Grado en Ingeniería Informática

Inteligencia Artificial

Febrero 2019. Primer examen parcial

Departamento de Informática
Universidad Carlos III de Madrid

Normas generales del examen

- El tiempo para realizar el examen es de **2 horas**
- No se responderá a ninguna pregunta sobre el examen
- 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. (2 puntos)

Dado el siguiente conjunto de reglas:

R1: IF p(X,Y) $\wedge$ n(Y,Z) $\wedge$ f(Z) $\wedge$ e(X,m) THEN p(X,Z) $\wedge$ $\sim$p(X,Y) $\wedge$ $\sim$f(Z)

R2: IF p(X,Y) $\wedge$ n(Y,Z) $\wedge$ $\sim$f(Z) THEN p(X,Z) $\wedge$ $\sim$p(X,Y)

Donde el símbolo $\sim$ significa borrar de la base de hechos, las mayúsculas representan variables y las minúsculas constantes. El contenido inicial de la base de hechos es: e(b1,h) e(b2,m) p(b1,h1) p(b2,h1) n(h1,h2) n(h2,h3) f(h3)

Se pide: Mostrar la ejecución del sistema de producción detallando los hechos de la memoria de trabajo y el conjunto conflicto para cada ciclo de ejecución, siguiendo una estrategia de resolución del conjunto conflicto LIFO (profundidad o último en entrar primero en salir).

Problema 2. (5 puntos)

Considere el siguiente conjunto de problemas en el que un jugador comienza en la posición más a la izquierda de una secuencia de N casillas, que pueden ser blancas, negras o rayadas. La longitud de la secuencia y el tipo de las casillas depende del problema concreto que se esté tratando de resolver. El objetivo es alcanzar cuanto antes la posición de más a la derecha. Cuando el jugador está situado en una casilla blanca puede desplazarse 1 ó 2 casillas a la derecha. Cuando el jugador está situado en una casilla negra, puede desplazarse 1 ó 4 casillas a la derecha. Cuando el jugador cae en una casilla rayada, automáticamente es desplazado a la primera casilla hacia la izquierda que no sea rayada.

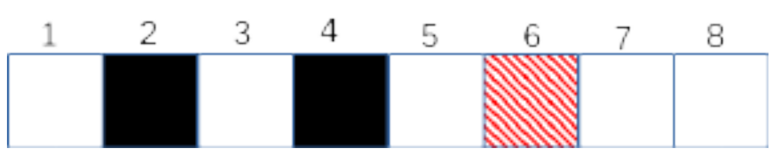


Figura 1: Ejemplo de configuración de la secuencia de casillas.

Se pide:

1. (1 punto) Defina el espacio de estados para el conjunto de problemas definido.
2. (1 punto) Defina el estado inicial y el/los estados meta.
3. (1 punto) Defina los operadores para el conjunto de problemas definido, incluyendo sus condiciones de aplicabilidad y su resultado. ¿Cuál es el factor de ramificación? ¿Por qué?
4. (1 punto) Defina una heurística admisible.
5. (1 punto) Muestre la ejecución de el algoritmo A* para el problema definido en la Figura 1, indicando toda la información necesaria para poder interpretarla.

1.)

MT	Conjunto Conflicto	Ejecuta
e(b1,h) e(b2,m) p(b1,h1) ~~p(b2,h1)~~ n(h1,h2) n(h2,h3) f(h3)	R1(X=b2, Y=h1, Z=h3) × R2(X=b1, Y=h1, Z=h2) R2(X=b2, Y=h1, Z=h2) →	p(b2,h2)
e(b1,h) e(b2,m) p(b1,h1) n(h1,h2) n(h2,h3) ~~f(h3)~~ ~~p(b2,h2)~~	R2(X=b1, Y=h1, Z=h2) R1(X=b2, Y=h2, Z=h3) →	p(b2,h3)
~~p(b1,h1)~~ e(b1,h) e(b2,m) n(h1,h2) n(h2,h3) p(b2,h3)	R2(X=b1, Y=h1, Z=h2) →	p(b1,h2)
e(b1,h) e(b2,m) n(h1,h2) n(h2,h3) p(b2,h3) ~~p(b1,h2)~~	R2(X=b1, Y=h2, Z=h3) →	p(b1,h3)
e(b1,h) e(b2,m) n(h1,h2) n(h2,h3) p(b2,h3) p(b1,h3)	No hay más	

2)

1. BH: casilla (x, stat) x es la posición de la casilla
stat es el tipo de casilla. stat ∈ ('b','n','r')

desp ('b', 1) desp('n', 1) } 'x' color casilla
desp('b', 2) desp('n', 4) } n nº de casillas

b = blanca
n = negra
r = rallada

mypos (x) x posición en la que nos encontramos

2. Est. inicial: mypos (0)

Est. final: mypos (N) / N es la longitud de la secuencia de casillas.

