

De Signal à AltaRica et inversement

Alain Griffault et Frédéric Herbreteau

(prenom.nom@labri.fr)

Mots-clés : Signal, AltaRica, Traduction de formalismes, Model-checking.

Sujet : Dans le cadre du projet ANR SPaCIFY, le thème "Modélisation et Vérification" du LaBRI a pour tâche le développement d'un outil de vérification pour le langage synchrone orienté flot de données Signal. L'équipe ESPRESSO de l'IRISA (qui développe Signal) propose l'outil de vérification formelle Sigali. Cependant, les partenaires du projet SPaCIFY souhaitent disposer d'un outil qui tire partie des techniques les plus modernes de vérification afin de pouvoir vérifier de très gros programmes comme les logiciels de vol des satellites. C'est pourquoi, il a été proposé de traduire les programmes Signal en modèles AltaRica afin d'utiliser l'outil de model-checking Arc développé au LaBRI. Cet outil bénéficie des dernières avancées, notamment la construction automatique de modèles abstraits et leur raffinement grâce à l'approche CEGAR (Counter Example-Guided Abstraction Refinement).

Le but du stage est de réaliser les tâches suivantes :

- proposer une traduction du langage Signal vers le formalisme AltaRica. Ceci comprend une définition formelle de la traduction, ainsi qu'une mise en œuvre d'un prototype assurant cette traduction.
- lorsqu'une propriété n'est pas satisfaite par le modèle AltaRica, l'outil Arc fournit un contre-exemple. La deuxième partie du stage consiste donc à traduire le contre-exemple AltaRica en un contre-exemple du programme Signal ayant servi à la construction du modèle AltaRica. Cependant, puisque Arc utilise l'approche CEGAR, le contre-exemple qu'il fournit est un chemin dans *l'abstraction du modèle AltaRica* qu'il a calculée et vérifiée. Il faut alors traduire ce contre-exemple abstrait en un contre-exemple concret du programme Signal de départ. On s'intéressera dans un premier temps au cas des contre-exemples AltaRica concrets avant de passer aux contre-exemples AltaRica abstraits.

Bibliographie :

1. A. Arnold, G. Point, A. Griffault and A. Rauzy. *The AltaRica Formalism for Describing Concurrent Systems*. Fundam. Inform., 40(2-3) :109-124, 1999.
2. A. Benveniste, P. Le Guernic and C. Jacquemot. *Synchronous programming with events and relations : the SIGNAL language and its semantics*. Science of Computer Programming, 16(2) :103-149, 1991.
3. P. Le Guernic, T. Gautier, M. Le Borgne and C. Le Maire. *Programming Real-Time Applications with Signal*. Proceedings of the IEEE, 79(9) :1321-1336, 1991.
4. E.M. Clarke, O. Grumberg, S. Jha, Y. Lu and H. Veith. *Counterexample-guided abstraction refinement for symbolic model checking*. Journal of the ACM, 50(5), pp.752-794, 2003.
5. A. Griffault, G. Point *On the partial translation of Lustre programs into the AltaRica language and vice versa*. Rapport interne LaBRI 1415-06