Inteligencia Artificial

Oscar Bedoya

oscar.bedoya@correounivalle.edu.co

* Algoritmo minimax
* Poda alfa-beta
* Juegos con decisiones imperfectas

Calla el arroz,

viento de otoño y risas

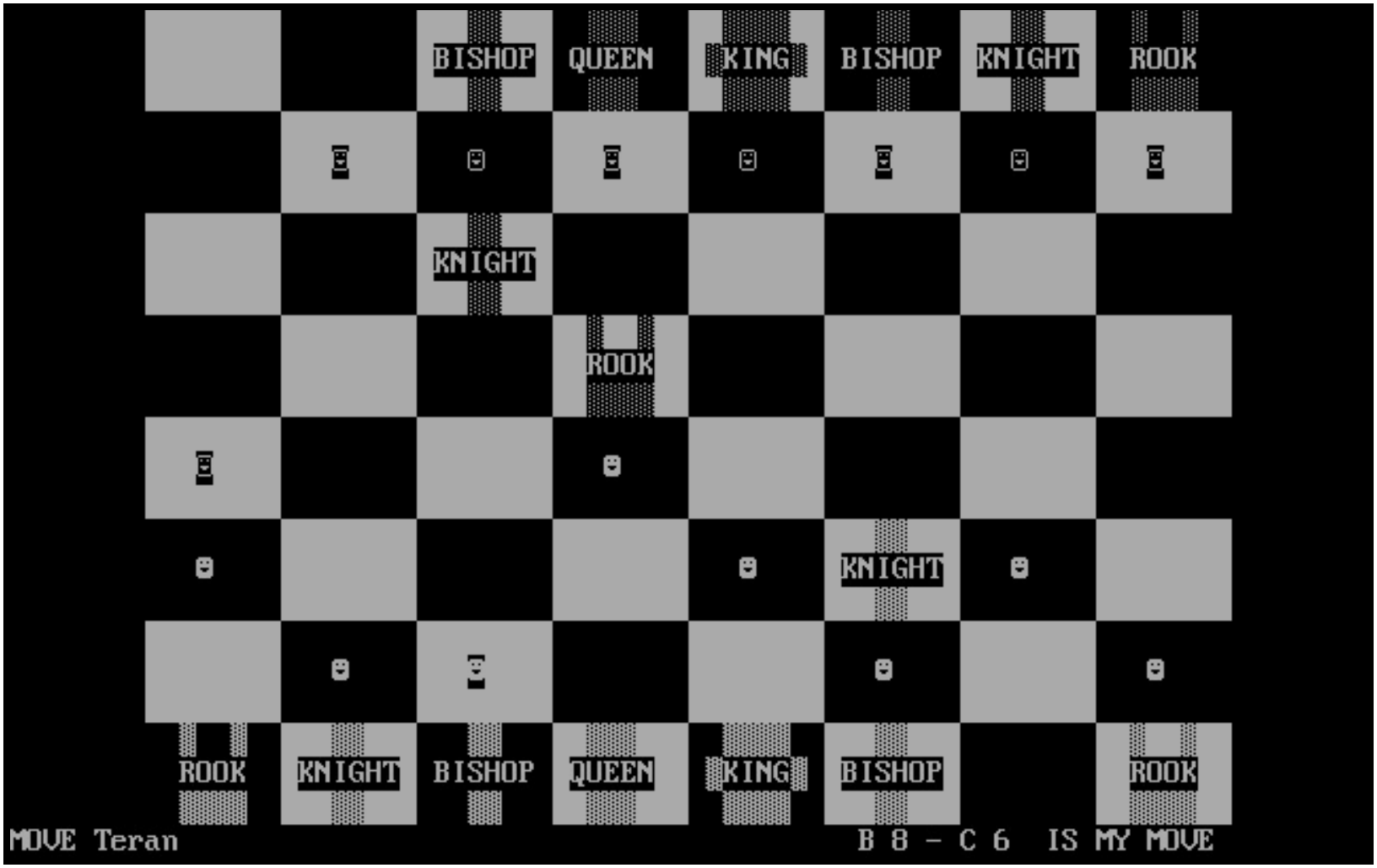
en el crepúsculo

Juegos

Tipos de contrincantes

- Humano Vs Humano
- Humano Vs Máquina
- Máquina Vs Máquina

Juegos


BISHOP
QUEEN
KING
BISHOP
KNIGHT
ROOK
KNIGHT
ROOK
KNIGHT
ROOK
KNIGHT
BISHOP
QUEEN
KING
BISHOP
ROOK
MOVE Teran
B 8 - C 6 IS MY MOVE

Juegos

Máquina Vs Máquina

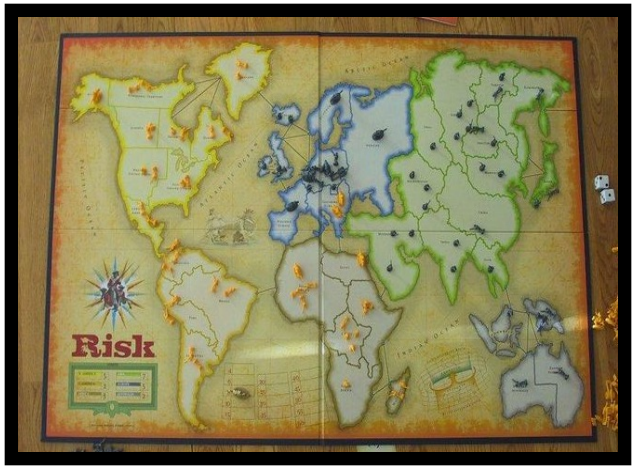
Juegos

Juegos como problemas de búsqueda

- La IA se centra en el análisis de juegos donde los estados se puedan **representar** fácilmente e intervenga en los movimientos la **toma de decisiones**

Juegos

Juegos como problemas de búsqueda



Juegos

- Si un juego tiene un único equilibrio de Nash y los jugadores son completamente racionales, escogerán las estrategias que forman el equilibrio

John Nash (1928 - 2015)

Juegos

Teoría de juegos

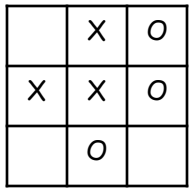
- Construir el árbol de juego
- Analizar el árbol

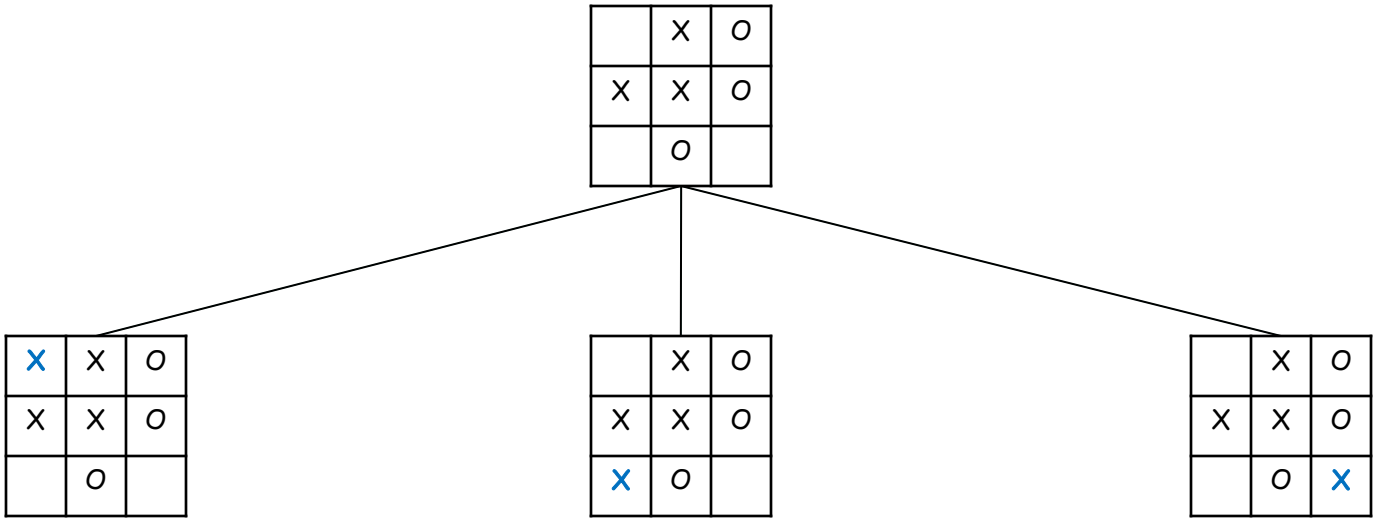
Árboles

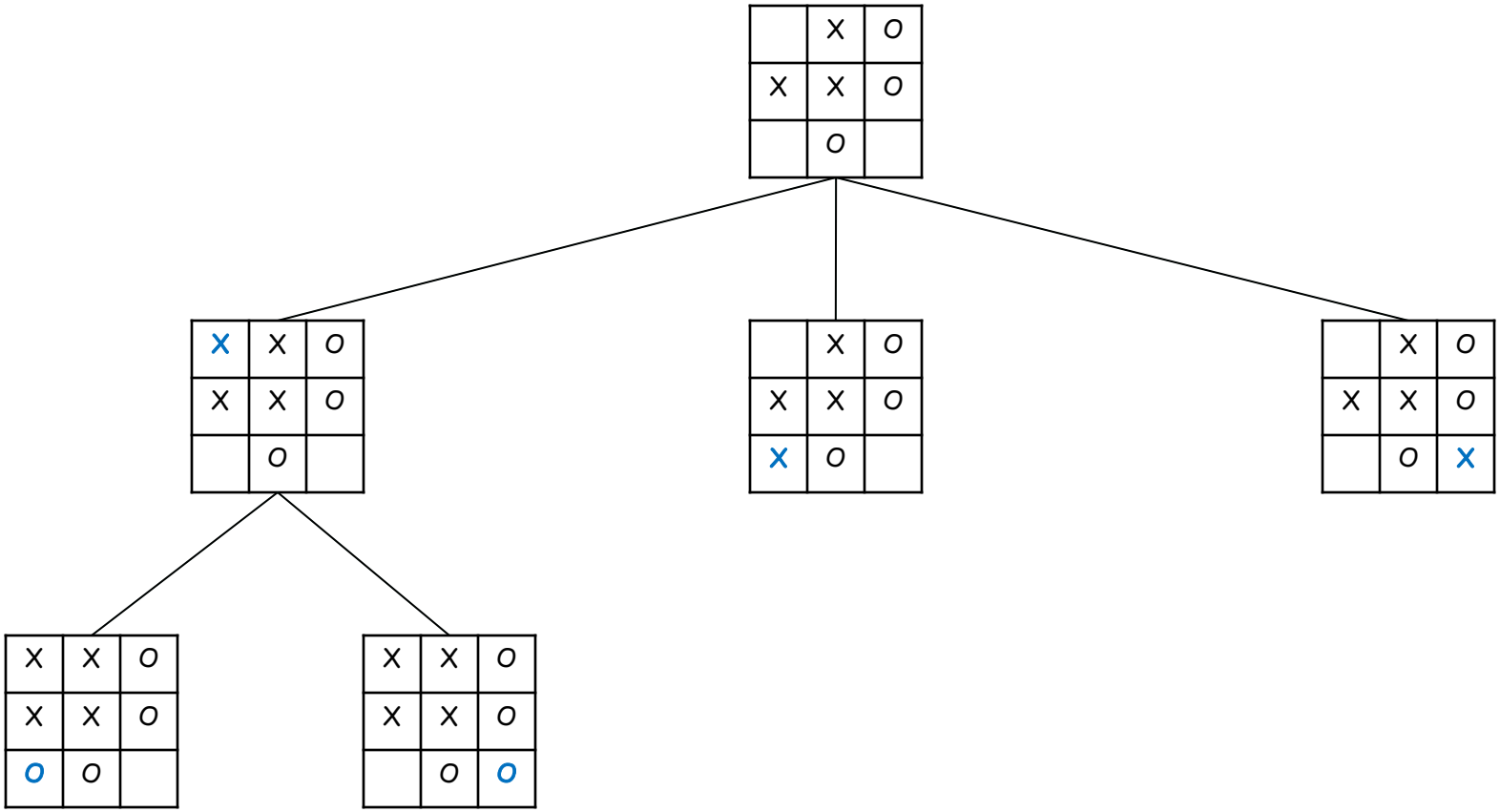
Construir el árbol de juego

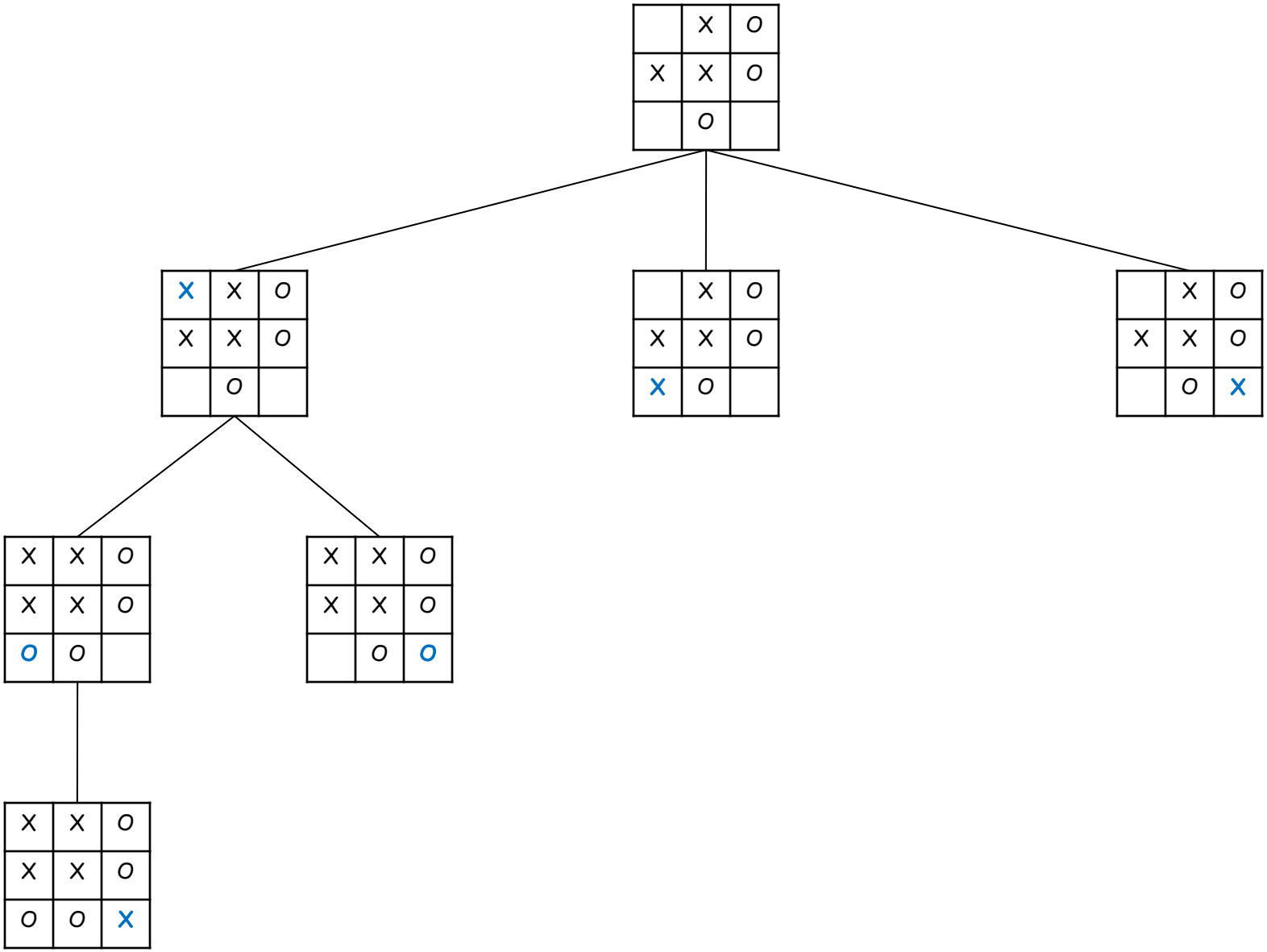
	X	O
X	X	O
	O	

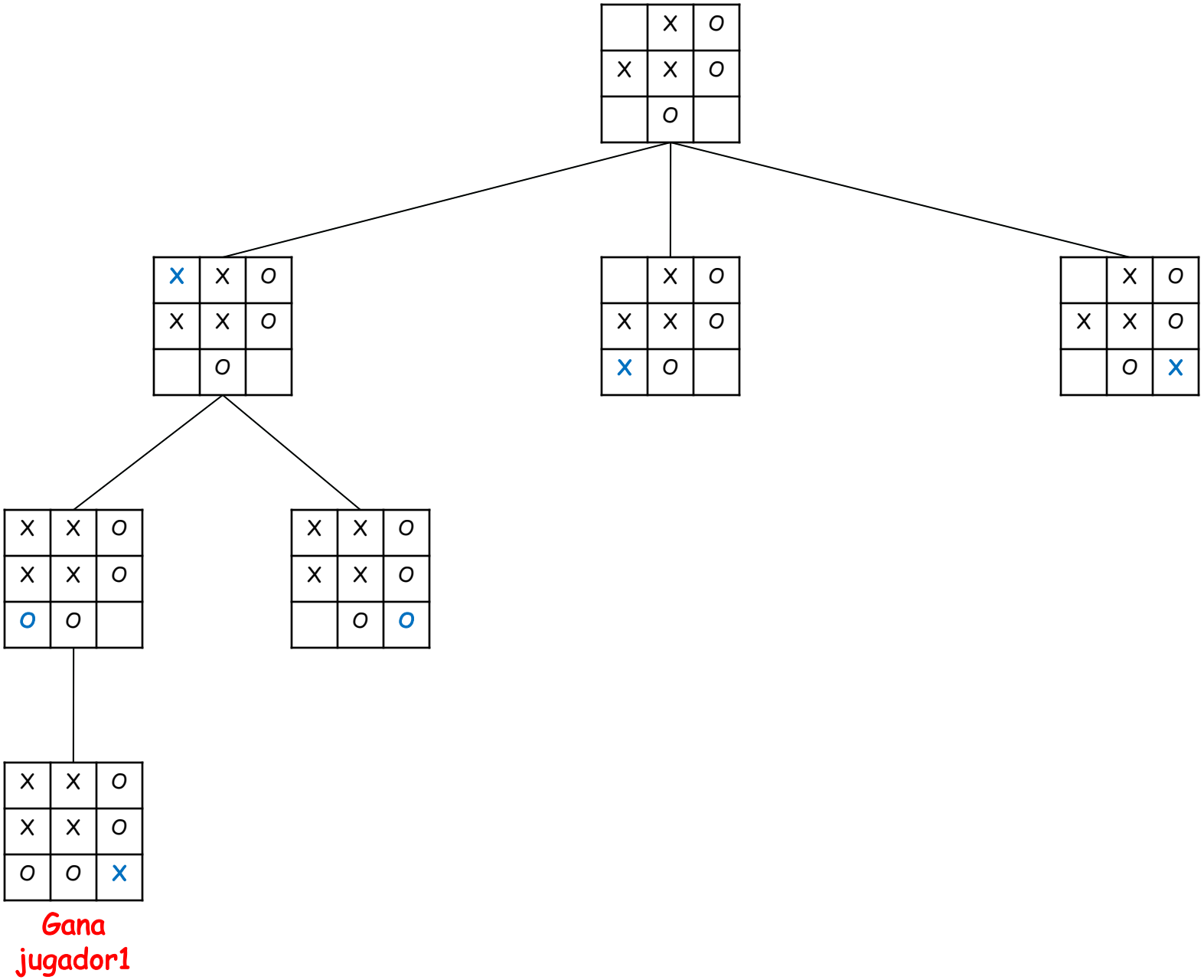
- La jugada es de (X)

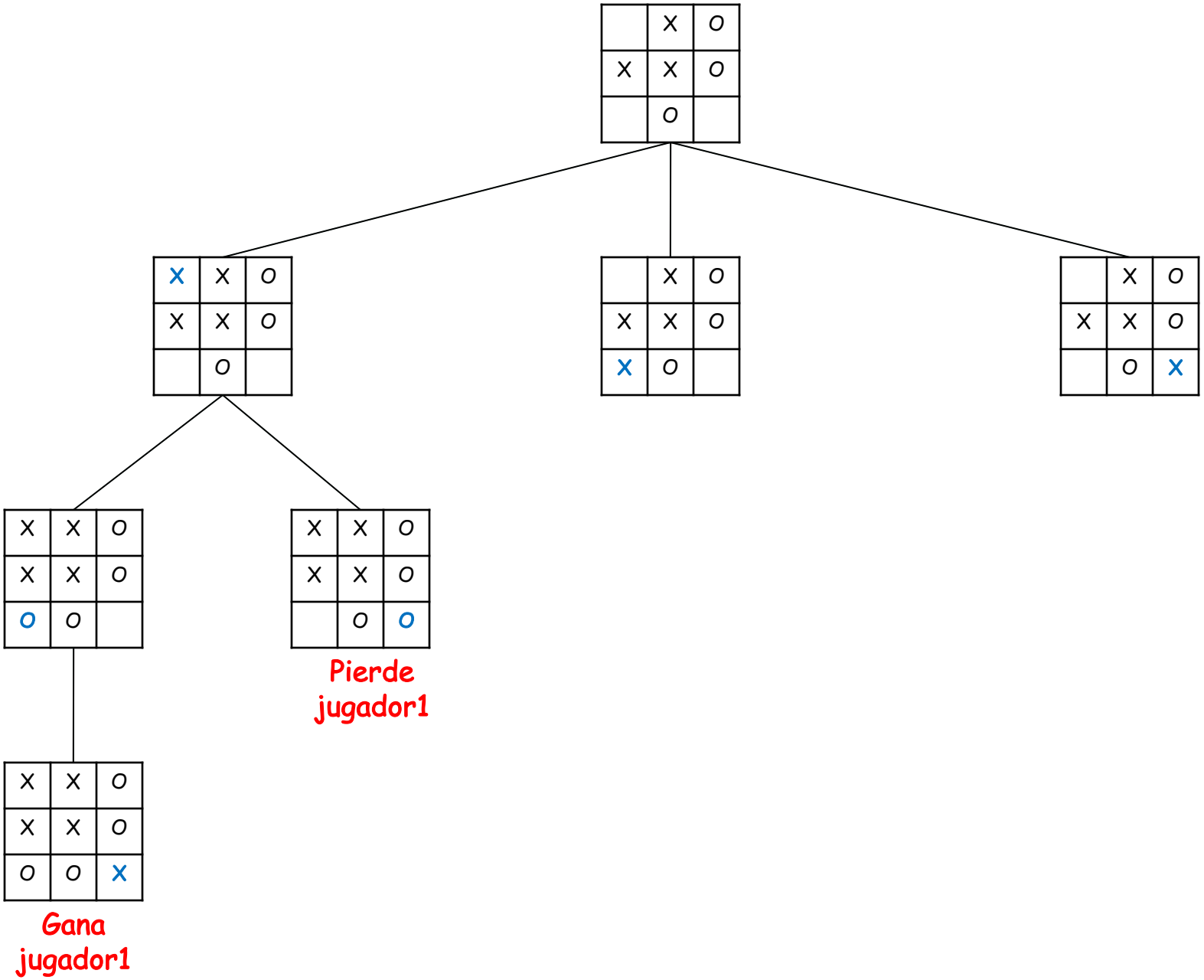

X
O
X
X
O
O

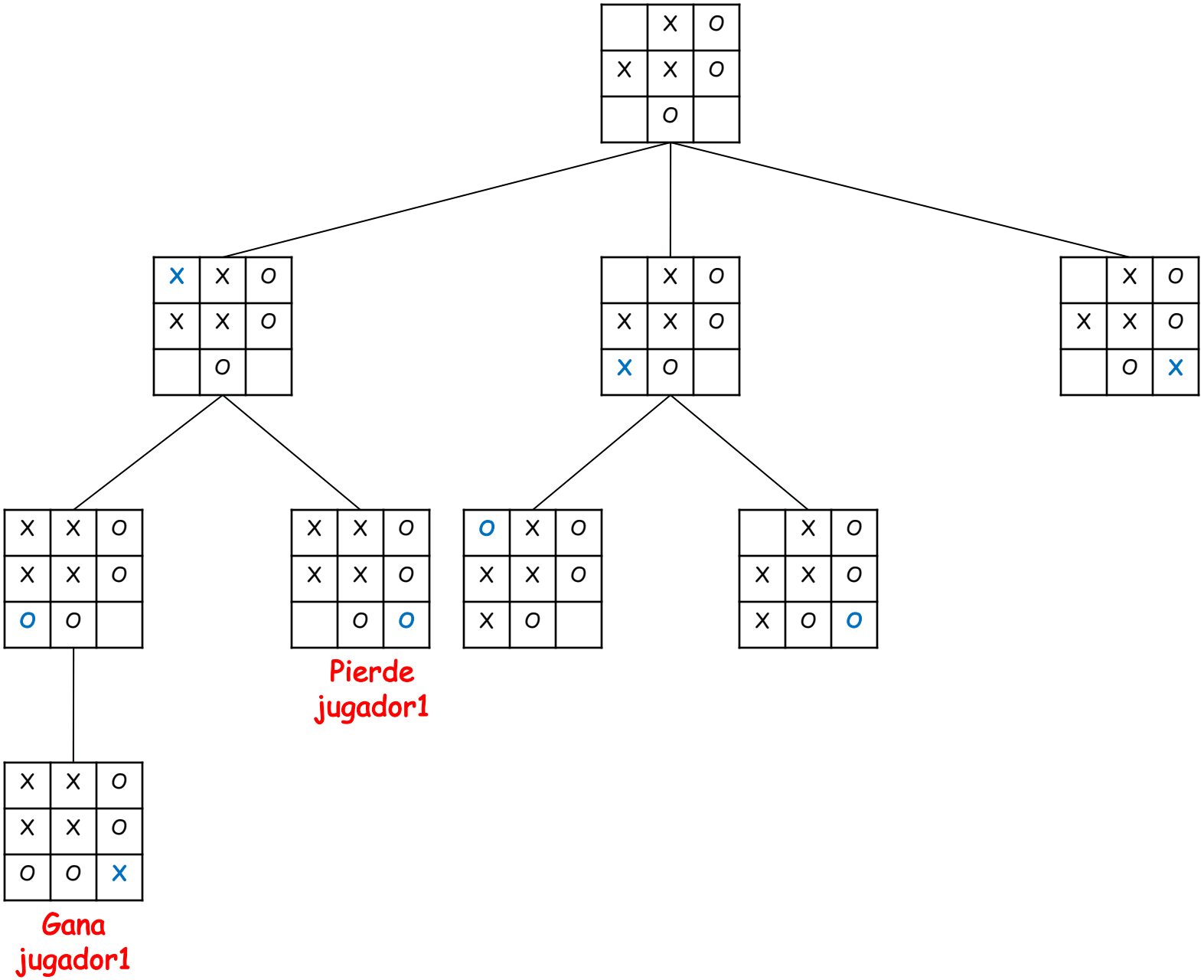

X O
X X O
O
X X O
X X O
O
X X O
X X O
X O
X X O
X X O
O X

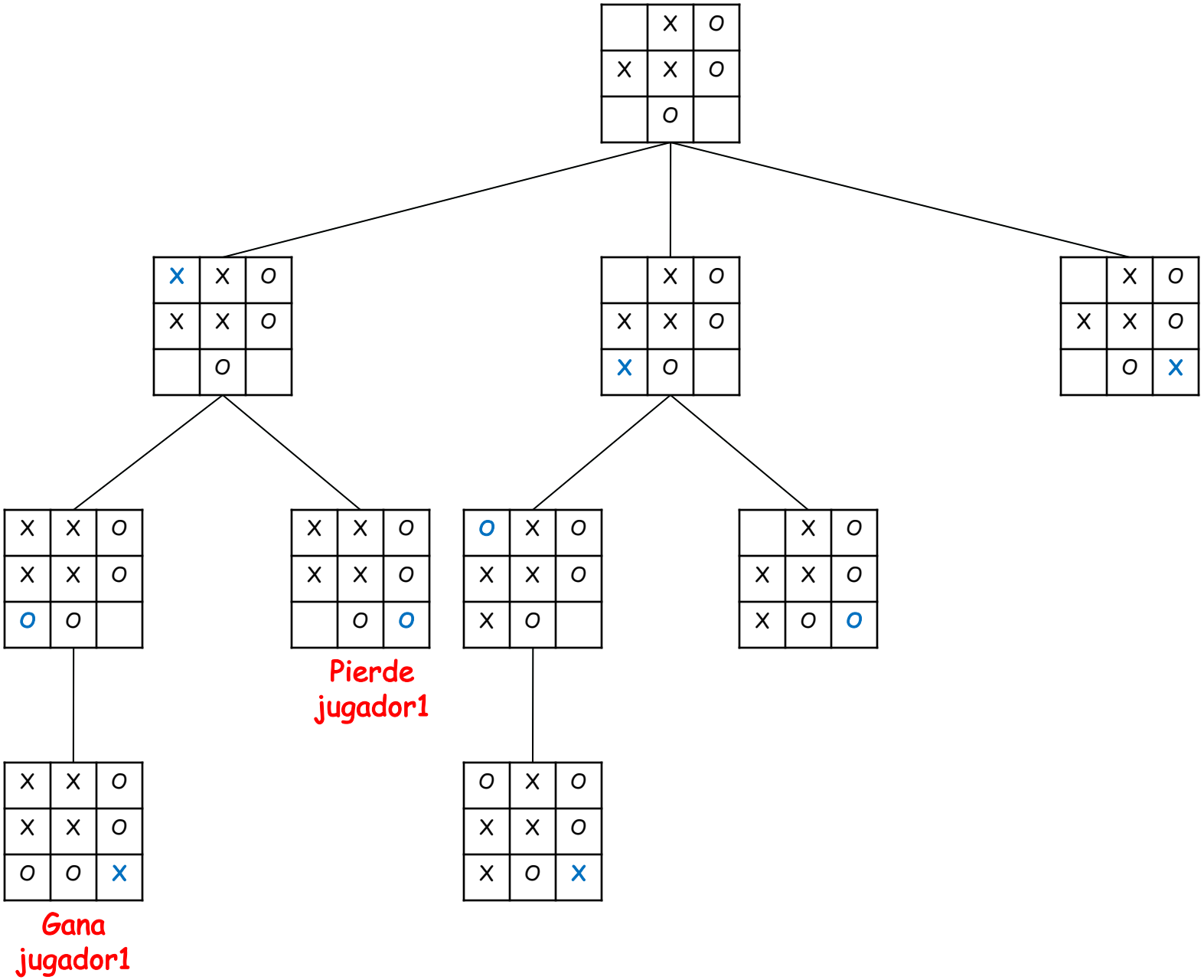

X O
X X O
O
X X O
X X O
O
X X O
X X O
X O
X X O
X X O
O O
X X O
X X O
O O

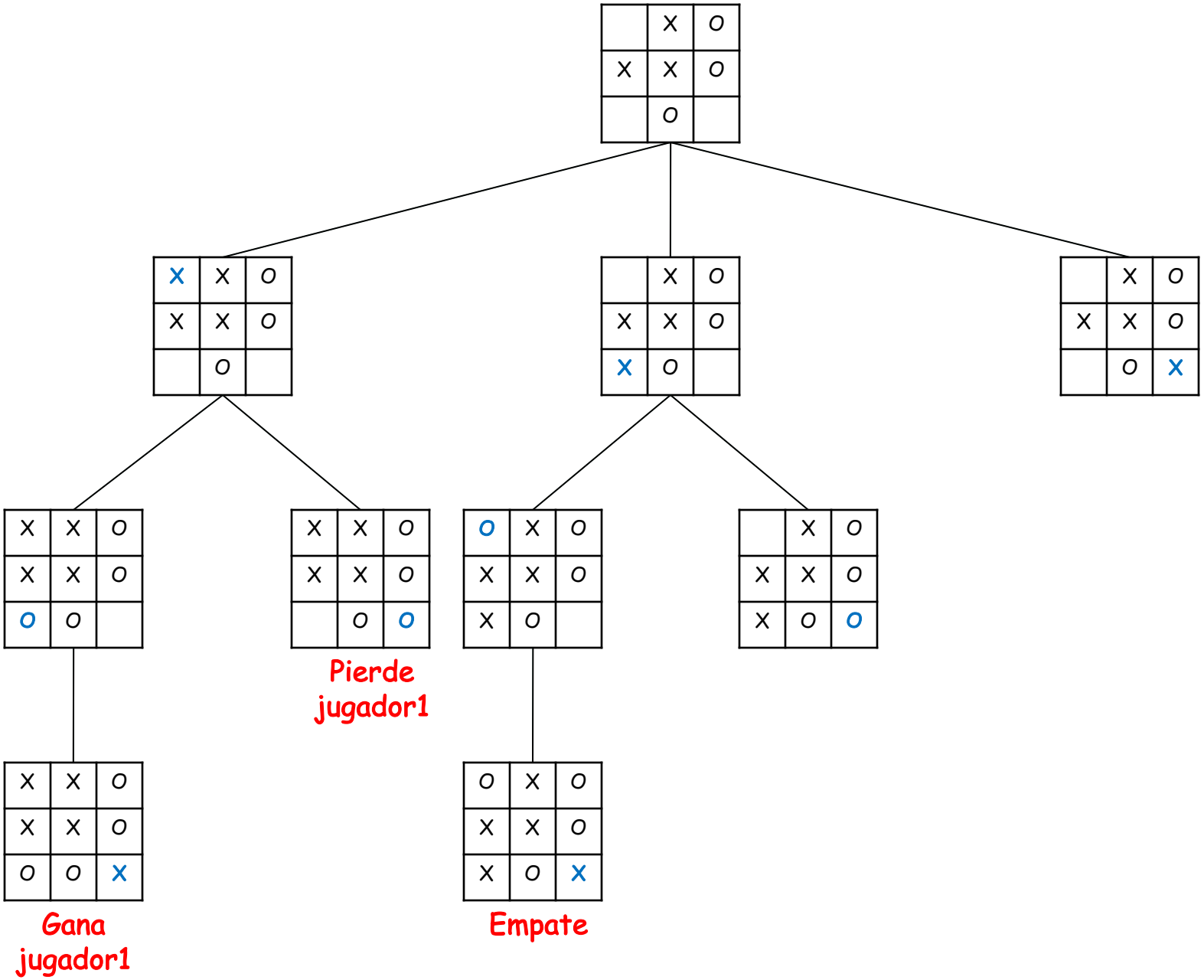

X O
X X O
O
X X O
X X O
O
X X O
X X O
X O
X X O
X X O
O O
X X O
X X O
O O
X X O
X X O
O O X

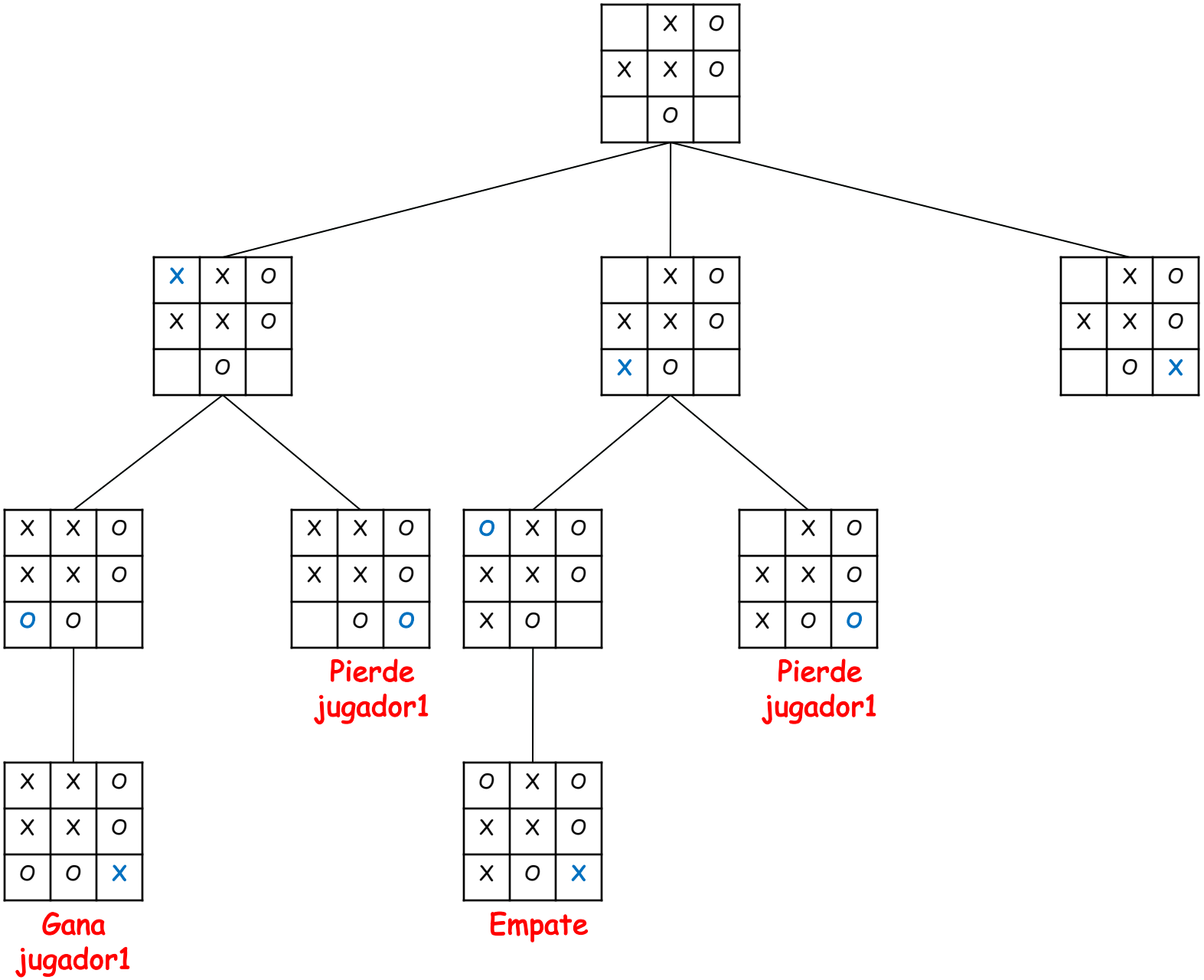

Gana
jugador1

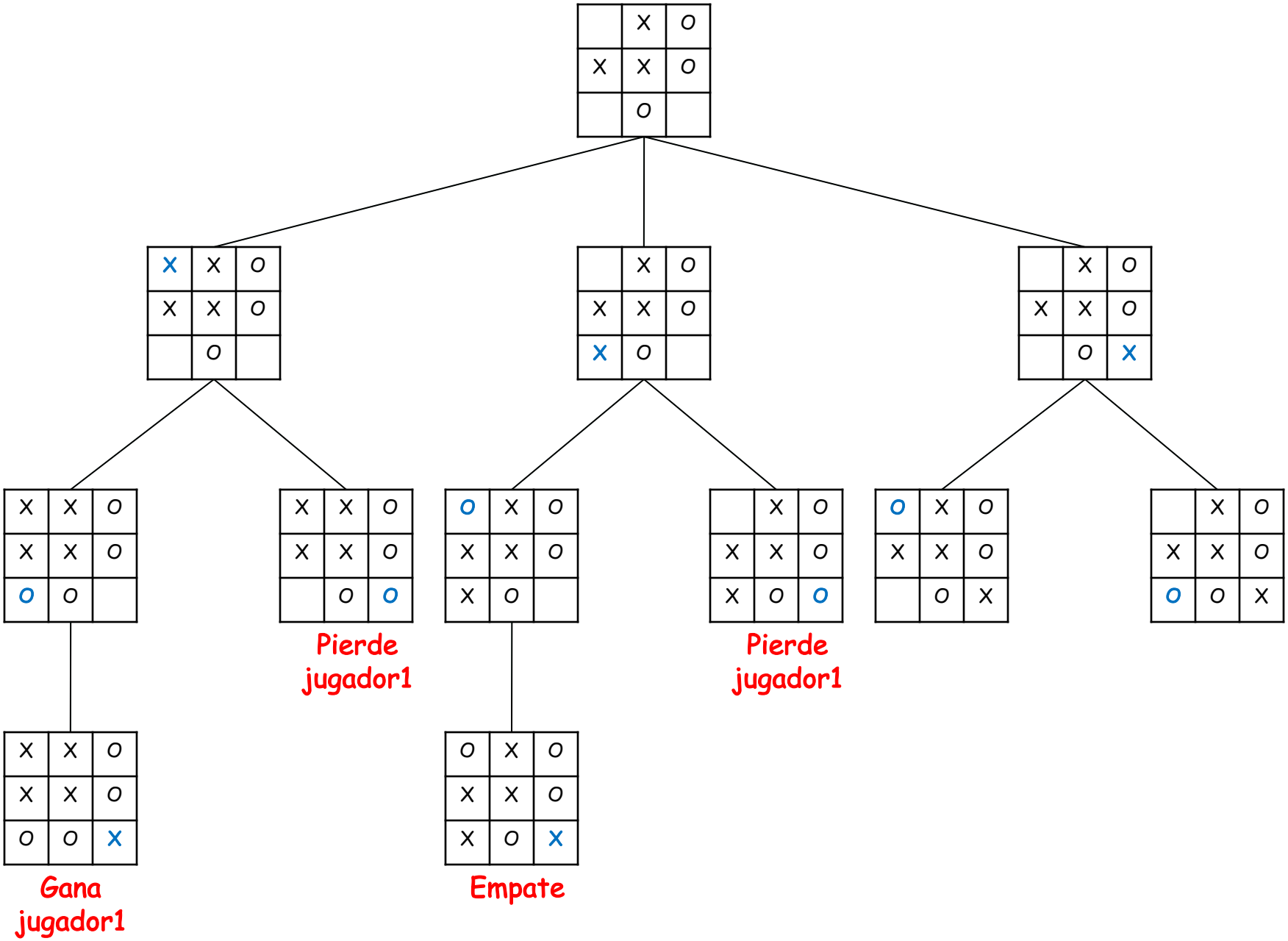

Pierde
jugador1
Gana
jugador1

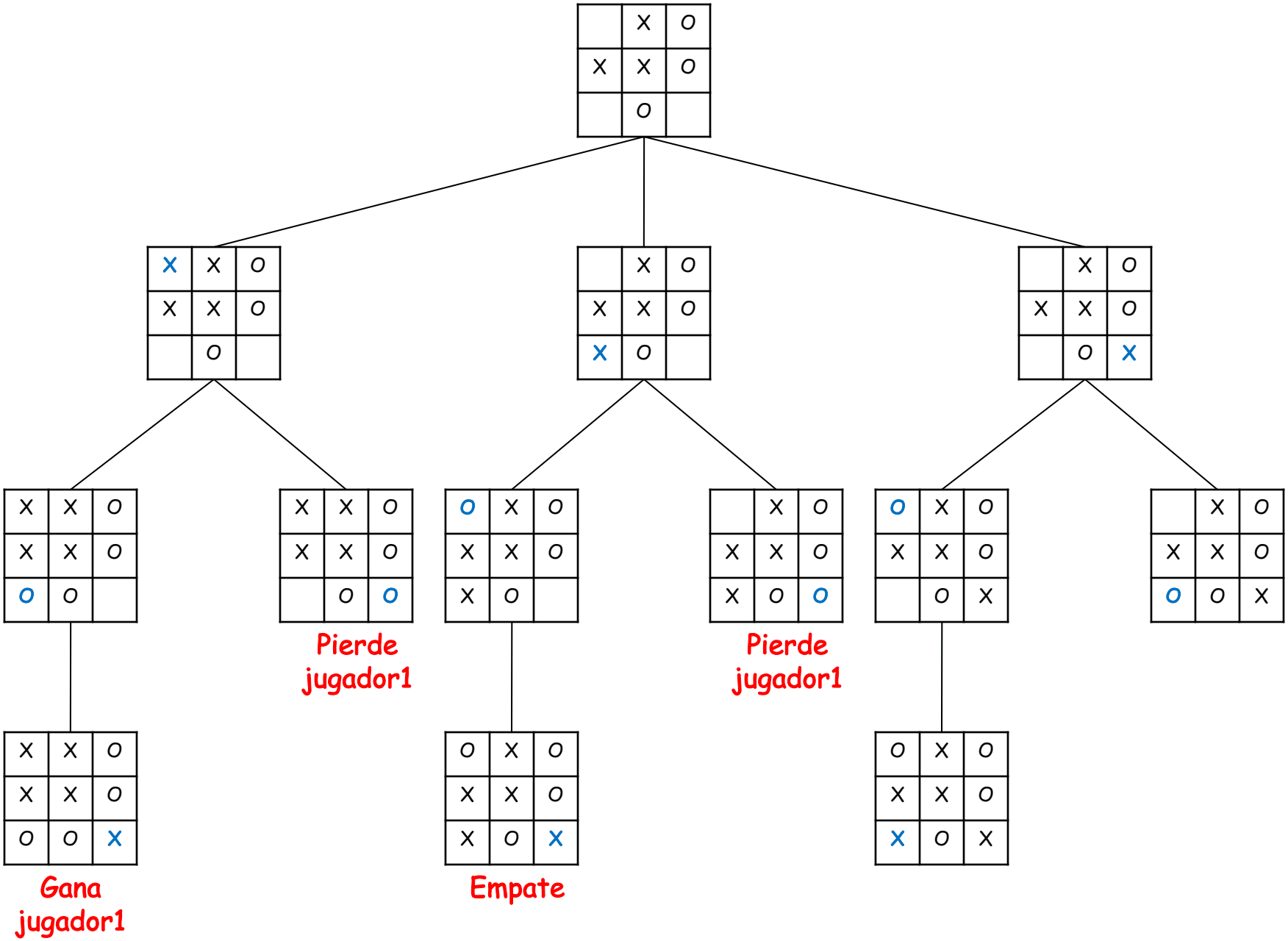

Pierde
jugador1
Gana
jugador1

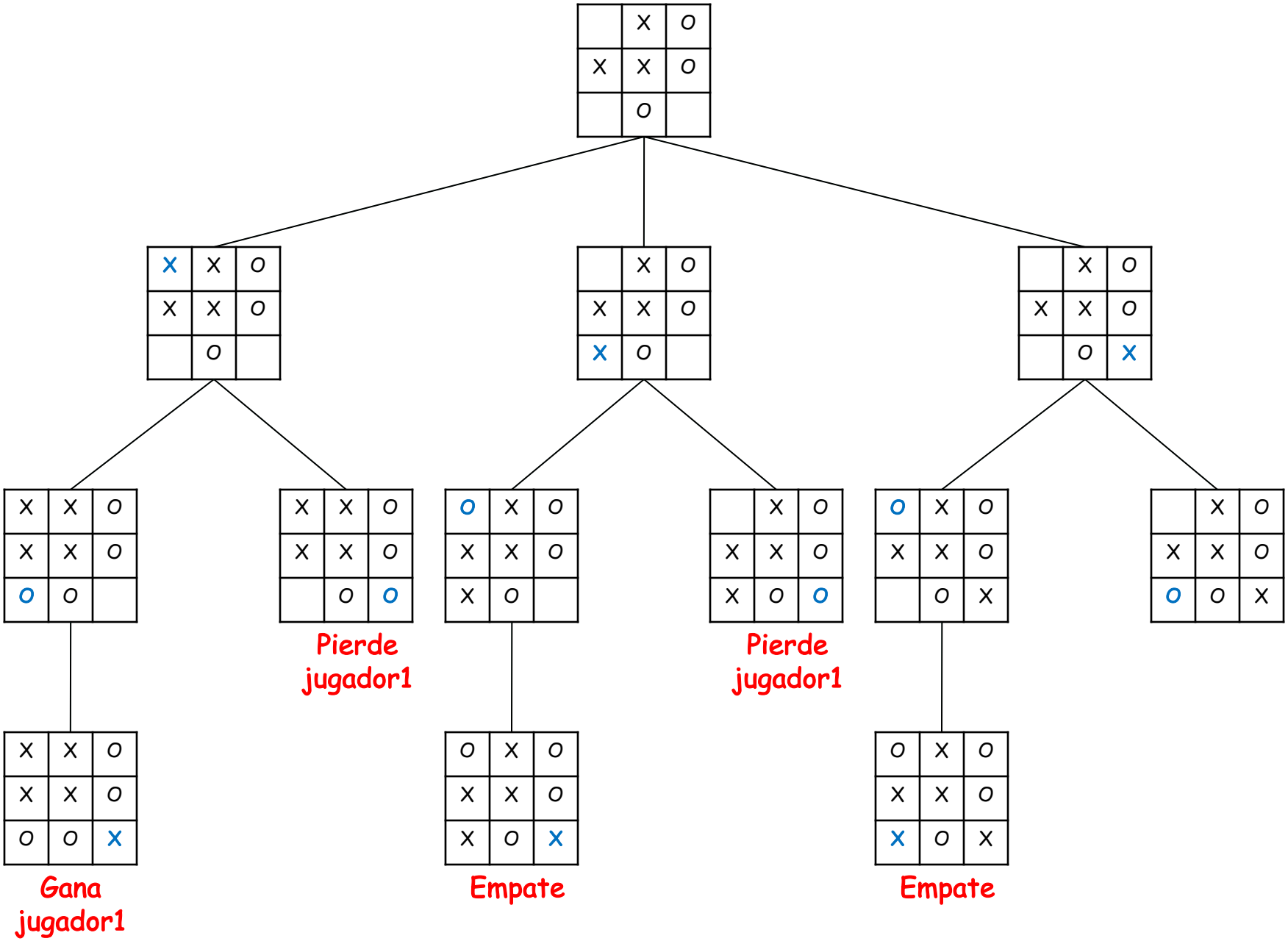

Pierde
jugador1
Gana
jugador1

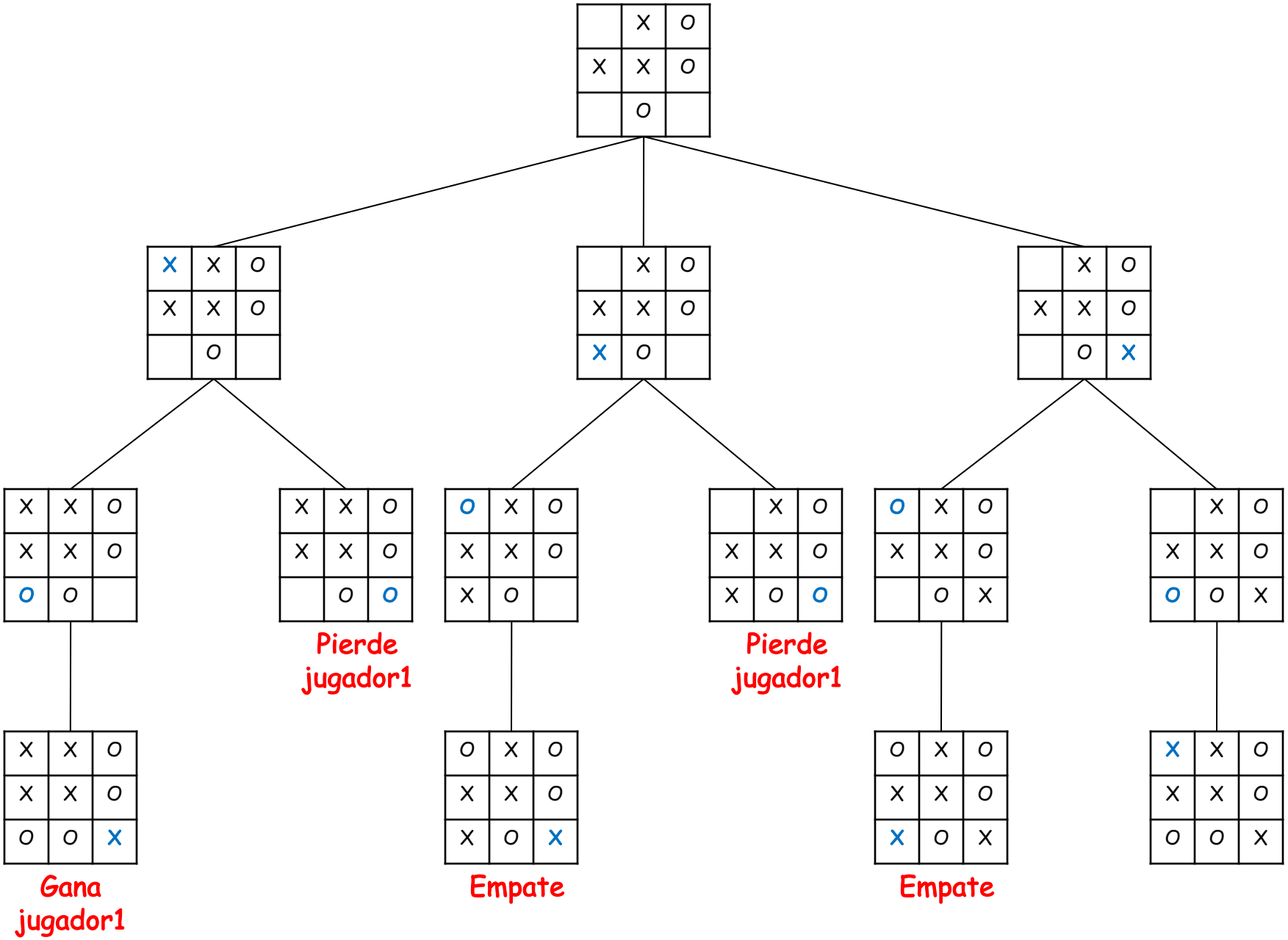

Pierde
jugador1
Gana
jugador1
Empate

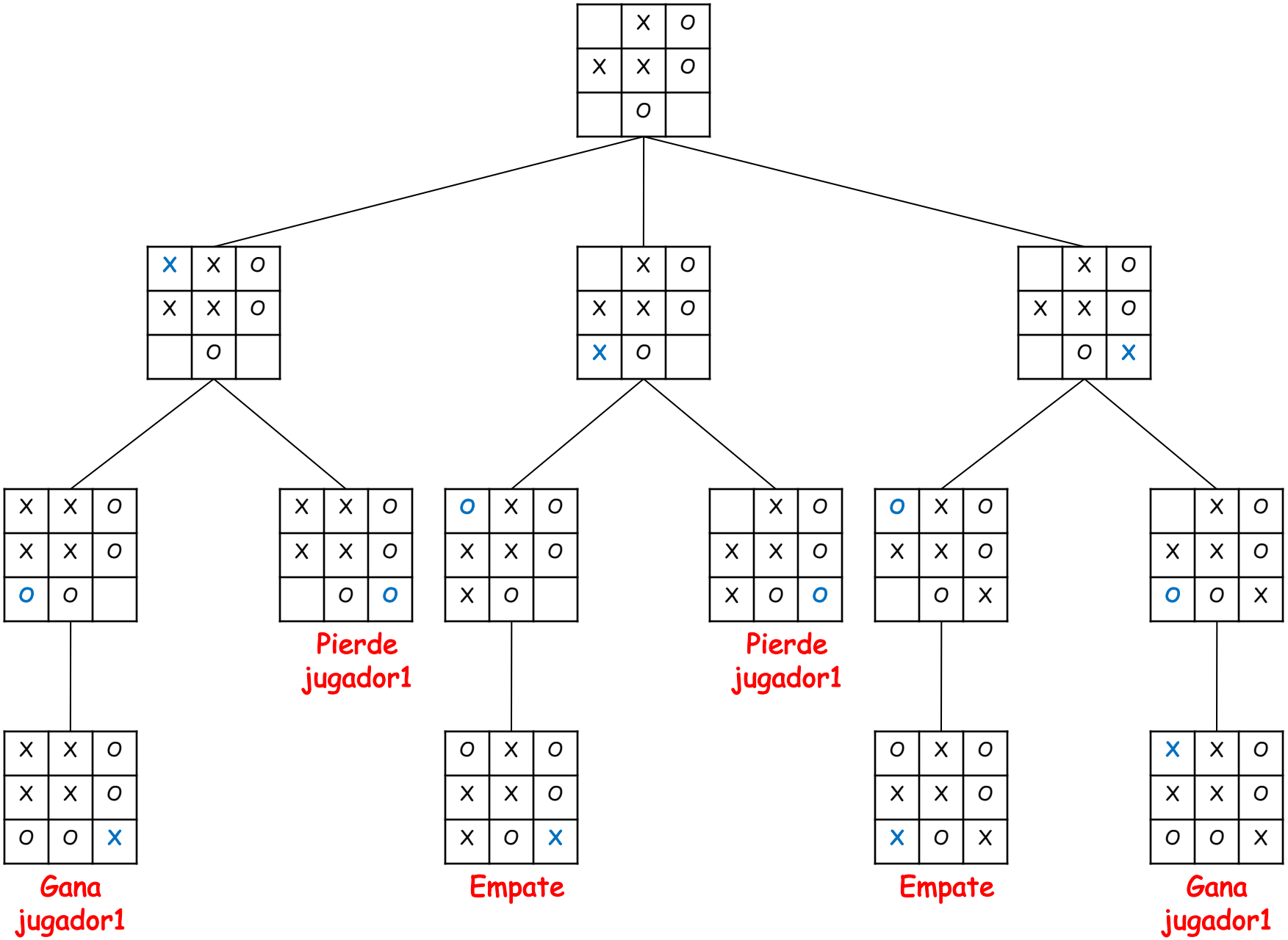

Pierde jugador1
Pierde jugador1
Gana jugador1
Empate


Pierde jugador1
Pierde jugador1
Gana jugador1
Empate

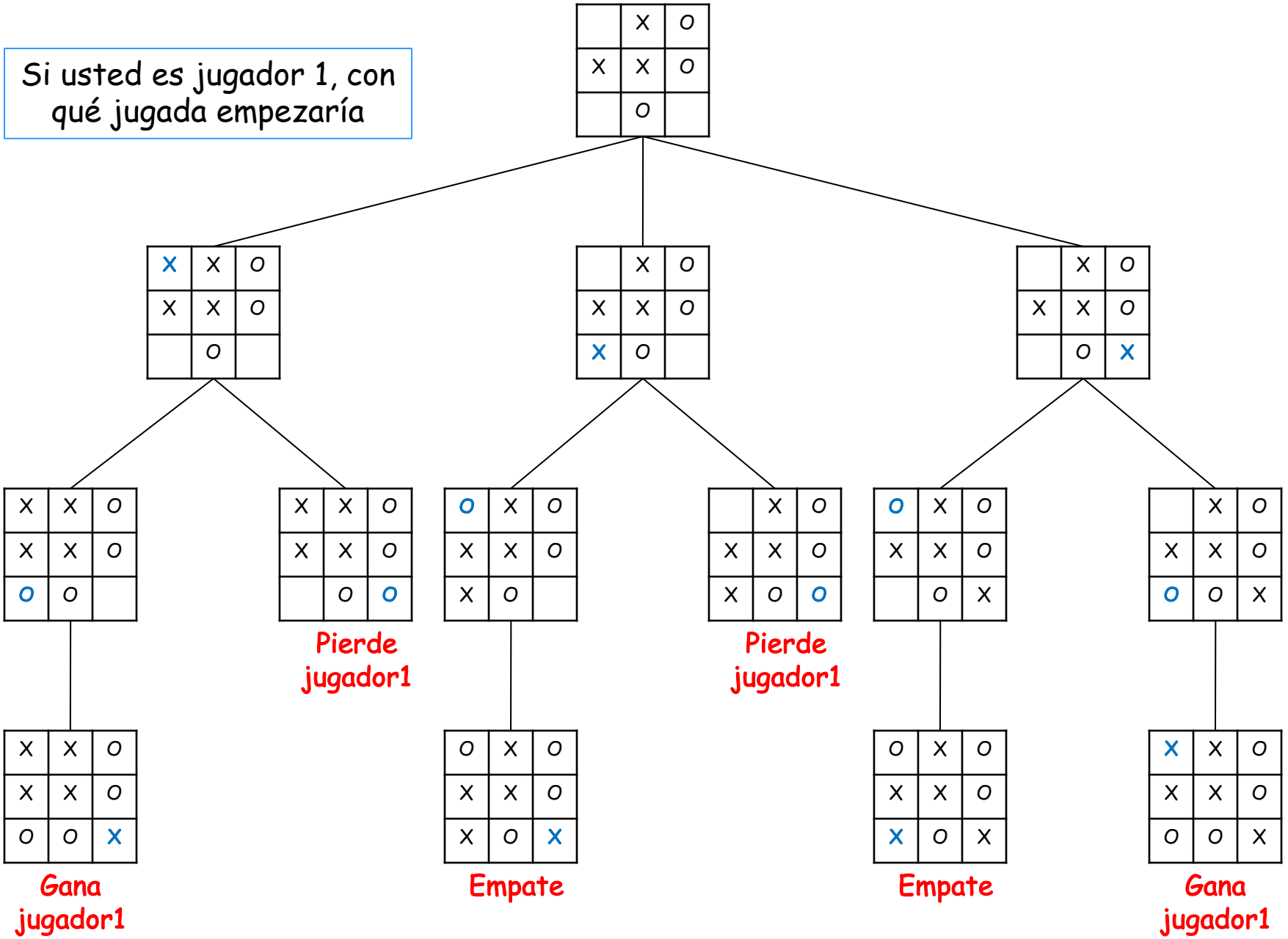

Pierde jugador1
Pierde jugador1
Gana jugador1
Empate


Pierde jugador1
Pierde jugador1
Gana jugador1
Empate
Empate

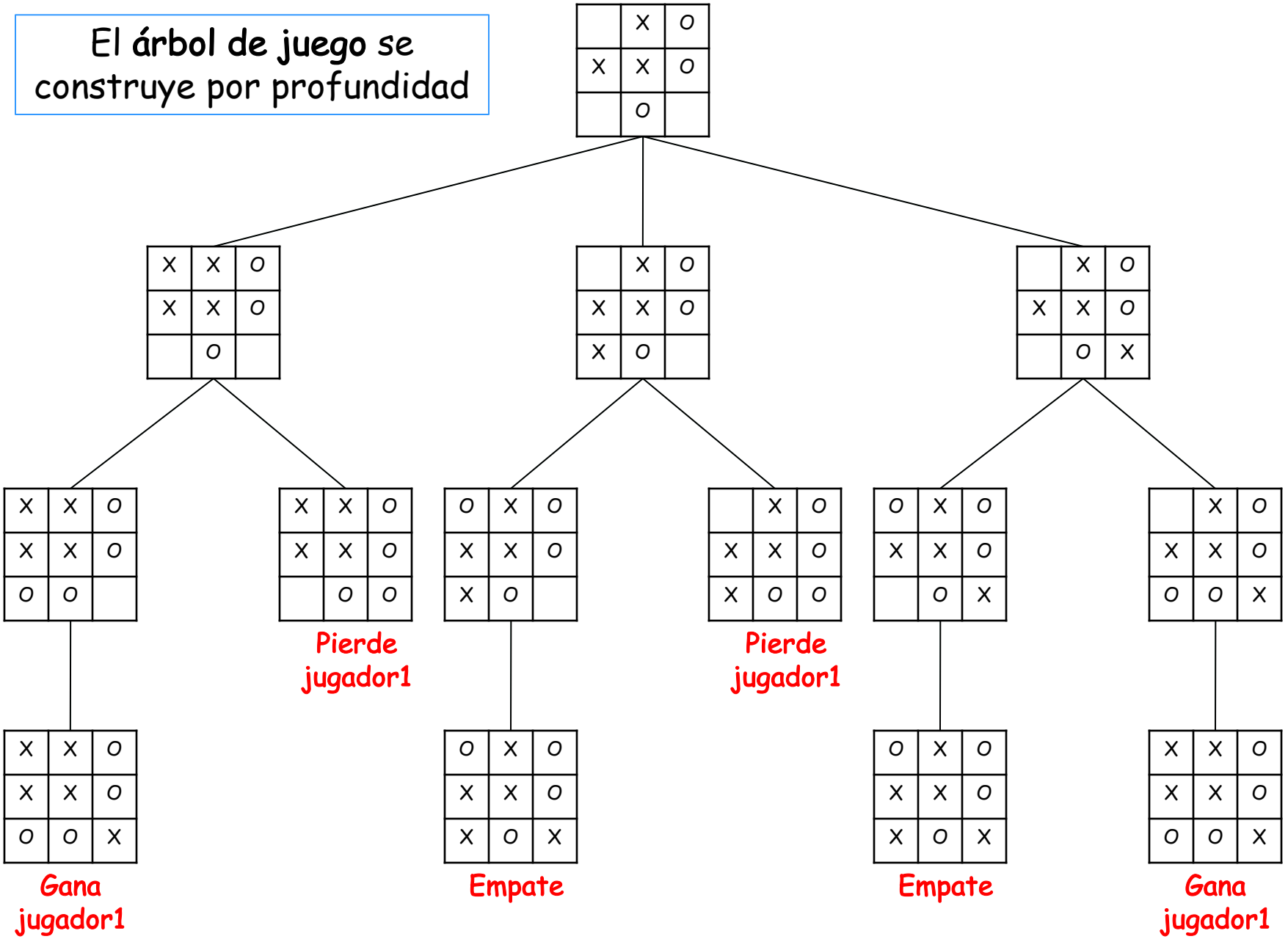

Pierde jugador1
Pierde jugador1
Gana jugador1
Empate
Empate


Pierde jugador1
Pierde jugador1
Gana jugador1
Empate
Empate
Gana jugador1

Si usted es jugador 1, con qué jugada empezaría



El **árbol de juego** se construye por profundidad

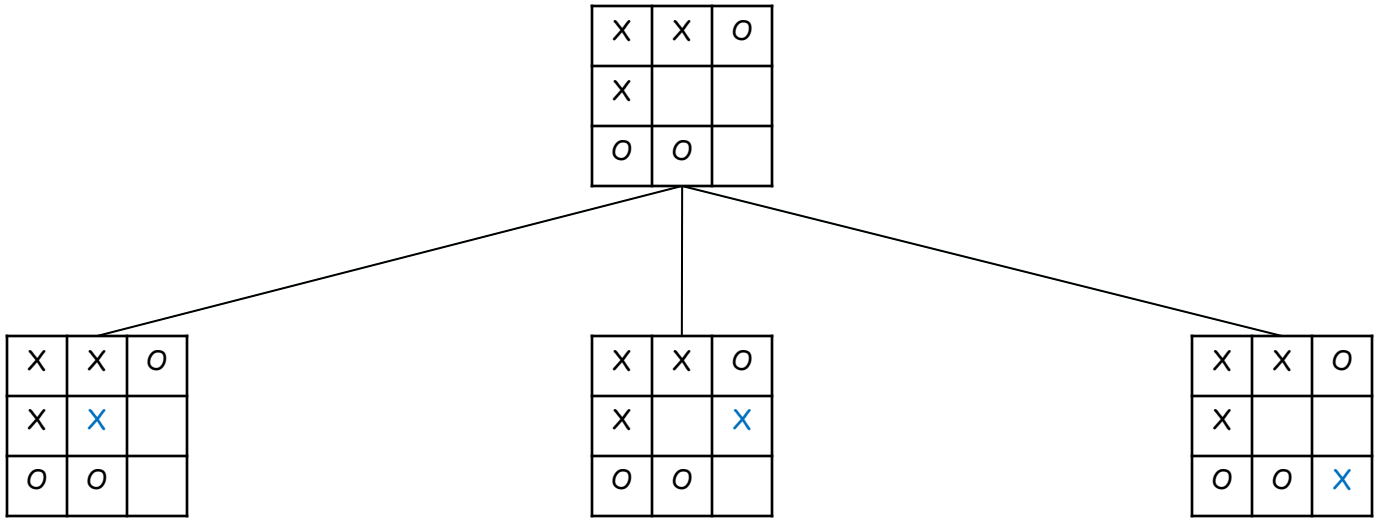


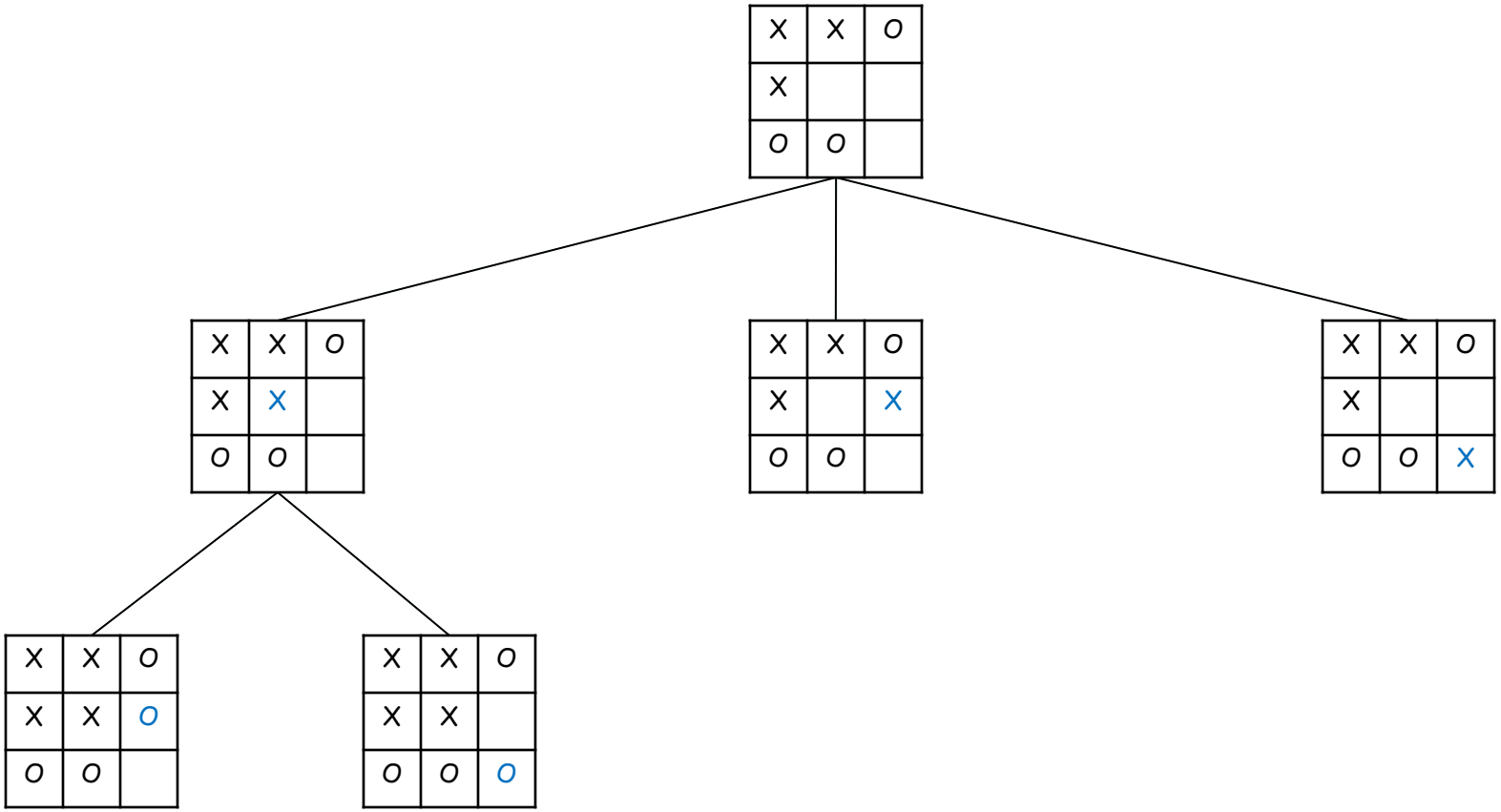
Juegos

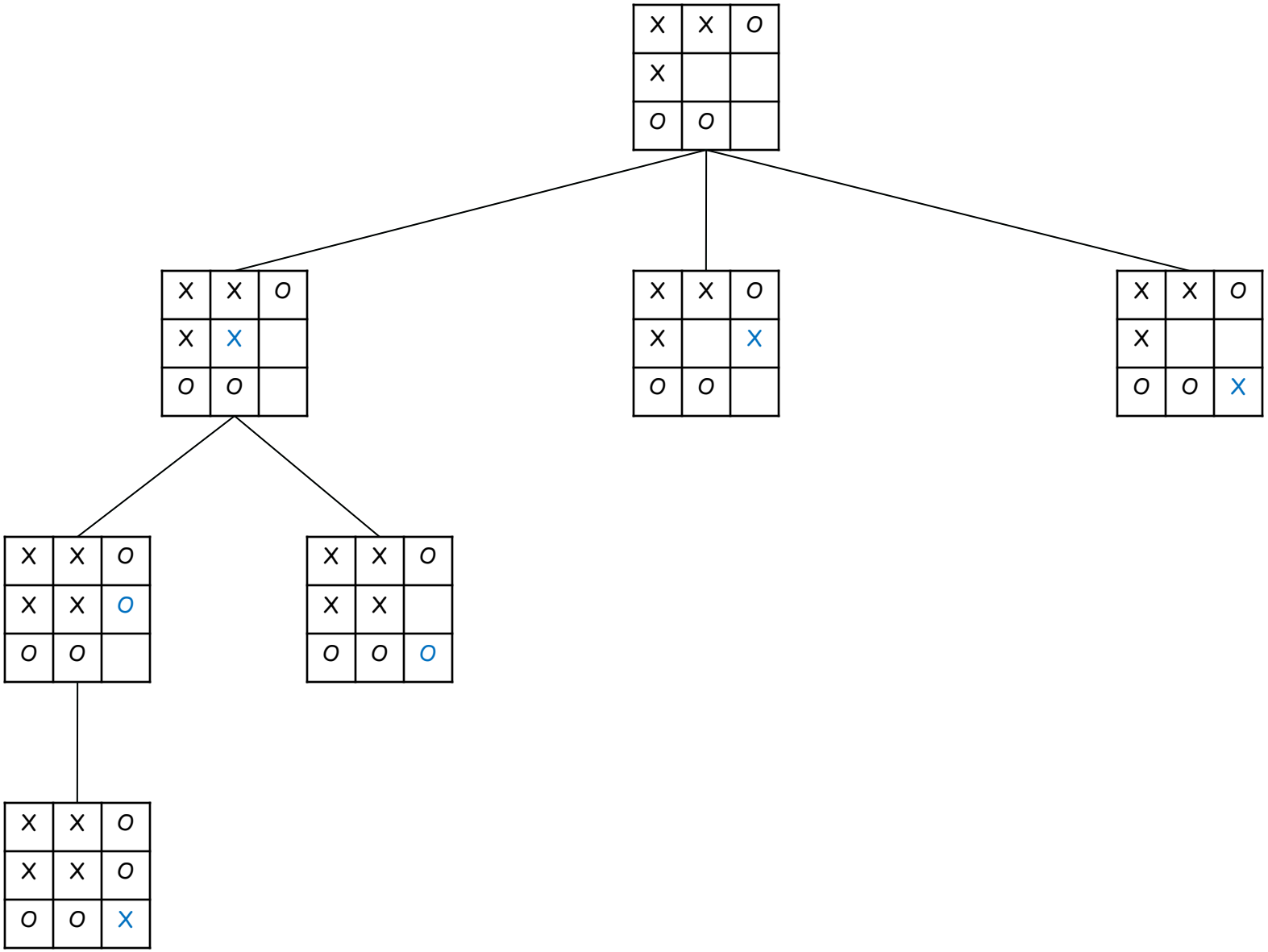
Construir el árbol de juego

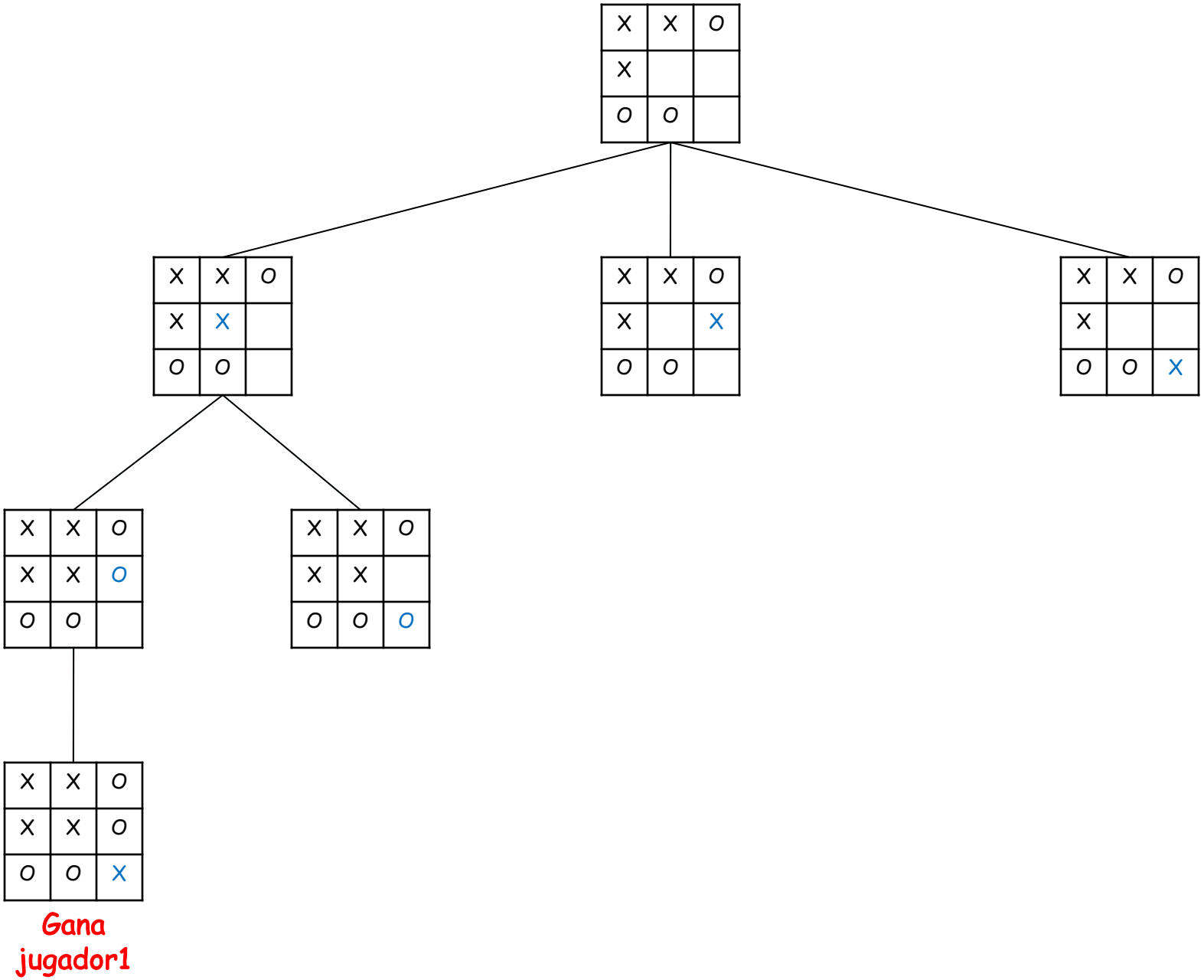
X	X	O
X		
O	O	

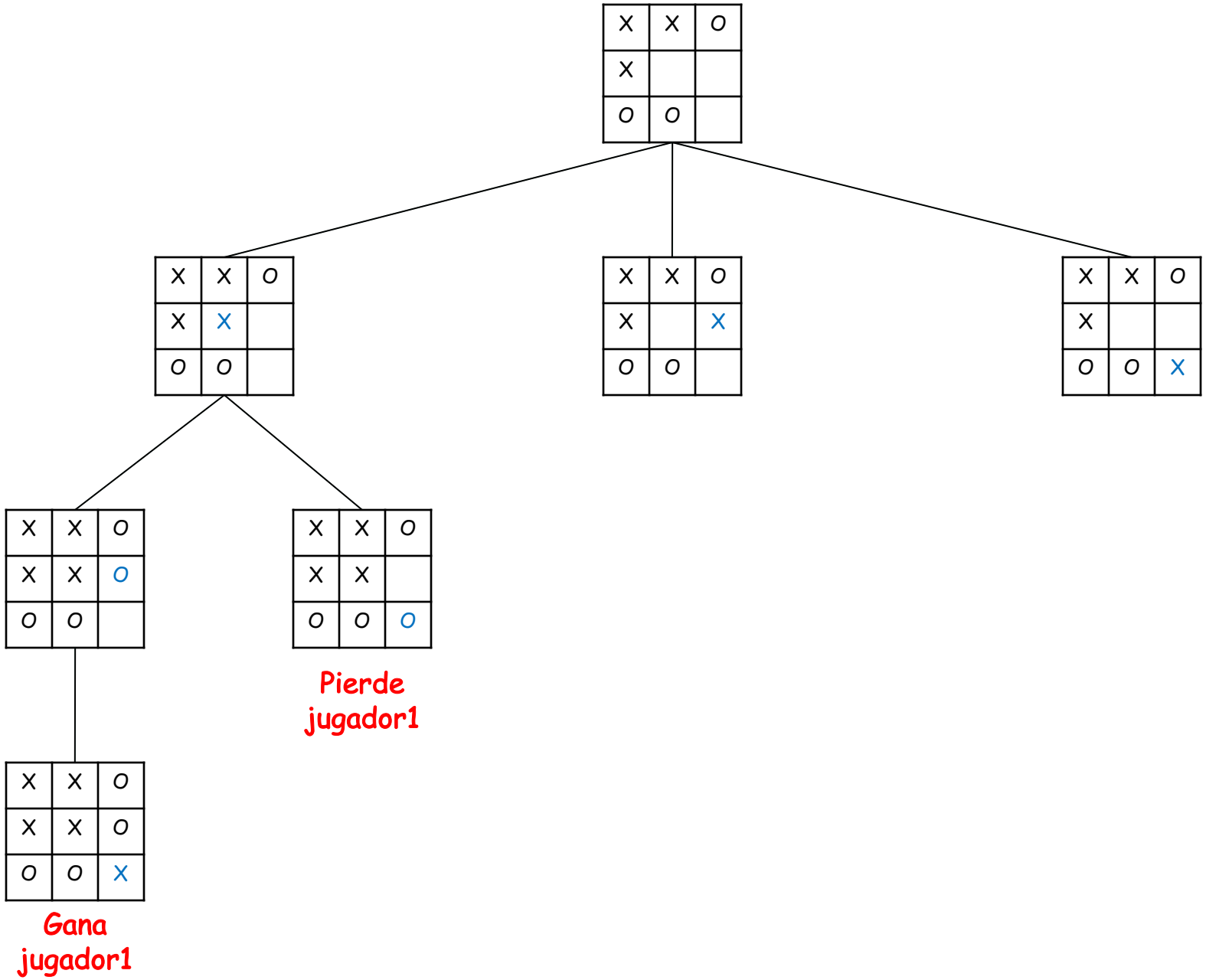
- La jugada es de (X)

