Année Universitaire 2025/2026
Licence Informatique

Université de La Rochelle
Département Informatique

Compilation

Autour des automates

K. Bertet, I. Nass

Correction

1 Préambule

Les exercices de cette feuille TD ont pour objectif d'acquérir les compétences suivantes :

1. Savoir définir et manipuler un automate (exercices 1 et 2 - pour aller plus loin : exercices 3 et 4)
2. Savoir manipuler un automate à sortie (exercice 5)
3. Savoir déterminiser un automate (exercices 7 et 8 - pour aller plus loin : exercice 9)

Il vous revient d'avancer à votre rythme, et de demander à l'enseignant des éléments de correction si nécessaire. Le travail à plusieurs est fortement conseillé.

2 Quelques rappels de cours

— Un automate se définit par un un tuple (V, E, I, F, Δ) où :
  — V est un alphabet d'entrée, i.e un ensemble de lettres (ou de symboles)
  — E est un ensemble d'*états*
  — $I \subseteq E$ est l'ensemble des *états initiaux* (généralement un ensemble singleton)
  — $F \subseteq E$ est l'ensemble des *états finaux* ou d'acceptation
  — $\Delta \subseteq E \times V \times E$ est l'ensemble des *transitions*, où une transition est un triplet $(e, a, e') \in \Delta$ signifiant *si je suis dans l'état e et que je lis la lettre a alors je vais dans l'état e'.*

— Un automate se représente graphiquement par un graphe où chaque noeud correspondent à un état $e \in E$, et chaque arc $e \xrightarrow{a} e'$ à une transition $(e, a, e') \in \Delta$. Les états initiaux et finaux sont indiqués. On peut représenter l'ensemble des transitions d'un automate par sa *table de transitions* avec les états en ligne, les symboles en colonnes.

— Un automate sert à reconnaître des mots. Pour un mot en entrée, et un état courant initialisé par l'état initial, il s'agit de *franchir* des transitions en consommant chaque caractère du mot d'entrée. Le mot est reconnu si, lorsque toutes les lettres sont consommées, l'état courant est un état acceptant. L'ensemble des mots acceptés par un automate A est appelé le *langage reconnu par l'automate.*

— Un automate à sortie permet en plus d'écrire un mot de sortie. Il s'agit d'un automate décrit par un tuple (V, E, O, I, F, Δ) où O est un alphabet de sortie, et une transition $(e, a, b, e') \in \Delta$, avec $b \in O$, signifie *si je suis dans l'état e et que je lis la lettre a alors j'écris la lettre b et je vais dans l'état e'.*

— Un automate est *complet* si, pour tout état $e \in E$, il existe au moins une transition pour tout symbole $a \in V$. Rendre un automate complet consiste à lui ajouter un état puit p, puis ajouter toutes les transitions manquantes (e, a, p).

— Un automate est *déterministe* si, pour tout état $e \in E$, il existe au plus une transition pour tout symbole $a \in V$. L'algorithme de déterminisation vu en cours permet de rendre un automate déterministe ;

— Une λ-transition est une transition sur le mot vide λ, qui peut étre franchie sans consommation de caractère. Pour un état e, sa λ-fermeture est l'ensemble des états atteignables à partir de e en franchissant des λ-transitions.

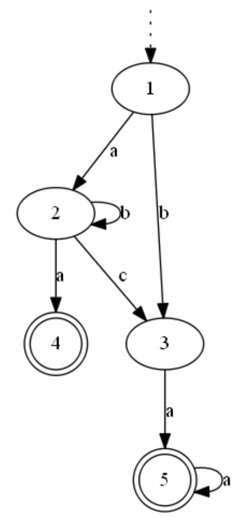
Pour plus d'informations sur les automates, consulter le document **Eléments d'analyse lexicale**

3 Automates et expressions régulières

Exercice 1.

On considère l'automate $\mathcal{A}$.

