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

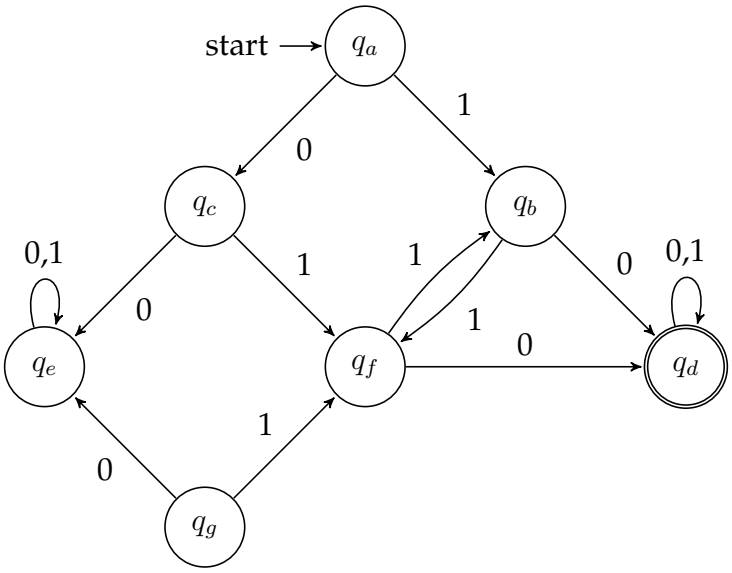

FIGURE 1: Automate A_B, exemple personnel

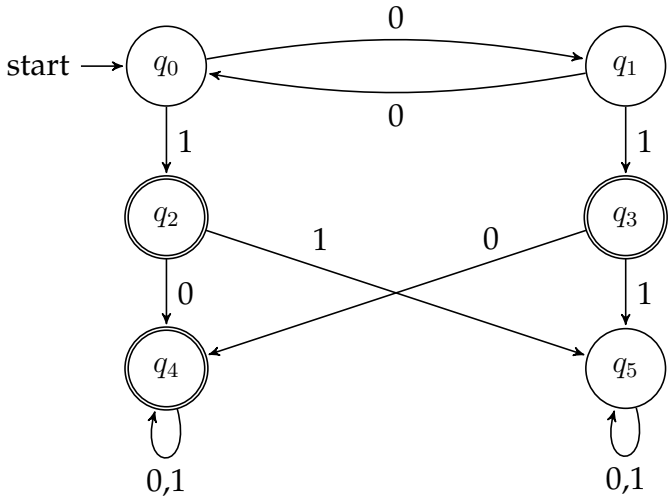

FIGURE 2: Automate A_H, exemple d'un livre de référence[1]

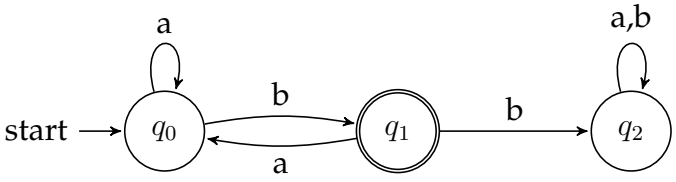

FIGURE 3: Automate A_N, exemple d'une thèse[2]

2 Bases théoriques

2.1 DFA

Soit un ensemble de symboles Σ. Soient $\Sigma^* = \{a_1a_2a_3...a_n | a_1, a_2, a_3, ..., a_n \in \Sigma\}$, 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

