Le problème d'universalité d'un langage rationnel est PSPACE-Complet

Le développement

Lemme 1

Le problème d'universalité d'un langage rationnel est ***P****SPACE-dur*

Soit un problème $f : \Gamma^* \to \{\top, \bot\}$ de la classe **P**SPACE et même SPACE(p) pour un certain polynôme p. On veut montrer que si l'on dispose d'une réduction de ce problème au problème d'universalité ie $h : \Gamma^* \to \mathrm{Rat}(\Sigma^*)$ tel que $h(x) = \Sigma^*$ **ssi** $f(x) = \bot$.

Et soit $\mathcal{M} = (Q, T, \Gamma \cup \{\_\}, \delta, \_, q_0, q_f)$ une machine de turing déterministe décidant f avec une complexité en espace $p(|x|)$ pour $x \in \Gamma^*$. Avec :

- Q l'ensemble des états ;.
- q_0 état initial.
- q_f état final.
- T alphabet de bande.
- Γ alphabet d'entrée. ($\Gamma \subset T$)
- δ fonction de transition.

Fixons $x = x_1 x_2 \cdots x_n \in \Gamma^*$ qui est lu par $\mathcal{M}$ en m étapes. On va donner l'expression rationnelle E d'un langage qui sera Σ^* **ssi** $f(x) = \bot$.

On note $\Sigma = T \cup \{qX \mid q \in Q \text{ et } X \in T\} \cup \{\sharp\}$ et $\Delta = \Sigma \setminus \{\sharp\}$.

Construisons le mot $w = \sharp w_1 \sharp \cdots \sharp w_m \sharp$ où chaque w_i représente la i[e] configuration de la machine $\mathcal{M}$ au cours de la lecture de x.

Comme la complexité en espace de la machine est décrite par le polynôme p, chaque mot w_i aura une longueur $p(|x|) = p(n)$ noté ϱ (ainsi on est sûr qu'ils pourront contenir toute l'information de l'état de $\mathcal{M}$).

Les conventions d'écriture pour w_i sont les suivantes $w_i = \alpha_1 \ldots \alpha_{j-1}(q\alpha_j)\alpha_{j+1} \ldots \alpha_k \_ \ldots \_$ où $(q\alpha_j) \in \Sigma$ représente le caractére donnant l'état courant dans Q et la lettre en cours de lecture. Et on complète par des symboles blancs ($\_$) pour faire un mot de longueur ϱ.

Un mot w **n'**est **pas** une exécution acceptante de $\mathcal{M}$ partant de x si l'un des cas suivant se produit :

1. Les $\sharp$ n'apparaissent pas aux bons endroit et/ou les w_i ne sont pas de la bonne taille.
2. Les w_i contiennent zéro ou plusieurs symboles indiquant la position de la tête de lecture (de la forme qX)
3. w ne contient pas l'état final $q_F X$ pour tout $X \in T$. (ie l'execution n'accepte pas)
4. w ne contient pas la configuration initiale $\sharp w_0$, ou celle-ci n'est pas de la bonne forme.
5. Enfin la configuration w_{i+1} ne se déduit pas de w_i conformément à δ.

Construction des expressions

- Posons E_3 qui représente les mots w ne rencontrant pas d'état final. Posons $Y = \{q_f X \mid X \in T\} \subset \Sigma^*$. On a donc

$$E_3 = \left(\sum_{a \in \Sigma \setminus Y} a \right)^*$$

On a de plus la relation $|E_3| = 2|T| + 2$.

