Licence Sciences et Technologies
Université Joseph Fourier

Examen final
INF 232: Langages et Automates
L2, 2012/2013

Rappel des consignes et quelques conseils/remarques

– Durée : 2 heures (13h00 → 15h00). 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. Justifier soigneusement vos réponses.
– Les exercices sont indépendants. Le barème est donné à titre indicatif.
– L'examen est sur 22 points. Il faut 20 points pour avoir la note maximale.

Exercice 1 (Vrai ou Faux - 3 points)

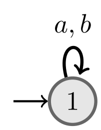
Répondre par Vrai ou Faux aux questions suivantes. Justifier *soigneusement* et sans preuve vos réponses.

1. Un automate complet reconnaît le langage universel.
2. Un automate complet déterministe reconnaît le langage universel.
3. Un automate complet déterministe dont tous les états sont accepteurs reconnaît le langage universel.
4. La différence de deux langages réguliers est un langage régulier.
5. Tous les langages réguliers satisfont le lemme de l'itération.
6. Il est possible que des langages non-réguliers satisfont le lemme de l'itération.
7. Dans la méthode de Floyd, la pré-condition doit être impliquée par la condition associée à chaque état initial.
8. Dans la méthode de Floyd, la post-condition doit être impliquée par au moins une des conditions des états terminaux.

Solution de l'exercice 1

Barème : 0,25 en moins pour chaque réponse mauvaise ou absente, 0,25 en moins pour chaque justification mauvaise ou absente.

1. Faux. Sur l'alphabet $\{a, b\}$, voici un contre-exemple :



2. Faux. L'automate de la question précédente peut servir de contre-exemple pour cette question.
3. Vrai. L'exécution de tout mot est définie et termine dans un état accepteur.
4. Vrai. Pour deux langages réguliers E, F quelcquonques, $E \setminus F = E \cap \overline{F}$. D'après la fermeture des langages réguliers par les opérations de complémentation et d'intersection, nous obtenons que $E \setminus F$ est régulier.
5. Vrai. Lemme de l'itération.
6. Vrai. Le lemme de l'itération est satisfait par tous les langages réguliers. Rien n'est indiqué pour les langages non-réguliers.
7. Faux. C'est la pré-condition qui doit impliquer tous les prédicats des états initiaux.
8. Faux. La post-condition doit être impliquée par tous les prédicats des états terminaux.

Exercice 2 (Expression régulière vers automate - 4 points)

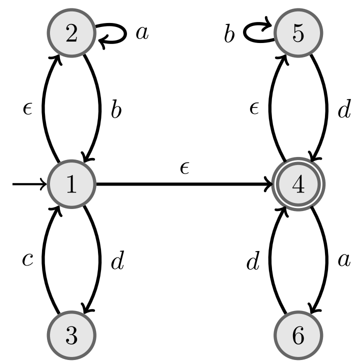
On considère l'expression régulière suivante sur l'alphabet $\{a, b, c, d\}$:

$$\left(a^* \cdot b + d \cdot c\right)^* \cdot \left(b^* \cdot d + a \cdot d\right)^*$$

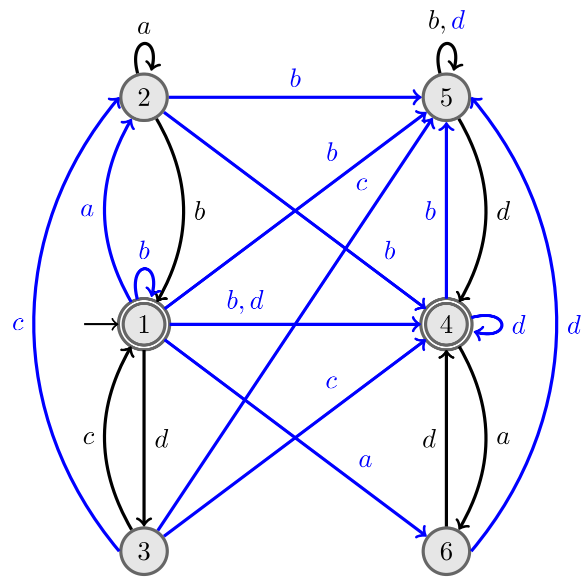
1. Donner un automate non-déterministe avec ϵ-transitions qui reconnaît le langage dénoté par cette expression régulière. Vous n'êtes pas obligés de suivre la méthode compositionnelle donnée en cours.
2. Éliminer les ϵ-transitions dans l'automate obtenu dans la question précédente.
3. Déterminiser l'automate obtenu dans la question précédente.
4. Est-ce que cet automate est minimal ? Si l'automate n'est pas minimal, donner l'automate minimisé.

