POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

LOG2810
STRUCTURES DISCRÈTES

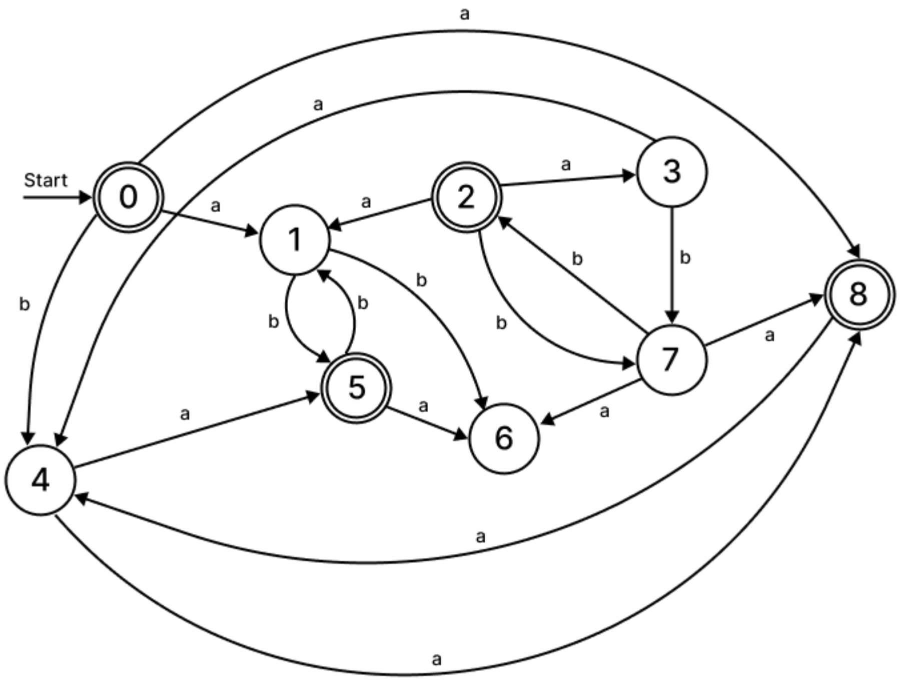
TD 13 : MODÉLISATION COMPUTATIONNELLE

H2023

SOLUTIONNAIRE

Exercice 1.

Transformez en automate déterministe l'automate suivant. Donnez la table d'états-transition, précisez les états finaux ou acceptants et construisez l'automate déterministe que vous proposez.



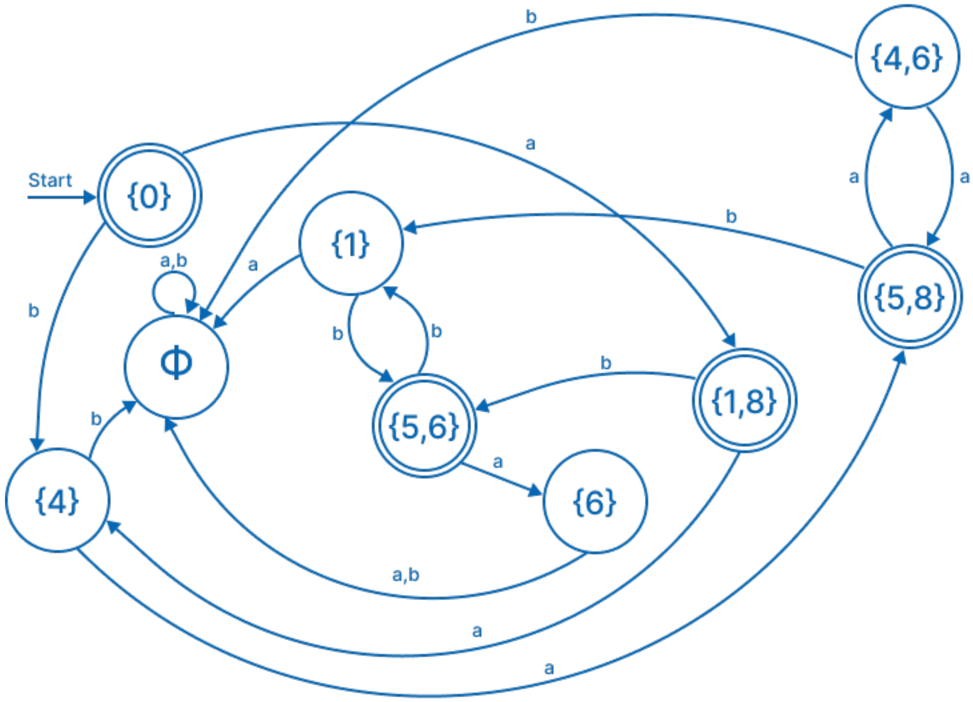
Réponse :

