UMONS

Faculté
des Sciences

Automates

Étudiant : Benjamin André
Directrice : Véronique Bruyère
23 avril 2020

Faculté des Sciences • Université de Mons • Place du Parc 20 • B-7000 Mons

Table des matières

1 Automates utilisés

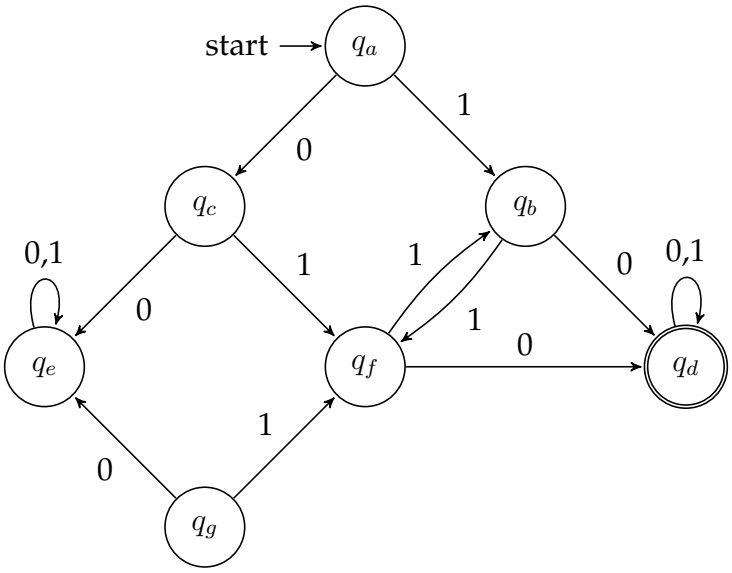

FIGURE 1: Automate A_B, exemple personnel

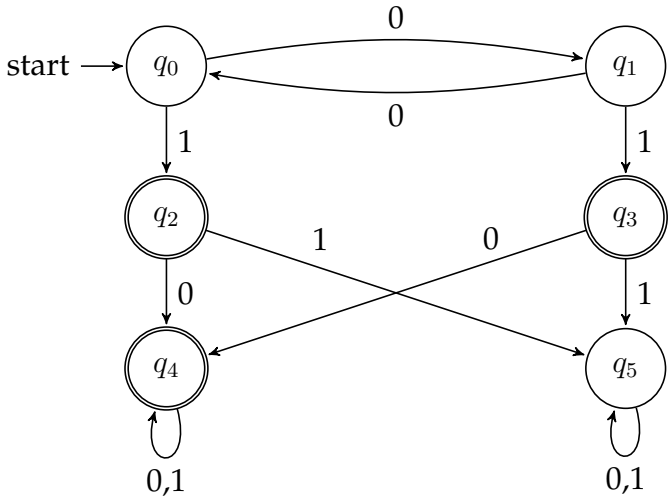

FIGURE 2: Automate A_H, exemple d'un livre de référence[1]

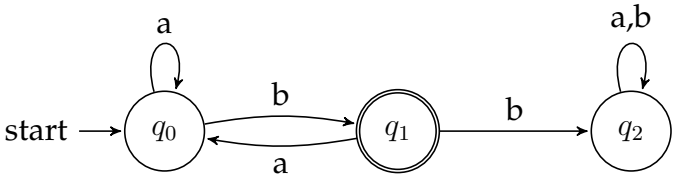

FIGURE 3: Automate A_N, exemple d'une thèse[2]

2 Bases théoriques

2.1 Alphabet

Un alphabet, nommé par convention Σ est un ensemble fini et non vide de symboles. Voici certains exemples d'alphabets :

— $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, l'alphabet des chiffres
— $\Sigma = \{a, b, c, ..., z, A, B, C, ..., Z\}$, l'alphabet latin
— $\Sigma = \{0, 1\}$ l'alphabet binaire

2.2 Mots

Comme Σ est un ensemble, on peut définir Σ^k, qui donne des k-uples de symboles, appartenant tous à Σ.

Un mot w de taille $|w| = k$ est un ensemble de symboles provenant de Σ^k. Dans le cas particulier où $k = 0$, on note le mot vide (sans symbole) $w = \epsilon$.

De façon générale, w est un mot sur Σ si il existe k tel que $w \in \Sigma^k$. Par convention, les mots sont nommés par une lettre minuscule, souvent w, x, y, z.

L'ensemble de tous ces mots possible sur Σ est noté Σ^*. Cet ensemble est infini.

2.3 Expression régulière

TODO : concatenation

2.4 Langage

Un langage L est défini sur un alphabet Σ. L est un ensemble de mots sur cet alphabet : $L \subseteq \Sigma^*$. Comme Σ^* contient une infinité de mots, L est susceptible de ne pas être fini non plus.

