

Projet : Compilateur SLIP vers NBC
IHDC B332 - Théorie des langages : Syntaxe et sémantique

Nicolay Matthias
Evrard Cédric

UNamur

1 Introduction

Dans ce rapport, nous allons présenter les différents aspects techniques du projet de syntaxe et sémantique ainsi que les choix techniques que nous avons effectués lors de la création du compilateur SLIP.

2 Démarche générale

Notre groupe était composé de deux personnes. La démarche que nous avons adopté a été de laisser Cédric commencer à travailler en chaque début de partie afin de créer l'architecture de l'application ainsi que la structure des différents packages. Une fois que cette structure était créée, une répartition était faite du travail.

Il a parfois été difficile de découper la réalisation du travail de manière équitable. En effet, une dépendance assez importante dans la grammaire de la partie *expression droite* a fait que celle-ci devait souvent être réalisée avant tout le reste et conditionnait énormément la structure du projet.

Avant la fin de la phase 2, un gros travail de refactoring a été effectué afin de rendre le code plus lisible et d'en faciliter la maintenance.

3 Structure de la table des symboles

La structure utilisée se trouve dans la classe *Context.java*.

La "Table des symboles" en elle-même est composée de quatre *Hashtables* qui sont accompagnées d'un *array* pour chaque symbole. Les quatre symboles différents sont :

- VariableBase
- Function
- Record
- Enumeration

Notre choix d'utiliser quatre *Hashtable* est simple, notre point de vue est qu'il y a quatre classe de symboles. Ainsi donc pour avoir une simplicité de codage, de stockage et d'accès aux symboles, nous avons utilisé cette implémentation.

Notre "Table des symbole" est donc utilisée comme suit:

- Lors du passage de notre visiteur, chaque symbole est stocké dans l'*array* assigné à son symbole et référencé dans sa *Hashtable*. Le tout dans le contexte dans lequel se trouve ce symbole.
- Lors du "test" du code, chaque *Function* est vérifiée par la validité de son type de retour et de la bonne présence des *VariableBase* dans leur contexte respectif pour qu'elles respectent leurs portées. Cette action est facilitée par la présence des *Hashtable*
- Chaque symbole est donc restitué en code NBC en conservant chaque fonctionnalité du code *Slip* entré dans le compilateur.

Ceci est donc le résumé de l'implémentation de notre "Table des symboles".

4 Architecture du compilateur

5 Conclusion

References