Aix*Marseille université

Domaine Sciences et Technologies
Portail 1 René Descartes

Introduction à l'informatique : TD 10

Code UE : S14IN1I1

Année 2021–2022

Automates

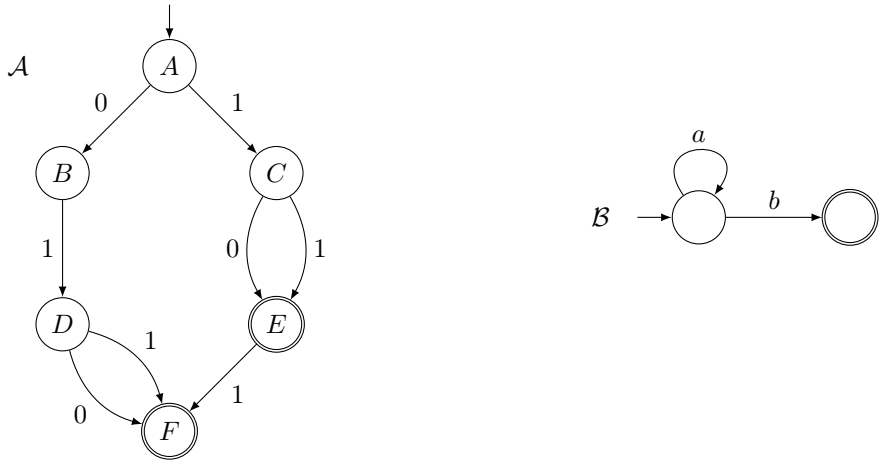
Un automate permet de décrire des ensembles de séquences sur un alphabet A donné :

— si l'alphabet est $A = \{0, 1\}$, l'automate décrit un ensemble de séquences de 0 et 1 : un automate peut ainsi accepter des codes binaires d'entiers ;

— si l'alphabet est $A = \{a, b, c, \ldots, y, z\}$, l'automate décrit un ensemble de mots sur l'alphabet latin : un automate peut ainsi accepter l'ensemble des mots du dictionnaire français ;

— si l'alphabet est $A = \{oui, non\}$, l'automate accepte des séquences de décisions à certaines questions : un automate permet alors de représenter une stratégie au jeu du *Qui est-ce ?* dans un arbre de décision.

Un automate est composé d'*états* et de *transitions* étiquetées par des lettres de l'alphabet A. Il possède également un état *initial* et des états *acceptants*. On représente ci-dessous deux automates : celui de gauche, $\mathcal{A}$, sur l'alphabet $\{0, 1\}$; celui de droite, $\mathcal{B}$, sur l'alphabet $\{a, b\}$.



Considérons l'automate $\mathcal{A}$ à gauche. Ses états sont A, B, C, D, E et F. L'état initial est A, distingué par une flèche entrante. Les états acceptants sont E et F distingués par un double cercle (cela diffère de la couleur verte que nous avons utilisé pendant le cours !). Cet automate possède 8 transitions : par exemple, il y en a une de l'état A vers l'état B étiquetée par la lettre 0, et une autre de l'état E à l'état F étiquetée par la lettre 1.

Notons que les deux automates vérifient la propriété suivante :

> « Pour tout état E de l'automate et pour toute lettre ℓ de l'alphabet, il existe au plus une transition sortant de l'état E étiquetée par la lettre ℓ. »

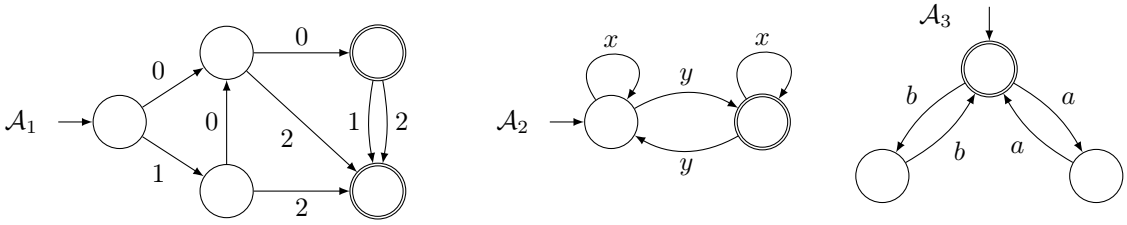
On dit que l'automate est *déterministe* en cela qu'à tout moment il n'y a aucun choix pour continuer la lecture d'une séquence de lettres fixée à l'avance : soit on bloque car il n'y a pas de transition étiquetée par la lettre voulue (par exemple, si on veut lire deux lettres 0 dans l'automate du dessus, on bloque en B qui ne peut pas lire une lettre 0 en suivant une transition), soit on suit l'unique transition étiquetée par la lettre voulue.

Un automate accepte un ensemble de séquences. Une séquence s est *acceptée* par l'automate s'il existe un chemin (et s'il existe, il est unique) de l'état initial à un état acceptant dont la séquence des étiquettes des transitions visitées (dans l'ordre) est s. Ainsi, la séquence 011 est acceptée par l'automate $\mathcal{A}$ de gauche ci-dessus, mais pas la séquence 01 qui ne termine pas dans un état acceptant, ni la séquence 110 pour laquelle il n'y a pas de chemin avec cette étiquette (on bloque en E lorsqu'il s'agit de lire la lettre 0). L'ensemble des séquences acceptées par l'automate $\mathcal{A}$ est $\{010, 011, 10, 11, 101, 111\}$: ce sont l'ensemble des codages binaires sur moins de 3 bits représentant des entiers premiers (2, 3, 5, 7).

