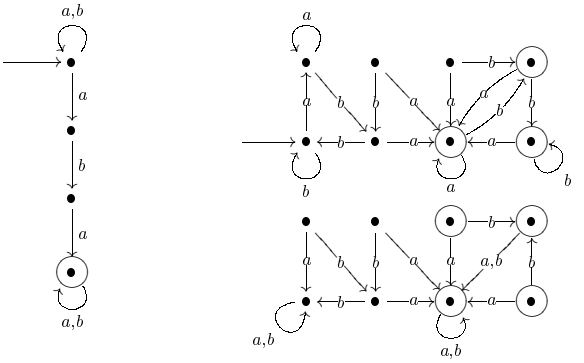
Automates et Calculabilité

– Minimisation –

Leçon 8 : 8 dec 2003

Uwe Nestmann

EPFL-I&C



Motivation : Minimisation des AFDs

Soit $M = (\mathcal{Q}, \Sigma, \delta, s, F)$ un AFD.

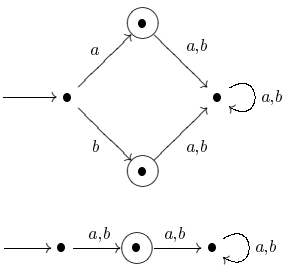
Nous cherchons l'automate ***minimal***
— par rapport au ***nombre d'états*** —
qui accepte le même langage que M.

Question : existe-t-il toujours un unique automate minimal ?

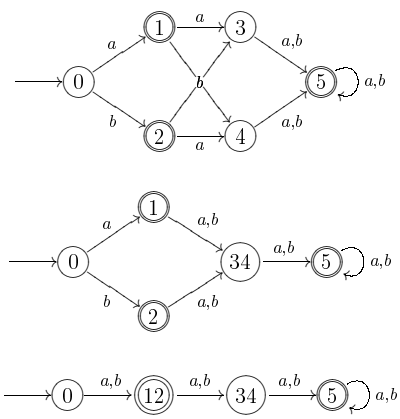
La minimisation se déroule en deux étapes :

1. Suppression des ***états inaccessibles*** depuis l'état initial.
   L'état $q \in \mathcal{Q}$ est inaccessible (à partir de s)
   s'il n'existe pas de mot $x \in \Sigma$ tel que $\widehat{\delta}(s, x) = q$.
2. Fusion des ***états équivalents*** . . .

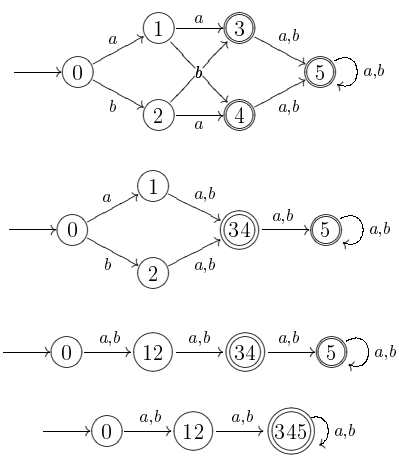
Exemple I



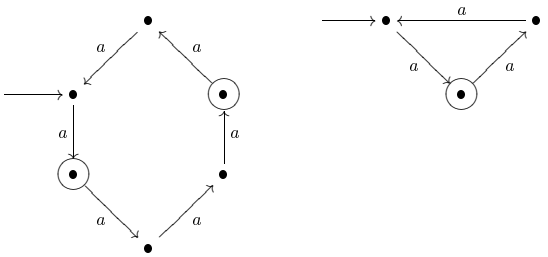
Exemple II



Exemple III



Exemple IV



Observations semi-formelles

Soit $M = (\mathcal{Q}, \Sigma, \delta, s, F)$ un AFD.

Nous notons p "$\approx$" q si nous pensons de manière informelle pouvoir fusionner deux états p et q, et p "$\not\approx$" q si nous pensons que cela n'est pas possible.

Soit $p = \widehat{\delta}(s, x)$ et $q = \widehat{\delta}(s, y)$,

- Si $p \in F$ et $q \notin F$, alors p "$\not\approx$" q.
- Si p "$\approx$" q, alors $\widehat{\delta}(s, xa)$ "$\approx$" $\widehat{\delta}(s, ya)$.