- Table d'états-transition

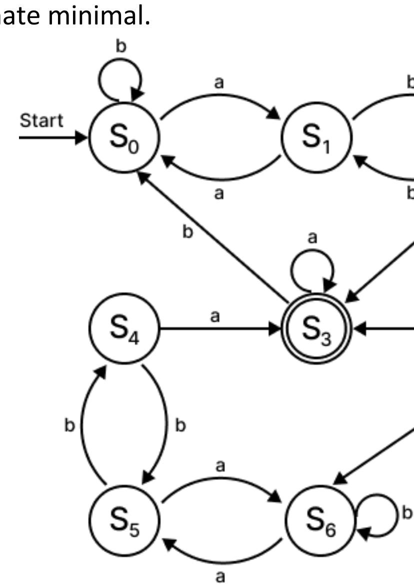
États	Entrée	
	a	b
{0}	{1,8}	{4}
{1,8}	{4}	{5,6}
{4}	{5,8}	∅
{5,6}	{6}	{1}
{5,8}	{4,6}	{1}
{6}	∅	∅
{1}	∅	{5,6}
{4,6}	{5,8}	∅

- États finaux : {0}, {1,8}, {5,6} et {5,8}.

- Automate



Exercice 2.
Vous êtes un.e ingénieur.e en informatique travaillant pour une entreprise de conception de drones autonomes. Votre équipe est en train de développer un logiciel de pilotage automatique pour les drones qui leur permettra de voler de manière autonome. Le logiciel utilise un automate fini pour identifier les obstacles et éviter les collisions en temps réel. Cependant, l'automate fini courant contient un grand nombre d'états, ce qui peut ralentir le système et réduire l'autonomie du drone. Minimisez l’automate ci-dessous afin d’améliorer la performance du logiciel. Donnez la table d’états-transition, précisez les états finaux et construisez l’automate minimal.

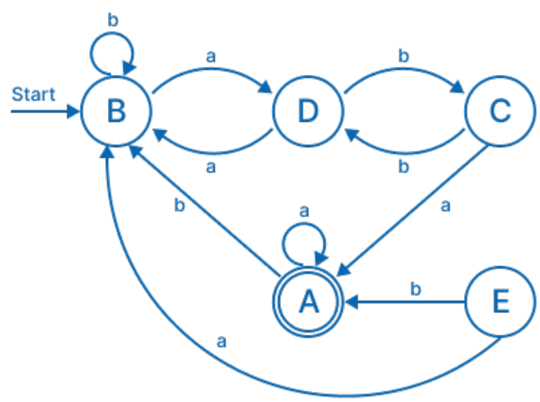


Réponse :

- Table d'états-transition

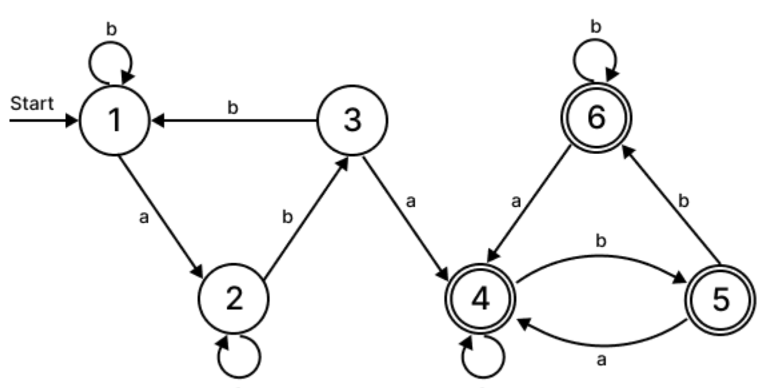
États	Entrée	
	a	**b**
A	A	B
B	D	B
C	A	D
D	B	C
E	B	A

- État final : *A*
- Automate



Exercice 3.

a) En utilisant le lemme d'Arden, trouvez le langage reconnu par la machine à états finis suivante. Présentez toutes les étapes de votre démarche.



