Langages rationnels et automates finis

0| Alphabet, mot et langages

Déf 1 : Un *alphabet* A est un ensemble de symboles.

Ex : $A_1=\{a,b\}$; $A_2=\{0;1\}$; $A_3=\{a;b;c;d;e;\dots;x;y;z\}$.

Def 2 : Un *mot* sur un alphabet A est une suite finie de symboles de A. On note A^* les mots sur l'alphabet A et ε le mot vide.

Ex : $abaab \in A_1^*$; $abaab \in A_3^*$; $bonjour \in A_3^*$; $0100 \in A_2^*$.

Def 3 La *concaténation* de deux mots $u=u_0u_1u_2\dots u_n$ et $v=v_0v_1\dots v_m$ sur l'alphabet A est le mot $u\cdot v = uv = u_0u_1u_2\dots u_n v_0v_1\dots v_m$.

Ex : bon·jour = bonjour $\in A_3^*$; $0100\cdot 1010 = 01001010 \in A_2^*$.

Def 4 u est un *préfixe* (resp *suffixe*) de w si il existe un mot v tel que w = uv (resp w = vu).

Ex : bon est un préfixe de bonjour. jour est un suffixe de bonjour

Def 5 u est un *facteur* de w si il existe deux mots v, v' tel que w = vuv'.

Ex : 010 est un facteur de 10110101.

11 n'est pas un facteur de 101001.

Def 6| u est un *sous-mot* de $w = w_0w_1w_2\dots w_n$ si il existe $0\le i_0<i_1<i_2<i_3<\dots<i_m\le n$ tel que $u = w_{i_0}w_{i_1}\dots w_{i_m}$.

Ex : 0100 est un sous-mot de 110011010.

011 n'est pas un sous-mot de 1110100.

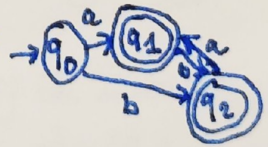
Def 7| Un langage $L\subset A^*$ sur l'alphabet A est un ensemble de mots sur A.

Ex : L'ensemble des nombres pairs écrits en binaire est un langage sur A_2^*.

1| Automates et langages reconnaissables

Def 8| Un *automate fini déterministe* (AFD) $\mathcal{A}$ est un tuple (Q,A,δ,i,F) tel que :
- Q est un ensemble fini d'états.
- A est un alphabet.
- δ est une fonction de transition partiel de $Q\times A$ dans Q.
- $i\in Q$ est un état initial.
- $F\subset Q$ est un ensemble d'états finaux.

Ex :



$Q=\{q_0;q_1;q_2\}$; $A=\{a,b\}$;
$\delta(q_0,a)=q_1$ $\delta(q_0,b)=q_2$ $\delta(q_1,b)=q_2$
$\delta(q_2,a)=q_1$; $i=q_0$; $F=\{q_1;q_2\}$

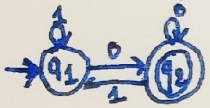
Def 9| On définit inductivement la fonction de transition de $\mathcal{A}$ $\delta^*: Q\times A^*\to Q$ sur les mots par $\delta^*(q,\varepsilon)=q$

$\delta^*(q,w) = \delta^*(\delta(q,w_1), w_2w_3\dots w_n)$ (si défini)

Def 10) Un mot w est reconnu par l'automate $\mathcal{A} = (Q, A, \delta, i, F)$ si $\delta^*(i, w) \in F$.

Le langage $\mathcal{L}(\mathcal{A})$ reconnu par l'automate $\mathcal{A}$ est l'ensemble des mots reconnus par $\mathcal{A}$.

Un langage est dit reconnaissable par automate fini si il existe un automate $\mathcal{A}$ le reconnaissant.

Ex :  L'automate ci contre reconnait les entiers pairs écrits en binaire, ce langage est donc reconnaissable par automate fini.

Not) : On note Rec la classe des langages reconnaissable par automate fini.

Def 11) Un automate est complet si sa fonction de transition est totale

Thm 1) Pour tout AFD $\mathcal{A}$, il existe un AFD complet $\mathcal{A}_c$ reconnaissant le même langage.

Def 12) Soit $q \in Q$ un état de l'AFD $\mathcal{A}$;

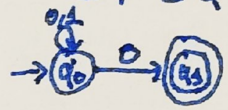
- q est accessible si il existe un mot w tel que $q = \delta^*(i, w)$.
- q est co-accessible si il existe un mot w et un état final $q' \in F$ tel que $q' = \delta^*(q, w)$.

Un automate est dit émondé si tout ses états sont accessibles et co-accessibles

Thm 2) Pour tout AFD $\mathcal{A}$, l'automate $\mathcal{A}_e$ restraint à ses états accessibles et co-accessibles est émondé et reconnait le même langage.