3. Despl1, Despl2, Despl4 y Volver.

Volver: mypos(x), casilla (x, 'r') → mypos (0)

Despl1: mypos (x), casilla (x, S), S ∈ (b, n) → modify mypos (x+1)
$x+1 \leq N$

Despl2: mypos(x), casilla (x, 'b') → modify mypos (x+2)
$x+2 \leq N$

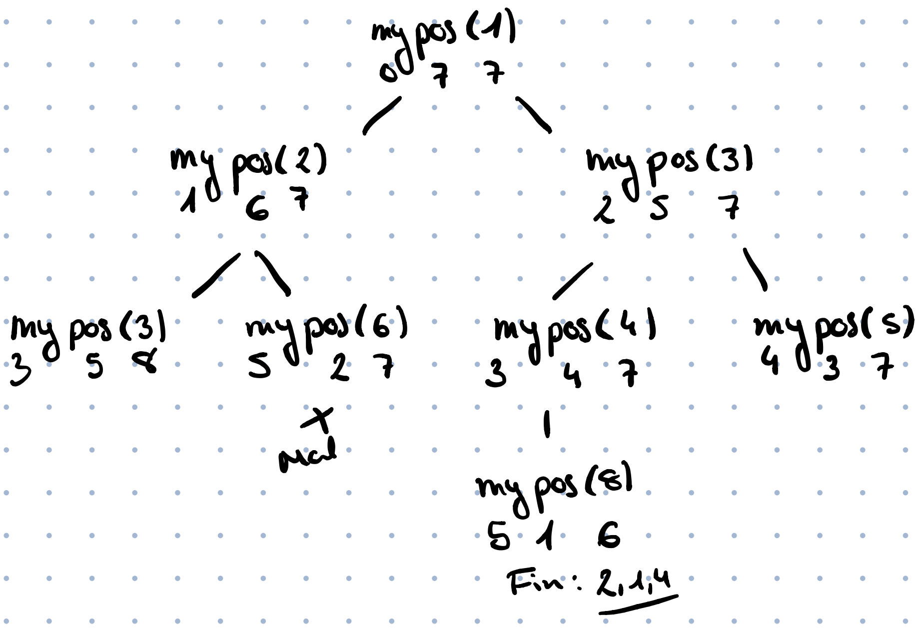
Despl4: mypos(x), casilla (x,'b') → modify mypos (x+4)
$x+4 \leq N$

Factor de ramificaciónes 2 para b y n, cada uno puede provocar 2 estados y volver es 1

4. h(x): casillas desde x hasta N

5. El coste de despl. g(x) = 1 si es negra, 2 si blanca y 4 si rayada.

Prioriza el que menos se desplaza.



Problema 3. (3 puntos)

Se quiere definir un sistema de producción (SP) para simular las operaciones de rescate de los bomberos cuando se produce un incendio en una planta de un edificio suponiendo lo siguiente:

- Los bomberos persiguen dos objetivos: apagar incendios y rescatar víctimas.
- Cada bombero puede llevar solo un tipo de equipación, que puede ser: un hacha para echar abajo puertas cerradas con llave, una manguera para apagar el fuego, o un equipo de reanimación (oxigeno y mascarilla)
- La planta del edificio incendiada se compone de distintas habitaciones conectadas entre sí, esta conexión puede ser diafana (puerta abierta o echada abajo), cerrada pero sin la llave echada, o cerrada con llave.
- Un bombero equipado con manguera situado en una habitación con un fuego vivo puede apagarlo. Solo entrarán en una habitación en llamas los bomberos con manguera
- Un bombero con hacha y una puerta cerrada con llave, puede echarla abajo
- Un bombero equipado con oxígeno y mascarilla situado en una habitación con una victima, puede reanimarla evitando que fallezca

Se pide:

1. (1 punto) Defina los hechos de la base de hechos para representar un edificio con 4 habitaciones consecutivas (la 1 conecta con la 2, la 2 con la 3 y la 3 con la 4), con 3 bomberos situados en la primera habitación, uno tiene un hacha, otro una manguera y el tercero el equipo de reanimación. La puerta entre la habitación 1 y 2 está cerrada sin llave, pero las otras puertas están cerradas con llave. Hay un fuego en la tercera habitación y una víctima en la última
2. (2 punto) Defina las reglas del SP.

