Chapitre 6

Automates Finis

6.1 Généralités

6.1.1 Automates finis non déterministes

Définition 6.1.1
On appelle automate fini non déterministe *(NDFA) tout quintuplet* $\mathcal{A} = \langle Q, \Sigma, I, T, \delta \rangle$ *où :*

-
-
-
-
-

Si $(p, a, q) \in \delta$*, on dit que*

On représentera plutôt les automates sous forme de graphe orienté : les sommets seront les états, les états initiaux seront distingués par une flèche entrante, et les états finals par un double rond. Enfin, les transitions seront des arêtes étiquetées par la lettre.

EXEMPLE

On peut considérer l'automate suivant

pour lequel

Les automates vont nous servir principalement à reconnaître des mots : on lit un mot lettre à lettre, et on suit les flèches correspondantes. Formellement, on fait un *calcul* dans l'automate

Définition 6.1.2
Soit $\mathcal{A} = \langle Q, \Sigma, I, T, \delta \rangle$ un automate. On appelle calcul *dans $\mathcal{A}$*
On notera

$$c =$$

p_0 est appelée état d'origine *du calcul, et p_n* état d'arrivée *du calcul.*

Le mot de Σ^* $a_0 \cdots a_{n-1}$ est appelé étiquette du calcul c, notée $|c|$.

NOTA

Par convention, il existe des calculs vides, ayant même état d'origine et de d'arrivée, d'étiquette ε.

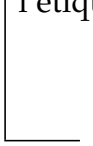
Dans un automate, certains calculs sont qualifiés de réussis, et d'autres non. Les ensembles I et T se justifient alors :

Définition 6.1.3
Un calcul c dans un automate est dit réussi *si*

On dit alors que $|c|$ est

EXEMPLE

Dans l'automate de l'exemple précédent, le mot *abbabb* est reconnu par l'automate : c'est l'étiquette du calcul



EXERCICE

Montrer qu'un automate reconnait le mot vide si et seulement s'il admet un état qui est à la fois initial et final.

On peut alors parler de langages :

Définition 6.1.4
On dit qu'un langage L sur un alphabet Σ est reconnu

pour tout mot $u \in \Sigma^*$,

On dit qu'un langage L *sur un alphabet* Σ *est* reconnaissable *s'il existe un automate fini qui le reconnaît.*

On note alors $\mathrm{Rec}(\Sigma^*)$ *l'ensemble des langages reconnaissables par automate fini.*

EXEMPLE

L'automate de l'exemple reconnaît le langage $L =$, *i.e.*

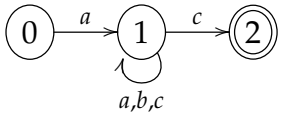
En effet,

- soit $u \in L$: on peut écrire $u = vbab$, où $v = v_0 \cdots v_{n-1} \in \Sigma^*$. Alors, le calcul

  est un calcul réussi de l'automate.
- réciproquement, soit $u = u_0 \dots u_{n-1}$ reconnu par l'automate. Pour pouvoir terminer dans l'état 3,

EXERCICE

Déterminer le langage reconnu par l'automate suivant :



Définition 6.1.5
On dit que deux automates sont équivalents *s'ils reconnaissent le même langage.*

6.1.2 Automates déterministes complets

La notion d'automate finin non-déterministe est très générale, mais difficilement utilisable en pratique : pour savoir si un mot est reconnu, il faut parcourir tous les calculs possibles de ce mot dans l'automate, et regarder si au moins l'un d'entre eux est réussi.

On peut alors remplacer les automates par des automates *déterministes*, c'est-à-dire pour lesquels il n'y a qu'un calcul possible.

Définition 6.1.6
On appelle automate fini déterministe *(DFA) tout quintuplet* $\mathcal{A} = \langle Q, \Sigma, I, T, \delta \rangle$ *où :*

-
-
-
-
-

En pratique, un automate est déterministe si pour tout état q et toute lettre a, il existe au plus une transition d'origine q étiquetée par a.

