Licence Sciences et Technologies
Université Joseph Fourier

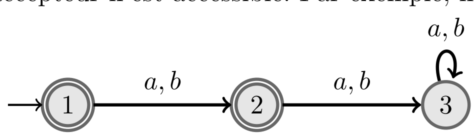
Examen à mi-parcours
INF 232 : Langages et Automates
L2, 2014/2015

Rappel à propos des consignes et quelques conseils et remarques

— Durée : 2 heures.
— Pas de sortie avant 30 minutes. Pas d'entrée après 30 minutes.
— Tout document du cours ou du TD est autorisé. Tout autre document est interdit.
— Tout dispositif électronique est interdit (calculette, téléphone, tablette, etc.).
— **Le soin de votre copie sera pris en compte (-1 point si manque de soin).**
— Les exercices sont indépendants.
— Le barème est donné à titre indicatif.

Answer of exercise 1

1. Faux. Le langage défini par l'expression régulière a^* est infini. Ce langage étant régulier, par le théorème de Kleene, il est également d'états-finis.
2. Vrai. Un langage fini contient un nombre fini de mots. Soient $w_1, \ldots, w_n$ les mots d'un langage fini. Chaque mot peut être représenté par une expression régulière. L'expression régulière pour ce langage est $w_1 + \ldots + w_n$.
3. Faux. Pour construire un contre-exemple, on peut considèrer n'importequel automate (déterministe ou non) sans état accepteur. Cet automate reconnaitra le langage vide.
4. Faux. Il est possible de construire un automate dont la boucle se trouve depuis lequel aucun état accepteur n'est accessible. Par exemple, nous pouvons considérer l'automate suivant :



5. Vrai. Considérons l'alphabet $\Sigma = \{a, b\}$ et les langages :
   — L_1 : langage des mots sur Σ qui contiennent autant de a que de b.
   — L_2 : langage des mots sur Σ qui contiennent un nombre différent de a et de b.
   Les langages L_1 et L_2 sont non réguliers. Cependant $L_1 \setminus L_2 = \emptyset$ est régulier.
6. Vrai. Nous avons par le théorème de Kleene que comme L est régulier, alors L est d'états finis. De plus, $L \cap L = L$.

Answer of exercise 2

Dans cet exercice, lorsque nous représentons un automate sous forme tabulaire, l'état initial est souligné et les états accepteurs sont associés au symbole $*$.

1. Représentons l'automate sous forme tabulaire :

	$\underline{1}*$	$2*$	$3*$	4
ϵ	1	4		
a	$1,3$	2		$3,4$
b	$2,3$	$2,4$		4
c		$2,3$	3	4

Supprimons les ϵ-transitions :

	$\underline{1}*$	$2*$	$3*$	$4*$
a	$1,3$	$2,3,4$		$3,4$
b	$2,3,4$	$2,3$		4
c		$2,3,4$	3	4

Déterminisons :

	$\underline{1}*$	$1,3*$	$2,3,4*$	$2,4*$	$3*$
a	$1,3$	$1,3$	$2,3,4$	$2,3,4$	
b	$2,3,4$	$2,3,4$	$2,4$	$2,4$	
c		3	$2,3,4$	$2,3,4$	3

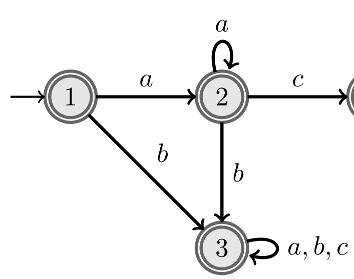
2. Renommons les états de l'automate et complétons le [1]

	$\underline{1}*$	$2*$	$3*$	$4*$	$5*$	6
a	2	2	3	3	6	6
b	3	3	4	4	6	6
c	6	5	3	3	5	6

Appliquons l'algorithme de minimisation :

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

Les états 3 et 4 sont équivalents. L'automate non-complet minimal est (après renommage des états) donc :



3. Le système d'équation associé à cet automate est :

$$
\begin{aligned}
X_1 &= aX_2 + bX_3 + \epsilon \\
X_2 &= aX_2 + bX_3 + cX_4 + \epsilon \\
X_3 &= (a+b+c)X_3 + \epsilon \\
X_4 &= cX4 + \epsilon
\end{aligned}
$$

