



DEPARTAMENTO DE INFORMÁTICA  
UNIVERSIDAD CARLOS III DE MADRID

# Grado en Ingeniería Informática

Inteligencia Artificial  
Febrero 2015. Problemas 1. Grupo 82

## Normas generales del examen

- El tiempo para realizar el examen es de **2 horas**
- Entregad cada problema en hojas separadas
- Solo se responderán preguntas sobre el examen los primeros 30 minutos
- 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)

En un sistema de producción se han introducido las siguientes reglas.

R1: IF elemento(X) THEN total(0)

R2: IF elemento(X), total(Y) THEN total(X+Y), elemento(X-1),  $\sim$ elemento(X),  $\sim$ total(Y)

R3: IF elemento(1) THEN  $\sim$ elemento(1), **halt**

El símbolo  $\sim$  significa borrar de la base de hechos y **halt** parar el sistema de producción. Las reglas están ordenadas de menor a mayor prioridad, es decir, en caso de que en el conjunto conflicto hubiese una instanciación de la R3 y otra de la R2 se ejecutará la instanciación de la R3. Del mismo modo, si hay una instanciación de la R1 y otra de la R2 se ejecutará la instanciación de la R2.

La base de hechos o memoria de trabajo inicial contiene: elemento(6).

Se pide:

1. (1,5 puntos) Mostrar la secuencia de reglas ejecutadas y los hechos de la memoria de trabajo para cada ciclo de ejecución para una estrategia de resolución del conjunto conflicto FIFO (amplitud o primero en entrar primero en salir).
2. (0,5 puntos) Si la base de hechos inicial contuviera el hecho elemento(10), ¿cual sería el contenido de la base de hechos al parar el sistema de producción?

## Problema 2. (3 puntos)

Se tienen dos jarras de agua: una de cinco litros de capacidad y otra de tres. Ninguna de ellas tiene marcas de medición. Se tiene un grifo que permite llenar las jarras de agua, también se pueden vaciar o traspasar el contenido de una jarra a otra. Si se traspasa, por ejemplo, el contenido de la jarra grande a la pequeña cuando la pequeña está vacía, la jarra grande acabará con dos litros y la pequeña se llenará. Si la pequeña contuviera ya un litro, la jarra grande acabaría con tres litros. Inicialmente las jarras están vacías y se desea que la jarra de cinco litros de capacidad contenga exactamente cuatro litros.

Suponiendo que se quiere resolver el problema utilizando sistemas de producción.

Se pide:

1. (1 punto) Definir la base de hechos o memoria de trabajo necesaria para representar el problema.
2. (2 punto) Definir las reglas del sistema de producción pedido.

### Problema 3. (2 puntos)

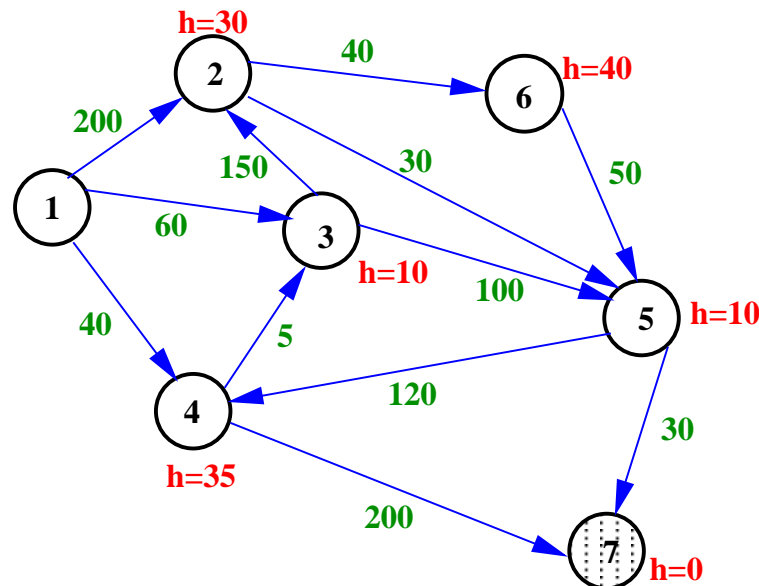
Suponer que en el problema anterior se considera que el coste de cada operador es el número de litros que se mueven, tanto del grifo a las jarras, de una jarra a otra o al vaciarlas.

Se pide:

1. (1 punto) Definir el espacio de problema.
2. (1 punto) Definir alguna heurística admisible para que el algoritmo A\* sea capaz de encontrar la solución óptima, es decir, la de menor coste según la definición de coste anterior.

### Problema 4. (3 puntos)

Dado el siguiente grafo donde se quiere ir desde el nodo 1 hasta el nodo 7 y el orden de generación de sucesores es del nodo de menor número a mayor.



