909

Langages rationnels. Exemples et applications.

Les langages rationnels sont au premier niveau de la hiérarchie de Chomsky. Ils servent notamment dans les éditeurs de texte, avec la recherche de motifs.

## I – Langages rationnels et langages reconnaissables :

### 1) Langages rationnels

Soit $A$ un alphabet.

[C] p.15 Déf 1 : Un langage $L$ sur $A$ est un ensemble de mots sur $A$.

[C] p.15 Déf 2 : L'union est l'opération qui à deux langages $L$ et $L'$ associe le langage $L \cup L' = \{u \mid u \in L \text{ ou } u \in L'\}$

• Le produit est l'opération qui à $L$ et $L'$ associe le langage $LL' = \{uv \mid u \in L, v \in L'\}$

• Soit $L$ un langage. On définit l'opération $*$ :
$L^0 = \{\varepsilon\}$, $\forall i \in \mathbb{N}$, $L^{i+1} = LL^i$, $L^* = \bigcup_{i \geq 0} L^i$.

[C] p.15 Ex 3 : • Soit $L = \{u \in A^* \mid |u| \text{ est paire}\}$ où $|u|$ = longueur de $u$.
Soit $L' = \{u \in A^* \mid |u| \text{ est impaire}\}$. Alors :
$L + L' = A^*$ ; $LL' = L'L = L'$

[C] p.15 • Soit $L = \{a^n b \mid n \geq 0\}$. $L^*$ est l'ensemble des mots qui se terminent par un $b$.

[C] p.34 Déf 4 : On définit la classe $\mathcal{E}$ des expressions rationnelles par induction : • $\emptyset \in \mathcal{E}$ et $\forall a \in A$, $a \in \mathcal{E}$

Il faut parenthéser • $\forall E, E' \in \mathcal{E}$, les expressions $E + E'$, $E \cdot E'$ et $E^* \in \mathcal{E}$.

[S] p.126 Rq : Cette définition inductive est non-ambiguë.

[S] p.126 Déf 5 : Sémantique. À toute expression rationnelle $E$, on associe le langage rationnel $L(E)$ par induction :

• $L(\emptyset) = \emptyset$ ; $\forall a \in A$, $L(a) = a$

• $\forall E, E' \in \mathcal{E}$, $L(E + E') = L(E) + L(E')$ ; $L(E \cdot E') = L(E)\, L(E')$
$L(E^*) = L(E)^*$.

[S] p.126 Déf 6 : Deux expressions rationnelles $E$ et $E'$ sont équivalentes lorsqu'elles décrivent le même langage. On note : $E \equiv E'$

[C] p.34 Ex 7 : • $L = A^* a A^*$ est l'ensemble des mots contenant au moins un $a$.
• $(0 + a(1 + b \cdot 1))^* \equiv (ab + a)^*$

[B] p.309 [S] p.125 Déf 8 : Un langage est rationnel lorsque il est décrit par une expression rationnelle.

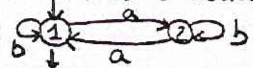
[C] p.33 Prop 9 : La classe $R$ des langages rationnels est la plus petite classe de langage telle que : $\emptyset \in R$ ; $\forall a \in A, \{a\} \in R$ ; et $R$ est close par union, produit et étoile.

### 2) Langages reconnaissables

[C] p.35 Déf 10 : Un automate $\mathcal{A}$ sur l'alphabet $A$ est un quintuplet $(Q, A, \delta, I, F)$ où • $Q$ est un ensemble fini d'état ;
• $I \subseteq Q$ et $F \subseteq Q$ sont respectivement les états initiaux et finaux.
• $\delta : Q \times A \to \mathcal{P}(Q)$ est la fonction transition.
On note $q \xrightarrow{a} q'$ lorsque $q' \in \delta(q, a)$

[C] p.35 Rq : On représente les automates par des graphes, avec des flèches entrantes et sortantes pour les états initiaux et finaux.

