Proposition de correction des TD langages

à ne pas distribuer aux étudiants

Séance 1 :
Faire en séance : exercices 1 et 2 Optionnel (à faire à la maison) exercice 3

Séance 2 :
Faire en séance : exercices 4 et 5 Optionnel (à faire à la maison) exercices 6 et 7

Séance 3 :
Faire en séance : exercices 8, 9 et 10

1.

Rappel de la définition :
Soit (E, †, λ) un monoïde.
(S, †, λ) est un sous-monoïde de (E, †, λ) si :
- $S \subset E$
- † est une loi interne dans S.
- $\lambda \in S$

A inclus dans V*
quel que soit x appartenant à A $|x| = 2n$
quel que soit y appartenant à A $|y| = 2m$
$|x . y| = 2n + 2m = 2(n + m)$
$|x . y|$ est pair, donc x . y appartient à A
ε appartient à A
(A, ., ε) est donc un sous-monoïde de (V*, ., ε) .

B inclus dans V*
mais ε n'appartient pas à B car $|\varepsilon| = 0$
(B, .) n'est donc pas un sous-monoïde de V*.

C inclus dans V*
quel que soit x appartenant à C, $x = (a_1 a_2)^n$
quel que soit y appartenant à C, $y = (a_1 a_2)^m$
$x . y = (a_1 a_2)^n . (a_1 a_2)^m = (a_1 a_2)^{n+m}$ donc x . y appartient à C
$\varepsilon = (a_1 a_2)^0$ appartient à C
(C, ., ε) est donc un sous-monoïde de (V*, ., ε).

D inclus dans V*
quel que soit x appartenant à D, $x = a_1^n a_2^n$
quel que soit y appartenant à D, $y = a_1^m a_2^m$
$|x . y| = a_1^n a_2^n a_1^m a_2^m$ n'appartient pas à D
(D, ., ε) n'est donc pas un sous-monoïde de (V*, ., ε).

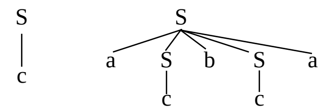
E inclus dans V*
quel que soit x appartenant à E, x comprend n a_1 et n a_2
quel que soit y appartenant à E, y comprend m a_1 et m a_2
x . y comprend (n+m) a_1 et (n+m) a_2 donc x . y appartient à E

ε comprend 0 a_1 et 0 a_2 donc ε appartient à E
(E, ., ε) est donc un sous-monoïde de (V*, ., ε).

2.

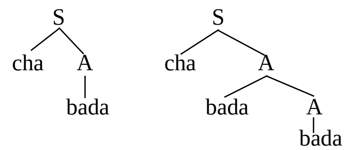
- G_1 de type 2 - grammaire hors-contexte
  $S \vdash c$ $|c| = 1$

  $S \vdash aSbSa \vdash aaSbSabSa \vdash^* aacbcabca$ $|aacbcabca| = 9$



- G_2 de type 0
  car règle Cbb -> da avec |Cbb| = 3 et |da| = 2
  ou encore règle A -> ε
  $S \vdash BCaCbbA \vdash^* chabaCbbA \vdash^* chabada$ |chabada| = 6 car ch est 1 symbole
  $S \vdash BCaCbbA \vdash ddCaCbbCaCbbA \vdash ddCaCbbCaCbbCaCbb \vdash^* chabadabadabada$
  |chabadabadabada| = 14

- G_3 de type 3 - grammaire régulière
  $S \vdash chaA \vdash chabada$ |chabada| = 2
  $S \vdash chaA \vdash chabadaA \vdash chabadabada$ |chabadabada| = 3



- G_4 de type 2 - hors-contexte
  $S \vdash Aa \vdash Saa \vdash bAaa$ ...
  Cette grammaire n'engendre pas de chaîne terminale
  $L(G_4) = \emptyset$

- G_5 de type 1
  $S \vdash AB \vdash aAB \vdash acABc \vdash accc$

3.

