Théorie des langages

Olivier Roques

2016-2017

1 Mots, Langages

Définition 1.1. On définit les termes suivants concernant les *mots* :

- Un *alphabet* est un ensemble fini de symbole, noté Σ.
- Un *mot* est une suite finie d'éléments de Σ. Le mot vide est noté ε.
- La *longueur d'un mot* u est notée $|u|$. On a $|\varepsilon| = 0$.
- u est un *facteur* de $v \in \Sigma^*$ s'il existe u_1, $u_2 \in \Sigma^*$ tels que $v = u_1 u u_2$. Si $u_1 = \varepsilon$, u est un *préfixe* de v, et si $u_2 = \varepsilon$, u est un *suffixe* de v.

Définition 1.2. On définit les termes suivants, concernant les *langages* :

- L'ensemble des mots sur Σ est noté Σ^*.
- Un *langage* est un sous-ensemble de Σ^*.
- L'opération de *concaténation* des langages L_1 et L_2 se définit par :

$$L_1 L_2 = \{u \in \Sigma^* \mid \exists (v, w) \in L_1 \times L_2 \text{ tq } u = vw\}$$

- L'opération de *fermeture de Kleene* d'un langage L se définit par : $L^* = \bigcup_{n \geq 0} L^n$.

2 Langages rationnels

Définition 2.1. Soit Σ un alphabet. Les *langages rationnels* sur Σ sont définis inductivement par :

(i) $\{\varepsilon\}$ et $\emptyset$ sont des langages rationnels.

(ii) $\forall a \in \Sigma$, $\{a\}$ est un langage rationnel.

(iii) Si L_1 et L_2 sont des langages rationnels, alors $L_1 \cup L_2$, $L_1 L_2$ et L_1^* sont des langages rationnels.

3 Automates finis

Définition 3.1. Un *automate fini déterministe complet* est défini par un quintuplet $A = (\Sigma, Q, q_0, F, \delta)$, où :

- Σ est un ensemble fini de symboles (l'alphabet) ;
- Q est un ensemble fini d'états ;
- $q_0 \in Q$ est l'*état initial* ;
- $F \subset Q$ est l'ensemble des *états finaux* ;
- $\delta : Q \times \Sigma \longrightarrow Q$ est appelée *fonction de transition.* Si elle n'est pas totale, l'automate n'est plus complet.

Définition 3.2. On définit les notions suivantes :

- Une *transition* est un triplet $(q, a, r) \in Q \times \Sigma \times Q$ tel que $\delta(q, a) = r$. a est alors appelé l'*étiquette* de cette transition.
- Un *calcul* dans A est une suite d'états $e_1 \dots e_n$ de A telle que pour tout $i \in [\![1\,, n-1]\!]$, il existe a_i tel que (e_i, a_i, e_{i+1}) soit une transition.
- L'*étiquette d'un calcul* est le mot construit par concaténation des étiquettes a_i.
- Un calcul dans A est dit *réussi* si le premier état est q_0 et l'état final est dans F.
- Le *langage reconnu* par l'automate A, noté $L(A)$, est l'ensemble des étiquettes des calculs réussis.

Définition 3.3. Un langage est dit *reconnaissable* s'il existe un automate fini qui le reconnaît. Deux automates finis A_1 et A_2 sont équivalents si et seulement s'ils reconnaissent le même langage.

Définition 3.4. Un état q de A est dit *accessible* s'il existe $u \in \Sigma^*$ tel que $\delta^*(q_0, u) = q$ (fonction δ étendue aux mots). Un état q est dit *co-accessible* s'il existe $u \in \Sigma^*$ tel que $\delta^*(q, u) \in F$. Un état q est dit *utile* s'il est à la fois accessible et co-accessible. Lorsque tous les états d'un automate sont utiles, on dit qu'il est *émondé.*

Théorème 3.1. Si $L(A) \neq \emptyset$ est un langage reconnaissable, alors il est également reconnu par un automate émondé.

Définition 3.5. Un *automate fini non-déterministe* est défini par un quintuplet $A = (\Sigma, Q, I, F, \delta)$, où :

- Σ, Q et F sont définies comme précédemment ;
- $I \subset Q$ sont les états initiaux ;
- $\delta \subset Q \times \Sigma \times Q$. Pour une paire (q, a), il peut donc exister dans A plusieurs transitions possibles.

