USTHB - FEI - Département d'Informatique
Master 2 "Systèmes Informatiques Intelligents" 2016/2017
Module "Programmation par Contraintes"
Date : 04/12/2016 <9h40-11h10>

Corrigé de l'interrogation

Exercice 1 (6 points)

On considère l'instance suivante du problème de coloriage d'un graphe : coloriage d'un triangle ABC de telle sorte que deux sommets adjacents n'aient pas la même couleur, avec V (Vert) l'unique couleur permise pour le sommet A, et V et R (Rouge) les couleurs permises pour chacun des deux autres sommets.

- 1) Modélisez l'instance à l'aide d'un CSP binaire discret $P=(X,D,C)$.
- 2) Appliquez l'algorithme de consistance d'arc AC3 au CSP P.
- 3) Que pouvez-vous en déduire ?

Solution :

1) Le CSP binaire discret $P=(X,D,C)$ modélisant l'instance est comme suit :

- $X=\{X_1, X_2, X_3\}$, avec X_1 , X_2 et X_3 représentant, respectivement, les couleurs associées aux sommets A, B et C
- $D(X_1)=\{V\}$, $D(X_2)=D(X_3)=\{V,R\}$
- $C=\{c_1 : X_1 \neq X_2, c_2 : X_1 \neq X_3, c_3 : X_2 \neq X_3\}$

2) **Application de AC3 :**

La file Q de AC3 est initialisée aux arcs sur lesquels il y a une contrainte, dans les deux sens : $Q=\{(X_1, X_2), (X_2, X_1), (X_1, X_3), (X_3, X_1), (X_2, X_3), (X_3, X_2)\}$. Les domaines, initialement, sont tels que donnés dans le CSP :

X_i	X_1	X_2	X_3
$D(X_i)$	{V}	{V,R}	{V,R}

- On prend l'arc (X_1, X_2) pour propagation. La file devient : $Q=\{(X_2, X_1), (X_1, X_3), (X_3, X_1), (X_2, X_3), (X_3, X_2)\}$.

L'unique valeur V de $D(X_1)$ a R comme support dans $D(X_2)$: rien à supprimer.

- On passe à la propagation de l'arc (X_2, X_1) . La file devient : $Q=\{(X_1, X_3), (X_3, X_1), (X_2, X_3), (X_3, X_2)\}$.

Cette fois, la valeur V de $D(X_2)$ n'a pas de support dans $D(X_1)$: on la supprime et les domaines deviennent (rien à ajouter à la file) :

X_i	X_1	X_2	X_3
$D(X_i)$	{V}	{R}	{V,R}

- On prend l'arc (X_1, X_3) pour propagation, qui ne produit aucun effet sur $D(X_1)$.
- On prend maintenant l'arc (X_3, X_1) . La file devient : $Q=\{(X_2, X_3), (X_3, X_2)\}$. La valeur V de $D(X_3)$, qui n'a pas de support dans $D(X_1)$, est supprimée ; et les domaines deviennent (ici aussi, rien à ajouter à la file) :

X_i	X_1	X_2	X_3
$D(X_i)$	$\{V\}$	$\{R\}$	$\{R\}$

- En prenant l'arc (X_2, X_3) pour propagation, on se rend compte que l'unique valeur R de $D(X_2)$ n'a plus de support dans $D(X_3)$: on la supprime et le domaine de $D(X_2)$ passe à l'ensemble vide :

$$Q = \{(X_3, X_2)\}$$

X_i	X_1	X_2	X_3
$D(X_i)$	$\{V\}$	$\{\}$	$\{R\}$

- Un domaine de variable étant devenu vide, AC3 s'arrête.

3) On déduit de la réponse à la question 2 que le CSP est inconsistant : il n'admet pas de solution.

Exercice 2 (4 points) :

- Expliquez comment modéliser une instance du problème 2-SAT à l'aide d'un CSP binaire discret.
- Donnez la forme des poids des arcs de la représentation graphique de la modélisation : illustrez avec des exemples.