L'automate est sans ϵ-transitions donc le lemme d'Arden nous donne une solution minimale est unique à chaque résolution d'équation.
En utilisant $X_3 = (a+b+c)X_3 + \epsilon$, et en appliquant le lemme d'Arden, nous obtenons $X_3 = (a+b+c)^*$.
En utilisant $X_4 = cX_4 + \epsilon$, nous obtenons $X_4 = c^*$.
En utilisant $X_2 = aX_2 + bX_3 + cX_4 + \epsilon$ et les langages trouvés pour X_3 et X_4, nous obtenons $X_2 = aX_2 + b(a+b+c)^* + c^+ + \epsilon$. En appliquant le lemme d'Arden, nous obtenons :

$$X_2 = a^*(b(a+b+c)^* + c^+ + \epsilon).$$

Finalement, en utilisant $X_1 = aX_2 + bX_3 + \epsilon$, nous obtenons :

$$X_1 = a^+(b(a+b+c)^* + c^+ + \epsilon) + b(a+b+c)^* + \epsilon.$$

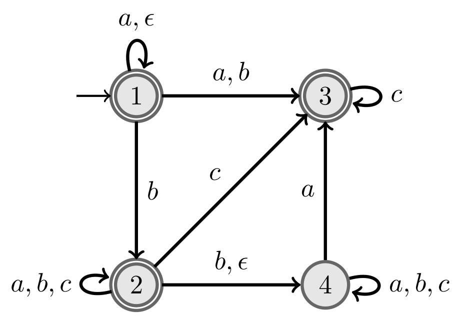


FIGURE 1: Automate pour l'exercice 2

Answer of exercise 3

Considérons l'alphabet $\Sigma = \{a, b\}$

1. Considérons les langages :
   — L_1 : langage des mots sur Σ qui contiennent plus de a que de b.
   — L_2 : langage des mots sur Σ qui contiennent strictement plus de a et de b.
   Le langage $L_1 \setminus L_2 = \emptyset$ est le langage des mots qui contiennent autant de a que de b. Les langages L_1, L_2 et $L_1 \setminus L_2 = \emptyset$ sont non réguliers.
2. Nous proposons deux solutions :
   — Considérons un langage L_1 non réguliers. Soit $L_2 = L_1$. Alors, nous avons $L_1 \setminus L_2 = \emptyset$, qui est un langage régulier.
   — Considérons les langages :
     — L_1 : langage des mots sur Σ qui contiennent autant de a que de b.
     — L_2 : langage des mots sur Σ qui contiennent un nombre différent de a et de b.
     Les langages L_1 et L_2 sont non réguliers. Cependant $L_1 \setminus L_2 = \emptyset$ est régulier.

Answer of exercise 4

1. Un algorithme qui répond au problème est donné ci-dessous :

Algorithm 1 Algorithme pour la question 1 de l'exercice 4

Require: E_1, E_2, E_3 : expressions régulières
Ensure: *vrai* si $L(E_1 \cdot E_2) = L(E_3^*)$, *faux* sinon
$auto :=$ ER2Automate$(E_1 \cdot E_2)$
$auto' :=$ ER2Automate(E_3^*)
return sont_equivalents$(auto, auto')$

2. Un algorithme qui répond au problème est donné ci-dessous :

1. La complétion de l'automate n'est pas obligatoire.

Algorithm 2 Algorithme pour la question 2 de l'exercice 4

Require: E_1, E_2 : expressions régulières
Ensure: une chaîne de caractère qui indique si $|L(E_1)| = |L(E_2)|$, $|L(E_1)| < |L(E_2)|$ ou $|L(E_1)| > |L(E_2)|$

```
auto1 := ER2Automate(E1) ; auto2 := ER2Automate(E2)
infini1 := reconnaitLangageInfini(auto1) ; infini2 := reconnaitLangageInfini(auto2)
if infini1 && infini2 then
    return "|L(E1)| = |L(E2)|"
else
    if infini1 then
        return "|L(E1)| > |L(E2)|"
    else
        if infini2 then
            return "|L(E1)| < |L(E2)|"
        else
            entier cpt1 := 0, cpt2 := 0
            entier n := max(|Q1|, |Q2|)          ▷ Q1, Q2 sont les ensembles d'états de auto1, auto2
            alphabet Σ = Σ1 ∪ Σ2                 ▷ Σ1, Σ2 sont les alphabets de auto1, auto2
            for chaque mot w de Σ^i, i = 1, ..., n do   ▷ On génère tous les mots de Σ^n
                if est_accepte(w, auto1) then
                    cpt1 := cpt1 + 1
                end if
                if est_accepte(w, auto2) then
                    cpt2 := cpt2 + 1
                end if
            end for
            if "cpt1 > cpt2" then
                return "|L(E1)| > |L(E2)|"
            else
                if "cpt1 < cpt2" then
                    return "|L(E1)| < |L(E2)|"
                else
                    return "|L(E1)| = |L(E2)|"
                end if
            end if
        end if
    end if
end if
```

Answer of exercise 5

Dans cet exercice, lorsque nous représentons un automate sous forme tabulaire, l'état initial est souligné et les états accepteurs sont associés au symbole $*$.