[C] p.35 Ex 11 :



[C] p.35 Déf 12 : Un mot $u = a_1 \dots a_n$ est reconnu par $\mathcal{A} = (Q, A, \delta, I, F)$ lorsque $\exists q_0, \dots, q_n \in Q$ ; $q_0 \in I$, $q_n \in F$ et $q_0 \xrightarrow{a_1} q_1 \to \dots \xrightarrow{a_n} q_n$. On note $\delta(q_0, u) = q_n$ ou $q_0 \xrightarrow{u} q_n$

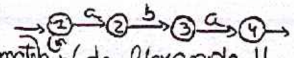
[C] p.35 Déf 13 : Le langage $L(\mathcal{A})$ reconnu par $\mathcal{A}$ est l'ensemble des mots $u$ reconnus par $\mathcal{A}$.

[C] p.35 Ex 14 : L'automate de l'exemple 11 reconnaît les mots ayant un nombre pair de $a$.

### 3) Théorème de Kleene :

[S] p.94 Prop 15 : À toute expression rationnelle, on peut associer un automate fini reconnaissant le même langage

[C] p.37 [S] p.98 ↳ La construction de l'automate se fait par exemple par la construction de Thompson (cf. annexe).

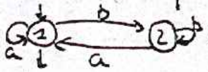
[B] p.236 Ex 16 : $*$ $L=(a+b)^*aba$ est reconnu par : 

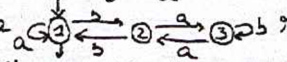
[C] $*$ $L=(b+ab^*a)^*$ est reconnu par l'automate de l'exemple 11.

p.3 **Lemme 17 : Arden** : Soient $K$ et $L$ deux langages, et l'équation

[C] p.40 $X=KX+L$. Alors $\cdot$ $\varepsilon\notin K$ : il existe une unique solution $X=K^*L$

$\cdot$ $\varepsilon\in K$ : les solutions sont de la forme $X=K^*(L+Y)$ où $Y\in A^*$

[S] p.107 Ex 18 : L'automate  reconnaît $L=(b^*a)^*$

[S] p.108 $*$ L'automate  reconnaît $(a+b(ab^*a)^*b)^*$

[C] p.36 **Théorème 19 : Kleene** : Un langage est rationnel si et seulement si il est reconnu par un automate fini.

## II – Propriétés des langages rationnels

### 1) Quotients d'un langage

[C] **Prop et déf. 20** : Tout automate fini est équivalent (ie reconnaît le même langage) à un automate fini déterministe complet, ie $|I|=1$ et $\forall a\in A, \forall q\in Q$, $|\delta(q,a)|=1$

[C] p.45 **Déf 21** : Soit $L$ un langage ; Soit $u$ un mot. Le quotient à gauche de $L$ par $u$ est le langage $u^{-1}L=\{v \mid uv\in L\}$

[C] p.45 **Prop 22** : Soit $a\in A$ ; Soient $K$ et $L$ deux langages :

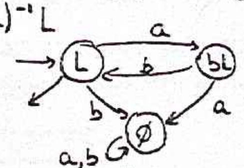
$\cdot$ $a^{-1}(K+L)=a^{-1}K+a^{-1}L$ ; $a^{-1}(KL)=(a^{-1}K)L+\varepsilon(K)a^{-1}L$ ($\varepsilon(K)=\varepsilon$ si $\varepsilon\in K$, $0$ sinon) ; $a^{-1}(L^*)=(a^{-1}L)L^*$.

[C] p.45 Ex 23 : $a^{-1}(ab^*a+b)^*=b^*a(ab^*a+b)^*$

[C] p.46 **Déf 24** : Soit $L$ un langage rationnel. L'automate minimal reconnaissant $L$ est : $\mathcal{A}_L=(Q,A,\delta,I,F)$ avec :

$Q=\{u^{-1}L \mid u\in A^*\}$, $I=\{L\}$, $F=\{u^{-1}L \mid u\in L\}$

$\delta(u^{-1}L,a)=(ua)^{-1}L$

[C] p.46 Ex 25 : $L=(ab)^*$ 

[C] p.15 **Prop 26** : Un langage est rationnel si et seulement si il admet un nombre fini de quotients à gauche. DVPT

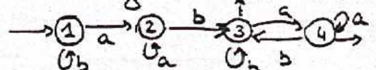
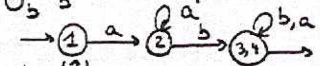
[B] p.317 **Prop 27** : L'automate minimal de $L$ est l'automate parmi les automates déterministes complets ayant le moins d'états qui reconnaît $L$.

