**Université de M'hamedBouguerraBoumerdes**

**Faculté des sciences**

**Département d'Informatique**

**Module : Théorie des Langages.** **Année : 2019-2020**

**Filière : LI- S4** **Document : Série 2 (Corrigé)**

**Chapitre 2 : grammaires**

**Objectif :** Comprendre :grammaire, type de langage/grammaire et la relation entre grammaire/langage.

**Exercice 01**

Pour trouver le langage généré par une grammaire, il faut trouver tous les mots w qui sont générés par celle-ci.

Pour le type de la grammaire, il faut voir de quelle forme sont ses règles de production. Par contre le type du langage est le type le plus élevé des grammaires qui l'engendre.

*Remarque :* Le type le plus élevé est le 3 puis le 2 puis le 1 puis le 0.

**1)** G1 = ({a, b}, {S}, S, {S → abS | b})

**Le langage :**

On a 2 productions: S → abS et S →b

On commence les dérivations:

S⇒**b** donc w= b c'est le premier et le plus petit mot qu'on peut obtenir à partir de l'axiome S en utilisant la deuxième règle.

Le prochain mot, on l'obtient en utilisant la première règle une fois puis la deuxième :

S ⇒abS⇒**abb** donc w=abb

Puis on va utiliser la première règle deux fois ensuite la première:

S ⇒abS⇒ ab abS⇒ababb = $\mathbf{(ab)^2b}$

En utilisant la première règle plusieurs fois, on obtient :

S ⇒abS⇒ ab abS⇒abababS⇒ $(ab)^4S \overset{*}{\Rightarrow} (ab)^n S \Rightarrow \mathbf{(ab)^n b}$

Donc, on peut déduire que le langage généré par cette grammaire est:

**L(G1)= { $(ab)^n$ b, n>=0 }**

**Remarque**:on voit bien qu'en remplaçant le n par « 0 », on obtient le premier mot « b » puis par « 1 » le mot « abb », par « 2 » le mot « $(ab)^2$b » et ainsi de suite.

**Le type :**

Les 2 règles de production sont de la forme A → αB ou A → α avec A, B ∈ N et α ∈T*

G1 est de type 3 ⇒**L(G3) est de type 3**

**2)** G2 = ({a}, {S}, S, {S →aSa|ε})

**Le langage :**

On a 2 productions: $S \rightarrow aSa$ et $S \rightarrow \varepsilon$

On commence les dérivations:

$S \Rightarrow \boldsymbol{\varepsilon}$ donc $w= \varepsilon$ c'est le premier et le plus petit mot qu'on peut obtenir à partir de l'axiome S en utilisant la deuxième règle.

Le prochain mot, on l'obtient en utilisant la première règle une fois puis la deuxième :

$S \Rightarrow aSa \Rightarrow a\varepsilon a = \mathbf{aa}$ donc $w=aa$

Puis on va utiliser la première règle deux fois ensuite la première:

$S \Rightarrow aSa \Rightarrow aaSaa \Rightarrow aa\varepsilon aa = \mathbf{a^4}$

En utilisant la première règle plusieurs fois, on obtient :

$S \Rightarrow aSa \Rightarrow aaSaa \Rightarrow a^3Sa^3 \Rightarrow a^4Sa^4 \overset{*}{\Rightarrow} a^n S a^n \Rightarrow a^n\varepsilon a^n \mathbf{= a^{2n}}$

Donc, on peut déduire que le langage généré par cette grammaire est:

**$L(G2)= \{ a^{2n}, n>=0 \}$ ou $L(G2)= \{ a^n$, n est pair }**

**Le type :**

G2 est de type 2 car ses 2 règles sont de la forme $A \rightarrow \alpha$ avec $A \in N$ et $\alpha \in (T \cup N)^*$

Mais **L(G2) est de type 3** car on peut trouver une grammaire G2' de type 3 tel que L(G2)=L(G2')

