

LF – Théorie des langages formels

Sylvain Brandel

2022 – 2023

sylvain.brandel@univ-lyon1.fr



CM 8

LANGAGES RATIONNELS

RATIONALITÉ

Langages rationnels

- Langage : **ensemble** de mots sur Σ
 - Éléments de base
 - L'ensemble \emptyset
 - Le mot vide ε
 - Les singletons sur Σ
 - Opérations
 - La concaténation de langages
 - La réunion de deux langages
 - L'intersection de deux langages
 - La fermeture de Kleene
- **Langages rationnels**
 - Représentation **finie** :
 - Éléments de base,
 - Concaténation,
 - Union,
 - Fermeture de Kleene

Langages rationnels

- Expressions régulières sur Σ : *plus petit* ensemble E tel que
 - $\emptyset \in E$
 - $\varepsilon \in E$
 - Si $\sigma \in \Sigma$, alors $\sigma \in E$
 - Si $e_1, e_2 \in E$, alors $e_1 + e_2 \in E$ et $e_1 \cdot e_2 \in E$
 - Si $e \in E$, alors $(e) \in E$, $e^* \in E$ et $e^+ \in E$
- Priorité : $*/+ > \cdot > +$
- Exemple
 - $\Sigma = \{a, b\}$ $(a + a \cdot b)^* \cdot a^+ + \varepsilon$

Langages rationnels

- Langage représenté :

- $\llbracket \emptyset \rrbracket = \emptyset$

- $\llbracket \sigma \rrbracket = \{\sigma\}$ pour $\sigma \in \Sigma$

- $\llbracket \varepsilon \rrbracket = \{\varepsilon\}$

- $\llbracket e_1 + e_2 \rrbracket = \llbracket e_1 \rrbracket \cup \llbracket e_2 \rrbracket$

- $\llbracket e_1 \cdot e_2 \rrbracket = \{ w_1 w_2 \mid w_1 \in \llbracket e_1 \rrbracket \text{ et } w_2 \in \llbracket e_2 \rrbracket \}$

- $\llbracket e^* \rrbracket = \bigcup_{n=0}^{\infty} \llbracket e^n \rrbracket$ où $e^0 = \{\varepsilon\}, e^1 = e, e^{n+1} = e \cdot e^n$

- $\llbracket e^+ \rrbracket = \bigcup_{n=1}^{\infty} \llbracket e^n \rrbracket$ où $e^1 = e, e^{n+1} = e \cdot e^n$

Langages rationnels

- Exemples

– $\Sigma = \{a, b, c\}$ $(a \cdot a^* \cdot c + (b + c))^* \cdot a^*$ Mots ne contenant pas ab

– $\Sigma = \{0, 1\}$ $0 + 1 \cdot (0 + 1)^*$ Entiers en binaire

– $\Sigma = \{0, 1\}$ $0 + 1 \cdot (0 + 1)^* \cdot 0$ Entiers en binaire pairs

– $\Sigma = \{0, 1\}$ $0^* + (((0^* \cdot (1 + (1 \cdot 1)))) \cdot ((0 \cdot 0^*) \cdot (1 + (1 \cdot 1))))^* \cdot 0^*$

Mots ne contenant pas 111

Langages rationnels

- Système d'équations linéaires gauche :

$$\begin{cases} X_1 = e_1^1 X_1 + \dots + e_n^1 X_n + f^1 \\ \vdots \\ X_n = e_1^n X_1 + \dots + e_n^n X_n + f^n \end{cases}$$

avec e_i^j expressions régulières $\in \Sigma \cup \{\varepsilon\} \cup \emptyset$, f^i quelconque (langage)

- Solution

$$\begin{aligned} X_1, \dots, X_n \text{ si } X_1 &= \llbracket e_1^1 \rrbracket . X_1 \cup \dots \cup \llbracket e_n^1 \rrbracket . X_n \cup \llbracket f^1 \rrbracket \\ &\vdots \\ X_n &= \llbracket e_1^n \rrbracket . X_1 \cup \dots \cup \llbracket e_n^n \rrbracket . X_n \cup \llbracket f^n \rrbracket \end{aligned}$$

Langages rationnels

- Exemple

$$\begin{cases} X_1 & = & aX_2 + bX_3 + \varepsilon \\ X_2 & = & aX_1 + bX_4 \\ X_3 & = & bX_1 + aX_4 \\ X_4 & = & bX_2 + aX_3 \end{cases}$$

$X_1 = \{ \text{mots ayant un nombre pair de } a \text{ et pair de } b \}$

$X_2 = \{ \text{mots ayant un nombre impair de } a \text{ et pair de } b \}$

$X_3 = \{ \text{mots ayant un nombre pair de } a \text{ et impair de } b \}$

$X_4 = \{ \text{mots ayant un nombre impair de } a \text{ et impair de } b \}$

Langages rationnels

- Soit L un langage.

