne *c*, il est possible de suivre l'évolution de l'acceptabilité de cet argument en fonction des arguments qui ont été énoncés. Par exemple, l'énonciation de *e* (colonne *e*) fait passer *c* de ◦ (cf colonne précédente) à •, c'est-à-dire de non acceptable à acceptable. Cette forme de représentation permet également de voir rapidement l'impact direct et indirect de l'ordre d'énonciation des arguments sur l'évolution du scénario complet. Cela est illustré avec l'exemple 2.

Exemple 2 *Modifions un peu le scénario de l'exemple 1. Le dialogue débute de la même façon avec l'énonciation des arguments a, b, c. À ce moment là, le médecin demande ensuite directement s'il n'est pas possible de faire l'IRM aujourd'hui même (l). Le radiologue répond qu'il ne peut que dans deux jours au plus tôt (m). Le médecin précise alors qu'il s'agit d'une urgence (n). Ensuite, le reste du dialogue se déroule dans l'ordre présenté dans la troisième ligne du tableau 2.*

Dans celui-ci, nous avons représenté uniquement l'évolution de l'acceptabilité de c, à partir de son énonciation, variable de décision avec a et l. Même si l'état final du graphe d'argumentation n'est pas modifié, on observe l'impact très important que peut avoir l'ordre des actions dans les étapes intermédiaires qui mènent à ce dernier. Ainsi, dans ce nouveau scénario, la variable de décision c est refusée dès la 6^e action, c'est-à-dire l'énonciation de n, et cela sans modification jusqu'à la fin. Ces nuances ne sont pas représentées par les approches statiques.

Cependant, ni les méthodes par extraction de sous-graphe, ni les représentations que nous venons de présenter ne s'intéressent ou ne mettent en valeur la chaîne causale qui a conduit à l'action ou à la décision, propriété importante pour une explication d'après Miller [11].

5.2 Causalité et explication

Dans [19], une autre catégorie de méthodes se rapproche de cette notion d'explication causale en recherchant quels arguments doivent être retirés d'un graphe d'argumentation pour rendre acceptable un argument qui ne l'était pas [5]. En matière de causalité, cela correspond à une recherche de *but-for* cause de la non acceptabilité d'un argument. Cependant, ce test ne permet pas de résoudre les cas où l'occurrence de l'un de deux évènements aurait été suffisante pour causer un effet en l'absence de l'autre, appelés surdétermination [10]. Pour cela, il faut utiliser d'autres méthodes comme celle présentée dans la section 3.2, ou encore les équations structurelles de Halpern et Pearl [8]. Bien que ces deux méthodes permettent de traiter les cas de surdétermination, elles ne le font pas de la même façon et ne s'accordent pas sur un même résultat. D'un point de vue philosophique, la définition de causalité sous-jacente au NESS test appartient à la famille des approches par régularité [1], alors que les définitions de Halpern et Pearl appartiennent à la famille des approches contrefactuelles [10]. Résoudre le débat sur quelle approche est la plus adéquate est en dehors du cadre de cet article. Discutons maintenant comment nous nous démarquons de la transformation des graphes d'argumentation abstraits acycliques proposée dans [12], permettant d'exploiter la dernière définition de causalité proposée par Halpern afin de générer des explications causales en argumentation.

