Langages rationnels et automates finis. Exemples et applications.

909

Cadre : Σ est un alphabet fini

I. Langages rationnels

Def 1: Les opérations suivantes sur l'ensemble des langages sur Σ^* seront appelées opérations rationnelles:

- Union: Soient L_1, L_2 deux langages, on définit $L_1+L_2=L_1 \cup L_2$
- Produit: Soient L_1, L_2 deux langages, on définit le produit $L_1 \cdot L_2=\{w_1 w_2 \mid w_1 \in L_1, w_2 \in L_2\}$ [Car]
- Puissance: Soit L un langage, on définit L^n par induction sur $\mathbb{N}$:
  - $L^0=\{\varepsilon\}$
  - $L^{n+1}=L\cdot L^n$
- Étoile: Soit L un langage, on définit $L^*=\bigcup_{n\in\mathbb{N}} L^n$.

Ex 2: $(\{a\}+\{b\})^*$ est l'ensemble des mots sur l'alphabet $\{a,b\}$.

Def 3: On définit l'ensemble des langages rationnels $Rat(\Sigma^*)$ comme étant la plus petite famille de langages closes par opérations rationnelles telles que $\emptyset \in Rat(\Sigma^*)$ et $\forall a \in \Sigma \quad \{a\} \in Rat(\Sigma^*)$. [Car]

Ex 4:
- $\{a\}^*=\{a^k \mid k \in \mathbb{N}\}$ est rationnel.
- Σ^* est rationnel.
- Tout langage fini est rationnel.

Def 5: On définit l'ensemble des expressions rationnelles $\mathcal{E}$ par induction: [Car]
- $\emptyset \in \mathcal{E}$; $a \in \mathcal{E}$ pour tout $a \in \Sigma$
- Si $E_1, E_2 \in \mathcal{E}$, alors E_1+E_2, $E_1 \cdot E_2$, E_1^* sont dans $\mathcal{E}$.

Def 6: Si $E \in \mathcal{E}$, alors on définit le langage $L(E)$ par induction: [Car]
- $L(\emptyset)=\emptyset$; $L(a)=\{a\}$ pour tout $a \in \Sigma$
- Si $E_1, E_2 \in \mathcal{E}$, alors $L(E_1 \cdot E_2)=L(E_1)\cdot L(E_2)$, $L(E_1+E_2)=L(E_1)+L(E_2)$ et $L(E_1^*)=L(E_1)^*$.

Deux expressions rationnelles sont dites équivalentes si et seulement si elles induisent le même langage.

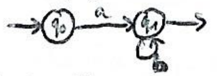
Ex 7: $L((a+b)^*)=(\{a\}+\{b\})^*$.

Prop 8: Les langages rationnels sont les langages induits par les expressions rationnels.

II. Automates finis et langages reconnaissables

Def 9: Un automate fini sur Σ est un quadruplet $\mathcal{A}=(Q,\delta,I,F)$ où : [Car]
- Q est un ensemble fini d'états
- $\delta: Q\times\Sigma \longrightarrow \mathcal{P}(Q)$ est la fonction de transition
- $F \subseteq Q$ est l'ensemble des états finaux
- $I \subseteq Q$ est l'ensemble des états initiaux

Ex 10: Si $Q=\{q_0,q_1\}$, $\Sigma=\{a,b\}$, $\delta(q_0.a)=q_1$, $\delta(q_1.b)=q_1$, $I=\{q_0\}$, $F=\{q_1\}$, on a

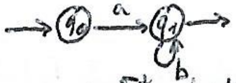


.

On étend δ à $\mathcal{P}(Q)\times\Sigma^*$ par $\delta(q,\varepsilon)=q$ et $\delta(q,w\cdot a)=\delta(\delta(q,w),a)$.

