Licence Sciences et Technologies
Univ. Grenoble Alpes

Examen final
INF 302: Langages et Automates
L2, 2016/2017

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 22 points.

Exercice 1 (Vrai ou Faux - 3 points)

Indiquer si les affirmations suivantes sont vraies ou fausses. Justifier *soigneusement* vos réponses.

1. Un automate complet déterministe dont tous les états sont accepteurs reconnaît le langage universel.
2. La différence de deux langages réguliers est un langage régulier.
3. Il est possible que des langages non-réguliers satisfassent le lemme de l'itération.
4. Dans la méthode de Floyd, la post-condition doit être impliquée par au moins une des prédicats associé à un des états terminaux.
5. Il est possible que l'entier naturel 0 soit la constante d'itération d'un langage.
6. Le nombre minimal d'étapes nécessaires lors de l'exécution de l'algorithme de minimisation est 1, c'est-à-dire que dans certain cas, il suffit de calculer $\equiv_0$.

Solution de l'exercice 1

1. Vrai. L'exécution de tout mot est définie et termine dans un état accepteur.
2. Vrai. Pour deux langages réguliers E, F quelconques, $E \setminus F = E \cap \overline{F}$. D'après la fermeture des langages réguliers par les opérations de complémentation et d'intersection, nous obtenons que $E \setminus F$ est régulier.
3. Vrai. Le lemme de l'itération est satisfait par tous les langages réguliers. Rien n'est indiqué pour les langages non-réguliers.
4. Faux. La post-condition doit être impliquée par tous les prédicats associés aux états terminaux.
5. Vrai. Considérons le langage vide (qui est régulier), le lemme de l'itération s'applique pour $N = 0$. Le langage ne contient aucun mot, donc le lemme de l'itération s'applique pour tout mot du langage de longueur supérieure ou égale à 0.
6. Vrai. Cela peut être le cas si l'automate (complet et déterministe) sur lequel on applique l'algorithme possède deux états, un accepteur, l'autre non-accepteur. Les deux états sont dans des classes d'équivalence différentes de cardinal 1.

Exercice 2 (Expression régulière vers automate - 3 points)

Nous considérons l'expression régulière suivante :

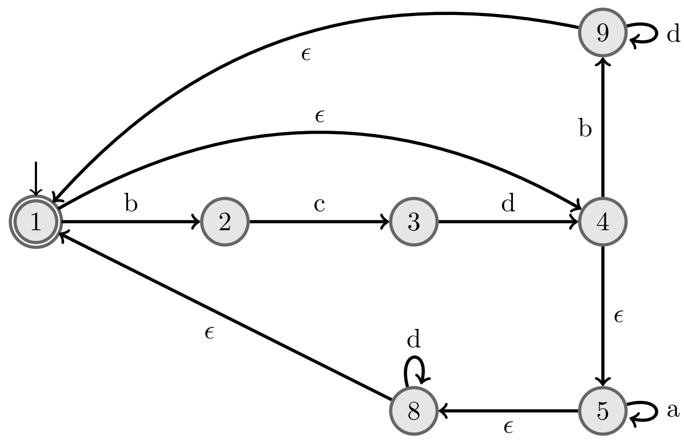
$$\left(b \cdot d \cdot d + (a^* + b \cdot c \cdot d + \epsilon) \cdot (a^* \cdot d^* + b \cdot d^*) + a^+ \cdot d^+ + \epsilon\right)^* \cdot (b \cdot c \cdot d)^*.$$

1. Simplifier l'expression régulière ; c'est-à-dire donner une expression régulière impliquant moins de symboles et dénotant le même langage.
2. Donner un automate non-déterministe avec ϵ-transitions qui reconnait le langage dénoté par l'expression régulière simplifiée obtenue à la question précédente.