- Posons E_4 qui représente les mots ne contenant pas la configuration initiale.

$$E_{4.1} = \Delta\Sigma^* + \Sigma(\Sigma \setminus \{q_0x_1\})\Sigma^* + \cdots + \Sigma^{n+1}(\Sigma \setminus \{x_n\})\Sigma^*$$

  (Représente les mots ne contenant pas soit le dièse initial, soit le symbole initial (q_0x_1) soit une lettre de x à la bonne position.)
  Longueur : $|E_{4.1}| = O(n^2|\Sigma|) = O(n^2)$. En effet $|\Sigma^n| = n(2|\Sigma| + 1)$.

$$E_{4.2} = (\Sigma^{n+1} + \Sigma^{n+2} + \cdots + \Sigma^{\varrho+1})(\Sigma \setminus \{\_\})\Sigma^*$$

  (Représente les mots complétés par autre chose que des blancs)
  Longueur : $|E_{4.2}| = O((\varrho - n)n|\Sigma|) = O(p(n)n)$

$$E_{4.3} = \varepsilon + \Sigma + \cdots + \Sigma^{\varrho}$$

  (Représente les mots trop courts)
  Longueur : $|E_{4.3}| = O(p(n)^2)$
  Ainsi finalement $E_4 = E_{4.1} + E_{4.2} + E_{4.3}$ désigne exactement les mots ne contenant pas la configuration initiale $\sharp w_1$.
  Longueur : $|E_4| = O(n|\Sigma|) = O(p(n)^2)$ (on peut supposer $n \in O(p(n))$ sans perte de généralité).
- E_1 représentera les mots qui n'ont pas la bonne forme $\sharp w_1 \sharp w_2 \sharp \cdots \sharp w_n \sharp$ avec $|w_i| = \varrho$:

$$E_{1.1} = \sum_{0 \leq i < \varrho} \Sigma^* \sharp \Delta^i \sharp \Sigma^*$$

  (Mots avec des configurations trop courtes)
  Longueur : $|E_{1.1}| = O(p(n)^2)$

$$E_{1.2} = \Sigma^* \sharp \Delta^{\varrho+1} \Delta^* \sharp \Sigma^*$$

  (Mots avec des configurations trop longues)
  Longueur : $|E_{1.2}| = O(p(n))$

$$E_{1.3} = \Delta^* + \Delta^* \sharp \Delta^* + \Delta\Sigma^* + \Sigma^*\Delta$$

  (Mots sans $\sharp$ ou avec un seul $\sharp$ ou ne commençant pas par $\sharp$ ou ne terminant pas par $\sharp$)
  Longueur : $|E_{1.3}| = O(1)$
  Ainsi $E_1 = E_{1.1} + E_{1.2} + E_{1.3}$ reconnait exactement les mots n'ayant pas la bonne forme pour représenter une execution. La longueur est borné par $p(n)^2$.
- E_2 représentera les mots avec des configurations ne contenant aucun état (aucune lettre de type qX avec $q \in Q$, $X \in T$, on note QT leur ensemble) ou en contenant trop (ie au moins 2) :

$$E_2 = \Sigma^* \sharp T^* \sharp \Sigma^* + \Sigma^* \sharp T^*(QT)T^*(QT)T^* \sharp \Sigma^*$$

  Longueur : $|E_2| = O(1)$
- E_5 représentera les mots $w = \sharp w_1 \sharp w_2 \sharp \cdots \sharp w_n \sharp$ tels que l'on ne puisse pas passer d'un certain w_i à un certain w_{i+1}. C'est l'expression la plus important qui assure que les mots décris correspondent à une bon fonctionnement de $\mathcal{M}$.
  Pour cela remarquons que w_{i+1} se déduit de w_i en ne modifiant que les lettres qui sont proche du symbole de type qX (emplacement de la tête de lecture). Cette propriété locale nous permet de définir une fonction $\varphi : \Delta^3 \to \Delta$ qui décrit la fonction de transition δ.

$$\blacksquare\underset{\triangle}{\blacksquare}\blacksquare \quad \to \quad \square\underset{\triangle}{\blacksquare}\square$$

On notera $\psi : \Delta^3 \to \mathcal{P}(\Sigma)$ la fonction définie par $\psi(c) = \Delta \setminus \{\varphi(c)\}$

$$E_5 = \sum_{c \in \Delta^3} \Sigma^* c \Sigma^{\varrho-1} \left(\sum_{b \in \psi(c)} b \right) \Sigma^*$$

Etrangement la longueur de E_5 n'est asymptotiquement dépendante que de $|\Sigma^{\varrho-1}|$ et donc en $O(p(n))$.

