

Travaux Dirigés d'analyse syntaxique n°5

Licence d'informatique

Traduction de séquences en arbres

L'objectif de ce TD est d'implémenter un traducteur qui accepte des expressions multiplicatives et les traduit en arbres.

► Exercice 1. *Jeu d'essais*

En prévision des tests des autres exercices, préparez au moins un fichier de test positif, c'est-à-dire qui contient une expression multiplicative correcte, et au moins un fichier de test négatif avec une erreur. Les opérateurs sont `*`, `/` et `%` et les opérandes sont les identificateurs en langage C, conformément à la grammaire suivante :

$$\begin{cases} E \rightarrow T \\ T \rightarrow T \text{ divstar } id \mid id \end{cases}$$

► Exercice 2. *Analyseur lexical*

Implémentez un analyseur lexical pour la grammaire de l'exercice 1. (On ne peut pas le tester sans avoir fait l'exercice 3.)

► Exercice 3. *Analyseur syntaxique*

Ajoutez à votre analyseur lexical un analyseur syntaxique pour la grammaire de l'exercice 1 et testez avec le jeu d'essais de l'exercice 1.

► Exercice 4. *Construction d'arbres de dérivation*

Ajoutez à votre analyseur syntaxique des actions qui construisent un arbre de dérivation et qui l'affichent une fois qu'il est fini. Utilisez le module `tree.c` fourni avec le sujet de TD. Adaptez votre `makefile` (parce que `tree.c` et `tree.h` sont des sources : vous aurez à les changer manuellement en travaillant sur votre projet) et testez.

► Exercice 5. *Construction d'arbres abstraits*

Implémentez une nouvelle version de votre traducteur qui construit un arbre abstrait sans les nœuds `E` et `T`. Testez.