

## Devoir surveillé - Informatique Théorique 2

Documents non autorisés

La clarté et la lisibilité de vos réponses seront prises en compte dans l'évaluation.

### 1 Langages et automates

Pour chacun des langages suivants sur l'alphabet à trois lettres  $\{a, b, c\}$ , donner un automate fini déterministe qui le reconnaît :

1. le langage des mots de longueur au moins deux dont la deuxième lettre n'est pas  $c$ .
2. le langage des mots qui ne contiennent pas le facteur  $ab$ .
3. le langage des mots qui ont un nombre de  $a$  multiple de 3 et qui se terminent par  $c$ .
4. le langage des mots de longueur au moins deux dont la deuxième lettre n'est pas  $c$  et qui ne contiennent pas le facteur  $ab$ .

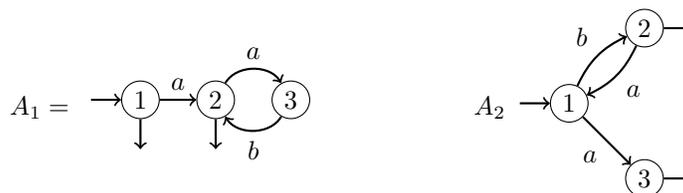
### 2 Détermination d'automates

Soit l'automate  $M = (Q, \Sigma, \delta, \{q_0\}, F)$  où  $Q = \{1, 2, 3\}$ ,  $\Sigma = \{a, b\}$ ,  $q_0 = 2$ ,  $F = \{1\}$  et  $\delta = \{(1, a, 1), (1, b, 2), (2, a, 1), (2, b, 2), (2, b, 3), (3, a, 2), (3, b, 3)\}$ .

1. Représenter l'automate sous la forme d'un graphe orienté étiqueté. Cet automate est-il complet ? Justifier la réponse.
2. Quel est le langage reconnu par cet automate ? Utiliser le lemme d'Arden en donnant les étapes de la résolution du système d'équations.
3. Déterminer cet automate. Quel est le langage reconnu par l'AFD obtenu ?

### 3 Construction d'automates par l'algorithme de Thompson

On considère les deux automates :



1. En utilisant l'algorithme de Thomson donner l'automate sans  $\epsilon$ -transition qui reconnaît le langage  $L(A_1) \bullet L(A_2)$ .
2. En utilisant l'algorithme de Thomson donner l'automate **avec**  $\epsilon$ -transitions qui reconnaît le langage  $(L(A_2))^*$ .

## 4 Autour de l'union et de l'intersection de langages

Soit  $B$  un langage. Soient  $A$  et  $L$  deux langages reconnaissables. Est-ce que les propriétés suivantes sont vraies? Prouver vos affirmations.

- Propriété 1 :  $A \cup B = L \implies B$  est reconnaissable
- Propriété 2 :  $A \cap B = L \implies B$  est reconnaissable
- Propriété 3 :  $A \cap L$  n'est pas reconnaissable  $\implies B$  est reconnaissable

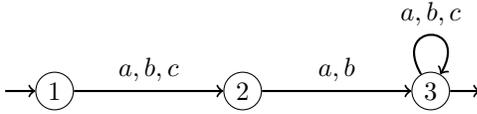
## 5 Automates complets

Soit un automate  $(Q, \Sigma, \delta, q_0, F)$  qui n'est pas complet. La fonction de transition est elle définie sur son domaine de définition tout entier?

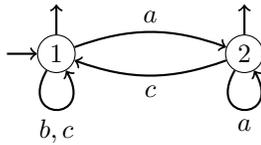
# Correction

## 1 Langages et automates

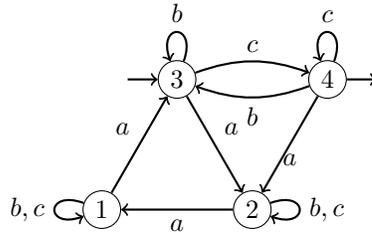
1.



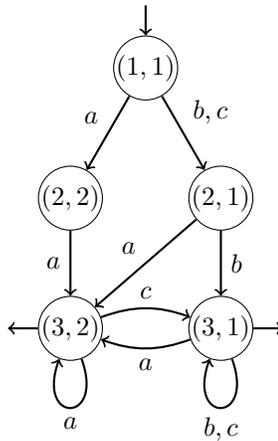
2.



3.

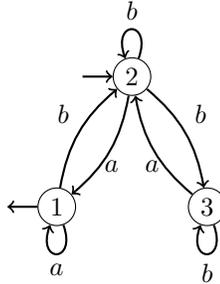


4. Le langage demandé est l'intersection du langage de la Question 1 et du langage de la Question 2. On construit donc l'automate produit des deux automates construits pour ces deux questions.