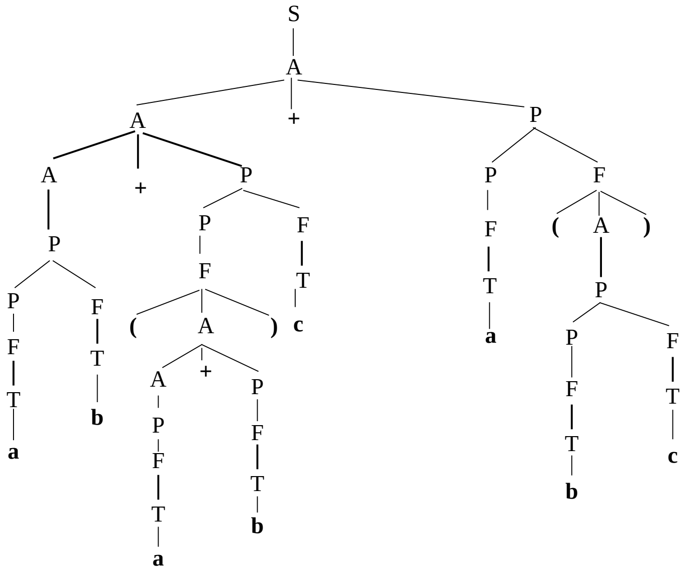
On peut commencer par réécrire la grammaire en changeant les noms des terminaux et en faisant apparaître les disjonctions (afin de simplifier les notations).

On obtient la grammaire G' qui engendre le même langage que G

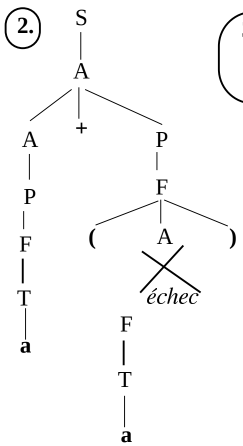
G' : ({a, b, ..., z, +, (,)}, {S, A, P, F, T}, S,
{ $S \to A$
$P \to P\ \ F \mid F$
$F \to (\ A\ ) \mid\ T$

$A \rightarrow A + P \mid P$
$T \rightarrow a \mid b \mid c \mid ... \mid y \mid z$ })

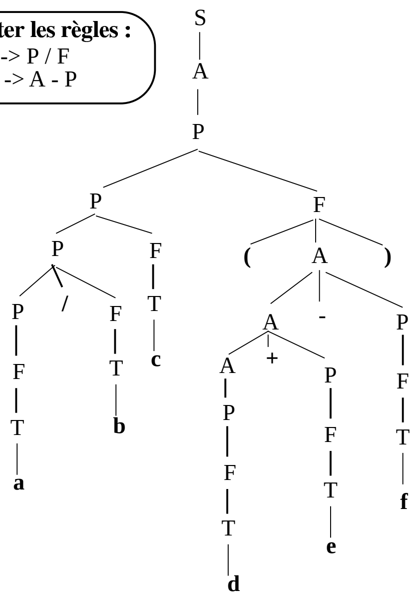
1



2.



3. Ajouter les règles :
P -> P / F
A -> A - P



4.

G1 type 0 (à cause de la règle $b\,B \rightarrow a$ ce n'est pas le type 1).

G2 type2 (hors-contexte)
L(G2) = {(⌐¬)^n, n > 0}
Si on choisit la règle $S \rightarrow U\,V$ pour terminer la dérivation, on a un nombre impair de ⌐¬, sinon avec la règle $A \rightarrow V\,U$ on en a un nombre pair

G3 type3 (régulière)
Les chaînes engendrées par G3 sont constituées en trois parties
- une suite de a et b,
- une suite uniquement de a ou uniquement de b,
- une suite de a et b.

