

Le système SESAME : Spécifier des sémantiques argumentatives

Philippe Besnard^{2,1} Sylvie Doutre^{3,1} Dominique Longin^{2,1}

¹ IRIT (UMR 5505), France

² CNRS, Université de Toulouse, France

³ Université de Toulouse 1, France

{philippe.besnard, sylvie.doutre, dominique.longin}@irit.fr

Résumé

SESAME est un système qui, dans le cadre de l'argumentation abstraite, permet de spécifier une sémantique argumentative et qui s'attaque au problème de vérification : pour un graphe d'argumentation donné, tel ou tel ensemble d'arguments constitue-t-il une extension sous la sémantique spécifiée par l'utilisateur ? Les sémantiques qui peuvent être spécifiées au travers du système SESAME ne se limitent pas aux sémantiques déjà connues, un utilisateur peut en spécifier de nouvelles. Le système SESAME fournit en sortie une représentation logique de la sémantique saisie sous la forme d'une formule paramétrée. Appliquée à un ensemble d'arguments d'un graphe d'argumentation donné, la formule instanciée va être satisfiable si et seulement si cet ensemble est une extension pour ce graphe sous la sémantique spécifiée.

Abstract

SESAME is a tool which allows to specify argumentation semantics for abstract argumentation frameworks, and to tackle the verification problem : is a given set of arguments an extension under a semantics for a given argumentation framework ? The semantics which can be specified go well beyond the range of semantics already known, and the user can indeed specify brand new semantics of her own. The system then provides a logical encoding in the form of a parametrized formula. When applied to a given subset of arguments of a given argumentation framework, the instantiated formula will be satisfiable if and only if the set is an extension for the framework according to the specified semantics.

Mots-clefs. Argumentation, logique, spécification.

1 Introduction

Cette note est une courte introduction à SESAME (SEmantics Specification in Abstract arguMENTation), un système qui permet à son utilisateur de spécifier une sémantique argumentative, et qui en donne un codage logique sous la forme d'une formule propositionnelle paramétrée [4].

Appliqué à un système (graphe) d'argumentation donné $G = (\mathcal{A}, \mathcal{R})$, pour $S \subseteq \mathcal{A}$, le principe est que la formule instanciée est *satisfiable* si et seulement si S est une *extension* pour G selon la sémantique argumentative spécifiée par l'utilisateur.

La FIGURE 1 présente une capture d'écran de SESAME.

Rappel. Un graphe d'argumentation est un graphe orienté fini $(\mathcal{A}, \mathcal{R})$. Les éléments de \mathcal{A} sont appelés arguments et $b\mathcal{R}a$ se lit « b attaque a » pour a et b appartenant à \mathcal{A} .

Exemple de codage de la la sémantique stable.

Pour la sémantique stable de [6] par exemple, il faut construire une formule $\sigma_{(\mathcal{A}, \mathcal{R}), S}$ pour le cas $\sigma = \text{stable}$.

Définition Pour un graphe d'argumentation $(\mathcal{A}, \mathcal{R})$, une *extension stable* est un ensemble $S \subseteq \mathcal{A}$ tel que :

- il n'existe pas deux arguments a et b de S t.q. $a\mathcal{R}b$;
- pour tout argument $a \notin S$, il existe $b \in S$ t.q. $b\mathcal{R}a$.

La formule $\sigma_{(\mathcal{A}, \mathcal{R}), S}$ va traduire une reformulation de ces deux conditions :

- pour tout a dans S , pour tout b attaquant a , il est faux que b est dans S ;
- pour tout a dans \mathcal{A} , si a n'est pas dans S alors il existe b attaquant a tel que b est dans S .

Pourquoi reformuler ? Afin d'obtenir des conditions qui sont des combinaisons d'énoncés de base

« l'argument x est dans l'ensemble X »

devant se coder par des symboles propositionnels

$$\varphi_{(x \in X)}$$

ainsi notés pour des raisons de lisibilité (dans les faits, ce sont de simples atomes du langage propositionnel).

