Fondements de l'informatique

Dans ce cours, nous nous intéresserons à des *modèles de calcul*, comme ceux ci-dessous :

(1) les automates finis, qui engendrent les langages rationnels ;
(2) les automates à pile, qui engendrent les langages algébriques ;
(3) les machines de Turing, qui engendrent les langages décidables.

Comme vu dans la thèse de Church-Turing, la machine de Turing est le modèle le plus complexe que l'on peut exécuter. Mais, il existe des modèles équivalents : les fonctions récursives, le λ-calcul.

On étudiera un peu de complexité : avec les classes **P** et **NP**, et le théorème de Cook-Levin.

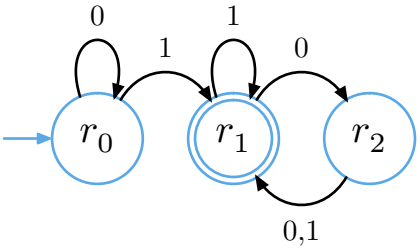
I. Automates finis.

On se place dans la situation suivante : on modélise une porte automatique, avec un capteur avant la porte et un capteur après la porte. On peut modéliser le mécanisme de contrôle de la porte par un automate.

Définition

Un automate fini est un 5-uplet $(Q, \Sigma, \delta, q_0, F)$ avec
(1) Q est l'*ensemble* fini *des états* ;
(2) Σ est l'*alphabet d'entrée* ;
(3) $\delta : Q \times \Sigma \to Q$ est la *fonction de transition* (totale) ;
(4) $q_0 \in Q$ est l'*état initial* ;
(5) $F \subseteq Q$ est l'*ensemble des états finaux*.

Exemple



Cet automate reconnaît l'ensemble des mots où il y a au moins un `1`, et le dernier `1` est suivi d'un nombre pair de `0`.

Définition

Étant donné l'état initial q_0 et un mot $w_1 w_2 \dots w_n \in \Sigma^\star$, la *suite de transition* d'un automate M est défini par la suite :

$$q_0 \to \delta(q_0, w_1) \to \delta(\delta(q_0, w_1), w_2) \to \cdots$$

avec $q_i = \delta\big(q_{i-1,w_i}\big)$.

On dit que w est *accepté* si $q_n \in F$.

Le langage reconnu est donc

$$\mathscr{L}(M) = \{w \in \Sigma^\star \mid w \text{ est accepté par } M\}$$

Un langage reconnu par un automate fini est dit *rationnel*.

On étend la fonction de transition δ : pour $w \in \Sigma^\star$, on note $\delta(q, w)$ (ou $\delta^\star(q, w)$) par l'état obtenu après lecture du mot w en partant de l'état q_0. On l'appelle la fonction de transition étendue.

I.1. | Propriétés de clôture.

On définit les opérations ci-dessous :

(1) Complément $\bar{L} = \{w \in \Sigma^\star \mid w \notin L\}$;
(2) Union $L \cup M$;
(3) Concaténation $L \cdot M = \{xy \mid x \in A \text{ et } y \in B\}$;

(4) Étoile $L^\star = \{x = x_1 x_2 \dots x_k \mid x_i \in L \text{ et } k \geqslant 0\}$

Théorème

L'ensemble des langages rationnels est clos par complément.

Preuve. On réalise la construction suivante. Soit L reconnu par $M = (Q, \Sigma, \delta, q_0, F)$, alors $\bar{L}$ est reconnu par $\bar{M} = (Q, \Sigma, \delta, q_0, \bar{F})$ avec $\bar{F} = Q \setminus F$. □

Théorème

L'ensemble des langages rationnels est clos par union.

Preuve. On réalise la construction suivante. Soit L_1 reconnu par $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ et L_2 reconnu par $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$. On veut construire M qui reconnaît $L_1 \cup L_2$. On définit M comme $M = (Q, \Sigma, \delta, q_0, F)$ par :

- $Q = Q_1 \times Q_2$;
- $\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a))$;
- $q_0 = (q_1, q_2)$;
- $F = \{(q_1, q_2) \in Q \mid q_1 \in F_1 \text{ ou } q_2 \in F_2\} = (F_1 \times Q_2) \cup (Q_1 \times F_2)$.

