POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

LOG1810

STRUCTURES DISCRÈTES

TD 12 : MODÉLISATION COMPUTATIONNELLE

H2025

SOLUTIONNAIRE

Exercice 1 :

Partie A

Pour chacune des grammaires ci-dessous, déterminez leur type en justifiant vos réponses, en commençant par les grammaires de type 3 et en progressant vers celles moins restrictives.

a) Considérez la grammaire $G_1 = (V_1, T_1, S, P_1)$ où $V_1 = \{a, b, S, A, B\}$ et $T_1 = \{a, b\}$. L'axiome est S, et l'ensemble des règles de production P_1 est le suivant :

$$S \rightarrow aA$$
$$A \rightarrow bB \mid b$$
$$B \rightarrow bA \mid \epsilon$$

Solution

- Type 3 : Elle n'est pas de type 3 à cause de la présence de la règle de production $B \rightarrow \epsilon$. Dans une grammaire de Type 3, seule l'axiome est autorisé à produire une chaîne vide.

- Type 2 : Elle est de type 2, car tous les symboles à gauche dans les productions sont des symboles uniques non terminaux.

Conclusion : G_1 est de type 2.

b) Considérez la grammaire $G_2 = (V_2, T_2, S, P_2)$ où $V_2 = \{a, b, S, A, B\}$ et $T_2 = \{a, b\}$. L'axiome est S, et l'ensemble des règles de production P_2 est le suivant :

$$S \rightarrow aA \mid \epsilon$$
$$A \rightarrow bB$$
$$B \rightarrow bA \mid a \mid b$$

Solution

- Type 3 : Elle est de type 3 car toutes les règles de production sont de la forme $w_1 \rightarrow a|aA \; ou \; S \rightarrow \epsilon$.

Conclusion : G_2 est de type 3.

c) Considérez la grammaire $G_3 = (V_3, T_3, S, P_3)$ où $V_3 = \{a, b, S, A, B\}$ et $T_3 = \{a, b\}$. L'axiome est S, et l'ensemble des règles de production P_3 est le suivant :

$$S \rightarrow AB$$
$$AB \rightarrow BA$$
$$A \rightarrow b$$
$$B \rightarrow a$$

Solution

- Type 3 : Elle n'est pas de type 3 car certaines des règles de production ne sont pas de la forme $w_1 \rightarrow a|aA \; ou \; S \rightarrow \epsilon$ $(S \rightarrow AB \; et \; AB \rightarrow BA)$.

- Type 2 : Elle n'est pas de type 2, à cause de la présence de la règle de production $AB \rightarrow BA$, où la partie gauche n'est pas un symbole unique non terminal.

- Type 1 : Elle n'est pas de type 1, à cause de la présence de la règle de production $AB \rightarrow BA$, la règle de production ne respecte pas $\alpha A \beta \rightarrow \alpha \gamma \beta$.

- Type 0 : Comme la grammaire n'est pas de type 1,2 ou 3, alors elle est de type 0.

Conclusion : G_3 est de type 0.

Partie B

Déterminez si la chaine $w_1 = abbba$ peut être générés par les grammaires G_1, G_2 et G_3.

Solution

La grammaire G_1 ne génère pas le mot w_1. Pour montrer cela, nous analyserons les contraintes imposées par la grammaire.

