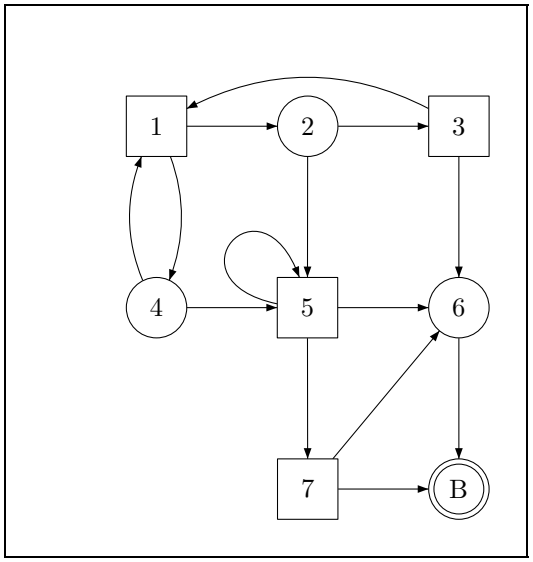
Automates avancés

Master 1 II et MI

Examen du 2 mai 2007 - petit corrigé.

1 Applications des cours

1.1 Jeux



On considère le jeu défini par le diagramme ci-dessus. Les états ronds appartiennent au joueur E, les états carrés au joueur A. L'objectif de E consiste à atteindre l'état B.

1. Pour trouver l'ensemble W_E de tous les états gagnants pour E on applique l'algo de cours.

$$\begin{aligned} W_0 &= \{B\} \\ W_1 &= \{B, 6\} \\ W_2 &= \{B, 6, 7\} \\ W_3 &= \{B, 6, 7\} = W_2, \end{aligned}$$

   d'où $W_E = W_2 = \{B, 6, 7\}$

2. *Décrire une stratégie gagnante pour E à partir de tous les états W_E.* En état B on a déjà gagné. En état 7 c'est A qui décide. Il reste à définir la stratégie pour l'état 6. On applique la méthode établie en cours : avancer de W_i vers $W_{<i}$ (dans notre cas de W_1 vers W_0

   Stratégie pour E. De 6 aller à B

3. *Décrire une stratégie gagnante pour A à partir de tous les autres états.* Il suffit de spécifier les actions de A pour les états $1, 3, 5$. On applique la méthode établie en cours : rester hors W_E.

   Stratégie pour A. De 1 aller à 2 ou 4. De 3 aller vers 1. De 5 rester sur place.

1.2 Mots infinis

On considère le langage L de tous les mots infinis sur $\{a, b\}$ qui contiennent un nombre infini de b sur des positions impaires.

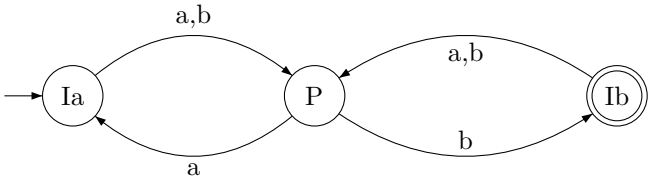
On suppose que le mot commence par la position 0.

1. Une expression ω-régulière définissant L

$$\big((a+b)((a+b)^2)^*b\big)^\omega$$

   Les b sont ici intercalés par des quantités impaires de caractères arbitraires.

2. *Trouver un automate de Büchi reconnaissant L.*



   Chaque fois quand il y a un b en position impaire on visite l'état accepteur.

3. *Trouver une formule MSO définissant L.* L'idée serait d'introduire une variable X de second ordre qui représente l'ensemble de positions impaires, et de dire qu'il y a des b dans X dans une infinité de positions.

$$\exists X\,(\neg X(0) \wedge (\forall i\,(X(i) \Leftrightarrow \neg X(S(i)))) \wedge (\forall m \exists n\,(n > m \wedge X(n) \wedge b(n))))$$

2 Des petits raisonnement : Mots périodiques et leurs applications

Pour donner des exemples des langages non-ω-réguliers de mots infinis on peut utiliser des lemmes de pompage. Dans cette exercice on trouve une méthode similaire mais plus simple. N'oubliez pas de justifier vos assertions.

