Chapitre 3

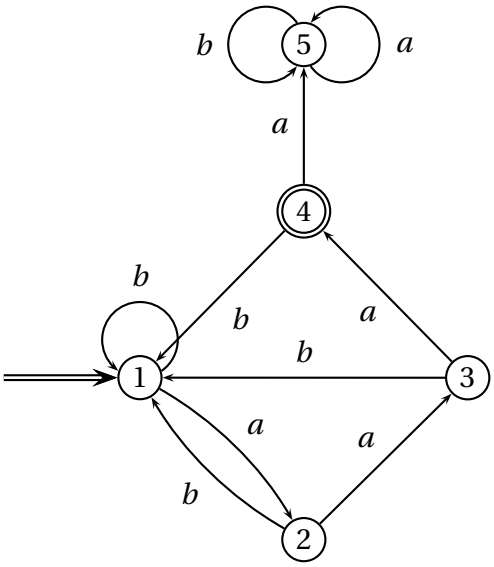
Graphes étiquetés

Sommaire

3.1 Quelques exemples

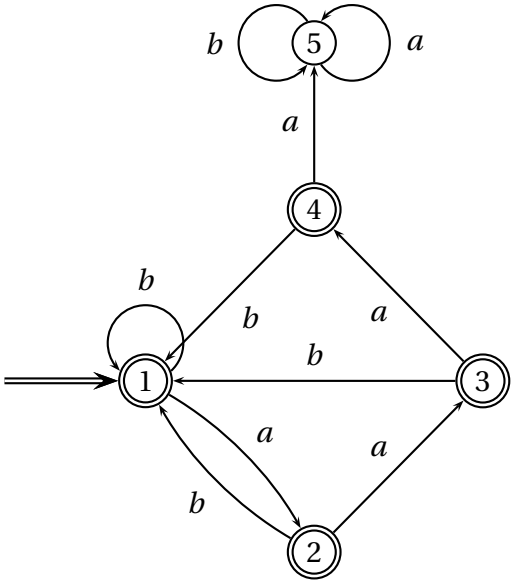
3.1.1 Le jeu du labyrinthe

Nous allons commencer par un jeu : on a représenté ci-dessous le plan d'un petit labyrinthe. Ce labyrinthe possède 5 salles, numérotées de 1 à 5 ; au départ, on est dans la salle 1, indiquée par une flèche. Les salles qui ouvrent sur l'extérieur sont entourées par un double rond ; ici il n'y en a qu'une, c'est la salle 4. De chaque salle partent des couloirs à sens unique, portant une lettre (a ou b), et allant à une autre salle (ou parfois revenant à la même salle, comme dans le cas de la salle 5).



Au début du jeu, on vous remet une suite de lettres. En lisant ces lettres l'une après l'autre, on suit un chemin partant de la salle 1 dans le labyrinthe. Si, après avoir lu toute la suite, on est dans une salle qui ouvre sur l'extérieur, on a gagné, sinon, on a perdu.

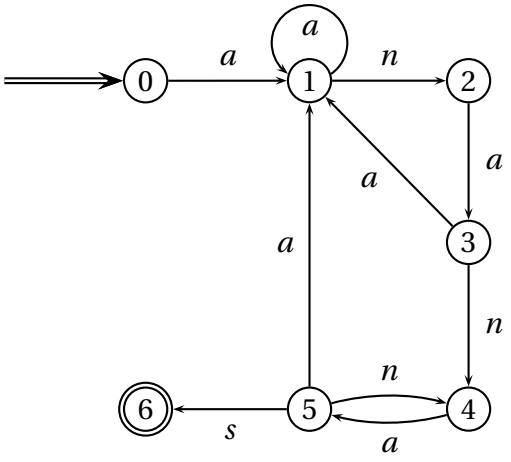
1. Le mot *abaab* est-il gagnant ? Et le mot *abaaa* ?
2. Caractériser les mots gagnants.
3. Caractériser les mots gagnants pour le labyrinthe suivant, petite modification du premier, où toutes les salles, sauf la salle 5, ouvrent sur l'extérieur.



Cet exercice peut paraître élémentaire. On a pourtant réalisé quelque chose de non évident : une machine qui sait reconnaître certains mots (car il n'est pas difficile de réaliser un équivalent électronique de ce labyrinthe), et on va voir que cela a de nombreuses applications.
Bien entendu, on va cesser de parler de salles et de labyrinthes. On reconnaît dans le plan de ce labyrinthe un graphe orienté, où chaque arête est munie d'un nom appelé *étiquette*.

