

Corrigé : Examen *Théorie des langages*, mai 2007

Documents autorisés : transparents du cours et notes de TD.

Le barème est indicatif. On attachera une grande importance à la clarté et à la concision des justifications.

Langages réguliers

(4+3+2 points)

Soit $E = a(b + ab)^* + b^*(a + bb)$.

1. Dans cet exercice on veut construire un automate non-déterministe *sans* transitions- ϵ qui accepte $\mathcal{L}(E)$, en appliquant la construction de Glushkov.

Rappel : la construction de Glushkov s'applique à l'expression renommée

$$E' = a_1(b_1 + a_2b_2)^* + b_3^*(a_3 + b_4b_5).$$

- (a) Déterminez l'ensemble $\text{first}(E')$ des lettres commençant au moins un mot de $\mathcal{L}(E')$, et l'ensemble $\text{last}(E')$ des lettres terminant au moins un mot de $\mathcal{L}(E')$. Justifiez votre réponse.

$$\text{first}(E') = \text{first}(a_1(b_1 + a_2b_2)^*) \cup \text{first}(b_3^*(a_3 + b_4b_5)) = \{a_1\} \cup \{b_3, a_3, b_4\}$$

$$\text{last}(E') = \text{last}(a_1(b_1 + a_2b_2)^*) \cup \text{last}(b_3^*(a_3 + b_4b_5)) = \{b_1, b_2, a_1\} \cup \{a_3, b_5\}$$

- (b) Déterminez l'ensemble $\text{follow}(E')$ des paires de lettres x, y telles que xy est facteur d'au moins un mot de $\mathcal{L}(E')$. Justifiez votre réponse.

$$\begin{aligned} \text{follow}(E') &= \text{follow}(a_1(b_1 + a_2b_2)^*) \cup \text{follow}(b_3^*(a_3 + b_4b_5)) \\ &= \{(a_1, b_1), (a_1, a_2)\} \cup \text{follow}((b_1 + a_2b_2)^*) \cup \text{follow}(b_3^*) \\ &\quad \cup \{(b_3, a_3), (b_3, b_4)\} \cup \text{follow}(a_3 + b_4b_5) \\ &= \{(a_1, b_1), (a_1, a_2)\} \cup \{(b_1, b_1), (b_1, a_2), (a_2, b_2), (b_2, b_1), (b_2, a_2)\} \\ &\quad \cup \{(b_3, b_3)\} \cup \{(b_3, a_3), (b_3, b_4)\} \cup \{(b_4, b_5)\} \end{aligned}$$

- (c) Construisez l'automate pour $\mathcal{L}(E')$ et ensuite celui pour $\mathcal{L}(E)$.

2. Proposez une expression rationnelle E_1 *sans-étoile* équivalente à E . Justifiez votre réponse.

Rappel : une expression rationnelle sans-étoile sur l'alphabet Σ est une expression construite à partir des lettres de Σ , en utilisant les opérations union, produit (concaténation) et complémentaire.

$$E_1 = a[(a + b)^* \setminus ((a + b)^*aa(a + b)^* + (a + b)^*a)] + [(a + b)^* \setminus (a + b)^*a(a + b)^*](a + bb)$$

justification : $(b + ab)^$ sont les mots qui ne contiennent pas de facteur aa et qui ne finissent pas par a*

3. Proposez une formule ϕ de la logique du premier ordre (FOL) qui décrit le langage $\mathcal{L}(E)$.

Vous pouvez utiliser les macros $\min(x)$ et $\max(x)$ définis en cours.

$$\begin{aligned}\phi &= \phi_1 \vee \phi_2 \\ \phi_1 &= \min(a) \wedge \max(b) \wedge \neg \exists x, y (a(x) \wedge a(y) \wedge y = x + 1) \\ \phi_2 &= (\forall x. \neg \max(x) \rightarrow b(x) \wedge \max(a)) \vee (\exists x, y. \min(x) \wedge \max(y) \wedge x \neq y \wedge \forall x. b(x))\end{aligned}$$

Automate d'arbres

(4 points)

Dans cet exercice on considère des arbres binaires sur l'alphabet $\Sigma = \Sigma_0 = \Sigma_2 = \{a, b\}$.

Soit $L \subseteq \mathcal{T}(\Sigma)$ l'ensemble des arbres tels que *chaque* chemin de la racine jusqu'à une feuille est étiqueté par un mot de $a^*(ba)^*b^*$. Construisez un automate d'arbre descendant *déterministe* \mathcal{A} qui reconnaît L .

On utilise un DFA $\mathcal{B} = \langle Q, \Sigma, \Delta, q_0, F \rangle$ qui reconnaît $a^*(ba)^*b^*$. Le DTA descendant qui reconnaît L est $\mathcal{A} = \langle Q, \Sigma, \delta, \{q_0\} \rangle$, avec $\delta_2(q, c) = (q', q')$ pour tous $q, q' \in Q$ et $c \in \Sigma$ tels que $\Delta(q, c) = q'$; $(q, c) \in \delta_0$ pour tous $q \in Q, c \in \Sigma$ tels que $\Delta(q, c) \in F$.

