Devoir 2, partie II : Grammaires de quotients

À rendre au plus tard le 20 mai 2010.
Tout retard sera pénalisé.

		(5)	6	7	8	9
10	11	12	13	14	15	16
17	18	19	(20)			

Une version électronique (PDF) peut m'être envoyée par mail à ⟨schmitz@lsv.ens-cachan.fr⟩, les versions papiers doivent m'être rendues physiquement le 20 mai ou mises dans mon casier au LSV. Il est obligatoire d'avoir rendu les *deux parties* pour bénéficier d'une note strictement positive.

On s'intéresse dans cette deuxième partie du devoir à des *grammaires de quotients*, qui sont utilisées dans des modélisations en langue naturelle.

Le chiffre en regard d'une question est une indication sur sa difficulté ou sa longueur.

Définition 1. Soit $P = \{p_1, \dots, p_n\}$ un ensemble fini de *types primitifs*. Un *type quotient* sur P est un terme défini par la syntaxe abstraite

$$q ::= p \mid q/q \mid q\backslash q$$

où p est un type primitif de P ; on note $Q(P)$ pour l'ensemble des types quotients définis sur P. On définit deux règles de réécritures sur des séquences de types quotients :

$$(q/q')\, q' \to q \qquad (/_e)$$
$$q'\, (q'\backslash q) \to q \qquad (\backslash_e)$$

Une *grammaire de quotients* est un tuple $\mathcal{Q} = \langle P, \Sigma, \tau, q_0 \rangle$ où P est un ensemble fini de types primitifs, Σ un alphabet fini, $\tau \subseteq \Sigma \times Q(P)$ une relation finie de typage, et $q_0 \in Q(P)$ un type distingué.

La relation τ s'étend naturellement en un homomorphisme de Σ^* dans $(2^{Q(P)})^*$ en considérant l'ensemble image de chaque symbole de Σ. Le langage d'une grammaire de quotients $\mathcal{Q}$ est défini par

$$L(\mathcal{Q}) = \{w \in \Sigma^* \mid \tau(w) = E_1 \cdots E_m, \exists q_1 \in E_1, \dots, q_m \in E_m, q_1 \cdots q_m \to^* q_0\} .$$

On étend les règles de réécriture sur les types quotients à des ensembles de types quotients par

$$E_1\, E_2 \to \{q\} \text{ si } \exists q_1 \in E_1, q_2 \in E_2,\ q_1\, q_2 \to q ,$$

où E_1 et E_2 sont dans $2^{Q(P)}$, ce qui permet d'écrire plus simplement

$$L(\mathcal{Q}) = \{w \in \Sigma^* \mid \exists E \in 2^{Q(P)}, q_0 \in E \text{ et } \tau(w) \to^* E\} .$$

Exemple 1. Soit $P = \{d, n, s\}$, $q_0 = s$, et la relation de typage τ définie par la table

Σ	$Q(P)$	note
le, un	d	déterminant
chien, os	$d\backslash n$	nom
croque, dort, mange	s	verbe impératif
dort, mange	$n\backslash s$	verbe intransitif, indicatif
croque, mange	$(n\backslash s)/n$	verbe transitif, indicatif
blanc, jaune	$(d\backslash n)\backslash(d\backslash n)$	adjectif post
petit, joli	$(d\backslash n)/(d\backslash n)$	adjectif pré

Une réécriture pour la phrase « Le chien mange. » serait par exemple :

$$\begin{aligned}\tau(\text{le chien mange}) &= \{d\}\ \{d\backslash n\}\ \{s, n\backslash s, (n\backslash s)/n\}\\ &\to \{n\}\ \{s, n\backslash s, (n\backslash s)/n\}\\ &\to \{s\}\ .\end{aligned}$$

Exercice 1 (Équivalence avec les grammaires algébriques)**.** On montre dans cet exercice que les grammaires de quotients définissent la classe des langages algébriques.

[2] 1. Montrer que, pour toute grammaire par quotient $\mathcal{Q}$, il existe une grammaire algébrique équivalente $\mathcal{G}$.

[3] 2. Montrer que, pour toute grammaire algébrique $\mathcal{G}$, il existe une grammaire de quotients équivalente $\mathcal{Q}$, sauf éventuellement pour le mot vide : $L(\mathcal{Q}) = L(\mathcal{G})\backslash\{\varepsilon\}$.

