POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

LOG2810
STRUCTURES DISCRÈTES

TD 13 : **MODÉLISATION COMPUTATIONNELLE**

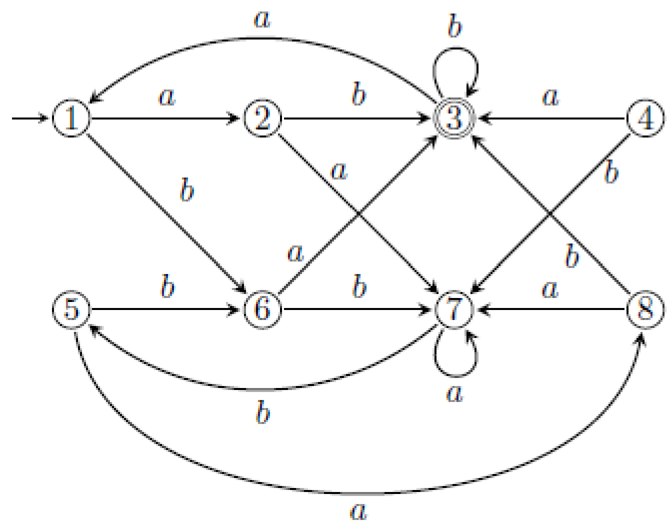
H2022

SOLUTIONNAIRE

Directives pour la remise :

- La remise est individuelle, mais le travail en équipe est encouragé.
- La remise est individuelle se fait à la fin de la séance de TD.
- Répondez directement sur ce document Word (docx). Dans l'intérêt de l'équité pour tous les étudiants, vous devez modifier le fichier Word. Modifiez le fichier **EXCLUT** le fait d'intégrer des scans de rédaction manuscrite ou d'y écrire avec un stylet.
- Lorsque vous avez terminé, générez un PDF avec le nom sous le format :
  Matricule-TDNuméro.pdf (exemple : 1234567-TD1.pdf).
- Téléversez votre fichier PDF dans la boîte de remise située dans la Zone TDs de la page Moodle du cours.
- Choisissez la boîte de remise qui correspond à votre section de TD.
- **Aucun retard et aucune remise par courriel ne seront acceptés.**
- **Le non-respect des consignes entraînera automatiquement la note 0 pour ce TD.**

Exercice 1. En utilisant les règles de transformation vues en cours, donnez une grammaire qui engendre le langage reconnu par l'automate ci-dessous.



Réponse :

Soit les symboles non terminaux associés aux états de l'automate comme suit :

- État 1 : S
- État 2 : A
- État 3 : B
- État 4 : C
- État 5 : D
- État 6 : E
- État 7 : F
- État 8 : H

La grammaire G = (V, T, S, P) est tel que V = {a, b, A, B, C, D, E, F, H, S}, T = {*a, b*} et P l'ensemble des productions :

S → aA | bE
A → b | bB | aF
B → b | bB | aS
C → a | aB | bF
D → bE | aH
E → a | aB | bF
F → bD | aF
H → b | bB | aF

Exercice 2. Soit la grammaire ***G*** *= (V, T, 1, P)* où ***V*** *= {0, 1, 2, 3, a, b, c}*, ***T*** *= {a, b, c}*. ***1*** est l'axiome, ***P*** l'ensemble des règles de production :

1 → a0 | b2| c1 |Ɛ
0 → b0 | a3 | c
2 → a1 | b| c3
3 → a| b | c0

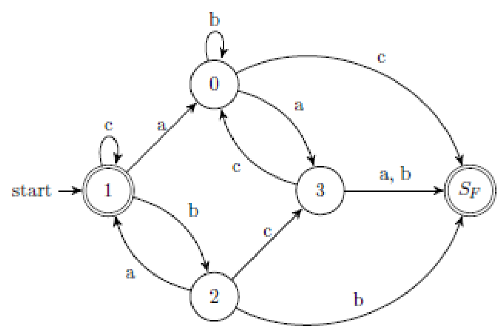
En utilisant les règles de transformation vues en cours, construisez l'automate qui reconnaît le langage généré par la grammaire ***G***. Vous pouvez juste donner la table d'états-transition et préciser les étaux finaux ou acceptants.

Réponse :

- Table d'états-transition

États	Entrée		
	a	b	c
1	0	2	1
0	3	0	S_F
2	1	S_F	3
3	S_F	S_F	0

- États finaux : 1 et S_F.
- Automate



Exercice 3. Construisez un automate fini reconnaissant sur l'alphabet ***V*** = *{a, b, c}* l'expression rationnelle : ***a((ba)²cb*)*.***
Vous pouvez juste donner la table d'états-transition et préciser les états finaux.