1. Combien d'états possède cet automate ? Combien de transitions ? Quel est son état initial ? Quels sont ses états finaux ?
2. Définissez cet automate sous sa forme (V, E, I, F, Δ) en représentant les transitions par des triplets puis par une table de transitions.
3. Dans quel état se trouve cet automate aprés avoir lu le mot *ba* ? Ce mot est-il reconnu par l'automate ? Mêmes questions avec les mots *ab*, *ac* puis *bc*.
4. Donner tous les mots de longueur 2, de longueur 3 et de longueur 4 reconnus par cet automate.
5. Décrire tous les mots reconnus par cet automate à l'aide d'une expression régulière.
6. Rendre cet automate complet en ajoutant un puit. Le mot *cba* est-il reconnu/accepté par l'automate ?



Automate $\mathcal{A}$

Correction:

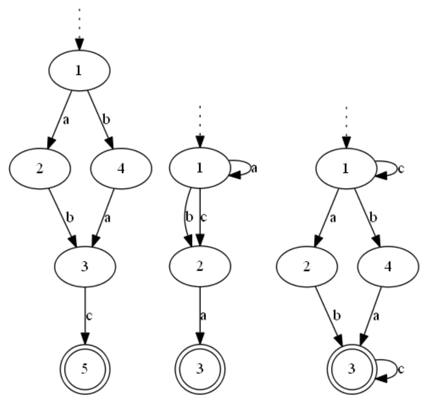
1. 5 états - 7 transitions/flèches - 1 état initial, l'état 1 - 2 états finaux, les états 4 et 5
2. Définition formelle :
   — Alphabet d'entrée : $V = \{a, b, c\}$
   — Ensemble des états : $E = \{1, 2, 3, 4, 5\}$
   — Etat initial : 1
   — Etats finaux : $F = \{4, 5\}$
   — Transitions (triplets ou table de transitions) : $(1, a, 2)$; $(2, b, 2)$; $(2, a, 4)$; $(1, b, 3)$; $(2, c, 3)$; $(3, a, 5)$; $(5, a, 5)$
3. $1 \stackrel{a}{\to} 2 \stackrel{b}{\to} 2$; $1 \stackrel{a}{\to} 2 \stackrel{c}{\to} 3$; $1 \stackrel{b}{\to} 3 \stackrel{c}{\to} ?$
4. 2 mots de longueur 2 : aa,ba - 3 mots de longeur 3 : aba, aca, baa - 4 mots de longeur 4 : abba, abca, acaa, baaa
5. Expression régulière : $ab^*a + ab^*caa^* + baa^* = ab^*a + ab^*ca^+ + ba^+$
6. Automate détermniste (pas de transition avec la même lettre partant d'un même état), mais pas complet, des transitions "manquantes" : $1 \stackrel{c}{\to} ?$; $3 \stackrel{b,c}{\to} ?$; $4 \stackrel{a,b,c}{\to} ?$; $5 \stackrel{b,c}{\to} ?$. Pour le rendre complet, ajouter un état puit vers lequel faire pointer toutes les transitions manquantes. Ne pas oublier les transitions boucles dans l'état puit : $puit \stackrel{a,b,c}{\to} puit$.

◇

Exercice 2. Donner les automates permettant de reconnaître les expressions régulières suivantes :

1. $(abc) + (bac)$
2. $a^*(b + c)a$
3. $c^*(ab)c^* + c^*(ba)c^*$

Correction: Il peut y avoir plusieurs solutions. Les solutions ici sont des automates non complets.

