LF – Théorie des langages formels

Sylvain Brandel

2023 – 2024

sylvain.brandel@univ-lyon1.fr

CM 8

LANGAGES RATIONNELS RATIONALITÉ

Langages rationnels

- Langage : ensemble de mots sur Σ
  - Éléments de base
    - L'ensemble $\varnothing$
    - 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 $e_1 . e_2 \in E$
  - Si $\mathrm{e} \in E$, alors $(\mathrm{e}) \in E$, $e^* \in E$ et $e^+ \in E$

- Priorité : $^{*/+}$ > . > +

- Exemple
  - Σ = {a, b} (a + a . b)* . a* + ε

Langages rationnels

- Langage représenté :
  - $[\![\emptyset]\!] = \emptyset$
  - $[\![\sigma]\!] = \{\sigma\}$ pour $\sigma \in \Sigma$
  - $[\![\varepsilon]\!] = \{\varepsilon\}$
  - $[\![e_1 + e_2]\!] = [\![e_1]\!] \cup [\![e_2]\!]$
  - $[\![e_1 \,.\, e_2]\!] = \{\, w_1 w_2 \mid w_1 \in [\![e_1]\!]$ et $w_2 \in [\![e_2]\!] \,\}$
  - $[\![e^*]\!] = \bigcup_{n=0}^{\infty} [\![e^n]\!]$ où $e^0 = \{\varepsilon\}, e^1 = e, e^{n+1} = e \,.\, e^n$
  - $[\![e^+]\!] = \bigcup_{n=1}^{\infty} [\![e^n]\!]$ où $e^1 = e, e^{n+1} = e \,.\, e^n$

Langages rationnels

- Exemples
  - $\Sigma = \{a, b, c\}$ $(a \,.\, a^* \,.\, c + (b + c))^* \,.\, a^*$ Mots ne contenant pas ab
  - $\Sigma = \{0, 1\}$ $0 + 1 \,.\, (0 + 1)^*$ Entiers en binaire
  - $\Sigma = \{0, 1\}$ $0 + 1 \,.\, (0 + 1)^* \,.\, 0$ Entiers en binaire pairs
  - $\Sigma = \{0, 1\}$ $0^* + (((0^* \,.\, (1 + (1.1))) \,.\, ((0 \,.\, 0^*) \,.\, (1 + (1 \,.\, 1)))^*) \,.\, 0^*)$

    Mots ne contenant pas 111

Langages rationnels

- Système d'équations linéaires gauche :

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

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

- Solution

$$X_1, \cdots, X_n \; si \; X_1 = [\![e_1^1]\!] \,.\, X_1 \cup \cdots \cup [\![e_n^1]\!] \,.\, X_n \cup [\![f^1]\!]$$

$$\vdots$$

$$X_n = [\![e_1^n]\!] \,.\, X_1 \cup \cdots \cup [\![e_n^n]\!] \,.\, X_n \cup [\![f^n]\!]$$

Langages rationnels

- Exemple

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

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

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

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

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

Langages rationnels

- Soit L un langage.

  L rationnel si ∃ S tq (**L**, L_1, L_2, …) est solution minimale de S

- Lemme

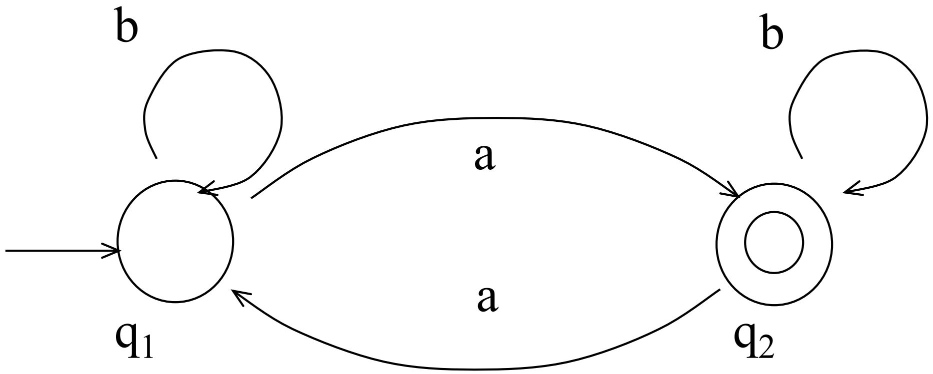
$X = eX + f$	avec e, f : expressions régulières	
$e^*.f$	est solution minimale de $X = eX + f$	si $\varepsilon \in$ e
	est solution unique de $X = eX + f$	si $\varepsilon \notin$ e

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) → équivalence entre automate et ER

→ il existe des algorithmes

automate → ER

ER → 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 déterminiser / 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

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

- L supposé rationnel
- L_1 rationnel $\quad$ $L \ \theta \ L_1$ rationnel $\quad$ (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 : $L = \{ w \in \{a,b\}^* \mid w$ contient autant de a 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 $uv^iw \in L$, $\forall i \geq 0$.*

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

$$z = uvw,\ |uv| \leq k,\ |v| > 0 \text{ et } \forall\, i \geq 0,\ uv^iw \in L$$

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 $uv^iw = a^p a^{qi} a^r b^k = a^{p+qi+r} b^k$
On a $\forall\, i \neq 1,\ p + qi + r \neq k$
Donc $\forall\, i \neq 1,\ uv^iw \notin L$
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