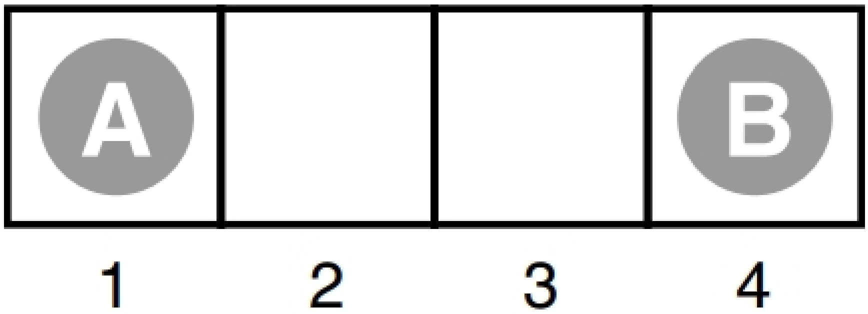
Práctico 2: Juegos

Ejercicio 1

Para este juego se utiliza una fila de 4 casillas numeradas del 1 al 4. Inicialmente, una ficha A se encuentra en la casilla 1 y una ficha B en la casilla 4, como se muestra a continuación.

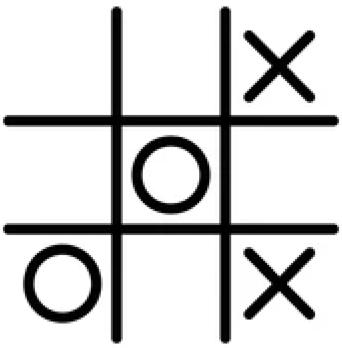


Los jugadores A y B mueven por turno, comenzando por el jugador A. En su turno, el jugador mueve su ficha a una casilla vacía adyacente en una u otra dirección. Si el oponente está ocupando una casilla adyacente, entonces el jugador puede saltar sobre el oponente a la siguiente casilla vacía si existe (por ejemplo, si A está sobre 3 y B está sobre 2, entonces A puede mover hacia atrás a 1 o hacia adelante a 4). El juego se termina cuando un jugador alcanza el extremo opuesto del tablero. Si el jugador A alcanza primero la casilla 4, el valor del juego para A es 1; si el jugador B alcanza primero la casilla 1, entonces el valor del juego para A es -1.

1. Formular este juego. Escribir cada estado como (i_A, i_B, p), donde $i_A \in \{1, 2, 3, 4\}$ es la posición de la ficha A, $i_B \in \{1, 2, 3, 4\}$ la de B y $p \in \{A, B\}$ indica el turno de juego.
2. Dibujar el árbol de juego completo, usando las siguientes convenciones:
   - Poner cada estado terminal en una caja cuadrada y escribir su utilidad en un círculo.
   - Poner los estados repetidos (estados que ya aparecen sobre el camino a la raíz) en cajas cuadradas dobles. Ya que no está claro cómo definir la utilidad en estados repetidos, anotarlo con un "?" en un círculo.
3. Encontrar el valor minimax de cada nodo del árbol y anotarlo en un círculo. Explicar cómo manejar los valores "?" y por qué.
4. Explicar por qué el algoritmo minimax estándar fallaría sobre este árbol de juego y mencionar brevemente cómo arreglarlo a partir de la respuesta anterior. ¿El algoritmo modificado proporciona las decisiones óptimas para todos los juegos con estados repetidos?
5. Determinar la decisión óptima para este mismo juego pero sobre una fila de 5 casillas (A empieza en la 1 y B en la 5).

Ejercicio 2

Considerar el juego del ta-te-ti. Para $n \in \{1, 2, 3\}$, definimos X_n como el número de filas, columnas o diagonales con exactamente n X's y ningún O. Del mismo modo, O_n es el número de filas, columnas o diagonales con exactamente n O's y ninguna X. Por ejemplo, para el siguiente estado s



se tiene $X_1(s) = 1$, $X_2(s) = 1$, $X_3(s) = 0$, $O_1(s) = 3$, $O_2(s) = 0$ y $O_3(s) = 0$. La función de utilidad asigna 1 a cualquier estado s con $X_3(s) = 1$ y -1 a cualquier estado con $O_3(s) = 1$. Todos los otros estados terminales tienen utilidad 0. Para estados no terminales, usamos una función de evaluación definida como $3X_2(s)+X_1(s) - (3O_2(s)+O_1(s))$.

1. Mostrar el árbol de juego entero hasta la profundidad 2 (es decir, una X y un O sobre el tablero), comenzando con un tablero vacío, teniendo en cuenta las simetrías.
2. Señalar sobre el árbol las evaluaciones de todos los nodos a profundidad 2.
3. Usando el algoritmo minimax con evaluación, marcar sobre el árbol los valores minimax para los nodos de profundidades 1 y 0, y use esos valores para elegir la mejor estrategia.