

# Algoritmo Minimax

Jorge Baier

Departamento de Ciencia de la Computación  
Pontificia Universidad Católica de Chile

Santiago, Chile



- Entender el funcionamiento del algoritmo Minimax
- Analizar la complejidad y eficiencia del algoritmo
- Discutir aplicaciones prácticas en juegos de dos jugadores



# Juegos con adversario

- En las clases anteriores hemos visto estrategias para encontrar soluciones a problemas de búsqueda que se pueden expresar por medio de un grafo. Estos procedimientos son muy útiles para resolver problemas de juegos solitarios, o con un jugador.
- Sin embargo, es difícil aplicar directamente estos procedimientos a juegos en los que dos jugadores intervienen y compiten entre sí.
- En los juegos solitarios los estados solo pueden alterarse por las acciones del jugador, en los juegos con adversario hay un contricante que realiza cambios en el estado que no podemos controlar a priori.



## Definición

El algoritmo Minimax es una técnica de decisión utilizada para minimizar la posible pérdida máxima propia o maximizar la mínima ganancia del contrincante en juegos con adversarios.

- Aplicado comúnmente en juegos de tablero como ajedrez, damas, etc.
- Consiste en simular todas las posibles movidas del juego y sus consecuencias futuras en el tablero.

## Objetivo

Elegir el mejor movimiento para ti suponiendo que tu contrincante escogerá el peor movimiento para ti.



# Funcionamiento del Minimax

1. Se construye un **árbol de juego** donde cada nodo representa un estado del juego. A partir del nodo que representa el estado actual, se generan todos los nodos hasta llegar a un estado terminal.
2. Se calculan los valores de la **función de evaluación** para cada nodo terminal del árbol construido. La función de evaluación devuelve valores positivos para indicar buenas situaciones para el jugador que hace uso del algoritmo, y valores negativos para indicar buenas situaciones para el adversario.
3. Se evalúan los nodos superiores a partir del valor de los inferiores. Según si estos nodos pertenecen a un nivel **MAX** o un nivel **MIN**, se elegirán los valores mínimos y máximos representando los movimientos del jugador y del oponente.
4. Se repite el paso 3 hasta llegar al nodo superior.
5. Se selecciona la jugada-nodo directamente accesible desde el nodo actual que optimiza el valor de la evaluación.



- Si no podemos generar el árbol completo del juego, es posible aplicar el algoritmo sobre una sección del mismo, pero no podemos asegurar optimalidad haciendo esto.
- En el caso anterior, la función de evaluación, en vez de calcularse en los estados terminales, se calcula en las hojas del árbol generado



# Pseudocódigo Minimax

## Función Minimax

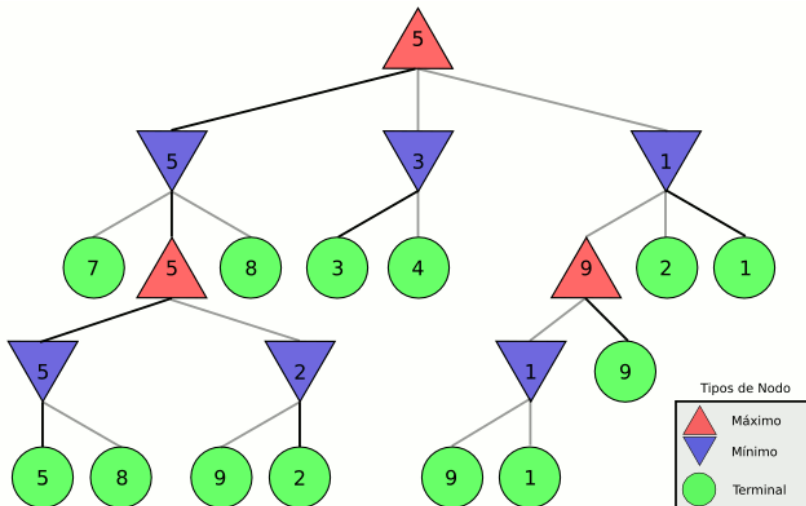
**Input:** Un estado  $S$  y un bool  $Maximizing$

**Output:** Valor de la evaluación de  $S$

1. **if**  $S$  es terminal **then**
2.     **return** Función\_de\_evaluación( $S$ )
3. **if**  $Maximizing = True$  **then**
4.      $v \leftarrow -\infty$
5.     **for each**  $hijo$  in  $S$  **do**
6.          $v \leftarrow Max(v, Minimax(hijo, False))$
7.     **return**  $v$
8. **else**
9.      $v \leftarrow +\infty$
10.     **for each**  $hijo$  in  $S$  **do**
11.          $v \leftarrow Min(v, Minimax(hijo, True))$
12.     **return**  $v$



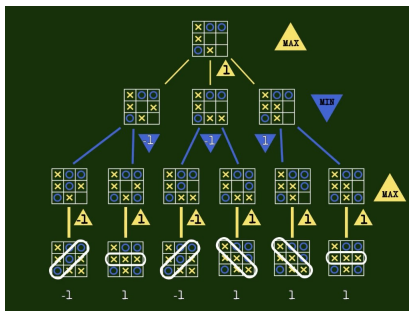
# Ejemplo





# Ejemplo Práctico: Juego del Gato

- Aplicación del Minimax para determinar la mejor jugada.
- Función de evaluación corresponde a asignarle un valor 1 a los estados donde el jugador gana y -1 a los estados donde el contrincante gana.



- Entender el funcionamiento del algoritmo Minimax
- Analizar la complejidad y eficiencia del algoritmo
- Discutir aplicaciones prácticas en juegos de dos jugadores

