Examen de démonstration automatique 2001

Documents autorisés (en particulier le poly).

I. Automates alternants généralisés

On considère une signature du premier ordre consistant en un ensemble fini arbitraire de symboles de fonction f, g, ..., et un ensemble fini de symboles de prédicats *unaires*.

On appelle *clause automatique* toute clause de la forme :

$$\neg P_1(t_1) \vee \ldots \vee \neg P_m(t_m) \vee P_{m+1}(t_{m+1}) \vee \ldots \vee P_n(t_n)$$

où $0 \leq m \leq n$, les t_i ($1 \leq i \leq n$) sont des termes tels que :

1. si tous les t_i sont des variables, alors ils sont tous la même variable x;
2. sinon, tous les t_i non variables sont identiques, de la forme $f(x_1, \ldots, x_p)$, où $\{x_1, \ldots, x_p\}$ contient toutes les variables libres de la clause.

Dans le premier cas, on dit que la clause est un ϵ-*bloc*.

L'intuition est la suivante. Chaque prédicat P correspond à un état d'un automate d'arbres classique. Une transition $f(P_1, \ldots, P_k) \rightarrow P$ (exprimant que si t_1 est reconnu en P_1, ..., t_k est reconnu en P_k, alors $f(t_1, \ldots, t_k)$ est reconnu en P) est codée par la clause:

$$\neg P_1(x_1) \vee \ldots \vee \neg P_n(x_n) \vee P(f(x_1, \ldots, x_n))$$

On peut coder les ϵ-transitions $P \xrightarrow{\epsilon} Q$ ("si t est reconnu en P, alors t est reconnu en Q") par:

$$\neg P(x) \vee Q(x)$$

Dans les automates usuels, un terme t est reconnu en P si et seulement s'il *existe* une transition aboutissant à P telle qu'un certain nombre de conditions soient vérifiées : la sémantique de la reconnaissance est une disjonction. Ici, on peut aussi coder les conjonctions $P_1 \wedge \ldots \wedge P_n \rightarrow P$ ("si t est reconnu en P_1 et ... et en P_n, alors il est reconnu en P"):

$$\neg P_1(x) \vee \ldots \vee \neg P_n(x) \vee P(x)$$

On peut aussi coder les négations ("P reconnaît exactement les termes non reconnus en Q") :

$$\neg P(x) \vee \neg Q(x)$$
$$P(x) \vee Q(x)$$

ainsi que les tests d'égalité entre frères ("si les t_i sont reconnus en P_i et $t_1 = t_2$ alors $f(t_1, \ldots, t_n)$ est reconnu en P"):

$$\neg P_1(x_1) \vee \neg P_2(x_1) \vee \neg P_3(x_3) \vee \ldots \vee \neg P_n(x_n) \vee P(f(x_1, x_1, x_3, \ldots, x_n))$$

1. Montrer qu'il existe un ordre strict $>$ sur les atomes tel que (i) $>$ est stable, autrement dit pour toute substitution σ, si $A > B$ alors $A\sigma \gg B\sigma$; et (ii) pour tous P, Q, f, x_1, ..., x_n, on a $P(f(x_1, \ldots, x_n)) > Q(x_i)$, $1 \leq i \leq n$. Dans la suite, on considérera la résolution ordonnée au-dessus de $>$.

2. Montrer que tout facteur d'une clause automatique est une clause automatique, dès que $>$ vérifie la condition (ii).

3. Montrer que tout résolvant binaire ordonné de deux clauses automatiques est une clause automatique ou une disjonction d'ϵ-blocs.

4. Montrer qu'à renommage près, il n'existe qu'un nombre fini de clauses automatiques. On estimera un majorant de ce nombre en fonction du nombre n de symboles de prédicats, m de symboles de fonctions, et l'arité maximum k des symboles de fonctions. (Attention ! Les symboles de fonctions sont appliqués à des variables qui ne sont pas nécessairement disjointes deux à deux.)

5. En déduire que, si tous les symboles de fonction de la signature sont d'arité au plus un, alors la satisfiabilité de tout ensemble de clauses automatiques est décidable.

6. Que peut-on dire lorsqu'il existe des symboles de fonction d'arité ≥ 2 ? (Indication : il s'agit de traiter le cas problématique des disjonctions non-triviales d'ϵ-blocs; on pourra penser à importer une règle de ramification des méthodes de tableaux.)

II. À propos de S2S

On définit en $\mathcal{A}_{S2S}$ une abréviation pour dire que des ensembles sont disjoints entre eux :

$$\textit{disjoint}(X_1, \ldots, X_n) \quad :\Leftrightarrow \quad \bigwedge_{1 \leq i < j \leq n} (\neg \exists y (y \in X_i \wedge y \in X_j))$$

1. Pour quelles valeurs de x est-ce que la formule suivante est vraie dans la structure $\mathcal{A}_{S2S}$?

$$\begin{aligned} \exists X_0, X_1, X_\bot \Big(\quad & \textit{disjoint}(X_0, X_1, X_\bot)) \\ & \wedge \epsilon \in X_0 \\ & \wedge x \in X_0 \\ & \wedge \forall y (y \in X_0 \rightarrow y0 \in X_1 \wedge y1 \in X_\bot) \\ & \wedge \forall y (y \in X_1 \rightarrow y0 \in X_0 \wedge y1 \in X_\bot) \\ & \wedge \forall y (y \in X_\bot \rightarrow y0 \in X_\bot \wedge y1 \in X_\bot) \quad \Big) \end{aligned}$$

2. Montrer le théorème suivant : pour tout langage régulier L sur l'alphabet $\{0, 1\}$ il existe une formule $F_L(x)$ du langage de la structure $\mathcal{A}_{S2S}$ telle que pour tout mot $w \in \{0, 1\}^*$:

$$\mathcal{A}_{S2S}, [x \mapsto w] \models F_L(x) \qquad \Leftrightarrow \qquad w \in L$$

3. On considère la structure $\mathcal{A}'_{S2S}$ qui a le même langage que $\mathcal{A}_{S2S}$. La seule différence dans l'interprétation est que le symbole de fonction 0 est interprété par $0(w) := 0w$, tandis que le symbole de fonction 1 garde son ancienne interprétation $1(w) := w1$. Montrer que la théorie de $\mathcal{A}'_{S2S}$ est indécidable.

   Indication: Réduire le problème indécidable suivant :

   Étant donnée une machine de Turing sur l'alphabet $\{a, b\}$ à ruban borné gauche et qui ne s'arrête jamais, et un état q, est-ce que l'état q est atteignable à partir de l'état initial et ruban infini (à droite) rempli avec 0 ?

   L'idée est de représenter un calcul d'une telle machine de Turing par une grille où chaque configuration est représentée par la partie du ruban à gauche de la tête, puis l'état, puis la partie du ruban à droite de la tête. Les cases de la grille sont donc remplies soit par a, b, ou par un symbole qui représente un état de la machine. Noter qu'une coordonnée (n, m) de la grille peut être représentée par le mot $0^n 1^m$. On utilisera des variables d'ensembles pour dénoter les cases qui sont remplies par a, b, et les états q_0, q_1, etc.