Langages Formels, Calculabilité, Complexité

Mickaël Thomazo
Lucas Larroque

12 octobre 2023

Table des matières

Première partie
Cours 1 28/09

1 Langages, Automates, RegExp, Monoïdes finis

Définition 1.0.1. *On appelle* alphabet *un ensemble fini* Σ *de lettres.*
On appelle mot *une suite finie de lettres.*
On appelle langage *un ensemble de mots*

Définition 1.0.2. *On appelle* automate sur l'alphabet Σ *un graphe orienté dont les arêtes sont étiquetées par les lettres de l'alphabet* Σ
Formellement, c'est un quadruplet $\mathcal{A} = (Q, \Sigma, I, F, \delta)$ *ou :*

— Q *est un ensemble fini d'états*
— Σ *est un alphabet*
— $I \subseteq Q$
— $F \subseteq Q$

— $\delta : Q \times \Sigma \to 2^Q$

Un calcul de $\mathcal{A}$ sur $w = a_0 \dots a_n$ est une séquence $q_0 \dots q_n$ telle que $q_0 \in I$, $\forall i \geq 1$, $q_i \in \delta(q_{i-1}, a_i)$
On appelle Langage reconnu par $\mathcal{A}$ l'ensemble $\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^* \mid \exists q_0 \dots q_n$ calcul de $\mathcal{A}$ sur w où $q_n \in F\}$
On dit que $\mathcal{A}$ est déterministe si :

— $\forall q, a, |\delta(q,a)| \leq 1$
— $|I| = 1$

Définition 1.0.3. *Une expression régulière est de la forme :*

— $a \in \Sigma$
— $\varnothing$
— $r + r$ *(+ désigne l'union : $L_1 + L_2 = \{w \in L_1 \cup L_2\}$)*
— $r \cdot r$ *($\cdot$ désigne la concaténation : $L_1 \cdot L_2 = \{w_1 w_2 \mid w_1 \in L_1,\ w_2 \in L_2\}$)*
— r^* *($*$ désigne l'étoile de Kleene, $L^* = \left\{ \bigodot_{w \in s} w \;\middle|\; s \in \bigcup_{n \in \mathbb{N}} L^n \right\}$)*

Définition 1.0.4 (Automate des Parties)**.** *On pose, si $\mathcal{A} = (Q, \Sigma, I, F, \delta)$ est un automate :*

— $\hat{Q} = 2^Q = \{q_S \mid S \subset Q\}$
— $\hat{I} = \{q_I\}$
— $\hat{F} = \{q_S \mid S \cap F \neq \varnothing\}$
— $\hat{\delta}(q_S, a) = \{q_{S'}\}$ *avec* $S' = \bigcup_{q \in S} \delta(q,a)$

Alors, $\hat{\mathcal{A}} = (\hat{Q}, \Sigma, \hat{I}, \hat{F}, \hat{\delta})$ est un automate déterministe reconnaissant $\mathcal{L}(\mathcal{A})$

Démonstration. On procède par double inclusion :

— ($\subset$) On introduit un calcul de $w \in \mathcal{L}(\mathcal{A})$ sur $\hat{\mathcal{A}}$ et on vérifie par récurrence que son dernier état est final.
— On procède de même pour la réciproque.

■

Définition 1.0.5. *Un monoïde est un magma associatif unifère.*
Un morphisme de monoïde est une application $\varphi : (N, \cdot_N) \to (M, \cdot_M)$ telle que :

— $\varphi(1_N) = 1_M$
— $\varphi(n_1 n_2) = \varphi(n_1)\varphi(n_2)$

Un langage L est reconnu par $(M, \times)$ ssi il existe $P \subset M$ tel que $L = \varphi^{-1}(P)$ où φ est un morphisme de Σ^* dans M

Proposition 1.0.1. *$L \subseteq \Sigma^*$ est reconnu par un automate ssi L est reconnu par un monoïde fini.*

Démonstration. — Soit L reconnu par un monoïde fini $(M, \times)$. Soit φ un morphisme tel que $L = \varphi^{-1}(P)$, $P \subset M$. On pose $\mathcal{A} = (M, \Sigma, \{1\}, P, \delta)$ où $\delta(q,a) = q \times \varphi(a)$. Alors, $\mathcal{A}$ reconnaît L.

— Soit $\mathcal{A}$, déterministe, complet, reconnaissant L. Pour $a \in \Sigma$, $a \to \varphi_a : q \in Q \mapsto \delta(q,a)$ induit par induction un morphisme de $(\Sigma^*, \cdot)$ dans $(Q^Q, \circ)$. Alors, avec $P = \left\{ f \in Q^Q \mid f(i) \in F_{\mathcal{A}} \right\}$. On a défini le monoïde des transitions de $\mathcal{A}$.