Sala(X, Y, Z, V)
X posición sala.
Y abierta ∈ (si, no, abrible)
Z fuego ∈ (si, no)
V victima ∈ (si, no)

3.) bomb(X, Y, Z)
X nom. bombero ∈ (1,3)
Y posición bombero ∈ (1,4)
Z con lo que carga ∈ (hacha, manguera, kit)

inv(si, no) inv(no, si)
Inversos de las variables de los estados.

BH:
Sala(1, abrible, no, no) bomb(1, 1, hacha)
Sala(2, no, no, no) bomb(2, 1, manguera)
Sala(3, no, si, no) bomb(3, 1, kit)
Sala(4, no, no, si)

2.)
Fuego: Sala(x, y, si, v), bomb(2, x, manguera) → modify sala(x, y, no, v)

Puerta: Sala(x, no, no, z), bomb(1, x, hacha) → modify sala(x, si, no, z)

Sala(x, abrible, no, z), bomb(A, x, B) → modify sala(x, si, no, z)

Victima: sala(x, y, no, si), bomb(3, x, kit) → modify sala(x, y, no, no)

Entran y dentro ya lo van arreglando

Movimiento: Sala(x, si, no, no), bomb(y, x, z) → modify bomb(y, x+1, z)

Fin: ~bomb(x, y, z), y<3 $\xrightarrow{x<4}$ STOP

Solución Problema 1

$CC_0 = \{\underline{R2(X = b2, Y = h1, Z = h2)}, R2(X = b1, Y = h1, Z = h2)\}$
$BH_1 = \{e(b1, h), e(b2, m), p(b1, h1), p(b2, h2), n(h1, h2), n(h2, h3), f(h3)\}$
$CC_1 = \{\underline{R1(X = b2, Y = h2, Z = h3)}, R2(X = b1, Y = h1, Z = h2)\}$
$BH_2 = \{e(b1, h), e(b2, m), p(b1, h1), p(b2, h3), n(h1, h2), n(h2, h3)\}$
$CC_2 = \{\underline{R2(X = b1, Y = h1, Z = h2)}\}$
$BH_3 = \{e(b1, h), e(b2, m), p(b1, h2), p(b2, h3), n(h1, h2), n(h2, h3)\}$
$CC_3 = \{\underline{R2(X = b1, Y = h2, Z = h3)}\}$
$BH_4 = \{e(b1, h), e(b2, m), p(b1, h3), p(b2, h3), n(h1, h2), n(h2, h3)\}$
$CC_4 = \emptyset$

Solución Problema 2

1. Hay que representar la configuración de la secuencia y la posición del jugador. La configuración de la secuencia es estática, luego no se debe introducir en los estados de la búsqueda. Asumiremos que la configuración de la secuencia se representa mediante un vector denominado C de N posiciones con índices en $\{1, \ldots, N\}$ y valores en $\{blanco, negro, rayado\}$. Asumiremos que la primera y última casillas de la secuencia nunca son rayadas. N representa el tamaño de la secuencia. En el caso de la Figura 1, $N = 8$.

   Los estados están representados por la posición del jugador:

   $$EE = \{p \mid p \in \{1 \ldots N\} \; Y \; C_p \neq rayado\}$$

2. $EI = 1$, $EF = 8$

3. Operador $avanzar(x)$, $x \in \{1, 2, 4\}$

   - Condiciones de aplicabilidad (en estado s con posición p): [($C_p = blanco$ AND $x \neq 4$) OR ($C_p = negro$ AND $x \neq 2$)]
   - Resultado: aplicar el operador en un estado s con posición p genera un estado s' con posicion p', donde:

   $$p' = \begin{cases} 8 & \text{si } p + x > 8 \\ i & \text{si } p + x \leq 8 \text{ y } C_{p+x} = rayado \\ p + x & \text{en otro caso} \end{cases} \tag{1}$$

   Donde $i = \text{máx}\{j \mid j \in \{1 \ldots N\}, j < p + x, C_j \neq rayado\}$

   Asumiremos que si no hay suficientes casillas para realizar el movimiento el jugador se queda en la última casilla.

