

ALGORITMO MINIMAX

- Algoritmo de decisión para minimizar la pérdida máxima aplicada en juegos de adversarios
- Información completa (cada jugador conoce el estado del otro)
- Elección del mejor movimiento para cada jugador, suponiendo que el contrincante escogerá el peor
- El espacio de estados se representa mediante árboles alternados, donde:
 - **Nodo:** Representa una situación del juego
 - **Sucesores de un nodo:** Situaciones del juego a las que se

accede por movimientos legales aplicando sus reglas

- **Nivel:** Contiene todas las situaciones posibles para uno de los jugadores
- 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.

Representación de los juegos

- Posición inicial.
- Conjunto de operadores o reglas del juego (definen movimientos legales)
- Estado terminal
- Función de utilidad, ej. gana, pierde, empata

Pasos del Algoritmo Minimax

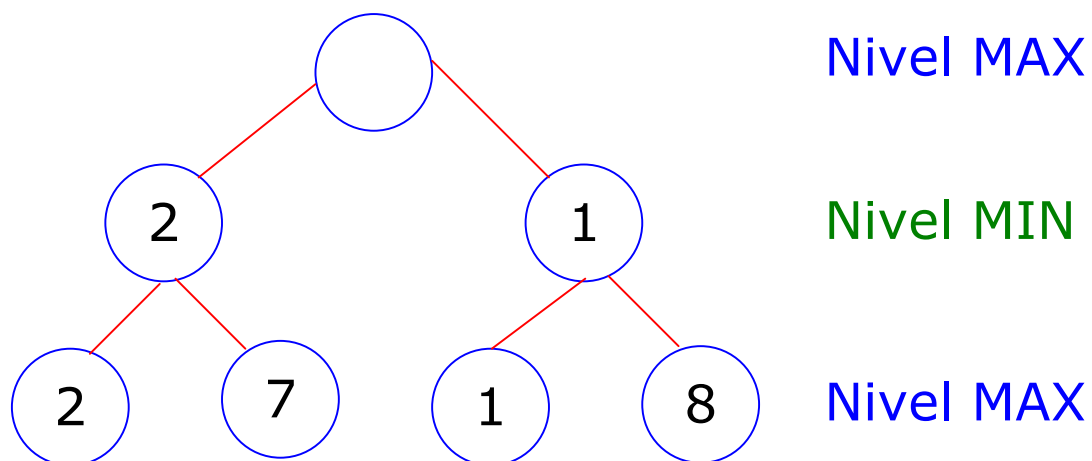
1. Generación del **árbol de juego**. Se generarán todos los nodos hasta llegar a un estado terminal.
2. Cálculo de los valores de la **función de utilidad** para cada nodo terminal.

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.
 - Colocar 0 ó 1 en los nodos terminales dependiendo si gana MIN o MAX

- La función de utilidad definirá lo buena que es la posición para un jugador cuando la alcanza.
- Se requiere de una estrategia que garantice llegar a estados terminales ganadores independientemente de lo que haga el oponente.
- Un valor positivo indica la ventaja de un jugador y uno negativo la ventaja del otro.
- El jugador que espera valores positivos se conoce como **maximizador**
- El jugador que espera valores negativos se conoce como **minimizador**

- El **maximizador** busca movimientos que lo conduzcan al mayor número positivo
- El **minimizador** busca movimientos que lo conduzcan al menor número negativo

- **P. ejemplo:**



- El **maximizador**:
 - Puede esperar llegar a un valor de 8

- Sabe que el **minimizador** puede escoger un movimiento que lo lleve a un valor de 1
- Desde el punto de vista de el **maximizador**, el **minimizador** puede escoger 2 ó 1
- Los resultados de un nivel determinan la acción y el resultado del nivel inmediato superior

