1 Introduction

- Un **automate fini déterministe** est un 4-uplet $\mathcal{A} = (Q, \delta, i, F)$ où Q est un ensemble *fini* d'états, $\delta : Q \times \Sigma \to Q$ est la fonction de transition, i l'état initial, et $F \subseteq Q$ l'ensemble des états finaux.

- Le **language reconnu** par $\mathcal{A}$ est $\mathcal{L}(\mathcal{A}) := \{u \in \Sigma^* \mid \delta(i, u) \in F\}$. Un langage $L \subseteq \Sigma^*$ est dit **reconnaissable** si il existe un automate fini $\mathcal{A}$ tel que $L = \mathcal{L}(\mathcal{A})$

- Un **automate fini non-déterministe** est un 4-uplet $\mathcal{A} = (Q, T, i, F)$ où Q est un ensemble *fini* d'états, $T \subseteq Q \times \Sigma \times Q$ est la table de transition, i l'état initial, et $F \subseteq Q$ l'ensemble des états finaux.

- On peut **déterminiser** tout automate fini.

Rq : La preuve se fait à l'aide de l'automate des parties.
Rq : On peut tester en $O(|\omega| \cdot |Q|^2)$ si $\omega \in \mathcal{L}(\mathcal{A})$

- On peut **émonder** tout automate fini.

- Les automates avec des ϵ-transitions sont équivalents aux automates classiques.

Rq : On note $\text{Rec}(\Sigma^*)$ la famille des langages reconnaissable sur Σ^*.

- $\text{Rec}(\Sigma^*)$ est **fermée** par union, intersection, complément et quotient.

Rappel : $K^{-1}L := \{\omega \in \Sigma^* \mid \exists (k, l) \in K \times L,\ \omega = kl\}$ est appelé quotient (à gauche) de $L \in Rec(\Sigma^*)$ par $K \subseteq \Sigma^*$.
Rq : $\text{Rec}(\Sigma^*)$ est de même fermée par préfixe, suffixe et facteur.

- Soit $f : A^* \to B^*$ un morphisme, $L_1 \in \text{Rec}(A^*)$, $L_2 \in \text{Rec}(B^*)$, on a : $\begin{cases} f(L_1) \in \text{Rec}(B^*) \\ f^{-1}(L_2) \in \text{Rec}(A^*) \end{cases}$.

- Une **substitution** est une application $\sigma : A \to \mathcal{P}(B^*)$. Elle s'étend naturellement en morphisme de A^* vers $\mathcal{P}(B^*)$.

On adopte les notations suivantes : $\begin{cases} \text{Pour } L \subseteq A^*,\ \sigma(L) := \bigcup\limits_{u \in L} \sigma(u) \\ \text{Pour } L \subseteq B^*,\ \sigma^{-1}(L) := \{u \in A^* \mid \sigma(u) \cap L \neq \emptyset\} \end{cases}$

Une substitution est **rationnelle** si elle est définie par une application $\sigma : A \to \text{Rec}(B^*)$.

- La famille des langages reconnaissables est fermée par substitution rationnelle et substitution rationnelle inverse, i.e si $\sigma : A \to \text{Rec}(B^*)$, on a : $\begin{cases} \forall L \in \text{Rec}(A^*), \sigma(L) \in \text{Rec}(B^*) \\ \forall L \in \text{Rec}(B^*), \sigma^{-1}(L) \in \text{Rec}(A^*) \end{cases}$.

Résumé : Les langages reconnaissables sont clos par union, intersection, complément, quotient, préfixe, suffixe, facteur, morphisme et substitution (inverses).

2 Langages rationnels

- L'ensemble des expressions **rationnelles** $\mathcal{E}$ est le plus petit ensemble qui contient Σ, et qui est stable par $+, \cdot$ et $*$. Un langage L est dit **rationnel** si il existe une expression rationnelle e telle que $L = \mathcal{L}(e)$. On définit : $\begin{cases} \mathcal{L}(e+f) := \mathcal{L}(e) \cup \mathcal{L}(f) \\ \mathcal{L}(e) \cdot \mathcal{L}(f) := \mathcal{L}(e) \cdot \mathcal{L}(f) \\ \mathcal{L}(e^*) := \mathcal{L}(e)^* \end{cases}$

Rq : Deux e.r sont dites **équivalentes** si leurs langages sont égaux.

Théorème (Kleene) : $\mathrm{Rec}(\Sigma^*) = \mathrm{Rat}(\Sigma^*)$

Rq : Une démonstration du sens réciproque peut se faire avec l'algo de McNaughton-Yamada, décrit ci-dessous.

