Théorie des langages

March 2, 2018

1 Alphabet, mot et langage

Un alphabet est un ensemble fini de symbole Σ.
Exemple

- $\Sigma = \{0, 1\}$
- $\Sigma = \{a, b, c, ..., z\}$
- $\Sigma = \{a, b, c\}$
- Σ = ensemble des mots de la langue française

$\boxed{\Sigma^* \text{= ensemble de tous les mots finie qu'on peut écrire avec les symboles dans } \Sigma}$

- $\Sigma = \{a\}$ Dans ce cas, $\Sigma^* = \{\varepsilon, a, aa, aaa, ...\}$ où ε est un mot vide (aucun symbole)
- $\Sigma = \{0, 1\}$ Dans ce cas, $\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, ...\}$

Remarque

- ε est le mot vide de longueur 0
- On pose $|u|$ la longueur du mot $u \in \Sigma^* |\varepsilon| = 0$

Définition

Si u et v sont deux mots dans Σ^*le mot uv est obtenu en concaténant u et v (v est après u)
Remarques

- $|uv| = |u| + |v|$
- Σ^* muni de la concaténation est un monoïde car la concaténation est :
  - associative $(uv)w = u(vw)$
  - possède un élément neutre ε ($\varepsilon u = u = u\varepsilon$)

  Attention : Il n'y a pas d'inverse !

Notations

- Si $u \in \Sigma^*$, $\left| \begin{array}{c} u^n \\ u^0 = E \end{array} \right.$ est le mot $uuu...u$ (n fois u)

- Si $u \in \Sigma^*$ tel que $u = xy$, $\left| \begin{array}{c} u/y = x \\ uy^{-1} = x \\ x\backslash u = y \\ x^{-1}u = y \end{array} \right.$ Attention : Notations

1.1 Propriétés combinatoires

Définition

Si $u = xy$ alors $\left| \begin{array}{l} x \text{ est un préfixe de } u \\ y \text{ est un suffixe de } w \end{array} \right.$ où $u, x, y \in \Sigma^*$

Objectif

Caractériser les mots qui ont un préfixe et un suffixe égaux.

Lemme

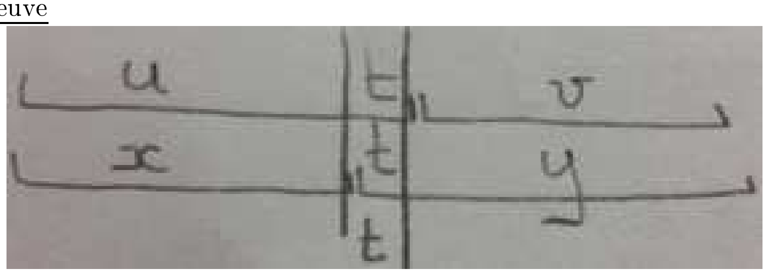
Si u, v, x, y sont dans Σ^* et :

- $|u| \geq |x|$

- $uv = xy$

alors il existe un mot $t \in \Sigma^*$ tel que $u = xt$ et $y = tv$.

Preuve



Théorème

Si $u, v, w \in \Sigma^*$ tels que

- $u \neq \varepsilon$

- $uv = vw$

alors, il existe deux mots x et y dans Σ^* et un entier naturel k tels que $u = xy$ et $v = (xy)^k x$ et $w = yx$.

Preuve

Si $|u| > |v|$ alors d'après le lemme précédent, il existe $t \in \Sigma^*$ tel que $u = vt$, et $w = tv$ donc le théorème est prouvé avec $k = 0$: $u = xy$, $v = (xy)^0 x = x$ et $w = yx$.

Si $|u| < |v|$ alors d'après le lemme précédent, il existe $t \in \Sigma^*$tel que $v = ut$, et $v = tw$ donc $ut = tw$ et $|v| > |t|$.

Si $|v| = 2$, alors $|u| = 1$ (car $0 < |u| < |v|$). Or, $v = ut = tw$ (c'est ce qu'on a obtenu en invoquant le lemme). On en déduit que u, t et w sont composés d'un seul symbole et que $t = u = w$. Donc, en passant $x = t$, on a : $v = xx$, $w = x$ ou encore $u = xy$, $v = (xy)$, $w = yx$ avec $y = \varepsilon$.