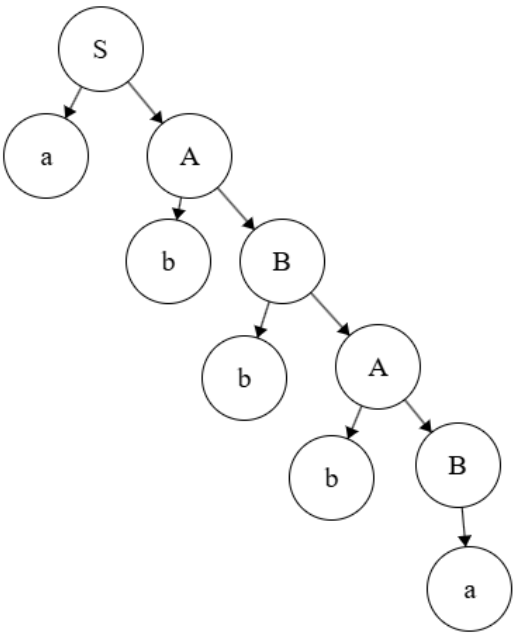
La grammaire G_1 impose de commencer avec un a ($S \rightarrow aA$). Ensuite, il n'est plus possible d'avoir un autre a. Les autres règles ne permettent que d'ajouter que des b à la séquence, il ne sera donc pas possible de terminer la séquence par un a.

La grammaire G_2 peut générer le mot w_1. Il est possible de procéder par la chaîne de dérivation ou l'arbre de dérivation pour le montrer.

Chaine de dérivation

$S \rightarrow aA$
$S \rightarrow abB$ $(A \rightarrow bB)$
$S \rightarrow abbA$ $(B \rightarrow bA)$
$S \rightarrow abbbB$ $(A \rightarrow bB)$
$S \rightarrow abbba$ $(B \rightarrow a)$

Arbre de dérivation



La grammaire G_3 ne génère pas le mot w_1. Pour montrer cela, nous analyserons les contraintes imposées par la grammaire.

La grammaire G_3permet de générer uniquement 2 mots : ab et ba. Effectivement, nous avons d'abord $S \rightarrow AB$. Il y a ensuite deux possibilités. La première est d'utiliser directement les règles $A \rightarrow b$ et $B \rightarrow a$, ce qui engendre le mot ba. Il y a ensuite la deuxième possibilité qui est d'utiliser la règle $AB \rightarrow BA$ puis $A \rightarrow b$ et $B \rightarrow a$ ce qui donnent le mot ab. Ainsi, il n'est pas possible de générer le mot $w_1 = abbba$ à partir de la grammaire G_3.

Exercice 2

Pour chacun des langages suivants, construisez un automate fini déterministe le reconnaissant. Considérez l'ensemble des symboles terminaux $I = \{a, \mathrm{b}\}$. Donnez ensuite l'expression régulière représentant ce langage. Justifiez vos réponses.

a) Le langage des mots qui contient un nombre de $a \equiv 1\ mod(4)$.

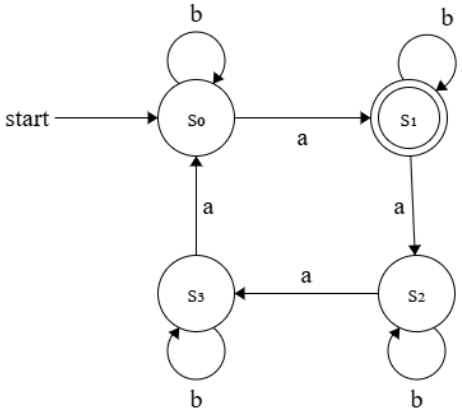
Solution

Considérons 4 états possibles. L'état correspondant à un nombre de a divisible par 4 (nombre de a congru à 0 modulo 4) : s_0. L'état correspondant à un nombre de a congru à 1 modulo 4 : s_1.L'état correspondant à un nombre de a congru à 2 modulo 4 : s_2. Et finalement l'état s_4 qui correspond à un nombre de a congru à 3 modulo 4. Le seul état terminal est l'état s_1 car le seul état ayant nombre de $a \equiv 1\ mod(4)$.

Initialement, nous avons zéro a, ce qui est un nombre de a divisible par 4. L'état de départ est donc s_0 . On reste à s_0 tant qu'on n'ajoute pas un a. Ensuite, quand on ajoute un a, on passe à l'état s_1, qui est terminal, et on reste à cet état tant que l'on n'ajoute pas un autre a.