[1] 3. Question facultative : si votre preuve pour la question précédente est celle que j'attends, vous pouvez aisément en déduire que, pour toute grammaire de quotients $\mathcal{Q}$, il existe une grammaire de quotients $\mathcal{Q}'$ équivalente qui n'utilise dans τ' que des types *normalisés*, c'est-à-dire de la forme p_1, p_1/p_2, ou $(p_1/p_2)/p_3$ avec p_1, p_2 et p_3 des types primitifs. Soit $N(P)$ l'ensemble des types quotients normalisés : cela revient à définir τ' comme une relation de Σ dans $N(P)$.

Exercice 2 (Théorème de représentation des langages algébriques)**.** Vous connaissez déjà un théorème de représentation des langages algébriques, à savoir le théorème de Chomsky-Schützenberger. Nous allons appliquer la forme normale des grammaires de quotients vue à l'exercice 1.3 pour montrer un théorème beaucoup plus fort : il existe un langage algébrique $\mathscr{L}_0$ *fixé* sur un alphabet $\{a, b, c, d\}$ tel que tout langage algébrique L soit une pré-image homomorphique de $\mathscr{L}_0$:

Théorème 1. *Il existe un langage algébrique $\mathscr{L}_0$ sur $\{a, b, c, d\}$ tel que, pour tout langage algébrique L sur un alphabet Σ, il existe un homomorphisme $h : \Sigma^* \to \{a, b, c, d\}^*$ tel que*

$$L\backslash\{\varepsilon\} = h^{-1}(\mathscr{L}_0)\ .$$

Ce théorème a d'importantes applications en théorie de la complexité, dont on verra un avant-goût très simple avec la question 3.

1. Soit $P_n = \{p_1, \ldots, p_n\}$ un ensemble de types primitifs quelconque. On définit un homomorphisme de codage ψ_n des séquences de multi-ensembles de types quotients normalisés dans $\{a, b, c, d\}^*$, i.e. de $(2^{N(P_n)})^*$ dans $\{a, b, c, d\}^*$, en codant un type quotient de la forme
   – p_i par $a^i bc$,
   – p_i/p_j par $a^i b a^j bc$,
   – $(p_i/p_j)/p_k$ par $a^i b a^j b a^k bc$,
   et un multi-ensemble de tels types comme une séquence finissant par d (en supposant un ordre arbitraire sur les éléments de du multi-ensemble). Par exemple,

$$\begin{aligned}\psi_n(\{(p_3/p_2)/p_1, p_1\}\,\{p_2, p_1/p_3\}\,\{p_1\}) &= \psi_n(\{(p_3/p_2)/p_1, p_1\}) \quad \psi_n(\{p_2, p_1/p_3\}) \quad \psi_n(\{p_1\}) \\ &= a^3 b\, a^2 b\, a^1 b\, c\, a^1 b\, c\, d \qquad a^2 bc\, a^1 b\, a^3 b\, c\, d \qquad a^1 b\, c\, d \\ &= aaabaababc\, abcd \qquad aabc\, abaaabcd \qquad abcd\,.\end{aligned}$$

[1] (a) Montrer que l'ensemble des codages de séquences dans $(2^{N(P_n)})^*$ par ψ_n *pour tout* n, i.e. $\{\psi_n(\alpha) \mid n > 0, \alpha \in (2^{N(P_n)})^*\}$, est un langage reconnaissable sur $\{a, b, c, d\}^*$.

[5] (b) Montrer que

$$\mathscr{L}_0 = \{\psi_n(\alpha) \mid n > 0, \alpha \in (2^{N(P_n)})^*, \exists E \in 2^{N(P_n)}, p_1 \in E \text{ et } \alpha \to^* E\}$$

est un langage algébrique sur $\{a, b, c, d\}$.

[1] (c) En déduire le théorème 1.

[1] 2. Montrer que $\mathscr{L}_0$ est inhéremment ambigu. Est-il déterministe ?

3. Soit $\mathcal{M}$ une machine de Turing reconnaissant un langage $L(\mathcal{M})$ sur un alphabet Δ en temps $t(n) \geq n$ (resp. en espace $e(n) \geq n$). Soit $h : \Sigma^* \to \Delta^*$ un homomorphisme.

[1] Montrer qu'il existe une machine de Turing $\mathcal{M}_h$ qui reconnaît $h^{-1}(L)$ en temps $t(n)$ (resp. en espace $e(n)$). Qu'en déduire quant à la difficulté de reconnaître un langage algébrique à l'aide d'une machine de Turing ?