Définition 6.1.7
Si pour tout état q *et toute lettre* a *il existe exactement une transition d'origine* q *étiquetée par* a*, on dit que l'automate est complet.*

EXEMPLE

L'automate

est un automate fini déterministe complet.

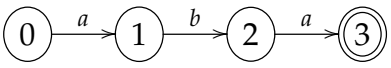
Proposition 6.1.8
Tout automate fini déterministe est équivalent à un automate fini déterministe complet.

Démonstration. Il suffit de considérer le même automate, en lui ajoutant

□

EXEMPLE

Considérons l'automate reconnaissant le langage aba :



Pour le compléter, on ajoute un autre état, qui sera un état puit :

Théorème 6.1.9
Tout automate fini non déterministe est équivalent à un automate fini déterministe complet.

Démonstration. Notons $\mathcal{A} = \langle Q, \Sigma, I, T, \delta \rangle$ l'automate à déterminiser, et posons $\mathcal{A}' = \langle Q', \Sigma, i, T', \delta' \rangle$, où

-
-
-
- pour tous $(X, a, Y) \in Q' \times \Sigma \times Q'$,

Il ne reste plus qu à se convaincre que ces automates sont bien équivalents. □

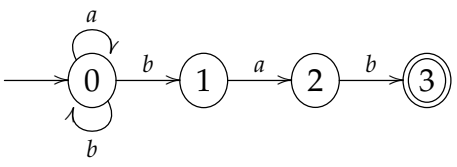
NOTA

En théorie, le déterminisé d'un automate à n états est un automate à 2^n états.
En pratique, il arrive très souvent que beaucoup de ces états soient inutiles. On constuit en fait le déterminisé petit à petit :
Q' et δ' sont initialisés à $\emptyset$, et E est un ensemble initialisé à $\{I\}$. Puis, tant que $E \neq \emptyset$:

-
- Pour toute lettre a,
  - On pose $Y =$
  - Si Y n'est pas dans Q',
  - On ajoute

EXEMPLE

Déterminisons l'automate du premier exemple



On commence avec

-

-

-

Dans l'exemple, l'automate déterminisé d'un automate à quatre états reste un automate à quatre états, au lieu de 16 prévus par la preuve.

Cependant, on peut montrer qu'il existe des automates non déterministes pour lesquels tout automate déterministe équivalent a au moins 2^n états.

Définition 6.1.10
Un automate fini est dit standard

Proposition 6.1.11
Tout langage reconnaissable est reconnaissable par un automate fini standard.

Démonstration. Soit L reconnu par $\mathcal{A} = \langle Q, \Sigma, I, T, delta \rangle$. Soit $d \notin Q$. On construit alors l'automate $\mathcal{A}' = \langle Q', \Sigma, I', T', delta' \rangle$ défini par

-
-
-

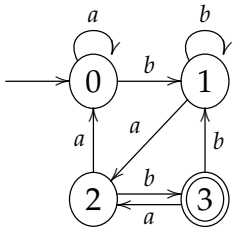
Alors $\mathcal{A}'$ est standard et reconnaît L. □

NOTA

En fait, on change l'état initial, et on rajoute un double de toutes les transitions partant d'un état initial de l'ancien automate.

EXEMPLE

Regardons l'automate précédent.



Alors son standardisé est

6.2 Stabilité des langages reconnaissables

Dans toute cette partie, au vu des résultats de la partie précédente, on considèrera que tous les automates sont des automates finis déterministes complets standards, si besoin.

Proposition 6.2.1
Tout langage à un élément est reconnaissable.

Démonstration. En exercice. □

Proposition 6.2.2
La réunion de deux langages reconnaissables est reconnaissable.

