Licence Sciences et Technologies
Université Joseph Fourier

Examen à mi-parcours
INF 232 : Langages et Automates
L2, 2015/2016

Rappel des consignes et quelques conseils/remarques

— Durée : 2 heures.
— Aucune sortie avant 30 minutes. Aucune entrée après 30 minutes.
— Tout document du cours ou du TD est autorisé.
— Tout dispositif électronique est interdit (calculatrice, téléphone, tablette, etc.).
— Le soin de la copie sera pris en compte (**-1 point en cas de manque de soin**).
— Les exercices sont indépendants.
— Le barème est donné à titre indicatif. L'examen est sur 21 points.

Exercice 1 (Vrai ou Faux - 2 points)

Répondre par Vrai ou Faux aux questions suivantes. Justifier *soigneusement* vos réponses.

1. Étant donné un langage, on peut trouver un automate d'états finis déterministe qui reconnaît ce langage.
2. Si L est d'états finis, alors pour tout langage $L' \subseteq L$, L' est d'états finis.
3. Pour tout langage régulier, on peut trouver un automate d'états finis déterministe qui satisfait les deux conditions suivantes :
   — l'automate reconnaît ce langage, et
   — si on enlève n'importe quel état de l'automate, alors le nouvel automate reconnaît un langage différent.
4. Tout langage d'états finis peut être reconnu par un automate déterministe avec un seul état terminal.

Exercice 2 (Vrai ou Faux suite - 3 points)

Répondre par Vrai ou Faux aux questions suivantes. Justifier *soigneusement* vos réponses.
Soient Σ et Σ' deux alphabets, soit Q un ensemble d'états, et soit $\Delta \subseteq Q \times \Sigma \times Q$ une relation de transition. On considère les deux automates définis par les quintuplets suivants : $A = (Q, q_0, \Sigma, \Delta, F)$ et $A' = (Q, q_0, \Sigma', \Delta, F)$. On note $\mathcal{L}(A)$ et $\mathcal{L}(A')$ les langages reconnus par A et A', respectivement.

1. Si $\Sigma \subseteq \Sigma'$, alors $\mathcal{L}(A) \subseteq \mathcal{L}(A')$.
2. On a toujours $\mathcal{L}(A) = \mathcal{L}(A')$.
3. Si $\Sigma \subset \Sigma'$, alors $\mathcal{L}(A) \subset \mathcal{L}(A')$.

Exercice 3 (Un peu d'algo - 2 points)

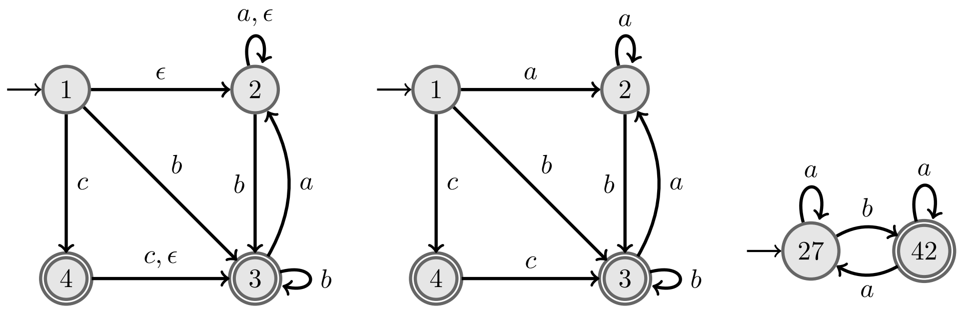
1. Donner un algorithme qui prend un ADEF sur un alphabet Σ en paramètre et détermine si le langage reconnu par cet automate est le langage universel sur Σ. Dans cette question, les algorithmes du cours peuvent être utilisés directement sans être redéfinis.

Exercice 4 (Expression régulière vers automate - 1.5 points)

On considère l'expression régulière suivante sur l'alphabet $\{a, b, c\}$:

$$((b \cdot c + \epsilon) \cdot (c \cdot a^* + a \cdot c) \cdot \epsilon)^*$$

1. Donner un automate non-déterministe avec ϵ-transitions qui reconnaît le langage dénoté par cette expression régulière. Vous n'êtes pas obligés de suivre la méthode compositionnelle donnée en cours. Vous pouvez peut-être simplifier l'expression (avec les justifications appropriées).



(a) Un automate à transformer

(b) Un automate pour la méthode associant des équations aux états

(c) Un automate pour la méthode associant des équations aux chemins

Exercice 5 (Transformations d'automates - 4,5 points)

On considère l'automate de la Figure 1a.

1. Éliminer les ϵ-transitions.
2. Déterminiser l'automate obtenu dans la question précédente.
3. Est-ce que cet automate est minimal ? Si l'automate n'est pas minimal, donner l'automate minimisé.

Exercice 6 (Automate vers expression régulière - 3 points)

On considère l'automate de la Figure 1b.

1. Donner une expression régulière associée à cet automate en utilisant **la méthode associant des équations aux *états***.

Exercice 7 (Automate vers expression régulière suite - 3 points)

On considère l'automate de la Figure 1c.

1. Donner une expression régulière associée à cet automate en utilisant la **méthode associant des équations aux *chemins***.

Exercice 8 (Un automate à trouver - 2 points)

On considère l'alphabet $\Sigma = \{a, b\}$. On s'intéresse au langage L des mots dont tous les préfixes ont un nombre de a qui n'excède pas le nombre de b de plus de 2 et dont le nombre de b n'excède pas le nombre de a de plus de 2. Autrement dit, $L = \{w \in \Sigma^* \mid \forall w' \in \Sigma^* : w' \preceq w \implies ||w'|_a - |w'|_b| \leq 2\}$ où $|w|_x$ dénote le nombre d'occurrences du symbole x dans le mot w. L est un langage d'état fini.

1. Donner un automate qui reconnaît L.