TELECOM Nancy

Projet Compilation

Rapport d'activité n°2

Thibault Boisseau
Nathan Cornélie
Noé-Laurent Laurent
Maxime Wirth

Responsable de module :
Suzanne Collin
Sébastien Da Silva

1 Introduction

Cette partie du projet se concentrait sur la création de l'arbre syntaxique abstrait (appelé AST ensuite) et la table des symboles. Nous n'avons pour l'instant travaillé que sur l'AST. Nous avons donc ajouté des labels à la grammaire, puis rempli les fichiers AstCreator.java et GraphVizVisitor.java.

2 Ajout des labels

Nous avons donc commencé par ajouter des labels à certaines règles de la grammaire. Nous l'avons fait pour chaque règle ayant plusieurs dérivations possibles. Puis, en recompilant la grammaire avec cette fois-ci l'option -visitor, les visiteurs ont été créés pour la création de l'AST, nous devions donc les remplir.

3 Remplissage d'AstCreator.java

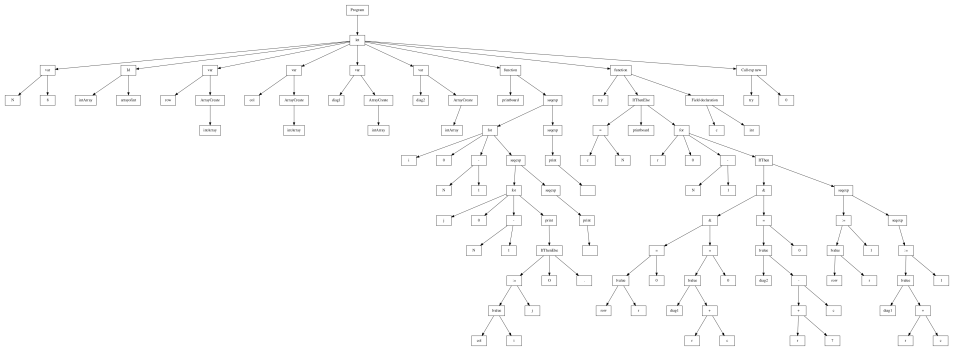
Après avoir compilé la grammaire, toutes les signatures des fonctions nécessaires étaient présentes dans exprBaseVisitor.java. Nous devions donc les implémenter dans un fichier AstCreator.java. Les fonctions avaient comme nom visit"NomRègle", et définissaient comment les noeuds de l'arbre étaient créés. Nous avons dû réfléchir à la structure de notre arbre pour savoir comment traiter chaque règle, si la règle devait créer ou non un noeud dans l'AST. Chaque règle pour lesquelles on souhaitait un nouveau noeud, il a fallu créer une nouvelle classe du nom de la règle. Dans le cas de règle pouvant amener au mot vide, nous utilisions des attributs de type $Optional < Ast >$.

4 Création de toutes les classes java

Afin de créer l'arbre abstrait, de nombreuses classes java ont été implémenté et cela, pour chaque label présent dans la grammaire ainsi que pour chaque règle qui n'en présentait pas.Toutes les classes implémentent l'interface Ast permettant l'utilisation de la méthode accept sur chacune des classes. Le constructeur de chacune de ces classes était construit de tels sorte qu'il y ai un paramètre pour chaque non terminal de la règle.

5 GraphVizVisitor.java

Une fois toutes les classes implémentées et l'Ast créé, il nous désormais avoir la possibilité de l'afficher. Ce rôle est rempli par GraphVizVisitor.java qui va permettre la création d'un fichier .dot représentant l'Ast qui sera alors visible dans un autre fichier .svg grâce à une commande. Pour chacune des classe décrite plus tôt on implémente une fonction visit qui va créer chacun des noeuds.



Arbre abstrait