
RICM3 – Automates et Grammaires

Durée : 2h00, sans documents.

- Tous les appareils électroniques sont interdits à l'exception des montres mécaniques.
- Le barème est donné à titre indicatif.
- Le sujet est sur 21 points et comporte 5 exercices indépendants.

5.5 pt

Exercice 1 : Comparaison d'automates (30 min)

On considère l'alphabet $\Sigma = \{a, b\}$. On considère le langage L formé des mots contenant aa et terminé par b .

0.25 pt

Q1. Donnez deux mots qui appartiennent à L et deux mots n'appartenant pas à L .

0.75 pt

Q2. Donnez un automate A *non-déterministe* qui reconnaît L sous forme de dessin et sous forme de tableau.

Indication : Vous pouvez continuer l'exercice même si vous n'êtes pas sûr de votre automate A . Ce ne sont moins les automates que les raisonnements qui m'intéressent.

1 pt

Q3. Déterminez l'automate A . Donnez l'automate déterministe sous forme de tableau uniquement.

Différences de deux automates Un élève prétend que l'automate déterministe E ci-dessous reconnaît le même langage que l'automate A . On cherche à savoir s'il a raison.

E	i	1	2	3	4^a
a	2	3	2	3	
b	1	1	4		

0.5 pt

Q4. Construisez l'automate complémentaire de E qu'on notera E^c sous forme de tableau.

0.25 pt

Q5. Donnez le langage reconnu par l'automate $A \times E^c$ en fonction de $\mathcal{L}(A)$ et $\mathcal{L}(E)$.

0.25 pt

Q6. On prétend que l'automate $A \times E^c$ permet d'estimer si l'élève s'est trompé. Expliquez pourquoi en donnant l'interprétation en français de ce que représente le langage $\mathcal{L}(A \times E^c)$

0.75 pt

Q7. Donnez un algorithme qui permet de savoir si un automate ne reconnaît aucun mot.

1.25 pt

Q8. Construisez l'automate $A \times E^c$ jusqu'au moment où vous pouvez conclure si l'élève s'est trompé ou pas. **On demande bien d'utiliser A et pas sa version déterministe.**

0.5 pt

Q9. Donnez une exécution de $A \times E^c$ qui accepte un mot et conclure.



5 pt

Exercice 2 : Grammaire des adresses web (30 min)

Les adresses web (ou URL de l'anglais *Uniform Resource Locator*) sont formées de "http" ou "https" pour les sites sécurisés, suivi de "://" puis d'un nom de site. Dans cet exercice les noms de site sont formés d'identificateurs séparés par des tirets '-' ou des soulignés '_'. Un nom de site doit comporter au moins un identificateur et se terminer par un nom de domaine précédé du caractère '.'.

Exemples :

- http://mon-site-web-perso.fr
- https://ma_banque_en_ligne.com

Les identificateurs sont des mots formés uniquement de lettres de l'alphabet (majuscule ou minuscule). Un identificateur doit comporter au moins une lettre.



0.25 pt

Q10. Donnez un automate qui reconnaît les identificateurs.

Indication : On dispose désormais d'un parser d'identificateur IDENT : $\text{stream char} \rightarrow \text{string}$ qui prend en entrée un flot de caractère et fait tourner l'automate sur ce flot. Si l'automate reconnaît un identificateur en début de flot le parser IDENT retourne l'identificateur reconnu sous forme de chaîne de caractère ; sinon le parser échoue et ne consomme pas le flot de caractère.

Vous pouvez donc utiliser le non-terminal IDENT dans vos grammaires.



0.25 pt

Q11. Donnez quatre noms de sites incorrects pour des raisons différentes



1 pt

Q12. Donnez une grammaire avec pour germe le non-terminal URL qui reconnaît les adresses web.



