

1 Automates reconnaissant un langage donné.

Donner des automates reconnaissant les langages suivants: es entiers sont codés en binaire. Pour les entiers comme pour les mots habituel, on considère que les caractères sont lus de gauche à droite, donc en commençant par les bits de poids forts.

- des entiers pairs, des entiers impairs, des puissances de 2
 - entier pair: $q_0 \xrightarrow{0,1} q_0 \xrightarrow{0} q_1$, q_1 final
 - entier impairs: $q_0 \xrightarrow{0,1} q_0 \xrightarrow{1} q_1$, q_1 final
 - puissance de deux $q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_1$, q_1 final
- $A = \{0, 1\}$, $L = \{w \mid w \text{ code une puissance de } 4\}$
 - C'est 1 suivi d'un nombre pair de 0. Donc $q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{0} q_1$, q_1 final.
- $A = \{0, 1\}$, $L = \{w \mid w \text{ code la somme de deux puissances de } 4 \text{ distinctes: } 4^k + 4^{k'}, k \neq k'\}$.
 - C'est 1 suivi d'un nombre impair de 0 suivi de 1 suivi d'un nombre pair de zéro, on peut réutiliser l'automate précédent $q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_2 \xrightarrow{0} q_1$, $q_2 \xrightarrow{1} q_1$ l'état initial de l'automate précédent
- $A = \{a, b\}$, $L = \{w \mid w \text{ commence par } abaaba\}$
- $A = \{a, b\}$, $L = \{w \mid w \text{ contient } aabaaab\}$
- $A = \{a, b\}$, $L = \{w \mid w \text{ commence par } abb \text{ et termine par } bba\}$
- Les écritures de nombre à virgule
- $A = \{a, b, c\}$, $L = \{w \mid w \text{ contient au moins une fois chacune des trois lettres}\}$
- Optionel. $A = \{a, b\}$, $L = \{w \mid |w|_a \text{ est pair, ainsi que } |w|_b\}$
- Optionel $A = \{a, b\}$, $L = \{w \mid w \text{ contient un nombre pair de fois le facteur } bab\}$

- Optionel. $A = \{0, 1\}$, $L = \{w \mid w \text{ en base } 2, w \text{ représente un nombre valant } 1 \text{ modulo } 3\}$
 - $L = \{w \mid w \text{ commence par } abaaba\}$.
On note P la poubelle Facile :
 $(0,a,1) (1,b,2) (2,a,3) (3,a,4) (4,b,5) (5,a,6) (6,a,6) (6,b,6), (0,b,P) (1,a,P) (2,b,P) (3,b,P) (4,a,P) (5,b,P) (P,a,P) (P,b,P)$. 6 est final.
 - $L = \{w \mid w \text{ finit par } aabaaab\}$. Plus sioux, faut faire attention aux fleches "retour" $(0,a,1) (1,a,2) (2,b,3) (3,a,4) (4,a,5) (5,a,6) (6,b,7) (0,b,0) (1,b,0) (2,a,2) (3,b,0) (4,b,0) (5,b,3) (6,a,2) (7,a,4) (7,b,0)$. 7 est final.
 - commence par abb et termine par bba attention, on oublie le mot $abba$ qui appartient au langage
 - Nombre à virgule: On commence par écrire l'expression régulière vu la semaine dernière A partir de cette expression, on construit l'automate en composant il faut au moins un chiffre après la virgule
 - au moins une fois chacune des trois lettres a,b,c: 8 états qui sont les parties de $\{a, b, c\}$. si x dans p, alors (p,x,p) , sinon (p,x,q) avec q égal p union singleton x. L'automate est en cube. On raisonne sur les états, en les écrivant tous, ensuite on rajoute les transitions. Etat initial est l'ensemble vide, l'état final est les trois lettres.
 - Nombre pair de facteur u : faire deux occurrences de l'automate cherchant le facteur u. Dans la première version, j'ai lu le facteur un nombre pair de fois (sauf à l'extrémité de l'automate ou je viens de lire une occurrence de plus), dans la deuxième impair. Quand j'ai réussi à lire un u de plus, la parité change, on se dirige donc vers l'autre automate. Donc pour chacune de deux versions, remplacer la fleche qui conduit à l'état le plus à droite, par une fleche qui passe sur l'autre version, dans l'état ou on a déjà lu v, v étant le plus long préfixe de u qui est aussi suffixe de u, avec v différent de u.

