



HAL
open science

L'équipe LILaC et l'IA

Dominique Longin, Emiliano Lorini

► **To cite this version:**

Dominique Longin, Emiliano Lorini. L'équipe LILaC et l'IA. Bulletin de l'Association Française pour l'Intelligence Artificielle, 2017, Dossier "Equipes de recherche en IA", 97. hal-03128354

HAL Id: hal-03128354

<https://hal.science/hal-03128354v1>

Submitted on 2 Feb 2021

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.

■ LILaC : Logique, Interaction, Langage, et Calcul

*Institut de Recherche en Informatique de Toulouse
(IRIT, UMR 5505)
CNRS, INPT, UPS, UT1C, UT2J
<https://www.irit.fr/-Equipe-LILaC->*

Dominique LONGIN
Dominique.Longin@irit.fr
+33 5 61 55 64 47

Emiliano LORINI
Emiliano.Lorini@irit.fr
+33 5 61 55 64 47

Membres

La liste exhaustive des membres de l'équipe LILaC peut être consultée en cliquant [ici](#). Les permanents sont au nombre de 14 (7 chercheurs CNRS, 3 enseignants-chercheurs de l'Université Paul Sabatier, 4 enseignants-chercheurs de l'Université Toulouse Capitole) et les non-permanents au nombre de 14 (1 chercheur associé, 1 chercheur invité, 1 collaborateur extérieur, 2 post-doctorants, 9 doctorants).

Thématique générale de l'équipe

Les recherches de l'équipe LILaC se focalisent sur les modèles formels de l'interaction en adoptant une approche normative (top-down) à travers cinq thématiques.

1. Logiques pour l'action et la dynamique des croyances. Un aspect important de l'interaction est la dynamique de l'environnement. Celui-ci évolue grâce à l'exécution d'événements et d'actions qui ont des répercussions sur les états mentaux des agents (via leurs observations ou leurs perceptions). L'équipe LILaC s'intéresse entre autres : aux logiques de l'action telle que la logique dynamique



Afia

Association française
pour l'Intelligence Artificielle

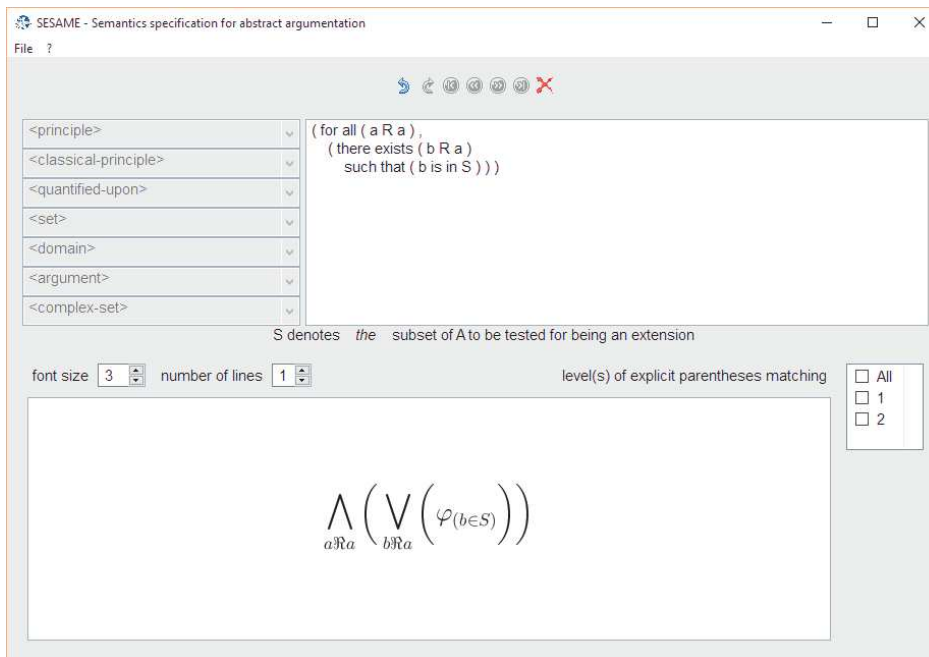


Figure 1.1 – Le logiciel SESAME de production de sémantiques argumentatives à base d'extension

dont elle étudie les propriétés formelles (complétude, décidabilité, complexité) ; à l'algorithmique du raisonnement sur les actions en étudiant les propriétés formelles de théories de l'action et des événements ; à la révision et à la mise à jour afin de prendre en compte une nouvelle information, en particulier dans le cas multi-agent ; aux aspects épistémiques de la planification multi-agent.

2. Sécurité des systèmes d'information et de communication. En environnement ouvert, l'interaction entre agents nécessite d'être protégée.