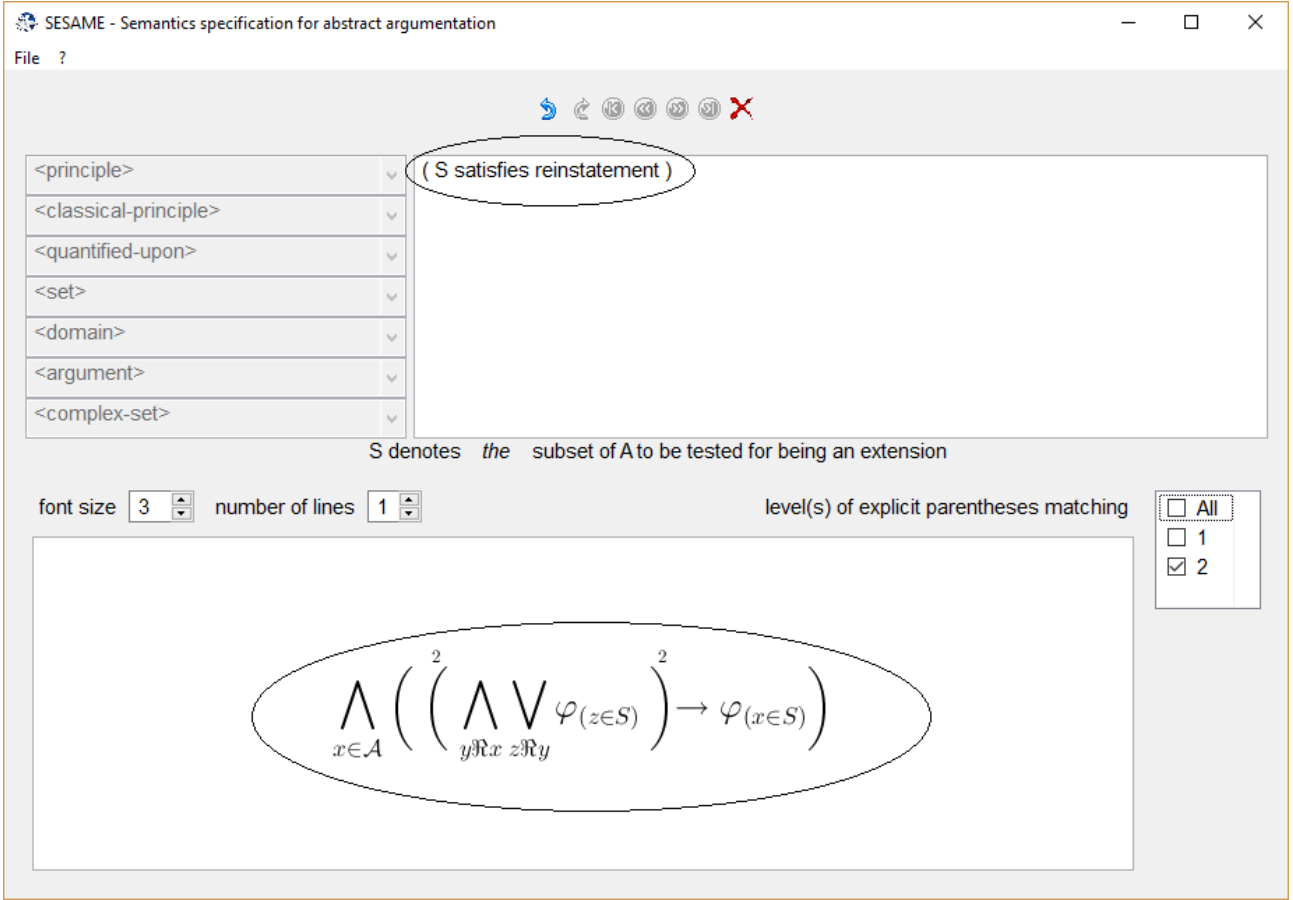


FIGURE 1 – Sémantique argumentative « S satisfait le principe de reinstatement » et la formule résultante $\sigma_{(\mathcal{A}, \mathcal{R}), S}$ (les chiffres visent à faciliter la visualisation de l'appariement des parenthèses)

