Langages formels et automates – cours 3

Constructions d'automates

Catalin Dima

Construction d'un automate de Moore à partir d'un automate de Mealy

- $\mathcal{A} = (Q, \Sigma, \delta, Q_0, Q_f)$ automate de Mealy.
- Comment construire (**de manière générique !**) aut. Moore $\mathcal{A}' = (Q', \Sigma, \delta', \lambda, Q_0', Q_f')$ t.q. $L(\mathcal{A}) = L(\mathcal{A}')$?
- L'idée : tirer les étiquetes des transitions vers les états source.
- **Formalisation** :

$$
\begin{aligned}
Q' &= \delta && \text{c'est tout simplement ça !} \\
\lambda(q \xrightarrow{a} r) &= r \\
\delta' &= \left\{ (q \xrightarrow{a} r, r \xrightarrow{b} s) \mid q \xrightarrow{a} r, r \xrightarrow{b} s \in \delta \right\} \\
Q_0' &= \left\{ q \xrightarrow{a} r \mid q \in Q_0 \right\} \\
Q_f' &= \left\{ q \xrightarrow{a} r \mid r \in Q_f \right\}
\end{aligned}
$$

- **Preuve** !
  - Double inclusion : $L(\mathcal{A}) \subseteq L(\mathcal{A}')$ et $L(\mathcal{A}') \subseteq L(\mathcal{A})$.

Construction d'un automate de Mealy à partir d'un automate de Moore

- $\mathcal{B} = (Q, \Sigma, \delta, \lambda, Q_0, Q_f)$ automate de Mealy.
- Comment construire $\mathcal{B} = (Q', \Sigma, \delta', Q'_0, Q'_f)$ automate de Moore avec $L(\mathcal{B}) = L(\mathcal{B}')$.
- Idée : déplacer les étiquettes des états vers toutes les transitions sortantes.
- Et bien choisir les états finaux.
- Formalisation :

$$
\begin{aligned}
Q'_f &= \{r' \mid r \in Q_f\} & \qquad & Q'_f \text{ copie disjointe de } Q_f \\
Q' &= Q \cup Q'_f \\
\delta' &= \{q \xrightarrow{a} r \mid q \in Q, r \in Q', (q, r) \in \delta, \lambda(q) = a\} \\
&\quad \cup \{q \xrightarrow{a} q' \mid q \in Q_f, q' \in Q_f \text{ étant la copie de } q, \lambda(q) = a\} \\
Q'_0 &= Q_0
\end{aligned}
$$

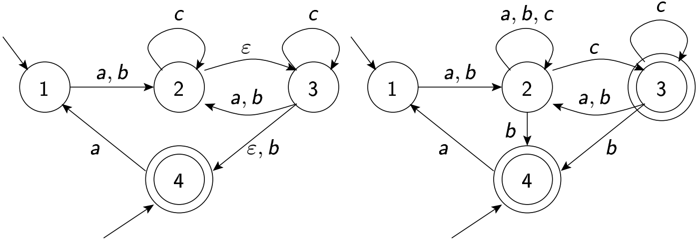
- Donc, attention, on n'a pas de transition sortante de Q'_f !

Automates avec ε-transitions

- On a vu, elles seraient bien utiles...
- Il faut étendre la définition d'automate !
- Automate (de Mealy) avec ε-transitions : $\mathcal{A} = (Q, \Sigma, \delta, Q_0, Q_f)$ avec :
  - Q, Q_0, Q_f les mêmes noms et caractéristiques que pour les automates finis normaux.
  - $\delta : Q \times \left(\Sigma \cup \{\varepsilon\}\right) \times Q$ relation de transition.
- Les notions de trajectoire et acceptation sont les mêmes.

Expressivité des ε-transitions

- Est-ce qu'utiliser des ε-transitions nous permet de représenter plus de langages ?
- Essayons :
  - On doit combiner une ε-transition avec une transition normale.



- Et si on a plusieurs ε-transitions qui s'enchaînent ?...
- On doit alors prendre toutes les séquences d'ε-transitions !

Expressivité des ε-transitions

- On a un automate avec ε-transitions : $\mathcal{A} = (Q, \Sigma, \delta, Q_0, Q_f)$.
- On veut construire un automate sans ε-transitions : $\mathcal{A}' = (Q', \Sigma, \delta', Q'_0, Q'_f)$.

$$Q' = Q, Q'_0 = Q_0$$

$$\delta' = \{q \xrightarrow{a} r \mid \exists s \in Q, \exists \rho \text{ trajectoire partant de } q$$
$$\text{et s'arrêtant en } s, \text{ composée que de } \varepsilon\text{-transitions, et } s \xrightarrow{a} r \in \delta\}$$

$$Q'_f = Q_f \cup \{q \in Q, \mid \exists r \in Q_f, \exists \rho \text{ trajectoire partant de } q$$
$$\text{et s'arrêtant en } r, \text{ composée que de } \varepsilon\text{-transitions } \}$$

- Bon, d'accord, c'est une définition utile mais... algorithmique ?
  1. Construire la fermeture transitive de la relation suivante : $R_\varepsilon = \{(q, r) \mid q \xrightarrow{\varepsilon} r \in \delta\}$.
     - Tout le monde sait faire, non ?...
  2. Alors

$$\delta' = \{q \xrightarrow{a} r \mid \exists s \in Q, (q, s) \in R_\varepsilon, s \xrightarrow{a} r \in \delta\}$$
$$Q'_f = Q_f \cup \{q \in Q, \mid \exists s \in Q_f, (q, s) \in R_\varepsilon\}$$

