le cnam

Codes et Automates finis

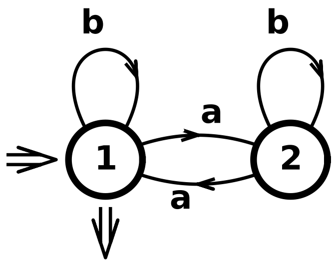
Cours 14 Transition spontanée

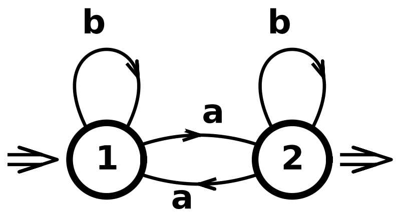
- Complémentarité

Soit $L \subset A^*$ un langage sur un alphabet fini A. Le langage complémentaire L^c est par définition formé des mots de A^* qui n'appartiennent pas à L: $L^c = A^* \backslash L = \{w \in A^*, w \notin L\}$.
On a la propriété suivante. Si le langage L est rationnel, alors le langage complémentaire L^c est rationnel.
La preuve consiste à considérer un automate fini déterministe $\mathscr{A} = (Q, A, T, I, F)$ qui accepte le langage L: $\mathscr{L}(\mathscr{A}) = L$. On considère alors un nouvel automate fini déterministe $\mathscr{A}' = (Q, A, T, I, F')$ avec les mêmes états, le même alphabet, les mêmes transitions et le même ensemble d'états initiaux. Seul l'ensemble des états finals est modifié et on choisit $F' = Q \backslash F = \{q \in Q, q \notin F\}$. On échange simplement les états acceptants et les états non acceptants. Ainsi, un chemin acceptant pour l'automate $\mathscr{A}$ part d'un état de I et aboutit à un état de F devient non acceptant pour $\mathscr{A}'$; réciproquement, un chemin non acceptant pour l'automate $\mathscr{A}$ qui part d'un état initial $i \in I$ et n'aboutit pas à un état final de F devient acceptant pour l'automate $\mathscr{A}'$ car F' est formé des états de l'ensemble Q qui n'appartiennent pas à F. $\square$

- Exemple de langage complémentaire

Nous considérons l'automate de la figure ci-dessous à gauche. Nous l'avons déjà rencontré lors d'une leçon précédente ; le langage des mots acceptés est égal à $L = (ab^*a + b)^*$ des mots qui contiennent un nombre pair de fois la lettre "a". Bien sûr, le langage complémentaire L^c est l'ensemble des mots qui contiennent un nombre impair de fois la lettre "a". L'automate $\mathscr{A}'$ décrit dans la partie droite de la figure ci-dessous s'obtient en transformant l'état ② en état de sortie et l'état ① en état non acceptant.





Nous pouvons vérifier à l'aide des équations de départ que le langage accepté par l'automate $\mathscr{A}'$ décrit dans la figure de droite est formé des mots qui comportent un nombre impair de fois la lettre "a".
Les équations de départ s'écrivent $X_1 = aX_2 + bX_1$, $X_2 = aX_1 + bX_2 + \varepsilon$. De plus $\mathscr{L}(\mathscr{A}') = X_1$.
Le lemme d'Arden appliqué à la seconde équation implique $X_2 = b^*(aX_1 + \varepsilon)$. On reporte cette

François Dubois, mai 2024.

expression dans la première équation : $X_1 = ab^*(aX_1+\varepsilon)+bX_1 = (ab^*a+b)X_1+ab^*$. On en déduit (lemme d'Arden) : $X_1 = (ab^*a+b)^*\,ab^*$, donc $\mathscr{L}(\mathscr{A}') = X_1 = L^{\mathrm{c}} = (ab^*a+b)^*\,ab^*$. Comme le langage initial L est formé des mots qui contiennent un nombre pair de fois la lettre "a", le nouveau langage $L^{\mathrm{c}} = \mathscr{L}(\mathscr{A}')$ est bien formé des mots qui contiennent un nombre impair de fois la lettre "a".

- Intersection de deux langages rationnels

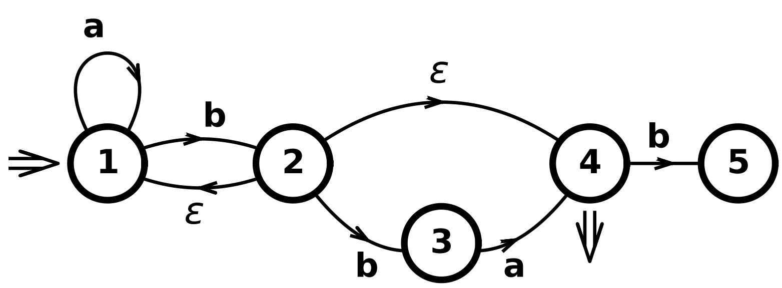
Si L_1 et L_2 sont deux langages rationnels, alors l'intersection $L_1 \cap L_2$ est un langage rationnel. Il suffit d'écrire $L_1 \cap L_2 = (L_1^{\mathrm{c}} \cup L_2^{\mathrm{c}})^{\mathrm{c}}$ et d'utiliser d'une part le fait que la somme, c'est à dire la réunion, de deux langages rationnels est un langage rationnel et d'autre part que le complémentaire d'un langage rationnel est un langage rationnel.

