

Vers un système de types pour un fragment de JavaScript

Encadrement & Contact : Xavier RIVAL

e-mail : rival@di.ens.fr

tél : 01 44 32 21 50

Description de l'objet du mémoire :

D'une manière générale, les systèmes de types ont pour but d'éliminer certaines classes d'erreurs ("Well typed do not go wrong", R. Milner) et de faciliter le raisonnement sur les programmes, que ce soit pour les compiler (en fournissant au compilateur des informations sur le format des données, et en permettant ainsi des optimisations) ou pour s'assurer de leur correction. Cela explique le succès de langages fortement typés.

Néanmoins, certains langages ont été développés sans système de type, le plus souvent pour conserver une certaine flexibilité. C'est le cas par exemple du langage JavaScript très utilisé dans le domaine du Web. De tels langages donnent une grande liberté aux programmeurs, mais ne permettent pas de garantir de propriétés fortes sur les programmes, ce qui est la source de problèmes de performance, de correction, et de sécurité. De plus, il est extrêmement difficile de rajouter des informations de type à un tel langage *a posteriori*, sans rejeter la plupart des programmes existants même si cela pourrait être souhaitable pour améliorer la qualité des programmes futurs.

Le but de ce sujet est d'étudier de nouvelles techniques de typage pour définir des informations de type utiles pour une proportion significative des programmes JavaScript. En particulier, la technique de *typage graduel* permet de donner un type à des fragments de programmes tandis que d'autres parties ne peuvent pas recevoir de type précis. Toutefois, la preuve de correction de tels systèmes est en général assez difficile. Nous proposons d'étudier l'application de cette technique au système *TypeScript*, qui permet de typer de nombreux programmes JavaScript, sa preuve de correction et ses limites.

Références bibliographiques :

- 1 Aseem Ratsogi, Nikhil Swammy, Cédric Fournet, Gavin Bierman, and Panagiotis Vekris.
Safe and Efficient Gradual Typing for JavaScript.
Symposium of Principles Of Programming Languages (POPL), ACM 2015.