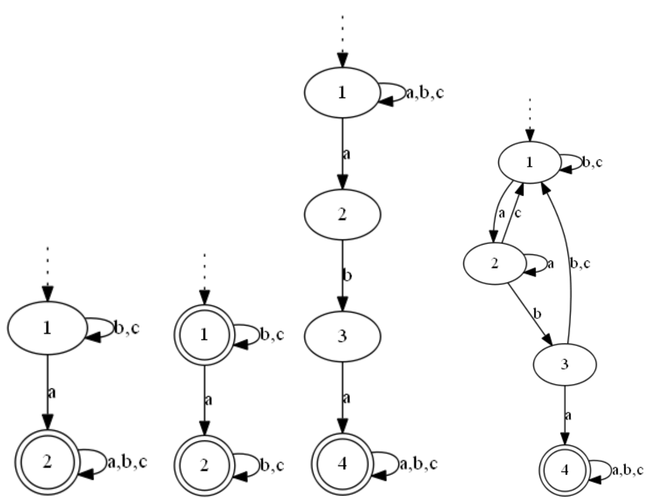


◇

Exercice 3. (pour aller plus loin) Donner les automates permettant de reconnaître les langages suivants sur l'alphabet $\{a, b, c\}$:

1. L'ensemble des mots contenant au moins une fois la lettre a
2. L'ensemble des mots contenant au plus une fois la lettre a
3. L'ensemble des mots admettant aba pour facteur

Correction: Deux solutions pour (3), l'une non déterministe, l'autre déterministe.

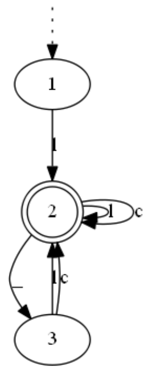


◇

Exercice 4. (pour aller plus loin) Un identificateur est une suite de lettres, de chiffres ou de _ qui commence par une lettre, ne finit pas par _ et ne contient pas deux fois _ à la suite.

1. Donner l'expression régulière permettant de décrire tous les identificateurs possibles, en utilisant l pour une lettre et c pour un chiffre.
2. Construire un automate permettant de reconnaître les identificateurs.

Correction:



Automate $\mathcal{A}$

Expression régulière : $L(A) = l(c|l)^*((\_(c|l)^+))^*$

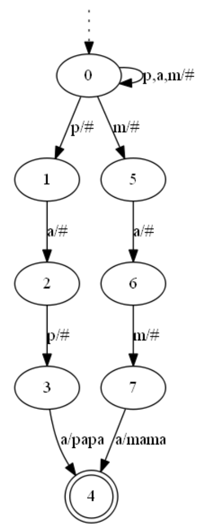
◇

4 Automates à sortie

Exercice 5.

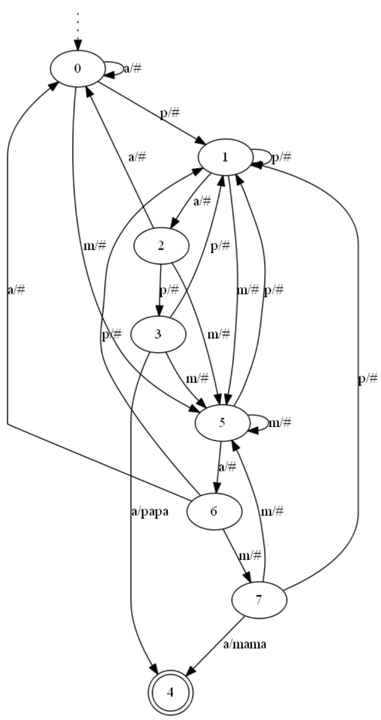
On considère l'automate à sortie $\mathcal{PM}$.

1. Donner l'alphabet d'entrée et de sortie de cet automate.
2. Sur quels mots d'entrée une sortie autre que # est-elle produite ?
3. Donner l'expression régulière décrivant les mots reconnus par cet automate.
4. Modifier cet automate afin qu'il soit complet et déterministe.



Automate $\mathcal{PM}$

Correction:

1. Alphabet d'entrée, 3 lettres : A={a,p,m}. Alphabet de sortie, deux mots/léxémes : {papa,mama}
2. Les mots *papa* et *mama* sont produits en sortie lorsqu'ils sont reconnus
3. Expression régulière : $A^*(papa + mama)$
4. 

