le cnam

Codes et Automates finis

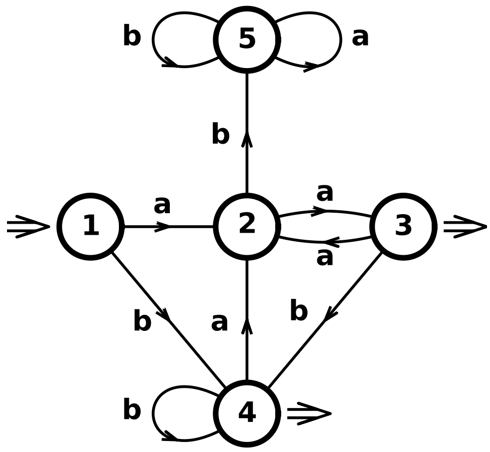
Cours 12 Simplification d'automates

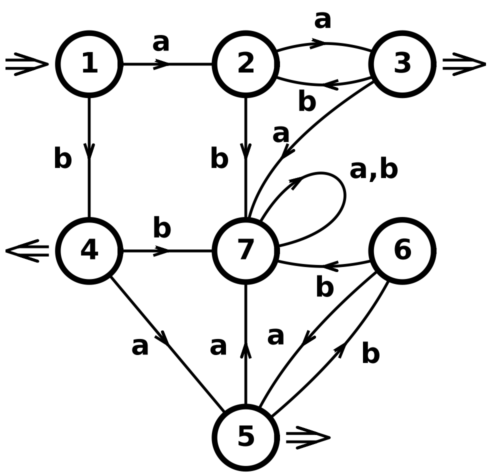
- Rappels sur les automates finis déterministes

On se donne un automate fini déterministe $\mathscr{A} = (Q, A, T, I, F)$. La transition T est définie dans ce cas par une application $\delta : Q \times A \longrightarrow Q$ qui à tout état $p \in Q$, et toute lettre $a \in A$ associe un nouvel état q de façon unique : $q = \delta(p, a)$. La transition $q = \delta(p, a)$ est représentée usuellement sous la forme $p \xrightarrow{a} q$ avec un graphe étiqueté qui permet de visualiser l'automate $\mathscr{A}$.

On note $L = \mathscr{L}(\mathscr{A})$ le langage des mots acceptés par l'automate. On cherche à réduire l'automate $\mathscr{A}$, c'est à dire trouver (quand c'est possible !) un automate $\mathscr{A}'$ plus simple, c'est à dire essentiellement avec moins d'états $q \in Q$ de sorte que le langage $\mathscr{L}(\mathscr{A}')$ des mots acceptés par le nouvel automate est exactement égal à L: $\mathscr{L}(\mathscr{A}) = \mathscr{L}(\mathscr{A}')$.

Dans ce chapitre, nous allons utiliser essentiellement les deux exemples suivants, extraits du livre d'exercices de Avérous, Gil, Santi et Vélu (Dunod, 2008). L'exemple 1 est à gauche et l'exemple 2 est à droite.





- Équivalence, congruence

Regrouper plusieurs états de l'ensemble E revient à définir une relation d'équivalence, appelée également congruence, sur l'ensemble des états. Nous rappelons les éléments fondamentaux de cette notion.

On se donne un ensemble E tout à fait quelconque et une relation $\sim$ entre deux éléments de E. On dit que la relation $\sim$ est une relation d'équivalence sur E si et seulement si

(i) $\sim$ est réflexive : $x \sim x$ ce pour tout $x \in E$

(ii) $\sim$ est symétrique : $(x \sim y) \Longrightarrow (y \sim x)$ pour tous les $x \in E$ et $y \in E$

(iii) $\sim$ est transitive : $\big((x \sim y)$ et $(y \sim z)\big) \Longrightarrow (x \sim z)$ pour tous les x, y, z dans E

François Dubois, mai 2024.