Def 13) Un automate fini non déterministe (AFND) $\mathcal{A}$ est un tuple (Q, A, Δ, I, F) tel que :

- Q est un ensemble fini d'états
- A est un alphabet fini
- $\Delta \subset Q \times A \times Q$ est une relation de transition
- $I \subset Q$ est un ensemble d'états initiaux
- $F \subset Q$ est un ensemble d'états finaux

Ex 

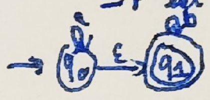
$Q = \{q_0 ; q_1\}$, $A = \{0 ; 1\}$

$\Delta = \{(q_0, 0, q_0) ; (q_0, 1, q_0) ; (q_0 ; 0, q_1)\}$

$I = \{q_0\}$; $F = \{q_1\}$

Def 14) Un AFND avec ε-transitions $\mathcal{A}$ est un tuple (Q, A, Δ, I, F) tel que :

- Q est un ensemble fini d'états
- A est un alphabet fini
- $\Delta \subset Q \times (A \cup \{\varepsilon\}) \times Q$ est une relation de transition
- I est un ensemble d'états initiaux
- F est un ensemble d'états finaux

Ex : 

$Q = \{q_0 ; q_1\}$; $A = \{a, b\}$; $\Delta = \{(q_0, a, q_0), (q_0, \varepsilon, q_1), (q_1, b, q_1)\}$

$I = \{q_0\}$; $F = \{q_1\}$

Def 15) Un mot w est reconnu par un AFND (avec ε-transitions) $\mathcal{A} = (Q, A, \Delta, I, F)$ si il existe un chemin $q_0 \xrightarrow{c_0} q_1 \xrightarrow{c_1} q_2 \xrightarrow{c_2} \cdots \xrightarrow{c_{n-1}} q_n$ tel que $\forall i \in [\![0 ; n-1]\!]$ $(q_i, c_i, q_{i+1}) \in \Delta$, $q_0 \in I$, $q_n \in F$ et $w = c_0 \cdot c_1 \cdot c_2 \cdot c_3 \cdots c_{n-1}$.

Thm 3) Pour tout AFND (avec ε-transitions), il existe un AFD reconnaissant le même langage.

2] Opérations sur les langages, expressions régulières

Thm 4] Rec est stable par union finie, intersection finie et passage au complémentaire.

Def 16] La concaténation des langages $K, L \subset A^*$ est le langage $K \cdot L = \{ uv \in A^* \mid u \in K, v \in L \}$.

Thm 5] Rec est stable par concaténation de langages

Def 17] L'étoile de Kleene d'un langage $L \subset A^*$ est le langage

$$L^* = \bigcup_{n \in \mathbb{N}} L^n \quad \text{où} \quad L^n = \underbrace{L \cdot L \cdot L \cdots L}_{n \text{ fois}}$$

Ex : $bon \cdot bon \cdot bon \in \{bon\}^3 \subset \{bon\}^*$

Thm 6] Rec est stable par passage à l'étoile de Kleene.

Def 18] La classe $\mathcal{E}$ des expressions régulières sur A est la plus petite classe tel que : $- \emptyset \in \mathcal{E}$ $\quad - \varepsilon \in \mathcal{E}$ $\quad - \forall a \in A \;\; a \in \mathcal{E}$

$- \forall e, f \in \mathcal{E}, \;\; e \cdot f \in \mathcal{E}, \; e | f \in \mathcal{E}$ et $e^* \in \mathcal{E}$

Def 19] Le langage reconnu par une expression régulière est défini inductivement par : $- L(\emptyset) = \emptyset$ $\quad - L(\varepsilon) = \{\varepsilon\}$ $\quad - \forall a \in A \;\; L(a) = \{a\}$

$- \forall e, f \in \mathcal{E}$

$$L(e \cdot f) = L(e) \cdot L(f)$$
$$L(e|f) = L(e) \cup L(f)$$
$$L(e^*) = L(e)^*$$

On note Reg la classe des langages reconnaissables par expressions régulières.

Thm 7] (Kleene) Rec = Reg. On appelle les langages de cette classe, les langages rationnels.

Preuve : $\boxed{\text{Dev 1}}$

3] Automate minimal

Def 20] On appelle quotient d'un langage $L \subset A^*$, les langages de la forme $Lw^{-1} = \{ u \in A^* \mid uw \in L \}$ avec $w \in A^*$.

Thm 8] Un langage est rationnel ssi et seulement si il a un nombre de quotients distincts finis. De plus, le nombre de quotients d'un langage rationnel correspond au nombre d'état minimum d'un AFD complet le reconnaissant.

Def 21] ~~Congruence~~ Une relation d'équivalence $\sim$ sur Q est une congruence si $\forall q, q' \in Q \quad q \sim q' \Rightarrow (q \in F \Leftrightarrow q' \in F)$
$\wedge (\forall a \in A \;\; \delta(q,a) \sim \delta(q',a))$

Def 22] La congruence de Nerode est une relation définie par $\forall q, q' \in Q \quad q \sim_N q' \Leftrightarrow (\forall w \in A^*, \; \delta^*(q,w) \in F \Leftrightarrow \delta^*(q',w) \in F)$

Thm 9] La congruence de Nerode est une congruence sur $\mathcal{A} = (Q, A, \delta, i, F)$. De plus, l'automate $\mathcal{A}_{/\sim_N}$ ~~dont les états~~ $= (Q_{/\sim_N}, A, \delta_{/\sim_N}, \bar{i}, F_{/\sim_N})$ où

- $Q_{/\sim_N}$ sont les classes d'équivalence $\bar{q}$ selon $\sim_N$. - $\bar{i}$ est la classe d'équivalence de
- $\forall (q,a) \in Q \times A \quad \delta_{/\sim_N}(\bar{q}, a) = \overline{\delta(q,a)}$ $\quad - F_{/\sim_N} = \{ \bar{q} \in Q_{/\sim_N} \mid q \in F \}$ est défini et minimal

Calcul de l'automate minimal : Algorithme de Moore $\boxed{\text{Dev 2}}$

Scanned with DocScan