TD2. Automates

I. Automates et langages

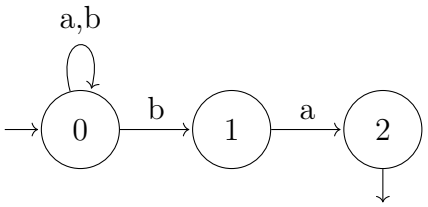
Exercice 1. Vocabulaire sur les automates

1. Représenter l'automate A sur l'alphabet $\{a, b, c\}$ d'états $0, 1, 2, 3$, d'état initial 0, d'état terminal 3 et ayant pour transitions
$0 \xrightarrow{a} 0; 0 \xrightarrow{a} 1; 0 \xrightarrow{b} 0; 0 \xrightarrow{c} 0; 1 \xrightarrow{a} 2; 1 \xrightarrow{b} 2; 1 \xrightarrow{c} 2; 2 \xrightarrow{a} 3; 2 \xrightarrow{b} 3; 2 \xrightarrow{c} 3$.
2. Cet automate est-il déterministe ? Est-il complet ? Justifier.
3. Les mots *baba* et *cabcb* sont-ils reconnus par A ?
4. Décrire $L(A)$ en langage ordinaire.

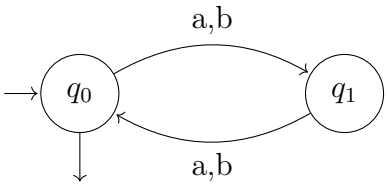
Exercice 2. Langages reconnus

Décrire le langage reconnu par chacun des automates suivants sur l'alphabet $\Sigma = \{a, b\}$.

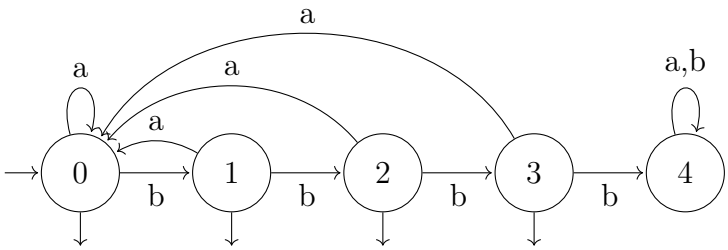
1.



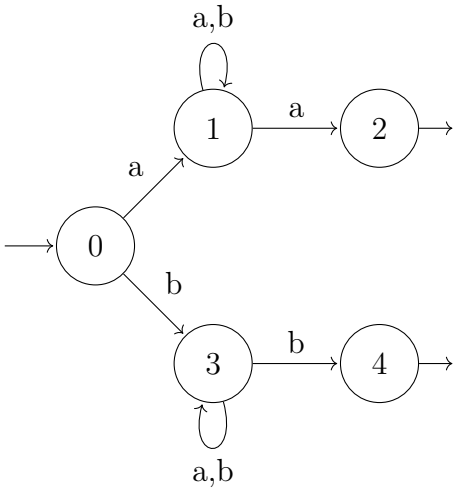
2.



3.



4.

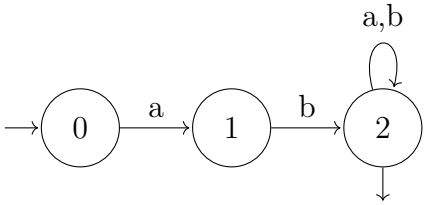


Exercice 3. Construction d'automates
Soit $\Sigma = \{a, b, c\}$. Donner des automates finis **déterministes** et **complets** reconnaissant les langages suivants :

1. L'ensemble des mots terminant par b.
2. L'ensemble des mots non vides ne se terminant pas par b.
3. L'ensemble des mots contenant au moins un b.
4. L'ensemble des mots contenant au plus un b.
5. L'ensemble des mots contenant exactement trois b.
6. L'ensemble des mots contenant un nombre pair de a.
7. L'ensemble des mots où le nombre d'occurrences de b est divisible par 3.
8. L'ensemble des mots contenant au moins un a et dont la première occurrence de a n'est pas suivie par un c.