L **rationnel** si $\exists S$ tq (L, L_1, L_2, \dots) est solution **minimale** de S

- Lemme

$X = eX + f$ avec e, f : expressions régulières

$e^*.f$ est solution **minimale** de $X = eX + f$ si $\varepsilon \in e$

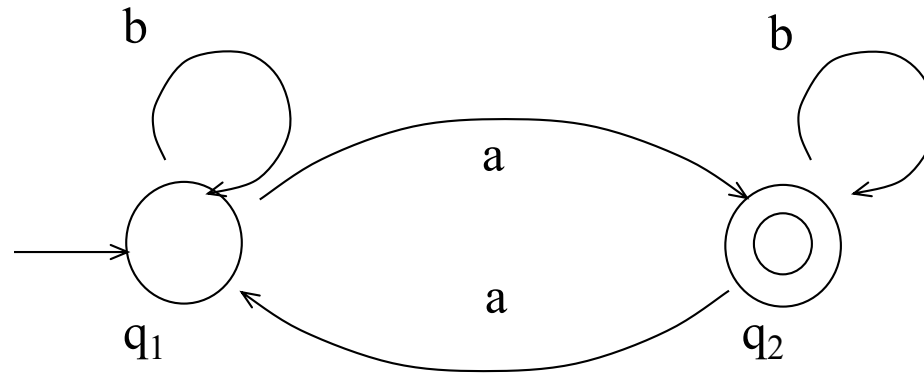
est solution **unique** de $X = eX + f$ si $\varepsilon \notin e$

Preuve

- $e^*.f$ solution
- $e^*.f$ solution minimale
- $e^*.f$ solution unique si $\varepsilon \notin e$

Langages rationnels

- Exemple



$$\begin{cases} X_1 = & bX_1 + aX_2 \\ X_2 = & bX_2 + aX_1 + \varepsilon \end{cases}$$

$$X_1 = b^*a(b + ab^*a)^*$$

$$X_2 = (b + ab^*a)^*$$

Langages rationnels

- Exemple

$$\begin{cases} X_1 = & 1X_1 + 1X_2 + \varepsilon \\ X_2 = & 0X_1 \\ X_3 = & 0X_2 + 0X_3 + 1X_3 \end{cases}$$

$$X_1 = (1 + 10)^*$$

$$X_2 = 0(1 + 10)^*$$

$$X_3 = (0 + 1)^* 00 (1 + 10)^*$$

Rationalité

- Montrer qu'un langage est **rationnel**

(1) **Stabilité** (rappel : la classe des langages acceptés par un automate est stable par union, concaténation, fermeture itérative, complément, intersection)

(2) **Caractérisation** (rappel : un langage est rationnel ssi il est accepté par un automate)

(3) Un langage est rationnel ssi il peut être décrit par une **expression régulière**.

(4) Un langage L est rationnel ssi \approx_L a un nombre fini de classes d'équivalence.

(2) et (3) \rightarrow équivalence entre automate et ER

\rightarrow il existe des algorithmes

automate \rightarrow ER

ER \rightarrow automate

Rationalité

- Montrer qu'un langage est **rationnel**
 - À partir de (1) : utiliser les propriétés de stabilité
 - décomposer le langage en sous ensembles par union, intersection, concaténation, et montrer que ces sous ensembles sont rationnels.
 - À partir de (2) : construire un automate acceptant ce langage (on peut éventuellement déterminer / minimiser cet automate)
 - À partir de (3) : construire une expression régulière décrivant ce langage
 - À partir de (4) : déterminer les classes d'équivalence de la relation \approx_L et montrer que leur nombre est fini

Non rationalité

- Il existe des langages non rationnels
 - L'ensemble des expressions régulières est dénombrable
 - L'ensemble des langages est non dénombrable
 - Tout langage **fini** est rationnel (il peut être décrit par une ER composée de l'union de tous les mots du langage)
- La question de **non rationalité** ne se pose que pour les langages **infinis**
- Montrer la **non rationalité**
 - Stabilité et raisonnement par l'absurde
 - Lemme de l'étoile

Non rationalité

Propriétés de stabilité

- Pour montrer que L est **non rationnel** :

on pose **l'hypothèse** que L est **rationnel**

et on détermine L_0 **non rationnel** et L_1 **rationnel** tels que

$$L_0 = L \theta L_1 \quad (\theta \in \{\cap, \cup, \cdot\})$$

- L supposé rationnel
- L_1 rationnel $L \theta L_1$ rationnel (stabilité de la classe des langages rationnels par θ)
- Or $L \theta L_1 = L_0$ avec L_0 connu (démontré) non rationnel
 - ➔ Contradiction
 - ➔ l'hypothèse (L rationnel) est fausse

Non rationalité

Propriétés de stabilité

- Exemple : $L = \{ w \in \{a,b\}^* \mid w \text{ contient autant de } a \text{ que de } b \}$
Montrons que L est non rationnel

Supposons que L est rationnel.