Ainsi, SESAME génère une sous-formule (de $\sigma_{(\mathcal{A}, \mathcal{R}), S}$) exprimant que S remplit les deux conditions requises pour être une extension stable :

$$\left(\bigwedge_{a \in S} \bigwedge_{b \mathcal{R} a} \neg \varphi(b \in S) \right) \wedge \bigwedge_{a \in \mathcal{A}} \left(\neg \varphi(a \in S) \rightarrow \bigvee_{b \mathcal{R} a} \varphi(b \in S) \right)$$

Enfin, l'ensemble Φ_S signifiant que tous les éléments de S sont dans S et que tous les arguments hors de S ne sont pas dans S est défini par¹

$$\Phi_S \stackrel{\text{def}}{=} \{ \varphi(a \in S) \mid a \in S \} \cup \{ \neg \varphi(a \in S) \mid a \in \mathcal{A} \setminus S \}$$

d'où une formule $\sigma_{(\mathcal{A}, \mathcal{R}), S}$ pour les extensions stables :

$$\left(\bigwedge_{a \in S} \bigwedge_{b \mathcal{R} a} \neg \varphi(b \in S) \right) \wedge \bigwedge_{a \in \mathcal{A}} \left(\neg \varphi(a \in S) \rightarrow \bigvee_{b \mathcal{R} a} \varphi(b \in S) \right) \wedge \bigwedge \Phi_S$$

1. Puisqu'une condition (sémantique autre que la sémantique stable) pourrait être vraie pour un sous-ensemble de S mais pas S , il faut préciser qu'on spécifie bien tout S (et rien que S).

Au final, le système SESAME demande à l'utilisateur de spécifier une sémantique argumentative (qui n'est pas nécessairement déjà connue) sous forme d'une combinaison de principes (notion introduite par [2] dans le domaine) — à l'exception de $\bigwedge \Phi_S$ qui reste transparent pour l'utilisateur.

2 Spécifier une sémantique

L'élément central du processus de spécification d'une sémantique argumentative via SESAME est un arbre de décomposition selon une certaine grammaire. L'utilisateur développe cet arbre en cliquant sur des boutons dans la fenêtre principale, qui affiche les développements possibles des non-terminaux apparaissant dans l'arbre partiellement développé.

De fait, la gamme des sémantiques argumentatives supportées par le système SESAME est déterminée par une grammaire hors contexte telle que les énoncés primitifs sont, comme mentionnés plus haut, de la forme « l'argument x est dans l'ensemble X ». A cet effet, la grammaire dispose