$G2'= (\{a\}, \{S\}, S, \{S \rightarrow aaS|\ \varepsilon\})$

3) $G3= (\{a,b,c\}, \{S,X, Y\}, S,\{S \rightarrow XY, X \rightarrow aXb/ \varepsilon, Y \rightarrow bYc/\varepsilon\} )$

**Le langage :**

On procède de la même façon que pour les deux premiers exemples :

$S \Rightarrow XY \Rightarrow \varepsilon Y \Rightarrow \varepsilon\varepsilon = \boldsymbol{\varepsilon}$

$S \Rightarrow XY \Rightarrow X\varepsilon \Rightarrow aXb \Rightarrow aaXbb \Rightarrow a^3Xb^3 \overset{*}{\Rightarrow} a^nXb^n \Rightarrow \mathbf{a^nb^n}$

$S \Rightarrow XY \Rightarrow \varepsilon Y \Rightarrow bYc \Rightarrow bbYcc \Rightarrow b^3Yc^3 \overset{*}{\Rightarrow} b^mYc^m \Rightarrow \mathbf{b^mc^m}$

On remarque que le non terminal « X » permet de dériver le langage $\{a^nb^n , n >=0\}$ et le non terminal « Y » le langage $\{ b^mc^m , m >=0 \}$ et puisque à partir de l'axiome « S », on a une seule dérivation qui est $S \rightarrow XY$, on peut déduire que le langage généré par cette grammaire est la concaténation des deux langages générés par X et Y.

Autrement dit :

$X \overset{*}{\Rightarrow} a^nb^n$ et $Y \overset{*}{\Rightarrow} b^mc^m$ donc $S \Rightarrow XY \Rightarrow \mathbf{a^nb^nb^mc^m = a^n\, b^{n+m}\, c^m}$

D'où **$L(G3)= \{ a^n\, b^{n+m}\, c^m , n,m>=0\}$**

**Le type :**

G3 est de type 2 selon ses règles de dérivation et **L(G3) est de type 2** car on ne peut pas trouver une grammaire de type 3 qui l'engendre. ( Dans les grammaires de type 3, on ne peut pas contrôler le faite que le nombre de b est la somme du nombre de a et c, reste à démontrer (voir chapitre4).

**4)** G4 = ({a,b}, {S, R}, S, {S→ aS | bR | b , R→aR | bS)

**Le langage :**

On procède de la même façon :

- Le plus petit mot

$S \Rightarrow \mathbf{b}$

- Si on commence par S→ aS , on aura :

$S \Rightarrow aS \Rightarrow \mathbf{ab}$

$S \Rightarrow aS \Rightarrow a^2S \overset{*}{\Rightarrow} a^nS \Rightarrow \mathbf{a^nb}$

- Sinon avec S→ bR, on aura :

$S \Rightarrow bR \Rightarrow b^2S \Rightarrow b^3R \Rightarrow b^4S \Rightarrow \ldots \Rightarrow b^{2n}S \Rightarrow \mathbf{b^{2n}b = b^{2n+1}}$

$S \Rightarrow bR \Rightarrow b\ aR \overset{*}{\Rightarrow} b\ a^mR \Rightarrow ba^mbS \Rightarrow \mathbf{ba^mb\ b}$

- On remarque qu'on peut avoir un nombre quelconque de a mais le nombre de b est toujours contrôlé par les deux symboles S et R. les mots toujours se terminent par un b

**Ou L(G4)={ w=w'b avec w'∈ {a,b}* et $|w'|_b$= 2n (pair), n>=0}**

**Le type**

G4 est de type 3 donc **L(G4) est de type 3**.

