Université de Montpellier – Master Informatique
Intelligence Artificielle – HMIN107

TD3

Quelques problèmes se résolvant avec un algorithme de backtrack

Exercice 1. Trouver un homomorphisme

On considère les ensembles d'atomes suivants :
$\mathcal{A}1$ = { p(x,y), p(y,z), r(z,x) } $\mathcal{A}2$ = { p(a,b), p(b,c), r(a,c), r(c,d), p(c,e), r(e,b), r(e,d) }

(Conseil : représenter $\mathcal{A}1$ et $\mathcal{A}2$ sous forme de graphe pour mettre en évidence leur structure)

Dessiner l'**arbre de recherche** construit par l'algorithme de backtrack, en supposant que :
- les variables de $\mathcal{A}1$ sont considérées dans l'ordre x, y et z
  [calculer à partir de cet ordre le *rang* de chaque atome de $\mathcal{A}1$ en vue du test de solution partielle]
- les termes (constantes ici) de $\mathcal{A}2$ sont considérés dans l'ordre a b c d e.

Y-a-t-il une solution ?

Rappels : problèmes de décision « CSP », « SAT » et « HOM »

Le problème **CSP** *(satisfaction de contraintes)* prend en entrée un réseau de contraintes et demande si ce réseau admet une solution. Un réseau de contraintes est composé d'un ensemble X = {x_1, …, x_n} de variables, d'un domaine D_i pour chaque variable x_i, et d'un ensemble C de contraintes portant sur les variables de X. On se restreint ici à des contraintes données en extension (c'est-à-dire par énumération des tuples de valeurs compatibles).

Le problème **SAT** prend en entrée une conjonction de clauses *S* (ou ensemble de clauses) en logique propositionnelle et demande si *S* est satisfiable.

Le problème **HOM** *(homomorphisme)* prend en entrée 2 ensembles d'atomes représentant des formules existentielles conjonctives de la logique du premier ordre, et demande s'il existe un homomorphisme du premier dans le second.

Exercice 2. Réduction de HOM à CSP (préparation au TP 2)

Soit Q et D deux ensembles d'atomes (représentant respectivement une requête conjonctive et une base de faits).

Q = { p(x,y), p(y,z), p(x,u), p(z,z), r(x), r(u) } où x, y, z et u sont des variables

D = { p(a,b), p(a,c), p(b,c), p(b,d), p(b,e), p(d,e), p(e,e), r(a), r(b), r(c) } où a, b, c, d et e sont des constantes.

1- Quels sont tous les homomorphismes de Q dans D ?

2- Transformer Q et D en un réseau de contraintes au plus binaires P = <X,D,C> de façon à ce que tout homomorphisme de Q dans D soit une solution à P, et réciproquement.

3- Définir précisément la transformation à effectuer dans le cas général ("réduction de HOM à CSP"). En particulier, ne pas oublier qu'une variable peut apparaître deux fois dans un même atome (comme dans l'exemple ci-dessus) et que les atomes peuvent comporter des constantes.

Exercice 3. Réduction de SAT à CSP

Il existe plusieurs réductions polynomiales de **SAT à CSP**. On vous demande de définir une telle transformation sur la base suivante :

- On transforme une instance de SAT en un réseau de contraintes *binaires*.
- A chaque clause Ci, on associe une variable xi, dont le domaine de valeurs est l'ensemble des littéraux qui rendent vrai cette clause (par exemple à C1 = $(\neg a \vee b)$, on associe une variable x1 dont le domaine est $\{\neg a, b\}$).

Complétez cette transformation en indiquant comment sont construites les *contraintes*. Illustrez-la sur l'instance de SAT suivante : $(\neg a \vee b) \wedge (a \vee c \vee \neg d) \wedge (\neg c \vee b)$.

Exercice 4 : Réduction de CSP à SAT

On veut construire une réduction de CSP à SAT, c'est-à-dire une transformation *t* de tout réseau de contraintes R en un ensemble de clauses *S* de façon à ce que R admette une solution si et seulement si S est satisfiable.

1. On cherche d'abord à traduire en logique des propositions le fait que "toute variable x de X doit avoir une valeur prise dans son domaine", sans chercher à traduire la notion de contrainte pour l'instant. On note "xv" le symbole propositionnel (ou "variable propositionnelle") exprimant que "la variable x prend la valeur v", où x est une variable de X et v une valeur de son domaine. Avec cette notation, exprimer sous forme de clauses les conditions suivantes :

   a. Toute variable x de X prend une valeur de son domaine ;
   b. Toute variable ne peut prendre qu'une valeur.

   Illustrer ceci sur l'exemple ci-dessous.

2. Ajouter à la transformation la représentation des contraintes. Illustrer la transformation obtenue sur l'exemple.

Indication : énumérer les tuples de valeurs interdits par les contraintes plutôt que les tuples autorisés.

Exemple

X = {x1, x2, x3, x4}
D1 = {a,b,c} **D2** = **D3** = {a, b} **D4** = {b, c}

C1

x1	x2
a	a
b	b
c	b

C2

x2	x3	x4
a	a	b
a	a	c
b	b	b
b	a	c