Se pide:

1. (0,5 puntos) Pintar el árbol de búsqueda que generaría el algoritmo en amplitud para resolver el problema, indicando el orden de los nodos generados y expandidos. Decir las características del algoritmo (completitud, admisibilidad y eficiencia).
2. (0,5 puntos) Pintar el árbol de búsqueda que generaría el algoritmo en profundidad para resolver el problema, indicando el orden de los nodos generados y expandidos. Decir las características del algoritmo (completitud, admisibilidad y eficiencia).
3. (0,5 puntos) Decir cual es el factor de ramificación medio.
4. (1,5 puntos) Pintar el árbol de búsqueda que generaría el algoritmo A\* indicando el orden de los nodos generados y expandidos. Decir las características del algoritmo (completitud, admisibilidad y eficiencia).

## Solución al problema 1

1. La memoria de trabajo y el conjunto conflicto ordenado por prioridad de las reglas y por estrategia FIFO en cada ciclo son:

$MT_0 = \{\text{elemento}(6)\}$   
 $CC_0 = \{(R1, X=6)\}$   
 $MT_1 = \{\text{elemento}(6), \text{total}(0)\}$   
 $CC_1 = \{(R2, X=6, Y=0)\}$   
 $MT_2 = \{\text{elemento}(5), \text{total}(6)\}$   
 $CC_2 = \{(R2, X=5, Y=6) (R1, X=5)\}$   
 $MT_3 = \{\text{elemento}(4), \text{total}(11)\}$   
 $CC_3 = \{(R2, X=4, Y=11) (R1, X=4)\}$   
 $MT_4 = \{\text{elemento}(3), \text{total}(15)\}$   
 $CC_4 = \{(R2, X=3, Y=15) (R1, X=3)\}$   
 $MT_5 = \{\text{elemento}(2), \text{total}(18)\}$   
 $CC_5 = \{(R2, X=2, Y=18) (R1, X=2)\}$   
 $MT_6 = \{\text{elemento}(1), \text{total}(20)\}$   
 $CC_6 = \{(R3) (R2, X=1, Y=20) (R1, X=1)\}$   
 $MT_7 = \{\text{total}(20)\}$

2.  $\text{total}(10+9+8+7+6+5+4+3+2)$

## Solución al problema 2

1. Utilizamos los siguientes predicados:

- $\text{contenido}(X, Y)$ : el contenido de la jarra X es Y
- $\text{capacidad}(X, Y)$ : la capacidad de la jarra X es Y

La base de hechos inicial será:  $MT_o = \{\text{contenido}(j1, 0), \text{capacidad}(j1, 5), \text{contenido}(j2, 0), \text{capacidad}(j2, 3)\}$

2. Las reglas serán:

R1: regla para llenar una de las jarras

SI  $\text{contenido}(X, Y), \text{capacidad}(X, Z), Y < Z$  ENTONCES  $\text{contenido}(X, Z), \sim \text{contenido}(X, Y)$

R2: regla para vaciar una de las jarras

SI  $\text{contenido}(X, Y), Y > 0$  ENTONCES  $\text{contenido}(X, 0), \sim \text{contenido}(X, Y)$

R3: regla para traspasar si cabe de X1 a X2

SI  $\text{contenido}(X1, Y1), \text{contenido}(X2, Y2), \text{capacidad}(X2, Z2), X1 \neq X2, Y1 > 0, Z2 \geq (Y1 + Y2)$  ENTONCES  $\text{contenido}(X1, 0), \text{contenido}(X2, Y2 + Y1), \sim \text{contenido}(X1, Y1), \sim \text{contenido}(X2, Y2)$

R4: regla para traspasar si no cabe de X1 a X2

SI  $\text{contenido}(X1, Y1), \text{contenido}(X2, Y2), \text{capacidad}(X2, Z2), X1 \neq X2, Y1 > 0, Z2 < (Y1 + Y2)$  ENTONCES  $\text{contenido}(X1, Y1 - (Z2 - Y2)), \text{contenido}(X2, Z2), \sim \text{contenido}(X1, Y1), \sim \text{contenido}(X2, Y2)$

## Solución al problema 3

1. Hay que definir los estados, operadores, estado inicial y final.

- Estados: se pueden definir por dos variables (X, Y), X representa los litros que contiene la jarra de 5 litros e Y los litros que tiene la jarra de 3
- El estado inicial será: (0, 0)
- El estado final será: (4, Y)