**Exercice 2**

1. Donner une grammaire pour les langages suivants.
2. Démontrer que $L_1$=L(G).

$L_1$= { w∈{a,b}* \ w =$a^nb^n$, n>=0}

$L_2$= {l'ensembles des palindromes}={ w∈{a,b}* \ w =$w^r$}

$L_3$= { w∈{a,b}* \ w =$a^nb^m$, n>m}

$L_4$={ w∈{a,b,c}* \ w=$a^{2n+2}(ab)^p c^2(bc)^{m+1}$ , n,m,p ≥ 0 }

**1) Grammaires**

Soit G1 <Vn,Vt,S,P> la grammaire qui génère le langage L1, avec

Vn={S}, Vt={a,b}, S axiome, P={S→aSb/ ε}

Soit G2 <Vn,Vt,S,P> la grammaire qui génère le langage L2, avec

Vn={S}, Vt={a,b}, S axiome, P={S→aSa/ bSb / ε}

Soit G3 <Vn,Vt,S,P> la grammaire qui génère le langage L3, avec

Vn={S}, Vt={a,b}, S axiome, P={S→aS/ aSb / a}

Soit G4 <Vn,Vt,A,P> la grammaire qui génère le langage L4, avec

Vn={A,B,D}, Vt={a,b,c}, A axiome, P={A→aaA/ aaB, B→abB/ccD, D→bcD/bc}

**2) Montrons que L1=L(G1)**

Pour monter que L1=L(G1) il faut monter les double inclusions $L1 \subseteq L(G1)$ et $L(G1) \subseteq L1$.
Nous allons le démontrer par récurrence sur la longueur de dérivation pour le cas et sur autre longueur d'un mot du langage dans l'autre cas.

**i) $L1 \subseteq L(G1)$**

Il faut démontrer que tout mot appartenant au langage est généré par la grammaire.
Autrement dit : $w \in L1 \Rightarrow w \in L(G1), \ \forall |w| \in N$

Cas 0

Soit $w \in L1$ et $|w|=0 \Rightarrow w=\varepsilon$ et w est bien généré par $S \rightarrow \varepsilon \Rightarrow w \in L(G1)$

Cas général

Nous supposons que $|w|=k$, $w \in L1 \Rightarrow w \in L(G1) \Rightarrow S \overset{*}{\Rightarrow} w$ et montrons quelle reste vraie pour k+1

Soit $|w|=k+1$, $w \in L1 \Rightarrow w = aw_1b$ et $|w|=k+1 \Rightarrow |w_1|=|w|-2=k+1-2=k-1 \Rightarrow |w_1|<=k$

$\Rightarrow w_1 \in L1$ et $w_1 \in L(G1) \Rightarrow S \overset{*}{\Rightarrow} w_1$ (selon hypothèse).

D'après les règles de la grammaire $S \Rightarrow aSb \Rightarrow aw_1b = w$, donc $w \in L(G1)$

$\Rightarrow$ **$L1 \subseteq L(G1)$**

**ii) $L(G1) \subseteq L1$**

Il faut démontrer que tout mot généré par la grammaire appartient au langage L1. Autrement dit, $w \in L(G1) \Rightarrow w \in L1, \forall |w| \in N$

Cas 0

$|w|= 0 \Rightarrow w=\varepsilon$ et $S \Rightarrow \varepsilon$, $w \in L(G1)$ et $\varepsilon = a^0b^0 \in L1$

Cas général

Supposons que $|w|=k$, $w \in L(G1)$ et $w \in L1$ soit vrai et montrons que pour $|w|=k+1$, si $w \in L(G1)$ alors $w \in L1$.

Soit $|w|=k+1$, et $w \in L(G1)$

Les dérivations donnent $S \Rightarrow aSb \overset{*}{\Rightarrow} aw_1b=w$

$|w|=|aw_1b|=k+1 \Rightarrow |w_1|=k+1-2=k-1 <=k$

$w_1 \in L1$ (selon hypothèse) $\Rightarrow w_1= a^nb^n \Rightarrow w=aa^nb^nb \Rightarrow w=a^{n+1}b^{n+1} \Rightarrow w \in L1$

$\Rightarrow L(G1) \subseteq L1$

**On peut conclure de i) et ii) que L(G1) = L1**

**Exercice 3**

$\mathbf{G_{exp}}$ = ($V_N$,$V_T$,Axiome,Règles)

