TouIST (Toulouse Integrated Satisfiability Tool) : utilisation de la plateforme pour la planification

Khaled Skander Ben Slimane, Alexis Comte, Olivier Gasquet, Abdelwahab Heba, Olivier Lezaud, Frédéric Maris, and Maël Valais

IRIT - CNRS UMR 5505
Université Paul Sabatier – Toulouse 3
118 route de Narbonne, 31062 Toulouse Cedex 9
Frederic.Maris@irit.fr

Résumé : La plateforme de résolution TouIST (Ben Slimane *et al.*, 2015a,b) que nous avons développée, permet de coder facilement des problèmes combinatoires génériques sous la forme de bases de clauses ou de bases de contraintes et d'utiliser des solveurs SAT ou SMT pour les résoudre. Cet environnement permet donc de coder la résolution de problèmes de planification classique ou temporelle. Pour faciliter et automatiser cette résolution, nous avons développé un module externe PDDL2TouISTL qui permet de générer à partir d'un problème de planification classique ou temporelle décrit dans le langage PDDL, des règles permettant de décrire le codage de ce problème dans le langage de notre plateforme.

1 Démonstration de TouIST pour la planification

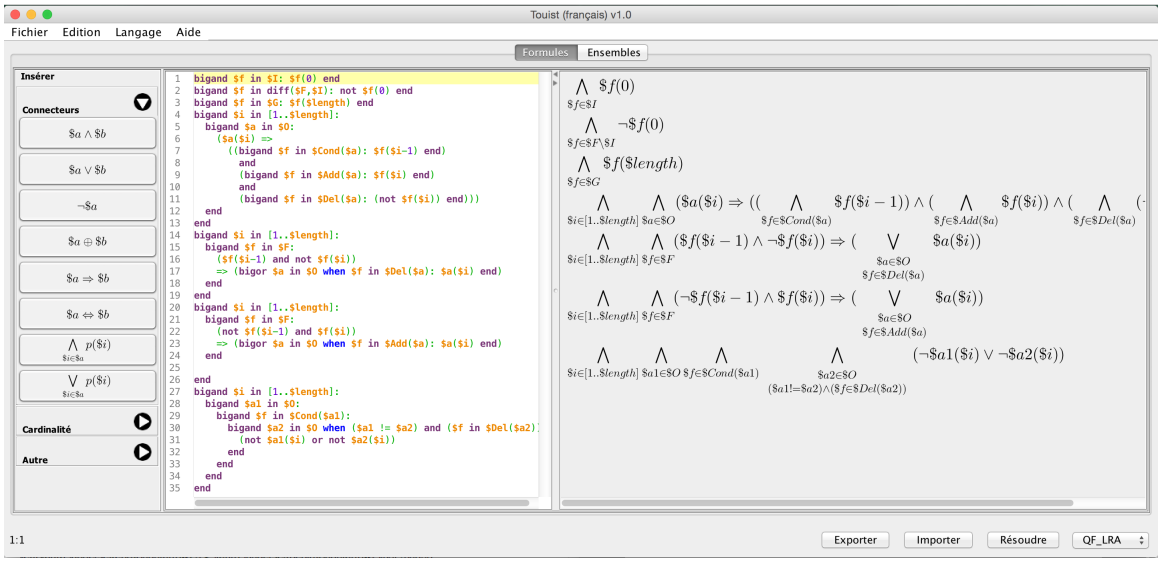

FIGURE 1 – Règles de codage (langage touistl à gauche et affichage LaTeX à droite).

La planification par satisfaction de base de clauses (planification SAT) a été introduite avec le planificateur SATPLAN (Kautz & Selman, 1992). Dans cette approche, on travaille directement sur un ensemble fini de variables propositionnelles. Deux actions identiques pouvant apparaître à des endroits différents d'un même plan doivent pouvoir être différenciées et on leur associe donc des propositions différentes. Comme on ne connaît pas à l'avance la longueur d'un plan solution d'un problème, on ne peut pas créer un codage unique permettant de le résoudre puisqu'il faudrait créer une infinité de variables propositionnelles pour représenter toutes les actions de tous les plans possibles. La solution la plus commune consiste alors à créer un codage représentant tous les plans d'une longueur k fixée. La base de clause ainsi obtenue est donnée en entrée à un

solveur SAT qui retourne, lorsqu'il existe, un modèle de cette base. Le décodage de ce modèle permet alors d'obtenir un plan-solution. Si la résolution du codage ne donne pas de modèle, la valeur de k est augmentée et le processus réitéré. Pour la complétude du procédé, tous ces modèles doivent correspondre exactement à tous les plans solutions d'une longueur fixée du problème. Notre plateforme de résolution de problèmes TouIST est capable de prendre en compte à la fois des formules logiques et des *ensembles de domaines*. Si notre objectif est de résoudre plusieurs problèmes de planification génériques, nous pouvons ainsi profiter de la flexibilité de TouIST qui permet à l'utilisateur de décrire une méthode générique de résolution avec des règles encodées comme formules et d'utiliser des ensembles de domaines pour décrire chaque problème de planification particulier. De nombreuses règles de codage pour la résolution de problèmes de planification ont déjà été proposées (Kautz & Selman, 1992; Mali & Kambhampati, 1999; Rintanen *et al.*, 2006). La figure 1 présente l'intégration de l'un de ces codages dans TouIST (codage dans les espaces d'états avec frame-axiomes explicatifs). La figure 2 illustre les ensembles de domaines définis par PDDL2TouISTL pour un problème particulier du célèbre monde des cubes "BlockWorld" et la figure 3 donne la solution renvoyée par le solveur.

