

Programación no determinista en Prolog: Coloreado de Quadrees

Curso 2003/04, Fecha:9/01/2004

Enunciado 1 (repiteN) Utilizar los predicados sobre quadrees definidos en las prácticas anteriores.

Para ello, puede utilizarse la directiva `include/1` que permite cargar el nombre de fichero que se pasa como argumento.

Por ejemplo, para importar las definiciones del fichero `fich.pl`, se utilizaría la línea

```
:- include('fich.pl').
```

Construir un predicado `repite(N,X,Y)` que se cumple si Y es un quadtree formado al repetir el N veces el quadtree X en cada cuadrante.

Enunciado 2 (cambiaColor) Definir un predicado `cambiaColor(C,Q1,Q2)` que se cumple si $Q2$ es el quadtree resultante de poner el color C en todas las casillas del quadtree $Q1$.

Enunciado 3 (juntos) El siguiente predicado comprueba que en un quadtree no hay dos cuadrantes juntos del mismo color

```
noJuntos(vacio).
noJuntos(rect(_)).
noJuntos(d(A,B,C,D)):-
    noJuntos(A), noJuntos(B), noJuntos(C), noJuntos(D),
    hazArbol(der,A,Ar), hazArbol(izq,B,Bl), noEq(Ar,Bl),
    hazArbol(der,C,Cr), hazArbol(izq,D,Dl), noEq(Cr,Dl),
    hazArbol(inf,A,Ad), hazArbol(sup,C,Cu), noEq(Ad,Cu),
    hazArbol(inf,B,Bd), hazArbol(sup,D,Du), noEq(Bd,Du).
```

```
noEq(vacio, _).
noEq(_, vacio).
noEq(rect(X), rect(Y)):- X=Y.
noEq(rect(X), f(A,B)):- noEsta(X,A), noEsta(X,B).
noEq(f(A,B), rect(X)):- noEsta(X,A), noEsta(X,B).
```

$noEq(f(A,B), f(C,D)) :- noEq(A,C), noEq(B,D).$

$noEsta(-, vacio).$

$noEsta(X, rect(Y)) :- X \setminus = Y.$

$noEsta(X, f(A,B)) :- noEsta(X,A), noEsta(X,B).$

$hazArbol(-, vacio, vacio).$

$hazArbol(-, rect(X), rect(X)).$

$hazArbol(X,D, f(V,W)) :-$

$toma(X,D,Y,Z), hazArbol(X,Y,V), hazArbol(X,Z,W).$

$toma(sup, d(A,B,-,-), A,B).$

$toma(inf, d(-,-,C,D), C,D).$

$toma(izq, d(A,-,C,-), A,C).$

$toma(der, d(-,B,-,D), B,D).$

Se pide:

- *Editar en un archivo y consultar el fragmento de código anterior*
- *Definir un quadtree de 4×4 celdas y dos colores que no tenga ninguna celda junta del mismo color y comprobar que el predicado funciona correctamente*
- *Definir un quadtree de 4×4 celdas con dos colores juntos y comprobar que el predicado lo detecta*
- *Definir un quadtree en forma de rombo que cumpla la condición y comprobar que el predicado funciona*

Enunciado 4 (colorea) *Definir un predicado colorea($Xs, Q1, Q2$) que se cumple si $Q2$ es un quadtree formado al cambiar los colores de $Q1$ por valores de la lista Xs .*

