

**SEP**



**SEIT**



**INSTITUTO TECNOLÓGICO  
De Nuevo Laredo**

**Especialidad:**

**Ingeniería en Sistemas  
Computacionales**

**Catedrático:**

**Ing. Bruno López Takeyas.**

**Asignatura:**

**Inteligencia Artificial.**

**Integrantes Equipo # 1:**

Javier Alonso Alvarado Rodríguez	01100160
Julissa Nereyda García Núñez	01100212
Wendy Hernández Santos	01100232
Norma Alicia Pimentel Vargas	01100281
Mario Tristán Martínez	01100313

**Tema:**

**Mínimax.**



**Nuevo Laredo, Tamps.**

**13 de Junio del 2005**

## INDICE

INTRODUCCION.....	3
DEFINICION.....	4
EJEMPLO DE LA ESTRUCTURA.....	5
CARACTERISTICAS GENERALES DEL ALGORITMO.....	6
ALGORITMO MINIMAX.....	7
PASOS DEL ALGORITMO.....	7
ALGORITMO MINIMAX.....	8
ALGORITMO MINIMAX CON BACKTRACKING.....	8
EJEMPLO.....	9
APLICACIONES.....	10
JUEGO TRES EN RAYA.....	15
BIBLIOGRAFIA.....	18

## MINIMAX.

### INTRODUCCIÒN.

En 1950, casi en cuanto fue posible programar las computadoras, Claude Shannon, inventor de la teoría de la información, y Alan Turing elaboraron los primeros programas de Inteligencia Artificial, basándose en el juego del ajedrez. Desde entonces, se han realizado continuos avances en el juego, al grado, de que los sistemas actuales son capaces de desafiar a campeones mundiales humanos sin temor de que puedan hacer el ridículo.

Los investigadores antes mencionados eligieron el ajedrez para su trabajo por varias razones. Una computadora capaz de jugar ajedrez sería la prueba viviente de una máquina que podía realizar algo para lo que se consideraba era necesario tener inteligencia. Además, la sencillez de las reglas, y el hecho de que el programa pueda acceder totalmente al estado del mundo significa que es fácil representar el juego como una búsqueda a través de un espacio de posibles posiciones de juego.

De hecho, la representación en computadora del juego es correcta, en todos sus detalles, a diferencia de la representación del problema de librar una batalla, por ejemplo: La presencia de un oponente complica más el problema de decisión. El oponente introduce la *incertidumbre*, puesto que es imposible saber qué va a hacer. El contrincante se esforzará todo lo que pueda porque sus jugadas sean lo menos benignas para su oponente.

La incertidumbre no es consecuencia de la información faltante, sino debido a que no hay tiempo suficiente para calcular las consecuencias exactas de una jugada. Sólo se intenta deducir lo que sería lo mejor con base en experiencias pasadas y proceder, sin estar completamente seguro de cuál sería la mejor acción.

### DEFINICIÓN:

El procedimiento de búsqueda Minimax es una búsqueda en profundidad (**DFS**) de profundidad limitada.

El nombre del algoritmo deriva de considerar que, dada una función estática que devuelve valores en relación al jugador maximizante, éste procura maximizar su valor mientras que su oponente procura minimizarlo. En un árbol de juego donde los valores de la función estática están en relación al jugador maximizante, se maximiza y minimiza alternadamente de un nivel a otro.

En **teoría de juegos** Minimax es un método de decisión para minimizar la pérdida máxima esperada en juegos con adversario y con información perfecta.

El funcionamiento de Minimax puede resumirse a como elegir el mejor movimiento para ti mismo; suponiendo que tu contrincante escogerá el peor para ti.

En la práctica el método Minimax es impracticable excepto en supuestos sencillos. Realizar la búsqueda completa requerirían cantidades excesivas de tiempo y memoria.

Claude Shannon en su texto sobre ajedrez de 1950 (*Programming a Computer for Playing Chess*) propuso limitar la profundidad de la búsqueda en el árbol de posibilidades y determinar su valor mediante una función heurística. Para optimizar Minimax puede limitarse la búsqueda por nivel de profundidad o por tiempo de ejecución. Otra posible técnica es el uso de la poda alfa-beta. Esta optimización se basa en la suposición que el jugador contrario no nos permitirá jugar nuestras mejores jugadas.