On a alors décrit la totalité du fonctionnement de $\mathcal{M}$ c'est-à-dire que si un mot w est dans le langage généré par $E = E_1 + E_2 + E_3 + E_4 + E_5$, alors il ne correspond pas à une exécution de $\mathcal{M}$ lisant le mot x.

Donc $[\![E]\!] = \Sigma^*$ **ssi** il n'y a pas d'exécution de $\mathcal{M}$ commençant par x qui termine **ssi** $f(x) = \bot$

Et enfin $|E|$ est un polynôme en n borné par $\lambda p(n)^2$, avec λ constante positive.

Résumé On a donc construit une fonction $h : \Gamma^* \to Rat(\Sigma^*)$ donnée par $x \mapsto E$, calculable (la construction est algorithmique) et satisfaisant pour tout x :
- $h(x) = \Sigma^*$ **ssi** $f(x) = \bot$
- $|h(x)| \in O(p(|x|)^2)$

Autour

Preuve du caractère PSPACE Rq : On s'intéresse à des problèmes de décision et de complexité sur des langages, on pourra considérer le langage défini par un automate ou par une expression rationnelle vu qu'on peut passer polynômialement (en temps et en espace) de l'un à l'autre.

Etant donné un langage rationnel L, le problème du vide (est ce que $L = \emptyset$) est décidable en temps et en espace polynômial (problème d'accessibilité dans un graphe). Mais l'universalité ($L = \Sigma$) ne se ramène au problème du vide qu'après déterminisation et passage à l'automate complémentaire. Or la procédure de déterminisation est exponentielle en espace occupé.

Cependant, on peut proposer un algorithme non déterministe qui vérifie l'universalité : $L = \Sigma^*$ (cf preuve du lemme 3) en temps polynômial.

Lemme 2 (*Admis*)

On peut passer d'un automate fini à une expression rationnelle décrivant le même langage et réciproquement. Et ce en temps polynômial et en espace polynômial.

Cf. Thm de Kleene, construction de l'automate par les dérivées d'antimirov (par ex) et création de l'expression rationnelle par le lemme d'Arden.

Lemme 3

Le problème d'universalité d'un langage rationnel est ***NP****SPACE*

On le considère décrit par un automate complet $\mathcal{A}$ à n états.

On considère une machine de turing non déterministe qui va choisir une lettre dans Σ et simuler le comportement de $\mathcal{A}$ sur la lecture de cette lettre en mémorisant l'ensemble des états activés.

Chaque instance de la machine poursuit son chemin jusqu'à un ensemble d'états activés ne contenant aucun état final. On sait qu'un tel ensemble s'il existe est au plus atteint $2^n - 1$ étapes après l'état initial. Il suffit de compter le nombre d'étapes et s'arrêter par défault au bout de $2^n - 1$.

On a bien construit une machine qui décide l'universalité de $L(\mathcal{A})$: elle accepte **ssi** l'une de ses instances non déterministe trouve un ensemble d'états non finaux.

La place utilisé en mémoire est celle du compteur $O(\log_2(2^n))$ et celle des états $O(n)$ donc globalement en $O(n)$.

Lemme 4 (*Admis*)

$\boldsymbol{P}SPACE = \boldsymbol{NP}SPACE$

Corrolaire du théorème de Savitch.

Références

[1] R.W. Floyd, P. Biegel, R. Beigel, and D. Krob. *Le langage des machines : introduction à la calculabilité et aux langages formels.* Vuibert informatique, 1995.

[2] Serge Haddad. Complexité (l3). poly, janvier 2011.