Solution de l'exercice 2

1. **(1 point)**
   L'automate non-déterministe avec ϵ-transitions est :



2. **(1 point)**
   Après élimination des ϵ-transitions, nous obtenons :



3. **(1 point)**
   Appliquons l'algorithme de déterminisation, nous obtenons :

	$1*$	$2,6$	$1,4,5*$	2	$3,4*$	$1,2,4,5*$	$3,4,5*$	5	6	$4*$	$4,5*$
a	$2,6$	2	$2,6$	2	6	$2,6$	6			6	6
b	$1,4,5$	$1,4,5$	$1,4,5$	$1,4,5$	5	$1,4,5$	5	5		5	5
c					$1,2,4,5$		$1,2,4,5$				
d	$3,4$	$4,5$	$3,4,5$		4	$3,4,5$	$4,5$	$4,5$	$4,5$	4	$4,5$

   Les étoiles sont utilisées pour marquer les états accepteurs.

4. **(0,75 point pour l'algorithme de minimisation et 0,25 pour l'automate minimisé)**

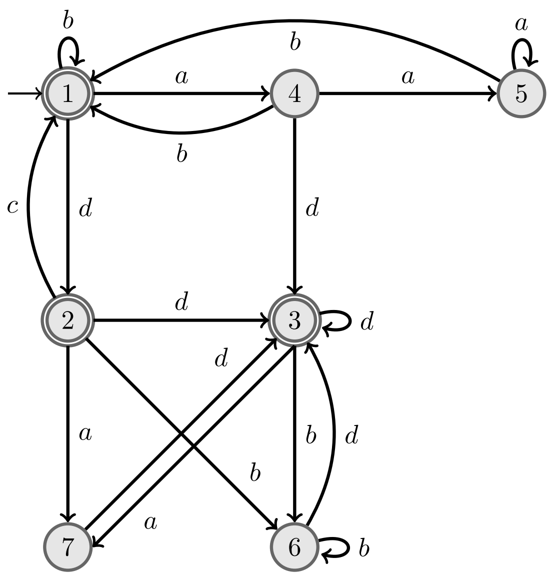
Avant d'appliquer l'algorithme de minimisation, renommons les états :

	$a*$	b	$c*$	d	$e*$	$f*$	$g*$	h	i	$j*$	$k*$
a	b	d	b	d	i	b	i			i	i
b	c	c	c	c	h	c	h	h		h	h
c					f		f				
d	e	k	g		j	g	k	k	k	j	k

L'algorithme de minimisation donne :

$\equiv_0$	$\equiv_1$	$\equiv_2$	$\equiv_3$
a	a	a	a
c	c	c	c
e	f	f	f
f	e	e	e
g	g	g	g
j	j	j	j
k	k	k	k
b	b	b	b
d	d	d	d
h	h	h	h
i	i	i	i
p	p	p	p

Lors du calcul de $\equiv_3$, nous obtenons $\equiv_2$. Donc l'algorithme s'arrete et l'automate n'est pas minimal. Nous prenons $acf \to 1, eg \to 2, jk \to 3, b \to 4, d \to 5, h \to 6, i \to 7$. L'automate minimisé est :



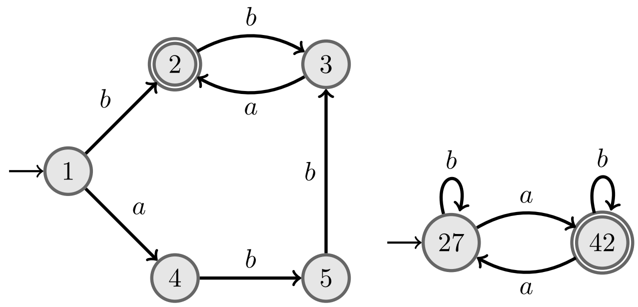
Exercice 3 (Automate vers expression régulière - 2 points)

Nous considérons l'automate de la Figure 1a.

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

Solution de l'exercice 3

1. **Barème :**



(a) Automate de l'exercice 3 (b) Automate de l'exercice 4

Figure 1: Automates pour les exercices 3 et 4

- **0,75 point pour le résultat,**
- **0,75 point pour la méthode,**
- **0,5 point pour les justifications (lemme d'Arden et $\epsilon \notin A$ pour une équation $X = AX + B$),**
- **0 à l'exercice si utilisation d'une autre méthode.**

Le système d'équations associé à cet automate est :

$$\begin{cases} X_1 = & bX_2 + aX_4 \\ X_2 = & bX_3 + \epsilon \\ X_3 = & aX_2 \\ X_4 = & bX_5 \\ X_5 = & bX_3 \end{cases}$$

Comme $X_3 = aX_2$, nous obtenons : $X_2 = baX_2 + \epsilon$.
En utilisant le lemme d'Arden, et comme $\epsilon \notin ba$, nous obtenons $X_2 = (ba)^*$.
En utilisant $X_5 = bX_3$ et $X_4 = bX_5$, nous obtenons $X_4 = bbX_3$. Puis $X_4 = bbaX_2 = bba(ba)^*$.
Au final :

$$X_1 = abba(ba)^* + b(ba)^*.$$

Exercice 4 (Automate vers expression régulière suite - 2 points)

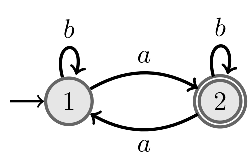
Nous considérons l'automate de la Figure 1b.