- Le mot sans lettre pour les automates finis déterministes ou non déterministes

Pour un automate déterministe, une transition $(p,\,a,\,q) \in T$ avec $p \in Q$, $a \in A$ et $q \in Q$ est définie à l'aide d'une application $\delta\colon\ Q \times A \longrightarrow Q$ et $q = \delta(p,\,a)$.
Une convention très utile consiste à écrire $p = \delta(p,\,\varepsilon)$ et on étend de cette façon l'application δ au mot sans lettre d'abord puis à tout mot $m \in A^*$ ensuite.
Les équations de départ s'écrivent $X_p = \sum_{a\in A,\,q=\delta(p,a)} aX_q$ si $p \notin F$ et
$X_p = \sum_{a\in A,\,q=\delta(p,a)} aX_q + \varepsilon$ si $p \in F$. Le mot sans lettre ε permet à l'automate de rejoindre un état de sortie si on est déjà positionné sur un tel état final !
Pour un automate non déterministe, une transition $(p,\,a,\,q) \in T$ avec $p \in Q$, $a \in A$
et $q \in Q$ n'obéit à aucune contrainte. La convention précédente avec le mot sans lettre consiste à écrire maintenant $(p,\,\varepsilon,\,p) \in T$ pour tout état $p \in Q$. Le mot sans lettre ne change pas l'état courant $p \in Q$ de l'automate. Comme pour les automates déterministes, cette convention permet d'écrire les équations de départ : $X_p = \sum_{q\in Q,\,a\in A,\,(p,a,q)\in T} aX_q$ si $p \notin F$ et
$X_p = \sum_{q\in Q,\,a\in A,\,(p,a,q)\in T} aX_q + \varepsilon$ si $p \in F$.

- Automates finis à transition spontanée, ou "avec ε-transition"

Il s'agit d'une généralisation de la notion d'automate fini non déterministe. Le mot sans lettre ε permet maintenant également une transition réelle entre deux états p et q différents :
$T \subset Q \times (A \cup \{\varepsilon\}) \times Q$. On peut sauter de l'état p à l'état q sans qu'il y ait besoin de lire la moindre lettre si on suppose $(p,\,\varepsilon,\,q) \in T$!



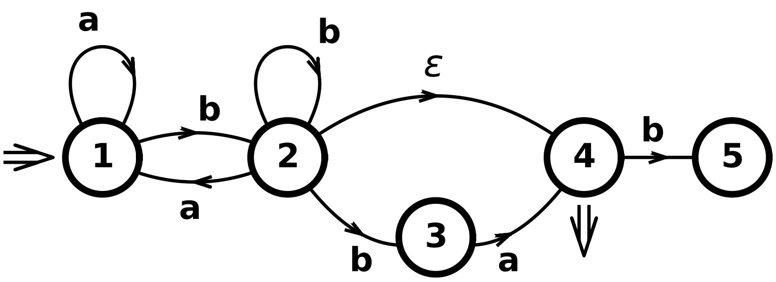
Dans l'exemple de la figure ci-dessus, le mot d'une seule lettre b par exemple est un mot accepté par l'automate : il suffit d'écrire $b = b\varepsilon$ et de suivre les flèches du graphe ci-dessus. On a les transitions ①$\xrightarrow{b}$② puis ②$\xrightarrow{\varepsilon}$④.

- Algorithme de fermeture arrière pour l'élimination du mot sans lettre

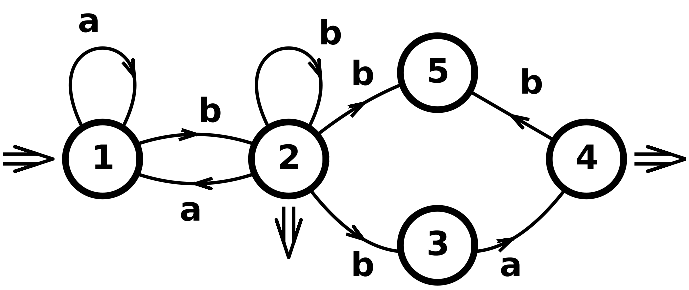
On remplace une transition spontanée suivie par un chemin de longueur unité par une nouvelle transition : pour tout $a \in A$, (p, ε, q) et (q, a, r) donnent naissance à la transition (p, a, r) ; on élimine ensuite la transition spontanée (p, ε, q).

