

J1IN7M13: Compilation Project

Due on Monday, January 1, 2012

CLEMENT Lionel

VER VALEM Willian , BAYOL Elmer , YOGA Josue , BOULMEDAIS Zaid

Contents

Introduction	3
Table des symboles	3
Implémentation dans le parser	3
Arbre de syntaxe abstraite	3
Vérification des types	3

Introduction

Dans ce second devoir, il nous est demandé de réaliser une table des symboles, de procéder à la vérification de types, de représenter les structures de contrôle ainsi que de calculer les coercition et les surcharges. Malheureusement, par manque de temps, nous avons pu implémenter la table des symboles ainsi que l'arbre de syntaxe abstraite ainsi qu'un début d'implémentation de la vérification des types sans pouvoir effectuer le reste du travail demandé.

Table des symboles

Nous avons choisi de créer 3 classes afin de représenter les différents objets (ClassType, Function, Variable). Dans ClassType et Function, nous avons ajouté un environnement afin de pouvoir traiter correctement les déclarations, en effet ces deux structures contiennent du code qu'il faut être capable d'exécuter. Variable ne contient que les valeurs ainsi que leur type.

Implémentation dans le parser

Dans le parser, nous avons 3 variables de type Env, l'une d'entre elles est la variable a proprement parler, les deux autres sont des pointeurs (un sur le bas de la pile, le second sur la position actuelle). Ces différentes variables sont nécessaires car il faut pouvoir travailler sur des parties non-accessibles de la pile.

Arbre de syntaxe abstraite

Nous avons utilisé la classe Stree proposée pour le projet. Il a fallu implémenter l'enumtag correspondant aux opérations autorisées par le langage. Nous avons choisi de ne travailler qu'avec des arbres binaires en spécifiant la construction des arbres plus complexes lorsque cela est nécessaire (de type if, while, for ...). Pour pouvoir travailler sur l'arbre de syntaxe abstraite, il a été nécessaire de changer les RESULT du parser car nous n'avons pas trouvé comment retourner de multiples éléments (une solution possible serait de créer un objet regroupant les retours qui serait renvoyé). Lors des tests de notre travail, nous avons rencontré des difficultés avec la création du fichier de visualisation de l'arbre (.dot).

Vérification des types

Cette partie n'est pas très avancée mais nous avons implémenté les méthodes toString, unify et la classe typeDiff. L'algorithme unify nous a posé de gros problèmes, nous avons donc choisi de nous concentrer sur les autres parties du devoir.