Théorème

Si $u, v \in \Sigma^*$

- $uv = vu$
- $u \neq \varepsilon$, $v \neq \varepsilon$

alors $u = x^i$ et $v = x^j$ pour un certain $x \in \Sigma^*$.

1.2 Langages

Définition

Un langage est un sous-ensemble de Σ^*.

Exemples

- $\Sigma = \{a, b\}$ le langage de tous les mots qui commencent par a

$$L = \{a, aa, ab, aaa, aab, aba, abb, ...\} = a\Sigma^*(concaténation)$$

- Le langage de tous les mots qui se terminent par b

$$L' = \{b, ab, bb, aab, abb, bab, bbb, ...\} = \Sigma^* b$$

- Le langage des mots qui commencent par a et se finissent par b

$$L'' = (a\Sigma^*) \cap (\Sigma^* b) = a\Sigma^* b$$

- Le langage de tous les mots qui ont un nombre pair de a

$$L''' = \{u \in \Sigma^* : |u|_a \equiv 0 \, mod \, 2\}$$

où u nombre d'occurence de a dans u

(Remarque : $\sum\limits_{a \in \Sigma} |u|_a = |u|$)

Opérations sur les langages

- $L \cup L'$, $L \cap L'$ et $L \backslash L'$ sont les opérations ensemblistes habituelles. L'opération $L \cup L'$ est aussi notées $L + L'$
- On peut concaténer deux langages

$$LL' = \{xy : x \in L \, et \, y \in L'\}$$

- $LLL...L = L^n$ et $L^0 = \{\varepsilon\}$
- L étoile (ou fermeture de Kleene)

$$L^* = \bigcup_{i=0}^{+\infty} L'$$

1.3 Langage rationnels

Les langages rationnels sont les langages (sur Σ, ou encore, ce sont des sous-ensembles de Σ^*) qui peuvent être construits de manière récursive en utilisant

- La réunion
- La concaténation
- La fermeture de Kleene

L'initialisation de cette procédure récursive est faite avec des langages élémentaires, les langages suivants:

- $\emptyset$, $\{\varepsilon\}$
- $\{a\}$ ou $a \in \Sigma$

<u>Définition</u>

1. $\emptyset$ et $\{\varepsilon\}$ sont des langages rationnels
2. $\{a\}$ où $a \in \Sigma$ sont des langages rationnels
3. Si L_1 et L_2 sont des langages rationnels alors L_1L_2 et $L_1 \cup L_2$ sont des langages rationnels
4. Si L est un langage rationnel alors L^* est un langage rationnel

<u>Propriété</u>

Puisque la construction d'un langage rationnel se fait en respectant un nombre fini de fois certaines opérations. On peut décrire ces langages comme des mots sur certains alphabet.

<u>Exemple</u>

- ε va noter $\{\varepsilon\}$
- a va noter $\{a\}$
- $\emptyset$ va noter $\emptyset$
- $a + b$ va noter le langage $\{a\} \cup \{b\}$ (et $L_1 + L_2$ note $L_1 \cup L_2$)
- ab note le langage $\{ab\}$
- a^* note le langage $\{a\}^*$
- $aa^* + b$ note $\{a\}\{a\}*\cup\{b\}$ où $\{a\}^* = \{a^n : n \geq 0\}$, $\{a\}\{a\}^* = \{a^n : n \geq 1\}$ et $\{b, a, aa, aaa, aaaa, ...\}$
- $\{re + dé\}^*$montrer

  $\{re, dé\}$

  $\{\epsilon, re, dé, rere, redé, dére, dédé, rerere, reredé, redére, redédé, dérere, déredé, dédére, dédédé, ...\}$

  $\{montrer, remontrer, démontrer, redémontrer, reremonter, dédémontrer\}$

Définition

Deux expressions rationnelles e_1 et e_2 sont dites équivalentes si elles dénotent le même langage.