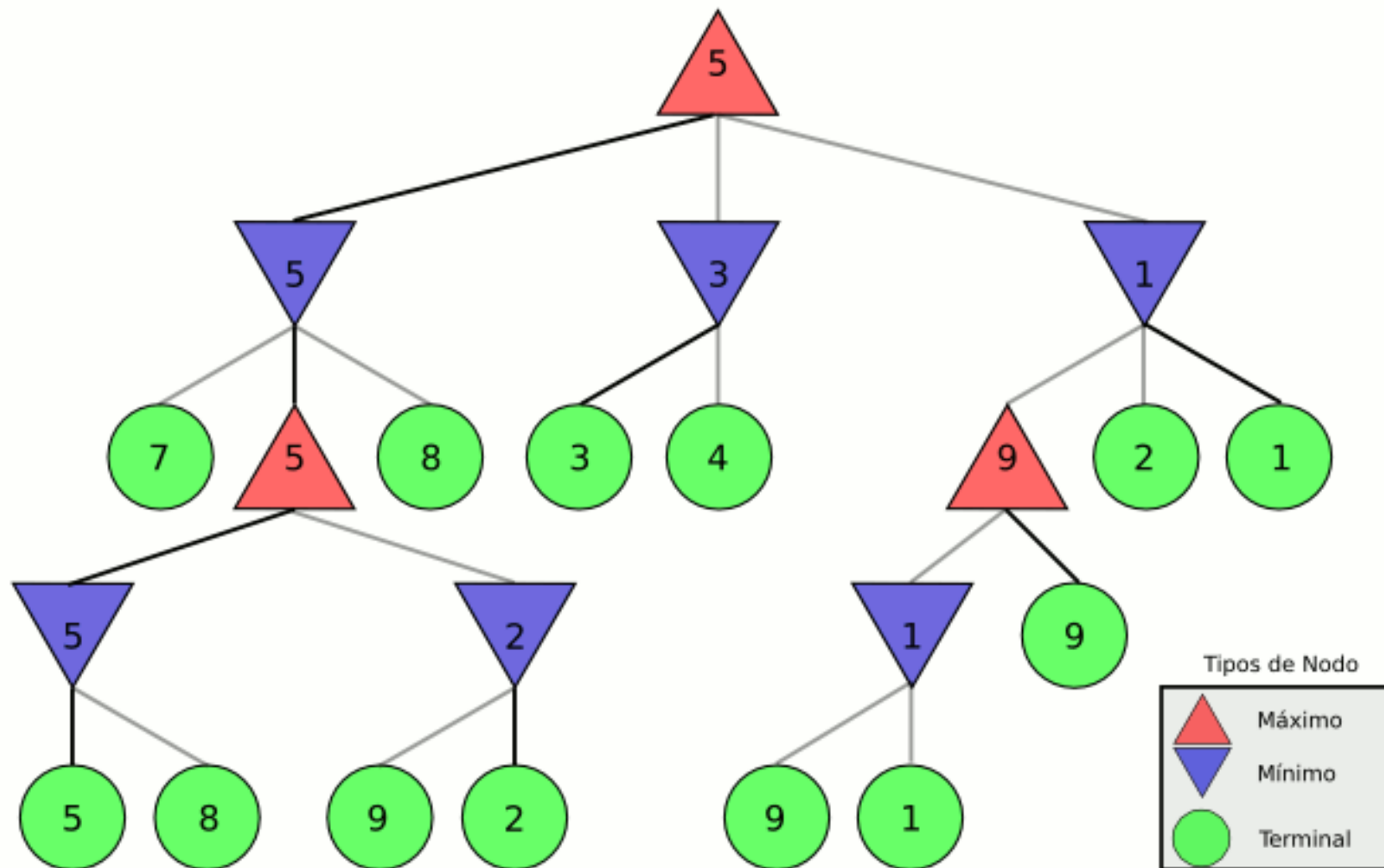
Cálculo de valores de la función de utilidad

```
Calcular el valor minimax del nodo J:  $V(J)$   
SI J es Terminal,  $V(J) \leftarrow ev(J)$   
SI NO  
    Genera los sucesores de J:  $J_1, J_2, \dots, J_n$   
    Evalúa  $V(J_1), V(J_2), \dots, V(J_n)$  de izq a der  
    SI J es nodo Max ENTONCES  
         $V(J) \leftarrow \max[V(J_1), V(J_2), \dots, V(J_n)]$   
    SI J es nodo Min ENTONCES  
         $V(J) \leftarrow \min[V(J_1), V(J_2), \dots, V(J_n)]$ 
```


