

SERIE DE TD N°:2 COMPILATION

ANALYSE SYNTAXIQUE : INTRODUCTION, RAPPELS ET COMPLEMENTS

Exercice 1

1) Soit la grammaire $G = (\{a, b\}, \{S, A, B, C\}, S, P)$ où P est défini par :

$S \rightarrow AB \mid BC$

$A \rightarrow BA \mid a$

$B \rightarrow CC \mid b$

$C \rightarrow AB \mid a$

Analyser la chaîne **babaab** de manière descendante puis ascendante, en construisant, à chaque fois, son arbre de dérivation.

Exercice 2

1) Soit la grammaire des expressions arithmétiques $G = (\{+, -, *, /, a, b, c, (,)\}, \{E\}, E, P)$ où P est défini par :

$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid (E) \mid -E \mid a \mid b \mid c$

Donner l'arbre de dérivation de la chaîne **b + a * b - c** ? Que peut-on en déduire ?

2) Soit la grammaire $G' = (\{+, -, *, /, a, b, (,)\}, \{E, T, F, L\}, E, P)$ où P est défini par :

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow -F \mid L$

$L \rightarrow (E) \mid a \mid b \mid c$

a) Donner l'arbre de dérivation de la chaîne **b + a * b - c**. Que peut-on en déduire ?

b) Donner l'arbre abstrait correspondant à la chaîne **b + a * b - c**.

c) Donner les arbres syntaxiques concret et abstrait de la chaîne **b - a + c**.

Exercice 3

1) Soit la grammaire des expressions booléennes $G = (\{\text{ou}, \text{et}, \text{non}, \text{vrai}, \text{faux}, (,)\}, \{A\}, A, P)$ où P est défini par :

$A \rightarrow A \text{ ou } A \mid A \text{ et } A \mid \text{non } A \mid (A) \mid \text{vrai} \mid \text{faux}$

Donner l'arbre de dérivation de la chaîne **non faux ou vrai et vrai** ? Que peut-on en déduire ?

2) Proposer une grammaire G' pour éliminer le problème posé par la grammaire G .

3) Donner l'arbre de dérivation de la chaîne **non faux ou vrai et vrai** en utilisant la grammaire G' .

Exercice 4

Soit la grammaire G d'un langage proche de Pascal, exprimée sous forme EBNF de la manière suivante, exprimer la grammaire G sous forme de diagrammes syntaxiques.

$\langle \text{Programme} \rangle ::= \underline{\text{Program}} \text{ ident} ; \langle \text{Bloc} \rangle .$

$\langle \text{Bloc} \rangle ::= [\underline{\text{Const}} \langle \text{SuitConst} \rangle ;] [\underline{\text{Var}} \langle \text{SuitVar} \rangle ;] \{ \underline{\text{Procédure}} \text{ ident} ; \langle \text{Bloc} \rangle ; \}$

$\underline{\text{Begin}} \langle \text{Inst} \rangle \{ ; \langle \text{Inst} \rangle \} \underline{\text{End}}$

$\langle \text{SuitConst} \rangle ::= \langle \text{DecConst} \rangle \{ , \langle \text{DecConst} \rangle \}$

$\langle \text{DecConst} \rangle ::= \text{ident} = \text{nbEnt}$

$\langle \text{SuitVar} \rangle ::= \text{ident} \{ , \text{ident} \}$

$\langle \text{Inst} \rangle ::= \text{ident} := \langle \text{Exp} \rangle \mid \text{If } \langle \text{Cond} \rangle \text{ Then } \langle \text{Inst} \rangle [\text{Else } \langle \text{Inst} \rangle]$

$\mid \text{Repeat } \langle \text{Inst} \rangle \text{ Until } \langle \text{Cond} \rangle \mid \text{While } \langle \text{Cond} \rangle \text{ Do } \langle \text{Inst} \rangle \mid \underline{\text{Begin}} \langle \text{Inst} \rangle \{ ; \langle \text{Inst} \rangle \} \underline{\text{End}}$

$\langle \text{Cond} \rangle ::= \langle \text{Exp} \rangle (= \mid < \mid > \mid < = \mid > =) \langle \text{Exp} \rangle$

$\langle \text{Exp} \rangle ::= \langle \text{Terme} \rangle \{ (+ \mid -) \langle \text{Terme} \rangle \}$

$\langle \text{Terme} \rangle ::= \langle \text{Facteur} \rangle \{ (* \mid /) \langle \text{Facteur} \rangle \}$

$\langle \text{Facteur} \rangle ::= \text{ident} \mid \text{nbEnt} \mid (\langle \text{Exp} \rangle)$