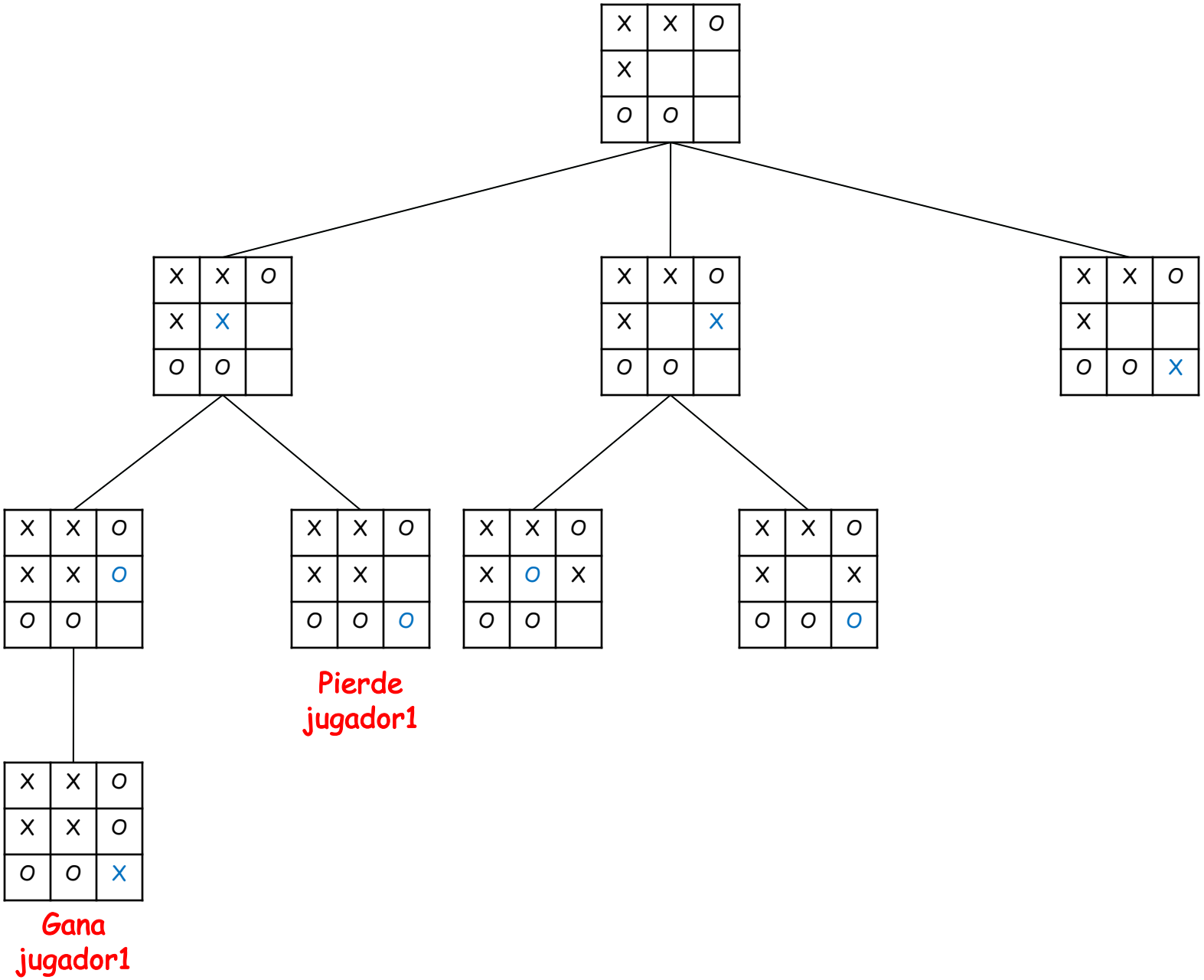

X X O
X
O O
X X O
X X
O O
X X O
X X
O O
X X O
X
O O X

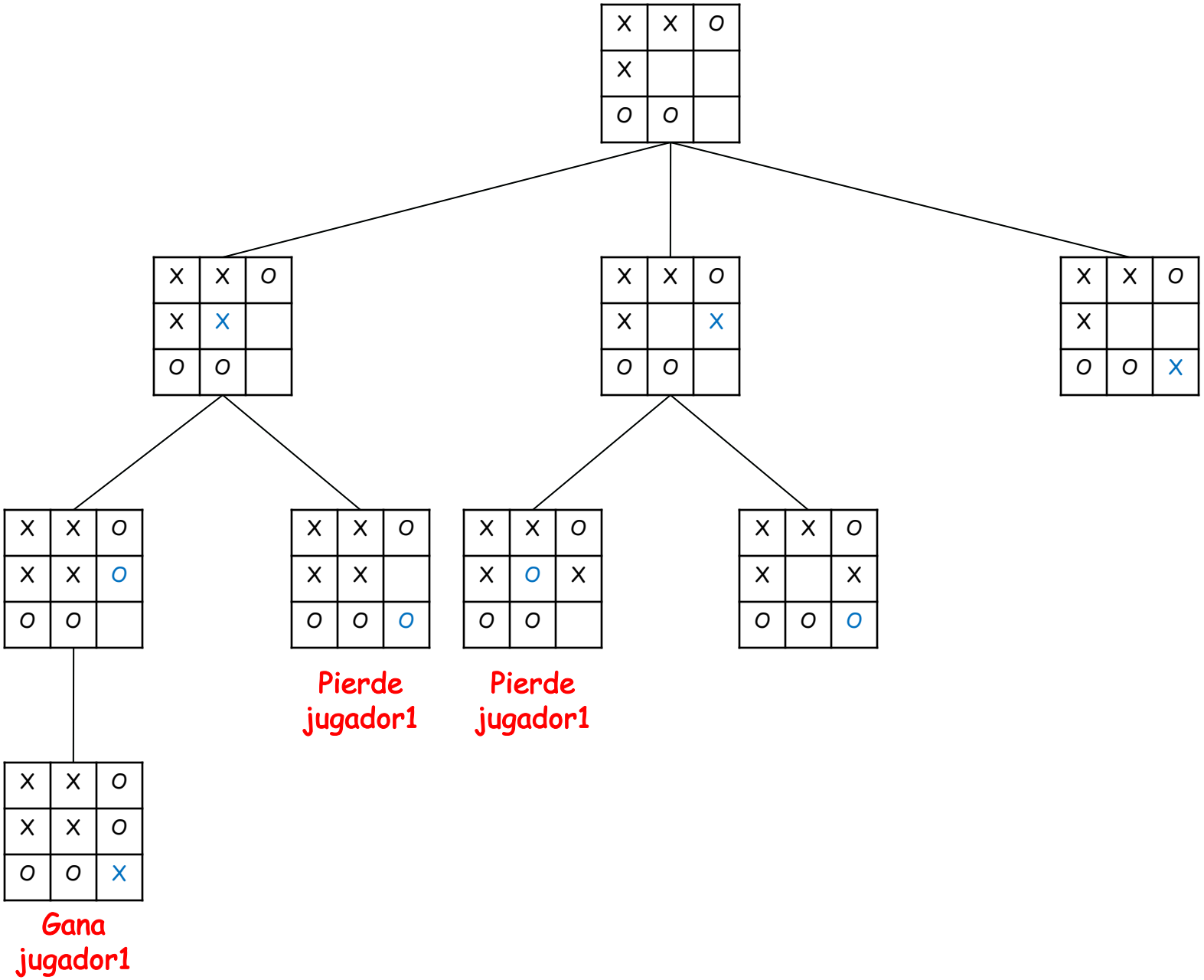

X X O
X
O O
X X O
X X
O O
X X O
X X
O O
X X O
X
O O X
X X O
X X O
O O
X X O
X X
O O O

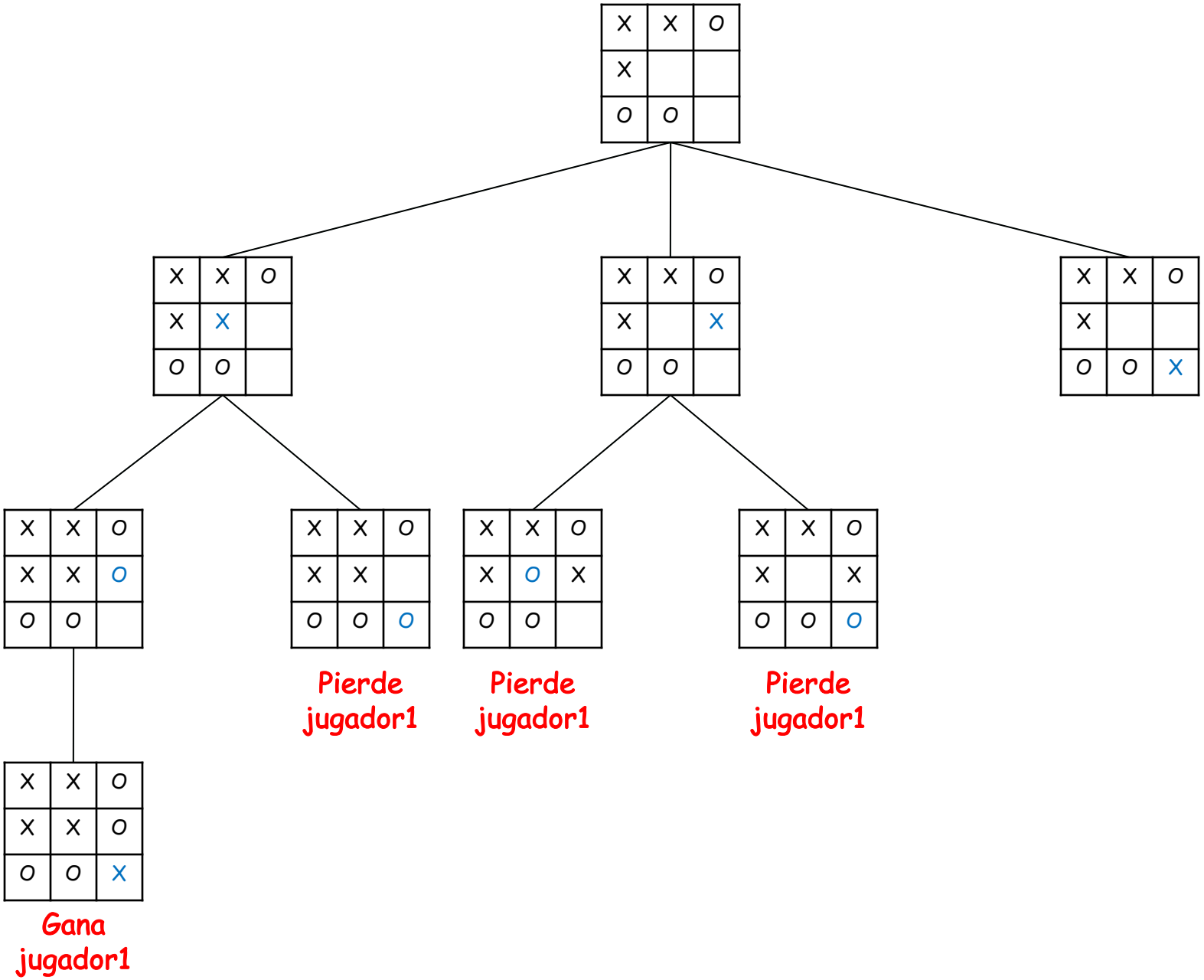

X X O
X
O O
X X O
X X
O O
X X O
X O
O O
X X O
X
O O X
X X O
X X O
O O
X X O
X X
O O O
X X O
X X O
O O X

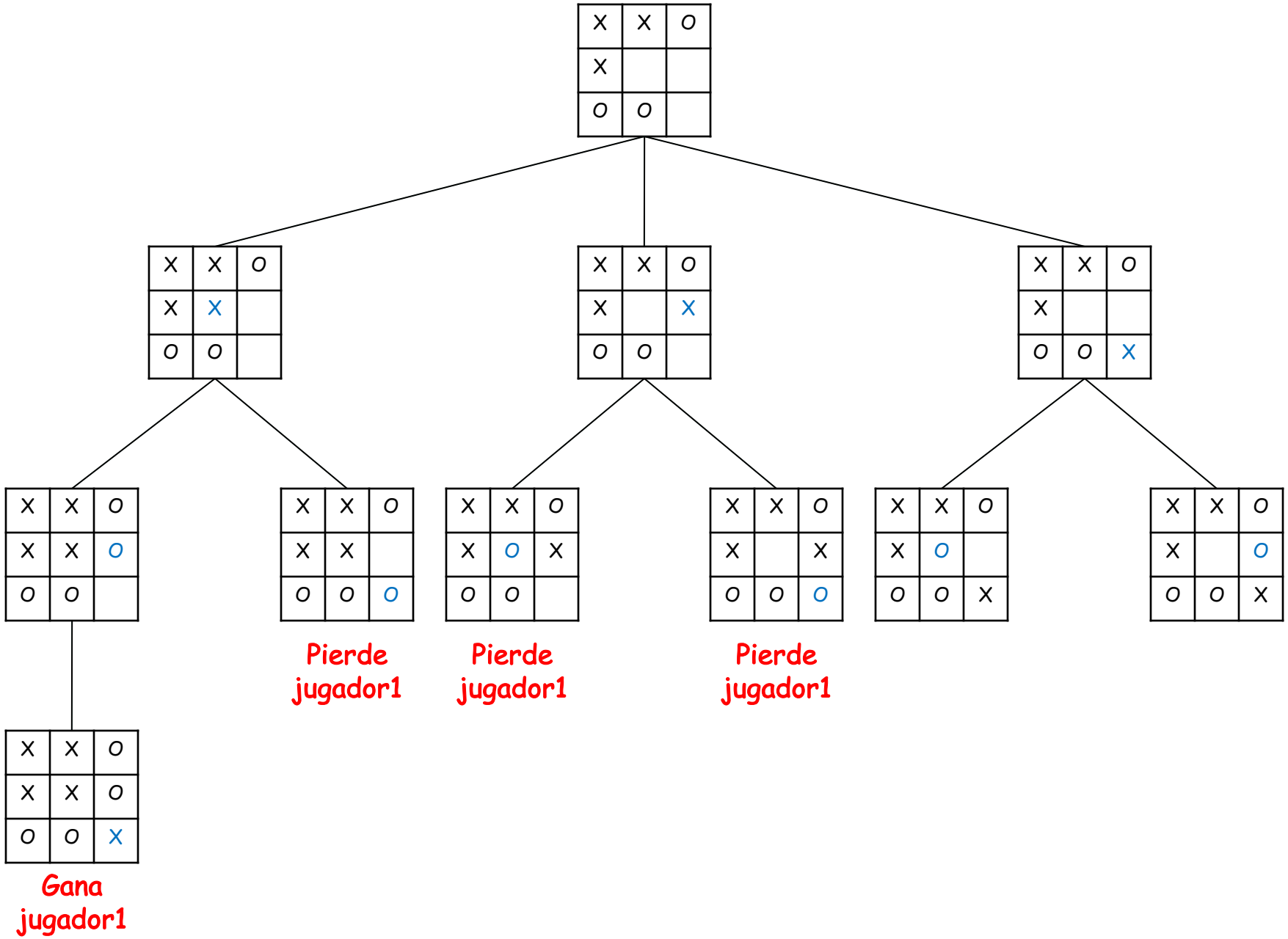

Gana
jugador1

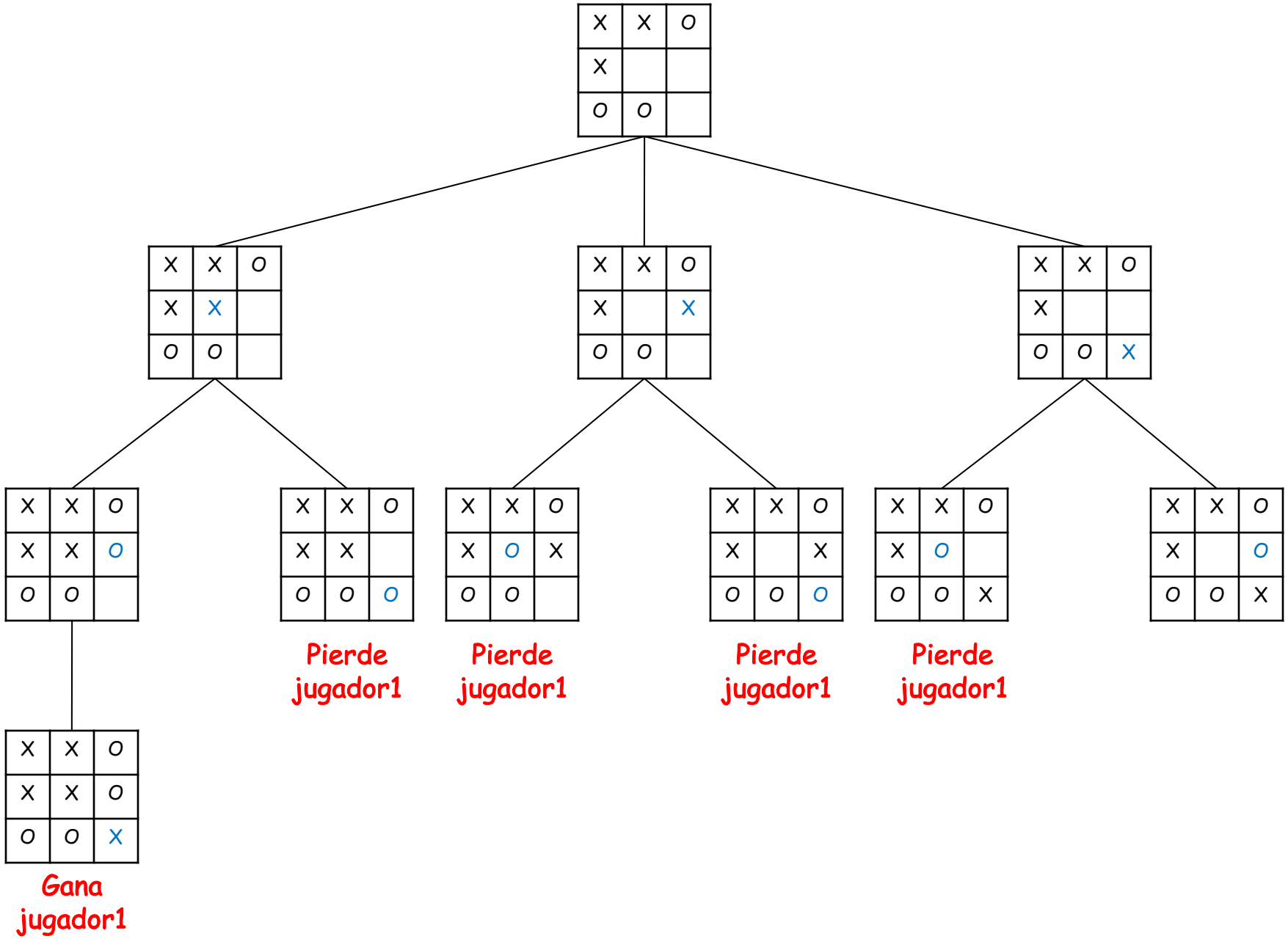

Pierde
jugador1
Gana
jugador1

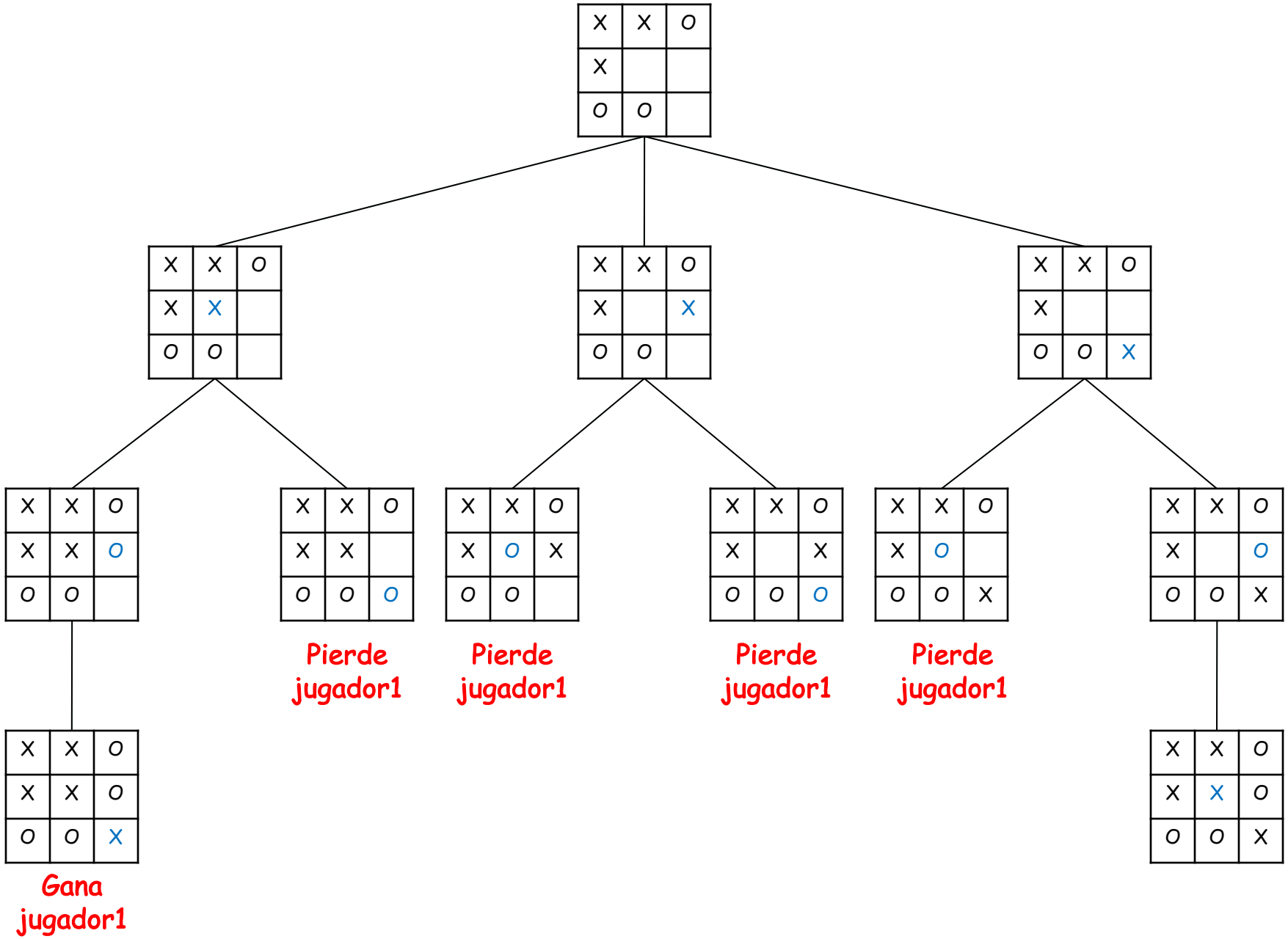

Pierde
jugador1
Gana
jugador1

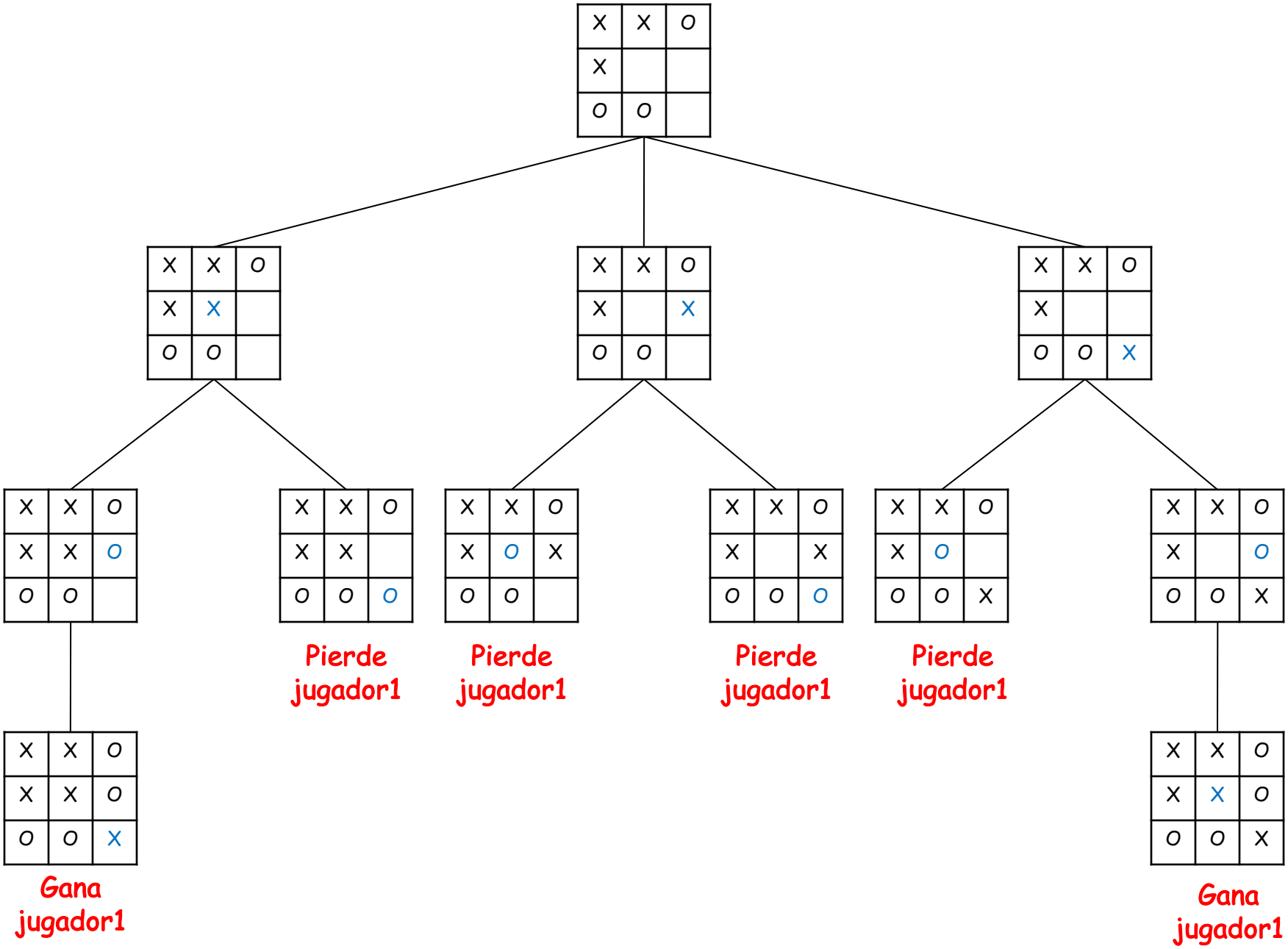

Pierde jugador1
Pierde jugador1
Gana jugador1


Pierde jugador1
Pierde jugador1
Pierde jugador1
Gana jugador1

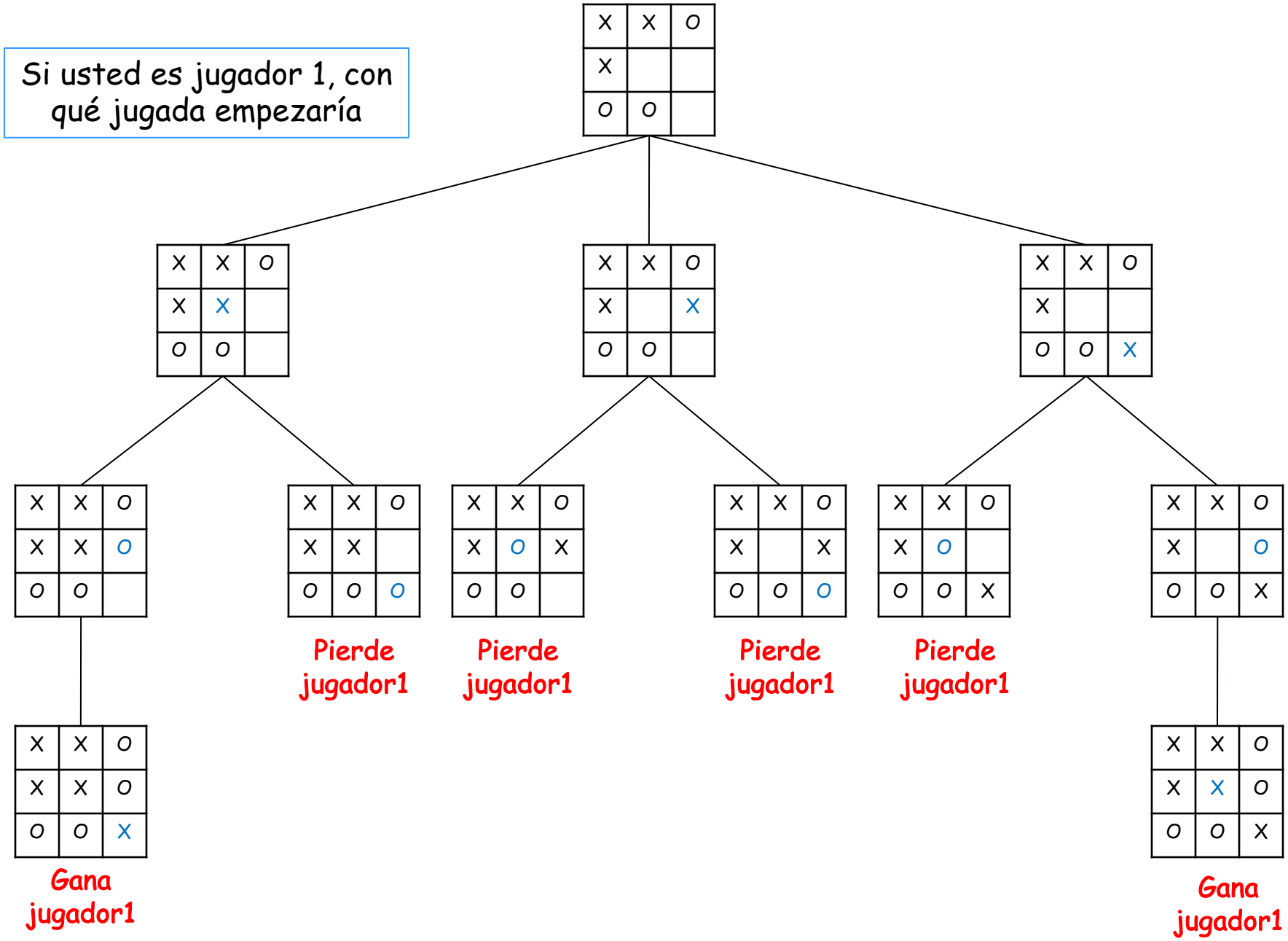

Pierde jugador1
Pierde jugador1
Pierde jugador1
Gana jugador1


Pierde jugador1
Pierde jugador1
Pierde jugador1
Pierde jugador1
Gana jugador1


Pierde jugador1
Pierde jugador1
Pierde jugador1
Pierde jugador1
Gana jugador1

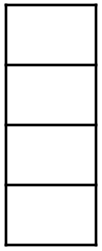

Pierde jugador1
Pierde jugador1
Pierde jugador1
Pierde jugador1
Gana jugador1
Gana jugador1

Si usted es jugador 1, con qué jugada empezaría



Árboles

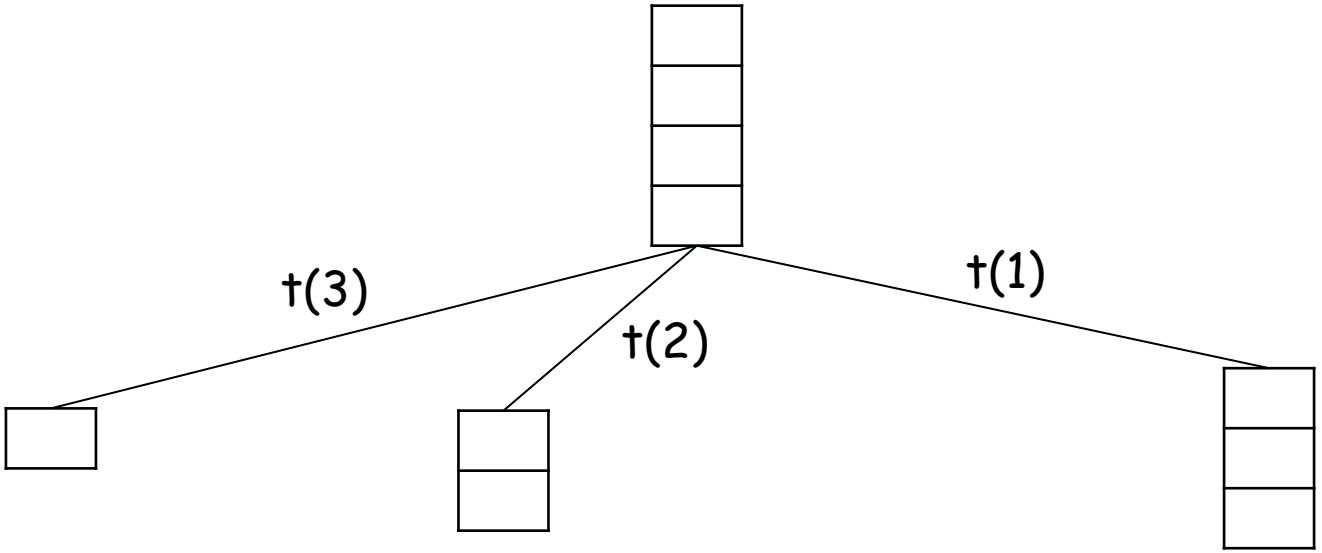
Construir el árbol de juego



• **El juego del NIM**. Se tiene una pila de 4 fichas de la cual cada jugador puede tomar 3, 2 ó 1. El objetivo de cada jugador es obligar a su adversario a tomar la última ficha. Como los elementos están apilados, solo se pueden tomar fichas de su tope

https://www.archimedes-lab.org/game_nim/play_nim_game.html

J1

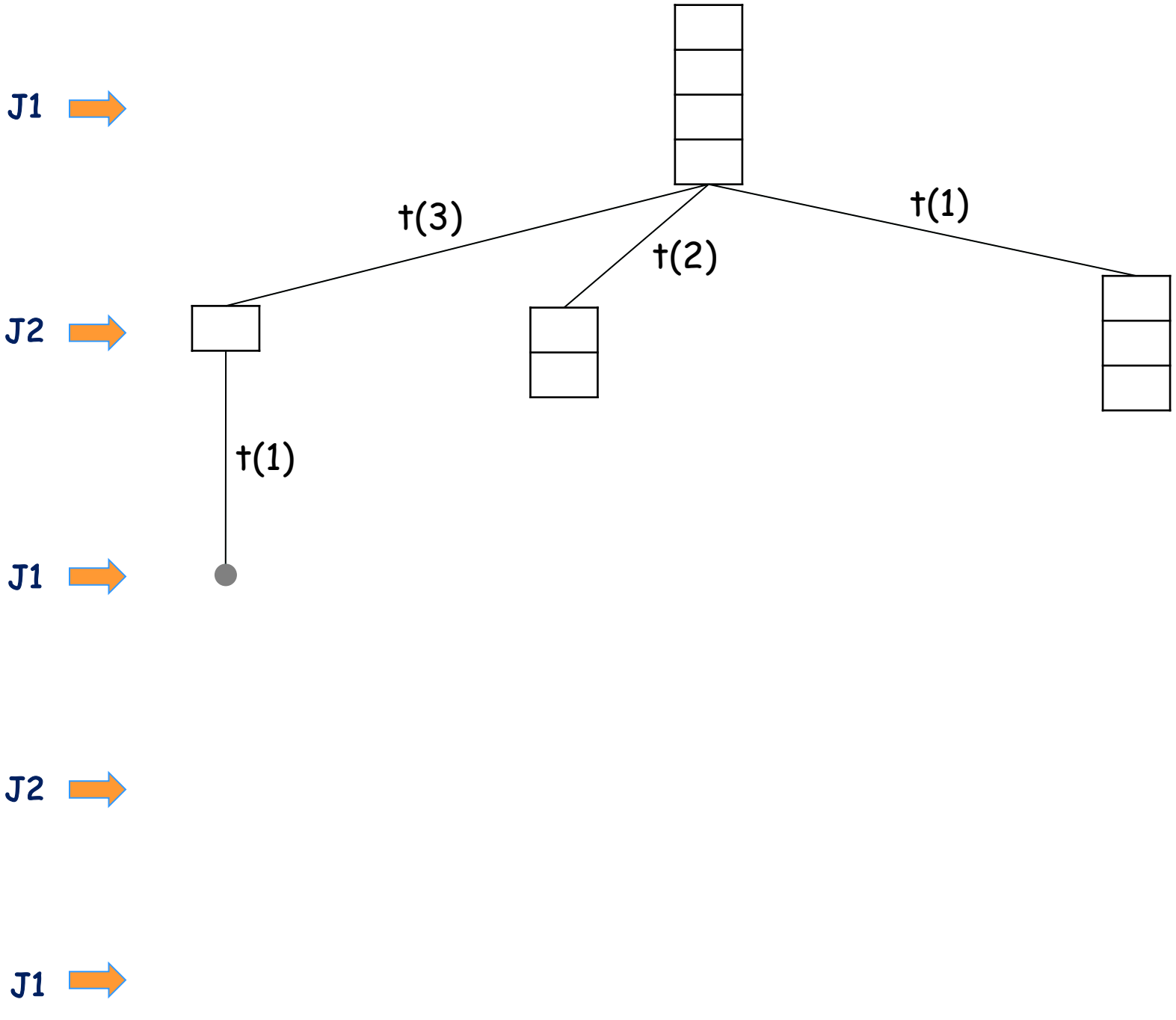


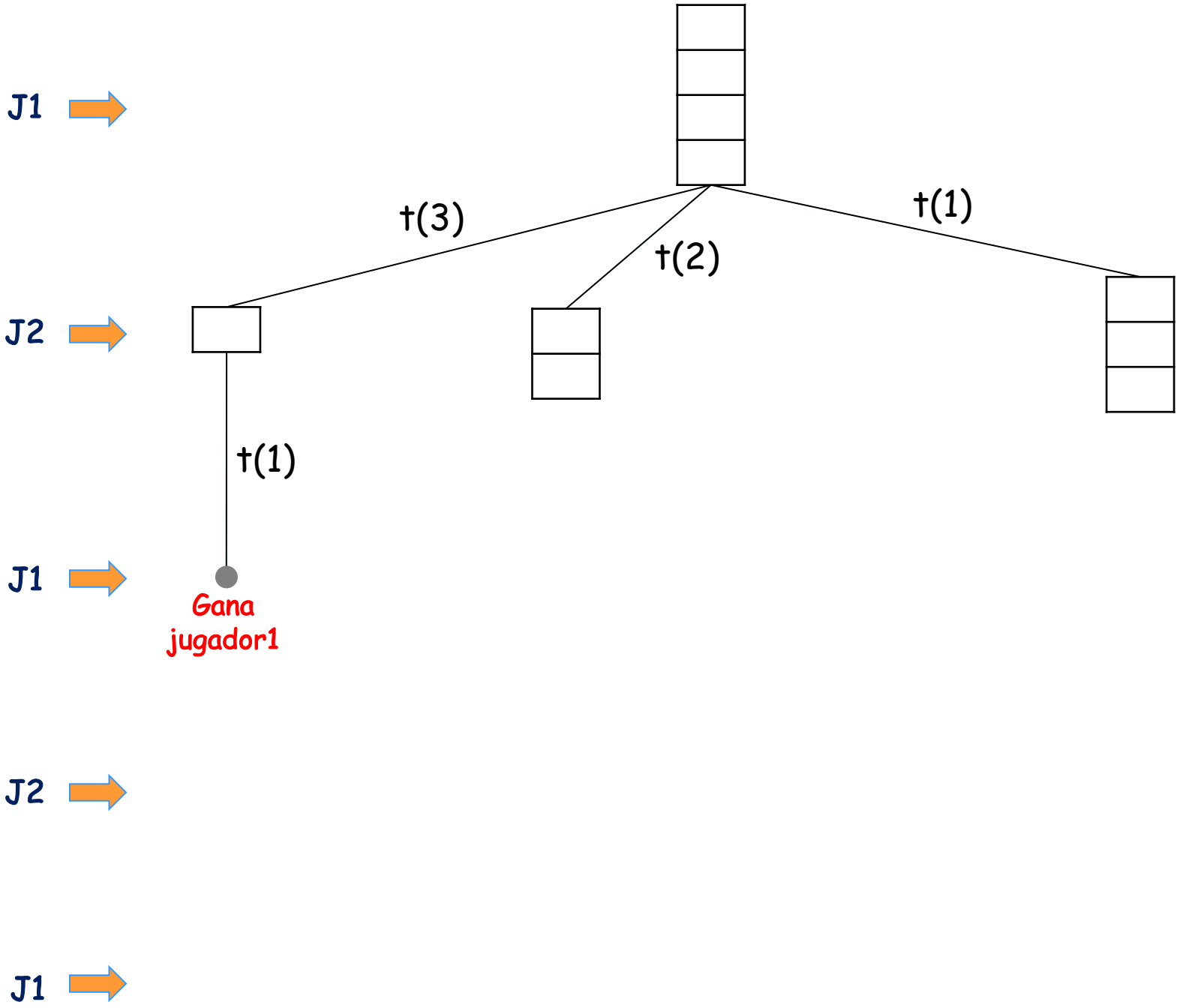
J2

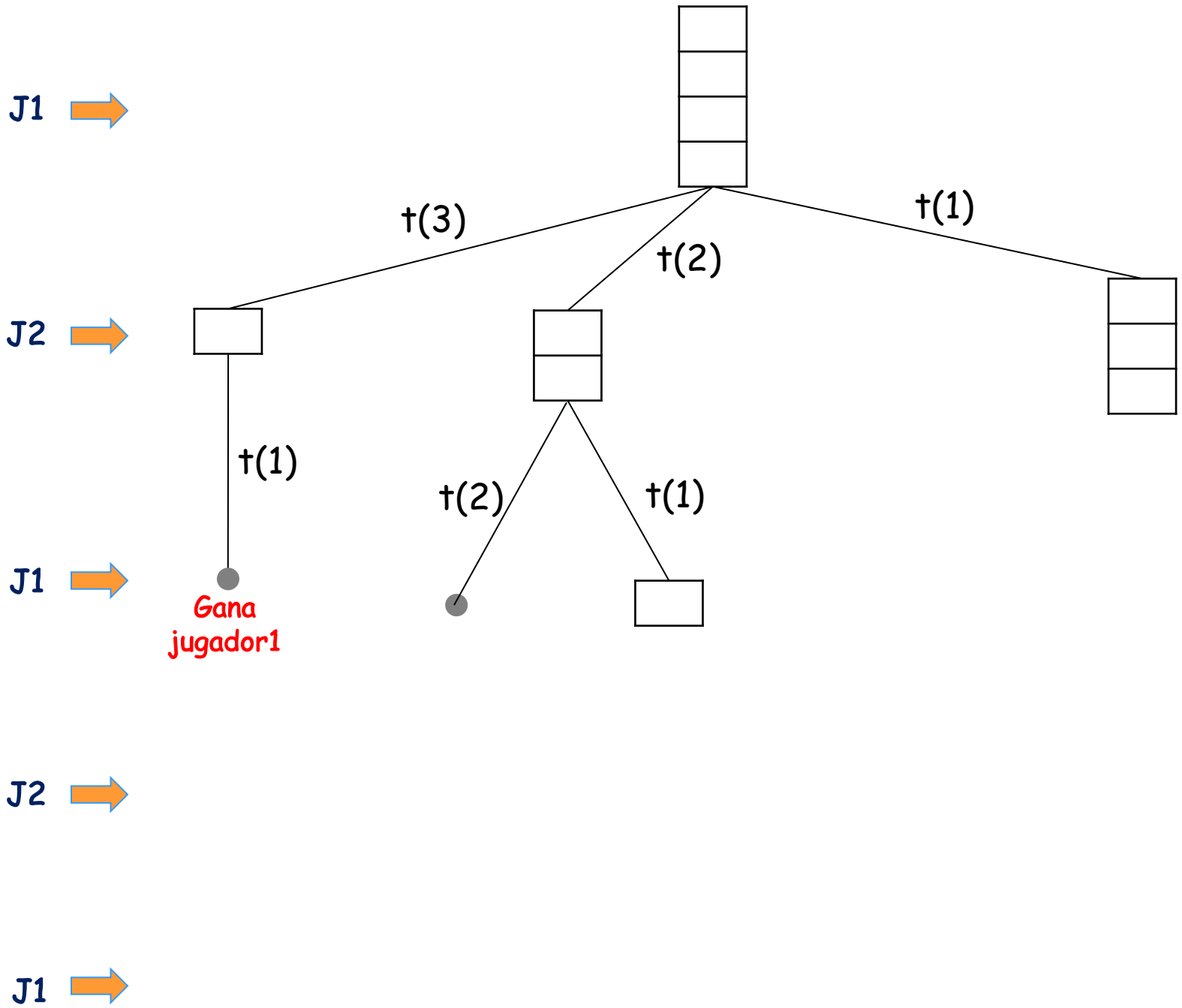
J1

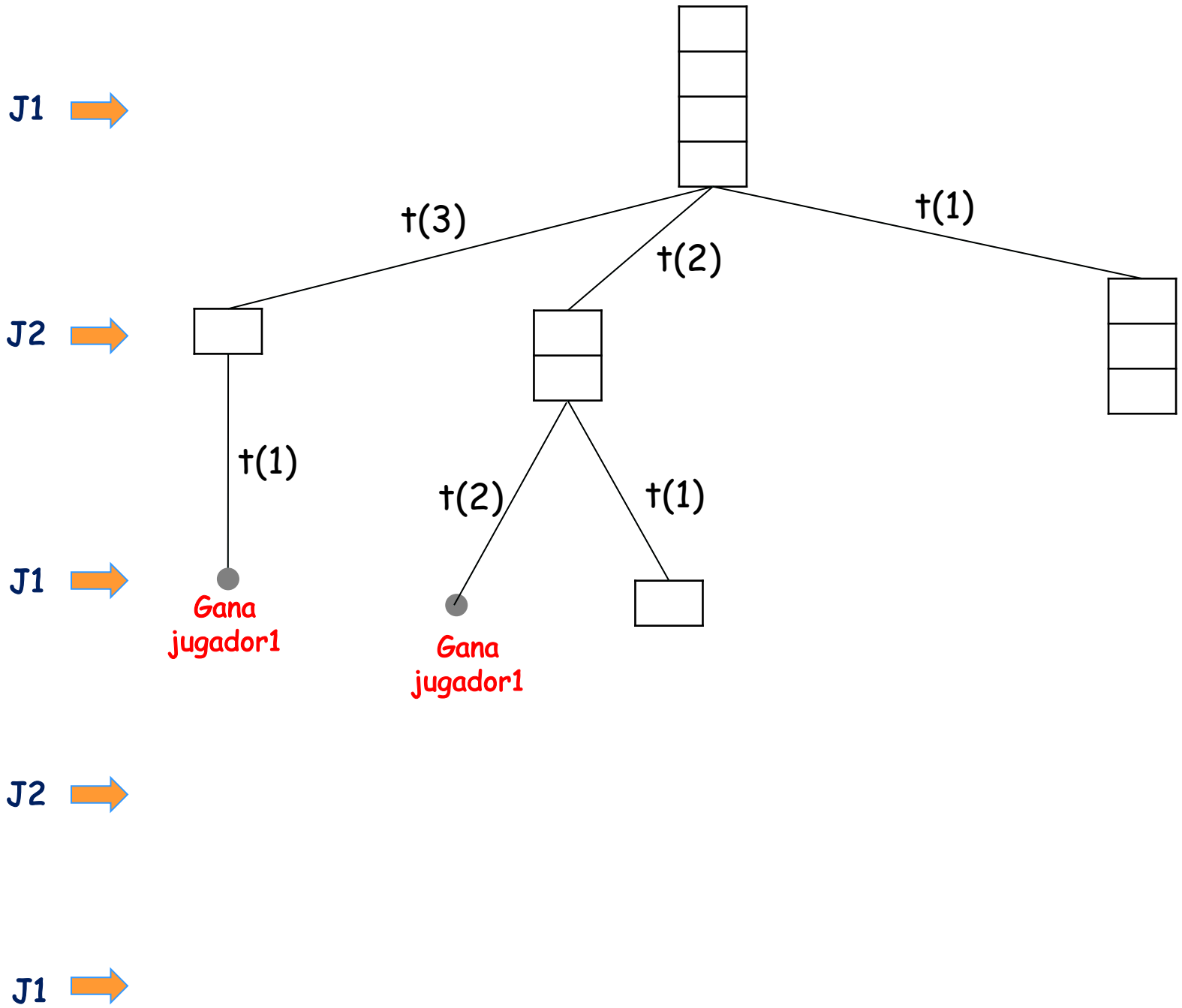
J2

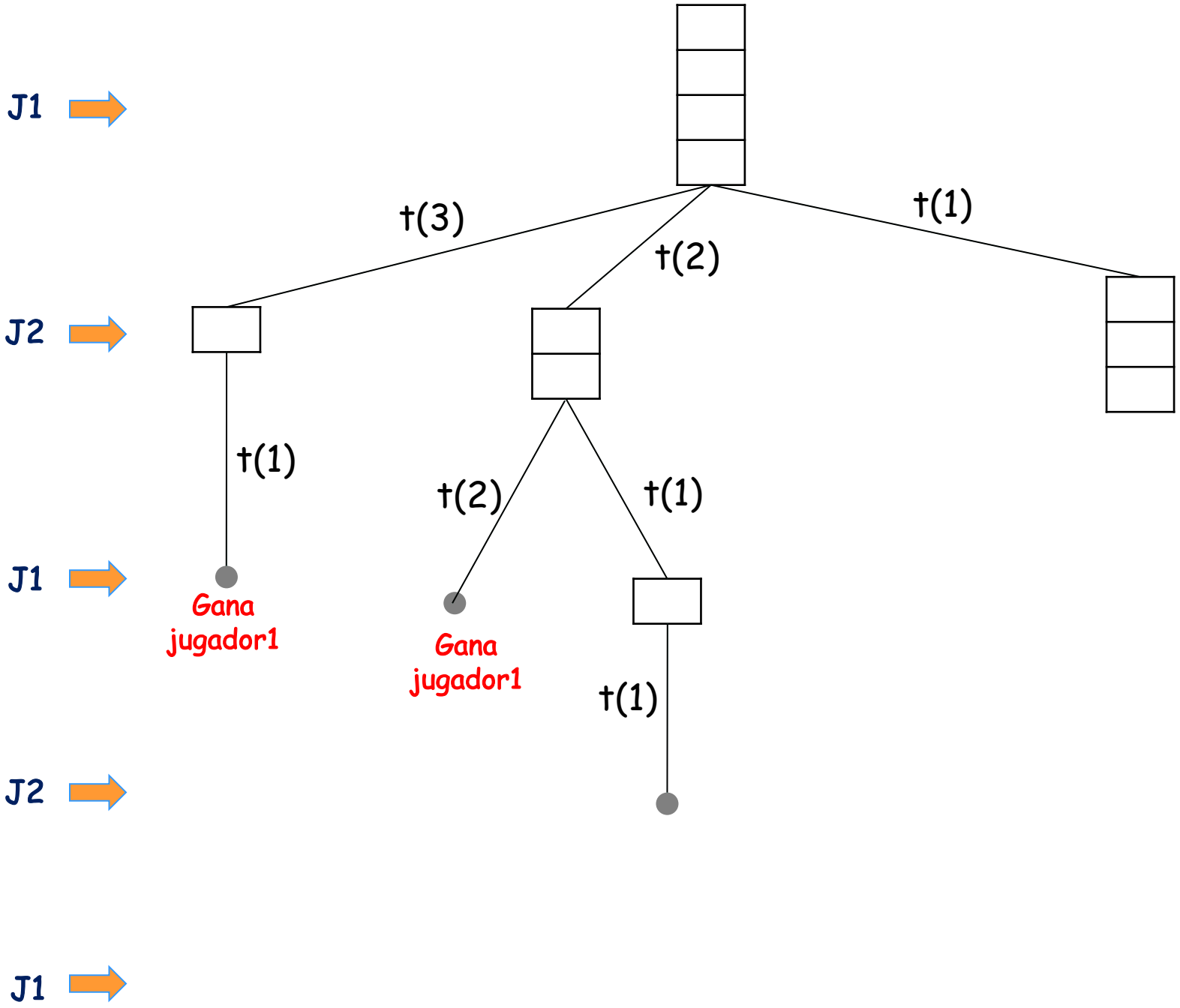
J1

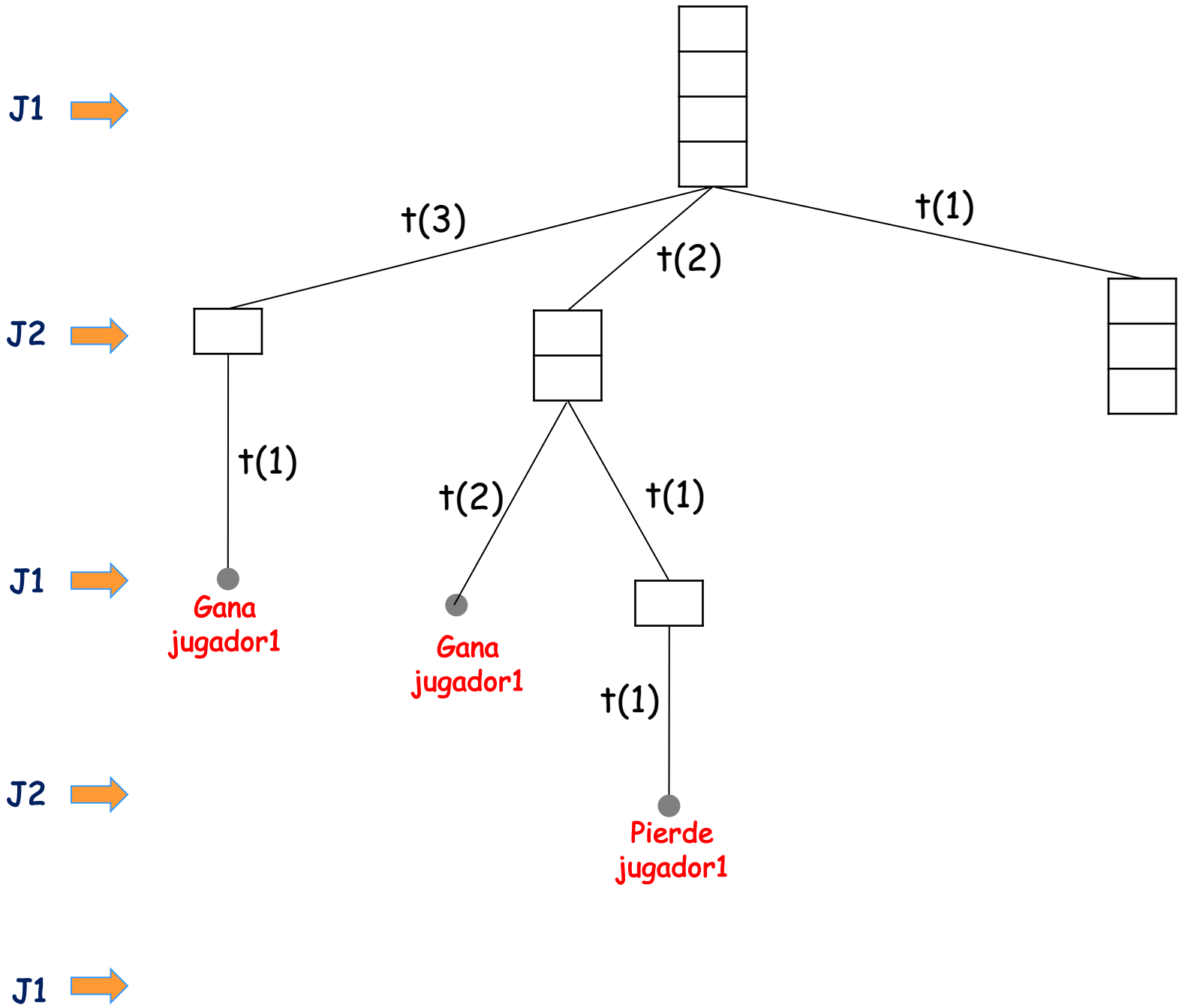

J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)

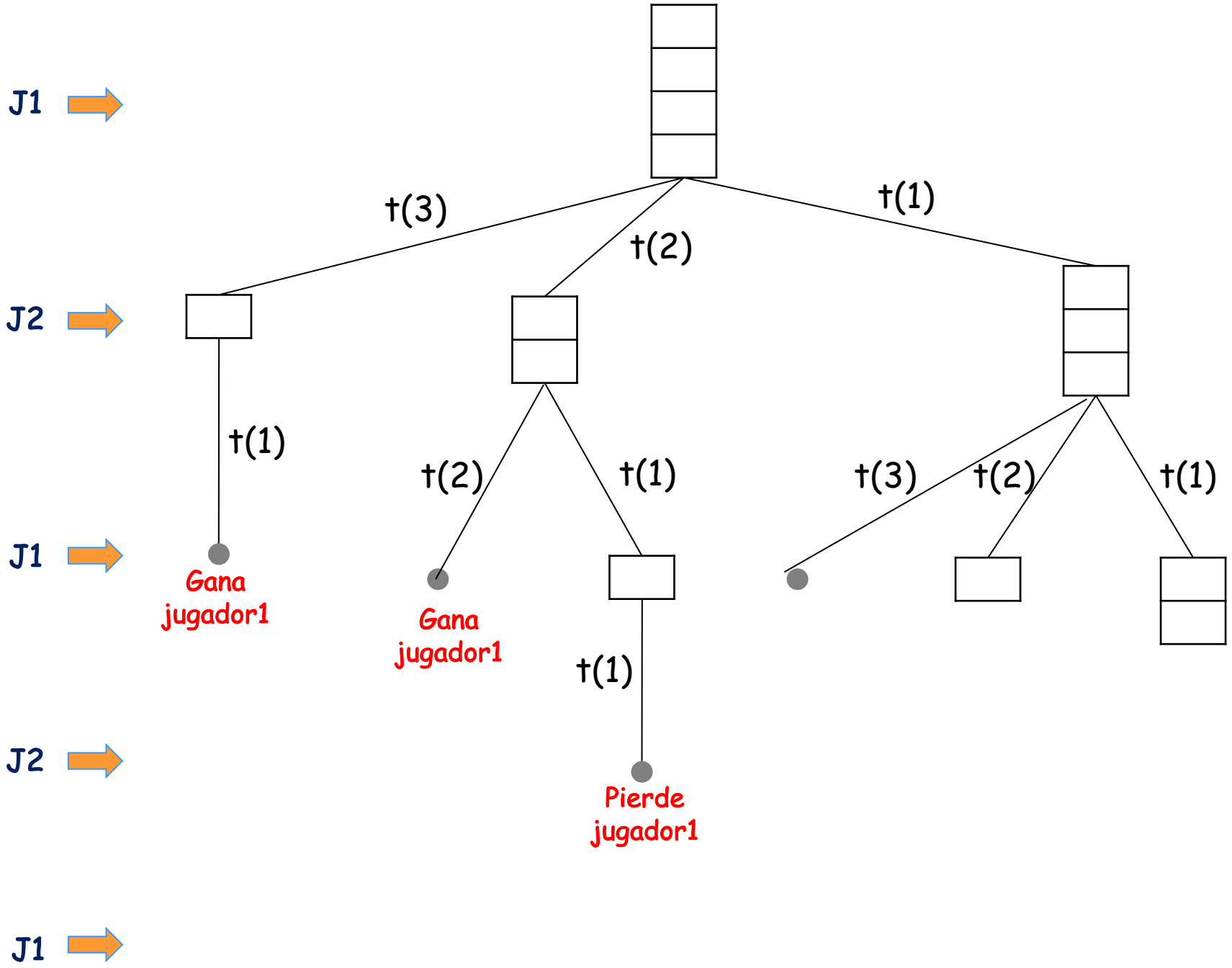

J1
t(3)
t(2)
t(1)
J2
t(1)
J1
Gana
jugador1
J2
J1

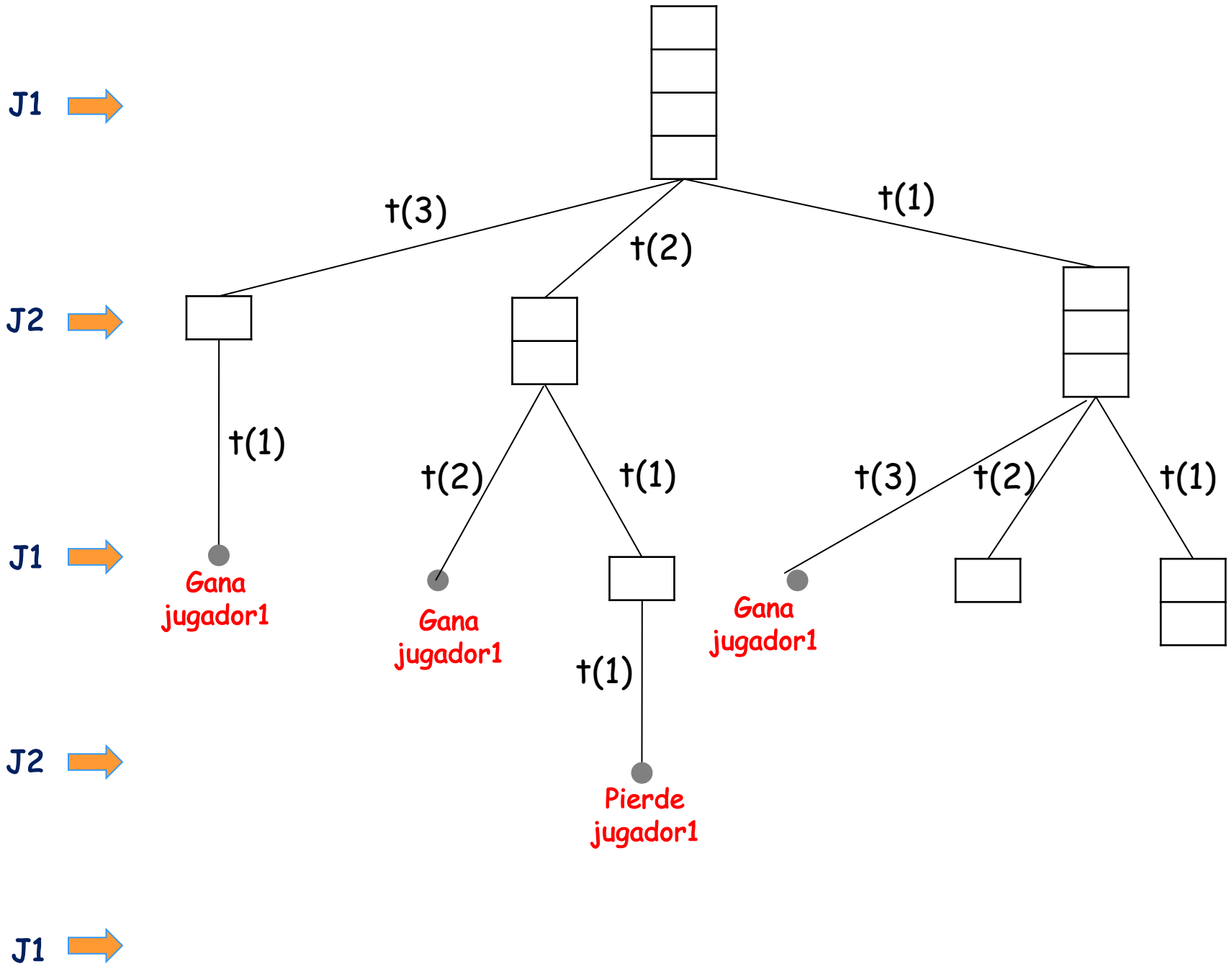

J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
Gana
jugador1

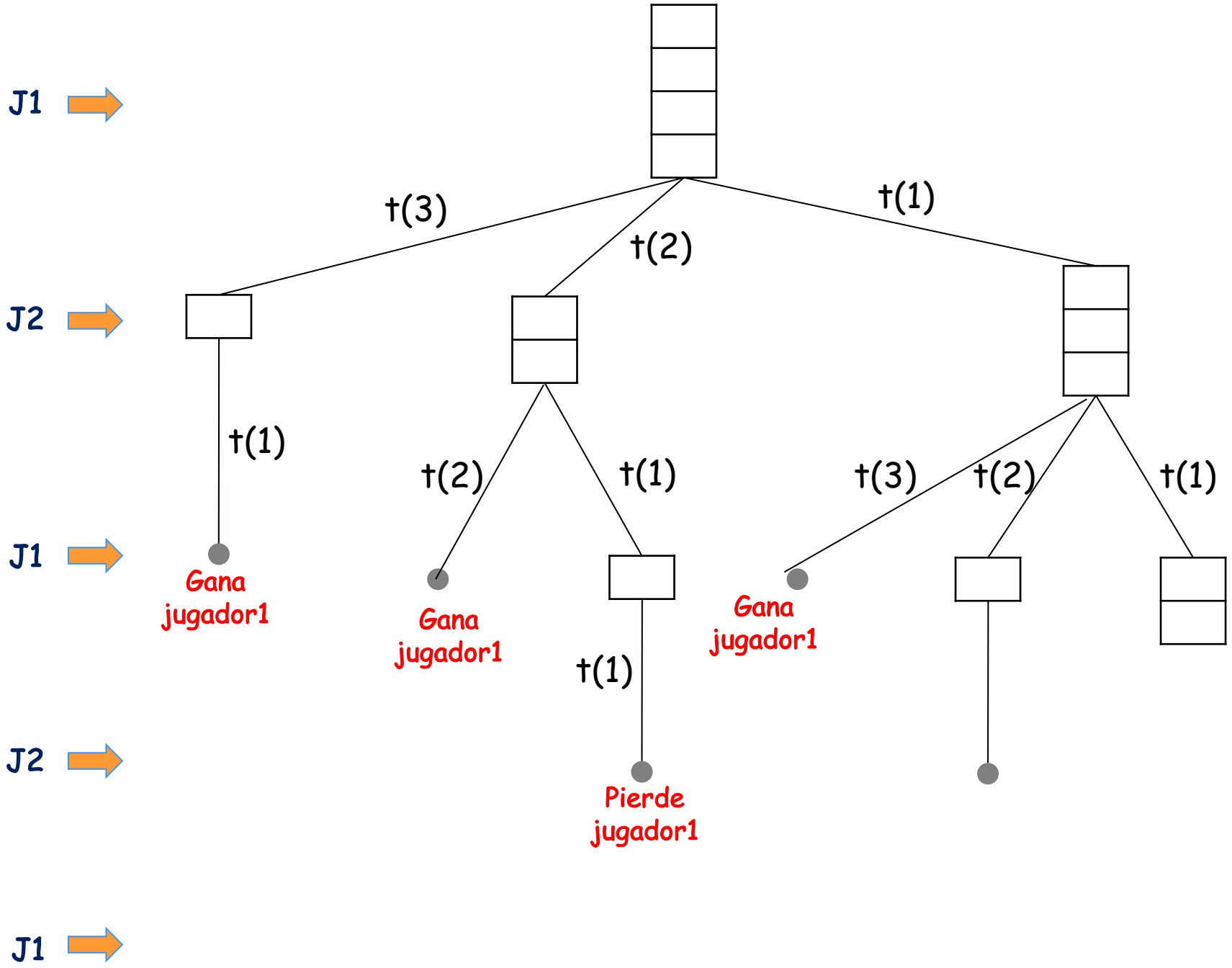

J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
Gana
jugador1
Gana
jugador1