EJEMPLO

- 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.
- **1er. Paso:** Calcular los nodos terminales, en verde.

- **2º. Paso:** Calcular el cuarto nivel, movimiento **MIN**, minimizando lo elegido (5, 2 y 1).
- **3er. Paso:** 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 maximize nuestra utilidad (5).



Aplicación: El Juego del Gato

- Dos jugadores **MIN** y **MAX**
- Los jugadores colocan fichas en un tablero de 3 X 3
- **MAX** usa las fichas X
- **MIN** usa las fichas O

| | | |
|---|---|---|
| X | O | X |
| O | X | O |
| X | O | X |

i **MAX** gana i

- **Reglas:**
- Inicialmente el tablero está vacío
- **MAX** empieza y se alternan los movimientos

| | | |
|---|---|---|
| O | O | X |
| O | X | O |
| O | X | X |

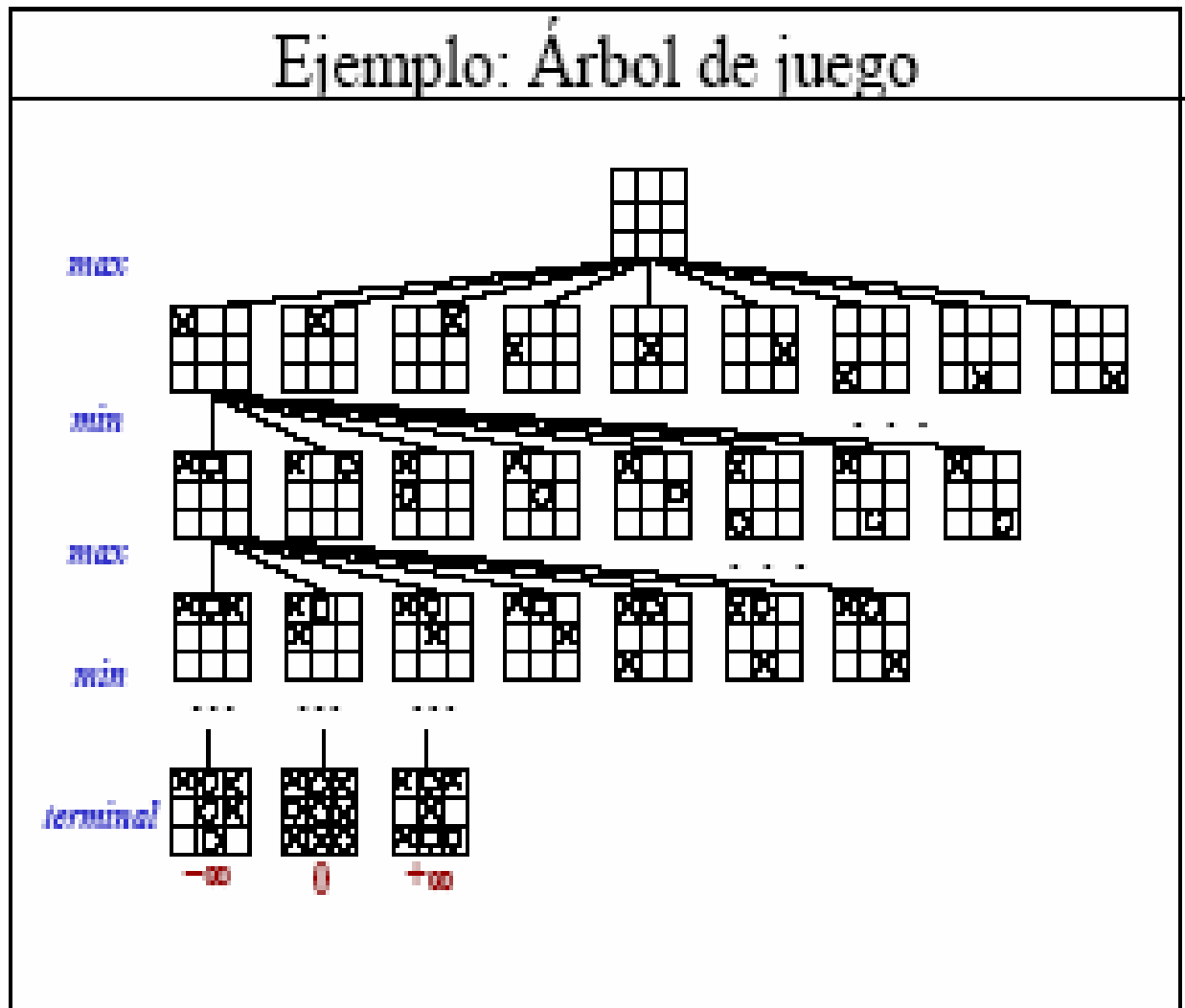
i **MIN** gana i

- **MAX** gana si obtiene una línea de 3 X's
- **MIN** gana si obtiene una línea de 3 O's
- Existe la posibilidad de empate

| | | |
|---|---|---|
| X | O | X |
| O | X | O |
| O | X | O |

i Empate i

Espacio de estados para el juego del gato



Procedimiento

- Se desarrolla una búsqueda por niveles, generando los nodos del cada nivel