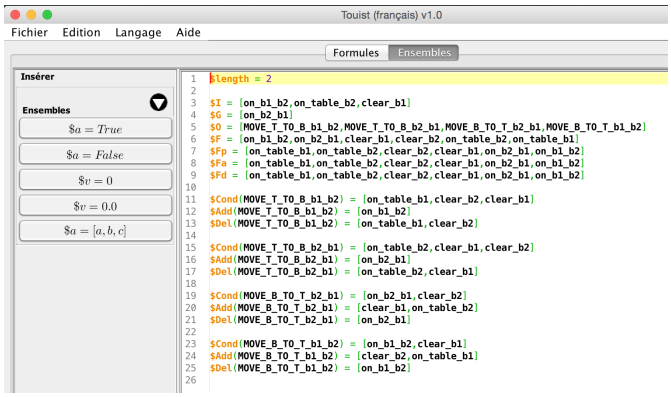

FIGURE 2 – Ensembles de domaines définis pour un problème de planification (BlocksWorld).

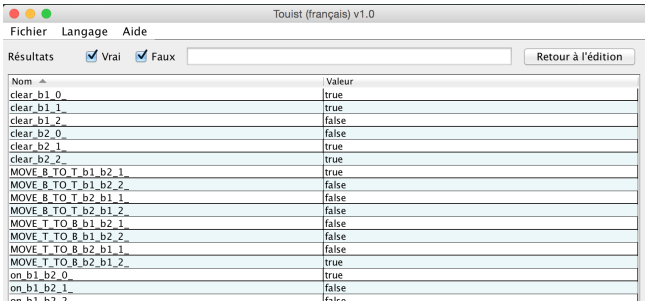

FIGURE 3 – Solution renvoyée par le solveur (BlocksWorld).

Pour résoudre des problèmes de planification réels, l'une des principales difficultés à surmonter consiste à prendre en compte la dimension temporelle. En effet, de nombreux problèmes du monde réel nécessitent, pour être résolus ou exécutés plus efficacement, la prise en compte de la durée des actions, des instants auxquels des évènements se produisent, ou encore la concurrence d'actions. En complément de SAT, notre plateforme TouIST est capable de gérer des théories comme la différence logique ou l'arithmétique linéaire sur les nombres entiers (QF_IDL, QF_LIA) ou les nombres réels (QF_RDL, QF_LRA), et d'appeler un solveur SMT pour trouver une solution. Pour être résolus, les problèmes de planification temporelle issus du monde réel nécessitent une représentation continue du temps, et donc, l'utilisation de nombres réels dans les codages logiques. TouIST peut être utilisé pour résoudre de tels problèmes impliquant des actions avec durée, des évènements exogènes et des buts temporellement étendus, par exemple avec les règles de codage proposées dans (Maris & Régnier, 2008).

Références

BEN SLIMANE S., COMTE A., GASQUET O., HEBA A., LEZAUD O., MARIS F. & VALAIS M. (2015a). La logique facile avec TouIST (formalisez et résolvez facilement des problèmes réel). In *Actes des 9es Journées d'Intelligence Artificielle Fondamentale (IAF'2015), 29 Juin-1er Juillet, 2015, Rennes, France*.

BEN SLIMANE S., COMTE A., GASQUET O., HEBA A., LEZAUD O., MARIS F. & VALAIS M. (2015b). Twist your Logic with TouIST. In *Proceedings of 4th International Conference on Tools for Teaching Logic (TTL'2015), June 9-12, 2015, Rennes, France*.

KAUTZ H. A. & SELMAN B. (1992). Planning as satisfiability. In *ECAI*, p. 359–363.

MALI A. D. & KAMBHAMPATI S. (1999). On the utility of plan-space (causal) encodings. In J. HENDLER & D. SUBRAMANIAN, Eds., *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22, 1999, Orlando, Florida, USA.*, p. 557–563 : AAAI Press / The MIT Press.

MARIS F. & RÉGNIER P. (2008). TLP-GP : new results on temporally-expressive planning benchmarks. In *20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2008), November 3-5, 2008, Dayton, Ohio, USA, Volume 1*, p. 507–514 : IEEE Computer Society.

RINTANEN J., HELJANKO K. & NIEMELÄ I. (2006). Planning as satisfiability : parallel plans and algorithms for plan search. *Artif. Intell.*, **170**(12-13), 1031–1080.