Université de Versailles Saint-Quentin-en-Yvelines
UFR des Sciences
45 Avenue des Etats-Unis,
78000 Versailles

Allan Mouhani
Abdoulaye balde
Date: 27/05/2018

Projet
de
Théorie des Langages

Rapport

1- Formalisation du problème :

Question 1 :

Soit G la grammaire définit par G = $\{\Sigma, N, S, R\}$, avec :

$\Sigma = \{(,), +, -, *, /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

N = $\{Expr, Int, Op\}$
S = $\{Expr\}$
R = Expr → Int | (Expr Op Expr) ; Op → + | -| * | /; Int → $(0|1|2|3|4|5|6|7|8|9)^+$

Question 2 :

Structure de données Token :

```
typedef struct token{
    int type;
    int data;
}token;;
```

Le champ "type" représente le type de token correspondant : 0 pour les opérateurs, 1 pour les entiers , 2 pour les parenthèses ouvrantes et 3 pour les parenthèses fermantes .
le champ "data" représente la valeur selon le type de token.

```
typedef struct List_token{
    token tok;
    struct List_token *suiv;
}*list_token;
```

Le champ "tok" représente un token.
le champ "suiv" représente un pointeur vers le token suivant .

Question 3 :

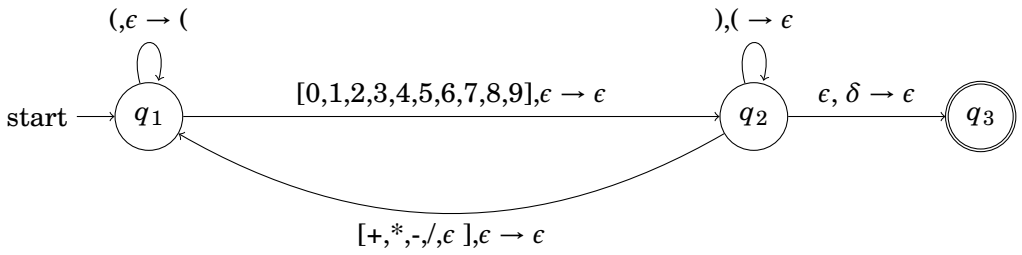
Soit L(G) le langage généré par la grammaire G , définit par :

L(G) = (w $\in \Sigma^*$, tel que w est une expression arithmétique correcte).

- Automate à Pile correspondant :

$A = \{\Sigma, Q, \Gamma, q_1, F, T, \delta\}$ avec :

$\Sigma = \{(,), +, -, *, /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
$Q = \{q_1, q_2, q_3\}$
Γ



Remarque:

La transition [0,1,2,3,4,5,6,7,8,9],$\epsilon \rightarrow \epsilon$ est à interpéter comme étant :
0,$\epsilon \rightarrow \epsilon$
1,$\epsilon \rightarrow \epsilon$
2,$\epsilon \rightarrow \epsilon$
3,$\epsilon \rightarrow \epsilon$
4,$\epsilon \rightarrow \epsilon$
5,$\epsilon \rightarrow \epsilon$
6,$\epsilon \rightarrow \epsilon$
7,$\epsilon \rightarrow \epsilon$
8,$\epsilon \rightarrow \epsilon$
9,$\epsilon \rightarrow \epsilon$

De même la transition [+,*,-,/,ϵ],$\epsilon \rightarrow \epsilon$ est à interpéter comme étant :
+,$\epsilon \rightarrow \epsilon$
-,$\epsilon \rightarrow \epsilon$
*,$\epsilon \rightarrow \epsilon$
/,$\epsilon \rightarrow \epsilon$
ϵ,$\epsilon \rightarrow \epsilon$, Cependant cette transition permet de retourner à l'état q_1 notamment lorsqu'on a en entrée un nombre à plusieurs chiffres.

Exemple : 12345

Question 4 :

Structure de données :

```
typedef struct noeud{
        token t;
        struct noeud* filsG;
        struct noeud* filsD;
}NOEUD;
```

Cette Structure représente comme son nom l'indique un noeud de l'arbre.
Le champ "token" représente un token.
Les champs "filsG,filsD" représentent les descendances dans l'arbre du token.

```
typedef struct arbre{
        NOEUD *n;
        struct arbre *suiv;
}*arbre_token,Arbre;
```

Cette Structure représente un arbre et une pile.
Le champ "n" représente un noeud de l'arbre.
Le champ "suiv" représente un pointeur vers le noeud suivant .

2- Modalités pratiques :

Compilation et Exécution du programme :

Le programme peut être compilé en utilisant la commande :
$ make test .

Concernant l'exécution du programme, l'utilisateur doit borner l'expression arithmétique avec des apostrophes ".

On retrouvera dans le Makefile 3 situations de test différents :
- Le premier cas de figure correspond au test d'une expression arithmétique lexicalement incorrecte.
- Le deuxième cas de figure correspond au test d'une expression arithmétique syntaxiquement incorrecte.
- Et enfin le dernier correspond au test d'une expression arithmétique lexicalement et syntaxiquement correcte.

On trouvera également dans le Makefile , d'autres cibles comme "eval" pour l'édition de lien, "eval.o" pour la compilation seule et "clean" pour la suppression de fichiers générés par le mécanisme de compilation.