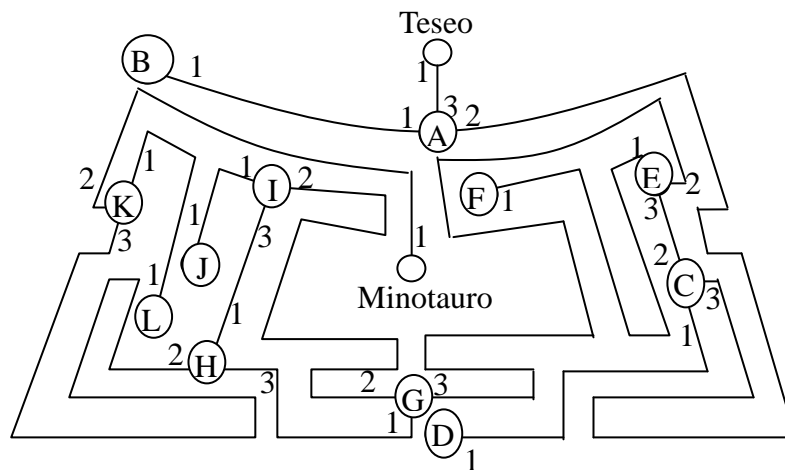


Inteligencia Artificial. Examen de problemas. Enero 2011.

Problema 1. Búsqueda no informada. El laberinto del Minotauro (5 puntos)



Asterio, el llamado Minotauro, tenía cara de toro y el resto de hombre. El rey Minos lo encerró en el laberinto y le puso vigilancia. El laberinto, construido por Dédalo, era una prisión que a base de intrincados corredores burlaba la salida. (...) Teseo, hijo de Egeo rey de Atenas, fue escogido como tributo para el Minotauro. Cuando llegó a Creta, Ariadna, la hija de Minos, se enamoró de él y se ofreció a ayudarlo si prometía llevarla a Atenas y hacerla su mujer. Y habiéndolo prometido Teseo bajo juramento, rogó ella a Dédalo que le revelara la salida del laberinto. Entonces por consejo de aquél, le dio a Teseo, cuando entraba en el laberinto un hilo: Teseo lo ató a la puerta y arrastrándolo tras de sí iba entrando. Cuando encontró al Minotauro en la parte extrema del laberinto, lo mató golpeándolo con sus puños; y recogiendo el hilo, salió. (Apolodoro, Biblioteca, III, 11 y Epítome, I, 8-9).



Actualizando este fragmento de mitología griega es necesario programar un Teseo que sea capaz de calcular el camino más directo desde su posición actual hasta la del Minotauro considerando el laberinto anterior. Interpreta las líneas como pasillos y los círculos (etiquetados con letras) como lugares singulares en el trayecto (intersecciones o finales de trayecto). En estos lugares singulares, Teseo decidirá el pasillo a seguir, tomando los pasillos en orden creciente de la numeración del pasillo que aparece en cada entrada de pasillo que da al lugar singular correspondiente. Antes de hacer la programación se solicita:

1. La formalización del problema de búsqueda, definiendo con claridad (de manera formal) el conjunto de estados.
2. Dibujad el árbol de búsqueda con cada nodo dentro de un cuadrado. Etiquetad las aristas del árbol con la acción que lleva de un estado a otro. Indicad las repeticiones de nodo mediante un cuadrado con contorno de doble línea. Subrayad los nombres de los estados cuyo trayecto no conduzca a Teseo de forma directa hasta el Minotauro. Utilizad doble subrayado para resaltar el nombre del/de los estado/s final/es.
3. Indicad la secuencia de estados que recorrerá el algoritmo de búsqueda en grafos primero en profundidad (o profundidad prioritaria). ¿Qué solución retornará?
4. Indicad la secuencia de estados que recorrerá el algoritmo de búsqueda en grafos primero en anchura (amplitud prioritaria). ¿Qué solución retornará?

Problema 2. Procesos de decisión de Markov. Juego de cartas solitario. (5 puntos)

Se pretende jugar a un juego de cartas donde se utiliza un mazo que cuenta con tres tipos de cartas diferentes (el uno, el dos y el tres). El juego consiste en llegar a sumar cinco tirando sobre el mantel la cantidad de cartas que se consideren oportunas. Se puede asumir que el mazo de cartas es infinito y que la proporción de cartas que hay en el mazo es tal que hay el doble de doses que de unos y que de treses (la cantidad de treses es la misma que la de unos). De esta forma, la proporción de cartas que tenemos es la siguiente:



El/la jugador/a empieza el juego tirando una carta y deberá decidir en cada momento si continúa tirando una carta más o si se planta (con lo que el juego termina), teniendo en cuenta que:

- Si se planta sin haber llegado a sumar 5, los puntos que ganará será la de la suma de las cartas que haya sobre el mantel. Así, en la siguiente secuencia de cartas de ejemplo obtendría, al acabar el juego, un total de 4 puntos.



- Si el/la jugador/a llega a tirar un total de cartas cuya suma supere los 5 puntos, perderá y el juego se dará por terminado. De este modo, en la siguiente secuencia de ejemplo, el/la jugador/a perderá.



- Y por último, si la tirada de cartas sobre el mantel suma exactamente 5, entonces el/la jugador/a ganará un total de 10 puntos.



1. Formaliza el problema como un proceso de decisión de Markov, identificando de forma clara el conjunto de estados, las acciones disponibles en cada uno de los estados, la función de transición y la función de recompensa.
2. Ejecuta hasta 4 iteraciones del algoritmo de iteración de valores y muestra los Q-valores y los valores de cada estado. Teniendo en cuenta que en los cálculos asumiremos un factor de descuento $\gamma = 1$ y que no se pide que se utilicen más de dos decimales ¿Cuál es la política óptima de juego? (si no te da tiempo puedes ejecutar 3 iteraciones).
3. Partiendo de la política anterior, ejecuta el algoritmo de iteración de políticas e indica qué sucede. No es necesario que hagas más de 2 iteraciones en tu explicación, tampoco se pide mucha precisión en los valores y seguiremos considerando $\gamma = 1$.