1. *Démontrer que chaque langage ω-régulier non-vide contient un mot ultimement périodique (c-à-d de la forme uv^ω).*

   Soit L un langage ω-régulier non-vide accepté par un automate de Büchi $\mathcal{A}$. Comme L est non-vide, il contient un mot infini x. Un calcul accepteur de $\mathcal{A}$ sur x doit exister. Ce calcul commence dans l'état initial i de $\mathcal{A}$ et visite infiniment souvent l'ensemble F de ses états finals. Comme F est fini, ce calcul doit visiter un certain $f \in F$ au moins deux fois. On a donc vu que le début de calcul sur x a nécessairement la forme

$$i \xrightarrow{u} f \xrightarrow{v} f \rightarrow \cdots$$

   avec u, v deux sous-mots finis de x. En particulier on déduit que $i \xrightarrow{u} f$ and $f \xrightarrow{v} f$ pour i initial et f final.

   Maintenant on construit un nouveau mot infini $y = uv^\omega$. Par construction il est ultimement périodique. Il possède un calcul accepteur suivant :

$$i \xrightarrow{u} f \xrightarrow{v} f \xrightarrow{v} f \xrightarrow{v} f \xrightarrow{v} f \xrightarrow{v} f \xrightarrow{v} f \xrightarrow{v} f \xrightarrow{v} f \xrightarrow{v} \cdots$$

   On déduit que y est accepté par $\mathcal{A}$ et appartient donc à L.

2. **Lemme 1** *Soit $\mathcal{A}$ un automate de Büchi avec n états acceptant au moins un mot infini. Alors il existe deux mots finis u et v, de taille inférieure à $n+1$, tels que uv^ω est accepté par $\mathcal{A}$.*

Dans le point précédents on a trouvé deux mots finis u, v tels que $i \xrightarrow{u} f$ and $f \xrightarrow{v} f$ pour i initial et f final. Il existe donc un chemin de i à f. Par conséquent, il existe un chemin simple (qui ne passe jamais par le même état). Un tel chemin contient au plus n transitions. Soit u' le mot qui étiquette ce chemin. On a

$$|u'| \leq n; \quad i \xrightarrow{u'} f$$

De la même manière on trouve une boucle simple de f vers f, étiquetée par un mot v' tel que

$$|v'| \leq n; \quad f \xrightarrow{v'} f$$

Le même raisonnement que dans le point précédent montre que le mot infini $y' = u'v'^{\omega}$ est accepté par $\mathcal{A}$.

3. *Donner un exemple de langage ω-régulier de mots infinis, qui ne peut être reconnu par aucun automate avec moins de 100 états.*
   Exemple : $L = (ba^{100})^{\omega}$. Ce langage est bien sûr ω-régulier et contient un seul mot infini $z = (ba^{100})^{\omega}$.
   Preuve : Supposons que L est accepté par un automate avec < 100 états. Par le lemme précédent L contient un mot de la forme uv^{ω} avec $|u|, |v| < 100$. Or z est le seul mot dans L. Par conséquent il faut que $z = uv^{\omega}$
   On déduit donc que le mot infini $z = (ba^{100})^{\omega}$ admet une représentation $z = uv^{\omega}$ avec $|u|, |v| < 100$. C'est impossible, parce que la plus courte période de z est 101. **Comment le démontrer rigoureusement ?**
4. *Donner un exemple de langage non-ω-régulier de mots infinis* **Exemple :** L contient un seul mot : la séquence des décimales de π : 314159265359...
   Preuve : L est non-vide et son unique élément n'est pas ultimement périodique. Or si L était ω-régulier, il devrait contenir un mot ultimement périodique. Donc L n'est pas ω-régulier.