Le premier point de différenciation est la façon dont nous représentons le monde. En effet, comme montré dans la section 5.1, l'utilisation d'un langage d'action nous permet de prendre en compte la dynamique du dialogue. Cela n'est pas sans importance dans la compréhension de celui-ci, étant donné que la temporalité est fondamentale dans la façon dont on se représente le monde. Le deuxième point de différenciation est lié à la causalité. D'un point de vue purement mathématique, la définition de causalité de Halpern peut être qualifiée de « *Contrastive actual weak sufficiency* » d'après Beckers [2], alors que celle utilisée ici est « *Minimal actual strong sufficiency* » d'après cette même typologie. En quelques mots, alors que la première accorde beaucoup d'importance au fait qu'une cause doit être nécessaire à l'effet, d'où l'aspect contrastif, la deuxième place la suffisance au premier plan et asservit la nécessité à cette dernière. Plus de détails sur les implications de ces différences sont discutés dans [2, 18]. D'un point de vue pratique, l'avantage de l'approche causale utilisée ici est qu'elle ne nécessite pas de raisonnement contrefactuel ni d'interventionnisme, des mécanismes coûteux d'un point de vue computationel et critiqués du fait qu'ils introduisent de la subjectivité dans l'enquête causale [16, 18]. Le fait d'être dans un cadre où l'analyse causale se fait a posteriori, et donc en pleine connaissance du déroulement des évènements, enlève un avantage aux méthodes contrefactuelles qui sont très utiles lorsque l'on raisonne a priori et que l'on souhaite explorer les autres déroulements possibles.

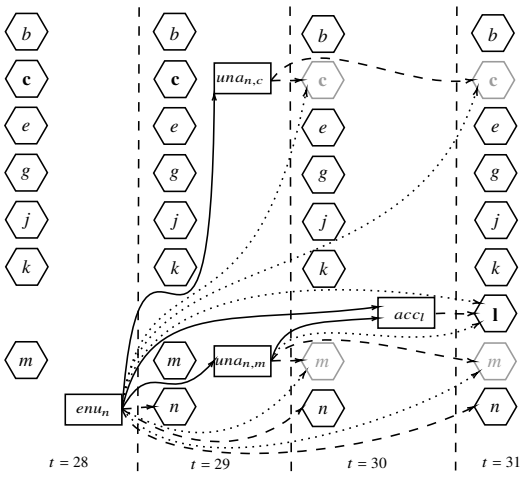

FIGURE 3 – Représentation possible d'un extrait de la trace d'évènements $\tau^e_{\varsigma,\kappa}$, de la trace d'états $\tau^s_{\varsigma,\kappa}$ et des relations causales. NESS-causes directes (– –), NESS-causes (· · · · ·), et causes effectives (——).

En ajoutant les relations causales trouvées par le programme à la représentation des traces d'évènements et d'états, il est possible de générer une représentation visuelle plus riche de l'interaction qui s'est produite. La figure 3 en est une illustration pour les quatre derniers états de trace de l'exemple 1. Il s'agit de la partie du dialogue correspondant à l'énonciation de l'argument *n* et aux mécanismes de mise à jour qui en découlent.

Exemple 1 *(suite) – La représentation des traces d'évènements et d'états étant la même que pour la figure 2, ici nous commentons uniquement les relations causales que l'on trouve dans la figure 3. Rappelons que les arguments a, c et l sont les variables de décision. L'argument a devenant non acceptable très tôt dans le dialogue et le restant tout le long, nous avons choisi de ne pas le représenter dans les figures.*

L'argument n autour duquel toute la figure est articulée est celui qui vient clore le débat. Cet argument, énoncé par le médecin demandeur, porte sur le caractère urgent de l'examen. Sur la figure 3 nous pouvons voir que l'énonciation de cet argument dans l'état $S(28)$ *est une NESS-cause directe de l'acceptabilité de l'argument dans les états suivants, relation que nous notons* $(enunciate_n, 28)$ *NESS-cause directe* $(a_n, 29{-}31)$*. De même, nous avons* $(makesUnacc_{n,c}, 29)$ *NESS-cause directe* $(\neg a_c, 30-31)$*,* $(makesUnacc_{n,m}, 29)$ *NESS-cause directe* $(\neg a_m, 30-31)$*, et* $(makesAcc_l, 30)$ *NESS-cause directe* $(a_l, 31)$*. Comme le montrent ces exemples, cette première relation est la brique de base de la causalité, elle s'intéresse aux relations causales étant donné les effets réels de l'occurrence d'un évènement. Toutefois, cette relation n'est pas suffisante. Si nous souhaitons savoir pour-*

quoi l'argument l est acceptable à la fin du dialogue, et donc pourquoi la décision prise est de faire une IRM le jour même, dire simplement que c'est à cause de l'occurrence $(makesAcc_l, 30)$ *n'est pas satisfaisant.*

