Devoir maison n°4

À rendre le lundi 10/11

⋆⋆⋆

Minimisation d'automate

Après la construction d'un automate, il est naturel de s'interroger sur la minimalité du nombre d'états. En effet, une fois l'automate construit, son exploitation et l'espace mémoire utilisé seront meilleurs s'il possède peu d'états.

Un résultat intéressant est que l'automate minimal d'un langage reconnaissable est unique, à permutation des états près. Cela permet notamment de tester l'équivalence de deux automates (c'est-à-dire l'égalité de leur langage reconnu).

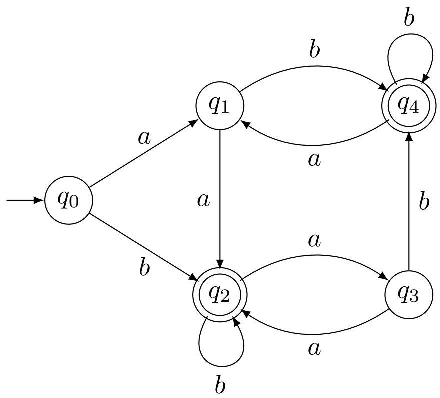
On propose dans ce devoir d'étudier deux méthodes de minimisation d'automate.

1 Algorithme de Brzozowski

1.1 Langage miroir

Pour $u = a_1 \dots a_n \in \Sigma^*$, on note $\overline{u}$ le **mot miroir** de u, défini par $\overline{u} = a_n a_{n-1} \dots a_1$. Pour $L \subseteq \Sigma^*$ un langage, on note $\overline{L} = \{\overline{u} \mid u \in L\}$ le **langage miroir** de L.

On considère l'automate A_0 :



Automate A_0

Question 1 Déterminer le langage reconnu par l'automate A_0. On donnera une expression régulière dont l'interprétation est $L(A_0)$.

Question 2 Représenter graphiquement un automate non déterministe à 5 états, qu'on notera $\overline{A_0}$, reconnaissant le langage miroir de $L(A_0)$.

Pour $A = (Q, \Sigma, \delta, q_0, F)$ un automate fini déterministe, on pose $\overline{A} = (Q, \Sigma, \Delta, I, F')$ l'automate fini non déterministe défini de la manière suivante :

- $I = F$;
- $F' = \{q_0\}$;
- pour $(p,q) \in Q^2$ et $a \in \Sigma$, $p \xrightarrow{a} q$ est une transition dans A si et seulement si $q \xrightarrow{a} p$ est une transition dans $\overline{A}$.

Question 3 Montrer que $\overline{L(A)} = L(\overline{A})$.

1.2 Déterminisation

Pour la suite, si A est un automate non déterministe, on note $\det(A)$ l'automate des parties de A où on n'a conservé que les parties accessibles comme états.

Question 4 Construire l'automate $A_1 = \det(\overline{A_0})$. On donnera la table de transition obtenue pendant la construction, et on représentera graphiquement l'automate ainsi obtenu.

Question 5 Représenter graphiquement l'automate non déterministe $\overline{A_1}$, reconnaissant le langage miroir de $L(A_1)$.

Question 6 Construire l'automate $A_2 = \det(\overline{A_1})$. Quel est le langage reconnu par A_2 ?

1.3 Algorithme de Brzozowski

L'algorithme de Brzozowski consiste, pour un automate A, à calculer l'automate $B = \det(\overline{\det(\overline{A})})$. On veut montrer dans cette partie que $L(B) = L(A)$ et que B contient un nombre minimal d'états parmi les automates déterministes complets équivalents à A.

Question 7 Avec les notations précédentes, montrer que $L(B) = L(A)$.

Pour la suite, on suppose que A est un automate déterministe dont tous les états sont accessibles. On pose $\overline{A} = (Q, \Sigma, \Delta, I, F)$ et on pose $\det(\overline{A}) = (X, \Sigma, \delta, I, F')$ le déterminisé accessible de $\overline{A}$. Attention, on note de la même manière I en tant que partie d'états de $\overline{A}$ et en tant qu'état de $\det(\overline{A})$.

Pour $u \in \Sigma^*$ et $L \subseteq \Sigma^*$, on note $u^{-1}L = \{v \in \Sigma^* \mid uv \in L\}$.

Question 8 Soit $q \in Q$ et $u \in \Sigma^*$. Montrer que si $q \in \delta^*(I, u)$, alors $u^{-1}L(\overline{A}) \neq \emptyset$.

Question 9 Montrer que si $u, v \in \Sigma^*$ vérifient $u^{-1}L(\overline{A}) = v^{-1}L(\overline{A})$, alors $\delta^*(I, u) = \delta^*(I, v)$.

Question 10 En déduire que tout automate déterministe complet reconnaissant $L(\overline{A})$ possède au moins $|X|$ états.

Question 11 Montrer la correction de l'algorithme de Brzozowski, c'est-à-dire que l'automate construit par l'algorithme contient un nombre minimal d'états parmi les automates équivalents.