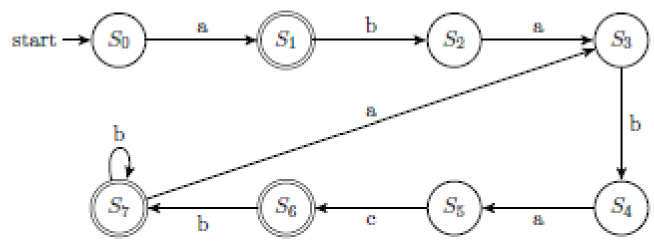
Réponse :

- Table d'états-transition

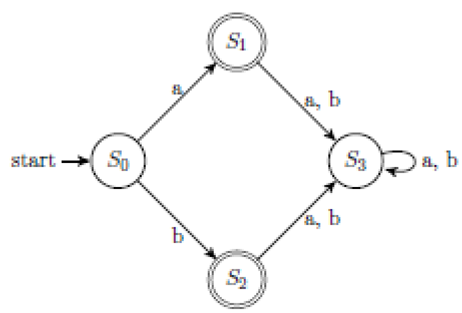
États	Entrées		
	a	b	c
S_0	S_1		
S_1		S_2	
S_2	S_3		
S_3		S_4	
S_4	S_5		
S_5			S_6
S_6		S_7	
S_7	S_3	S_7	

- États finaux : S_1, S_6 et S_7.

- Automate



Exercice 4. Minimisez l'automate ci-dessous. Vous pouvez juste donner la table d'états-transition et préciser les états finaux.

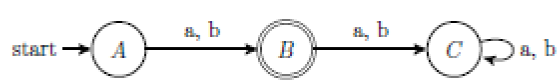


Réponse :

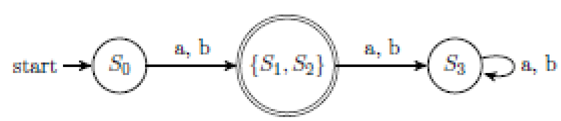
- Table d'états-transition

États	Entrées	
	a	b
A = {S_0}	B	B
B = {S_1, S_2}	C	C
C = {S_3}	C	C

- État final : B.
- Automate

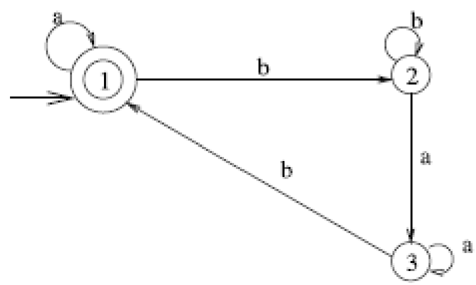


OU



Exercice 5. En utilisant le lemme d'Arden, trouvez le langage reconnu par la machine à états finis suivante. Justifiez votre réponse.

Note : Il existe certaines identités avec les expressions régulières. Elles peuvent parfois être utilisées pour réécrire une expression. **Exemples** : $(a+b)^* = (a^*b)^*a^*$ $(a+b)^* = a^*(ba^*)^*$



Réponse : .

Soit Y1, Y2 et Y3 les étiquettes associées aux états 1, 2 et 3, respectivement.

Le système d'équation s'écrit :

- Y1 = aY1 + bY2 + Ɛ
- Y2 = bY2 + aY3
- Y3 = aY3 + bY1

En appliquant le lemme d'Arden, on a :

- Y3 = a*bY1
- Y2 = bY2 + aY3
  Y2 = bY2 + aa*bY1
  Y2 = b*aa*bY1
- Y1 = aY1 + bY2 + Ɛ
  Y1 = aY1 + b(b*aa*bY1) + Ɛ
  Y1 = aY1 + bb*aa*bY1 + Ɛ
  Y1 = (a + bb*aa*b)Y1 + Ɛ
  Y1 = (a + bb*aa*b)*Ɛ
  Y1 = (a + bb*aa*b)*

Le langage reconnu par la machine à état est donc (a + bb*aa*b)*.

Note : En tenant compte des notes écrites dans l'énoncé, l'expression (a + bb*aa*b)* peut être réécrite autrement. Exemples : **(a*bb*aa*b)*a*** ou encore **a*(bb*aa*ba*)***.

Exercice 6. Montrez que le langage ***L*** n'est pas régulier.

$$L = \{ 0^{2n}1^n, n \in \mathbb{N} \}$$

Réponse :

Raisonnons par l'absurde. Supposons que le langage est régulier. Il vérifie donc le lemme de pompage. Soit p la constante du lemme de pompage.