3.1.2 Un digicode

Prenons comme deuxième exemple celui d'un digicode comme ceux permettant l'ouverture de la porte de nombreux bureaux ou appartements. Considérons un digicode constitué d'un clavier à 26 touches (les 26 lettres) qui constitue l'organe d'entrée d'un automate dont la sortie commande l'ouverture d'une porte. La porte s'ouvre dès que l'on a tapé la bonne suite de lettres et cela (au moins pour le digicode considéré ici) même si l'on a commencé par se tromper. Si le mot permettant l'ouverture de la porte est « *ananas* », le système peut être modélisé par le graphe étiqueté représenté par la figure ci-dessous.



Dans ce chapitre, les sommets du graphe seront appelés, pour suivre la terminologie habituelle en la matière, des *états* et les arcs étiquetés des *transitions*. Pour ne pas alourdir le schéma ci-dessus nous n'avons pas représenté les transitions provoquées par la frappe des autres lettres.

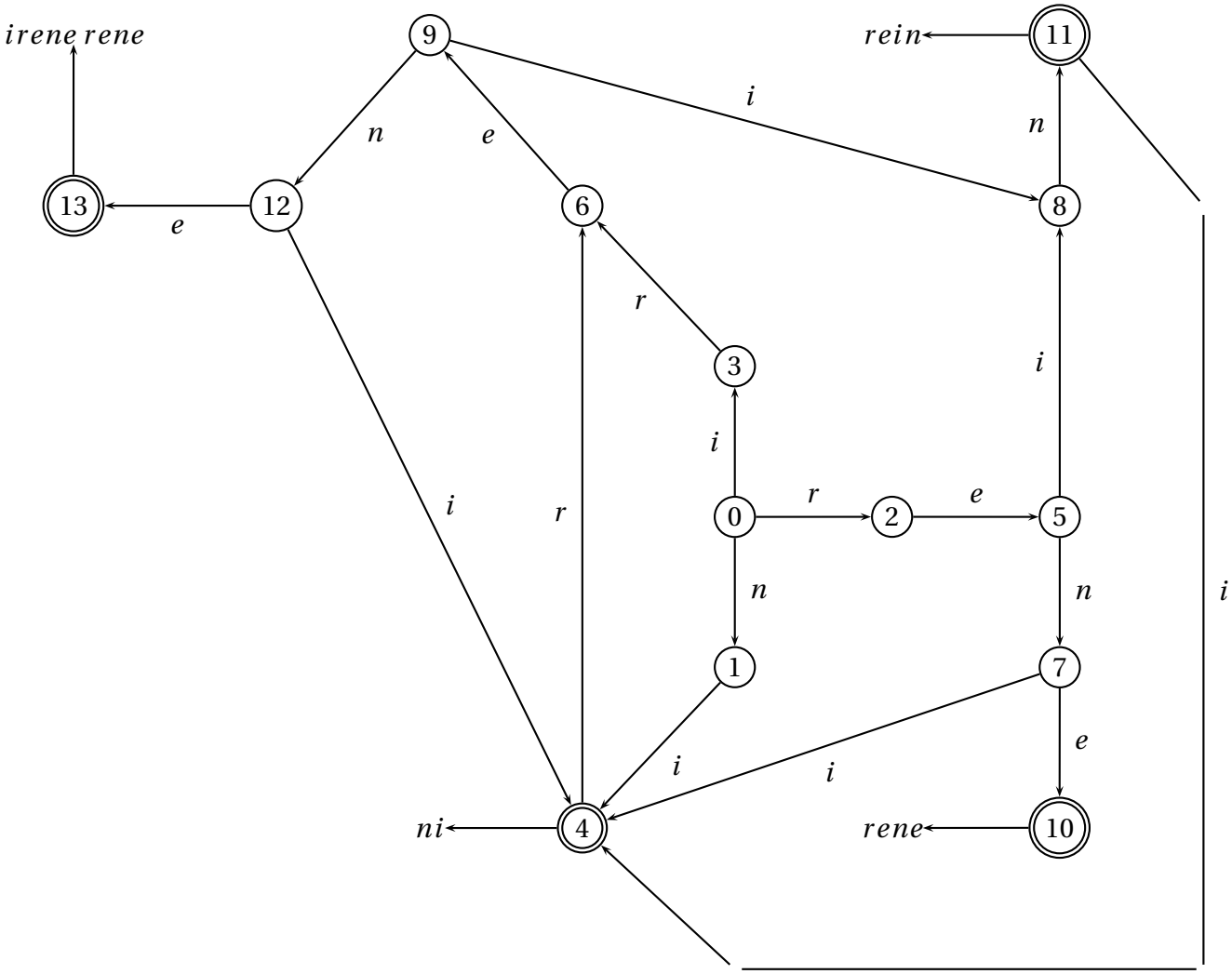
Chaque état (y compris l'état 0) est relié en fait à l'état 0 par autant d'arcs étiquetés par toutes les autres lettres. Par exemple, on n'a pas représenté les 25 arcs qui bouclent sur l'état 0 et qui sont étiquetés par toutes les lettres différentes de « a ». Il est facile de constater que dès que la suite de lettres a, n, a, n, a, s est frappée, on atteint l'état 6 (par convention un état final est entouré par deux cercles concentriques) qui déclenche l'ouverture.

