

PROGRAMACIÓN LÓGICA

Lights out

Programación Declarativa

Pedro Álvarez Piedehierro

INTRODUCCION AL PROYECTO

El juego Lights Out tiene muchas versiones distintas, aunque siempre basadas en el mismo principio.

Se trata de un juego individual. Normalmente se juega en un tablero cuadrado. En cada posición se encuentra una bombilla que puede estar apagada o encendida.

Cada vez que se marca una posición, las bombillas que ocupan esa posición y las casillas adyacentes cambian de estado (de encendida a apagada, o viceversa). Generalmente, el patrón de activación de las casillas adyacentes suele cambiar el estado de las casillas que se encuentran arriba, abajo, a la derecha y a la izquierda de la marcada.

Partiendo de una situación inicial de bombillas encendidas, el objetivo es ir seleccionando casillas hasta conseguir que se apaguen todas las bombillas.

El **objetivo de este proyecto** es escribir un programa en Prolog que, dada una configuración inicial de luces encendidas en un tablero cuadrado, devuelva una lista con las coordenadas de las casillas que hay que marcar para conseguir que todas las luces del tablero se apaguen.

PREDICADOS DEL PROYECTO

Aquí podremos ver todos los predicados de los que se compone el proyecto, dando a conocer que hace cada uno, y para que sirven los argumentos de cada predicado.

lights(+N,+LEncendidas, ?LMarcadas)

Devuelve en LMarcadas la combinación de casillas que hay que pulsar para apagar la combinación de luces encendidas LEncendidas. Devuelve como alternativas todas las posibilidades, sin permutaciones de éstas.

Argumentos:

- **N:** Entero que indica el lado de el tablero
- **LEncendidas:** Lista de casillas que indica las casillas que están encendidas.
- **LMarcadas:** Lista de casillas que indica que casillas hay que pulsar para apagar la combinación LEncendidas.

generarPosibles(+N, +M, ?LCombinacion)

Devuelve en LMarcadas una posible combinación de casillas de una matriz de lado N, para que devuelva todas las combinaciones posibles hay que pasarle un '1' en la variable N. Devuelve como alternativas todas las combinaciones de casillas .

Argumentos:

- **N:** Entero que indica el lado de el tablero
- **M:** Entero que se utiliza como índice para el método, al llamar al método debe ponerse a 1.
- **LCombinacion:** Lista de casillas que indica una combinación de casillas contenidas en el tablero de lado N.

itocoord(+L,+I,?F,?C)

Devuelve en F y C la fila y columna respectivamente que ocupa la posición I de un vector en una matriz cuadrada de lado L.

Argumentos:

- **L:** Entero que indica el lado de el tablero
- **I:** Entero que indica el índice del vector
- **F:** Entero, fila de la casilla
- **C:** Entero, columna de la casilla.

buscar(+X, +L)

Busca el valor X en la lista L, si no lo encuentra falla el predicado.

Argumentos:

- **X:** Elemento a buscar en la lista L.
- **L:** Lista en la que buscar el elemento X.

borrar(+L, +X, ?L2)

Borra la primera aparición de valor X en la lista L. Devuelve el resultado en la lista L2.

Argumentos:

- **L:** Lista en la que borrar el elemento X.
- **X:** Elemento a borrar en la lista L.
- **L2:** Lista resultado, en la que se ha borrado el elemento X.

insertar(+L, +X, ?L2)

Inserta en la lista L el valor X, devolviendo el resultado en la lista L2

Argumentos:

- **L:** Lista en la que insertar el elemento X.
- **X:** Elemento a insertar en la lista L.
- **L2:** Lista resultado, en la que se ha insertado el elemento X.

casillavalida(+N, +C)

El predicado falla si la casilla C no pertenece a la matriz cuadrada de lado N.

Argumentos:

- **N:** Entero que indica el lado del tablero.
- **C:** Casilla que se quiere comprobar si esta dentro del tablero o no.

modificar(+C, +LEncendidas, ?LEncendidas2)

Si está C en LEncendidas, lo borra, sino lo inserta. Devuelve el resultado de la acción en LEncendidas2.

Argumentos:

- **C:** Indica la casilla la cual se quiere cambiar de estado
- **LEncendidas:** Lista de casillas que están encendidas.
- **LEncendidas2:** Lista de casillas encendidas resultante después de modificar el estado de la casilla C.

activarDesactivarCasilla(+N,+C,+LEncendidas,?LEncendidas2)

Modifica el estado la casilla C de un tablero cuadrado de lado N. Comprueba primero si la casilla es valida, si lo es llama al metodo modificar.

Argumentos:

- **N:** Indica el valor del lado del tablero
- **C:** Casilla que se quiere modificar el estado lights out.
- **LEncendidas:** Lista de casillas que están encendidas.
- **LEncendidas2:** Lista de casillas encendidas resultante después de modificar el estado de la casilla C.

cambiar(+N, +C, +LEncendidas, ?LEncendidasR)

Modifica el estado de la lista LEncendidas según el patrón Lights Out cuando se pulsa la casilla C en un tablero de lado N. Devuelve el resultado en LEncendidasR.