II. Transformations en automates équivalents

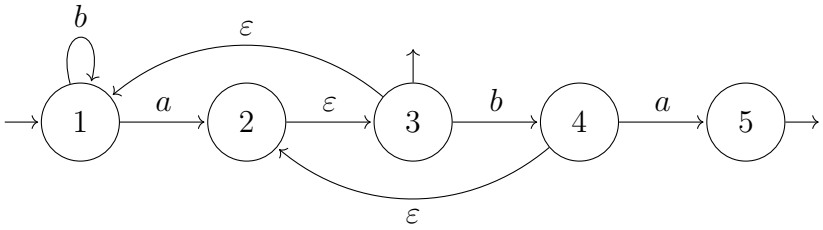
Exercice 4. Complétion d'automates
1. Expliquer pourquoi l'automate suivant n'est pas complet.



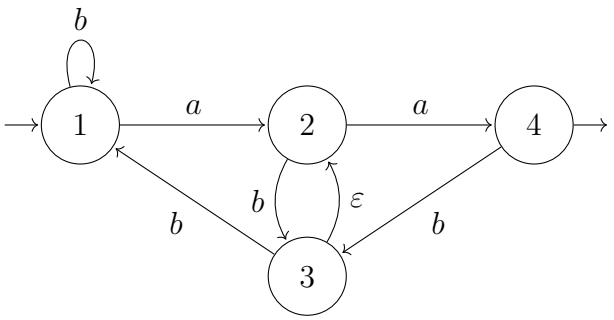
2. Quel langage reconnaît-il ?
3. Donner un automate complet équivalent.

Exercice 5. Déterminisations
1. Parmi les automates de l'exercice 2, lesquels sont déterministes ? Déterminiser ceux qui ne le sont pas.
2. Soit $\Sigma = \{a, b\}$. Construire un automate non déterministe sans ε-transition qui reconnaisse $\{w = w_1...w_n \in \Sigma^* |\ w_{n-2} = a\}$.
3. Déterminiser l'automate obtenu à la question précédente.

Exercice 6. Suppression des ε-transitions
Donner des automates sans ε-transitions équivalents aux l'automate suivant :
1.



2.



3. Déterminiser les automates obtenus à la question précédente.

Exercice 7. Déterminisation de taille exponentielle
Dans cet exercice, on appellera taille d'un automate son nombre d'états.
1. Soit A un automate fini non déterministe de taille n, et A_D l'automate déterminisé associé, obtenu grâce à l'algorithme vu en cours. Donner une borne supérieure de la taille de A_D en fonction de n.
2. Donner un exemple de deux automates A_1 et A_2 tels que :

- A_1 et A_2 sont équivalents
- A_1 est déterministe, et A_2 ne l'est pas
- En appliquant l'algorithme de déterminisation à A_2, on obtient un automate A_3 ayant strictement plus d'états que A_1.

La question précédente peut laisser penser que l'explosion du nombre d'états lors de la déterminisation est due à l'algorithme utilisé. Dans le reste de cet exercice on démontre que ce n'est pas le cas. Plus précisément on démontre que pour tout $n \in \mathbb{N}$, il existe un automate A_n ayant $n+1$ états qu'on ne peut déterminiser avec moins de 2^n états, c'est-à-dire tel que tout automate déterministe équivalent a au moins 2^n états.
3. On note L_n le langage des mots sur $\Sigma = \{a, b\}$ de longueur supérieurs à n, et dont la n ième lettre en partant de la fin est un a. Construire un automate non déterministe A_n ayant au plus $n+1$ états tel que $L(A_n) = L_n$.
4. Représenter A_3 et le déterminiser.
Dans la suite, on suppose donné un automate complet déterministe D_n reconnaissant L_n. On note alors q_0 son unique état initial et δ^* sa fonction de transition étendue. **5.** Soient deux mots distincts u et v de Σ^n. Montrer que $\delta^*(q_0, u) \neq \delta^*(q_0, v)$.
6. En déduire un minorant sur le nombre d'états de D_n.
7. Conclure.