Dado que Minimax se basa en juegos y la habilidad de jugar es considerada como una distinción de inteligencia. Sus características son:

### **CARACTERÍSTICAS:**

- Fácil de crear situaciones complicadas con reglas sencillas.
- Se pueden probar contra humanos en donde existen escalas
- Son adictivos.
- A diferencia de búsqueda, el oponente introduce incertidumbre porque no sabemos que va a tirar, lo cuál se asemeja más a problemas reales.
- El algoritmo Minimax es un procedimiento recursivo, y el corte de la recursión está dado por alguna de las siguientes condiciones:
  - Gana algún jugador
  - Se han explorado N capas, siendo N el límite establecido
  - Se ha agotado el tiempo de exploración
  - Se ha llegado a una situación estática donde no hay grandes cambios de un nivel a otro.

### **ALGORITMO MINIMAX.**

#### **Pasos del algoritmo Minimax:**

1. Generación del árbol de juego. Se generarán todos los nodos hasta llegar a un estado terminal.
2. Cálculo de los valores de la función de utilidad para cada nodo terminal.
3. Calcular el valor de los nodos superiores a partir del valor de los inferiores. Alternativamente se elegirán los valores mínimos y máximos representando los movimientos del jugador y del oponente, de ahí el nombre de Minimax.
4. Elegir la jugada valorando los valores que han llegado al nivel superior.

El algoritmo explorará los nodos del árbol asignándoles un valor numérico mediante una función de utilidad, empezando por los nodos terminales y subiendo hacia la raíz. La función de utilidad definirá lo buena que es la posición para un jugador cuando la alcanza.

En el caso del juego del gato los posibles valores son (+1,0,-1) que se corresponden con ganar, empatar y perder respectivamente.

#### **En un juego tenemos:**

- Posición inicial.
- Conjunto de operadores (definen movidas legales).
- Estado terminal.
- Función de utilidad, ej. gana, pierde, empata (pero puede haber más, por ejemplo en backgammon). las reglas del juego determinan los posibles movimientos.

Se requiere de una estrategia que garantice llegar a estados terminales ganadores independientemente de lo que haga el oponente.

**Algoritmo Minimax:**

```

Para determinar el valor minimax de  $J$ :  $V(J)$ 
si  $J$  es terminal,  $V(J) \leftarrow ev(J)$ 
sino genera los sucesores de  $J$  :  $J_1, J_2, \dots, J_n$ 
    evalua  $V(J_1), V(J_2), \dots, V(J_n)$  de izquierda a derecha
    si  $J$  es nodo max,  $V(J) \leftarrow \max[V(J_1), \dots, V(J_n)]$ 
    si  $J$  es nodo min,  $V(J) \leftarrow \min[V(J_1), \dots, V(J_n)]$ 
    
```