3.1.3 Reconnaissance de modèles

Le problème de la reconnaissance de modèles (*pattern matching*) consiste à rechercher les occurrences de certaines séquences de caractères, que l'on appellera mots-clés, dans un texte t.
L'application la plus typique est la recherche documentaire. Étant donné un fond documentaire comprenant des références bibliographiques accompagnées de résumés indiquant les thèmes de chaque ouvrage, on désire chercher les références dans lesquelles certains mots-clés apparaissent.
Mais il existe bien d'autres domaines d'applications, comme celle du génome par exemple : on recherche des séquences d'acides aminés dans une très longue suite. L'algorithme trivial qui consiste à parcourir tout le texte pour rechercher les occurrences du premier mot-clé, puis à le parcourir à nouveau pour rechercher le second, etc. est inefficace, voire impraticable si le texte comporte plusieurs millions de caractères et que l'on recherche plusieurs mots. Le graphe étiqueté ci-dessous a été construit par l'algorithme du à AHO et CORASICK (1 975), il permet de rechercher plusieurs mots en ne parcourant le texte qu'une seule fois (donc en n'effectuant que n comparaisons si n est la longueur du texte t) et ceci quel que soit le nombre de mots recherchés ! Supposons que les mots clés soient : « ni », « rein », « rene »et « irene ». À partir de cet ensemble de mots recherchés, l'algorithme de AHO et CORASICK fournit le graphe étiqueté de la figure 3.1 page suivante.
Pour des raisons de lisibilité nous n'avons pas représenté toutes les transitions. L'état 0 est l'état initial et partant de l'état 0, bouclant sur ce même état il y a autant de transitions étiquetées par les caractères autres que « n », « r », « i ». Partant des autres états, sauf arc explicite, les transitions étiquetées par « n », « r », « i »sont reliées respectivement aux états 1, 2 et 3 et les transitions étiquetées par les caractères autres que « n », « r », « i »sont toutes reliées à l'état 0.
L'utilisation de ce graphe étiqueté pour le problème de reconnaissance de modèles est très simple : on explore le texte t, en partant de l'état 0 et en suivant les transitions. À chaque étape, si on est dans l'état i, on examine le caractère c du texte t sur lequel on est. S'il existe une transition partant de l'état i, étiquetée c, conduisant à un état j, on passe à l'état j ; dans le cas contraire on passe dans l'état 0. Dans les deux cas on avance d'un caractère dans le texte t. Chaque fois qu'on passe par un état dont l'ensemble des mots clés associés n'est pas vide, on sait que l'on vient de déceler tous les mots-clés qui se trouvent dans cet ensemble. Sur le schéma, ces ensembles de mots sont reliés à l'état correspondant.

Remarque. De tels graphes étiquetés sont utilisés de manière intensive sur tous les traitements de texte. Ces programmes possèdent tous une fonction « recherche et remplacement » qui construit un graphe étiqueté pour rechercher un mot (ou un type de mots) dans un texte. Ces fonctions de recherche permettent en général de chercher des expressions plus compliquées qu'un simple mot. C'est souvent utilisé dans la recherche de fichiers : si l'on veut chercher tous les fichiers dont le nom commence par « lettre » et se termine par « .txt », on peut demander les fichiers dont le nom est du type « lettre*.txt », où * est un « joker » qui remplace n'importe quel mot.
Les fichiers « lettre1.txt » ou « lettrebis.txt » sont de ce type.

FIGURE 3.1: Graphe étiqueté construit par l'algorithme du à AHO et CORASICK



3.2 Récapitulation : définitions et résultats

Pour parler de graphe étiqueté, il faut d'abord choisir un *alphabet*, dans lequel seront choisies les étiquettes du graphe. On notera A cet alphabet ; dans tous les exercices, l'alphabet sera petit (habituellement 2 ou 3 lettres, exceptionnellement l'alphabet usuel, mais dans ce cas on ne représentera bien sûr pas toutes les arêtes).

