



DEPARTAMENTO DE INFORMÁTICA  
UNIVERSIDAD CARLOS III DE MADRID

# Grado en Ingeniería Informática

## Inteligencia Artificial

Febrero 2017. 1ª examen

### Normas generales del examen

- El tiempo para realizar el examen es de **1,5 horas**
- Si se sale del aula, no se podrá volver a entrar durante el examen
- No se puede presentar el examen escrito a lápiz

### Problema 1. (5 puntos)

Se tiene un juego que consiste en un grid de bombillas 4x4. Cuando el juego empieza, un número aleatorio de bombillas están apagadas. Encender una bombilla apagada cambia su estado y el estado de la bombilla adyacente a la derecha. En el borde derecho del grid, que no hay bombillas adyacentes a la derecha, al encender una bombilla también se cambia el estado de la bombilla adyacente arriba en caso de que exista. La meta es encender todas las bombillas, preferiblemente presionando el menor número de veces posible. Considerar que sólo se presionan bombillas apagadas.

Si se quiere modelar el problema utilizando Sistemas de Producción, **se pide:**

1. (1,5 punto) Diseñar qué se debería representar en la base de hechos o memoria de trabajo y poner un ejemplo.
2. (1,5 punto) Definir una regla para encender cualquier bombilla que no esté en el borde derecho.
3. (1 puntos) Definir la regla de parada del sistema de producción.
4. (1 puntos) Razonar si las dos reglas anteriores son suficientes para que el motor de inferencia pueda encontrar eventualmente la solución a cualquier problema del juego. En caso contrario explicar brevemente qué otras reglas faltarían especificando la prioridad de cada regla. ¿Encontraría soluciones óptimas?

### Problema 2. (5 puntos)

Si se quiere resolver el problema anterior utilizando algoritmos de búsqueda, **se pide:**

1. (1 punto) Definir alguna heurística admisible.
2. (2 punto) Dado el estado inicial de la Figura 1. Dibujar el árbol de búsqueda que generaría A\* al expandir los **4 primeros nodos** (además del nodo inicial), indicando el orden de los nodos generados y expandidos y los valores de  $g$  y  $h$  de todos los nodos generados. Para la generación de sucesores considerar el orden de menor a mayor fila primero y luego de menor a mayor columna. En caso de empate expandir el nodo con menor valor de  $h$ , si sigue el empate expandir el nodo antes generado. Suponer que se definen dos operadores  $E1(X)$  y  $E2(X)$ , donde  $X$  es la bombilla que se presiona (tiene que estar apagada),  $E1$  es para las bombillas que no están en el borde derecho y  $E2$  para las del borde. Los nodos repetidos con mayor o igual valor de  $g$  se pueden omitir.
3. (1 puntos) Razonar cuál es el factor de ramificación máximo y mínimo para el problema del punto anterior.
4. (1 puntos) Explicar cuál es el tamaño del espacio de estados.

1.)

	1	2	3	4
1	x	z		x
2				
3				z
4				x

Solo se pulsian bombillas apagadas.

st: estado  $\in \{on, off\}$

x: pos x  $0 \leq x < 5$

y: pos y  $0 \leq y < 5$

1. MT: casilla (x, y, st)

inv(on, off) inv(off, on)

2. <sup>Relax:</sup>

casilla(x, y, off), casilla(x+1, y, t)  $\rightarrow$  modify casilla(x, y, on)  
inv(t, v) modify casilla(x+1, y, v)

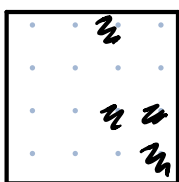
3.  $\sim$  casilla(x, y, off)  $\rightarrow$  STOP, halt

4. Faltan reglas, una para el borde y otra para la esquina. En cuanto a la prioridad la de fin será la primera, en cuanto se cumpla termine.

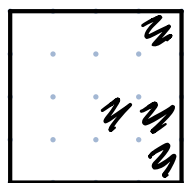
2)

1. h(n): número de bombillas apagadas en el estado n.

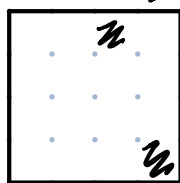
2.