1 pt

Q13. Donnez l'arbre de dérivation qui permet à votre grammaire de générer «http://charlie-hebdo.fr»



1.25 pt

Q14. Modifiez la grammaire initiale pour lui ajouter un/des attributs afin qu'elle compte le nombre d'identificateurs dans le nom du site, sans compter celui du domaine. Le non-terminal URL est désormais un parser de type $\text{stream char} \rightarrow \mathbb{N}$

Exemples :

- URL [`< http://monsiteweb.com >`] = 1
- URL [`< http://polytech-grenoble.fr >`] = 2



1.25 pt

Q15. Modifiez la grammaire initiale pour lui ajouter un/des attributs afin qu'elle retourne une adresse web qui ne contiennent plus de séparateurs '-' ni '_'.

Le non-terminal URL est désormais un parser de type $\text{stream char} \rightarrow \text{string}$

Exemples :

- URL [`< http://mon_site_web.com >`] = "http://monsiteweb.com"
- URL [`< http://polytech-grenoble.fr >`] = "http://polytechgrenoble.fr"



3 pt

Exercice 3 : Algorithme (20 min)

On considère l'alphabet $\Sigma = \{a, b\}$.



0.5 pt

Q16. Dessinez un automate *A déterministe et complet* qui reconnaît le langage $\{a, b\}$.



0.5 pt

Q17. Dessinez l'automate qui reconnaît $\overline{\mathcal{L}(A)}$.



0.5 pt

Q18. Le langage $\overline{\mathcal{L}(A)}$ est-il fini? Justifiez votre réponse.

1.5 pt

Q19. Décrivez (en français) un algorithme qui permet de tester si le langage reconnu par un automate A est fini.

3.5 pt

Exercice 4 : Questions de cours (20 min)

0.5 pt

Q20. Décrivez par une phrase en français le langage correspondant à l'expression régulière suivante « $c \cdot (a^* | b^*)$ ». Est-il fini ou infini ?

0.5 pt

Q21. Étant donné un alphabet $\Sigma = \{a, b, c\}$. Peut-on construire un mot qui n'appartient pas à Σ^* ? SI OUI *donnez deux exemples*. SI NON *expliquez pourquoi cela est impossible*.

0.5 pt

Q22. **Donnez le langage correspondant à (répondez sur votre copie)**

- (1) $\{a, b\} \cdot \{a, b\} =$
- (2) $\{a, b\} \cdot \{c, \epsilon\} =$
- (3) $\Sigma^* \cdot \{\} =$
- (4) $\{\}^* \cdot \Sigma =$
- (5) $\{a, \epsilon\}^* =$

0.5 pt

Q23. Étant donné un alphabet Σ . Peut-on construire un automate qui reconnaît le langage Σ^* ? SI OUI *donnez l'automate*. SI NON *expliquez pourquoi cela est impossible*.

0.75 pt

Q24. Étant donné un alphabet $\Sigma_1 \subseteq \Sigma$. Peut-on construire un automate qui reconnaît les mots qui possèdent *au moins un symbole ne faisant pas partie de Σ_1* ? SI OUI *expliquez le principe de construction de l'automate*. SI NON *expliquez pourquoi cela est impossible*.

0.75 pt

Q25. Étant donné l'alphabet $\Sigma = \{a\}$. Est-il possible de construire un automate (à nombre) d'états fini qui reconnait un langage fini contenant exactement n mots ? SI OUI *donnez un exemple d'automate (qui dépend de n)*. SI NON *expliquez pourquoi cela est impossible*.

4 pt

Exercice 5 : Minimisation (20 min)

On considère l'automate suivant, les états initiaux sont annotés par i et les états accepteurs par a :

A	1^a	2	3	4	5	$i6$	7	8^a	9
b	4	7	9	9	7	1	6	7	6
c	5	2	8	5	2	2	2	8	2
a	1	4	6	4	4	3	9	3	9

0.5 pt

Q26. Le langage reconnu par l'automate ci-dessus est-il vide ? **Justifiez votre réponse.**

1.5 pt

Q27. Minimisez l'automate en justifiant chaque étape de l'algorithme. Les $\frac{2}{3}$ des points sont attribués aux justifications.

1 pt

Q28. Dessinez l'automate minimisé.

1 pt

Q29. Si l'automate n'avait aucun état accepteur quel serait l'automate minisé ? L'algorithme de minimisation donnerait-il cet automate ? *Justifiez votre réponse.*