Avec cette construction, on a :

$$\delta^\star(q_0, q) = (\delta_1^\star(q_1, w), \delta_2^\star(q_2, w)),$$

et au vue de la définition de F, on peut en conclure que M reconnaît bien $L_1 \cup L_2$. □

On souhaite démontrer la clôture par concaténation, mais pour cela, on va devoir utiliser des automates non déterministes.

Définition

Un *automate fini non déterministe* (*NFA*) est un 5-uplet $N = (Q, \Sigma, \delta, q_0, F)$:

(1) Q est l'*ensemble* fini *des états* ;
(2) Σ est l'*alphabet d'entrée* ;
(3) $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \wp(Q)$ est la *fonction de transition* ;
(4) $q_0 \in Q$ est l'*état initial* ;
(5) $F \subseteq Q$ est l'*ensemble des états finaux*.

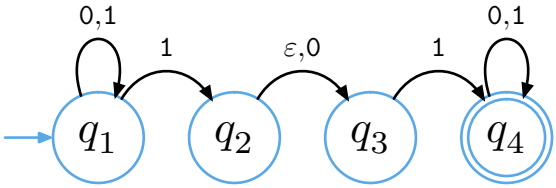
Dans un automate non déterministe, il est possible de se trouver dans le cas $\delta(q, a) = \emptyset$. Dans ce cas, la chaîne de transitions est rompue.

Définition

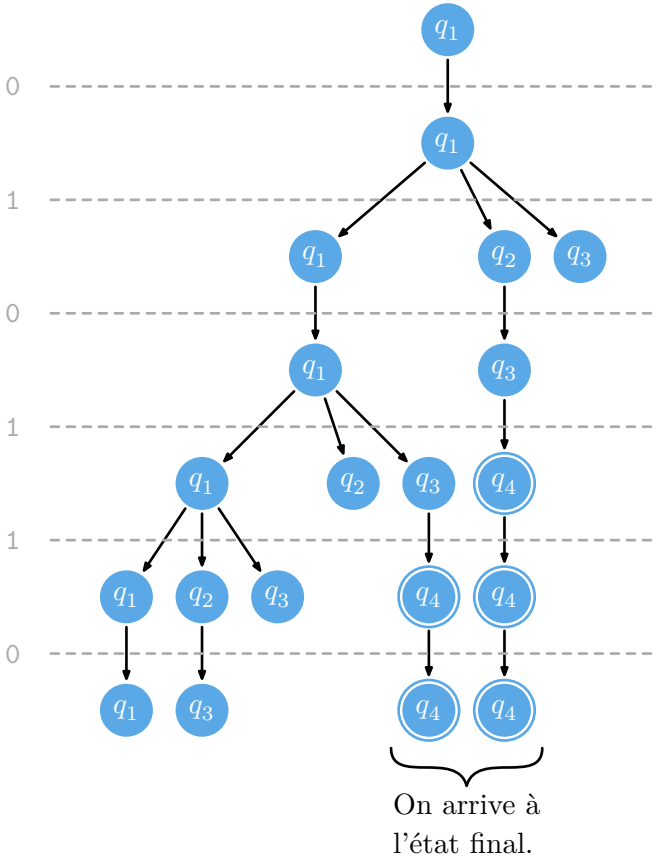
On dit qu'un mot w est *accepté* s'il existe un calcul acceptant sur l'entrée w.

Formellement, w est accepté si $w = y_1 y_2 \cdots y_m$ avec $y_i \in \Sigma \cup \{\varepsilon\}$ de sorte qu'il existe une suite de m états $r_1, r_2, \ldots, r_m \in Q$ avec :
(1) $r_0 = q_0$;
(2) $r_i \in \delta(r_{i-1}, y_i)$ pour $i \in [\![1, m]\!]$;
(3) $r_m \in F$.

Exemple



On lit le mot `010110` avec l'automate ci-dessus, et on obtient l'arbre de calculs ci-dessous.



Théorème

Un langage est reconnaissable par NFA si, et seulement s'il est rationnel.

Preuve. Dans un sens, cela est évident : un automate fini déterministe est un NFA. L'autre sens demande une construction.

