Langages Formels

Licence Informatique – ENS Cachan

Examen du 26 mai 2011
durée 3 heures

Document autorisé : polycopié du cours.
Toutes les réponses devront être correctement justifiées.
Le chiffre en regard d'une question est une indication sur sa difficulté ou sa longueur.

1 Langages linéaires et automates à un pic

Un *automate à un pic* est un automate à pile tel que dans tout calcul valide, la taille de la pile n'augmente plus une fois qu'elle a diminué. La taille de la pile peut donc augmenter (au sens large) pendant une première partie du calcul, puis elle ne fait que diminuer (au sens large).

Un *langage est à un pic* s'il peut être accepté *par pile vide* par un automate à un pic.

[3] **a)** Montrer que le langage $L = \{a^n b^n \mid n > 0\} \cup \{a^n b^{2n} \mid n > 0\}$ est un langage linéaire.
Montrer que L est un langage à un pic.
Montrer que le langage $K = \{ba^{i_1}ba^{i_2}b \cdots ba^{i_n}b \mid n > 0 \text{ et } \exists j, i_j \neq j\}$ est un langage à un pic.

[2] **b)** Montrer que tout langage linéaire est un langage à un pic.

[3] **c)** Soit $G = (\Sigma, V_1 \cup V_2, P_1 \cup P_2)$ une grammaire vérifiant :

$$\begin{aligned} &V_1 \cap V_2 = \emptyset \\ &P_1 \subseteq V_1 \times (V_1\Sigma^* \cup \Sigma^*) \\ &P_2 \subseteq V_2 \times (\Sigma^* V_2 V_1^* \cup \Sigma^*) \end{aligned}$$

Montrer que pour tout $x \in V$, $\mathcal{L}_G(x)$ est un langage linéaire.

Soit $\mathcal{A} = (Q, \Sigma, Z, T, (q_0, z_0))$ un automate à pile arbitraire.
Pour $p, q \in Q$ et $z \in Z$ on définit les langages

$$\begin{aligned} K_{zp,q} &= \{w \in \Sigma^* \mid \exists\ zp \xrightarrow{w} q \text{ calcul dans } \mathcal{A} \text{ avec une pile de taille toujours } \leq 1\}, \\ L_{zp,q} &= \{w \in \Sigma^* \mid \exists\ zp \xrightarrow{w} q \text{ calcul à un pic dans } \mathcal{A}\}. \end{aligned}$$

[2] **d)** Montrer que le langage $K_{zp,q}$ peut être engendré par une grammaire linéaire *droite*.

[2] **e)** Construire une grammaire vérifiant les conditions de la question **(c)** et qui engendre les langages $L_{zp,q}$ avec les variables de V_2 et les langages $K_{zp,q}$ avec les variables de V_1.
Montrer que tout langage à un pic est un langage linéaire.

2 Fonctions séquentielles et logique temporelle

On s'intéresse ici à une logique utilisant des modalités temporelles utiles pour spécifier les propriétés de certains systèmes. On fixe un ensemble fini AP de propositions atomiques. Un état du système est abstrait en l'ensemble $s \subseteq \mathrm{AP}$ des propositions atomiques qui sont vraies dans cet état. Une exécution du système sera donc décrite par un mot $\sigma = s_1 s_2 \cdots \in \Sigma^*$ où $\Sigma = 2^{\mathrm{AP}}$. On note $|\sigma|$ la longueur du mot σ.

Les formules de la logique temporelle $\mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{S})$ sont obtenues à partir des propositions atomiques en utilisant les connecteurs booléens, la modalité unaire Y (Yesterday) et la modalité binaire S (Since) :

$$\alpha ::= p \in \mathrm{AP} \mid \neg\alpha \mid \alpha \vee \alpha \mid \mathsf{Y}\,\alpha \mid \alpha\,\mathsf{S}\,\alpha$$

Soit $\alpha \in \mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{S})$ une formule, $\sigma = s_1 s_2 \cdots \in \Sigma^*$ un mot et i une position vérifiant $1 \leq i \leq |\sigma|$. On note $\sigma, i \models \alpha$ lorsque le mot σ à la position i satisfait la formule α. Cette sémantique se définit inductivement comme suit :

$$\begin{array}{lll}
\sigma, i \models p & \text{si} & p \in s_i \\
\sigma, i \models \neg\alpha & \text{si} & \sigma, i \not\models \alpha \\
\sigma, i \models \alpha \vee \beta & \text{si} & \sigma, i \models \alpha \text{ ou } \sigma, i \models \beta \\
\sigma, i \models \mathsf{Y}\,\alpha & \text{si} & i > 1 \text{ et } \sigma, i-1 \models \alpha \\
\sigma, i \models \alpha\,\mathsf{S}\,\beta & \text{si} & \exists j \in \{1, \ldots, i\},\ \sigma, j \models \beta \text{ et } \forall k \in \{j+1, \ldots, i\},\ \sigma, k \models \alpha
\end{array}$$

Par exemple, soit $\mathrm{AP} = \{p, q\}$ et soit $\sigma = \{p\}\{p,q\}\{q\}\{q\}\{p,q\}\{\}\{\}\{p\}\{q\} \in \Sigma^*$. On a $\sigma, 1 \models p \wedge \neg q$, $\sigma, 3 \models \mathsf{Y}\,p$, $\sigma, 4 \models q\,\mathsf{S}\,p$ et $\sigma, 8 \models \mathsf{Y}(\neg p\,\mathsf{S}\,q)$.

