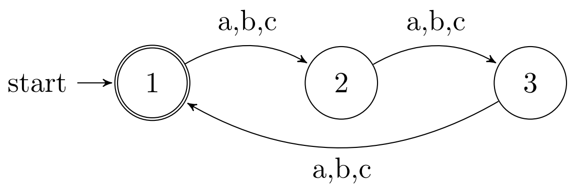
TD 1 Langages rationnels

Timothée Bernard (timothee.bernard@ens-lyon.org)

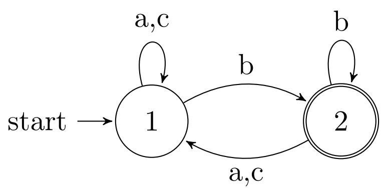
16 septembre 2015

1. Soit $A = \{a, b, c\}$. Pour chacun des langages suivants, donner un automate fini déterministe (AFD) le reconnaissant :
   (a) l'ensemble des mots dont la longueur est un multiple de 3 ;



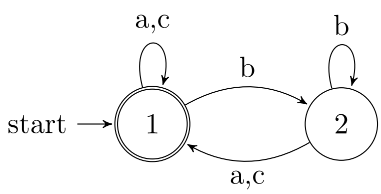
Solution:

   (b) l'ensemble des mots dans lesquels le motif ab, s'il apparaît, est suivi de ccc ;
   (c) l'ensemble des mots se terminant par b ;



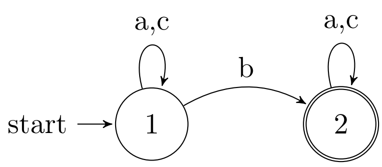
Solution:

   (d) l'ensemble des mots ne se terminant pas par b ;



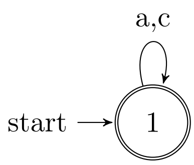
Solution:

   (e) l'ensemble des mots contenant exactement un b ;



Solution:

   (f) l'ensemble des mots ne contenant aucun b ;

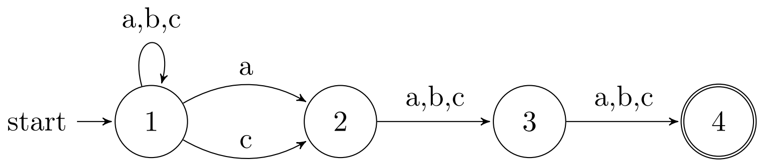
Solution:


(g) l'ensemble des mots contenant au moins un a ;

(h) l'ensemble des mots comportant au moins 3 lettres et dont la troisième lettre à partir de la fin est un a ou un c ;

Solution:

— Soit déterminiser l'automate suivant :

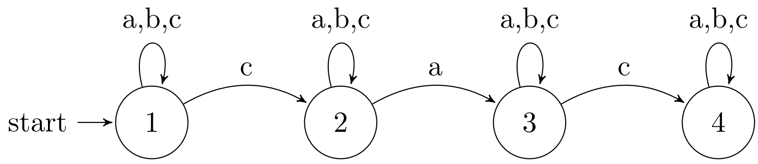


— soit directement construire un automate à 8 états, chacun correspondant à une distribution des a et c dans les 3 dernières lettres lues (avec des transitions bien choisies évidemment).

(i) l'ensemble des mots de longueur paire ;

(j) l'ensemble des mots se terminant par *bc*.

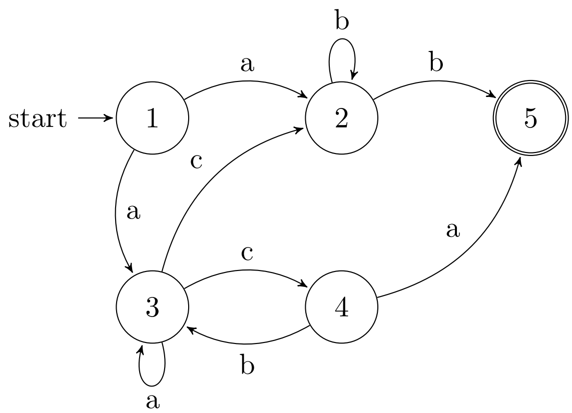
2. Proposer un automate et une expression rationnelle pour le langage de tous les mots de $\{a, b, c\}^*$ dont *cac* est un sous-mot [1].