Soit $N = (Q, \Sigma, \delta, q_0, F)$ un automate fini non déterministe reconnaissant $L \subseteq \Sigma^\star$. On construit $M = (Q', \Sigma, \delta', q_{0'}, F')$ un automate fini déterministe reconnaissant L : on pose

(1) États : $Q' = \wp(Q)$,

(2) États finaux : $F' = \{R \in Q \mid R \cap F \neq \emptyset\}$,

(3) Fonction de transition :

- si N ne contient pas d'ε-transitions, alors on a $\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$

- sinon, on a $\delta'(R,a) = \bigcup_{r\in R} E(\delta(r,a))$

(4) État initial : $q_{0'} = E(\{q_0\})$.

On définit la fonction E comme :

$$E(R) = \{q \in Q \mid q \text{ peut-être atteint à partir d'un état de } R \text{ en prenant une suite finie d'}\varepsilon\text{-transitions}\}.$$

On la nomme la *clôture* par ε-transitions. (Dans l'exemple précédent, on a $E(\{q_2\}) = \{q_2, q_3\}$).)

On peut démontrer que l'on a $\delta^\star(q_0, w) = \delta'^\star(q_{0'}, w)$ par récurrence pour conclure. □

Théorème

L'ensemble des langages rationnels est clos par concaténation.

Preuve. Soient deux NFA N_1 et N_2 reconnaissant L_1 et L_2. On construit un automate N qui reconnaît $L_1 \cdot L_2$. L'*idée* est la suivante : si $x \in L_1 \cdot L_2$ alors $x = y_1 y_2$ avec $y_1 \in L_1$ et $y_2 \in L_2$. Il suffit donc d'enchaîner les états initiaux N_2 à la suite des états finaux de N_1, avec des ε-transitions.

Posons $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ et $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ et on construit $N = (Q, \Sigma, \delta, q_0, F)$ par :

(1) États : $Q = Q_1 \uplus Q_2$;
(2) État initial : $q_0 = q_1$;
(3) États finaux : $F = F_2$;
(3) Fonction de transition :

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & \text{si } q \in Q_1 \setminus F_1 \\ \delta_1(q,a) & \text{si } q \in F_1 \text{ et } a \neq \varepsilon \\ \delta_1(q,\varepsilon) \cup \{q_2\} & \text{si } q \in F_1 \text{ et } a = \varepsilon \\ \delta_2(q,a) & \text{si } q \in Q_2 \end{cases}$$

□

Théorème

L'ensemble des langages rationnels est clos par étoile.

Preuve. Soit un NFA N reconnaissant L. On construit un automate $N^\star$ qui reconnaît $L^\star$.

On construit un automate comme décrit ci-après :

(1) on ajoute un état initial final q ;
(2) on ajoute une ε-transition de q à l'état initial de N ;
(3) on ajoute des ε-transitions entre les états finaux de N et l'état initial de N.

Cet automate reconnaît bien $L^\star$. □

II. Expressions rationnelles.

Une expression rationnelle (« *regular expressions* » en anglais) est une expression de la forme $(0 \cup 1)^\star 1 (00)^\star$.

Définition

Les expressions rationnelles sont de la forme :
(1) a avec $a \in \Sigma$;
(2) ε, le mot vide ;
(3) $\emptyset$, l'ensemble vide ;
(4) $R_1 \cup R_2$ où R_1 et R_2 sont deux expressions rationnelles déjà construites ;
(5) $R_1 \cdot R_2$ où R_1 et R_2 sont deux expressions rationnelles déjà construites ;
(6) $R^\star$ où R est une expression rationnelle.

On note $\mathscr{R}(\Sigma)$ l'ensemble des expressions rationnelles sur l'alphabet Σ.

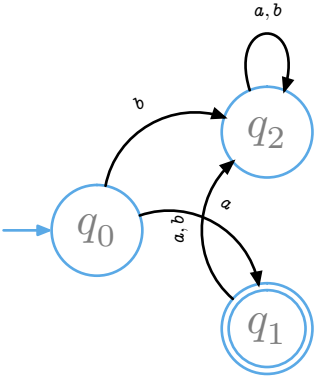
Définition

On définit $\mathscr{L}(R) \subseteq \Sigma^\star$ le langage de l'expression R :
(1) $\mathscr{L}(a) = \{a\}$ avec $a \in \Sigma$;
(2) $\mathscr{L}(\varepsilon) = \{\varepsilon\}$;
(3) $\mathscr{L}(\emptyset) = \emptyset$;
(4) $\mathscr{L}(R_1 \cup R_2) = \mathscr{L}(R_1) \cup \mathscr{L}(R_2)$;
(5) $\mathscr{L}(R_1 \cdot R_2) = \mathscr{L}(R_1) \cdot \mathscr{L}(R_2)$;
(6) $\mathscr{L}(R^\star) = \mathscr{L}(R)^\star$.