G4 type 2
S se dérive obligatoirement en A, et A se dérive obligatoirement en S.
Cette grammaire n'engendre aucune chaîne terminale.
L(G4) = { }
Faire remarquer que { } (ensemble vide) est différent de {ε} (ensemble ne comprenant que la chaîne vide).

G5 type 3
L(G5) = {ε, $a^n b$ avec n positif ou nul}

G6 type 2
L(G6) = {ε, $a^n b^n$ avec n positif }

G7 type0
S1 peut se dériver en ε

G8 type 0
S peut se dériver en ε et S apparaît en partie droite d'une règle de grammaire

5.

Pour chaque grammaire faire éventuellement la génération d'une ou deux chaînes.

1. $V_T = \{a, b\}$ $L = \{w \in V_T^* \mid w = a^n b^n \text{ avec } n > 0\}$
   G = ({a, b}, {S}, S,
   { $S \rightarrow a\,S\,b \mid a\,b$ })
   type 2 (hors-contexte)

2. $V_T = \{a, b, c\}$ $L = \{w \in V_T^* \mid w = a^m b^n c^p \text{ avec } m > 0, n > 0 \text{ et } p > 0\}$
   G = ({a, b, c}, {S, S_1, S_2}, S,
   { $S \rightarrow a\,S \mid a\,S_1$
   $S_1 \rightarrow b\,S_1 \mid b\,S_2$
   $S_2 \rightarrow c\,S_2 \mid c$ })
   type 3 (régulière)

3. $V_T = \{a, b, c\}$ $L = \{w \in V_T^* \mid w = a^n b^n c^p \text{ avec } n > 0 \text{ et } p > 0\}$

$G = (\{a, b, c\}, \{S, S_1, S_2\}, S,$
$\{ S \rightarrow S_1\ S_2$
$S_1 \rightarrow a\ S_1\ b \mid ab$
$S_2 \rightarrow c\ S_2 \mid c \quad \})$
type 2 (hors-contexte)

4. $V_T = \{a, b, c\}$ $L = \{w \in V_T^* \mid w = a^p b^q c^r$ avec $(p = q$ ou $q = r)$ et $p > 0$ et $q > 0$ et $r > 0\}$

$G = (\{a, b, c\}, \{S, S_1, S_2, T_1, T_2\}, S,$
$\{ S \rightarrow S_1\ S_2 \mid T_1\ T_2$
$S_1 \rightarrow a\ S_1\ b \mid ab$
$S_2 \rightarrow c\ S_2 \mid c$
$T_1 \rightarrow a\ T_1 \mid a$
$T_2 \rightarrow b\ T_2\ c \mid bc \quad \})$
type 2 (hors-contexte)

5. $V_T = \{a, c\}$ $L = \{w \in V_T^* \mid w = a^m\ c\ a^p$ avec $m \geq p > 0\}$

$G = (\{a, c\}, \{S, S_1\}, S,$
$\{ S \rightarrow a\ S\ a \mid a\ S_1\ a$
$S_1 \rightarrow a\ S_1 \mid c \quad \})$
type 2 (hors-contexte)

6. $V_T = \{a, b\}$ $L = \{w \in V_T^* \mid w = a^p b^q$ avec $p \neq q$ et $p \geq 0$ et $q \geq 0\}$

$G = (\{a, b\}, \{S, S_1, S_2\}, S,$
$\{ S \rightarrow a\ S\ b \mid S_1 \mid S_2$
$S_1 \rightarrow a\ S_1 \mid a$
$S_2 \rightarrow b\ S_2 \mid b \quad \})$
type 2 (hors-contexte)

7. $V_T = \{a, b, c\}$ $L = \{w \in V_T^* \mid w = a^p b^q c^r\ ;\ p + q \geq r\ ;\ p > 0,\ q > 0$ et $r \geq 0\}$
une règle crée autant de a et de b que de c, puis on ajoute des a et / ou des c

$G = (\{a, b, c\}, \{S, S_1\}, S,$
$\{ S \rightarrow a\ S\ c \mid a\ S \mid a\ S_1 \mid a\ S_1 c$
$S_1 \rightarrow b\ S_1\ c \mid b\ S_1 \mid b \mid b\ c\})$
type 2 (hors-contexte)

6.

