Sémantique Mécanisée et Compilation Vérifiée pour un Langage Synchrone à Flots de Données avec Réinitialisation

Lélio Brun*[1]

[1]Ecole Normale Supérieure Paris-Saclay (ENS Paris Saclay) – université Paris Dauphine, PSL Resarch University – 61 av du Pdt Wilson94230 Cachan, France

Résumé

Les spécifications basées sur les schémas-blocs et machines à états sont utilisées pour la conception de systèmes de contrôle-commande, particulièrement dans le développement d'applications critiques. Des outils tels que Scade et Simulink/Stateflow sont équipés de compilateurs qui traduisent de telles spécifications en code exécutable. Ils proposent des langages de programmation permettant de composer des fonctions sur des flots, tel que l'illustre le langage synchrone à flots de données Lustre.

Cette thèse présente Vélus, un compilateur Lustre vérifié dans l'assistant de preuves interactif Coq. Nous développons des modèles sémantiques pour les langages de la chaîne de compilation, et utilisons le compilateur C vérifié CompCert pour générer du code exécutable et donner une preuve de correction de bout en bout. Le défi principal est de montrer la préservation de la sémantique entre le paradigme flots de données et le paradigme impératif, et de raisonner sur la représentation bas niveau de l'état d'un programme.

En particulier, nous traitons le reset modulaire, une primitive pour réinitialiser des sous-systèmes. Ceci implique la mise en place de modèles sémantiques adéquats, d'algorithmes de compilation et des preuves de correction correspondantes. Nous présentons un nouveau langage intermédiaire dans le schéma habituel de compilation modulaire dirigé par les horloges de Lustre. Ceci débouche sur l'implémentation de passes de compilation permettant de générer un meilleur code séquentiel, et facilite le raisonnement sur la correction des transformations successives du reset modulaire.

*Intervenant