$\mathbf{V_T}$ = { **0**, ..., **9**, **+**,**-**, **/**, *****, **)**, **(**}

$\mathbf{V_N}$ = { Expr, Nbr, Cte, Oper}

**Axiome** : Expr

**Règles :**

Expr→Nbr | (Expr) | Expr Oper Expr

Nbr→Cte | CteNbr

Cte→**0** | **1** |...| **9**

Oper→**+** | **-** | ***** | **/**

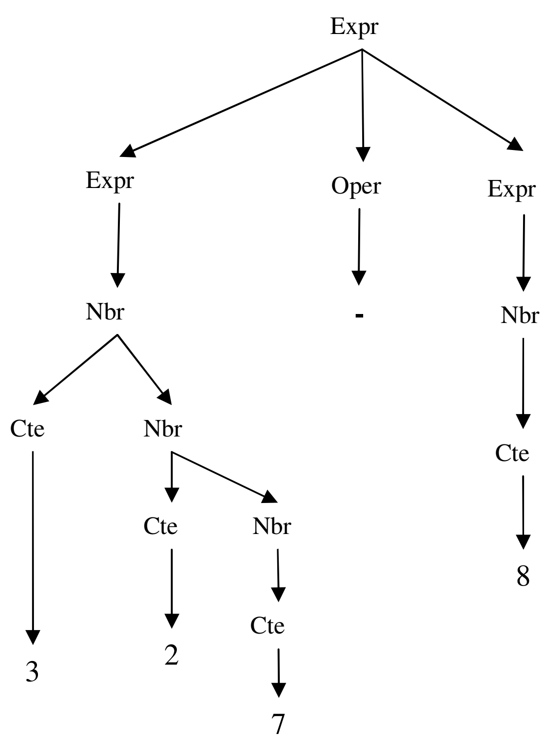
1. Montrer que le mot w = 327 - 8 ∈ L(G).

   Pour montrer que w ∈ L(G), il suffit de le dériver à partir de l'axiome Expr .

   Expr⇒Expr Oper Expr ⇒Expr – Expr⇒Expr - Nbr⇒Expr - cte⇒Expr - 8⇒Nbr – 8

   ⇒CteNbr- 8⇒CteCteNbr -8⇒CteCteCte -8⇒CteCte 7 -8⇒Cte 27 -8⇒**327-8**

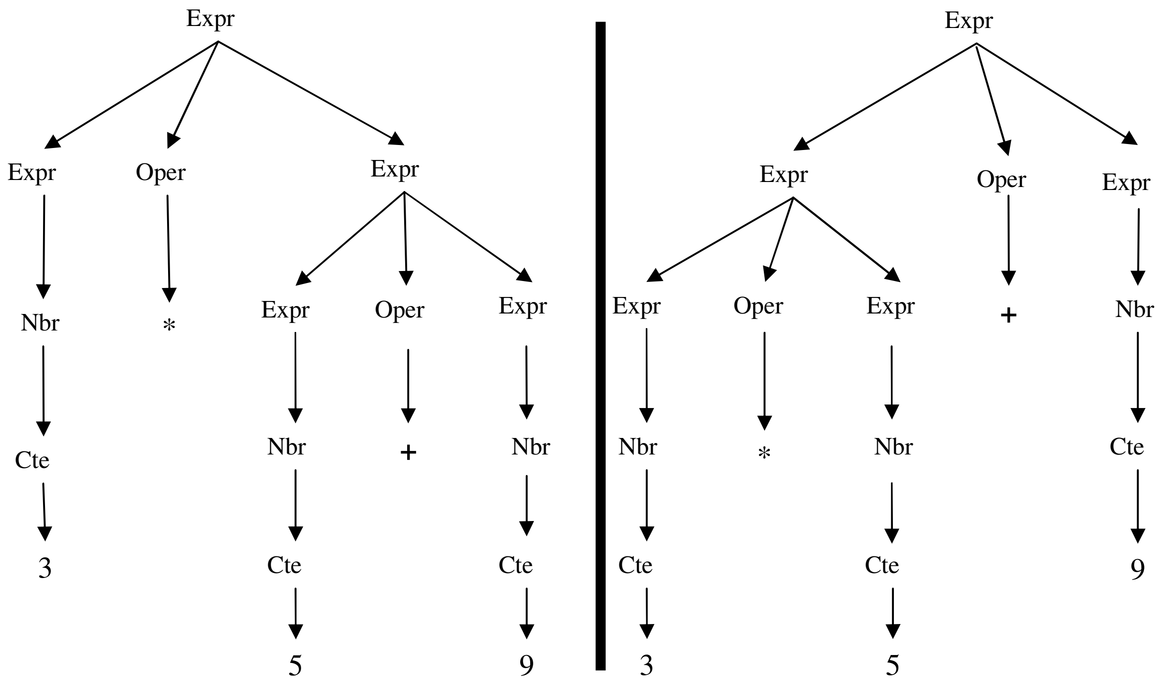
   On peut aussi utiliser l'arbre de dérivation




