Université Paris Cité
Automates et Analyse Lexicale

UFR Informatique
Année 2023–2024

Handout 12

Minimisation, et autres conséquences du théorème de de Myhill-Nerode

1 Un nouveau critère pour montrer qu'un langage n'est pas rationnel

Le théorème de Myhill-Nerode nous donne une deuxième méthode, après le lemme d'itération, pour montrer qu'un langage L n'est pas rationnel: montrer que $\sim_L$ est d'indice infini.

Pour cela il suffit de construire un ensemble infini $R \subseteq \Sigma^*$ tel que

$$\forall x, y \in R : x \neq y \Rightarrow x \not\sim_L y$$

On n'a pas besoin d'identifier toutes les classes d'équivalence de $\sim_L$ pour conclure que L n'est pas rationnel, il suffit d'en exhiber un nombre infini.

2 Les automates non-déterministes peuvent être beaucoup plus petits que les déterministes

Nous avons vu que la déterminisation d'un automate non-déterministe avec n états donne *en théorie* un automate déterministe avec 2^n états, mais dans les exemples, une grande partie de ces états était non-accessible. Nous pouvons maintenant donner un exemple de langages reconnaissables pour lesquels les automates non-déterministes sont beaucoup plus compactes que les automates déterministes.

Soit $\Sigma = \{a, b\}$, et pour tout $n \geq 0$:

$$L_n = \{w \in \Sigma^* \mid \text{ existe } v \in \Sigma^*, r \in \Sigma^n \text{ tel que } w = v \cdot a \cdot r\}.$$

Il y a évidement un automate non-déterministe avec $n + 2$ états qui reconnaît L_n. Quand deux mots x et y diffèrent sur les $n + 1$ dernières positions, c.-à-d. quand

$$\begin{aligned} x &= x_1 \cdot x_2 \\ y &= y_1 \cdot y_2 \end{aligned}$$

où $|x_2| = |y_2| = n + 1$, et $x_2 \neq y_2$, alors $x \not\sim_{L_n} y$. Par conséquent, $\mathit{indice}(\sim_{L_n}) \geq 2^{n+1}$, et selon Myhill-Nerode tout automate déterministe complet qui reconnaît L_n a au moins 2^{n+1} états.

3 Pour les automates minimales, $\sim_A$ et $\sim_L$ coïncident

Soit A un automate déterministe complet minimal qui reconnaît le langage L. Alors on a que $\sim_A = \sim_L$.

4 Unicité de l'automate minimal

L'automate minimal est unique, dans le sens que deux automates déterministes et complets qui reconnaissent le même langage, et qui ont un nombre d'états minimal, sont identiques à renommage des états près (on dit que les deux automates sont *isomorphes*).

5 Un critère pour être minimal

Un automate déterministe et complet $A = (\Sigma, Q, q_0, F, \delta)$ est minimal (c-à-d, il a un nombre minimal d'états parmi tous les automates déterministes complets qui reconnaissent le même langage) si et seulement si :

1. tous ses états sont accessibles à partir de q_0,
2. et pour tous états différents $q_1, q_2 \in Q$ il existe un mot $w \in \Sigma^*$ tel que $\delta^*(q_1, w) \in F$ et $\delta^*(q_2, w) \notin F$, ou $\delta^*(q_1, w) \notin F$ et $\delta^*(q_2, w) \in F$.

6 Minimalité de l'automate des résiduels

L'automate des résiduels construit au cours 9 est minimal, si on a bien pris soin d'identifier les expressions rationnelles qui sont équivalentes. On le montre facilement avec le critère de la section 5 : par construction tous ses états sont accessibles, et quand $x^{-1}L$ et $y^{-1}L$ sont deux états différents on a que $x \not\sim_L y$.

On obtient donc en principe une méthode pour minimiser un automate : d'abord on transforme l'automate en expression rationnelle, suivant le lemme d'Arden ou la construction de Brzozowski-McCluskey, et puis on construit l'automate des résiduels pour cette expression rationnelle. Cette méthode est un peu laborieuse, la méthode de la section 7 est beaucoup plus simple.

7 Minimisation par partition d'états (méthode de Moore)

Soit $A = (\Sigma, Q, q_0, F, \delta)$ un automate complet et déterministe.

Supprimer d'abord tous les états non-accessibles de A.

Une *partition* de Q est une famille $(Q_1, \ldots, Q_n)$ avec $Q_i \subseteq Q$ pour tout i, et :

- $Q = Q_1 \cup \ldots \cup Q_n$
- $Q_i \cap Q_j = \emptyset$ si $i \neq j$
- $Q_i \neq \emptyset$ pour tout i

Si C est une partition de Q, $P \in C$ et $a \in \Sigma$, on dit que *a sépare* P en $P_1, \ldots, P_n$ si :

- $(P_1, \ldots, P_m)$ est une partition de P ;
- pour tout i il existe une partie Q_i de C telle que $P_i = \{q \in Q_i \mid \delta(q, a) \in Q_i\}$.

L'algorithme construit une séquence $C_0, C_1, \ldots, C_m$ de partitions de Q. La première partition est $C_0 = (F, Q \setminus F)$. Les partitions suivantes sont obtenues par séparation : on choisit un $a \in \Sigma$ et une partie P de C_i tel que a sépare P en $P_1, \ldots, P_n$, et on obtient C_{i+1} en remplaçant dans

C_i la partie P par $P_1, \ldots, P_n$. Le processus s'arrête quand la partition obtenue ne peut plus être raffinée.

Finalement on obtient l'automate déterministe et complet suivant :

- son ensemble d'états est la dernière partition construite ;
- l'état initial est la partie qui contient q_0 ;
- un état P est acceptant quand il contient un état de F ;
- on a une transition de P_1 par le symbole a vers P_2 quand il y a $p_1 \in P_1$ tel que dans l'automate A : $\delta(p_1, a) \in P_2$.

Selon le critère de la section 5 cet automate est minimal.

8 Équivalence entre automates

On obtient donc une méthode relativement simple pour montrer que deux automates déterministes complets A_1 et A_2 reconnaissent le même langage : on minimise les deux, et on vérifie que les deux automates minimaux obtenus sont les mêmes à renommage des états près.

9 Minimisation par la méthode de Brzozowksi

Brzozowski a proposé une méthode de minimisation étonnante : étant donné un automate déterministe et complet, on construit l'automate pour son miroir, on déterminise, puis on fait de nouveau le miroir, et on déterminise. Il est sous-entendu que la déterminisation ne crée que des états accessibles, comme nous l'avons toujours fait dans les exemples :

$$A_0 \xrightarrow{mir} A_1 \xrightarrow{det} A_2 \xrightarrow{mir} A_3 \xrightarrow{det} A_4$$

Évidement, A_0 et A_4 reconnaissent le même langage, mais on a aussi que A_4 est minimal ! Ceci est dû au lemme suivant qu'on peut montrer avec le critère de la section 5 : Soit A un automate déterministe tel que tous ses états sont accessibles, $\overline{A}$ son miroir, et $\overline{A}^D$ l'automate obtenu par déterminisation de $\overline{A}$, on prenant soin que $\overline{A}^D$ est complet et que tous ses états sont accessibles. Alors $\overline{A}^D$ est minimal.

On applique ce lemme à l'automate A_2 de la construction de Brzozowksi.