Langages et Automates – Travaux dirigés – Feuille d'exercices n°2
AUTOMATES FINIS

Partie 1. Construction d'automates.

1- Un distributeur délivre des tasses de café au prix 0,25 euros. Il accepte des pièces de 5, 10 et 20 cents. On suppose que le distributeur ne rend pas la monnaie et qu'il accepte des sommes allant jusqu'à 0,40 euros. Construire l'automate fini qui modélise un tel distributeur.

2- Construire un automate qui détermine si un nombre entier codé en binaire est un multiple de 4

3- Un jeune pilote stagiaire de la compagnie « L3INFO Airlines », encore inexpérimenté, se voit confier le pilotage d'un appareil bimoteur. Heureusement, le fonctionnement de l'appareil est assez simple : l'avion réagit à quelques commandes comme décrit ci-dessous.

- Les moteurs 1 et 2 sont commandés indépendamment. La commande *on1* met en marche le moteur 1 et la commande *off1* l'arrête (même chose avec *on2* et *off2* pour le moteur 2). La commande *on1* (resp. *on2*) est sans effet[1] lorsque le moteur 1 (resp. le moteur 2) est déjà en marche ; de même, *off1* (resp. *off2*) est sans effet quand le moteur 1 (resp. le moteur 2) est à l'arrêt. Les moteurs peuvent être mis en marche l'un après l'autre dans n'importe quel ordre. Il est interdit d'arrêter un moteur si l'avion n'est pas immobile sur la piste (sauf exception pour le moteur 2, voir plus loin…).
- La commande *go* permet de faire rouler l'avion lorsqu'il est sur la piste et quand les deux moteurs sont en marche (si l'un des moteurs est arrêté, *go* est sans effet). Inversement, la commande *stop* permet d'immobiliser l'avion lorsqu'il roule sur la piste.
- Un décollage n'est autorisé que lorsque les deux moteurs sont en marche. La commande *dec* permet de faire décoller l'avion quand il roule sur la piste alors que la commande *att* permet de le faire atterrir s'il est en vol (en conséquence, il roule sur la piste). On peut redécoller après avoir atterri sans avoir à immobiliser l'avion.
- Exceptionnellement, on peut arrêter le moteur 2 pendant le vol. Il peut être remis en marche au moyen de la commande *on2*. Il est interdit d'arrêter le moteur 1 pendant le vol. En cas d'arrêt du moteur 2, l'atterrissage reste possible avec le moteur 1 seul.
- Initialement, l'appareil est immobile sur la piste et les deux moteurs sont arrêtés. Toute utilisation de l'avion conforme aux règles doit le ramener dans cet état.

En guise de manuel, la compagnie décide de fournir au jeune pilote un automate fini qui décrit les commandes et leurs effets conformément aux règles d'utilisation. Elle fait appel à vous pour construire cet automate.

a) Construire l'automate fini modélisant le fonctionnement du système. Le représenter sous forme de graphe. Identifier les états et préciser l'alphabet.
b) Votre automate est-il déterministe ? Justifier

4- Construire un automate fini qui reconnaît les nombres entiers multiples de 3 (rappel : un entier est multiple de 3 si la somme de ses chiffres est multiple de 3).

5- Soient $X = \{a, b\}$, $L = \{w \in X^* \ / \ |w|_a = 2n \ , n \in \mathbb{N}\}$ et $L' = \{w \in X^* \ / \ |w|_b = 2n + 1, n \in \mathbb{N}\}$

[1] Une commande sans effet est une commande possible (autorisée) mais qui ne change pas l'état du système.

a) Caractériser en français les langages L et L' .
b) Construire les automates qui reconnaissent respectivement L et L'.

6- On considère des nombres entiers non signés en base 2, en base 8 et en base 16. Chaque nombre est représenté par une suite de symboles pris dans l'alphabet X = {H,K,0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F} qui peut se terminer par K (respectivement par H) pour indiquer explicitement que le nombre est en base 8 (respectivement 16). La présence de K ou de H n'est pas obligatoire, les nombres ayant alors un type par défaut déduit des symboles qui le composent.