S'il existe z tel que $\widehat{\delta}(p, z) \in F$ et $\widehat{\delta}(q, z) \notin F$, alors p "$\not\approx$" q.

Idée : p "$\approx$" q ssi $(\forall z \in \Sigma^*)(\widehat{\delta}(p, z) \in F \Leftrightarrow \widehat{\delta}(q, z) \in F)$

Définition:
Toute équivalence $\equiv$ sur un ensemble X détermine un ***quotient*** $X/\equiv$ défini par ses ***classes d'équivalence*** :

$$[p]_\equiv \stackrel{\text{def}}{=} \{\, q \in X \mid q \equiv p \,\} \qquad X/\equiv \stackrel{\text{def}}{=} \{\, [p]_\equiv \mid p \in X \,\}$$

Lorsqu'il n'y a pas d'ambiguïté on écrit $[p]$ à la place de $[p]_\equiv$.

Exemple: Ecrire le quotient $\mathbb{N}_{\equiv_3}$ qui est déterminé par la relation $\equiv_3$ (égalité modulo 3) sur $\mathbb{N}$.

Lemme:
Soit $\equiv$ une équivalence sur un ensemble X.

- $p \equiv q$ ssi $[p] = [q]$.
- Si $[p] \neq [q]$, alors $p \notin [q]$ et $q \notin [p]$.
- $X = \bigcup_{p \in X} [p]$.

Propriétés du quotient (I)

Lemme:
Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD.
Soit $M/\approx = (Q/\approx, \Sigma, \delta_\approx, [s], F_\approx)$.

1. Pour tout $p \in Q$: $p \in F \iff [p] \in F_\approx$.
2. Pour tout $p \in Q$, et pour tout $x \in \Sigma^*$: $\widehat{\delta_\approx}([p], x) = [\widehat{\delta}(p, x)]$.

Preuves : (Idées)

1. $\Rightarrow$ Immédiate par définition de $F_\approx$.
   $\Leftarrow$ Observer que $p \notin F$ et $p \approx q$ implique que $q \notin F$.
   Alors, si $p \notin F$ il n'existe pas de q tel que $p \approx q$ et $[q] \in F_\approx$.
2. Par induction sur x :
   - $\widehat{\delta_\approx}([p], \epsilon) = [\widehat{\delta}(p, \epsilon)]$.
   - $\widehat{\delta_\approx}([p], xa) = [\widehat{\delta}(p, xa)]$.

Propriétés du quotient (III)

Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD.
Soit $M/\approx = (Q/\approx, \Sigma, \delta_\approx, [s], F_\approx)$, qui est aussi un AFD.
Nous définissons « le quotient du quotient » par :

$$[p] \sim [q] \text{ ssi } \forall x \in \Sigma^* : \big(\widehat{\delta_\approx}([p], x) \in F_\approx \iff \widehat{\delta_\approx}([q], x) \in F_\approx \big)$$

$\sim$ est une relation définie sur l'ensemble des $\approx$-classes de Q.

$$\begin{array}{lll}
[p] \sim [q] & & \\
\Rightarrow \quad \forall x : \big(\widehat{\delta_\approx}([p], x) \in F_\approx \iff \widehat{\delta_\approx}([q], x) \in F_\approx \big) & & \text{déf. de } \sim \\
\Rightarrow \quad \forall x : \big([\widehat{\delta}(p, x)] \in F_\approx \iff [\widehat{\delta}(q, x)] \in F_\approx \big) & & \text{Lemme p.11 (2)} \\
\Rightarrow \quad \forall x : \big(\widehat{\delta}(p, x) \in F \iff \widehat{\delta}(q, x) \in F \big) & & \text{Lemme p.11 (1)} \\
\Rightarrow \quad p \approx q \;\Rightarrow\; [p] = [q] & &
\end{array}$$

Il suffit donc d'exécuter la minimisation une seule fois.

Algorithme de minimisation (II)

- Il se peut que l'on soit obligé de vérifier l'application de la règle (STEP) plusieurs fois pour le même couple.
  Un prémisse manquant à un moment peut apparaître plus tard dans la procédure ...
- La procédure termine parce que ...
  -
  -
- Il nous reste à démontrer que les relations $\approx$ et $\approx_{\mathrm{m}}$ coincident. $\Rightarrow$ Exercice de la série 8 !

