Automates Finis

L2 Informatique - UFR S.A.T

Pr. Ousmane THIARE

ousmane.thiare@ugb.edu.sn
http://www.ousmanethiare.com

10 mai 2024

Automates Finis

Automates Finis
Introduction

On va dégager dans ce paragraphe la notion de machine comme modèle conceptuel pour la description de dispositifs informatiques aussi variés qu'un ordinateur entier, un logiciel ou un compilateur.

Définition

Une machine est un dispositif doté d'un certain nombre d'états, susceptible d'évoluer d'un état à un autre en fonction de divers paramètres, comme le temps (la machine est alors dotée d'une machine interne).
Elle est de plus apte à communiquer avec l'extérieur : elle peut accepter des données en provenance de l'extérieur (des entrées) ou communiquer des résultats à l'extérieur (des sorties).

Remarque. À chaque instant, la condition interne de la machine, y compris la mémoire, constitue son état.
C'est le type le plus simple de machine :

Définition

Un mécanisme est totalement imperméable au monde extérieur, il n'accepte aucune entrée ni aucune sortie. C'est une machine à nombre fini d'états, dont le comportement est gouverné uniquement par le temps, mesuré par une horloge interne.

Un mécanisme peut être entièrement décrit par un couple (E, t), où E est un ensemble fini d'états et $t : E \longrightarrow E$ est une fonction de transition des états.

Propriété

Un mécanisme entre nécessairement dans une boucle infinie (on dit : un cycle).

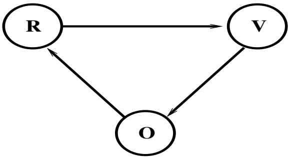
En effet, le nombre d'états est fini.

Propriété

S'il existe un état $e \in E$ tel que t(e)=e, cet état est appelé état-repos .

Remarque. Un mécanisme qui entre dans un tel état n'en sort évidemment plus.

Exemple. Un feu de circulation routière peut être décrit par un mécanisme à trois états : V , O et R, donc $E = \{V, O, R\}$.
La fonction de transition des états est telle que t(V)=O, t(O)=R et t(R)=V. On peut représenter ce mécanisme par le graphe de transition des états



ou par la matrice booléenne T représentant t :

$$t=\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Définition

On appelle automate fini à com- portement déterminé (AFD) tout triplet (E, I, t), où

- *E est un ensemble fini (l'ensemble des états),*

Pour $i \in I$, *on définit la fonction* $t_i : E \longrightarrow E$ *par* $t_i(e) = t(i, e)$.

Définition

On appelle automate fini à com- portement déterminé (AFD) tout triplet (E, I, t), où

- *E est un ensemble fini (l'ensemble des états),*
- *I est le vocabulaire de l'automate : c'est l'ensemble fini des symboles admis en entrée,*

Pour $i \in I$, on définit la fonction $t_i : E \longrightarrow E$ par $t_i(e) = t(i, e)$.

Définition

On appelle automate fini à com- portement déterminé (AFD) tout triplet (E, I, t), où

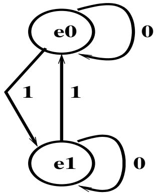
- *E est un ensemble fini (l'ensemble des états),*
- *I est le vocabulaire de l'automate : c'est l'ensemble fini des symboles admis en entrée,*
- $t : E \times I \longrightarrow E$ *est la fonction de transition d'états : si l'automate se trouve dans l'état* $e \in E$*, il réagit à l'entrée* $i \in I$ *en passant à l'état t(e, i).*

Pour $i \in I$*, on définit la fonction* $t_i : E \longrightarrow E$ *par* $t_i(e) = t(i, e)$.

Exemple. Soit $E = \{e_0, e_1\}$, $I = \{0, 1\}$ et t telle que

- l'entrée 0 laisse inchangé chacun des états,
- l'entrée 1 échange les états.

Un tel dispositif, en électronique, est appelé un T-flip-flop , il est abondamment utilisé dans les ordinateurs...



La table qui donne les valeurs de la fonction t est appelée table de transition d'états de l'automate considéré :

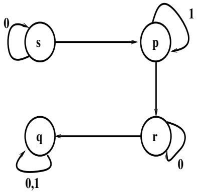
t	0	1
e_0	e_0	e_1
e_1	e_1	e_0

Définition

Une machine de Moore est un sextuplet $M = (E, I, t, e_0, V, g)$ tel que (E,I,t) est un AFD, et

- *$e_0 \in E$ est un état appelé état initial, dans lequel se trouve la machine au départ de chaque exécution.*
- *V est un ensemble fini, dit ensemble des sorties,*
- *$g : t\langle E, I\rangle \Longrightarrow V$, où $t\langle E, I\rangle$ est l'image de t, est la fonction de sortie telle que, chaque fois que la machine entre dans l'état e, elle produise la sortie $g(e) \in V$.*

