Université de Montpellier
-
Faculté des Sciences

Examen d'Intelligence Artificielle (HMIN107)
Master informatique

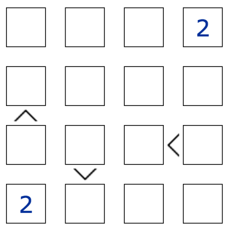
Session : 1	Durée de l'épreuve : 2 heures
Date : Janvier 2017	Documents autorisés : aucun

1 Problème de satisfaction de contraintes

Le jeu du Futoshiki consiste à compléter de nombres une grille n x n partiellement remplie de telle manière que :

- chaque ligne et chaque colonne possède les nombres de 1 à n ;
- les signes $<$ et $>$ imposés entre 2 cases soient respectés.

Modélisez comme un problème de satisfaction de contraintes *la grille de Futoshiki suivante* :



1. Décrivez avec précision la modélisation proposée : variables (nombre et leurs noms), domaines associés (taille et contenu), contraintes (cardinalité, variables en jeu et tuples). Si vous utilisez des contraintes en intension précisez leur sémantique et donnez quelques exemples des tuples qu'elles acceptent.
2. Précisez la taille de l'arbre de l'espace de recherche à explorer (expliquez).

2 Questions de cours

Question 2.1 Décrivez en quelques lignes le principe du Forward Checking par différence avec le Backtrack de base.

Question 2.2 On considère une base de faits $\mathcal{F}$ et une base de règles (positives et conjonctives) $\mathcal{R}$ en *logique des propositions*. Etant donné un symbole propositionnel (ou : atome, variable propositionnelle) a, on veut déterminer si $\mathcal{F}, \mathcal{R} \models a$ à l'aide d'un solveur SAT. Que doit-on fournir en entrée au solveur et comment interpréter la réponse du solveur ?

Question 2.3 On rappelle qu'un littéral est un atome (littéral positif) ou la négation d'un atome (littéral négatif). On considère la négation logique classique. Le chaînage avant sur les règles positives et conjonctives en logique des propositions n'est plus complet lorsque les règles et les faits peuvent comporter des littéraux négatifs. Prouvez-le.

3 Chaînage avant avec des règles positives d'ordre 1

On considère la base de connaissances $\mathcal{K} = (\mathcal{F}, \mathcal{R})$ ci-dessous.

- $\mathcal{F}$ (Base de faits) : $\{\ p(A, B), s(A, B)\ \}$ où A et B sont des constantes

- $\mathcal{R}$ (Base de règles) :
  $R_1 : p(x,y) \rightarrow p(x,x)$
  $R_2 : r(x,x,y) \rightarrow p(y,x)$
  $R_3 : s(x,y) \rightarrow s(y,x)$
  $R_4 : p(x,y) \wedge s(y,z) \rightarrow r(x,y,z)$

Question 1 Saturer la base de faits avec les règles, en procédant en largeur (à chaque étape, calcul de tous les nouveaux homomorphismes des hypothèses de règles dans la base de faits courante, puis application des règles selon ces homomorphismes avant de passer à l'étape suivante ; un homomorphisme est nouveau s'il n'a pas déjà été trouvé à l'étape précédente). Présenter les résultats selon le format suivant :

Etape	Règle applicable	Homomorphisme	fait produit	utile ?
n^o étape	*n^o règle*			*oui/non*
...				

Question 2 On considère la requête booléenne $Q = r(B,u,v)$, où u et v sont des variables ("existe-t-il u et v tels que $r(B,u,v)$" ?). Déterminer si $\mathcal{K}$ répond positivement à Q :
2.1. en vous basant sur le mécanisme de chaînage avant ;
2.2. en vous basant sur le mécanisme de chaînage arrière.

4 De CSP à Homomorphisme

Rappels.

- Le problème *CSP (satisfaction de contraintes)* prend en entrée un réseau de contraintes $P = (X,D,C)$ [où $X = \{x_1,\ldots,x_n\}$ est un ensemble de variables, $D = \{D_i | x_i \in X\}$ est l'ensemble des domaines des variables, C est l'ensemble des contraintes portant sur les variables de X], et demande si ce réseau admet une solution. On suppose que les contraintes sont définies en extension, c'est-à-dire par énumération de tous les tuples autorisés.
- Le problème *HOM (homomorphisme)* prend en entrée un couple $(\mathcal{A}_1,\mathcal{A}_2)$ [où $\mathcal{A}_1$ et $\mathcal{A}_2$ sont des ensembles d'atomes représentant des formules existentielles conjonctives de la logique du premier ordre] et demande s'il existe un homomorphisme de $\mathcal{A}_1$ dans $\mathcal{A}_2$.
- Une *réduction* f d'un problème de décision P_1 à un problème de décision P_2 transforme toute instance I de P_1 ("la donnée en entrée de P_1") en une instance $f(I)$ de P_2 de façon à ce que la réponse à P_1 pour I soit oui si et seulement si la réponse à P_2 pour $f(I)$ est oui.
- Une telle réduction est *polynomiale* si pour tout I elle se calcule en temps polynomial en la taille de I.

Question 1 Définir une réduction polynomiale de CSP à HOM, et l'illustrer avec le réseau de contraintes (X,D,C) ci-dessous :
$X = \{x_1,x_2,x_3,x_4,x_5\}$ $D = \{D_1,...,D_5\}$ avec pour tout $1 \leq i \leq 5$, $D_i = \{a,b,c\}$.
$C = \{C_1,C_2,C_3\}$, avec :
$C_1(x_1,x_2,x_3) = \{(a,a,a),(a,b,c),(b,a,c)\}$
$C_2(x_2,x_3,x_4) = \{(a,a,b),(a,b,c),(c,c,c)\}$
$C_3(x_4,x_5) = \{(a,a),(a,b),(b,c),(c,c)\}$.

Le but est de définir une réduction la plus "simple" possible, en se disant que l'algorithme de recherche d'homomorphismes effectuera les prétraitements qui seraient souhaitables.

Question 2 On étend les réseaux de contraintes considérés dans la question précédente en ajoutant des contraintes d'égalité et de différence données en *intension*. Comment étendre la transformation de CSP à HOM ?