De plus, pour chaque transition (p, ε, q) d'un état $p \in Q$ à un état terminal $q \in F$, on ajoute l'état p à l'ensemble des états terminaux.

Pour l'exemple ci-desus, on a deux transitions contenant le mot sans lettre : $(2, \varepsilon, 1)$ et $(2, \varepsilon, 4)$. Le chemin $(2, \varepsilon, 1)$ suivi de $(1, a, 1)$ est équivalent au chemin $(2, a, 1)$. De plus, le chemin $(2, \varepsilon, 1)$ suivi de $(1, b, 2)$ est équivalent à la boucle $(2, b, 2)$. On garde cette transition et on élimine la transition $(2, \varepsilon, 1)$. On obtient ainsi un nouvel automate qui ne contient plus qu'un seul mot sans lettre parmi les transitions autorisées ; il est proposé à la figure ci-dessous.



On réitère le processus avec ce nouvel automate. De la transition $(2, \varepsilon, 4)$, on aboutit à l'état ④ qui est un état de sortie. On peut donc considérer que l'état ② est un état de sortie. Enfin, on remplace les deux transitions $②\xrightarrow{\varepsilon}④$ suivi de $④\xrightarrow{b}⑤$ par l'unique transition $②\xrightarrow{b}⑤$. On obtient après ces modifications un automate fini non déterministe décrit à la figure ci-dessous.



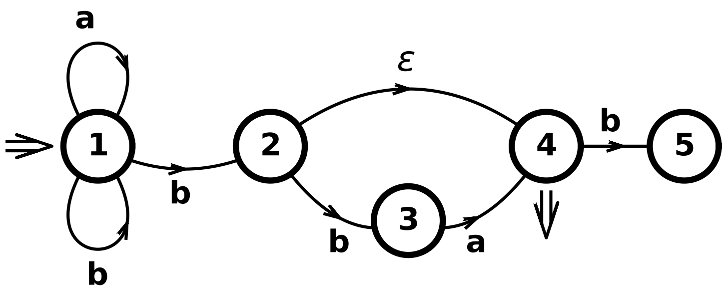
On peut chercher le langage des mots acceptés par l'automate de la figure ci-dessus avec les méthodes usuelles. On a $L = X_1$ avec $X_1 = aX_1 + bX_2$, $X_2 = aX_1 + bX_2 + bX_3 + bX_5 + \varepsilon$,
$X_3 = aX_4$, $X_4 = bX_5 + \varepsilon$, $X_5 = \emptyset$. Donc $X_4 = \varepsilon$, $X_3 = a$, $X_1 = aX_1 + bX_2$,
$X_2 = aX_1 + bX_2 + ba + \varepsilon$. Donc $X_2 = X_1 + ba + \varepsilon$ et $X_1 = aX_1 + b(X_1 + ba + \varepsilon)$,
$X_1 = (a+b)X_1 + b^2a + b$ et finalement $X_1 = (a+b)^*(b^2a+b) = L$.

En conclusion, l'algorithme de fermeture arrière ne modifie pas le langage des mots acceptés par l'automate.

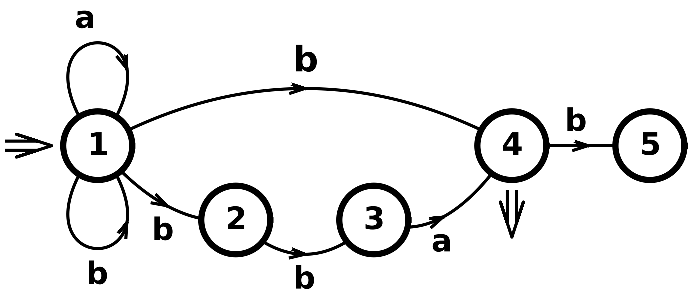
• Algorithme de fermeture avant pour l'élimination du mot sans lettre

On remplace maintenant chaque chemin de longueur unité qui se termine par le mot sans lettre par une nouvelle transition qui décrit ce chemin. De plus, pour chaque transition (p, ε, q) d'un état initial $p \in I$ à un état $q \in Q$, on ajoute l'état q à l'ensemble des états initiaux.

Pour l'exemple précédent décrit à la figure de la page 2, on remplace la transition ①$\xrightarrow{b}$② suivie de ②$\xrightarrow{\varepsilon}$① par ①$\xrightarrow{b}$①. On obtient ainsi l'automate présenté à la figure ci-dessous.



Puis on élimine le second mot sans lettre en remplaçant les transitions ①$\xrightarrow{b}$② et ②$\xrightarrow{\varepsilon}$④ par l'unique transition ①$\xrightarrow{b}$④. Cette transformation permet de créer l'automate proposé à la figure ci-dessous.



