AUTOMATES FINIS NON DÉTERMINISTES

Dans ce chapitre, on décrit mathématiquement des machines très simples, qu'on appelle automates finis non déterministes. Ce sont des automates finis qui ont la possibilité de « choisir » les transitions : il peut y avoir plusieurs transitions étiquetées par la même lettre depuis un même état.

Comme toujours, A est un alphabet fini non vide.

1 Généralités

1.1 Définitions et vocabulaire

Définition. On appelle automate fini non déterministe sur l'alphabet A tout quadruplet $\mathscr{A} = (E, I, T, \delta)$ tel que

- E est un ensemble fini, appelé ensemble des états de l'automate $\mathscr{A}$;
- I est un sous-ensemble de E, dont les éléments sont appelés états initiaux ;
- T est un sous-ensemble de E, dont les éléments sont appelés états terminaux ;
- δ, appelé fonction de transition, est une application de $E \times A$ dans $\mathscr{P}(E)$:
  lorsque $\delta(p, a)$ est non vide, si $q \in \delta(p, a)$, on dit que (p, a, q) est une transition de l'automate et que a est l'étiquette de la transition.

Lorsque (p, a, q) est une transition d'étiquette a, on note souvent $p \xrightarrow{a} q$

Cette fois-ci, la fonction de transition peut être représentée par un tableau à doubles entrées (les états, les lettres) dont chaque case contient la liste des extrémités possibles.

par exemple, si $A = \{a, b, c\}$, et $E = \{1, 2, 3, 4\}$, le tableau

	a	b	c
1	$2,4$	3	1
2	1	$1,2,3$	
3		3	4
4	$4,1$	4	

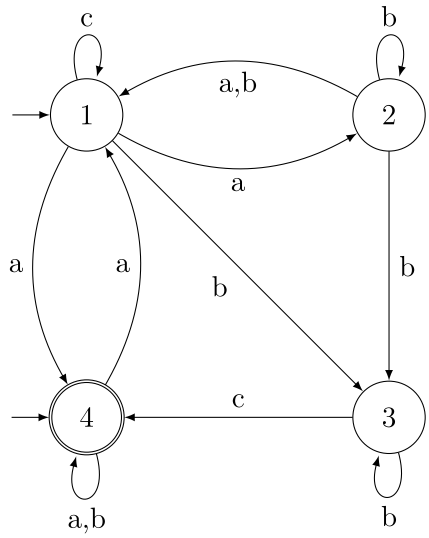
définit une fonction de transition.

Remarque. Plutôt que de définir les transitions par une fonction de transition δ, on peut aussi les définir en vrac comme les éléments d'une partie de $E \times A \times E$, ce qu'on peut faire aussi avec les automates déterministes. Avec cette définition, un automate déterministe est un cas particulier d'automate non déterministe, dans lequel pour tout état p et lettre a, il existe au plus un seul état q tel que (p, a, q) est une transition.

1.2 Représentation graphique

Quand le nombre d'états et de lettres n'est pas trop élevé, on peut représenter un automate par un graphe orienté valué : les sommets sont les états et les lettres sont les valeurs des arcs. Ce graphe n'est pas forcément simple (il peut y avoir des boucles et des arcs multiples), les arcs multiples sont représentés une seule fois avec une étiquette constitué de plusieurs lettres séparées par des virgules.

L'exemple ci-dessous représente l'automate $\mathscr{A} = (E, \{1, 4\}, \{4\}, \delta)$ précédent :



1.3 Langage reconnu par un automate non déterministe

Définition. Soit $\mathscr{A} = (E, i, T, \delta)$ un automate non déterministe. On définit la fonction de transition étendue δ^*, définie sur des mots de A^* :
si u est un mot et p un état, on définit inductivement $\delta^*(p, u)$:

- si $u = \varepsilon$, on pose $\delta^*(p, \varepsilon) = \{p\}$;
- si $u = av$ (a une lettre et v un mot), on pose $\delta^*(p, u) = \bigcup\limits_{q \in \delta(p,a)} \delta^*(q, v)$;

En notant $u = a_1 a_2 \dots a_n$ et $q \in \delta^*(p, u)$, alors en notant $p_0 = p$, il existe n états $(p_1, \dots, p_n)$ tels que pour tout $k \in [\![0, n-1]\!]$, $p_{k+1} \in \delta(p_k, a_k)$ et $q = p_n$:
autrement dit, il existe une succession de transitions