Langages hors-contexte

(2+2+2+2 points)

Soit $L = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b + |w|_c\}$.

1. Montrez que L n'est pas reconnaissable par application directe du lemme d'itération (ou calcul des quotients).

Soit $w = a^{N+1}b^Nc$, N constante du lemme. En écrivant $w = uvx$, $|uv| \leq N$, on a $uv = a^i$, $x = a^{N+1-i}b^Nc$. Mais $ux \notin L$.

2. Montrez que L n'est pas reconnaissable par application des propriétés de clôture des langages reconnaissables et du fait que le langage $\{a^n b^n \mid n \geq 0\}$ n'est pas reconnaissable. Justifiez votre réponse.

Soit $h : \{a, b, c\} \rightarrow \{a, b\}$ le morphisme défini par $h(a) = a, h(b) = h(c) = b$. On a $h(L) \cap a^*b^* = \{a^n b^n \mid n \geq 0\}$. Comme les réguliers sont fermés par morphismes et intersection, L régulier entraîne $\{a^n b^n \mid n \geq 0\}$ régulier, contradiction.

3. Montrez que chaque mot non-vide w de L peut s'écrire comme $w = \alpha u \beta v$, avec $u, v \in \{a, b, c\}^*$ des mots appartenant à L et α, β des lettres telles que $(\alpha, \beta) \in \{(a, b), (b, a), (a, c), (c, a)\}$. (Par exemple, si $w = baca$, alors $\alpha = b, \beta = a, u = \epsilon$ et $v = ca$. Si $w = ccabaaab$, alors $\alpha = c, \beta = a, u = caba$ et $v = ab$.)

Pour chaque préfixe x de w on définit $\Delta(x) = |x|_a - (|x|_b + |x|_c)$. Soit x le plus court préfixe de w tel que $\Delta(x) = 0$ ($x = w$ est possible!) Alors x peut s'écrire $x = \alpha u \beta$ avec $(\alpha, \beta) \in \{(a, b), (b, a), (a, c), (c, a)\}$, donc $u \in L$. Le suffixe v tel que $w = xv$ satisfait aussi $v \in L$.

4. Construisez une grammaire hors-contexte G telle que $\mathcal{L}(G) = L$. Justifiez votre réponse.

$S \rightarrow aSbS \mid bSaS \mid aScS \mid cSaS \mid \epsilon$

Expressions arithmétiques

(2+1+2+4 points)

Dans les expressions arithmétiques en notation polonaise on met le symbole de l'opération avant les opérands et on n'utilise pas de parenthèses. Par exemple, $a + b$ s'écrit $+ ab$, et $(a + b) \times (b + c)$ s'écrit $\times + ab + bc$.

1. Décrivez l'ensemble L des expressions en notation polonaise avec les opérations binaires $+$, \times et les constantes a, b, c par une grammaire hors-contexte G .

$$E \rightarrow +EE \mid *EE \mid a \mid b \mid c$$

2. Quel est l'arbre de dérivation de $\times + ab + bc$?

3. On considère le langage $\mathcal{T}(G)$ des arbres de dérivation de la grammaire G , associés aux mots de $\mathcal{L}(G)$.

- (a) Précisez l'alphabet Σ des étiquettes des arbres de $\mathcal{T}(G)$, ainsi que le(s) arité(s) de chaque symbole.

$$\Sigma_3 = \Sigma_1 = \{E\}, \Sigma_0 = \{a, b, c, +, *\}$$

- (b) Construisez un automate d'arbre montant \mathcal{A} qui reconnaît le langage $\mathcal{T}(G)$.

$$a \rightarrow q, b \rightarrow q, c \rightarrow q, + \rightarrow r, * \rightarrow r, E, r, q, q \rightarrow q$$

4. Construisez un automate à pile qui reconnaît le langage L .

$$\Gamma = \{+, *, E\}, Q = \{q, q_1, q_2\}$$

$$\delta(q, op, A, q, opA) \text{ pour } op \in \{+, *\}, A \in \Gamma \text{ (empiler } +, *)$$

$$\delta(q, v, A, q, Ev) \text{ pour } v \in \{a, b, c\}, A \in \Gamma \text{ (pour chaque opérand, empiler } E)$$

$$\delta(q, \epsilon, E, q_1, \epsilon), \delta(q_1, \epsilon, E, q_2, \epsilon), \delta(q_2, \epsilon, op, q, E) \text{ (à chaque moment, on peut tester si le mot de pile finit par } opEE \text{ et remplacer par } E, \text{ si oui)}$$

$$\delta(q_1, \epsilon, op, q, Eop) \text{ (test négatif, rétablir la pile)}$$