Licence Sciences et Technologies
Univ. Grenoble Alpes

Examen à mi-parcours
INF 302 : Langages et Automates
L2, 2020/2021

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).

Exercice 1 (Exécution d'un automate déterministe - 1,5 points)

Considérons un automate déterministe $A = (Q, \Sigma, q_0, \delta, F)$ qui accepte les mots $w \in \Sigma^*$, avec $|w| = n$. Une exécution de A sur w :

1. est une séquence de longueur exactement n
2. est une séquence de longueur exactement $n + 1$
3. est une séquence de longueur arbitraire
4. est une séquence de longueur strictement supérieure à $n + 1$

Solution de l'exercice 1

(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.

Exercice 2 (Automate reconnaissant le langage universel - 2 points)

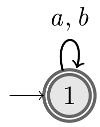
Considérons un automate déterministe $A = (Q, \Sigma, q_0, \delta, F)$ qui accepte le langage universel sur son alphabet.

1. Chaque état de A est final/accepteur.
2. Il y a au moins un état de A qui n'est pas final/accepteur.
3. Chaque état accessible est final/accepteur.
4. Il y a au moins un élément dans Σ.
5. Il y a au moins un état final/accepteur qui n'est pas q_0.

Solution de l'exercice 2

(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 $\{a, b\}$.



Exercice 3 (Automate produit - 1,5 points)

Considérons 2 automates déterministes accessibles et complets ayant respectivement 5 et 4 états, et 2 et 3 états finaux, et reconnaissant respectivement les langages L_1 et L_2. Indiquer le nombre d'états finaux dans l'automate produit reconnaissant l'intersection de L_1 et L_2.

Solution de l'exercice 3

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

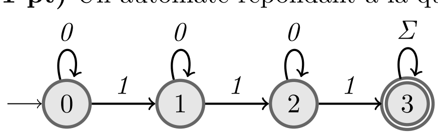
Exercice 4 (Dessiner un automate - 4 points)

Pour chacun des langages suivants, dessiner un automate déterministe reconnaisseur. Pour les trois premières questions, l'automate doit avoir au plus 4 états.

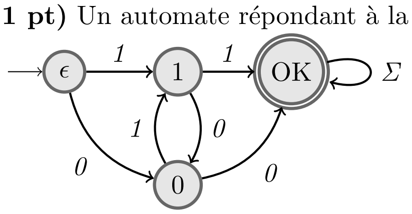
1. **(1 pt)** $\{w \in \{0,1\}^* \mid w$ contient au moins trois occurrences du symbole $1\}$.
2. **(1 pt)** $\{w \in \{0,1\}^* \mid w$ contient au moins une occurence du facteur 00 ou du facteur $11\}$.
3. **(1 pt)** $\{w \in \{0,1\}^* \mid w$ ne contient pas le facteur $110\}$.
4. **(1 pt)** $\{w \in \{0,1\}^* \mid w$ finit par $00\}$.

Solution de l'exercice 4

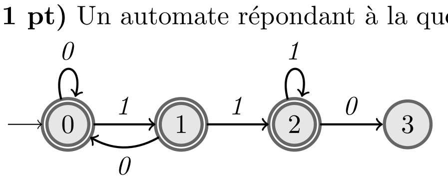
1. **(1 pt)** Un automate répondant à la question est donné ci-dessous :



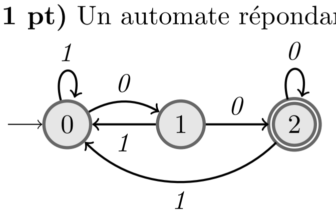
2. **(1 pt)** Un automate répondant à la question est donné ci-dessous :



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

Exercice 5 (Dessiner un automate - 4 points)

Considérons les 2 automates, A et B, dont les représentations tabulaires sont données ci-dessous.

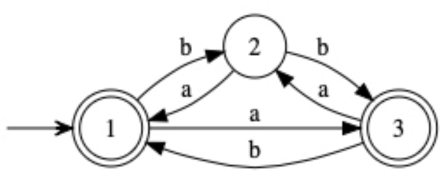
A	↓1	2*	3*
a	2	3	1
b	3	1	2

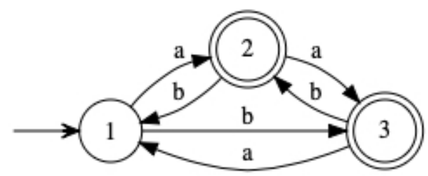
B	↓1*	2	3*
a	3	1	2
b	2	3	1

1. **(2 pt)** Dessiner l'automate reconnaissant l'*intersection* des langages reconnus par A et B.
2. **(2 pt)** Dessiner l'automate reconnaissant l'*union* des langages reconnus par A et B.