Cherchons alors à en savoir un peu plus sur les raisons pour lesquelles l'occurrence $(makesAcc_l, 30)$ *a eu lieu. Pour cela, il faut s'intéresser aux NESS causes et causes effectives afin de construire la chaîne causale qui a mené à* $(makesAcc_l, 30)$*. Puis par transitivité, nous obtenons que* $(makesUnacc_{n,m}, 29)$ *est une cause du fait que* $makesAcc_l$ *se soit déclenché, et donc des effets que ce déclenchement a pu avoir. En remontant encore plus loin en recherchant les causes pour lesquelles l'occurrence* $(makesUnacc_{n,m}, 29)$ *a eu lieu, nous trouvons* $(enunciate_n, 28)$ *cause effective* $(makesUnacc_{n,m}, 29)$ *et donc* $(enunciate_n, 28)$ *NESS-cause* $(\neg a_m, 30 - 31)$*. Par transitivité, nous pouvons déduire* $(enunciate_n, 28)$ *NESS-cause* $(a_l, 31)$*. Cette nouvelle relation nous permet de dire que le médecin demandeur précisant qu'il s'agit d'une urgence est une des causes de la décision finale, réponse qui paraît déjà plus satisfaisante et pouvant faire partie d'une explication. Ce même raisonnement peut-être appliqué pour trouver les causes de* $(\neg a_c, 31)$*, l'autre variable décisionnelle.*

Discussion – L'exemple précédent montre que les chaînes causales sont composées d'un nombre important de relations même si le nombre d'états étudiés est restreint. Dans le contexte de l'intelligence artificielle explicable (XAI), Miller explique dans [11] que dans le cadre d'une explication, la chaîne causale est très importante. Pour autant, il ajoute qu'une explication doit également être courte. De fait, la question des relations à mettre en avant reste à résoudre si nous voulons utiliser cette méthode pour générer des explications.

Miller précise également [11] qu'une explication est contrastive. De fait, lors du processus de recherche d'explications, il est important de pouvoir raisonner sur des scénarios contrefactuels afin de fournir des explications effectivement contrastives. Or, comme nous l'avons décrit précédemment, l'approche causale adoptée n'utilise pas ce genre de raisonnement.

Pour ces raisons, dans notre approche nous ne proposons pour le moment qu'une représentation visuelle du mécanisme conduisant à la décision. Cette dernière a l'avantage de permettre de représenter toute la chaîne causale grâce à des schémas comme celui présenté en figure 3. Ils peuvent évidemment être un support à une explication mais n'en constituent pas une indépendamment.

6 Conclusion

Nous avons proposé dans cet article une formalisation des systèmes abstraits d'argumentation acycliques dans le langage d'action présenté dans [16]. Cette transformation permet tout d'abord d'augmenter l'expressivité de ces modèles grâce à l'intégration de la temporalité, ce qui permet d'examiner l'effet de l'ordre d'énonciation des arguments. De plus, elle nous permet aussi d'exploiter la notion de causalité associée au langage d'action, offrant la possibilité de donner des informations supplémentaires sur l'acceptation ou le rejet d'un argument ainsi que des justifications sur ce dernier. Pour cela, nous avons proposé deux types de représentations graphiques du processus d'argumentation formant un support visuel et ouvrant la voie à de nouvelles formes d'explications en argumentation.

Les perspectives de ce travail visent à développer de telles explications, en appliquant les principes développés dans le contexte de l'intelligence artificielle explicable, par exemple détaillés dans [11] : les chaînes causales sont établies comme essentielles pour les explications, mais elles doivent également être courtes. La question des relations à privilégier reste donc ouverte, ainsi que la manière dont elles peuvent être utilisées pour définir des explications contrastives, nécessitant de pouvoir raisonner sur des scénarios contrefactuels.

