TD 3 de Langages Formels

MIM2, 2001-2002

Laure Danthony

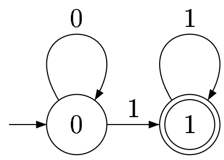
Eléments de correction

11 octobre 2001

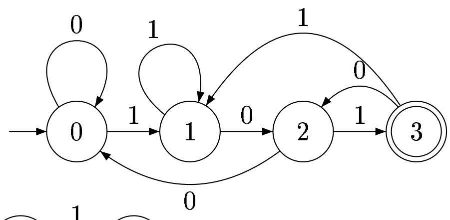
1 Automates à partir de langages

REMARQUE 1 En toute rigueur il faudrait montrer les égalités $L_i = \mathcal{L}(\mathcal{A}_i)$

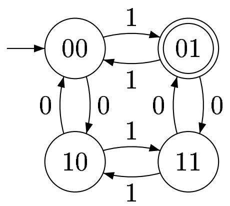
- Pour L_1 :



- Pour L_2 :



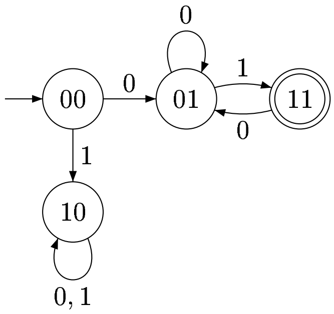
- Pour L_3 :



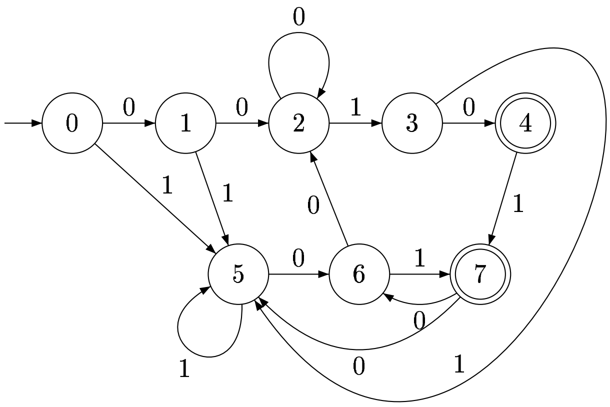
REMARQUE 2 L'état (i, j) est tel que i est le nombre de 0 déjà lus *mod* 2 et j est le nombre de 1 déjà lus *mod* 2. On montre alors facilement par récurrence que cet automate reconnaît bien le langage désiré.

2 Automates minimaux à partir de langages

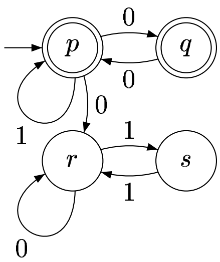
- On obtient facilement l'automate suivant, et si on essaie de le minimiser, on tombe sur un automate à 4 états. Il est donc minimal :



- On écrit l'automate non déterministe qui réalise un "ou" des automates qui reconnaissent les mots finissant par 101 et 0010. Ensuite on déterminise et on obtient l'automate suivant. Si on essaie de minimiser, tous les états sont séparés, donc l'automate est minimal :



- On obtient facilement l'automate non déterministe suivant :



On montre que cet automate est bien un automate voulu :

$\Rightarrow$: Si $u \in L$, on écrit $u = v01w$ où $v0$ est le plus petit préfixe de u contenant un nombre impair de 0 :

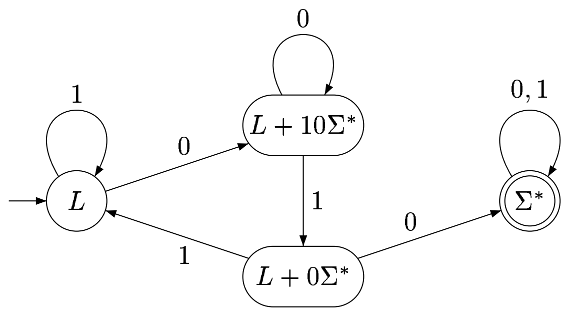
- Si cette décomposition n'existe pas, c'est que ou bien $u = v0$, on lit vit avec les états p et q, et le dernier 0 nous ramène dans l'état r, ou bien $u \in (00+1)^*$, on reconnaît le mot avec les états p et q.
- Si elle existe, on lit v avec les états p et q, avec le 0 on passe à l'état r, puis on lit $1w$ avec r et s.

$\Leftarrow$ Soit $u \in L(\mathcal{A})$.

- Si la lecture de u se termine dans l'état p, c'est que $u \in (00+1)^* \subseteq L$.
- Si la lecture de u se termine dans l'état r, c'est que u s'écrit $v0w$ avec $v \in (00+1)^*$ et $w \in (11+0)^*$ (donc $0w \in (11+0)^*$ aussi), et finalement $u \in L$.

3 Résiduels

On obtient avec la méthode du cours l'automate suivant :



Cet automate étant complétement spécifié, on en déduit immédiatement l'automate complémentaire en changeant les états non finaux en états finaux et vice-versa.

4 Automate $\mathcal{A}_i$

On définit $Q = \{u, |u| \leq i\}$. Ensuite, $q_0 = \varepsilon$, $q_f = \{\text{mots de i lettres dont la } 1^\circ \text{ est un } 0\}$, et la fonction de transition : $\delta(q,a) = \begin{cases} ua & \text{si } q = u, \text{ avec } |u| < i \\ \text{les i dernières lettres de } ua & \text{sinon} \end{cases}$

On vérifie ensuite qu'il fait bien le travail demandé.

5 Langages L' et L''

- L' : l'idée est d'avoir une case mémoire, *i.e.* $Q' = Q \cup Q \times \Sigma^*$. On définit $q'(\overline{q}, a)$ de la façon suivante : $\delta(\overline{q}, a) = \begin{cases} \delta(q, ba) & \text{si } \overline{q} = (q, b) \\ (q, a) & \text{sinon} \end{cases}$.
  On a aussi $F' = F$, et on se convainc que ça marche !
- L'' : on crée l'automate non déterministe suivant, que l'on déterminise ensuite : $\mathcal{A}'' = (Q'', \Sigma, \delta'', q_0'', F'')$ avec $q_0'' = q_0$, $Q'' = Q$ et $\delta''(q,a) = \{\delta(p, ab), b \in \Sigma\}$, et $F'' = F$. On vérifie ensuite que ça marche.

6 Langage $K^{-1}L$

On rappelle que $K^{-1}L = \bigcup_{u \in K} u^{-1}L = \{z \in \Sigma^*, \exists u \in K, uz \in L\}$.

Deux solutions :

- Soit $\mathcal{M} = (Q, \Sigma, \delta, q_0, F)$ un automate non déterministe reconnaissant L. Construisons un automate déterministe $\mathcal{M}' = (Q', \Sigma, \delta', q_0', F')$ reconnaissant $K^{-1}L$. On prend $Q' = \mathcal{P}(Q)$, $q_0' = \bigcup_{v \in K} \delta(q_0, v)$, $\delta'(X, a) = \bigcup_{q \in X} \delta(q, a)$, et $F' = \{X \subseteq Q, X \cap F = \emptyset\}$.
- Comme L est rationnel, le résiduel à gauche $u^{-1}L$ est rationnel (prendre le même automate avec $q_0' = \delta(q_0, u)$), et comme $K^{-1}L = \bigcup_{u \in K} u^{-1}L$, et que c'est une union finie car L est rationnel, alors $K^{-1}L$ est rationnel.