

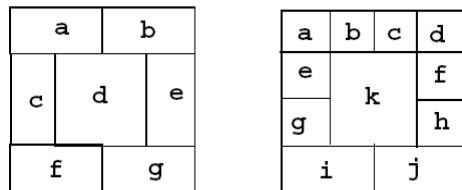
## Programación Declarativa

19 de diciembre de 2005

---

Un mapa puede representarse mediante la relación  $\text{mapa}(N,L)$  donde  $N$  es el nombre del mapa y  $L$  es la lista de los pares formados por cada una de las regiones del mapa y la lista de sus regiones vecinas.

Por ejemplo, los mapas siguientes:



se pueden representar por

```
mapa(ejemplo_1,  
  [a-[b,c,d], b-[a,d,e], c-[a,d,f], d-[a,b,c,e,f,g],  
  e-[b,d,g], f-[c,d,g], g-[d,e,f]]).
```

```
mapa(ejemplo_2,  
  [a-[b,e,k], b-[a,c,e,k], c-[b,d,f,k], d-[c,f,k], e-[a,b,g,k],  
  f-[c,d,h,k], g-[e,i,k], h-[f,j,k], i-[g,j,k], j-[i,h,k],  
  k-[a,b,c,d,e,f,g,h,i,j]]).
```

Definir la relación  $\text{coloracion}(M,LC,S)$  que se verifica si  $S$  es una lista de pares formados por una región del mapa  $M$  y uno de los colores de la lista de colores  $LC$  tal que las regiones vecinas tengan colores distintos.

Por ejemplo,

```
?- coloracion(ejemplo_1,[rojo,verde,azul],S).  
S = [a-verde, b-rojo, c-rojo, d-azul, e-verde, f-verde, g-rojo]
```

Como ayuda para la resolución de este problema podremos seguir los siguientes pasos e implementar una serie de predicados tales como

1. En primer lugar deberemos tener un predicado que se cumpla cuando unifique un Area con una lista de Area-Color y un Color.  
**colorde/3**
2. Tendremos que evitar el conflicto entre áreas vecinas, por lo que deberemos construir un predicado que se cumpla cuando exista algún conflicto entre un área y sus vecinos.  
**conflicto/3**
3. Escribir un predicado que unifique una lista de zonas con sus vecinos, una lista de colores y una lista de cada una de las zonas con sus colores  
**colorear/3**
4. Por último, realiza un predicado que resuelva este problema cuando unifique un mapa con una lista de colores y una lista de areas y colores.  
**coloracion/3**