CTD 3: Compléments sur les automates et langages rationnels

BUT 2 – Automates et Langages – R4.A12

1 Automates finis non déterministes (AFN)

Dans un AFD, pour chaque état et chaque symbole il existe au plus (ou *exactement* pour un AFD complet) une transition vers un état successeur. Il est possible de relacher cette contrainte et d'autoriser plusieurs états successeurs, on obtient des automates dits *non déterministes.*

Définition 1 (Automates finis non déterministes)**.** Un **automate fini non déterministe** (AFN) A est un quintuplet $A = (Q, \Sigma, \delta_0, q_0, F)$ où

- Q est un ensemble fini d'états ;
- Σ est un alphabet ;
- $q_0 \in Q$ est l'état initial ;
- $F \subseteq Q$ est l'ensemble des états acceptants (ou finaux) ;
- $\delta : Q \times \Sigma \to 2^Q$ est la fonction de transition qui à un état et un symbol associe un *ensemble* d'états.

Reamrque : Une façon équivalente de définir les AFN est de décrire les transitions à l'aide d'une relation $\delta \subseteq Q \times \Sigma \times Q$. De même, certaines définitions d'AFN autorisent plusieurs états initiaux (i.e. $q_0 \in Q$ est remplacé par $I \subseteq Q$).

La simulation d'un ANF pour un mot d'entrée peut potentiellement aboutir à plusieurs états. Il suffit qu'un de ces états soit acceptant pour que le mot soit accepté par l'automate.

Définition 2 (Langage reconnu par un AFN)**.** $\hat{\delta}$ est l'extension de la fonction de transition δ aux mots et est définie par :

$$\begin{array}{rcl} \hat{\delta} : Q \times \Sigma^* & \to & 2^Q \\ (q, \epsilon) & \mapsto & \{q\} \\ (q, xw) & \mapsto & \bigcup_{q' \in \delta(q,x)} \hat{\delta}(q', w) \end{array}$$

Soit $A = (Q, \Sigma, \delta, q_0, F)$ un AFN, le **langage reconnu** par A, noté $\mathcal{L}(A)$, est défini par

$$\mathcal{L}(A) = \{w \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$$

Exercice 1. Soit $\Sigma = \{a, b, c\}$, on désire construire un automate, le plus simple possible, qui reconnaisse le langage des mots sur Σ qui contiennent le mot *bac* **et** se terminent par *a*.

1. Donnez une expression rationnelle qui représente ce langage.
2. Dessinez un AFN A_0, le plus simple possible, qui reconnaisse ce langage.
3. Indiquez comment A_0 lit chacun des mots suivants : *abaca*, *abacb*, *abacbaca*, *bcaa*, *bcab*, ϵ. On demande d'indiquer toutes les lectures de chaque mot !
4. Mêmes questions pour le langage des mots qui contiennent le mot *bac* **ou** (inclusif) se terminent par *a*. On nommera l'AFN A_1.

Les AFD sont évidemment des cas particuliers d'AFN. Donc, tout langage reconnu par un AFN est aussi reconnu par un AFD. L'inverse n'est pas immédiat... Il se trouve que les AFN n'augmentent par le pouvoir d'expressivité des AFD, c'est-à-dire qu'ils ne reconnaissent pas plus de langages. C'est la proprété qu'énoncent les théorèmes suivants.

Théorème 1. Soient $A_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ un AFN, et $A_D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ l'AFD où

— $Q_D = 2^{Q_N}$

— $F_D = \{S \mid S \subseteq Q_N \text{ et } F_N \cap Q_N \neq \emptyset\}$

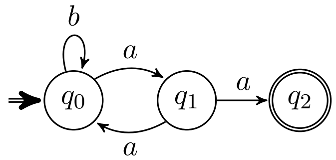
— $\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$

alors, $\mathcal{L}(A_N) = \mathcal{L}(A_D)$.

En pratique, il n'est pas nécessaire de construire tous les états (i.e. toutes les parties de Q_N), il suffit de construire ceux qui sont atteignables depuis l'état initial $\{q_0\}$.

Théorème 2. Un langage est reconnu par un AFN si et seulement s'il est reconnu par un AFD.

Exercice 2. Donnez un AFD équivalent à cet AFN :



Exercice 3. Déduisez des AFN A_0 et A_1 de l'exercice 1, des AFD qui reconnaissent les mêmes langages.

2 Automates et langages rationnels

Théorème 3. Les langages reconnaissables par automates finis (déterministes ou non) sont exactement les langages rationnels.

Il existe des langages qui ne sont pas rationnels (et donc pour lesquels il n'existe pas d'expressions rationnelles qui les représentent). Cette propriété se démontre, par exemple, à l'aide du lemme de la pompe pour les langages rationnels.

Lemme 1 (Lemme de la pompe pour les langages rationnels)**.** Soit L un langage rationnel, il existe une constante n (qui dépend de L) telle que pour ton mot w pour lequel $|w| \geq n$, w peut être décomposé en 3 mots $w = v_1 u v_2$ tels que :

1. $u \neq \epsilon$
2. $|v_1 u| \leq n$
3. $\forall k \geq 0, v_1 u^k v_2 \in L$

Considérons le langage L des mots sur $\Sigma_{\mathbb{B}} = \{0, 1\}$ qui ont un nombre pair de 0 ; L est rationnel. La constante n associée à L dans le lemme de la pompe est égale à 2. En effet, pour le mot $00 \in L$ de longueur 2, en posant $v_1 = v_2 = \epsilon$ et $u = 00$, on a bien $(00)^k \in L$ pour tout $k \geq 0$. De même, pour le mot 010, en posant $v_1 = v_2 = 0$ et $u = 1$, on a bien $01^k0 \in L$ pour tout $k \geq 0$.

Exercice 4. Montrer que le langage $L = \{a^n b^n | n \neq 0\}$ sur l'alphabet $\Sigma = \{a, b\}$ n'est pas un langage rationnel.