2. Montrer que la grammaire G est ambigüe.

Pour montrer que la grammaire est ambigüe, il suffit de trouver un mot w ∈ L(G) qui a deux (2) arbres de dérivation différents.

Exemple : **w= 3*5+9**




**Arbre 1** **Arbre 2**

**Remarque** : tous les mots contenant au moins 2 opérateurs (+,-,*,/) ont 2 arbres de dérivation différents.

**Exercice 4**

Soit la grammaire G dont les règles de production sont :

$S \rightarrow aAb \mid \varepsilon$

$A \rightarrow aSb$

$Ab \rightarrow \varepsilon$

1) Déterminer L(G).

On remarque qu'on a deux types de mots : Soit $a^n b^n$ si n est pair **ou** $a^n\ b^{n-1}$ si n impair

Donc **L(G)= { $a^n b^n$ , n est pair} ∪{ $a^n b^{n-1}$ , n impair }**

**Ou L(G)= { $a^{2n} b^{2n}$ , n >=0} ∪ { $a^{2n+1}\ b^{2n}$ , n >=0}**

2) Une grammaire de type 2 équivalente à G est

**G'=( {a,b} ,{S}, S, {S→aaSbb | a | ε } )**

**Exercice 5**

1. Montrer que la grammaire G ci-dessous est ambiguë.

G= ({condition, si, alors, instruction, sinon}, {S}, S, R ) où R est l'ensembles des règles suivantes :

S→ si condition alors S

S→ si condition alors S sinon S

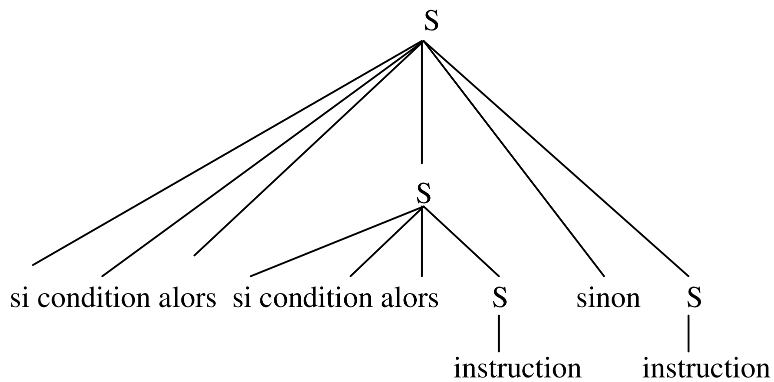
S→ instruction

2. Quel est le résultat du programme suivant, selon l'arbre de dérivation utilisé pour analyser l'instruction conditionnelle suivante :x :=1 ;  si x > 5 alors si x<10 alors x :=x+1 sinon x :=x-1

