

Transcription de l'entretien audio avec Benjamin Lion

Je m'appelle Benjamin Lion et je travaille au centre Inria de l'Université de Rennes. J'étudie les effets de bord de programmes. C'est des effets qui ne sont pas fonctionnels, pas dans la logique, mais qui sont des conséquences de l'exécution d'un programme sur la mémoire ou sur son environnement. Parmi tous ces effets de bord, il y en a un qui est le temps. Un programme prend du temps pour s'exécuter.

Je m'intéresse à des méthodes pour comprendre ce temps qui s'exécute et à des domaines où avoir une maîtrise du temps est critique. Et un de ces domaines, par exemple, c'est, on va dire l'Internet des objets où on a un contrôleur qui interagit avec son environnement, avoir des instructions qui s'exécutent à un temps précis. C'est crucial.

Aujourd'hui, on n'arrive pas à prouver formellement. On a des méthodes d'analyse, mais de façon générique. Quand un programme préserve des propriétés de temps sa compilation, on a du mal à avoir une méthode générique. Alors il y a des dysfonctionnements qui sont causés par le fait qu'on n'arrive pas à comprendre précisément comment le temps évolue lors de l'exécution. Mais il y a aussi une ambiguïté pour le développeur lorsqu'il programme pour savoir a priori quel va être le comportement de son programme et donc pour l'instant, il n'y a pas moyen d'avoir une sécurité sur l'exécution de son programme, a priori.

J'ai perçu une bourse de post-doctorat Marie Curie qui va me permettre d'étudier pendant deux ans des moyens pour certifier le temps d'exécution du programme. Alors, il y a deux défis principaux. Le premier, c'est arriver à exprimer le temps dans une sémantique, ce qui permettrait d'étudier la sémantique d'un programme en incluant des aspects temporels et de déduire des propriétés de temps d'un programme par rapport à des propriétés temps de sous-parties.

Le deuxième objectif, c'est de préserver ces propriétés à travers une chaîne de compilation, ce qui permettrait de prouver des propriétés sur un programme et d'avoir ces propriétés qui sont maintenues parce que le programme est exécuté. Je ne vais pas créer le compilateur de toute pièce, mais je vais me baser sur un travail qui a déjà été développé, conséquent ; à Inria, qui est la certification d'un compilateur pour le langage C. Et ce compilateur a été formalisé dans un assistant de preuve et plusieurs propriétés ont été prouvées, dont une de préservation de sémantique. C'est-à-dire que le programme qu'on va exécuter après compilation a le même comportement que le programme qu'on a écrit en C.

Le compilateur certifié s'appelle CompCert et l'assistant de preuve s'appelle Coq. Et c'est un grand domaine de recherche et aussi un grand domaine d'application. Toutes les applications qui sont sensibles aux temps auraient besoin d'une certification pour assurer que leur programme s'exécute dans un temps précis. Donc, des systèmes comme l'industrie connectée où les programmes s'exécutent à des temps précis pour opérer sur une machine ou des systèmes plus critiques qui doivent interagir avec la physique, doivent inclure dans leur compilation des certifications de préservation de propriété temporelle.