Algorithme (McNaughton-Yamada) :
Soit $\mathcal{A} = \{Q, T, I, F\}$ un automate ND. On va construire $e \in \mathcal{E}$ telle que $\mathcal{L}(e) = \mathcal{L}(\mathcal{A})$.
On dénote les états de $\mathcal{A}$ par $\{1, \dots, n\}$. On note $L_{p,q}$ (resp. $L_{p,q}^{(k)}$) le langage accepté par $\mathcal{A}$ avec p initial et q final (resp. et ne passant que par les états $1, \dots, k$ entre p et q).
On va exprimer $L_{p,q}^{(k)}$ par une expression rationnelle $e_{p,q}^{(k)}$, calculée récursivement :

- $e_{p,q}^{(0)} = \Sigma\{a \mid p \xrightarrow{a} q\}$
- Pour $1 \leq k \leq n$, $e_{p,q}^{(k)} = e_{p,q}^{(k-1)} + e_{p,q}^{(k-1)}(e_{k,k}^{(k-1)})^* e_{p,q}^{(k-1)}$

Alors, $L_{p,q}^{(k)} = \begin{cases} \mathcal{L}(e_{p,q}^{(n)}) & \text{si } p \neq q \\ \mathcal{L}(e_{p,q}^{(n)} + \emptyset^*) & \text{si } p = q \end{cases}$ et donc $\mathcal{L}(\mathcal{A}) = \bigcup_{i \in I} \bigcup_{f \in F} L_{i,f}$.

- **Lemme (Étoile) :** Si $L \in \mathrm{Rec}(\Sigma^*)$, alors il existe $N \geq 0$ tel que pour tout $x \in L$, on ait :

- Si $|x| \geq N$, alors $\exists u_1, u_2, u_3 \in \Sigma^*$ tels que $x = u_1 u_2 u_3$, $u_2 \neq \epsilon$ et $u_1 u_2^* u_3 \subseteq L$
- Si $x = w_1 w_2 w_3$ avec $|w_2| \geq N$ alors $\exists u_1, u_2, u_3 \in \Sigma^*$ tels que $w_2 = u_1 u_2 u_3, u_2 \neq \epsilon$ et $w_1 u_1 u_2^* u_3 w_3 \subseteq L$
- Si $x = u_1 v_1 \dots v_N w$ avec $|v_i| \geq 1$ alors il existe $0 \leq j < k \leq N$ tels que $u_1 v_1 \dots v_j (v_{j+1} \dots v_k)^* v_{k+1} \dots v_N w \subseteq L$

3 Résiduels

- Soient $u \in \Sigma^*$, et $L \subseteq \Sigma^*$. Le **résiduel** de L par u est le quotient $u^{-1}L = \{v \in \Sigma^* \mid uv \in L\}$.

- Soit $L \subseteq \Sigma^*$. L'**automate des résiduels** de L est $\mathcal{R}(L) = (Q_L, \delta_L, i_L, F_L)$ avec :

$$\begin{cases} Q_L = \{u^{-1}L \mid u \in \Sigma^*\} \\ \delta_L(u^{-1}L, a) = a^{-1}(u^{-1}L) = (ua)^{-1}L \\ i_L = L = \epsilon^{-1}L \\ F_L = \{u^{-1}L \mid \epsilon \in u^{-1}L\} = \{u^{-1}L \mid u \in L\} \end{cases}$$

Théorème : Un langage est reconnaissable ssi il a un nombre fini de résiduels.

- Soit $\mathcal{A}$ un automate DC. Une relation d'équivalence $\sim$ sur Q est une congruence si :

- $\forall p, q \in Q,\ \forall a \in \Sigma,\ p \sim q \Rightarrow \delta(p, a) \sim \delta(q, a)$
- F est saturé par $\sim$, i.e : $\forall p \in F,\ [p] = \{q \in Q \mid p \sim q\} \subseteq F$

Le **quotient** de $\mathcal{A}$ par $\sim$ est $A/\sim = (Q/\sim, \delta_\sim, [i], F/\sim)$, où $\delta_\sim$ est définie par $\delta_\sim([p], a) = [\delta(p, a)]$

Théorème : $\mathcal{L}(\mathcal{A}/\sim) = \mathcal{L}(\mathcal{A})$

- Soit $\mathcal{A} = (Q, \delta, i, F)$ un automate DCA (DC et accessible) reconnaissant L. Pour $q \in Q$, on note $\mathcal{L}(\mathcal{A}, q) = \{u \in \Sigma^* \mid \delta(q, u) \in F\}$. L'**équivalence de Nerode** de $\mathcal{A}$ est définie par $p \sim q$ si $\mathcal{L}(\mathcal{A}, p) = \mathcal{L}(\mathcal{A}, q)$.

- L'équivalence de Nerode est une congruence.
- L'automate $\mathcal{A}/\sim$ est appelé **quotient de Nerode** de $\mathcal{A}$.

