

Définition 2.1 - *automate fini déterministe*

Un automate fini déterministe \mathcal{A} est défini par un quintuplet $(\Sigma, Q, q_0, F, \delta)$, où :

1. Σ est un alphabet fini ;
2. Q est un *ensemble fini d'états* ;
3. $q_0 \in Q$ est l'*état initial* ;
4. $F \subset Q$ est un *ensemble d'états finaux* ;
5. δ est une application d'une partie de $Q \times \Sigma$ dans Q est la *fonction de transition*.

Il est commun de représenter par un tableau à double entrées, dit *table de transition*, les valeurs prises par δ .

Définition 2.2 - *chemin, étiquette d'un chemin*

Un *chemin* dans un automate est une suite finie d'états (q_1, \dots, q_n) telle qu'il existe a_1, \dots, a_{n-1} dans Σ , tel que :

$$\forall i \in \llbracket 1, n \rrbracket, \delta(q_i, a_i) = q_{i+1}$$

Assertion que l'on notera :

$$q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} \dots \xrightarrow{a_{n-1}} q_n$$

L'*étiquette du chemin* est alors le mot $a_1 \dots a_{n-1}$.

Définition 2.3 - *chemin acceptant*

Un chemin (q_1, \dots, q_n) d'un automate \mathcal{A} est *acceptant* lorsque q_1 est l'état initial (ou un état initial si AFND) de \mathcal{A} et q_n est un état final de \mathcal{A} .

On dit alors que l'étiquette de (q_1, \dots, q_n) , qui est un mot, *est reconnue par \mathcal{A}* .

Définition 2.4 - *langage reconnu*

L'ensemble des mots reconnus par \mathcal{A} un automate fini est noté $\mathcal{L}(\mathcal{A})$ et est appelé *langage reconnu par \mathcal{A}* .

Définition 2.5 - *état accessible*

Un état q d'un automate fini \mathcal{A} est dit accessible lorsqu'il existe un chemin de \mathcal{A} de l'état initial (ou d'un état initial) de \mathcal{A} menant à q .

Définition 2.6 - *état co-accessible*

Un état q d'un automate fini \mathcal{A} est dit co-accessible lorsqu'il existe un chemin de \mathcal{A} reliant q à un état final de \mathcal{A} .

Définition 2.7 - *état utile*

Un état d'un automate fini est dit utile s'il est accessible et co-accessible.

Définition 2.8 - *automate fini émondé*

La présence d'états non utiles (non accessibles ou non co-accessibles) n'altère pas le langage reconnu par un automate fini \mathcal{A} .

On dit alors qu'un automate \mathcal{A}' est *émondé* s'il ne contient que des états utiles.

Définition 2.9 - *automate des parties d'un AFND*

Soit $\mathcal{A}_{\text{ND}} = (\Sigma, Q, I, F, \delta)$ un automate fini non déterministe. On appelle *automate des parties de \mathcal{A}_{ND}* , noté $\mathcal{A}_{\text{D}} = (\Sigma, Q_{\text{D}}, q_{0,\text{D}}, F_{\text{D}}, \delta_{\text{D}})$ tel que :

1. $Q_{\text{D}} = \mathcal{P}(Q)$;
2. $q_{0,\text{D}} = I$;
3. $F_{\text{D}} = \{P \in Q_{\text{D}}, P \cap F \neq \emptyset\}$, l'ensemble des états de \mathcal{A}_{D} contenant au moins un état final de \mathcal{A}_{ND} .
4. $\delta_{\text{D}} : \begin{array}{ccc} Q_{\text{D}} \times \Sigma & \rightarrow & Q_{\text{D}} \\ (P, a) & \mapsto & \{q \in Q, \exists p \in P, q \in \delta(p, a)\} = \bigcup_{p \in P} \{q \in Q, q \in \delta(p, a)\} \end{array}$

Pour $P \in Q_{\text{D}}$ et $a \in \Sigma$, $\delta_{\text{D}}(P, a)$ est l'ensemble des états de Q accessibles en lisant a depuis un élément de P . \mathcal{A}_{D} est alors un automate fini déterministe.

Définition 2.10 - *ϵ -fermeture d'un état d'un ϵ -AFND*

Soit $(\Sigma, Q, I, F, \delta)$ un ϵ -AFND. On appelle ϵ -fermeture d'un état q est l'ensemble des états accessibles depuis un chemin dont l'étiquette est le mot vide. On la note $\epsilon\text{-}F(q)$.

Définition 2.11 - *langage local*

Soit Σ un alphabet fini et $L \subset \Sigma^*$ un langage. On dit que L est *local* s'il existe $P \subset \Sigma$, $S \subset \Sigma$ et $F \in \Sigma_2$ telles que :

$$L \setminus \{\epsilon\} = \left(P\Sigma^* \cap \Sigma^*S \right) \setminus \left(\Sigma^*(\Sigma_2 \setminus F)\Sigma^* \right)$$

Définition 2.12 - langages locaux d'un langage

Soit Σ un alphabet fini et $L \in \Sigma^*$ un langage. On définit les *langages locaux de L* :

1. $\epsilon(L) = \{\epsilon\} \cap L$;
2. $P(L) = \{a \in \Sigma, \{a\}\Sigma^* \cap L \neq \emptyset\}$;
3. $S(L) = \{a \in \Sigma, \Sigma^*\{a\} \cap L \neq \emptyset\}$;
4. $F(L) = \{ab \in \Sigma_2, \Sigma^*\{ab\}\Sigma^* \cap L \neq \emptyset\}$.