En premier lieu, cette protection peut se décliner sous la forme de trois propriétés : disponibilité, intégrité, confidentialité. En partant de la spécification formelle d'un système, l'équipe LILaC s'intéresse d'une part, à fournir la preuve que ce système vérifie les trois propriétés ci-dessus, et d'autre part à obtenir de manière automatique une implémentation qui correspond à la spécification et dont la correction est prouvée par construction. Au niveau du contrôle d'accès, il s'agit d'étendre le modèle de la matrice du contrôle d'accès et de tracer, pour ce qui concerne le problème de la sûreté des modèles

étendus, la limite entre décidabilité et indécidabilité ; Au niveau de la vérification de protocoles, il s'agit de définir des algorithmes pour vérifier automatiquement des protocoles qui vont permettre à des agents de garantir la confidentialité ou l'intégrité des informations qu'ils échangent.

En second lieu, la protection d'un système est également étudiée dans l'équipe via la modélisation (en termes de rôles et de relations entre utilisateurs) de l'organisation dans laquelle s'insère ce système informatique : il s'agit alors de caractériser à la fois les aspects prédictifs de la notion de rôle et ses aspects déontiques qui font appel aux concepts de permission et d'obligation, à la base également d'autres notions comme celles de confiance, de responsabilité, ou de délégation.

3. Logique, théorie des jeux, et choix social.

Dans cet axe, l'équipe LILaC s'intéresse aux mécanismes sociaux qui régissent les interactions entre agents (artificiels ou humains). Cela concerne notamment : la théorie du vote, qui fournit des modèles d'interaction et de coordination pour les systèmes multi-agents, où l'utilisation de la logique



Afia

Association française
pour l'Intelligence Artificielle

permet de spécifier et vérifier ces systèmes ; la théorie des jeux épistémiques et les logiques pour les jeux ; la théorie de la diffusion d'opinion et de l'influence sociale. Des simples systèmes sont implémentés afin de tester les différents mécanismes élaborés sur différentes populations d'agents.

4. Logiques pour les systèmes multi-agents et argumentation.

Un des objectifs majeurs de l'équipe est d'élaborer un modèle logique général de l'interaction. Cela se traduit par de nombreux travaux visant l'intégration des modèles provenant des thématiques précédentes en un formalisme unifié. Ce dernier s'appuie sur des logiques BDI et des logiques d'action, afin de pouvoir exprimer à la fois des croyances, désirs et intentions des agents, ainsi que les actions pouvant être accomplies par les agents, tout en prenant en compte l'évolution des croyances des agents. Cette intégration considère aussi l'argumentation abstraite, en tant que modèle de raisonnement basé sur des interactions entre arguments ; l'évaluation des interactions permet de déterminer des ensembles d'arguments collectivement acceptables (extensions), ou un ordre d'acceptabilité entre les arguments. La dynamique des systèmes d'argumentation, la définition de différents modes d'évaluation (sémantiques argumentatives), et les liens entre argumentation et logique, sont au coeur des recherches actuelles de l'équipe.

5. Logiques et raisonnement automatique.

Les modèles de l'interaction qui précèdent posent des problèmes calculatoires et nécessitent la mise au point d'algorithmes permettant de répondre à des questions de type « telle formule est-elle conséquence de telle théorie logique ? ». En tant que spécialiste des logiques non classiques, l'équipe LILaC a substantiellement renouvelé la méthode des tableaux sur le plan théorique (complétude et complexité de logiques admettant une sémantique de Kripke) ; sur le plan pratique, ce travail a été implémenté dans la plate-forme LoTREC qui offre un langage de programmation de très haut niveau pour la définition rapide de démonstrateurs pour toute logique modale normale. Nous avons également exploré certains systèmes paraconsistants et proposé une traduction des problèmes posés en termes de formules booléennes quantifiées.

Plus récemment, nous nous intéressons : au développement de la plateforme ToulST qui constitue une interface entre un langage propositionnel de haut niveau et très concis, et différents solveurs (SAT, SMTP, et plus récemment QBF) ; à la traduction (de fragments) de certains langages en un langage utilisable par ToulST afin d'automatiser des procédures de preuve pour lesquelles nous n'avons pas de démonstrateur ; à l'implémentation de certains jeux pour la recherche de stratégies gagnantes ; à la planification épistémique d'actions (robotique cognitive).

Contributions

L'équipe LILaC publie ses résultats dans les plus grands journaux internationaux (Artificial Intelligence, Journal of Artificial Intelligence Research, Journal of Logic and Computation, Studia Logica, Synthese, Journal of Autonomous Agents and Multi-Agent Systems, etc.) ainsi que dans les plus grandes conférences internationales (IJCAI, AAMAS, TARK, etc.).