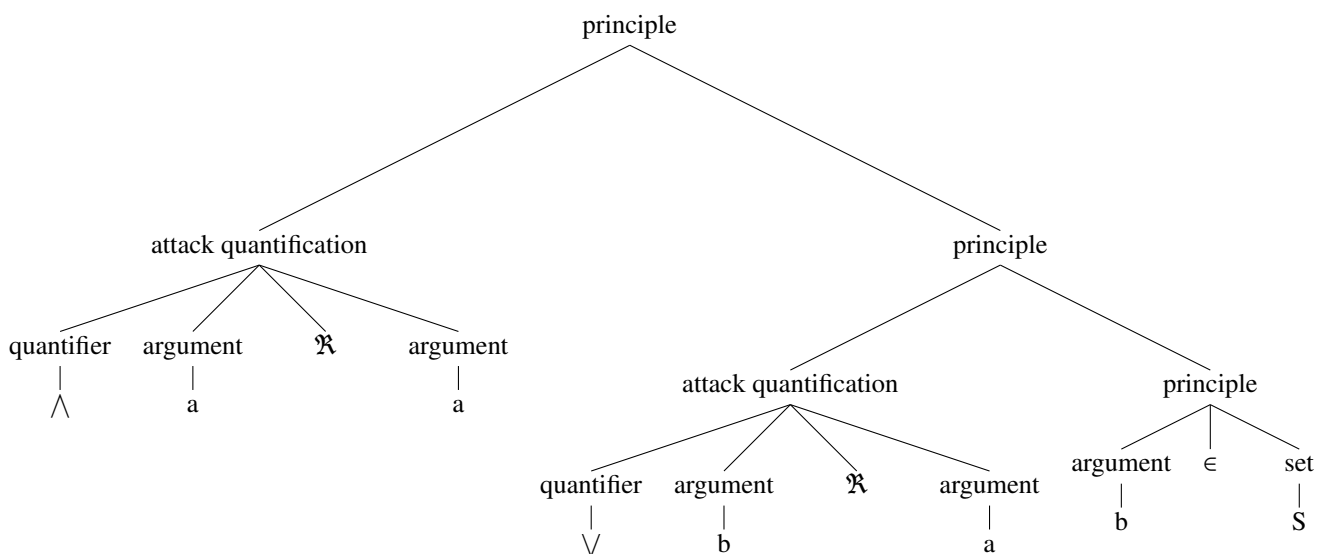


FIGURE 2 – Arbre de décomposition « tout argument s’attaquant lui-même est attaqué par au moins un argument de S »

de terminaux a, b, c, \dots et $\mathcal{A}, S, S_1, S_2, \dots$ (Attention, \mathcal{A} désigne l’ensemble des arguments du graphe alors que S désigne *le* sous-ensemble candidat au statut d’extension — par exemple, S est une extension stable si et seulement si S satisfait \dots). La grammaire permet divers types de combinaisons, booléennes et aussi par quantification.

Concrètement, quand l’utilisateur clique sur le bouton ⟨principe⟩, apparaît un menu déroulant qui indique quelles chaînes peuvent remplacer ⟨principe⟩, à savoir « il est faux que ⟨principe⟩ » ou « ⟨principe⟩ and ⟨principe⟩ » ou \dots

Après chaque choix effectué par l’utilisateur le système SESAME d’une part met à jour une description (en pseudo langage naturel) de l’arbre de décomposition et, d’autre part, affiche (en L^AT_EX) la version actualisée de la formule en cours de construction.

Si par exemple l’utilisateur souhaite définir le principe que tout argument qui s’attaque lui-même doit être attaqué par au moins un argument de S , il va *de facto* développer l’arbre de décomposition de la FIGURE 2.

Cet arbre de décomposition (invisible pour l’utilisateur) produit instantanément à l’écran l’affichage de la formule

$$\bigwedge_{a \mathcal{R} a} \bigvee_{b \in S} \varphi_{(b \in S)}$$

qui s’obtient via une séquence appropriée de clics sur les boutons de développement.

3 Aide à l’utilisateur

Naturellement, exprimer une sémantique argumentative selon les termes de la grammaire peut exiger une bonne dose d’ingéniosité de la part de l’utilisateur.

C’est une conséquence directe de l’idée que les boutons de développement correspondent aux non-terminaux de la grammaire. Chacun de ces boutons est associé à un menu déroulant comprenant plusieurs items qui correspondent aux règles développant ledit non-terminal (par exemple, le menu déroulant pour ⟨argument⟩ affiche la liste a, b, c, \dots, x, y, z). Il faut donc dépasser cela de sorte qu’en exception à cette idée, certaines options dans le menu déroulant ne se limitent pas à ces items mais soient en réalité des macros explicitées dans le but d’aider l’utilisateur dans son activité de spécification. Par exemple, la grammaire permet d’exprimer qu’un sous-ensemble des arguments est maximal pour un principe. Ou similairement, elle permet d’exprimer que les attaquants d’un certain argument forment un sous-ensemble de tel autre sous-ensemble. Il serait exagéré d’exiger qu’un utilisateur usuel fasse l’effort de déterminer comment exprimer cela à partir des seules règles de la grammaire. C’est pourquoi le bouton de développement pour le nonterminal ⟨principe⟩ comporte des options telles que : « ⟨ensemble⟩ est maximal pour ⟨principe⟩ », « ⟨ensemble-complexe⟩ est inclus dans ⟨ensemble-complexe⟩ », \dots