- Se aplica una *función de evaluación* a cada nodo
- La *función de evaluación* considera los siguientes factores:
 - Número de casillas restantes
 - Posición de casillas vacías
- La *función de evaluación* devolverá los siguientes valores:
 - *Positivos altos*: Si la situación de uno de los jugadores es ventajosa
 - *Negativos altos*: Si la situación del otro jugador es ventajosa
 - *Cero*: Si ninguno de los jugadores tiene ventaja

función de evaluación para el juego del gato

- Si **s** no es ganadora para cualquiera de los jugadores (MAX o MIN):

$f(s) = \text{No. filas abiertas para MAX} - \text{No. Filas, columnas o diagonales abiertas para MIN}$

$f(s) = \text{No. Líneas que no contiene una "O"} - \text{No. Líneas que no contienen una "X"}$

esto es:

- Si **s** es ganadora para el jugador MAX

$f(s) = \infty$ (mayor número positivo posible)

- Si **s** es ganadora para el jugador MIN

$f(s) = -\infty$ (mayor número negativo posible)

- MAX elegirá los nodos de mayor evaluación
- MIN elegirá los nodos de menor evaluación

Caso práctico de función de evaluación para el juego del gato

- Se define la función de evaluación:

$$f(s) = N_{MAX}(s) - N_{MIN}(s)$$

donde:

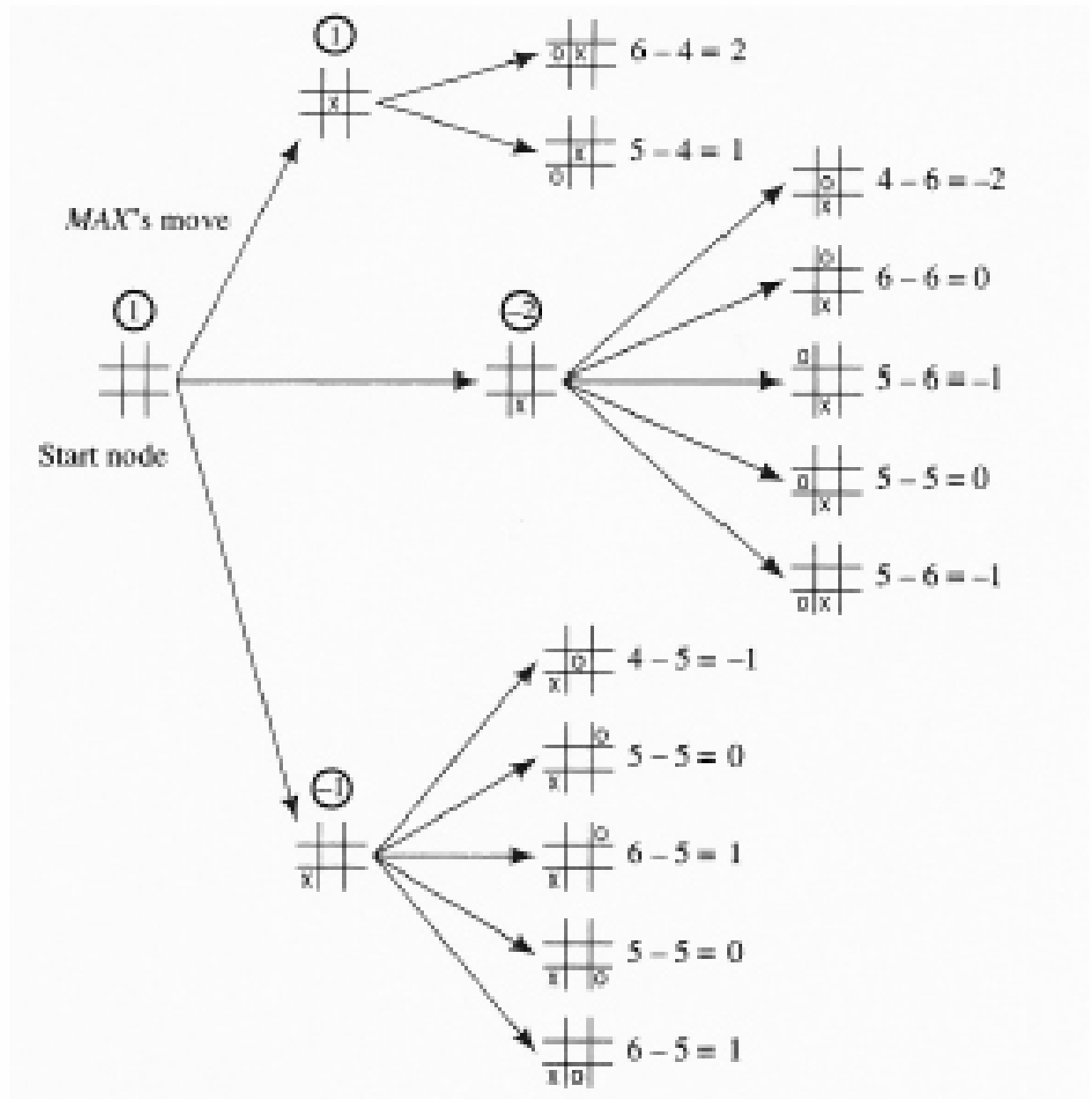
S : Situación o distribución del tablero

$f(s)$: Función de evaluación del tablero (nodo del espacio de estados)