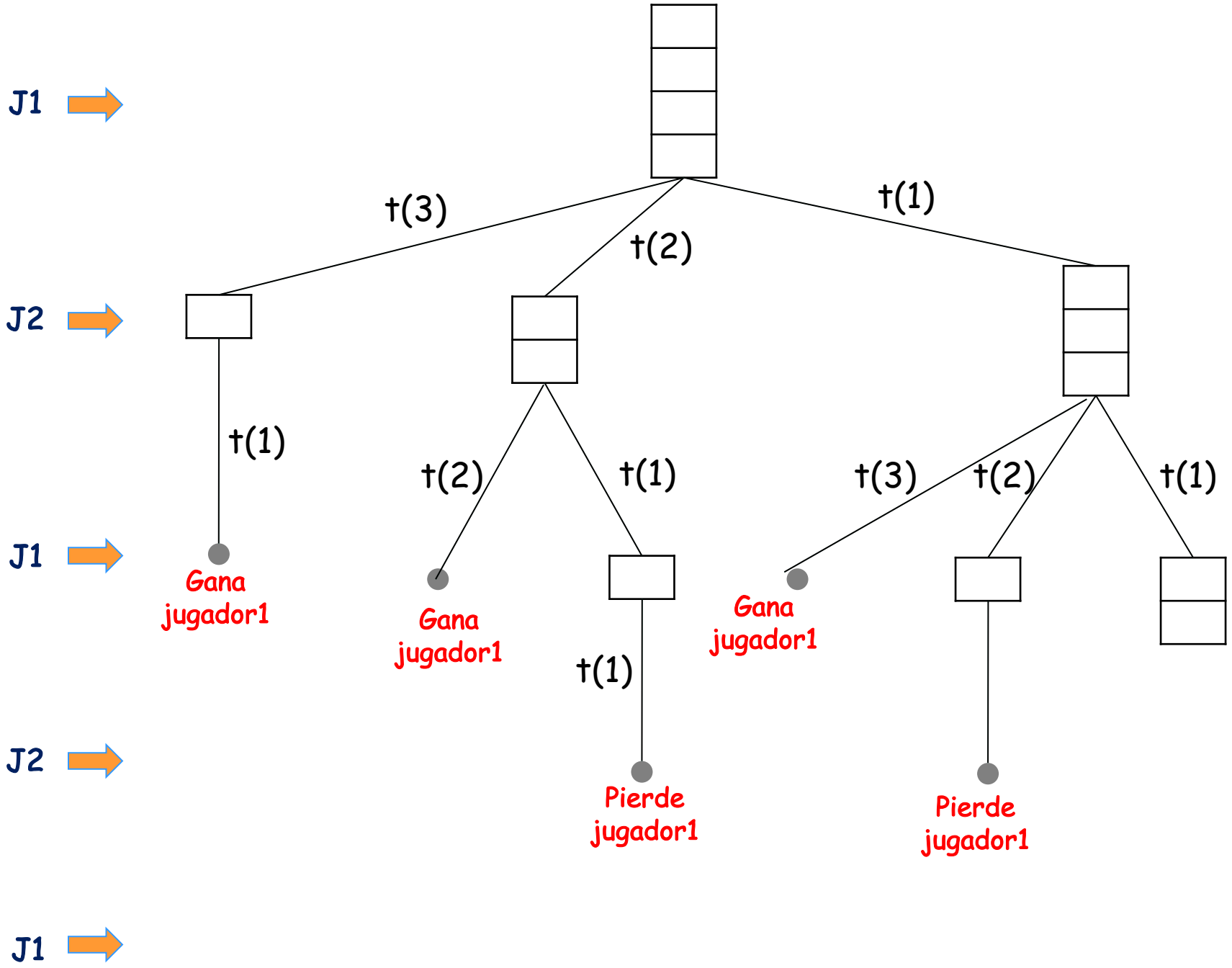

J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1

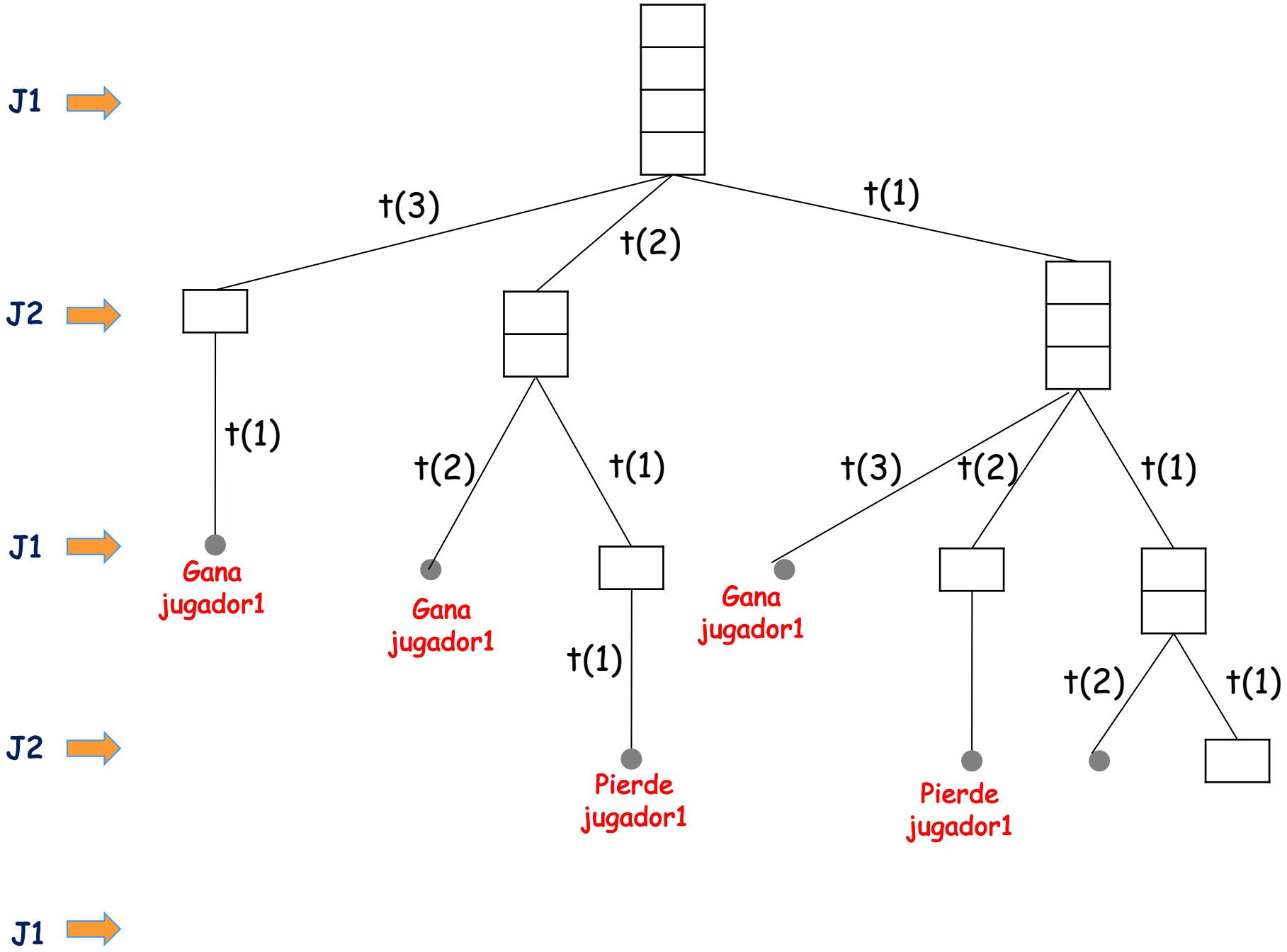

J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1
Pierde jugador1

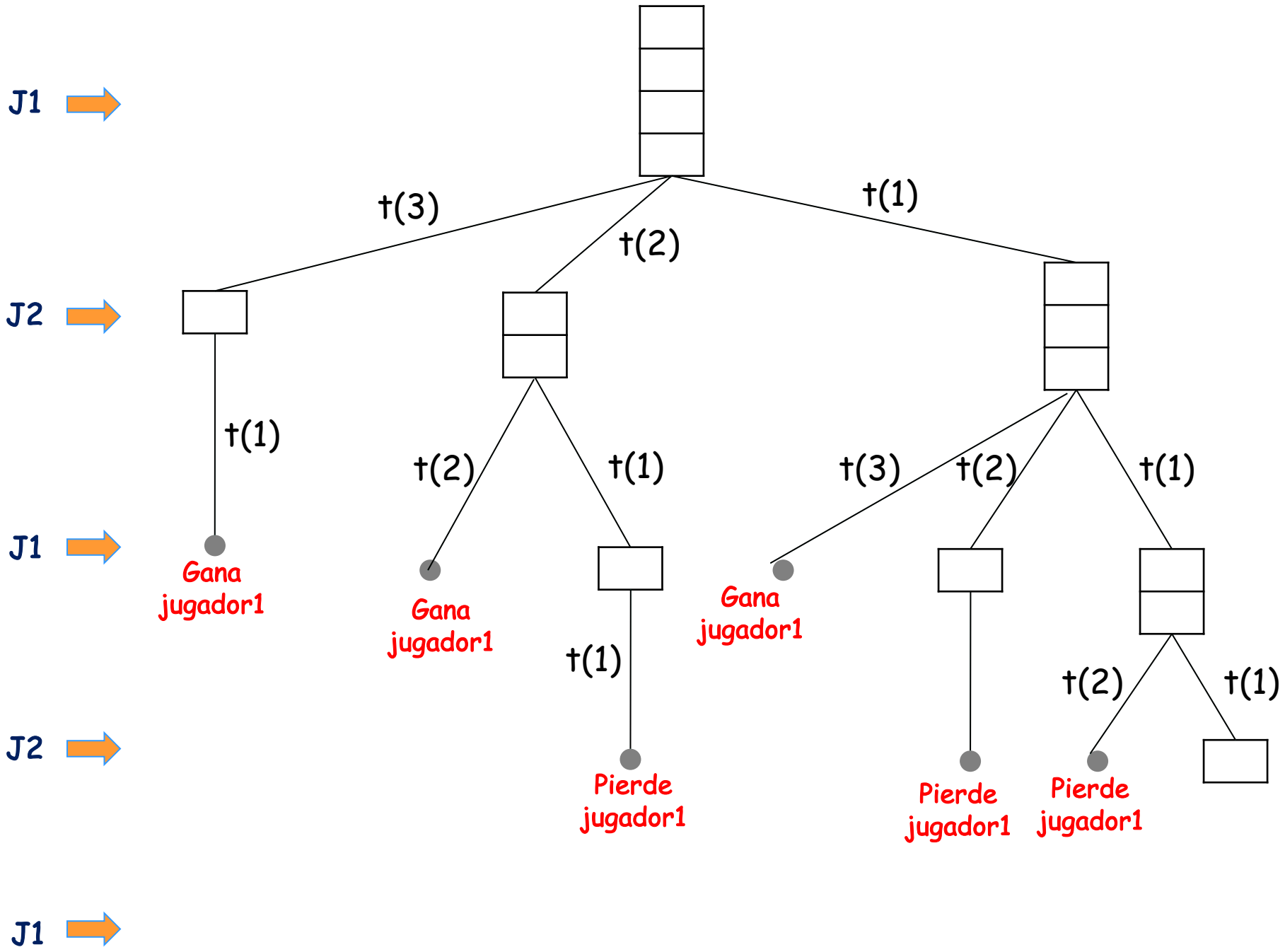

J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1
Pierde jugador1

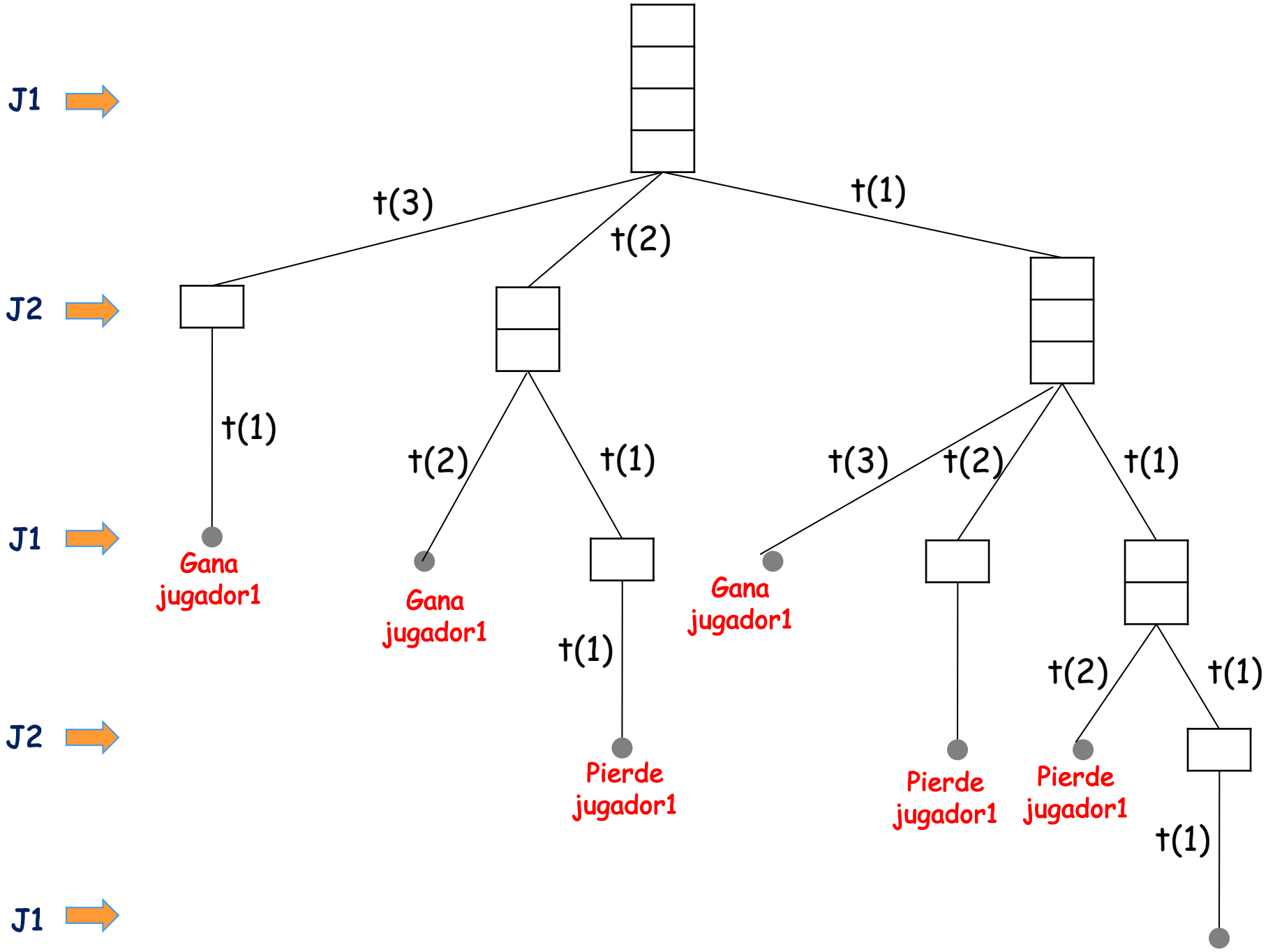

J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1
Gana jugador1
Pierde jugador1

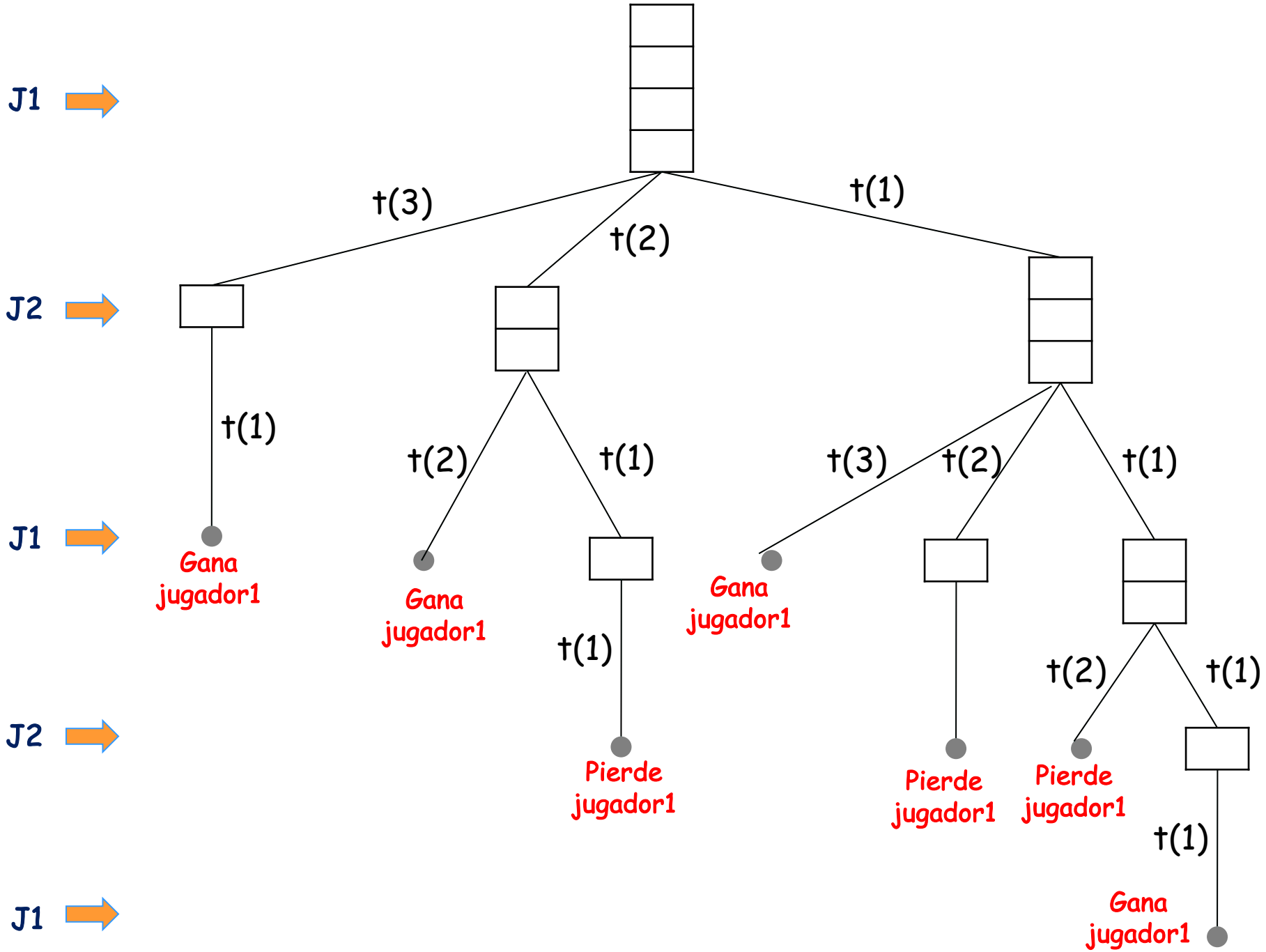

J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1
Gana jugador1
Pierde jugador1


J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1
Gana jugador1
Pierde jugador1
Pierde jugador1


J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
Gana jugador1
Gana jugador1
Pierde jugador1
Gana jugador1
Pierde jugador1


J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
Gana jugador1
Gana jugador1
Gana jugador1
Pierde jugador1
Pierde jugador1
Pierde jugador1


J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1
Pierde jugador1
Gana jugador1
Pierde jugador1
Pierde jugador1

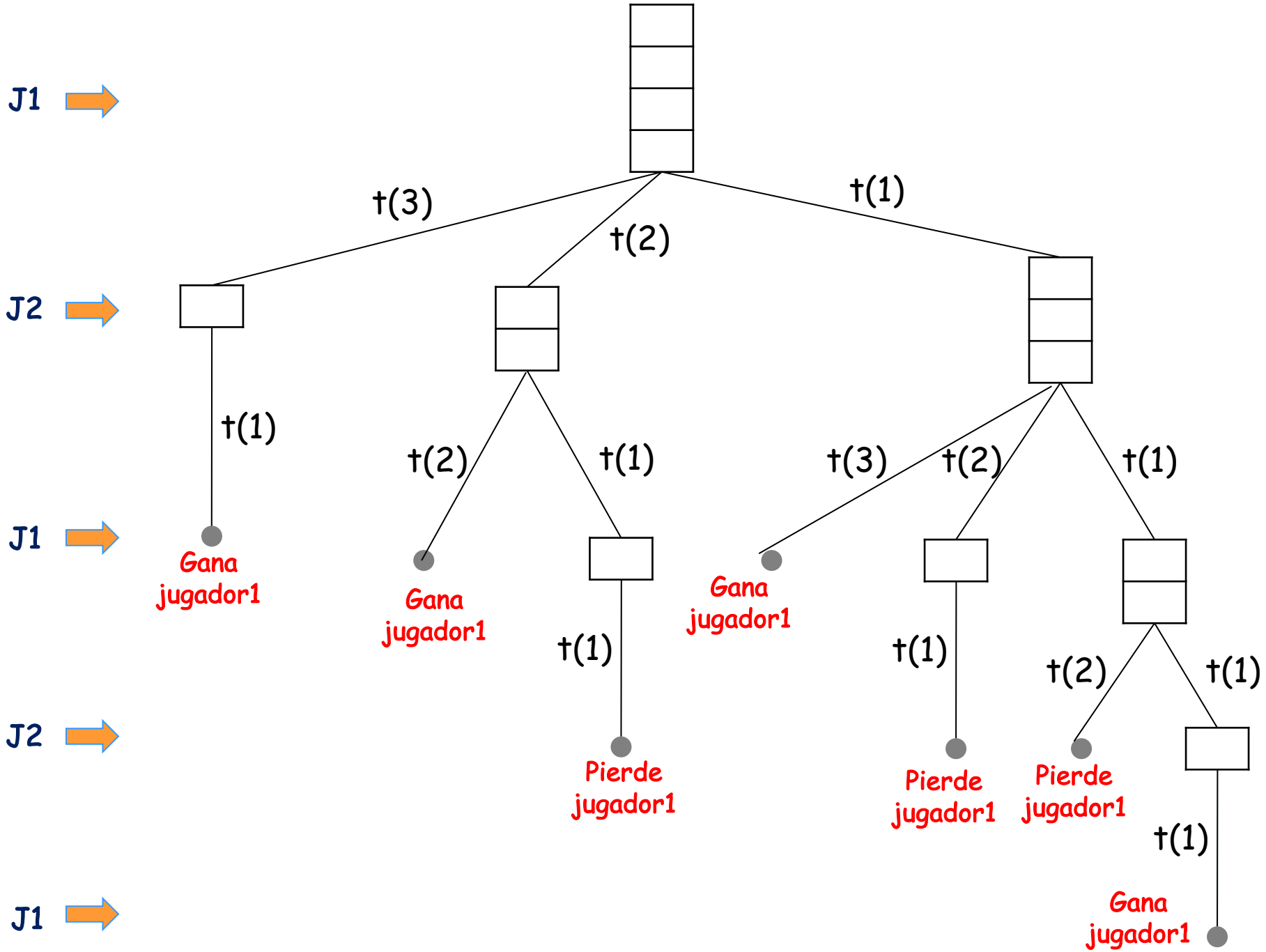

J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1
Pierde jugador1
Gana jugador1
Pierde jugador1
Pierde jugador1
Gana jugador1

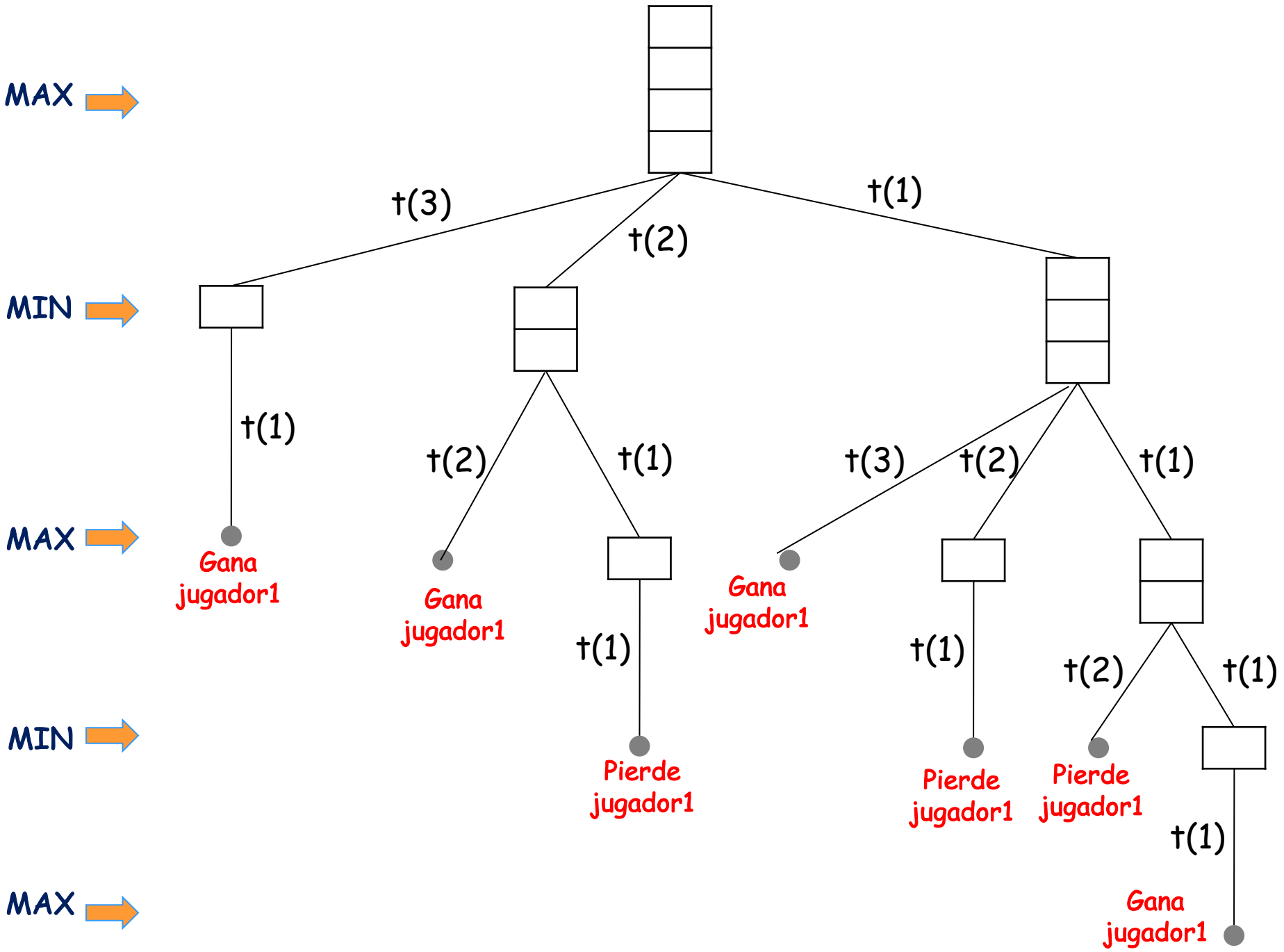
Juegos

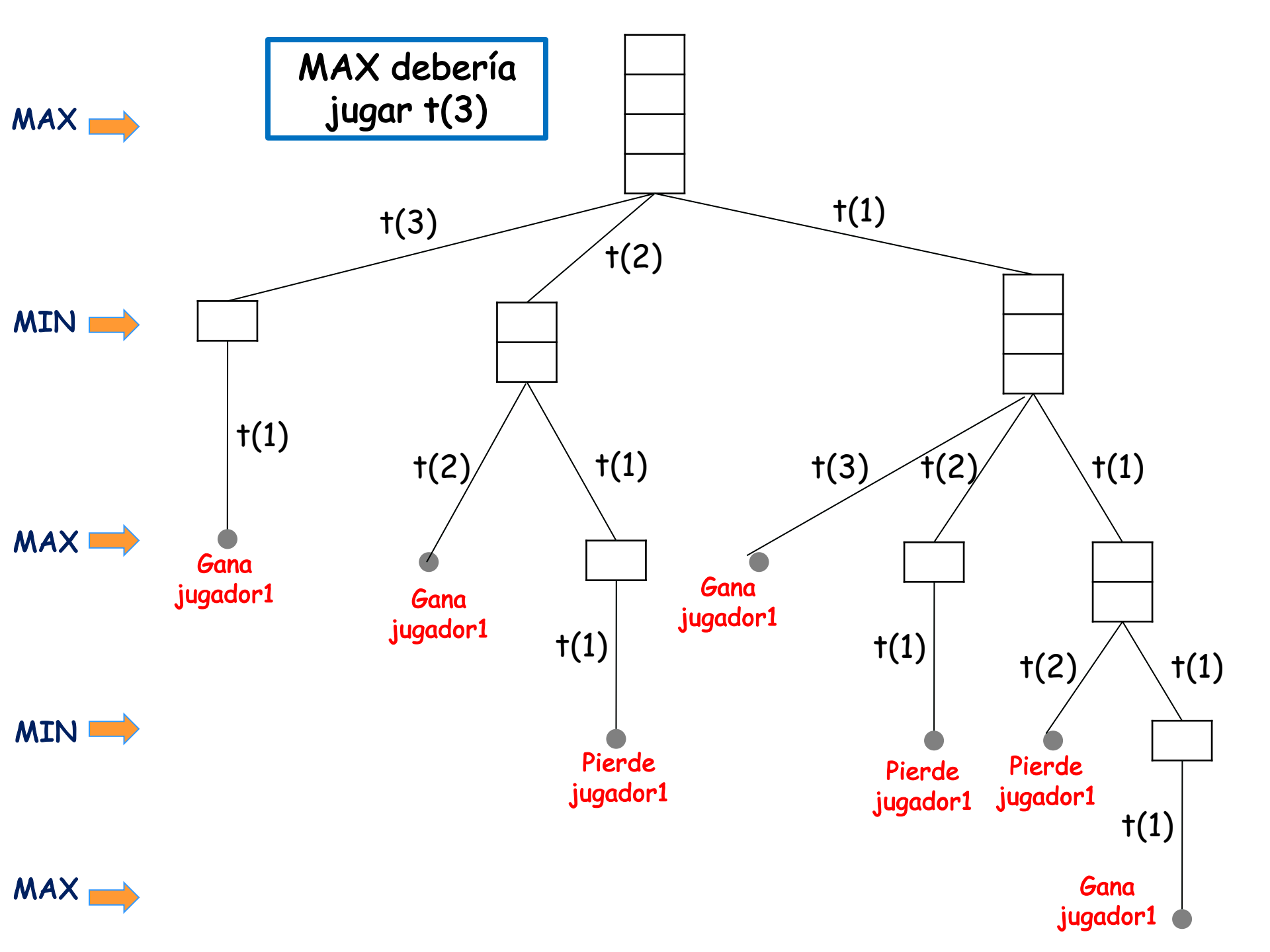
Algoritmo minimax

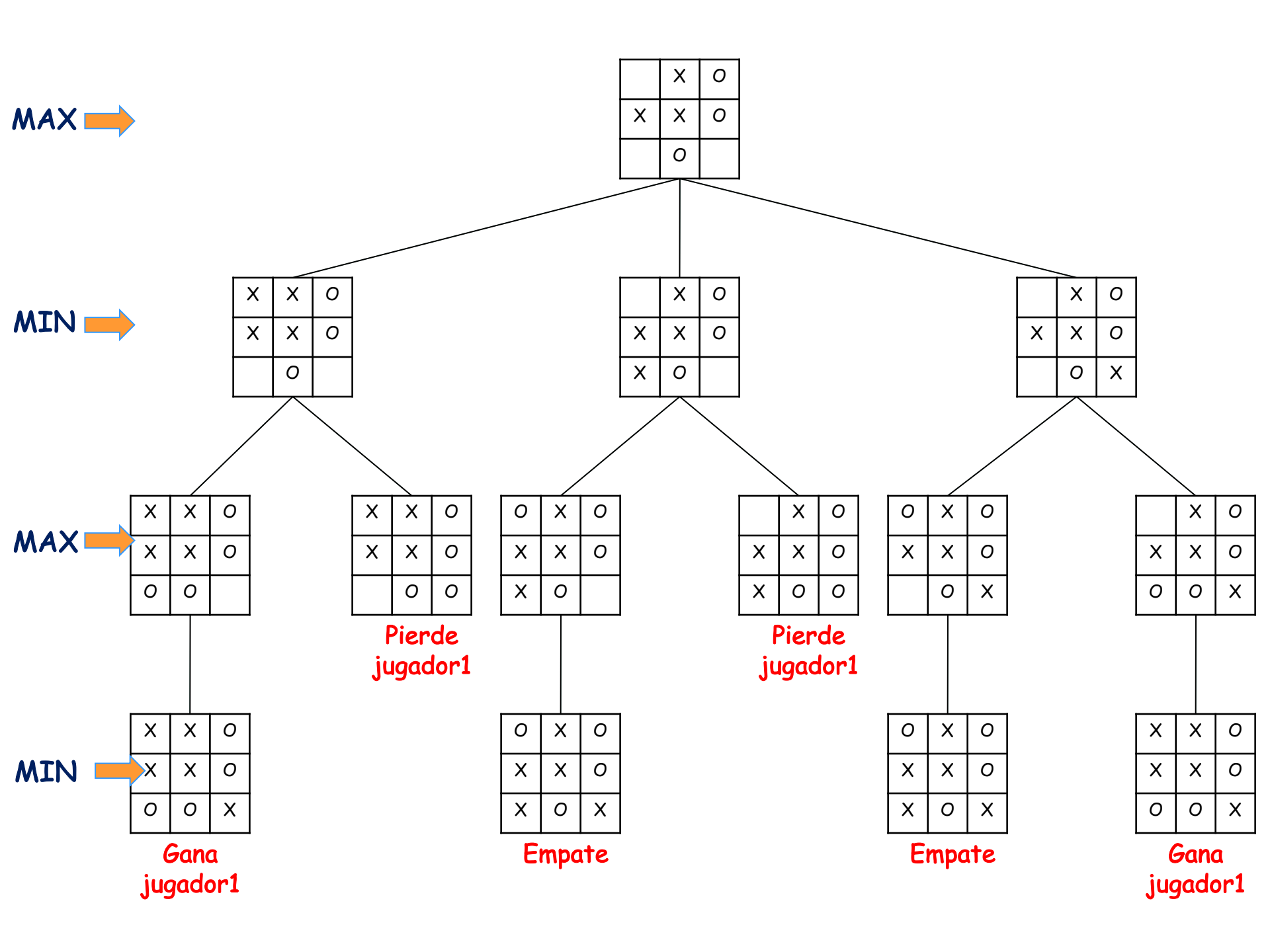
- El algoritmo minimax se aplica para el caso de juegos de dos participantes, MAX y MIN

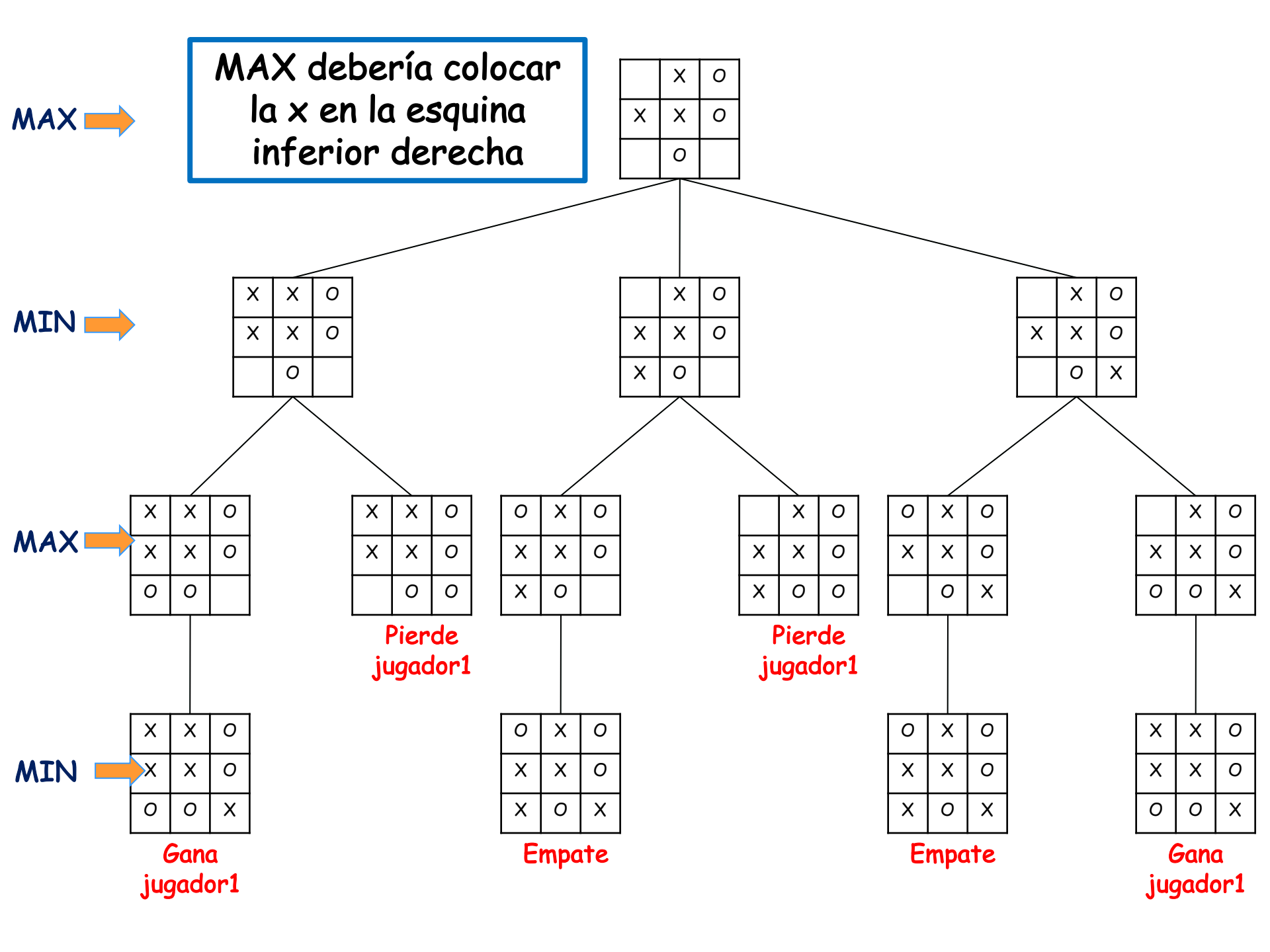
La salida del algoritmo es la jugada que debería realizar MAX en la raíz


J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1
Pierde jugador1
Gana jugador1
Pierde jugador1
Pierde jugador1
Gana jugador1


MAX
MIN
MAX
MIN
MAX
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1
Pierde jugador1
Gana jugador1
Pierde jugador1
Pierde jugador1
Gana jugador1


MAX debería jugar t(3)
MAX
MIN
MAX
MIN
MAX
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1
Pierde jugador1
Gana jugador1
Pierde jugador1
Pierde jugador1
Gana jugador1


MAX
MIN
MAX
MIN
Pierde jugador1
Pierde jugador1
Gana jugador1
Empate
Empate
Gana jugador1


MAX debería colocar la x en la esquina inferior derecha
MAX
MIN
MAX
MIN
Pierde jugador1
Pierde jugador1
Gana jugador1
Empate
Empate
Gana jugador1

Juegos

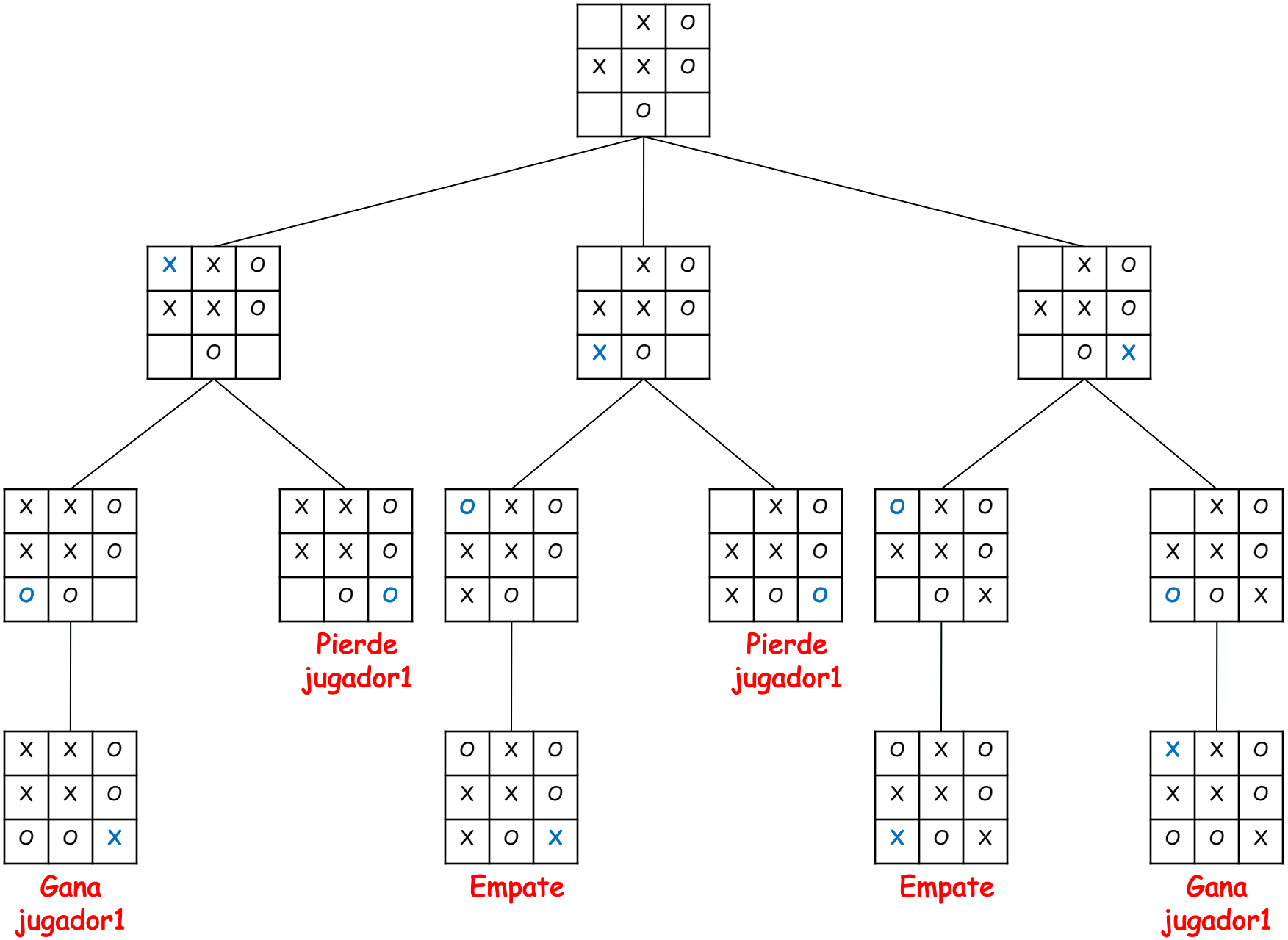
Algoritmo minimax

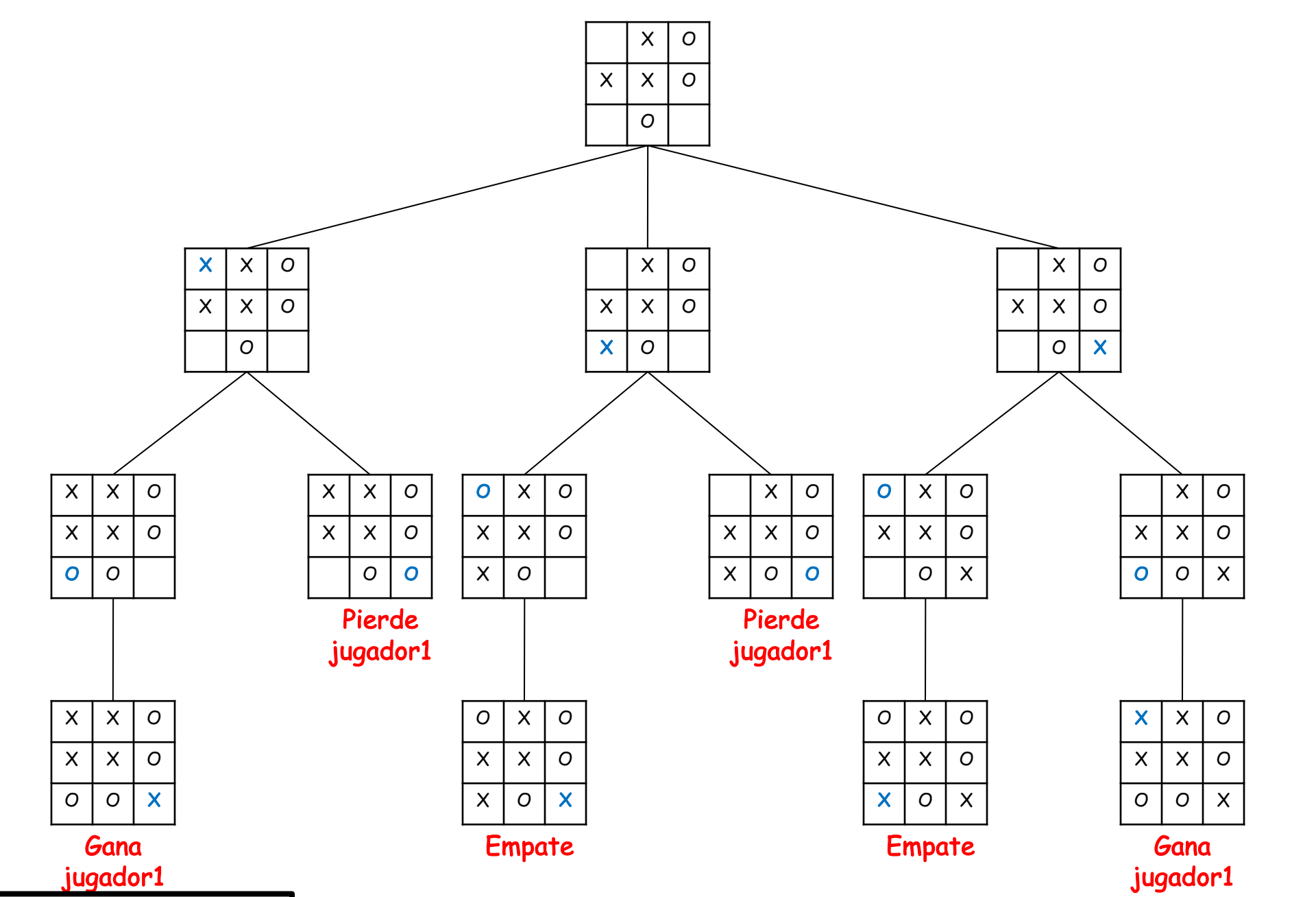
- La definición formal de juego requiere:
  - **Estado inicial**
  - **Conjunto de operadores**
  - **Prueba terminal**
  - **Función de utilidad**

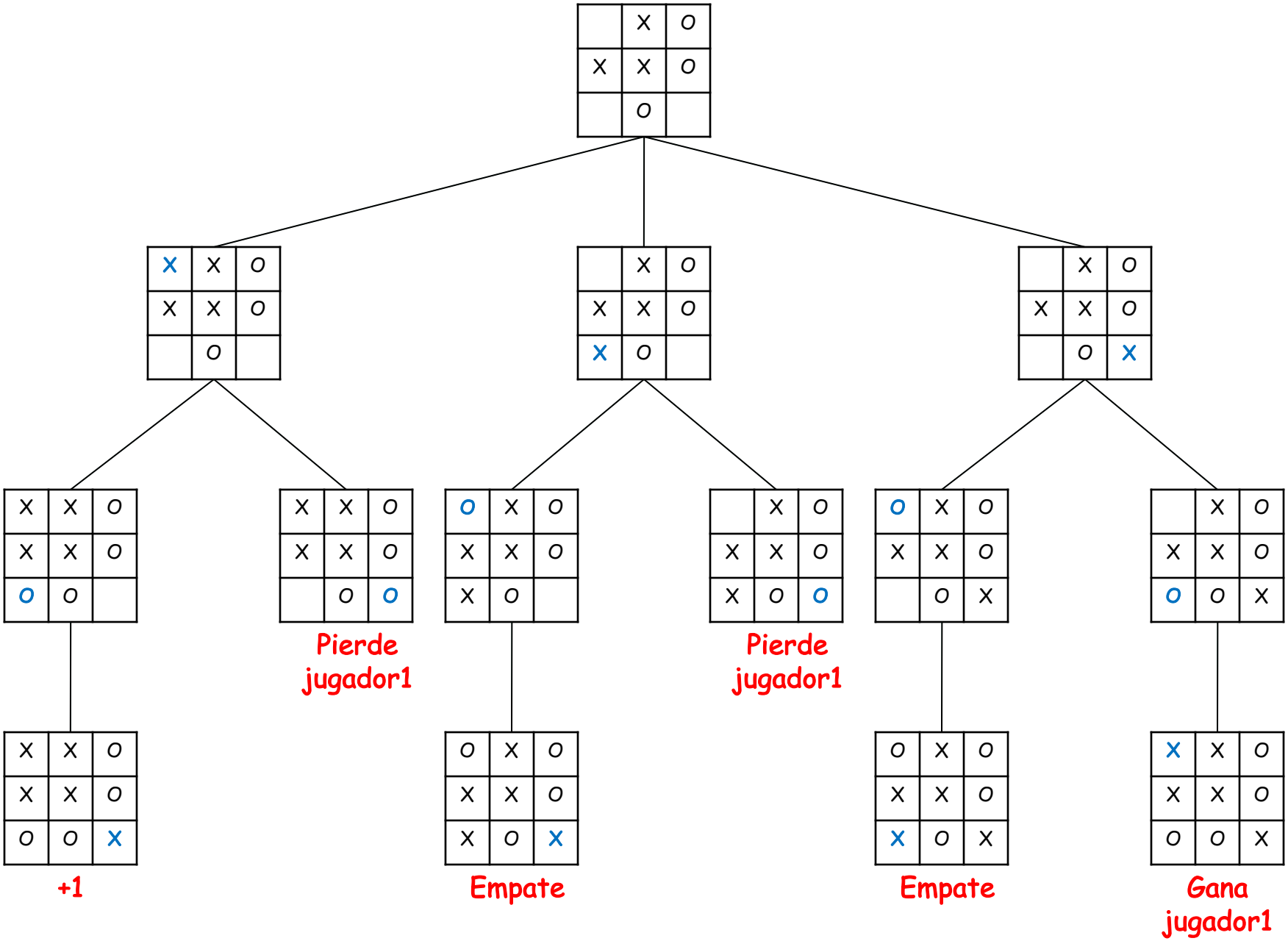
Juegos

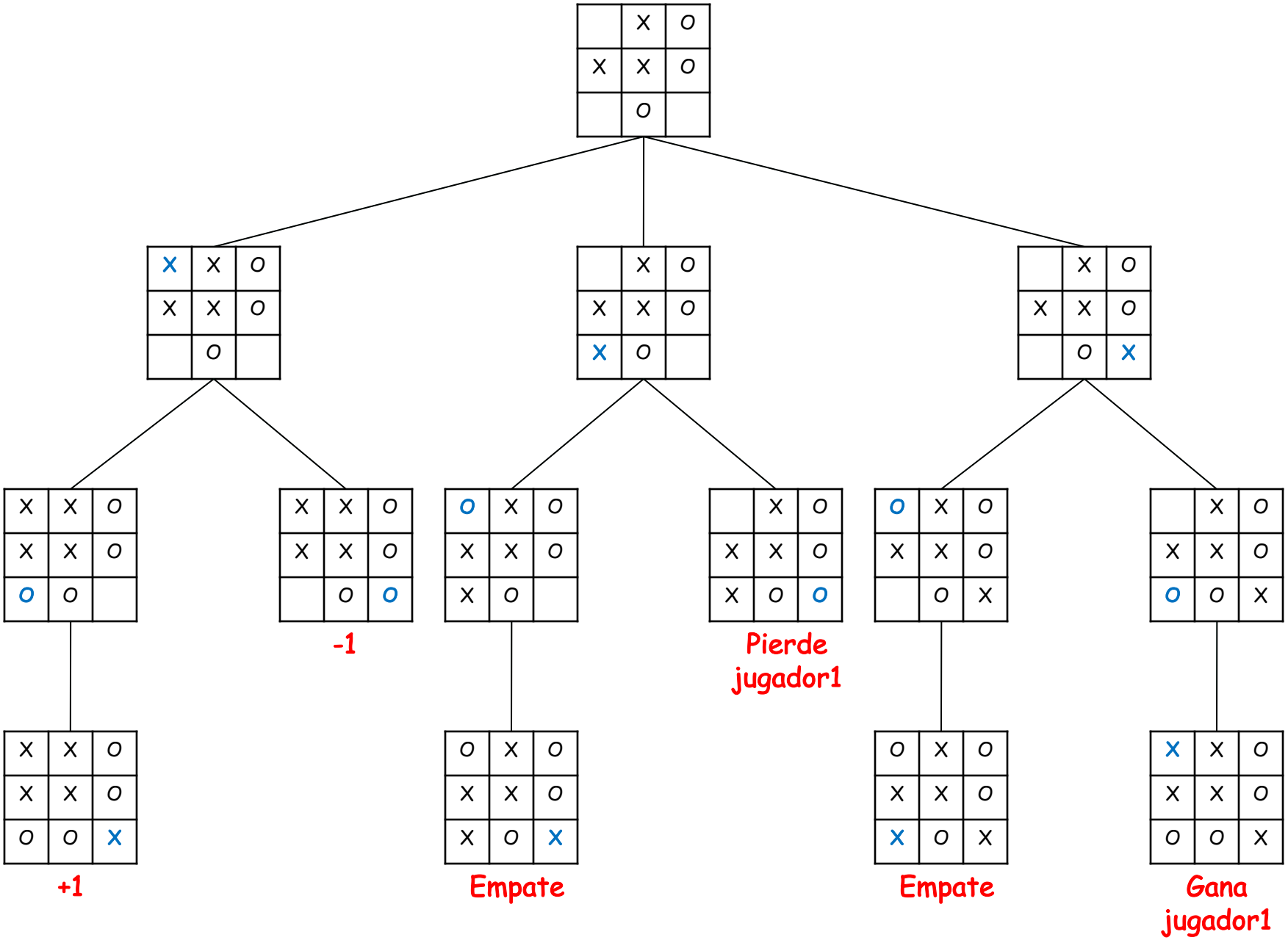
Algoritmo minimax

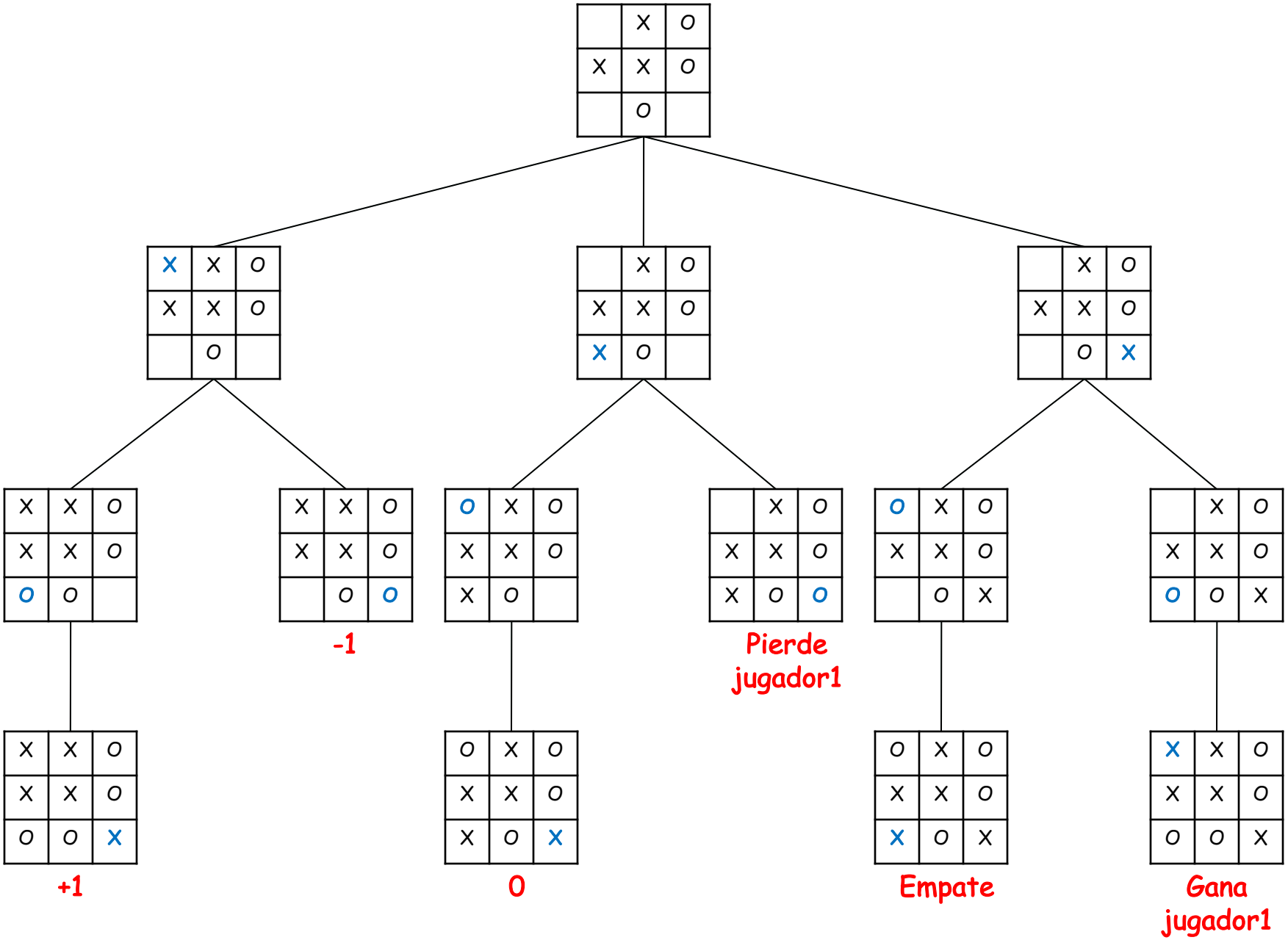
- La definición formal de juego requiere:
  - **Estado inicial**
  - **Conjunto de operadores**
  - **Prueba terminal**
  - **Función de utilidad**, asigna un valor numérico al resultado obtenido de MAX en el juego

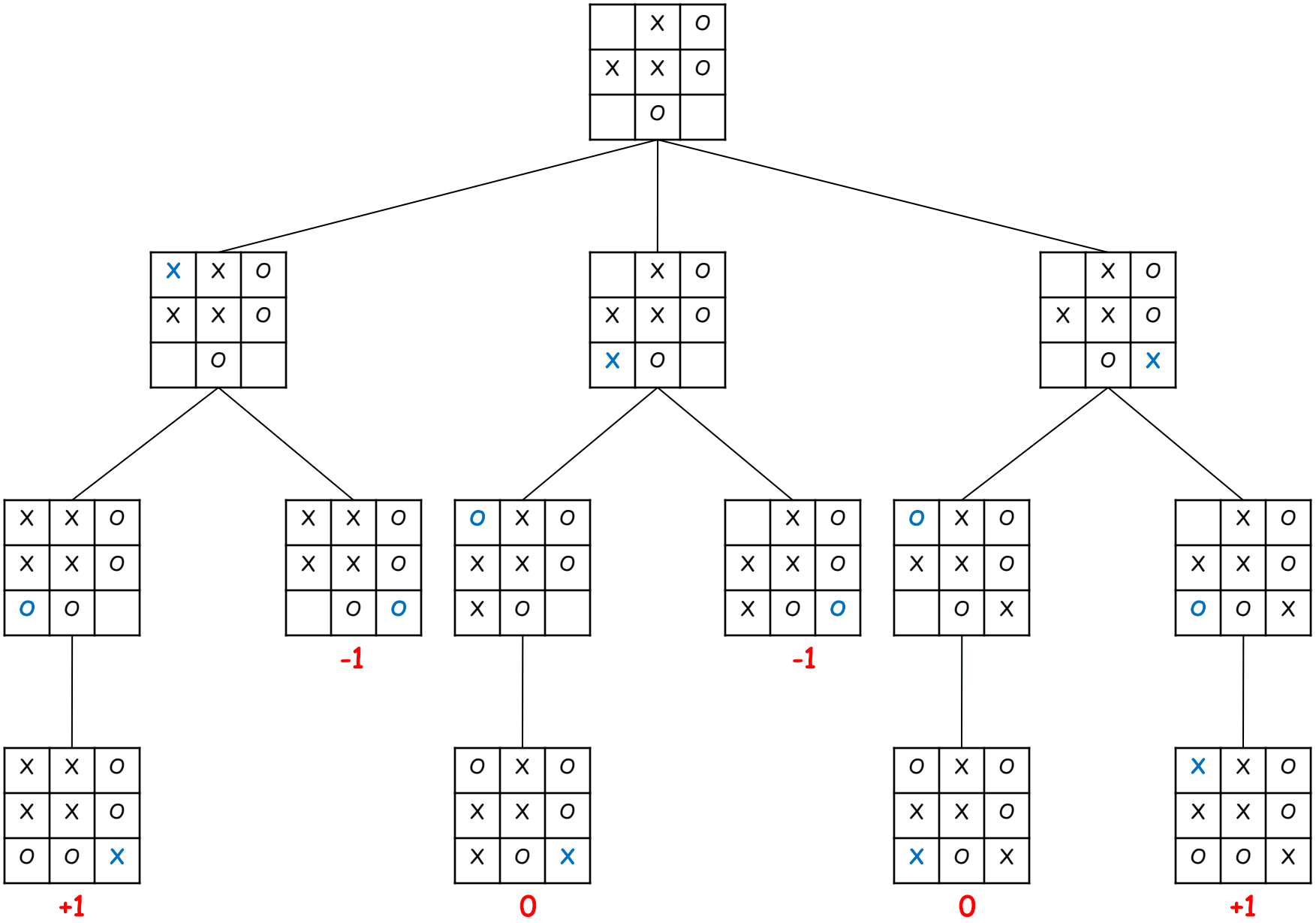

Pierde jugador1
Pierde jugador1
Gana jugador1
Empate
Empate
Gana jugador1


Pierde jugador1
Pierde jugador1
Gana jugador1
Empate
Empate
Gana jugador1
¿Cuál es la utilidad para MAX en este camino?

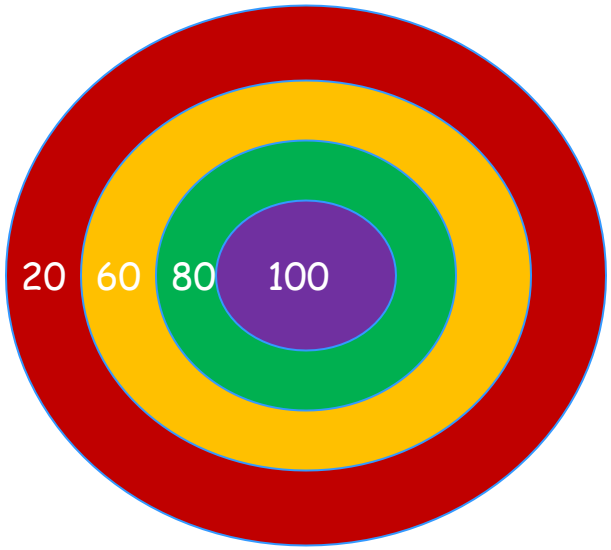

Pierde jugador1
Pierde jugador1
+1
Empate
Empate
Gana jugador1


-1
Pierde jugador1
+1
Empate
Empate
Gana jugador1


-1
Pierde jugador1
+1
0
Empate
Gana jugador1


-1
-1
+1
0
0
+1

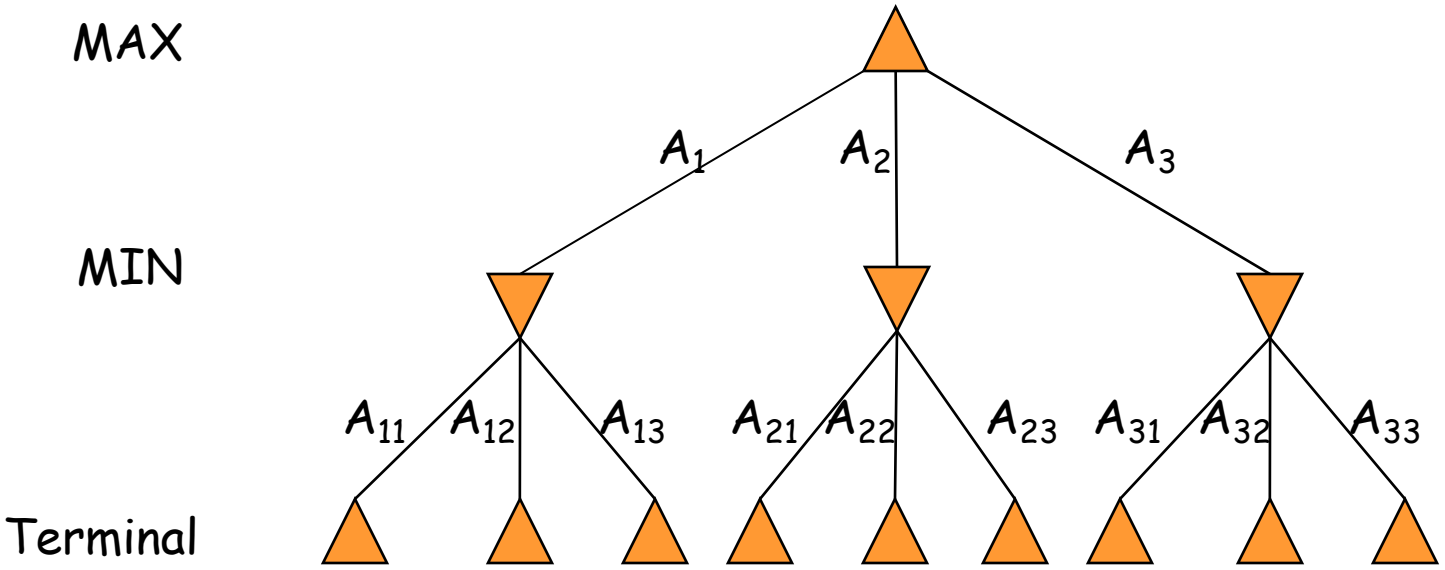
Juegos



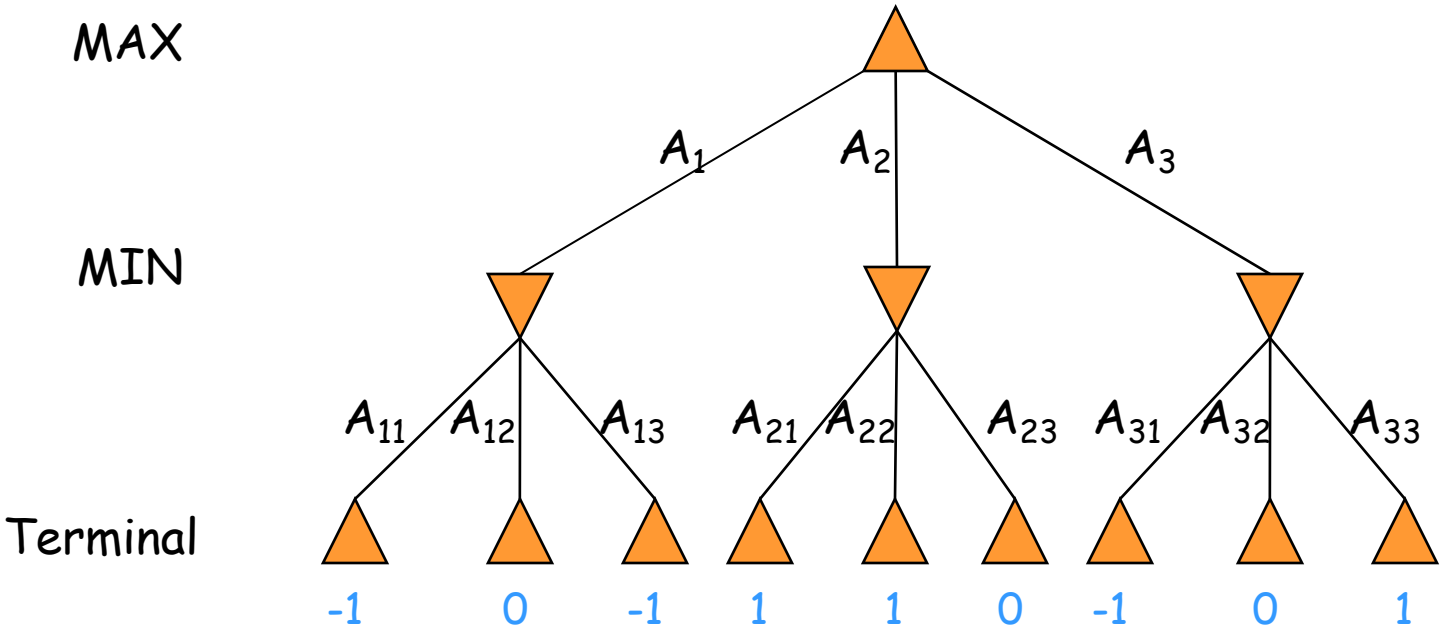
- Cada jugador tiene 3 lanzamientos
- Clasifica quien obtenga 200 puntos acumulados

En juegos donde se gana o pierde, f=1 si gana, f=-1 si pierde

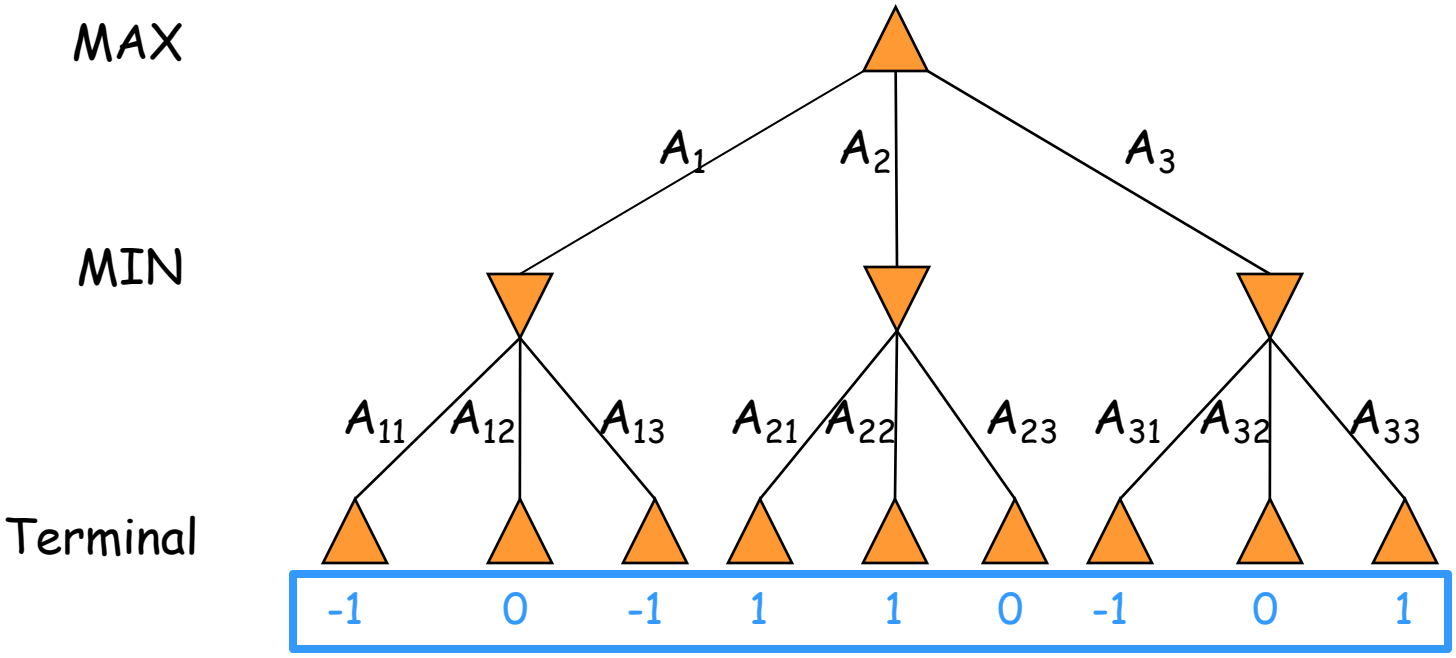
Juegos



Juegos

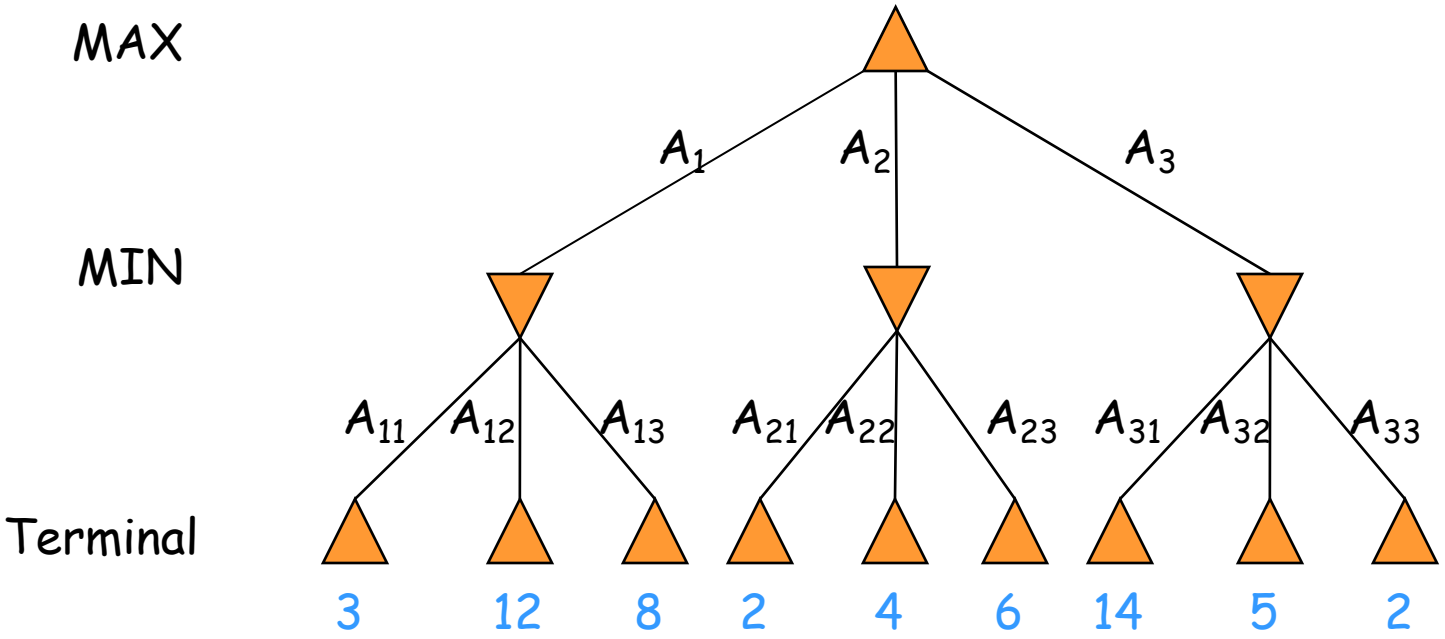

MAX
MIN
Terminal
A1
A2
A3
A11
A12
A13
A21
A22
A23
A31
A32
A33
-1
0
-1
1
1
0
-1
0
1

Juegos

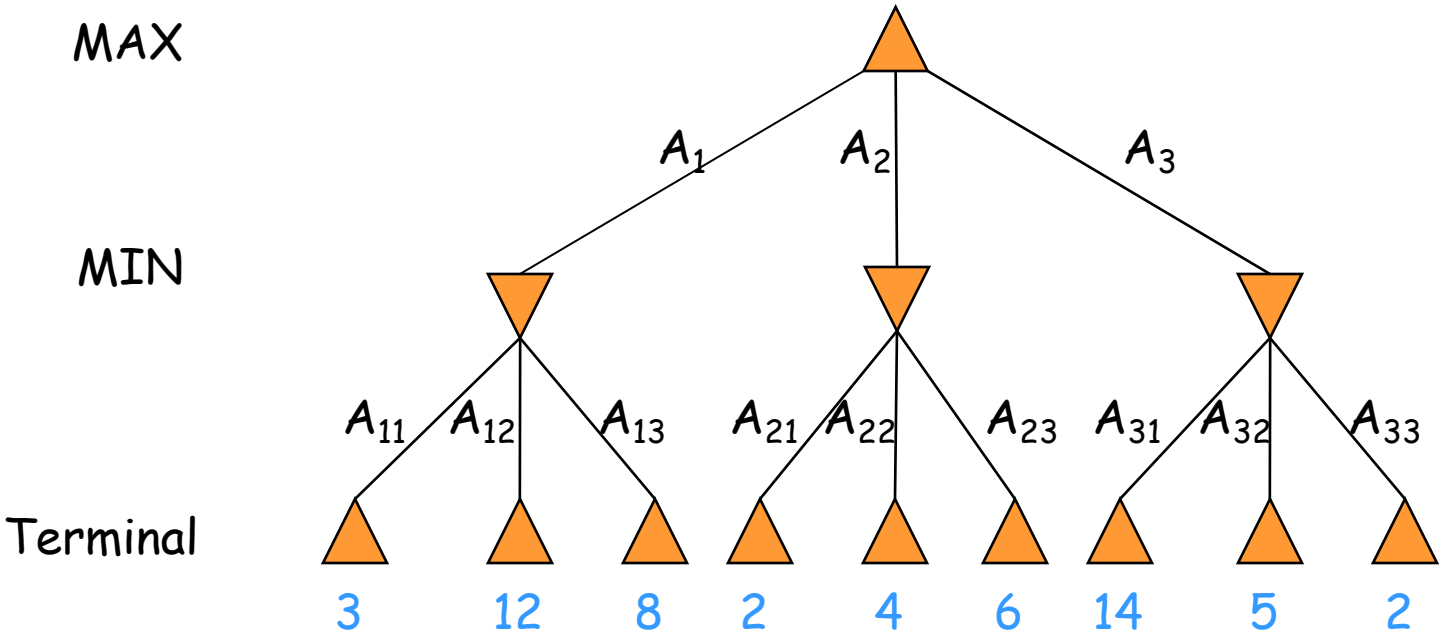


Las utilidades se calculan en las hojas (cuando el juego ha terminado)

Juegos

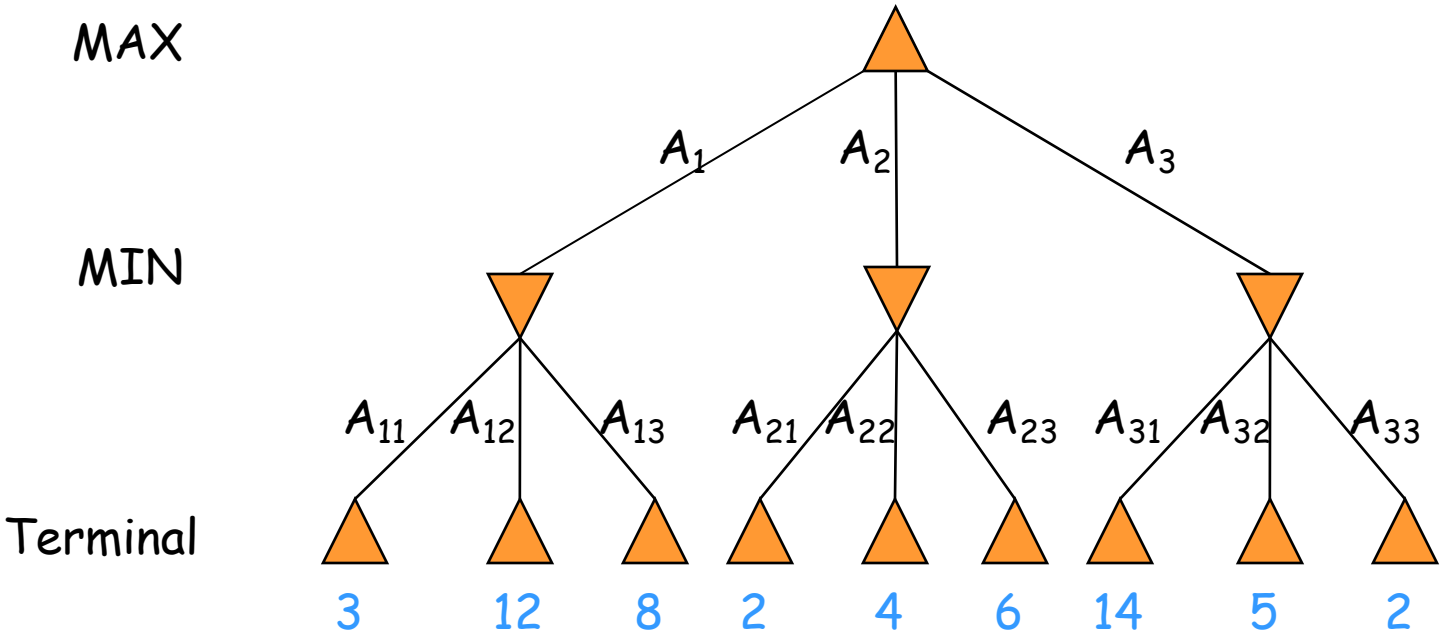

MAX
MIN
Terminal
A1
A2
A3
A11
A12
A13
A21
A22
A23
A31
A32
A33
3
12
8
2
4
6
14
5
2

Juegos

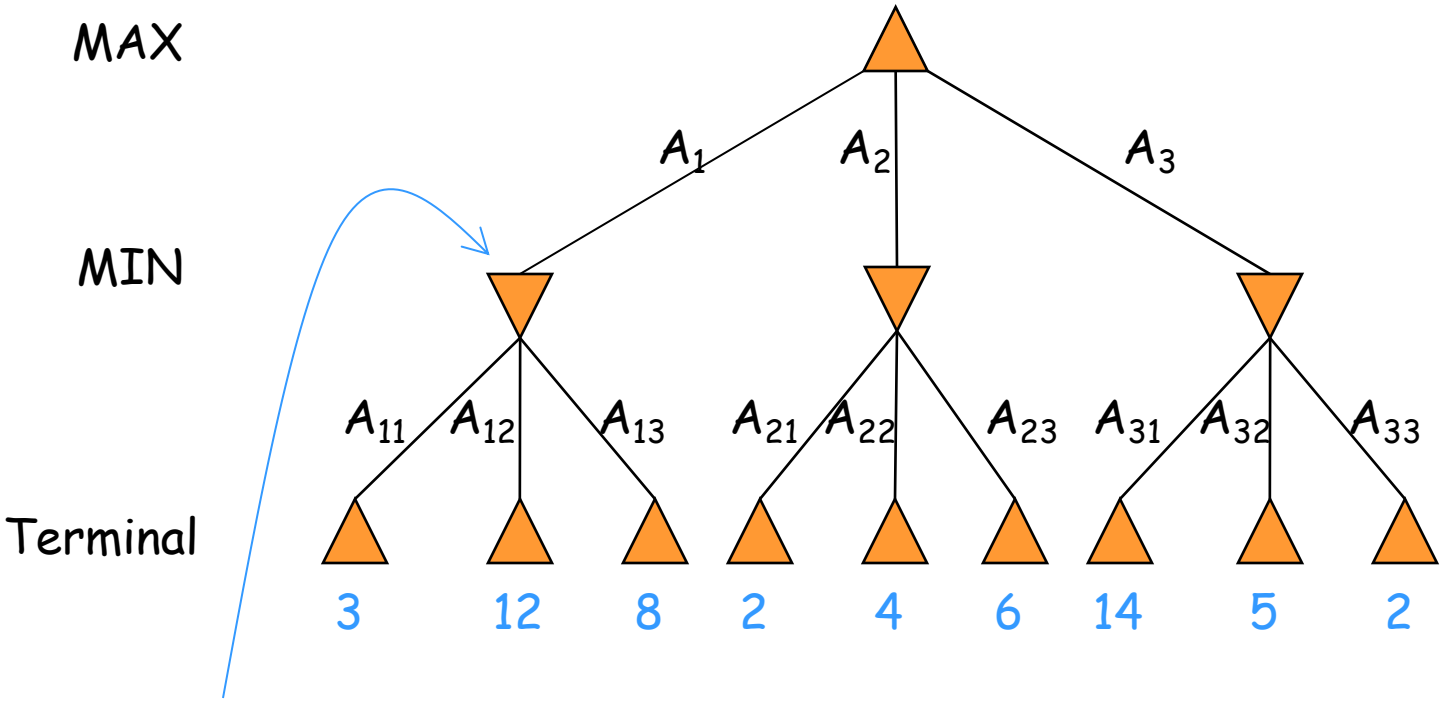


¿Cuál es la decisión que debería tomar MAX?