6.1

version 1
<S> ::= aa <A> | bb <B>
<A> ::= ba <B> | babb
<B> ::= ab <A> | abaa

version 2
On optimise la grammaire afin d'éviter des règles avec des disjonctions dans le cas où la chaîne engendrée/analysée a un début identique, comme ici : <A> ::= ba <B> | babb. Si la reconnaissance d'une chaîne échoue sur <B> on va réanalyser ba !

<S> ::= aaba <A'> | bbab <B'>
<A'> ::= ab <B'> | bb
<B'> ::= ba <A> | aa

ou
<S> ::= aa <A> | bb <B>
<A> ::= ba <A'>
<A'> ::= <B> | bb
<B> ::= ab <B'>
<B'> ::= <A> | aa

6.2

version 1
<A> ::= <liste_X> <B>
<liste_X> ::= babb <liste_X> | ba
<B> ::= <liste_Z> | ε
<liste_Z> ::= <liste_Y> a <liste_Z> | <liste_Y> a
<liste_Y> ::= abaa <liste_Y> | ab

mais cette grammaire est de type 0 à cause de **<B> ::=** ε
Il faut la réécrire

version 2
<A> ::= <liste_X> <B> | <liste_X>
<liste_X> ::= babb <liste_X> | b
<B> ::= <liste_Y> a <B> | <liste_Y> a
<liste_Y> ::= abaa <liste_Y> | ab

<liste_Z> a été renommé **<B>**

6.3

<A> ::= aa | bb | cc | dd

6.4

version 1
<A> ::= aa <suiteA1> | <suiteA1> | aa | ε
<suiteA1> ::= bb <suiteA2> | <suiteA2> | bb
<suiteA2> ::= cc <suiteA3> | <suiteA3> | cc
<suiteA3> ::= dd

7.

Il faut voir l'expression comme parenthésée

$G = (\{a, +, =\}, \{S, S_1, S_2\}, S,$
$\{ S \rightarrow a\ S_2\ a$
$S_2 \rightarrow a\ S_2\ a \mid +\ S_1$
$S_1 \rightarrow a\ S_1\ a \mid a = a \quad \})$
type 2

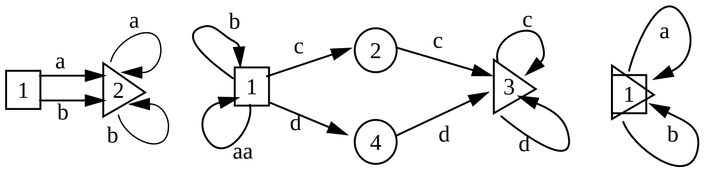
8.

Remarque : on doit toujours obtenir une grammaire régulière.
Technique : à chaque état on fait correspondre un symbole non terminal, on réécrit éventuellement la grammaire

1. <S> := a <A> | b<B>
   <A> := b <A> | c <B> | a
   <B> := a <C>
   <C> := a <C> | ε
   Les deux dernières règles doivent être réécrites : <B> := a <B> | a
2. <S> := a <A> | ε
   <A> := b <B> | b
   <B> := c <S>
   doit être réécrit :

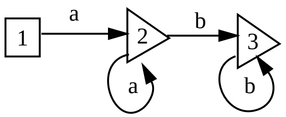
   <S> := a <A> | ε
   <A> := b <B> | b
   <B> := c <C> | c
   <C> := a <A>

9. Plus difficile qu'il n'y parait : faire dériver quelques chaînes

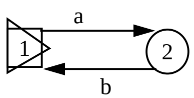


10.

1



2



3