Définition:
Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD.
La relation binaire $\approx$ sur Q est une équivalence définie par :

$$p \approx q \text{ ssi } (\forall x \in \Sigma^*)\big(\widehat{\delta}(p, x) \in F \iff \widehat{\delta}(q, x) \in F \big)$$

L'automate $M/\approx$ est défini par $(Q/\approx, \Sigma, \delta_\approx, [s], F_\approx)$.

- $Q/\approx \stackrel{\text{def}}{=} \{\, [p] \mid p \in Q \,\}$
- $\delta_\approx([p], a) \stackrel{\text{def}}{=} [\delta(q, a)]$.
- $F_\approx \stackrel{\text{def}}{=} \{\, [q] \mid q \in F \,\}$

Lemme: La fonction $\delta_\approx$ est bien définie :

- Si $p \approx q$, alors $\delta(p, a) \approx \delta(q, a)$.
- Si $[p] = [q]$, alors $[\delta(p, a)] = [\delta(q, a)]$.

Propriétés du quotient (II)

Théorème:
Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD.
Soit $M/\approx = (Q/\approx, \Sigma, \delta_\approx, [s], F_\approx)$.
Alors,

$$L(M/\approx) = L(M).$$

Preuve :
Pour tout $x \in \Sigma^*$:

$$\begin{array}{lll}
x \in L(M/\approx) & \widehat{\delta_\approx}([s], x) \in F_\approx & \text{acceptation de } x \text{ par } M/\approx \\
\iff & [\widehat{\delta}(s, x)] \in F_\approx & \text{Lemme p.11 (2)} \\
\iff & \widehat{\delta}(s, x) \in F & \text{Lemme p.11 (1)} \\
\iff & x \in L(M) & \text{acceptation de } x \text{ par } M
\end{array}$$

Algorithme de minimisation (I)

Nous calculons la relation $\approx$ pour un AFD $M = (Q, \Sigma, \delta, s, F)$ dont tous les états sont atteignables par la procédure suivante:

1. Construire un tableau contenant une entrée pour tous les couples non-ordonnés $\{p, q\}$ avec $p, q \in Q$.
2. Marquer tous les couples $\{p, q\}$ tels que $p \in F$ et $q \notin F$.

$$\text{(INIT)}\ \frac{p \in F \qquad q \notin F}{\{p, q\} \in \surd}$$

3. Répéter (appliquer) autant que possible :

$$\text{(STEP)}\ \frac{a \in \Sigma \qquad \delta(p, a) = p' \qquad \delta(q, a) = q' \qquad \{p', q'\} \in \surd}{\{p, q\} \in \surd}$$

4. Nous définissons : $p \approx_{\mathrm{m}} q$ si $\{p, q\} \notin \surd$.

La minimisation par automate

La technique appliquée dans la procédure de minimisation est souvent utilisée dans le domaine de la vérification des programmes sous le nom de ***backwards reachability analysis*** (BRA).

Soit $M = (Q, \Sigma, \delta, s, F)$ un AFD.

Définissons un AFN $M_{\mathrm{BRA}} = (Q_{\mathrm{BRA}}, \Sigma, \Delta_{\mathrm{BRA}}, S_{\mathrm{BRA}}, F_{\mathrm{BRA}})$ par :

- $Q_{\mathrm{BRA}} \stackrel{\text{def}}{=} \{\, \{p, q\} \mid p, q \in Q \wedge p \neq q \,\}$
- $\Delta_{\mathrm{BRA}}(\{p', q'\}, a) \stackrel{\text{def}}{=} \{\, \{p, q\} \mid p' = \delta(p, a) \wedge q' = \delta(q, a) \,\}$
- $S_{\mathrm{BRA}} \stackrel{\text{def}}{=} \{\, \{p, q\} \mid p \in F \wedge q \notin F \,\}$
- $F_{\mathrm{BRA}} \stackrel{\text{def}}{=} \{\, \{p, q\} \in Q_{\mathrm{BRA}} \mid \exists x \in \Sigma^* : \{p, q\} \in \widehat{\Delta_{\mathrm{BRA}}}(S_{\mathrm{BRA}}, x) \,\}$