3 Une théorie décidable d'après Boigelot, Rassart et Wolper

L'objectif de cet exercice consiste à démontrer la décidabilité d'une théorie et trouver un algorithme de décision (un tel algorithme est appliqué pour la vérification de programmes). Il s'agit d'une théorie $\mathcal{T}$ de premier ordre avec la signature suivante :

$$\mathcal{S} = (0, 1, \leq, +, Z).$$

On l'interprète sur le domaine de réels $\mathbb{R}$. L' interprétation de $0, 1, \leq, +$ est usuelle. Le prédicat $Z(x)$ est vrai si et seulement si x est un entier. On souhaite pour chaque formule close décider si elle est vraie dans cette interprétation.

1. *Décrire en termes usuels l'ensemble de $x \in \mathbb{R}$ satisfaisant la formule $exo(x)$ ci-dessous :*

$$exo(x) \;=\; \exists y \exists z \, (y + y + y = 1 \wedge Z(z) \wedge x = y + z)$$

   Les x avec la partie fractionnelle $1/3$.
2. *Proposer une méthode pour représenter un nombre réel par un mot infini.* On le représente en système décimal. Le mot commence par un signe, quelques chiffres pour la partie entière, virgule, et une infinité de chiffres pour la partie fractionnelle. L'alphabet utile serait $\Sigma = \{01 \ldots 9 + -, \}$. Par exemple $+7,66666666666666 \ldots$ représente $7\frac{2}{3}$
   Un vecteur réel de dimension k par un mot infini. Pour un vecteur $(x_1, \ldots, x_k)$ on écrit les mots pour x_1, pour x_2, pour x_k un au-dessus de l'autre, en alignant les signes et les virgules.

Par exemple, pour $(-25,5;7\frac{2}{3})$ on écrit

-	2	5	,	5	0	0	0	0	0	0	0	0	...
+	0	7	,	6	6	6	6	6	6	6	6	6	...

Une telle table est en fait un mot infini sur l'alphabet Σ^k, où chaque colonne correspond à une lettre.

Un ensemble E de vecteurs de dimension k par un langage de mots infinis. On prend le langage $L(E)$ de tous les mots sur l'alphabet Σ^k qui correspondent aux vecteurs de E.

3. *Associer un langage $L(f)$ de mots infinis à chaque formule dans la signature $\mathcal{S}$.* Une formule f avec k variables libres définit un ensemble E de vecteurs de dimension k. Dans la question précédente on y a associé un langage L de mots infinis. On appelera ce langage $L(f)$. **Petit problème : comment faire pour une formule close ?**
4. *Montrer que le langage $L(exo)$ associé à la formule $exo(x)$ introduite au début de ce problème est ω-régulier (en exhibant un automate de Büchi ou une expression ω-régulière).*
   $L(exo) = +(0..9)^*, 3^\omega \cup -(0..9)^*, 6^\omega$
5. *Donner un plan de preuve par induction structurelle du lemme principal suivant :*

   Lemme 2 *Pour toute formule f le langage $L(f)$ est ω-régulier.*

   On passe tout d'abord à une théorie $\mathcal{T}'$ de premier ordre avec la signature suivante :

   $$\mathcal{S}' = (0, 1, \leq, P, Z).$$

   On a remplacé la fonction + par un prédicat $P(x, y, z)$ avec le sens "$x+y = z$". On remarque que la théorie $\mathcal{T}$ se réduit à $\mathcal{T}'$.

   L'avantage de cette nouvelle théorie est qu'elle possède trois types de termes seulement : 0, 1 et des variables.

   On va donc démontrer le lemme pour chaque formule f en signature S', en utilisant l'induction structurelle sur la construction de f.