Juegos

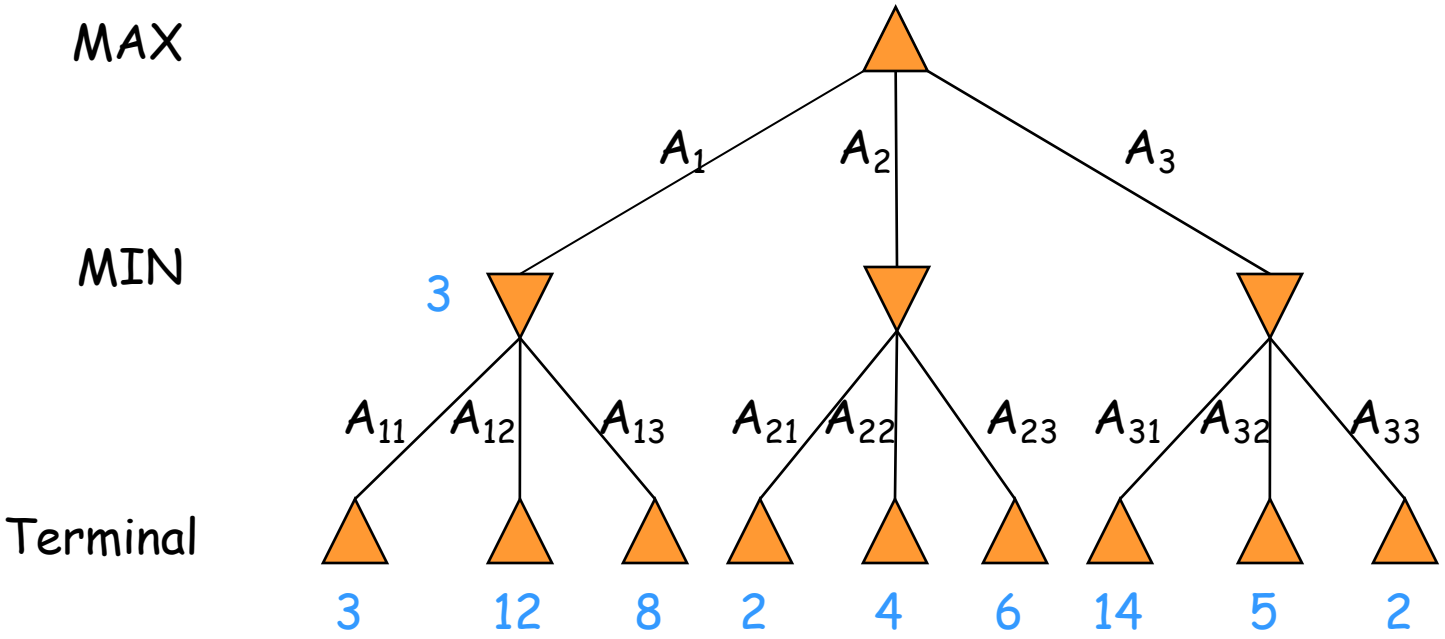

MAX
MIN
Terminal
A1
A2
A3
A11
A12
A13
A21
A22
A23
A31
A32
A33
3
12
8
2
4
6
14
5
2

Juegos

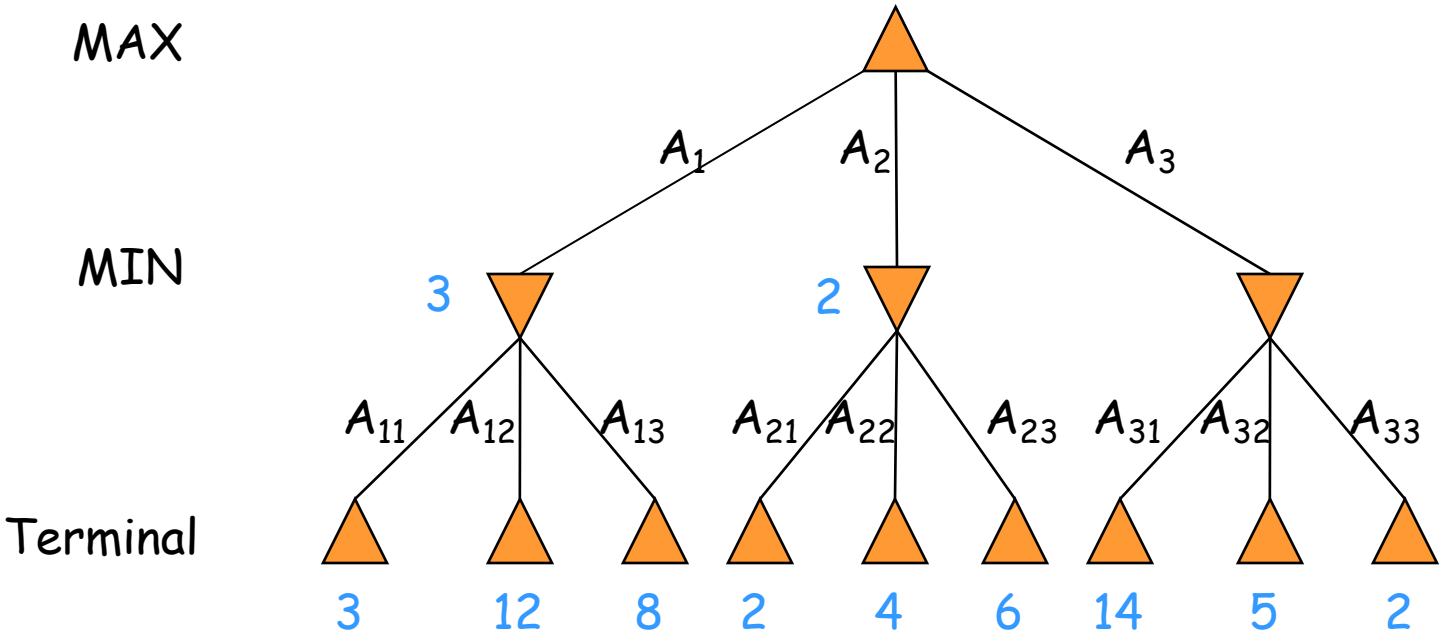


Si MIN sabe de IA, qué acción emprenderá en este nodo?

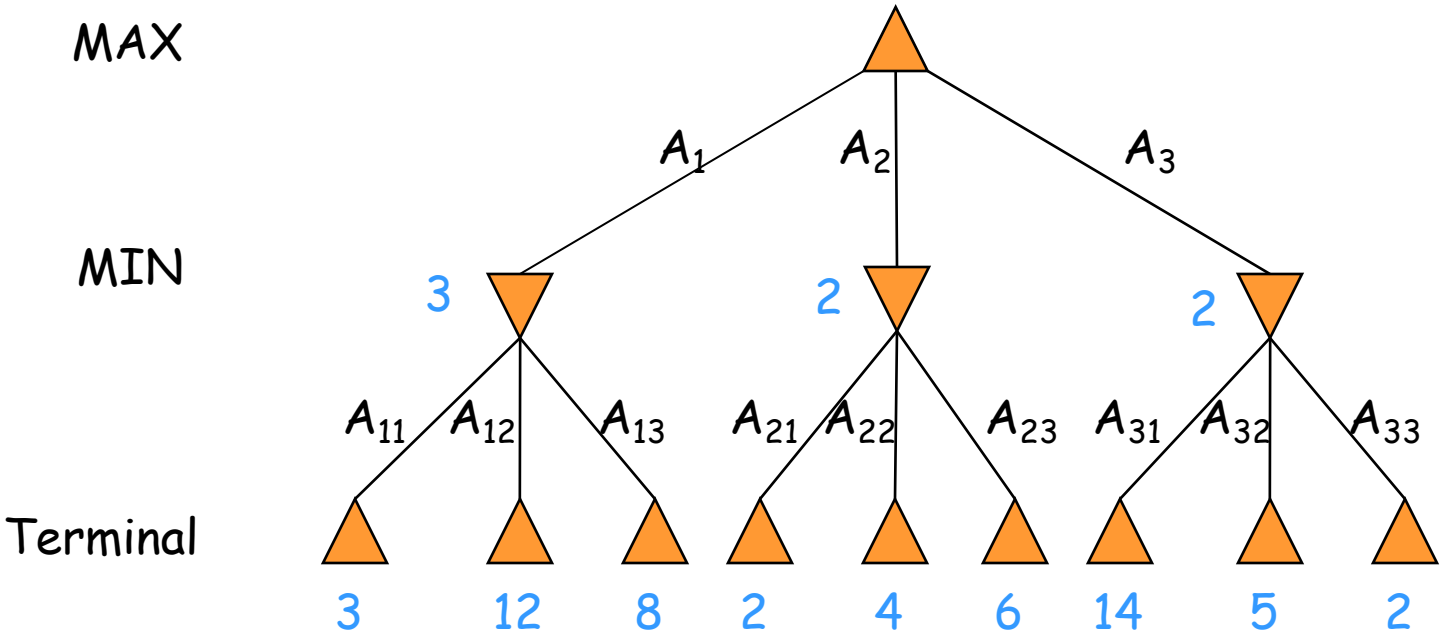
Juegos


MAX
MIN
Terminal
A1
A2
A3
3
A11
A12
A13
A21
A22
A23
A31
A32
A33
3
12
8
2
4
6
14
5
2

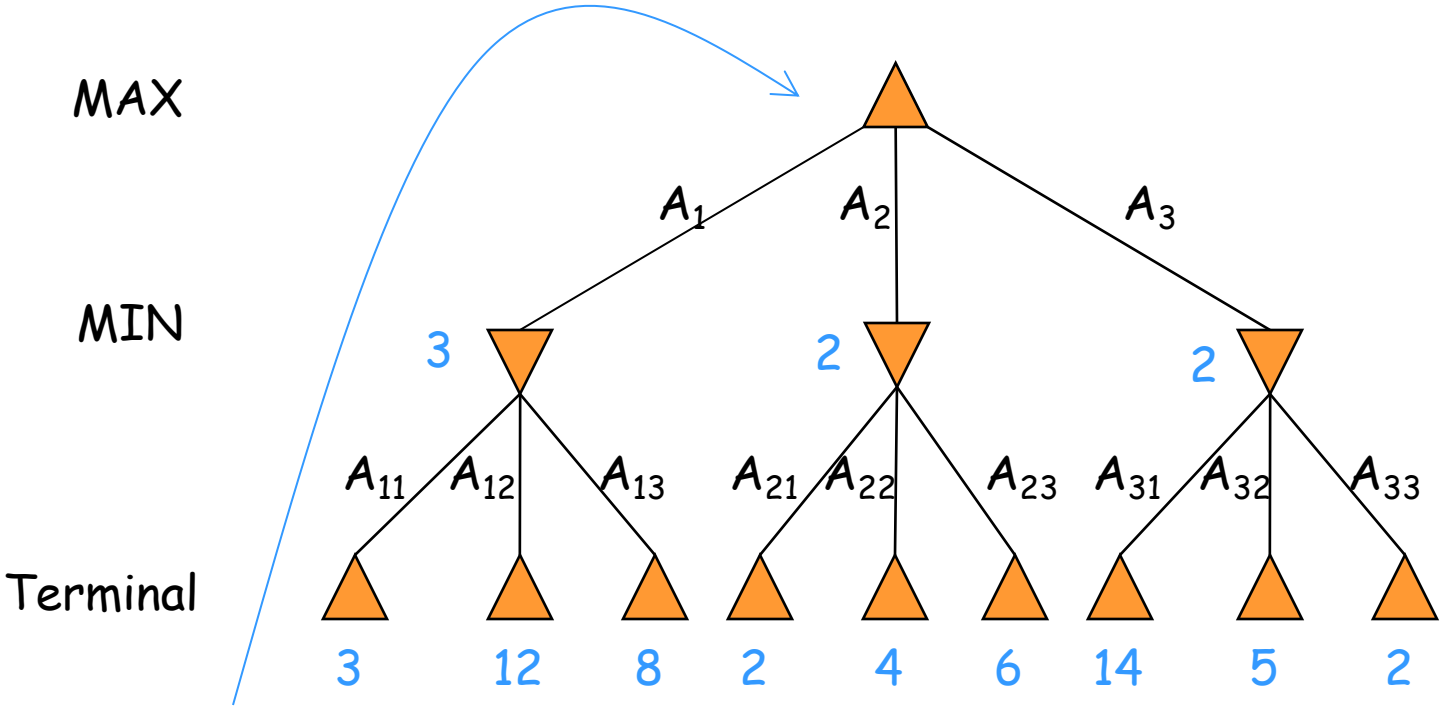
Juegos


MAX
MIN
Terminal
A1
A2
A3
3
2
A11
A12
A13
A21
A22
A23
A31
A32
A33
3
12
8
2
4
6
14
5
2

Juegos

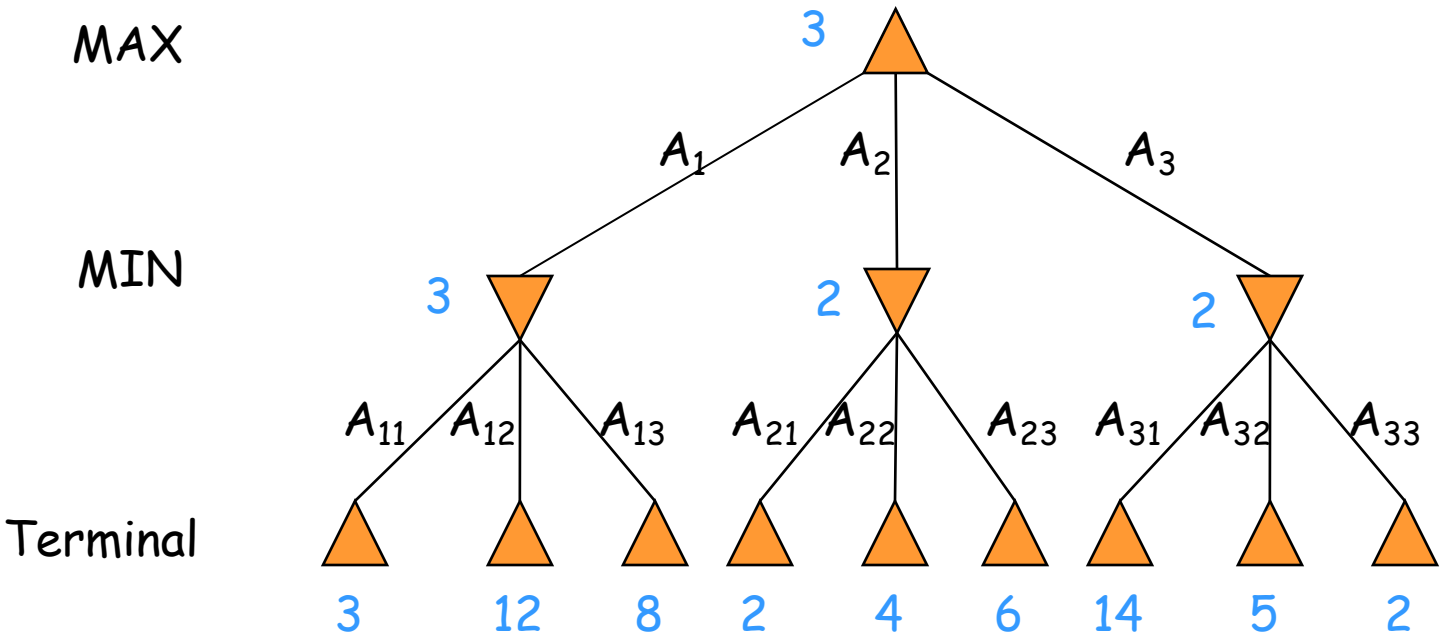

MAX
MIN
Terminal
A1
A2
A3
3
2
2
A11
A12
A13
A21
A22
A23
A31
A32
A33
3
12
8
2
4
6
14
5
2

Juegos



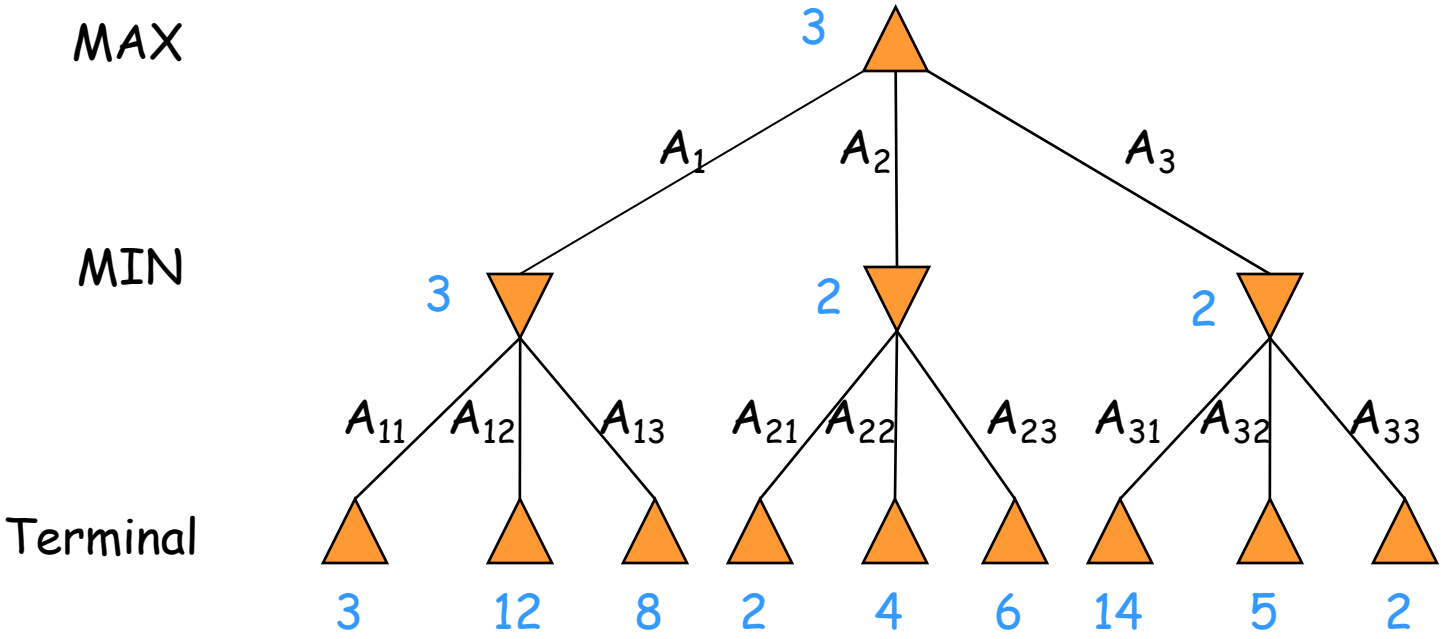
Si MAX sabe de IA, qué acción emprenderá en este nodo?

Juegos



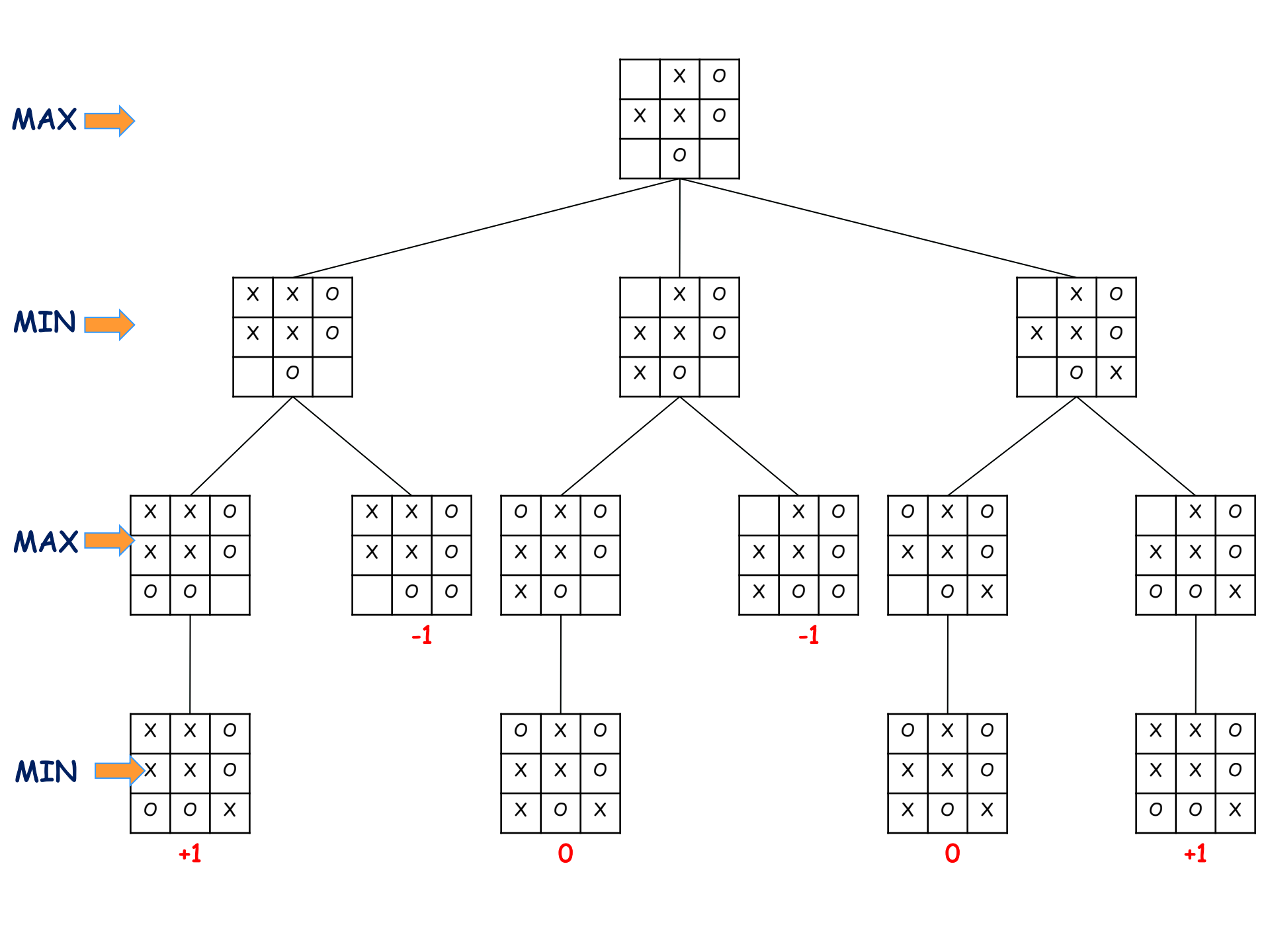
MAX intentará obtener el mayor valor. En este caso será 3 para el nodo raíz

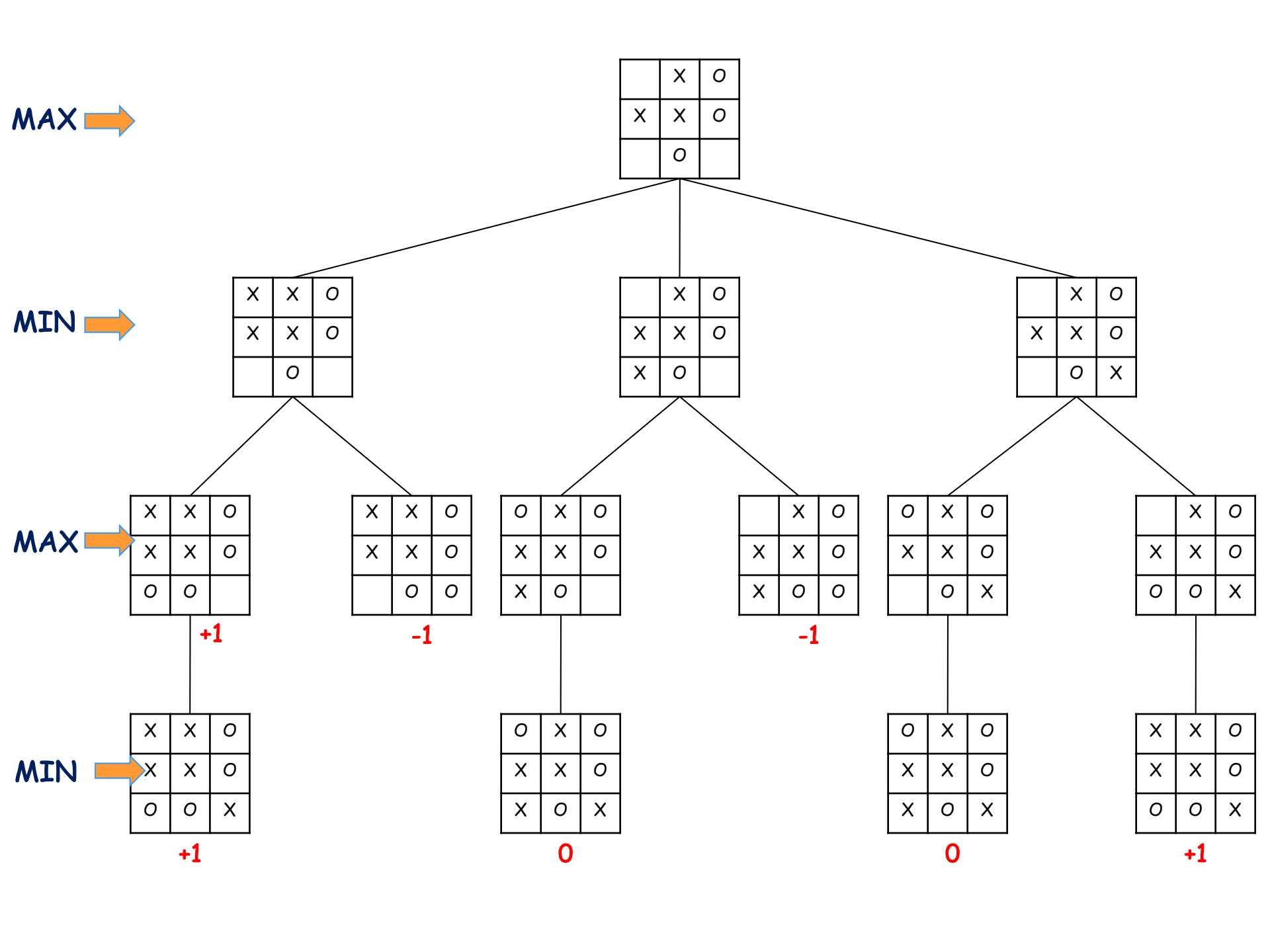
Juegos

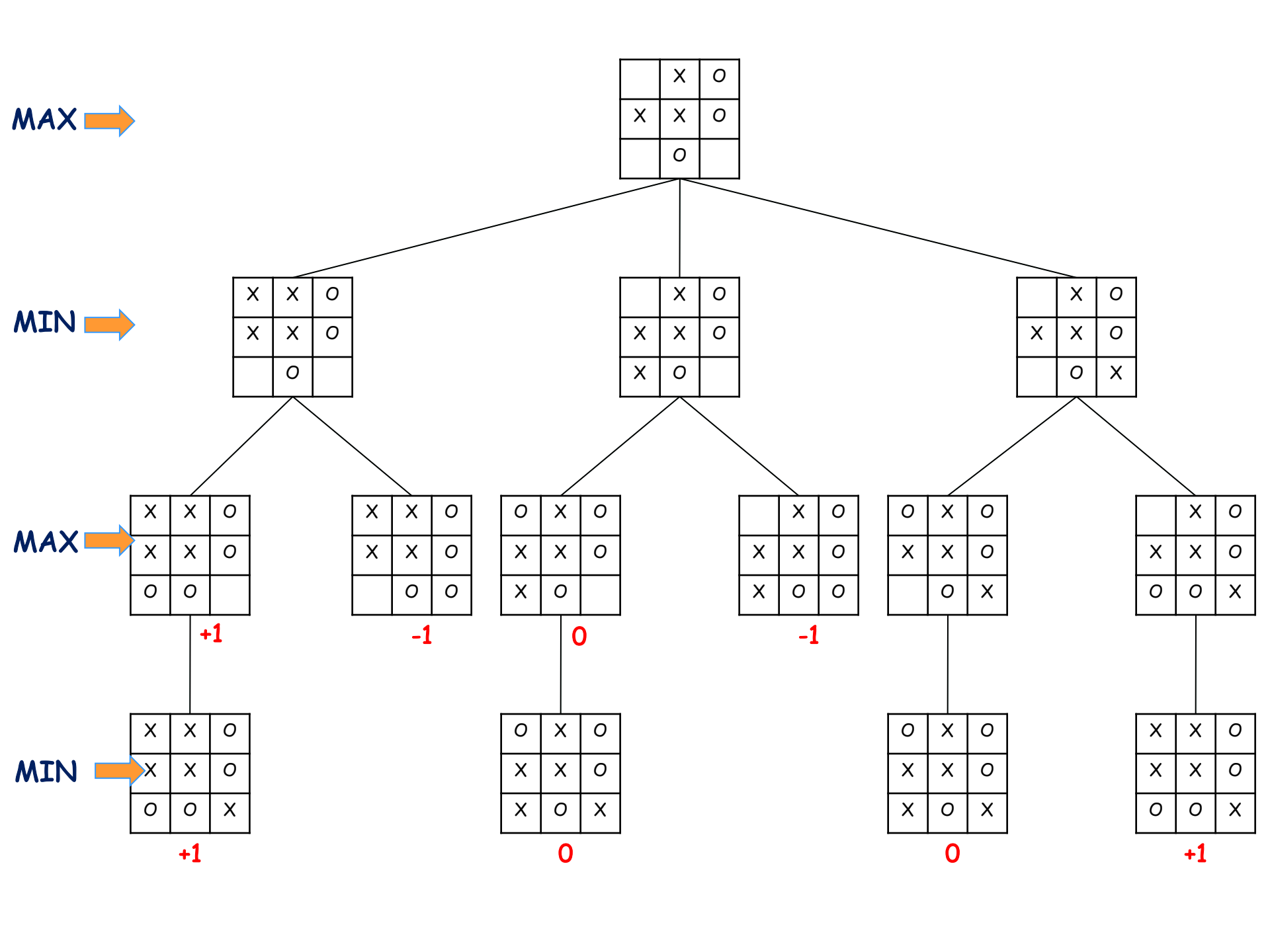


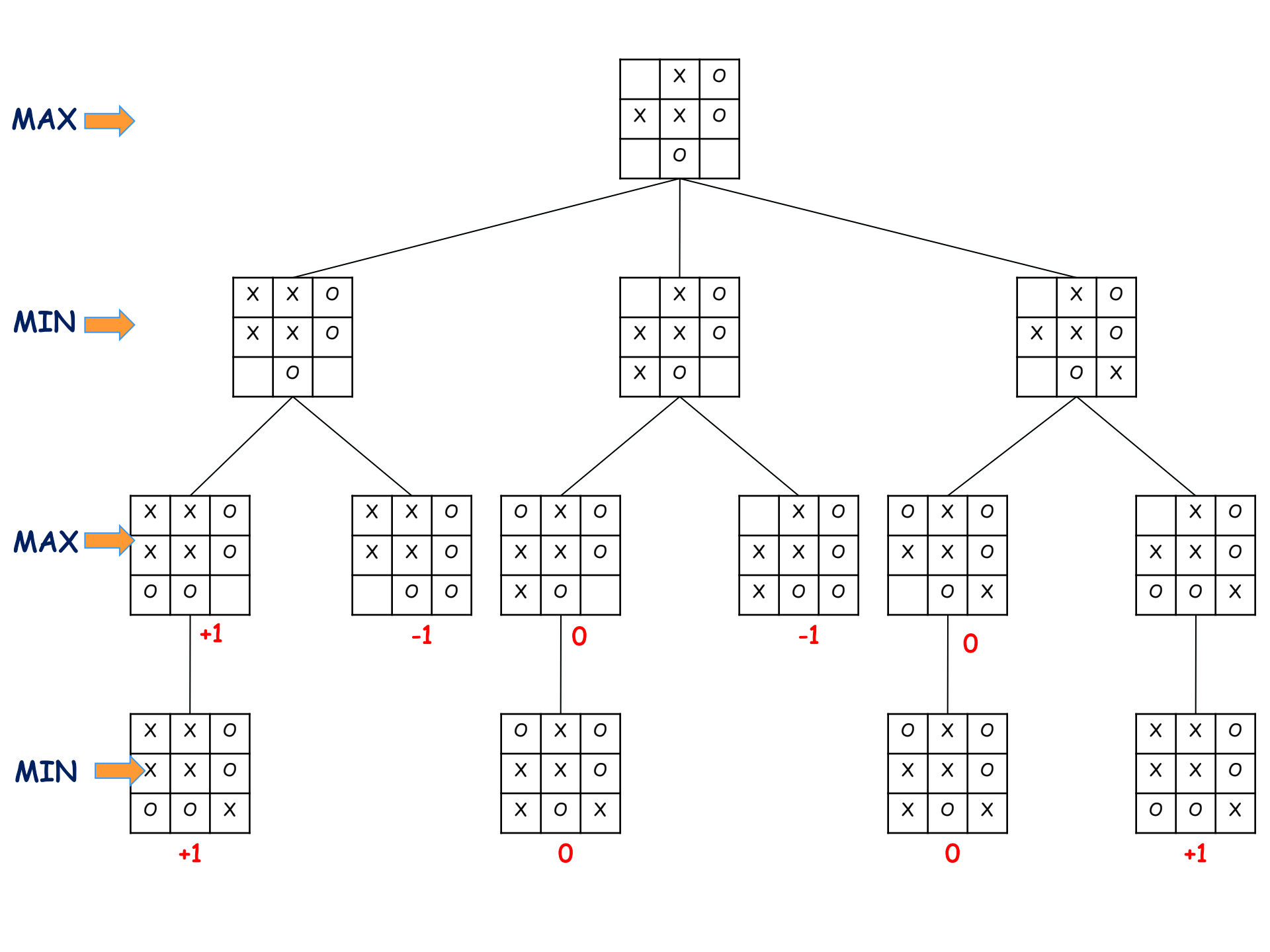
La mejor opción para MAX es A_1

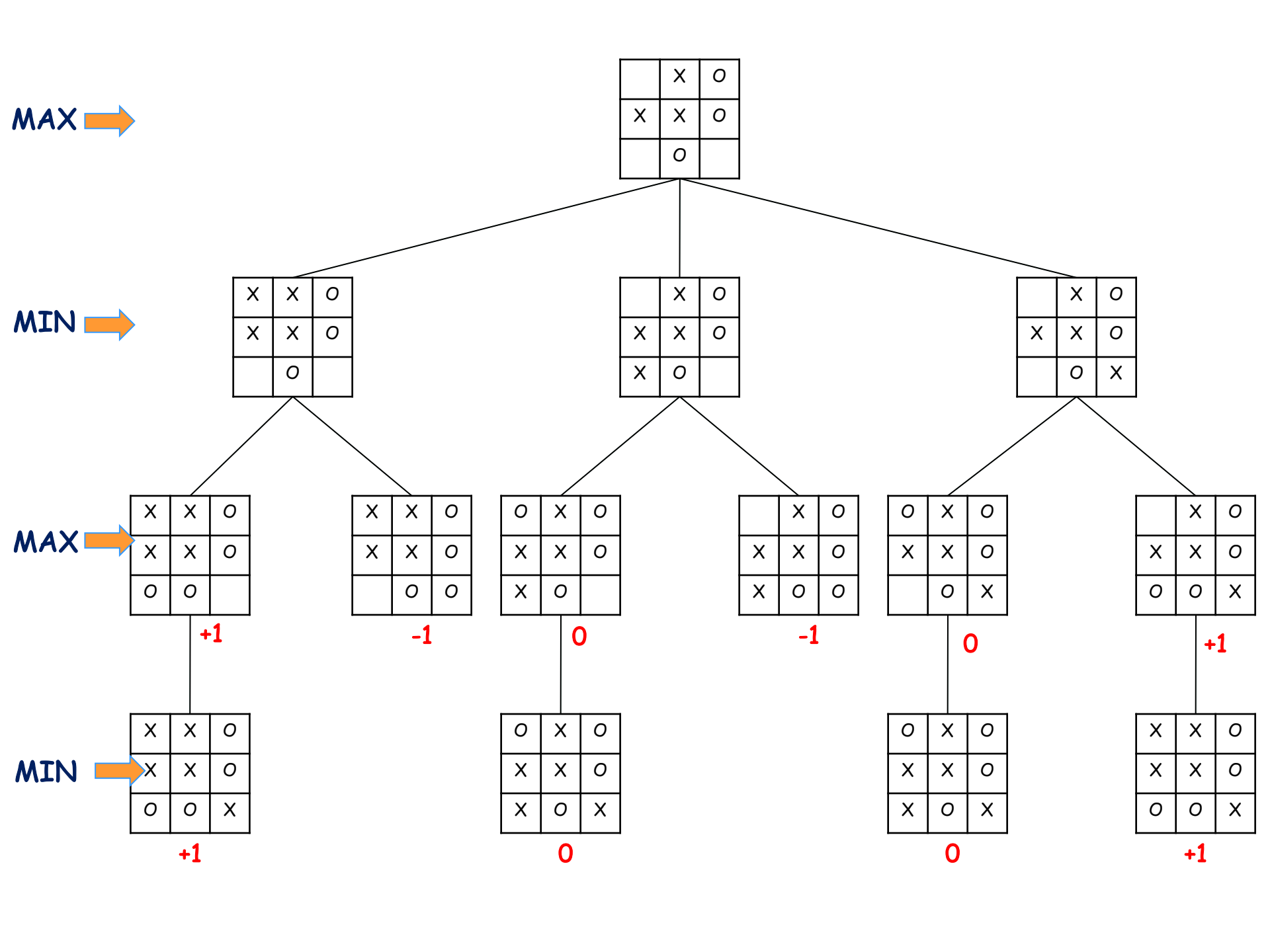
Esta se conoce como la **decisión minimax**. Se supone que MIN siempre juega con la intensión de disminuir al máximo la utilidad de MAX

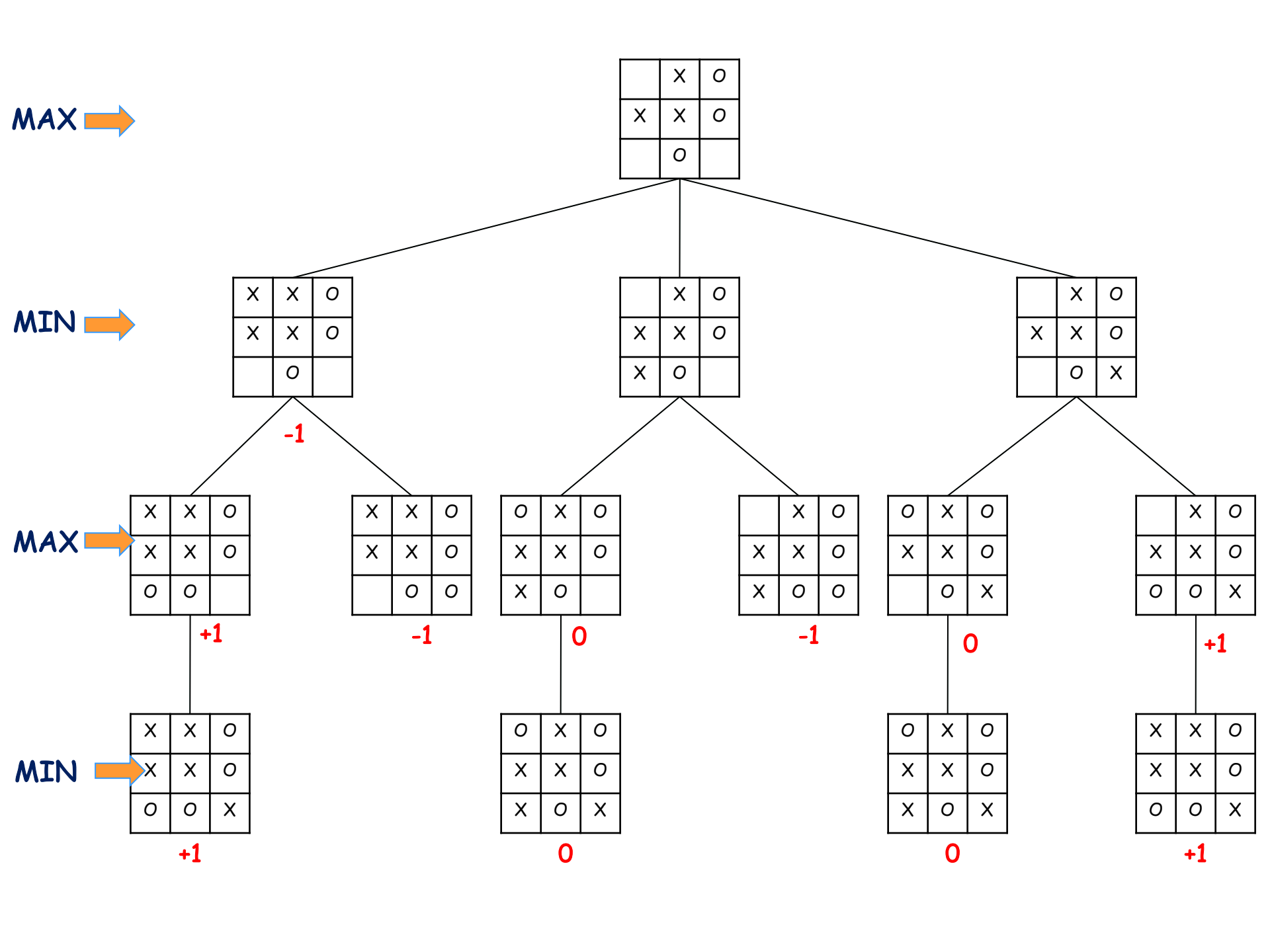

MAX
MIN
MAX
MIN
-1
-1
+1
0
0
+1

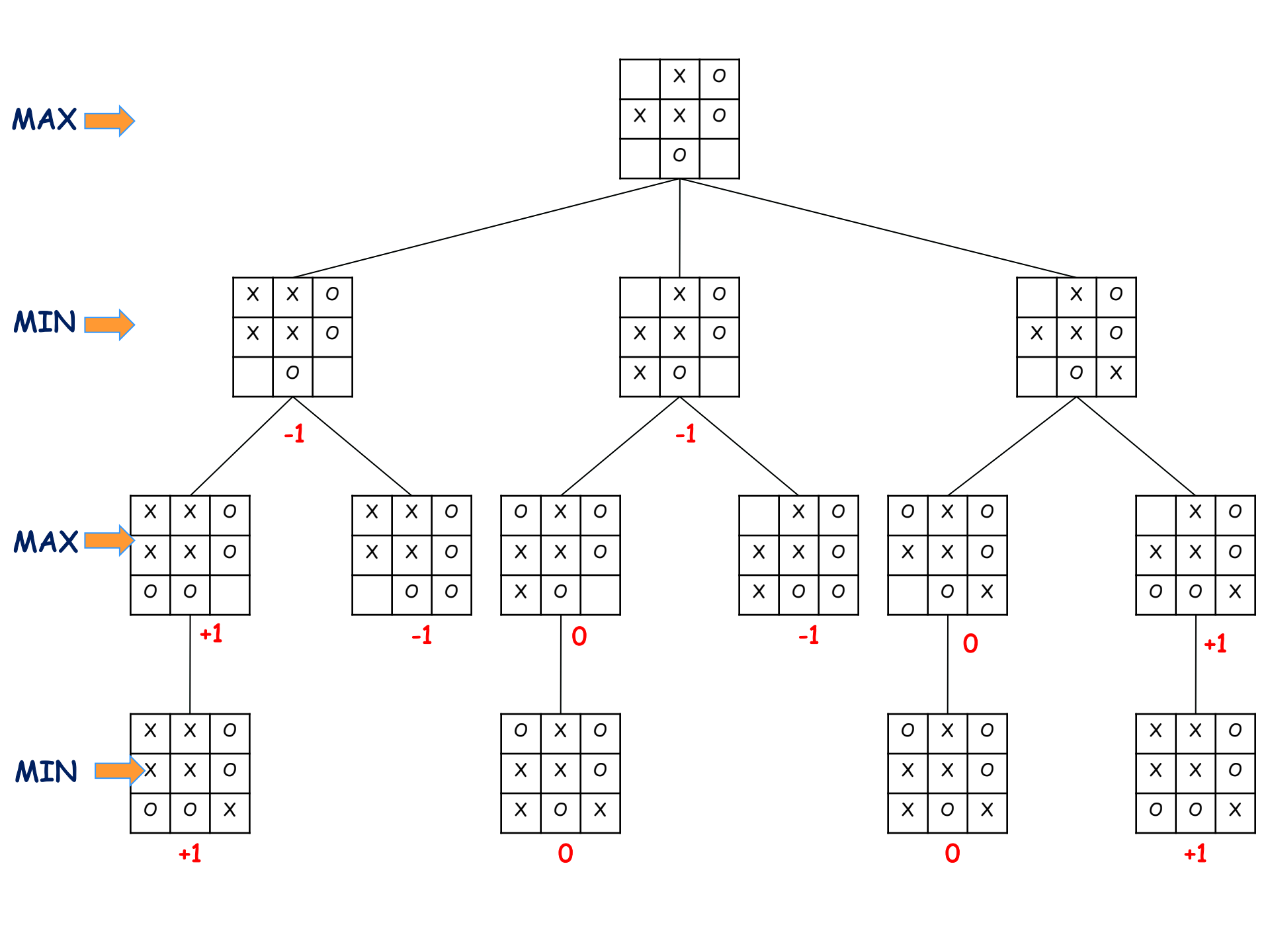

MAX
MIN
MAX
MIN
+1
-1
-1
+1
0
0
+1

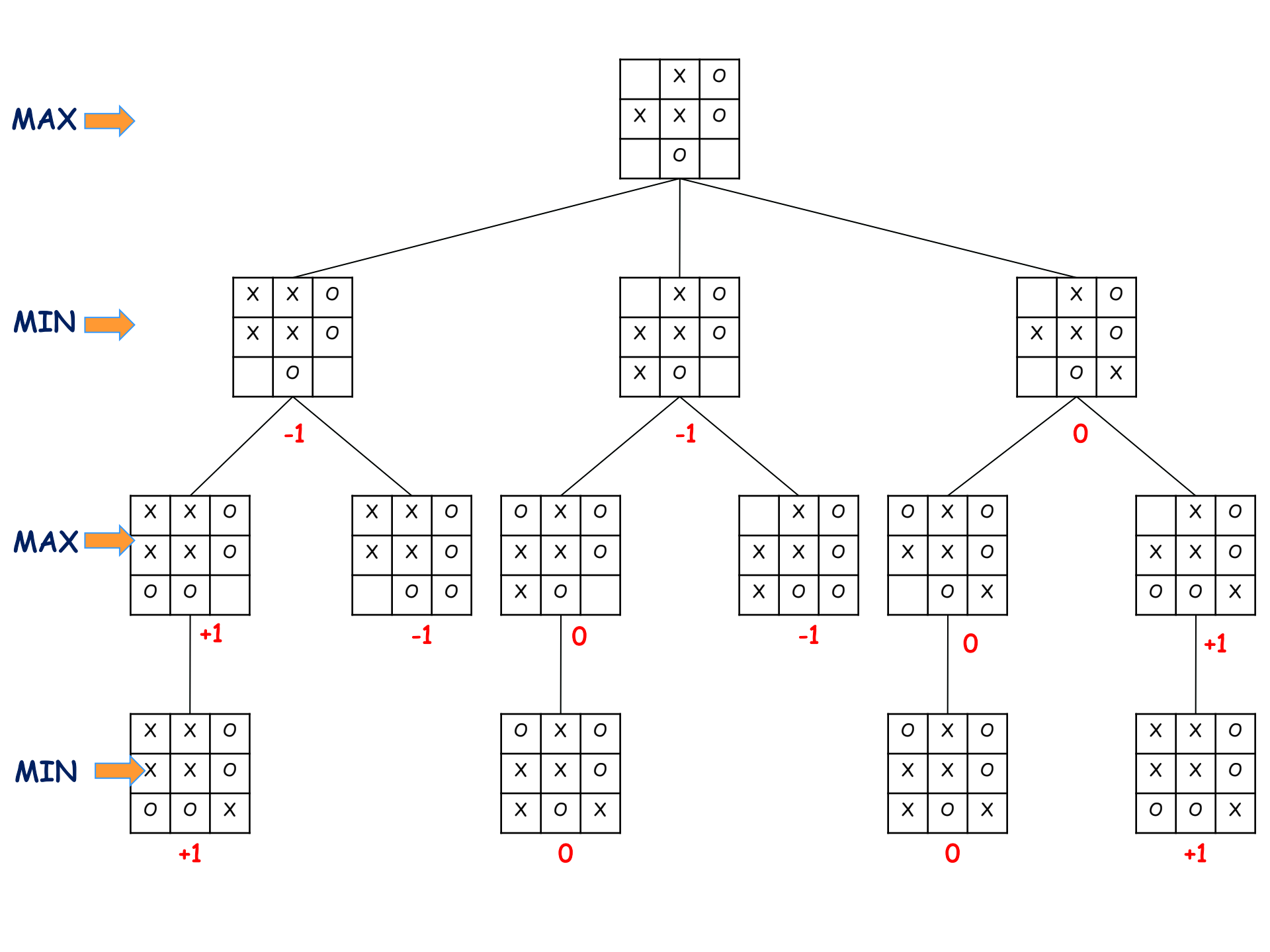

MAX
MIN
MAX
MIN
+1
-1
0
-1
+1
0
0
+1

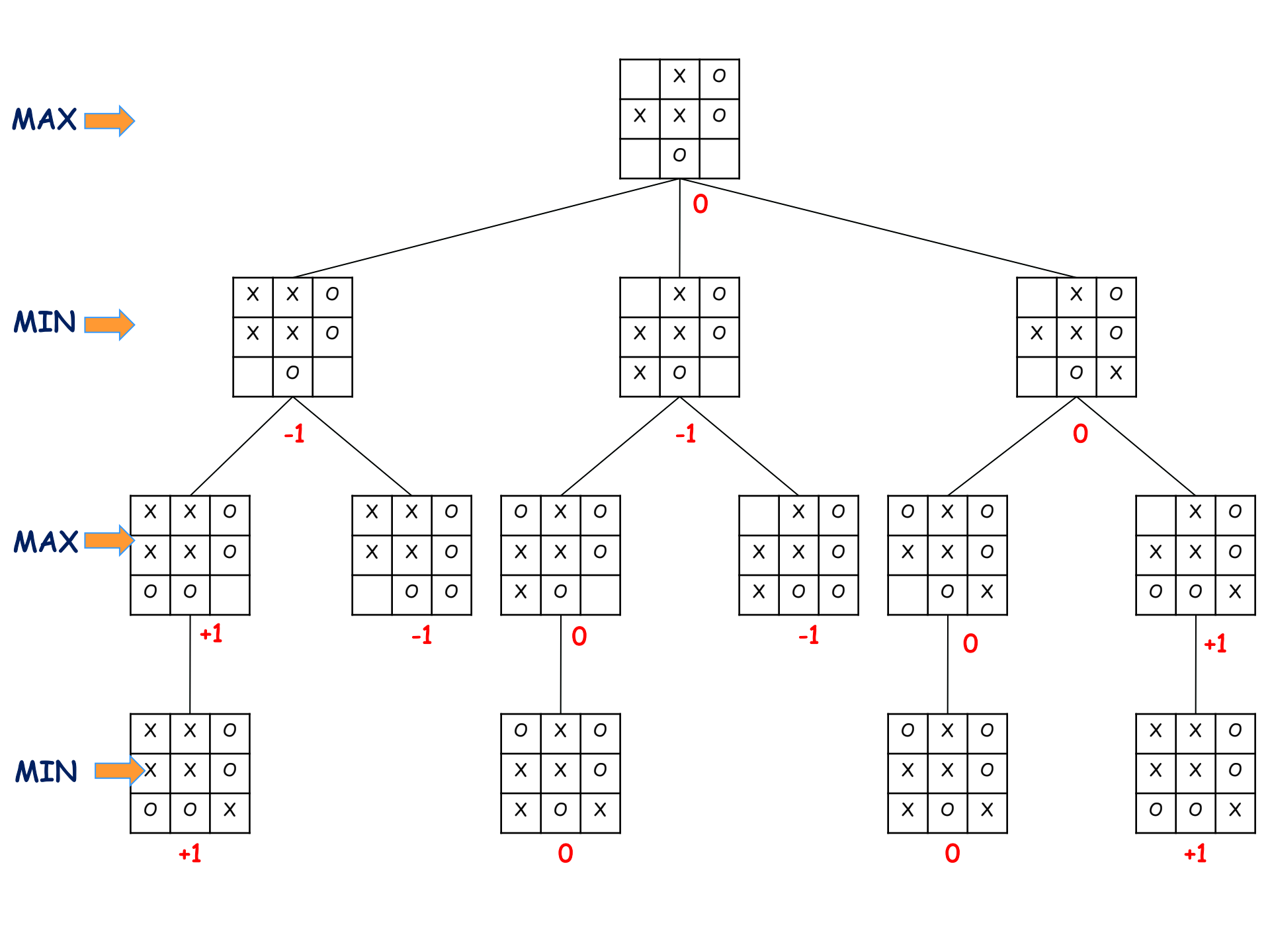

MAX
MIN
MAX
MIN
+1
-1
0
-1
0
+1
+1
0
0
+1


MAX
MIN
MAX
MIN
+1
-1
0
-1
0
+1
+1
0
0
+1

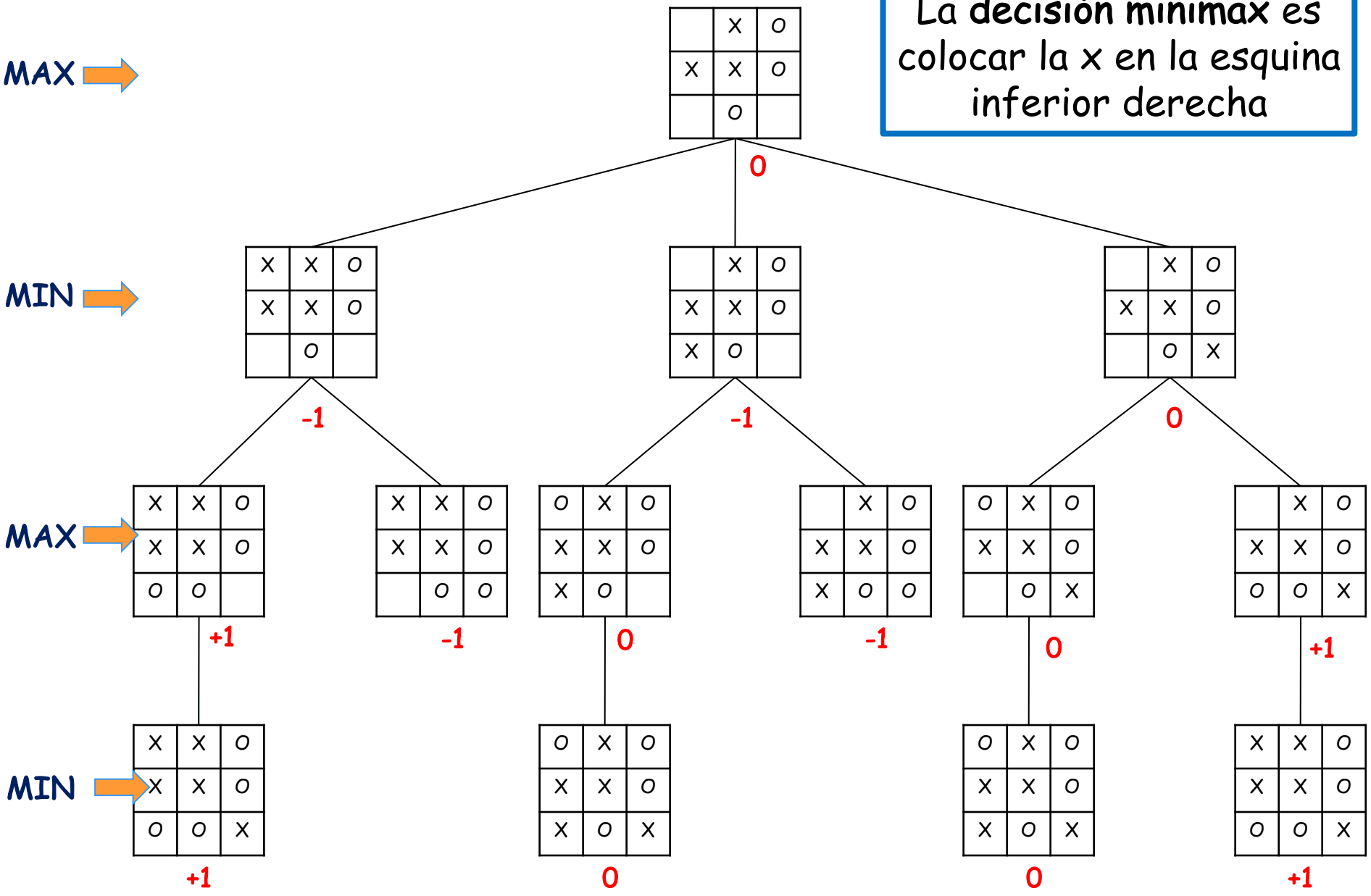

MAX
MIN
MAX
MIN
-1
+1
-1
0
-1
0
+1
+1
0
0
+1


MAX
MIN
MAX
MIN
-1
-1
+1
-1
0
-1
0
+1
+1
0
0
+1

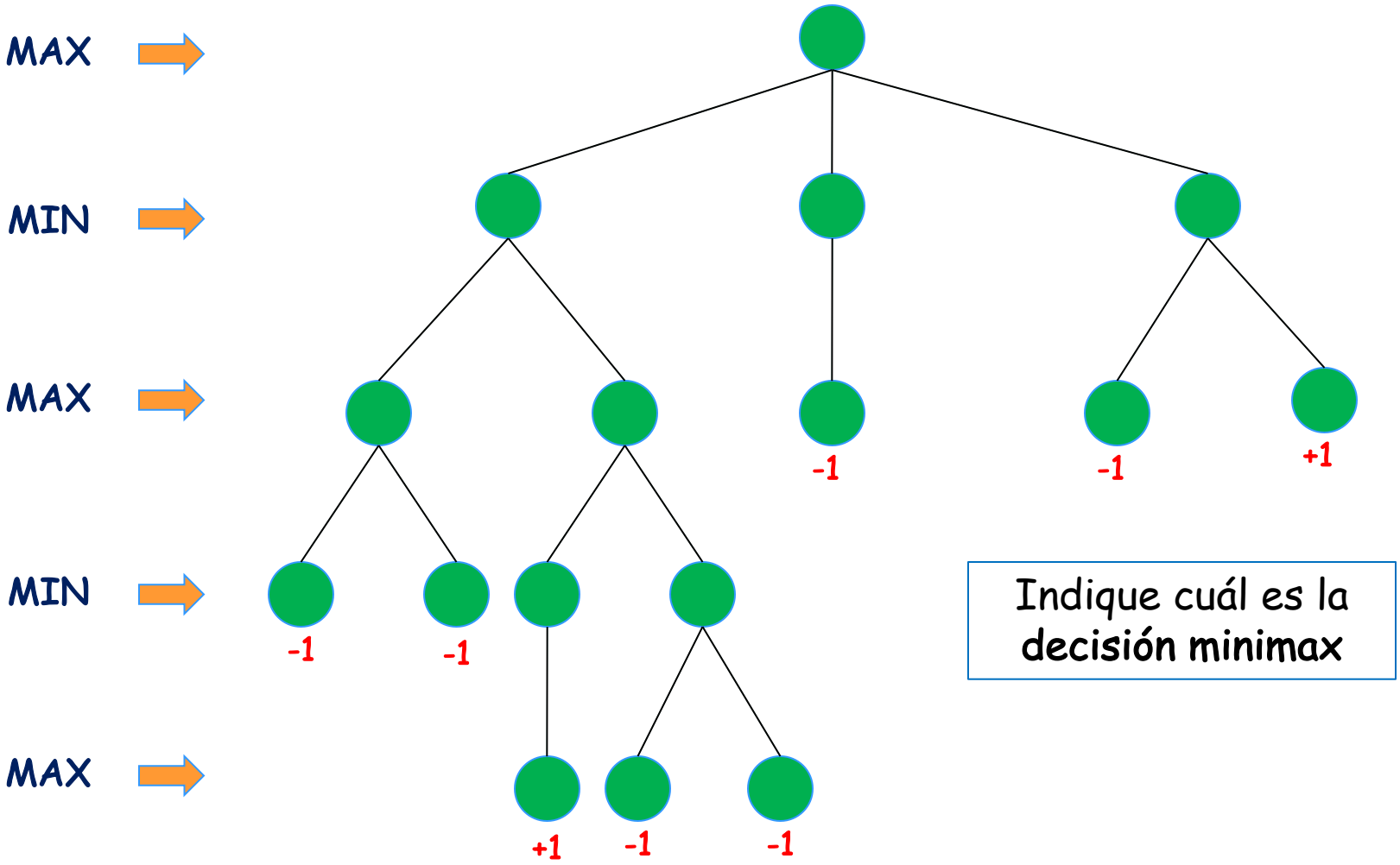

MAX
MIN
MAX
MIN
-1
-1
0
+1
-1
0
-1
0
0
+1
+1
0
+1


MAX
MIN
MAX
MIN
0
-1
-1
0
+1
-1
0
-1
0
0
+1
+1
0
0

La **decisión minimax** es colocar la x en la esquina inferior derecha

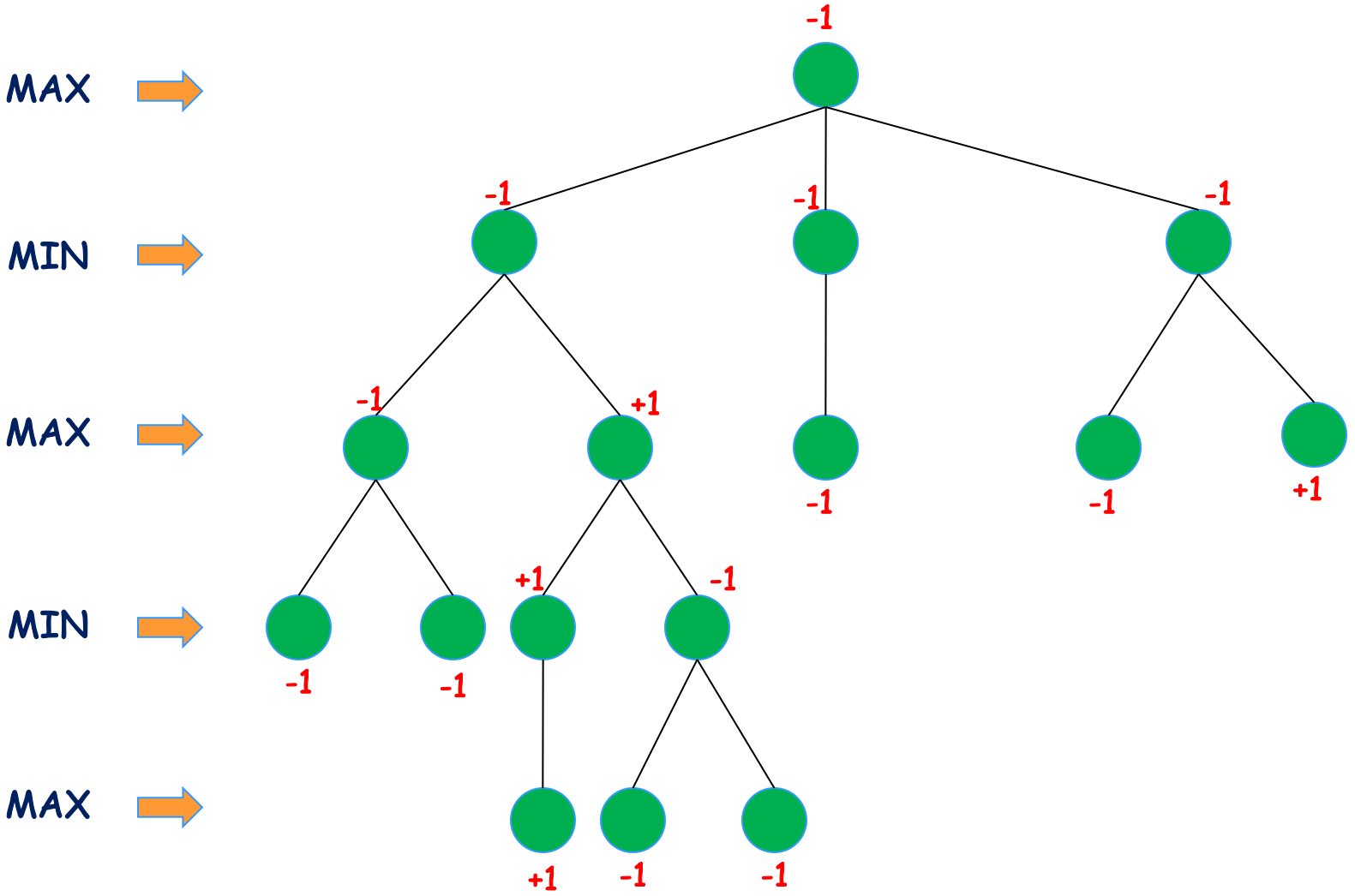


Juegos

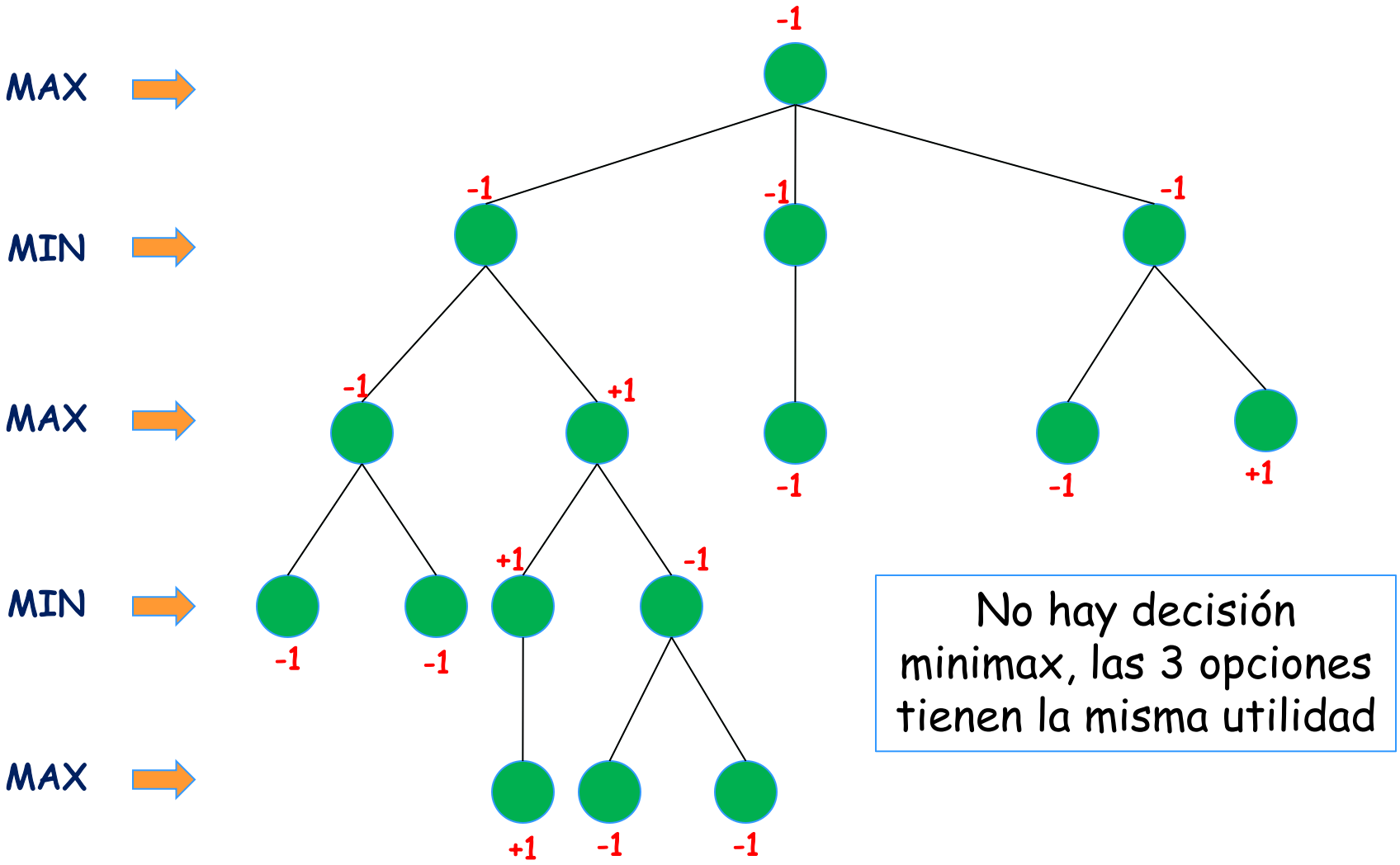


Indique cuál es la **decisión minimax**

Juegos



Juegos



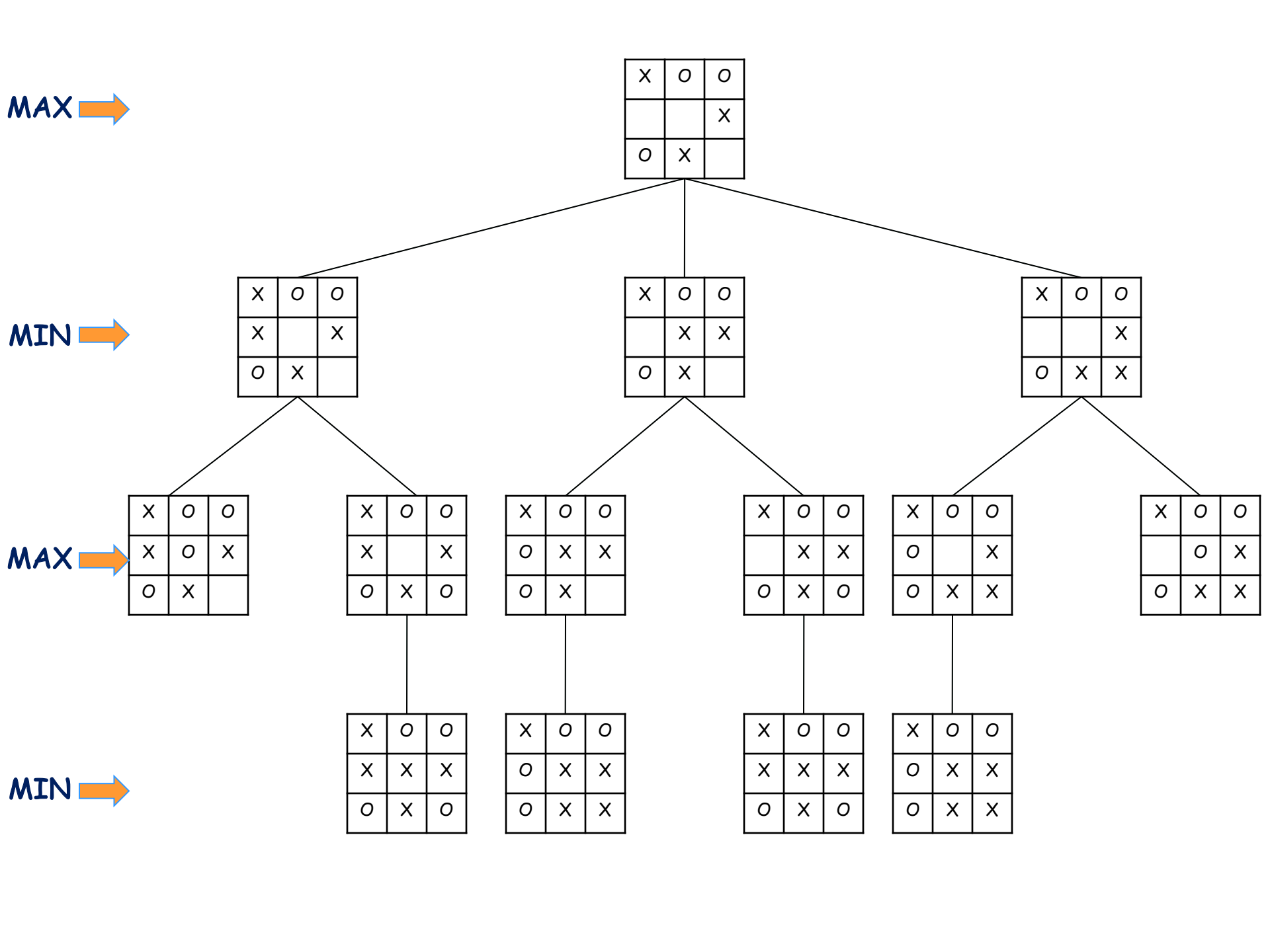
No hay decisión minimax, las 3 opciones tienen la misma utilidad