Exemple

$(a+b)^*$ et $(a^*b)^*a^*$ sont équivalente (le langage qu'elles dénotent est $\{a,b\}^*$)

(a^*b)Des mots avec des a (peut-être aucun) suivi d'un unique b

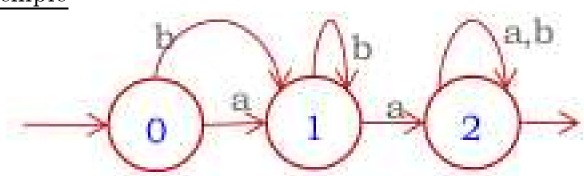
2 Automate finis déterministes

Définition

Un automate fini déterministe complet est un quintuplet $(\Sigma, Q, q_0, F, \delta)$ où

- Σ est un alphabet
- Q est un ensemble d'état
- $q_0 \in Q$ est l'état initial
- $F \subset Q$ est l'ensemble des états finaux
- $\delta.Q * F \to Q$ est une application

Exemple



$\Sigma = \{a,b\}$
$Q = \{0,1,2\}$
$q_0 = 0$
$F = \{2\}$

$\delta(0,a) = 1 \quad \delta(0,b) = 1 \quad \delta(1,a) = 2 \quad \delta(1,b) = 1 \quad \delta(2,a) = 2 \quad \delta(2,b) = 2$

Définition

Un calcul d est une suite $t_0, ..., t_u$ de transitions (flèches dans le dessin) telle que la destination t_{i-1} est d'origine de t_i pour tout i. L'étiquette du calcul est obtenue en concaténant les symboles de chaque transition du calcul (dans l'ordre). On dit qu'un calcul est réussi si il se termine dans un état final (et commence dans l'état initial).

Définition

Le langage reconnu par l'automate A, noté $L(A)$ est l'ensemble des étiquettes des calculs réussis dans A.

Exemple

$(a+b)b^*a(a+b)^*$ (Le langage reconnu par l'automate dans l'exemple précédent)

Définition

Un langage est dit reconnaissable si il existe un automate fini déterministe complet qui le reconnaît.

Algorithme : décide si le mot $u = u_1, ..., u_l$ est reconnu par A.

Pour $q \leftarrow q_0$ et $i \leftarrow 1$
Tant que $i \leq l$ Faire
    Poser $q \leftarrow \delta(q, u_i)$
    Poser $i \leftarrow i + 1$
Fin tant que
Si $q \in F$ alors u est reconnu
Sinon u n'est pas reconnu
    (complexité $= l$)

Remarque

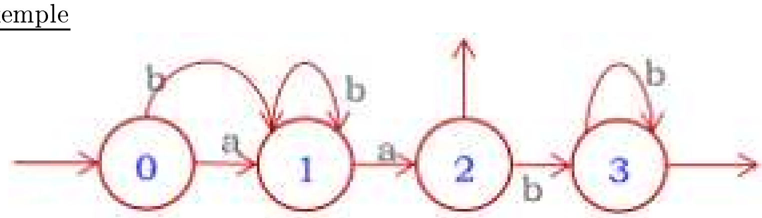
On peut généraliser δ par une application $\delta^* : Q * \Sigma^* \rightarrow Q$:

- $\delta^*(q, a) = \delta(q, a)$ si $a \in \Sigma$
- $\delta^*(q, au) = \delta^*(\delta(q, a), u)$ si $a \in \Sigma$ et $u \in \Sigma^*$

Définition

Un automate fini déterministe se définit comme un automate fini déterministe complet à la différence que δ n'est plus forcément une application mais une fonction.

Exemple



$L(A) = (a + b)b^*a + (a + b)b^*abb^* = (a + b)b^*a(\varepsilon + bb^*) = (a + b)b^*(a + cbb^*)$

Algorithme (même obj. que au-dessus)

Pour $q \leftarrow q_0$ et $i \leftarrow 1$
Tant que $i \leq l$ Faire
Si $\delta(q, u_i)$ existe alors
    Poser $q \leftarrow \delta(q, u_i)$
    Poser $i \leftarrow i + 1$
Sinon u n'est pas reconnu
Fin tant que
Si $q \in F$ alors u est reconnu
Sinon u n'est pas reconnu