**Algoritmo Minimax con *backtracking*.**

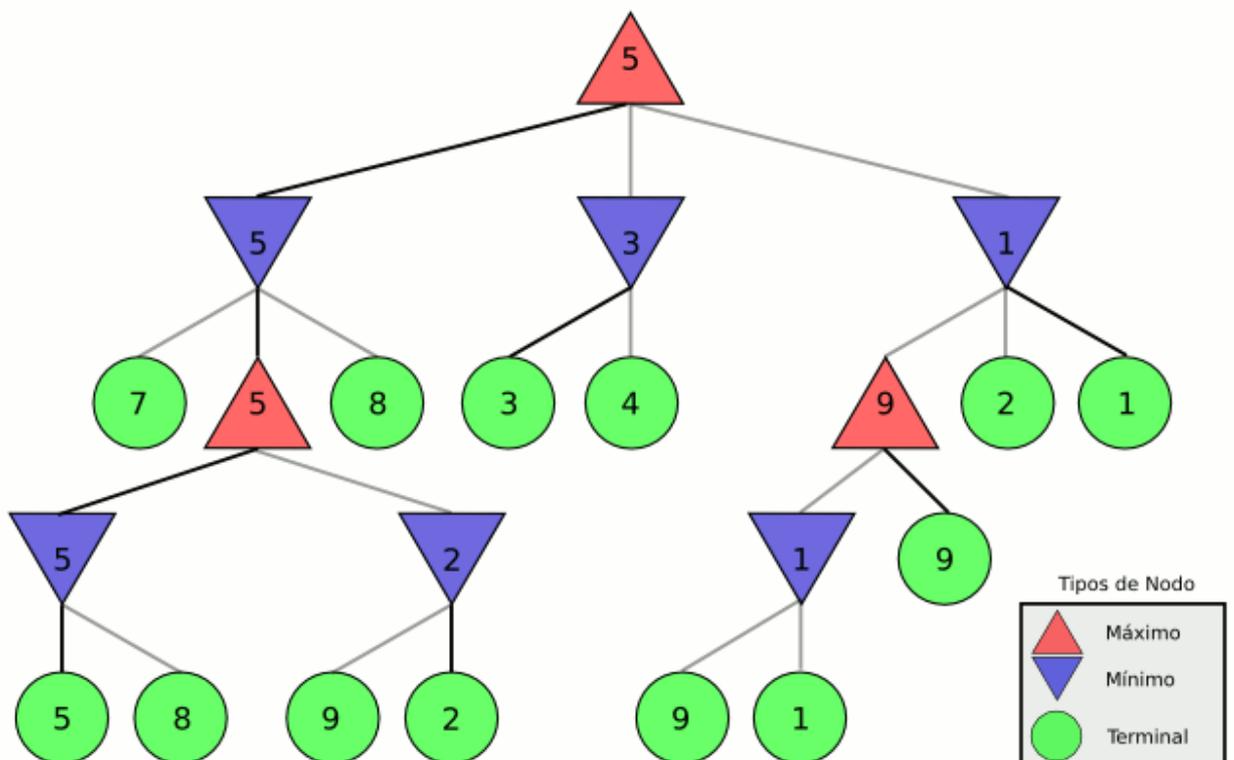
```

Algoritmo minimax (backtracking)
si  $J$  es terminal,  $V(J) \leftarrow ev(J)$ , sino
for  $k = 1, \dots, n$  do
    generar  $J_k$  (el  $k$ -esimo sucesor de  $J$ )
    evalua  $V(J_k)$ 
    si  $k = 1$ ,  $VA(J) \leftarrow V(J_1)$  sino
    para  $K \geq 2$ 
        si  $J$  es nodo max,  $VA(J) \leftarrow \max[VA(J), V(J_k)]$ 
        si  $J$  es nodo min,  $VA(J) \leftarrow \min[VA(J), V(J_k)]$ 
regresa  $V(J) \leftarrow VA(J)$ 
    
```

**EJEMPLO :**

En el siguiente ejemplo puede verse el funcionamiento de minimax en un árbol generado para un juego imaginario. Los posibles valores de la función de utilidad tienen un rango de [1-9]. En los movimientos del contrincante suponemos que escogerá los movimientos que minimicen nuestra utilidad, en nuestros movimientos suponemos que escogeremos los movimientos que maximizan nuestra utilidad.

El primer paso será calcular los nodos terminales, en verde. Posteriormente calcularemos el cuarto nivel, movimiento min, minimizando lo elegido (5, 2 y 1). Después podremos calcular el tercer nivel, movimiento max, maximizando la utilidad (5, 9). El segundo nivel es un movimiento min (5, 3 y 1). Finalmente llegamos al primer nivel, el movimiento actual, elegiremos el nodo que maximice nuestra utilidad (5).