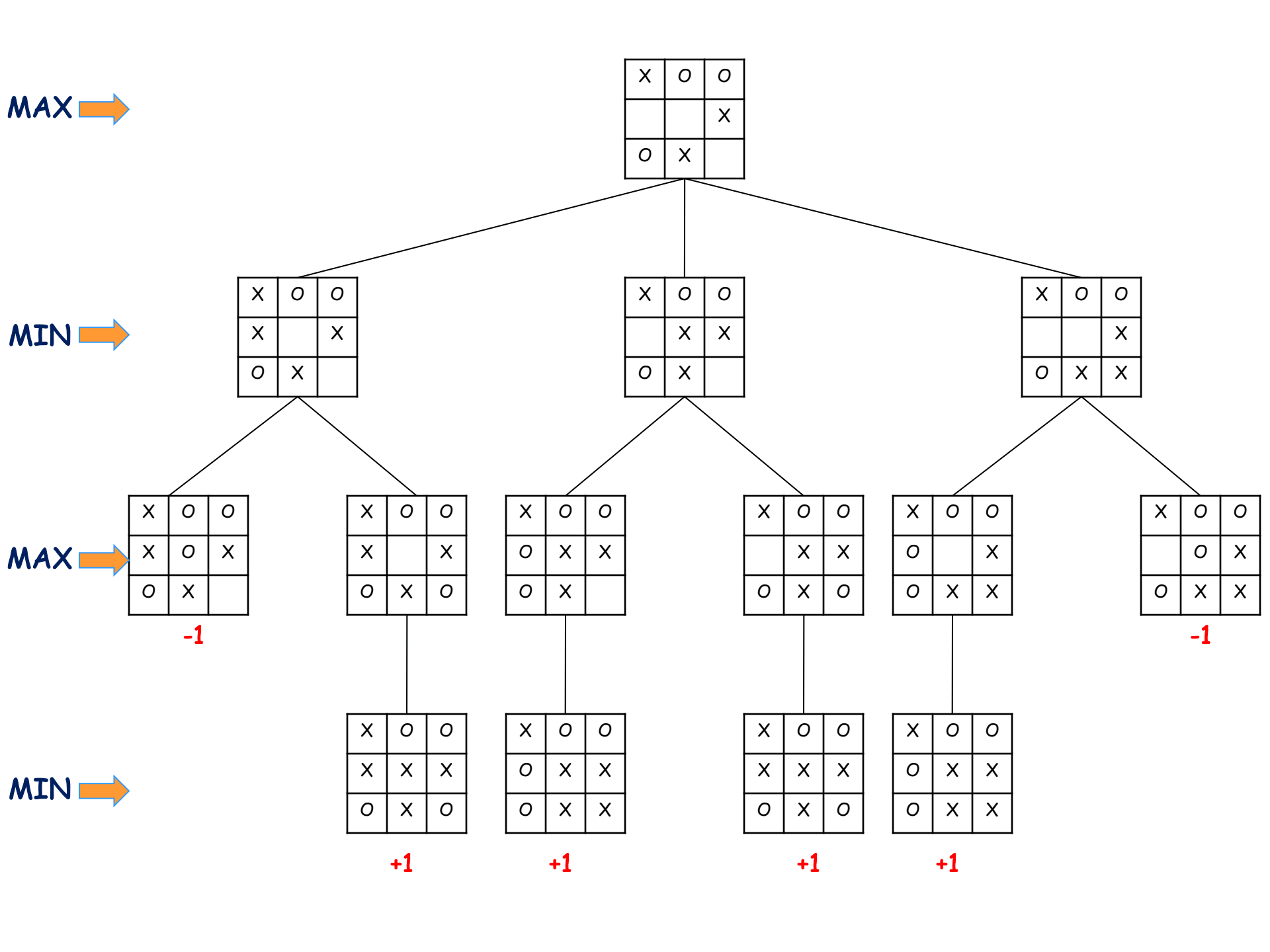
Juegos

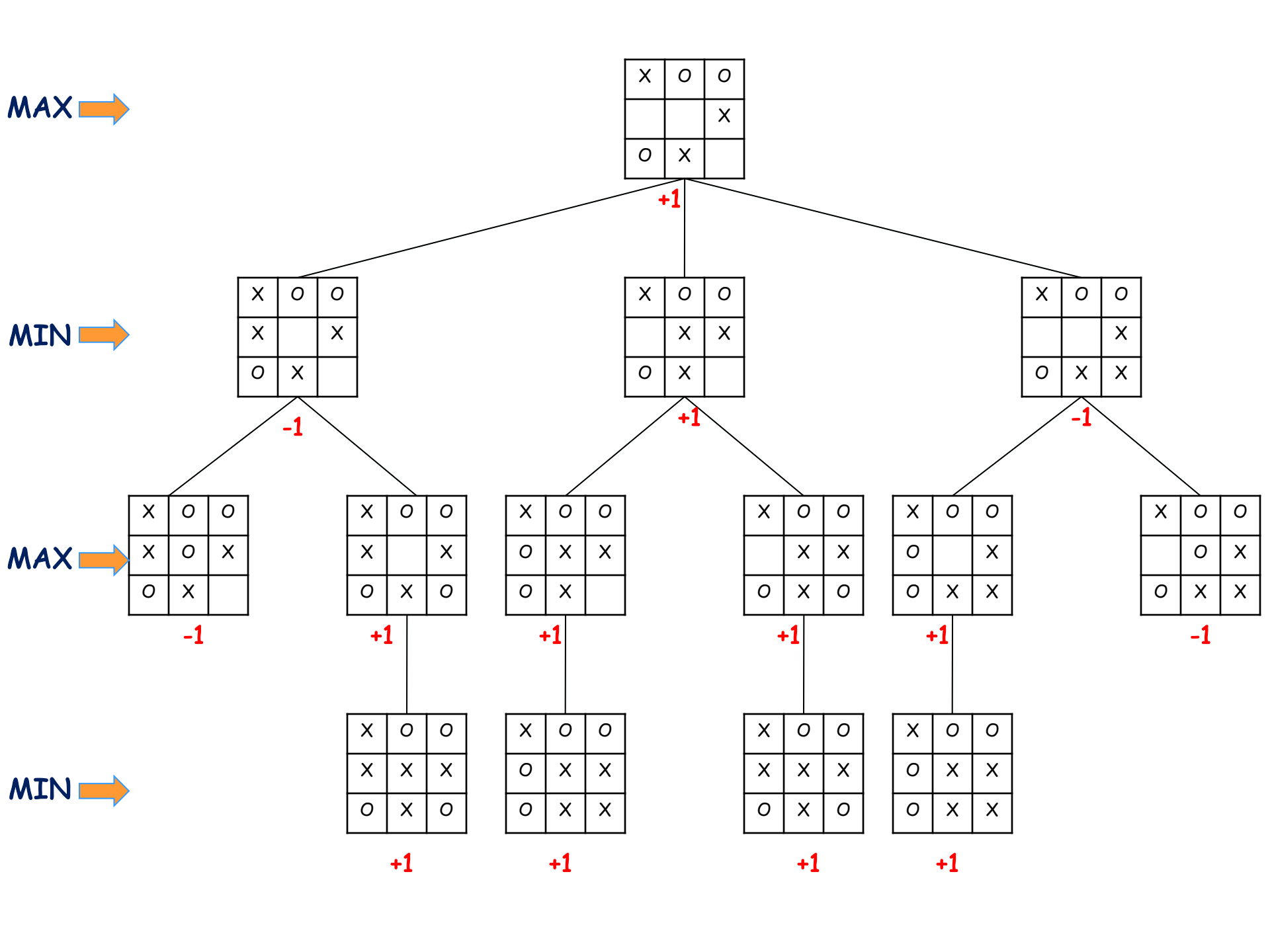
Aplique el algoritmo minimax

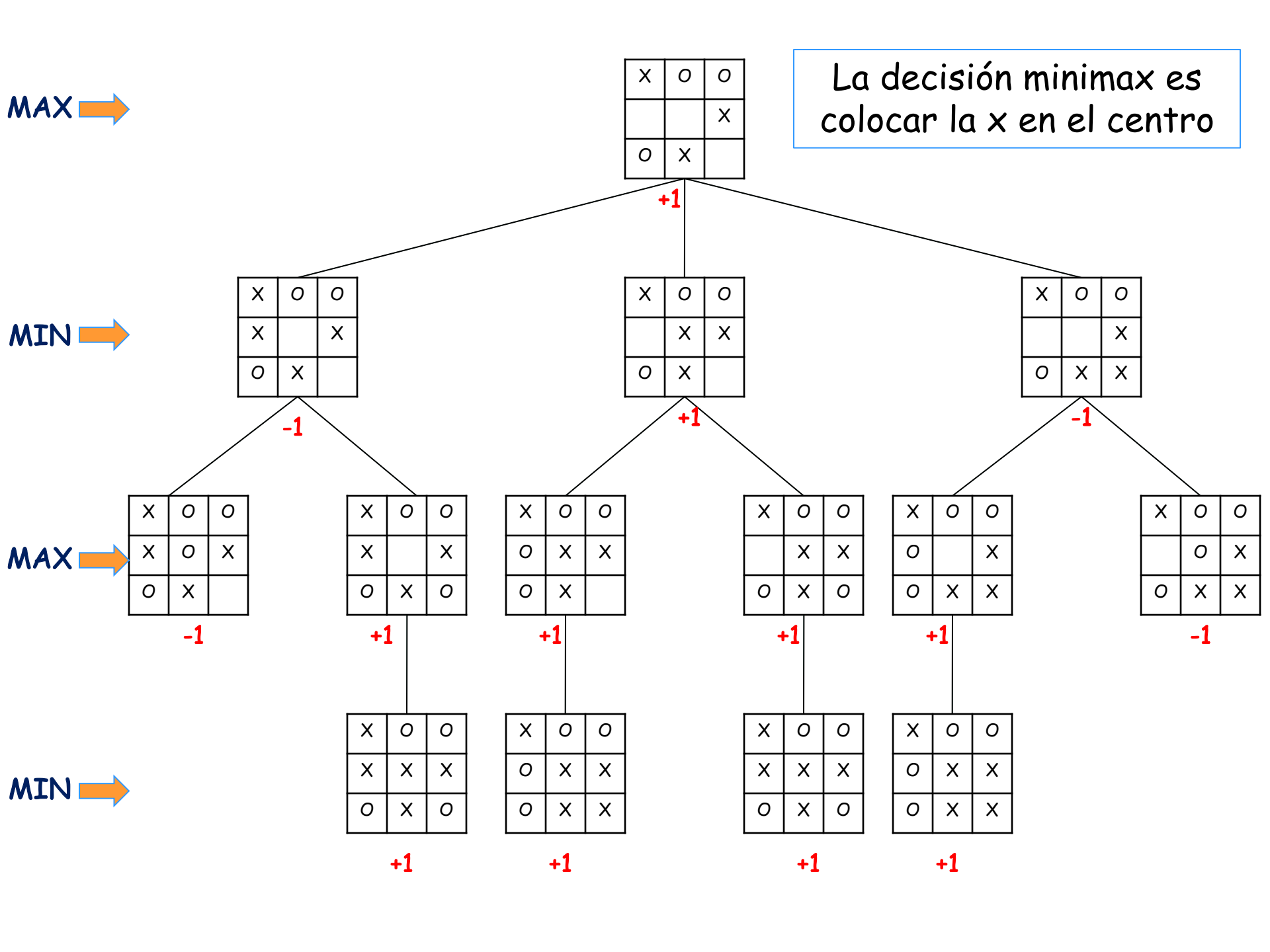
X	O	O
		X
O	X	

- La jugada es de MAX (X)
- Muestre la decisión minimax


MAX
MIN
MAX
MIN

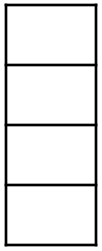

MAX
MIN
MAX
MIN
-1
-1
+1
+1
+1
+1


MAX
MIN
MAX
MIN
+1
-1
+1
-1
-1
+1
+1
+1
+1
+1
+1
+1
+1
-1

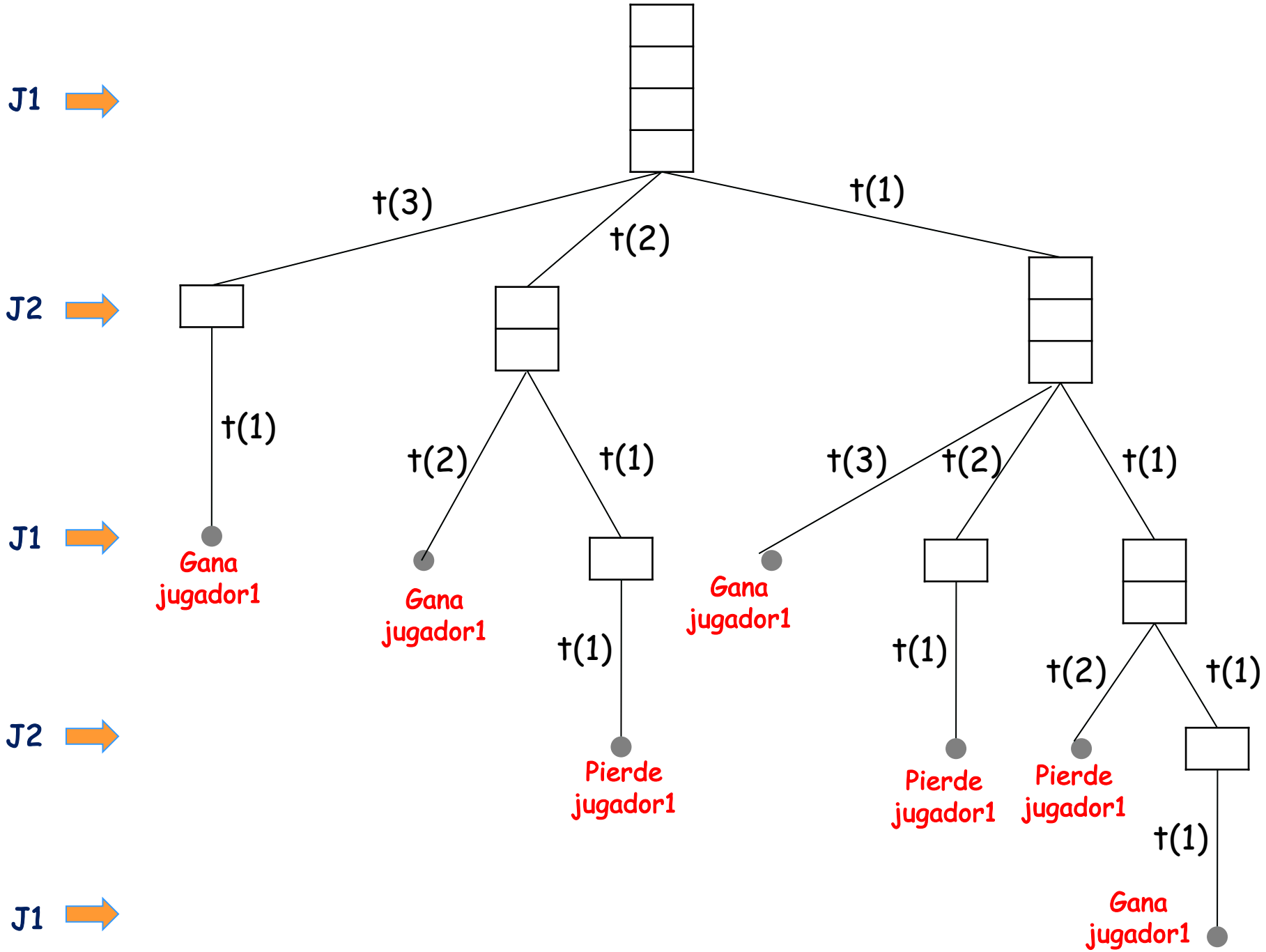

La decisión minimax es colocar la x en el centro
MAX
MIN
MAX
MIN
+1
-1
+1
-1
-1
+1
+1
+1
+1
-1
+1
+1
+1
+1

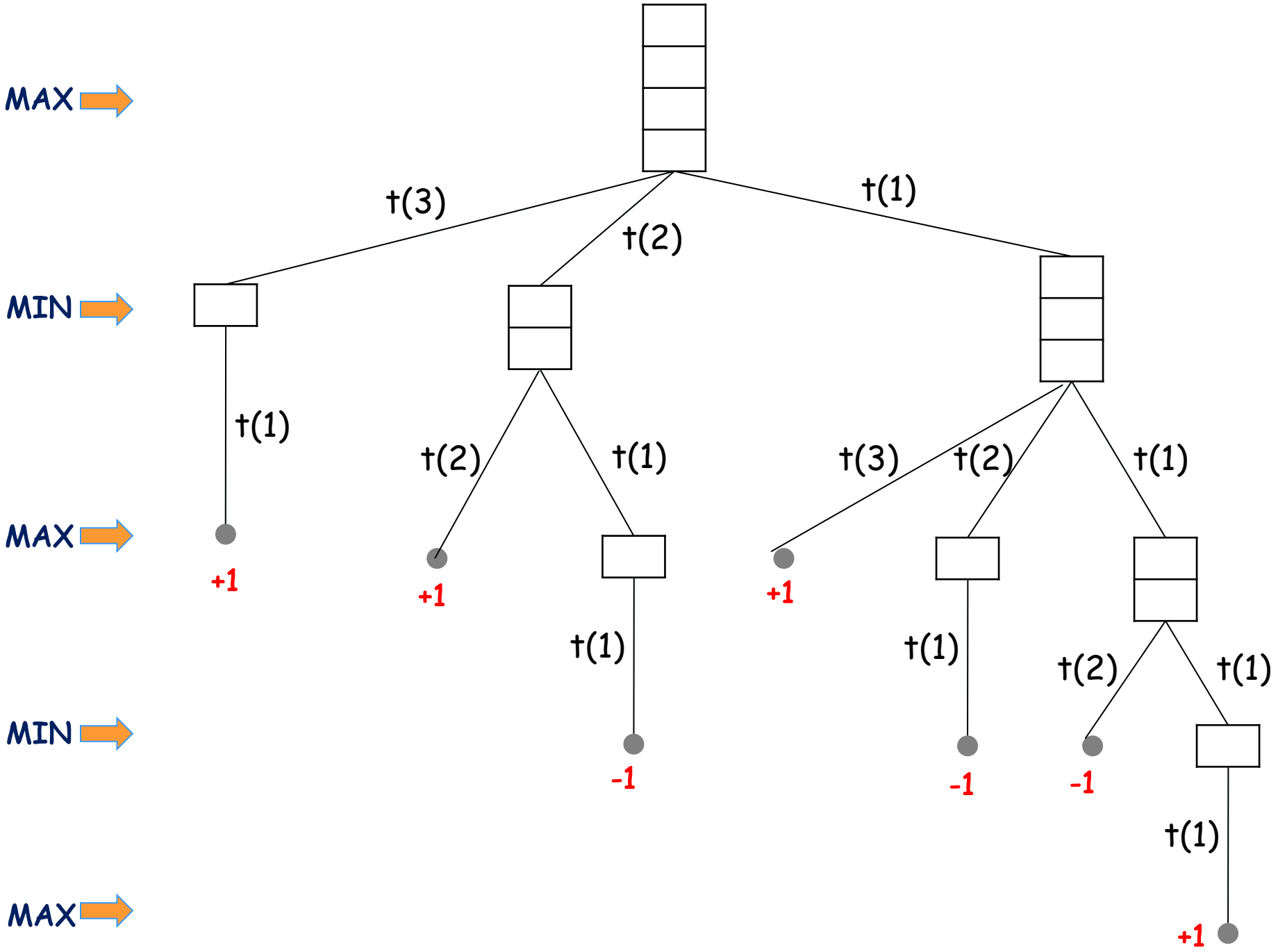
Juegos

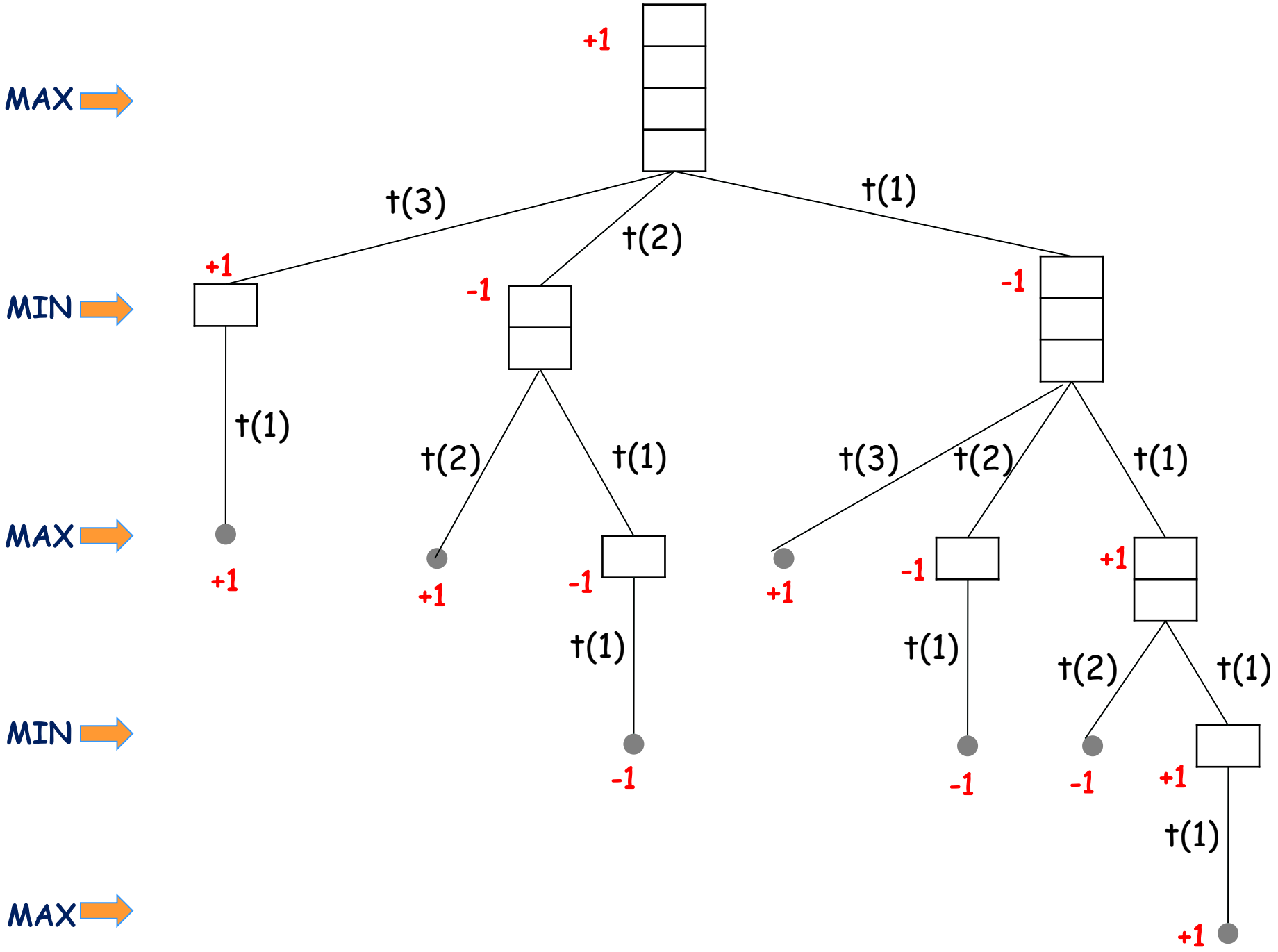
Aplique el algoritmo minimax

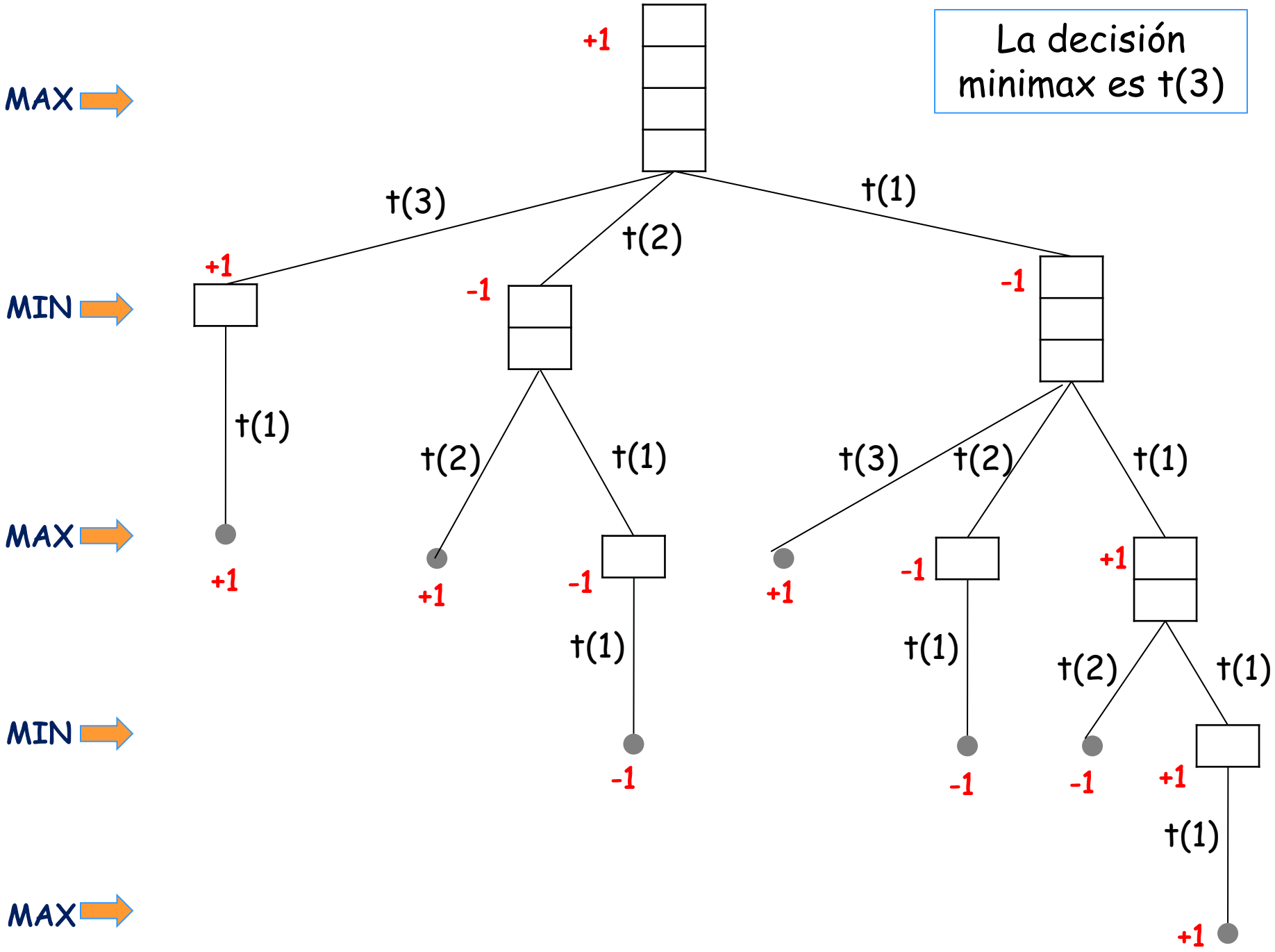


- **El juego del NIM**. Se tiene una pila de 4 fichas de la cual cada jugador puede tomar 1, 2 ó 3. El objetivo de cada jugador es obligar a su adversario a tomar la última ficha. Como los elementos están apilados, solo se pueden tomar fichas de su tope
- Indique la decisión minimax


J1
J2
J1
J2
J1
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(1)
Gana jugador1
Gana jugador1
Pierde jugador1
Gana jugador1
Pierde jugador1
Pierde jugador1
Gana jugador1


MAX
MIN
MAX
MIN
MAX
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
t(1)
t(2)
t(1)
+1
+1
+1
-1
-1
-1
+1


MAX
MIN
MAX
MIN
MAX
+1
t(3)
t(2)
t(1)
+1
-1
-1
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
+1
+1
-1
-1
-1
+1
t(1)
t(1)
t(2)
t(1)
-1
-1
-1
+1
t(1)
+1


La decisión minimax es t(3)
MAX
MIN
MAX
MIN
MAX
+1
t(3)
t(2)
t(1)
+1
-1
-1
t(1)
t(2)
t(1)
t(3)
t(2)
t(1)
+1
+1
-1
-1
t(1)
+1
-1
-1
t(2)
t(1)
+1
t(1)
+1

Juegos

Complejidad de minimax

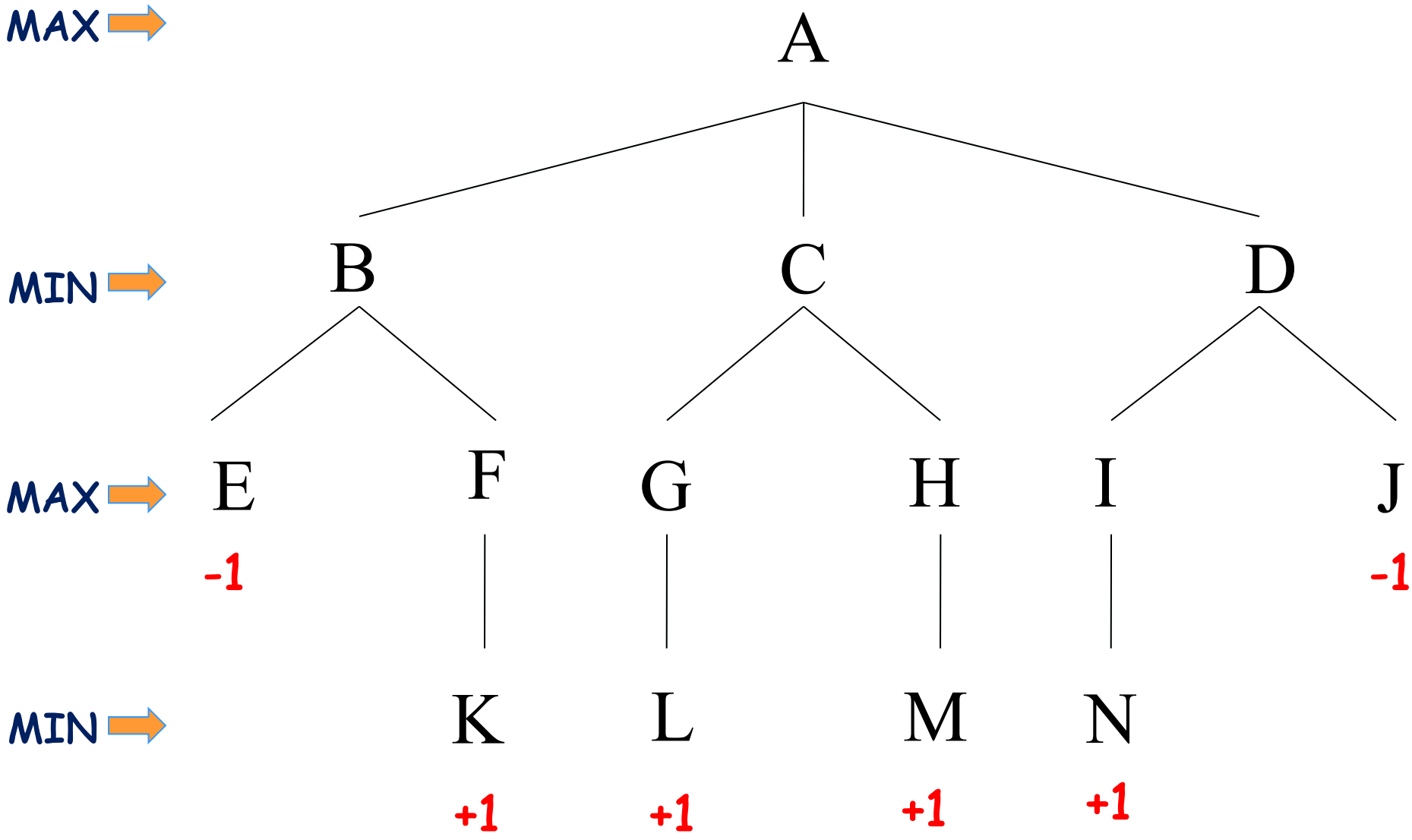
Si la profundidad máxima del árbol es **m** y hay **b** movimientos legales en cada punto, se tiene:

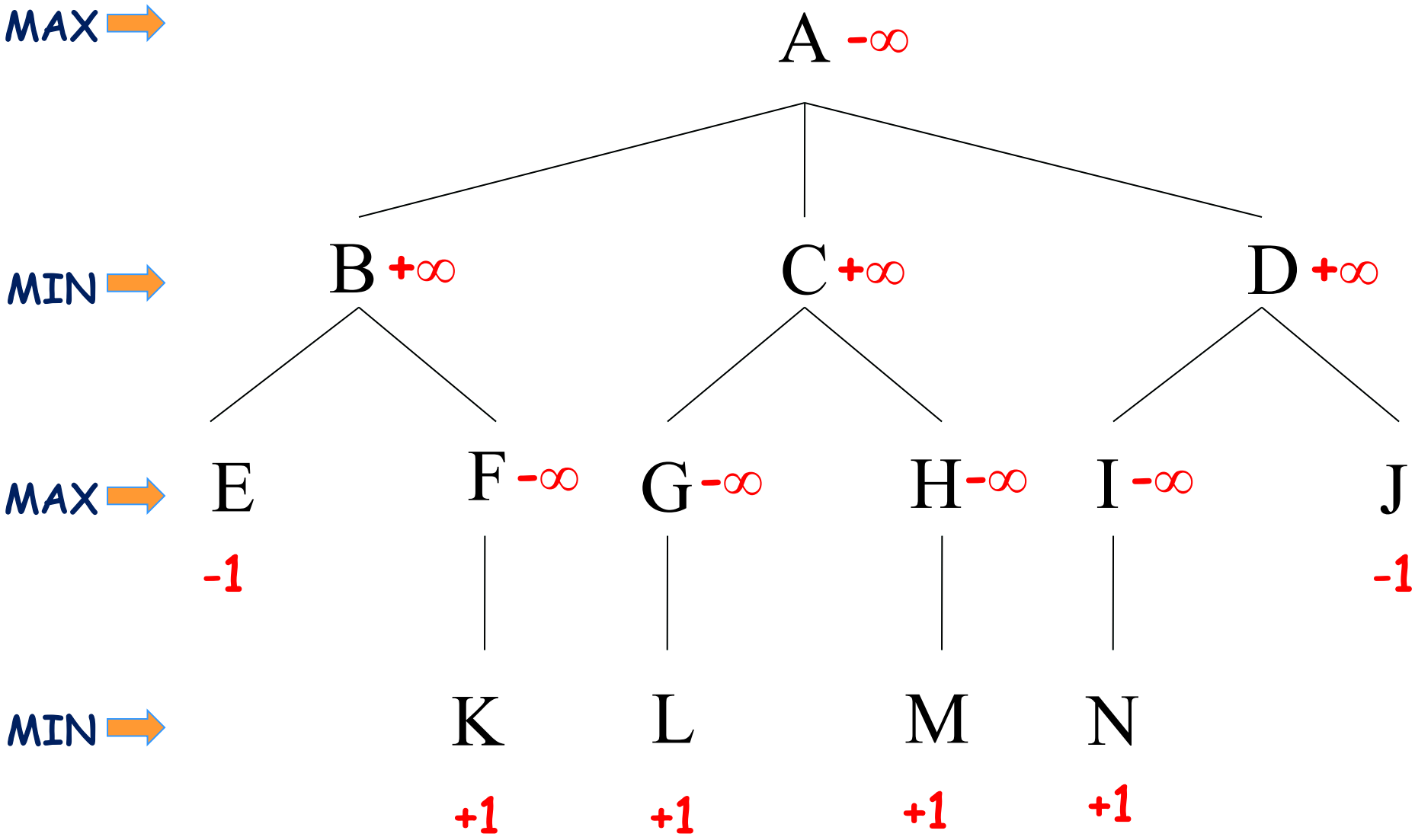
- **Complejidad temporal:** $O(b^m)$
- **Complejidad espacial:** $O(b*m)$

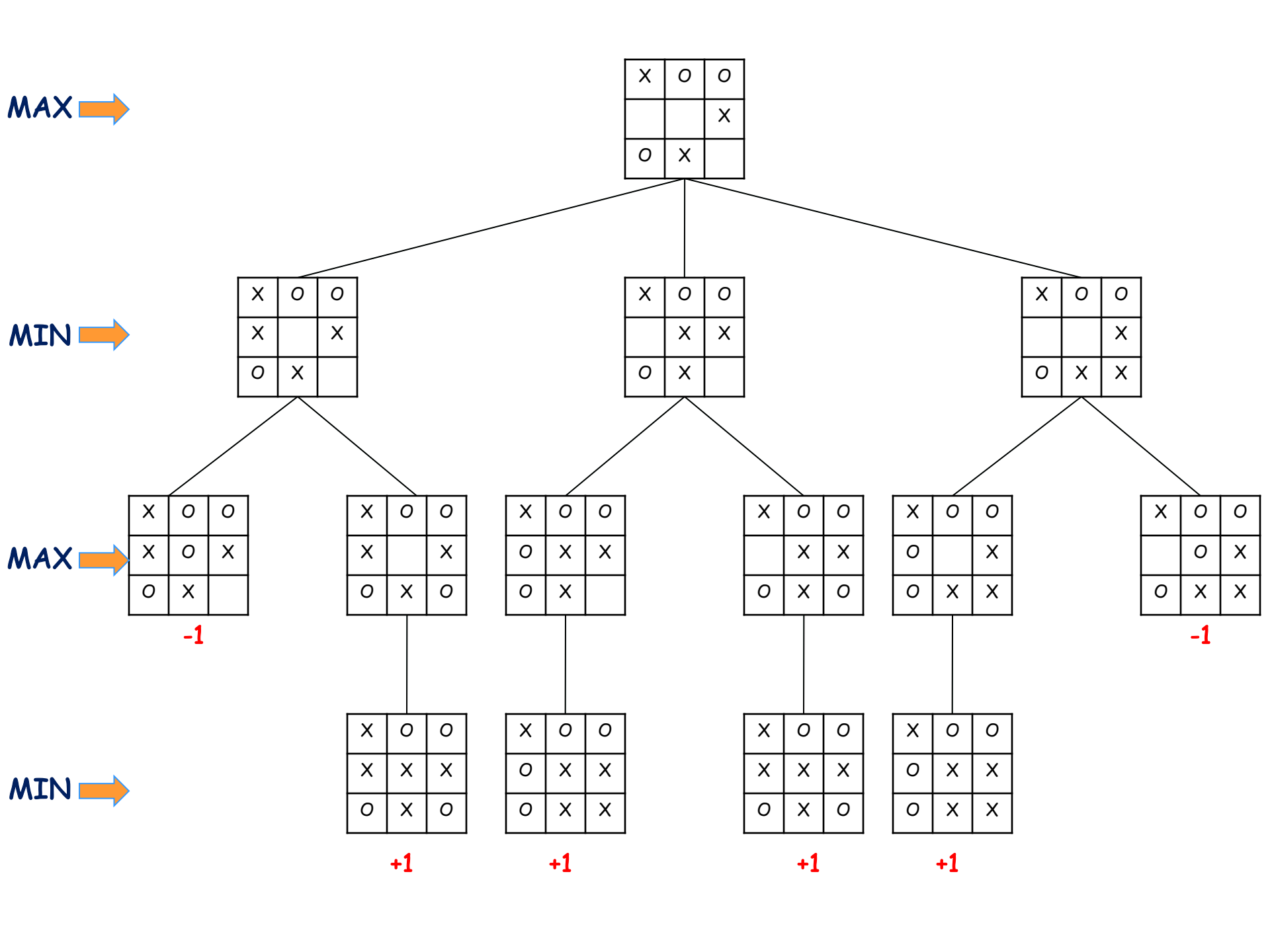
Juegos

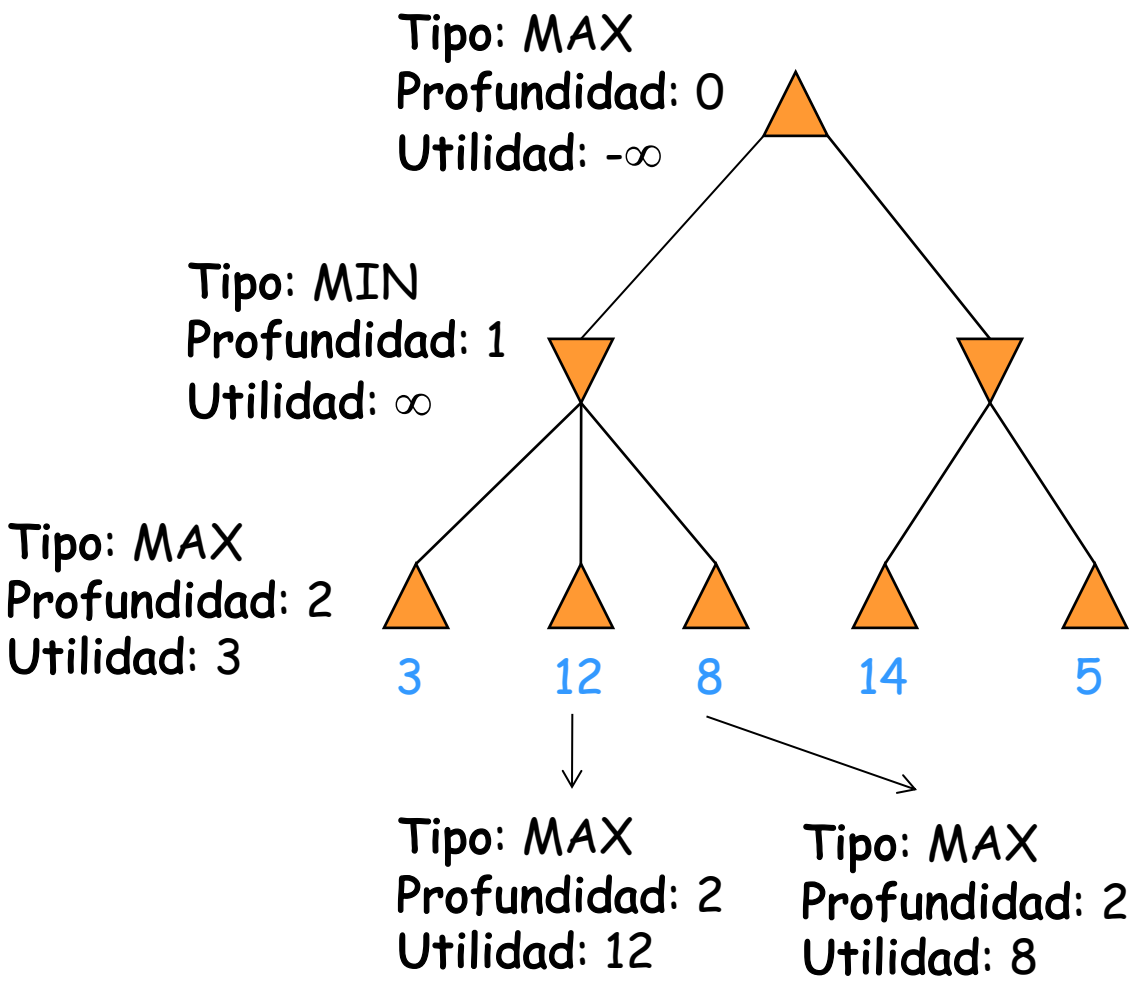
Implementación

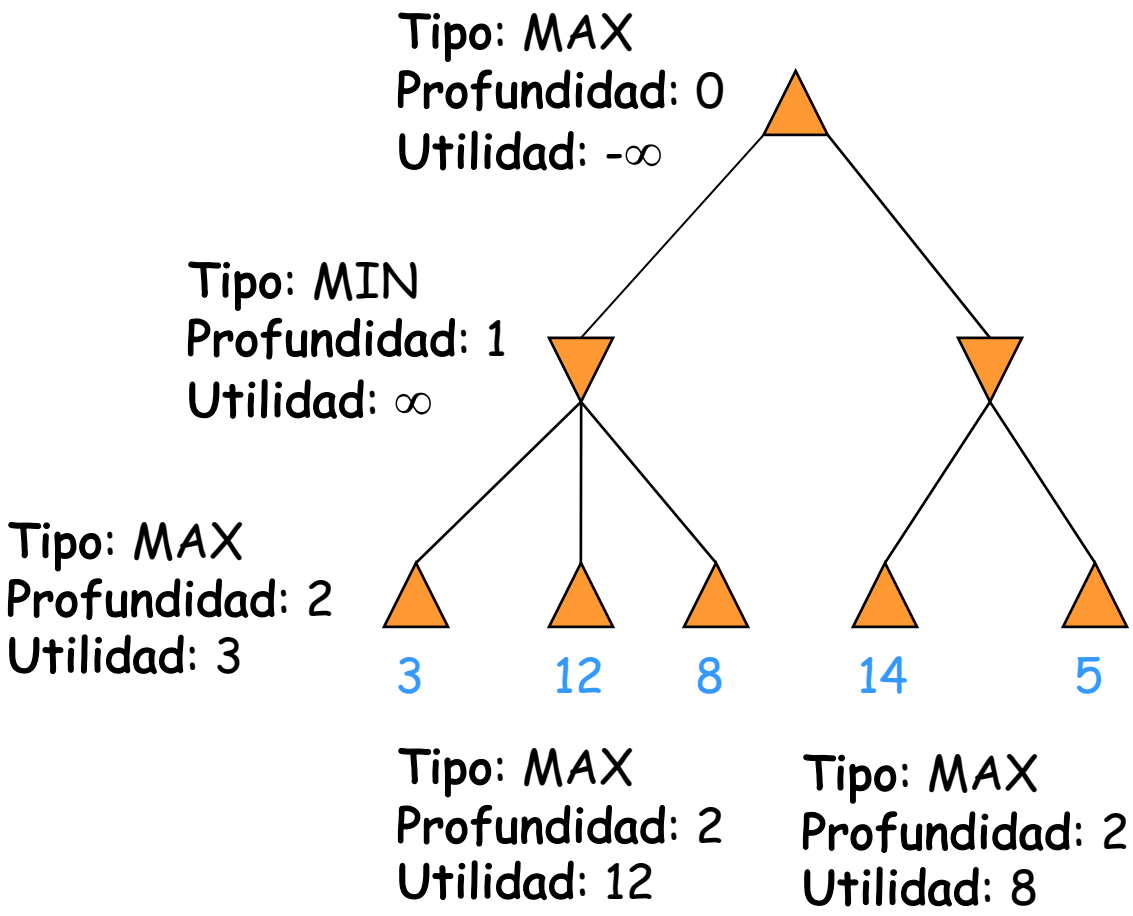
- Se debe guardar el tipo de nodo, MAX o MIN, la profundidad y la utilidad. Inicializar la utilidad de los nodos MAX en -∞ y los MIN en +∞
- Se utiliza una pila como estructura de datos
- Cuando se expanda un nodo hoja se calcula su utilidad
- Una vez terminada la construcción, se recorren los nodos en un arreglo desde los más profundos hasta la raíz. Cada nodo informa al padre su utilidad
- Cuando se llegue a la raíz se tendrá el valor minimax


MAX
A
MIN
B
C
D
MAX
E
F
G
H
I
J
-1
-1
MIN
K
L
M
N
+1
+1
+1
+1

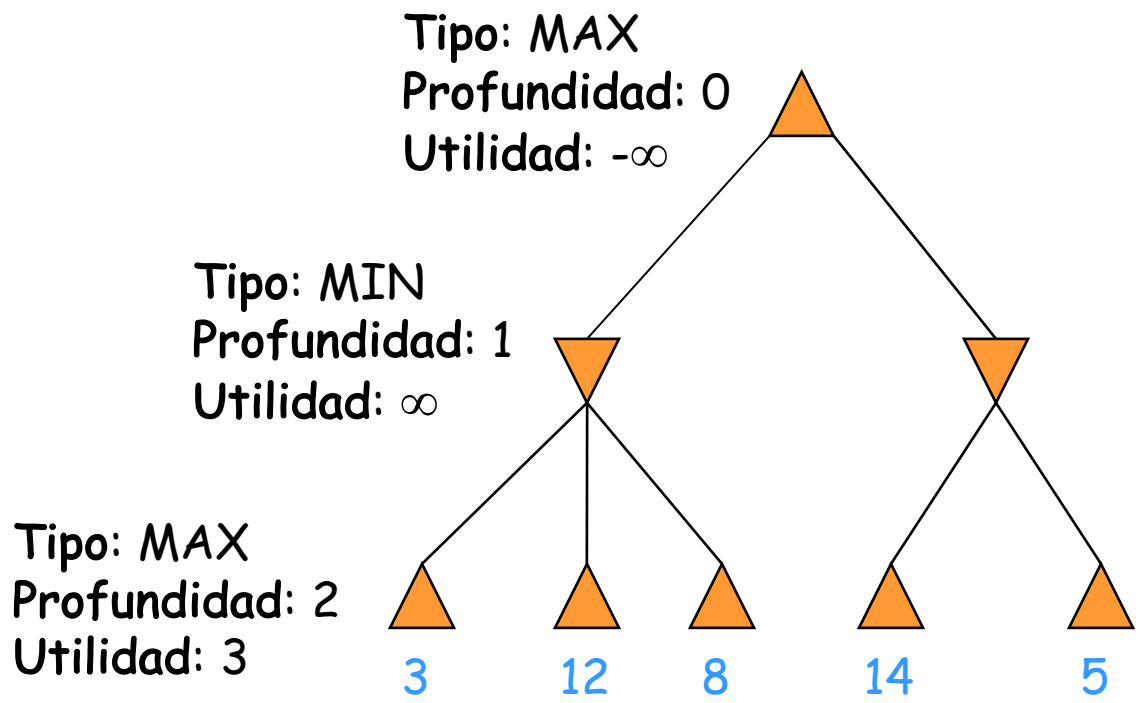

MAX
A -∞
MIN
B +∞
C +∞
D +∞
MAX
E
F -∞
G -∞
H -∞
I -∞
J
-1
-1
MIN
K
L
M
N
+1
+1
+1
+1


MAX
MIN
MAX
MIN
-1
-1
+1
+1
+1
+1


Tipo: MAX
Profundidad: 0
Utilidad: -∞
Tipo: MIN
Profundidad: 1
Utilidad: ∞
Tipo: MAX
Profundidad: 2
Utilidad: 3
3
12
8
14
5
Tipo: MAX
Profundidad: 2
Utilidad: 12
Tipo: MAX
Profundidad: 2
Utilidad: 8

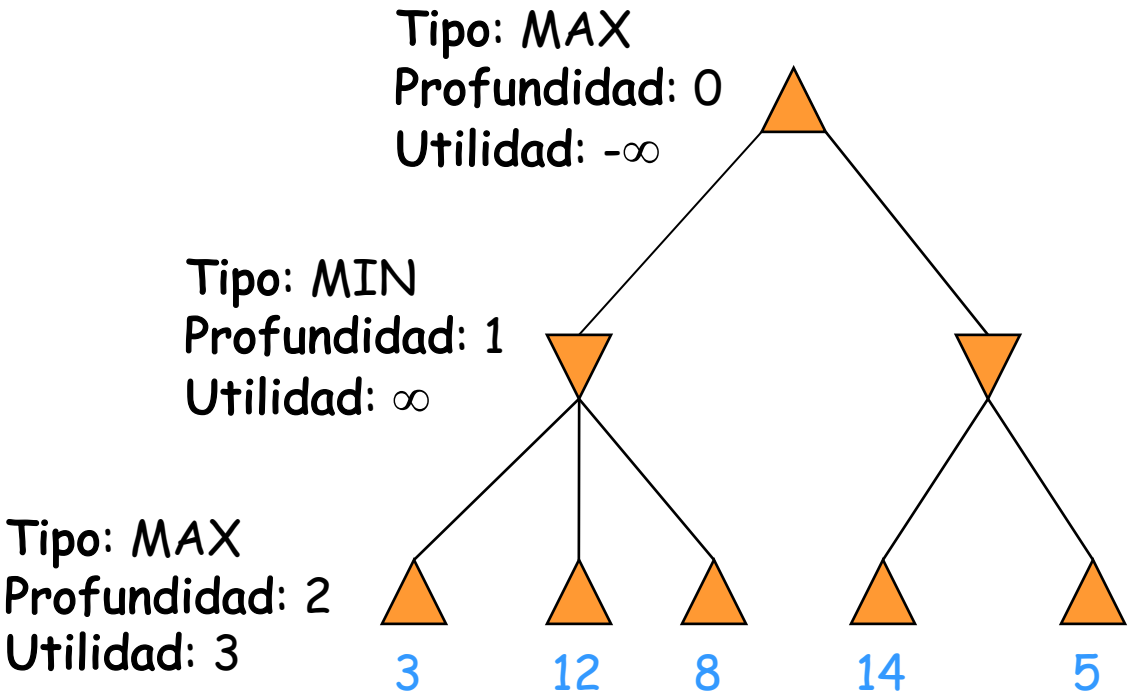


Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: ∞	Tipo: MAX Profundidad: 2 Utilidad: 3	



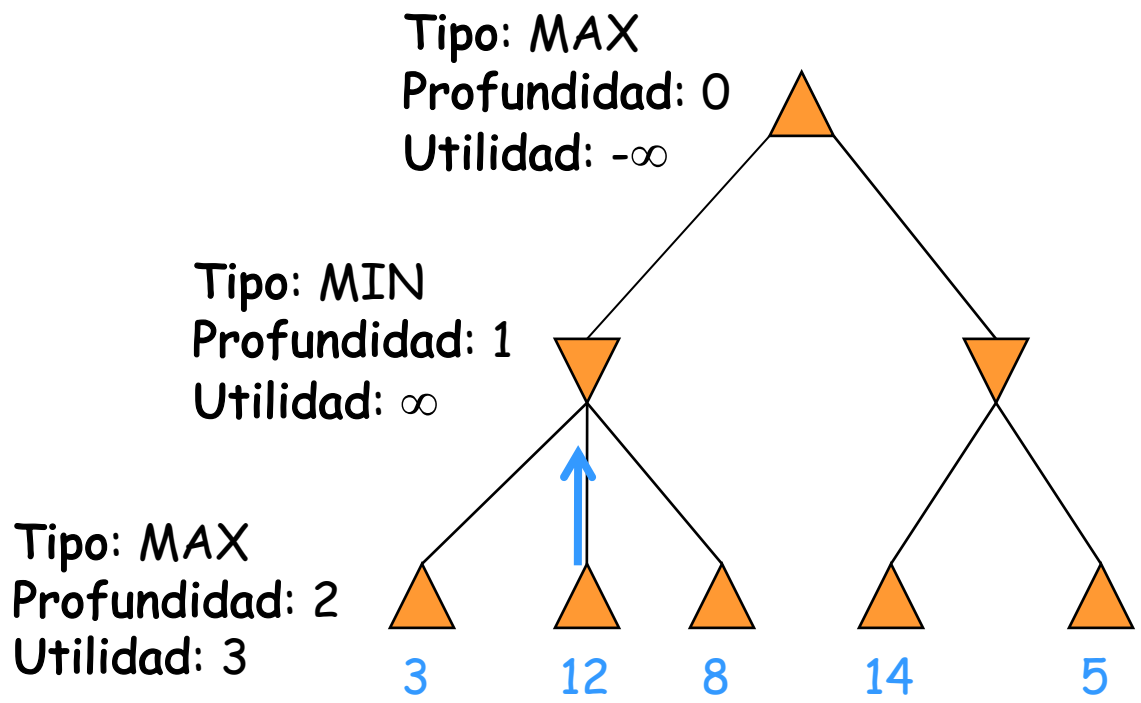
1. Se busca un nodo en la profundidad máxima presente en el vector

Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: ∞	Tipo: MAX Profundidad: 2 Utilidad: 3	



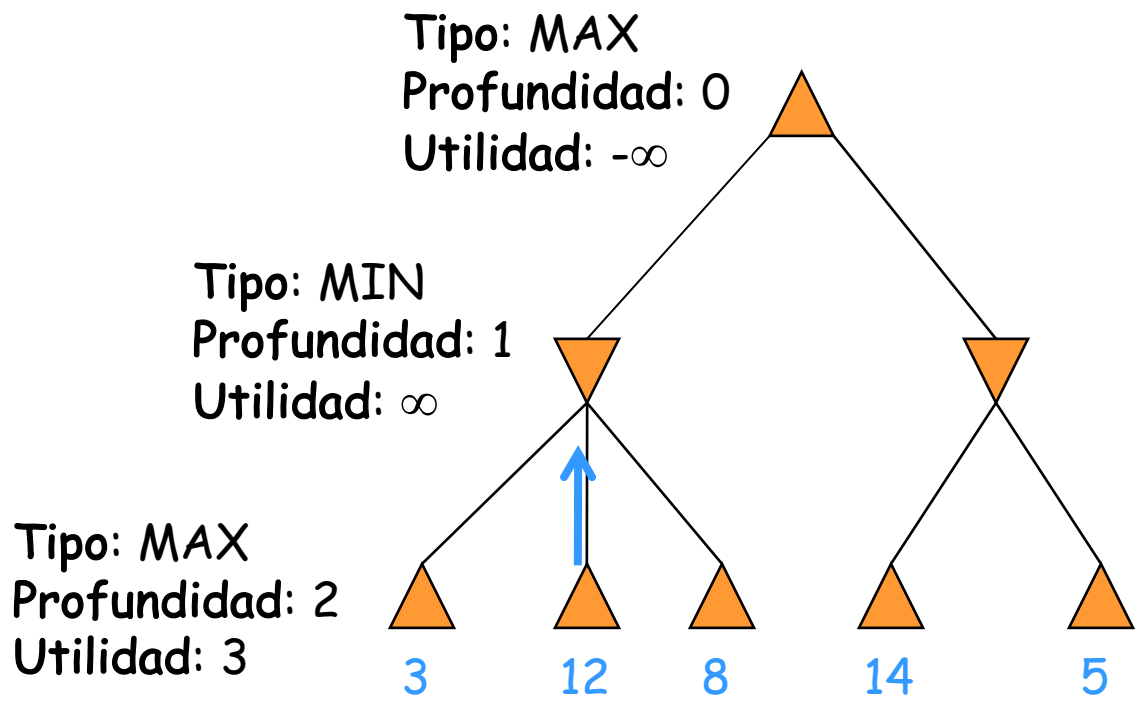
1. Se busca un nodo en la profundidad máxima presente en el vector

Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: ∞	Tipo: MAX Profundidad: 2 Utilidad: 3	



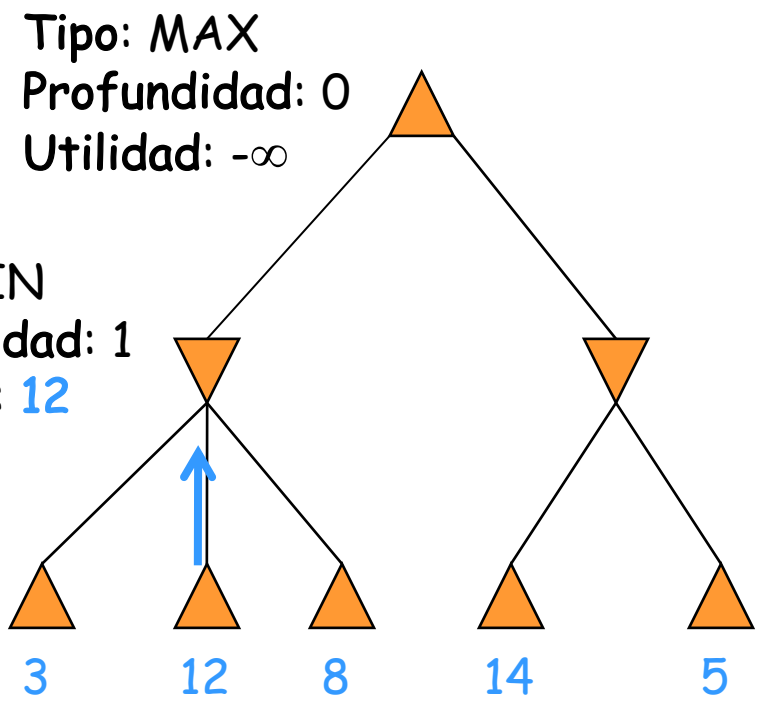
2. El nodo seleccionado informa a su padre el valor de su utilidad

Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: ∞	Tipo: MAX Profundidad: 2 Utilidad: 3	



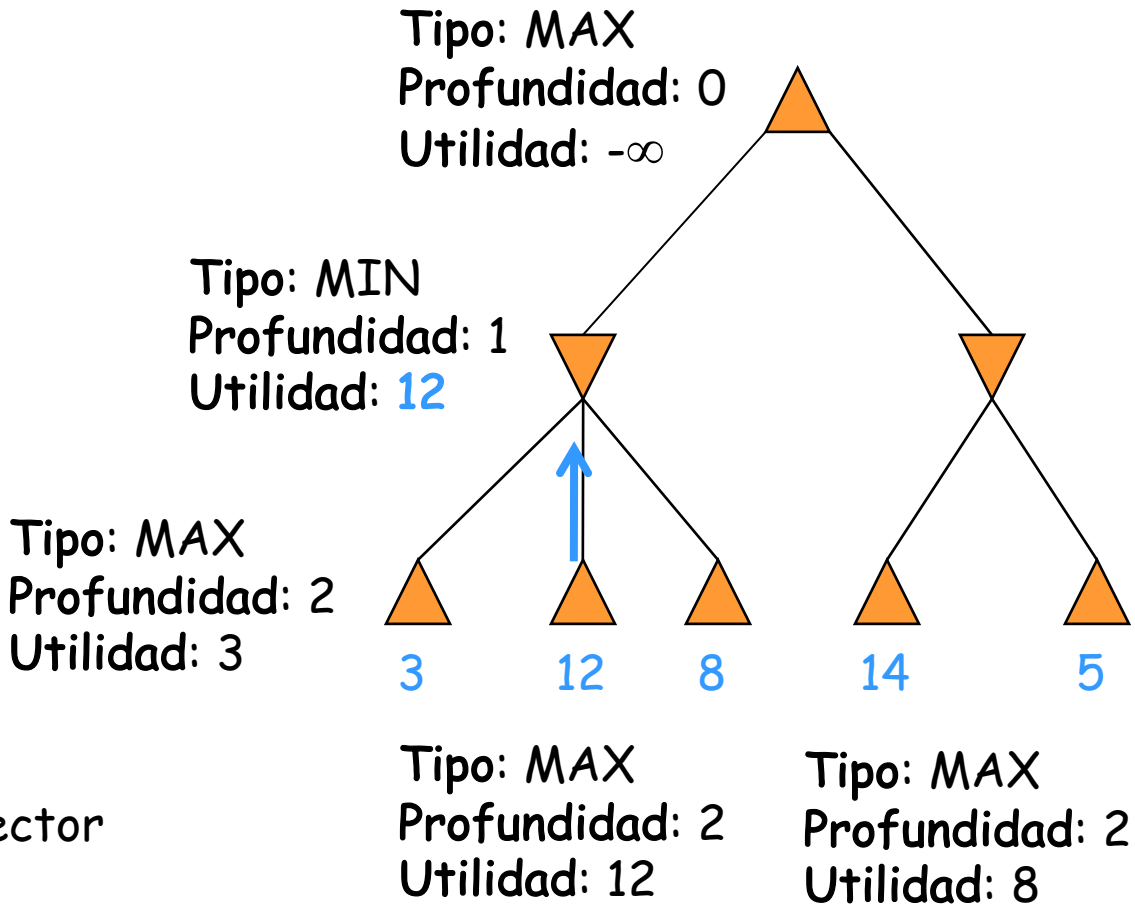
3. Se actualiza la utilidad del padre. Si es MAX se saca el máximo, si es MIN el mínimo

Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: ∞	Tipo: MAX Profundidad: 2 Utilidad: 3	



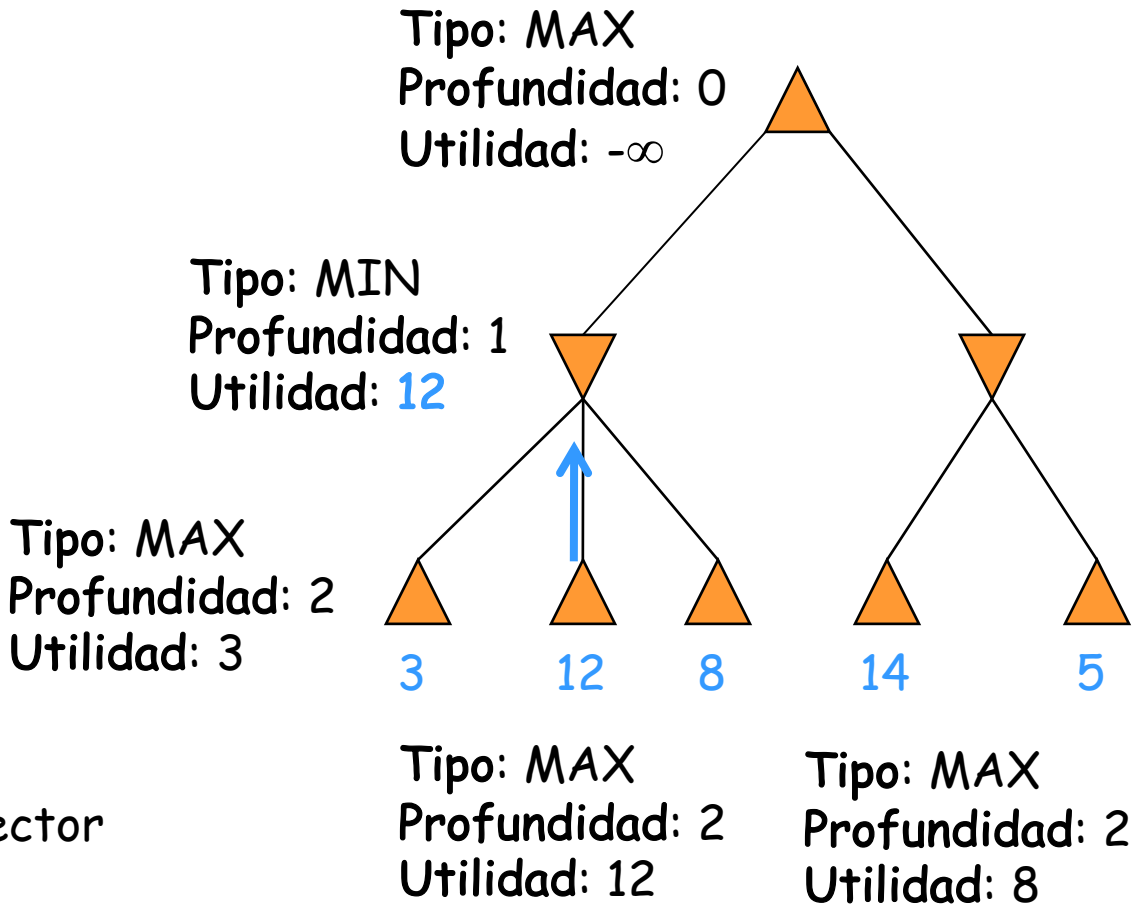
3. Se actualiza la utilidad del padre. Si es MAX se saca el máximo, si es MIN el mínimo

Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 3	



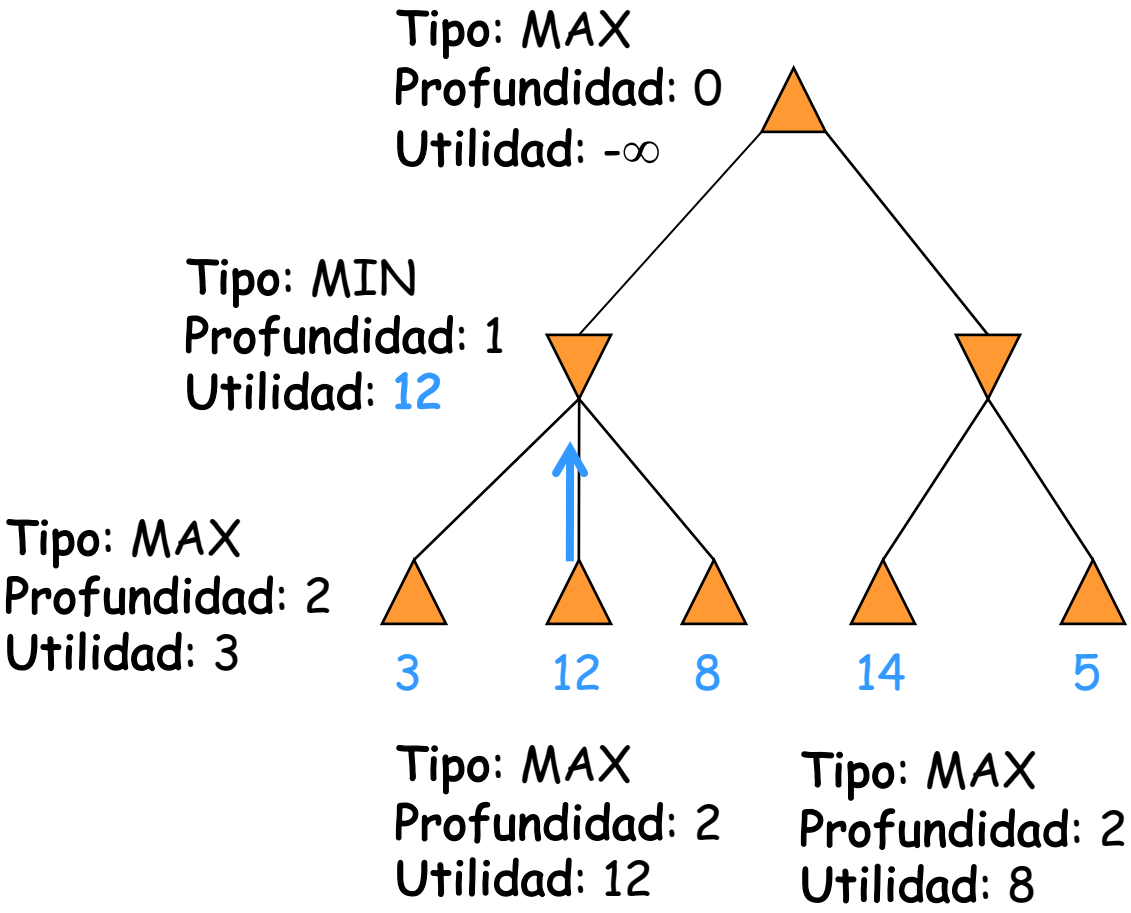
4. Se elimina el nodo del vector

Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 3	



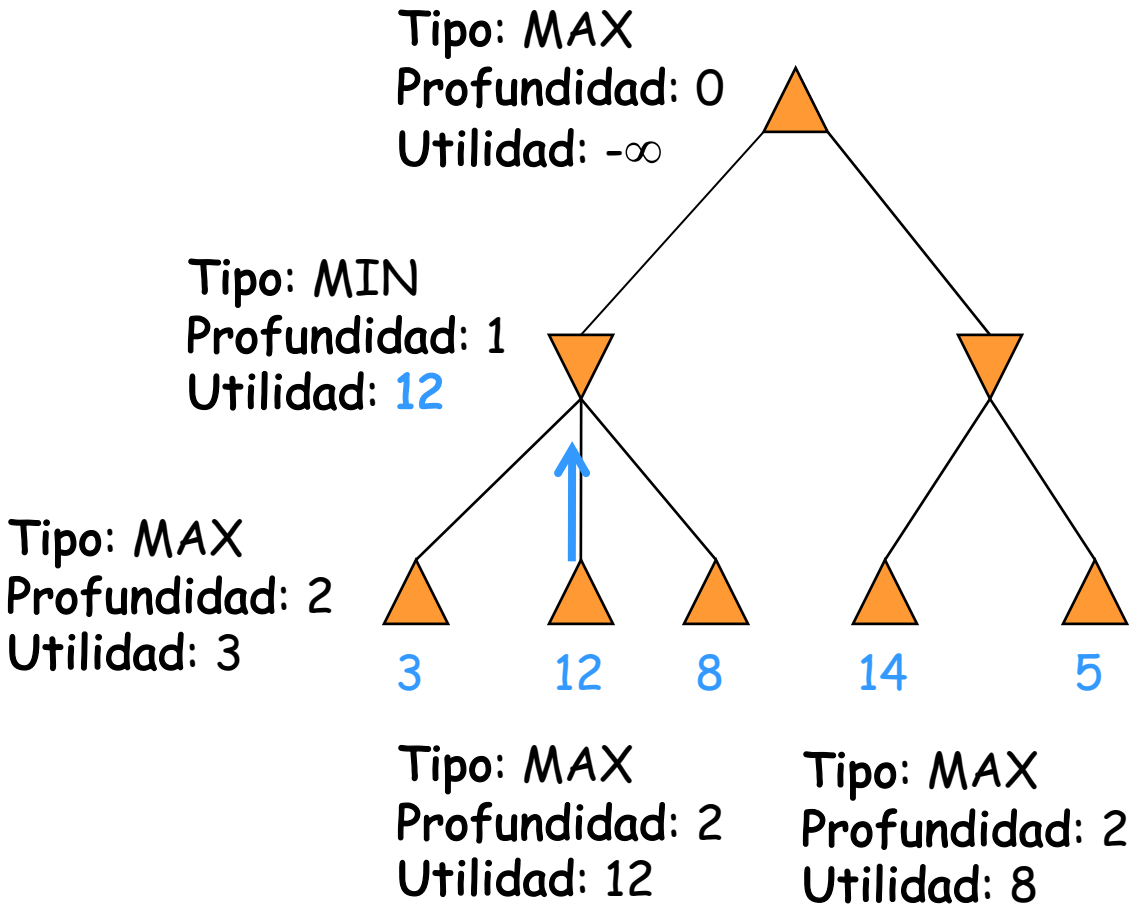
4. Se elimina el nodo del vector

Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 3		

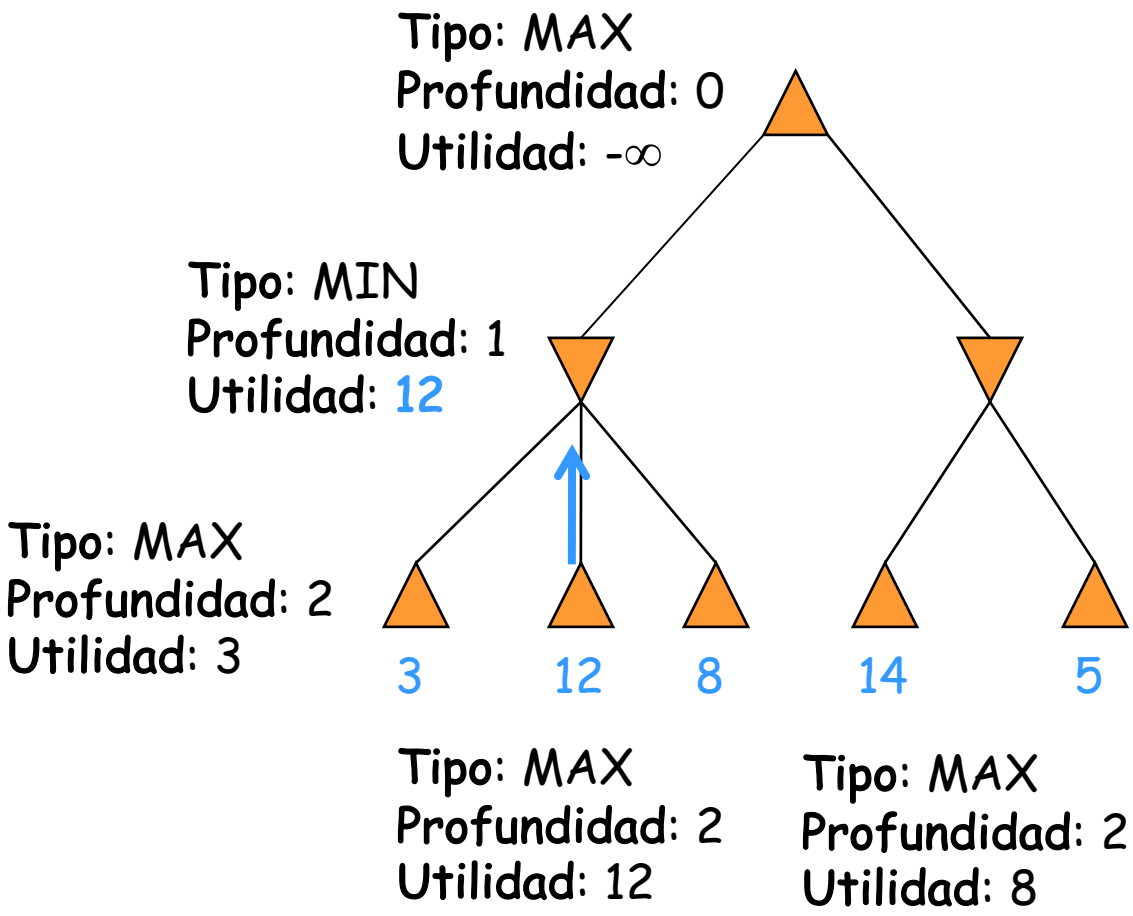


5. Repetir el Paso 1

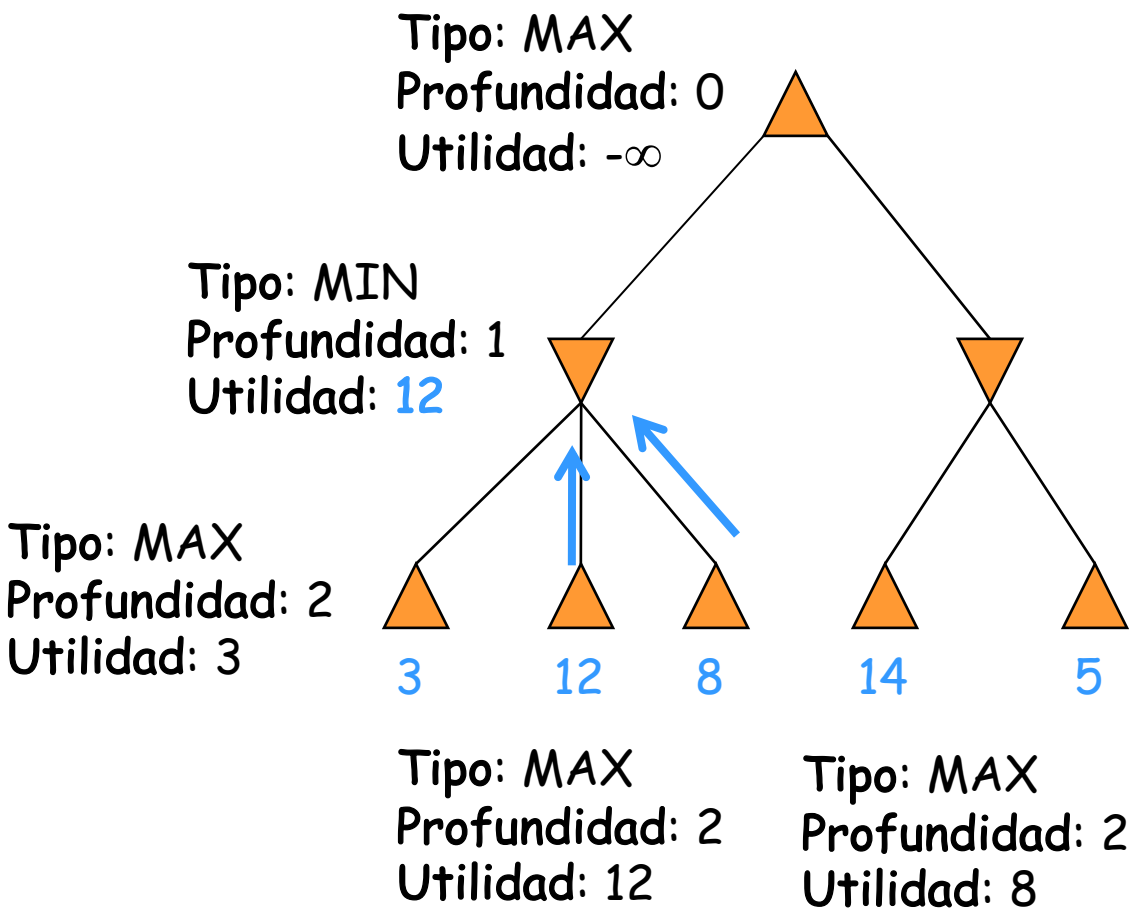
Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 3		



