



HAL
open science

Poster sur “La logique facile avec TouIST”

Olivier Gasquet, Andreas Herzig, Dominique Longin, Frédéric Maris, Maël Valais

► To cite this version:

Olivier Gasquet, Andreas Herzig, Dominique Longin, Frédéric Maris, Maël Valais. Poster sur “La logique facile avec TouIST”. Journées Francophones sur la Planification, la Décision et l’Apprentissage pour la conduite de systèmes (JFPDA 2017), Jul 2017, Caen, France. , 2017. hal-03116309

HAL Id: hal-03116309

<https://hal.science/hal-03116309v1>

Submitted on 20 Jan 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.



Formalisez et résolvez facilement des problèmes avec des solveurs SAT, SMT ou QBF

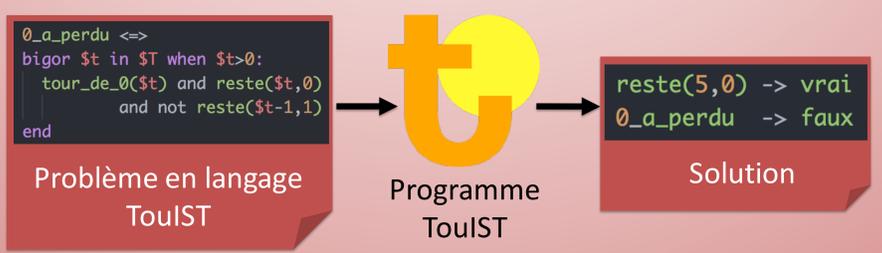
Olivier Gasquet¹, Andreas Herzig², Dominique Longin², Frédéric Maris¹, Maël Valais¹
IRIT (Institut de Recherche en Informatique de Toulouse)
²CNRS, ¹Université Paul Sabatier, Toulouse, France

{Olivier.Gasquet, Andreas.Herzig, Dominique.Longin, Frederic.Maris, Mael.Valais}@irit.fr

TouIST

(Toulouse Integrated Satisfiability Tool)

Un compilateur de langages logiques étendus
et de haut niveau vers des prouveurs
efficaces et indépendants



Exemple : le jeu de Nim



- Chacun son tour ($t \in T$), un joueur prend 1 ou 2 allumettes ($n \in A$)
- Celui qui ne peut plus prendre d'allumettes a perdu
- Nous jouons le rôle du **joueur 0** et l'adversaire est le **joueur 1**

Étape 1. Définir les règles du jeu (Formalisation des règles en logique)

Jouer. Selon son choix, le joueur laisse **une** ou **deux** allumettes en moins :

$$\bigwedge_{\substack{t \in T \\ n \in A \\ n \geq 2}} \left(\begin{aligned} &(\text{reste}(t, n) \wedge \neg \text{prend_2}(t) \rightarrow \text{reste}(t + 1, n - 1)) \\ &\wedge (\text{reste}(t, n) \wedge \text{prend_2}(t) \rightarrow \text{reste}(t + 1, n - 2)) \end{aligned} \right)$$

Dernier coup. S'il ne reste plus qu'**une** allumette, le joueur n'a pas le choix, il doit la prendre et ne laisse **aucune** allumette :

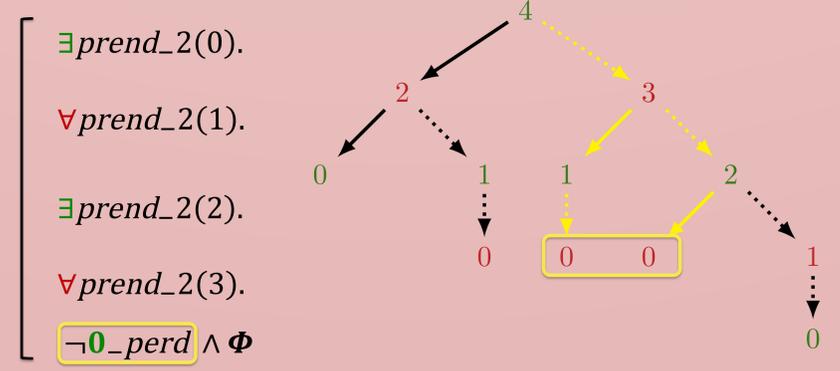
$$\bigwedge_{t \in T} (\text{reste}(t, 1) \rightarrow \neg \text{prend_2}(t) \wedge \text{reste}(t + 1, 0))$$

Perdre. Le **joueur 0** a perdu ssi à un tour t , c'est à lui de jouer et il ne reste **aucune** allumette :

$$0_perd \leftrightarrow \bigvee_{\substack{t \in T \\ t > 0}} (\text{tour_de_0}(t) \wedge \text{reste}(t, 0))$$

Étape 2. Jouer et gagner à tous les coups ! (Formalisation d'une stratégie gagnante à l'aide de QBF)

Quels coups le **joueur 0** doit-il faire pour **être sûr de gagner** quels que soient les coups du **joueur 1** ?



où Φ représente les règles du jeu présentées en étape 1

Spécificités

- Langage riche
- Édition visuelle
- Multiples solveurs (SAT, SMT, QBF)
- Facilement extensible
- Projet github, licence MIT