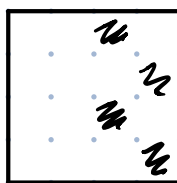
0  
4  
4



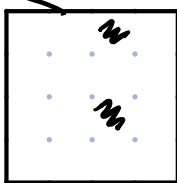
1 4 5



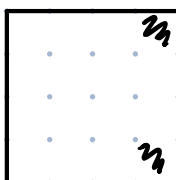
1 2 3



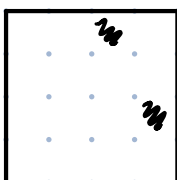
1 4 5



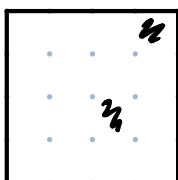
1 2 3



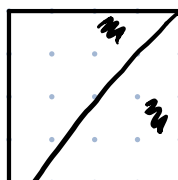
2 2 4



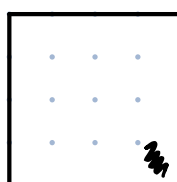
2 2 4



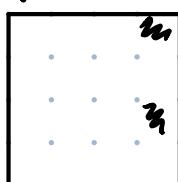
2 2 4



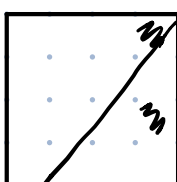
2 2 4



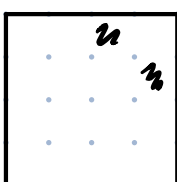
3 1 4



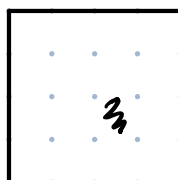
3 2 5



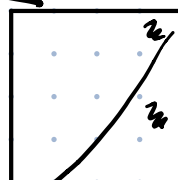
3 2 5



3 2 5

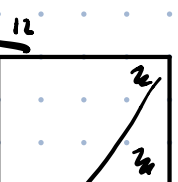


3 1 4



3 2 5

Repetido = cost



3 2 5

Repe

En 4 par 9 termina de leer

3. El factor de ramificación máximo es 4 y el mínimo 1  
las 4 iniciales apagadas es el máximo de apagados posible. de los que a un único posible estado. Cuando se pulsa el último estado ya llendo a fin
4. Es un tablero de  $4 \times 4$  con 2 posibles estados cada casilla, por lo tanto  $2^{16}$  estados

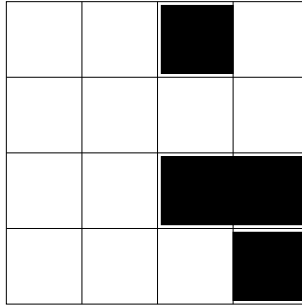


Figura 1: Las casillas negras representan bombillas apagadas y las blancas las encendidas.

## Solucion Problema 1

1. Para representar este problema hacen falta dos predicados:

inverso(X, Y): representa que el inverso de X es Y, los posibles valores de las variables X e Y son {off,on}

casilla(X,Y,Z): representa que la bombilla de la casilla de la fila X y columna Y está en el estado Z. X e Y pueden valer {1,2,3,4} y Z {off,on}

En la BH siempre tiene que haber dos hechos con el predicado inverso, inverso(on,off) e inverso(off,on) y 16 hechos con el predicado casilla que indica el estado de la bombilla de cada casilla. Ejemplo:

```
(inverso off on) (inverso on off)
(casilla 1 1 off) (casilla 1 2 off) (casilla 1 3 off) (casilla 1 4 off)
(casilla 2 1 off) (casilla 2 2 off) (casilla 2 3 off) (casilla 2 4 off)
(casilla 3 1 off) (casilla 3 2 off) (casilla 3 3 off) (casilla 3 4 off)
(casilla 4 1 off) (casilla 4 2 off) (casilla 4 3 off) (casilla 4 4 off)
```

2. (defrule encender2

```
?c1 <- (casilla ?x ?y off)
?c2 <- (casilla ?x ?y1 ?es)
(inverso ?es ?new)
(test (= ?y1 (+ ?y 1)))
=>
(retract ?c1 ?c2)
(assert (casilla ?x ?y on))
(assert (casilla ?x1 ?y1 ?new))
(printout t "Enciende (" ?x ", " ?y ") junto con (" ?x ", " ?y1 ")" crlf ))
```

3. (defrule fin

```
(declare (salience 100))
(not (casilla ?x ?y off))
=>
(halt)
(printout t " SOLUCION " crlf))
```

4. Faltarían 2 reglas, una para encender la bombilla superior derecha y otra regla para encender las otras bombillas del borde. Tiene que tener menos prioridad que la de fin pero no está claro que tenga mas o menos prioridad que la de encender2 anterior. No asegura encontrar soluciones óptimas, con ninguna estrategia de resolución del conjunto conflicto para todos los problemas.

## Solución problema 2

1. Número de bombillas apagadas / 2
2. La Figura muestra el árbol pedido

3. El factor de ramificación de un nodo es el número de sucesores que tiene. Para este problema el máximo es 4 porque solo se pueden pulsar bombillas apagadas y como mucho habrá 4. El mínimo es 1 que ocurrirá, por ejemplo, cuando se expanda el nodo 9. El algoritmo A\* termina cuando saque de abierta un nodo meta, en este caso, cuando todas las bombillas estén encendidas y no necesita generar sus sucesores.
4. Hay 16 bombillas que puede estar en dos estados (on/off), por tanto el tamaño del espacio de búsqueda es  $2^{16}$