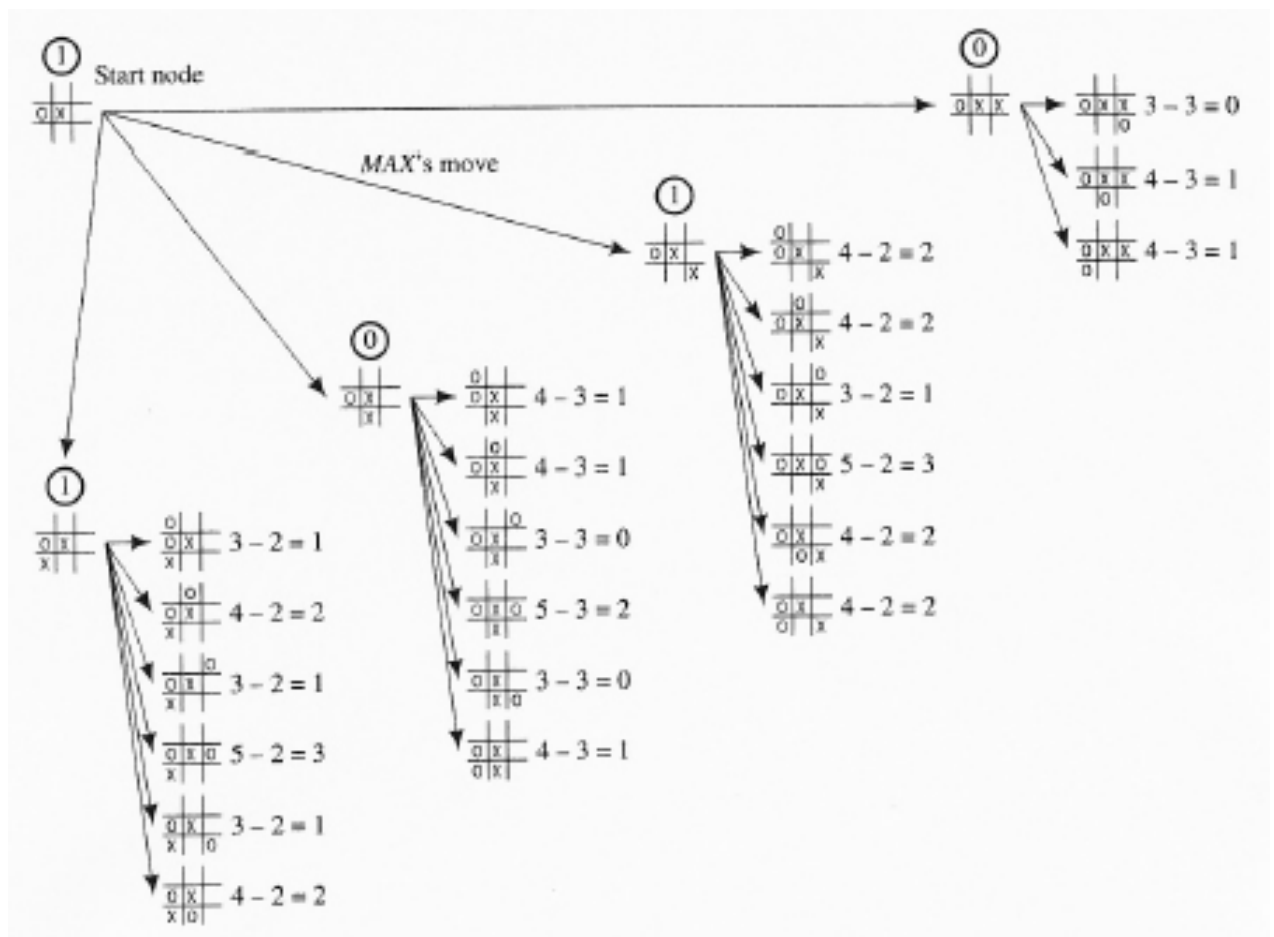
$N_{MAX}(s)$: No. de filas, columnas o diagonales abiertas para MAX (donde aún puede ganar)

$N_{MIN}(s)$: No. de filas, columnas o diagonales abiertas para MIN (donde aún puede ganar)

1º. Etapa del Espacio de Estados



2º. Etapa del Espacio de Estados



3ª. Etapa del Espacio de Estados