Réponse :

Soient X_1, X_2, X_3, X_4, X_5 et X_6 les étiquettes associées aux états 1, 2, 3, 4, 5 et 6, respectivement.

Le système d'équations décrivant les états de l'automate est :

$$\begin{cases} X_1 = bX_1 + aX_2 \\ X_2 = aX_2 + bX_3 \\ X_3 = bX_1 + aX_4 \\ X_4 = aX_4 + bX_5 + \varepsilon \\ X_5 = aX_4 + bX_6 + \varepsilon \\ X_6 = bX_6 + aX_4 + \varepsilon \end{cases}$$

En appliquant le lemme d'Arden à X_6, on a :

$$\begin{aligned} X_6 \quad &= b^*(aX_4 + \varepsilon) \\ &= b^*aX_4 + b^* \end{aligned}$$

En substituant le résultat obtenu pour X_6 dans X_5, on a :

$$\begin{aligned} X_5 \quad &= aX_4 + b(b^*aX_4 + b^*) + \varepsilon \\ &= aX_4 + bb^*aX_4 + bb^* + \varepsilon \\ &= aX_4 + b^+aX_4 + b^+ + \varepsilon \end{aligned}$$

En substituant le résultat obtenu pour X_5 dans X_4, on obtient :

$$\begin{aligned} X_4 \quad &= aX_4 + b(aX_4 + b^+aX_4 + b^+ + \varepsilon) + \varepsilon \\ &= aX_4 + baX_4 + bb^+aX_4 + bb^+ + b + \varepsilon \\ &= (a + ba + bb^+a)X_4 + (bb^+ + b + \varepsilon) \\ &= (a + ba + bb^+a)^*(bb^+ + b + \varepsilon) \qquad \textit{lemme d'Arden} \end{aligned}$$

En appliquant le lemme d'Arden à X_2, on a : $\quad X_2 = a^*bX_3$

En substituant le résultat obtenu pour X_2 dans X_1, on obtient :

$$\begin{aligned} X_1 \quad &= bX_1 + a(a^*bX_3) \\ &= b^*aa^*bX_3 \qquad \textit{lemme d'Arden} \\ &= b^*a^+bX_3 \end{aligned}$$

En substituant maintenant les résultats obtenus pour X_4 et X_1 dans X_3, on obtient :

$$\begin{aligned} X_3 \quad &= b(b^*a^+bX_3) + a((a + ba + bb^+a)^*(bb^+ + b + \varepsilon)) \\ &= bb^*a^+bX_3 + a(a + ba + bb^+a)^*(bb^+ + b + \varepsilon) \\ &= b^+a^+bX_3 + a(a + ba + bb^+a)^*(bb^+ + b + \varepsilon) \\ &= (b^+a^+b)^*a(a + ba + bb^+a)^*(bb^+ + b + \varepsilon) \qquad \textit{lemme d'Arden} \end{aligned}$$

Ainsi, $X_1 = b^*a^+b(b^+a^+b)^*a(a + ba + bb^+a)^*(bb^+ + b + \varepsilon)$
Le langage reconnu par cette machine à état est donc $b^*a^+b(b^+a^+b)^*a(a + ba + bb^+a)^*(bb^+ + b + \varepsilon)$

b) Donnez la grammaire $\boldsymbol{G}$ qui génère le langage reconnu par l'automate en a). Vous devez préciser l'alphabet $\boldsymbol{V}$, l'ensemble des symboles terminaux $\boldsymbol{T}$, l'axiome $\boldsymbol{S}$ et l'ensemble des règles de production $\boldsymbol{P}$.

