Desconecta4Boom

Memoria de la Práctica 3. Inteligencia artificial.

Esther García Gallego

2ºB Ingeniería Informática.

**Índice**

[1. ANÁLISIS DEL PROBLEMA 3](#_Toc74148763)

[2. SOLUCIÓN PLANTEADA 3](#_Toc74148764)

[2.1 Heurística 5](#_Toc74148765)

# ANÁLISIS DEL PROBLEMA

El problema planteado en esta práctica es la implementación de un agente deliberativo que juegue contra un adversario, siguiendo las técnicas de búsqueda como el algoritmo MiniMax o la Poda Alfa-Beta.

En este caso, nuestro entorno es un tablero de siete filas y siete columnas, en el que deben ir colocándose fichas. En cada turno, los jugadores (verde y azul) podrán colocar dos fichas de su color en cualquier columna del tablero que se encuentre disponible, excepto en la primera jugada, que constará de una única ficha verde (por ser éste el jugador que inicia el juego).

Además de fichas normales, hemos de notar la existencia de las fichas bomba. Éstas permiten al jugador eliminar las fichas del oponente que se encuentren en la misma fila mediante una “explosión”.

El juego concluye cuando alguno de los jugadores consigue hacer “cuatro en raya”, es decir, coloca cuatro fichas de su color en línea. Esto puede suceder de forma vertical, horizontal o diagonal. En caso de que se llene el tablero por completo, se considera un empate.

# SOLUCIÓN PLANTEADA

Como solución, se ha implementado el algoritmo de Poda Alfa-Beta siguiendo el esquema mostrado en el código 1, aunque se han incluido algunas modificaciones para una mejor legibilidad y una menor cantidad de código.

**función** alfa-beta(nodo //en nuestro caso el tablero, profundidad, α, β, jugador)

**si** nodo es un nodo terminal **o** profundidad = 0

**devolver** el valor heurístico del nodo

**si** jugador1

**para cada** hijo de nodo

α := max(α, alfa-beta(hijo, profundidad-1, α, β, jugador2))

**si** β≤α

**romper** *(\* poda β \*)*

**devolver** α

**si no**

**para cada** hijo de nodo

β := min(β, alfa-beta(hijo, profundidad-1, α, β, jugador1))

**si** β≤α

**romper** *(\* poda α \*)*

**devolver** β

Código 1. Esquema de la poda Alfa-Beta.

En primer lugar, se comprueba si se ha llegado a la profundidad límite, establecida por la variable PROFUNDIDAD\_ALFABETA que se proporciona por parámetros o si se ha alcanzado un nodo terminal (lo que daría por finalizado el juego).

Ilustración 1. Condición de parada.

Posteriormente, se inicia un bucle que concluye cuando todos los hijos han sido explorados o cuando beta es menor o igual a alfa. Dentro del mismo se calcula el valor del nodo mediante una llamada recursiva al método de poda. Seguidamente, comprueba si el nodo a explorar es MIN o MAX, es decir, si el jugador actual es el introducido por parámetro o es el oponente, respectivamente.

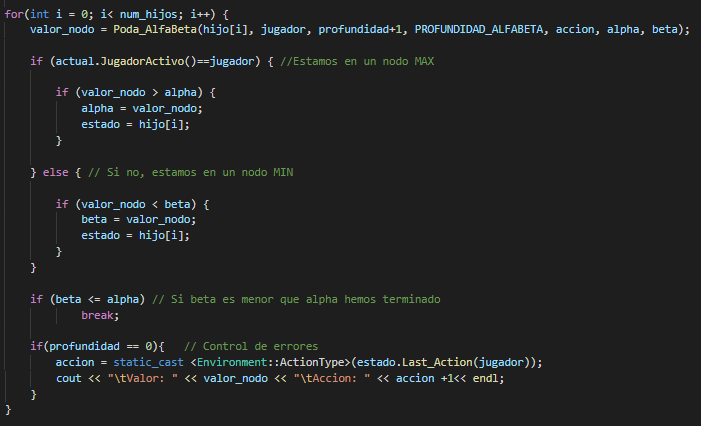
En el primer caso, si el valor del nodo es mayor a alfa, se actualizará tanto el valor almacenado en la variable Alpha. Si el que juega es el oponente y se determina que el valor del nodo es más pequeño que beta, se actualizará beta. En ambos casos se debe modificar la variable estado al estado del hijo actual.

Ilustración 2. Implementación del bucle.

Como podemos observar, he incluido en el método un pequeño punto de control de errores que me permite visualizar el valor del nodo escogido y la acción que representa el mismo. Como las acciones van de 0 a 7, siendo la última la de explotar la bomba, he sumado una unidad para obtener directamente la columna en la que se va a colocar la ficha.

## Heurística

La función heurística utilizada en este caso es muy sencilla.