- Exemple de relation d'équivalence : les entiers modulo deux

Cet exemple n'a *a priori* rien à voir avec les automates finis. On prend $E = \mathbb{Z}$, ensemble des entiers positifs ou négatifs et la relation $\sim$ définie par $(x \sim y)$ si et seulement si $(x-y)$ est un multiple de 2. On a $0 \sim 2$, $0 \sim 4$, $-2 \sim 0$, ... mais $0 \nsim 1$, alors que $1 \sim 3$, $1 \sim -1$, *etc.*
On remarque aussi que l'égalité est une équivalence particulière.

- Ensemble quotient $E/\!\sim$

Pour $x \in E$ donné : on regroupe tous les $y \in E$ de sorte que $x \sim y$. On appelle classe d'équivalence de l'élément x, ou plus simplement "classe de x" et on note $\dot{x}$ l'ensemble de tous les $y \in E$ tels que $x \sim y$: $\dot{x} = \{y \in E, x \sim y\}$ par définition.
Pour l'exemple précédent, on a deux classes d'équivalence : $\dot{0} = \{x \in \mathbb{Z},\ x$ est multiple de $2\}$, c'est à dire l'ensemble des nombres pairs et $\dot{1} = \{x \in \mathbb{Z},\ x$ n'est pas multiple de $2\}$, soit l'ensemble des nombres impairs.
L'ensemble quotient $E/\sim$ est par définition l'ensemble des classes pour tous les éléments $x \in E$: $E/\!\sim = \{\dot{x}, x \in E\}$.
Pour l'exemple précédent des entiers modulo deux, on a deux classes seulement :
$\mathbb{Z}/\!\sim = \{\ \dot{0},\ \dot{1}\ \}$.

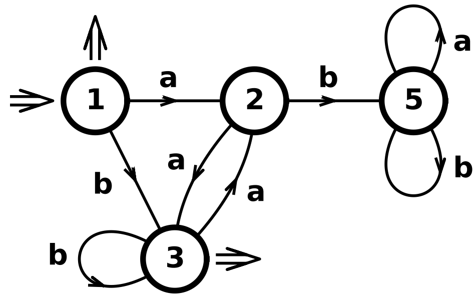
- Congruence compatible avec l'automate $\mathscr{A}$

On se donne un automate fini déterministe $\mathscr{A} = (Q, A, T, I, F)$. Une relation d'équivalence, une congruence $\sim$ sur l'ensemble des états Q est bien entendu réflexive, symétrique et transitive. C'est de plus une congruence compatible avec l'automate $\mathscr{A}$ si elle vérifie les deux conditions suivantes
(i) respect des états finals : $(q \sim q') \Longrightarrow (q \in F \iff q' \in F)$; deux états liés par la relation $\sim$ doivent être tous deux des états finals ou bien tous deux des états non finals. En d'autres termes, on ne peut relier par une congruence compatible avec l'automate $\mathscr{A}$ un état final et un état qui n'est pas un état final.
(ii) respect des transitions : $(q \sim q') \Longrightarrow (\forall a \in A,\ \delta(q, a) \sim \delta(q', a))$; la congruence $\sim$ doit être compatible avec les transitions de l'automate $\mathscr{A}$.

- Automate quotient $\mathscr{A}/\!\sim$

On se donne un automate fini déterministe $\mathscr{A} = (Q, A, T, I, F)$ avec un seul état initial $i \in I$. L'automate quotient $\mathscr{A}/\!\sim$ est un automate dont les états $\dot{Q} = Q/\!\sim$ sont les classes d'équivalence des états de l'ensemble Q ; l'alphabet A reste inchangé, les transitions $T/\!\sim$ sont de la forme $(\dot{p}, a) \longrightarrow \dot{q}$ et l'état $\dot{q}$ est la classe d'équivalence de $\delta(p, a)$, l'ensemble $I/\!\sim$ des états initiaux est la classe de l'état i: $I/\!\sim = \{p \in Q, p \sim i\}$ et $F/\!\sim$ est l'ensemble des classes des états finals : $F/\!\sim = \{\dot{f}, f \in F\}$.
En pratique, on garde un seul état de claque classe et on ne représente plus les autres états. Pour le premier exemple, les états ③ et ④ sont tous deux des états de sortie. On a de plus les transitions $③ \xrightarrow{a} ②$ et $④ \xrightarrow{a} ②$ d'une part, $③ \xrightarrow{b} ④$ et $④ \xrightarrow{b} ④$ d'autre part. Après avoir mis les états ③ et ④ en équivalence [③ $\sim$ ④], c'est à dire après les avoir identifiés pour l'automate quotient, on peut construire le graphe correspondant. On a simplement enlevé l'état ④ et on change si nécessaire le nom de l'état ③. On a donc le graphe qui suit après une première