Réponse :

Soit les symboles non terminaux associés aux états comme suit :

- État 1 : Symbole non terminal $\boldsymbol{S}$, axiome de la grammaire
- État 2 : Symbole non terminal $\boldsymbol{A}$
- État 3 : Symbole non terminal $\boldsymbol{B}$
- État 4 : Symbole non terminal $\boldsymbol{C}$
- État 5 : Symbole non terminal $\boldsymbol{D}$
- État 6 : Symbole non terminal $\boldsymbol{E}$

Nous avons les ensembles suivants :

$\boldsymbol{V} = \{\mathbf{a}, \mathbf{b}, \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{E}\}$

$\boldsymbol{T} = \{\mathbf{a}, \mathbf{b}\}$

Les productions de $\boldsymbol{P}$ sont :

$\boldsymbol{S} \rightarrow \mathbf{a}\boldsymbol{A} \mid \mathbf{b}\boldsymbol{S}$

$\boldsymbol{A} \rightarrow \mathbf{a}\boldsymbol{A} \mid \mathbf{bB}$

$\boldsymbol{B} \rightarrow \mathbf{a}\boldsymbol{C} \mid \mathbf{b}\boldsymbol{S} \mid \mathbf{a}$

$\boldsymbol{C} \rightarrow \mathbf{a}\boldsymbol{C} \mid \mathbf{b}\boldsymbol{D} \mid \mathbf{a}$

$\boldsymbol{D} \rightarrow \mathbf{a}\boldsymbol{C} \mid \mathbf{b}\boldsymbol{E} \mid \mathbf{b}$

$\boldsymbol{E} \rightarrow \mathbf{a}\boldsymbol{C} \mid \mathbf{b}\boldsymbol{E} \mid \mathbf{a} \mid \mathbf{b}$

Exercice 4.

Soit la grammaire $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{T}, \boldsymbol{S}, \boldsymbol{P})$ où $\boldsymbol{V} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{E}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{H}\}$, $\boldsymbol{T} = \{\boldsymbol{a}, \boldsymbol{b}\}$. $\boldsymbol{S}$ est l'axiome et $\boldsymbol{P}$ l'ensemble des règles de production suivantes.

$\boldsymbol{S} \rightarrow \boldsymbol{aA} \mid \boldsymbol{bB} \mid \boldsymbol{\varepsilon}$

$\boldsymbol{A} \rightarrow \boldsymbol{aC} \mid \boldsymbol{bD} \mid \boldsymbol{a}$

$\boldsymbol{B} \rightarrow \boldsymbol{aD} \mid \boldsymbol{bE} \mid \boldsymbol{b}$

$\boldsymbol{C} \rightarrow \boldsymbol{aC} \mid \boldsymbol{bD} \mid \boldsymbol{\varepsilon}$

$\boldsymbol{D} \rightarrow \boldsymbol{aF} \mid \boldsymbol{bG}$

$\boldsymbol{E} \rightarrow \boldsymbol{aD} \mid \boldsymbol{bE} \mid \boldsymbol{\varepsilon}$

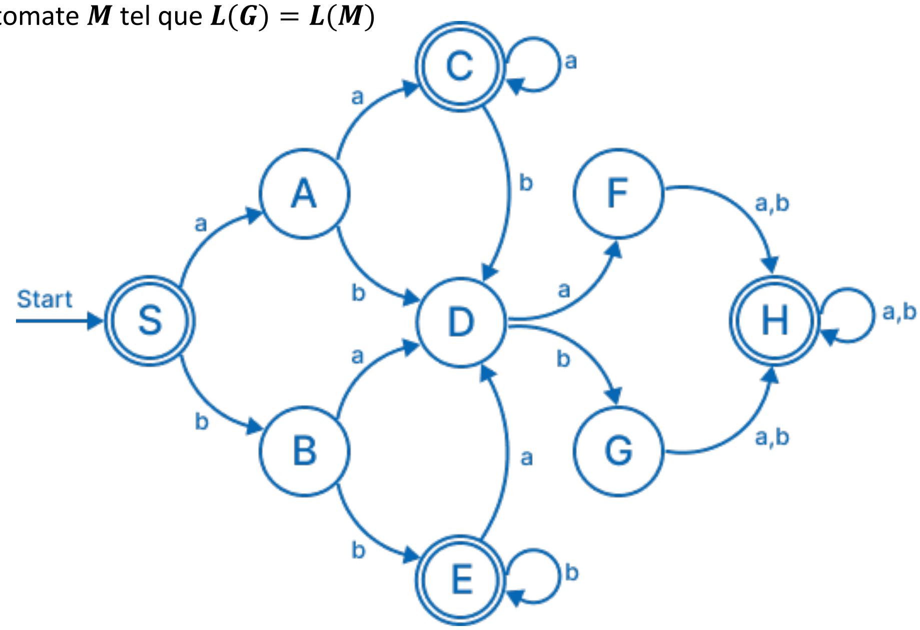
$\boldsymbol{F} \rightarrow \boldsymbol{aH} \mid \boldsymbol{bH} \mid \boldsymbol{a} \mid \boldsymbol{b}$

