Langages et Automates : LA3
Partie 2 : Automates Finis - Introduction - Determinisation

Automate Fini Déterministe

Définition

Un Automate Fini Déterministe (ou AFD) $\mathcal{A}$ est la donnée d'un quintuplet $(\Sigma, Q, q_0, F, \delta)$ dans lequel

- Σ *est un alphabet fini.*
- Q *est l'ensemble fini des états.*
- q_0 *est un élément de* Q *appelé état initial.*
- $F \subset Q$ est l'ensemble des *états finaux* de Q.
- $\delta : Q \times \Sigma \to Q \cup \{\emptyset\}$ *est la fonction de transition*

Automate Fini Déterministe

Définition

Un Automate Fini Déterministe (ou AFD) $\mathcal{A}$ est la donnée d'un quintuplet $(\Sigma, Q, q_0, F, \delta)$ dans lequel

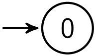
- Σ *est un alphabet fini.*
- Q *est l'ensemble fini des états.*
- q_0 *est un élément de Q appelé état initial.*
- $F \subset Q$ est l'ensemble des *états finaux* de Q.
- $\delta : Q \times \Sigma \to Q \cup \{\emptyset\}$ est la *fonction de transition*

? ? ?

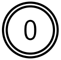
Représentation d'un automate

On peut représenter cela par le dessin d'un graphe orienté :

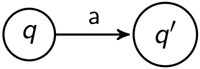
- les sommes sont étiquetés par les états de l'automate
- On indique par une flèche entrante l'état initial :



- On entoure doublement un état final :



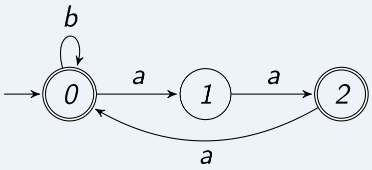
- Un arc étiqueté par a va du sommet q au sommet q' si $\delta(q, a) = q'$



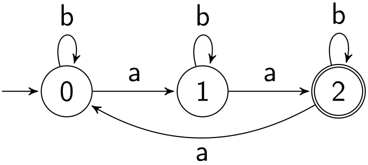
Représentation d'un automate

Exemple

Considérons l'automate $(\{a, b\}, \{0, 1, 2\}, \{0\}, \{0, 2\}, \delta)$ *où* δ *est donnée par*
$\delta(0, a) = 1$, $\delta(0, b) = 0$,
$\delta(1, a) = 2$, $\delta(1, b) = \emptyset$,
$\delta(2, a) = 0$, $\delta(2, b) = \emptyset$.
On le dessine de la façon suivante :



Représentation d'un automate



Au lieu de décrire la fonction comme ça :
$\delta(0, a) = 1$, $\delta(0, b) = 0$,
$\delta(1, a) = 2$, $\delta(1, b) = 1$,
$\delta(2, a) = 0$, $\delta(2, b) = 2$.

il est souvent plus pratique de l'écrire sous forme de table :

	a	b
0	1	0
1	2	1
2	0	2

Calcul d'un mot

Soit $\mathcal{A} = (\Sigma, Q, q_{init}, F, \delta)$ un AFD.
Le calcul du mot $w = x_1 ... x_n$ à partir d'un état q_0 où $x_i \in \Sigma$ se fait en partant de q_0 et en parcourant m états $q_0, q_1, \ldots, q_m$ (avec $m \leq n$) vérifiant $q_i = \delta(q_{i-1}, x_i)$ pour tout i.
Si à un moment la transition n'est pas définie ($\delta(q_{i-1}, x_i) = \emptyset$) on dit que le calcul bloque.
On lit un mot et on se déplace dans l'automate en suivant à chaque fois la transition sortante étiquetée par la lettre du mot qu'on est en train de lire.

Fonction de Transition étendue

Formellement, on étend $\delta : Q \times \Sigma \rightarrow Q \cup \{\emptyset\}$ en $\delta^* : Q \times \Sigma^* \rightarrow Q \cup \{\emptyset\}$ par

- $\delta^*(q, \varepsilon) = q$ pour tout état q
- $\delta^*(q, x_1 ... x_n) = \begin{cases} \emptyset & \text{si } \delta(q, x_1) = \emptyset \\ \delta^*(\delta(q, x_1), x_2 ... x_n) & \text{sinon} \end{cases}$

Fonction de Transition étendue

Formellement, on étend $\delta : Q \times \Sigma \to Q \cup \{\emptyset\}$ en $\delta^* : Q \times \Sigma^* \to Q \cup \{\emptyset\}$ par

- $\delta^*(q, \varepsilon) = q$ pour tout état q
- $\delta^*(q, x_1 ... x_n) = \begin{cases} \emptyset & \text{si } \delta(q, x_1) = \emptyset \\ \delta^*(\delta(q, x_1), x_2 ... x_n) & \text{sinon} \end{cases}$

$\delta^*(q, w) = q'$ siginifie donc : en partant de l'état q, on effectue le calcul du mot w sans bloquer et on arrive dans l'état q'.