L'automate minimal peut être construit à partir d'algorithmes qui s'appuient sur :

**Déf. 28** : Congruence de Nérode : Soit $\mathcal{A}=(Q,A,\delta,I,F)$ un automate déterministe complet. $\forall q,q'\in Q$. [C] p.47

$q\sim_N q' \iff \forall w\in A^*, (q\cdot w\in F \iff q'\cdot w\in F)$.

**Prop 29** : Soit $\mathcal{A}$ un automate déterministe complet. L'automate minimal équivalent est isomorphe à $\mathcal{A}/\sim_N$ où $\sim_N$ est la congruence de Nérode de $\mathcal{A}$. [C] p.48

Ex 30 :  équivaut à l'automate minimal suivant :  [S] p.125

Rq : L'automate minimal est unique à isomorphisme près

### 2) Lemme de l'étoile

**Thm 31** : Lemme de l'étoile : Soit $L$ un langage rationnel.

1) $\exists N\in\mathbb{N}, \forall u\in L, |u|\geq N, \exists\, u=vtw$, $\cancel{v\neq\varepsilon}$ $t\neq\varepsilon$, $\forall n\in\mathbb{N}, vt^nw\in L$ [S]

2) $\exists N\in\mathbb{N}, \forall u\in L, \forall u=v_1u_1w_1 \,/\, |u_1|\geq N : \exists\, u_1=vtw$, $\cancel{v\neq\varepsilon}$ $t\neq\varepsilon$ p.78 tq : $\forall n\in\mathbb{N}$, $v_1vt^nww_1\in L$

3) $\exists N\in\mathbb{N}, \forall u\in L, u=vu_1\ldots u_Nw$, les $u_i\neq\varepsilon$, $\exists\, 0\leq j<k\leq N$ tq : $vu_1\ldots u_j(u_{j+1}\ldots u_k)^*u_{k+1}\ldots u_Nw\in L$

Rq : Ce résultat sert à montrer que des langages ne sont pas rationnels. [S] p.79

Ex 32 : $\cdot$ $L=\{a^nb^n \mid n\in\mathbb{N}\}$ n'est pas rationnel [S] p.79

$\cdot$ $L=\{a^nb^m \mid n\in\mathbb{N}\}$ satisfait 1) mais pas 2) donc n'est pas rationnel [S] p.80

$\cdot$ $L=\{a^{k_1}ba^{k_2}b\ldots a^{k_n}b \mid \exists i, i\neq k_i\}$ satisfait 1), 2) et 3) mais n'est pas rationnel ! [S] p.80

La réciproque est fausse !

**Rq 33** : Une réciproque au lemme de l'étoile [S] p.127

Soit $L$ un langage.

$L$ et $A^*\setminus L$ vérifient le lemme de l'étoile par blocs (3) si et seulement si $L$ et $A^*\setminus L$ sont rationnels.

## III - Applications.

[S] Les langages rationnels ont de nombreuses applications, notamment liées à l'analyse lexicale des langages de programmation et à la recherche de motifs.

### 1) Recherche de motif

[B] On considère un texte $t = t_1 \dots t_k$ ; on cherche une occurrence du motif $x = x_1 \dots x_n$ dans $t$.

[B] a - Algorithme naïf :

On compare le motif à chaque facteur de $t$ de longueur $n$ ; on s'arrête si on trouve une occurence de $x$.

Complexité en pire cas : $O(nk)$ lorsque $n$ petit devant $k$.

[B] b - Algorithme de Knuth-Morris et Pratt et automate de Simon.

On optimise le décalage dans la comparaison des motifs en considérant le plus grand suffixe lu qui est aussi préfixe du motif $x$.

On peut également construire l'automate des occurrences et l'utiliser dans la recherche du motif

Complexité en pire cas : $O(n+k)$.