Solution de l'exercice 2

1.
$$((b \cdot c \cdot d + \epsilon) \cdot (a^* \cdot d^* + b \cdot d^*))^*$$

2. Un automate reconnaissant l'expression régulière est donné ci-dessous :



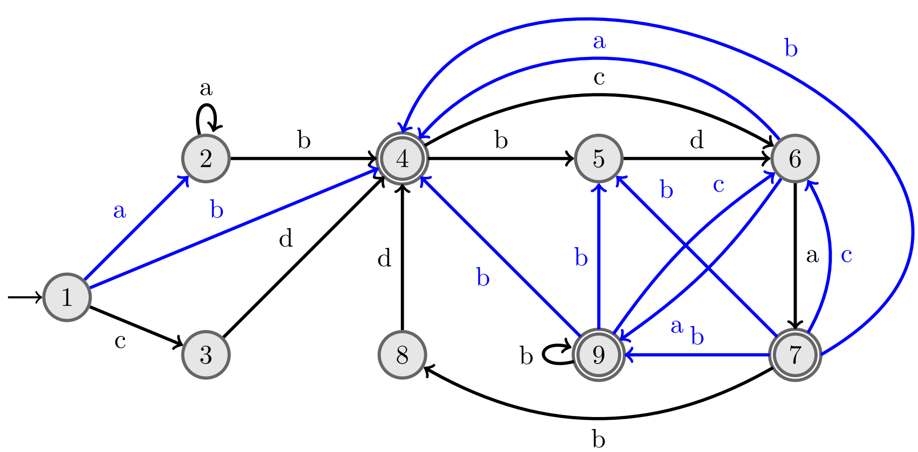
Exercice 3 (Transformations d'automate - 4 points)

Nous considérons l'automate d'états finis non déterministe avec ϵ-transitions représenté dans la Figure 1a. Il n'est pas autorisé de répondre aux deux questions en même temps.

1. Supprimer les ϵ-transitions, c'est-à-dire, donner un automate d'états finis non-déterministe sans ϵ-transitions qui reconnaît le même langage.
2. Déterminiser l'automate obtenu à la question précédente.

Solution de l'exercice 3

1. L'automate ci-dessous est l'automate résultant de la suppression des ϵ-transitions. Les transitions ajoutées sont en bleu. Les états 7 et 9 deviennent accepteurs car, dans l'automate intitial, l'ϵ-cloture de ces états contient un état accepteur (l'état 4).



2. Nous représentons l'automate de la question précédente sous forme tabulaire avant déterminisation.

	1	2	3	4	5	6	7	8	9
a	2	2				$4,7,9$			
b	4	4		5			$4,5,8,9$		$4,5,9$
c	3			6			6		6
d			4		6			4	

Nous appliquons l'algorithme de déterminisation. Nous obtenons l'automate représenté par le tableau suivant.

	1	2	3	4	5	6	$4,7,9$	$4,5,8,9$	$4,5,9$	$4,6$
a	2	2				$4,7,9$				$4,7,9$
b	4	4		5			$4,5,8,9$	$4,5,9$	$4,5,9$	5
c	3			6			6	6	6	6
d			4		6			$4,6$	6	

Exercice 4 (Langages non-réguliers - 4 points)

1. Montrer que le langage $L = \{w \mid w \in \{0,1\}^* \text{ et } w = w^R\}$ n'est pas régulier.
2. En supposant que le langage $L_1 = \{a^n b^l c^{l+n} \mid n \geq 0 \text{ et } l \geq 0\}$ est non régulier, déduire que le langage $L_2 = \{w \in \{a,b,c\}^* \mid |w|_a + |w|_b = |w|_c\}$ est non régulier.

Solution de l'exercice 4

1. Supposons que L soit un langage régulier. Soit n la constante du lemme d'itération et soit $w = 0^n 1 0^n$. Il est clair que $w \in L$, et $|w| \geq n$. Soit $w = xyz$ la décomposition avec x, y, z fourni par le lemme de l'itération avec $|xy| \leq n$ et $|y| \geq 1$, Soit $k = |y|$, il faut noter que $0 < k \leq n$. Alors $xy^0z = 0^{n-k}10^n$ n'appartient pas à L, parce que si c'était le cas, alors on aurait $0^{n-k}10^n = (0^{n-k}10^n)^R = 0^n10^{n-k}$, et donc $n - k = n$, ce qui est impossible car $k \neq 0$.
2. Nous utilisons la propriété de fermeture des langages réguliers par intersection ensembliste et nous observons que $L_1 = L_2 \cap a^* \cdot b^* \cdot c^*$. Ainsi si L_2 était régulier, alors L_1 serait régulier.

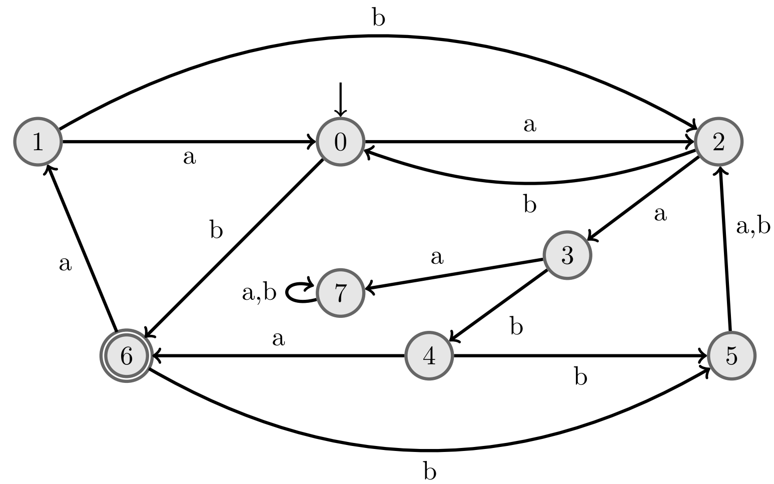
Exercice 5 (Minimisation d'automate - 3 points)

Nous considérons l'automate d'états finis déterministe représenté dans la Figure 1b.

