LifLF – Théorie des langages formels

Sylvain Brandel
2016 – 2017
sylvain.brandel@univ-lyon1.fr

CM 6

RATIONALITÉ
NON RATIONALITÉ

Rationalité

- On peut monter qu'un langage est rationnel avec un des théorèmes :
  - (1) Stabilité

    (rappel : la classe des langages acceptés par un automate est stable par union, concaténation, Kleene star, 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 rationnelle.

(2) et (3) → équivalence entre automate et ER

→ il existe des algorithmes

automate → ER

ER → automate

Rationalité

- Pour montrer qu'un langage est rationnel :
  - A 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.

  - A partir de (2) : construire un automate acceptant ce langage
    (on peut éventuellement déterminiser / minimiser cet automate)

  - A partir de (3) : construire une expression rationnelle décrivant ce langage.

Non rationalité

- Il existe des langages non rationnels :
  - L'ensemble des expressions rationnelles est dénombrable
  - L'ensemble des langages est non dénombrable

- Tout langage <u>fini</u> 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 <u>infinis</u>.

- 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,

  <u>on pose l'hypothèse que L est rationnel</u>,

  et on détermine L_0 non rationnel et L_1 rationnel tels que

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

- L supposé rationnel
- L_1 rationnel

  $L \ \theta \ L_1$ rationnel par 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
Lemme de l'étoile

- <u>Théorème</u> Lemme de l'étoile

  Soit L un langage rationnel infini accepté par un automate <u>déterministe</u> 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

- Preuve
  - Lemme 1

    Soit G le graphe d'un automate déterministe à k états.

    Tout chemin de longueur k dans G contient un cycle.
  - Lemme 2

    Soit G le graphe d'un automate déterministe à k états.

    Soit p un chemin de longueur k ou plus.

    p peut être décomposé en sous chemins q, r et s tels que p = qrs, |qr| ≤ k, r est un cycle.

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

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

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 (sans démonstrations précises)

  (i) Il existe un algorithme exponentiel (en le nombre d'états)

  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)

  Entrée : une expression rationnelle

  Sortie : un automate non déterministe équivalent

Complexité d'algorithmes pour les automates

(iii) Il existe un algorithme exponentiel (en fonction du nombre d'états)

Entrée : un automate non déterministe

Sortie : une expression rationnelle équivalente

(la taille des R(i, j, k) est multipliée par 3 à chaque incrément de k)

$R(i, j, k) = R(i, j, k\text{-}1) \cup R(i, k, k\text{-}1)\ R(k, k, k\text{-}1)^*\ R(k, j, k\text{-}1)$

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

Entrée : un automate déterministe

Sortie : l'automate déterministe minimal (standard) équivalent

Complexité d'algorithmes pour les automates

(v) il existe un algorithme <u>polynomial</u> pour décider si deux automates déterministes sont équivalents

(Passe par l'automate standard)

(vi) Il existe un algorithme <u>exponentiel</u> pour déterminer si deux automates <u>non</u> déterministes son équivalents

Complexité d'algorithmes pour les automates

- <u>Théorème</u>

  L = langage rationnel (donné par un automate ou une expression rationnelle) et $w \in \Sigma^*$

  Il existe un algorithme qui teste si $w \in L$ *avec une complexité en temps de* $O(|w|)$

- <u>Théorème</u>

  $M = (K, \Sigma, \Delta, s, F)$ *non déterministe et* $w \in \Sigma^*$

  Il existe un algorithme qui teste si $w \in L(M)$ *avec une complexité en temps de* $O(|K|^2\,|w|)$