## APLICACIONES.

MINIMAX
<p><b>Aplicación:</b> juegos de antagonismo</p> <p><b>Características</b> de este tipo de problemas:</p> <ul style="list-style-type: none"><li>• la acción la desarrollan <b>2 antagonistas</b></li><li>• el turno pasa <b>alternativamente de uno a otro</b></li><li>• la <b>información es completa</b>: cada antagonista conoce el estado del otro (ajedrez, damas, <del>dominó</del>, <del>cartas</del>)</li><li>• <b>juegos de suma nula</b>: lo que gana uno de los antagonistas es lo que pierde el otro</li><li>• <b>puede haber tablas o empate</b></li></ul>

MINIMAX
<ul style="list-style-type: none"><li>▪ El espacio de búsqueda se representa mediante <b>árboles alternados</b><ul style="list-style-type: none"><li>- <b>nodo</b>: representa una situación (del juego)</li><li>- <b>sucesores de un nodo</b>: situaciones a las que se puede acceder aplicando las reglas del juego</li><li>- <b>cada nivel</b>: contiene las situaciones posibles para uno de los antagonistas</li></ul></li><li>▪ Si primero juega el antagonista A, lo hará en los niveles pares (0,2,4, ...) del árbol alternado</li><li>▪ B lo hará en los niveles impares (1,3,5, ...) del árbol alternado</li></ul>

MINIMAX
<ul style="list-style-type: none"><li>▪ En juegos complejos (ajedrez, damas) se descarta una exploración total del espacio de búsqueda (explosión combinatoria)</li><li>▪ Se realiza una exploración parcial en torno a una situación dada (nodo)<ul style="list-style-type: none"><li>- explorar solo hasta una determinada profundidad</li><li>- evaluar una situación y sus sucesoras por medio de una función heurística</li></ul></li></ul>

## MINIMAX

---

### MINIMAX

Heurística :

- no garantiza el éxito
- el camino seleccionado es "razonablemente" un camino hacia la victoria o el empate
- imita el comportamiento humano al examinar por anticipado un pequeño número de jugadas antes de decidirse por una

### MINIMAX

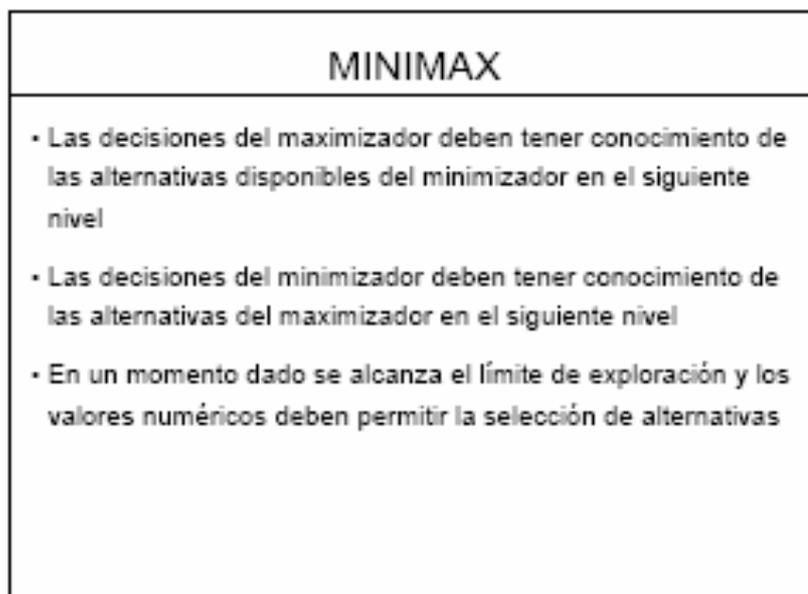
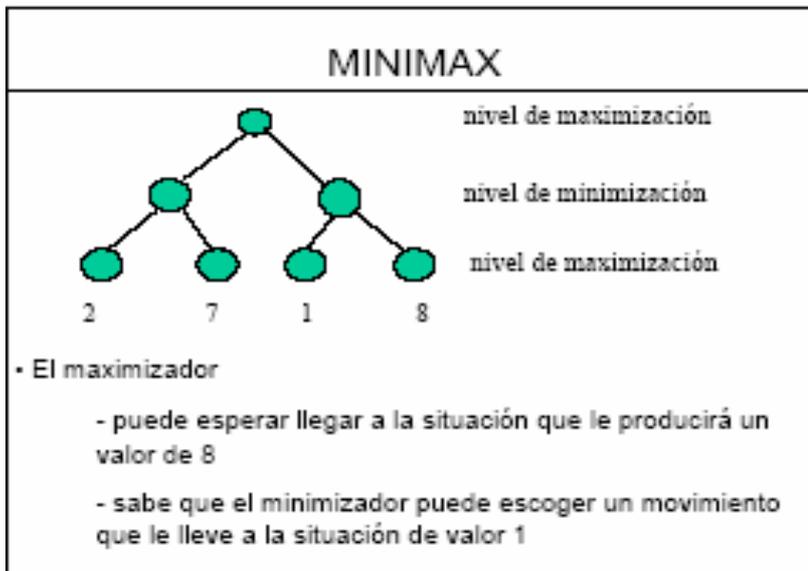
- Se dispone de un analizador de situaciones que las representa mediante un valor numérico simple
- Por convención, un valor numérico positivo indicará la ventaja de un jugador y uno negativo la ventaja del otro jugador
- El grado de ventaja entre jugadores aumentará con el valor absoluto del número
- El jugador que espera resultados positivos se conoce como jugador de maximización o maximizador
- El jugador que espera resultados negativos se conoce como jugador de minimización o minimizador

