Titre : Widening pour les NDD
Mots clefs : Représentation symbolique par automates et polyèdres, Widening.
Encadrant :
Jérôme Leroux `jleroux@irisa.fr` +33 299 84 71 91
Bertrand Jeannet `Bertrand.Jeannet@irisa.fr` +33 299 84 74 72
Sujet :
La vérification d'un système et la synthèse de contrôleurs, se réduisent en général au calcul de l'ensemble des *états co-accessibles* d'un système, qui peuvent amener à un état considéré comme critique, et que l'on souhaite par exemple éviter.

Pour décrire et manipuler ces ensembles d'états en général infinis, une des méthodes est l'utilisation de représentations symboliques adaptées à la structure du système étudié. Ainsi, pour les systèmes utilisant des variables entières, différentes représentations symboliques ont été développée. Citons par exemple les contraintes linéaires ($y \geq x \wedge x = 3.z$) [CH78], permettant de représenter tout polyèdre convexe de vecteurs d'entiers, et les NDD [WB00], des automates représentant des ensembles réguliers d'entiers encodés par des chaines de bits, permettant de représenter n'importe quel ensemble définissable dans la logiques additive du première ordre sur les entiers ($\exists z.y = x + 2.z$). La première représentation est par exemple utilisée dans STG et NBAC, et la seconde dans FAST [Fas] et LASH [Las].

Comme l'ensemble des états co-accessibles d'un système correspondant au calcul d'un point fixe n'est en général pas calculable, un calcul itératif *exact* des états accessibles diverge en général. Pour obtenir une convergence, on utilise des techniques approchées permettant d'obtenir une *sur-approximation*. Par exemple, dans le cas des polyèdres convexes, on pourra utiliser le *widening* [CH78] qui utilise la structure *géométrique* des ensembles représenter pour en donner une sur-approximation.

Cette technique géométrique d'approximation des polyèdres n'a pas d'équivalent pour les NDD. En utilisant la structure particulière des NDD exposée dans [Ler04], l'objectif du stage est d'étendre le widening au NDD. Cette extension sera guidée par des expérimentations faites avec les outils FAST, STG et NBAC disponibles au sein de l'équipe.

Références

[CH78] Patrick Cousot and Nicholas Halbwachs. Automatic discovery of linear restraints among variables of a program. In *Conference Record of the Fifth annual ACM Symposium on Principles of Programming Languages*, pages 84–96. ACM, ACM, January 1978.

[Fas] FAST homepage. `http://www.lsv.ens-cachan.fr/fast/`.

[Las] LASH homepage. `http://www.montefiore.ulg.ac.be/~boigelot/research/lash/`.

[Ler04] Jérôme Leroux. A polynomial time presburger criterion and synthesis for number decision diagrams. Technical Report 1251, Département d'Informatique et de Recherche Opérationnelle. Faculté des arts et des sciences. Université de Montréal. Montréal, Quebec, Canada, september 2004.

[WB00] Pierre Wolper and Bernard Boigelot. On the construction of automata from linear arithmetic constraints. In *Proc. 6th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2000), Berlin, Germany, Mar.-Apr. 2000*, volume 1785 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2000.