Université Paris Cité
Automates et Analyse Lexicale

UFR Informatique
Année 2023–2024

Handout 4

Automates Non Déterministes

1 Automates Non Déterministes

Un *automate fini non déterministe* (AFND) est un quintuplet $(\Sigma, Q, I, F, \delta)$ tel que

- Σ est un alphabet fini ;
- Q est un ensemble fini, appelé l'ensemble des *états* ;
- $I \subseteq Q$, appelé *ensemble des états initiaux* ;
- $F \subseteq Q$, appelé *ensemble des états acceptants* ;
- $\delta\colon Q \times \Sigma \mapsto \mathcal{P}(Q)$ est une fonction, appelée la *fonction de transition.*

Dans un automate non déterministe, la fonction de transition associe à un état et une lettre un *ensemble* d'états. Cet ensemble peut être vide.

On se permet de considérer les AFD comme un cas particulier des AFND, même s'il y a formellement une différence dans les types.

Intuitivement, pour exécuter un AFND sur un mot il faut exploiter *toutes* les possibilités de suivre les flèches. Un AFND accepte un mot quand il y a *au moins une* possibilité d'atteindre un état acceptant à partir d'un état initial.

Formellement : Étant donné un AFND $(\Sigma, Q, I, F, \delta)$, on définit une fonction $\delta^*\colon Q \times \Sigma^* \mapsto \mathcal{P}(Q)$ par récurrence sur le deuxième argument :

$$\begin{aligned} \delta^*(q, \epsilon) &= \{q\} \\ \delta^*(q, wa) &= \bigcup_{p \in \delta^*(q,w)} \delta(p, a) \end{aligned}$$

Le langage *reconnu* par l'AFND A est

$$\mathcal{L}(A) = \{w \in \Sigma^* \mid \text{il existe } q_0 \in I \text{ tel que } \delta^*(q_0, w) \cap F \neq \emptyset\}$$

L'AFND A *accepte* un mot w si $w \in \mathcal{L}(A)$.

2 Déterminisation

Pour tout AFND $A = (\Sigma, Q, I, F, \delta)$ il existe un AFD $A' = (\Sigma, Q', q_0', F', \delta')$ tel que $\mathcal{L}(A') = \mathcal{L}(A)$.

La construction de A' est comme suit :

- $Q' = \mathcal{P}(Q)$, donc tout sous-ensemble de Q est un élément de Q'.

- $q'_0 = I$
- $F' = \{P \subseteq Q \mid P \cap F \neq \emptyset\}$
- $\delta'(P, a) = \bigcup_{p \in Q} \delta(p, a)$

Donc, si l'automate non déterministe a n état, l'automate déterministe correspondant en a 2^n (en théorie, en pratique certains de ces états peuvent être inutiles).

3 Applications des Automates Non Déterministes

3.1 Union et intersection de langages reconnaissables

On obtient comme corollaire que *Rec* est close sous union : étant donné deux AFD $A_1 = (\Sigma, Q_1, q_1, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, q_2, F_2, \delta_2)$, on s'assure d'abord que Q_1 et Q_2 sont disjoints. Puis on défini un AFND

$$A_{12} = (\Sigma, Q_1 \cup Q_2, \{q_1, q_2\}, F_1 \cup F_2, \delta_{12})$$

où

$$\delta_{12}(q, a) = \begin{cases} \delta_1(q, a) & \text{si } q \in Q_1 \\ \delta_2(q, a) & \text{si } q \in Q_2 \end{cases}$$

puis on déterminise l'automate A_{12}.

Puisque *Rec* est close sous complément et union, elle est aussi close sous intersection, car $L_1 \cap L_2 = \left(L_1^{comp} \cup L_2^{comp}\right)^{comp}$.

3.2 Miroir d'un langage reconnaissable

La classe des langages reconnaissables est close sous miroir : Si L est ŕeconnaissable, alors $\overline{L}$ l'est également : Soit $A = (\Sigma, Q, q_0, F, \delta)$ un automate déterministe tel que $\mathcal{L}(A) = L$. On construit un automate non-déterministe

$$A' = (\Sigma, Q, F, \{q_0\}, \delta')$$

où $\delta'(q, a) = \{p \in Q \mid \delta(p, a) = q\}$. Il est évident que $\mathcal{L}(A') = \overline{L}$. Finalement, on obtient un automate déterministe reconnaissant L en déterminisant A'.

3.3 Construction d'un automate pour la recherche d'un motif

Étant donné un mot m, on peut très facilement construire un automate non-déterministe reconnaissant tous les mots qui ont un facteur m, puis déterminiser cet automate. Par contre, dans ce cas précis, la construction directe de l'algorithme KMP est plus rapide, et donne un automate plus petit.