$\boldsymbol{G} \rightarrow \boldsymbol{aH} \mid \boldsymbol{bH} \mid \boldsymbol{a} \mid \boldsymbol{b}$

$\boldsymbol{H} \rightarrow \boldsymbol{aH} \mid \boldsymbol{bH} \mid \boldsymbol{\varepsilon}$

Construisez l'automate $\boldsymbol{M}$ tel que $\boldsymbol{L}(\boldsymbol{G}) = \boldsymbol{L}(\boldsymbol{M})$

Réponse :



Exercice 5.
Montrez que le langage $\boldsymbol{L}$ n'est pas régulier.

$$\boldsymbol{L} = \{0^{n!} | n \in \mathbb{N}\}$$

Réponse :

Raisonnons par l'absurde.
Supposons que le langage $\boldsymbol{L}$ est régulier. Il vérifie donc le lemme de pompage.
Soit p le seuil de pompage.
Le mot $w = 0^{p!}$ est un mot de $\boldsymbol{L}$ (sauf si $p < 3$, auquel cas nous choisirons $0^{3!}$).
Il existe une décomposition $w = xyz$ tel que $x = 0^q, y = 0^r$ et $z = 0^{p!-q-r}$
avec $q + r \leq p$ (car $|xy| \leq p$) et $r > 0$ (car $y \neq \varepsilon$).
D'après le lemme de pompage, $\forall i \geq 0, xy^iz \in \boldsymbol{L}$.
Ainsi, le mot xy^0z devrait être aussi un mot de $\boldsymbol{L}$.
Pour $i = 0$, on a :

$$\begin{aligned} xy^iz &= 0^q(0^r)^0 0^{p!-q-r} \\ &= 0^q 0^{p!-q-r} \\ &= 0^{p!-r} \end{aligned}$$

Pour que $0^{p!-r}$ soit un mot de $\boldsymbol{L}$, il doit y avoir un entier s tel que $s! = p! - r$.
Cependant, cela n'est pas possible puisque lorsque $p \geq 3$ et $r \leq p$, on a :

$$p! - p \leq p! - r \qquad (\mathbf{1})$$

Or, $p! - p = p \cdot (p-1)! - p = p\big((p-1)! - 1\big)$

Et $(p-1)! < p\big((p-1)! - 1\big)$ $, car\ p \geq 3$
Soit

$$(p-1)! < p! - p \qquad (\mathbf{2})$$

Par $(\mathbf{1})$ et $(\mathbf{2})$, on obtient :

$$(p-1)! < p! - p < p! - r$$

Soit

$$(p-1)! < p! - r \qquad (\mathbf{3})$$

Aussi, lorsque $p \geq 3$ et $r \leq p$, on a :

$$p! - r < p! \qquad (\mathbf{4})$$

Avec $(\mathbf{3})$ et $(\mathbf{4})$, on obtient :

$$(p-1)! < p! - r < p!$$

Ainsi, on en déduit que $p! - r$ ne peut être factoriel d'un entier.
Donc lorsque $i = 0$, $xy^iz \notin \boldsymbol{L}$. Le lemme de pompage n'est pas vérifié.
D'où le langage $\boldsymbol{L}$ n'est pas régulier.
CQFD