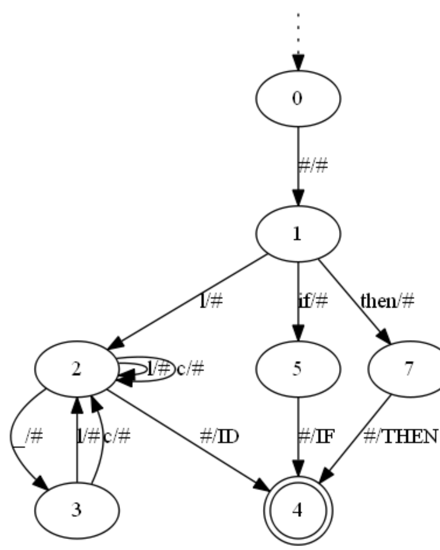
◇

Exercice 6. (pour aller plus loin)

1. Transfomer l'automate reconnaissant les identificateurs en un automate à sortie renvoyant le token ID lorsqu'un identificateur est reconnu.

2. Modifier cet automate afin qu'il puisse également reconnaître les mots *if*, *then* et *else*, et renvoyer les tokens ID, IF et $THEN$.
3. Cet automate est-il complet ? déterministe ?

Correction: Solution avec des multilabels $1 \overset{if}{\to} 5$ et $1 \overset{then}{\to} 7$. Solution non déterministe : dans l'état 1, si un identificateur commence par un i ou un t, on ne sait pas vers quel état aller.

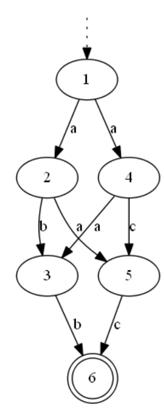


◇

5 Déterminisation

Exercice 7.

1. On considère l'automate $\mathcal{B}$ ci-contre.
   (a) Calculez X =transiter($\{1\}$,a)
   (b) Calculez Y =transiter(X,a)
   (c) Calculez Z =transiter(X,b)
   (d) Calculez T =transiter(Y,b)
2. Calculez l'automate déterministe équivalent à $\mathcal{N}$ par la méthode du cours. Détailler les différents calculs à chaque étape, puis donner l'automate final en indiquant son état initial et ses état acceptants.



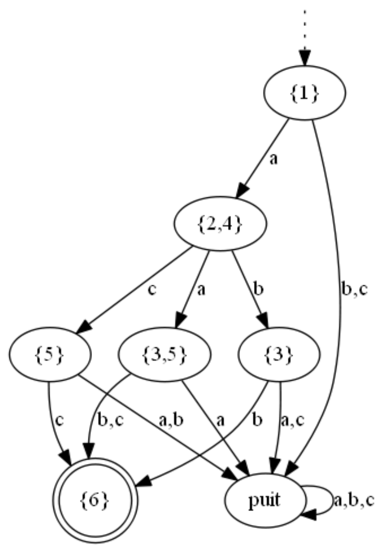
Automate $\mathcal{B}$

Correction:

1. — X =transiter($\{1\}$,a)= $\{2, 4\}$
   — Y =transiter(X,a)= $\{3, 5\}$
   — Z =transiter(X,b)= $\{3\}$
   — T =transiter(Y,b)= $\{6\}$
2. Déterminisation : l'état initial sera le groupe d'états $\{1\}$, car 1 est l'état inital. Puis on applique **transiter(X,a)**, **transiter(X,b)** et **transiter(X,c)** sur chaque groupe d'états X créé, en commençant par $X = \{1\}$. On obtient la table de transitions suivante, où les nouveaux états sont des groupes d'états :

groupe d'états X	**transiter(X,a)**	**transiter(X,b)**	**transiter(X,b)**
$\{1\}$	$\{2,4\}$		
$\{2,4\}$	$\{3,5\}$	$\{3\}$	$\{5\}$
$\{3,5\}$		$\{6\}$	$\{6\}$
$\{3\}$		$\{6\}$	
$\{5\}$			$\{6\}$
$\{6\}$			

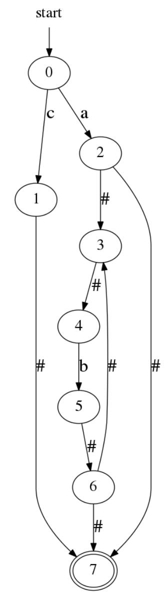
Les états finaux sont les groupes d'états contenant un état final, un seul ici : $\{6\}$. Il s'agit d'un automate déterministe (un seul groupe d'état au plus par case), mais non complet (cases vides). Pour rendre l'automate complet, il suffit de rajouter un état puit P. On obtient alors l'automate équivalent suivant, déterministe et complet :



Automate $\mathcal{B}$ déterminisé

◇

Exercice 8. On considère l'automate $\mathcal{N}$ ci-contre.