2. Operadores: se podrían definir los siguientes operadores

- VaciarX: vaciar la jarra grande. SI  $X > 0$  ENTONCES  $(0, Y)$ ,  $\text{coste} = X$
- VaciarY: vaciar la jarra pequeña. SI  $Y > 0$  ENTONCES  $(X, 0)$ ,  $\text{coste} = Y$
- LlenarX: llenar la jarra grande. SI  $X < 5$  ENTONCES  $(5, Y)$ ,  $\text{coste} = 5 - X$
- LlenarY: llenar la jarra pequeña. SI  $Y < 3$  ENTONCES  $(X, 3)$ ,  $\text{coste} = 3 - Y$
- VerterenX: vuelca el contenido de la jarra pequeña en la jarra grande.  
SI  $Y > 0, X < 5$  ENTONCES  $(X + \min\{5 - X, Y\}, Y - \min\{5 - X, Y\})$ ,  $\text{coste} = \min\{5 - X, Y\}$
- VerterenY: vuelca el contenido de la jarra grande en la jarra pequeña.  
SI  $X > 0, Y < 3$  ENTONCES  $(X - \min\{X, 3 - Y\}, Y + \min\{X, 3 - Y\})$ ,  $\text{coste} = \min\{X, 3 - Y\}$

3. Heurística admisible:  $h((X, Y)) = |4 - X|$  (litros que faltan para conseguir la meta)

## Solución al problema 4

Los nodos repetidos no se pintan. En el caso del A\*, si el nodo repetido tiene mejor función de evaluación que el anterior, se pone una cruz roja para indicar que ese nodo ya no se explora, si la función de evaluación es mayor no se pinta en el árbol.

1. El árbol se muestra en la figura 1. En este caso el algoritmo en amplitud es completo pero no es admisible por no tener costes uniformes. La complejidad espacial y temporal es exponencial.
2. El árbol se muestra en la figura 2. En este caso el algoritmo en profundidad es completo si se fija la profundidad como el n° de nodos del grafo, no es admisible. La complejidad espacial es lineal y la temporal exponencial.
3. El factor de ramificación medio es  $(3 + 2 + 2 + 2 + 2 + 1) / 7$
4. El árbol se muestra en la figura 3. El algoritmo A\* es completo, admisible y la memoria espacial y temporal es exponencial.

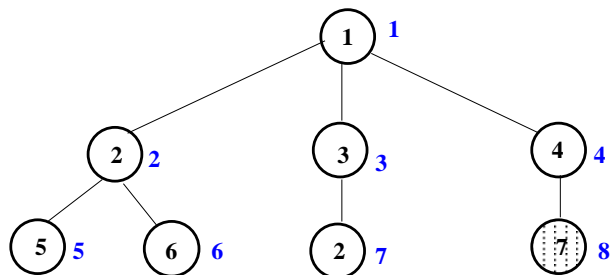


Figura 1: Árbol de búsqueda generado por el algoritmo Amplitud

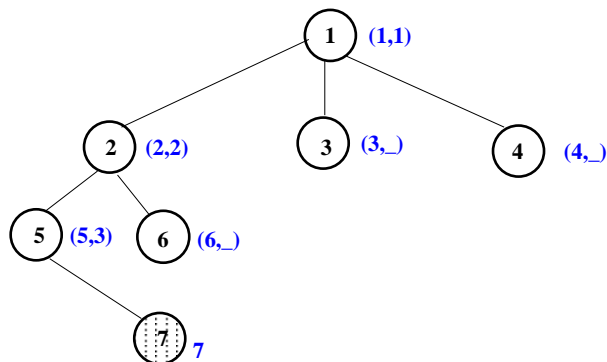


Figura 2: Árbol de búsqueda generado por el algoritmo Profundidad

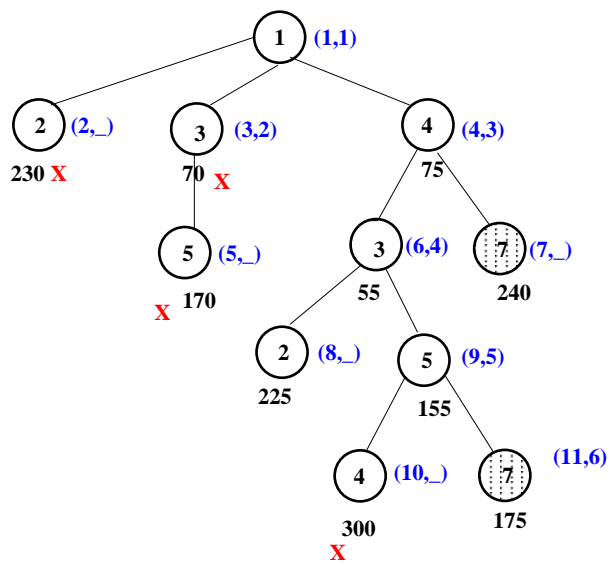


Figura 3: Árbol de búsqueda generado por el algoritmo A\*