Université Côte d'Azur — L3 Info, L3 MI, L3 ST
Automates & Langages — 2023–24

# TD n° 7

## Grammaires hors-contexte (suite)

**Exercice 1)** Considérons la grammaire $G$ suivante :

Grammaire $G$

Axiome = S
N = {S, X, Y, Z, T}
T = {0, 1, 2}
P = { S → X Y | Z T
X → 0 X 1 | 0 1
Y → 2 | Y 2
Z → 0 | Z 0
T → 1 T 2 | 1 2 }

1. Cette grammaire est déjà *propre* (elle ne contient ni $\varepsilon$-transitions, ni renommages, ni symboles improductifs). Mettez-la sous Forme Normale de Chomsky.
2. Procédez à l'analyse du mot $001122$ par l'algorithme de Cocke, Younger et Kasami (CYK).
3. Ce mot $001122$ appartient-il au langage $L(G)$ ? Possède-t-il des facteurs dans $L(G)$ ? Quel est le langage engendré par la grammaire $G$ ?
4. La grammaire $G$ initiale est-elle ambiguë ? Pourquoi ?

**Exercice 2)** Considérons la grammaire hors-contexte $G$ ci-dessous :

Grammaire $G$

Axiome = S
N = {S, W, Z}
T = {0, 1, 2}
P = { S → Z W | 1 S 1 | 2
W → S Z
Z → 0 }

1. La grammaire $G$ est propre. Mettez-la sous Forme Normale de Chomsky à droite (toutes les productions seront de la forme $A \to BC$ ou $A \to a$ avec $A, B, C$ non-terminaux et $a$ terminal).
2. Analysez le mot $01020102$ grâce à l'algorithme de Cocke, Younger et Kasami (CYK) afin de décider de son appartenance à $L(G)$. Citez également tous ses facteurs appartenant à $L(G)$.
3. Finalement, quel est le langage engendré par la grammaire $G$ ?

## Exercices complémentaires

**Exercice 3)** Considérons ci-dessous à gauche la grammaire hors-contexte $G$ *(on note le terminal 'ε' entre ' ' afin de le distinguer du mot vide habituel ε)* :

Grammaire $G$

Axiome = S
N = {S}
T = {∅, 'ε', 0,1,(,),+, *}
P = { S → ∅ | 'ε' | 0 | 1
S → ( S + S )
S → ( S S )
S → S * }

1. Quel est l'ensemble bien connu $L(G)$ engendré par la grammaire $G$ ?
2. Mettez la grammaire $G$ sous forme normale de Chomsky
3. Analysez le mot ( ( 0 + 1 ) * 0 ) à l'aide de l'algorithme de Cocke, Younger et Kasami (CYK).
4. Dites si le mot $((0 + 1)^*0)$ appartient à $L(G)$ et énumérez tous ses facteurs dans $L(G)$.

**Exercice 4)** On considère la grammaire $G$ suivante :

Grammaire $G$

Axiome = C
N = {C, S, $S_1$, $S_2$, A, W, V, U, T}
T = {'begin', 'end', 'pred', 'succ', ' :=', 'while', 'do', ' ;', '<>', '(', ')', '0', 'x', 'y'}
P = { C → 'begin' $S_1$ 'end'
$S_1$ → S $S_2$
$S_2$ → ' ;' $S_1$ | ε
S → A | W | C
A → V ' :=' T
T → 'pred' (' V ')' | 'succ' (' V ')' | '0'
W → 'while' V '<>' V 'do' S
V → 'x' | 'y' }

Le programme suivant est-il syntaxiquement correct, relativement à la grammaire $G$ précédente ?

```
begin
   while x <> y do
      begin
         x := 0 ;
         y := succ(x) ;
      end
end
```