Langages formels et calculabilité

L3 RI

Table des matières

1 Langages Formels

1.1 Automates

Analogie langages et équations

$\emptyset$	0
$\{\epsilon\}$	1
$\cup$	$+$
$.$	$\times$

Expressions rationnelles Générées par la grammaire :

$$E = \emptyset \mid \epsilon \mid (E.E) \mid (E \cup E) \mid E^*$$

Logique monadique du second ordre Exemple : $\exists x, a(x)$ il existe une position s dans le mot où la lettre est a.

Générées par la grammaire :

$$\varphi = x \mid a(x) \mid S(x,y) \mid x = y \mid x < y \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \to \varphi) \mid \exists x\varphi \mid \exists X\varphi \mid \forall x\varphi \mid \forall X\varphi$$

Théorème de Kleene $Rat(\Sigma) = Rec(\Sigma)$

Construction de l'automate depuis une expression e en $\mathcal{O}(|e|)$ états. Construction de l'expr depuis un automate à n états en $\mathcal{O}(4^n)$. On crée un automate généralisé puis on en supprime les états.

Lemme de l'étoile / de pompage L rationnel. $\exists N \in \mathbb{N},\ \forall m \in L$ avec $|m| \geq N$, alors $m = uwv$, $w \neq \epsilon$, et $\forall n \in \mathbb{N},\ uw^{\mathrm{n}}v \in L$.

Langages rationnels stables par intersection

Unicité (à iso. près) de l'automate (det. complet) minimal On quotiente l'automate par la relation d'équivalence entre les états : deux états sont équiv. si les langages reconnus à partir d'eux sont les mêmes.

Algorithme de Moore Pour obtenir l'automate minimal. Algorithme des "patates qui se stabilisent". Au début, une classe d'équivalence pour les états finaux et une pour les autres. Ensuite, on reste dans la même classe d'éq. si en lisant n'importe quelle lettre on arrivait dans la même classe pour la relation d'éq. précédente. Dès qu'on a les mêmes classes d'éq, on s'arrête et on quotiente l'automate initial par cette relation d'éq. pour avoir le minimal.

Complexité moyenne $\mathcal{O}(n \log(n))$ pire $\mathcal{O}(|\Sigma| n^2)$.

Reconnaissance par morphisme L rec. par $\mu \colon \Sigma^* \mapsto M$ ssi $\exists P \subseteq M$, $L = \mu^{-1}(P)$.

L rationnel ssi reconnu par morphisme de Σ dans M fini.

Monoïde syntaxique Pour un langage L, on crée la rel. d'éq. : $m_1 \sim m_2$ ssi $\forall u, v \in \Sigma^*, (um_1v \in L \ ssi \ um_2v \in L)$. Σ^* quotienté par cette relation est le monoïde syntaxique, isomorphe au mono. des transitions de l'automate minimal de L.

Expressions rationnelles généralisées

$$E = \emptyset \mid \epsilon \mid E.E \mid E \cup E \mid E^{\mathrm{c}} \mid E^*$$

Langage sans étoile L sans étoile s'il existe E expr. généralisée, $L(E) = L$ et E de hauteur d'étoile généralisée (nombre max d'étoiles imbriquées, en admettant les complémentaires) nulle.

Théorème Schützenberger L rationnel. L sans étoile ssi son mono. syntaxique est apériodique ($\forall m, \exists i \in \mathbb{N}, m^{\mathrm{i}} = m^{\mathrm{i+1}}$).

1.2 Grammaires

Grammaire

$$G = (\Sigma, V, S, R)$$

V non terminaux, $S \in V$ axiome, $R \subseteq V \times (V \cup \Sigma)^*$.

Dérivation $u \to v$ si $\exists (X, m) \in R$ et $u = u_1 X u_2,\ v = u_1 m u_2$. Dériv. gauche si $u_1 \in \Sigma^*$.

Grammaire linéaire à droite pour des règles de la forme

$$A \to B,\ A \to aB,\ A \to a$$

.