Pour l'automate $\mathcal{B}$, à droite ci-dessus, l'ensemble des séquences acceptées sont de la forme $aa\cdots aab$, c'est-à-dire une séquence de a avec un unique b à la fin. Dans les dessins, il n'est pas

toujours nécessaire de donner des noms aux états, comme dans l'exemple de l'automate $\mathcal{B}$. Mais on peut évidemment donner des noms si on le souhaite, pour aider à comprendre la signification de l'état : ici, l'état de gauche de $\mathcal{B}$ pourrait s'appeler « que des a » et l'état de droite « b à la fin », ou simplement « a » et « b » si on veut des noms plus courts...

Exercice 1 Pour chacun des trois automates $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$ ci-dessous, décrire l'alphabet ainsi que l'ensemble des séquences acceptées.



Exercice 2 On se place, dans cet exercice, sur l'alphabet $\{0, 1\}$.

1. Dessiner un automate qui accepte l'ensemble des séquences possédant un nombre de 1 multiple de 3 : ainsi la séquence 0011010 doit être acceptée, mais pas la séquence 1101011.
2. Dessiner un automate qui accepte l'ensemble des séquences de longueur au plus 4 qui possède autant de 0 que de 1 (dans n'importe quel ordre). Essayer d'obtenir un automate avec un nombre minimal d'états, en modifiant éventuellement votre première tentative.
3. À partir de l'automate obtenu à la question précédente, en déduire un automate qui accepte l'ensemble des séquences de longueur au plus 6 qui possède autant de 0 que de 1.

Exercice 3 L'objectif de l'exercice est de modéliser le comportement des portes automatiques des grands bus de la RTM par le biais d'un automate. On peut demander à ouvrir la porte grâce à un bouton. La porte est aussi équipée d'un capteur de proximité qui permet d'empêcher la fermeture de la porte si une personne est proche. Le principe est que la porte ne doit pas s'ouvrir si un individu passe simplement devant mais s'il l'indique explicitement en appuyant sur le bouton. Le bouton est soit dans l'état *relâché* si personne ne le touche, soit dans l'état *appuyé* si quelqu'un est en train d'appuyer sur le bouton. Le système possède 4 états possibles :

— Etat 0 : personne n'est à proximité de la porte. La porte est fermée.
— Etat 1 : un individu est à proximité de la porte (le capteur de proximité le détecte) et la porte est fermée.
— Etat 2 : un individu a appuyé sur le bouton. La porte est ouverte et le capteur de proximité détecte la personne.
— Etat 3 : un individu est à proximité de la porte (le capteur de proximité le détecte), il n'a pas la main sur le bouton mais la porte est ouverte. Ce cas survient si l'individu appuyait auparavant sur le bouton.

Les événements qui permettent la transition entre états sont :

— **p**résence : le capteur de proximité détecte un individu à proximité mais celui-ci ne touche encore pas le bouton.
— **a**bsence : le capteur de proximité n'a détecté aucun individu à proximité depuis au moins 2 secondes. On suppose dans ce cas que personne ne peut appuyer sur le bouton.
— **t**ouche : l'individu vient d'appuyer sur le bouton alors qu'il était déjà dans le champ du capteur de proximité mais qu'il n'avait pas encore touché le bouton.
— **r**elâche : le bouton vient d'être relâché mais l'individu est toujours dans le champ du capteur.

Définir un automate permettant de modéliser le comportement de la porte automatique tel qu'il est décrit ci-dessus : l'automate doit accepter toutes les séquences d'évènements valides, si bien que tous les états sont acceptants. Quelles sont les transitions de votre automate où le système doit donner l'ordre au vérin de la porte d'ouvrir ou de fermer la porte ?

Exercice 4 On se place, dans cet exercice, sur l'alphabet $\{a, b, c\}$.

1. Dessiner un automate qui accepte l'ensemble des séquences qui se terminent par *cca* : ainsi, le mot *bacbcca* doit être accepté, mais pas le mot *bccba*, ni le mot *bccab*.
2. Dessiner un automate qui accepte l'ensemble des séquences qui se terminent par *abba* : attention, le mot *abbabba* doit être accepté !

Exercice 5 L'alphabet morse donne un code pour les 26 lettres de l'alphabet composé d'impulsions courtes (•) et longues (▬) :

A •▬	B ▬•••	C ▬•▬•	D ▬••	E •	F ••▬•
G ▬▬•	H ••••	I ••	J •▬▬▬	K ▬•▬	L •▬••
M ▬▬	N ▬•	O ▬▬▬	P •▬▬•	Q ▬▬•▬	R •▬•
S •••	T ▬	U ••▬	V •••▬	W •▬▬	X ▬••▬
Y ▬•▬▬	Z ▬▬••				

Trouver un automate, ayant le plus petit nombre d'états possibles, qui accepte exactement l'ensemble des codes morse.