Examen de démonstration automatique 2001

Correction.

Partie “Théorie des structures de données concrètes”

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 $\mathcal{A}_{S2S}$?

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

Les trois dernières lignes font penser à un automate à trois états X_0, X_1 et $X_\perp$. Les X_0, X_1, $X_\perp$ minimaux qui vérifient toutes les lignes sauf la première et la troisième sont les langages $(00)^*$ (une suite d'un nombre pair de zéros), $(00)^*0$ (un nombre impair de zéros), 0^*1A^* respectivement, où $A = \{0, 1\}$ (tout mot contenant au moins un 1).

Il est alors clair que tous les x dans $(00)^*$ vérifient la formule. (En particulier, $(00)^*$, $(00)^*0$, 0^*1A^* sont bien disjoints.) Réciproquement, notons que l'automate est déterministe et complet, autrement dit les langages $(00)^*$, $(00)^*0$, et 0^*1A^* sont non seulement disjoints mais encore leur union est exactement A^*. En particulier, si x n'est pas dans $(00)^*$ alors il est soit dans $(00)^*0$ soit dans 0^*1A^*. Dans le deux cas, montrons que x ne vérifie pas la formule. Supposons par contradiction que la formule est vraie, en particulier on a trouvé X_0, X_1 et $X_\perp$ et $x \in X_0$:

- *si $x \in (00)^*0$ alors comme $x \in X_0$, $x0 \in X_1$; mais $x0 \in (00)^*00 \subseteq X_0$, donc X_0 et X_1 ne seraient pas disjoints;*
- *si $x \in 0^*1A^*$ alors comme $x \in X_0$, $x0 \in X_1$; mais $x1 \in 0^*1A^*0 \subseteq X_\perp$, contredisant le fait que X_1 et $X_\perp$ sont disjoints.*

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 $w \in \{0, 1\}^*$:

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

Soit $\mathcal{A}$ un automate déterministe complet (pour toute lettre et tout état, il existe exactement une transition sortant de l'état étiquetée par la lettre) reconnaissant L. Pour chaque état q, créer une *variable d'ensemble X_q. Soit Q l'ensemble des états, q_0 l'état initial, et F l'ensemble des états finaux de $\mathcal{A}$. On crée alors la formule :*

$$\exists (X_q)_{q\in Q}\Big(\begin{array}{l} \textit{disjoint}((X_q)_{q\in Q}) \\ \wedge \epsilon \in X_{q_0} \\ \wedge \bigvee_{q\in F} x \in X_q \\ \wedge \bigwedge_{q \xrightarrow{a} q'} \forall y (y \in X_q \to ya \in X_{q'}) \end{array} \Big)$$

La démonstration de correction est essentiellement la même que dans la question précédente : les ensembles X_q minimaux vérifiant la deuxième et la quatrième ligne sont exactement les langages reconnus en q dans $\mathcal{A}$. Comme $\mathcal{A}$ est déterministe et complet, ces ensembles sont de plus disjoints deux à deux et leur union vaut A^*. On peut alors rejouer l'argument de la question précédente.

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.

   Par hypothèse, la grille est $\mathbb{N}^2$. À létape i de l'exécution de la machine de Turing, on représente sa bande et son état par un mot de A^*QA^ω, où $A = \{a, b\}$ et Q est l'alphabet des états de la machine (la tête étant positionnée sur la lettre à droite de l'état). La grille mentionnée est l'empilement de ces bandes les unes sur les autres, verticalement. On a besoin pour la décrire de dire quelle est la configuration initiale (un mot $qa_1 \dots a_k 0^*$): $\epsilon \in X_q$ et $1 \in X_{a_1}$ et $11 \in X_{a_2}$ et … $1^k \in X_{a_k}$ et $\exists X \cdot 1^{k+1} \in X \wedge (\forall y \in X \cdot y1 \in X) \wedge \forall y \in X \cdot y \in X_0$. Ensuite on a besoin d'exprimer la bande $i+1$ en fonction de la bande i; c'est facile parce que les opérations $w \mapsto 0w$ et $w \mapsto w1$ nous permettent de monter d'un cran et de se déplacer d'un cran à droite respectivement. Par exemple, pour représenter les mouvements de la machine de Turing à l'état q et regardant la lettre a consistant en réécrire a en b, passer à l'état q' et se déplacer à droite, on veut transformer tout mot à la hauteur i de la forme $wqaw'$ en le mot $wbq'w'$. On écrira donc pour ces mouvements :

   - *pour dire que w ne bouge pas : pour toute position y telle qu'il existe la lettre q à droite (aussi loin qu'on veut), alors la lettre une case plus haut est la même qu'en y. Pour dire qu'on trouve q à droite, on utilise la formule $\textit{droite}_q(y) = \forall X \cdot y1 \in X \wedge (\forall y' \cdot y' \in X \to y'1 \in X) \to \exists y' \in X \cdot X_q(y)$. On écrit alors la formule $\forall y \cdot \textit{droite}_q(y) \to (X_a(y) \Leftrightarrow X_a(0y)) \wedge (X_b(y) \Leftrightarrow X_b(0y))$.*
   - *Pour dire que qaw' se transforme en $bq'w'$: $\forall y \cdot X_q(y) \wedge X_a(y1) \to X_b(0y) \wedge X_{q'}(0y1) \wedge \forall X \cdot (y11 \in X \wedge (\forall y' \in X \cdot y'1 \in X) \to \forall y' \in X \cdot (X_a(y') \Leftrightarrow X_a(0y')) \wedge (X_b(y') \Leftrightarrow X_b(0y')))$.*

   On écrit des formules similaires pour les paires (q, a) telles que la machine de Turing doit se déplacer à gauche, c'est-à-dire pour les transformations de mots $wcqaw'$ en $wq'cbw'$. Ceci nécessite de remplacer $\textit{droite}_q(y)$ par $\textit{droite}_q(y1)$ pour dire que w ne bouge pas, et il faut une conjonction de *plus pour dire que* $cqaw'$ *se transforme en* $q'cbw'$*. Il n'y a plus qu'à prendre la conjonction de toutes ces formules, ainsi qu'avec une formule imposant que tous les* X_q*,* X_a*,* X_b *sont disjoints, et finalement de la formule exprimant que l'état final souhaité* q *n'est pas atteignable, soit* $\forall y \cdot \neg X_q(y)$*. La formule résultante est fausse si et seulement si* q *est atteignable.*