

Analyse Syntaxique - Année 2010-2011

Feuille 3

Construction d'analyseurs syntaxiques avec yacc

Exercice 3.1

Soit la grammaire :

$$\begin{array}{lcl}
 E & \rightarrow & T + E \quad | \quad T \\
 T & \rightarrow & F * T \quad | \quad F \\
 F & \rightarrow & (E) \quad | \quad \text{id}
 \end{array}$$

Le texte suivant représente la même grammaire en format accepté par *Yacc* :

```

%token    NUM
%%
expr      : terme '+' expr
          | terme
          ;
terme     : facteur '*' terme
          | facteur
          ;
facteur   : '(' expr ')'
          | NUM
          ;
%%
#include <stdio.h>
#include "lex.yy.c"

int main(int argc, char *argv[]){
    yyparse();
}

void yyerror(char *s) {
    printf ("Erreur syntaxique: %s\n",s);
}

```

1. Ecrire le fichier `lex` associé. Compiler, en utilisant l'option `-d` de `yacc`. Que contient le fichier `y.tab.h`?
2. Modifier le fichier de telle sorte que l'analyseur syntaxique affiche toutes les règles de grammaires utilisées au cours de l'analyse, sous la forme `E -> E + T` , ...
3. Utiliser l'affichage obtenu pour reconstituer (sur papier) les arbres syntaxiques des expressions `2+3` , `5*(2+3)` .

Exercice 3.2

Reprendre la grammaire de l'exercice précédent.

1. Rajouter des règles à la grammaire, pour pouvoir analyser des expressions comportant des soustractions et des divisions (on ne s'occupera pas pour l'instant des divisions par zero).
2. Tester votre programme pour l'expression $4 - 2 + 3$. Corriger la grammaire, si nécessaire.

Exercice 3.3

On se propose d'analyser des affectations avec les règles de grammaire suivantes :

$$\begin{aligned} \textit{Start} &\rightarrow \textit{Expression} \textit{ ';' } \\ &| \textit{Assign} \\ \textit{Expression} &\rightarrow \textit{ID} \textit{ ',' } \textit{Expression} \\ &| \textit{ID} \\ \textit{Assign} &\rightarrow \textit{ID} \textit{ '=' } \textit{Assign} \\ &| \textit{ID} \textit{ '=' } \textit{NUM} \textit{ ';' } \\ &| \textit{'*'} \textit{ID} \textit{ '=' } \textit{Assign} \\ &| \textit{ID} \textit{ ';' } \end{aligned}$$

1. Ecrire les fichiers lex et yacc correspondants. Compiler avec l'option $-v$ de yacc. Que constatez-vous ? D'après les informations contenues dans le fichier `y.output`, sur quel symbole se produit l'erreur ? Essayer de modifier la grammaire.
2. On se propose maintenant d'utiliser les règles de grammaire suivantes :

$$\begin{aligned} \textit{Expression} &\rightarrow \textit{LeftHandSide} \textit{ ASSIGNOP } \textit{RightHandSide} \\ &| \textit{RightHandSide} \\ \textit{LeftHandSide} &\rightarrow \textit{'*'} \textit{ID} \\ &| \textit{ID} \\ \textit{RightHandSide} &\rightarrow \textit{Expression} \\ &| \textit{ID} \\ &| \textit{NUM} \end{aligned}$$

Ecrire le fichier yacc correspondant. Compiler avec l'option $-v$ de yacc. Que constatez-vous ? L'erreur est-elle la même qu'à la question précédente ? D'après les informations contenues dans le fichier `y.output` quelles sont les règles de la grammaire qui posent problème ?

3. essayer d'autres grammaires, permettant de reconnaître les mêmes expressions.
4. Etendre la grammaire pour que la partie gauche de l'affectation puisse être un accès à un élément d'un tableau, ou un accès à un champ d'une structure.