COMPTE-RENDU DE TRAVAUX PRATIQUES

DE PROGRAMMATION LOGIQUE (SEANCE 1)

Mikaël CAPELLE - Estelle Chauveau

INSA-DGEI

7 décembre 2013

# Familiarisation sur un csp simple

Ci-dessous le code final du fichier essai.ecl après ajout de commentaire et modification :

:- lib(fd).

% Donne le problème et affiche la 1ère solution trouvée (+ more pour afficher les suivantes).

main1(X) :-

csp(X),

instancier(X).

% Donne le problème et donne l'ensemble des solutions.

main2(S) :-

csp(X),

findall(X, instancier(X), S).

% Donne le problème, l'ensemble des solutions, et affiche le domaine de chaque variable à chaque étape de la résolution.

main3(S) :-

csp(X),

findall(X, instancier2(X), S).

% Prédicat qui pose le problème sans le rédoure :

% - On définit les variables du CSP [A, B, C]

% - Leurs domaines (le même pour tous)

% - Les contraintes du CSP

csp(X) :-

variables(X),

domaines(X),

contraintes(X).

% Stocke dans X une liste de 3 variables non instanciées.

variables(X) :-

X = [A,B,C].

% Définit le domaine de la variable (ou des variables de la liste) X.

domaines(X) :-

X :: [bleu, rouge].

% Définit les contraintes sur les variables de X.

contraintes(X) :-

X = [A,B,C],

A #\= B,

B #\= C. %,

% A #\= C.

% Equivalent à labeling

% Instancie chaque n-uplet de variables possibles parmis X

instancier(Vars) :-

( foreach(X, Vars) do indomain(X) ).

% Instancie chaque n-uplet de variables possibles parmis X

instancier2(Vars) :-

( foreach(X, Vars) do dom(X, L), write(L), write(' '), indomain(X) ),

nl.

## Réponses aux questions de "cours"

* Que signifie le message “There are 3 delayed goals” ? Lorsqu’on exécute la requête :

Cela signifie que les contraintes affichées sont consistantes vis-à-vis du premier algorithme utilisé pour la propagation de contrainte (arc consistance, incomplet) et qu’elles sont donc actives et non violées pour l’instant.

* Le problème a-t-il des solutions ? Relâcher la 3ème contrainte « A #\= C » en la plaçant en commentaire. Quelles sont les solutions obtenues ?

Le problème n’a pas, initialement, de solution. Après relâchement de la 3ème contrainte, on obtient 2 solutions : (rouge, bleu, rouge), (bleu, rouge, bleu).

* Comment utiliser le tracer pour suivre la résolution pas à pas ?

?- trace

# Programme des N reines

Ci-dessous le code de notre programme de résolution du problème des N reines, avec différentes politiques pour le choix des variables et des valeurs.

:- lib(fd).

:- lib(fd\_search).

queens(N, X) :-

csp(X, N),

labeling(X).

queens\_ff(N, X) :-

csp(X, N),

labeling\_ff(X).

queens\_med(N, X) :-

csp(X, N),

labeling\_med(X).

queens\_ff\_med(N, X) :-

csp(X, N),

labeling\_ff\_med(X).

% Prédicat qui pose le problème des N reines sans le rédoure :

% - On définit les variables du CSP

% - Leurs domaines

% - Les contraintes du CSP

csp(X, N) :-

variables(X, N),

domains(X),

constraints(X).

% Stocke dans L une liste des variables non instanciées.

variables(L, N) :-

length(L, N).

% Définit le domaine de chaque variable.

domains(X) :-

length(X, N),

X :: 1..N.

% Définit les contraintes sur les variables de X.

constraints(X) :-

alldifferent(X),

(foreach(V, X), foreach(VS, DS), foreach(VD, DD), count(I, 0, \_) do VS = V + I, VD = V - I),

alldifferent(DS),

alldifferent(DD).

% Instancie les variables en utilisant le prédicat deleteff

labeling\_ff(L) :-

(fromto(L, I, O, []) do deleteff(X, I, O), indomain(X)).

labeling\_med(L) :-

foreach(X, L) do fd\_search:indomain(X, median).

labeling\_ff\_med(L) :-

(fromto(L, I, O, []) do deleteff(X, I, O), fd\_search:indomain(X, median)).