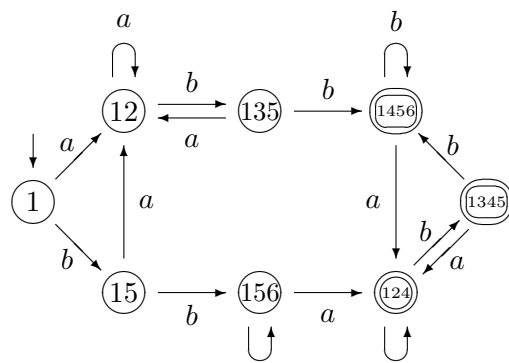
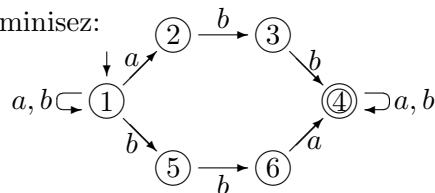
Pour $u = bab$, ca fait six etats (1, a, 1) (1, b, 2) (2, b, 2) (2, a, 3) (3, a, 1) (3, b, 2') (1', a, 1') (1', b, 2') (2', b, 2') (2', a, 3') (3', a, 1') (3', b, 2). Initial 1, finals 1 2 et 3

- Nombre pair de a et de b: Encore plus dur, l'automate à 4 etats: on se souviens si on a lu un nombre pair/im-pair de a et itou pour b (0, a, 1), (1, a, 0), (2, a, 3), (3, a, 2), (0, b, 2), (2, b, 0), (1, b, 3), (3, b, 1) final is 0 En utilisant Arden, on trouve : $(aa+bb+(ab+ba)(bb+aa)^*(ab+ba))^*$.
- multiples de trois : difficile, surtout l'expression. il faut commencer par l'automate. 3 etats. On est en i quand a lu un nombre qui vaut i modulo 3. Pas besoin de faire un cas spécial pour epsilon car 0 pas reconnu. Si j'ai lu un i mod 3 et que la prochaine lettre est j, je vais en $(2i+j)[3]$ (0, 0, 0), (0, 1, 1), (1, 1, 0), (1, 0, 2), (2, 0, 1), (2, 1, 2). L'état final est 1. Pour l'expression, On utilise Arden : $(0 + 1(01^*0)^*1)^*$

2 Déterminisation

2.1 Methode de déterminisation.

Déterminisez:



j'ai déjà fait un exemple en cours. Il faut faire une table avec une colonne pour les etats, une colonne pour le resultat des transitions par a, et une autre pour b.

2.2 Boum !

Soit L_n l'ensemble des mots sur $\{a, b\}$ de longueur au moins n dont la $n^{\text{ième}}$ lettre avant la fin est un b . Donnez un petit automate non-déterministe pour L_3 . puis son déterminisé. Comparez leur nombre d'états. Au lieu de faire marcher l'algorithme de déterminisation, on commencera par réfléchir quels doivent être les états, puis on rajoutera les transitions.

- Il y a $n+1$ états dans le non déterministe, et 2^n dans le déterministe Moralite: ca explose. Le non déterministe est facile, (1, a ou b, 1), (1, b, 2), (2, a ou b, 3) (3 a ou b, 4). Pour le déterminisé, on remarque que il faut stocker les trois dernières lettres lues dans l'état. Au début l'état initial est aaa, les état finaux sont ceux qui ont b a la bonne place;

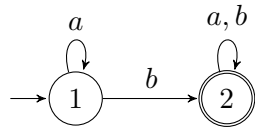
3 Resolution d'équations

Rappel de cours: a tout automate on peut associer un système d'équations dont les variables

représentent les langages reconnus par cet automate à partir de chacun de ses états.

3.1 Exemple introductif

À l'aide du système d'équations précédent, que l'on résoudra par élimination et utilisation du lemme d'Arden, déterminer une expression rationnelle correspondant aux automates suivants : (sur l'alphabet $\mathcal{A} = \{a, b\}$)



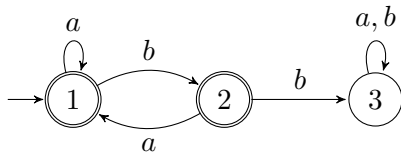
- Un exemple a déjà été traité en cours.
Rappel des règles :

- $R = \alpha R + \gamma$ a pour solution (unique si $\epsilon \notin \alpha$) $R = \alpha^* \gamma$;
- On ajoute aX s'il y a une transition a vers X
- on ajoute $+1$ aux états terminaux ;
- Il y a une inconnue par état qui représente le "Futur" de l'état, le langage reconnu en prenant cet état pour état initial.
- Le langage reconnu est donc l'inconnue associée à l'état initial.

$$\begin{cases} X_1 = aX_1 + bX_2 \\ X_2 = (a+b)X_2 + \epsilon \end{cases}$$

On a donc $X_2 = (a+b)^*$. D'où, en remplaçant X_2 dans X_1 :

$$X_1 = aX_1 + b(a+b)^* = a^*b(a+b)^*.$$



- Correspond à $\{w \in \mathcal{A}^* \mid w \text{ ne contient aucune occurrence de la chaîne } bb\}$

On obtient $a^*(ba^+)^*(\epsilon+b)$. Ou encore $(a+ba)^*(\epsilon+b)$.