Def 11: Soit $\mathcal{A}$ un automate, on appelle langage accepté par $\mathcal{A}$ le langage $L(\mathcal{A})=\{w\in\Sigma^* \mid \exists q_0 \in I \quad \delta(q_0.w)\cap F \neq \emptyset\}$. [Car]

Ex 12: Si $\mathcal{A}$:



, on a $L(\mathcal{A})=L(a\cdot b^*)$.

Def 13: Un langage sur Σ^* est dit reconnaissable s'il est accepté par un automate fini. On note $Rec(\Sigma^*)$ l'ensemble des langages reconnaissables sur Σ^*.

Def 14:
- Un automate $\mathcal{A}$ est déterministe si $|I|=1$ et $\forall (q,a)\in Q\times\Sigma \quad |\delta(q,a)|\leq 1$. [Car] [Saka]
- $\mathcal{A}$ est complet si $\forall q \in Q \ \forall a \in \Sigma \quad \delta(q,a)\neq\emptyset$.
- $\mathcal{A}$ est émondé si $\forall q \in Q \ \exists i \in I \ \exists w \in \Sigma^* \ q \in \delta(i,w)$ et $\forall q \in Q \ \exists f \in F \ \exists w \in \Sigma^* \ f \in \delta(q,w)$

Prop 15: Pour tout automate fini, on peut construire un automate fini déterministe (resp. complet, resp. émondé) qui reconnait le même langage.

Appli 16: Le problème du langage vide et le problème du mot sont décidables pour les langages reconnaissables.

Prop 17: L'ensemble des langages reconnaissables est stable par intersection, passage au complémentaire et opérations rationnelles. [Saka]

Thm 18 (Kleene): $Rec(\Sigma^*)=Rat(\Sigma^*)$.

Passage des automates aux expressions: [BBCS]
- Algorithme de McNaughton-Yamada
- Algorithme de Brzozowski-McCluskey
- Élimination de Gauss avec le lemme d'Arden

Lem 19 (d'Arden): Soient $K, L \subseteq K^*$ et soit l'équation $X = KX + L$ où l'inconnue X désigne un langage.
- si $\varepsilon \notin K$, l'unique solution de l'équation est $X = K^*L$.
- si $\varepsilon \in K$, les solutions sont de la forme $X = K^*(L+Y)$ où $Y \subseteq \Sigma^*$.

Passage des expressions aux automates:
- Algorithme de Thompson

[CaRo]

Lem 20 (de l'étoile): Soit L un langage rationnel, il existe $N \in \mathbb{N}$ tel que pour tout $w \in L$, $|w| \geq N$, il existe $u, \alpha, v \in \Sigma^*$ avec $\alpha \neq \varepsilon$ tels que $w = u\alpha v$ et $\forall n \in \mathbb{N}^*$ $u\alpha^n v \in L$.

Appli 21: Le langage $\{a^n b^n \mid n \in \mathbb{N}\}$ n'est pas rationnel.

Thm 22: Le problème UNIVERSALITE est Pspace-complet,

UNIVERSALITE : | entrée : une expression rationnelle E
| sortie : oui si $L(E) = \Sigma^*$, non sinon

III. Réduction d'automates

Def 23: Soient $\mathcal{A}_1 = (Q_1, \delta_1, I_1, F_1)$ et $\mathcal{A}_2 = (Q_2, \delta_2, I_2, F_2)$ deux automates finis. Un morphisme $\varphi : \mathcal{A}_1 \to \mathcal{A}_2$ est une application de Q_1 dans Q_2 telle que :
- $\varphi(I_1) \subseteq I_2$
- $\varphi(F_1) \subseteq F_2$
- $\forall p, q \in Q_1$ $q \in \delta_1(p,a) \Rightarrow \varphi(q) \in \delta_2(\varphi(p), a)$.

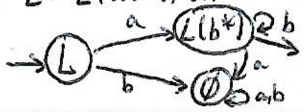
Prop 24: Si $\varphi : \mathcal{A}_1 \to \mathcal{A}_2$ est un morphisme d'automates surjectif entre deux automates déterministes complets, alors $L(\mathcal{A}_1) = L(\mathcal{A}_2)$.