Solution:

Expression rationnelle : (a|b|c)*c(a|b|c)*a(a|b|c)*c(a|b|c)*

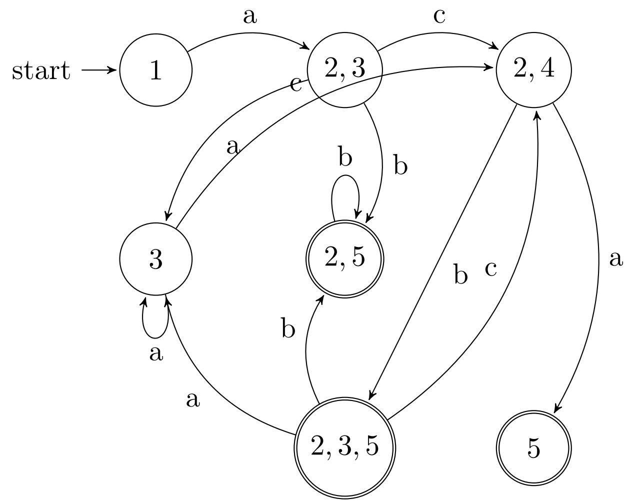
3. Déterminiser l'automate suivant en appliquant l'algorithme vu en cours [2] :



1. Un *sous-mot* de u est une sous-suite de lettres – non nécessairement contiguë – de u. À distinguer d'un facteur. Exemple : *pis* est un sous-mot de *produits*.
2. poly p.9

« Se convaincre » que l'automate déterminisé reconnaît le même langage en testant l'appartenance au langage d'un petit nombre de mots.

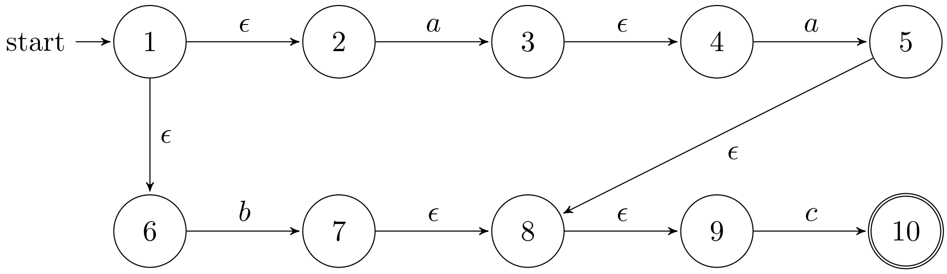
Solution:



4. Soit l'expression rationnelle $(aa|b)c$.

   (a) Utiliser la méthode vue en cours pour construire un automate reconnaissant le même langage[3].

   Solution:

   (b) Á partir de l'automate, proposer une grammaire régulière engendrant le même langage.

   Solution:

   $$\begin{aligned} S_1 &\rightarrow aS_3|bS_7 \\ S_3 &\rightarrow aS_5 \\ S_5 &\rightarrow cS_{10} \\ S_7 &\rightarrow cS_{10} \\ S_{10} &\rightarrow \epsilon \end{aligned}$$

   (N'oubliez pas que les règles singulières sont interdites dans une grammaire régulière.)

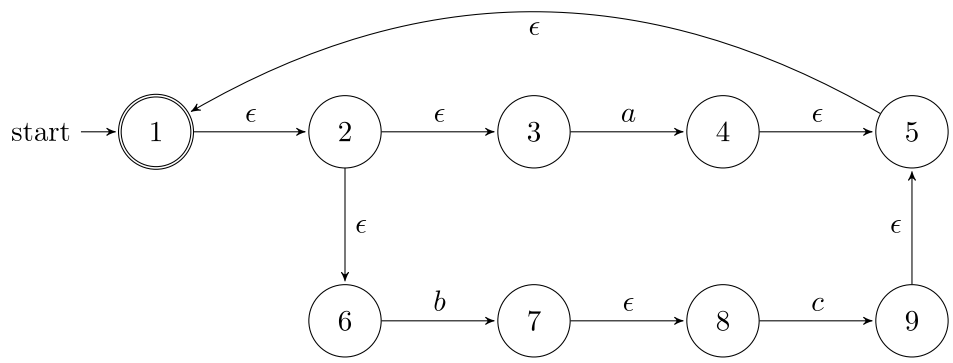
   (c) Donner un arbre syntaxique avec la grammaire précédente pour le mot *aac*.

5. Soit $A = \{a, b, c\}$, considérons l'expression rationnelle $(a|bc)^*$.

3. poly p.18

(a) Utiliser la méthode vue en cours pour construire un AF reconnaissant le même langage [4].

Solution:



(b) Éliminer les ϵ-transitions [5].