1. Donner l'automate minimisé reconnaissant le langage reconnu par cet automate.

Solution de l'exercice 5

1. Nous complétons l'automate avant d'appliquer l'algorithme de minimisation.



Les étapes du calcul sont représentées ci-dessous.

$\equiv_0$	$\equiv_1$	$\equiv_2$	$\equiv_3$
0	0	0	0
1	1	1	1
2	2	2	2
3	3	3	3
4	5	5	5
7	7	7	7
5	4	4	4
6	6	6	6

Chaque classe d'équivalence est de cardinal 1. L'automate est donc minimal.

Exercice 6 (Méthode de Floyd - 5 points)

Nous considérons l'automate étendu A avec variables entières et représenté sur la Figure 2. L'état de contrôle q_1 est l'état de contrôle initial et l'état de contrôle q_t est l'unique état de contrôle final.

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. En utilisant la méthode de Floyd, 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 $n!$. Lors de la preuve d'inductivité de l'automate, ne considérer que les transitions de q_3 vers q_4, de q_4 vers q_5 et de q_3 vers q_t.

Solution de l'exercice 6

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

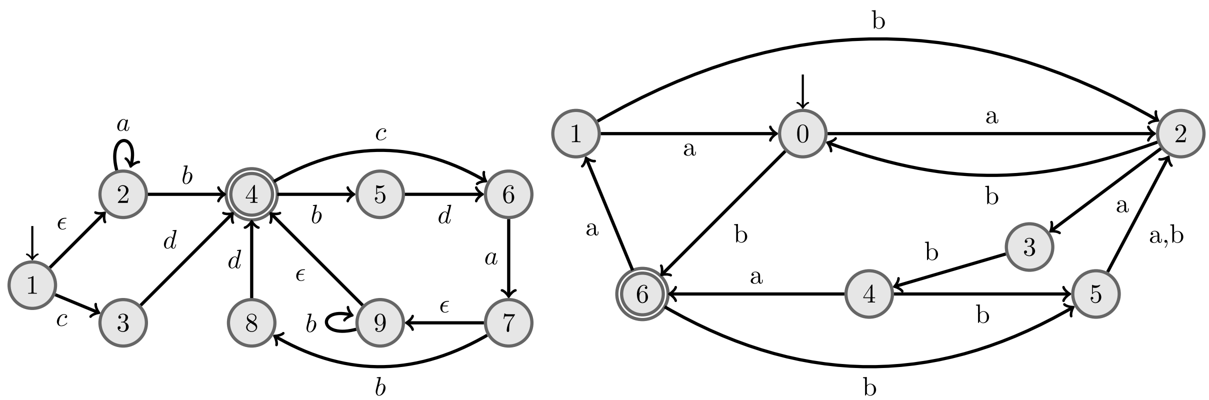
2. **(0,5 points)**

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 $\sigma\left[\llbracket e \rrbracket_\sigma / x\right] \models P_{q'}$



(a) Un automate pour l'Exercice 3

(b) Un automate pour l'Exercice 5

Figure 1: Automates pour les Exercices 3 et 5

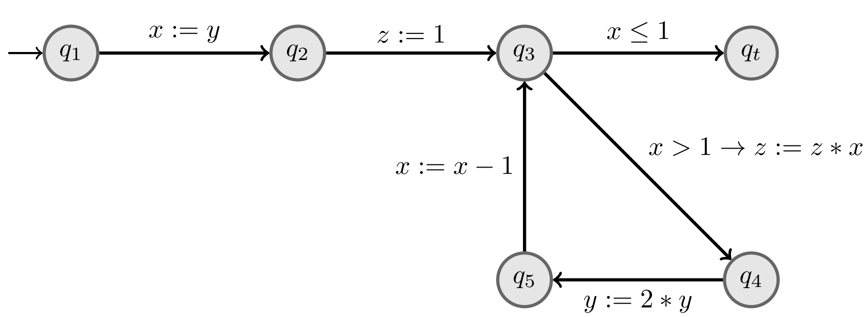


Figure 2: Automate étendu A pour l'Exercice 6