Lemme fondamental $\forall k \in \mathbb{N}, \forall \alpha_1, \alpha_2 \in (\Sigma \cup V)^*, u \in (\Sigma \cup V)^*$, si $\alpha_1\alpha_2 \to^k u$, alors $\exists u_1, u_2 \in (\Sigma \cup V)^*, \exists k_1, k_2 \in \mathbb{N}$ tels que $u = u_1u_2$, $\alpha_1 \to^{k_1} u_1$, $\alpha_2 \to^{k_2} u_2$ et $k = k_1 + k_2$.

Stabilité de Alg(Σ) par composition, union, étoile et morphisme On en déduit aussi $Rat(\Sigma) \subseteq Alg(\Sigma)$.

Pas stable par intersection et complémentaire !

Grammaires ambigües $\exists u \in L(G)$ tel que u est le mot des feuilles de 2 arbres de dérivation. Décider de l'ambiguité est indécidable.

Lemme de l'étoile / Bar-Hillel, Perles et Shamir L algébrique. $\exists K \in \mathbb{N}$ tel que $\forall u \in L,\ |u| \geq K$, alors $u = xvywz$ avec $vw \neq \epsilon$, $|vyw| \leq K$, $\forall i \in \mathbb{N}, xv^iyw^iz \in L$.

Nettoyage de grammaire On supprime d'abord les règles contenant des symboles non productifs (qui ne terminent jamais sur une lettre de Σ). Puis les non accessibles depuis S.

Formes normales de Chomsky Les règles dont de la forme $N \to BC$ ou $N \to a$ ou $S \to \epsilon$. La taille de la forme normale est au pire quadratique en la taille de la grammaire initiale.

Algorithme CYK / Cocke, Younger et Kasari Prend une grammaire G en FNC. Prog dynamique. On calcule les $\mathcal{N}(i,j) = \{N \in V \mid N \to^* w[i..j]\}$.

Cas récursif : $i < j$, $\mathcal{N}(i,j) = \{N \in V \mid \exists N \to BC, \exists k \in \{i..j-1\}$avec $B \in \mathcal{N}(i,k), C \in \mathcal{N}(k+1,j)\}$.

On fait varier l: $2 \to |w|$, i: $1 \to |w| - l + 1$, $j = i + l - 1$.

Complexité $\mathcal{O}(|w|^3 P(|G|))$ où P polynôme.

Théorème de Chomsky-Schützenberger $L \in Alg(\Sigma)$ ssi $\exists n \in \mathbb{N}, \mu : \{a_1, \ldots, a_n, \bar{a_1}, \ldots, \bar{a_n}\} \to \Sigma^*, k \in Rat(\Sigma_n)$ tels que $L = \mu({D_n}^* \cap K)$. Où ${D_n}^*$ est le langage de Dyck de n parenthèses (ouvertes et fermées).

1.3 Automates à piles

Automates à piles On rajoute Γ un alphabet de pile, et $\delta \subseteq Q \times \Sigma \cup \{\epsilon\} \times \Gamma \cup \{\epsilon\} \times Q \times \Gamma^*$ finie. Équivalence avec $Alg(\Sigma)$. $Rat(\Sigma)$ stable par complémentaire.

Automate à pile déterministe On peut voir δ comme une fonction partielle : $Q \times \Sigma \cup \{\epsilon\} \times \Gamma \cup \{\epsilon\} \to Q \times \Gamma^*$ telle que $\forall q \in Q, a \in \Sigma, x \in \Gamma$, exactement une de ces valeurs est définie :

$$\delta(q,a,x),\ \delta(q,a,\epsilon),\ \delta(q\epsilon,x),\ \delta(q,\epsilon,\epsilon).$$

$\{a^n b^n\}$ déterministe. Langage des palindromes non déterministe.

1.4 Analyse syntaxique

Analyse LL(1) Analyse descendante. Glouton : on choisit la règle en fonction du caractère suivant.

premier(α) = $\{a \in \Sigma \mid \exists \beta \in (\Sigma \cup V)^*$ tel que $\alpha \to {}^*a\beta\}$ ($\cup\{$poisson$\}$ si $\alpha \to \epsilon$).

suivant(N) = $\{a \in \Sigma \mid \exists \gamma\beta \in (\Sigma \cup V)^*$ tels que $S \to {}^*\gamma N a\beta\}$ ($\cup\{$EOF$\}$ si $\exists \alpha \mid S \to \alpha N$).