réduction de l'automate initial.

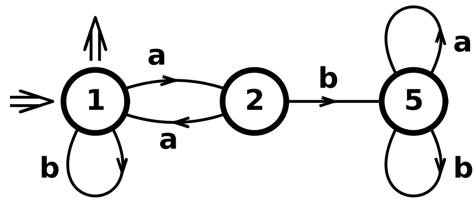


- Égalité des langages acceptés

On se donne une congruence $\sim$ compatible sur l'automate fini $\mathscr{A} = (Q, A, T, I, F)$. Donc $q \sim q'$ implique qu'on ne peut pas avoir q état final et q' état non final et $q \sim q'$ entraîne $\delta(q, a) \sim \delta(q', a)$ pour toute lettre $a \in A$. Alors les langages $\mathscr{L}(\mathscr{A})$ de l'automate initial et $\mathscr{L}(\mathscr{A}/\sim)$ de l'automate quotient sont égaux. Les mots acceptés par l'automate quotient sont exactement les mêmes que les mots acceptés par l'automate initial.

On peut illustrer cette propriété en cherchant les mots de petite longueur acceptés par l'automate de la figure de gauche initiale : ε, b, aa, bb, baa, aab, *etc.* Quand on examine maintenant la figure ci-dessus de l'automate réduit, on trouve exactement la même liste pour les mots acceptés de petite longueur.

On peut encore regrouper deux états de l'automate réduit de la figure ci-dessus. En effet, on a $1 \xrightarrow{a} 2$ et $3 \xrightarrow{a} 2$ d'une part, $1 \xrightarrow{b} 3$ et $3 \xrightarrow{b} 3$ d'autre part. De plus, les états 1 et 3 sont tous deux des états de sortie. On peut donc considérer la congruence où $1 \sim 3$; dans une représentation *via* un graphe de l'automate quotient, faire simplement disparaître l'état 3. On obtient alors un nouvel automate quotient décrit par la figure ci-dessous.



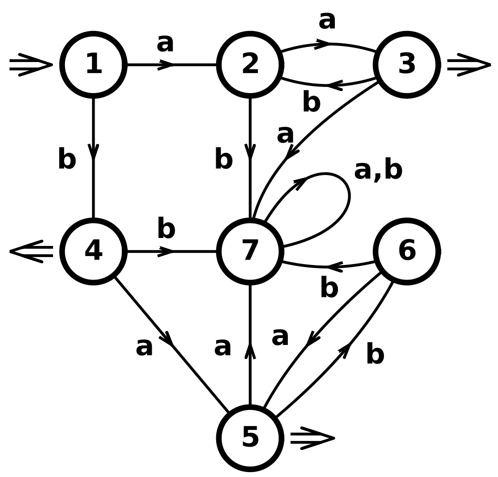
On peut vérifier que les mots acceptés de courte longueur de l'automate de la figure ci-dessus sont ceux de la liste ε, b, aa, bb, baa, aab, *etc.*