### MINIMAX

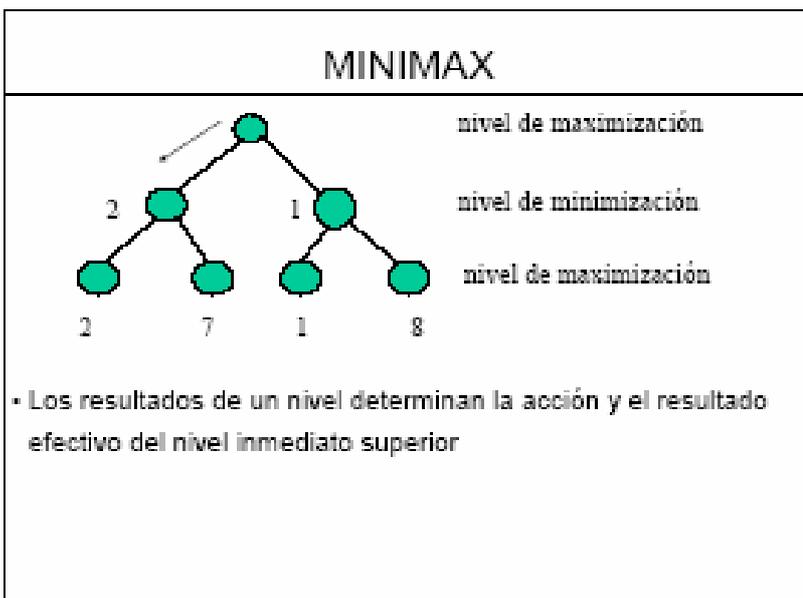
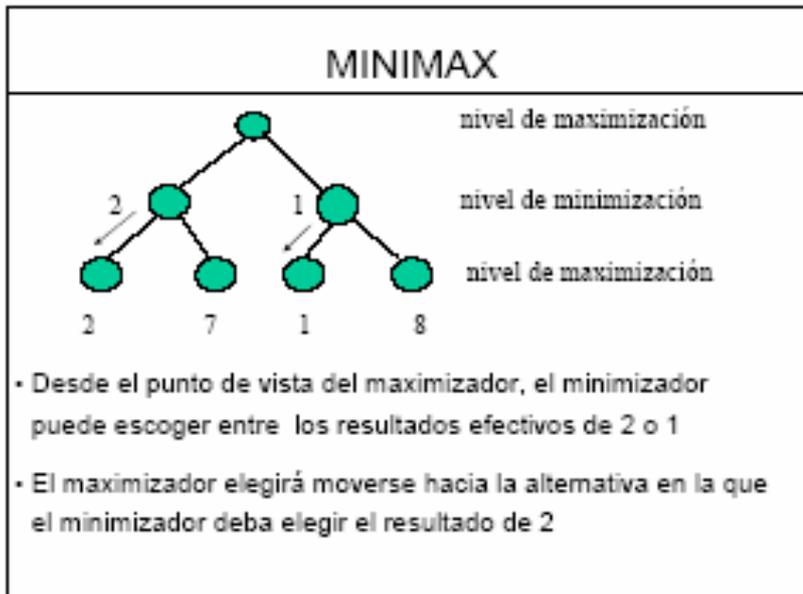
- El maximizador busca un movimiento que le lleve al mayor número positivo
- El minimizador busca un movimiento que le lleve al menor número negativo