Exemple. Ici, $E = \{e_0, e_1, e_2, e_3\}$, $I = \{0, 1\}$, $V = \{a, b, c\}$, t est donnée, soit par le graphe de l'automate :



soit par la table de transition d'états

t	0	1
e_0	e_1	e_3
e_1	e_2	e_2
e_2	e_3	e_2
e_3	e_3	e_2

g se lit dans le graphe : $g(e_1) = a, g(e_2) = c, g(e_3) = b$.
Remarque. Une telle machine est aussi appelée traducteur (elle « traduit » l'entrée 0001001 en une sortie acbcbbc).

Définition

On obtient une machine de Mealy lorsque la sortie est déterminée, non pas par l'état atteint, mais par la transition d'états.
C'est donc un sextuplet (E, I, t, e_0, V, h) *où la fonction de sortie h est une application de* $E \times I$ *vers V*

Remarque. Il est clair que, pour une machine de Moore (E, I, t, e_0, V, g), on peut définir une machine de Mealy équivalente (c'est-à-dire qui produit la même sortie sur toute séquence d'entrée), en posant h(e, i)=g(t(e, i)), soit $h = g \circ t$.
Réciproquement, en introduisant au besoin des états supplémentaires, on montre qu'on peut rem- placer toute machine de Mealy par une machine de Moore équivalente.

Définition

Une machine de Moore telle que l'ensemble V des sorties est réduit à la paire booléenne $\{FAUX, VRAI\}$ *ou* $\{0, 1\}$*,*

- *tout état qui donne lieu à FAUX est appelé état de rejet,*
- *tout état qui donne lieu à la sortie VRAI est appelé état d'acceptation .*

est appelée automate de Moore ou machine d'acceptation.

Remarque. Inutile d'exhiber ici la fonction de sortie, il suffit de se donner l'ensemble A des états d'acceptation, sous-ensemble de E.
Un automate de Moore est donc défini par le quintuplet (E, I, t, e_0, A).

Sur le graphe représentant un automate de Moore, on représentera un état d'acceptation en l'entourant d'un double cercle.
Les autres états (simplement cerclés) sont les états de rejet.

Soit M un automate de Moore.
L'ensemble des entrées I peut être considéré comme l'alphabet d'un système formel.
L'ensemble des « mots » construits avec cet alphabet (suite d'éléments de l'alphabet) qui conduisent la machine à un état d'acceptation peut être considéré comme l'ensemble des formules bien formées de ce système formel.

Définition

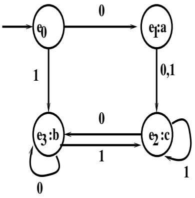
Ce système formel constitue le langage associé à l'automate M.

Notation : $\mathcal{L}(M)$

Réciproquement, étant donné un langage L, on peut éventuellement construire un automate de Moore M tel que le langage associé à M soit $\mathcal{L}$: $\mathcal{L} = \mathcal{L}(M)$.

Remarque. Cela n'est pas possible pour tous les langages. Quand c'est possible, cet automate analyse les mots du langage.

Exemple. Construction de l'automate qui reconnaît le langage défini par l'expression suivante... Un mot du langage est constitué d'un nombre quelconque, mais non nul, de 0, suivi d'un nombre quelconque, mais non nul, de 1.



Les automates considérés jusqu'à présent ont un comportement complètement « déterminé » : pour chaque configuration état-entrée $(e, i) \in E \times I$, une et une seule transition d'état est fixée.

Cela résulte du fait qu'ils sont régis par une fonction de transition d'états t (de $E \times I$ dans E).

On peut imaginer des automates moins « rigides », pour lesquels, dans certaines configurations, plusieurs transitions d'e ?tats sont possibles ou, au contraire, aucune n'est prévue.

Pour un tel automate, qualifié de non-déterministe, t n'est plus une fonction, mais une relation binaire quelconque.

Définition

Un automate fini non déterministe à états d'acceptation est défini par (E, I, t, S, A) où :

- *E est un ensemble (fini) d'états,*
- *I est l'ensemble des entrées,*
- *t est la relation de transition des états,*
- *S, partie de E, est l'ensemble des états initiaux,*
- *A, partie de E, est l'ensemble des états d'acceptation.*

Remarque. Il se peut donc qu'une entrée puisse conduire un automate vers plusieurs états possibles ou qu'elle laisse l'automate indifférent.

Exemple. Dans cet exemple, lorsque l'automate se trouve dans l'état e_0, l'entrée a peut le faire passer dans l'état e_1 ou dans l'état e_3 et, lorsqu'il se trouve dans l'état e_1, rien n'est prévu pour l'entrée b.

