



Practica Clingo Concurrente

Representación do Coñecemento e Razoamento Automático (Universidade da Coruña)

%Daniel Dobarro Landeira
%Roi Millan Miguez

#program initial.

%delimitamos las casillas con la constante para que no falle al sumar o al restar
c(0..n-1,0..n-1).

%definimos la accion de tocar
action(tocar(X,Y)) :- c(X,Y).

%Definimos cuando un par de casillas son adyacentes

% La de abajo

adyacente(c(X,Y),c(X2,Y2)) :- c(X,Y), c(X2,Y2), X2=X+1, Y2=Y.

% La de arriba

adyacente(c(X,Y),c(X2,Y2)) :- c(X,Y), c(X2,Y2), X2=X-1, Y2=Y.

% La de la derecha

adyacente(c(X,Y),c(X2,Y2)) :- c(X,Y), c(X2,Y2), X2=X, Y2=Y+1.

% La de la izquierda

adyacente(c(X,Y),c(X2,Y2)) :- c(X,Y), c(X2,Y2), X2=X, Y2=Y-1.

% Marcamos la propia casilla como adyacente

adyacente(c(X,Y),c(X2,Y2)) :- c(X,Y), c(X2,Y2), X2=X, Y2=Y.

% Cierre reflexivo

adyacente(c(X,Y),c(X2,Y2)) :- adyacente(c(X2,Y2),c(X,Y)).

#program dynamic.

% Generacion de acciones

% Ahora permitimos que se generen tantas acciones como quiera

{ o(A): _action(A) }.

% Registramos las casillas que ya han sido tocadas e impedimos que se vuelvan a tocar ya que en la solución mínima no se tocan dos veces la

% misma casilla

tocada(X,Y) :- o(tocar(X,Y)).

tocada(X,Y) :- 'tocada(X,Y).

:- o(tocar(X,Y)), 'tocada(X,Y).

% Axiomas de efecto

% Anteriormente pensabamos en tocar una casilla, modificar las adyacentes y no permitir que se toquen otras casillas que modificarían las adyacentes de otras

% Ahora cambiamos la manera de pensar y nos centramos en la propia casilla, contamos cuantas veces ha sido modificada y si es un numero impar de veces el valor

% Cambia y si es un numero par el valor se mantiene por la inercia

h(c(X,Y),on) :- 'h(c(X,Y),off), _c(X,Y), #count{ X2,Y2 : o(tocar(X2,Y2)), _adyacente(c(X,Y),c(X2,Y2))} = S, (S\2) == 1.

h(c(X,Y),off) :- 'h(c(X,Y),on), _c(X,Y), #count{ X2,Y2 : o(tocar(X2,Y2)), _adyacente(c(X,Y),c(X2,Y2))} = S, (S\2) == 1.

```

% Inercia
% Si no se le ha cambiado el valor a una casilla se le mantiene el mismo del anterior estado
h(F,V) :- 'h(F,V), not c(F).
c(F)  :- 'h(F,V), h(F,W), V!=W.

#program final.

% Mientras haya casillas encendidas no acaba
:- h(c(_,_),on).

% Minimizamos el numero de tocadas para tener la solución óptima
#minimize { 1,X,Y : tocada(X,Y) }.
#show o/1.

```