

Rappel à propos des consignes et quelques conseils et remarques

- Durée : 2 heures.
- Aucune sortie avant 30 minutes.
- Aucune entrée après 30 minutes.
- 3 feuilles A4 R/V autorisées.
- Le soin de la copie sera pris en compte.
- Tout dispositif électronique est interdit (calculatrice, téléphone, tablette, montre connectée, etc.).
- Le barème est donné à titre indicatif.
- L'examen est sur 23 points, vous devez obtenir 20 points pour obtenir la note maximale.

Questions de cours

Pour les exercices de cette section (Exercice 1 à Exercice 3), il faut 1) indiquer la ou les propositions correctes et 2) fournir une brève justification (4-5 lignes).

Solution de l'exercice ??

(0,75 pt) pour la bonne réponse

(0,75 pt) pour la justification

Lorsque l'exécution d'un automate est définie sur un mot w de longueur n , l'exécution est une séquence de configurations de longueur $n + 1$, commençant avec avec la configuration (q_0, w) et terminant dans une configuration de la forme (q, ϵ) avec $q \in Q$. Ainsi, seule la deuxième proposition est correcte.

Solution de l'exercice ??

(0,4 pt) par question

1. Vrai. Tous les mots sont acceptés et l'automate est déterministe, il ne peut donc y avoir aucun état non-accepteur (sauf si ces états sont non-accessibles).
2. Faux. Rien ne permet d'affirmer qu'il existe un état qui n'est pas final/accepteur.
3. Vrai. Voir question 1.
4. Vrai. Σ étant l'alphabet de l'automate, il est non vide.
5. Faux. Rien ne permet d'affirmer cela. Par exemple, l'automate ci-dessous accepte le langage universel sur l'alphabet $\{a, b\}$.



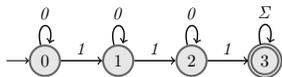
Solution de l'exercice ??

Les deux automates étant déterministes et complets, leur produit est déterministe et complet. Aussi le cardinal de l'ensemble des états accepteurs de l'automate produit est le produit des cardinaux des ensembles des états accepteurs des automates de départ : 6. Ces états ne sont pas nécessairement accessibles.

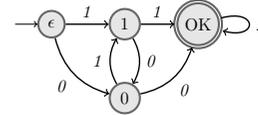
Automates déterministes

Solution de l'exercice ??

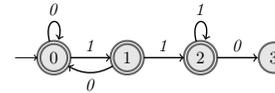
1. (1 pt) Un automate répondant à la question est donné ci-dessous :



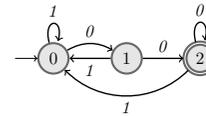
2. (1 pt) Un automate répondant à la question est donné ci-dessous :



3. (1 pt) Un automate répondant à la question est donné ci-dessous :



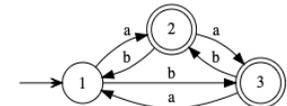
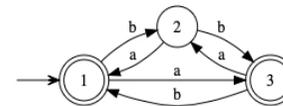
4. (1 pt) Un automate répondant à la question est donné ci-dessous :



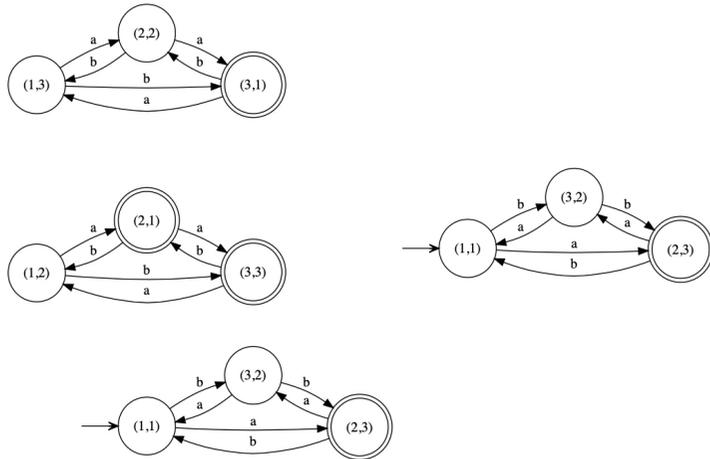
Produit d'automates

Solution de l'exercice ??

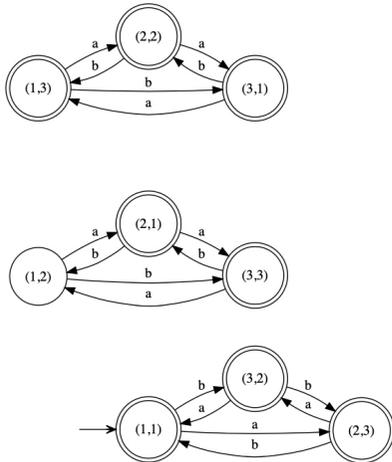
Nous commençons par représenter graphiquement les automates.



1. (2 pt) Nous utilisons le produit entre automate tel que défini en cours. Nous représentons l'automate ci-dessous (gauche) en appliquant la définition et en construisant l'automate à la volée (droite).



2. (2 pt) Nous adaptons la construction de l'automate produit. Un état est accepteur dans l'automate produit si cet état contient un des deux états formant le couple est un état accepteur de son automate. Nous obtenons l'automate ci-dessous qui reconnaît le langage universel.



Cet automate n'est pas minimal.

Algorithmes et preuves

Solution de l'exercice ??