Exemples :

1A7F0 est un nombre en base 16 (type par défaut)	K562 n'est pas un nombre
1A7F0H est un nombre en base 16	1A7F0K n'est pas un nombre
07437 est un nombre en base 8 (type par défaut)	H n'est pas un nombre
07437K est un nombre en base 8	K n'est pas un nombre
07437H est un nombre en base 16	
100101 est un nombre en base 2 (type par défaut)	
100101H est un nombre en base 16	
0, 0K, 0H, 00, 00K… sont des nombres	

Construire un automate fini déterministe qui reconnait les nombres définis ci-dessus.

7- Construire un automate fini reconnaissant les constantes numériques signées ou non signées, entières ou avec une partie décimale… comme par exemple :

12   -12.34   +12E3   12.34E-5   -12.34 E+56

8- On considère les additions de bits suivantes :

0 + 0 = 0	représenté par 0 (000)
0 + 0 = 1	représenté par 1 (001)
0 + 1 = 0	représenté par 2 (010)
0 + 1 = 1	représenté par 3 (011)
1 + 0 = 0	représenté par 4 (100)
1 + 0 = 1	représenté par 5 (101)
1 + 1 = 0	représenté par 6 (110)
1 + 1 = 1	représenté par 7 (111)

Les additions de bits doivent tenir compte d'une retenue implicite (0 ou 1) : elles sont donc correctes ou non selon la donnée de cette retenue. Par ailleurs, les additions de bits produisent une retenue.

Par exemple :

- 0 + 1 = 1 est correcte si la retenue est 0 car 0 + 0 + 1 = 1 et produit une retenue égale à 0
- mais 0 + 1 = 1 est incorrecte si la retenue est 1 car 1 + 0 + 1 = 0 (et pas 1)
- 1 + 0 = 0 est correcte si la retenue est 1 car 1 + 1 + 0 = 0 et produit une retenue égale à 1
- mais 1 + 0 = 0 est incorrecte si la retenue est 0 car 0 + 1 + 0 = 1 (et pas 0)

On considère ici des suites d'additions de bits représentées par des mots sur l'alphabet X={0,1,2,3,4,5,6,7}. Une suite d'additions de bits est correcte si la première addition est correcte avec une retenue initiale donnée égale à 0, puis si les additions suivantes sont correctes en tenant compte de la retenue obtenue à l'étape précédente (la dernière retenue obtenue doit être égale à 0).

Exemples :

- les mots 35, 61, 061 et 641 représentent des suites d'additions de bits correctes,
- les mots 6, 07, 25, 63, 241 et 625 représentent des suites d'additions de bits incorrectes.

Précisions :

- 35 c'est d'abord 3 avec une retenue initiale égale à 0 c'est-à-dire 0 + 0 + 1 = 1 (correcte) qui produit une retenue égale à 0, puis c'est 5 (avec la retenue 0) c'est-à-dire 0 + 1 + 0 = 1 (correcte) qui produit une retenue égale à 0
- 641 c'est d'abord 6 avec une retenue initiale égale à 0 c'est-à-dire 0 + 1 + 1 = 0 (correcte) qui produit une retenue égale à 1, puis c'est 4 (avec la retenue 1) c'est-à-dire 1 + 1 + 0 = 0 (correcte) qui produit une retenue égale à 1, et pour finir c'est 1 (avec la retenue 1) c'est-à-dire 1 + 0 + 0 = 1 (correcte) qui produit une retenue égale à 0
- mais 25 c'est d'abord 2 avec une retenue initiale égale à 0 c'est-à-dire 0 + 0 + 1 = 0 (incorrecte)

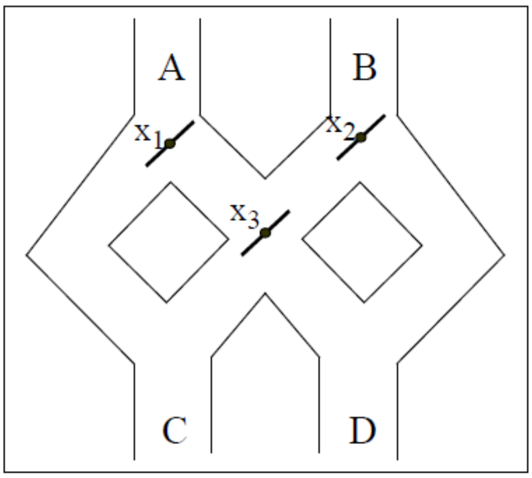
Construire un automate fini qui reconnait les suites d'additions de bits correctes. Expliquer.