[Saka]

Def 25: On définit la relation d'ordre $\lesssim$ sur l'ensemble des automates déterministes complets par $\mathcal{A}_1 \lesssim \mathcal{A}_2$ s'il existe un morphisme $\varphi : \mathcal{A}_1 \to \mathcal{A}_2$ surjectif.
Un automate est dit minimal s'il est minimal pour $\lesssim$.

Def 26: Soit $L \subseteq \Sigma^*$, $u \in \Sigma^*$. On définit le résiduel de L par u comme le langage $u^{-1}L = \{v \in \Sigma^* \mid uv \in L\}$, et l'automate des résiduels par $R(L) = \{Q_L, \delta_L, \{i_L\}, F_L\}$ avec :
- $Q_L = \{u^{-1}L \mid u \in \Sigma^*\}$
- $\delta_L(u^{-1}L, a) = a^{-1}(u^{-1}L) = (ua)^{-1}L$
- $i_L = L = \varepsilon^{-1}L$
- $F_L = \{u^{-1}L \mid u \in L\} = \{u^{-1}L \mid \varepsilon \in u^{-1}L\}$

Ex 27: Pour $L = L(ab^*)$, on a l'automate des résiduels suivant:



Prop 28: L est reconnaissable si et seulement s'il possède un nombre fini de résiduels. De plus, $R(L)$ est minimal.

Def 29: Soit $\mathcal{A}$ un automate fini déterministe. Une relation d'équivalence $\sim$ sur Q est une congruence si :
- $\forall p,q$ $p \sim q \Rightarrow \forall a \in \Sigma$ $\delta(p,a) \sim \delta(q,a)$ (compatibilité avec δ)
- $\forall p,q$ $p \sim q \Rightarrow p \in F$ ssi $q \in F$ (saturation de F)

Def 30: Si $\mathcal{A}$ est un automate fini déterministe et $\sim$ est une relation d'équivalence sur Q, on définit le quotient de $\mathcal{A}$ par $\sim$ comme : $\mathcal{A}/\!\sim\; = (Q/\!\sim, \delta_\sim, \{[i]\}, \{[f] \mid f \in F\})$ où $\delta_\sim([p], a) = [\delta(, a)]$.

Prop 31: On a $L(\mathcal{A}/\!\sim) = L(\mathcal{A})$.

Def 32: Soit $\mathcal{A}$ un automate fini. L'équivalence sur Q définie par $p \equiv q$ ssi $\forall w \in \Sigma^*$ $\delta(p,w) \in F \Leftrightarrow \delta(q,w) \in F$ est appelée congruence de Nerode.

Rem 33: $\mathcal{A}/\!\equiv$ est isomorphe à $R(L)$.

Algorithme de Hopcroft:

Principe: Calcul de la congruence de Nerode par raffinements successifs

Def 34: Soit $\mathcal{A} = (Q, \delta, \{i\}, F)$ un automate complet, déterministe et émondé, soient $a \in \Sigma$, $B, C \subseteq Q$. La partie B est stable pour (C,a) si $B \cdot a \subseteq C$ ou $B \cdot a \cap C = \emptyset$ $(B \cdot a = \{q \cdot a \mid q \in B\})$.
Sinon, la paire (C,a) coupe B en deux parties B_1 et B_2 :
$B_1 = \{q \in B \mid q \cdot a \in C\}$, $B_2 = \{q \in B \mid q \cdot a \notin C\}$.