Démonstration. Si L_1 est reconnu par $\mathcal{A}_1 = \langle Q_1, \Sigma, I_1, T_1, \delta_1 \rangle$ et L_2 par $\mathcal{A}_2 = \langle Q_2, \Sigma, I_2, T_2, \delta_2 \rangle$, alors, en supposant sans perte de généralité que Q_1 et Q_2 sont disjoints, l'automate

reconnaît $L_1 + L_2$. □

Corollaire 6.2.3
Tout langage fini est reconnaissable.

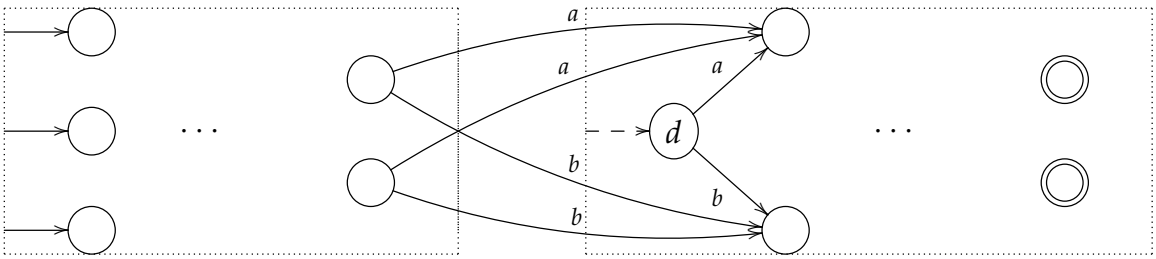
Proposition 6.2.4
La concaténation de langages reconnaissables est reconnaissable.

Démonstration. Soient L_i reconnus par $\mathcal{A}_i = \langle Q_i, \Sigma, I_i, T_i, \delta_i \rangle$, $i = 1, 2$, avec $\mathcal{A}_2$ standard ; on pose $I_2 = \{d\}$. On pose

-
-
-
- $\delta = \delta_1 \cup \delta_2' \cup \delta_3$ où

□

Pour résumer, il suffit de brancher les états finals du premier automate aux états étant juste après l'état initial du second.



Proposition 6.2.5
La fermeture de Kleene d'un langage reconnaissable est reconnaissable.

Démonstration. Il suffit de transformer l'automate $\mathcal{A} = \langle Q, \Sigma, \{d\}, T, \delta \rangle$ en $\mathcal{A}' = \langle Q, \Sigma, \{d\}, T \cup \{d\}, \delta' \rangle$ où

□

Proposition 6.2.6
Le complémentaire d'un langage reconnaissable est reconnaissable, et l'intersection et la différence de langages reconnaissables sont reconnaissables.

Démonstration. Si L est reconnu par un automate fini déterministe complet, alors $\Sigma^* \setminus L$ est reconnu par le même automate où

Ensuite, il suffit de noter que

□

6.3 Algorithme de Berry-Sethi

L'objectif de cette partie est de montrer la moitié du *théorème de Kleene* : tout langage rationnel est reconnaissable par un automate fini. L'algorithme nous fournit même une façon de calculer un automate fini qui reconnaît exactement les mots dénotés par une expression rationnelle.

6.3.1 Automates locaux

Définition 6.3.1
Un automate fini déterministe est dit local *si*

NOTA

Dans le cas d'un automate local, on a tendance à nommer les états avec les lettres qui pointent vers eux.

EXEMPLE

L'automate

est local.

Proposition 6.3.2
Tout langage local est reconnu par un automate fini local et standard. Plus précisemment, le langage L sur l'alphabet Σ est reconnu par l'automate $\mathcal{A} = \langle Q, \Sigma, \alpha, T, \delta \rangle$ où

-
-
-
-

EXEMPLE

Regardons l'automate local pour le langage local $(012)^*$: on avait vu $P = \{0\}$, $S = \{c\}$ et $F = \{01, 12, , 20\}$. On obtient l'automate

6.3.2 L'algorithme

On a vu que les langages dénotés par une expression linéaire étaient locaux, et on connaît des algorithmes pour calculer les ensembles P, S et F. Il est donc facile de calculer l'automate local correspondant.