À chaque formule $\alpha \in \mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{S})$ on associe une fonction $[\![\alpha]\!] : \Sigma^* \to \{0,1\}^*$ définie pour $\sigma \in \Sigma^*$ par $[\![\alpha]\!](\sigma) = c_1 \cdots c_{|\sigma|}$ avec pour tout $1 \leq i \leq |\sigma|$, $c_i = 1$ si et seulement si $\sigma, i \models \alpha$.

En reprenant le mot σ de l'exemple ci-dessus on a $[\![q]\!](\sigma) = 011110001$, $[\![\mathsf{Y}\,p]\!](\sigma) = 011001001$ et $[\![q\,\mathsf{S}\,p]\!](\sigma) = 111110011$.

[3] **a)** Soit $\mathrm{AP} = \{p, q\}$. Montrer que les fonctions suivantes sont séquentielles pures. Donner à chaque fois l'automate séquentiel pur (si possible minimal) qui calcule la fonction.

1. $[\![p]\!]$
2. $[\![p \to \mathsf{Y}\,q]\!]$
3. $[\![p\,\mathsf{S}\,q]\!]$
4. $p \to \mathsf{Y}((\neg p)\,\mathsf{S}\,q)$

À chaque opérateur op d'arité k on associe une fonction $f_{\mathrm{op}} : (\{0,1\}^k)^* \to \{0,1\}^*$ qui décrit sa sémantique. Par exemple,

- $f_\neg(a_1 \cdots a_n) = c_1 \cdots c_n$ avec pour tout $1 \leq i \leq n$, $c_i = 1 - a_1$,
- $f_{\mathsf{Y}}(a_1 \cdots a_n) = c_1 \cdots c_n$ avec $c_1 = 0$ et pour tout $1 < i \leq n$, $c_i = a_{i-1}$
- $f_{\mathsf{S}}((a_1, b_1) \cdots (a_n, b_n)) = c_1 \cdots c_n$ avec pour tout $1 \leq i \leq n$,
  $c_i = 1$ si et seulement si $\exists j \in \{1, \ldots, i\}$ tel que $b_j = 1$ et $\forall k \in \{j+1, \ldots, i\}$, $a_k = 1$.

[2] **b)** Montrer que les fonctions $f_\neg$, $f_\vee$, f_{Y} et f_{S} sont séquentielles pures. En déduire que pour chaque formule $\alpha \in \mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{S})$, la fonction $[\![\alpha]\!]$ est séquentielle pure.

Le problème SAT pour les formules LTL est le suivant :

Donnée Une formule $\alpha \in \mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{S})$,

Question Existe-t-il un mot $\sigma \in \Sigma^+$ et une position $1 \leq\ i \leq |\sigma|$ tels que $\sigma, i \models \alpha$?

Le problème Rec-MC pour les formules LTL est le suivant :

Donnée Une formule $\alpha \in \mathrm{LTL}(\mathrm{AP}, \mathsf{Y}, \mathsf{S})$ et un langage reconnaissable $M \subseteq \Sigma^+$,

Question A-t-on $\sigma, |\sigma| \models \alpha$ pour tout $\sigma \in M$?

[2] **c)** Montrer que les problèmes SAT et Rec-MC sont décidables.
Facultatif : Montrer que ces problèmes sont dans PSPACE.

On introduit maintenant deux modalités "futur" : X (neXt) et U (Until) dont la sémantique est définie (en utilisant les notations précédentes) par :

$$\begin{array}{lll} \sigma, i \models \mathsf{X}\,\alpha & \text{si} & i < |\sigma| \text{ et } \sigma, i+1 \models \alpha \\ \sigma, i \models \alpha\,\mathsf{U}\,\beta & \text{si} & \exists\, j \in \{i, \ldots, |\sigma|\},\ \sigma, j \models \beta \text{ et } \forall k \in \{i, \ldots, j-1\},\ \sigma, k \models \alpha \end{array}$$

[2] **d)** Montrer que la fonction f_{X} est séquentielle. Est-elle séquentielle pure ? Justifier.
La fonction f_{U} est-elle séquentielle ? Justifier.

La restriction aux entrées d'un automate séquentiel doit être déterministe. Pour un automate fonctionnel non ambigu $\mathcal{A} = (Q, A, I, T, F, B, \varphi)$, on assouplit cette condition en demandant seulement que la restriction aux entrées soit non ambigu : Q est l'ensemble fini d'états, $I, F \subseteq Q$ définissent les états initiaux et finaux, $T \subseteq Q \times A \times Q$ est la relation de transitions et $\varphi : T \to B^*$ la fonction de sortie. Chaque mot $u \in A^*$ doit admettre au plus un calcul acceptant dans l'automate réduit aux entrées (Q, A, I, T, F) et la fonction calculée $[\![\mathcal{A}]\!](u)$ est la concaténation des sorties de ce calcul.

[2] **e)** Montrer que la fonction f_{U} peut être calculée par un automate fonctionnel non ambigu.