DVPT

Rq : Cette recherche de motif est utilisée pour rechercher les différents lexèmes dans le cadre de l'analyse lexicale.

### [FB] 2) Séparation par automates.

Soient $L$ et $K$ deux langages finis. Soit $k \in \mathbb{N}$.

Existe-t-il un automate déterministe à $k$ états tel que $L \subseteq \mathcal{L}(A)$ et $T \cap L(A) = \emptyset$ ?

Thm 34 : Ce problème est NP-complet.


Références

[C] : Carton : Langages formels

[S] : Sakarovitch : Éléments de théorie des automates

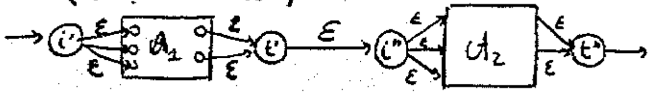
[B] : Beauquier - Berstel - Chrétienne : Éléments d'algorithmique

[FB] : Floyd-Beigel : Language of machines

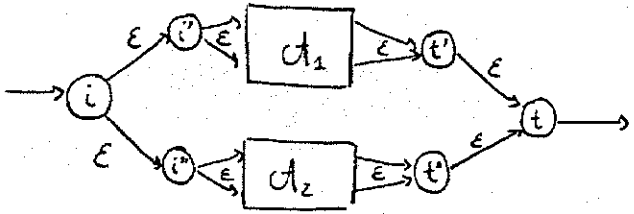
## Annexe 1 : Construction de Thompson :

· $\to \bigcirc \xrightarrow{a} \bigcirc \to$ (cas de base).

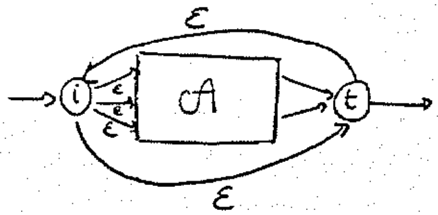
· produit :




· union :




· étoile :



# Recherche de motif.

Entrée : Un texte $t = t_1 \ldots t_n$
Un motif $x = x_1 \ldots x_m$.
Sortie : Les positions auxquelles apparaît le motif $x$ dans $t$.
(S'il n'apparaît pas, rien)

## I Algorithme de Simon :

On construit l'automate du motif.
Lorsqu'on se trouve à l'état $q$, c'est qu'on a lu les $q$ premières lettres du motif.

Ex : $x = abaa$

On doit construire l'automate en pré-traitement :

Déf : Soit $\sigma$ la fonction suffixe définie par :

$\sigma : A^* \longrightarrow [\![0, m]\!]$

$u \longmapsto \sigma(u) = \max\{k : x_1 \ldots x_k \sqsupset u\}$ $\quad$ ($x = x_1 \ldots x_m$ est le motif)

($\sigma$ correspond à la longueur du plus long suffixe de $u$ qui est préfixe de $x$.)

**Rq** : $\varepsilon$ est suffixe de tout mot donc $\sigma$ est bien défini
($\{k : x_1 \ldots x_k \sqsupset u\}$ est non-vide avec la convention :
$x_1 \ldots x_k = \varepsilon$ lorsque $k = 0$)

On définit l'automate du motif $\mathcal{A} = (Q, A, \delta, I, F)$ par
- $Q = [\![0, m]\!]$, $I = \{0\}$, $F = \{m\}$
- $\forall q \in Q, \forall a \in A, \delta(q, a) = \sigma(x_1 \ldots x_q a)$
(ie plus grand préfixe de $x$ qui soit suffixe de $x_1 \ldots x_q a$)

**Rq** : Cet automate est déterministe complet.

Algorithme :

Simon $(t, x)$ :
$q \leftarrow 0$
pour $i = 1$ à $n$ :
$q \leftarrow \delta(q, t_i)$
si $q = m$ alors imprimer "position $i - m$".

L'algorithme termine (facile) et est correct (ça se montre)

On peut montrer que sa complexité est en $O(n + m^3 |A|)$

II. L'algorithme de Knuth-Morris et Pratt :

On veut améliorer l'algorithme précédant avec un prétraitement moins coûteux.

