Contenu

Mots et langages

Definition (Alphabet)

Un alphabet est un ensemble noté Σ dont les éléments sont appelés *lettres* ou *symboles*.

Definition (Mot)

Un *mot* sur Σ est une suite (ou chaîne) finie de lettres de Σ.

- Le mot *vide* est noté ϵ.
- La longueur d'un mot u est notée $|u|$. ϵ est le mot de longueur nulle.
- L'ensemble des mots finis sur Σ est noté Σ^*.

Mots et langages

Definition (Concaténation de mots)

Σ^* est muni d'une opération binaire, la *concaténation*. La concaténation du mot u avec le mot v et dénotée $u \cdot v$ ou simplement uv en omettant le $\cdot$, est le mot obtenu en ajoutant v à la suite de u.
Cette opération est :

- *associative* et
- possède le mot vide comme *élément neutre*.

Mots et langages

Definition (Langage)

Une partie de Σ^* est appelée *langage* sur Σ.

Definition (Constructeurs de langages)

- La *concaténation* : $L \cdot L' = \{u \cdot u' : u \in L \wedge u' \in L'\}$

Remarque

- $L \cdot \emptyset = \emptyset \cdot L = \emptyset$
- $L \cdot \Sigma^* \neq \Sigma^* \neq \Sigma^* \cdot L$
- Si $\epsilon \in L$ alors $L \cdot \Sigma^* = \Sigma^* \cdot L = \Sigma^*$
- $L \cdot \{\epsilon\} = \{\epsilon\} \cdot L = L$

- La *fermeture de Kleene* : $L^* = \bigcup_{n \in \mathbb{N}} L^n$ où

$$L^n = \{u_1 u_2 \ldots u_n : u_1, u_2, \ldots u_n \in L\}$$

Mots et langages

Example

Soit les langages $L_1 = \{bb\}$ et $L_2 = \{\epsilon, bb, bbbb\}$. Les langages L_1^* et L_2^* sont formés, tous les deux, de tous les mots sur l'alphabet $\{b\}$ qui contiennent un nombre pairs de b.

Example

Le langage $L_1 = \{a, b\}^*\{bb\}\{a, b\}^*$ est formé de tous les mots sur l'alphabet $\{a, b\}$ qui contiennent le facteur bb.

Example

Le langage $\{aa, ab, ba, bb\}^*$ est formé de tous les mots sur l'alphabet $\{a, b\}$ de longueur paire : $\{a, b\}^* \setminus \{aa, ab, ba, bb\}^*$ est formé de tous les mots sur l'alphabet $\{a, b\}$ de longueur impaire. De façon alternative : $\{a, b\}\{aa, ab, ba, bb\}^*$

Mots et langages

Definition (Expressions régulières)

- ϵ, $\emptyset$ et a (pour tout $a \in \Sigma$) sont des ER
- Si u, v sont des ER alors $u \cdot v$, $u + v$ et u^* sont des ER

Definition (Sémantique des expressions régulières)

La sémantique des expressions régulières est donnée par l'application $L : ER \rightarrow \mathcal{P}(\Sigma^*)$ définie par

- $L(\epsilon) = \{\epsilon\}$ et $L(\emptyset) = \emptyset$
- Pour tout $a \in \Sigma$, $L(a) = \{a\}$
- Si u, v sont des ER alors $L(u \cdot v) = L(u) \cdot L(v)$, $L(u + v) = L(u) \cup L(v)$ et $L(u^*) = L(u)^*$

Definition (Langages réguliers)

L est régulier ssi il existe une ER u telle que $L(u) = L$

Automates d'états finis

Definition (Automates à états finis déterministe)

Un *automate à états finis déterministe* (AFD) est un quintuplet $(S, \Sigma, \delta, s_0, F)$ où

- S est un ensemble fini d'*états*
- Σ, un alphabet
- $\delta : S \times \Sigma \to S$ est appelée *fonction de transition* .
- s_0, un *état initial*
- Un ensemble F d'*états finaux ou acceptants* avec $F \subseteq S$

Automates d'états finis

Definition (Exécution d'un AFD)

On étend δ à une fonction $\Delta : S \times \Sigma^* \to S$:

- $\Delta(q, \epsilon) = q$
- $\Delta(q, xa) = \delta(\Delta(q, x), a)$.