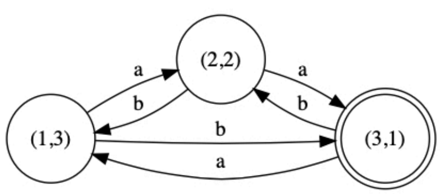
Solution de l'exercice 5

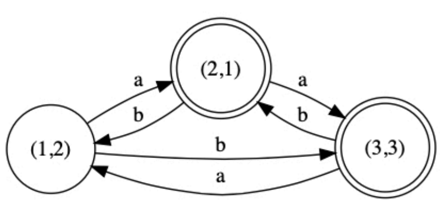
Nous commençons par représenter graphiquement les automates.

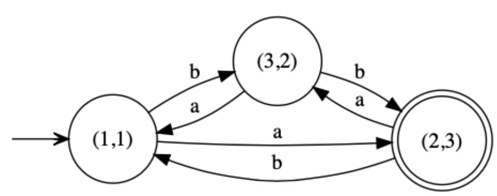


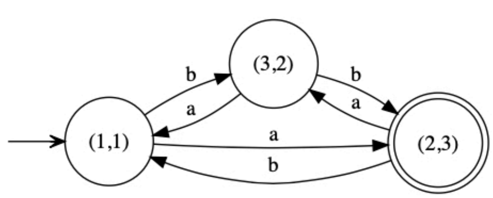


1. **(2 pt)** Nous utilisons le produit entre automate tel que défini en cours. Nous représentons l'automate ci-dessous (gauche) en appliquant la définition et en construisant l'automate à la volée (droite).

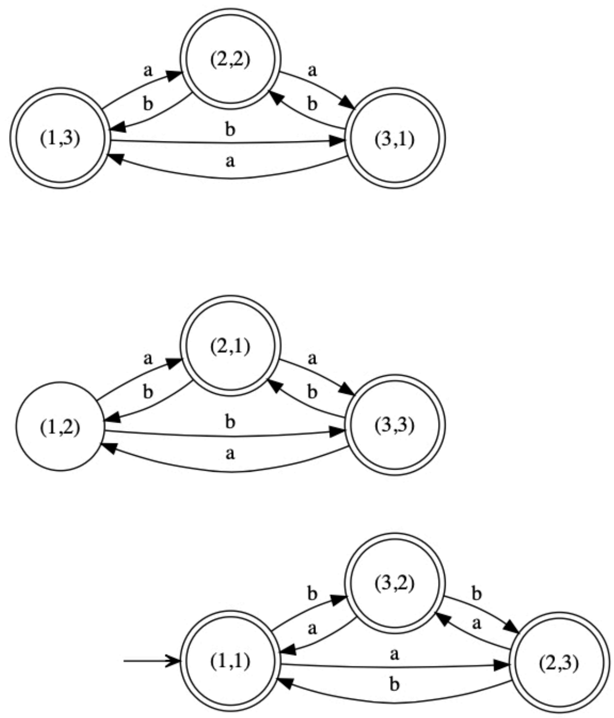








2. **(2 pt)** Nous adaptons la construction de l'automate produit. Un état est accepteur dans l'automate produit si cet état contient un des deux états formant le couple est un état accepteur de son automate. Nous obtenons l'automate ci-dessous qui reconnait le langage universel.



Cet automate n'est pas minimal.

Algorithmes et preuves

Exercice 6 (Miroir miroir - 6 points)

Pour rappel, le miroir d'un langage est le langage des (mots) miroir de ce langage. Le miroir d'un mot est le mot lu de droite à gauche. Plus formellement, étant donné $w \in \Sigma^*$, avec $w = a_1 \cdots a_n$, le mot miroir de w est noté w^R et est défini par $w^R = a_n \cdots a_1$. Le miroir d'un langage est le langage noté L^R et défini par $L^R = \{w^R \mid w \in L\}$.

1. **(2 pt)** Soit L un langage. Montrer que L est de la forme $L' \cup L'^R$ pour un certain langage L' si et seulement si $L = L^R$.
2. **(2 pt)** Donner un algorithme qui liste tous les mots d'un automate reconnaissant un langage fini.
3. **(2 pt)** Donner un algorithme qui, étant donné un automate déterministe ne comportant pas de cycle, détermine s'il existe un langage L (fini) tel que le langage reconnu par cet automate soit $L \cup L^R$.