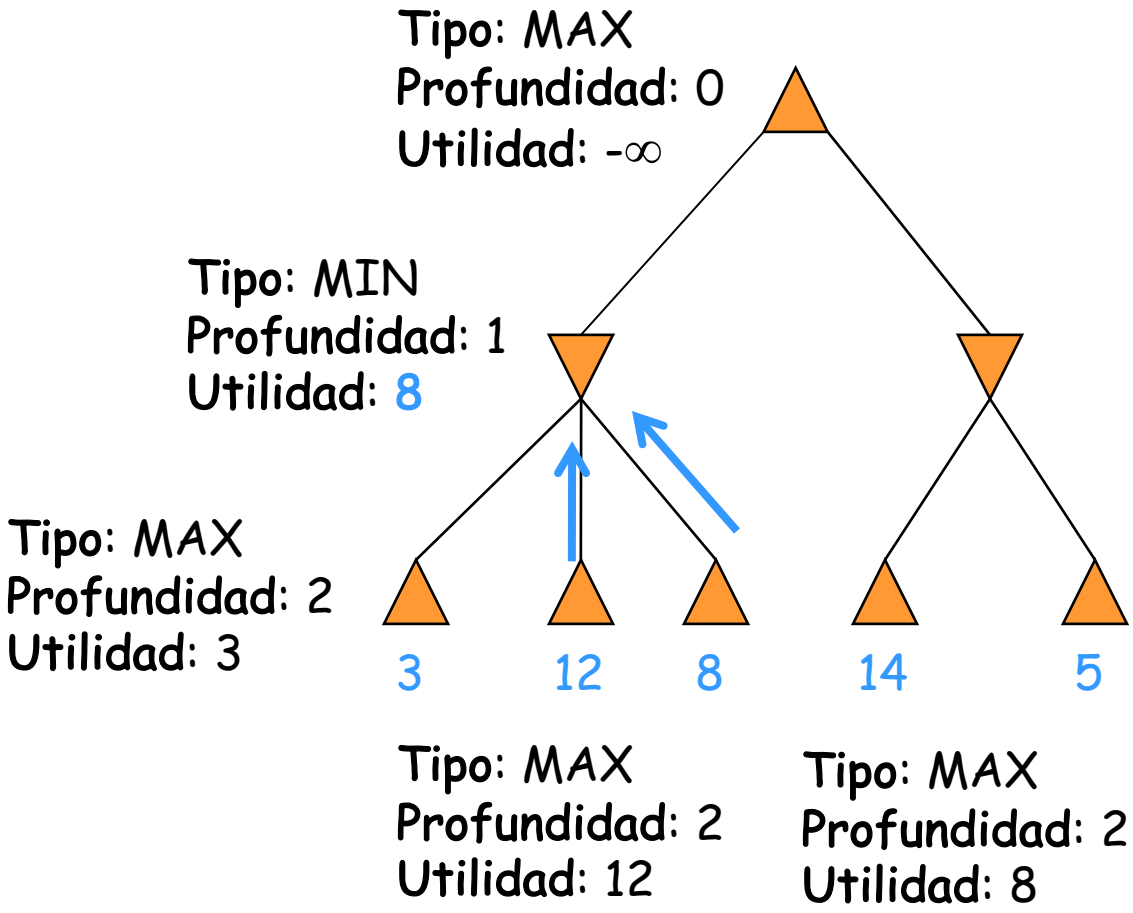
Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 3		



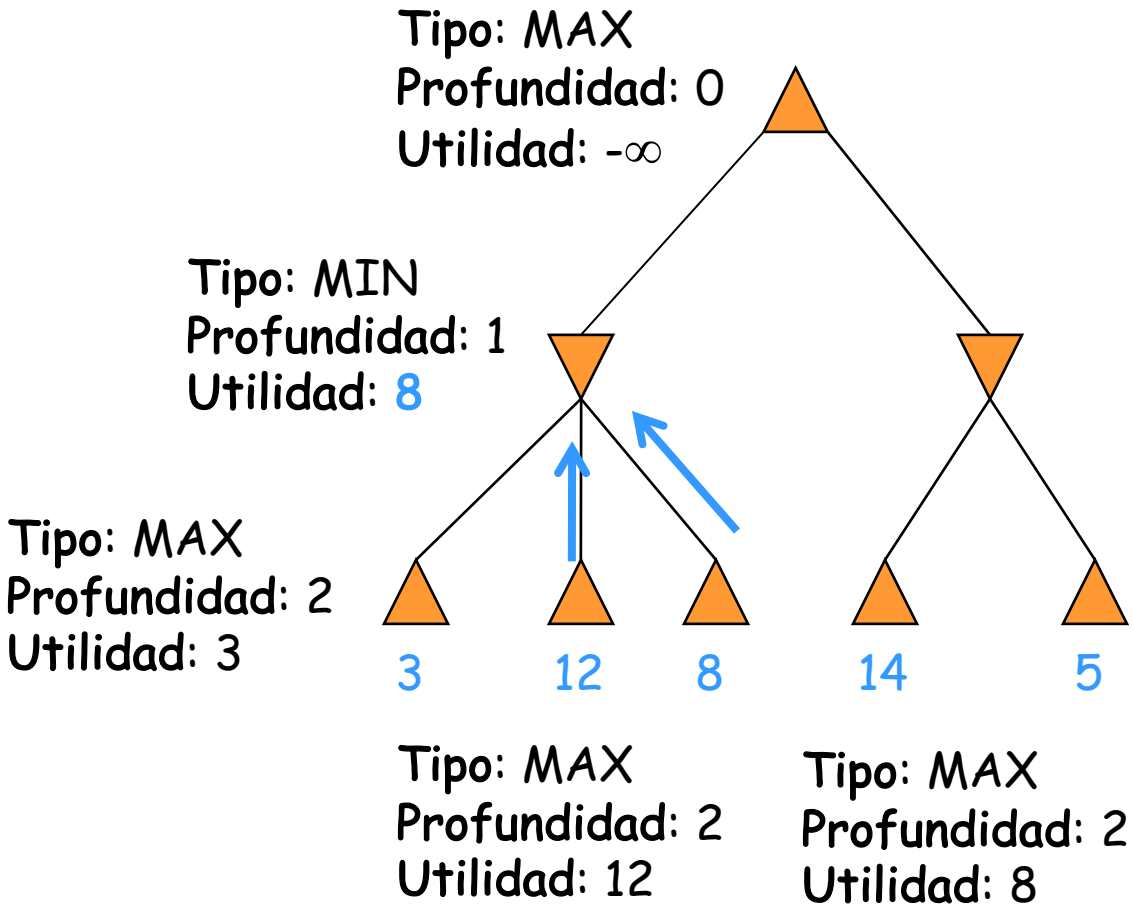
Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 3		



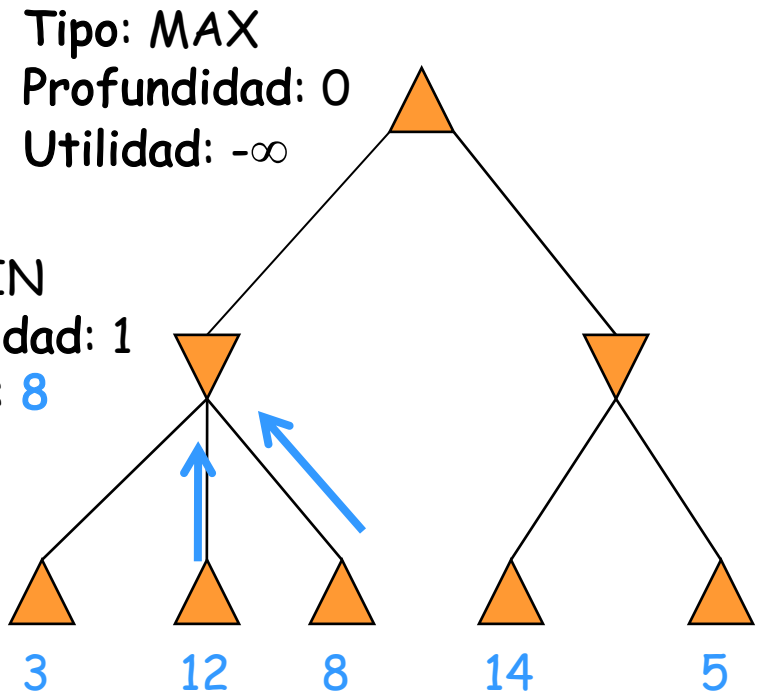
Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: 12	Tipo: MAX Profundidad: 2 Utilidad: 3		



Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MAX Profundidad: 2 Utilidad: 8	Tipo: MIN Profundidad: 1 Utilidad: 8	Tipo: MAX Profundidad: 2 Utilidad: 3		

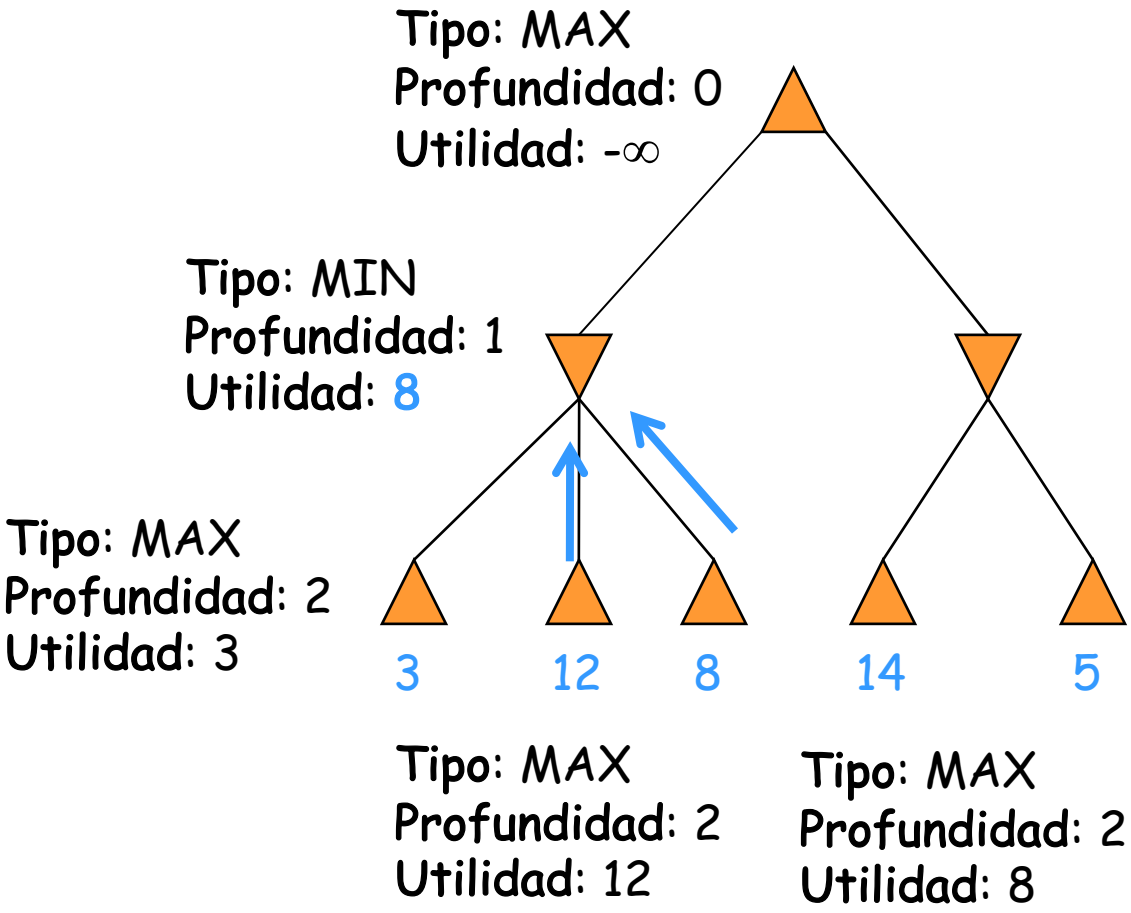


Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MIN Profundidad: 1 Utilidad: 8	Tipo: MAX Profundidad: 2 Utilidad: 3			

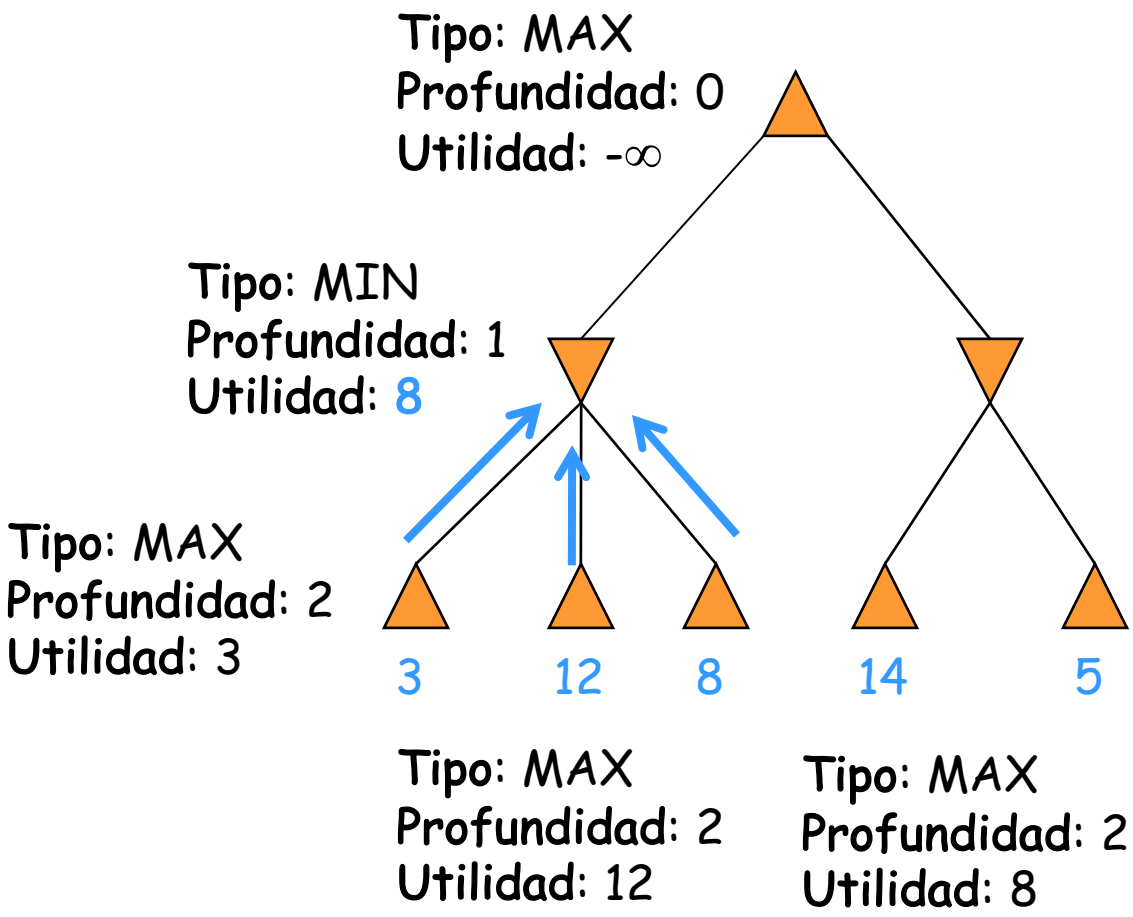


5. Repetir el Paso 1

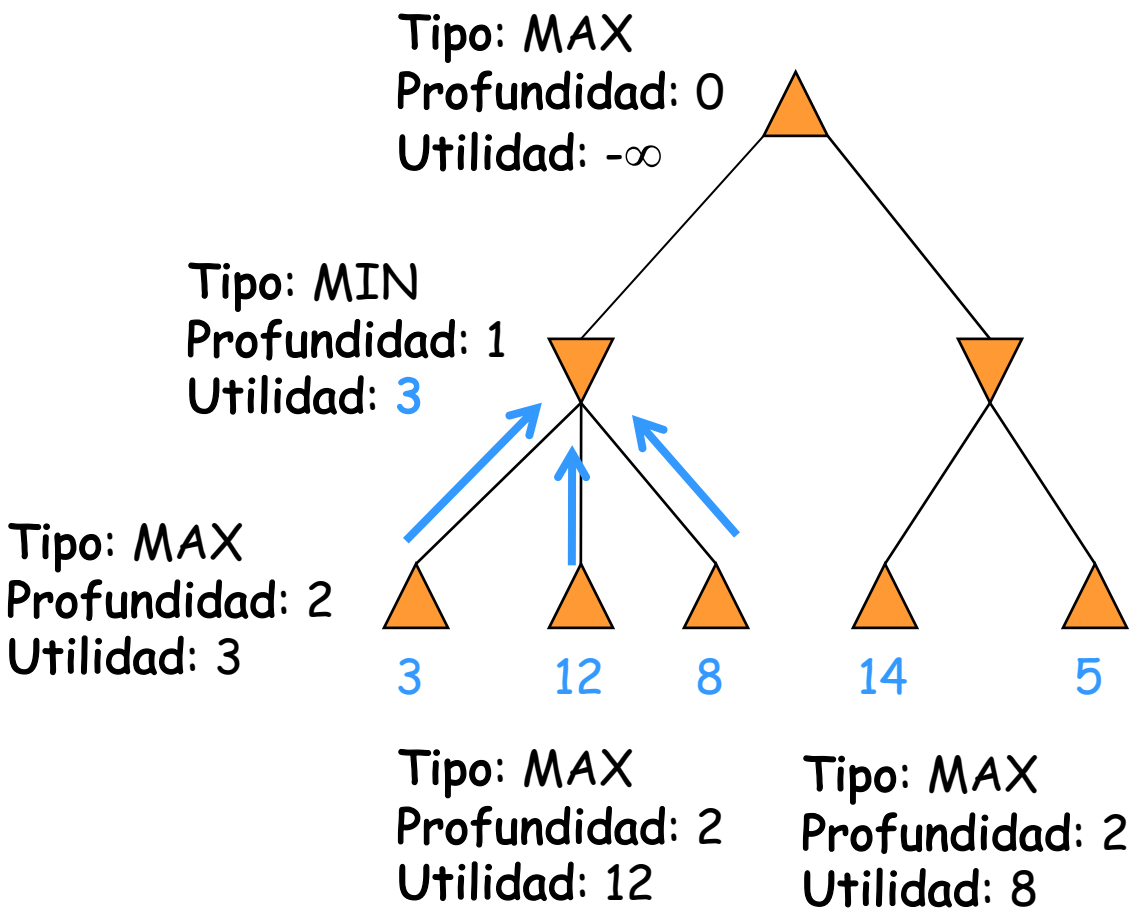
Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MIN Profundidad: 1 Utilidad: 8	Tipo: MAX Profundidad: 2 Utilidad: 3			



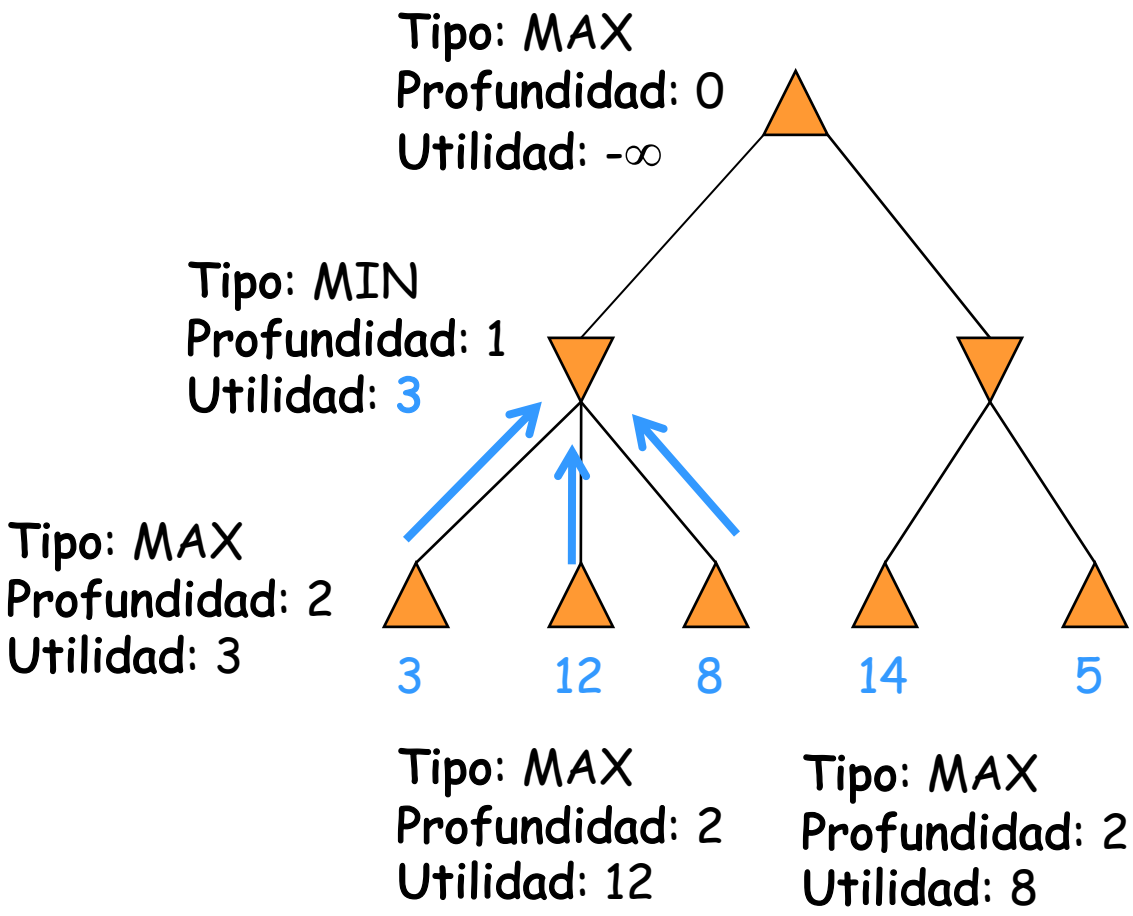
Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MIN Profundidad: 1 Utilidad: 8	Tipo: MAX Profundidad: 2 Utilidad: 3			



Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MIN Profundidad: 1 Utilidad: 8	Tipo: MAX Profundidad: 2 Utilidad: 3			



Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MIN Profundidad: 1 Utilidad: 3	Tipo: MAX Profundidad: 2 Utilidad: 3			



Tipo: MAX Profundidad: 0 Utilidad: -∞	Tipo: MIN Profundidad: 1 Utilidad: 3				

Juegos

Poda alfa-beta

- Según McCarthy, es posible calcular la decisión minimax correcta **sin examinar** todos los nodos del árbol
- Podar el árbol minimax

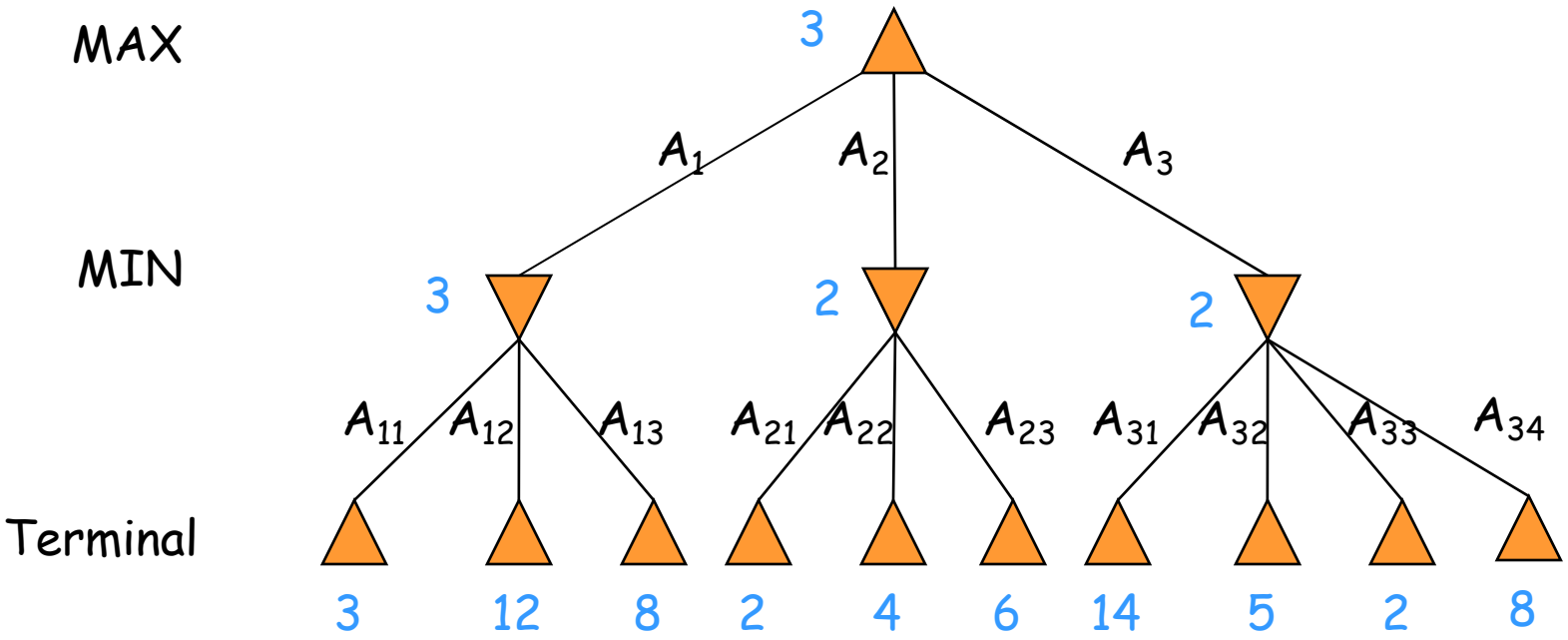
John McCarthy (1927 - 2011)

Juegos

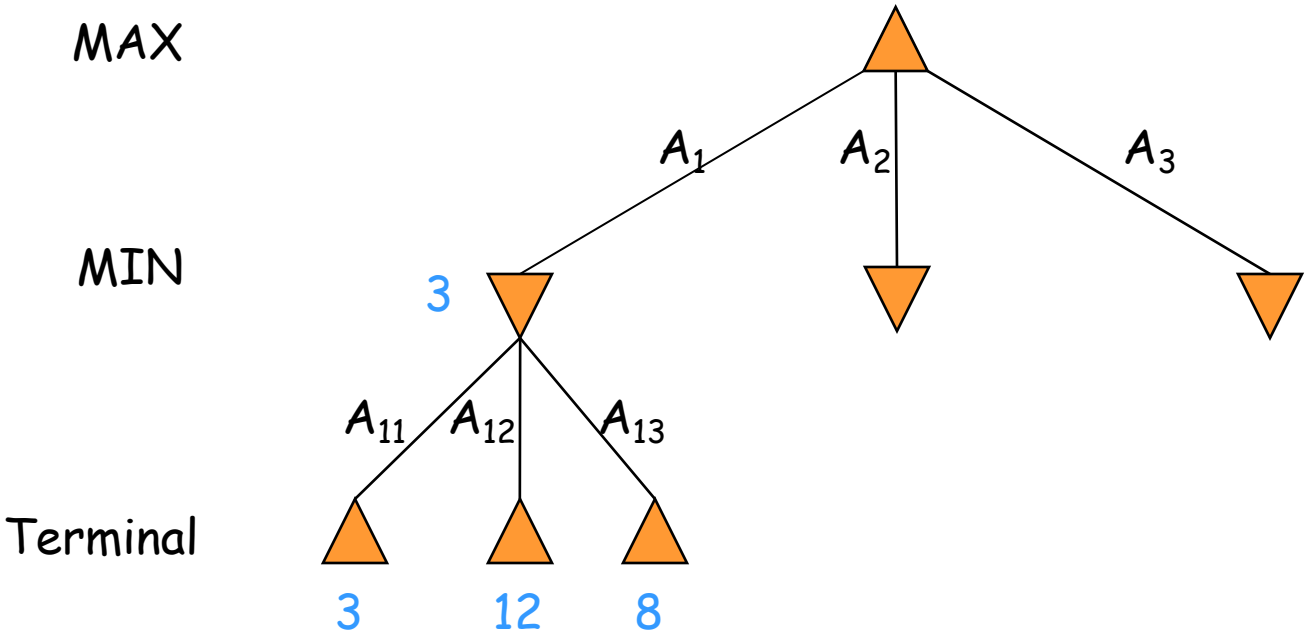
Poda alfa-beta

- Aplicada a un árbol minimax, produce la misma jugada que se obtendría sin ella

Juegos

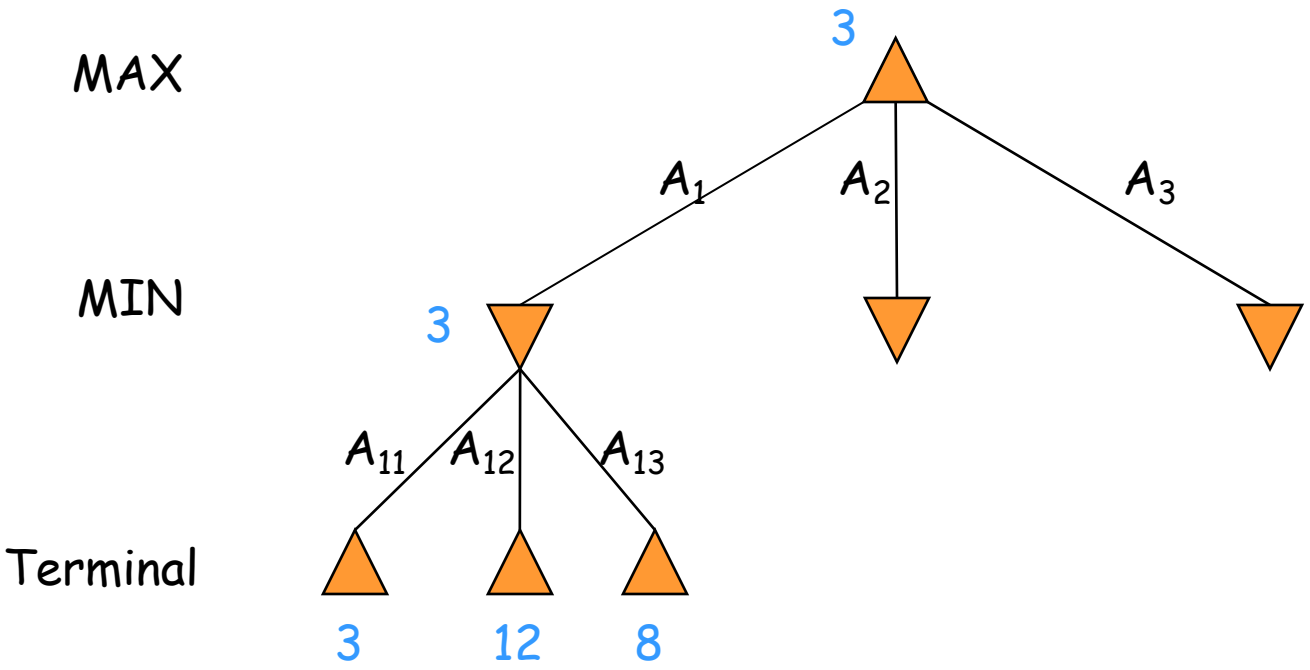

MAX
MIN
Terminal
3
A1
A2
A3
3
2
2
A11
A12
A13
A21
A22
A23
A31
A32
A33
A34
3
12
8
2
4
6
14
5
2
8

Juegos

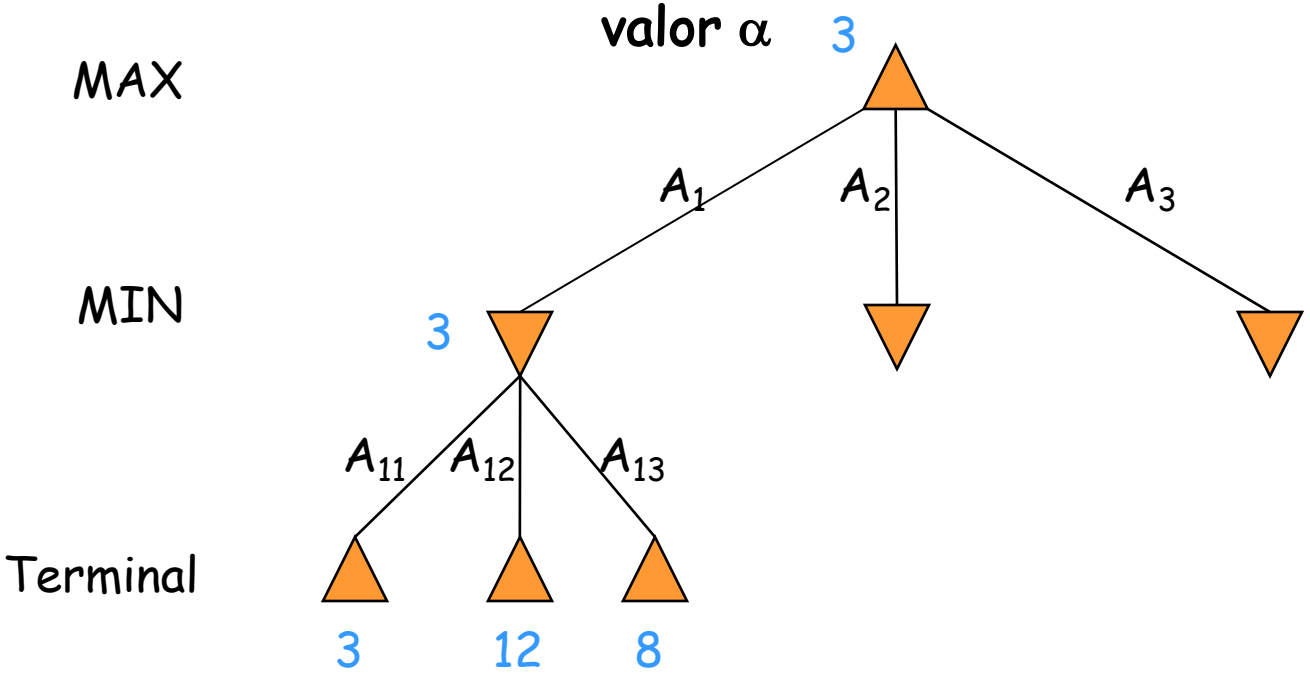


Se calcula la utilidad para A_1

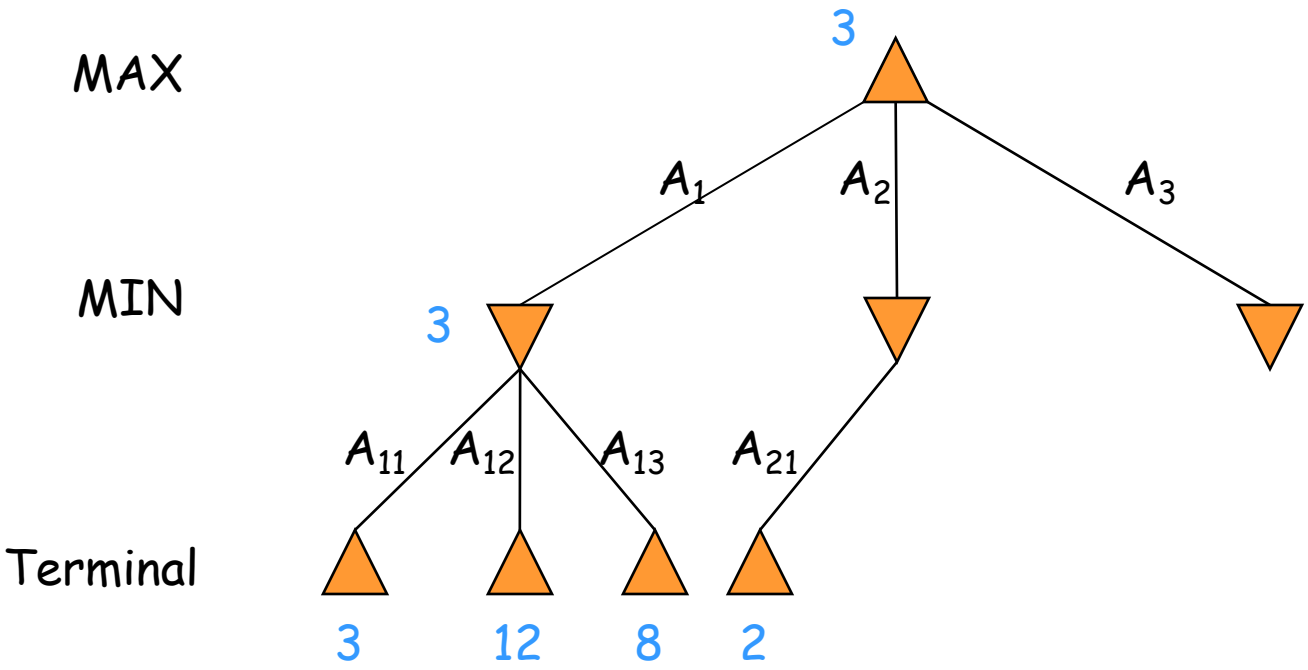
Juegos



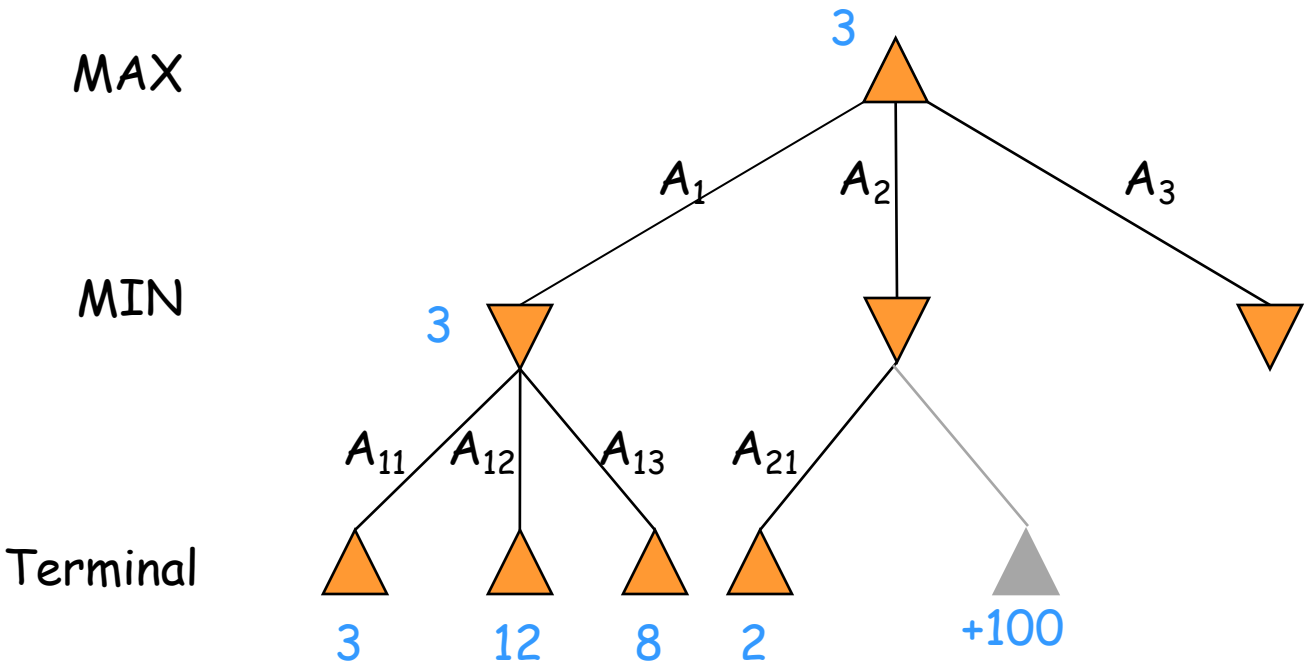
Juegos



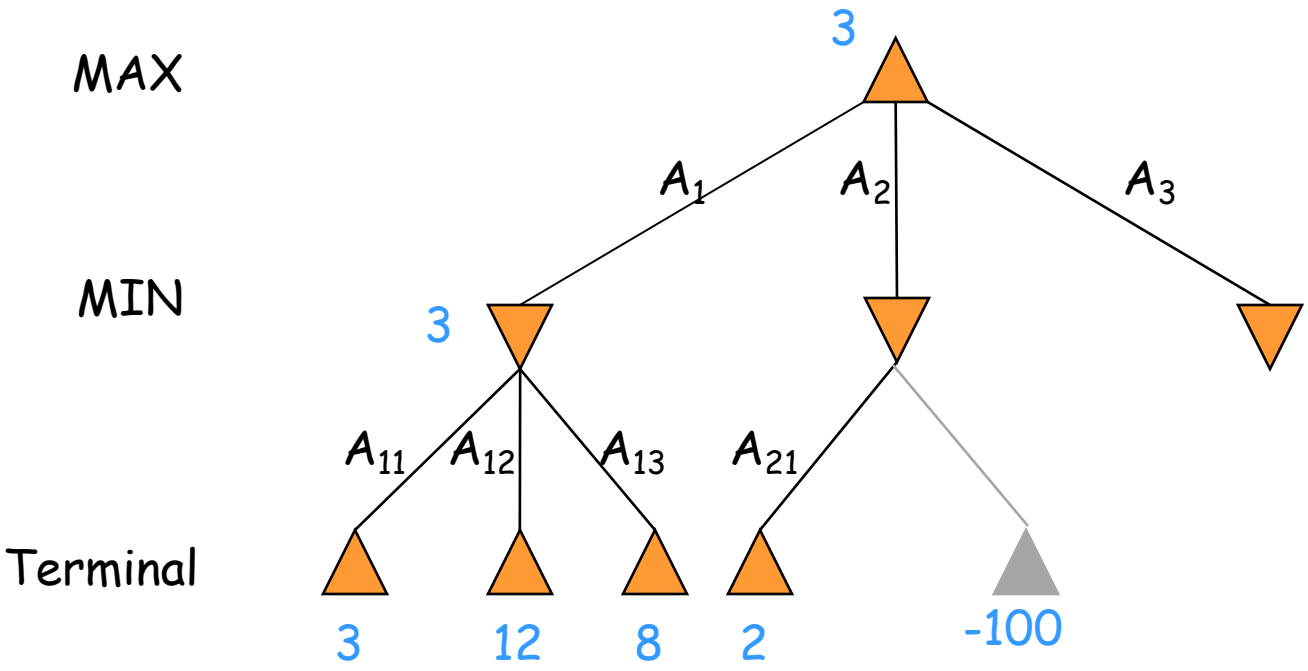
Juegos



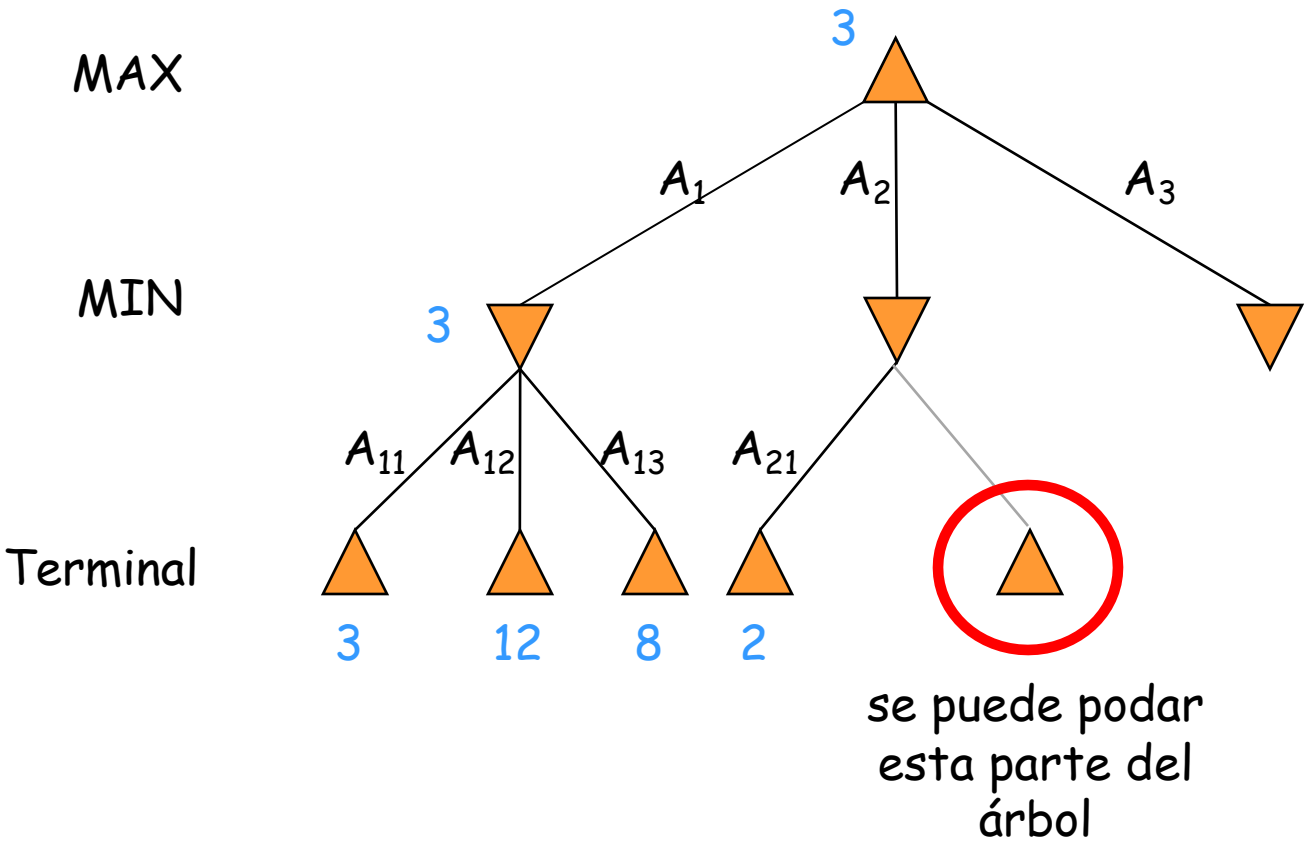
Juegos



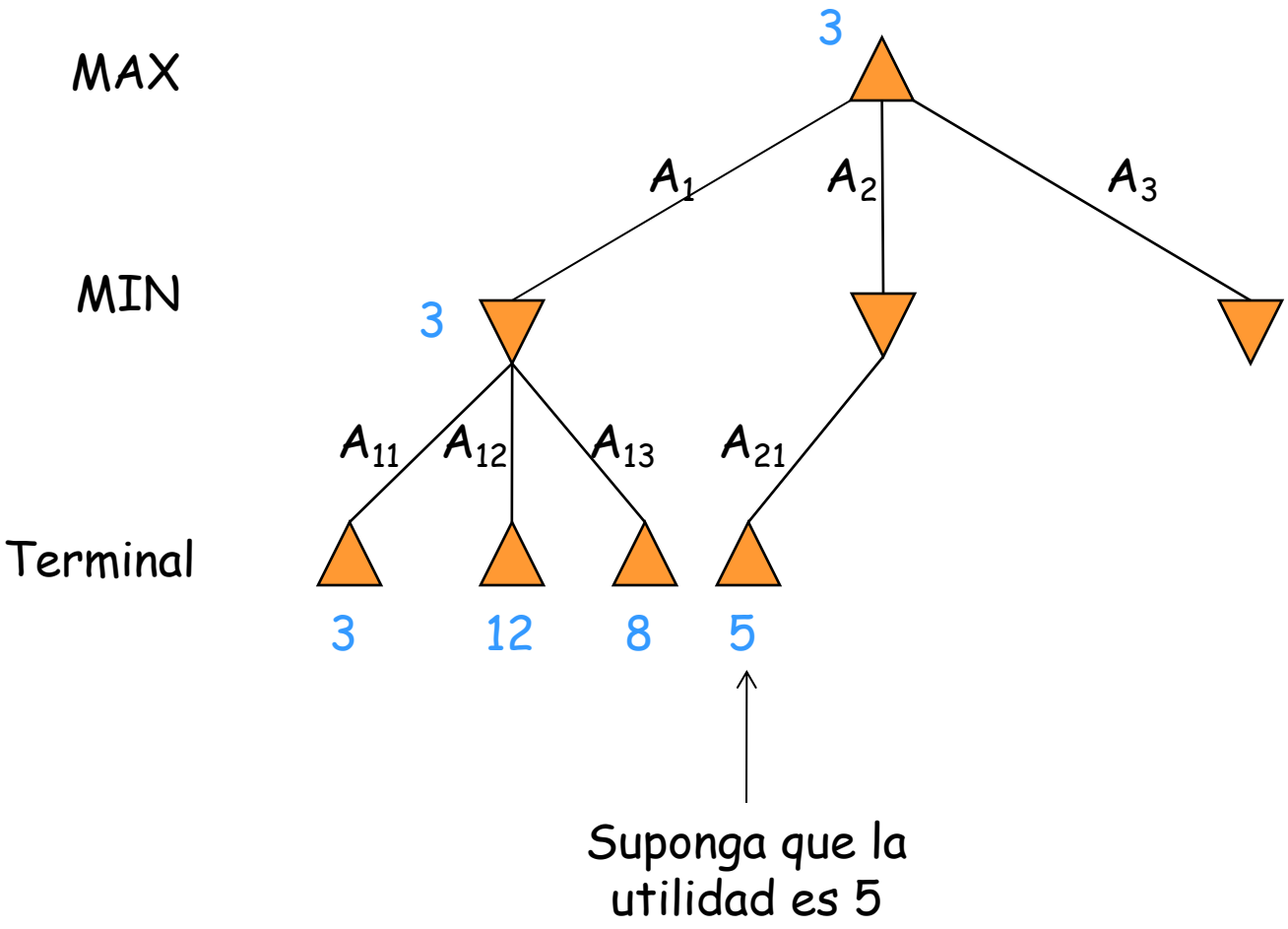
Juegos



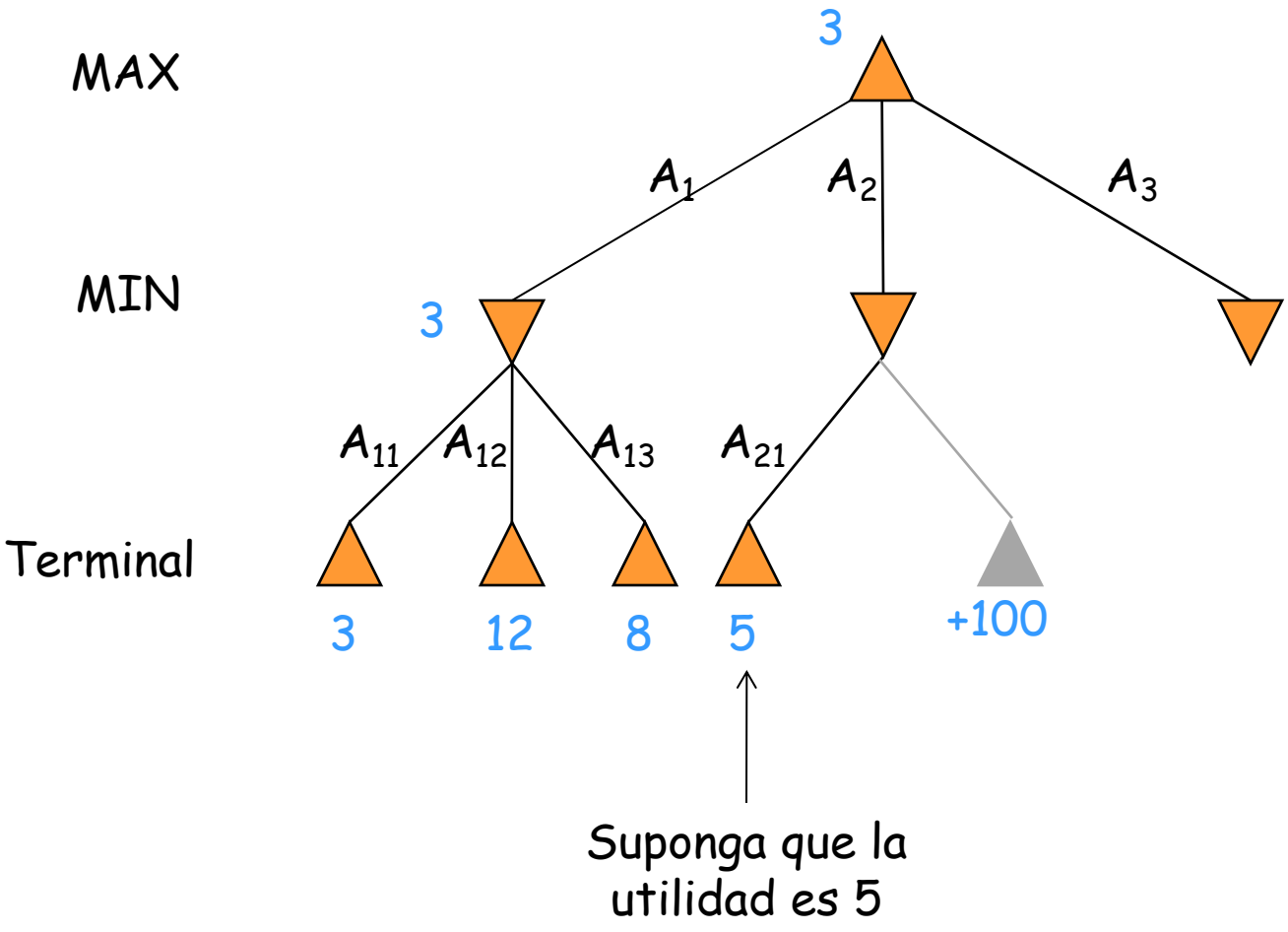
Juegos


MAX
MIN
Terminal
3
A_1
A_2
A_3
3
A_{11}
A_{12}
A_{13}
A_{21}
3
12
8
2
se puede podar esta parte del árbol

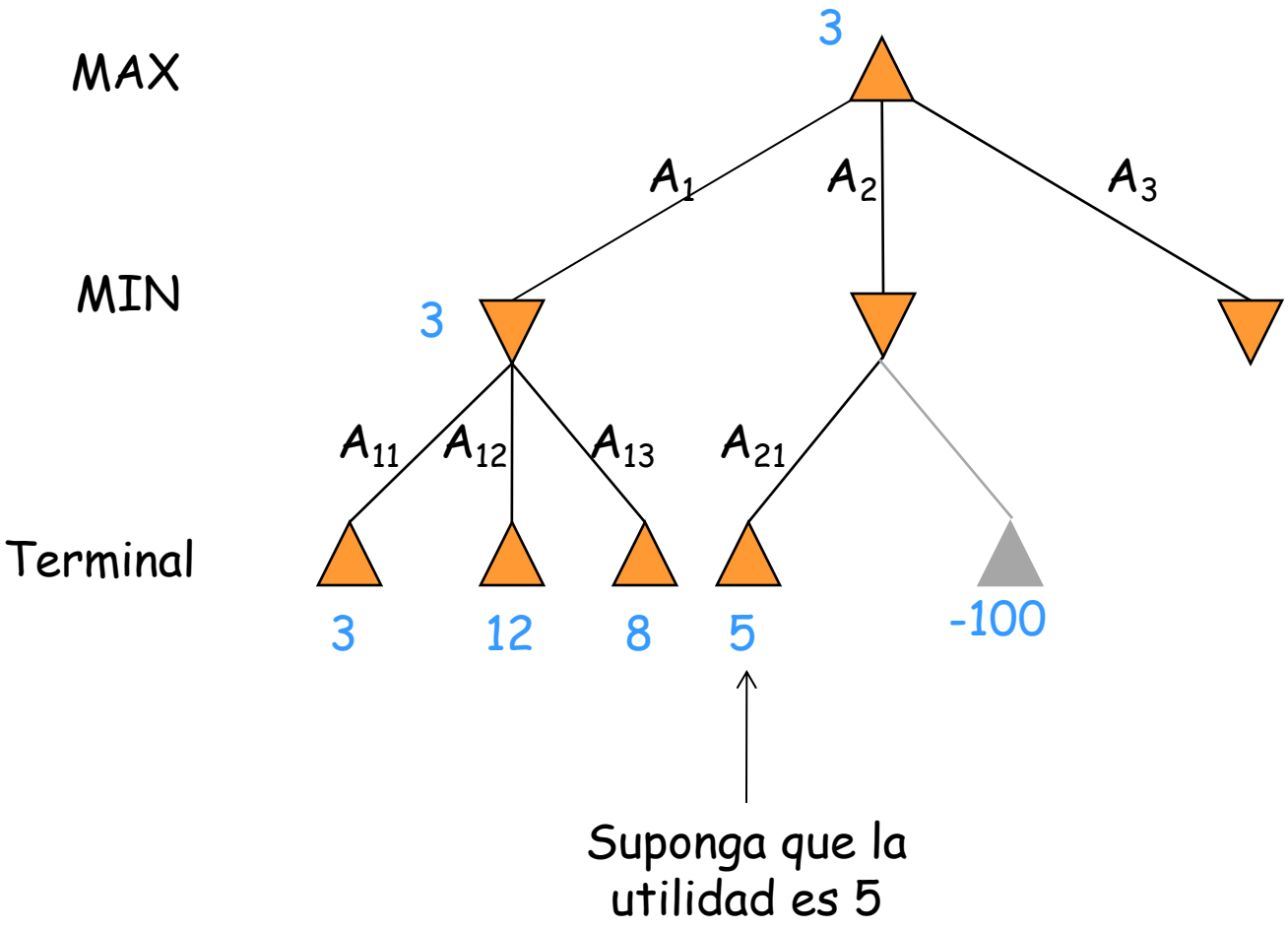
Juegos


MAX
MIN
Terminal
3
A1
A2
A3
3
A11
A12
A13
A21
3
12
8
5
Suponga que la utilidad es 5

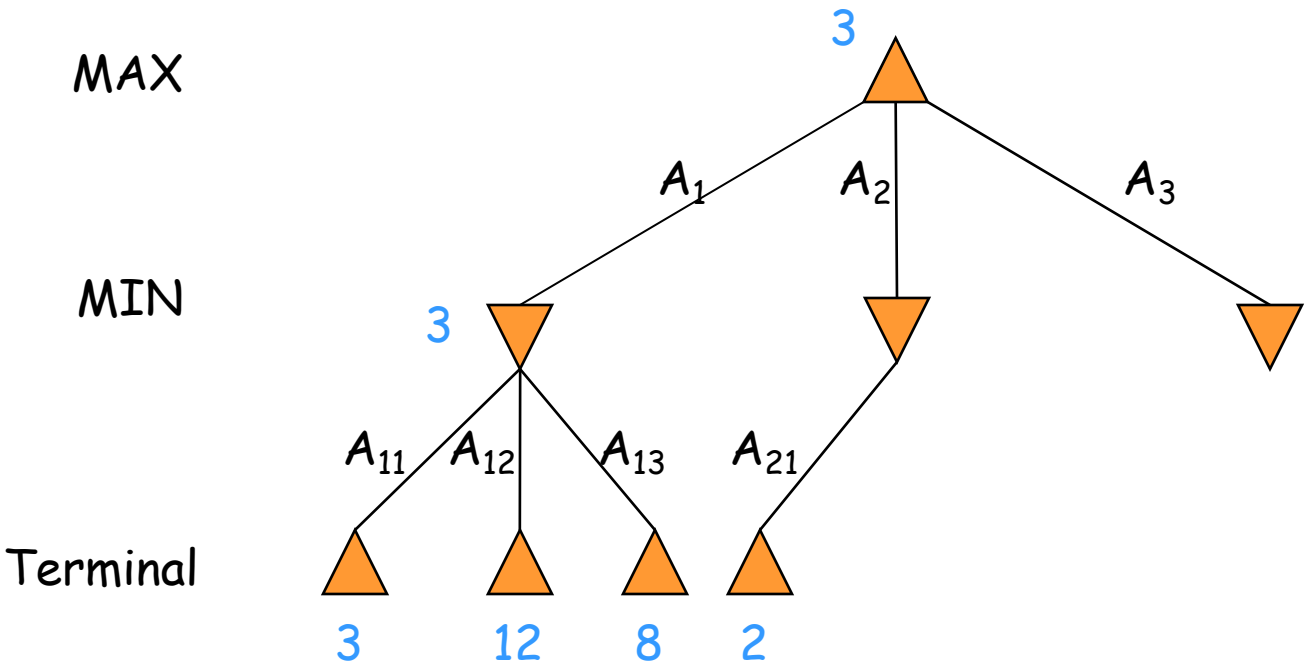
Juegos


MAX
MIN
Terminal
3
A1
A2
A3
3
A11
A12
A13
A21
3
12
8
5
+100
Suponga que la utilidad es 5

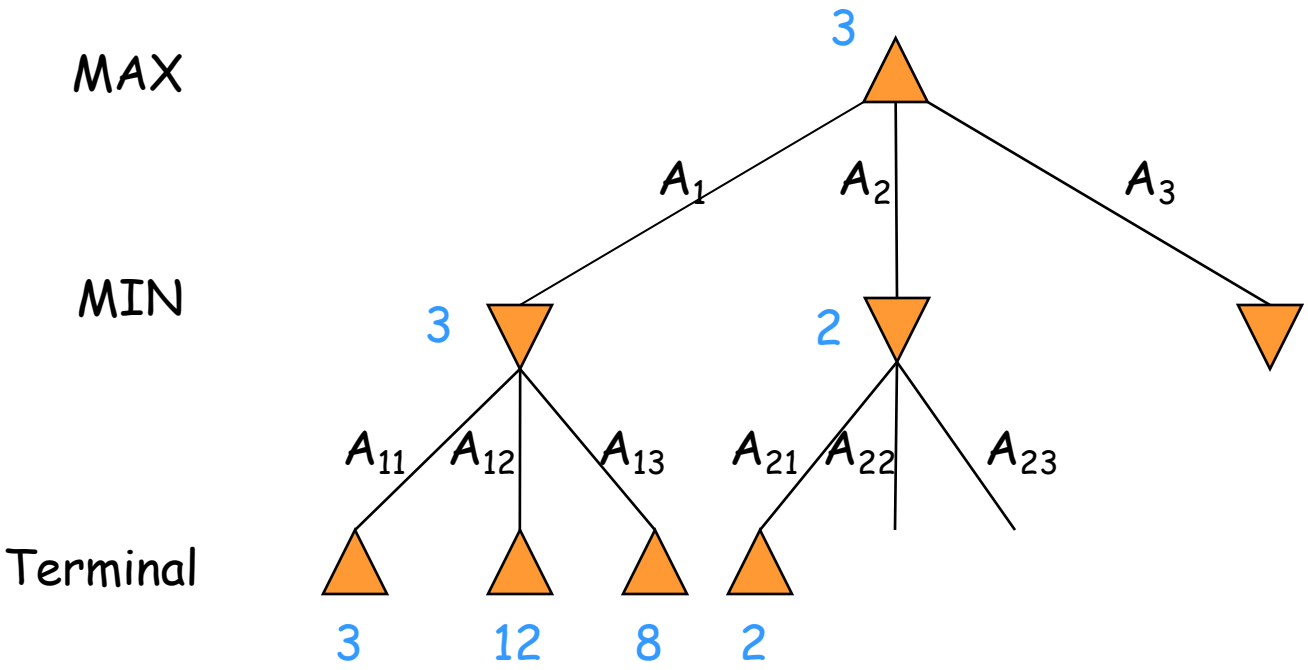
Juegos


MAX
MIN
Terminal
3
A1
A2
A3
3
A11
A12
A13
A21
3
12
8
5
-100
Suponga que la utilidad es 5

Juegos

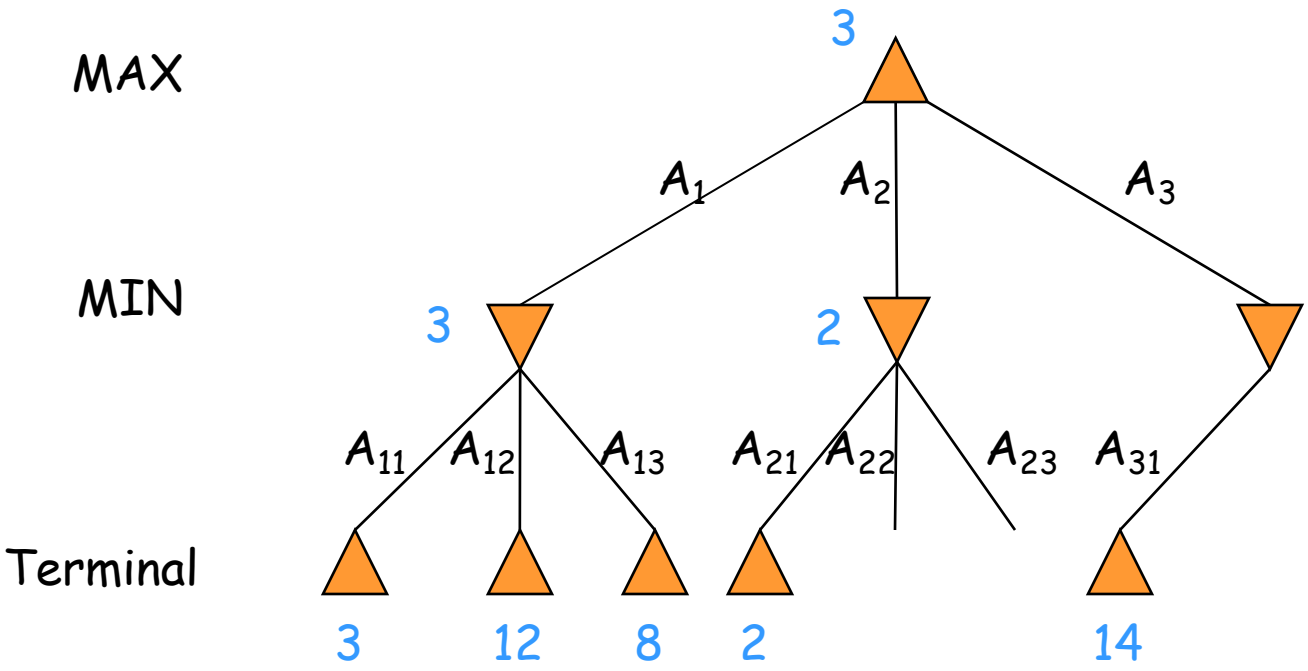


Juegos

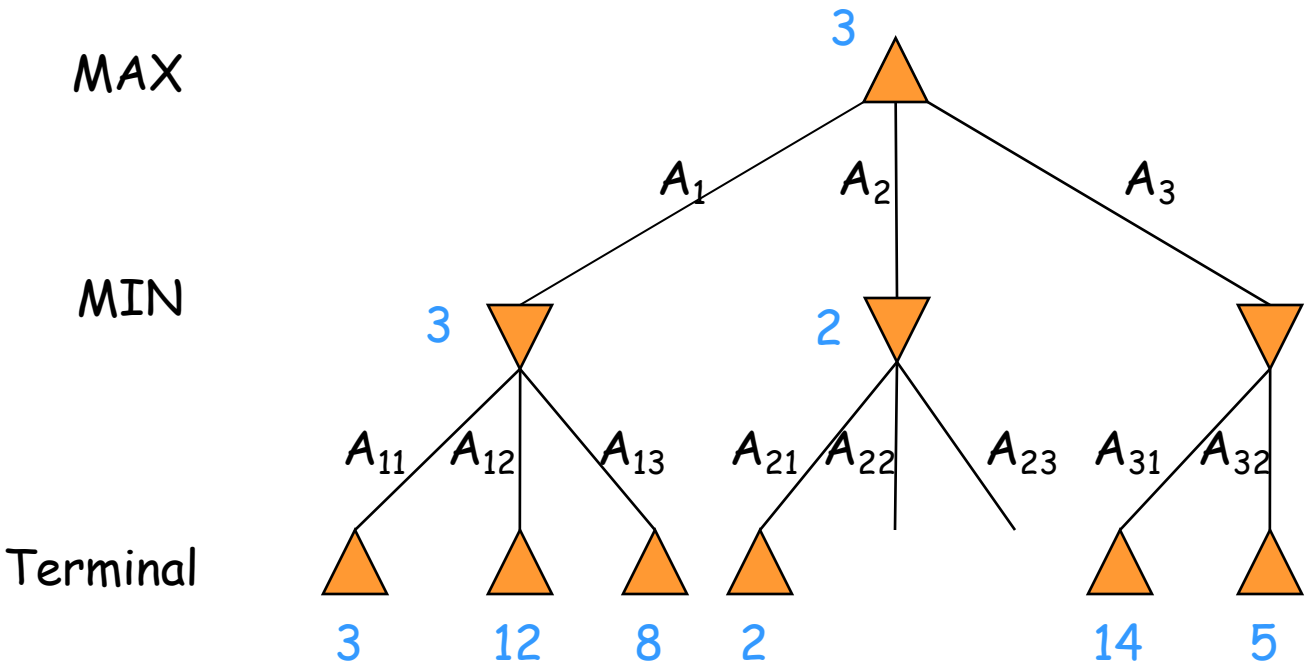


Como MIN va a escoger el menor entre sus hijos, se encontró un nodo con valor menor que 3 y MAX va a escoger el valor máximo, no se necesita explorar A_{22} ni A_{23}

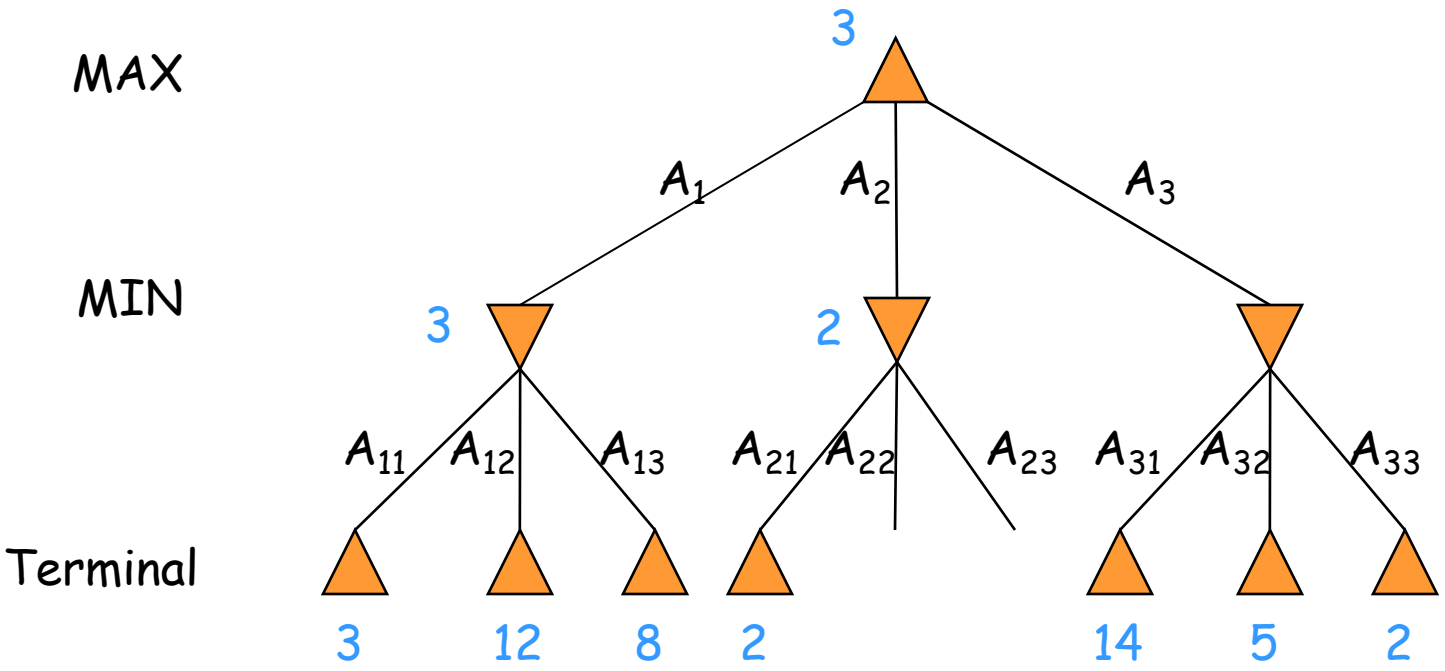
Juegos


MAX
MIN
Terminal
3
A1
A2
A3
3
2
A11
A12
A13
A21
A22
A23
A31
3
12
8
2
14

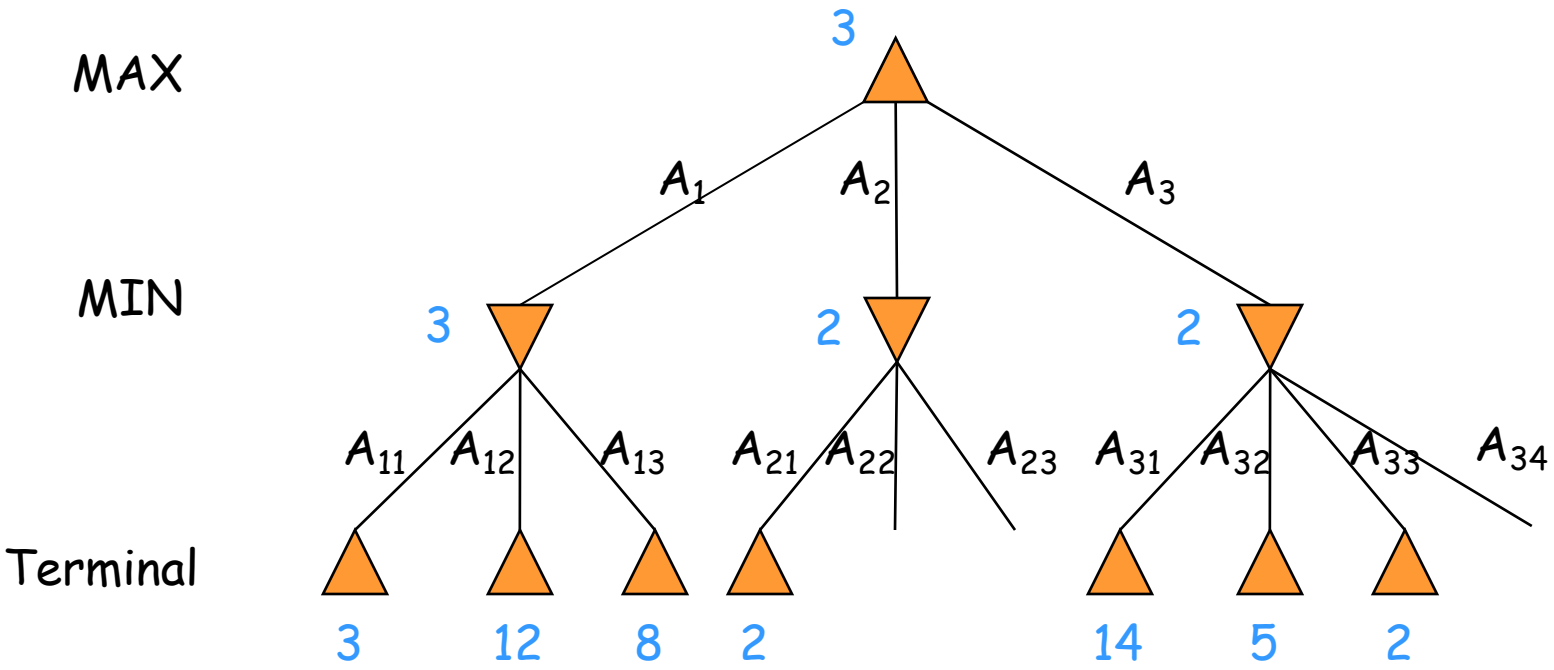
Juegos


MAX
MIN
Terminal
3
A1
A2
A3
3
2
A11
A12
A13
A21
A22
A23
A31
A32
3
12
8
2
14
5

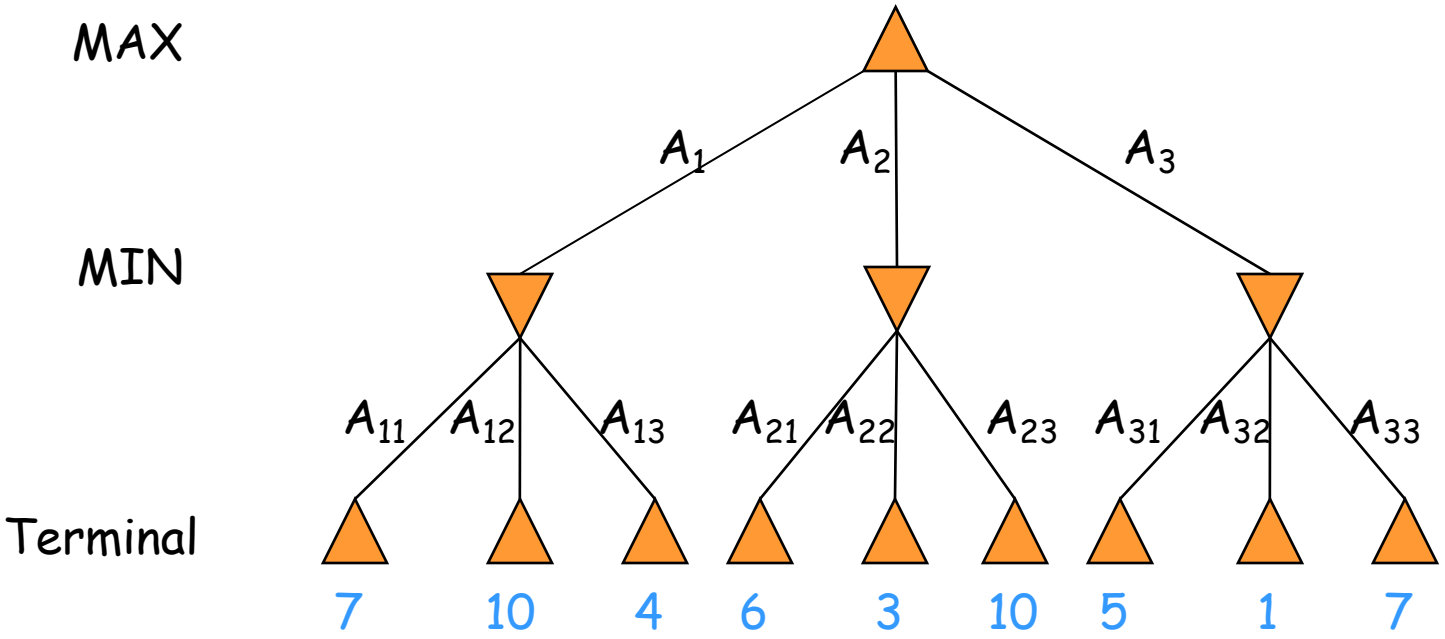
Juegos


MAX
MIN
Terminal
3
A1
A2
A3
3
2
A11
A12
A13
A21
A22
A23
A31
A32
A33
3
12
8
2
14
5
2

