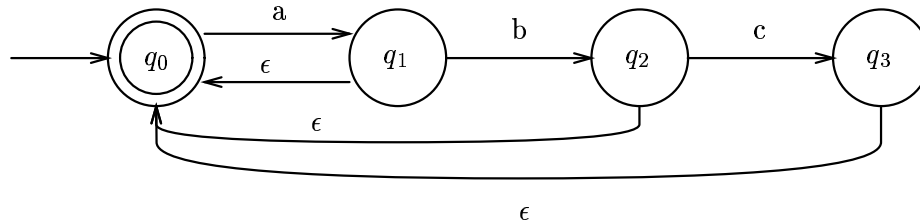


Eléments d' informatique théorique

Série 3

Exercice 3.1 Construire un automate déterministe équivalent à l'automate:



Définition 3.1 Automate minimal

Un AF déterministe $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ est *minimal* si et seulement si pour tout AF déterministe $M' = \langle Q', \Sigma, \delta', q'_0, F' \rangle$ on a $L(M) = L(M')$ implique $|Q| \leq |Q'|$.

Théorème 3.1 Unicité d'automate minimal

Soit $\mathcal{M}$ l'ensemble de tous les automates déterministes $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ qui reconnaissent le même langage $L \subseteq \Sigma^*$. Soit $M' = \langle Q', \Sigma', \delta', q'_0, F' \rangle$ un automate déterministe minimal dans $\mathcal{M}$. Alors M' est unique à isomorphisme près (renommage des états).

Définition 3.2 Algorithme de construction d'un automate minimal

1. Eliminer les états inutiles. Un état est inutile s'il n'y a pas de mot m tel que $\delta^*(q_0, m) = q$. L'état q est aussi inutile s'il n'y a pas de mot n tel que $\delta^*(q, n) \in F$.
2. Trouver les classes d'équivalence d'états selon la relation R suivante:

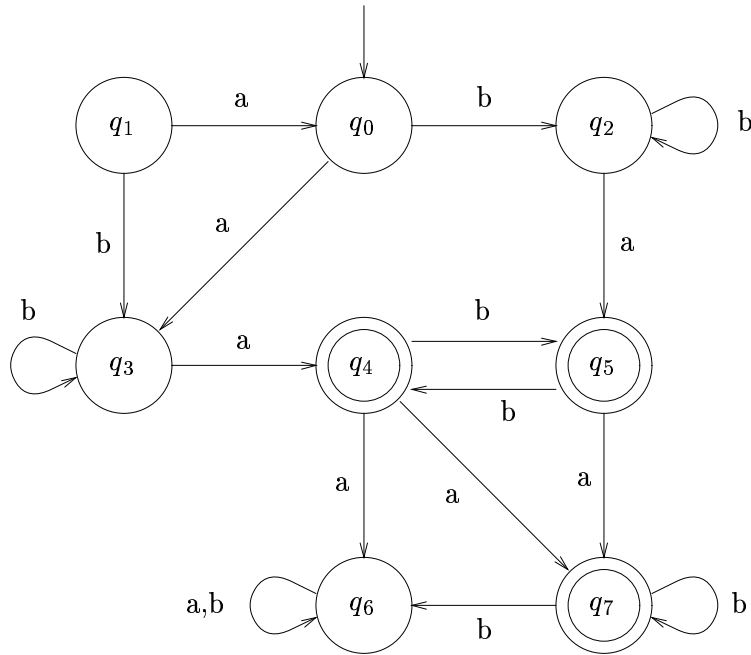
la relation $R(q, q')$ est vraie si et seulement si pour tout mot $m \in \Sigma^*$

- soit $\delta^*(q, m) \in F$ et $\delta^*(q', m) \in F$
- soit $\delta^*(q, m) \notin F$ et $\delta^*(q', m) \notin F$

Il suffit en fait de ne considérer que les mots m avec $|m| \leq |Q|$.

3. Définir: $Q' = \{[q]\}$ (c.à.d les classes d'équivalence trouvées),
 $q'_0 = [q_0]$,
 $\delta'([q], a) = [\delta(q, a)]$,
 $F' = \{[q] \mid q \in F\}$ et $\Sigma' = \Sigma$.

Exercice 3.2 Minimaliser l'automate :



Exercice 3.3

Soient L_1 et L_2 les langages décidés par M_1 et M_2 respectivement. Construire les automates acceptant $L_1 L_2$, $L_1 \cup L_2$, $\bar{L}_1$, $\bar{L}_2$ et $L_1 \cap L_2$.