## MINIMAX

---

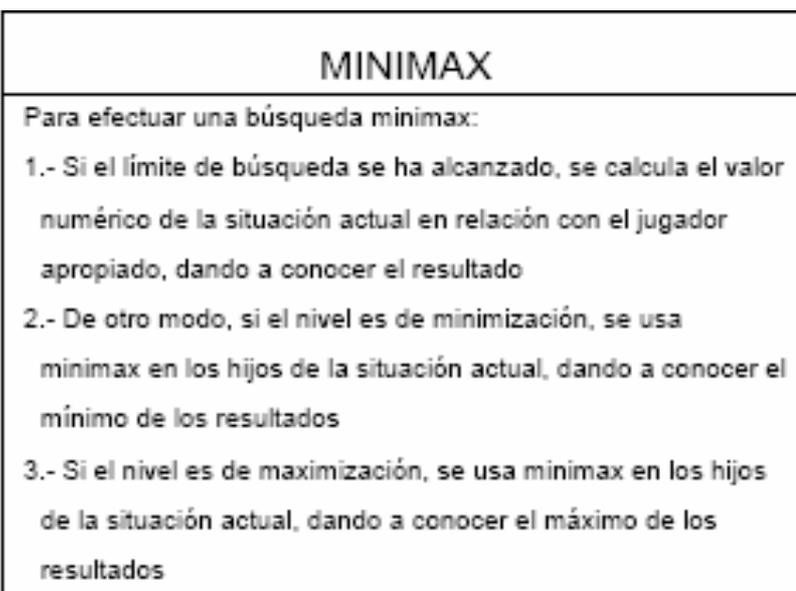
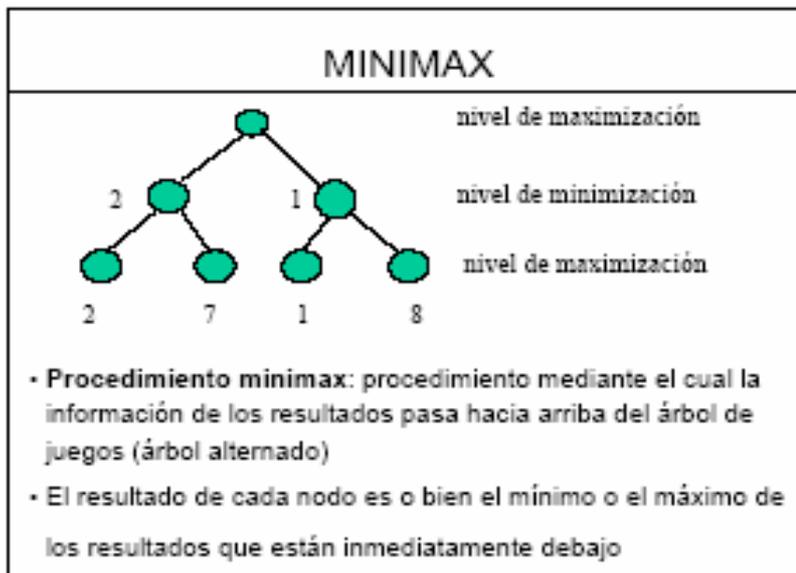


## MINIMAX



## MINIMAX

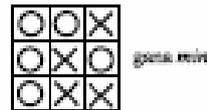
---



### Ejemplo: Tres en Raya

**Juego:**

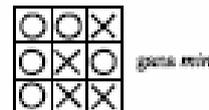
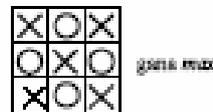
- dos jugadores (*min* y *max*)
- los jugadores van poniendo fichas en las casillas de un tablero 3x3
  - *max* (A) usa las fichas **X** / *min* (B) usa las fichas **O**
  - una casilla puede contener como mucho una ficha



