Analyse Syntaxique et Compilation
TD11 de la semaine du 15 au 21 Décembre 2000

Programmes en Yacc

Exercice 1. Nombre minimum de registres

Dans la procédure de génération de code donnée la semaine dernière en Yacc (`etfcode.y`) l'algorithme appliqué est le suivant :

Le code de l'expression $exp1 \ + \ exp2$ (ou de $exp1 \ * \ exp2$) est obtenu en concaténant

- le code de $exp1$
- le translaté du code de $exp2$ (*i. e.* on incrémente les numéros de registres de ce code)
- l'instruction `Add(R1,R2)` (ou `Mult(R1,R2)`)

Cette technique peut donner lieu à l'utilisation d'un très grand nombre de registres alors qu'on pourraitt faire beuacoup mieux Ainsi par exemple l'expression :

$$a * (b + c * (d + e * (f + g)))$$

aura besoin de 7 registres alors qu'on pourrait la calculer par le programme suivant qui n'en utilise que 2 :

```
Load(f,R1); Load (g,R2); Add(R1,R2); Load(e,R2); Mult(R1,R2); Load(d,R2); Add(R1,R2);
Load(c,R2); Mult(R1,R2);  Load(b,R2); Add(R1,R2); Load(a,R2); Mult(R1,R2);
```

Le programme ci-dessus a été obtenu en appliquant l'algorithme suivant :

Le code de l'expression $exp1 \ + \ exp2$ (ou de $exp1 \ * \ exp2$) est obtenu en concaténant

- le code de $expi$ qui est l'expression qui demande le plus de registres
- le translaté du code de $exp(3 - i)$
- l'instruction `Add(R1,R2)` (ou `Mult(R1,R2)`

Question 1. Donner un programme Yacc qui calcule le nombre de registres utilisé par ce deuxième alogorithme.

Question 2. Donner un programme Yacc qui construit le code en utilisant ce nouvel algorithme. On utilisera pour les variables sémantiques un type **struct** contenant un champ entier et un champ chaîne de caractères.

```
%{ #define YYSTYPE char*
#include <math.h>
#define LMAX 1000
trans (char *u){
      int i;
      for (i = 0; u[i] != '\0'; i++)
         if(isdigit(u[i])) u[i]++;}
  %}
%token NUMBER PLUS POINTV  MULT PARO PARF ID EGAL
%%
statement : expression POINTV
          {printf("\t %s \n", $1);}
          ;
expression : expression PLUS terme { trans($3); strcat(strcat ($1,$3), "add R1 R2 \n") ;
                     $$ = $1;}
            | terme {$$ = $1;}
            ;
terme:      terme MULT  facteur {trans($3); strcat(strcat($1,$3), "mult R1 R2 \n");
                       $$ = $1;}
            | facteur {$$ = $1;}
            ;
facteur : ID  {$$ = (char *) malloc(LMAX); strcpy($$,"load " );
               strcat($$, $1); strcat($$ ," R1 \n");}
         | PARO  expression PARF  {$$ = $2;}
          ;
```

Exercice 2

On utilise ici la grammaire suivante qui engendre des systèmes de parenthèses.

$$S_o \Rightarrow S; \quad S \ \rightarrow \ aSbS \qquad S \ \rightarrow \ aSb \quad S \ \rightarrow \ abS \qquad S \ \rightarrow \ ab$$

On souhaite effectuer la transformation suivante sur les systèmes de paranthèses : Les couples a, b qui ne sont pas inclus dans un autre sont à tranbsformer en c, d. Ainsi, $a\ b\ a\ a\ b\ a\ b\ b\ a\ a\ b\ b$ doit etre transformé en $c\ d\ c\ a\ b\ a\ b\ d\ c\ a\ b\ d$

Ecrire un programe Yacc qui effectue cette transformation.