Soit un mot $w = x_0 x_1 \ldots x_n$. La suite $s_0 q_1 \ldots q_{n+1}$ telle que

$$q_{i+1} = \Delta(s_0, x_0 x_1 \ldots x_i),\ \text{pour tout } 0 \leq i \leq n$$

est appelé exécution de w.

Example (Exécution de 110101 d'un AFD qui reconnait $\Sigma^* 01 \Sigma^*$)

Automates d'états finis

Definition (Langage reconnu par un AFD)

Le langage $L(\mathcal{A})$ défini par :

$$L(\mathcal{A}) = \{w \in \Sigma^* : \Delta(s_0, w) \in F\}$$

est appelé le langage reconnu par $\mathcal{A}$.

Example (AFD qui reconnait $\Sigma^*01\Sigma^*$)

Automates d'états finis

Definition (Automates à états finis non-déterministe)

Un *automate à états finis non-déterministe* (AFN) est un quintuplet $(S, \Sigma, \delta, s_0, F)$ où

- S est un ensemble fini d'*états*
- Σ, un alphabet
- $\delta : S \times \Sigma \rightarrow \mathcal{P}(S)$ est appelée *fonction de transition* .
- s_0, un *état initial*
- Un ensemble F d'*états finaux ou acceptants* avec $F \subseteq S$

Automates d'états finis

Definition (Exécution d'un AFN)

On étend δ à une fonction $\Delta : S \times \Sigma^* \to \mathcal{P}(S)$:

- $\Delta(q, \epsilon) = \{q\}$
- Soit $w = xa$ où $x \in \Sigma^*$ et $a \in \Sigma$ alors

$$\Delta(q, w) = \bigcup_{i=1}^{k} \delta(p_i, a)$$

  où $\Delta(q, x) = \{p_1, p_2, \ldots, p_k\}$.

Soit un mot $w = x_0 x_1 \ldots x_n$. Une suite $s_0 q_1 \ldots q_{n+1}$ telle que

$$q_{i+1} \in \Delta(s_0, x_0 x_1 \ldots x_i), \text{ pour tout } 0 \leq i \leq n$$

est appelé exécution de w.

Exemple (Exécution de 110101 d'un AFN qui reconnait $\Sigma^* 01 \Sigma^*$)

Automates d'états finis

Definition (Langage reconnu par un AFN)

Le langage $L(\mathcal{A})$ défini par :

$$L(\mathcal{A}) = \{w \in \Sigma^* : \Delta(s_0, w) \cap F \neq \emptyset\}$$

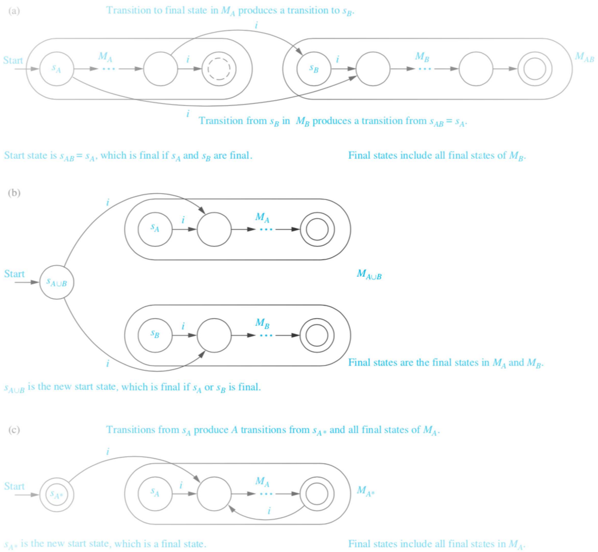
est appelé le langage reconnu par $\mathcal{A}$.