Constructions : union

- On nous donne deux automates : $\mathcal{A}_1 = (Q_1, \Sigma_1, \delta_1, Q_0^1, Q_f^1)$ et $\mathcal{A}_2 = (Q_2, \Sigma_2, \delta_2, Q_0^2, Q_f^2)$.
- Et on voudrait savoir si $L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$ peut être aussi reconnu par un automate fini.
- Idée : rajouter un nouvel état initial, relié aux anciens états initiaux par des ε-transitions !
- Exemple ...
- Formalisation : $\mathcal{A} = (Q, \Sigma, \delta, Q_0, Q_f)$ avec

$$Q = Q_1 \cup Q_2$$

Mais on suppose que $Q_1 \cap Q_2 = \emptyset$!

$$\begin{aligned} \Sigma &= \Sigma_1 \cup \Sigma_2 \\ Q_0 &= Q_0^1 \cup Q_0^2 \\ Q_f &= Q_f^1 \cup Q_f^2 \\ \delta &= \delta_1 \cup \delta_2 \end{aligned}$$

Concaténation

- Même situation, avec deux automates donnés :
  $\mathcal{A}_1 = (Q_i, \Sigma_i, \delta_i, Q_0^i, Q_f^i)$, $i = 1, 2$.
- On veut cette fois-ci un automate pour $L(\mathcal{A}_1) \cdot L(\mathcal{A}_2)$.
- Idée : relier par ε-transitions chaque état final de $\mathcal{A}_1$ à chaque état initial de $\mathcal{A}_2$.
- Exemple ...
- Formalisation : $\mathcal{A} = (Q, \Sigma, \delta, Q_0, Q_f)$ avec

$$\begin{aligned} Q &= Q_1 \cup Q_2 \\ \Sigma &= \Sigma_1 \cup \Sigma_2 \\ Q_0 &= Q_0^1 \\ Q_f &= Q_f^2 \\ \delta &= \delta_1 \cup \delta_2 \cup \left\{ q_1 \xrightarrow{\varepsilon} q_2 \mid q_1 \in Q_f^1, q_2 \in Q_0^2 \right\} \end{aligned}$$

Étoile

- Cette fois-ci on nous donne un seul automate : $\mathcal{A} = (Q, \Sigma, \delta, Q_0, Q_f)$.
- Et on veut construire l'automate pour $L(\mathcal{A})^*$.
- On peut s'inspirer de la concaténation, sauf qu'il faut faire attention au ε, qui appartient à $L(\mathcal{A})^*$ même s'il n'est pas dans $L(\mathcal{A})$!
- On met donc alors un nouvel état q_0 :
  - Le seul état initial et final dans $\mathcal{A}'$.
  - Source d'une ε-transition vers tout état initial.
  - Et destination d'une ε-transition provenant de tous les états finaux.
- Exemple ...
- Formalisation : $\mathcal{A}' = (Q', \Sigma, \delta', Q'_0, Q'_f)$ avec

$$
\begin{aligned}
Q' &= Q \cup \{q_0\} \text{ nouvel état, } q_0 \notin Q \\
Q'_0 &= Q'_f = \{q_0\} \\
\delta &= \delta \cup \left\{ q_0 \xrightarrow{\varepsilon} q \mid q \in Q_0 \right\} \cup \left\{ q \xrightarrow{\varepsilon} q_0 \mid q \in Q_f \right\}
\end{aligned}
$$

Intersection

- Maintenant on veut un automate pour $L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$.
  - Attention ! Il faut que les deux automates **partagent le même alphabet** !
- Idée : simuler synchronémment $\mathcal{A}_1$ et $\mathcal{A}_2$ sur le même automate :
  - L'état de $\mathcal{A}$ serait une paire d'états, l'état de $\mathcal{A}_1$ et l'état de $\mathcal{A}_2$.
  - Etats initiaux : paire d'états initiaux dans chaque automate.
  - Pareil pour les états finaux.
- Exemple :

$$\begin{aligned}L_5 = \{w \in \{a,b\}^* \mid \#_a(w) \text{ est un multiple de } 3 \\ \text{et } \#_b(w) \text{ est un multiple de } 4\}\end{aligned}$$

- Formalisation : $\mathcal{A} = (Q, \Sigma, \delta, Q_0, Q_f)$ avec

$$\begin{aligned}
Q &= Q_1 \times Q_2 \\
Q_0 &= Q_0^1 \times Q_0^2 \\
Q_f &= Q_f^1 \times Q_f^2 \\
\delta &= \{(q_1, q_2) \xrightarrow{a} (r_1, r_2) \mid q_1 \xrightarrow{a} r_1 \in \delta_1, q_2 \xrightarrow{a} r_2 \in \delta_2\}
\end{aligned}$$

Shuffle

- Et enfin, on veut construire un automate pour le shuffle $L(\mathcal{A}_1) \sqcup\!\sqcup L(\mathcal{A}_2)$.
- On doit pouvoir faire des transitions soit dans une, soit dans l'autre des automates.
- Il faut donc avoir en "mémoire" l'état de chaque automate.
  - Un peu comme pour l'intersection.
- Exemple ...
- Formalisation : $\mathcal{A} = (Q, \Sigma, \delta, Q_0, Q_f)$ avec

$$\begin{aligned}
Q &= Q_1 \times Q_2 \\
Q_0 &= Q_0^1 \times Q_0^2 \\
Q_f &= Q_f^1 \times Q_f^2 \\
\delta &= \left\{(q_1, q_2) \xrightarrow{a} (r_1, q_2) \mid q_1 \xrightarrow{a} r_1 \in \delta_1, q_2 \in Q_2\right\} \\
&\cup \left\{(q_1, q_2) \xrightarrow{a} (q_1, r_2) \mid q_1 \in Q_1, q_2 \xrightarrow{a} r_2 \in \delta_2\right\}
\end{aligned}$$