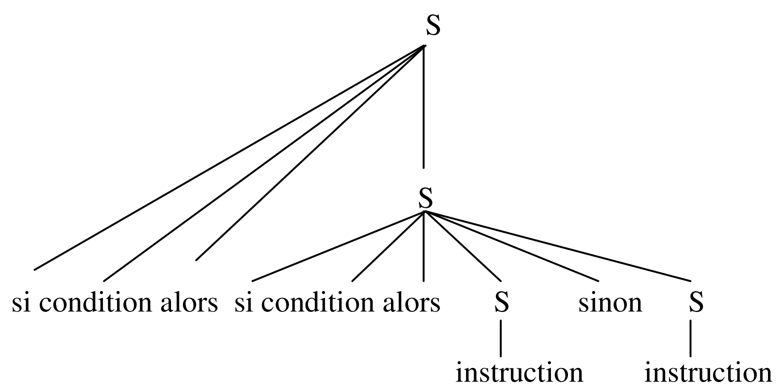
Solution

1. soit w  le mot  « si condition alors  si condition alors instruction sinon instruction »

Représentons deux arbres de dérivations a1 et a2




Arbre A1




Arbre A2

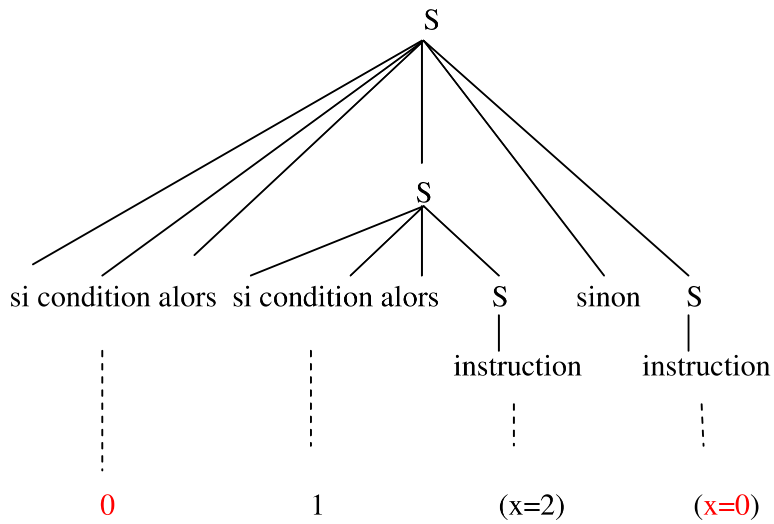
Deux arbres **différents** pour le **même** mot, la grammaire G est donc ambiguë.

2. **Résultats du programme selon les arbres A1 et A2**

Pour analyser l'instruction conditionnelle il faudrait compléter la grammaire. A ce stade du cours introductif nous allons simplifier l'exercice en gardant cette grammaire et se contentant de certains résultats intermédiaires.

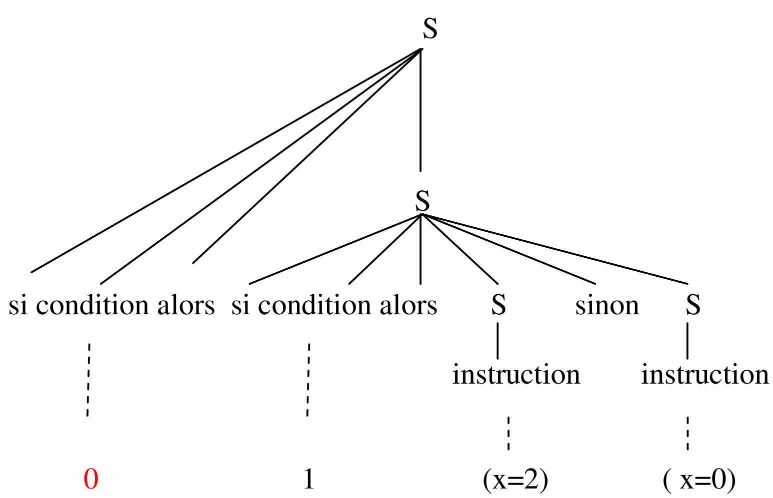
Résultats intermédiaires ;
x :=1 donc x < 5 d'où condition = 0
x<10 donc condition =1 alors x :=x+1 sinon x :=x-1

Arbre 1




La condition est fausse et il faut exécuter le sinon donc la valeur de x change et x=0

Arbre 2




La condition est fausse et il n’y a pas de sinon donc la valeur de x reste inchangée, x=1.