## UE MIC0602T – Théorie des Langages Travaux Dirigés n° 3

Mise en œuvre de l'Analyse descendante

Traiter les questions suivantes.

1 - Soit la grammaire des expressions booléennes G7 (V $_{\text{N}},\,\text{V}_{\text{T,,}}\,\,\text{S},\,\text{R})$  :

```
\begin{array}{lll} V_N = & \{A,\,B,\,C\} \\ V_T = & \{ou,\,et,\,non,\,(,\,),\,vrai,\,faux\,\} \\ S = & A \\ R = & \{ & & \\ & 1\,2 & A \rightarrow A \ ou \ B \ | \ B \\ & 3\,4 & B \rightarrow B \ et \ C \ | \ C \\ & 5\,6\,7\,8 & C \rightarrow non \ C \ | \ (\,A\,) \ | \ vrai \ | \ faux \\ & \} \end{array}
```

- 1.1 Construire la grammaire G71 équivalente non récursive à gauche et factorisée si nécessaire traité au TD 1.
- 1.2 Calculer les ensembles PREMIER de chaque non terminal.
- 1.3 Calculer les ensembles SUIVANT de chaque non terminal.
- 1.4 A partir de la version LL (1) de cette grammaire (cf corrigé n° 1), donner l'arbre de dérivation du mot :

```
non ( vrai ou faux ) et vrai
```

- 1.5 Dresser le tableau (Pile, Entrée, Sortie) sur le mot du 1.4
- 1.7 Construire la table d'analyse descendante de la grammaire initiale. Que constatez-vous ?
- 2 Les règles suivantes sont extraites de la grammaire qui reconnaît une partie des instructions d'un langage de programmation.

```
INSTRUCTION
                  \rightarrow
                        if EXPRESSION then BLOC else BLOC endif
                        | VARIABLE := EXPRESSION
BLOC
                        INSTRUCTION SOUS_BLOC
SOUS_BLOC
                        ; BLOC | ε
EXPRESSION
                  \rightarrow
                         VARIABLE SOUS EXPRESSION | NOMBRE
SOUS_EXPRESSION→
                        + EXPRESSION | = EXPRESSION | \varepsilon
                        LETTRE SOUS_VARIABLE
VARIABLE
                  \rightarrow
SOUS_VARIABLE →
                        VARIABLE | ε
                  \rightarrow
LETTRE
                         "a" | "b" |... | "z"
NOMBRE
                         CHIFFRE NOMBRE | CHIFFRE
                         "0" | "1" |... | "9"
CHIFFRE
```

- 2.1 Donner les éléments VN, VT et la racine de cette grammaire
- 2.2 Eliminer la récursivité et factoriser si nécessaire
- 2.3 Calculer les ensembles PREMIER de chaque non terminal.
- 2.4 Calculer les ensembles SUIVANT de chaque non terminal.
- 2.5 Construire la table d'analyse LL (1)
- 2.6 Dresser le tableau (Pile, Entrée, Sortie) sur le programme (mot) suivant :

```
if x = 2
then
y := 3
else
y := zzz + 3
endif
```