C'est l'idée utilisée dans l'algorithme de Berry-Sethi (ou algorithme de Glushkov), qui fait les étapes suivantes pour trouver une automate fini reconnaissant un langage dénoté par une expression rationnelle e :

-

-
-
-

Si on veut trouver un automate reconnaissant le langage L dénoté par $e = (ab + b)^*ba$.

- Le linéarisé est $e' =$. On calcule alors pour cette expression linéaire

  $P =$, $S =$ et $F =$

- On peut construire l'automate correspondant :

- On supprimer les annotations pour trouver une automate (non déterministe) qui reconnaît le langage :

- On le déterminise :

6.4 Compléments

6.4.1 Réciproque du théorème de Kleene

On va donner une démonstration de la réciproque du théorème précédent pour pouvoir énoncer le théorème :

Théorème 6.4.1 : de Kleene
Un langage est reconnaissable si et seulement s'il est rationnel. Autrement dit,

La démonstration sera basée sur le lemme d'Arden, qu'on rappelle

Théorème 6.4.2 : Lemme d'Arden
Soient A et B deux langages sur le même alphabet, $\varepsilon \notin A$. Alors

On peut même résoudre des systèmes d'équations linéaires de langages :

Proposition 6.4.3
Soient A_{ij}, $i, j \in [\![1, n]\!]$ des langages ne contenant pas ε. Soient B_i, $i \in [\![1, n]\!]$ des langages. Alors le système

$$\forall i \in [\![1, n]\!],\ L_i = \sum_{j=1}^{n} A_{ij} L_j + B_i$$

admet une unique solution $(L_1, \ldots, L_n)$. De plus, les L_i sont rationnels si les A_{ij} et B_i le sont.

On peut maintenant démontrer notre réciproque :

Démonstration. Soit L un langage reconnu par $\mathcal{A} = \langle Q, \Sigma, I, T, \delta \rangle$.

Pour tous $i, j \in Q$, notons A_{ij}

Définissons aussi pour tout état i L_i

On a alors trivialement

$$L =$$

On note qu'on a aussi

On a donc un système d'équation linéaire, qui admet une unique solution, qui est rationnelle. □

EXEMPLE

Prenons l'automate des multiples de 3 en base 2 sur l'alphabet $\Sigma = \{0, 1\}$. Il est reconnu par

On a alors

La troisième équation donne , qu'on injecte dans la deuxième équation, pour obtenir .
On obtient donc , et dans la première équation, , d'où

6.4.2 Lemme de l'étoile

S'il suffit de trouver une expression rationnelle pour déterminer si un langage est rationnel, il peut être difficile de voir qu'un langage ne l'est pas. Le lemme suivant donne une condition nécessaire pour être rationnel, dont on utilise souvent la contraposée.

Lemme 6.4.4 : de l'étoile
Soit L un langage. Alors si L est rationnel, alors

Démonstration. Soit $\mathcal{A}$ un automate reconnaissant L. Soit N le nombre d'état de l'automates augmenté de 1. Alors

□

EXEMPLE

L'ensemble $L = \{a^n b^n \mid n \in \mathbb{N}\}$ n'est pas rationnel. En effet, sinon, soit N comme dans le lemme de l'étoile.

EXERCICE

Montrer que le langage des carrés $C = \{m^2 \mid m \in \Sigma^*\}$ n'est pas rationnel.

6.5 Exercices

Exercice 1. Donner des automates reconnaissant les langages sur $\Sigma = \{a, b\}$ suivants :

a) Le langage des mots ayant au moins un a

b) Le langage des mots ayant au plus un a

c) Le langages des mots ayant un nombre pair de a

d) Le langage des mots admettant aba pour facteur

e) Le langage des mots admettant aba pour sous-mot.

f) Le langage des mots ayant au moins deux occurences de leur dernier caractère.

