

TP2: Réalisation d'un Analyseur Lexical

Nombre de Séances : 2

La réalisation d'un analyseur lexical consiste à :

1. Écrire une expression régulière pour chaque unité lexicale. Soit donc r_1, r_2, \dots, r_n ;
2. Construire un AEF qui reconnaît $r = (r_1 \mid r_2 \mid \dots \mid r_n)$;
3. Transformer l'AEF obtenu AEF déterministe ;
4. Réaliser le programme qui code l'AEF obtenu.

L'objectif de ce TP est d'accomplir l'étape n°4 en C du langage décrit par la grammaire suivante :

```
Program -> program ident vardecllist fundeclist instr  
vardecllist -> ε | varsdecl | varsdecl ; vardecllist  
varsdecl -> var identlist : typename  
identlist -> ident | ident , identlist  
typename -> atomictype | arraytype  
atomictype -> unit | bool | int  
arraytype -> array [ rangelist ] of atomictype  
rangelist -> int .. int | int .. int , rangelist  
fundeclist -> ε | fundecl ; fundeclist  
fundecl -> function ident ( arglist ) : atomictype vardecllist instr  
arglist -> ε | arg | arg , arglist  
arg -> ident : typename | ref ident : typename  
instr -> if expr then instr | if expr then instr else instr  
| while expr do instr | lvalue := expr | return expr | return  
| ident ( exprlist ) | begin sequence end | begin end  
sequence -> instr ; sequence | instr ; | instr  
lvalue -> ident | ident [ exprlist ]  
exprlist -> expr | expr , exprlist  
expr -> cte | ( expr ) | expr opb expr | opu expr  
| ident ( exprlist ) | ident [ exprlist ] | ident
```

opb -> +|-|*|/|=|>|<| and|or
opu -> +|-|not

Les terminaux apparaissent en gras. Le terminal **cte** représente les constantes entières et booléennes (true et false); le terminal **ident** représente les identificateurs qui sont formés d'un caractère alphabétique suivi d'une suite de caractères alphanumériques ou d'une apostrophe ou du caractère de soulignement. Les commentaires débutent par les deux caractères (* et se terminent par les deux caractères *), ils peuvent être imbriqués.

Remarque : Vous ferez en sorte que le texte à analyser soit lu dans un fichier.

Bon courage