Juegos

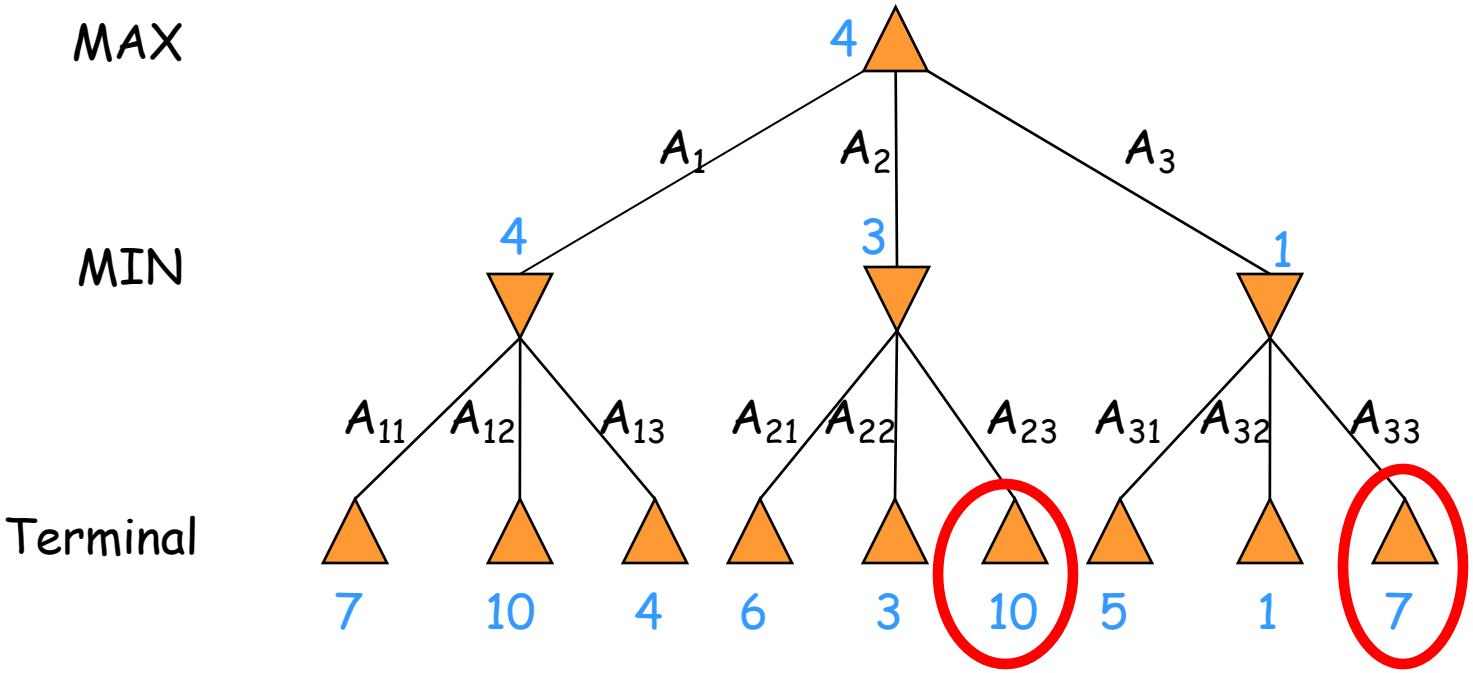

MAX
MIN
Terminal
3
A1
A2
A3
3
2
2
A11
A12
A13
A21
A22
A23
A31
A32
A33
A34
3
12
8
2
14
5
2

Juegos

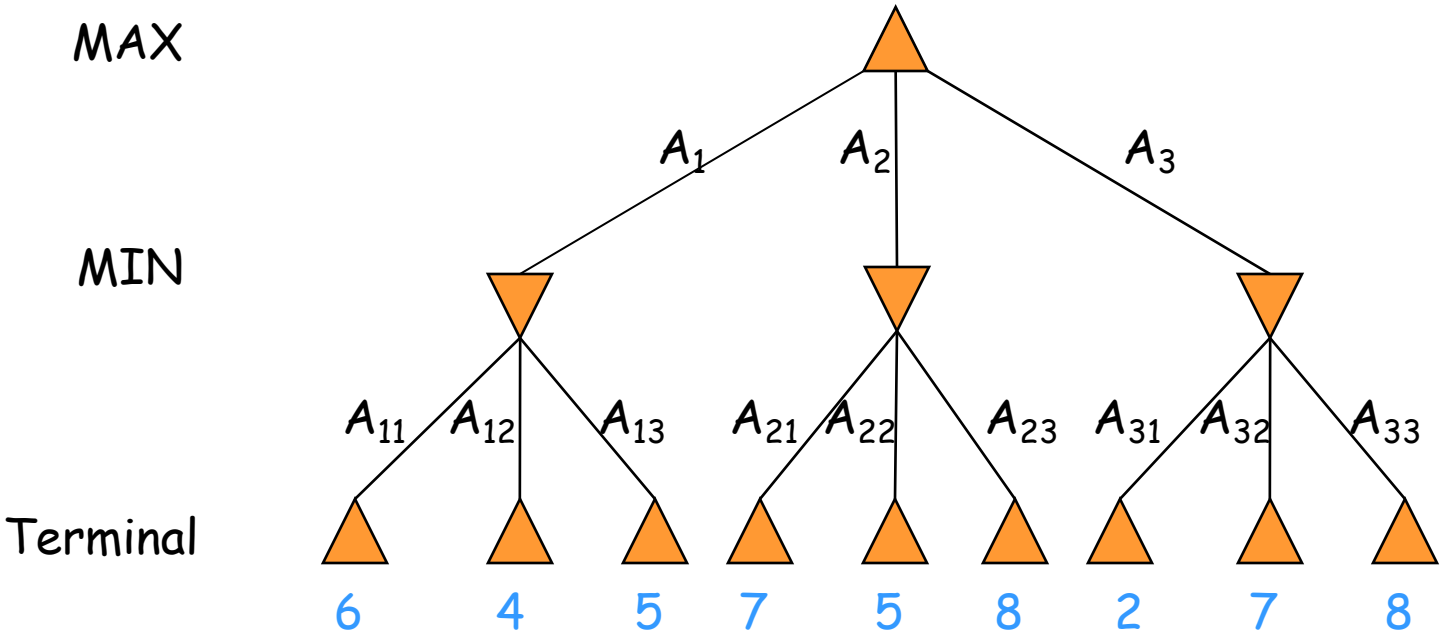


Indique qué nodos se podan

Juegos

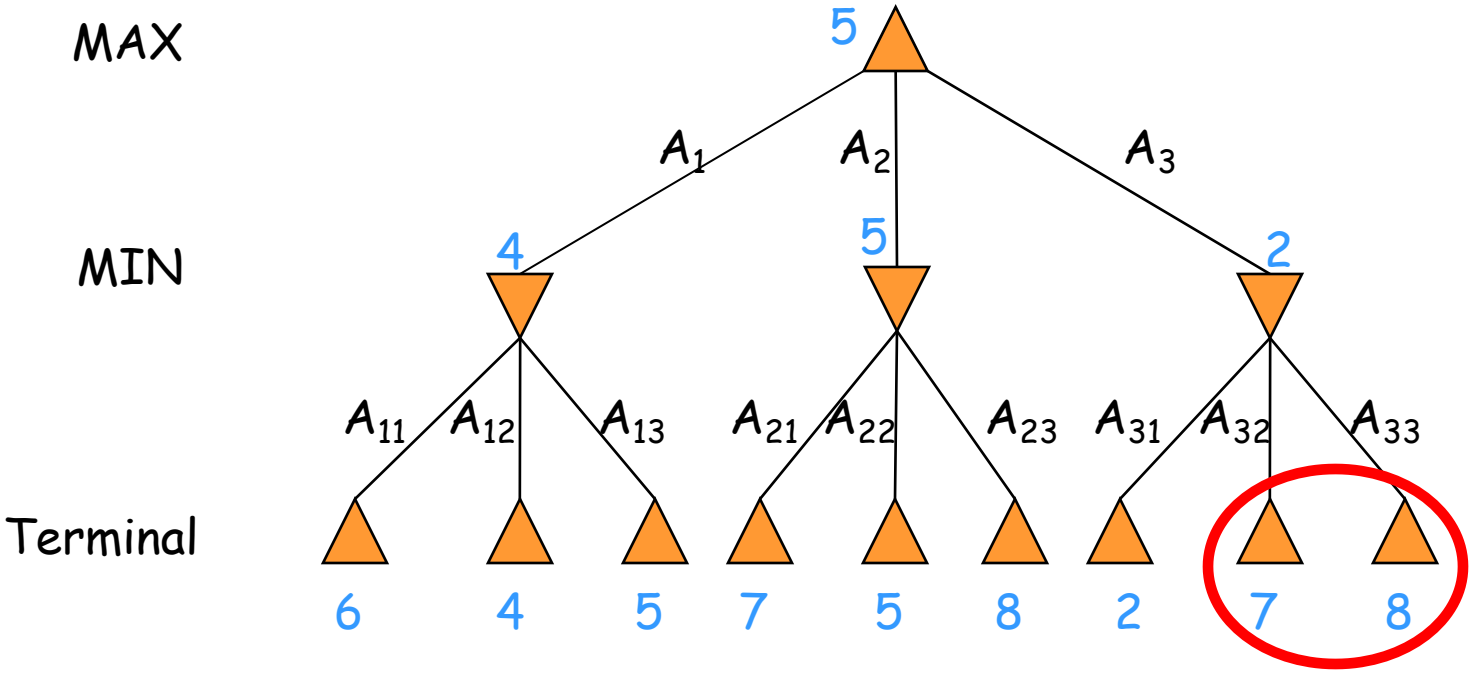

MAX
MIN
Terminal
4
A1
A2
A3
4
3
1
A11
A12
A13
A21
A22
A23
A31
A32
A33
7
10
4
6
3
10
5
1
7

Juegos

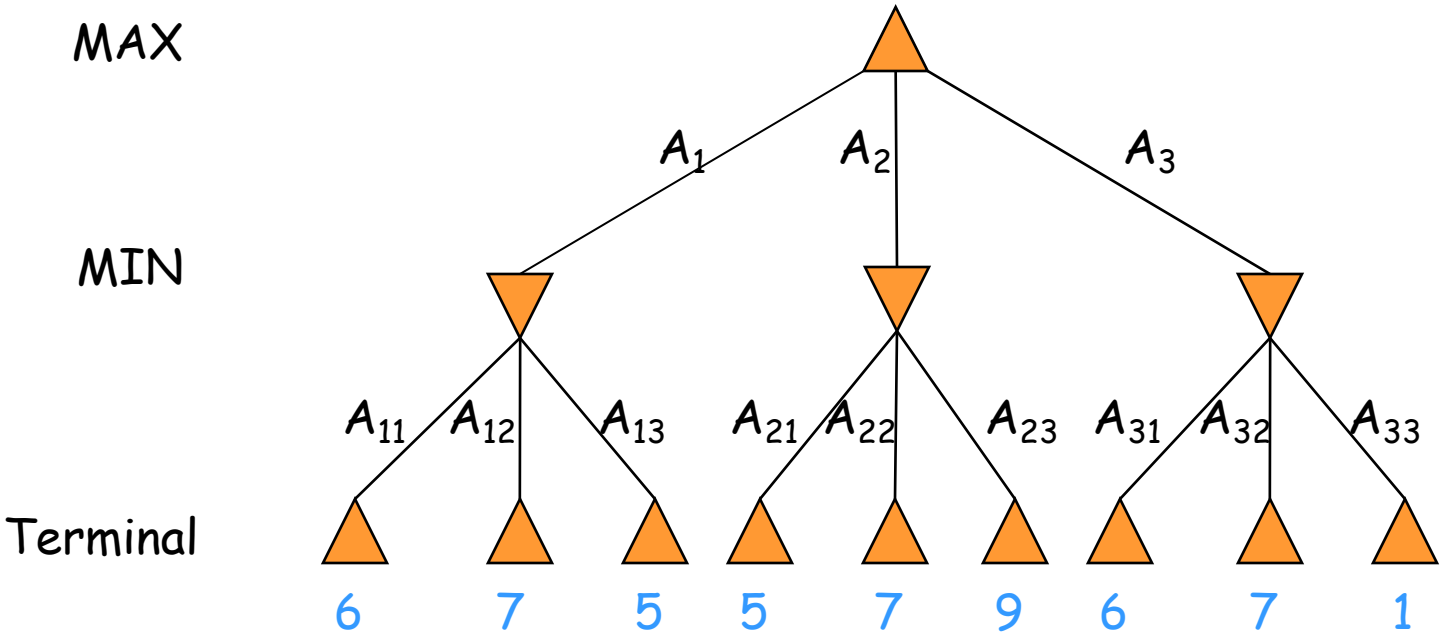


Indique qué nodos se podan

Juegos

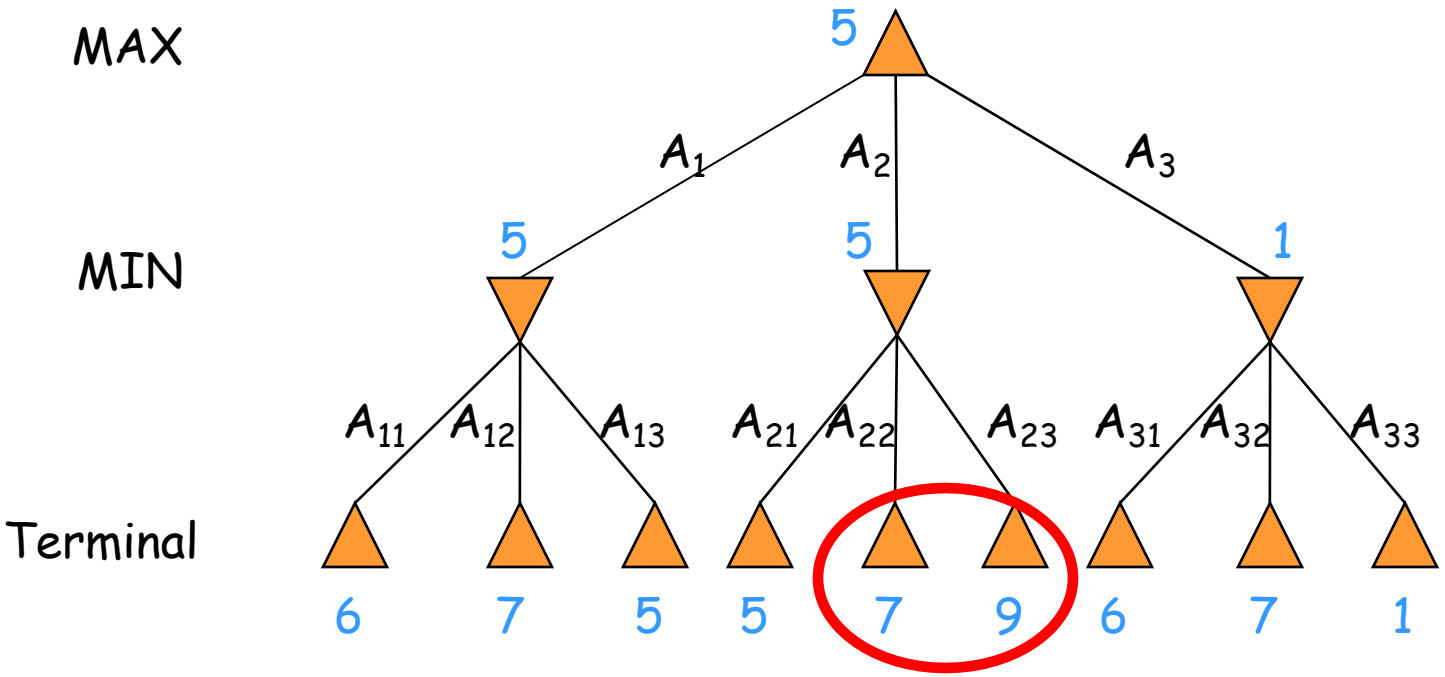

MAX
MIN
Terminal
5
A1
A2
A3
4
5
2
A11
A12
A13
A21
A22
A23
A31
A32
A33
6
4
5
7
5
8
2
7
8

Juegos

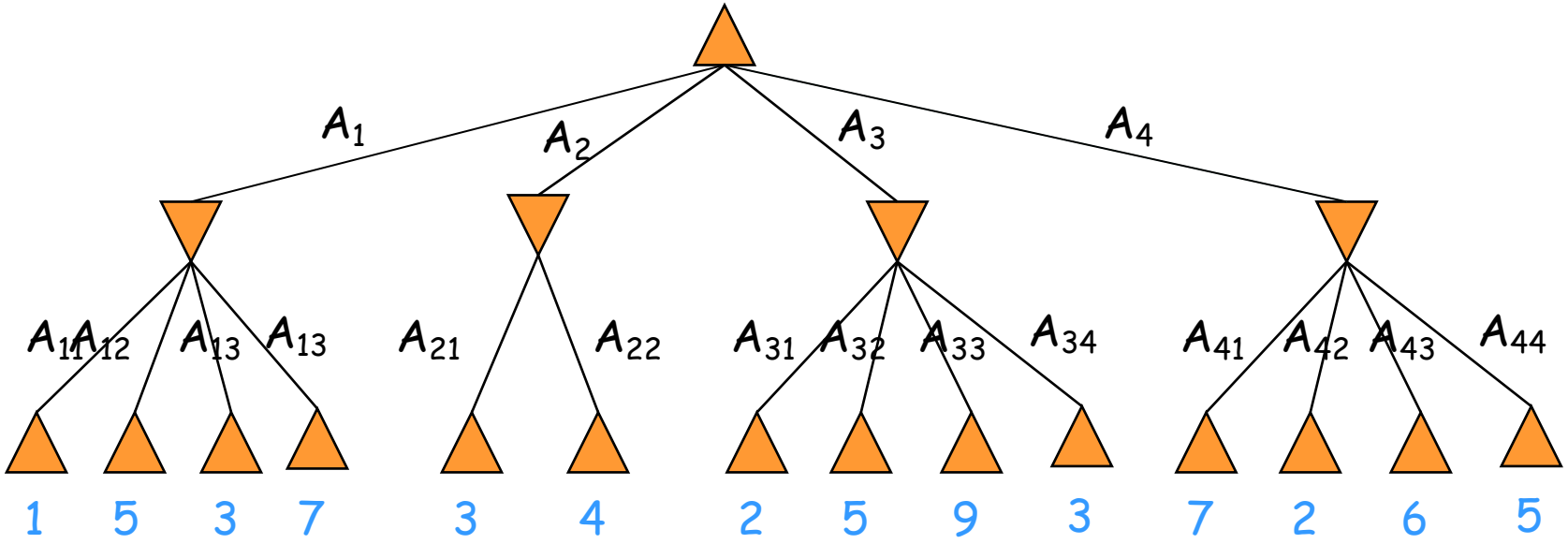


Indique qué nodos se podan

Juegos

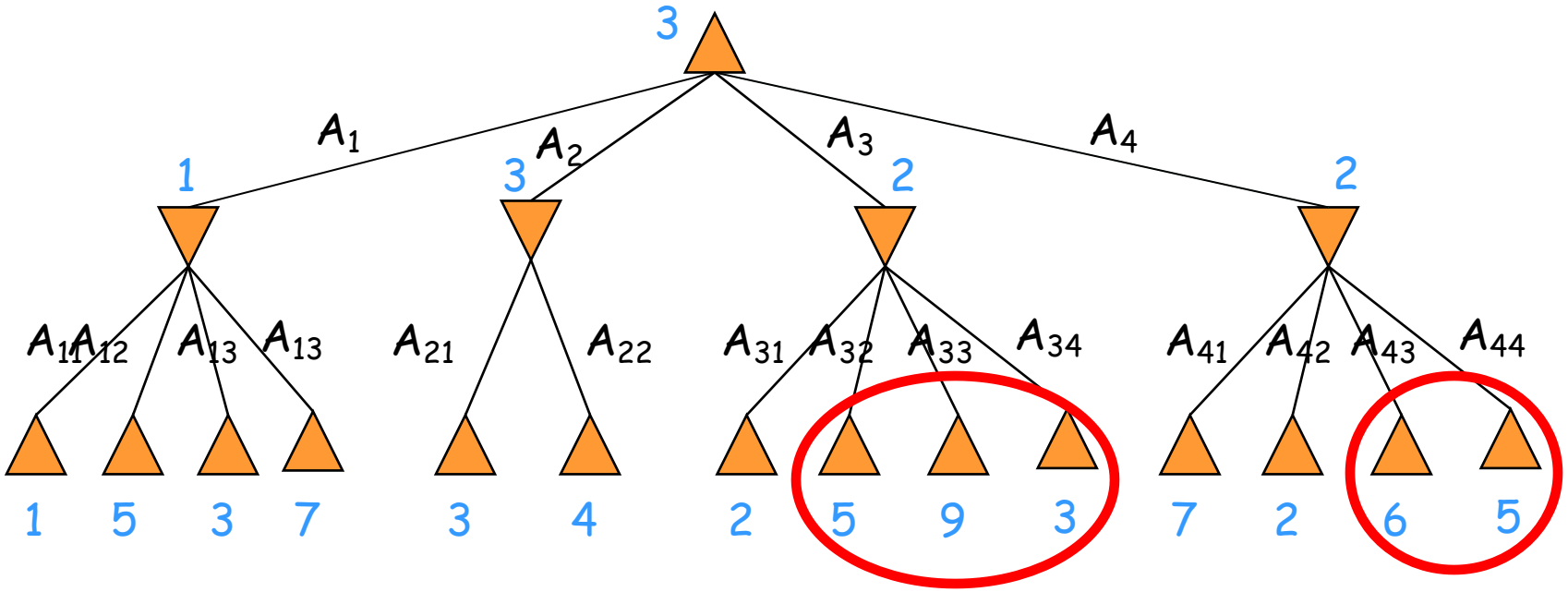

MAX
MIN
Terminal
5
A1
A2
A3
5
5
1
A11
A12
A13
A21
A22
A23
A31
A32
A33
6
7
5
5
7
9
6
7
1

Juegos

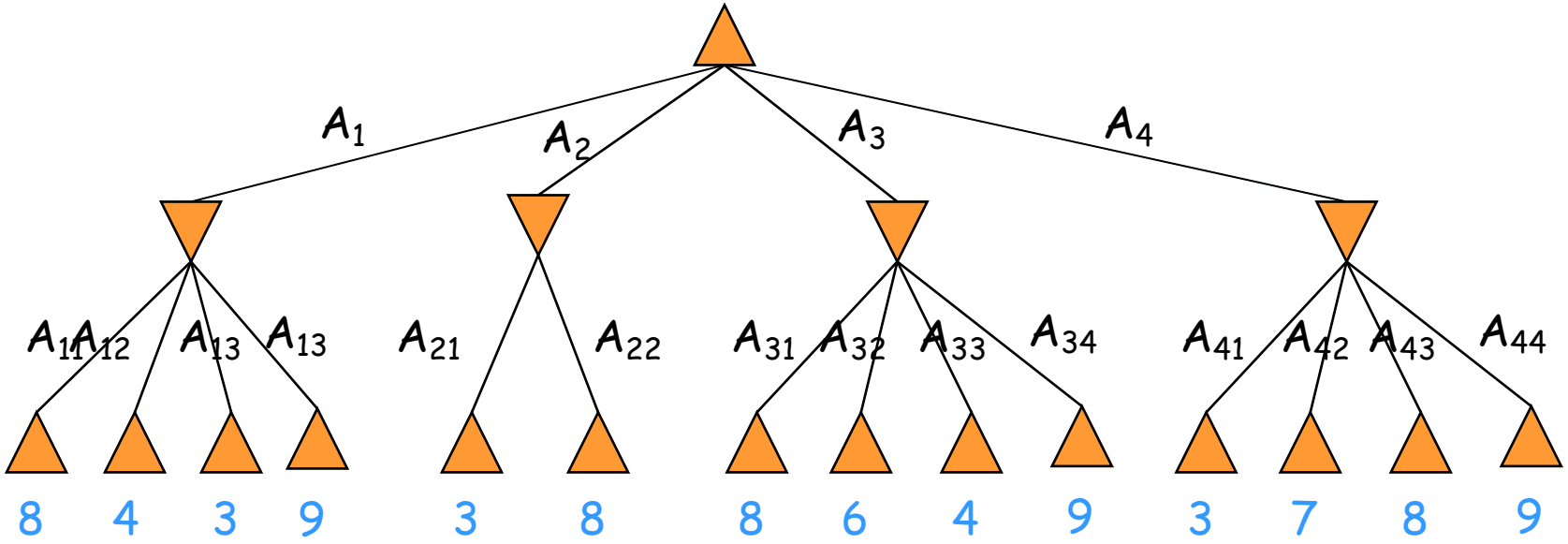


Indique qué nodos se podan

Juegos

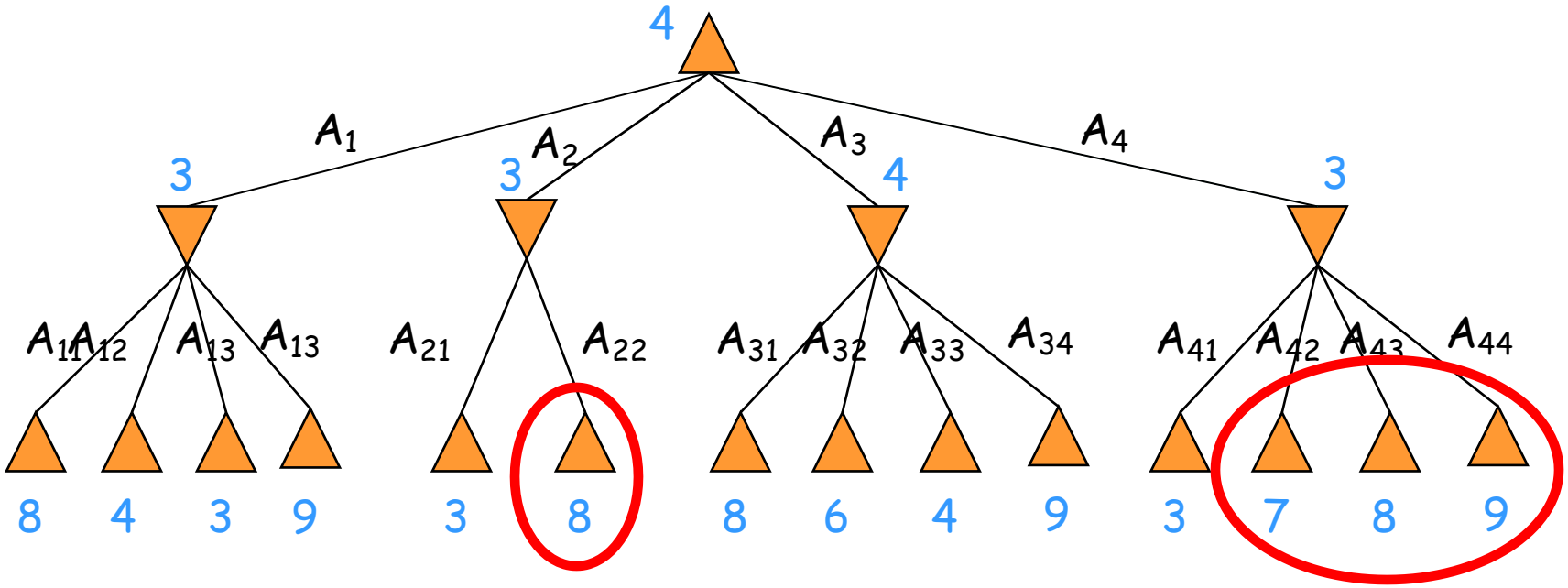

3
A1
A2
A3
A4
1
3
2
2
A11 A12
A13
A13
A21
A22
A31
A32
A33
A34
A41
A42
A43
A44
1
5
3
7
3
4
2
5
9
3
7
2
6
5

Juegos

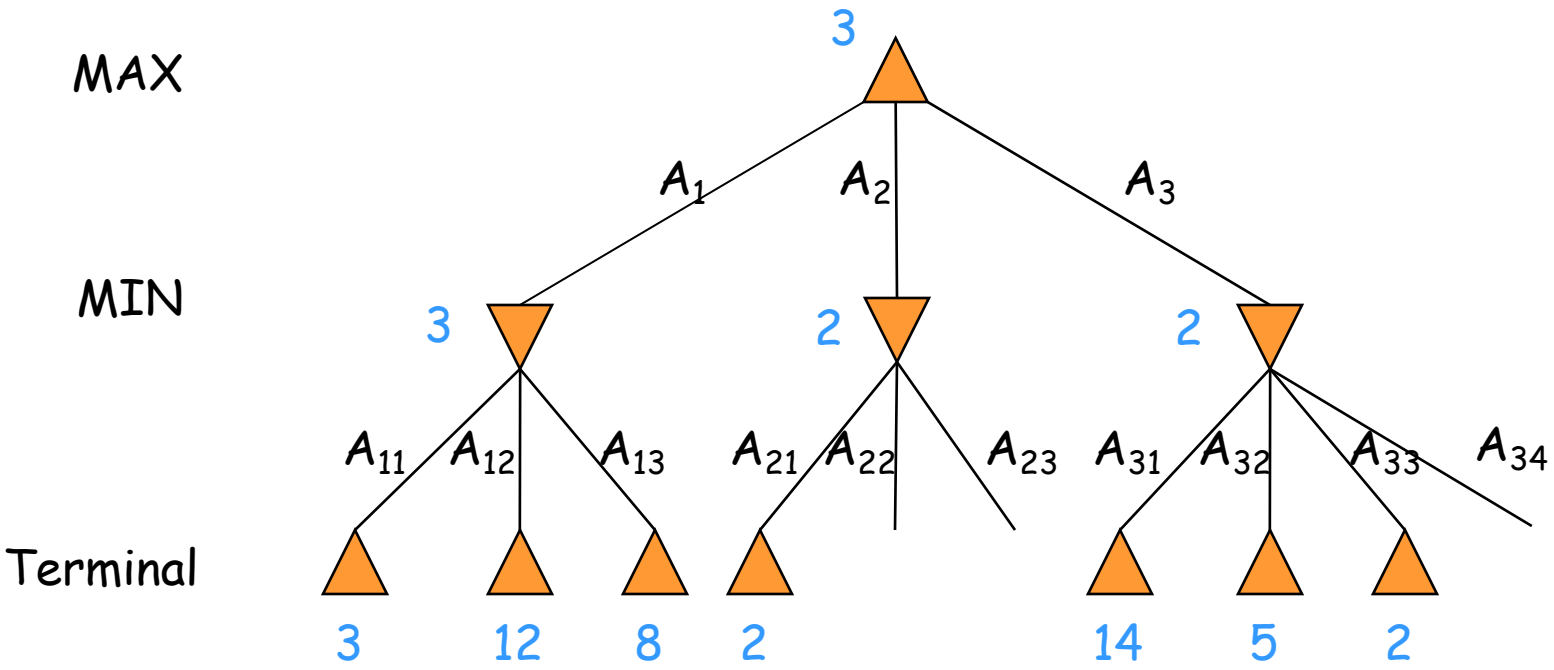


Indique qué nodos se podan

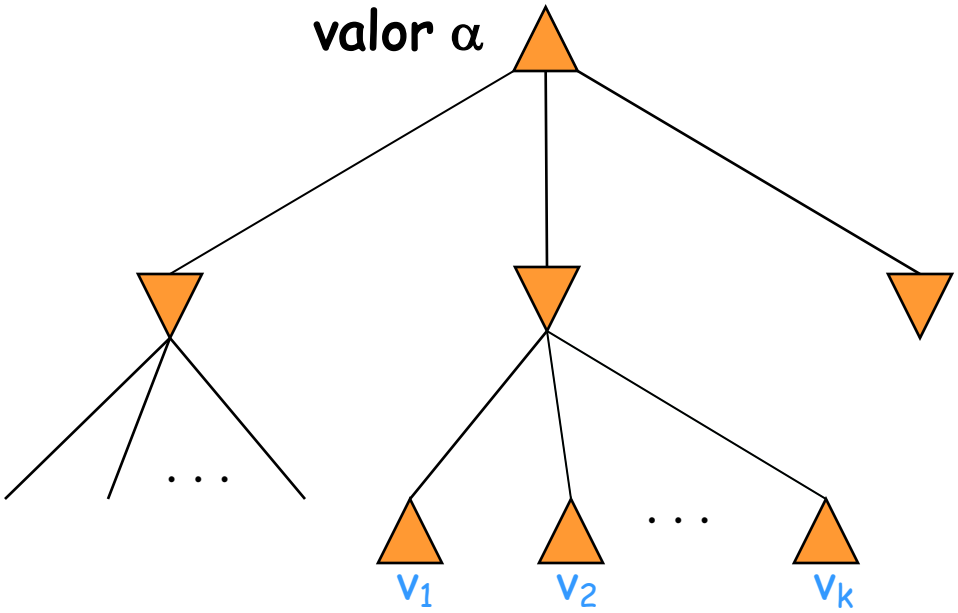
Juegos


4
A1
A2
A3
A4
3
3
4
3
A11 A12
A13
A13
A21
A22
A31
A32
A33
A34
A41
A42
A43
A44
8
4
3
9
3
8
8
6
4
9
3
7
8
9

Juegos


MAX
MIN
Terminal
3
A1
A2
A3
3
2
2
A11
A12
A13
A21
A22
A23
A31
A32
A33
A34
3
12
8
2
14
5
2