Proposition

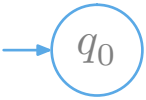
Si L est décrit par une expression rationnelle, alors L est reconnaissable par un automate.

Preuve. On traîte les différents cas :
(1) si $R = a$ avec $a \in \Sigma$, alors on construit l'automate ci-dessous ;



(2) si $R = \varepsilon$, alors on construit l'automate ci-dessous ;

(3) si $R = \emptyset$ alors on construit l'automate ci-dessous;



(4) propriétés de clôture des langages rationnels (réunion) ;
(5) propriétés de clôture des langages rationnels (concaténation) ;
(6) propriétés de clôture des langages rationnels (étoile).

□

Un *automate non déterministe généralisé* (GNFA) est un automate où

- les transitions sont étiquetées par des expressions rationnelles ;
- de l'état initial, une flèche va vers chaque état, mais ne reçoit aucune autre flèche ;
- un unique état final, qui reçoit une flèche de chaque état, mais n'en émet aucune ;
- pour les autres états : une flèche vers tous les autres états, sauf l'état initial.

Définition (GNFA, formellement)

Un *GNFA* est un 5-uplet $(Q, \Sigma, \delta, q_0, q_\mathrm{f})$ avec :
- Q l'ensemble fini des états ;
- Σ l'alphabet fini ;
- $\delta : (Q \setminus \{q_\mathrm{f}\}) \times (Q \setminus \{q_0\}) \to \mathscr{R}(\Sigma)$ la fonction d'étiquetage de transition ;
- q_0 l'état initial ;
- q_f l'état final.

Définition

On dit d'un mot $w \in \Sigma^\star$ qu'il est *accepté* s'il existe k mots $w_1, \dots, w_k$ tels que $w = w_1 \dots w_k$ et des états $q_0, \dots, q_k$ avec q_0 l'état initial, et $q_k = q_\mathrm{f}$ l'état final, et que $w_i \in \mathscr{L}(\delta(q_{i-1}, q_i))$, quel que soit i.

Proposition

Si L est reconnaissable par un automate fini, alors L peut être décrit par une expression rationnelle.

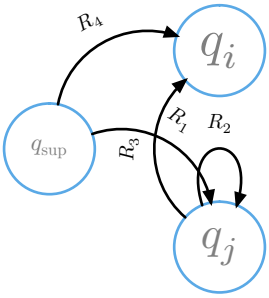
Lemme (1)

Tout GNFA avec $k > 2$ états est équivalent à un GNFA avec $k - 1$ états.

Lemme (2)

Tout DFA admet un GNFA équivalent.

Preuve (du lemme (1)). On veut supprimer q_sup dans G, avec $q_\mathrm{sup} \notin \{q_0, q_\mathrm{f}\}$.



On peut le remplacer par une unique transition $q_i \xrightarrow{R_4 \cup R_1 R_2^* R_3} q_j$. On fait cette transformation pour tous les couples (i, j).

On obtient un automate généralisé G' équivalent à G. En effet, soit w un mot accepté par G et soit $q_0; \ldots, q_{k-1}, q_{\text{f}}$ la suite des états dans un calcul acceptant de G sur l'entrée w. On obtient une ou des états dans un calcul acceptant □

Preuve (du lemme (2)). □

Preuve (de la proposition). On passe d'un DFA à un GNFA puis à une expression rationnelle.

□

Théorème

Un langage peut être décrit par une expression rationnelle si et seulement s'il est rationnel.

La suite du cours de Fondements de l'Informatique (FDI) ne sera pas tapé à l'ordinateur. Regardez le livre *Introduction to the Theory of Computation* de Michael Sipser. Le cours de FDI est basé sur ce livre, et il contient bien plus de détails.