Le 04/01/2015

Projet de compilation

**Sujet**: Réalisation d’un compilateur pour le langage dot.

**Réalisé par**: Ibrahima FALL

Abderrahmen DRIDI

Sommaire

Introduction 3

I – Front-End 4

II- Middle-End 4

III- Back-End 4

Conclusion 5

# Introduction

Le projet de compilation, consistait en la mise en œuvre d’une solution de représentation graphique suite à la lecture d’un fichier au langage dot.

Il a été réalisé en 3 parties :

* Le front end : où a été implémenté les différentes types nécessaires à l’implémentation finale. ( ie : typage, lexer, parser ), son but était la reconnaissance d’un programme en langage dot ainsi que sa vérification syntaxique.
* Le middle end : où a été implémenté les différentes fonctions nécessaire à la récupération du ficher dot et de son traitement.
* Le back end : affichage du graphe en utilisant le code généré, du html ainsi que du javascript.

# I – Front-End

Cette partie se décompose en 3 sous parties :

* **Le typage**: implémentation du type « graph » ainsi que du type « stmt ». Cela a permit de base pour les fichiers lexer et parser où l’analyse lexicale et syntaxique on lieu.
* **Le lexer**( ou analyse lexicale ) : Le rôle du lexer est de réalisation une analyse lexicale. Il doit donc contenir toutes les définitions de mot clé et de structure pouvant être rencontré au cours de la lecture du langage étudié.
* **Le Parser**(ou analyse syntaxique) : Son rôle est d’assurer que la syntaxe du fichier écrit en langage dot est correcte. Il permet de vérifier la suite logique des informations contenues dans le fichier.

# II- Middle-End

Ici, la mise en forme du graphe aura lieu. Elle permet de faire une transition entre les informations obtenues depuis le lexer et le parser à la présentation d’un graphe.

Entres autres, il permet de retourner le nombre de nœuds et d’arêtes en ayant obtenu au préalable un arbre de syntaxe abstraite. Ici il faudra faire bien attention à gérer tous les cas possibles lors du parcours de l’arbre syntaxique.

# III- Back-End

Dernière étape logique de la réalisation du projet ; il permet d’afficher le graphe ainsi obtenu suite aux deux premières parties.

Le but de cette partie est la production d’une page HTML contenant du javascript. Il permet entres autres de générer un élément SVG à l’aide de la bibliothèque D3.js contenant toute les informations nécessaires à l’affichage final du graphe.

# Conclusion

La réalisation du projet a permit d’étudier tous les aspects entrant en jeu lors de la réalisation d’un compilateur. Pour des choix pratique il a été entièrement réalisé avec le langage OCAML permettant plus de souplesse avec le typage, le lexer, le parser ainsi que la construction des fonctions. Notons que le langage C serait plus adapté pour la génération du code HTML ; ce serait donc une manière d’améliorer le code ou même les performances.