Le mot m = $0^{2p}1^p$ est un mot de ***L***. La décomposition m = uvw tel que u = 0^i, v = 0^j et w = $0^{(2p-i-j)}1^p$ avec i+j < p et j ≥ 1 satisfait aux conditions du lemme de pompage. Ainsi, le mot uv^0w devrait être aussi un mot de ***L***.

$uv^0w = uw$

$uv^0w = 0^i0^{(2p-i-j)}1^p$

$uv^0w = 0^{(2p-j)}1^p$

$0^{(2p-j)}1^p$ n'est pas un mot du langage. Puisque j ≥ 1 on a (2p-j) ≠ 2p. Le lemme de pompage n'est donc pas vérifié.

D'où le langage ***L*** n'est pas régulier.

Exercice 7. Soit ***T*** la machine de Turing dont l'état initial est S_0 et définie par les 5-tuples :

(S_0, 0, S_1, 1, R) ; (S_0, 1, S_1, 0, R) ; (S_0, B, S_1, 0, R) ; (S_1, 0, S_2, 1, L) ; (S_1, 1, S_1, 0, R) ; (S_1, B, S_2, 0, L)

En considérant le ruban initial suivant, déterminez le ruban final lorsque ***T*** s'arrête. On suppose que *T* commence en position initiale.

...	B	B	1	1	B	0	1	B	...

Réponse :

a. Position initiale

			S_0						
...	B	B	1	1	B	0	1	B	...

b.

				S_1					
...	B	B	0	1	B	0	1	B	...

c.

					S_1				
...	B	B	0	0	B	0	1	B	...

d.

				S_2					
...	B	B	0	0	0	0	1	B	...

e. La machine s'arrête. Le ruban final est donc

...	B	B	0	0	0	0	1	B	...

Exercice 8 (facultatif). Soit ***T*** la machine de Turing dont l'état initial est S_0 et définie par les 5-tuples :

(S_0, 0, S_1, 1, R) ; (S_0, 1, S_1, 0, R) ; (S_0, B, S_1, 0, R) ; (S_1, 0, S_2, 1, L) ; (S_1, 1, S_1, 0, R) ; (S_1, B, S_2, 0, L)

En considérant le ruban initial suivant, déterminez le ruban final lorsque ***T*** s'arrête. On suppose que *T* commence en position initiale.

...	B	B	1	0	1	B	B	B	...

Réponse :

a. Position initiale

			S_0						
...	B	B	1	0	1	B	B	B	...

b.

				S_1					
...	B	B	0	0	1	B	B	B	...

c.

			S_2						
...	B	B	0	1	1	B	B	B	...

d. La machine s'arrête. Le ruban final est donc

...	B	B	0	1	1	B	B	B	...

Exercice 9 (facultatif). Le nom du groupe de musique ABBA est un exemple de palindrome. La ville de LAVAL est un autre exemple. Voici une grammaire qui génère tous les palindromes sur l'alphabet $\{a, b\}$

$S \rightarrow \lambda$
$S \rightarrow a$
$S \rightarrow b$
$S \rightarrow aSa$
$S \rightarrow bSb$

Utilisez le lemme de pompage afin de démontrer que l'ensemble des palindromes définit sur $\{a, b\}$ n'est pas régulier.

Indice : Considérez par exemple la chaîne $a^n bba^n$

Réponse :

Supposons que l'ensemble soit régulier. Soit p la valeur du lemme de pompage. Considérons la chaîne $\omega = a^p bba^p$ qui est clairement un palindrome et $|\omega| \geq p$. Par le lemme de pompage, il doit exister des sous-chaînes x, y et z satisfaisant les trois contraintes du lemme de pompage.

Donc choisissons n'importe quel x, y et z tel que $\omega = xyz$, $|xy| \leq p$ et $|y| \geq 1$. Parce que $|xy| \leq p$, xy est entièrement contenu dans le a^p au début de ω. Donc x et y sont entièrement constitués de a, c'est-à-dire que $x = a^i$ et $y = a^j$. Alors z doit ressembler à $a^k bba^p$ où $i + j + k = p$.

Maintenant, considérons $xy^m z$. Par le lemme de pompage, avec $m = 0$, xz doit être dans le langage.

Mais $xz = a^i a^k bba^p = a^{i+k} bba^p$. De $|y| \geq 1$, on sait que $j \geq 1$. Donc $i + k < p$. Cela signifie que xz n'est pas un palindrome, car les nombres de a aux deux extrémités ne correspondent pas à p.

Cela signifie que l'ensemble des palindromes ne satisfait pas le lemme de pompage et, ainsi, l'ensemble des palindromes définit sur l'ensemble $\{a, b\}$ ne peut pas être régulier.