Argumentos:

- **N:** Indica el valor del lado del tablero
- **C:** Casilla que se quiere pulsar, usando el patrón lights out.
- **LEncendidas:** Lista de casillas que están encendidas.
- **LEncendidasR:** Lista de casillas encendidas resultante después realizar las modificaciones

probarsolucion(+N,+LEncendidas,+LMarcadas)

Prueba la solución LMarcadas en el tablero de lado N siendo las casillas contenidas en LEncendidas las casillas que están encendidas. Si la solución no deja el tablero apagado, es decir, la lista LEncendidas vacía, falla el predicado.

Argumentos:

- **N:** Indica el valor del lado del tablero
- **LEncendidas:** Lista de casillas que están encendidas.
- **LMarcadas:** Lista de casillas de la posible solución que se quiere probar.

EJEMPLOS DE FUNCIONAMIENTO

En este apartado podemos ver 3 ejemplos de la ejecución del predicado principal ‘lights’, hay un ejemplo de un tablero de 2x2, otro de 3x3 y otro de 4x4.

El ejemplo de 5x5 no lo presentamos ya que tarda demasiado tiempo en dar los resultados.

Como podemos ver en la ejecución del código, en el ejemplo de 4x4 nos da las 16 alternativas diferentes para resolver el juego.

```
1 ?- lights(2, [casilla(1,2),casilla(2,1),casilla(2,2)], R).
R = [casilla(2, 2)] ;

No
2 ?- lights(3, [casilla(1,2),casilla(2,1),casilla(2,2)], R).
R = [casilla(1, 2), casilla(1, 3), casilla(2, 1), casilla(2, 2), casilla(3, 1)] ;

No
3 ?- lights(4, [casilla(2,1),casilla(2,2),casilla(3,1),casilla(3,3),casilla(3,4),casilla(4,3),casilla(4,4)], R).
R = [casilla(3, 1), casilla(3, 2), casilla(4, 1), casilla(4, 4)] ;
R = [casilla(1, 4), casilla(2, 3), casilla(2, 4), casilla(3, 1), casilla(3, 4), casilla(4, 2), casilla(4, 3), casilla(4, 4)] ;
R = [casilla(1, 3), casilla(2, 2), casilla(2, 3), casilla(2, 4), casilla(3, 2), casilla(4, 3)] ;
R = [casilla(1, 3), casilla(1, 4), casilla(2, 2), casilla(3, 4), casilla(4, 1), casilla(4, 2)] ;
R = [casilla(1, 2), casilla(2, 1), casilla(2, 2), casilla(2, 3), casilla(3, 1), casilla(3, 2), casilla(3, 4), casilla(4, 2)] ;
R = [casilla(1, 2), casilla(1, 4), casilla(2, 1), casilla(2, 2), casilla(2, 4), casilla(3, 1), casilla(4, 1), casilla(4, 3)] ;
R = [casilla(1, 2), casilla(1, 3), casilla(2, 1), casilla(2, 4), casilla(3, 2), casilla(3, 4), casilla(4, 1), casilla(4, 2), c
R = [casilla(1, 2), casilla(1, 3), casilla(1, 4), casilla(2, 1), casilla(2, 3), casilla(4, 4)] ;
R = [casilla(1, 1), casilla(2, 1), casilla(2, 2), casilla(3, 2), casilla(3, 3), casilla(4, 1), casilla(4, 2), casilla(4, 3)] ;
R = [casilla(1, 1), casilla(1, 4), casilla(2, 1), casilla(2, 2), casilla(2, 3), casilla(2, 4), casilla(3, 3), casilla(3, 4)] ;
R = [casilla(1, 1), casilla(1, 3), casilla(2, 1), casilla(2, 3), casilla(2, 4), casilla(3, 1), casilla(3, 2), casilla(3, 3), c
R = [casilla(1, 1), casilla(1, 3), casilla(1, 4), casilla(2, 1), casilla(3, 1), casilla(3, 3), casilla(3, 4), casilla(4, 1), c
R = [casilla(1, 1), casilla(1, 2), casilla(2, 3), casilla(3, 2), casilla(3, 3), casilla(3, 4), casilla(4, 3), casilla(4, 4)] ;
R = [casilla(1, 1), casilla(1, 2), casilla(1, 4), casilla(2, 4), casilla(3, 3), casilla(4, 1), casilla(4, 2), casilla(4, 4)] ;
R = [casilla(1, 1), casilla(1, 2), casilla(1, 3), casilla(2, 2), casilla(2, 4), casilla(3, 1), casilla(3, 2), casilla(3, 3), c
R = [casilla(1, 1), casilla(1, 2), casilla(1, 3), casilla(1, 4), casilla(2, 2), casilla(2, 3), casilla(3, 1), casilla(3, 3), c
No
```

NOTA: Los resultados de la ejecución del predicado ‘lights’ en un tablero de 4x4 no aparecen enteros en la imagen, ya que la hemos recortado para poder mostrar mejor la ejecución del programa.

REFERENCIAS

- <http://www.anarkasis.com/rafa/homees.htm>. Esta Web es la que hemos usado para probar el predicado, comparando los resultados que nos dan los algoritmos que usa con los resultados que nos da nuestro predicado.
- http://en.wikipedia.org/wiki/Lights_Out_%28game%29. En esta Web nos hemos documentado acerca de como funciona el juego y de posibles algoritmos para resolverlo.