

# Programación recursiva y aritmética en Prolog

Curso 2002/03, Fecha:10/01/2002

**Enunciado 1 (blanco)** Utilizar los predicados definidos en las dos 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').
```

Definir un `blanco(N,B)` que se cumple si  $B$  es un quadtree de  $2^N \times 2^N$  casillas de color blanco (`rgb(255,255,255)`)

**Enunciado 2 (mapQT)** Definir un predicado `mapQT` que cambie los colores de un quadtree por otros colores. Puede utilizarse una tabla de conversión del tipo:

```
tabla(rgb(255,255,255),rgb(0,255,0)).  
tabla(rgb(255,255,255),rgb(0,255,255)).  
tabla(rgb(255,255,0),rgb(0,0,255)).  
tabla(X,X).
```

**Enunciado 3 (triang)** Definir un predicado `triang(N,C,Q)` que se cumple si  $Q$  es un quadtree que representa el triángulo rectángulo de color  $C$  formado al dividir un cuadrado en diagonal.  $N$  indica la resolución con la que se realiza la división.

**Enunciado 4 (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),
```

$hazArbol(inf, B, Bd), hazArbol(sup, D, Du), noEq(Bd, Du).$

$noEq(vacio, -).$   
 $noEq(-, vacio).$   
 $noEq(rect(X), rect(Y)):-X=Y.$   
 $noEq(rect(X), f(A, B)):-noEsta(X, A), noEsta(X, B).$   
 $noEq(f(A, B), rect(X)):-noEsta(X, A), noEsta(X, B).$   
 $noEq(f(A, B), f(C, D)):-noEq(A, C), noEq(B, D).$

$noEsta(-, vacio).$   
 $noEsta(X, rect(Y)):-X=Y.$   
 $noEsta(X, f(A, B)):-noEsta(X, A), noEsta(X, B).$

$hazArbol(-, vacio, vacio).$   
 $hazArbol(-, rect(X), rect(X)).$   
 $hazArbol(X, D, f(V, W)):-$   
 $toma(X, D, Y, Z), hazArbol(X, Y, V), hazArbol(X, Z, W).$

$toma(sup, d(A, B, -, -), A, B).$   
 $toma(inf, d(-, -, C, D), C, D).$   
 $toma(izq, d(A, -, C, -), A, C).$   
 $toma(der, d(-, B, -, D), B, D).$

Se pide:

- Añadir y consultar el fragmento de código anterior
- Definir un quadtree de  $4 \times 4$  celdas y dos colores que no tenga ninguna celda junta del mismo color y comprobar que el predicado funciona correctamente
- Definir un quadtree de  $4 \times 4$  celdas con dos colores juntos y comprobar que el predicado lo detecta
- Definir un quadtree en forma triangular (como los generados por el predicado triang que cumpla la condición y comprobar que el predicado funciona

**Enunciado 5 (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  $Xs$ .

?- *blanco* (2, B), *colorea* ([*rgb* (255, 0, 0), *rgb* (0, 255, 0)], B, Q).

B = ...

Q =  $d(rect(rgb(255, 0, 0)), rect(rgb(255, 0, 0)),$   
 $rect(rgb(255, 0, 0)), rect(rgb(255, 0, 0)))$  ;

B = ...

Q =  $d(rect(rgb(255, 0, 0)), rect(rgb(255, 0, 0)),$   
 $rect(rgb(255, 0, 0)), rect(rgb(0, 255, 0)))$  ;

$$B = \dots$$

$$Q = d(\text{rect}(\text{rgb}(255, 0, 0)), \text{rect}(\text{rgb}(255, 0, 0)), \\ \text{rect}(\text{rgb}(0, 255, 0)), \text{rect}(\text{rgb}(255, 0, 0))) ;$$

$$B = \dots$$

$$Q = d(\text{rect}(\text{rgb}(255, 0, 0)), \text{rect}(\text{rgb}(255, 0, 0)), \\ \text{rect}(\text{rgb}(0, 255, 0)), \text{rect}(\text{rgb}(0, 255, 0)))$$

...

*Pista: Puede utilizarse el predicado pertenece de la práctica 2*

**Enunciado 6 (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

?- blanco(2, B), colorBruto([rgb(255, 0, 0), rgb(0, 255, 0)], B, Q).

$$B = \dots$$

$$Q = d(\text{rect}(\text{rgb}(255, 0, 0)), \text{rect}(\text{rgb}(0, 255, 0)), \\ \text{rect}(\text{rgb}(0, 255, 0)), \text{rect}(\text{rgb}(255, 0, 0))) ;$$

$$B = \dots$$

$$Q = d(\text{rect}(\text{rgb}(0, 255, 0)), \text{rect}(\text{rgb}(255, 0, 0)), \\ \text{rect}(\text{rgb}(255, 0, 0)), \text{rect}(\text{rgb}(0, 255, 0))) ;$$

No

*Resolver el problema para un quadtree de  $4 \times 4$  y 3 colores*

**Enunciado 7 (sacaAlea)** Construir un predicado  $\text{sacaAlea}(Xs, X)$  que se cumple si  $X$  es un elemento de la lista  $Xs$  en una posición aleatoria

**Enunciado 8 (coloreaAlea)** 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 9 (colorAlea)** Construir un predicado  $\text{colorAlea}(Xs, Q1, Q2)$  con el mismo comportamiento que  $\text{colorBruto}$  pero que seleccione aleatoriamente los colores de la lista  $Xs$

**Enunciado 10 (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?.