Soit $L_1 = a^*b^*$.

L_1 rationnel (parce que décrit par une expression rationnelle).

Donc $L \cap L_1$ rationnel (par stabilité de la classe des langages rationnels par \cap).

Or $L \cap L_1 = L_0 = \{a^n b^n \mid n \geq 0\}$.

L_0 non rationnel (démontré plus loin).

Donc contradiction.

Donc l'hypothèse (L rationnel) est fausse.

Donc L non rationnel.

Non rationalité

Lemme de l'étoile

- Théorème Lemme de l'étoile

Soit L un langage *rationnel* infini accepté par un automate *déterministe* M à k états.

Soit z un mot quelconque de L tel que $|z| \geq k$.

Alors z peut être décomposé en uvw

avec $|uv| \leq k$, $|v| \neq 0$ et $uv^i w \in L, \forall i \geq 0$.

Non rationalité

Lemme de l'étoile

- Exemple : Montrons que $L = \{a^n b^n \mid n \geq 0\}$ est non rationnel.

Supposons que L est rationnel.

L est reconnu par un automate M à k états.

D'après le lemme de l'étoile, $\forall z \in L, |z| \geq k, \exists u, v, w \in \Sigma^*$ tels que
 $z = uvw, |uv| \leq k, |v| > 0$ et $\forall i \geq 0, uv^i w \in L$

Soit $z_0 = a^k b^k$.

On a bien $z_0 \in L$ et $|z_0| = 2k \geq k$.

Toutes les décompositions possibles $z_0 = uvw$ telles que $|uv| \leq k, |v| > 0$
sont de la forme $u = a^p, v = a^q, w = a^r b^k$ avec $q > 0$ et $p+q+r = k$.

Or $uv^i w = a^p a^{qi} a^r b^k = a^{p+qi+r} b^k$

On a $\forall i \neq 1, p+qi+r \neq k$

Donc $\forall i \neq 1, uv^i w \notin L$

Donc contradiction dans la propriété

Donc l'hypothèse (L rationnel) est fausse

Donc L non rationnel

Complexité d'algorithmes pour les automates

- Théorèmes

- (i) Il existe un algorithme *exponentiel* (en le nombre d'états) *déterminisation*
Entrée : un automate fini non déterministe
Sortie : un automate fini déterministe équivalent
- (ii) Il existe un algorithme *polynomial* (en fonction de la taille de l'expression ou du nombre d'opérateurs) *ER → automate*
Entrée : une expression régulière
Sortie : un automate non déterministe équivalent
- (iii) Il existe un algorithme *exponentiel* (en fonction du nombre d'états) *automate → ER*
Entrée : un automate non déterministe
Sortie : une expression régulière équivalente
(le nombre des $R(i, j, k)$ est multipliée par 4 à chaque incrément de k)
$$R(i, j, k) = R(i, j, k-1) \cup R(i, k, k-1) R(k, k, k-1)^* R(k, j, k-1)$$

Complexité d'algorithmes pour les automates

(iv) Il existe un algorithme *polynomial* (en fonction du nombre d'états)

minimisation

Entrée : un automate déterministe

Sortie : l'automate déterministe minimal (standard) équivalent

(v) Il existe un algorithme *polynomial* pour décider si deux automates déterministes sont équivalents (passe par l'automate standard)

équivalence

Entrée : deux automates déterministes

Sortie : vrai s'ils sont équivalents, faux sinon

(vi) Il existe un algorithme *exponentiel* pour déterminer si deux automates non déterministes son équivalents

équivalence

Entrée : deux automates non déterministes

Sortie : vrai s'ils sont équivalents, faux sinon

Complexité d'algorithmes pour les automates

- (vii) Il existe un algorithme *polynomial* qui teste si $w \in L$ *appartenance*
avec une complexité en temps de $O(|w|)$
Entrée : un langage rationnel L et un mot $w \in \Sigma^*$
Sortie : vrai si $w \in L$, faux sinon
- (viii) Il existe un algorithme *polynomial* qui teste si $w \in L$ *appartenance*
avec une complexité en temps de $O(|K|^2|w|)$
Entrée : un automate M non déterministe et un mot $w \in \Sigma^*$
Sortie : vrai si $w \in L(M)$, faux sinon