G est LL(1) si $\forall N \in V$, $N \to \alpha_1 \dots N \to \alpha_k$, premier($\alpha_i$) disjoints 2 à 2 et si $N \to {}^*\epsilon$, suivant(N) disjoints des premier(α_i).

On peut alors prendre la règle qui correspond au premier caractère à chaque fois.

Analyse LR(0) On construit l'automate des items : états ensembles des règles possibles et curseur. On est LR(0) s'il n'y a pas de conflits (réduc, lecture).

Comparaison	CYK	générique, cubique en |m|, poly en |G|
	LL(1)	linéaire en |m|, poly en |G|
	LR(0)	linéaire en |m|, expo en |G|

2 Calculabilité

2.1 Machines de Turing

Machine de Turing Septuplet $(Q, \Sigma, R, \delta, q_0, A, \#)$ où R alphabet de ruban. Une transition part d'un état, lit une lettre, écrit une autre lettre, va dans un autre état, bouge à droite ou à gauche sur le ruban.

Une configuration : état, contenu du ruban, position curseur

Une représentation : mot du ruban avant curseur, état, mot du ruban après $\subseteq R^* \times Q \times (R^*(R\backslash\{\#\}) \cup \{\epsilon\})$.

Thèse de Church : reconnu par une procédure effective ssi décidé par une MT.

Équivalence avec ruban double, infini des deux côtés, plusieurs rubans, mémoire accès direct, non déterministe.

R et RE Langage accepté : ensemble des mots pour lesquels la machine termine.
Langage décidé : M accepte L, et sans calcul infini.
R ensemble des décidés $\subset$ RE ensemble des acceptés.

Fonctions récursives primitives Base : 0(), successeur, projections. On peut utiliser des règles de composition et récursion.

Calculables et dénombrables mais pas suffisant (Ackermann, diagonalisation)

Fonctions μ-récursives Minimisation non bornée : plus petit indice tel qu'un prédicat est vrai. 0 si ça n'existe pas. Un prédicat est sûr si ça existe. Pour obtenir les μ-récursives, on s'autorise les mêmes constructeurs que les récursives primitives + minimisation non bornée de prédicats sûrs.

Èquivalent à calculable par une MT.

2.2 Non calculabilité

Diagonalisation Ensemble des machines et Σ^* dénombrables, pas l'ensemble des langages.

Déduction sur R et RE 3 situations

— $L \in R \wedge \bar{L} \in R$
— $L \notin RE \wedge \bar{L} \notin RE$
— $L \notin RE \wedge \bar{L} \in RE \backslash R$

Réduction On sait L_1 indécidable, on montre L_2 indécidable. On dit qu'on réduit de L_1 à L_2. On note $L_1 \leq_C L_2$. On suppose avoir une MT décidant L_2 et on construit une MT décidant L_1 à partir de cette machine $\mapsto$ absurde.

Théorème de Rice Propriété $S \subseteq RE$. Non-triviale ($\neq \emptyset$ et $\neq RE$) $\Rightarrow$ indécidable.

Exemples de problèmes indécidables

Problème LU : $LU = \{(M, w) : M \textit{ accepte } w\}$
Problème de l'arrêt : $H = \{(M, w) : M \textit{ s'arrête en lisant } w\}$
Problème de Post

3 Vu en TD

Lemme d'Arden A et B deux langages. $\epsilon \notin A$. L'équation $X = (A.X) \cup B$ a pour unique solution A^*B. (si $\epsilon \in A$ la solution n'est pas unique mais il s'agit de la plus petite).

Lemme d'Ogden L algébrique. $\exists N$ tel que $\forall w \in N, |w| \geq N$, pour tout choix de N positions distinguées dans w, $\exists$ factorisation $w = xuyvz$ telle que xuy ou yvz contiennent au moins une position distinguée, uyv a au plus N positions distinguées et $xu^n yv^n z \in L \ \forall n \in \mathbb{N}$.