Exercice 6.
Soit M_T la machine de Turing dont l'état initial est S_0 et définie par les sept quintuples suivants :

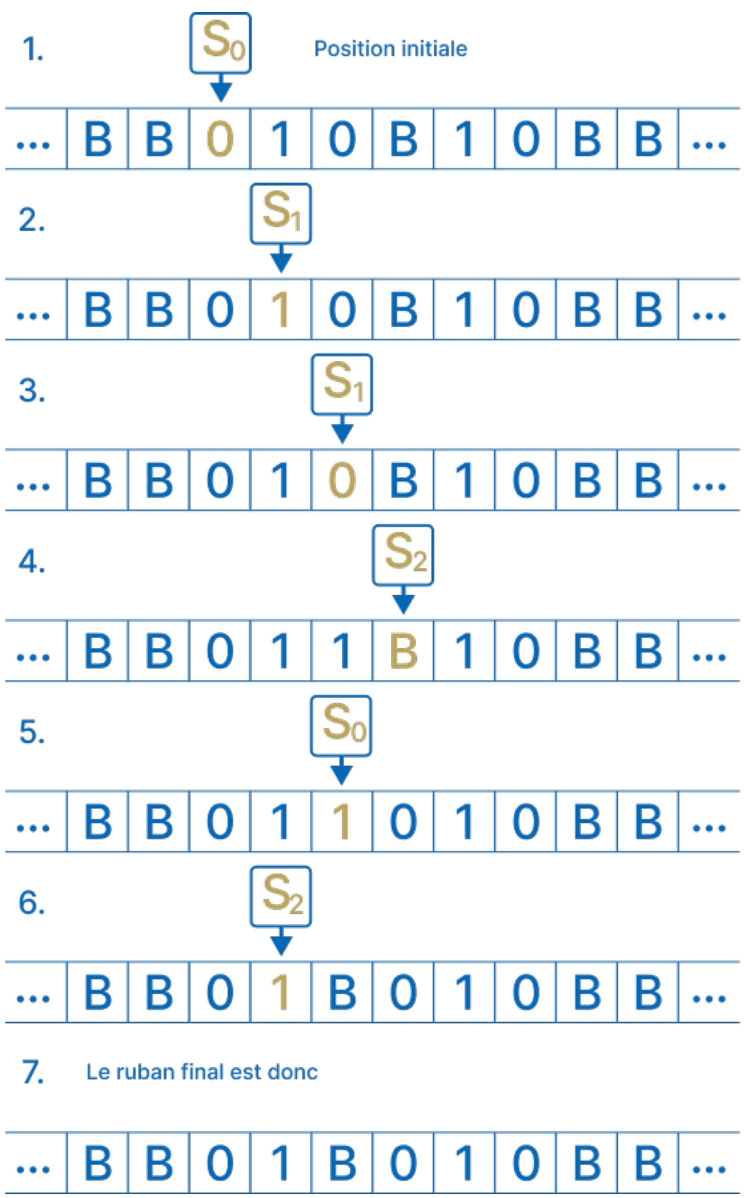
$$(S_0, 0, S_1, 0, R)\ ;\ (S_0, 1, S_2, B, L)\ ;\ (S_0, B, S_1, 1, R)\ ;\ (S_1, 0, S_2, 1, R)\ ;$$

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

En considérant le ruban initial suivant, déterminez le ruban final lorsque M_T s'arrête. On suppose que M_T commence en position initial.

…	B	B	0	1	0	B	1	0	B	B	…

Réponse :



$(S_0, 0, S_1, 0, R)$

$(S_1, 1, S_1, 1, R)$

$(S_1, 0, S_2, 1, R)$

$(S_2, B, S_0, 0, L)$

$(S_0, 1, S_2, B, L)$

Finalement, à la sixième étape, M_T s'arrête car il n'y a aucun quintuple commençant par la paire $(S_2, 1)$.

Exercice 7.
Construisez une machine de Turing qui reconnaît l'ensemble de toutes les chaînes de bits qui contiennent au moins deux '1'. Justifiez votre réponse.