- (2 pt) Soit Σ un alphabet. Considérons un langage $L \subseteq \Sigma^*$. La preuve se fait en deux temps, en montrant chaque implication.
 - Supposons que L est égal à $L' \cup L'^R$, pour un certain langage $L' \subseteq \Sigma^*$. Nous avons $L^R = (L' \cup L'^R)^R = L'^R \cup L'$, par définition du langage miroir (qui est l'application de l'opérateur miroir à chaque mot du langage). Ainsi, $L^R = L' \cup L'^R$ car $(w^R)^R = w$ pour tout mot w (et donc $(L^R)^R = L$ pour tout langage L).
 - Supposons que $L = L^R$. Alors, nous avons $L \cup L^R = L$ d'après la définition de l'union ensembliste.
- (2 pt) Comme l'automate reconnaît un langage fini, d'une part il existe une longueur maximale pour les mots acceptés et d'autre part il n'existe pas de cycle. Cette longueur maximale est le nombre d'états de l'automate (qui correspond à la longueur maximale des exécutions sans cycle moins 1). Les algorithmes répondant à la question sont donnés dans la Figure 1.

Algorithmes pour la question 2

lister_mot_longueur

Entrée : Σ, n (* un alphabet et un entier *)
 Sortie : M (* ensemble des mots de longueur inférieure ou égale à n *)

- ensemble de mots $M := \{\epsilon\}$
- ensemble de mots $M_\Sigma := \{m \mid |m| = 1 \text{ et } m(1) \in \Sigma\}$
 (* M_Σ est ici un ensemble de mots de longueur 1 construit avec les symboles de l'alphabet *)
- ensemble de mots $M_{\text{temp}} := \{\epsilon\}$
- pour $1 \leq i \leq n$ faire (* Répéter n fois. *)
- $M_{\text{temp}} := M_{\text{temp}} \cdot M_\Sigma$ (* Concaténation de langages *)
- $M := M \cup M_{\text{temp}}$
- fin pour
- retourner M

mots_acceptes

Entrée : $A = (Q, \Sigma, q_{\text{init}}, \delta, F)$ (* un AFED ne comportant pas de cycle *)
 Sortie : L (* ensemble des mots acceptés par A *)

- entier $n = |Q|$ (* es mots acceptés sont de longueur maximale $|Q|$ *)
- pour chaque mot m de lister_mot_longueur(Σ, n) faire
- si m est accepté par A alors
- $L := L \cup \{m\}$
- fin si
- fin pour
- retourner L

FIGURE 1: Algorithmes pour la question 2 - calculer l'ensemble des mots d'une certaine longueur sur un alphabet.

- (2 pt) L'automate ne comportant pas de cycle, le langage reconnu par cet automate est fini. Nous pouvons donc réutiliser les algorithmes de la question précédente. Les algorithmes répondant à la question sont donnés dans la Figure 2.

Minimisation

Solution de l'exercice 7

Algorithmes pour la question 3

miroir

Entrée : m (* un mot *)
 Sortie : w^R (* le mot miroir de m *)
 1: **mot** $mi := \epsilon$ (* le mot miroir *)
 2: **entier** $l := |m|$ (* $l = |\text{dom}(m)|$, la longueur de m *)
 3: **pour** $1 \leq i \leq l$ **faire**
 4: $mi := mi \cdot m(l - i + 1)$
 5: **fin pour**
 6: **retourner** mi

Algorithme répondant à la question 3

Entrée : $A = (Q, \Sigma, q_{\text{init}}, \delta, F)$ (* un AFED reconnaissant un langage fini *)
 Sortie : vrai ssi le langage reconnu par l'algorithme satisfait la condition indiquée dans la question
 1: **ensemble de mots** $L := \text{mots_acceptes}(A)$
 2: **ensemble de mots** $L_{\text{miroir}} := \{\text{miroir}(m) \mid m \in L\}$ (* $\text{miroir}(w) = w^R$ *)
 3: **retourner** $L = L_{\text{miroir}}$

FIGURE 2: Algorithmes pour la question 3.

Nous observons que l'état 3 n'est pas accessible. Nous le supprimons pour pouvoir appliquer les algorithmes de cet exercice. L'automate en représentation graphique est donné dans la Figure 3.

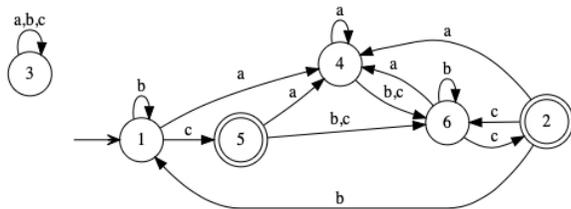


FIGURE 3: Automate initial de l'exercice 6 en représentation graphique.

- (1 pt) Les états sont tous distinguables deux à deux à l'exception des états (2 et 5) et des états (1 et 6) entre eux. Nous représentons le résultat de l'exécution de l'algorithme dans la Figure 4.
- (1 pt) Les états équivalents sont les états non-distinguables. Ainsi, les états équivalents sont les états 4 et 5.
- (1 pt) Oui, voir question précédente.
- (1 pt) Nous représentons l'automate minimal dans la Figure 5.

2	x				
3	x	x			
4	x	x	x		
5	x		x	x	
6		x	x	x	x
	1	2	3	4	5

FIGURE 4: Représentation de l'algorithme de calcul des états distinguables.

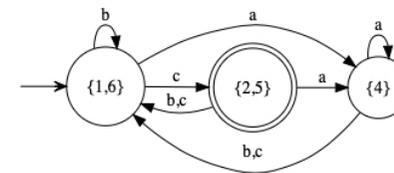


FIGURE 5: Représentation de l'automate minimisé.