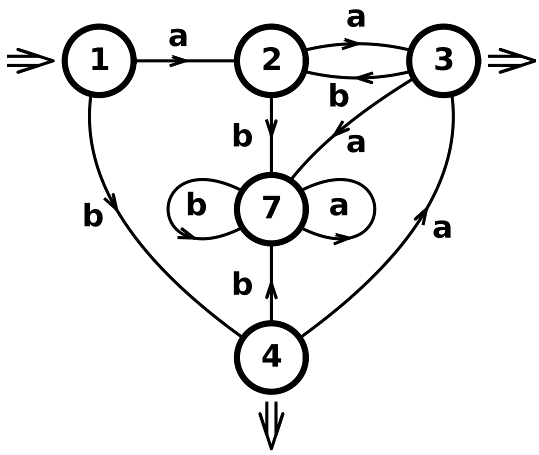
- Exemple de réduction d'un automate

Nous considérons l'automate présenté à la figure de droite de la première page et présenté à nouveau sur la figure de gauche ci-dessous.

Les états 3 et 5 sont tous deux des états de sortie. De plus, on a les transitions $3 \xrightarrow{a} 7$ et $5 \xrightarrow{a} 7$ d'une part, $3 \xrightarrow{b} 2$ et $5 \xrightarrow{b} 6$ d'autre part. Ceci ne garantit pas complètement l'équivalence $3 \sim 5$ et incite è étudier l'équivalence entre les états 2 et 6, qui sont tous deux des états non finals. On a les transitions $2 \xrightarrow{a} 3$ et $6 \xrightarrow{a} 5$ d'une part et d'autre part

② $\xrightarrow{b}$ ⑦ et ⑥ $\xrightarrow{b}$ ⑦. Donc il est possible de mettre en équivalence les états ③ et ⑤ d'une part et les états ② et ⑥ d'autre part. Cette équivalence conduit à l'automate réduit présenté à droite ci-dessous.





On se rend compte facilement que les mots de petite longueur du langage de l'automate de la figure de gauche sont identiques à ceux de l'automate de la figure de droite : b, ba, aa, $aaba$, *etc*.

• Congruence de Nerode [Anil Nerode, né en 1932, mathématicien américain]

Avant de définir la congruence de Nerode, nous rappelons que nous pouvons étendre à tous les mots $w \in A^*$ l'opérateur de transition $Q \times A \ni (p, a) \longmapsto q = \delta(p, a) \in Q$. En effet, si $w = uv$ est la concaténation des mots u et v, on pose $\delta(p, uv) = \delta(\delta(p, u), v))$. Cette définition permet d'étendre de proche en proche la définition de l'opérateur δ à des mots de longueur arbitraire sur l'alphabet A. Rappelons que l'on a $\delta(p, \varepsilon) = p$.

• Définition de la congruence de Nerode (1958)

On se donne un automate $\mathscr{A} = (Q, A, T, \{i\}, F)$ avec un unique état i d'entrée. La congruence de Nerode est définine par $q \sim q'$ si et seulement si $\forall\, w \in A^*$, $(\delta(q, w) \in F \iff \delta(q', w) \in F)$. Pour relier les deux états q et q', on regarde tous les chemins issus de q et q' et pilotés par les mêmes mots w. On peut écrire la congruence $q \sim q'$, l'équivalence $q \sim q'$, lorsqu'on atteint un état de sortie que l'on parte de l'état q ou que l'on parte de l'état q' et lorsqu'on atteint un état qui n'est pas un état final que l'on parte de $q \in Q$ ou que l'on parte de $q' \in Q$.

• La congruence de Nerode est une congruence compatible avec l'automate

On peut vérifier sans difficulté que la congruence de Nerode est une relation réflexive, symétrique et transitive sur l'ensemble Q des états de l'automate $\mathscr{A} = (Q, A, T, \{i\}, F)$.

De plus, si $q \sim q'$, alors $\delta(q, \varepsilon) = q$ appartient à F si et seulement si $\delta(q', \varepsilon) = q'$ appartient à l'ensemble F des états finals. Le premier axiome de respect des états finals est satisfait par la congruence de Nerode.