### Ejemplo: Tres en Raya

**Reglas:**

- Inicialmente el tablero está vacío
- *max* empieza y los jugadores se van alternando en poner sus fichas
- *max* gana si obtiene una raya de tres fichas **X**
- *min* gana si obtiene una raya de tres fichas **O**
- si todas las casillas están ocupadas sin que haya una raya de 3 fichas del mismo tipo, hay empate



## MINIMAX

---

### MINIMAX

- Se desarrolla una búsqueda por niveles hasta que se generan los nodos (situaciones) de nivel  $p$
- Se aplica una **función de evaluación** a esos nodos
- La función de evaluación deberá tener en cuenta los factores determinantes de cada juego:
  - nº casillas restantes
  - posición casillas vacías
  - ...

### MINIMAX

- La función de evaluación devolverá valores:
  - *positivos altos*: si la situación de uno de los antagonistas es ventajosa
  - *negativos altos*: si la situación del otro antagonistas es ventajosa
  - *próximos a cero*: si ninguno de los jugadores tiene ventaja

### MINIMAX

- Una función de evaluación  $f(s)$  para el juego del 3 en raya:
  - Si  $s$  no es ganadora para cualquiera de los jugadores:  
 $f(s) = \text{nº filas, columnas o diagonales abiertas para A} - \text{nº filas, columnas o diagonales abiertas para B} = \text{nº líneas que no contienen un "O"} - \text{nº líneas que no contienen una "X"}$
  - Si  $s$  es ganadora para A  
 $f(s) = \infty$  (mayor número positivo posible)
  - Si  $s$  es ganadora para B  
 $f(s) = -\infty$  (mayor número negativo posible)

## MINIMAX

---

### MINIMAX

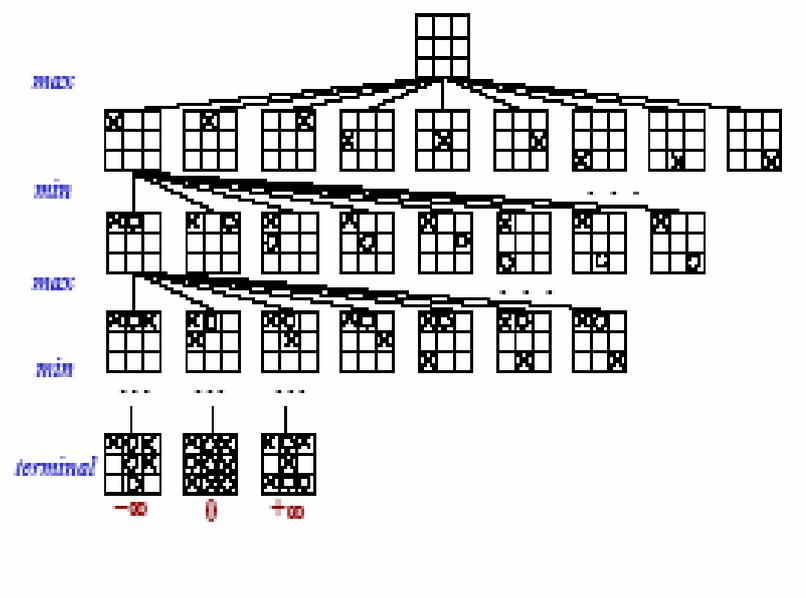
Si juega A: elegirá los nodos de mayor valor de evaluación

Si juega B: elegirá los nodos de menor valor de evaluación

A: maximizante o max

B: minimizante o min

### Ejemplo: Árbol de juego



## BIBLIOGRAFÍA:

### Páginas Web:

 <http://w3.mor.itesm.mx/~emorales/Cursos/Busqueda04/node43.html>

 <http://www.lania.mx/~asanchez/IA/minimax2.html>

 <http://dmi.uib.es/~abasolo/intart/2-juegos.html#2.3>

 <http://es.wikipedia.org/wiki/Minimax>

 <http://www.escet.urjc.es/~rmartine/EDAA-minimax.pdf>