Théorème : $\mathcal{A}/\sim = \mathcal{R}(L)$, i.e : Le quotient de Nerode est isomorphe à l'automate des résiduels.
$\varphi : Q/\sim \to Q_L$ définie par $\varphi([q]) = \mathcal{L}(\mathcal{A}, q)$ est un isomorphisme de $\mathcal{A}/\sim$ sur $\mathcal{R}(L)$.

Théorème : Soit $L \in \mathrm{Rec}(\Sigma^*)$.

- Si $\mathcal{A}$ est un automate DCA qui reconnaît L, alors $\mathcal{R}(L)$ est un quotient de $\mathcal{A}$.
- $\mathcal{R}(L)$ est minimal parmis les automates DCA reconnaissant L (en nombre d'états).
- Si $\mathcal{A}$ est un automate DC reconnaissant L avec un nombre minimal d'états, alors $\mathcal{A}$ est isomorphe à $\mathcal{R}(L)$ (i.e l'automate minimal est unique).

Algorithme de Moore : Pour $n \geq 0$, on définie $\sim_n$ sur Q par: $p \sim_n q$ si $\mathcal{L}(\mathcal{A}, p) \cap \Sigma^{\leq n} = \mathcal{L}(\mathcal{A}, q) \cap \Sigma^{\leq n}$
Alors, en notant $\sim = \bigcap_{n \geq 0} \sim_n$, on a les propriétés suivantes :

- $p \sim_{n+1} q \ \Leftrightarrow\ p \sim_n q$ et $\forall a \in \Sigma,\ \delta(p, a) \sim_n \delta(q, a)$
- Si $\sim_n = \sim_{n+1}$ alors $\sim = \sim_n$
- $\sim = \sim_{|Q|-2}$

<u>Rq</u> : $\sim_0$ a pour classes d'équivalences F et $Q \setminus F$.

4 Monoïdes et congruences

- Soit M un monoïde fini, $\varphi : \Sigma^* \to M$ un morphisme, et $L \subseteq \Sigma^*$. On donne les définitions suivantes :

- L est **reconnu** (ou saturé) par φ si $L = \varphi^{-1}(\varphi(L))$ ($\subseteq$ est toujours vraie)
- Un langage est reconnu par un monoïde fini si il existe un morphisme de ce monoïde qui reconnait ce langage.
- Un langage est **reconnaissable par morphisme** si il existe un monoïde fini qui reconnait ce langage.

- Soit $\mathcal{A} = (Q, \delta, i, F)$ un automate DC. Le **monoïde de transitions** de $\mathcal{A}$ est le sous monoïde de $(Q^Q, *)$ engendré par les applications $\delta_a : Q \to Q$ $(a \in \Sigma)$ définies par $\delta_a(q) = \delta(q, a)$ et muni de la composition.

Lemme : Le monoïde de transitions de $\mathcal{A}$ reconnait $\mathcal{L}(\mathcal{A})$.

Rq : Si L est reconnu par $\varphi : \Sigma^* \to M$, alors l'automate $\mathcal{A} = (M, \delta, \varphi(\epsilon), \varphi(L))$, avec $\delta(m, a) := m{\cdot}\varphi(a)$ reconnaît L.
Théorème : Un langage est reconnaissable par morphisme ssi il est reconnaissable par automate.

Rq : On en déduit que $\mathrm{Rec}(\Sigma^*)$ est fermé par morphisme inverse.

- Une relation d'équivalence $\equiv$ sur Σ^* s'appelle **congruence** si : $u \equiv v \Rightarrow \forall x, y\ xuy \equiv xvy$.

- Soit $L \subseteq \Sigma^*$ et $\equiv$ une congruence sur Σ^*. L est **saturé** par $\equiv$ si : $\forall u, v \in \Sigma^*, u \equiv v \Rightarrow (u \in L \Leftrightarrow v \in L)$

Théorème : Soit $L \subseteq \Sigma^*$. L est reconnaissable ssi L est saturé par une congruence d'index fini.

- Soit $L \subseteq \Sigma^*$, on considère $\equiv_L$ définie par $u \equiv_L v$ si $\forall x, y \in \Sigma^*, xuy \in L \Leftrightarrow xvy \in L$. On a les propriétés suivantes :

- $\equiv_L$ est une congruence qui sature L
- $\equiv_L$ est la plus grande congruence qui sature L
- L est reconnaissable ss $\equiv_L$ est d'index fini.

- Soit $L \subseteq \Sigma^*$, on considère $M_L = \Sigma^*/{\equiv_L}$ le **monoïde syntaxique** de L. On a les propriétés suivantes :

- M_L est le monoïde des transitions de l'automate minimal de L.
- M_L divise (i.e est quotient d'un sous-monoïde) tout monoïde qui reconnaît L.