Solution :

1) Soit $\text{Conj} = c_1 \wedge \dots \wedge c_i \wedge \dots \wedge c_m$ une conjonction de m clauses ayant chacune au plus deux littéraux : en d'autres termes, Conj est une instance du problème 2-SAT. Soit $\text{Prop} = \{p_1, \dots, p_n\}$ l'ensemble des propositions atomiques occurrant dans Conj . L'instance peut être modélisée par le CSP binaire discret $P = (X, D, C)$ suivant :

- $X = \{X_1, \dots, X_n\}$
- $D(X_i) = \{V(\text{rai}), F(\text{aux})\}$, pour tout $i \in \{1, \dots, n\}$
- $C = \{c_1 : f(c_1) = V, \dots, c_i : f(c_i) = V, \dots, f(c_m) = V\}$, $f(c_i)$, pour tout $i \in \{1, \dots, m\}$, étant la clause c_i dans laquelle, pour tout $j \in \{1, \dots, n\}$, les occurrences éventuelles de la proposition atomique p_j ont été remplacées par la variable X_j .

2) Les poids des arcs de la représentation graphique de la modélisation sont des matrices carrées 2×2 booléennes (matrices de 0 et de 1).

- Exemple 1 (clause binaire) :** $c_i = \neg p_1 \vee p_2 \rightarrow f(c_i) = \neg X_1 \vee X_2$

Le poids $l(X_1, X_2)$ de l'arc (X_1, X_2) sera la matrice booléenne $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$

- Exemple 2 (clause unaire) :** $c_i = \neg p_1 \rightarrow f(c_i) = \neg X_1$

Le poids $l(X_1, X_1)$ de l'arc (X_1, X_1) sera la matrice booléenne $\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

Exercice 3 (5 points) :

- Expliquez comment adapter l'algorithme Générer et Tester à la résolution d'un CSP de directions cardinales. Justifiez la complétude de l'adaptation.
- Expliquez comment adapter l'algorithme Générer et Tester à la résolution d'un TCSP. Justifiez la complétude de l'adaptation.

Solution :

1) On adapte l'algorithme Générer et Tester à la résolution d'un CSP de directions cardinales de la façon suivante : on considère tous les raffinements atomiques (c'est-à-dire, dont les entrées de la représentation matricielle sont toutes des relations atomiques). Pour chacun de ces raffinements,

l'algorithme applique PC2. L'algorithme s'arrête, avec une réponse positive au problème de consistance du CSP de départ, si un raffinement est trouvé pour lequel PC2 ne détecte la relation vide (ne transforme pas une entrée de la représentation matricielle en la relation vide). Si tous les raffinements sont parcourus sans en trouver un pour lequel PC2 ne détecte pas la relation vide, l'algorithme prend fin avec une réponse négative au problème de consistance du CSP de départ. La complétude de l'adaptation est justifiée par la complétude de PC2 pour les CSP atomiques de directions cardinales.

- 2) On adapte l'algorithme Générer et Tester à la résolution d'un TCSP de la façon suivante : on considère tous les raffinements convexes maximaux. Pour chacun de ces raffinements, l'algorithme applique PC2. L'algorithme s'arrête, avec une réponse positive au problème de consistance du TCSP de départ, si un raffinement est trouvé pour lequel PC2 ne détecte pas l'ensemble vide (ne transforme pas une entrée de la représentation matricielle en l'ensemble vide). Si tous les raffinements sont parcourus sans en trouver un pour lequel PC2 ne détecte pas l'ensemble vide, l'algorithme prend fin avec une réponse négative au problème de consistance du TCSP de départ. La complétude de l'adaptation est justifiée par la complétude de PC2 pour les STP.

Exercice 4 (5 points) :

- 1) Donnez un programme Prolog extrayant les éléments d'une liste autres que 5, dont les positions sont multiples de 3.
- 2) Soit le CSP binaire discret $P=(X,D,C)$ défini comme suit :
 - $X=\{X_1,X_2,X_3\}$
 - $D(X_i)=\{1,2,3,4,5\}$, pour tout $i \in \{1,2,3\}$
 - $C=\{c_1 : X_1+3 \leq X_2, c_2 : X_1+5=X_3, c_3 : X_2 \neq X_3\}$

Ecrire un programme Prolog saisissant le CSP et faisant appel au solveur sur domaines finis pour sa résolution.

Solution :

- 1)

```
extraire(L,[]) :-length(L,Longueur),
    Longueur<3,
    !.
extraire([_,_5|L1],L2) :-!,
    extraire(L1,L2).
extraire([_,_X|L1],[X|L2]) :-extraire(L1,L2).
```
- 2)

```
:- use_module(library('clp/bounds')).
csp_interro(Vars) :- Vars = [X1,X2,X3],
    Vars in 1..5,
    X1+3 #=< X2,
    X1+5 #=X3,
    X2 #\=X3,
    label(Vars).
```