Remerciements : Les auteurs remercient la Professeure Catherine Adamsbaum, radio-pédiatre, pour les discussions sur les exemples. Ce travail a été en partie financé par la chaire d'I. Bloch en intelligence artificielle (Sorbonne Université et SCAI).

Références

[1] Andreas, H. et M. Guenther: *Regularity and Inferential Theories of Causation*. Dans *The Stanford Encyclopedia of Philosophy*. Stanford University, 2021.

[2] Beckers, S.: *Causal Sufficiency and Actual Causation*. J. Philos. Log., 50(6) :1341–1374, 2021.

[3] Doutre, Sylvie, Faustine Maffre et Peter McBurney: *A dynamic logic framework for abstract argumentation : adding and removing arguments*. Dans *Advances in Artificial Intelligence : From Theory to Practice : 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2017*. Springer, 2017.

[4] 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.

[5] Fan, X. et F. Toni: *On explanations for non-acceptable arguments*. Dans *Theory and Applications of Formal Argumentation : 3rd Int. Workshop*, pages 112–127. Springer, 2015.

[6] Fox, M. et D. Long: *Modelling Mixed Discrete-Continuous Domains for Planning*. Journal of Artificial Intelligence Research, 27 :235–297, 2006.

[7] Giunchiglia, E. et V. Lifschitz: *An Action Language Based on Causal Explanation : Preliminary Report*. Dans *AAAI*, pages 623–630, 1998.

[8] Halpern, J. Y. et J. Pearl: *Causes and explanations : A structural-model approach. Part I : Causes*. The British J. Philosophy of Science, 2005.

[9] Lippi, M. et P. Torroni: *Argumentation mining : State of the art and emerging trends*. ACM Trans. on Internet Technology, 16(2) :1–25, 2016.

[10] Menzies, P. et H. Beebee: *Counterfactual Theories of Causation*. Dans *The Stanford Encyclopedia of Philosophy*. Stanford University, 2020.

[11] Miller, T.: *Explanation in Artificial Intelligence : Insights from the Social Sciences*. Artificial Intelligence, 267 :1–38, 2019.

[12] Munro, Y., I. Bloch, M. Chetouani, M.-J. Lesot et C. Pelachaud: *Argumentation and Causal Models in Human-Machine Interaction : A Round Trip*. Dans *8th Int. Workshop on AI and Cognition*, 2022.

[13] Munro, Y., C. Sarmiento, I. Bloch, G. Bourgne et M.-J. Lesot: *Temporality and Causality in Abstract Argumentation*. CoRR, abs/2303.09197, 2023.

[14] Rahwan, I. et G. R. Simari: *Argumentation in artificial intelligence*, tome 47. Springer, 2009.

[15] Saint-Cyr, Florence Dupin de, Pierre Bisquert, Claudette Cayrol et Marie Christine Lagasquie-Schiex: *Argumentation update in YALLA (yet another logic language for argumentation)*. International Journal of Approximate Reasoning, 75 :57–92, 2016.

[16] Sarmiento, C., G. Bourgne, K. Inoue et J. G. Ganascia: *Action Languages Based Actual Causality in Decision Making Contexts*. Dans *PRIMA*, pages 243–259, 2022.

[17] Sarmiento, Camilo, Gauvain Bourgne, Katsumi Inoue, Daniele Cavalli et Jean Gabriel Ganascia: *Action Languages Based Actual Causality for Computational Ethics : a Sound and Complete Implementation in ASP*. CoRR, abs/2205.02919, 2023.

[18] Wright, R. W.: *Causation in Tort Law*. California Law Review, 73(6) :1735–1828, 1985.

[19] Čyras, K., A. Rago, E. Albini, P. Baroni et F. Toni: *Argumentative XAI : A Survey*. Dans *IJCAI-21*, pages 4392–4399, 2021.