III. Théorème de Kleene

Exercice 8. Des expressions régulières aux automates
Pour chacune des expressions régulières suivantes, donner un automate reconnaissant le langage associé en utilisant les constructions de Thompson, puis supprimer les ϵ-transitions :

1. $(a|b)^*$
2. $a^*(b|c)a$

Exercice 9. Des automates aux expressions régulières
Pour chacun des automates de l'exercice 2, donner une expression régulière associée, en utilisant l'algorithme de McNaughton et Yamada.

Exercice 10. Langage préfixe, suffixe et facteur
Si L est un langage sur un alphabet Σ, on note $pref(L)$ l'ensemble des préfixes des mots de L, $suf(L)$ l'ensemble des suffixes, $fact(L)$ l'ensemble des facteurs.
Montrer que si L est régulier, il en est de même de ces trois langages.

Exercice 11. Langage "racine"
Si L est un langage sur un alphabet Σ, on note $\sqrt{L} = \{u \in \Sigma^* | u.u \in L\}$.
Montrer que si L est régulier alors $\sqrt{L}$ l'est aussi.

IV. Automates et calculs

Exercice 12. Calcul du modulo
Dans cet exercice on s'intéresse à la définition d'un automate permettant de tester si deux nombres, représentés par leur écriture en base 2, sont congrus modulo 3.
1. Donner un automate sur l'alphabet $\Sigma = \{0\}$ acceptant le langage des $\{0^p | 2^p \equiv 1[3]\}$, à savoir l'ensemble des nombres $n \in \mathbb{N}$, écrits en unaire, tels que $2^n \equiv 1[3]$.
2. Grâce à la question précédente, donner un automate sur l'alphabet $\Sigma = \{0, 1\}$ acceptant le langage des $\{b_0 b_1 ... b_{n-1} | b_{n-1} b_{n-2} ... b_0 \equiv 1[3]\}$. **Attention** : les bits de poids faibles sont donnés en premier.
3. Définir un automate sur l'alphabet $\Sigma = \{0, 1\}^2$ reconnaissant le langage suivant :

$$\{(b_0, c_0)(b_1, c_1)...(b_{n-1}, c_{n-1}) | b_{n-1} b_{n-2} ... b_0 \equiv c_{n-1} c_{n-2} ... c_0[3]\}$$

4. Grâce à la question précédente, donner un automate sur l'alphabet $\Sigma = \{0, 1\}$ acceptant le langage des $\{b_n b_{n-1} ... b_0 | b_{n-1} b_{n-2} ... b_0 \equiv 1[3]\}$. **Attention** : les bits de poids forts sont donnés en premier.

Exercice 13. Addition binaire
Dans cet exercice on s'intéresse au problème de la vérification, par un automate, de la somme de deux nombres écrits en binaire : l'automate reçoit en entrée trois mots u, v et w représentant trois entiers p, q et r et doit décider si $p + q = r$.
On suppose que les trois nombres font la même taille.

On se donne pour alphabet $\Sigma = \{0, 1\}$. Étant donné un mot $w = w_0 w_1 ... w_n$, on note w2 = ni=0 wi 2i , la valeur de w en base 2. On souhaite construire dans cette section un automate sur l'alphabet $\Sigma \times \Sigma \times \Sigma$ reconnaissant le langage suivant :

$$L = \{(u_0, v_0, w_0)(u_1, v_1, w_1)...(u_{n-1}, v_{n-1}, w_{n-1}) | u_0 u_1 ... u_{n-1} + v_0 v_1 ... v_{n-1} = w_0 w_1 ... w_{n-1}\}$$

1. Proposer un automate déterministe à 2 états reconnaissant L.
2. Énoncer une propriété invariante sur l'automate proposé. C'est une propriété de la forme : "la lecture d'un mot $(u_0, v_0, w_0)(u_1, v_1, w_1)...(u_{n-1}, v_{n-1}, w_{n-1})$ conduit à un état q si et seulement si ... ".
3. Prouver l'invariant proposé. En déduire la correction de l'automate proposé.