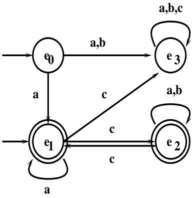


Table de transitions

t	a	b	c
$\{e_0\}$	$\{e_1, e_3\}$	$\{e_3\}$	$\emptyset$
$\{e_1\}$	$\{e_1\}$	$\emptyset$	$\{e_2, e_3\}$
$\{e_2\}$	$\{e_2\}$	$\{e_2\}$	$\{e_1\}$
$\{e_3\}$	$\{e_3\}$	$\{e_3\}$	$\{e_3\}$

Il est évidemment possible de concevoir des AFND produisant des sorties, et, en particulier, des états d'acceptation et de rejet. On admettra par ailleurs qu'il puisse y avoir, dans ces cas, plusieurs états initiaux possibles.

Soit alors M = (E,I,t,S,A) un AFND.

Définition

On dit qu'une suite w d'entrées est reconnue par l'automate si cette suite peut conduire l'automate à un état d'acceptation.

Exemple. Dans l'exemple précédent,

- Comme e_1 est à la fois un état initial et d'acceptation, le mot vide fait partie du langage reconnu par l'automate.
- Le mot aaacc est reconnu par l'automate.
- Les mots refusés sont ceux qui n'ont aucun chemin vers un état d'acceptation, comme bbb.

On définit aussi de cette manière le langage $\mathcal{L}(M)$ associé à un AFND. Il est constitué de l'ensemble des mots qui, depuis l'un des états initiaux, peut conduire à l'un des états d'acceptation.

Les AFND sont beaucoup plus simple à construire que les AFD.
Ainsi, les algorithmes de construction automatique d'automates produisent des AFND, et les algorithmes de simplification d'automate utilisent des AFND.
Mais, étant non déterministes, ils ne sont pas programmables. Heureusement, on sait les déterminer (i.e. construire un automate de Moore qui reconnaît le même langage).

L'algorithme exposé dans ce paragraphe est appelé méthode de construction par sous-ensemble. Il s'agit d'une méthode qui permet d'obtenir un automate de Moore qui reconnaît le même langage qu'un AFND.

Soit donc M=(E, I, t, S, A) un AFND à états d'acceptation. Soit Y une partie quelconque de E et $x \in I$ une entrée quelconque.

Notation : On note Y_x l'ensemble des états de M accessibles à partir de l'un quelconque des états de Y sur l'entrée x.

Exemple. Dans l'exemple précédent, et pour $Y = \{e_1, e_3\}$:

- $Y_a = \{e_1\} \cup \{e_3\} = \{e_1, e_3\}$,
- $Y_b = \{e_3\} \cup \emptyset = \{e_3\}$,
- $Y_c = \{e_2, e_3\} \cup \{e_3\} = \{e_2, e_3\}$,

On obtient un automate de Moore $M = (E, I, T, E_0, A)$ de la manière suivante :

1. L'ensemble E des états de M est le sous-ensemble de P(E) défini par :
   - $S \in E$,
   - $\forall x \in I, \forall Y \in E, Y_x \in E$.
2. L'état initial de $\mathcal{M}$ est $E_0 = S$.
3. L'ensemble $\mathcal{A}$ des états d'acceptation de $\mathcal{M}$ est défini par $\mathcal{A} = \{Y \in E$ tel que $Y \cap A \neq \emptyset\}$.
4. La fonction de transition d'états est définie par $\mathcal{T} : \mathcal{E} \times \mathcal{I} \longrightarrow \mathcal{E}, (Y, x) \mapsto \mathcal{T}(Y, x) = Y_x$.

En pratique, on part de l'état initial de $\mathcal{M}$, c'est-à-dire de S.
Pour chacune des entrées, on forme l'ensemble S_x des états de $\mathcal{M}$ que la relation de transition t permet d'atteindre à partir de tous les états de S, et on pose $\mathcal{T}(S, x) = S_x$.
On recommence l'opération pour chacun des états Sx ainsi obtenus (pour les diverses entrées x), etc.
Remarque. Le processus a une fin, parce que E est fini, donc aussi $\mathcal{P}(E)$: si l'automate de départ a n états, l'automate déterminé en aura au plus 2^n.

Remarque. Il se peut qu'aucun état ne soit accessible depuis l'un quelconque des états d'un état Y_x de $\mathcal{M}$, sur une entrée y.
On prend alors pour état d'arrivée de $\mathcal{M}$ l'ensemble vide ; celui-ci constitue un état particulier de $\mathcal{M}$, dont on ne peut sortir sur aucune entrée (c.f. exemple ci-dessous).

Exemple. Un exemple.