Si on ajoute un autre a, on passe à l'état s_2 et comme précédemment, on reste à s_2 tant que l'on n'ajoute pas un nouveau a. On passe à s_3 si l'on ajoute de nouveau un a. Finalement, on passe à l'état s_0 à l'ajout d'un autre a et ainsi de suite.

L'automate fini est donc simplement :



Le langage reconnu est donc :

$$L = b^*ab^*\{b^*ab^*ab^*ab^*ab^*\}^*$$

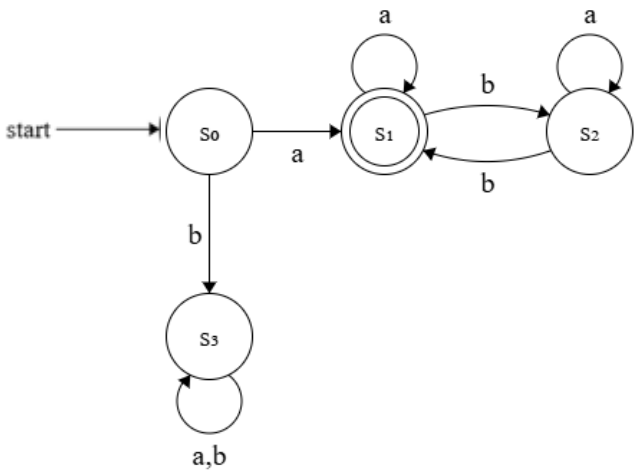
b) Le langage des mots qui commence par a suivi d'une chaine de caractère contenant un nombre pair de b.

Solution

Nous aurons besoins de 4 états.

- Un état de départ s_0.
- Un état « poubelle » s_3 qui boucle sur lui-même dans le cas où le premier caractère de la séquence est un b.
- Un état s_1 correspondant à un nombre de b pair (état terminal).
- Un état s_2 correspondant à un nombre de b impair.

L'automate fini est donc simplement :



Le langage reconnu est donc :

$$L = a^{+}\{ba^{*}ba^{*}\}^{*}$$

Exercice 3 :

Pour les langages suivants, proposez une grammaire $G = (V, T, S, P)$ qui engendre le langage. Précisez *V*, *T*, *S* et *P*.

a) Soit le langage construit sur l'alphabet $I = \{a, b\}$ qui ne contient qu'un seul a. Proposez une grammaire de type 3.

Solution *

$G = (V, T, S, P)$
$V = \{a, b, A, S\}$
$T = \{a, b\}$
S est l'axiome
P est constitué des productions suivantes :

$S \rightarrow bS \mid aA \mid a \mid \epsilon$
$A \rightarrow bA \mid$ b

*plusieurs solutions sont possibles.

b) Soit le langage $L_2 = \{a^n b^m \mid n \neq m\}$ construit sur l'alphabet $I = \{a, b\}$.

Solution *

$G = (V, T, S, P)$
$V = \{a, b, A, B, S\}$
$T = \{a, b\}$
S est l'axiome
P est constitué des productions suivantes :

$S \rightarrow A \mid B$
$A \rightarrow aAb \mid aA \mid a$
$B \rightarrow aBb \mid bB \mid b$

Explications :

- A gère les mots où $n > m$. La règle $A \rightarrow aAb$ permet d'ajouter le nombre m de b en s'assurant d'ajouter le même nombre de a. Les autres règles de A permettent d'ajouter un nombre de a supplémentaire afin d'avoir $n > m$.
- B gère les mots où $n < m$. La règle $B \rightarrow aBb$ permet d'ajouter le nombre n de a en s'assurant d'ajouter le même nombre de b. Les autres règles de B permettent d'ajouter un nombre de b supplémentaire afin d'avoir $n < m$.

*plusieurs solutions sont possibles.

c) Soit le langage $L_3 = \{a^n b^m c^k \mid n = m \; ou \; m = k\}$ construit sur l'alphabet $I = \{a, b, c\}$.