Le graphe de ce nouvel automate non déterministe est assez différent de l'automate non déterministe obtenu avec l'algorithme de fermeture arrière et décrit au milieu de la page 3. Nous allons vérifier que le langage $\widetilde{L}$ des mots acceptés par l'automate ci-dessus est bien égal au langage L calculé page 3. On a $\widetilde{L} = X_1 = (a+b)X_1 + bX_2 + bX_4$, $X_2 = bX_3$, $X_3 = aX_4$, $X_4 = bX_5 + \varepsilon$, $X_5 = \emptyset$. Donc $X_4 = \varepsilon$, $X_3 = a$, $X_2 = ba$, $X_1 = (a+b)X_1 + bba + b$ et comme plus haut $X_1 = (a+b)^* \, (b^2 a + b) = \widetilde{L}$ grâce au lemme d'Arden. Nous notons que l'algorithme de fermeture avant ne modifie pas le langage des mots acceptés par l'automate.

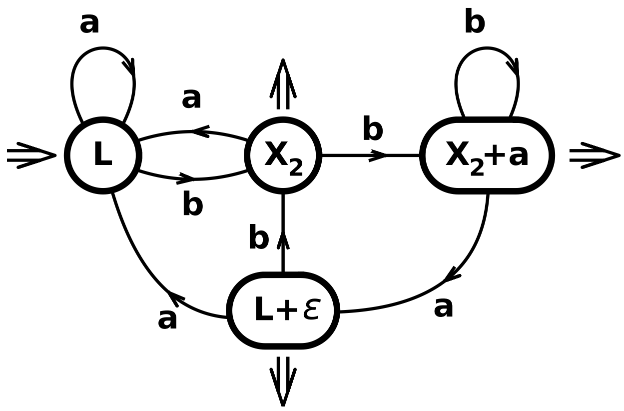
On a le résultat très général : il existe un algorithme d'élimination du mot sans lettre pour un automate fini avec transition spontanée.

• Théorème

Le langage des mots acceptés par un automate fini avec transition spontanée est un langage rationnel.

Nous illustrons ce dernier résultat avec l'exemple présenté dans cette leçon ; nous calculons "directement" un automate fini déterministe $\mathscr{A}$ tel que $\mathscr{L}(\mathscr{A}) = L = \widetilde{L}$. Les équations de départ de l'automate avec transition spontanée s'écrivent : $X_1 = aX_1 + bX_2$, $X_2 = X_1 + X_4 + bX_3$, $X_3 = aX_4$, $X_4 = bX_5 + \varepsilon$, $X_5 = \emptyset$. Donc $X_4 = \varepsilon$, $X_3 = a$, $X_2 = X_1 + ba + \varepsilon$,
$X_1 = aX_1 + b\,(X_1 + ba + \varepsilon) = (a+b)X_1 + b + bba$ et compte tenu du lemme d'Arden,

$X_1 = L = (a+b)^*(b+b^2a)$.
Nous en d'eduisons les premiers quotients à gauche : $a^{-1}L = L$, $b^{-1}L = X_2 = L+ba+\varepsilon$, $a^{-1}X_2 = a^{-1}L = L$, $b^{-1}X_2 = b^{-1}L+a = L+ba+a+\varepsilon = X_2+a$, $a^{-1}(X_2+a) = a^{-1}X_2+\varepsilon$, $a^{-1}(X_2+a) = L+\varepsilon$, $b^{-1}(X_2+a) = b^{-1}X_2+b^{-1}a = b^{-1}X_2 = X_2+a$, $a^{-1}(L+\varepsilon) = a^{-1}L = L$, $b^{-1}(L+\varepsilon) = b^{-1}L = X_2$. On dispose donc de quatre quotients à gauche : L, X_2, X_2+a et $L+\varepsilon$. Ils sont bien tous différents. Les états de sortie F sont déterminés par les conditions de l'automate minimal : $F = \{w^{-1}L, w \in L\}$. Comme $b \in L$ et $b^{-1}L = X_2$, on a $X_2 \in F$. De même, $b^2a \in L$ et $a^{-1}(b^{-1}(b^{-1}L))) = a^{-1}(X_2+a) = L+\varepsilon$ donc $L+\varepsilon \in F$. De plus, $b^2 \in L$ et $b^{-1}(b^{-1}L)) = b^{-1}X_2 = X_2+a$ et $X_2+a \in F$; tous les états autres que l'état initial L sont des états finals pour cet automate. Il est représenté à la figure ci-dessous.



Exercice

• Automate non déterministe avec transition spontanée

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

a) Quels sont les mots de longueur ≤ 4 acceptés par cet automate ?

b) Proposer un automate non déterministe équivalent en utilisant l'algorithme de fermeture arrière.

c) Proposer un automate non déterministe équivalent avec l'algorithme de fermeture avant.

d) En déduire un automate fini déterministe équivalent aux trois automates considérés plus haut.