

Projet de compilation

Le but de ce projet de compilation est d'implémenter un mini-compilateur réalisant les différentes phases de la compilation à savoir l'analyse lexicale en utilisant l'outil FLEX et l'analyse syntaxico-sémantique en utilisant l'outil BISON. Les traitements parallèles concernant la gestion de la table des symboles ainsi que le traitement des différentes erreurs doivent être également réalisés lors des différentes phases d'analyse du processus de compilation.

Ainsi il vous est demandé de définir votre propre langage sur lequel le projet va porter. La définition du langage doit au minimum comporter les aspects suivants :

- Structure générale du programme.
- Commentaires.
- Déclaration de variables simples et structurées.
- Définition des différents types (appellation, plage de valeurs,...etc.).
- Instructions de base (affectation, conditions, boucles, entrée/sortie,...etc.).
- Opérateurs logiques, de comparaison, et arithmétiques.
- les différentes priorités et règles d'associativité.

L'analyse Lexicale avec l'outil FLEX a pour but d'associer à chaque mot du programme source la catégorie lexicale à laquelle il appartient. Pour cela, il est demandé de définir les différentes entités lexicales à l'aide d'expressions régulières et de générer le programme FLEX correspondant.

L'analyse syntaxico-sémantique avec l'outil BISON se base sur la grammaire qui génère votre langage de programmation. La grammaire associée doit être LALR. En effet l'outil BISON est un analyseur ascendant qui opère sur des grammaires LALR. Il faudra spécifier dans le fichier BISON les différentes règles de la grammaire ainsi que les règles de priorités pour les opérateurs afin de résoudre les conflits. Les routines sémantiques doivent être associées aux règles dans le fichier BISON.

(Selon l'avancement du cours) La génération du code intermédiaire permet de transformer le code source en ensemble d'instructions élémentaire proche du code objet sans être pour autant spécifique à une machine donnée. Il vous est demandé de baser votre transformation sur le format quadruplets.

La table de symboles doit être créée lors de la phase de l'analyse lexicale. Elle doit regrouper l'ensemble des variables et constantes définies par le programmeur avec toutes les informations nécessaires pour le processus de compilation. Cette table sera mise à jour au fur et à mesure de l'avancement de la compilation. Il est demandé de prévoir des procédures pour permettre de rechercher et d'insérer des éléments dans la table de symboles. Les variables structurées de type tableau doivent aussi figurer dans la table.

Il est aussi demandé d'afficher les messages d'erreurs adéquats à chaque étape du processus de compilation. Ainsi, lorsqu'une erreur lexicale ou syntaxique est détectée par votre compilateur, elle doit être signalée le plus précisément possible, par sa nature et sa localisation dans le fichier source. On adoptera le format suivant pour cette signalisation : File "Test", line 4, character 56: syntax error

Il vous est demandé de vous constituer en équipe de 4 étudiants faisant partie du même groupe. Les délais prévisionnels des livrables sont définis comme suit (seront amenés à évoluer éventuellement selon l'avancement du cours)

Livable 1 : liste des groupes	9/11 (inclus)
Livable 2 : proposition du langage	13/11 (inclus)
Livable 3 : analyse lexicale	30/11 (inclus)
Livable 4 : analyse syntaxique	21/12 (inclus)
Livable 5 : analyse sémantique et code objet	15/01 (inclus)
Livable 7 : démonstration finale	TBA

Les livrables contiennent le code source ainsi qu'un rapport mettant en exergue les principaux aspects traités. Les différents livrables doivent être transmis par e-mail à l'enseignant. Ils peuvent aussi faire l'objet de vérification durant les séances de TD/TP.