Solution *

$G = (V, T, S, P)$
$V = \{a, b, c, A, B, C, D, S\}$
$T = \{a, b, c\}$
S est l'axiome
P est constitué des productions suivantes :

$S \rightarrow AB \mid CD$
$A \rightarrow aAb \mid \epsilon$
$B \rightarrow cB \mid \epsilon$
$C \rightarrow aC \mid \epsilon$
$D \rightarrow bDc \mid \epsilon$

Explications :

Le langage L_3 contient toutes les chaînes de la forme $a^n b^m c^k$ telles que le nombre de a est égal au nombre de b, ou le nombre de b est égal au nombre de c.

Ce langage peut être vu comme l'union de deux sous-langages :

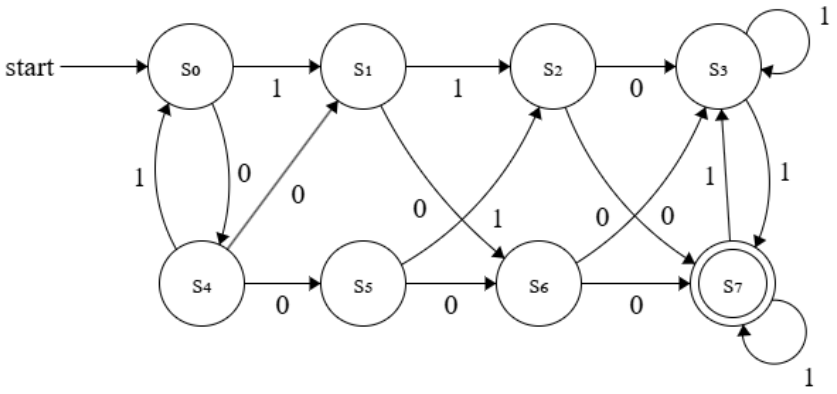
- $L_{3,1} = \{a^n b^n c^k \mid n, k \in \mathbb{N}\}$
- $L_{3,2} = \{a^n b^k c^k \mid n, k \in \mathbb{N}\}$

La branche $S \rightarrow AB$ traite le premier cas. A permet d'ajouter le même nombre de a et b.
B permet ensuite d'ajouter un nombre quelconque de c.

La branche $S \rightarrow CD$ traite le deuxième cas. C permet d'ajouter un nombre quelconque de a.
D permet ensuite d'ajouter le même nombre de b et c.

Exercice 4 :

Transformez en automate déterministe l'automate suivant.



Solution

Table d'états-transition de l'automate initial :

États	Entrées	
	0	1
$\rightarrow S_0$	$\{S_4\}$	$\{S_1\}$
S_1	$\{S_6\}$	$\{S_2\}$
S_2	$\{S_3, S_7\}$	$\emptyset$
S_3	$\emptyset$	$\{S_3, S_7\}$
S_4	$\{S_1, S_5\}$	$\{S_0\}$
S_5	$\{S_6\}$	$\{S_2\}$
S_6	$\{S_3, S_7\}$	$\emptyset$
$\leftarrow S_7$	$\emptyset$	$\{S_3, S_7\}$

Table d'états-transition de l'automate déterministe :

États	Entrées	
	a	b
$\rightarrow \{S_0\}$	$\{S_4\}$	$\{S_1\}$
$\{S_4\}$	$\{S_1, S_5\}$	$\{S_0\}$
$\{S_1\}$	$\{S_6\}$	$\{S_2\}$
$\{S_1, S_5\}$	$\{S_6\}$	$\{S_2\}$
$\{S_6\}$	$\{S_3, S_7\}$	$\emptyset$
$\{S_2\}$	$\{S_3, S_7\}$	$\emptyset$
$\leftarrow \{S_3, S_7\}$	$\emptyset$	$\{S_3, S_7\}$

L’automate est :