## 2 Déterminisation d'automates

1. L'automate donné est complet : pour tout état  $q \in Q$  et toute lettre  $\ell \in \Sigma$ , il existe  $r \in Q$  tel que  $(q, \ell, r) \in \delta$ . On représente cet automate ci-dessous,



2. À partir de l'automate, on obtient le système d'équations suivant :

$$\begin{cases} L_1 = a.L_1 + b.L_2 + \epsilon \\ L_2 = a.L_1 + b.L_2 + b.L_3 \\ L_3 = a.L_2 + b.L_3 \end{cases}$$

L'état initial étant l'état 2, on cherche une expression rationnelle pour le langage  $L_2$ . On résout le système de la façon suivante :

- On commence par appliquer Arden sur la première équation et on obtient :

$$L_1 = a^*(b.L_2 + \epsilon)$$

- Ensuite, on applique Arden sur la troisième équation ce qui donne :

$$L_3 = b^*a.L_2$$

- On substitue maintenant ces deux égalités dans la deuxième équation pour obtenir :

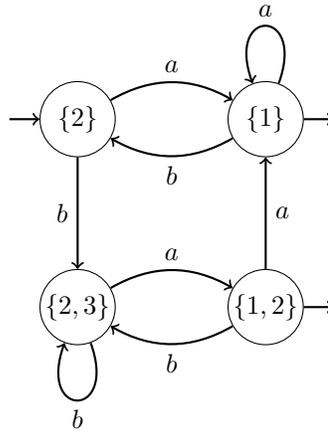
$$\{ L_2 = (aa^*b + b + bb^*a).L_2 + aa^*$$

- En utilisant le lemme d'Arden une dernière fois, on obtient :

$$L_2 = (aa^*b + b + bb^*a)^*aa^*$$

Langage de l'automate :  $L = L_2$ .

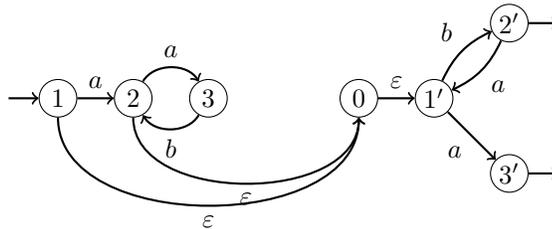
3. L'automate déterministe est le suivant :



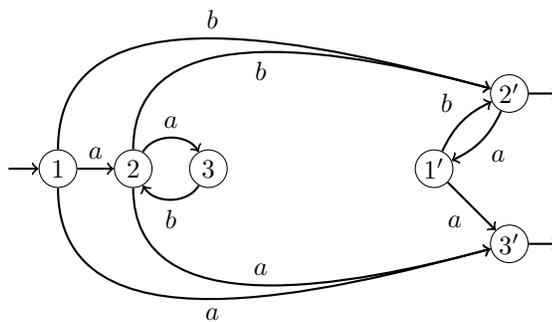
La procédure de déterminisation construit un automate qui reconnaît le même langage que celui reconnu par l'automate d'origine. Cet automate reconnaît donc le même langage que l'automate des questions précédentes.

### 3 Construction d'automates par l'algorithme de Thomson

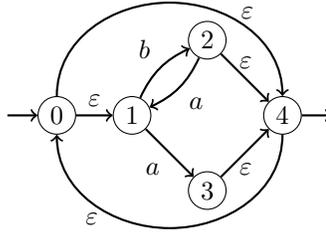
1. On commence par donner une version avec transition  $\varepsilon$ ,



On élimine ensuite les transitions  $\varepsilon$ ,



2. On donne un automate avec transitions  $\varepsilon$ ,



## 4 Autour de l'union et de l'intersection de langages

1. Faux.

Prenons  $A = L = (a + b)^*$  (qui est reconnaissable) et  $B = \{a^n b^n \mid n \geq 0\}$  (qui n'est pas reconnaissable). On a bien  $L = A \cup B = (a + b)^*$  et pourtant  $B$  n'est pas reconnaissable.

2. Faux.

Prenons  $A = L = \emptyset$  (qui est reconnaissable) et  $B = \{a^n b^n \mid n \geq 0\}$  (qui n'est pas reconnaissable). On a bien  $L = A \cap B = \emptyset$  et pourtant  $B$  n'est pas reconnaissable.

3. Vrai.

La table de vérité de l'implication est

$\Rightarrow$	V	F
V	V	F
F	V	V

Ainsi,  $L$  et  $A$  étant tous deux reconnaissables,  $A \cap L$  est reconnaissable et l'assertion " $A \cap L$  n'est pas reconnaissable" est fausse. Ainsi, d'après la table de vérité, la propriété 3 est vraie.

En fait, cette propriété traduit seulement le fait suivant : si vous réussissez à fournir une preuve de l'absurdité suivante : " $A \cap L$  n'est pas reconnaissable", alors c'est que les mathématiques ne sont pas cohérentes (ou plus probablement que vous vous êtes trompé quelque part) et dans ce cas vous pouvez démontrer la véracité de n'importe quelle assertion (ne particulier celle affirmant que  $B$  est reconnaissable).

## 5 Automates complets

Si on définit  $\delta$  comme une fonction de  $\delta : Q \times \Sigma \rightarrow Q$ , alors  $\delta$  n'est pas définie sur son domaine de définition tout entier lorsque l'automate n'est pas complet.

Autrement il existe un état  $q \in Q$  et une lettre  $a \in \Sigma$  tels que  $\delta(q, a)$  n'est pas défini.