

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Electronique et d'Informatique
Département Informatique



TP Compilation

Les quadruplets

Les quadruplets

L'instruction :

if (v) A=5; else B=2;

Les quadruplets que nous devons obtenir :

0 - (BZ , 3 , temp_cond , vide)

1 - (= , 5 , vide , A)

2 - (BR , 4 , vide , vide)

3 - (= , 2 , vide , B)

Les quadruplets

Partie Flex:

```
% {  
#include "pgm.tab.h"  
extern YYSTYPE yylval;  
#include "pgm.h"  
% }
```

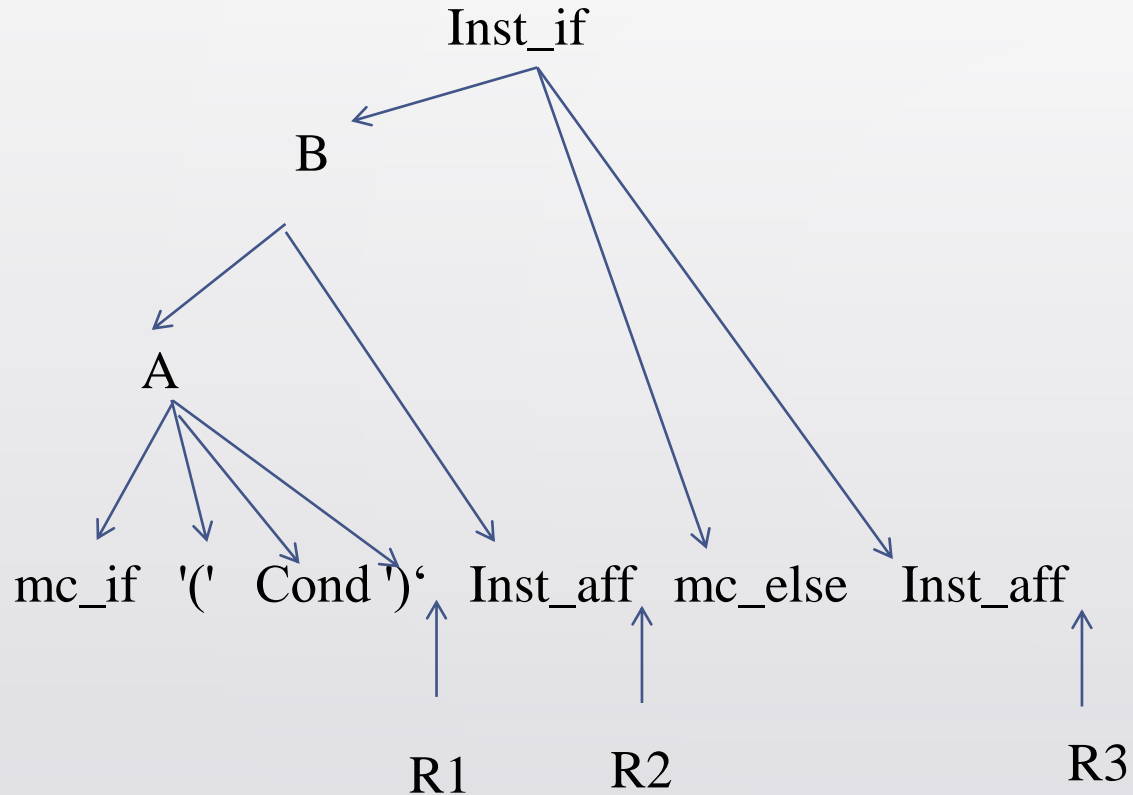
```
idf [A-Z][a-zA-Z0-9]*  
cst [0-9] +
```

```
%%  
if { return mc_if;}  
else { return mc_else;}  
{idf} { yylval.str=strdup(yytext); return idf;}  
{cst} { yylval.entier=atoi(yytext); return cst;}  
[=;()v] { return yytext[0];}  
[ \t\n]
```

- printf ("erreur lexicale");

Les quadruplets

La grammaire dans le cas de schémas de Traduction ascendant :



```
inst_if : B mc_else Inst_aff { //R3
                                }
```

```
;
B: A inst_aff { //R2
                }
```

```
;
A:mc_if '(' cond ')' {//R1
                      }
```

```
;
inst_aff: idf '=' cst ';'
```

```
;
cond: 'v'
```

```
;
```

Les quadruplets

Partie Bison:

```
%union {  
    int enter;  
    char* str;  
}  
%token <str>idf <entier>cst  
mc_if mc_else ( ) = ; v  
  
%%  
inst_if : B mc_else inst_aff { //R3  
    }  
;  
  
B : A inst_aff { //R2  
    }  
;  

```

```
A:mc_if '(' cond ')' { //R1  
    }  
;  
  
inst_aff : idf '=' cst ';' ;  
cond: 'v'  
;  
%%  
main()  
{  
    yyparse();  
}
```

Les quadruplets

Partie Bison:

```
% {
int deb_else=0;
int qc=0;
int Fin_if=0;
char tmp [20];
% }
%union {
int entier;
char* str;
}
%token <str>idf <entier>cst mc_if mc_else
%%
inst_if : B mc_else Inst_aff { printf(tmp,"%d",qc);
                               ajour_quad(Fin_if,1,tmp);
                               printf("pgm juste");
};
B: A inst_aff {
    Fin_if=qc;
    quadr("BR", "", "vide", "vide");
    sprintf(tmp,"%d",qc); // transformer entier vers
string
    ajour_quad(deb_else,1,tmp);
};
```

```
A:mc_if '(' cond ')'
{
    deb_else=qc; // J'ai laisser le champs 2 vide. Je dois le remplir
    apres
    quadr("BZ", "", "temp_cond", "vide");
}
;
```

```
inst_aff : idf '=' cst ';'
;
cond: 'v'
;
%%
main()
{
yyparse();
afficher_qdr();
}
```

```
0 - ( BZ , 3 , temp_cond , vide )
-----
1 - ( = , 5 , vide , A )
-----
2 - ( BR , 4 , vide , vide )
-----
3 - ( = , 2 , vide , B )
-----
```


Les quadruplets

Le fichier « pgm.h »:

```
typedef struct
{
    char oper[100];
    char op1[100];
    char op2[100];
    char res[100];

}qdr;

qdr quad[1000];

extern int qc;
```


Les quadruplets

```
void quadr(char opr[],char op1[],char op2[],char res[])
{

    strcpy(quad[qc].oper , opr);
    strcpy(quad[qc].op1  , op1);
    strcpy(quad[qc].op2  , op2);
    strcpy(quad[qc].res   , res);

    qc++;

}
```

Les quadruplets

```
void ajour_quad(int num_quad, int colon_quad, char val [])  
{  
    if (colon_quad==0)    strcpy(quad[num_quad].oper , val);  
    else if (colon_quad==1)    strcpy(quad[num_quad].op1 , val);  
        else if (colon_quad==2)    strcpy(quad[num_quad].op2 , val);  
            else if (colon_quad==3)    strcpy(quad[num_quad].res , val);  
}
```

Les quadruplets

```
void afficher_qdr()
{
printf("*****LesQuadruplets*****\n");

int i;

for(i=0;i<qc;i++)
{
printf("\n %d - ( %s , %s , %s , %s )",i,quad[i].oper,quad[i].op1,quad[i].op2,quad[i].res);
printf("\n-----\n");
}
}
```