Définition 3.1. On appelle *graphe étiqueté* un graphe orienté où toutes les arêtes portent une étiquette choisie dans l'alphabet A, et qui possède un sommet initial (indiqué en général par une flèche pointant vers ce sommet) et un ou plusieurs sommets finaux (indiqués en général par un double rond entourant le sommet).
On ne considérera que des graphes étiquetés déterministes, c'est-à-dire tels que de chaque sommet parte une seule arête portant une étiquette donnée.

L'intérêt d'un graphe étiqueté est de reconnaître des mots sur l'alphabet A ; précisons un peu :

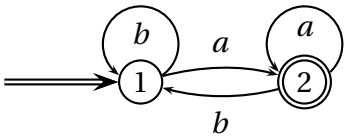
Définition 3.2. On appelle *mot* sur l'alphabet A une suite finie de lettres de A.

Définition 3.3. Soit G un graphe étiqueté par l'alphabet A. On dit qu'un mot sur l'alphabet A est *reconnu* par le graphe G s'il existe un chemin orienté sur le graphe G, partant du sommet initial, arrivant à un sommet final, et étiqueté par ce mot.

Définition 3.4. On appelle *langage associé* à un graphe étiqueté l'ensemble des mots reconnus par ce graphe étiqueté.

Le contenu de ce chapitre se borne à faire comprendre ces notions de mot reconnu par un graphe étiqueté et de langage associé à un graphe étiqueté, et à les faire fonctionner dans des cas simples.

Exemple 3.1. Voici un graphe étiqueté très simple :

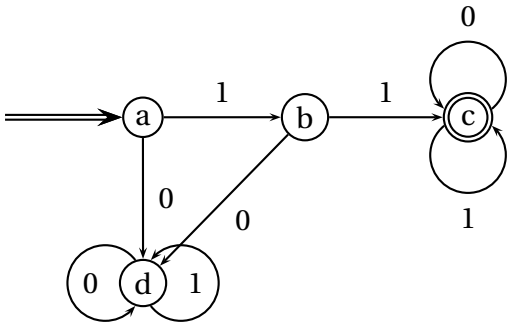


Ce graphe étiqueté reconnaît les mots sur l'alphabet $A = \{a\,;\,b\}$ (c'est-à-dire les mots formés à partir des lettres a et b) se terminant par la lettre a. Le langage L associé à cet automate est l'ensemble des suites finies de a ou de b se terminant par a.

Remarque. Aucun théorème n'est au programme. Il ne faudrait pas s'imaginer que c'est parce qu'il n'y en a pas ! Les graphes étiquetés, ou automates, ont donné lieu depuis une cinquantaine d'années à une théorie mathématique abstraite, riche et diversifiée, possédant de nombreuses applications. Ils sont en particulier fondamentaux en informatique, et c'est pourquoi on a jugé utile de les faire connaître aux élèves de terminale.

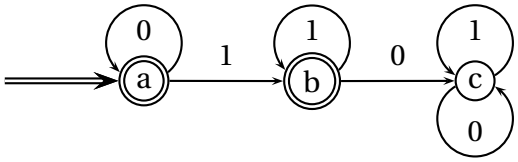
3.3 Exercices

EXERCICE 3.1.
Soit l'automate ci-contre.



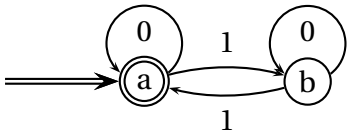
1. Les mots « 11 », « 101 », « 110 », « 1011 » sont-ils reconnus par cet automate ?
2. Donner la liste des mots de quatre lettres reconnus par celui-ci.
3. Caractériser les mots reconnus.

EXERCICE 3.2.
Soit l'automate ci-contre.



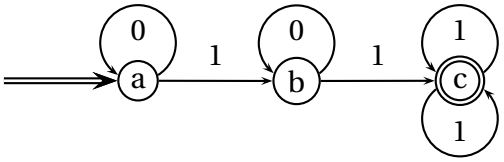
1. Les mots « 000 », « 001 », « 010 », « 0011 » sont-ils reconnus par cet automate ?
2. Donner la liste des mots de moins de quatre lettres reconnus par celui-ci.
3. Caractériser les mots reconnus.

EXERCICE 3.3.
Soit l'automate ci-contre.



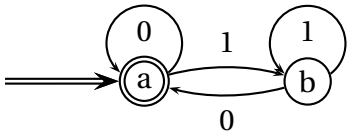
1. Les mots « 11 », « 101 », « 110 », « 1011 » sont-ils reconnus par cet automate ?
2. Donner la liste des mots de trois lettres reconnus par celui-ci.
3. Caractériser les mots reconnus.