Au niveau des réalisations logicielles, l'équipe LILaC a développé : LoTREC (un démonstrateur pour la quasi totalité des logiques modales) ; SESAME (un logiciel de production de sémantiques argumentatives encodées dans un langage de type logique propositionnelle, voir Figure 1.1) ; ToulST (une interface dotée d'un langage très compact pour la logique propositionnelle permettant d'utiliser des solveurs SAT, SMT et QBF, et développé en collaboration avec l'équipe ADRIA de l'IRIT, voir Figure 1.2).

Au niveau international ses chercheurs travaillent avec de nombreux collaborateurs issus de différents pays européens, américains ou asiatiques.

Au niveau national, l'équipe LILaC est investie dans l'Institut Cognition (Tremplin Carnot), un réseau de 14 laboratoires répartis sur le sol français en relation avec les sciences cognitives dont le but est de travailler de manière privilégiée avec le monde de la recherche industrielle. L'équipe adhère à l'Afia en tant que personne morale.

Au niveau local, l'équipe LILaC est membre du Labex CIMI qui lui finance régulièrement des projets ou étudiants en thèse.

Voir [ici](#) pour la liste des publications de l'équipe LILaC.



Afia

Association française
pour l'Intelligence Artificielle

The screenshot shows the Toulist interface with the following content:

Left Pane (Code):

```
13
14 exists prend_2(0): \ \ ; ; joueur 0 (nous)
15 forall prend_2(1): \ \ ; ; joueur 1 (adversaire)
16 exists prend_2(2): \ \ ; ; joueur 0 (nous)
17 forall prend_2(3): \ \ ; ; joueur 1 (adversaire)
18 exists prend_2(4): \ \ ; ; joueur 0 (nous)
19
20 ; ; But
21 not 0_perd \ \
22 ; ; Conditions initiales
23 and reste(0,$NA)
24 and tour_de_0(0) \ \
25 and
26
27 ; ; Resultat de l'action de la prise : si on a pris, il faut b
28 ; ; le nombre d'allumettes.
29 bigand $t,$n in $T,$A when $n>=2:
30   (reste($t,$n) and prend_2($t) => reste($t+1,$n-2))
31   and \ \
32   (reste($t,$n) and not prend_2($t) => reste($t+1,$n-1))
33 end \ \
34 and
35
36 ; ; On ne peut pas prendre 2 allumettes si il n'en reste plus
37 bigand $t in $T:
38   reste($t,1) => (not prend_2($t) and reste($t+1,0))
39 end \ \
40 and
41
42 ; ; Au moins un nombre d'allumettes par tour.
43 bigand $t in $T:
44   bigor $n in $A:
45     reste($t,$n)
46 end
```

Right Pane (Formula):

$$\begin{aligned} & \mathbf{A} \leftarrow [0..NA] \\ & \mathbf{T} \leftarrow [0..NA] \\ & \exists \text{prend_}2_0. \\ & \forall \text{prend_}2_1. \exists \text{prend_}2_2. \\ & \forall \text{prend_}2_3. \exists \text{prend_}2_4. \\ & \neg 0_perd \\ & \wedge \text{reste}_{0,NA} \wedge \text{tour_de_}0_0 \\ & \wedge \bigwedge_{\substack{t,n \in T,A \\ n \geq 2}} \left(((\text{reste}_{t,n} \wedge \text{prend_}2_t) \Rightarrow \text{reste}_{t+1,n-2}) \wedge \right. \\ & \quad \left. ((\text{reste}_{t,n} \wedge \neg \text{prend_}2_t) \Rightarrow \text{reste}_{t+1,n-1}) \right) \\ & \wedge \bigwedge_{t \in T} (\text{reste}_{t,1} \Rightarrow (\neg \text{prend_}2_t \wedge \text{reste}_{t+1,0})) \\ & \wedge \bigwedge_{t \in T} \bigvee_{n \in A} \text{reste}_{t,n} \\ & \wedge \bigwedge_{t,n_1 \in T,A} \left(\text{reste}_{t,n_1} \Rightarrow \bigwedge_{\substack{n_2 \in A \\ n_1 \neq n_2}} \neg \text{reste}_{t,n_2} \right) \\ & \wedge \left(0_perd \Leftrightarrow \right. \\ & \quad \left. \bigvee_{t \in T} (\text{tour_de_}0_t \wedge \text{reste}_{t,0} \wedge \neg \text{reste}_{t-1,1}) \right) \\ & \wedge \bigwedge_{t \in T} (\neg \text{tour_de_}0_t \Leftrightarrow \text{tour_de_}0_{t+1}) \end{aligned}$$

Bottom Bar: 21:14 | Exporter | Importer | Résoudre | QBF

Figure 1.2 – L'interface Toulist pour les démonstrateurs SAT, SMT et QBF