Proposition 2.13 - caractérisation par les langages locaux

Soit Σ un alphabet fini et $L \in \Sigma^*$ un langage. L est local si et seulement si l'égalité suivante est vérifiée :

$$L = \left(\left(P(L)\Sigma^* \cap \Sigma^*S(L) \right) \setminus \left(\Sigma^*(\Sigma_2 \setminus F(L))\Sigma^* \right) \right) \cup \epsilon(L)$$

Définition 2.14 - expression régulière linéaire

Une expression régulière dans laquelle chaque lettre apparaît au plus une fois est dite *linéaire*.

Proposition 2.15 - concernant les expressions régulières linéaires

Tout langage local régulier est dénoté par une expression régulière linéaire.

Définition 2.16 - automate local

Un automate fini déterministe est *local* lorsque seules des transitions de même étiquette arrivent au même état.

Définition 2.17 - automate standard

Un automate fini déterministe est dit *standard* si aucune transition n'arrive en son état initial.

Proposition 2.18 - concernant les automates locaux standards

Tout langage local régulier est reconnu par un automate local standard.

Définition 2.19 - *automate fini généralisé*

Un *automate fini généralisé* \mathcal{A} est d'abord défini par un quintuplet $(\Sigma, Q, q_0, f_0, \delta)$, avec :

1. Σ est un alphabet fini ;
2. Q est un *ensemble fini d'états* ;
3. $q_0 \in Q$ est l'*état initial* ;
4. $q_f \in Q$ est l'*état final* ;
5. δ , la *fonction de transition*, est une application d'une partie de $Q \times E(\Sigma)$ dans $\mathcal{P}(Q)$. Ici, $E(\Sigma)$ désigne l'ensemble des expressions régulières sur Σ .

Puis :

1. dont aucune transition n'arrive en q_0 ;
2. dont aucune transition ne part de q_f .

Définition 2.20 - *automate reconnaissant l'union de deux langages reconnaissables*

Soit $A_1 = (\Sigma, Q_1, q_{0,1}, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, q_{0,2}, F_2, \delta_2)$ deux automates finis déterministes. Un *automate fini déterministe reconnaissant le langage $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$* est l'automate produit $A = (\Sigma, Q, q_0, F, \delta)$, où :

1. $Q = Q_1 \times Q_2$
2. $q_0 = (q_{0,1}, q_{0,2})$;
3. $F = Q_1 \times F_2 \cup F_1 \times Q_2$;
4. $\delta((q, p), a) = (\delta_1(q, a), \delta_2(p, a))$.

Définition 2.21 - *automate reconnaissant l'intersection de deux langages reconnaissables*

Soit $A_1 = (\Sigma, Q_1, q_{0,1}, F_1, \delta_1)$ et $A_2 = (\Sigma, Q_2, q_{0,2}, F_2, \delta_2)$ deux automates finis déterministes. Un *automate fini déterministe reconnaissant le langage $\mathcal{L}(A_1) \cap \mathcal{L}(A_2)$* est l'automate produit $A = (\Sigma, Q, q_0, F, \delta)$, où :

1. $Q = Q_1 \times Q_2$
2. $q_0 = (q_{0,1}, q_{0,2})$;
3. $F = F_1 \times F_2$;
4. $\delta((q, p), a) = (\delta_1(q, a), \delta_2(p, a))$.

Définition 2.22 - *automate reconnaissant le complémentaire d'un langage reconnaissable*

Soit $A_1 = (\Sigma, Q_1, q_{0,1}, F_1, \delta_1)$ un automate fini déterministe et **complet**. Un *automate fini déterministe reconnaissant* $\mathcal{L}_{\Sigma^*}(A_1)$ est l'automate $A = (\Sigma, Q, q_0, F, \delta)$ où :

1. $Q = Q_1$;
2. $q_0 = q_{0,1}$;
3. $F = Q_1 \setminus F_1$;
4. $\delta = \delta_1$.

Définition 2.23 - *automate reconnaissant le complémentaire d'un langage reconnaissable*

Soit $A_1 = (\Sigma, Q_1, q_{0,1}, F_1, \delta_1)$ un automate fini déterministe et **complet**. Un *automate fini déterministe reconnaissant* $\mathcal{L}_{\Sigma^*}(A_1)$ est l'automate $A = (\Sigma, Q, q_0, F, \delta)$ où :

1. $Q = Q_1$;
2. $q_0 = q_{0,1}$;
3. $F = Q_1 \setminus F_1$;
4. $\delta = \delta_1$.

Question 2.24 - *D'après Mines-Telecom 2023*

Soit L un langage régulier sur un alphabet Σ et $A = (\Sigma, Q, q_0, F, \delta)$ un automate fini déterministe complet reconnaissant le langage L .

1. Soit $q, q' \in Q$ et $L_{q,q'} = \{\omega \in \Sigma^*, \delta^*(q, \omega) = q'\}$. Montrer que $L_{q,q'}$ est rationnel.
2. En déduire que $\sqrt{L} = \{\omega, \omega^2 \in L\}$ est régulier.

Théorème 2.25 - *lemme de l'étoile*

Soit L un langage régulier. Il existe $N \in \mathbb{N}$ tel que :

$$\forall u \in L, |u| \geq N \implies \exists (x, y, z) \in (\Sigma^*)^3, \begin{cases} u = xyz \\ y \neq \epsilon \\ |xy| \leq N \\ \{x\}\{y\}^*\{z\} \subset L \end{cases}$$

Théorème 2.26 - *lemme d'Arden*

Soit A et B deux langages sur un même alphabet Σ . Si $\epsilon \notin A$, l'équation $L = (A \cdot L) \cup B$ admet pour unique solution le langage A^*B .