   El coste de los operadores es 1.

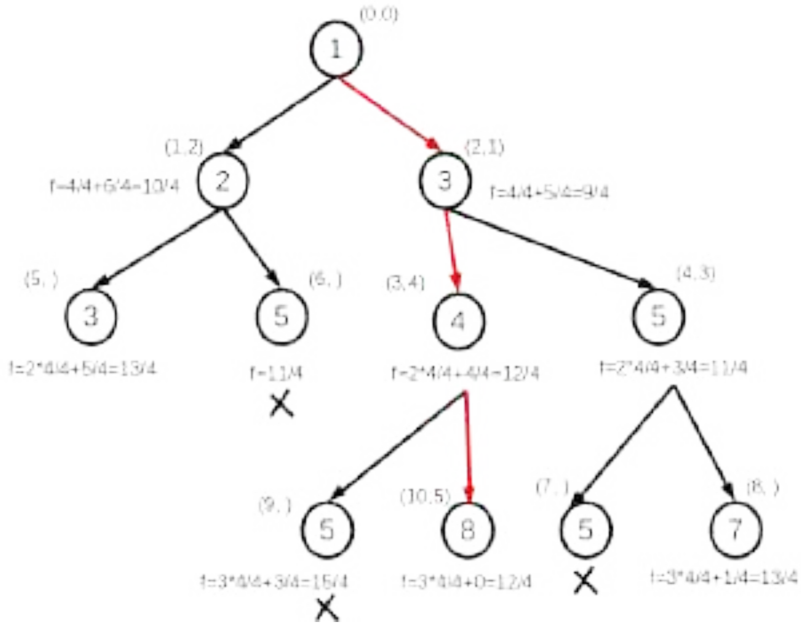
   El factor de ramificación es 2 porque en cada estado se pueden aplicar dos operadores.

4. Una heurística sencilla que es admisible sería:

   $$h(s) = \frac{N - p}{4}$$

   Para generar esta heurística estamos relajando el hecho de que haya casillas rojas y también el hecho de que haya casillas blancas y negras. La heurística cuenta lo que falta para llegar a la posición final considerando que en cada movimiento se pueden avanzar 4 casillas. Puesto que este es el número de casillas máximo que se puede avanzar en la realidad, la heurística es admisible.

5. El árbol de búsqueda generado por A* podría ser el que aparece a continuación. El camino solución se indica en color rojo: avanzar(2), avanzar(1), avanzar(4). Esta solución es óptima.



Solución Problema 3

```
R_avanzar: SI  posicion(B, H) Y conectada(H1, H2, diafana) Y  not fuego(H2))
           ENTONCES
             not posicion(B, H)
             posicion(B, H2)

R_avanzar_apagar: SI posicion(B, H) Y conectada(H1, H2, diafana) Y  fuego(H2)
                     Y equipacion(B, manguera)
                  ENTONCES
                    not posicion(B, H)
                    not fuego(H2)
                    posicion(B, H2)

R_abrir_puerta: SI posicion(B, H) Y conectada(H1, H2, cerrada)
                ENTONCES
                    not conectada(H1, H2, cerrada)
                    conectada(H1, H2, diafana)

R_tirar_puerta: SI posicion(B, H) Y conectada(H1, H2, llave) Y equipacion(B, hacha)
                ENTONCES
                  not conectada(H1, H2, llave)
                  conectada(H1, H2, diafana)

R_reanimar: SI posicion(B, H) Y conectada(H1, H2, diafana) Y  victima(H2)
               Y equipacion(B, reanimacion)
                  ENTONCES
                    not posicion(B, H)
                    not victima(H2)
                    posicion(B, H2)


BH = {equipacion(b1, hacha), equipacion(b2,manguera),  equipacion(b3,reanimacion),
      posicion(b1, h1), posicion(b2, h1), posicion(b3, h1), conectada(h1, h2, cerrada),
      conectada(h2, h3, llave), conectada(h3, h4, llave), victima(h4), fuego(h3)}
```