Question 12 Quelle est la complexité temporelle de cet algorithme en fonction du nombre d'états de l'automate initial ? Est-il utilisable en pratique ? Justifier.

2 Algorithme de Hopcroft

2.1 Équivalence de Nerode

Définition

Soit $A = (Q, \Sigma, \delta, q_0, F)$ un AFD, et $p, q \in Q$ deux états. On dit que p et q sont **équivalents** si et seulement si pour tout $u \in \Sigma^*$:

$$\delta^*(p,u) \in F \Leftrightarrow \delta^*(q,u) \in F$$

On notera $p \sim_A q$ ou $p \sim q$ s'il n'y a pas d'ambiguïté de l'automate.

On dit que p et q sont **séparables** s'ils ne sont pas équivalents. Si $(\delta^*(p,u), \delta^*(q,u)) \in F \times \overline{F} \cup \overline{F} \times F$, on dit que u **sépare** p et q.

Question 13 Parmi les paires de sommets $\{q_1, q_4\}, \{q_0, q_1\}, \{q_2, q_4\}$ de l'automate A_0 de la partie précédente, préciser lesquelles sont séparables ou équivalentes. On donnera un mot u qui les sépare pour une paire séparable, et une justification rapide de leur équivalence pour une paire équivalente.

Soit $A = (Q, \Sigma, \delta, q_0, F)$ un AFD, $q \in Q$ et $a \in \Sigma$. On note $a^{-1}q = \{p \in Q, \delta(p,a) = q\}$. Par exemple, dans l'automate A_0, $b^{-1}q_4 = \{q_1, q_3, q_4\}$.

Question 14 On suppose que A est complet. Montrer que pour $a \in \Sigma$, $\sum\limits_{q \in Q} |a^{-1}q| = |Q|$.

On considère l'algorithme donné figure 1, appelé algorithme de séparation.

Entrée : AFD complet $(Q, \Sigma, \delta, q_0, F)$
Début algorithme
    $\Phi \leftarrow$ file vide
    $D \leftarrow \varnothing$
    Pour $(p,q) \in F \times \overline{F}$ **Faire**
        Insérer (p,q) à Φ
        $D \leftarrow D \cup \{(p,q),(q,p)\}$
    Tant que $\Phi \neq \varnothing$ **Faire**
        $(p,q) \leftarrow$ extraction de Φ
        Pour $a \in \Sigma$ **Faire**
            Pour $(p', q') \in a^{-1}p \times a^{-1}q$ **Faire**
                Si $(p', q') \notin D$ **Alors**
                    Insérer (p', q') à Φ
                    $D \leftarrow D \cup \{(p', q'), (q', p')\}$
    Renvoyer D

Figure 1 – Algorithme de séparation

Question 15 Représenter, sous forme de tableau à double entrée, l'ensemble D obtenu en appliquant l'algorithme à l'automate de l'exemple précédent. On utilisera un tableau à double entrée pour représenter D.

Question 16 Montrer qu'après l'exécution de l'algorithme, $(p,q) \in D$ si et seulement si (p,q) est séparable.

Question 17 Déterminer la complexité de l'algorithme en fonction de $|\Sigma|$ et $|Q|$. On attend une borne la plus précise possible.

2.2 Automate minimal

L'objectif de la construction de l'automate minimal est de pouvoir « fusionner » des états qui se comportent de la même manière. Pour simplifier la construction, on ne cherche à minimiser dans ce problème que des automates fini déterministes complets.

Soit $A = (Q, \Sigma, \delta, q_0, F)$ un automate fini déterministe complet. L'algorithme de minimisation est le suivant.

- on construit A' en enlevant de A les états non accessibles ;
- on utilise l'algorithme de séparation sur A' pour obtenir une partition de Q en classes d'équivalences Q_0, Q_1, ..., Q_k.

On construit un automate intermédiaire $B' = (Q'_B, \Sigma, \delta_B, Q_0, F_B)$ tel que, en notant $\overline{q}$ la classe d'équivalence de $q \in Q$:

- $Q'_B = \{Q_0, Q_1, \ldots, Q_k\}$;
- $Q_0 = \overline{q_0}$;
- $F_B = \{\overline{q} \mid q \in F, q \text{ accessible}\}$;
- $\forall i \in [\![0, k]\!], \forall a \in \Sigma$, $\delta_B(Q_i, a) = \overline{\delta(q, a)}$, où $q \in Q_i$.

Question 18 Montrer que δ_B est bien définie.

Question 19 Montrer que l'automate B' possède au plus un état non co-accessible.

Dès lors, l'automate minimal de A est l'automate B correspondant à B' dans lequel on a supprimé l'éventuel état non co-accessible.

Question 20 Montrer que A et B sont équivalents.

Question 21 Montrer que si $C = (Q_C, \Sigma, \delta_C, q_C, F_C)$ est un automate fini déterministe équivalent à A, alors $|Q_C| \geqslant |Q_B|$.

Question 22 Déterminer l'automate minimal de l'automate A_0.

Question 23 Déterminer l'automate minimal de l'automate A_3 suivant :