Solution:

On donne ϵ^+

$$\begin{array}{lll} 1 & \rightarrow & 2,\ 3,\ 6 \\ 2 & \rightarrow & 3, 6 \\ 4 & \rightarrow & 5,\ 1,\ 2,\ 3,\ 6 \\ 5 & \rightarrow & 1,\ 2,\ 3,\ 6 \\ 7 & \rightarrow & 8 \\ 9 & \rightarrow & 5,\ 1,\ 2,\ 3,\ 6 \end{array}$$

Solution:

δ'	a	b	c
1 (i/f)	4	7	
2	4	7	
3	4		
4 (f)	4	7	
5 (f)	4	7	
6		7	
7			9
8			9
9 (f)	4	7	

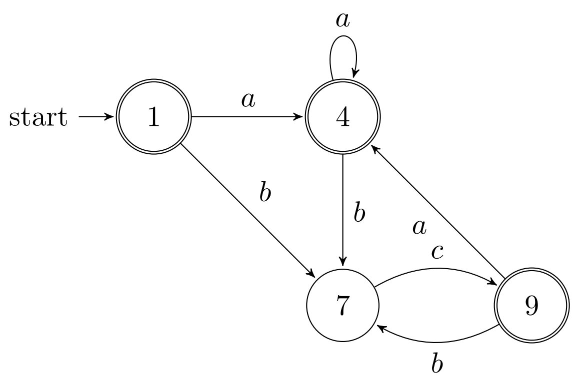
$(i) \rightarrow$ Etat initial

$(a) \rightarrow$ Etat final

On remarque que les états 2, 3, 5, 6 et 8 ne sont plus accessibles. Ils ne sont donc plus utiles et on peut les retirer.

Solution:

4. poly p.18
5. poly p.7

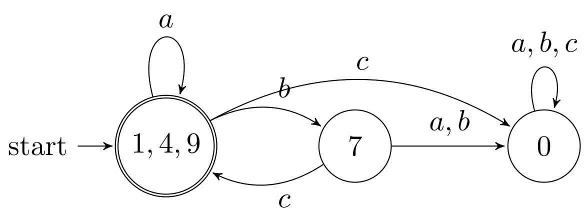


(c) Déterminiser l'automate résultant [6].

Solution: On a de la chance (mais ce n'est pas nécessairement le cas) : on peut vérifier facilement que l'automate est déjà déterministe.

(d) Compléter [7] puis minimiser [8] l'AFD.

Solution:



(e) Construire la grammaire régulière (à droite) [9] reconnaissant le même langage que l'AFDC.

Solution:

$$\begin{aligned} S_1 &\rightarrow aS_1|bS_7|cS_0|\epsilon \\ S_7 &\rightarrow cS_1|aS_0|bS_0 \\ S_0 &\rightarrow aS_0|bS_0|cS_0 \end{aligned}$$

(Il est cependant aisé de montrer qu'il n'existe aucun chemin de l'état 0 vers un état final, on peut donc supprimer cet état et utiliser l'algorithme pour produire une grammaire plus simple.)

6. Soit la grammaire suivante : $G_1 = (\{a,b\},\{S\},S,\{S \rightarrow aSbS|bSaS|\epsilon\})$. Quel est le langage L reconnu ? (Le prouver par double inclusion, c-à-d en montrant d'une part que tout mot reconnu appartient à L et d'autre part que tout mot de L est reconnu.)

Solution: Il s'agit de l'ensemble des mots contenant autant de a que de b. Ce langage n'est en fait pas régulier (on peut le montrer avec le lemme de pompage) !

6. poly p.9
7. poly p.6
8. poly p.10
9. poly p.17

Montrer que tous les mots contenant autant de a que de b sont bien générés est un peu technique mais très intéressant : on peut procéder par récurrence sur la taille des mots.

7. Soit L un langage rationnel, en utilisant des considérations sur les automates montrer que :
   (a) l'ensemble des préfixes de L, $L' = \{u | \exists v, uv \in L\}$ est aussi rationnel ;
   (b) l'ensemble des suffixes de L, $L'' = \{v | \exists u, uv \in L\}$ est aussi rationnel.

Solution: Soit L un langage régulier,
soit A un AF reconnaissant L,

— soit A' l'AF construit à partir de A où tous les états sont acceptants : on peut montrer facilement que A' reconnait L',
ainsi L' est régulier ;

— soit A'' l'AF construit à partir de A où tous les états sont initiaux : on peut montrer facilement que A'' reconnait L'',
ainsi L'' est régulier.