■

Deuxième partie

Cours 2 - 5/10

2 Lemme de Pompage

Théorème 2.0.1 (Lemme de Pompage/Lemme de l'Etoile)**.** *Si L est un langage régulier, $\exists n \in \mathbb{N}$ $\forall w \in L, |w| \geq n \Rightarrow \exists x, y, z$ tels que :*

— $w = xyz$
— $|xy| \leq n$
— $y \neq \varepsilon$
— $\forall n \geq 0, xy^n z \in L$

Démonstration. Faire un calcul de $\mathcal{A}$ sur w tel que $|w| \geq n$. Celui-ci passe deux fois par le même état. ∎

3 Langages Quotients

3.1 Quotients d'un Langage à Gauche

Définition 3.1.1 (Quotient à Gauche)**.** *Soit $L, K \subseteq \Sigma^*, u \in \Sigma^*$.*
Le quotient à gauche de L par u noté $u^{-1}L$ est : $\{v \in \Sigma^* \mid uv \in L\}$
Le quotient à gauche de L par K, $K^{-1}L$ est $\bigcup_{u \in K} u^{-1}L$

Proposition 3.1.1. — $w^{-1}(K + L) = w^{-1}K + w^{-1}L$
— $(wa)^{-1}L = a^{-1}(w^{-1}L)$
— $w^{-1}(KL) = (w^{-1}K \cdot L) + \sum_{u \in L, v \in \Sigma^* w = uv} v^{-1}L$

3.2 Quotient d'un Automate à Gauche

Définition 3.2.1. *On définit le quotient à gauche d'un automate par un mot u comme celui obtenu en remplaçant les états initiaux par les résultats d'un calcul de l'automate sur u.*

Proposition 3.2.1. *L est régulier si et seulement si il a un nombre fini de quotients à gauche.*

Démonstration. — Un automate reconnaissant L a au plus un quotient par état.
— Posons $A_L = \left(\Sigma, \left\{u^{-1}L \mid u \in \Sigma^*\right\}, I = L = \varepsilon^{-1}L, F =, \delta(w^{-1}L, a) = a^{-1}(w^{-1}L)\right)$
Par récurrence, le calcul de A_L sur w termine en $w^{-1}L$

∎

3.3 Construction de l'Automate Minimal

Définition 3.3.1. *Deux états q_1, q_2 sont distingables si : $\exists w \in \Sigma^*, \delta(q_1, w) \in F, \delta(q_2 \notin F)$.*

Proposition 3.3.1. *q_1 et q_2 sont distingables s'ils n'ont pas même quotient à gauche. Si $\delta(q, a)$ est distingable $\delta(q^{'}, a)$, $q, q^{'}$ sont distingables.*
La relation q, q' sont distingables est une relation d'équivalence.

Troisième partie
Cours 3 - 12/10

4 Langages et Logique

4.1 Objectif

On associe à $w \in \Sigma^*$ une structure D_w et à $L \subseteq \Sigma^*$ une structure φ_L telles que : $w \in L \setminus \{\varepsilon\} \Leftrightarrow D_w \vdash \varphi_L$. On se place dans le cadre de la logique du premier ordre et de la monadique du second ordre.

Définition 4.1.1. *On définit* $\mathsf{pos}(w) = \{0, \ldots, |w| - 1\}$*. On définit une signature i.e. un ensemble de relations :*

$$\forall a \in \Sigma,\ L_a \ \textit{d'arité}\ 1$$
$$\leq,\ \textit{l'ordre strict sur}\ \mathsf{pos}(w)$$

On définit alors la structure $D_w = \left(\mathsf{pos}(w), \left\{L_a^{D_w}\right\}_{a \in \Sigma}, <_w\right)$

Remarque 4.1.0.1. *On aurait pû remplacer* $<_w$ *par* $succ_w$*, mais on perd en expressivité.*

4.2 Logique du Premier Ordre et Monadique du Second Ordre

Définition 4.2.1 (Logique du Premier Ordre)**.** *On définit par induction la logique du premier ordre.*

— *Constantes*

— *Variables*

— *Si* R *est une relation d'arité* n*,* t_i *des termes :* $R(t_1, \ldots, t_n)$

— $\neg\varphi$*,* $\varphi_1 \wedge \varphi_2$*,* $\varphi_1 \vee \varphi_2$

— $\forall x, \varphi, \exists \varphi$

On cherche à associer à $\varphi : \exists x,\ (L_0(x) \wedge \forall y(y < x \to (L_1(y))))$, un langage $L_\varphi = \{w \mid D_w \vdash \varphi\}$.