Fonction de Transition étendue

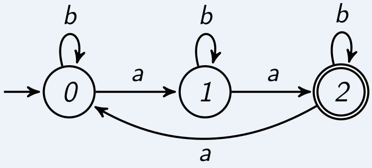
Formellement, on étend $\delta : Q \times \Sigma \to Q \cup \{\emptyset\}$ en $\delta^* : Q \times \Sigma^* \to Q \cup \{\emptyset\}$ par

- $\delta^*(q, \varepsilon) = q$ pour tout état q
- $\delta^*(q, x_1 ... x_n) = \begin{cases} \emptyset & \text{si } \delta(q, x_1) = \emptyset \\ \delta^*(\delta(q, x_1), x_2 ... x_n) & \text{sinon} \end{cases}$

$\delta^*(q, w) = q'$ siginifie donc : en partant de l'état q, on effectue le calcul du mot w sans bloquer et on arrive dans l'état q'.

Exemple

Dans l'exemple suivant que valent $\delta^*(0, aaba)$, $\delta^*(0, aaa)$, $\delta^*(1, aaa)$?



Définition

Soit $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$ *un AFD.*

- *On dit qu'un mot* $w \in \Sigma^*$ *est* ***accepté*** *par* $\mathcal{A}$ *si* $\delta^*(q_0, w) \in F$.
- *L'ensemble* $\mathcal{L}(\mathcal{A})$ *de tous les mots qui sont acceptés par est appelé* ***langage reconnu par l'automate*** $\mathcal{A}$.

Définition

Un langage L *est dit* ***reconnaissable*** *si il existe un automate* $\mathcal{A}$ *tel que* $L = \mathcal{L}(\mathcal{A})$.

Acceptation - Langages Reconnaissables

Trouver un automate reconnaissant les langages suivants sur l'alphabet $\{a, b\}$:

- $L_1 = \{aba, bab\}$
- $L_2 = \{a, aba, abb, babb\}$

Proposition

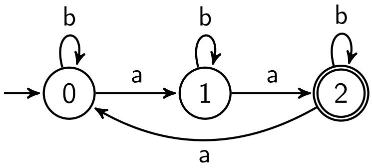
Tout langage fini est reconnaissable.

Remarque

De même que dans le cas des expressions rationnelles, deux automates non identiques peuvent reconnaître le même langage.

Exercices

Quel est le langage reconnu par l'automate suivant ?



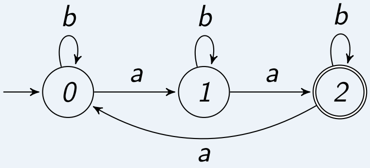
Trouver un automate pour l'ensemble des mots de longueur impaire
Trouver un automate pour l'ensemble des mots commençant par un a
Trouver un automate pour l'ensemble des mots contenant le facteur *aba*.

AFD complets

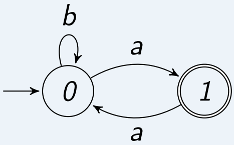
Définition (AFD complet)

Un AFD est dit *complet* *si pour tout état q et tout lettre* $a \in \Sigma$, $\delta(q, a) \neq \emptyset$.

Exemple



Complet



Non Complet

Complété

Proposition

Pour tout AFD $\mathcal{A} = (\Sigma, Q, q_0, F, \delta)$*, il existe un AFD complet* $\mathcal{A}' = (\Sigma', Q', {q_0}', F', \delta')$ *reconnaissant le même langage.*

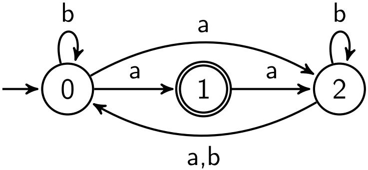
Définition (AFND)

Un Automate Fini Non Deterministe $\mathcal{A}$ *est la donnée d'un quintuplet* $(\Sigma, Q, Q_0, F, \delta)$ *dans lequel*

- Σ *est un alphabet fini.*
- Q *est l'ensemble fini des états.*
- $Q_0 \subset Q$ *est l'ensemble des états initiaux.*
- $F \subset Q$ *est l'ensemble des états finaux.*
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ *est la fonction de transition*

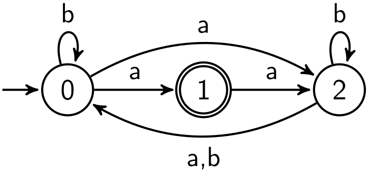
$\mathcal{P}(Q)$ désigne l'ensemble des parties de Q :
D'un même état q peuvent partir deux transitions (ou plus) étiquetées par la même lettre a vers deux états différents.