EXERCICE 3.4.
Soit l'automate ci-contre.



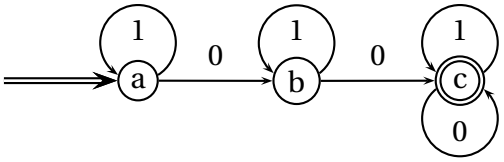
1. Les mots « 11 », « 101 », « 110 », « 1011 » sont-ils reconnus par cet automate ?
2. Donner la liste des mots de trois lettres reconnus par celui-ci.
3. Caractériser les mots reconnus.

EXERCICE 3.5.
Soit l'automate ci-contre.



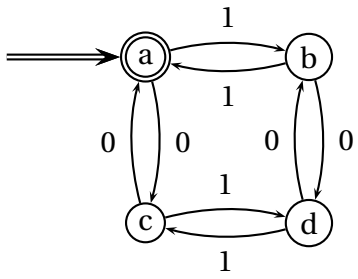
1. Les mots « 11 », « 010 », « 110 », « 101010 » sont-ils reconnus par cet automate ?
2. Donner la liste des mots de quatre lettres reconnus par celui-ci.
3. Caractériser les mots reconnus.

EXERCICE 3.6.
Soit l'automate ci-contre.



1. Les mots « 11 », « 101 », « 110 », « 1011 » sont-ils reconnus par cet automate ?
2. Donner la liste des mots de trois lettres reconnus par celui-ci.
3. Caractériser les mots reconnus.

EXERCICE 3.7.
Soit l'automate ci-contre.



1. Les mots « 11 », « 101 », « 1010 », « 1001 » sont-ils reconnus par cet automate ?
2. Donner la liste des mots de trois lettres reconnus par celui-ci.
3. Caractériser les mots reconnus.

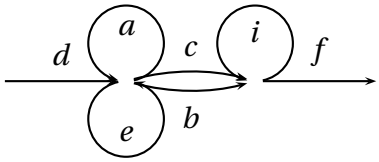
EXERCICE 3.8.
Donner la suite des états visités par l'automate de la reconnaissance de modèles page 20 si les mots sont recherchés dans le texte t suivant : « annie n'honnit ni irene ni nina »

EXERCICE 3.9.
Représenter les automates suivants :

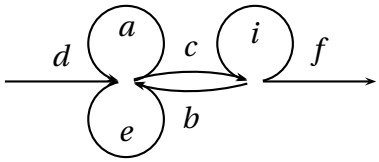
1. reconnait les mots ne comportant que des 0 et des 1, et dont le nombre de 1 est impair ;
2. reconnait les mots ne comportant que des 0 et des 1, et dont le nombre de 0 est pair ;
3. reconnait les mots ne comportant que des 0 et des 1, et comportant la chaine 00 ;
4. reconnait les mots ne comportant que des 0 et des 1, et finissant par la chaine 11.

EXERCICE 3.10.
Un réseau informatique doit être accessible à un grand nombre de personnes, qui ne doivent cependant pas avoir le même code d'accès. Cet accès est régi par un des graphes étiquetés ci-dessous ; un mot est accepté comme code d'accès (ou reconnu) si c'est une liste de lettres commençant par d et terminant par f, associée à une chaîne de ce graphe.

Graphe 1

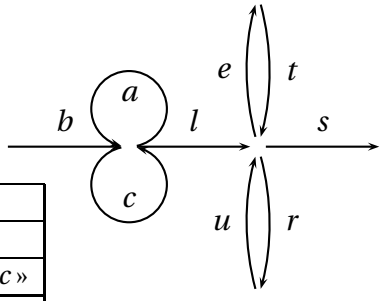


Graphe 2



1. Les mots « $decif$ »et « $daaeebiif$ »sont-ils des mots reconnus par les graphes étiquetés ci-dessus ?
2. Donner, pour chaque graphe ci-contre, la liste des mots de 5 lettres reconnus.
3. Caractériser pour chaque graphe les mots reconnus.
4. Caractériser les mots reconnus par les deux graphes ci-dessus.

EXERCICE 3.11.
Le graphe étiqueté ci-contre permet de reconnaître des mots (un mot est une suite finie de lettres, n'ayant pas forcément un sens).



VRAI	FAUX	
		il reconnaît le mot « $baccalets$»
		il reconnaît tous les mots commençant par « bac »
		il reconnaît le mot « $bleus$»
		il reconnaît exactement six mots de cinq lettres