Par exemple, par rapport à tous les mots disponibles avec les lettres de l'alphabet latin (Σ^*), seulement certains font partie de la langue française (L).

L peut être défini :

— en énumérant les mots en faisant partie : $L = \{12, 35, 42, 7, 0\}$
— via un notation ensembliste : $L = \{0^k 1^j | k + j = 7\}$ ou $L = \{w | w \text{ est un mot français}\}$

Dans ces deux cas, Σ est souvent implicite.

TODO : Représentation patate des langages et régulier

Parmi tous les langages possibles, certains possèdent la propriété d'être réguliers. Ceux-ci ont la caractéristique supplémentaire de pouvoir être définis par une expression régulière. Un des conséquences de cette propriété est qu'ils peuvent être représentés par un automate déterministe fini.

2.5 DFA

Soit un ensemble de symboles Σ. Soient $\Sigma^* = \{a_1 a_2 a_3 ... a_n | a_1, a_2, a_3, ..., a_n \in \Sigma\}$, l'ensemble des mots de taille arbitraire qu'il est possible de former à partir de Σ et $|w|, w \in \Sigma$ la longueur

de w, le nombre de symboles utilisés. Si $|w| = 0$, on note $w = \epsilon$.

Un automate est défini par $A = (Q, \Sigma, q_0, \delta, F)$ où

— Q est un ensemble d'états, différenciés par leur indice $q_1, q_2, ..., q_n$ ou $n = |Q|$.

— Σ est un ensemble de symboles

— $q_0 \in Q$ est l'état initial

— $\delta : Qx\Sigma \rightarrow Q$ est la fonction de transition. A partir d'un état de Q, en fonction d'un symbole, elle retourne un nouvel état faisant partie de Q.

— $F \subseteq Q$ est un ensemble d'état finaux.

A définir

— Accepter un langage

— Congruence à droite

2.6 Théorème de Myhill-Nerode

Théorème 2.1 *Les 3 énoncés suivants sont équivalents :*

1. *Un langage $L \subseteq \Sigma^*$ est accepté par un DFA*
2. *L est l'union de certaines classes d'équivalence d'index fini respectant une relation d'équivalence et de congruence à droite*
3. *Soit la relation d'équivalence $R_L : xR_Ly \Leftrightarrow \forall z \in \Sigma^*, xz \in L \Leftrightarrow yz \in L$. R_L est d'index fini.*

Preuve 2.1.1 *La preuve d'équivalence se fait en prouvant chaque implication de façon cyclique :*

$(1) \rightarrow (2)$ *Soit un langage $L \subseteq \Sigma^*$ qui est accepté par un automate déterministe fini (ADF) A. Soit la relation $R_A : xR_Ay \Leftrightarrow \delta(q_0, x) = \delta(q_0, y)$ qui détermine si deux mots, une fois parcourus dans l'automates, finissent sur le même état.*

C'est une relation d'équivalence (réflexive, transitive et symétrique), et congruente à droite :

— ***Réflexivité :*** *Soit le mot $x \in \Sigma^*$. Alors, par définition, $xR_Mx \Leftrightarrow \delta(q_0, x) = \delta(q_0, x)$.*

— ***Transitivité :*** *Soient les mots $x, y, z \in \Sigma^*$ tels que xR_My et yR_Mz. Alors, $\delta(q_0, x) = \delta(q_0, y) = \delta(q_0, z)$ par la transitivité de l'égalité. Dès lors, xR_Mz*

— ***Symétrie :*** *Soient les mots $x, y \in \Sigma^*$ tels que xR_My. Comme $\delta(q_0, x) = \delta(q_0, y)$, $\delta(q_0, y) = \delta(q_0, x)$. Donc, yR_Mx.*

— ***Congruence à droite :*** *Soient les mots $x, y \in \Sigma^*$ tels que xR_My. Soit un mot $w \in \Sigma^*$. $\delta(q_0, xw) = \delta(\delta(q_0, x), w) = \delta(\delta(q_0, y), w) = \delta(q_0, yw)$.*

Il peut au plus y avoir une classe d'équivalence par état (valeurs possibles retournées par δ). Ce nombre d'état étant fini dans M, R_M est donc d'index fini. Le langage L correspond aux mots menant à un état appartenant à F, et F peut être écrit comme une union d'état, qui correspondent à une classe d'équivalence de R_M qui est bien d'index fini et respectant la congruence à droite.