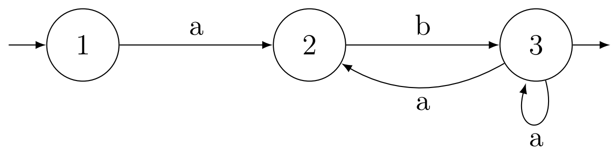
$$p = p_0 \xrightarrow{a_1} p_1 \xrightarrow{a_2} p_2 \xrightarrow{a_3} \dots \xrightarrow{a_n} p_n = q$$

dont q_n est l'état final.
On note alors $p \xrightarrow{u} q$ et on dit que la suite des transitions est un calcul d'étiquette u dans l'automate.

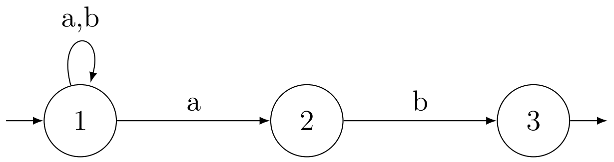
Définition. Soit $\mathscr{A} = (E, I, T, \delta)$ un automate non déterministe et u un mot.
On dit que le mot est reconnu (ou accepté) par l'automate $\mathscr{A}$ si il existe un état initial i tel que $\delta^*(i, u) \cap T \neq \varnothing$ (les états terminaux sont aussi appelés états acceptants), autrement dit s'il existe une suite de transitions débutant par un état initial, se terminant par un état terminal et étiqueté par u. Une suite de transitions associée est appelé un calcul réussi dans l'automate.
On appelle langage reconnu par l'automate $\mathscr{A}$ l'ensemble des mots reconnus par $\mathscr{A}$, noté en général $L(\mathscr{A})$.

Le gros avantage des automates non déterministes est qu'ils sont plus faciles à définir que les déterministes, car ils nécessitent moins d'états et de transitions.

Exemples

a) L'automate 

reconnaît les mots qui ont pour préfixe ab et ne contenant pas deux b consécutifs

b) L'automate 

reconnaît les mots ayant pour suffixe ab.

Deux automates qui reconnaissant le même langage sont dits équivalents.

1.4 Langages reconnaissables par un automate non déterministe

Tout l'intérêt des automates non déterministes réside dans le théorème suivant : contrairement à ce qu'on pourrait penser, les automates non déterministes n'apportent rien de plus en terme de langages reconnus.

Théorème 1 *Les langages reconnaissables par automates non déterministes sont les mêmes que ceux reconnus par automates déterministes.*
Autrement dit, si un langage est reconnu par un automate non déterministe, alors il est reconnu par un automate déterministe.

Même si les automates non déterministes sont plus simples, ils ne prêtent absolument pas à une traduction algorithmique du fait de leur non déterminisme. Pour effectuer concrètement des calculs, on a besoin quand même d'automates déterministes.

2 Déterminisation d'un automate non déterministe

Le théorème précédent permet de construire un automate déterministe équivalent à un automate non déterministe donné, mais celui-ci est souvent encombré d'états non accessibles. On peut réduire le nombre d'états accessibles en étant un peu plus subtil : on obtient l'algorithme des parties.

On considère un automate non déterministe $\mathscr{A}$.

> On pose I l'ensemble des états initiaux de $\mathscr{A}$
> O l'ensemble $\{I\}$
> F un ensemble vide d'ensembles d'états de $\mathscr{A}$
> T un ensemble vide de transitions entre éléments de F
> Tant que O est non vide,
>> on choisit un élément e dans O (e est un ensemble d'états de $\mathscr{A}$)
>> on retire e de O et on l'ajoute à F
>> pour chaque lettre a de A
>>> on pose f l'ensemble des états de $\mathscr{A}$ accessibles en une transition étiquetée par a depuis un quelconque état de e
>>> on ajoute dans T la transition (e, a, f)
>>> si f n'est pas déjà dans F ou dans O, on l'ajoute à O
>
> fin tant que
> F est l'ensemble des états d'un automate déterministe $\mathscr{A}'$, dont
>> les transitions sont les éléments de T,
>> l'état initial est I
>> les états finaux sont les éléments de F qui contiennent au moins un état terminal

Par rapport au théorème brut précédent, l'automate ainsi construit a souvent moins d'états et moins d'états non co-accessibles : en l'émondant, on peut obtenir un automate déterministe pas trop gros en général.