

# Théorie des Langages et de la Compilation (Partie pratique)

## Énoncés de la séance 12 (Yacc / Bison)

S. Collette                      G. Geeraerts

Année académique 2003-2004

### Exercice : manipulation de polynômes

Vous avez reçu une spécification Lex et une spécification Yacc (voir feuilles annexes).

1. Décrivez en français le langage reconnu par le compilateur qu'on obtiendrait si on compilait ces spécifications ;
2. Modifiez la spécification pour n'accepter que les polynômes d'une seule variable. On entre un polynôme par ligne, et chaque polynôme peut porter sur une variable différente ;
3. Ajouter le code nécessaire pour afficher la dérivée première du polynôme. Exemple d'utilisation :

```
2x^3+2x^2+5
Derivee premiere: 6x^2+4x
```

4. Ajouter la possibilité de reconnaître des produits de polynômes. Le symbole de multiplication \* devra être précisé explicitement. Adapter les actions qui affichent la dérivée. Exemple d'utilisation :

```
(3x^2+6x)*(9x+4)
La derivee premiere de (3x^2+6x)*(9x+4) est ((3x^2+6x)*(9))+((6x+6)*(9x+4))
```

5. Ajouter la possibilité d'évaluer le polynôme et sa dérivée première en un point donné. L'utilisateur entrera la valeur de la variable, suivie de ; puis du polynôme (le tout sur une même ligne). Par exemple :

```
2 ; (3x^2+6x)*(9x+4)
La derivee premiere de (3x^2+6x)*(9x+4) est ((3x^2+6x)*(9))+((6x+6)*(9x+4))
p(2)=528 , p'(2)=612
```

# Annexe

## Spécification Lex

```
/* ***** */
/* *  Théorie des Langages et de la Compilation * */
/* *                                     * */
/* *          TP12: Yacc et Bison          * */
/* *                                     * */
/* *          Spécification Lex          * */
/* ***** */
```

```
chiffre [0-9]
lettre [a-zA-Z]
entier {chiffre}+
var {lettre}+
%{
#include "derive.tab.h"
#define yywrap() 1
}%

{entier} {return ENTIER ;}
{var}    {return VAR ;}
" "      {}
.        {return yytext[0] ;}
"\n"    {return yytext[0] ;}

%%
```

## Spécification Yacc

```
/* ***** */
/* * Théorie des Langages et de la Compilation * */
/* * * * * * */
/* * TP12: Yacc et Bison * */
/* * * * * * */
/* * Spécification Yacc * */
/* ***** */

%token ENTIER
%token VAR

%left '+' '-'
%left '*' '/'

%%

input : line input      {}
      | line            {}
      ;

line : polynome '\n' {printf("OK\n")} ;
      ;

polynome : polynome '+' terme {}
          | polynome '-' terme {}
          | terme {}
          ;

terme : '-' terme      {}
       | VAR '^' ENTIER {}
       | ENTIER VAR '^' ENTIER {}
       | VAR           {}
       | ENTIER VAR    {}
       | ENTIER        {}
       ;

%%

int main (void)
{
    yyparse() ;
}

int yyerror(char * s)
{
    printf("yyerror: I encountered an error: %s.\n",s) ;
}
```