   Il faut démontrer :
   - **Le cas de base**. $L(f)$ est ω-régulier pour les formules atomiques. Il y a les cas suivants à traiter.
     - $Z(0), Z(1), Z(x)$;
     - les formules de la forme $s \leq t$ où s, t sont des termes (il suffit de considérer 16 cas : $s, t \in \{0, 1, x, y\}$). Les seuls cas non-triviaux sont $0 \leq x; x \leq 0; 1 \leq x; x \leq 1; x \leq y$.
     - et encore toutes les formules possible de la forme $P(s, t, r)$ où s, t, r sont des termes (il suffit de considérer un nombre fini de cas : $s, t, r \in \{0, 1, x, y, z\}$).
   - **Le cas inductif**. Si $L(f)$ et $L(g)$ sont ω-réguliers, alors $L(\neg f)$, $L(f \vee g)$, $L(\exists x f)$ sont aussi ω-réguliers.
6. *Démontrer le cas de base.* Il faut exhiber des automates (ou des expressions régulières pour les formules énumérées précédemment. On le fera au tableau pour $Z(x)$, $x \leq y$ et $P(x, y, z)$. Les autres cas sont plus simples et similaires.
7. *Démontrer le cas inductif* Supposons que $L(f)$ et $L(g)$ sont ω-réguliers.
   - $L(\neg f) = overline L(f)$, il est ω-régulier en tant que complément d'un langage ω-régulier (résultat énoncé sans preuve en cours.)**Problème : cette construction est un peu fausse, comme elle donne un langage qui contient des mots mal formés, tels que $+9, -899, , , , + + +$. Corrigez la preuve.**
   - $L(f \vee g) = L(f) \cup L(g)$, il est ω-régulier en tant qu'union de deux langages ω-réguliers. **Problème : ce raisonnement marche si f et g ont les mêmes variables libres. Que faire dans le cas général.**
   - $L(\exists x f)$ est une image homomorphe de $L(x)$ **Pourquoi ? Décrire le morphisme utilisé.** Donc, ce langage est aussi ω-régulier. On a utilisé un résultat suivant : une image homomorphe d'un langage ω-régulier est aussi ω-régulière. textbfdémontrer cette propriété pour les homomorphismes remplaçant une lettre par une lettre

8. *Déduire la décidabilité de la théorie $\mathcal{T}$ et expliquer l' algorithme de décision.*

   L'algorithme est le suivant : pour une formule close f donnée construire , en appliquent la méthode ci-dessus un automate de Büchi (A) pour $L(f)$. Tester en appliquant un algo connu si le langage de (A) est vide. S'il est vide f est fausse, sinon elle est vraie

9. *Vous avez sans doute oublié que certains nombres réels admettent plus d'un développement en fraction décimale/ou binaire. Par exemple*

$$\begin{aligned} 2,5 &= 2,49999999999999999\ldots \quad \textit{base 10} \\ 10,1 &= 10,0111111111111111111\ldots \quad \textit{base 2} \end{aligned}$$

   Comment corriger la preuve pour tenir compte de ce phénomène désagréable ?

   – Une solution possible : **interdire** les développements avec 999 : on définit le langage de tous les mots "jolis", c'est-a-dire qui correspondent à un vecteur de réels sans une infinité de 9 à la fin. Pour la dimension 1 ça s'écrit

$$J = (+ \vee -)(0..9)^*, ((0..9)^*(0..8))^\omega$$

     Pour une dimension arbitraire c'est un peu plus difficile, mais J est toujours ω-régulier. Ensuite, à chaque étape de la construction de l'automate pour $L(f)$ on intersecte avec J.

   – Une autre possibilité : **autoriser** les séquences infinies de 9. Il faudra corriger légèrement l'automate pour $Z(x)$ qui doit accepter, par exemple $+7, 9^\omega$ et l'automate pour $x \leq y$, qui doit accepter $x = 2, 39999\ldots$ et $y = 2, 40000\ldots$. **Comment ?**