2.2 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...*

2.3 Table Filling Algorithm

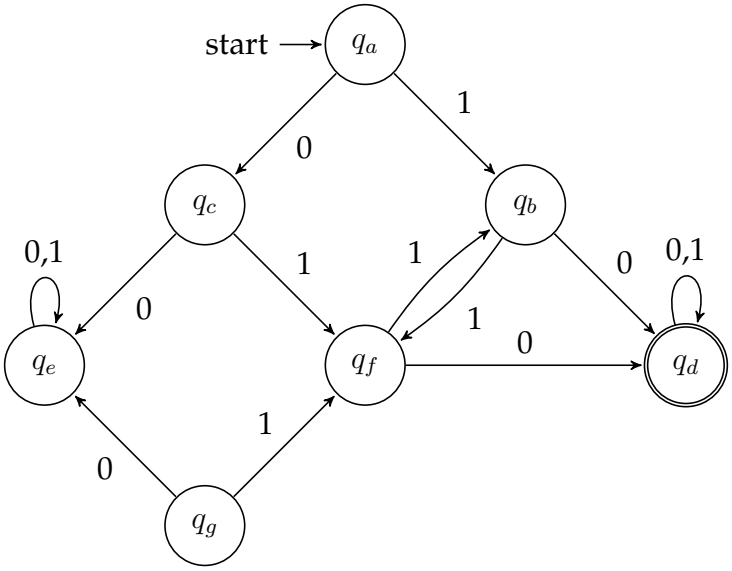

FIGURE 4: Automate A_1

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

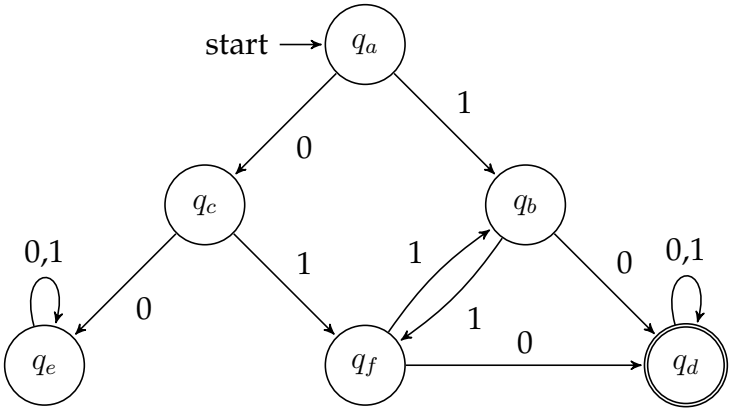

FIGURE 5: Automate A_2

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

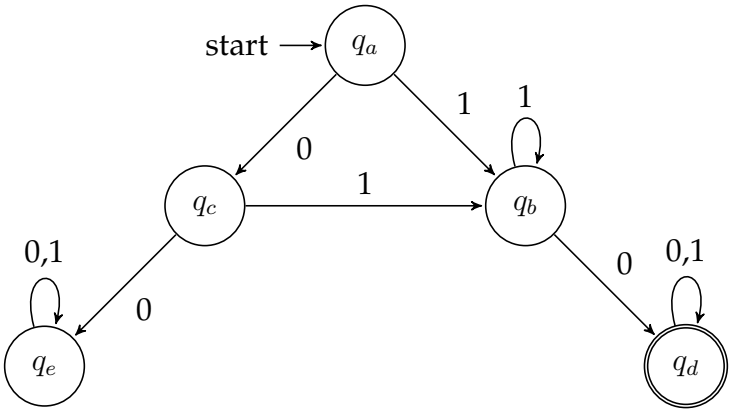

FIGURE 7: Automate A_3

3 Velleda

Complexité 4.4.2, bad pair pq, rs q et q' vont sur un meme p Attention a bien comprendre le cas de base, utilisation du mot témoin w qui différencie. La contradiction est sur la table pas sur w le plus petit (c'est un élemenet qu'on a introduit, ça nous avance à rien de le contredire)

Fig 4.13 + Exemple + exemple preuve éq minimaux

A rédiger, DFA/Notations, Rédiger, Avec des exemples persos

Prouver Reflexif, transitif, symetrique

Lire 4.23

Important : la notion de congruence à droite

Angluin : créer un contre-exemple

4.24.3 tout état d'une classe S a une transition vers un état de la classe T par construction de la table

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.