Example (AFN qui reconnait $\Sigma^* 01 \Sigma^*$)

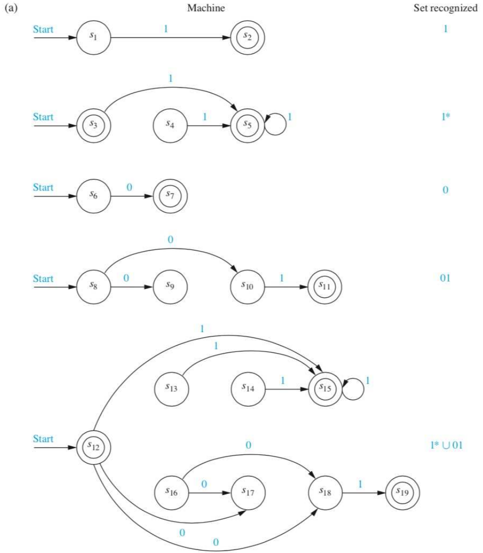
Theorem (Kleene)

Un langage est reconnaissable par un AFD si et seulement s'il est régulier.

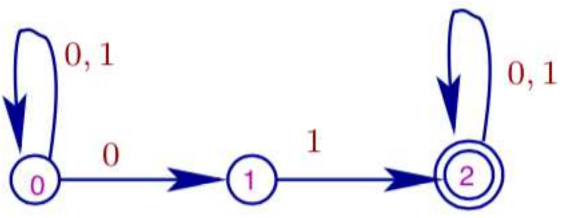
Tout lang. régulier est reconnaissable par un AFN

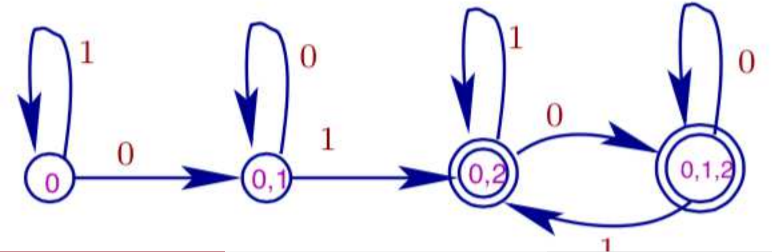


Exemple : AFN reconnaissant $1^* + 01$

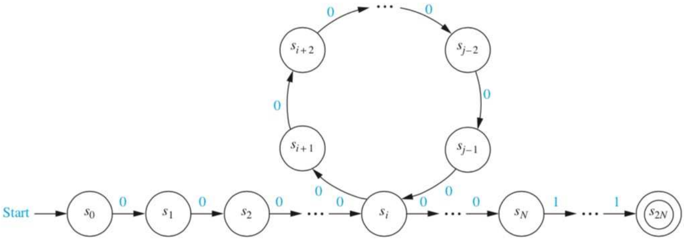


Déterminisation (Rabin-Scott, 1959)





$L = \{0^n 1^n : n \in \mathbb{N}\}$ n'est pas régulier



- N : nombre d'états de l'AFD qui reconnait $L = \{0^n 1^n : n \in \mathbb{N}\}$
- On considère une exécution de $0^N 1^N$
- $s_0 \xrightarrow{u} s_i$
- $s_i \xrightarrow{v} s_j$
- $s_j \xrightarrow{w} s_{2N}$
- Pour tout $n \in \mathbb{N}$, on a $0^i 0^{(n(j-i)} 0^{N-j} 1^N \in L$ (contradiction)

Problèmes de décisions

D'autres propriétés de fermeture

1. Intersection
2. Complémentation

Problème du vide

Étant donné un automate fini $\mathcal{A}$ on on peut décider si $L(\mathcal{A}) = \emptyset$

Problème d'inclusion

Étant donné des automate fini $\mathcal{A}$ et $\mathcal{B}$ on on peut décider si $L(\mathcal{A}) \subseteq L(\mathcal{B})$