Exercice 2. On joue à un jeu : quatre pièces sont disposées en carré, côté pile ou face vers le haut.

Un des joueurs, aveugles, ne connaissant pas la disposition de départ, peut faire les trois actions suivantes :

- tourner une pièce
- tourner deux pièces voisins
- tourner deux pièces opposées

L'autre joueur, peut faire tourner la table d'un ou plusieurs quarts de tours entre chaque coup.

Le but pour le joueur aveugle est de mettre toutes les pièces dans le même sens.

a) Montrer que le jeu peut être représenté par un automate fini, en utilisant le fait qu'il y a quatre configurations possibles.

b) Déterminiser l'automate, puis donner une stratégie gagnante.

Exercice 3. Pour un langage L, on appelle *racine de* L le langage $\{u \in \Sigma^* \mid u^2 \in L\}$.

Montrer que si L est rationnel, sa racine aussi.

Exercice 4. Soit L le langage dénoté par $(b^*a^2b^*)^*$. Donner une expression rationnelle dénontant le complémentaire de L.

Indication : on pourra utiliser les étapes suivantes : trouver un automate reconnaissant L, puis $\overline{L}$, puis retrouver une expression rationnelle avec le lemme d'Arden.

Exercice 5. Si L et L' sont deux langages sur un alphabet Σ, on appelle *mélange* de L et L' le langage sur Σ défini par

$$L \sqcup\!\sqcup L' = \{u_0v_0u_1v_1 \cdots u_nv_n \mid u_0 \cdots u_n \in L \text{ et } v_0 \cdots v_n \in L'\}.$$

Montrer que si L et L' sont rationnels, alors $L \sqcup\!\sqcup L'$ aussi.

Exercice 6. Conjecture de Černý.

Soit $\mathcal{A}$ un automate fini déterministe complet. Pour un état p, on note $p \cdot w$ l'état q (s'il existe) tel qu'il existe un calcul dans $\mathcal{A}$ d'origine p, d'arrivée q et d'étiquette w.

L'automate est dit synchronisant s'il existe un mot w tel que

$$\exists q \in Q,\ \forall p \in Q,\ p \cdot w = q.$$

w est alors dit synchronisant pour $\mathcal{A}$.

On définit les automates $\mathcal{C}_n$ sur $\Sigma = \{a, b\}$ comme suit :

- L'ensemble des états est $[\![0, n-1]\!]$
- L'état initial est 0
- L'état final est 0
- L'ensemble des transitions est

$$\{0 \xrightarrow{a} 1\} \cup \{i \xrightarrow{a} i \mid i \in [\![1, n-1]\!]\} \cup \{i \xrightarrow{b} i+1 \mid i \in [\![1, n-2]\!]\} \cup \{n-1 \xrightarrow{b} 0\}.$$

a) Dessiner $\mathcal{C}_3$ et $\mathcal{C}_4$.

b) Montrer que $u_3 = ab^2a$ est un mot synchronisant de $\mathcal{C}_3$. En trouver un pour $\mathcal{C}_4$, puis pour $\mathcal{C}_n$.

c) Pour un automate $\mathcal{A}$, on appelle $\mathfrak{p}(A)$ l'automate obtenu en rendant tous les états de $\mathcal{A}$ initiaux, puis en déterminisant (sans faire apparître les états inutiles).
Calculer $\mathfrak{p}(\mathcal{C}_3)$.

d) Montrer qu'un automate $\mathcal{A}$ est synchronisable si et seulement s'il existe dans $\mathfrak{p}(\mathcal{A})$ un état singleton, du type $\{p\}$.

e) Montrer que pour un automate synchronisant, le plus petit mot synchronisant est de longueur strictement inférieure à $2^n - n$.