Enfin, si $q \sim q'$, posons $q_a = \delta(q, a)$ et $q'_a = \delta(q', a)$ pour toute lettre $a \in A$. Alors pour tout mot w de la forme $w = aw'$, on a $\delta(q, aw') \in F$ si et seulement si $\delta(q', aw') \in F$, ce qui signifie $\delta(q_a, w') \in F$ si et seulement si $\delta(q'_a, w') \in F$ puisque $\delta(q, aw') = \delta(\delta(q, a), w') = \delta(q_a, w')$ et une relation analogue pour q'. On a donc $q_a \sim q'_a$ et l'axiome de compatibilité des transitions est bien satisfait.

- Théorème de l'automate minimal

On note $\sim$ la congruence de Nerode pour l'automate fini déterministe $\mathscr{A} = (Q, A, T, \{i\}, F)$. Alors l'automate quotient $\mathscr{A}/\sim$ est équivalent à l'automate minimal du langage $\mathscr{L}(\mathscr{A})$.

- Quotients à gauche et langages de départ

Soit $\mathscr{A} = (Q, A, T, \{i\}, F)$ un automate fini déterministe. Pour $q \in Q$, on note X_q le langage de départ associé à l'état q: $X_q = \{w \in A^*, \delta(q, w) \in F\}$. Si on note $\sim$ la congruence de Nerode, on a l'équivalence $q \sim q'$ si et seulement si $X_q = X_{q'}$.
En pratique, la détermination de l'automate minimal par la méthode des congruences de Nerode consiste en un processus que l'on peut décrire comme suit : écrire les équations de départ, enlever les puits (les flèches inutiles !), regrouper les états qui ont le même langage de départ et veiller à laisser bien séparés au cours de l'ensemble de ce processus les états finals des autres états.

- Exemple 1

Le système des équations de départ du premier exemple proposé dans cette leçon s'écrit
$X_1 = aX_2 + bX_4 + \varepsilon$, $X_2 = aX_3 + bX_5$, $X_3 = aX_2 + bX_4 + \varepsilon$, $X_4 = aX_2 + bX_4 + \varepsilon$,
$X_5 = (a+b)X_5$. On voit immédiatement à l'aide du lemme d'Arden que $X_5 = \emptyset$. Surtout, on a à la lecture des équations $X_1 = X_3 = X_4$ et on peut regrouper les états ①, ③ et ④ pour construire l'automate minimal. On en déduit un simple système de deux équations : $X_1 = aX_2 + bX_1 + \varepsilon$, $X_2 = aX_1$. On retrouve bien l'automate décrit à la figure au bas de la page 3. On en déduit $X_1 = (a^2 + b)X_1 + \varepsilon$ et $X_1 = (a^2 + b)^* \varepsilon = (a^2 + b)^*$ compte tenu du lemme d'Arden. Enfin, le langage des mots acceptés par l'automate s'écrit $L = X_1 = (a^2 + b)^*$.

- Exemple 2

Les équations de départ de ce second automate (voir la première page, figure de droite), prennent la forme algébrique suivante : $X_1 = aX_2 + bX_4$, $X_2 = aX_3 + bX_7$, $X_3 = aX_7 + bX_2 + \varepsilon$,
$X_4 = aX_5 + bX_7 + \varepsilon$, $X_5 = aX_7 + bX_6 + \varepsilon$, $X_6 = aX_5 + bX_7$, $X_7 = (a+b)X_7$. On en déduit que $X_7 = \emptyset$ et on reporte cette valeur dans les autres équations. On a donc $X_1 = aX_2 + bX_4$,
$X_2 = aX_3$, $X_3 = bX_2 + \varepsilon$, $X_4 = aX_5 + \varepsilon$, $X_5 = bX_6 + \varepsilon$, $X_6 = aX_5$ et on en déduit
$X_3 = (ba)X_3 + \varepsilon$, $X_5 = (ba)X_5 + \varepsilon$. Les deux langages X_3 et X_5 sont solutions de la même équation (qui se résout à l'aide du lemme d'Arden). Donc ils sont égaux : $X_3 = X_5$. On en tire l'égalité $X_2 = X_6$ et on a réduit la taille du système à résoudre à quatre équations :
$X_1 = aX_2 + bX_4$, $X_2 = aX_3$, $X_3 = (ba)X_3 + \varepsilon$, $X_4 = aX_3 + \varepsilon$. Ces quatre langages sont différents : $X_3 = (ba)^*$, $X_2 = a(ba)^*$, $X_4 = a(ba)^* + \varepsilon$, $X_1 = a^2(ba)^* + ba(ba)^* + b$. On en déduit $L = X_1 = (a+b)a(ba)^* + b$. L'automate réduit présenté à la figure du bas à droite de la page 4 ne peut plus être réduit ; c'est l'automate minimal recherché.