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

Solution de l'exercice 4

Barème :
- **0,25 point pour le renommage des états,**
- **0,25 point pour expressions de la forme générale de l'équation,**
- **0,25 point pour équations k=0,**
- **0,5 point pour équations k=1,2,**
- **0 à l'exercice si utilisation d'une autre méthode.**

1. D'abord, nous renommons les états de l'automate.



Nous suivons la méthode vue en cours qui consiste à calculer R_{ij}^k pour $k = 0, \ldots, 2$.
Les équations du cas de base sont :

$$\begin{array}{ll} R_{11}^0 = b + \epsilon & R_{21}^0 = a \\ R_{12}^0 = a & R_{22}^0 = b + \epsilon \end{array}$$

L'équation inductive est, pour $k \geq 1$:

$$R_{ij}^k = R_{ij}^{k-1} + R_{ik}^{k-1} \cdot (R_{kk}^{k-1})^* \cdot R_{kj}^{k-1}$$

Nous faisons donc le calcul en partant de $k = 1$ jusqu'à $k = 2$ (car le plus grand état est numéroté 2).

$$\begin{array}{ll} R_{11}^1 = (b+\epsilon) + (b+\epsilon)(b+\epsilon)^*(b+\epsilon) = b^* & R_{21}^1 = a + a(b+\epsilon)^*(b+\epsilon) = ab^* \\ R_{12}^1 = a + (b+\epsilon)(b+\epsilon)^* a = b^* a & R_{22}^1 = b + \epsilon + a(b+\epsilon)^* a = \epsilon + b + ab^* a \end{array}$$

Pour $k = 2$, nous sommes intéressés par R_{12}^2 :

$$\begin{aligned} R_{12}^2 &= b^*a + b^*a(\epsilon + b + ab^*a)^*(\epsilon + b + ab^*a) \\ &= b^*a(b + ab^*a)^*(\epsilon + b + ab^*a) \\ &= b^*a(b + ab^*a)^* \end{aligned}$$

Au final, l'expression régulière associée à cet automate est :

$$b^*a(b + ab^*a)^*.$$

Exercice 5 (Un automate à trouver - 3 points)

Étant donné un mot w et un sous-mot w' de w, on note $|w|_{w'}$ le nombre d'occurrences du mot w' dans w.

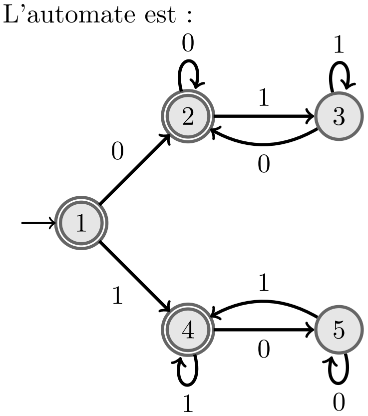
1. On considère l'alphabet $\{0, 1\}$. Donner, sans preuve ni justification, un automate qui reconnaît le langage

$$\{w \in \{0,1\}^* \mid |w|_{01} = |w|_{10}\}.$$

Solution de l'exercice 5

Barème : 3 automate juste, 1,5 petite erreur, 0 dans les autres cas.

1. L'automate est :



Exercice 6 (Un langage non régulier - 3 points)

Étant donné un alphabet quelconque, on note w^R, le mot miroir de w, c'est-à-dire le mot obtenu en lisant w de droite à gauche.

1. On considère l'alphabet $\{a, b\}$. Prouver que $\{w \cdot w^R \mid w \in \{a, b\}^*\}$ n'est pas un langage régulier.

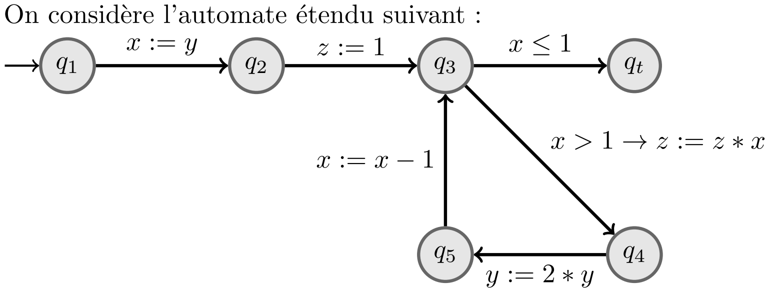
Solution de l'exercice 6

Barème : 1 point pour le lemme de l'itération correctement énoncé, 2 points en fonction de la qualité de la démonstration.