Définition 3.6. Un *automate fini à transitions spontanées* est défini par un quintuplet $A = (\Sigma, Q, I, F, \delta)$, où :

- $\delta \subset Q \times (\Sigma \cup \{\varepsilon\}) \times Q$
- Σ, Q, I et F sont définies comme précédemment.

Théorème 3.2. Tout langage reconnu par un automate fini non-déterministe ou un automate fini à transitions spontanées est aussi reconnu par un automate fini déterministe.

Théorème 3.3. L'ensemble des langages reconnaissables est stable par passage au complémentaire, union, intersection, concaténation, passage à l'étoile, passage au miroir.

Théorème 3.4 (Théorème de Kleene)**.** Un langage est reconnaissable si et seulement s'il est rationnel.

Algorithme 3.1. Soit L un langage rationnel. L'*algorithme de Thompson* construit un automate fini à transitions spontanées A tel que :

- A reconnaît L,
- A possède un seul état initial et un seul état final ;
- Aucune transition ne sort de l'état final.

Théorème 3.5 (Lemme de l'étoile)**.** Soit L un langage rationnel. Alors il existe un entier N tel que pour tout mot $x \in L$ de longueur $|x| \geq N$, x se factorise en $x = uvw$ où :

- $v \neq \varepsilon$;
- $|uv| \geq N$;
- $\forall n \in \mathbb{N}^*$, $uv^n w \in L$.

Théorème 3.6 (Automate canonique)**.** Il existe un unique automate A_C déterministe complet minimisant le nombre d'états et reconnaissant L. Cet automate s'appelle l'*automate canonique* de L.

4 Grammaires syntagmatiques

Définition 4.1. Une *grammaire syntagmatique* G (de *type* 0) est définie par un quadruplet (N, Σ, P, S) où :

- N est un ensemble fini de symboles appelés *variables* ou *non-terminaux* ;
- Σ est un ensemble fini de symboles appelés *terminaux*. On note $V = (N \cup \Sigma)$ qu'on appelle *vocabulaire* de la grammaire.
- P est une partie de $(N \cup \Sigma)^* \times (N \cup \Sigma)^*$ dont les éléments sont appelés *productions*. Une production (α, β) est notée $\alpha \to \beta$, où α est appelée *partie gauche* et β *partie droite* de la production.
- S est un élément particulier de N appelé *symbole initial* ou *axiome* de la grammaire.

Définition 4.2. On définit la relation *dérive immédiatement*, notée $\Rightarrow_G$, définie sur l'ensemble $V^* \times V^*$ par $\gamma\alpha\delta \Rightarrow_G \gamma\beta\delta$ si et seulement si $\alpha \to \beta$ est une production de G.

Définition 4.3. On définit la relation de *dérivation*, notée $\Rightarrow_G^*$, définie sur l'ensemble $V^* \times V^*$ par $\alpha_1 \Rightarrow_G^* \alpha_m$ si et seulement s'il existe $\alpha_2, \ldots, \alpha_{m-1} \in V^*$ tels que $\alpha_1 \Rightarrow_G \alpha_2 \Rightarrow_G \cdots \Rightarrow_G \alpha_m$.

Définition 4.4. On appelle *langage engendré par* G, noté $L(G)$, le sous-ensemble de Σ^* défini par $L(G) = \{\omega \in \Sigma \mid S \Rightarrow_G^* \omega\}$. Deux grammaires G_1, G_2 sont alors dites *équivalentes* si et seulement si elles engendrent le même langage.

Définition 4.5. On appelle *grammaire contextuelle* (de *type* 1) une grammaire G telle que toute production de G est de la forme $\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$ avec α_1, α_2 et $\beta \in V^*$, $\beta \neq \varepsilon$ et $A \in N$.

Définition 4.6. On appelle *grammaire hors-contexte* (de *type* 2) une grammaire G telle que toute production de G est de la forme $A \to \beta$ avec $A \in N$ et $\beta \in V^*$.

Définition 4.7. On appelle *grammaire régulière* (de *type* 3) une grammaire G telle que toute production de G est soit de la forme $A \to aB$ avec $a \in \Sigma$ et A, $B \in N$, soit de la forme $A \to a$.