Considérons l'automate $(\{a, b\}, \{0, 1, 2\}, \{0\}, \{1\}, \delta)$ où δ est donnée par
$\delta(0, a) = \{1, 2\}$, $\delta(0, b) = 0$,
$\delta(1, a) = 2$, $\delta(1, b) = \emptyset$,
$\delta(2, a) = 0$, $\delta(2, b) = \{2, 0\}$.



AFND - Calcul d'un mot

Dans le cas déterministe, pour effectuer le calcul d'un mot, on doit explorer toutes les possibilités de lire ce mot dans l'automate. On obtient une liste d'état possibles. Ainsi sur l'exemple précédent



on obtient pour les deux mots *aaba* et *bab* : $\delta^*(0, aaba) = \{0, 1\}$ et $\delta^*(2, bab) = \{0\}$.

AFND - Langage Reconnu

Formellement la fonction de transition étendue $\delta^* : Q \times \Sigma \rightarrow \mathcal{P}(Q)$ est définie par :

- $\delta^*(q, \varepsilon) = q$ pour tout état q
- $\delta^*(q, x_1 ... x_n) = \bigcup_{q' \in \delta(q, x_1)} \delta^*(q', x_2 ... x_n)$

Un mot w est alors accepté si l'ensemble des destinations possibles $\delta^*(q_0, w)$ contient au moins un état final.

Union de deux langages reconnaissables

Si L_1 et L_2 sont deux langages reconnaissables, une question naturelle est de savoir si $L_1 \cup L_2$ est aussi reconnaissable, c'est à dire : existe-t-il un AFD qui reconnait ce langage ?
Si on s'autorise à utiliser un automate non déterministe, la réponse est oui très facilement. Il suffit de faire l'union disjointe des deux automates.

Union de deux langages reconnaissables

Si L_1 et L_2 sont deux langages reconnaissables, une question naturelle est de savoir si $L_1 \cup L_2$ est aussi reconnaissable, c'est à dire : existe-t-il un AFD qui reconnait ce langage ?
Si on s'autorise à utiliser un automate non déterministe, la réponse est oui très facilement. Il suffit de faire l'union disjointe des deux automates.

Mais ceci ne répond pas à la question, puisque pour être reconnaissable, il faut trouver un automate DETERMINISTE qui reconnait ce langage.

Peut-on transformer un automate non deterministe en automate deterministe reconnaissant les mêmes mots ?

Théoreme

Pour tout AFND $\mathcal{A}$ il existe un AFD $\mathcal{A}'$ tel que $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.

Preuve :

Soit $\mathcal{A} = (\Sigma, Q, Q_0, F, \delta)$ un automate fini non déterministe.
On définit un automate $\mathcal{A}'$ déterministe dont chaque état correspond à un ensemble d'état de $\mathcal{A}$.
$\mathcal{A}'$ est défini par le quintuplet $(\Sigma, \mathcal{P}(Q), Q_0, F', \delta')$ où :

- $F' = \{P \in \mathcal{P}(Q), P \cap F \neq \emptyset\}$
- $\forall P \in \mathcal{P}(Q), \forall a \in \Sigma, \ \delta'(P, a) = \bigcup_{q \in P} \delta(q, a)$

Il est facile de prouver (par recurrence sur la longueur du mot) que pour tout mot w, lorsque on le lit dans $\mathcal{A}'$, on arrive dans un état étiqueté par l'ensemble des états de $\mathcal{A}$ que l'on aurait pu atteindre en le lisant de toutes les façons possibles (cad $\delta'^*(Q_0, w) = \delta^*(q_0, w)$.)
L'automate $\mathcal{A}'$ reconnait donc le même langage que $\mathcal{A}$.

Applications

- L'union de deux langages reconnaissables est reconnaissables
- Clotures par prefixe, suffixe, facteurs.
- Le miroir d'un langage reconnaissable est reconnaissable
- Le complémentaire d'un rec est rec. (?) et du coup l'intersection aussi