1. Notons L le langage de l'énoncé. Nous faisons une démonstration par contradiction. Supposons que L soit régulier. D'après le lemme de l'itération, il existe une constante $n \in \mathbb{N}$ telle que, pour tout mot w de L de longueur supérieure à n, il existe une décomposition de w avec trois sous-mots $x, y, z \in \{a, b\}^*$ tels que :
   – $w = x \cdot y \cdot z$
   – $|x \cdot y| \leq n$
   – $y \neq \epsilon$
   – $\forall k \in \mathbb{N} : x \cdot y^k \cdot z \in L$.
   Considérons le mot $w = a^n \cdot b \cdot b \cdot a^n$. Nous avons $w \in L$ et $|w| \geq n$. En utilisant $w = xyz$, $|xy| \leq n$, on obtient $y = a^i$ avec $i \geq 0$. En utilisant $y \neq \epsilon$, nous obtenons $i > 0$. En utilisant $\forall k \in \mathbb{N} : xy^kz \in L$, pour $k = 0$, nous obtenons $a^{n-i} \cdot b \cdot b \cdot a^n \in L$. Or $n - i < n$ car $i > 0$. On ne peut pas trouver de $w' \in \{a, b\}^*$ tel que $a^{n-i} \cdot b \cdot b \cdot a^n = w' \cdot {w'}^R$.
   Ceci est une contradiction.

Exercice 7 (Vérification d'automate étendu - 5 points)

On considère l'automate étendu suivant :



1. Donner l'exécution de cet automate sur un état initial tel que la valeur de y est 1, et toutes les autres variables sont initialisées à 0.
2. Donner l'exécution de cet automate sur un état initial tel que la valeur de y est 2, et toutes les autres variables sont initialisées à 0.
3. Quelles sont les valeurs finales de y et z après une exécution où y est initialisée à 3. On ne demande pas d'exécution.
4. Montrer que cet automate étendu est partiellement correct par rapport à la spécification

$$\Big(y > 0, z = y_0! \wedge y = 2^{y_0 - 1} * y_0\Big).$$

   L'invariant est de la forme $\ldots! * z = y_0! \wedge \ldots = y_0 * 2^{\cdots}$.
   Rappel : la factorielle d'un entier naturel n est notée $n!$.

Solution de l'exercice 7

1. **(0,5 points)**

etat	x	y	z
q_1	0	1	0
q_2	1	1	0
q_3	1	1	1
q_t	1	1	1

2. **(0,5 points)**

etat	x	y	z
q_1	0	2	0
q_2	2	2	0
q_3	2	2	1
q_4	2	2	2
q_5	2	4	2
q_3	1	4	2
q_t	1	4	2

3. **(0,5 points)**

etat	x	y	z
q_1	0	3	0
q_2	3	3	0
q_3	3	3	1
q_4	3	3	3
q_5	3	6	3
q_3	2	6	3
q_4	2	6	6
q_5	2	12	6
q_3	1	12	6
q_t	1	**12**	**6**

4. **(1 point pour l'invariant correct, 0,5 point pour pre/post, 2 points pour la correction des transitions)**
Nous prenons les prédicats suivants :
– $P_{q_1} \equiv y > 0$,
– $P_{q_2} \equiv y_0 = x! \wedge y = y_0 * 2^{y_0 - x} \wedge x > 0$,
– $P_{q_3} \equiv x! * z = y_0! \wedge y = y_0 * 2^{y_0 - x} \wedge x > 0$ (invariant),
– $P_{q_4} \equiv (x-1)! * z = y_0! \wedge y = y_0 * 2^{y_0 - x} \wedge x > 0$,
– $P_{q_5} \equiv (x-1)! * z = y_0! \wedge y = y_0 * 2^{y_0 - x + 1} \wedge x > 0$,
– $P_{q_t} \equiv z = y_0! \wedge y = 2^{y_0 - 1} * y_0$.
Il faut ensuite montrer que :
– La pré-condition implique P_{q_1}.
– La post-condition est impliquée par P_{q_t}
– L'automate est inductif, c'est-à-dire pour chaque transition $q \overset{b \to x := e}{\longrightarrow} q'$ de l'automate où q et q' sont des états, b une expression booléenne, x une variable et e une expression arithmétique, pour tout état σ, nous devons montrer : si $\sigma \models P_q \wedge b$ alors $\sigma\left[\llbracket e \rrbracket_\sigma / x\right] \models P_{q'}$