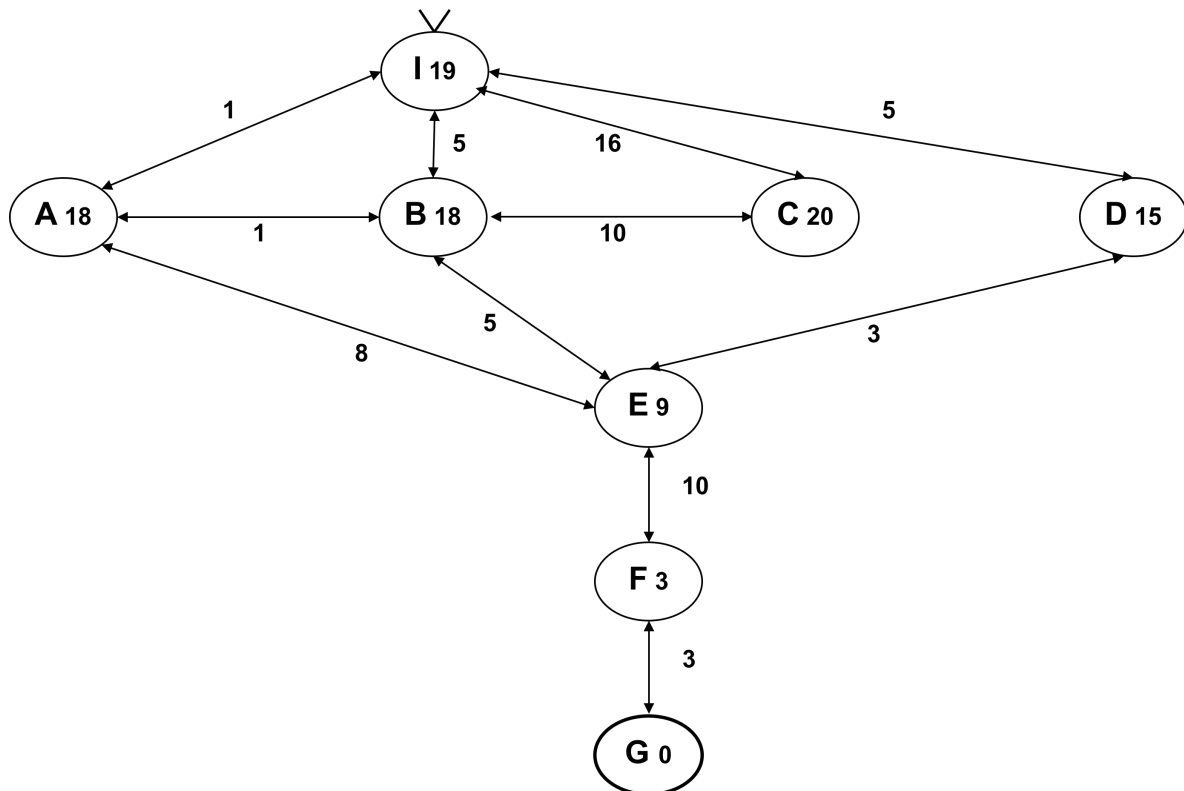


# Examen final de Inteligencia Artificial (Primer Parcial)

(Grado en Ingeniería Informática, 22/6/2015)

## Ejercicio 1 (4 puntos)

Dado el espacio de estados representado en la figura, consideramos que cada nodo representa un estado, siendo I el estado inicial y G el estado objetivo. El número indicado en cada arista representa su coste y el número que hay dentro de cada nodo corresponde al valor de la heurística  $h'$  para ese estado.



Se pide:

- Describir el comportamiento del algoritmo A\* con GraphSearch para este espacio de estados con una tabla que represente la evolución de la estructura Abiertos (Frontera), indicando para cada nodo su nodo padre y los valores de  $g$ ,  $h'$  y  $f'$ . Indicar también los nodos almacenados en la estructura Cerrados por orden de expansión, el camino encontrado y el coste de la solución. Supóngase que los hijos de un nodo se generan siempre por orden alfabético.
- ¿Es admisible la heurística utilizada en este problema? ¿Por qué? ¿Es consistente? ¿Por qué? ¿El camino obtenido es óptimo? ¿Por qué? ¿Cómo cambiarías la heurística para mejorar el comportamiento de A\*?
- Si aplicáramos el algoritmo de primero en anchura, evitando la repetición de estados, ¿qué coste tendría la solución encontrada en este caso y por qué estados pasaría?

- d) Si aplicáramos el algoritmo de escalada por máxima pendiente, ¿qué coste tendría la solución encontrada en este caso y por qué estados pasaría?

## **Ejercicio 2** (5 puntos)

Considérese una versión miniatura del puzle de 8, conocida como el puzle de 3, con los siguientes estados inicial y objetivo:

<u>Inicial</u>	<u>Objetivo</u>
- 1	1 2
3 2	- 3

Se asume que el hueco puede moverse hacia la derecha, hacia la izquierda, hacia arriba y hacia abajo, siempre y cuando se mantenga dentro de los límites del puzle. Se pide:

- Representar este problema en Prolog, según el paradigma del espacio de estados, eligiendo una representación adecuada para estados y operadores.
- Construir un programa en Prolog que determine si es posible alcanzar el estado final desde el inicial y devuelva la secuencia de operaciones necesarias, evitando la repetición de estados.

## **Cuestión** (1 punto)

¿Para qué sirve el algoritmo RETE? ¿En qué se basa?