Pour cela, on va calculer une fonction préfixe $\pi$ (qui dépendra du motif), qui exprimera les correspondances entre le motif et ses propres décalages.

Préfixe $(x_1 \ldots x_m)$ :
$\pi(1) \leftarrow 0$
$k \leftarrow 0$
pour $q = 2$ à $m$ :
tant que $k > 0$ et $x_{k+1} \neq x_q$
$k \leftarrow \pi(k)$
si $x_{k+1} = x_q$
$k \leftarrow k+1$
$\pi(q) \leftarrow k$
retourner $\pi$.

KMP $(x_1 \ldots x_m, t_1 \ldots t_n)$ :
$\pi \leftarrow$ Préfixe $(x_1 \ldots x_m)$
$q \leftarrow 0$
pour $i = 1$ à $n$ :
tant que $q > 0$ et $x_{q+1} \neq t_i$
$q \leftarrow \pi(q)$
si $x_{q+1} = t_i$
$q \leftarrow q+1$
si $q = m$
imprimer "position $i - m$"

Cet algorithme termine (ok) et est valide (long).

Complexité :

On compte le nombre de comparaisons de lettres.

• Dans KMP :

On compte le nombre de tests positifs et négatifs.

• Si le test est positif : On incrémente $i$

Or $1 \leq i \leq n$ donc $\rightsquigarrow$ au plus $n$ tests

• Si le test est négatif, $i$ inchangé mais $q$ diminue strict$^t$

donc $i-q$ augmente strictement à chaque test négatif

(+ reste inchangé à chaque test positif)

donc ($i-q=0$ au début)

$\#(\text{tests} < 0) \leq i-q$ (à la fin)

$\leq i \leq n$

Donc la complexité est en $O(n)$.

• Dans Préfixe :

idem : • au plus $m$ tests positifs

• $\#(\text{test} < 0) \leq q-k$ $\left(\begin{array}{l}\nearrow \text{strict quand} < 0 \\ \nearrow \text{quand} > 0\end{array}\right)$

$\leq m$

la complexité est en $O(m)$.

$\boxed{\text{Au final, l'algorithme est en } O(n+m).}$

## Automate quotient = automate minimal

**Prop** L'automate minimal $\mathcal{A}_L$ de $L$ est l'automate ayant le moins d'états parmi les automates déterministes complets.

dém 1) $\mathcal{A}_L = (Q, A, \delta, I, F)$ est déterministe et complet.

$I = \{L\}$ et $\forall u \in A^*$, $\forall a \in A$ $|\delta(u,a)| = |\{(ua)^{-1}L\}| = 1$

2) $\mathcal{A}_L$ possède un nombre d'états minimal.

On commence par démontrer le lemme suivant :

Lemme : Soit $L$ un langage rationnel et $\mathcal{A} = (Q', A, \delta', i, F')$ un automate déterministe complet accessible reconnaissant $L$.

Pour $q \in Q'$, on pose $L_q(\mathcal{A}) = \{w \in A^* / \delta'(q, w) \in F'\}$.

Alors $Q = \{L_q(\mathcal{A}), q \in Q'\}$.

Déf : Un état $q \in Q'$ est dit accessible si $\exists u \in A^*$, $\delta'(i,u) = q$.

$\mathcal{A}$ est accessible si tout état de $\mathcal{A}$ est accessible.

dém du lemme a) Mq $Q \subseteq \{L_q(\mathcal{A}), q \in Q'\}$.

Soit $u \in A^*$, comme $\mathcal{A}$ est complet $\exists q \in Q'$, $\delta'(i,u) = q$

Donc $w \in u^{-1}L \Leftrightarrow \underset{L(\mathcal{A}) = L}{uw \in L} \Leftrightarrow \underset{\delta'(i,u) = q}{\delta'(i, uw) \in F'} \Leftrightarrow \delta'(q,w) \in F' \Leftrightarrow w \in L_q(\mathcal{A})$

donc $\forall u$, $L_q(\mathcal{A}) = u^{-1}L \in Q$ donc $Q = \{u^{-1}L, u \in A^*\} \subseteq \{L_q(\mathcal{A}), q \in Q'\}$.

b) Mq $\{L_q(\mathcal{A}), q \in Q'\} \subseteq Q$

Soit $q \in Q'$, comme $\mathcal{A}$ est accessible $\exists u \in A^*$, $\delta'(i,u) = q$

on a comme précédemment $u^{-1}L = L_q(\mathcal{A})$ donc $\{L_q(\mathcal{A}), q \in Q'\} \subseteq Q$.