Théorème 4.1. On a les résultats suivants (cf. 6. *Notion de calculabilité*) :

- Les langages récursivement énumérables sont exactement les langages engendrés par une grammaire de type 0.
- Tout langage contextuel (engendré par une grammaire de type 1) est récursif.
- Les langages réguliers (de type 3) sont exactement les langages reconnaissables par un automate fini, *i.e.* les langages rationnels.

5 Langages hors-contexte

Définition 5.1. On appelle *dérivation gauche* (resp. *dérivation droite*) d'une grammaire hors-contexte G une dérivation dans laquelle chaque étape de dérivation récrit le non-terminal le plus à gauche (resp. le plus à droite) du pseudo-mot courant. On note $A \Rightarrow_G^L u$ (resp. $A \Rightarrow_G^R u$) un mot u dérivant de A par une dérivation gauche (resp. droite).

Théorème 5.1. Soit G une grammaire hors-contexte d'axiome S et $u \in \Sigma^*$. Alors $S \Rightarrow_G^* u$ si et seulement si $S \Rightarrow_G^L u$ (idem pour $S \Rightarrow_G^R$).

Définition 5.2. Un arbre de dérivation dans G est un arbre $\mathcal{A}$ tel que :

(i) tous les noeuds de $\mathcal{A}$ sont étiquetés par un symbole de V ;

(ii) la racine est étiquetée par S ;

(iii) si un noeud n n'est pas une feuille et porte l'étiquette X, alors $X \in N$;

(iv) si $n_1, \ldots, n_k$ sont les fils de n dans $\mathcal{A}$, d'étiquettes respectives $X_1, \ldots, X_k$, alors $X \longrightarrow X_1 X_2 \ldots X_k$ est une production de G.

Définition 5.3. Une grammaire est dite *ambigüe* s'il existe un mot admettant plusieurs dérivations gauches dans la grammaire. De manière équivalente, une grammaire est ambigüe s'il existe un mot qui admet plusieurs arbres de dérivation.

Définition 5.4. Un langage hors-contexte est *intrinsèquement ambigü* lorsque toutes les grammaires qui l'engendrent sont ambigües.

Théorème 5.2 (Lemme de l'étoile pour les grammaires)**.** Si L est un langage hors-contexte, alors il existe un entier N tel que tout mot $m \in L$ de longueur supérieur à N se décompose en $m = uvwxy$ avec :

(i) $vx \neq \varepsilon$
(ii) $|vwx| < N$
(iii) Pour tout $n \in \mathbb{N}$, $uv^nwx^ny \in L$

Théorème 5.3. Les langages hors-contexte sont stables par union, concaténation, passage à l'étoile, mais pas par concaténation ou complémentation. L'intersection d'un langage régulier et d'un langage hors-contexte est un langage hors-contexte.

6 Notion de calculabilité

Définition 6.1. Un langage L est dit *récursivement énumérable* (ou *semi-décidable*) s'il existe un algorithme $\mathcal{A}$ qui énumère tous les mots de L.

Théorème 6.1. Un langage L est récursivement énumérable si et seulement s'il existe un algorithme $\mathcal{A}$ tel que pour tout mot $u \in \Sigma^*$:

- si $u \in L$, alors $\mathcal{A}$ termine sur u en retournant `true` ;
- si $u \notin L$, alors soit $\mathcal{A}$ termine sur u en retournant `false`, soit $\mathcal{A}$ ne termine pas sur u.

Définition 6.2. Un langage L est dit *récursif* (ou *décidable*) s'il existe un algorithme $\mathcal{A}$ qui, prenant un mot de u de Σ^* renvoie `true` si u est dans L ou `false` sinon. L'algorithme $\mathcal{A}$ *décide* le langage L.

Propriété 6.1. Tout langage récursif est récursivement énumérable. L'inverse n'est pas vrai : il existe des langages récursivement énumérables mais non récursif (*langage d'arrêt* par exemple).

Définition 6.3. Une *machine de Turing* (déterministe) est la donnée de :

(i) un ensemble fini Q d'états ;
(ii) un alphabet de travail Γ ;
(iii) un symbole spécial $b \in \Gamma$ appelé *blanc* ;
(iv) un alphabet $\Sigma \subset \Gamma \setminus \{b\}$ d'entrée / sortie ;
(v) un état initial q_0 ;
(vi) un ensemble d'états finaux $F \subset Q$;
(vii) une fonction de transition $\delta : Q \setminus F \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$.