

Université des Sciences et de la Technologie Houari Boumediene
Faculté d'Electronique et d'Informatique
Département Informatique



TP Compilation

20/03/2022

Les routines sémantique

20/03/2022

Variables et fonctions prédéfinies de FLEX

- **char yytext[]**: tableau de caractères qui contient la chaîne d'entrée en cours d'analyse.

```
IDF [a-zA-Z]([a-zA-Z][0-9])*  
%%  
{IDF} {printf("idf: "); printf(" %s ",yytext);}
```

```
var  
x
```

PgmSource

```
Idf: var  
Idf: x
```

PgmCompilé

- **int yyleng**: retourne la longueur de la chaîne d'entrée en cours d'analyse.

```
IDF [a-zA-Z]([a-zA-Z][0-9])*  
%%  
{IDF} {if (yyleng <=7) printf(" idf ");  
      else printf("erreur lexicale : idf trop long");  
      }
```

```
var  
vartoplogue
```

PgmSource

```
Idf  
Erreur lexicale :idf trop long
```

PgmCompilé

Variables et fonctions prédéfinies de FLEX

- **yylex()**: c'est la fonction qui lance l'analyseur lexical. Si on change le main on doit pas oublier de l'ajouter dans le main

```
%{  
int nb_ligne=0;  
%}  
%%  
\n nb_ligne++;  
%%  
int main()  
{  
  yylex();  
  printf("nombre de ligne %d",nb_ligne);  
}
```

Var

X

Y

Z

PgmSource

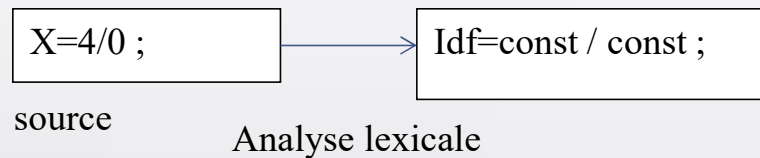
Nombre de ligne

4

PgmCompilé

Partage de variable entre FLEX et BISON

Comment Bison peut traiter les valeurs des entités lexicales, sachant que l'analyseur lexical à coder tout les entités de programme source?



Solution

Flex peut envoyer la valeur d'une entité avant de la coder en utilisant la variable prédéfinie **yyval**

Utilisation de yylval

1) Déclaration

Déclaration de **yylval** dans **FLEX**

```
%{  
extern YYSTYPE yylval;  
%}
```

Déclaration de type **YYSTYPE** dans **BISON** Comme type composé

```
%union {  
int    entier;  
char*  str;  
}
```

2) Utilisation

Envoie de valeur par **FLEX**

```
{cst} {  
    yylval.entier=atoi(yytext);  
    return cst;  
}  
  
{idf} {yylval.str=strdup(yytext);  
    return idf;  
}
```

Traitement des valeurs par **BISON**

```
%token <str>idf  <entier>cst  
  
%%;  
s : idf= idf '/' cst if ($5==0) printf(" erreur : division par 0")  
ELSE printf("la Division est%s= %s/%d", $1,$3,$5);  
    YYACCEPT;}
```

-Le symbole \$\$ référence la valeur associée au non terminal de la partie gauche d'une règle de grammaire.

-Le symbole \$i référence la valeur associée au i^{ème} symbole terminal **ou** non terminal de la partie droite d'une règle de grammaire.

Routine sémantique de double déclaration d'une variable

Comment savoir si une variable est double déclarée ou non ?

Vérifier le champs Type de cette variable dans la table de symbole:

- Si il est vide alors la variable n'est pas encore déclarée : Insérer le type.
- Sinon variable est déclarée déjà, signaler double déclaration.

Ex.y

```
declaration: type listeparam ';' declaration | type listeparam ';'
;
```

```
listeparam: listeparam ',' idf      { /*vérification de double déclaration*/ }
          | idf;                    { /*vérification de double déclaration*/ }
```

```
type: mc_integer      {strcpy(suavType,$1); }
    | mc_real          {strcpy(suavType,$1);}
;
```

Routine sémantique de double déclaration d'une variable

```
declaration: type listeparam ';' declaration | type listeparam ';'
;

listeparam: listeparam ',' idf { if (doubleDeclaration($3)==0) insererType($3,suavType);
                                else printf("erreur Sémantique: double déclation de %s, à la ligne %d\n", $3, nb_ligne); }

    | idf { if ( doubleDeclaration($1)==0)insererType($1,suavType);
            else printf("erreur Sémantique: double déclation de %s, à la ligne %d\n", $1, nb_ligne);}
;

type: mc_integer {strcpy(suavType,$1); }
    | mc_real {strcpy(suavType,$1);}
;
```


Exemple d'une erreur sémantique de double déclaration

```
exp.txt  synt.y  lex.l  ex.l  ex.y  ex.y
```

```
1 program L3
2 integer i,j,k;
3 real k;
4 begin
5 i=2;
6 j=i/0;
7 end
8
```

```
D:\enseignement (10.10.10.3)\compil_Tos\2016_2017\exp>comp<exp.txt
erreur Sémantique: double déclaration de k, à la ligne 3
Erreur sémantique: Division par zéro à la ligne 6
programme syntaxiquement juste

/*****Table des symboles *****/

```

	NomEntite	CodeEntite	TypeEntite
	L3	idf	
	i	idf	integer
	j	idf	integer
	k	idf	integer

Routine sémantique de variable non déclarée

Comment savoir si une variable est déclarée ou non ?

Vérifier le champs Type de cette variable

- Si il est vide alors la variable n'est pas encore déclarée :
Signaler erreur de non déclaration.
- Sinon variable est déclarée : utiliser la variable selon sa position

Ex.y

```
instaff: idf '=' idf'; { /* vérification de la déclaration des variables */  
}
```