D'où $Q = \{L_q(\mathcal{A}), q \in Q'\}$ □

Retour à la preuve de la propriété :

D'après le lemme, tout automate déterministe complet accessible reconnaissant $L$ vérifie que $Q = \{L_q(\mathcal{A}), q \in Q'\}$ ie $|Q| \leq |Q'|$.

De plus comme tout automate déterministe complet ~~peut être rendu accessible en~~ est équivalent à un automate déterministe complet accessible ayant moins d'états, on a bien que $\mathcal{A}_L$

est l'automate déterministe complet reconnaissant $L$ ayant le minimum d'états.
(car les états non-accessibles ne font partie d'aucun chemin accepté par l'automate). $\square$.

Prop | Un langage est rationnel ssi il admet un nombre fini de quotients à gauche

dém $\Rightarrow$) Soit $L$ un langage rationnel, alors il existe un automate fini reconnaissant $L$ noté $\mathcal{A}$. $\mathcal{A}$ est équivalent à un automate fini déterministe complet $\mathcal{A}'$.

~~Quitte à réduire le nombre d'états de $\mathcal{A}'$, on peut supposer qu'il est accessible. Or alors d'après le lemme utilisé dans la preuve~~

Comme $\mathcal{A}_L$ est l'automate minimal de $L$, il a le nombre minimal d'états parmi les automates déterministes complets. Donc $|Q| = |\{u^{-1}L, u \in A^*\}| \leq |\{\text{états de } \mathcal{A}'\}| < \infty$

Donc $L$ admet un nombre fini de quotients à gauche.

$\Leftarrow$) Soit $L$ un langage ayant un nombre fini de quotients à gauche alors l'automate $\mathcal{A}_L$ est un automate fini reconnaissant $L$ donc $L$ est rationnel. $\square$

---

Vérification que $L(\mathcal{A}_L) = L$.

On montre par récurrence sur la longueur du mot $u \in A^*$ que pour tout état $X \in Q$, $\delta(X, u) = u^{-1}X$.

- $n=1$ : $u = a \in A$ Soit $X \in Q$ alors $\exists v \in A^*$, $X = v^{-1}L$

$$\delta(X,a) = \delta(v^{-1}L, a) = (va)^{-1}L = \{w \in A^*, vaw \in L\} = \{w \in A^*, aw \in v^{-1}L\} = \{w \in A^*, aw \in X\} = a^{-1}X$$

- on suppose le résultat vrai pour $k \leq n$ et on considère $u \in A^*$, $|u| = n+1$

on a donc $u = au'$ avec $a \in A$ et $u' \in A^*$ tel que $|u'| = n$

Soit $X \in Q$ $\delta(X,u) = \delta(X, au') = \delta(\delta(X,a), u') \underset{HR:1}{=} \delta(a^{-1}X, u') \underset{HR:n}{=} u'^{-1}(a^{-1}X)$

or $u'^{-1}(a^{-1}X) = \{w \in A^*, u'w \in a^{-1}X\} = \{w \in A^*, au'w \in X\} = \{w \in A^*, uw \in X\} = u^{-1}X$

D'où $\forall u \in A^*, \forall X \in Q, \delta(X,u) = u^{-1}X$.

- $w \in L(\mathcal{A}_L) \underset{I=\{L\}}{\Leftrightarrow} \delta(L, w) \in F \underset{\text{Récurrence}}{\Leftrightarrow} w^{-1}L \in F \underset{\text{déf. de } F}{\Leftrightarrow} w \in L$.

Ainsi $L(\mathcal{A}_L) = L$.