Departamento de Informática
Universidad Carlos III de Madrid

Ingeniería Informática

Inteligencia Artificial

Prueba de evaluación

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. (5 puntos)

¡Fernando Alonso desea ganar el próximo premio de Fórmula 1! Y para ello, ha decidido contar con la ayuda de los alumnos de la Universidad Carlos III de Madrid, para crear un sistema de simulación inteligente de carreras, tanto en el modo contrarreloj como en carrera individual contra otro piloto. En una primera aproximación, se supone que el circuito de carreras está constituido por celdas del mismo tamaño entre las que se mueven los vehículos si éstas están desocupadas, o contra las que pueden estrellarse, si aparecen rayadas en negro. Además, con el propósito de tener en cuenta el efecto de la aceleración sobre los vehículos, se considera que un coche puede pasar de las coordenadas (x_0, y_0) a una cualquiera de las posiciones del cuadrado inscrito entre las coordenadas $(x_r, y_r) = (x_0+d_x-2, y_0+d_y-2)$ y $(x_R, y_R) = (x_0+d_x+2, y_0+d_y+2)$, donde (d_x, d_y) es el vector de desplazamiento que llevó al monoplaza hasta la posición (x_0, y_0). Obviamente, $(d_x, d_y) = (0, 0)$ al inicio de la carrera para cualquier coche.

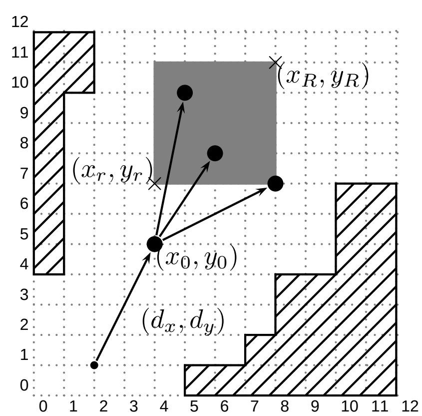


Figura 1: Modelo discreto de desplazamiento de coches

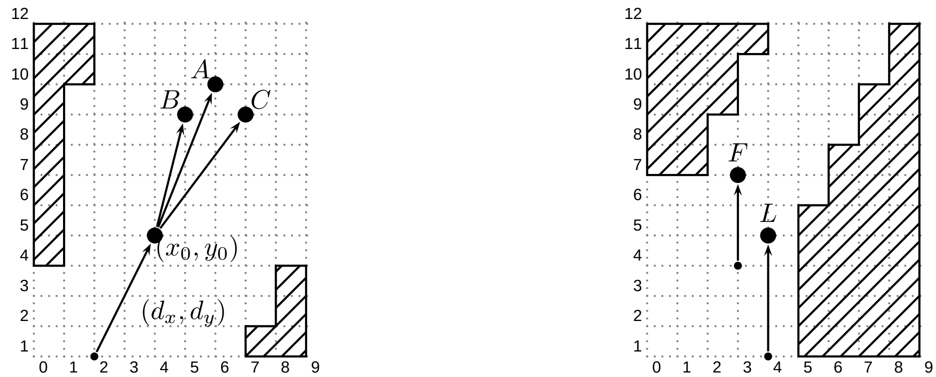
Para la aplicación contrarreloj, se desea encontrar la mejor secuencia de movimientos que lleven al monoplaza desde la línea de salida hasta la línea de meta **en el menor tiempo posible** (esto es, movimientos), en un circuito cualquiera descrito con celdas como las mencionadas anteriormente. Para ello, se pide:

1. **(0,5 puntos)** Formalizar el espacio de estados con el uso de marcos
2. **(0,5 puntos)** Modelizar los operadores del dominio

4. **(0,5 puntos)** Sin considerar los obstáculos, ¿cuáles son el valor mínimo y máximo del factor de ramificación en el árbol de búsqueda desarrollado a una profundidad arbitraria d?