Réponse :

Nous pouvons rester dans $\boldsymbol{S_0}$ jusqu'à ce que nous atteignions le premier '1' et puis rester dans l'état $\boldsymbol{S_1}$ jusqu'à ce que nous atteignions le deuxième '1'. À ce stade, nous pouvons entrer dans l'état $\boldsymbol{S_2}$ qui sera un état d'acceptation. Si nous arrivons au dernier blanc alors que nous sommes toujours dans les états $\boldsymbol{S_0}$ ou $\boldsymbol{S_1}$, nous n'accepterons pas cette chaîne de bits.

Ainsi, les quintuples sont donc :

$$(\boldsymbol{S_0}, \boldsymbol{0}, \boldsymbol{S_0}, \boldsymbol{0}, \boldsymbol{R})\ ;\ (\boldsymbol{S_0}, \boldsymbol{1}, \boldsymbol{S_1}, \boldsymbol{1}, \boldsymbol{R})\ ;\ (\boldsymbol{S_1}, \boldsymbol{0}, \boldsymbol{S_1}, \boldsymbol{0}, \boldsymbol{R})\ ;\ (\boldsymbol{S_1}, \boldsymbol{1}, \boldsymbol{S_2}, \boldsymbol{1}, \boldsymbol{R})\ ;$$

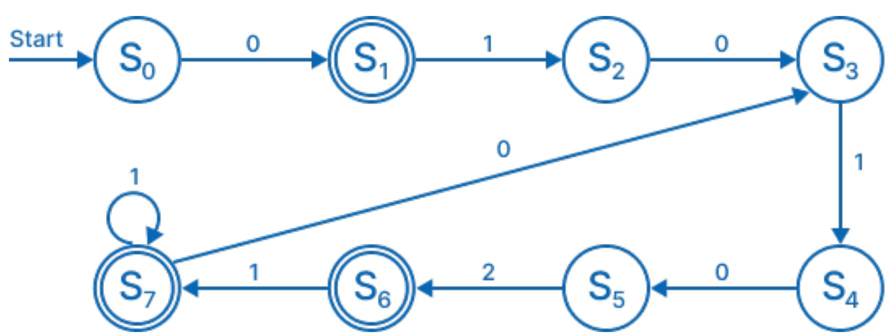
Exercice 8.
Construisez un automate fini reconnaissant l'expression suivante :

$$0(10102(1^*))^*$$

Réponse :

- $\boldsymbol{S_0}$ est l'état initial de l'automate.
- $\boldsymbol{S_1}, \boldsymbol{S_6}$ et $\boldsymbol{S_7}$ sont les états d'acceptation, finaux ou terminaux de l'automate.

- Automate



Exercice 9. (Facultatif)
Montrez que le langage $\boldsymbol{L}$ n'est pas régulier.

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

Réponse :

Raisonnons par l'absurde.
Supposons que le langage $\boldsymbol{L}$ est régulier. Il vérifie donc le lemme de pompage.
Soit p le seuil de pompage.
Le mot $w = 0^p 1^p 2^p$ est un mot de $\boldsymbol{L}$.
Il existe une décomposition $w = xyz$ tel que $x = 0^q, y = 0^r$ et $z = 0^{p-q-r} 1^p 2^p$ avec $q + r \leq p$ (car $|xy| \leq p$) et $r > 0$ (car $y \neq \varepsilon$).
D'après le lemme de pompage, $\forall i \geq 0, xy^i z \in \boldsymbol{L}$.
Ainsi, le mot $xy^0 z$ devrait être aussi un mot de $\boldsymbol{L}$.
Pour $i = 0$, on a :

$$\begin{aligned} xy^i z &= 0^q (0^r)^0 0^{p-q-r} 1^p 2^p \\ &= 0^q 0^{p-q-r} 1^p 2^p \\ &= 0^{p-r} 1^p 2^p \end{aligned}$$

Puisque $r > 0$, on a $p - r < p$.
Donc lorsque $i = 0$, $xy^i z \notin \boldsymbol{L}$. Le lemme de pompage n'est pas vérifié.
D'où le langage $\boldsymbol{L}$ n'est pas régulier.
CQFD