Définition 4.2.2. *Monadique du Second Ordre Sont des formules :*

— $\forall X,\ \varphi$ *avec* X*, variable du second ordre qui a une arité associée.*

— $\exists X,\ \varphi$ *avec* X*, variable du second ordre qui a une arité associée.*

— $(x_1, \ldots, x_n) \in X$ *avec* X *d'arité* n*. On trouve aussi une formule pour les graphes qui mettent en relation deux sommets* s, t *:*

On se restreint dans la suite à des variables d'arité 1

On considère un vocabulaire qui contient une relation E ("arêtes d'un graphe"). On représente un graphe par D_G l'ensemble de ses sommets et E^{D_G} l'ensemble des arêtes de ce graphe.

Exemple 4.1. *On trouve alors une formule pour représenter tous les graphes* 3*-coloriables :*

$$\begin{aligned}&\exists X_1, \exists X_2, \exists X_3 \left(\forall x \left(X_1(x) \vee X_2(x) \vee X_3(x)\right)\right) \\ &\qquad \wedge \left(\forall x \forall y \left(E(x,y) \to \left(\neg \left(X_1(x) \wedge X_1(y)\right) \wedge \neg \left(X_2(x) \wedge X_2(y)\right) \wedge \neg \left(X_3(x) \wedge X_3(y)\right)\right)\right)\right)\end{aligned} \tag{1}$$

Exemple 4.2. *On trouve aussi une formule pour les graphes qui mettent en relation deux sommets* s, t*. Il s'agit de trouver une relation close par successeur qui contient* s *:*

$$\forall R \left([s \in R \wedge \forall x, y, (R(x) \wedge E(x,y)) \to R(y)] \to t \in R\right)$$

Ainsi, on peut en déduire une méthode pour reconnaître le langage d'un automate $\mathcal{A} = (\{0, \ldots, k\}, \Sigma, 0, \Delta, F)$ avec une formule $\varphi_{\mathcal{A}}$ de la monadique du second ordre.

Théorème 4.2.1. *Un langage* $L = L(\mathcal{A})$ *est régulier, si et seulement si il existe une formule* $\varphi = \varphi_{\mathcal{A}}$ *telle que* $\forall w \in L, D_w \vdash \varphi$*.*

Démonstration. — ($\Rightarrow$) : On peut obtenir le premier élément d'un mot par la formule $\mathsf{first}(x) = \forall y((x = y) \vee x < y)$. On peut faire de même pour savoir si un couple est composé d'une paire successeur-successeuse de l'automate, ou si x est la dernière lettre.
On sépare les positions d'un mot selon l'état de l'automate depuis lequel on part. Il faut alors vérifier que le premier élément est bien dans un état initial, que toute transition est bien valide, qu'on est dans au moins un état avant chaque lettre, et que la dernière position

est bien écrite depuis une transition vers un état acceptant.
On obtient alors, en notant k le nombre d'états, et 0 l'état initial :

$$
\begin{aligned}
\varphi_{\mathcal{A}} : \exists X_0, \ldots, \exists X_k \left(\bigwedge_{i \neq j} \forall x,\ \neg \left(X_i(x) \wedge X_j(x) \right) \right) & \\
\forall x \ \left(\mathsf{first}(x) \rightarrow X_0(x) \right) & \\
\forall x, \forall y \ \left(\mathsf{succ}(x, y) \rightarrow \bigvee_{(i,a,j) \in \Delta} \left(X_i(x) \wedge L_a(x) \wedge X_j(y) \right) \right) & \\
\forall x \ \left(\mathsf{last}(x) \rightarrow \bigvee_{\exists j \in F | (i,a,j) \in \Delta} \left(X_i(x) \wedge L_a(x) \right) \right) &
\end{aligned}
\tag{2}
$$

- ($\Leftarrow$) : On procède par induction.
  - Initialisation : On peut facilement exhiber des automates qui reconnaissent les formules atomiques : $\mathsf{Sing}(X), X \subseteq Y, X \subseteq L_a$.
  - Hérédité : On raisonne sur les connecteurs, et on vérifie aisément, par les propriétés de cloture des langages réguliers le résultat. Pour ce qui est de la quantification existentielle, si le langage L défini par $\psi(X_1, \ldots, X_n)$ sur $\Sigma \times \{0,1\}^n$ est reconnu par $\mathcal{A}$. On exhibe un automate reconnu par $\varphi(X_1, \ldots, X_{n-1}) = \exists X_n \psi(X_1, \ldots, X_n)$, il n'a alors plus qu'a trouver une suite de $0-1$ qui définit la n-ième composante additionnelle et fonctionne sur $\Sigma \times \{0,1\}^n$ comme $\mathcal{A}$. Pour le $\forall$, il suffit de prendre la négation du $\exists$

$\blacksquare$