[BBC]

Algo 35: Hopcroft $(\mathcal{A})$ =

```
P ← $(F, Q \setminus F)$;
S ← $\{(\min(F, Q \setminus F), a) \mid a \in A\}$;      $\left(\min(A,B) = \begin{cases} A & \text{si } |A| \leq |B| \\ B & \text{sinon} \end{cases}\right)$
tant que $S \neq \emptyset$:
    choisir $(C,a) \in S$;
    $S \leftarrow S \setminus \{(C,a)\}$;
    pour $B \in P$ coupé par $(C,a)$ en $B_1, B_2$:
        remplacer $B$ par $B_1, B_2$ dans $P$;
        pour $b \in \Sigma$
            si $(B,b) \in S$ alors
                remplacer $(B,b)$ par $(B_1,b), (B_2,b)$ dans $S$
            sinon
                ajouter $(\min(B_1,B_2), b)$ à $S$
```

(Complexité : pire cas en $O(|\Sigma| \times |Q| \log |Q|)$)

DEV 1

IV. Applications

1. Recherche de motifs

Problème : trouver un mot fini $w \in \Sigma^*$ dans un texte $t \in \Sigma^*$.

entrée : $w = w_1 \cdots w_m$ (mot à trouver), $t = t_1 \cdots t_n$ (texte)
sortie : k tel que $w_1 \cdots w_m = t_{k+1} \cdots t_{k+m}$

[BBC]

Algo 36 : Recherche_naïve (w,t) :

```
  $i=1$ ; $j=1$
  tant que $i \leq m$ et $j \leq n$ :
      si $t[j]=w[i]$ alors:
          $i=i+1$ ; $j=j+1$
      sinon
          $j=j-i+2$ ; $i=1$
  si $i>m$ alors
      renvoyer $j-m$
  sinon
      pas d'occurrence
```

Complexité :
pire cas en $O(nm)$

Def 37 : Soit $v \in \Sigma^*$. Le bord de v est le plus long sous-mot strict de v qui est à la fois préfixe et suffixe de v.
On définit $\beta : \{0, \dots, m\} \longrightarrow \{-1, \dots, m-1\}$ par
$\beta(0) = -1$, $\beta(i) = |\mathrm{bord}(w_1 \cdots w_i)|$

Ex 38 : Pour $w = abacaba$, on a $\mathrm{bord}(w) = aba$,
$\beta(3) = 1$, $\beta(6) = 2$, $\beta(7) = 3$.

[Saka]

Def 39 : On définit l'automate des occurences d'un mot u : $\mathcal{A}(Q, \delta, \varepsilon, u)$
avec :
- $Q = \{$préfixes de $u\}$;
- ε état initial (préfixe vide de u) ;
- u état final (préfixe plein de u) ;
- si $v \in Q$ et $a \in \Sigma$, $\delta(v,a) = \begin{cases} va & \text{si } va \text{ est préfixe de } u \\ \mathrm{bord}(v) \cdot a & \text{sinon} \end{cases}$

Prop 40 : $\mathcal{A}$ reconnaît le langage $\Sigma^* \{u\}$.

Algorithme de Morris-Pratt :
Présentation de l'algorithme, terminaison, correction et complexité.

DEV 2

2. Arithmétique de Presburger

[Car]

Def 41 : L'arithmétique de Presburger est la théorie du premier ordre des entiers munis de l'addition.
C'est l'ensemble des formules logiques engendrées par la grammaire
$\varphi = x = 0 \mid x = y + z \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \neg\varphi \mid \forall x\ \varphi \mid \exists x\ \varphi$

Def 42 : On dit qu'une théorie logique est décidable si et seulement si il est décidable de savoir si une formule close est satisfiable.

Thm 43 : L'arithmétique de Presburger est décidable.
entrée : φ une formule close de l'arithmétique de Presburger
sortie : oui si φ est satisfiable, non sinon.

Peut aussi rentrer dans la leçon :
- Applications à l'analyse lexicale.
- Extensions du lemme de l'étoile.

Références :
- [BBC] Beauquier, Berstel, Chrétienne – Éléments d'algorithmique
- [Car] Carton – Langages formels, calculabilité, complexité
- [Saka] Sakarovitch – Éléments de théorie des automates