5. **(1 punto)** Definir una función heurística $h(n)$ que sea *admisible* e *informada* explicando claramente su construcción

Para la construcción del sistema de simulación de carreras contra un único piloto, se asume que ambos jugadores toman decisiones alternadamente, decidiendo en cada turno cuál es la posición que ocuparán. Por otra parte, el cálculo de colisiones se ha simplificado notablemente, considerando que ambos monoplazas chocan sólo si después de que uno se mueve, llega a la misma posición que ocupa el otro. Por último, el conjunto de posiciones siguientes para esta segunda aplicación se ha restringido significativamente, y ahora sólo puede pasarse a una cualquiera de las tres posiciones siguientes: $A(x_0 + d_x + 0, y_0 + d_y + 1)$, $B(x_0 + d_x - 1, y_0 + d_y)$ y $C(x_0 + d_x + 1, y_0 + d_y)$, tal y como se muestra en la figura 2(a)



(a) Posiciones permitidas en el modo de carrera

(b) Fernando Alonso (F) seguido por Lewis Hamilton (L)

Figura 2: Definiciones para el modo de carrera contra otro piloto

Se pide:

6. **(1 punto)** Aplicar el algoritmo MINIMAX a profundidad 2 al caso de la figura 2(b) para Fernando Alonso, usando una función de evaluación que devuelve $-\infty$ si el vehículo de Fernando Alonso (señalado con una F) choca y el de Lewis Hamilton (etiquetado con L) no; $+\infty$ si choca el de Hamilton y no el de Fernando; 0 si ambos chocan y, por último, la diferencia en Y entre el coche de Fernando Alonso y el de Lewis Hamilton en cualquier otro caso.

   Indicar claramente el valor MINIMAX del nodo raíz, así como la secuencia elegida de movimientos. Interpreta el resultado, ¿qué significa?

7. **(1 punto)** Definir una función de evaluación más realista que la anterior explicando claramente su construcción

Departamento de Informática
Universidad Carlos III de Madrid

Ingeniería Informática

Inteligencia Artificial

Prueba de evaluación

Normas generales del examen

- El tiempo para realizar el examen es de **1.5 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 2. (5 puntos)

La matrioska o matrioska son unas muñecas tradicionales rusas creadas en 1890, cuya originalidad consiste en que se encuentran huecas por dentro, de tal manera que en su interior albergan una nueva muñeca, y ésta a su vez a otra, y ésta a su vez otra, en un número variable que puede ir desde cinco hasta el número que se desee, aunque es raro que pasen de veinte. Se caracterizan por ser multicolores, o por la presencia de elementos decorativos en la pintura tales como jarrones o recipientes sostenidos por las muñecas, pero en la práctica pueden identificarse por su tamaño.

Las matrioskas pueden verse como un juego. Las matrioskas pueden estar abiertas (el cuerpo separado de la cabeza), o cerradas. Una matrioska puede contener a otra matrioska. Los estados inicial y final pueden ser cualquier configuración en la que todas las matrioskas están: (i) o bien cerradas en la mesa y/o unas dentro de otras, (ii) o bien abiertas encima de la mesa. Por tanto, unas matrioskas que inicialmente tienen la configuración que se muestra en la Figura 2 no tendrían una configuración correcta, puesto que hay unas matrioskas abiertas dentro de otras. Una matrioska sólo se puede abrir si está encima de la mesa. Una matrioska cerrada que está encima de la mesa puede meterse en otra si la segunda está abierta, encima de la mesa y es más grande que la primera.

1. (**0,5 puntos**) Formalizar el enunciado como un dominio de planificación. Se valorará que la formalización dada sea fácilmente escalable a cualquier número de matrioskas.
2. (**0,5 puntos**) Formalizar un problema de 5 matrioskas. En el estado inicial, la más grande está cerrada y vacía encima de la mesa, y las demás están todas cerradas dentro de la de tamaño inmediatamente superior de la que le corresponde (excepto para la cuarta más grande que está encima de la mesa, puesto que la más grande está vacía). En su configuración final están todas cerradas y dentro de la de tamaño inmediatamente superior.
3. (**1,5 puntos**) Obtener un plan que resuelva el problema anterior, y justificar la solución obtenida. Para justificar el plan obtenido, es necesario elegir un planificador y desarrollar formalmente el árbol de planificación.
4. (**0,5 puntos**) ¿Es el plan obtenido óptimo? ¿Garantiza el planificador utilizado que la solución sea óptima?
5. (**0,5 puntos**) ¿Qué literales habría que añadir al estado inicial si agregamos al problema 2 matrioskas más?
6. (**1,5 puntos**) ¿Cuál es el tamaño del espacio de estados para n matrioskas, si asumimos que los estados inicial y meta pueden ser cualquiera (que cumpla las especificaciones del dominio)?

Figura 3: Configuración incorrecta de Matrioskas