1. Représentons l'automate sous forme tabulaire :

	$\underline{1}$	$2*$	3	$4*$	5
ϵ	4		2	5	
a	2	4			5
b		3			

Supprimons les ϵ-transitions :

	$\underline{1}$	$2*$	3	$4*$	5
a	$2,5$	$4,5$	$4,5$	5	5
b		$2,3$	3		

Déterminisons :

	$\underline{1}$	$2,5*$	$4,5*$	$2,3*$	5
a	$2,5$	$4,5$	5	$4,5$	5
b		$2,3$		$2,3$	

2. Renommons les états et complétons l'automate :

	$\underline{1}$	$2*$	$3*$	$4*$	5	6
a	2	3	5	3	5	6
b	6	4	6	4	6	6

Appliquons l'algorithme de minimisation :

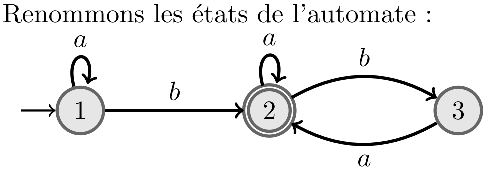
$\equiv_0$	$\equiv_1$	$\equiv_2$
2	3	3
3	2	2
4	4	4
1	1	1
5	5	5
6	6	6

L'automate minimisé est le suivant :

	$\underline{1}$	$24*$	$3*$	56
a	2	3	56	56
b	56	24	56	56

Answer of exercise 6

1. Renommons les états de l'automate :



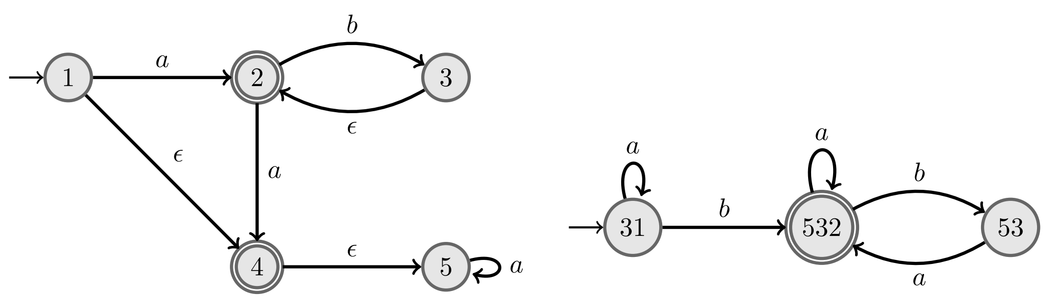
— $R^0_{11} = \epsilon + a$
— $R^0_{12} = b$
— $R^0_{13} = \emptyset$
— $R^0_{21} = \emptyset$
— $R^0_{22} = \epsilon + a$
— $R^0_{23} = b$
— $R^0_{31} = \emptyset$
— $R^0_{32} = a$
— $R^0_{33} = \epsilon$

— $R^1_{11} = a^*$
— $R^1_{12} = b + (\epsilon + a)(\epsilon + a)^* b = a^* b$
— $R^1_{13} = \emptyset$
— $R^1_{21} = \emptyset$
— $R^1_{22} = a + \epsilon$
— $R^1_{23} = b$
— $R^1_{31} = \emptyset$
— $R^1_{32} = a$
— $R^1_{33} = \epsilon$

— $R^2_{11} = a^*$
— $R^2_{12} = a^* b a^*$
— $R^2_{13} = a^* b a^* b$
— $R^2_{21} = \emptyset$
— $R^2_{22} = a^*$
— $R^2_{23} = a^* b$
— $R^2_{31} = \emptyset$
— $R^2_{32} = a^+$
— $R^2_{33} = a^+ b + \epsilon$

L'expression régulière de l'automate est

$$\begin{aligned} R^3_{12} &= R^2_{12} + R^2_{13}(R^2_{33})^* R^2_{32} \\ &= a^* b a^* + a^* b a^* b (a^+ b + \epsilon)^* a^+ \\ &= a^* b a^* (\epsilon + b(a^+ b + \epsilon)^* a^+) \\ &= a^* b (a + ba)^* \end{aligned}$$



(a) Automate pour l'exercice 5 (b) Automate pour l'exercice 6

FIGURE 2: Automates pour les exercices 5 et 6