Exercices

• Réduction d'un automate

On s'intéresse à l'automate décrit à la figure ci-dessous.

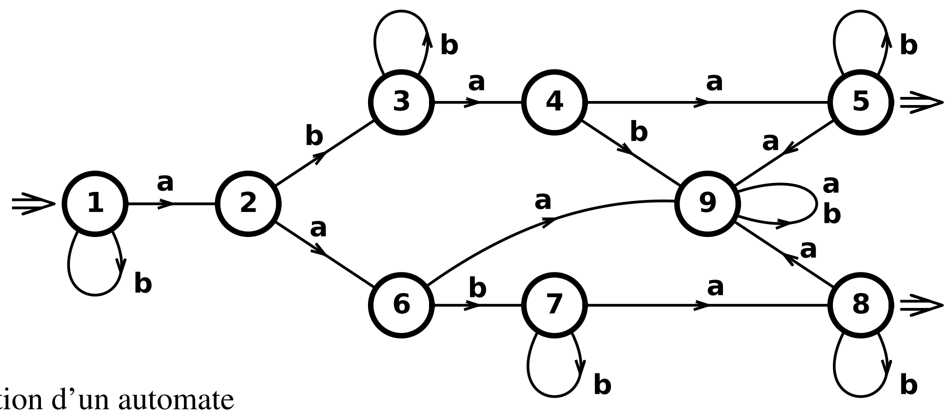
a) Écrire les équations de départ de cet automate.

b) Comparer les langages X_5 et X_8.

c) Comparer les langages X_4 et X_7.

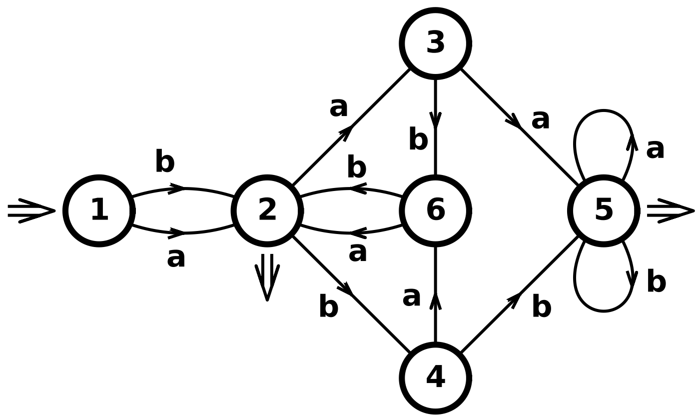
d) Achever la résolution des équations de départ et déterminer le langage L des mots acceptés par cet automate.

e) Dessiner le graphe de l'automate minimal associé aux équations de départ.



• Réduction d'un automate

On considère l'automate $\mathscr{A}$ décrit à la figure suivante



a) Écrire les équations de départ de cet automate.

b) Comparer les langages X_1 et X_6.

c) Construire l'automate réduit associé.

d) Quelles sont les équations de départ de cet automate réduit ?

e) Donner la liste de tous les quotients à gauche du langage $L = \mathscr{L}(\mathscr{A})$ des mots acceptés par l'automate. Cette question ne demande pas de longs calculs.

f) Vérifier avec des mots de petites longueur qu'il y a cinq quotients à gauche différents.

g) Déterminer le langage L des mots acceptés par cet automate.