Solution de l'exercice 6

1. **(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 $L^R = (L' \cup L'^R)^R = L'^R \cup L'^{R^R}$, par définition du langage miroir (qui est l'application de l'opérateur miroir à chaque mot du langage). Ainsi, $L^R = L'^R \cup L$ 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. **(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 $M_{\mathrm{tmp}} := \{\epsilon\}$
pour $1 \leq i \leq n$ **faire** (* Répéter n fois. *)
  $M_{\mathrm{tmp}} := M_{\mathrm{tmp}} \cdot M_\Sigma$ (* Concaténation de langages *)
  $M := M \cup M_{\mathrm{tmp}}$
fin pour
retourner M

mots_acceptes

Entrée : $A = (Q, \Sigma, q_{\mathrm{init}}, \delta, F)$ (* 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.

3. **(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

Exercice 7 (Minimisation - 4 points)

Considérons l'automate A dont la représentation tabulaire est données ci-dessous.

A	1	2*	3	4	5*	6
a	4	4	3	4	4	4
b	1	1	3	6	6	6
c	5	6	3	6	6	2

Algorithmes pour la question 3

miroir

Entrée : m (* un mot *)

Sortie : w^R (* le mot miroir de m *)

mot $mi := \epsilon$ (* le mot miroir *)
entier $l := |m|$ (* $l = |\mathrm{dom}(m)|$, la longueur de m *)
pour $1 \leq i \leq l$ **faire**
    $mi := mi \cdot m(l - i + 1)$
fin pour
retourner mi

Algorithme répondant à la question 3

Entrée : $A = (Q, \Sigma, q_{\mathrm{init}}, \delta, F)$ (* un AFED reconnaissant un langage fini *)

Sortie : vrai ssi le langage reconnu par l'algorithme satisfait la condition indiquée dans la question

ensemble de mots $L := \mathrm{mots\_acceptes}(A)$
ensemble de mots $L_{\mathrm{miroir}} := \{\mathrm{miroir}(m) \mid m \in L\}$ (* $\mathrm{miroir}(w) = m^R$ *)
retourner $L = L_{\mathrm{miroir}}$

FIGURE 2: Algorithmes pour la question 3.

1. **(1 pt)** Calculer l'ensemble des états distinguables de l'automate A.
2. **(1 pt)** Calculer l'ensemble des états équivalents de A.
3. **(1 pt)** Est-ce que les états équivalents sont non distinguables ? Justifier.
4. **(1 pt)** Minimiser l'automate A.

Solution de l'exercice 7

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.

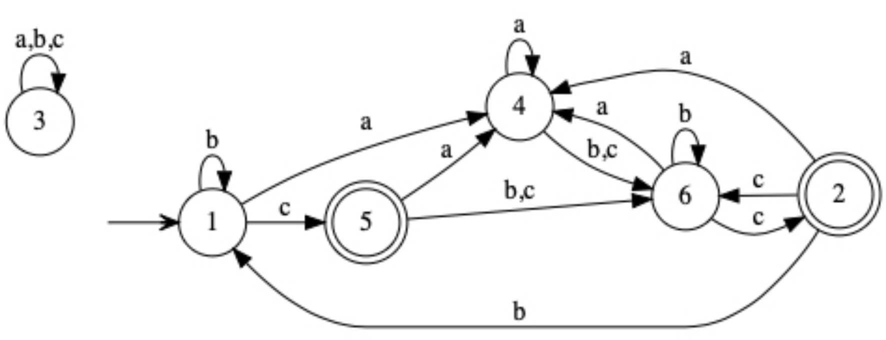


FIGURE 3: Automate initial de l'exercice 6 en représentation graphique.

1. **(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.
2. **(1 pt)** Les états équivalents sont les états non-distinguables. Ainsi, les états équivalents sont les

2	×				
3	×	×			
4	×	×	×		
5	×		×	×	
6		×	×	×	×
	1	2	3	4	5

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

états 4 et 5.

3. **(1 pt)** Oui, voir question précédente.
4. **(1 pt)** Nous représentons l'automate minimal dans la Figure 5.

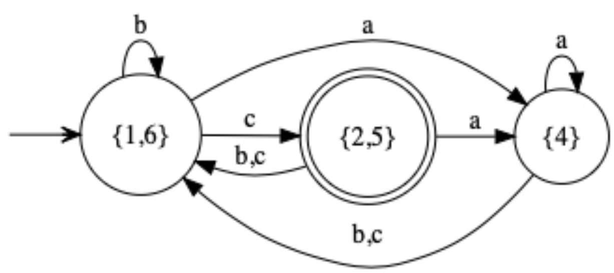


FIGURE 5: Représentation de l'automate minimisé.