Automate $\mathcal{N}$

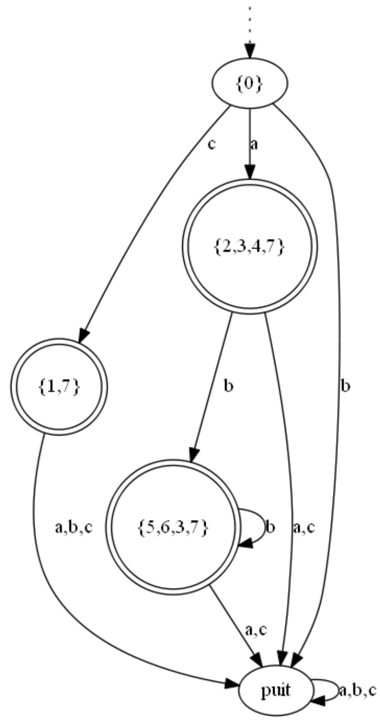
1. Petits calculs de transitions et λ-fermetures
   (a) Calculez $A = \lambda$-fermeture($\{2\}$)
   (b) Calculez $T_{A,b}$ =transiter(A,b)
   (c) Calculez λ-fermeture($T_{A,b}$)
2. Calculez l'automate déterministe équivalent à $\mathcal{N}$ par la méthode du cours. Détailler les différents calculs à chaque étape, puis donner l'automate final en indiquanr son état initial et ses état acceptants.

Correction:

1. — $A = \lambda$-fermeture($\{2\}$)=$\{2, 3, 4, 7\}$.
   — $T_{A,b}$ =transiter(A,b)=$\{5\}$.
   — λ-fermeture($T_{A,b}$)=$\{5, 6, 3, 7\}$.
2. Déterminisation : l'état initial sera le groupe d'états λ-fermeture($\{0\}$)=$\{0\}$, car 0 est l'état inital. Puis on applique λ-fermeture(transiter(X,a)), λ-fermeture(transiter(X,b)) et λ-fermeture(transiter(X,c)) sur chaque groupe d'états X créé. On obtient la table de transitions suivante, où les nouveaux états sont des groupes d'états :

X	λ-fermeture(**trans.(X,a)**)	λ-fermeture(**trans.(X,b)**)	λ-fermeture(**trans.(X,c)**)
$\{0\}$	$\{2, 3, 4, 7\}$		$\{1, 7\}$
$\{2, 3, 4, 7\}$		$\{5, 6, 3, 7\}$	
$\{1, 7\}$			
$\{5, 6, 3, 7\}$		$\{5, 6, 3, 7\}$	

   Les états finaux sont les groupes d'états contenant un état final, ici : $\{2, 3, 4, 7\}$, $\{5, 6, 3, 7\}$ et $\{1, 7\}$. Il s'agit d'un automate déterministe (un seul groupe d'état au plus par case), mais non complet (cases vides). Pour rendre l'automate complet, il suffit de rajouter un état puit P. On obtient alors l'automate équivalent suivant, déterministe et complet :



Automate $\mathcal{N}$ déterminisé

◇

Exercice 9. (pour aller plus loin) Donner l'automate complet et déterministe permettant de reconnaître l'expression régulière $c^*(ab)c^* + c^*(ba)c^*$ en utilisant :

1. La méthode générale du cours qui permet de construire un automate non déterministe avec des λ-transitions à partir d'une expression régulière ;
2. L'algorithme de déterminisation pour rendre cet automate déterministe et complet.