9- Soit le jeu dont les règles sont les suivantes (cf. figure) :

- On peut faire tomber des billes successivement dans l'entrée A ou dans l'entrée B.
- Les volets x1, x2, et x3 ont 2 positions possibles et laissent passer la bille à droite ou à gauche (selon la position). Chaque fois qu'une bille franchit un volet, elle fait changer le volet de position, donc la bille suivante prendra un chemin différent.
- Initialement, les volets sont positionnés comme indiqué dans la figure.

Le jeu consiste à faire tomber une série de billes dans l'un ou l'autre des entrées. Une séquence de jeu est gagnante si la dernière bille de la série sort en D.

Modéliser ce jouet à l'aide d'un automate fini (a priori, il est plus simple de construire un automate non déterministe).

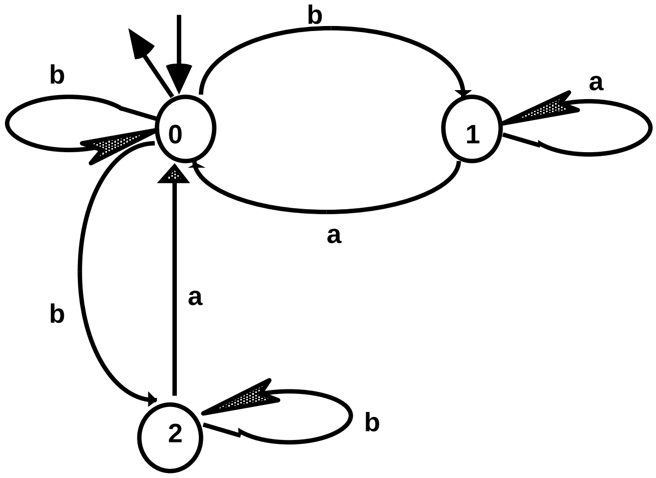


<u>Remarque</u> : pour simplifier le problème, on peut modéliser le même jouet mais en supposant que le volet x2 reste bloqué dans la position initiale

Partie 2. Opérations sur les automates.

Soit l'alphabet X = {a,b}.

1- Déterminisation et Complémentaire. Soit l'automate M :



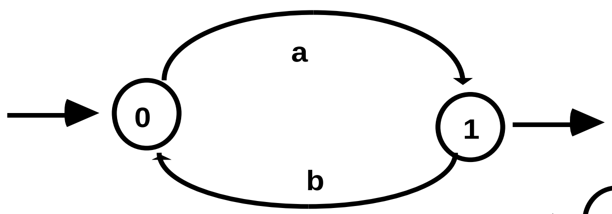
a) Donner le système d'équations de M. Donner une grammaire linéaire à droite qui engendre L(M).
b) Déterminiser l'automate M.
c) Caractériser le langage L(M) reconnu par M.
d) Donner une autre grammaire linéaire à droite qui engendre L(M).
e) Construire l'automate qui reconnaît X*- L(M), complémentaire de L(M).

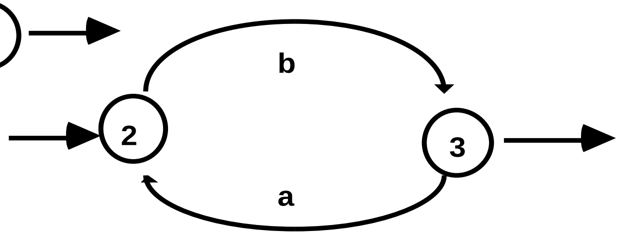
2- Union.

a- Soient $L = \{w \in X^* \ / \ |w|_a = 2n \ , \ n \in \mathbb{N}\}$ et $L' = \{w \in X^* \ / \ |w|_b = 2n + 1, \ n \in \mathbb{N}\}$. Construire l'automate qui reconnaît $L \cup L'$.

b- Soient deux automates quelconques A1 = < X1, Q1, F1, q1, t1 > et A2 = < X2, Q2, F2, q2, t2 >. Définir formellement l'automate U qui reconnait $L(A1) \cup L(A2)$.