?- $repiteN(2, rect(rgb(0,0,0)), B), colorea([rgb(1,0,0), rgb(0,1,0)], B, Q).$

$B = \dots$

$Q = d(rect(rgb(1,0,0)), rect(rgb(1,0,0)), rect(rgb(1,0,0)), rect(rgb(1,$

$B = \dots$

$Q = d(rect(rgb(1,0,0)), rect(rgb(1,0,0)), rect(rgb(1,0,0)), rect(rgb(0,$

$B = \dots$

$Q = d(rect(rgb(1,0,0)), rect(rgb(1,0,0)), rect(rgb(0,1,0)), rect(rgb(1,$

$B = \dots$

$Q = d(rect(rgb(1,0,0)), rect(rgb(1,0,0)), rect(rgb(0,1,0)), rect(rgb(1,$

$B = \dots$
 $Q = d(\text{rect}(\text{rgb}(1, 0, 0)), \text{rect}(\text{rgb}(1, 0, 0)), \text{rect}(\text{rgb}(0, 1, 0)), \text{rect}(\text{rgb}(0,$

$B = \dots$
 $Q = d(\text{rect}(\text{rgb}(1, 0, 0)), \text{rect}(\text{rgb}(0, 1, 0)), \text{rect}(\text{rgb}(1, 0, 0)), \text{rect}(\text{rgb}(1,$

$B = \dots$
 $Q = d(\text{rect}(\text{rgb}(1, 0, 0)), \text{rect}(\text{rgb}(0, 1, 0)), \text{rect}(\text{rgb}(1, 0, 0)), \text{rect}(\text{rgb}(0,$

$B = \dots$
 $Q = d(\text{rect}(\text{rgb}(1, 0, 0)), \text{rect}(\text{rgb}(0, 1, 0)), \text{rect}(\text{rgb}(0, 1, 0)), \text{rect}(\text{rgb}(1,$

$B = \dots$
 $Q = d(\text{rect}(\text{rgb}(1, 0, 0)), \text{rect}(\text{rgb}(0, 1, 0)), \text{rect}(\text{rgb}(0, 1, 0)), \text{rect}(\text{rgb}(0,$

\dots

Pista: Puede utilizarse el predicado pertenece de la práctica 2

Enunciado 5 (colorBruto) Construir un predicado $\text{colorBruto}(Xs, Q1, Q2)$ que se cumple si $Q2$ es el quadtree obtenido al colorear el quadtree $Q1$ con los colores de la lista Xs de forma que no haya dos cuadrantes juntos con el mismo color

?- repiteN(2, rect(rgb(0,0,0)), B), colorBruto([rgb(1,0,0), rgb(0,1,0)], B, Q).

$B = \dots$
 $Q = d(\text{rect}(\text{rgb}(1, 0, 0)), \text{rect}(\text{rgb}(0, 1, 0)), \text{rect}(\text{rgb}(0, 1, 0)), \text{rect}(\text{rgb}(1,$

$B = \dots$
 $Q = d(\text{rect}(\text{rgb}(0, 1, 0)), \text{rect}(\text{rgb}(1, 0, 0)), \text{rect}(\text{rgb}(1, 0, 0)), \text{rect}(\text{rgb}(0,$

No

Resolver el problema para un quadtree de 4×4 y 3 colores

Enunciado 6 (sacaAlea(Opcional)) Construir un predicado $\text{sacaAlea}(Xs, X)$ que se cumple si X es un elemento de la lista Xs en una posición aleatoria

Enunciado 7 (coloreaAlea(Opcional)) Construir un predicado $\text{coloreaAlea}(Xs, Q1, Q2)$ que se cumple si $Q2$ es un quadtree obtenido al colorear el quadtree $Q1$ con colores de la lista Xs obtenidos de forma aleatoria

Enunciado 8 (colorAlea(Opcional)) Construir un predicado $\text{colorAlea}(Xs, Q1, Q2)$ con el mismo comportamiento que colorBruto pero que seleccione aleatoriamente los colores de la lista Xs

Enunciado 9 (algoritmo(Opcional)) *Buscar un algoritmo más eficiente que permita colorear un quadtree de forma que no haya dos casillas juntas del mismo color. ¿Cuál es el mínimo número de colores que se necesitan?. En Algorithms for coloring quadtrees se pueden encontrar diversos algoritmos para colorear quadtrees. ¿Serías capaz de implementarlos en Prolog?*