Définition

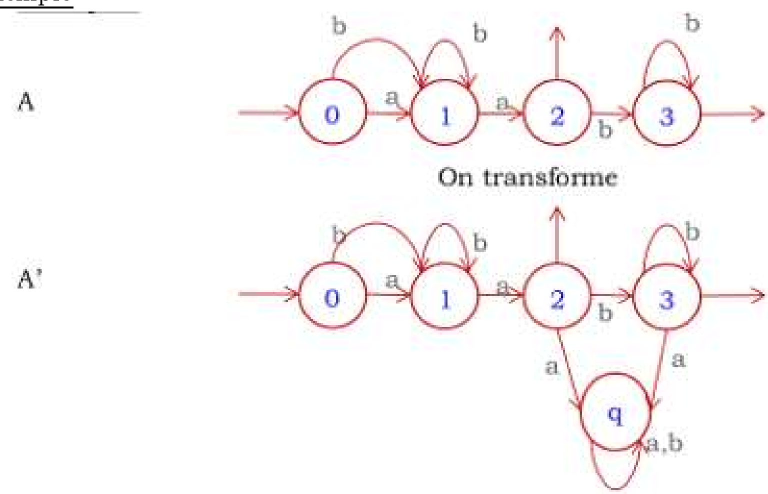
Deux automates A et A' sont équivalents si $L(A) = L(A')$

Théorème

Si A est un automate fini déterministe, alors il existe un automate fini déterministe complet A' tel que $L(A) = L(A')$

Preuve

On ajoute un état supplémentaire $\tilde{q}$ qui sera la destination de toutes transitions qui manquent pour rendre l'automate complet.
Exemple

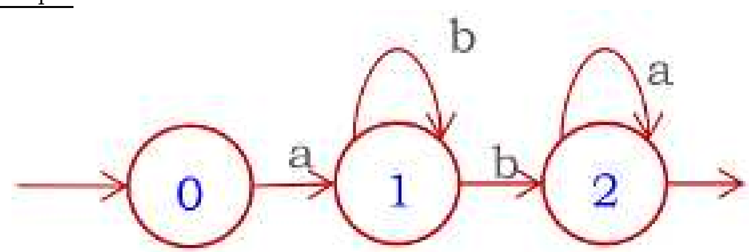


2.1 Automates finis non-déterministes

Définition
Un automate fini non-déterministe est un quintuplet $(\Sigma, Q, I, F, \delta)$

- Σ est un alphabet
- Q est un ensemble des états
- $I \subset Q$ est l'ensemble des états initiaux
- $F \subset Q$ est l'ensemble des états finaux
- $\delta . Q * F \to P(Q)$

Exemple



$I = \{0\}$
$F = \{2\}$
$\delta(0, a) = \{1\} \quad \delta(1, b) = \{1, 2\} \quad \delta(2, a) = \{2\}$
Remarques

- Les automates finis non déterministes généralisent les automates finis déterministes (donc attention un automate fini déterministe est nécessairement non-déterministe).

- Les notions de calcul d'étiquette et de calcul réussis se généralisent naturellement

- On peut aussi définir δ^*

  - $\delta^*(q, a) = \delta(q, a)$ si $a \in \Sigma$
  - $\delta^*(q, au) = \bigcup\limits_{r \in \delta(q,a)} \delta^*(r, u)$

Théorème

Si A est un automate fini non-déterministe, il existe un automate fini déterministe A tel que $L(A) = L(A')$

Théorème : (Contor)

Il n'existe plus de bijection entre E et $P(E)$

Preuve

Supposons que $\Phi : E \to P(E)$ est une bijection

Soit F le sous-ensemble de E contenant les $x \in E$ tels que $x \notin \Phi(a)$

$$F = \{x \in E : x \notin \Phi(x)\} \subset E$$

Il y a un x tel que $\Phi(x) = F$ car Φ est une bijection.

- Si $x \in F$ alors $x \notin \Phi(x)$ donc $x \notin F$

- Si $x \notin F$ alors $x \notin \Phi(x)$ donc $x \in F$.

On en déduit que Σ^* - ensemble des mots - est "plus petit" que $P(\Sigma^*)$ - ensemble des langages -

Par conséquent

- L'ensemble des langages rationnels est dénombrable (chaque expression rationnelle dénote un langage rationnel et chaque langage rationnel est dénoté par au moins une expression rationnelles).

- L'ensemble des langages sur Σ^* n'est pas dénombrable (car c'est $P(\Sigma^*)$)

Donc, il existe des langages qui ne sont pas rationnels.