3- Produit. Soient les automates M1 et M2 :





Construire l'automate qui reconnaît le langage L(M1) **.** L(M2) .

4- Intersection.

Soit le langage L engendré par la grammaire <N = {S,T,U}, X = {a,b}, $\mathcal{P}$, S>

avec $\mathcal{P}$ = { $S \to a\,S \mid b\,T \mid \lambda$;
$T \to b\,T \mid a\,U \mid \lambda$;
$U \to a\,U \mid \lambda$ }

a) Donner l'automate M3 tel que L(M3) = L.
b) Construire l'automate qui reconnaît $L(M1) \cap L(M3)$.
Rappel : $\forall a \in X,\ a.(L1 \cap L2) = a.L1 \cap a.L2$

Partie 3. Opérations constructives sur les automates.

1- Soient les langages sur X = {a,b}: $J = X^*a\,a\,X^*$ $K = a\,X^*$ $Y = X^* - J$ $Z = X^* - K$

a) Caractériser en français les propriétés respectives des mots des langages J, K, Y et Z.
b) Donner une grammaire pour J.
c) Construire des automates qui reconnaissent respectivement J, K , Y et Z . En déduire une grammaire linéaire à droite pour J.
d) A partir de ceux-ci, déduire les automates qui reconnaissent $L = J + K$ et $T = Y \cap Z$. Comparer L et T et caractériser les mots de T.
e) la concaténation est-elle une loi : - interne dans Y ?
- interne dans T ?

2- Construire un automate qui reconnaît les mots de $(a+b)^*$ ne contenant pas la chaîne aba.

3- Soit l'automate fini $A = < X=\{a,b\}, Q=\{q_0,q_1\}, q_0, F=\{q_1\}, t_A >$ avec $t_A(q_0,a) = \{q_1\}$,
$t_A(q_0,b) = \emptyset$,
$t_A(q_1,a) = \{q_1\}$,
$t_A(q_1,b) = \{q_1\}$

Soit l'automate fini B dérivé de la grammaire linéaire à droite suivante :

$< X=\{a,b\}, N=\{N_2,N_3\}, S=N_2, P=\{ N_2 \to a\,N_2 \mid b\,N_2 \mid b\,N_3 ;\ N_3 \to \lambda \} >$

a) Caractériser en français les langages L(A) et L(B).
b) Par le calcul sur les systèmes d'équations de langages, construire un automate déterministe qui reconnait le langage $L(A) \bullet L(B)$. Tracer le graphe.
c) Par le calcul sur les systèmes d'équations de langages de A et de B, construire un automate qui reconnait $L(A) \cap L(B)$. Tracer le graphe.
d) Comparer les automates obtenus en b et en c, et expliquer le résultat.

Soit le langage $K = L(A) \bullet L(B) \cup \{\lambda\}$.

e) Par le calcul sur les systèmes d'équations de langages, montrer que $K \bullet K = K$.
f) En déduire $K^* = K$.

Partie 4. Déterminisme et ambiguïté.

1- Montrer que la grammaire linéaire à droite correspondant à un automate fini déterministe est non ambiguë.
2- Trouver un langage régulier L et un automate fini non déterministe M tel que L(M) = L, dont la grammaire linéaire à droite correspondant à M est non ambiguë.

Partie 5. Automate normalisé.

Définition : Un automate déterministe $A = \langle X, Q, F, q_0, t \rangle$ est normalisé si $\forall x \in X, \forall q \in Q, q0 \notin t(q,x)$

a) Soit $A = \langle X, Q, F, q_0, t \rangle$ un automate déterministe normalisé reconnaissant le langage L.
Soit $B = \langle X, Q, F \cup \{q_0\}, q_0, \delta \rangle$ l'automate défini par :

$$\forall q \in Q - F \text{ et } \forall x \in X, \qquad \delta(q,x) = t(q,x)$$
$$\forall q \in F \text{ et } \forall x \in X, \qquad \delta(q,x) = t(q,x) \cup t(q_0,x)$$

Prouver que l'automate B reconnait L* c'est-à-dire L(B) = L(A)*.

b) Trouver un automate non normalisé A tel que l'automate B construit comme à la question précédente ne reconnaisse pas L(A)*.

c) Montrer que, pour tout automate, il existe un automate déterministe normalisé équivalent.