Si une spécification contradictoire lui est soumise, le système SESAME génère une formule insatisfaisable quelle que soit la valeur des paramètres. En revanche, le système n’effectue aucune vérification sémantique : l’utilisateur peut spécifier quelque chose de douteux, *e.g.*, que S soit une « extension » si et seulement si S contient tout argument qui s’attaque lui-même.

4 Conclusion

Le système SESAME permet à l'utilisateur d'exprimer des sémantiques argumentatives, connues ou originales, afin d'en obtenir une caractérisation sous la forme d'une formule propositionnelle paramétrée qu'il est possible d'instancier sur tout graphe argumentatif $(\mathcal{A}, \mathcal{R})$ et tout $S \subseteq \mathcal{A}$ pour tester sa satisfiabilité. Le système est téléchargeable à l'adresse suivante : <http://www.irit.fr/SESAME>.

Pour le moment, SESAME permet seulement de spécifier des sémantiques argumentatives de type extension-based par opposition au type ranking-based en plein essor actuellement [1, 5], et une suite logique sur laquelle nous commençons déjà à réfléchir est d'autoriser la spécification de sémantiques de ce second type. Par ailleurs, nous avons aussi le projet de coupler SESAME à un logiciel permettant de faire l'instanciation de la sémantique par un graphe argumentatif afin de vérifier *via* un solveur SAT comme TouIST [3] si un sous-ensemble S de l'ensemble des arguments est une extension ou non.

Remerciements

Ce travail a bénéficié du soutien du projet AMANDE (ANR-13-BS02-0004) de l'Agence Nationale pour la Recherche. La première version de SESAME a été développée par Van Hieu Ho.

Références

- [1] Amgoud, Leila et Jonathan Ben-Naim: *Ranking-Based Semantics for Argumentation Frameworks*. Dans Liu, Weiru, V. S. Subrahmanian et Jef Wijsen (éditeurs) : *7th International Conference on Scalable Uncertainty Management (SUM'13)*, tome 8078 de *Lecture Notes in Artificial Intelligence*, pages 134–147, Washington, DC, USA, 2013. Springer.
- [2] Baroni, Pietro et Massimiliano Giacomin: *On Principle-based Evaluation of Extension-based Argumentation Semantics*. *Artificial Intelligence*, 171(10) :675–700, 2007.
- [3] Ben Slimane, Khaled Skander, Alexis Comte, Olivier Gasquet, Abdelwahab Heba, Olivier Lezaud, Frederic Maris et Mael Valais: *Twist your logic with TouIST*. CoRR, 2015. [abs/1507.03663](https://arxiv.org/abs/1507.03663).
- [4] Besnard, Philippe, Sylvie Doutre, Van Hieu Ho et Dominique Longin: *SESAME - A System for Specifying Semantics in Abstract Argumentation*. Dans Thimm, Matthias, Federico Cerutti, Hannes Strass et Mauro Vallati (éditeurs) : *International Workshop on Systems and Algorithms for Formal Argumentation (SAFA)*, tome 1672, pages 40–51, <http://CEUR-WS.org>, 2016. CEUR Workshop Proceedings.
- [5] Bonzon, Elise, Jérôme Delobelle, Sébastien Konieczny et Nicolas Maudet: *A Comparative Study of Ranking-Based Semantics for Abstract Argumentation*. Dans Schuurmans, Dale et Michael P. Wellman (éditeurs) : *30th AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 914–920, Phoenix, Arizona, USA, 2016. AAAI Press.
- [6] Dung, Phan Minh: *On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-person Games*. *Artificial Intelligence*, 77(2) :321–357, 1995.