$(2) \rightarrow (3)$ *toute relation E de 2 est un refinement de RL du coup chaque c.eq est completement contenue dans une c.Eq de RL. on part de xRMy, cong droite*

$(3) \rightarrow (1)$ *Mq RL cong droite xRLy, utiliser définitions*

Corrolaire 2.1.1 *Possibilité de créer l'automate canonique...*

2.7 Table Filling Algorithm

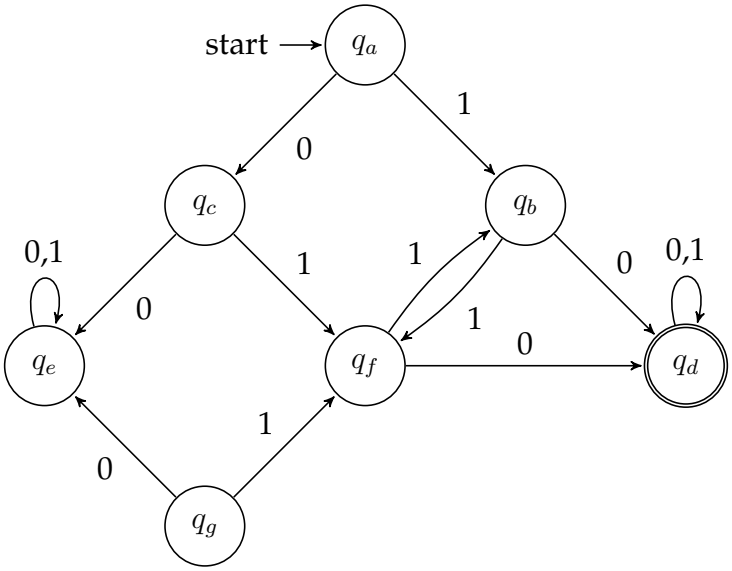

FIGURE 4: Automate A_1

L'état q_g n'est pas atteignable : il peut être simplement supprimé.

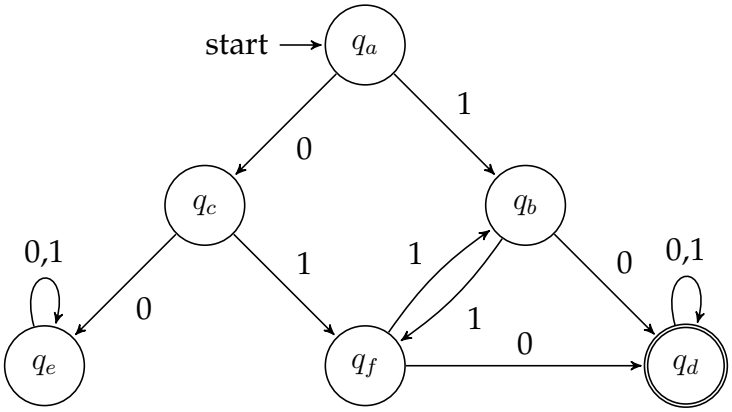

FIGURE 5: Automate A_2

Par l'algorithme de minimisation, on obtient A_3. De cet automate, on peut déduire une écriture de L sous forme d'expression régulière : $(1|01)1^*0(0|1)^*$

	A	B	C	D	E
B	x				
C	x	x			
D	x	x	x		
E	x	x	x	x	
F	x		x	x	x

FIGURE 6: Table filling pour A_2, décelant des équivalences d'états

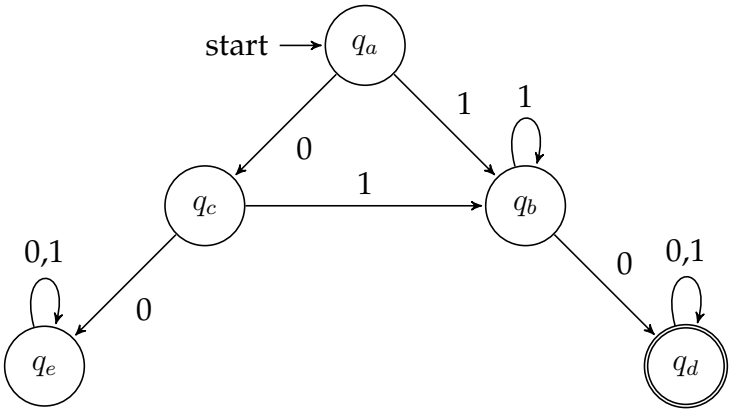

FIGURE 7: Automate A_3

Références

[1] J. E. HOPCROFT AND J. D. ULLMAN, *Introduction to automata theory, languages and computation. adison-wesley*, Reading, Mass, (1979).

[2] D. NEIDER, *Applications of automata learning in verification and synthesis*, PhD thesis, Hochschulbibliothek der Rheinisch-Westfälischen Technischen Hochschule Aachen, 2014.