

Titre : Implémentation du sous-typage dans un générateur de mappings Tom¹-Java

Équipe : PAREO

Contact : Jean-Christophe Bach, jeanchristophe.bach@inria.fr

Mots-clef : langage Tom, Java, Gom, type, sous-typage, structure algébrique, génération de code, arbre

Sujet :

Le langage Tom est un langage fonctionnant avec des langages hôtes (Java, C, Python, *etc*) pour les enrichir. Nous ajoutons donc des constructions dédiées fournissant des fonctionnalités spécifiques. Parmi elles, nous retiendrons les deux suivantes :

- le *pattern-matching* (ou filtrage de motifs) : cette fonctionnalité peut-être vue comme une généralisation de la construction *switch-case* que l'on retrouve dans beaucoup de langages. Contrairement à la plupart des autres langages qui implémentent cette construction valide pour les entiers seulement, le langage Tom permet de filtrer sur n'importe quelle structure de données.
- les *mappings* : il s'agit d'un mécanisme permettant de représenter une structure de données externe (*e.g.* objet Java) dans le monde algébrique de Tom. Cela nous permet entre autres d'utiliser la fonctionnalité de filtrage décrite ci-dessus sur des objets quelconques.

Afin d'améliorer la confiance que le développeur peut avoir dans le code qu'il écrit, nous avons intégré la fonctionnalité de sous-typage dans le langage Tom. Nous pouvons donc décrire et manipuler des termes en fonction de leur type et de leur(s) surtype(s). Ce mécanisme existe dans d'autres langages tels que Java.

Pour faciliter le travail du développeur, nous disposons d'un outil de génération automatique de mappings et de code Java : Gom. À partir d'une description simple d'une structure de données algébriques, il permet de générer l'implémentation Java correspondante ainsi que les mappings utilisables dans un programme Tom+Java.

Nous souhaitons donc améliorer l'outil Gom en y intégrant la fonctionnalité de sous-typage pour pouvoir exprimer les mêmes relations entre les types que dans le langage Tom. Il faudra dans un premier temps apprendre à utiliser le langage Tom. Ensuite, l'outil Gom devra être étudié en vue de l'implémentation du sous-typage. Enfin, le candidat mettra en œuvre le sous-typage dans Gom (parser, compilateur et générateur de code).

En cas d'avancée rapide sur le sujet, nous proposons de continuer à étendre l'outil en ajoutant des *backends* (générateurs de code) pour d'autres langages tels que Ada. Il s'agira là de reprendre un *proof of concept*, de vérifier sa conception et de le passer en phase de production.

Contexte :

Le stage se déroulera au sein de l'équipe PAREO² dirigée par PierreÉtienne Moreau.

Prérequis : Le candidat devra maîtriser la langage Java.

Connaissances utiles :

- antlr3
- système de gestion de code source Git
- ant

1. <http://tom.loria.fr>

2. <http://pareo.loria.fr>