TELECOM Nancy

Projet Compilation

Rapport d'activité n°1

Thibault Boisseau
Nathan Cornélie
Noé-Laurent Laurent
Maxime Wirth

Responsable de module :
Suzanne Collin
Sébastien Da Silva

1 Introduction

Le but de ce projet est de réaliser un compilateur pour le langage CHAOS. Dans ce rapport, nous traiterons la partie de la construction de la grammaire de ce langage ainsi que quelques optimisations. Notamment, nous avons supprimé la récursivité gauche de la grammaire, et avons traité les priorités d'opérations. Finalement, nous avons rendu notre grammaire LL(1), et créé certains exemples pour tester si notre grammaire fonctionnait correctement.

2 Construction de la grammaire

Nous avons d'abord suivi la 1ère version de la documentation du langage CHAOS fournie avec le sujet. Cette première version possédait plusieurs problèmes. Elle était très récursive gauche, et en outre ne reconnaissait pas certaines parties des exemples du document.

Lors de l'envoi de la 2ème version, nous avons revu la structure de la grammaire pour correspondre à la nouvelle spécification. Cette opération a déjà enlevé plusieurs problèmes que nous avions au début. Cependant, celle-ci n'était pas LL(1). En effet, notre grammaire possédait encore plusieurs cas de récursivité gauche directe et indirecte, ne reconnaissait pas certains opérateurs et pouvait être factorisée.

3 Correction de la grammaire

D'abord, nous avons enlevé les cas de récursivité gauche directes, qui concernait en tout 5 règles. Nous avons appliqué l'algorithme vu en cours de TLA de l'année dernière.

Puis nous avons cherché à implémenter les priorités des opérations. En effet, nous avons rencontré un problème avec une de nos règles, ce qui bloquait notre avancée. La règle pour les opérations binaires s'écrivait :

$$expr \rightarrow expr\ BINARY\_OPERATOR\ expr$$

avec $BINARY\_OPERATOR$ qui renvoyait vers chacun des opérateurs binaires. Avec cette grammaire, les caractères '=' et '-' n'étaient jamais reconnus. Après avoir géré les priorités des opérations, ces problèmes avaient disparus, et avaient aussi supprimé le cas de récursivité gauche indirecte.

Ensuite subsistaient des non-terminaux ayant différentes règles avec les mêmes préfixes. Nous avons appliqué un algorithme classique de factorisation pour régler ce problème. Notre grammaire est donc finalement LL(1).

4 Exemples

Pour concevoir des exemples, nous avons déjà repris les exemples bons qui sont présents dans la documentation. Ensuite, nous avons cherché à créer des

exemples avec des erreurs, qu'elles soient lexicales ou syntaxiques. En tout une quinzaine de tests ont été implémentés. Nous testons par exemple les différentes opérations pour visualiser leur priorité, et les différents mots-clés.

Problème : nous n'avons pas réglé l'erreur en cas d'égalité a=b=c qui doit plutôt s'écrire a=(b=c).