Juegos

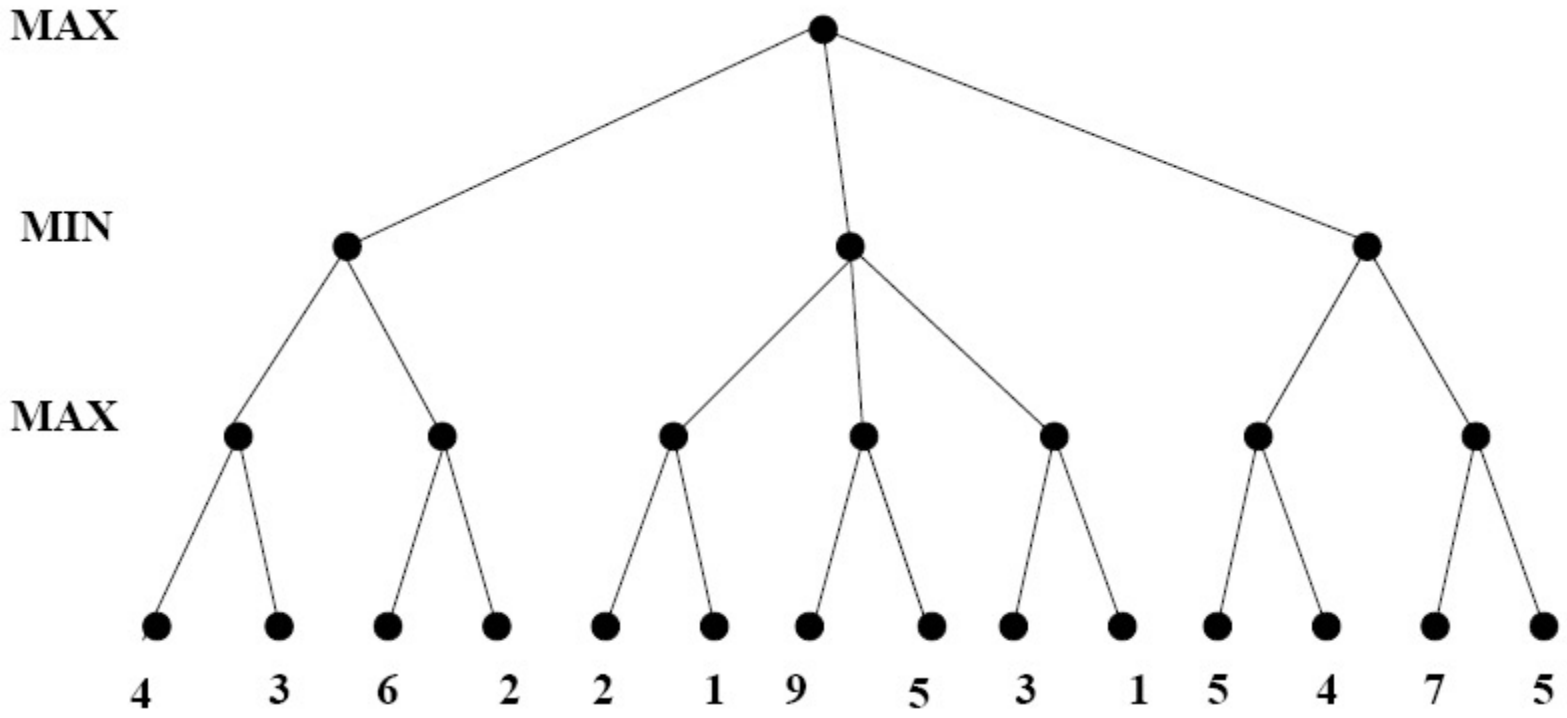


- Poda de los hijos de MIN

Podar los nodos a la derecha del nodo cuyo v_i es menor o igual que α

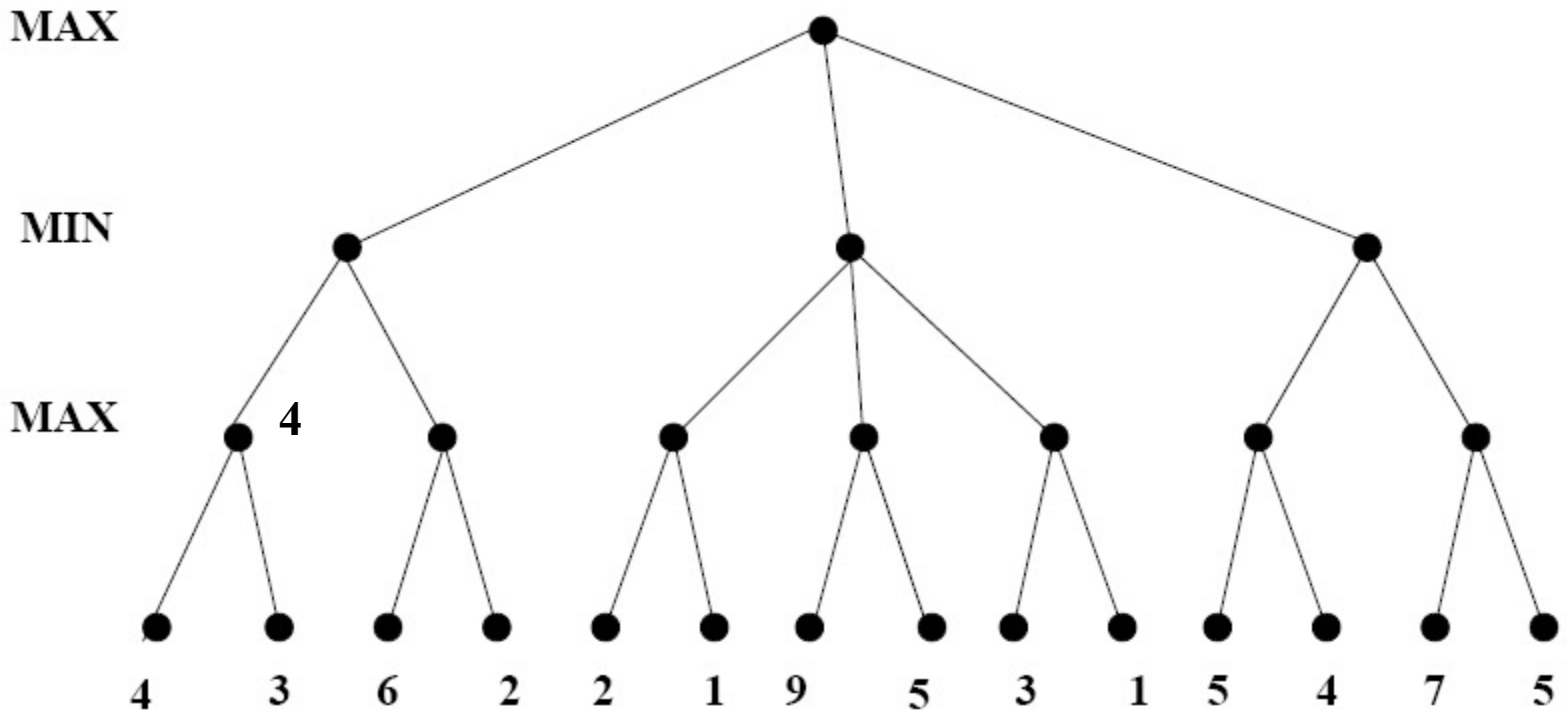
Juegos

Poda alfa-beta



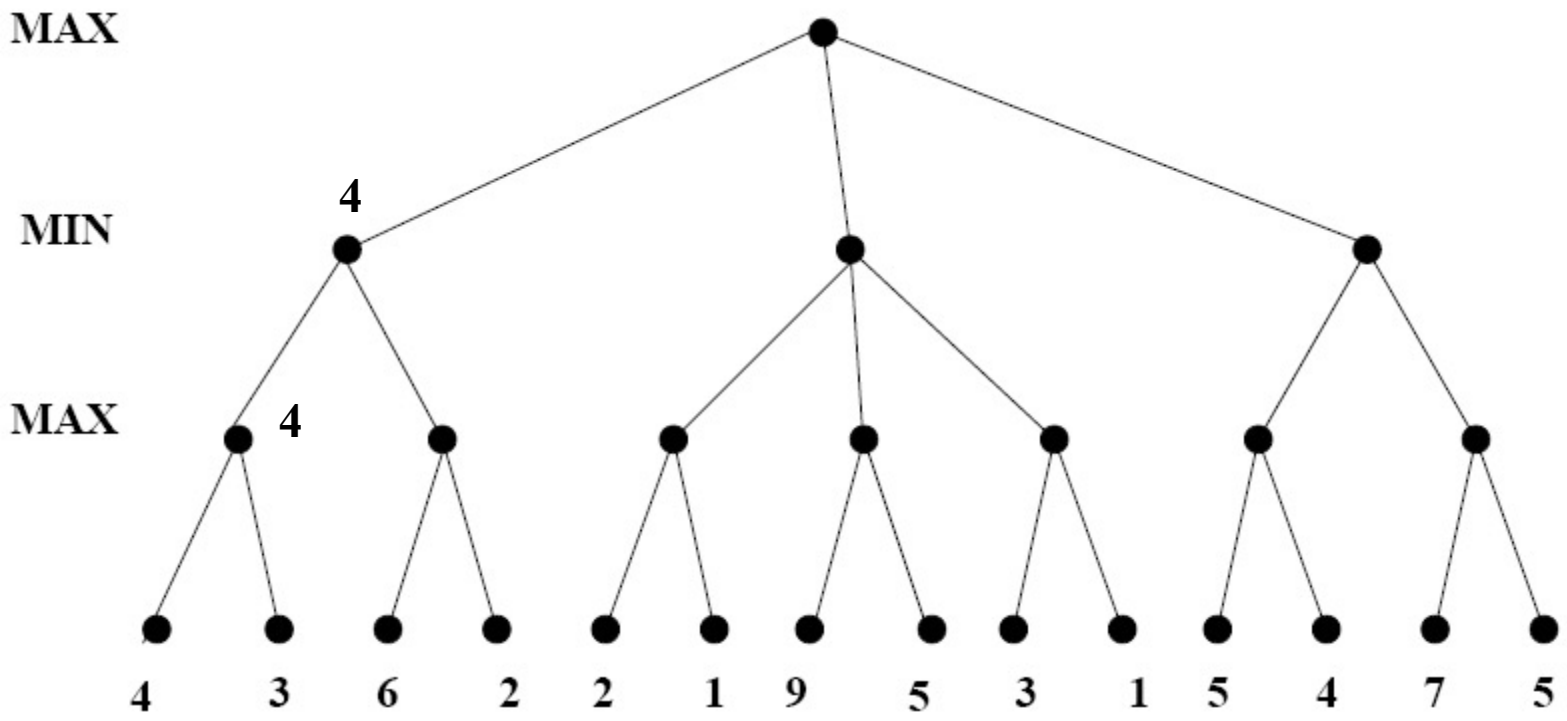
Juegos

Poda alfa-beta



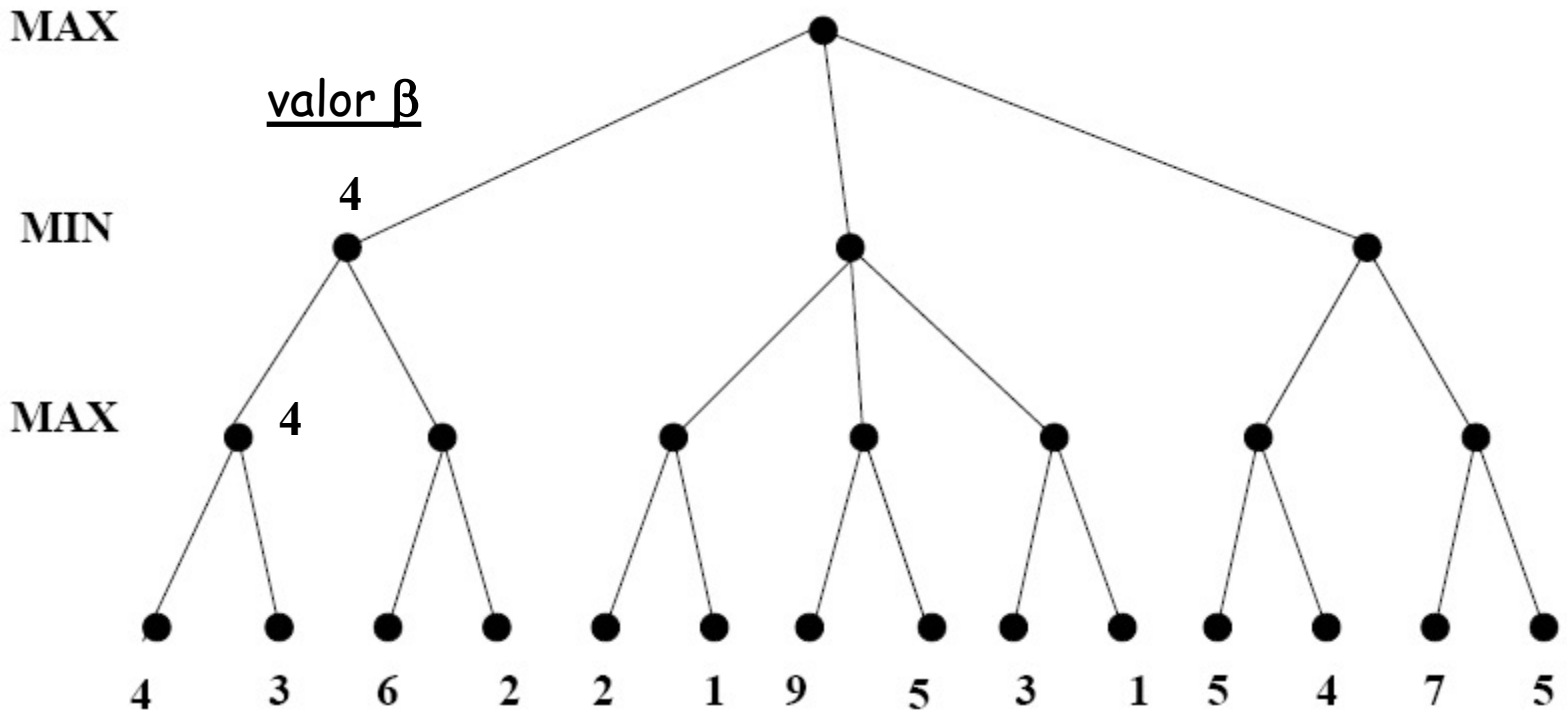
Juegos

Poda alfa-beta



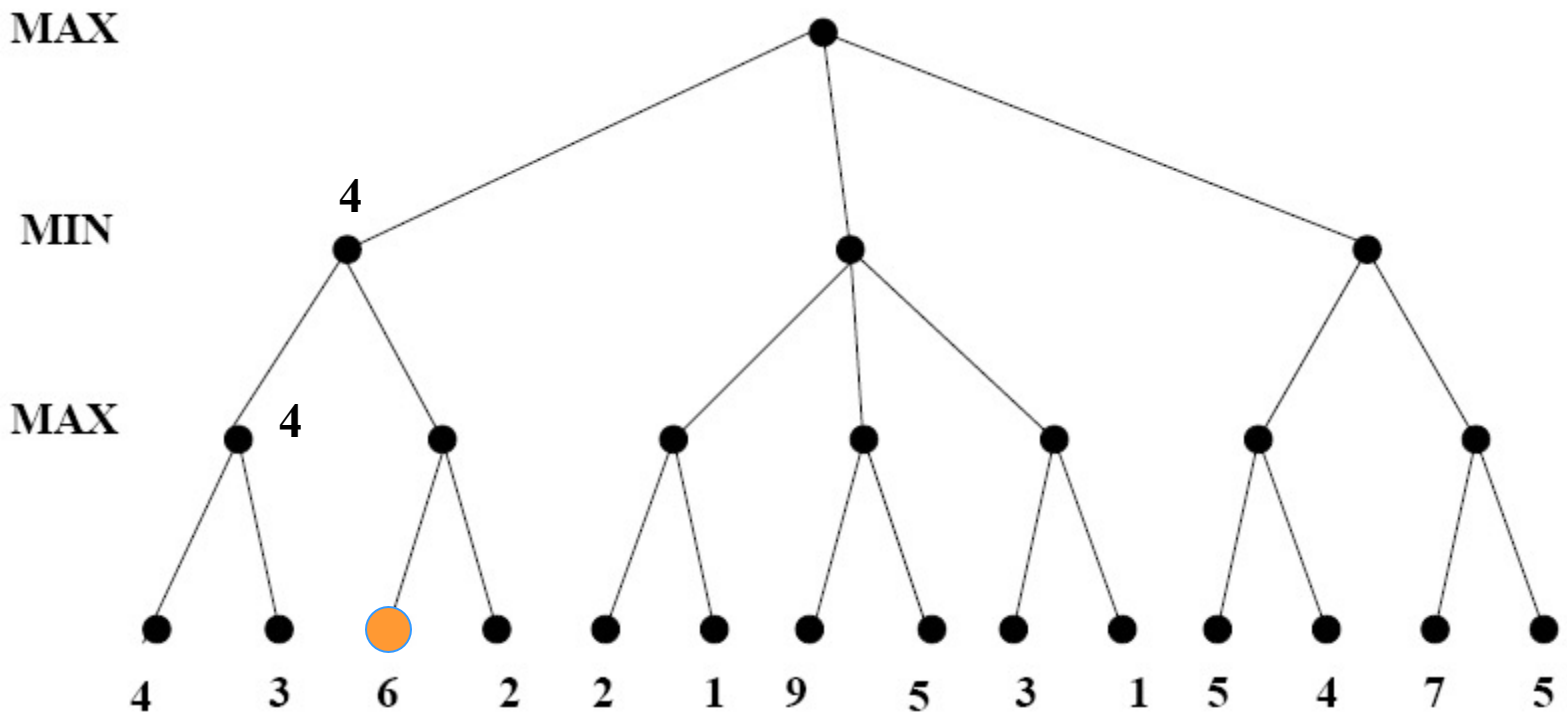
Juegos

Poda alfa-beta



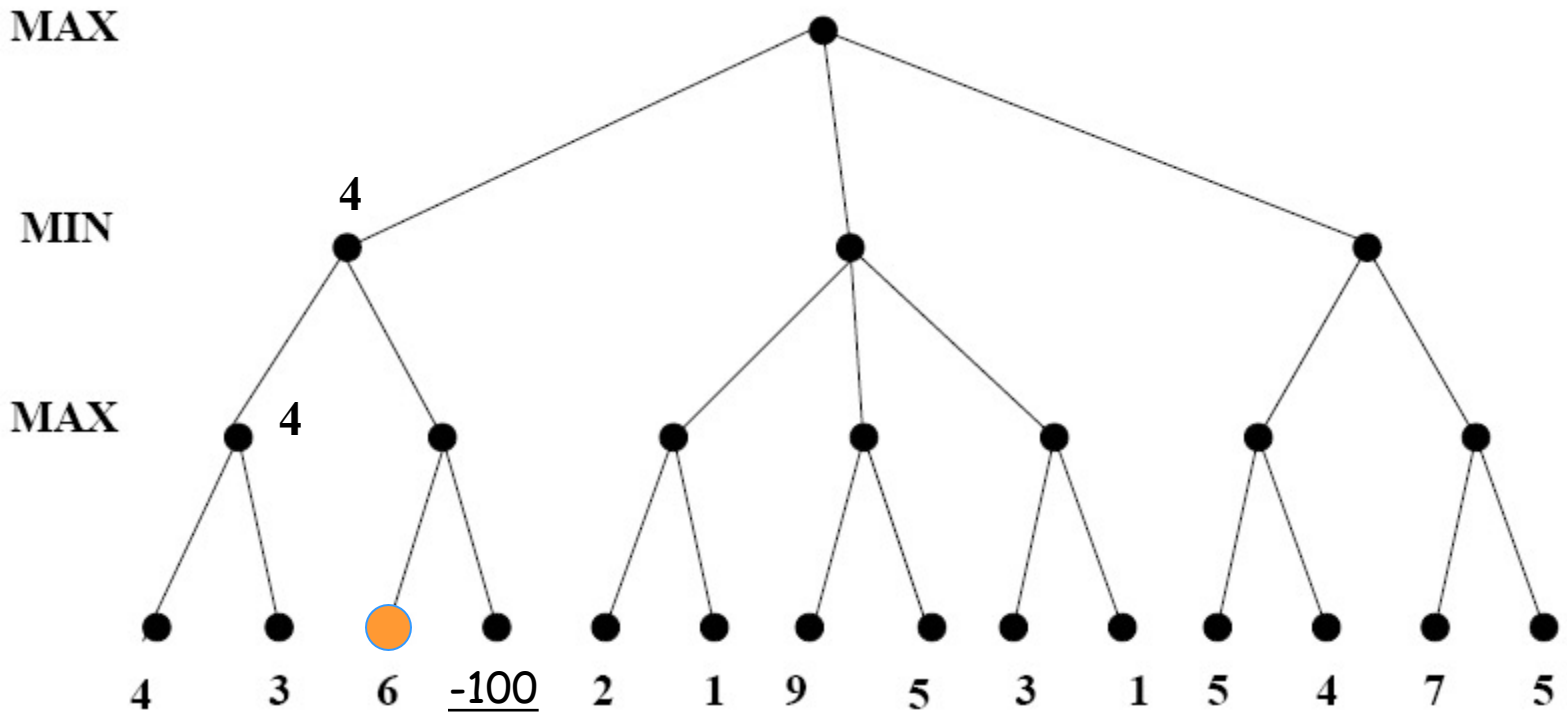
Juegos

Poda alfa-beta



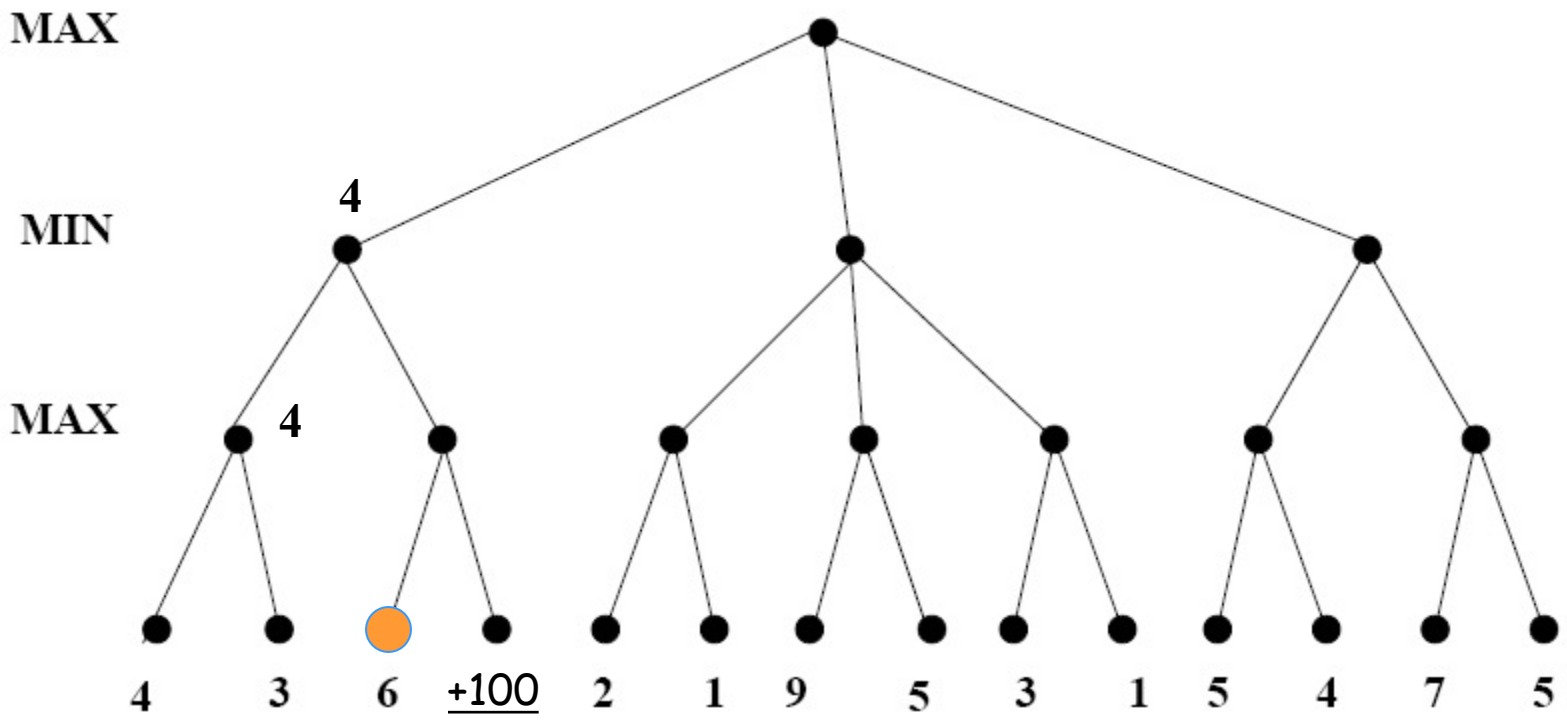
Juegos

Poda alfa-beta



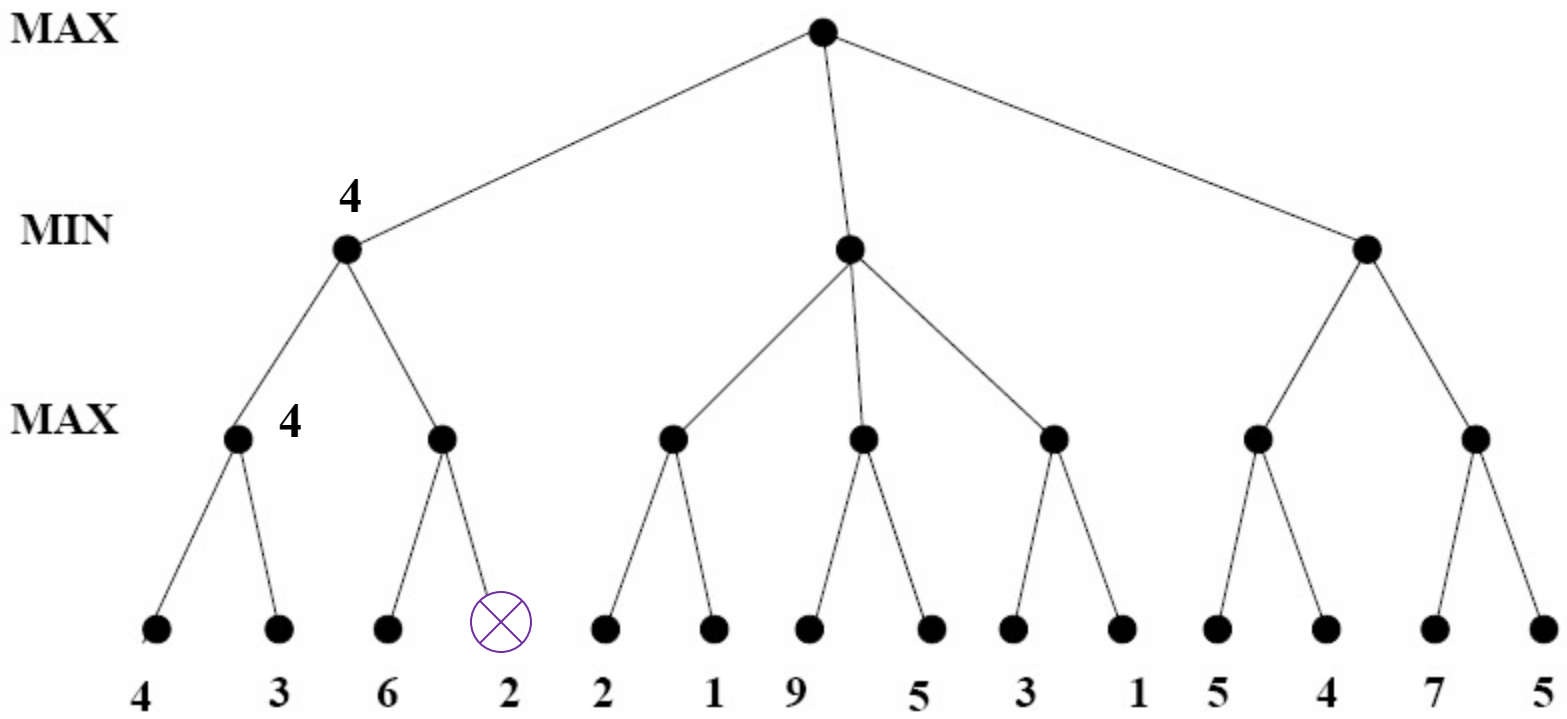
Juegos

Poda alfa-beta



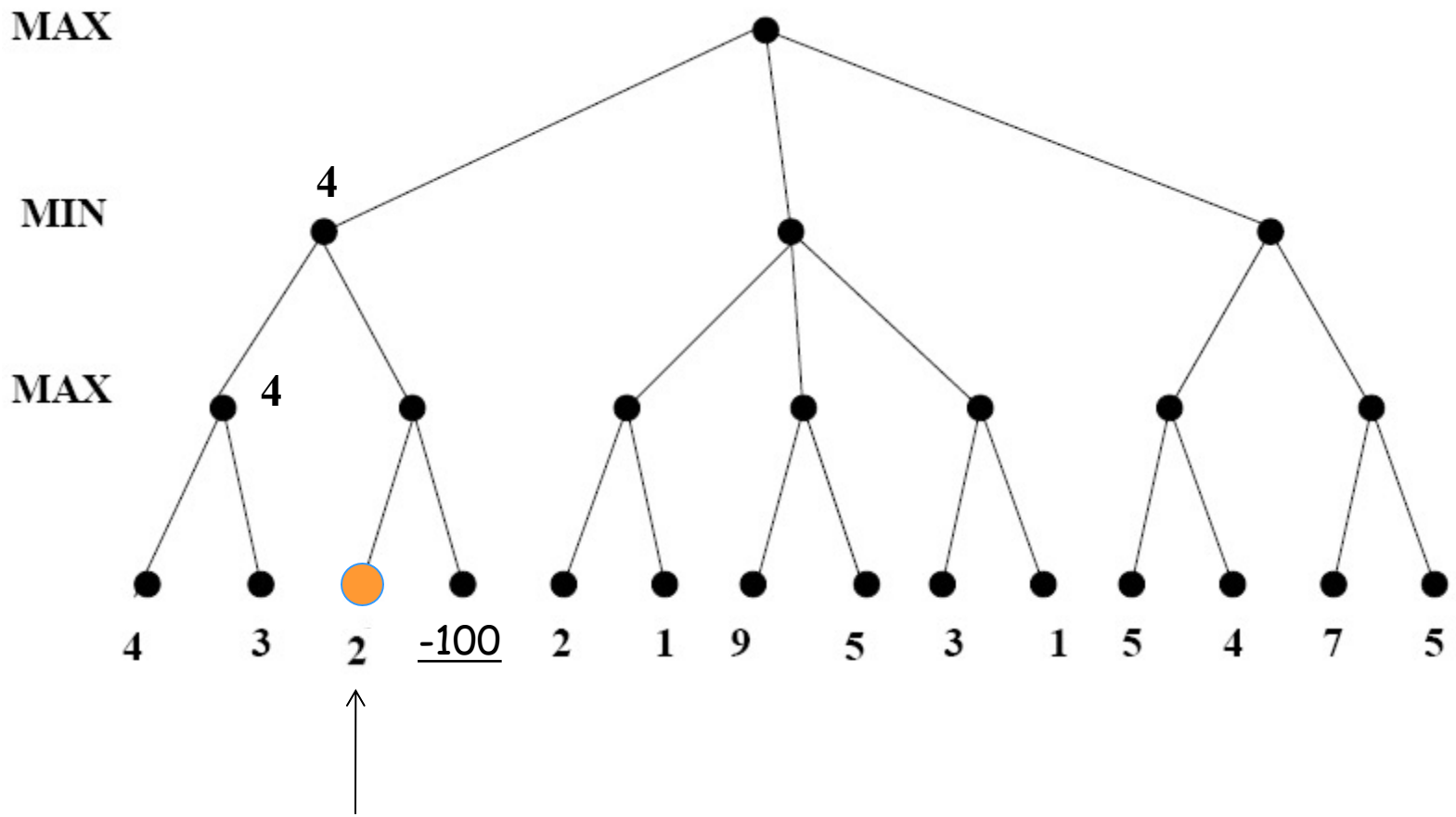
Juegos

Poda alfa-beta



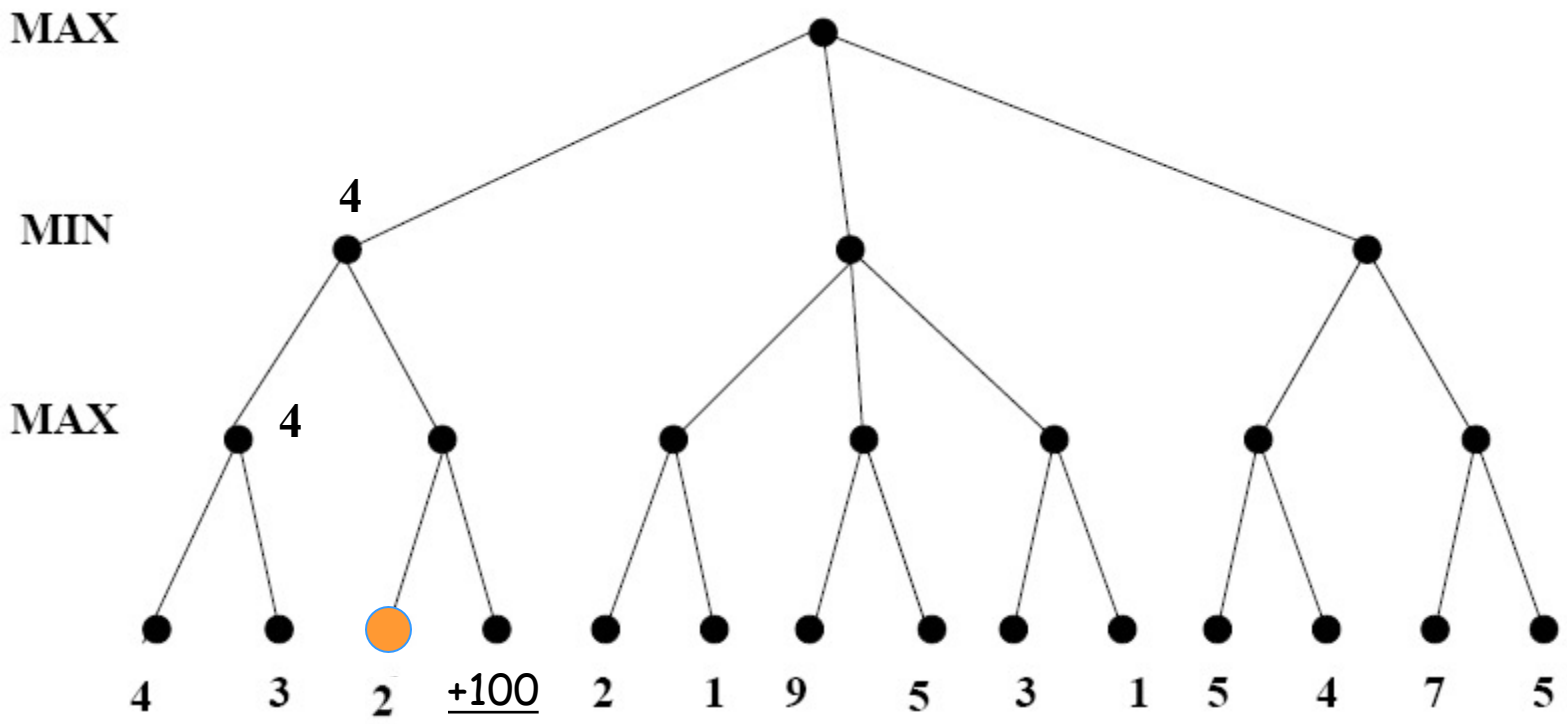
Juegos

Poda alfa-beta

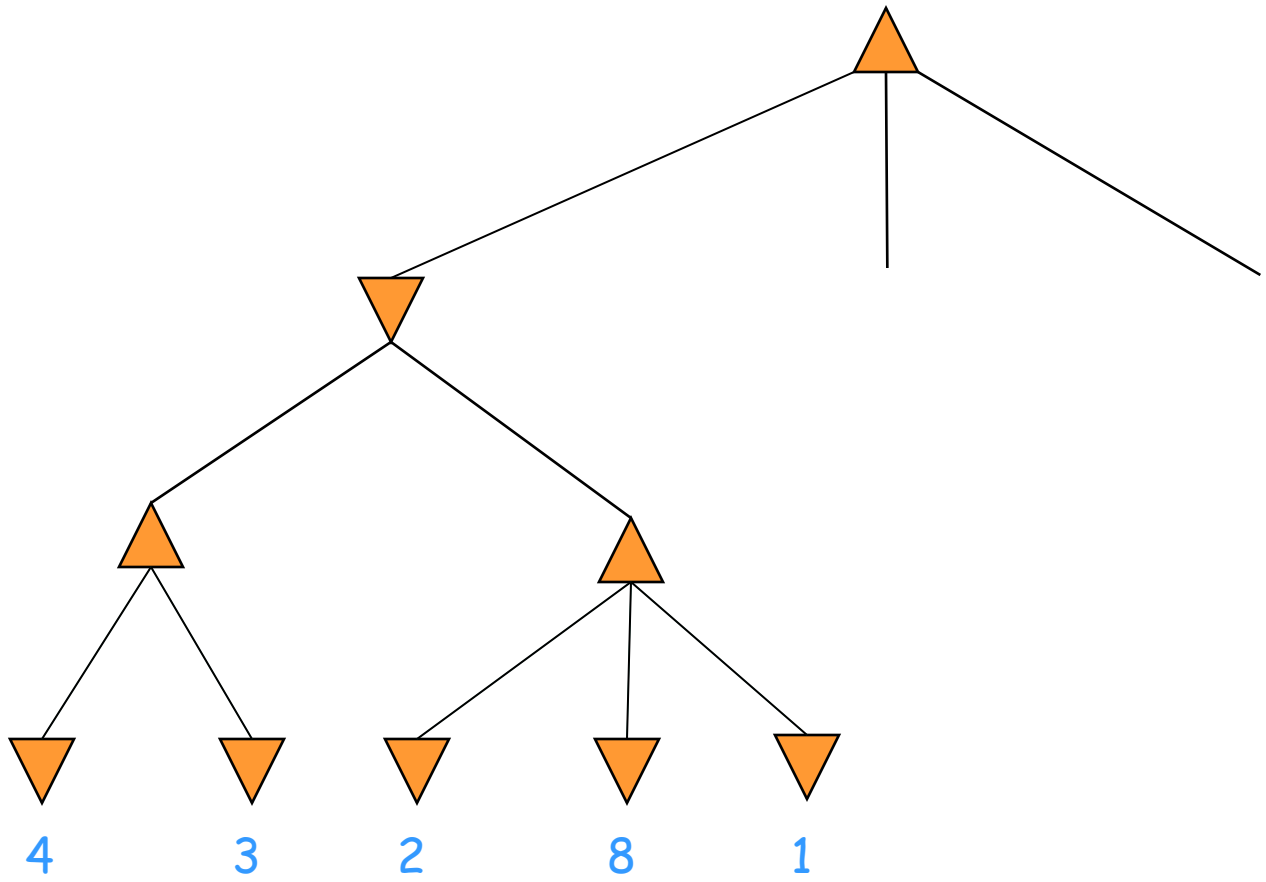


Juegos

Poda alfa-beta

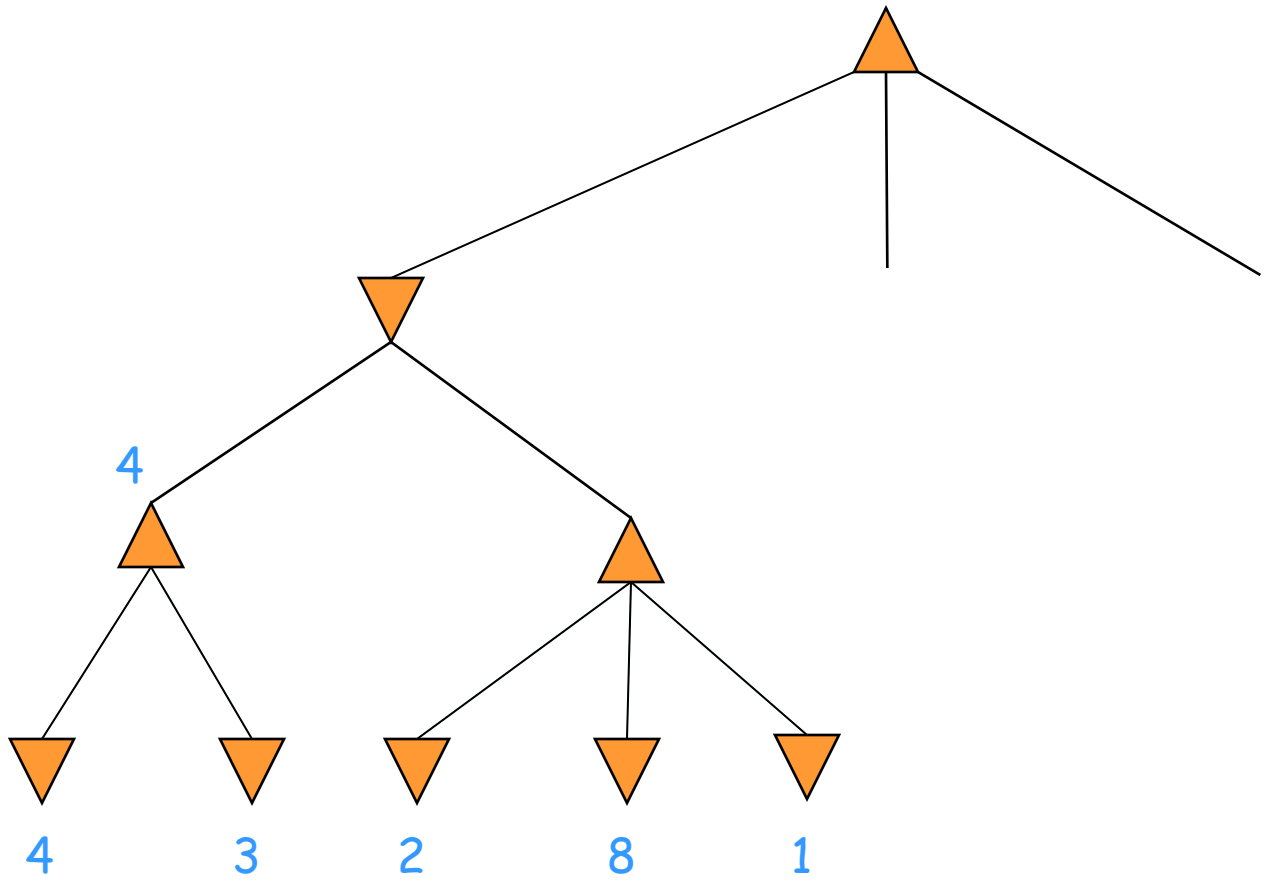


Juegos

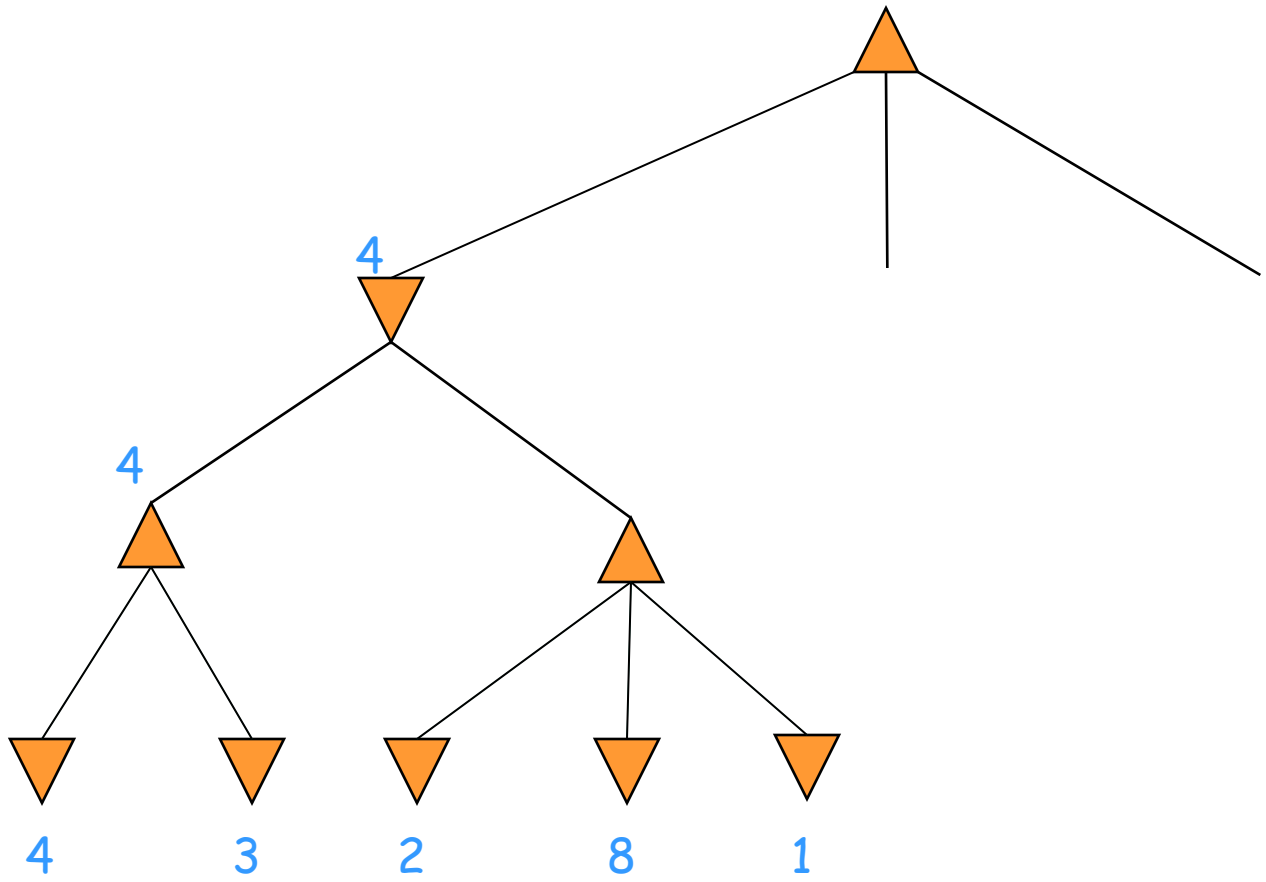


Indique qué nodos se podan

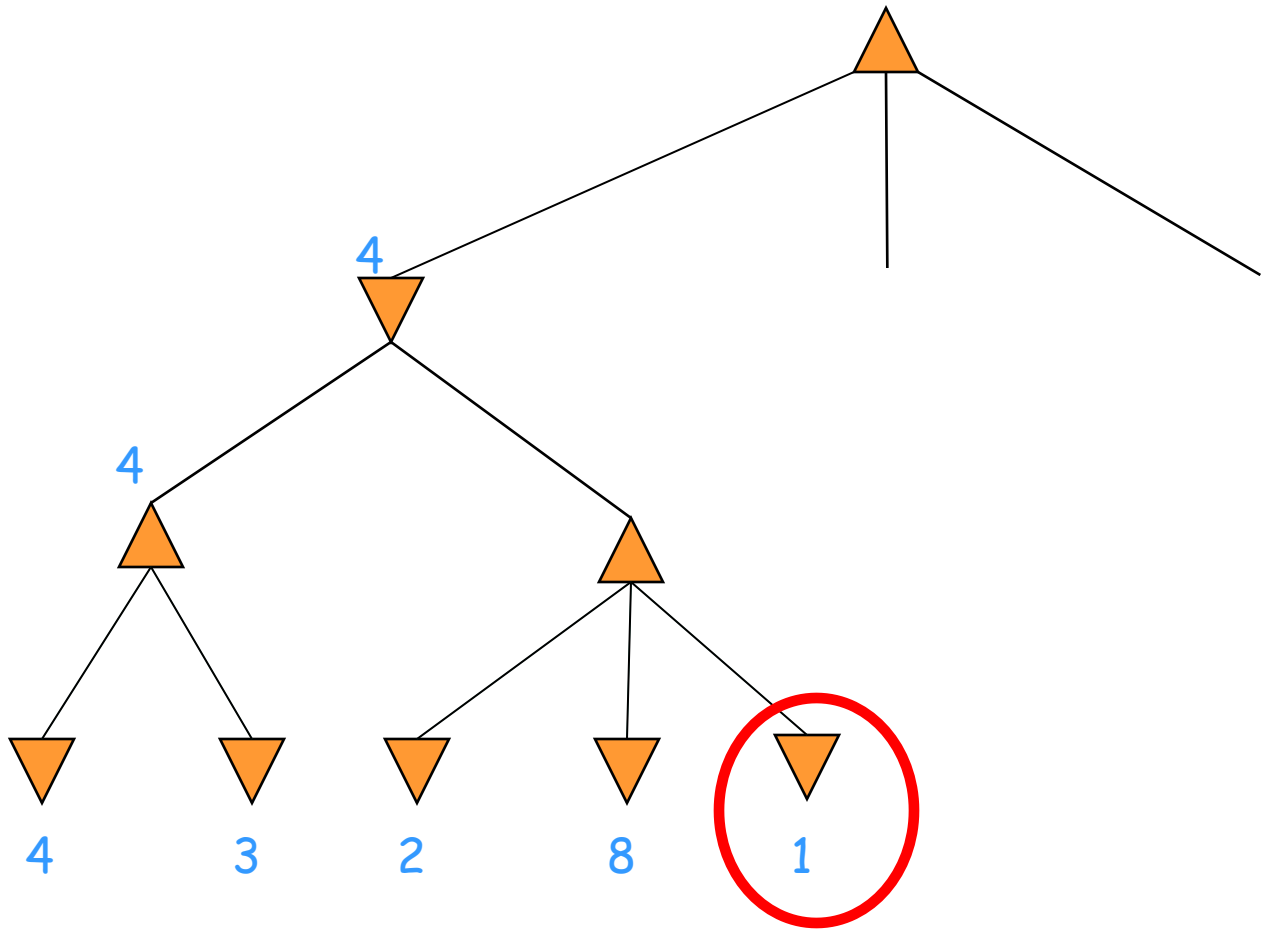
Juegos



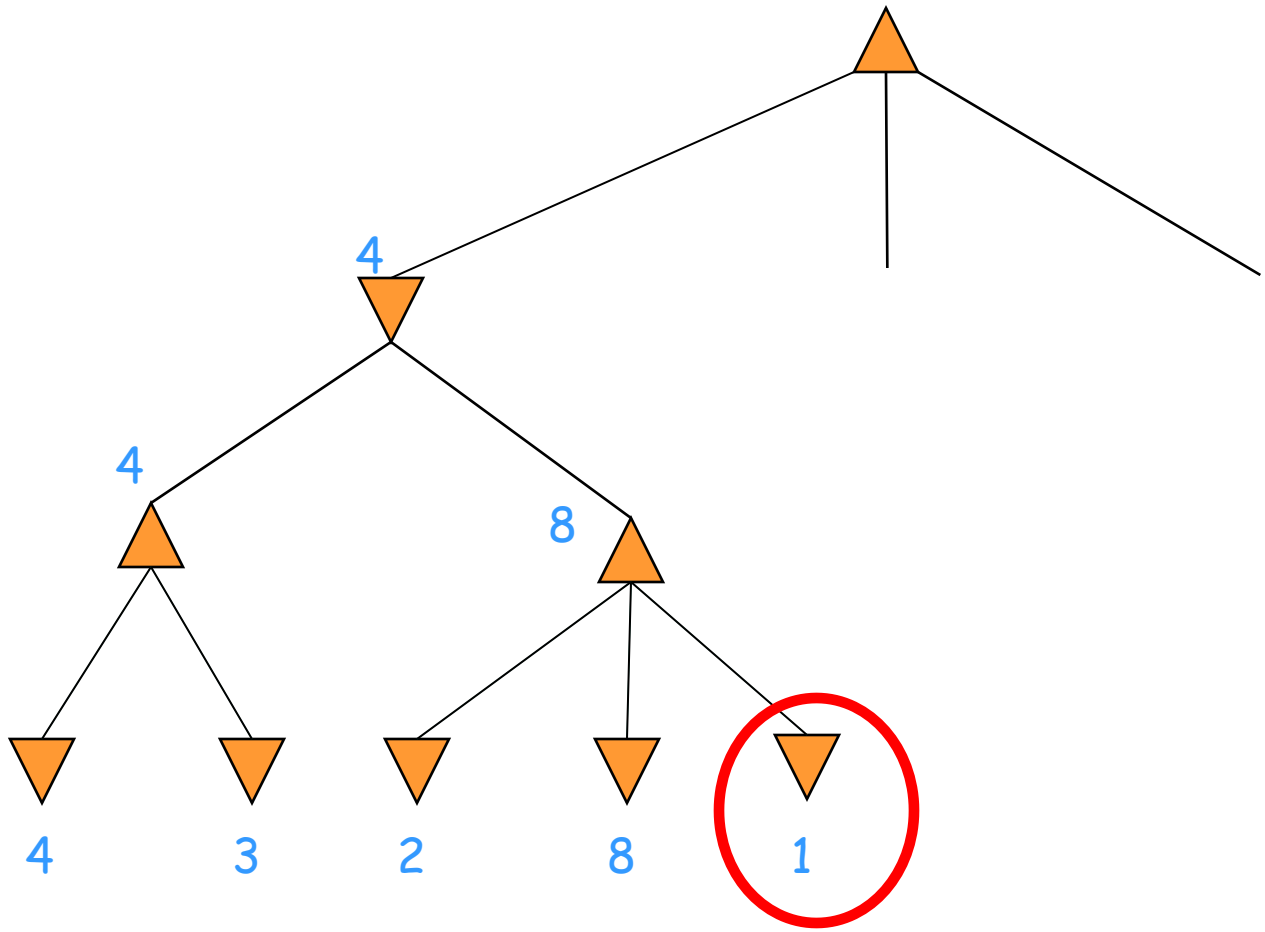
Juegos



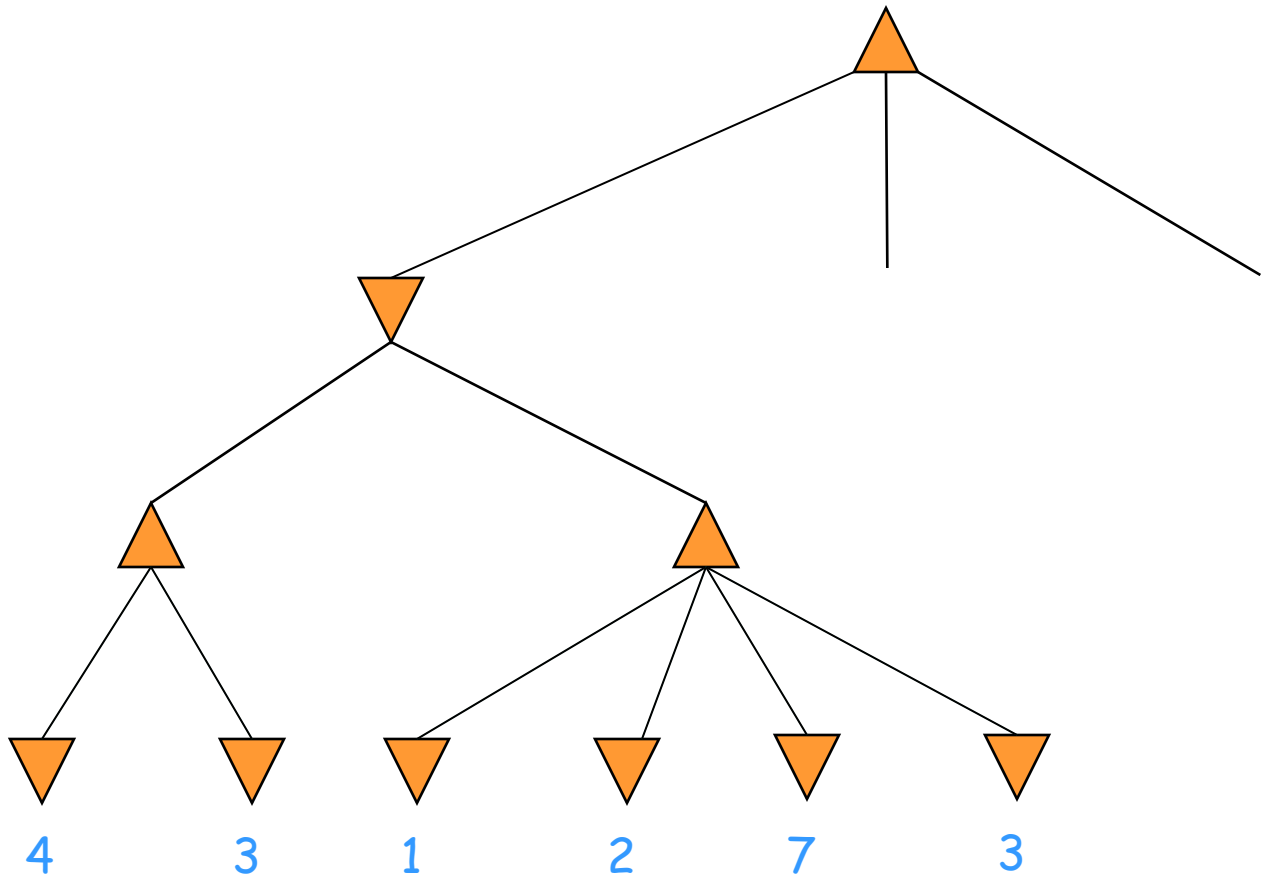
Juegos



Juegos

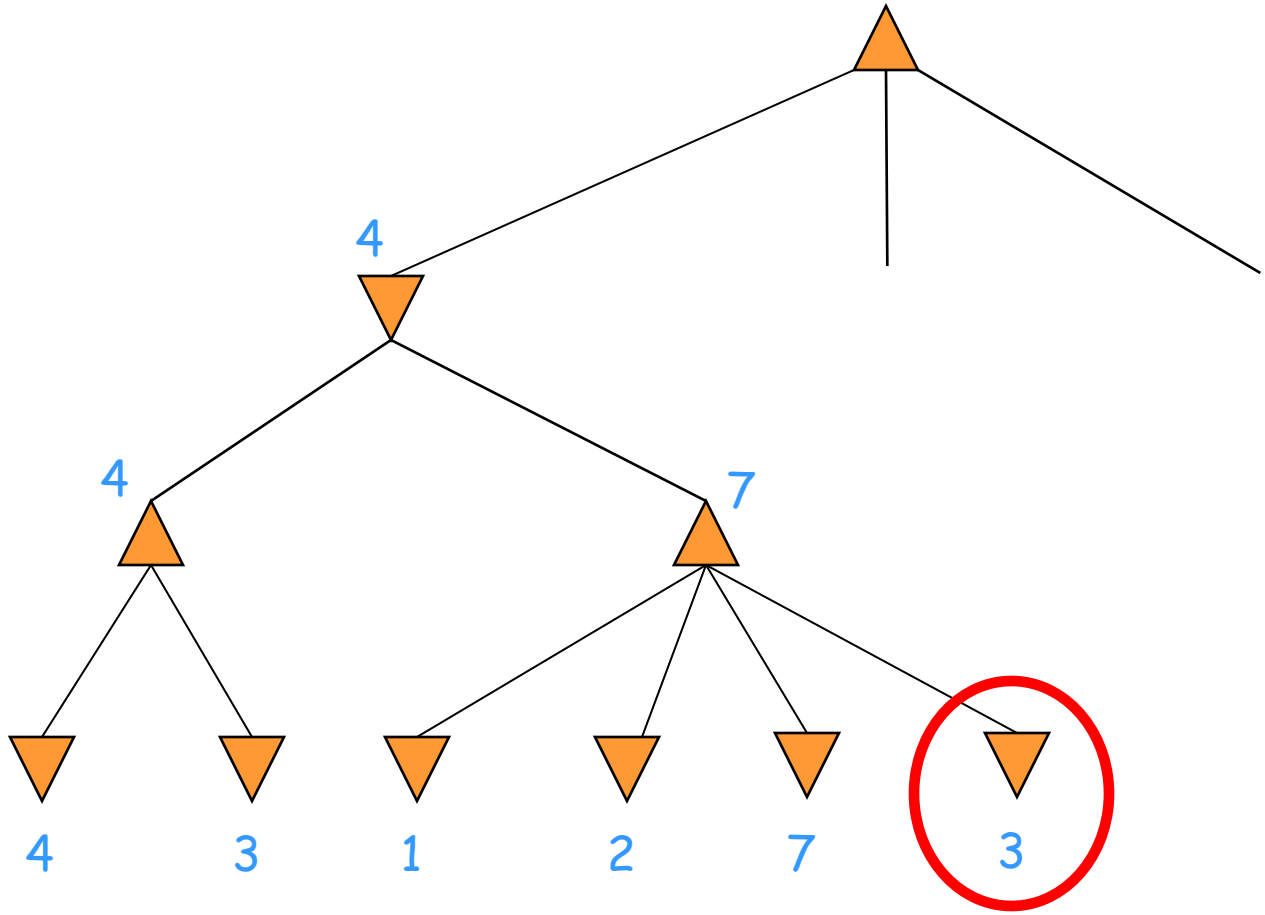


Juegos

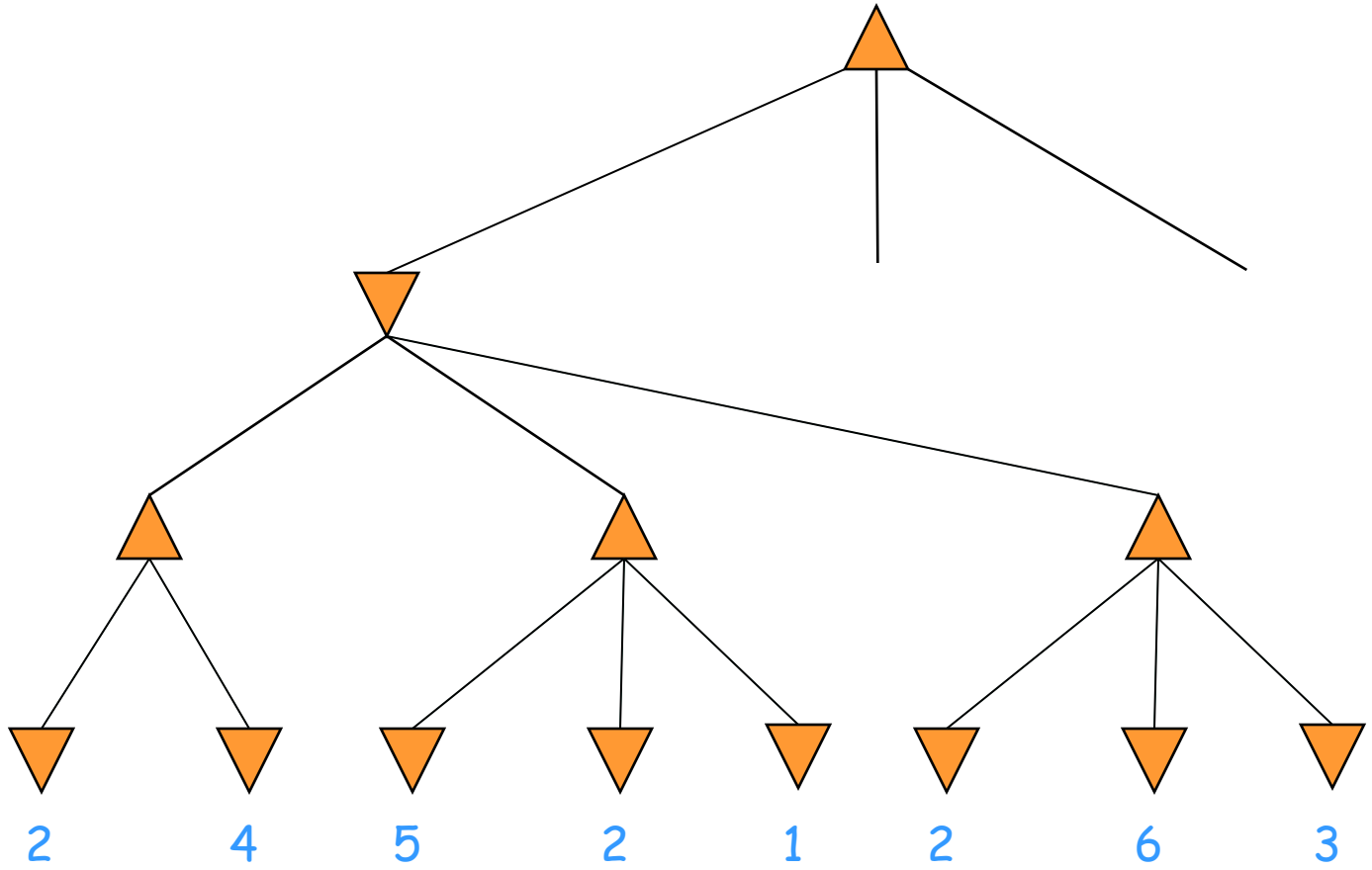


Indique qué nodos se podan

Juegos

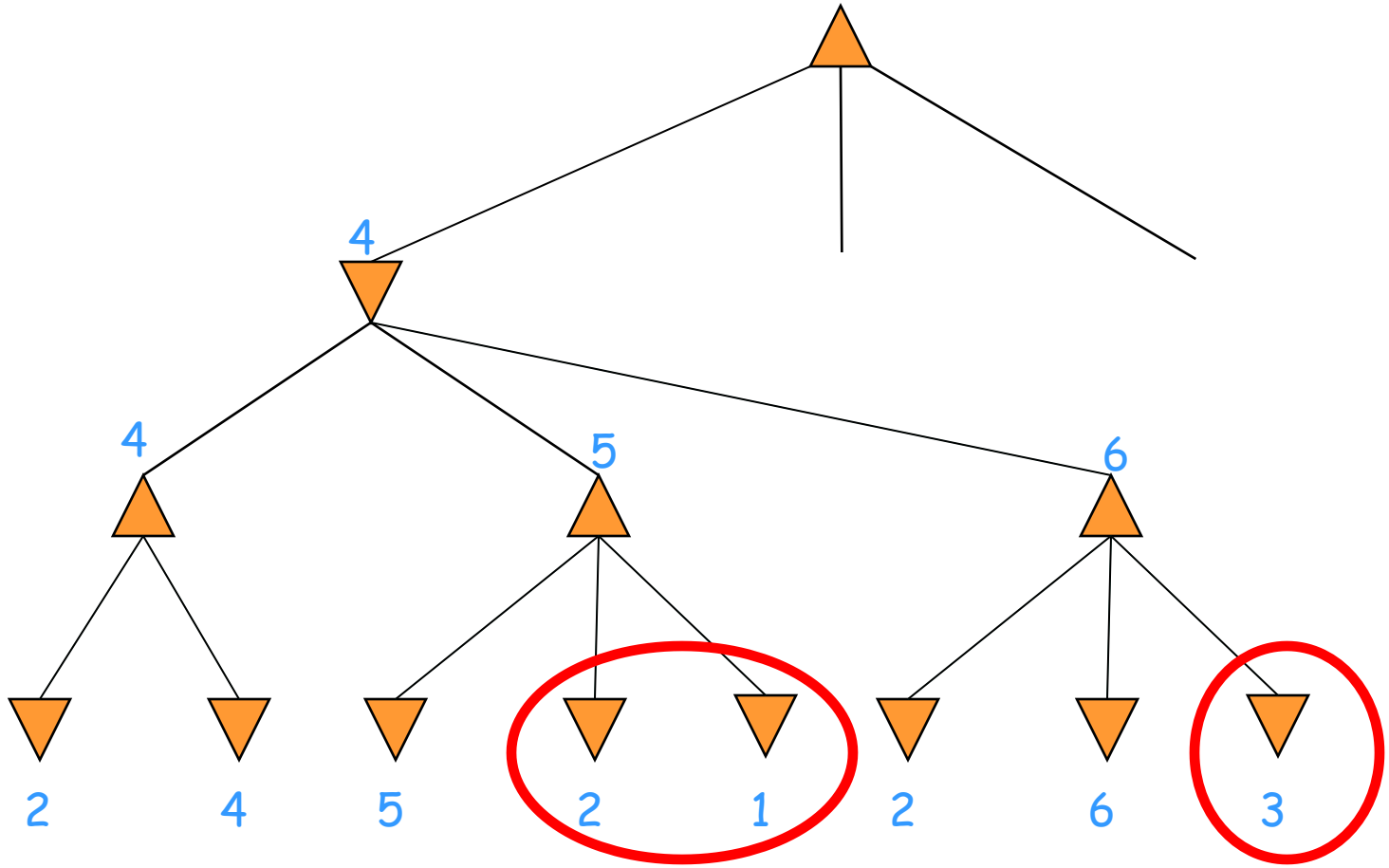


Juegos

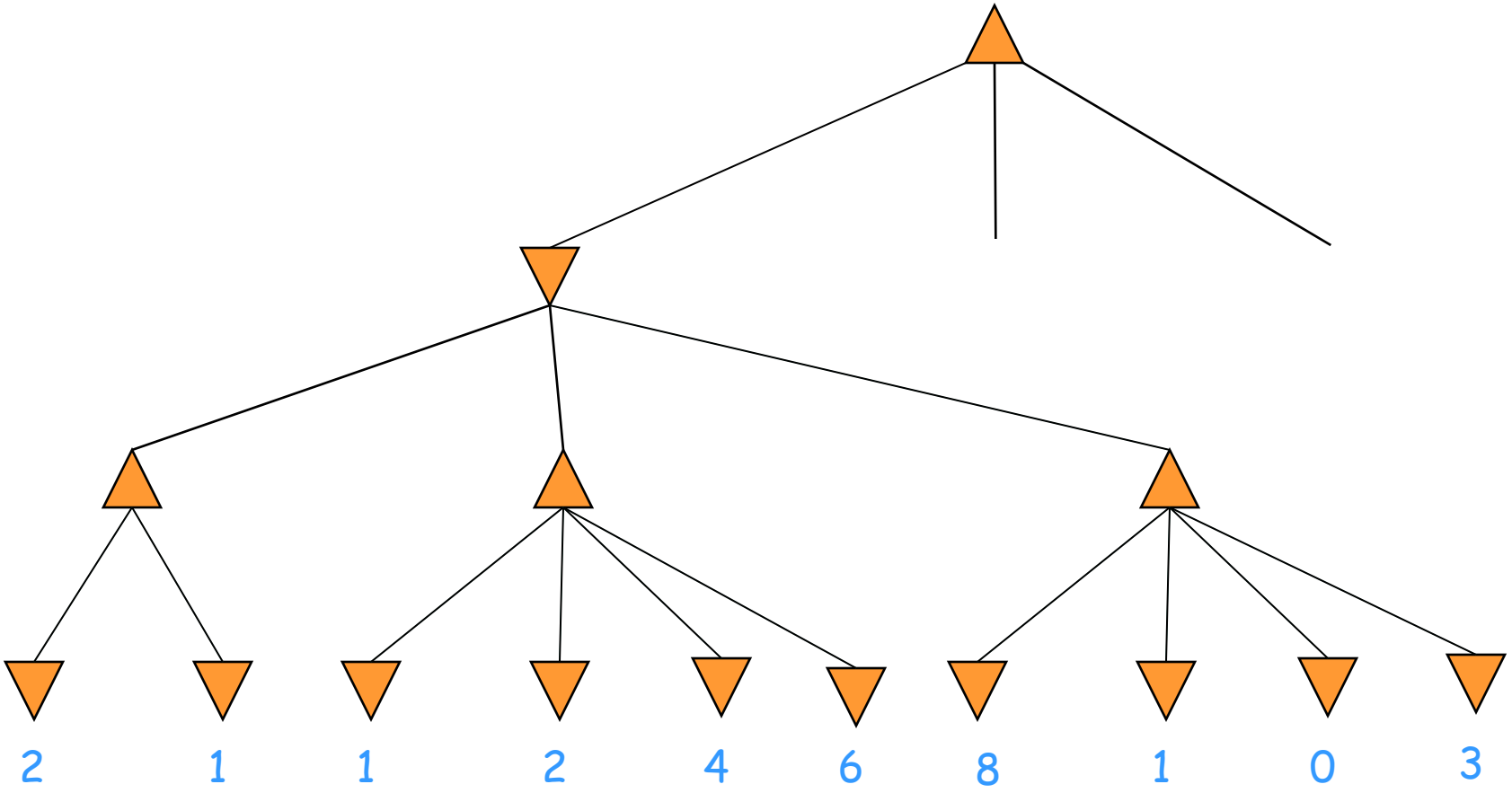


Indique qué nodos se podan

Juegos

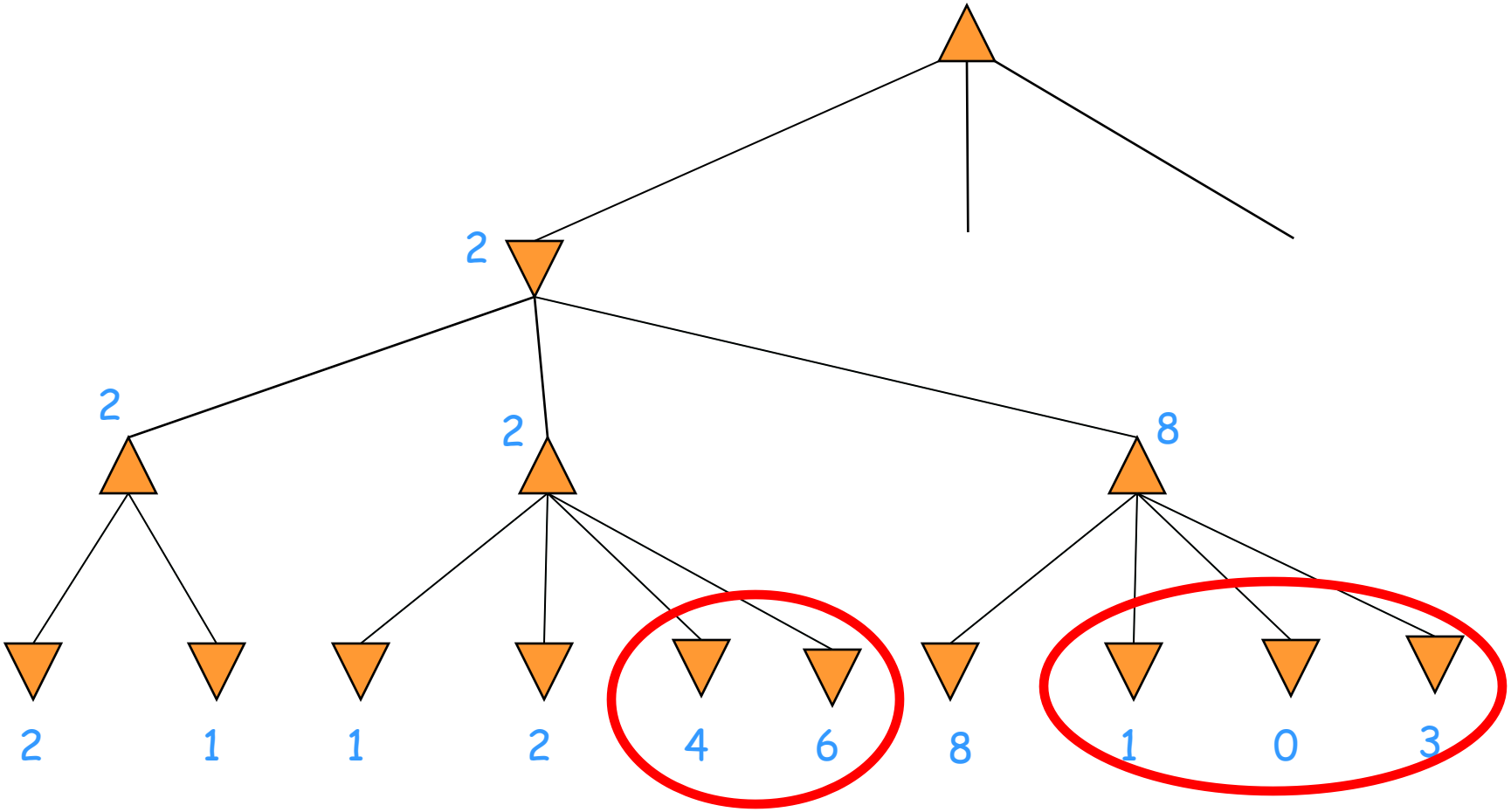


Juegos

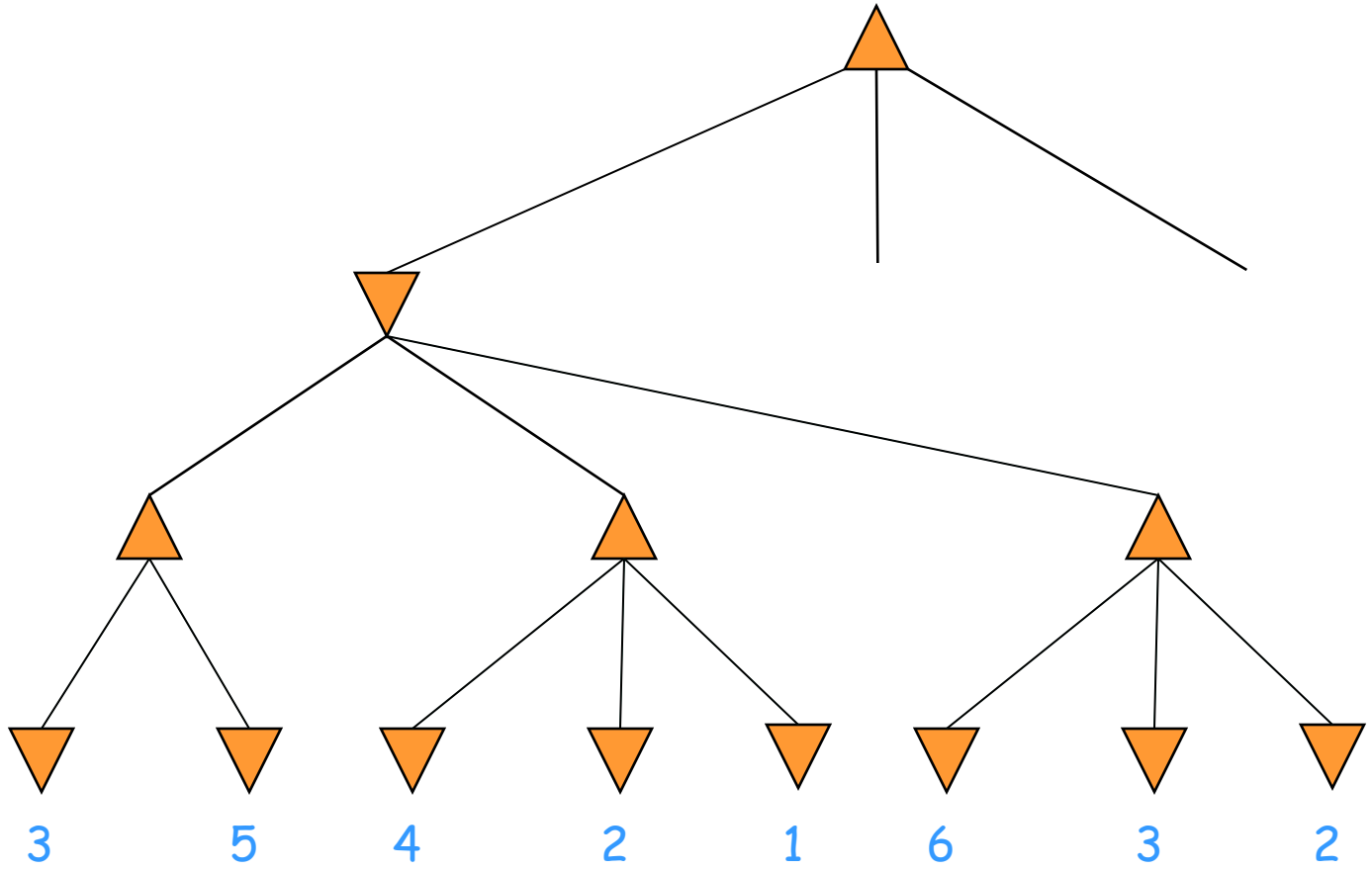


Indique qué nodos se podan

Juegos

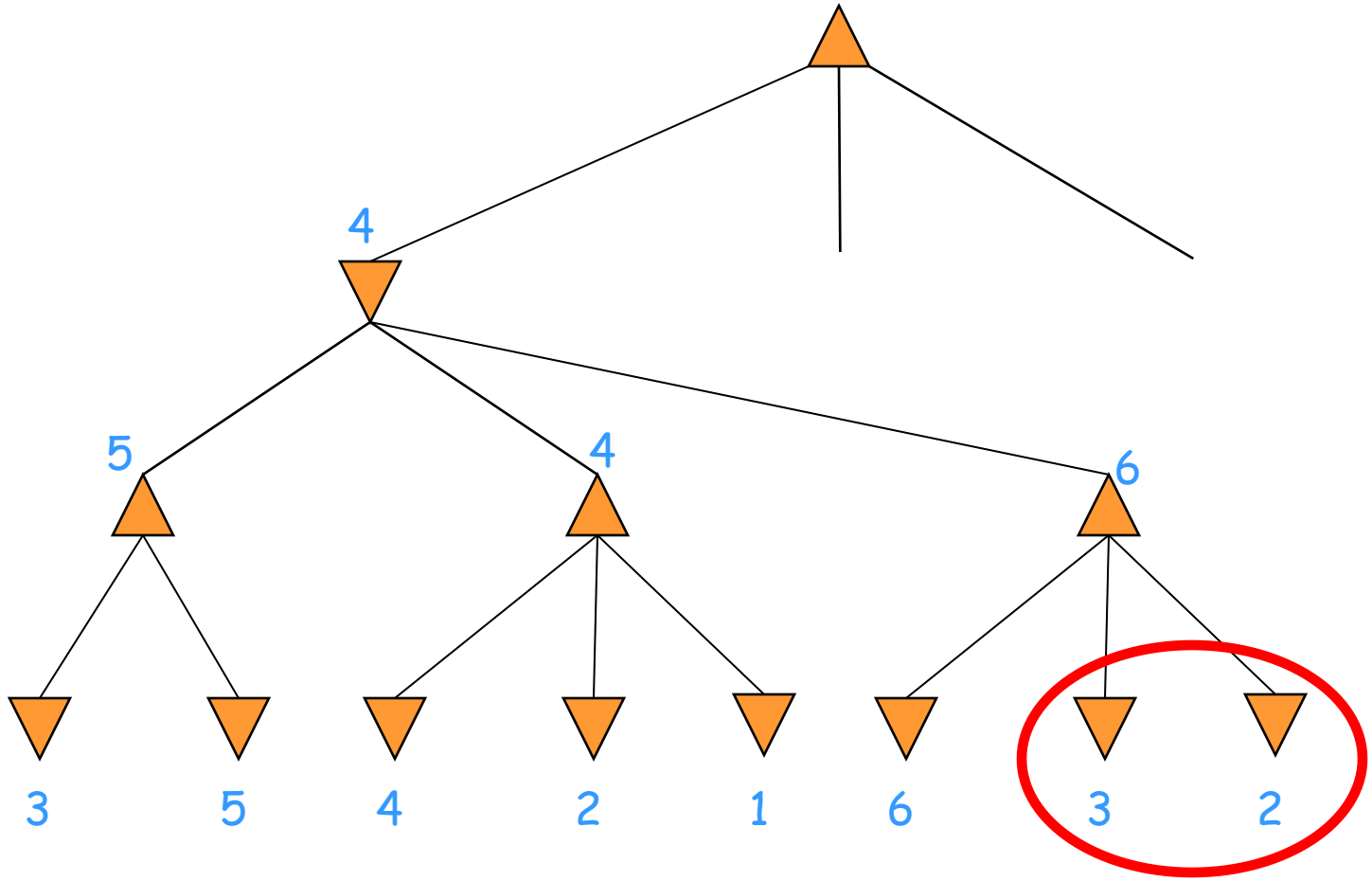


Juegos

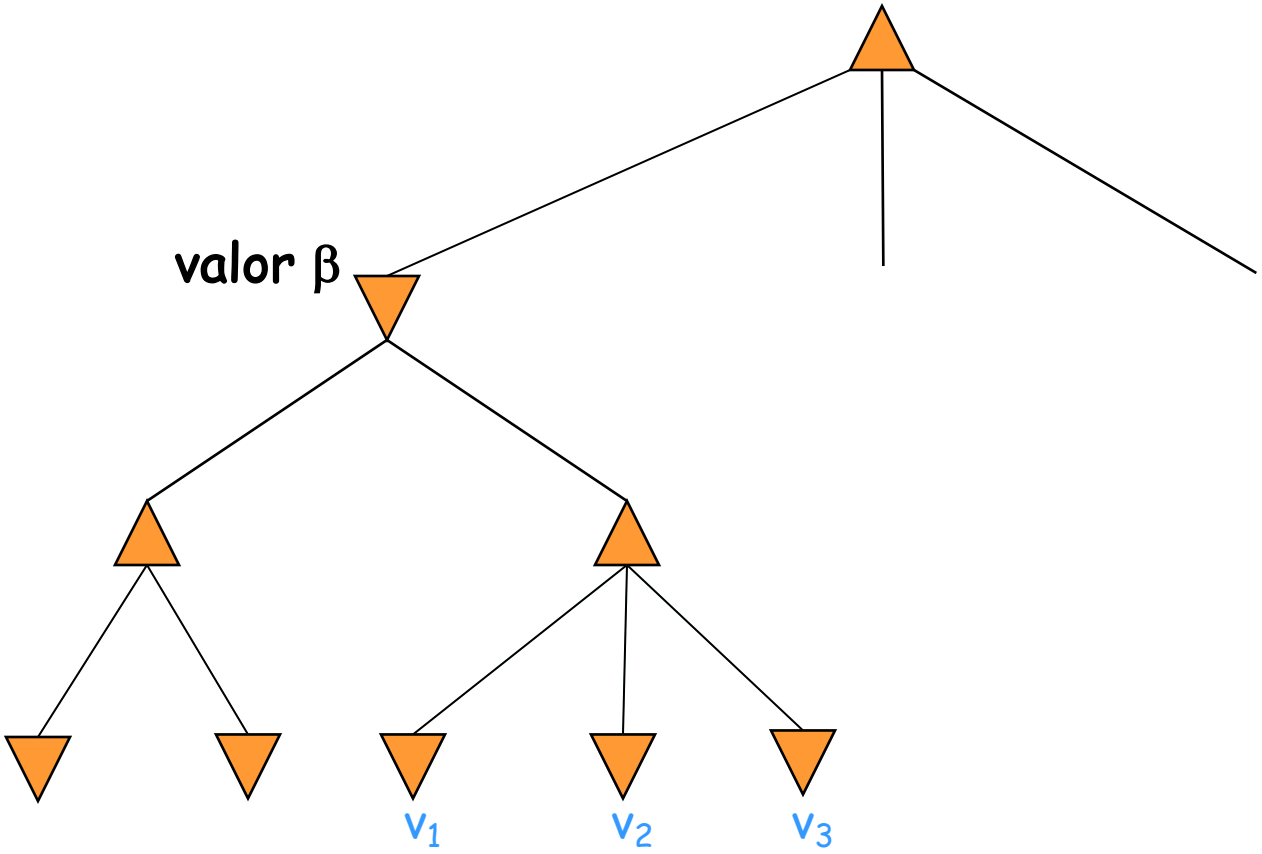


Indique qué nodos se podan

Juegos



Juegos

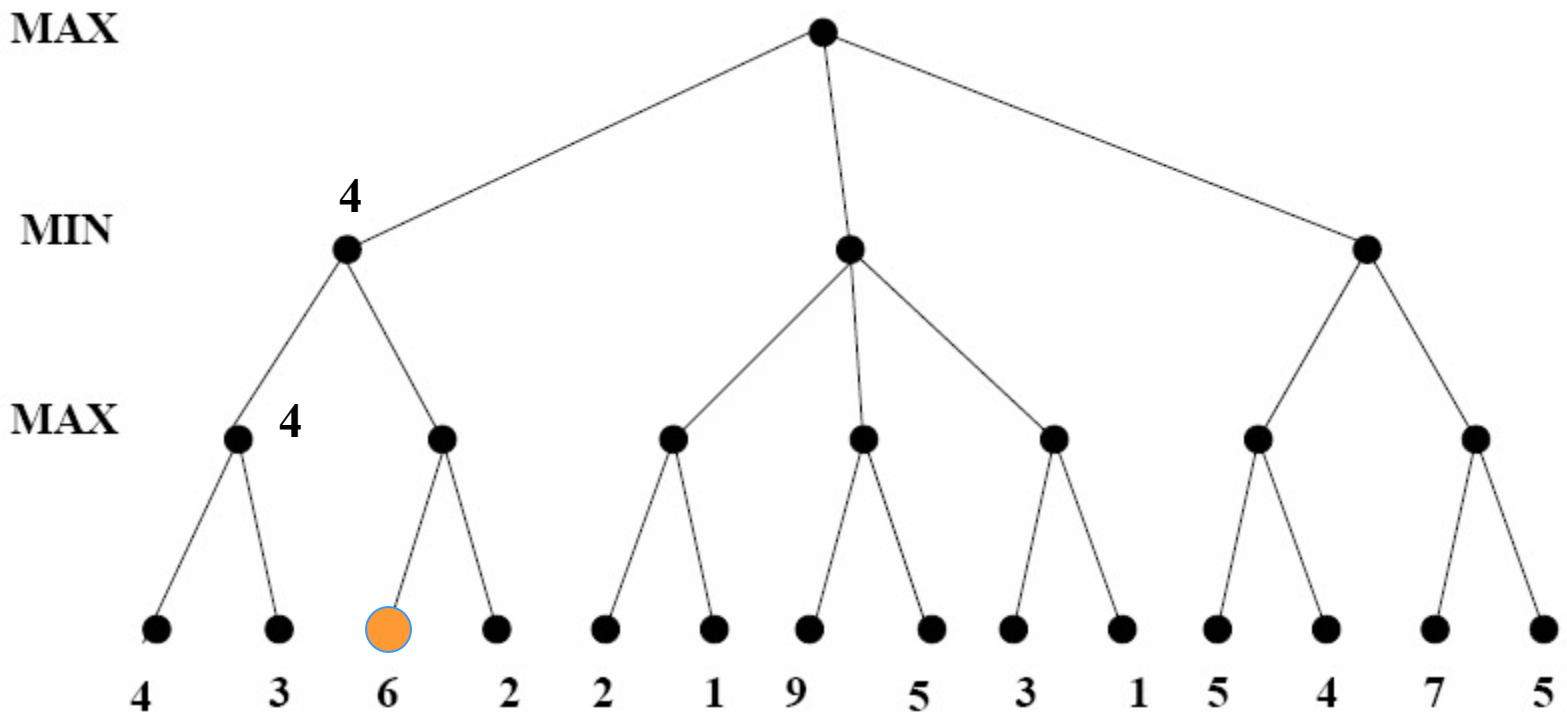


- Poda de los hijos de MAX

Podar los nodos a la derecha del nodo cuyo v_i es mayor o igual que β

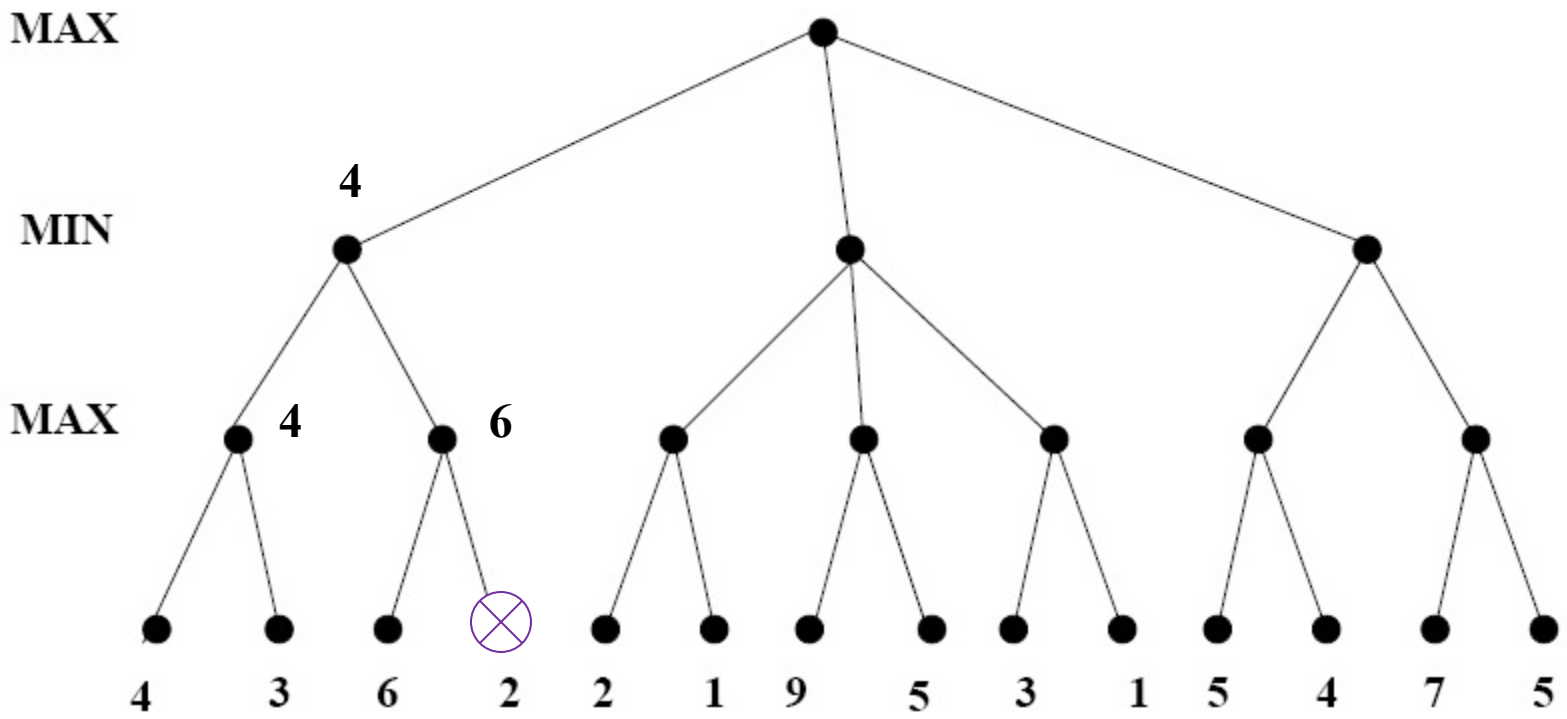
Juegos

Poda alfa-beta



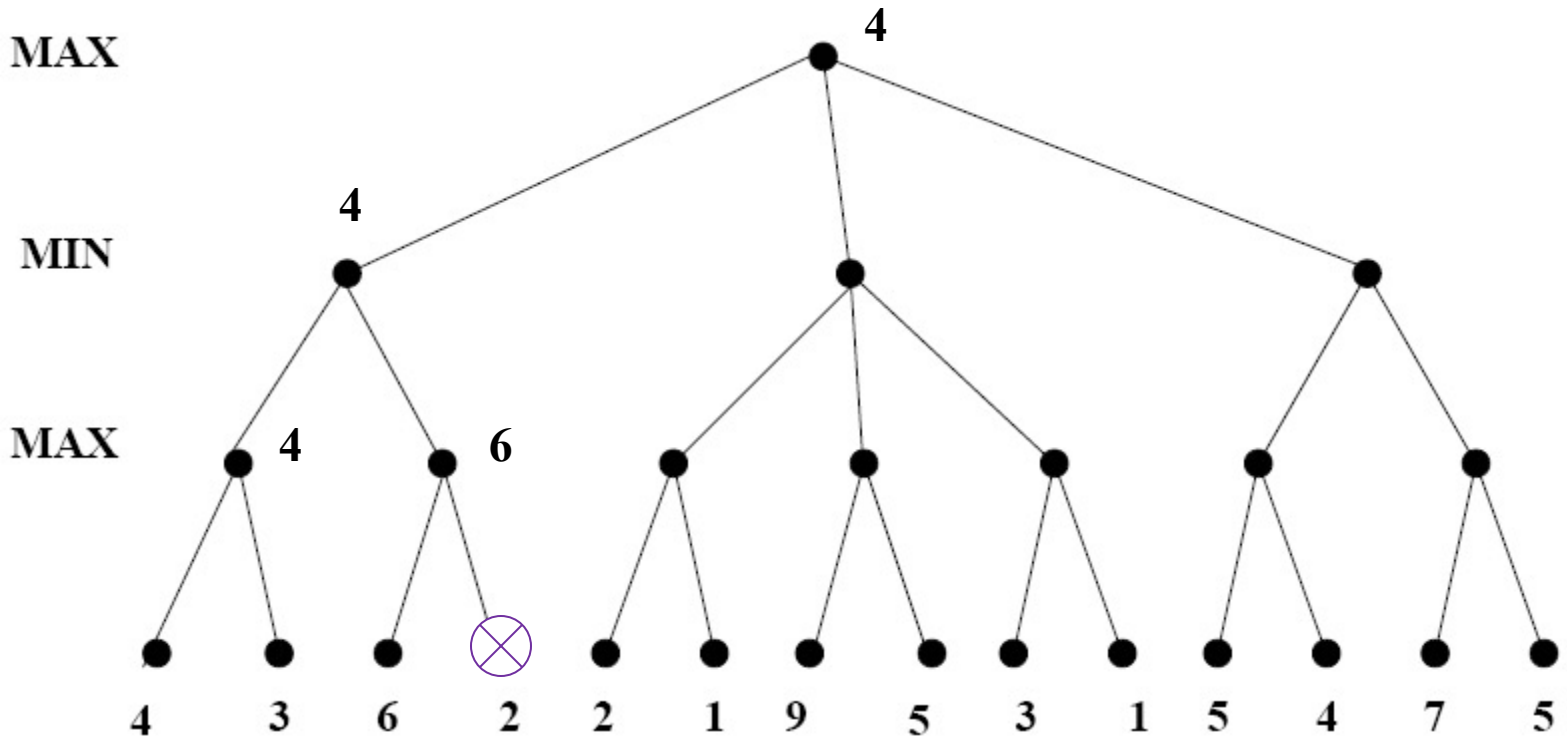
Juegos

Poda alfa-beta



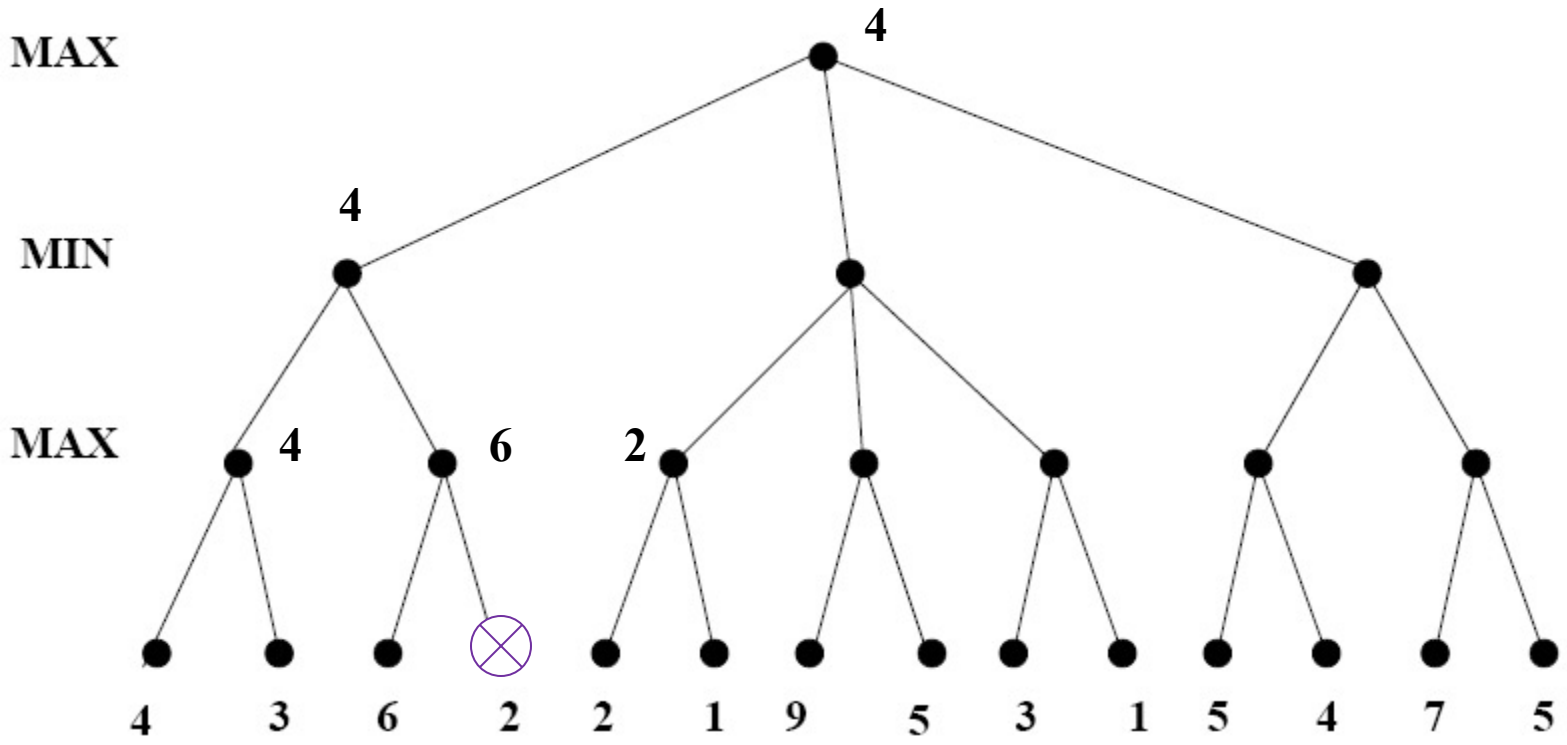
Juegos

Poda alfa-beta



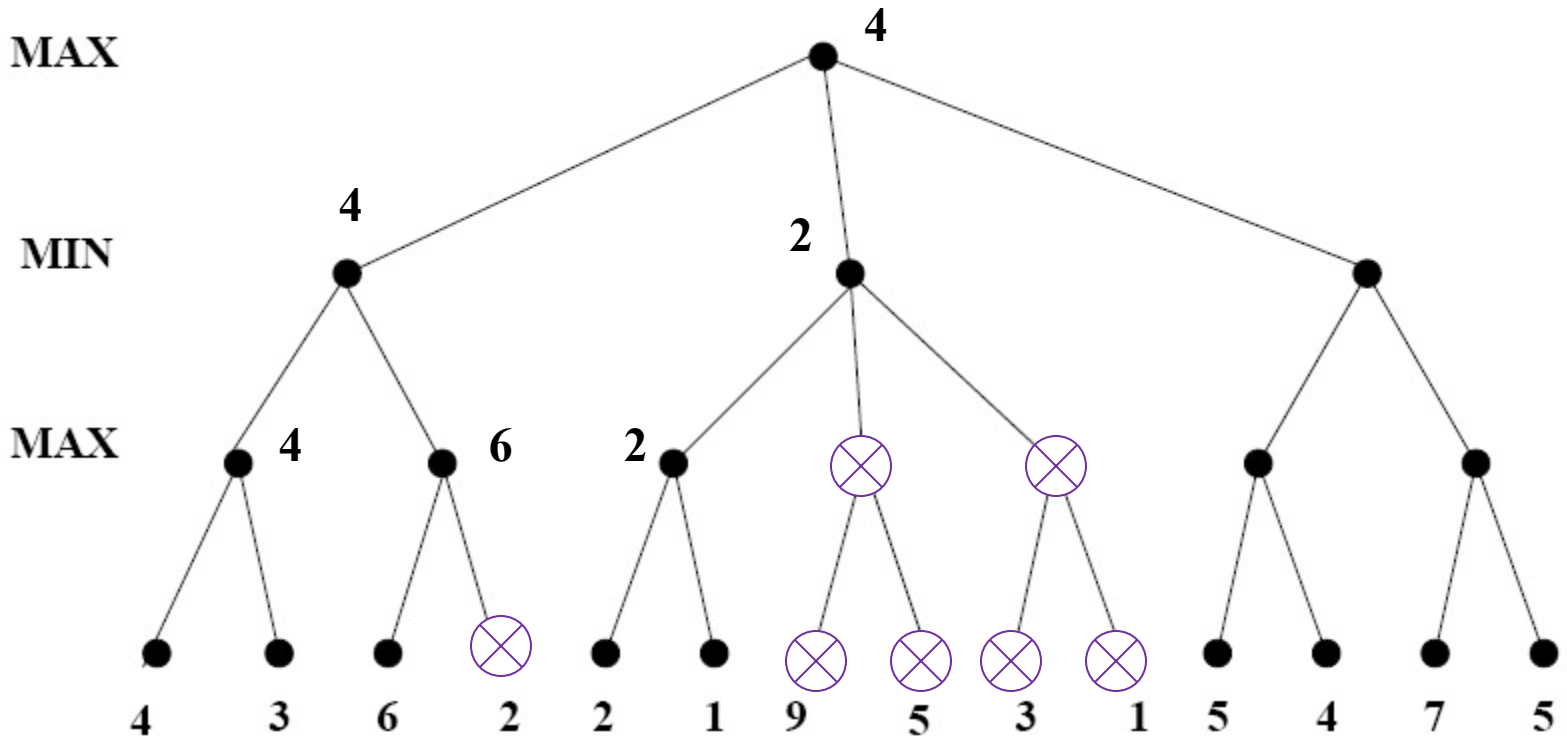
Juegos

Poda alfa-beta



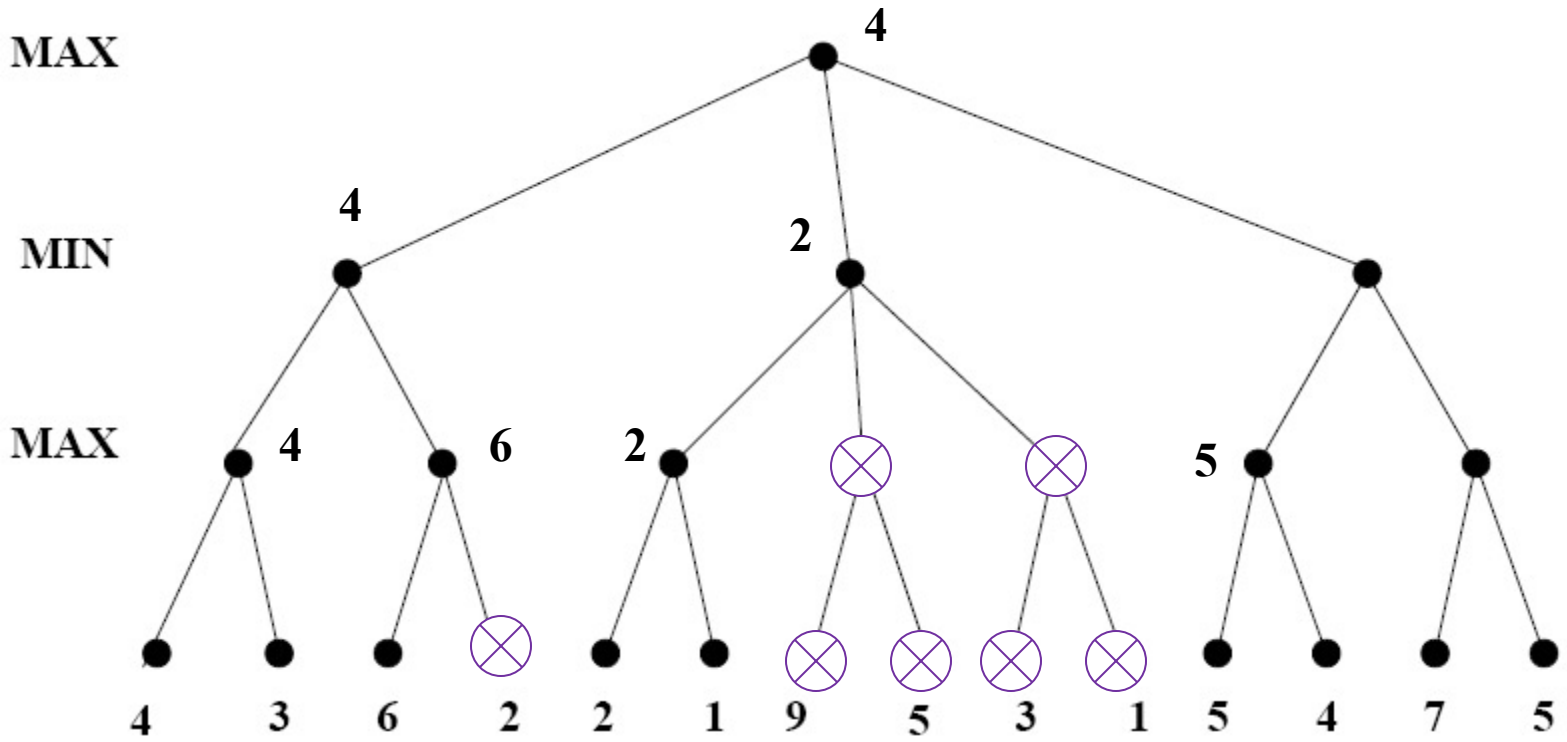
Juegos

Poda alfa-beta



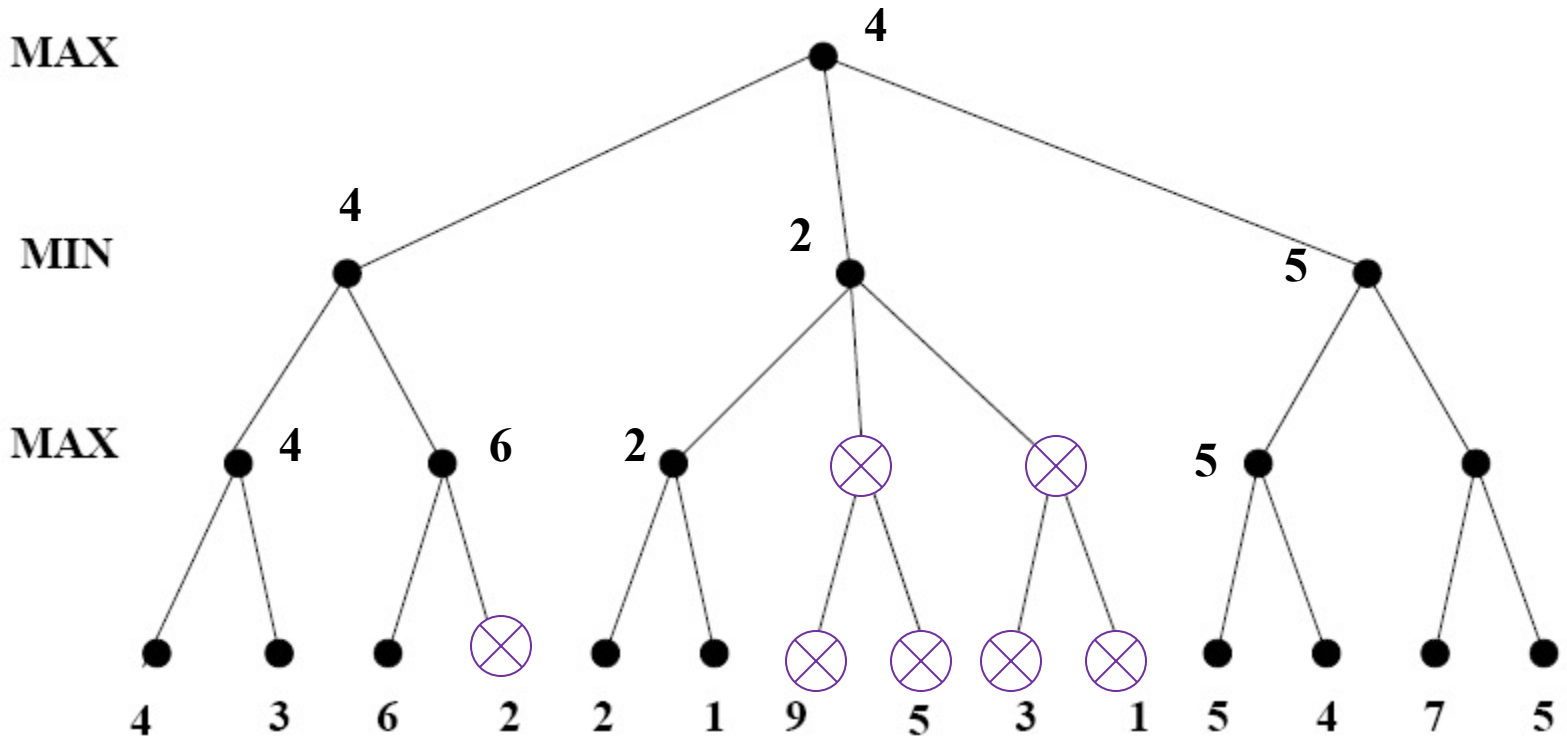
Juegos

Poda alfa-beta



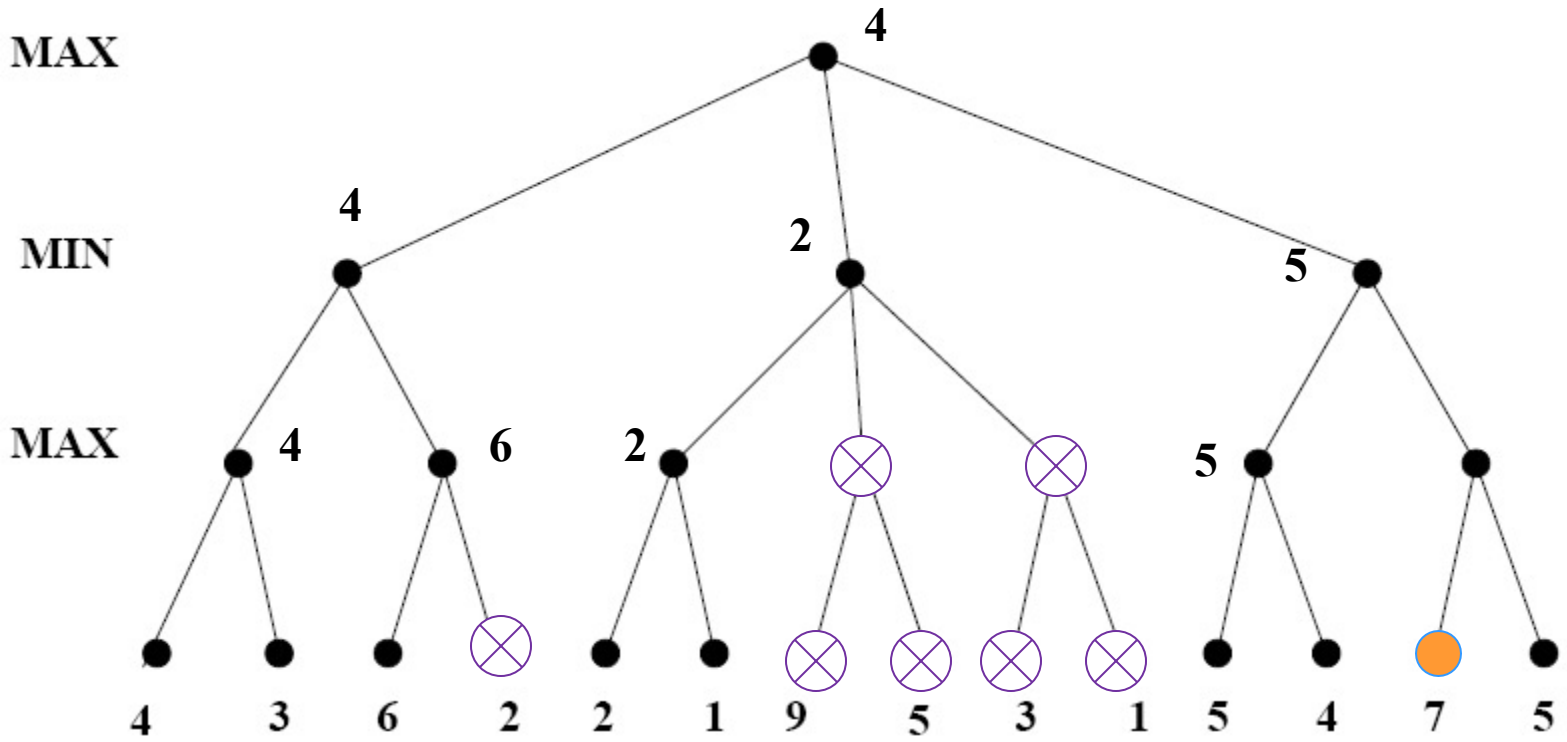
Juegos

Poda alfa-beta



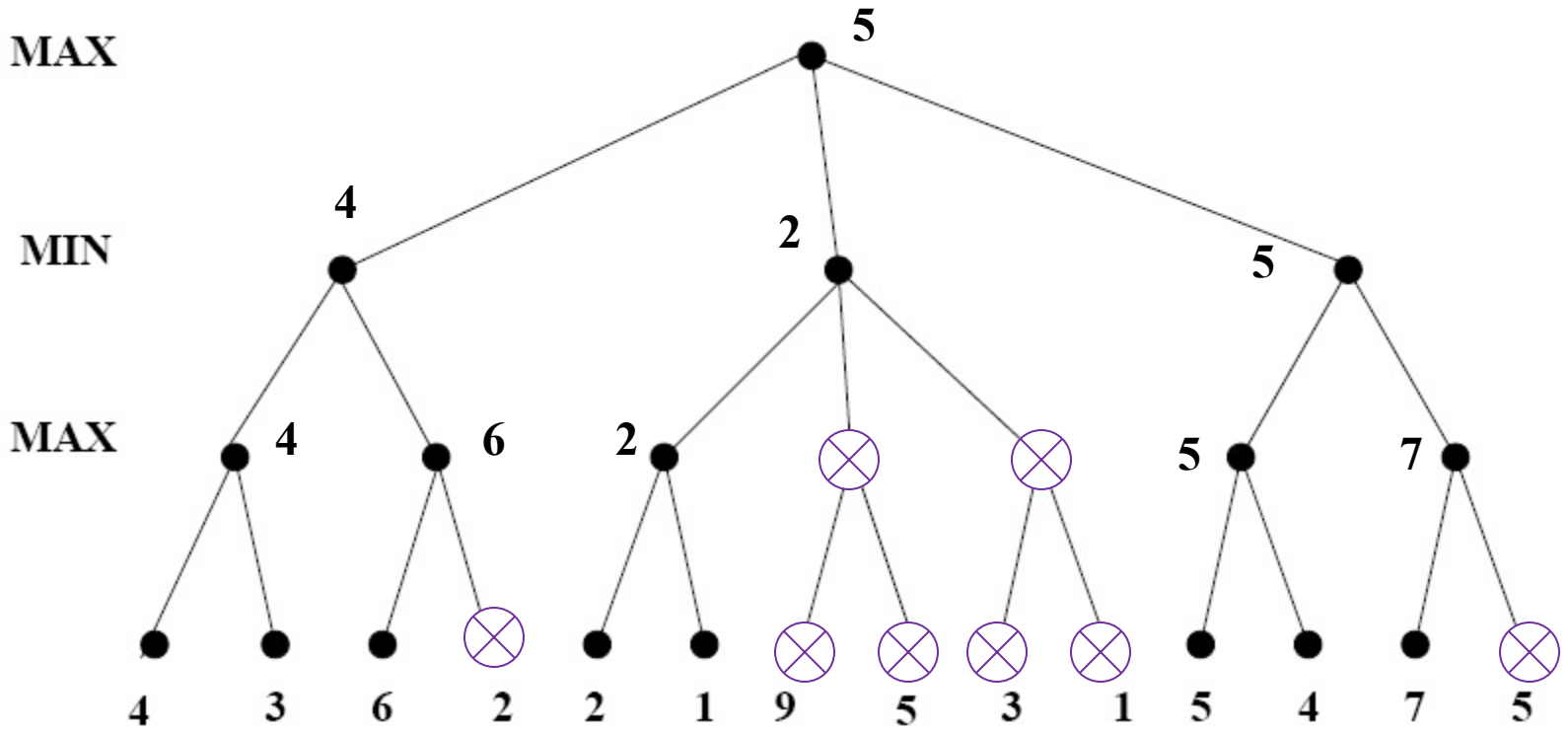
Juegos

Poda alfa-beta



Juegos

Poda alfa-beta

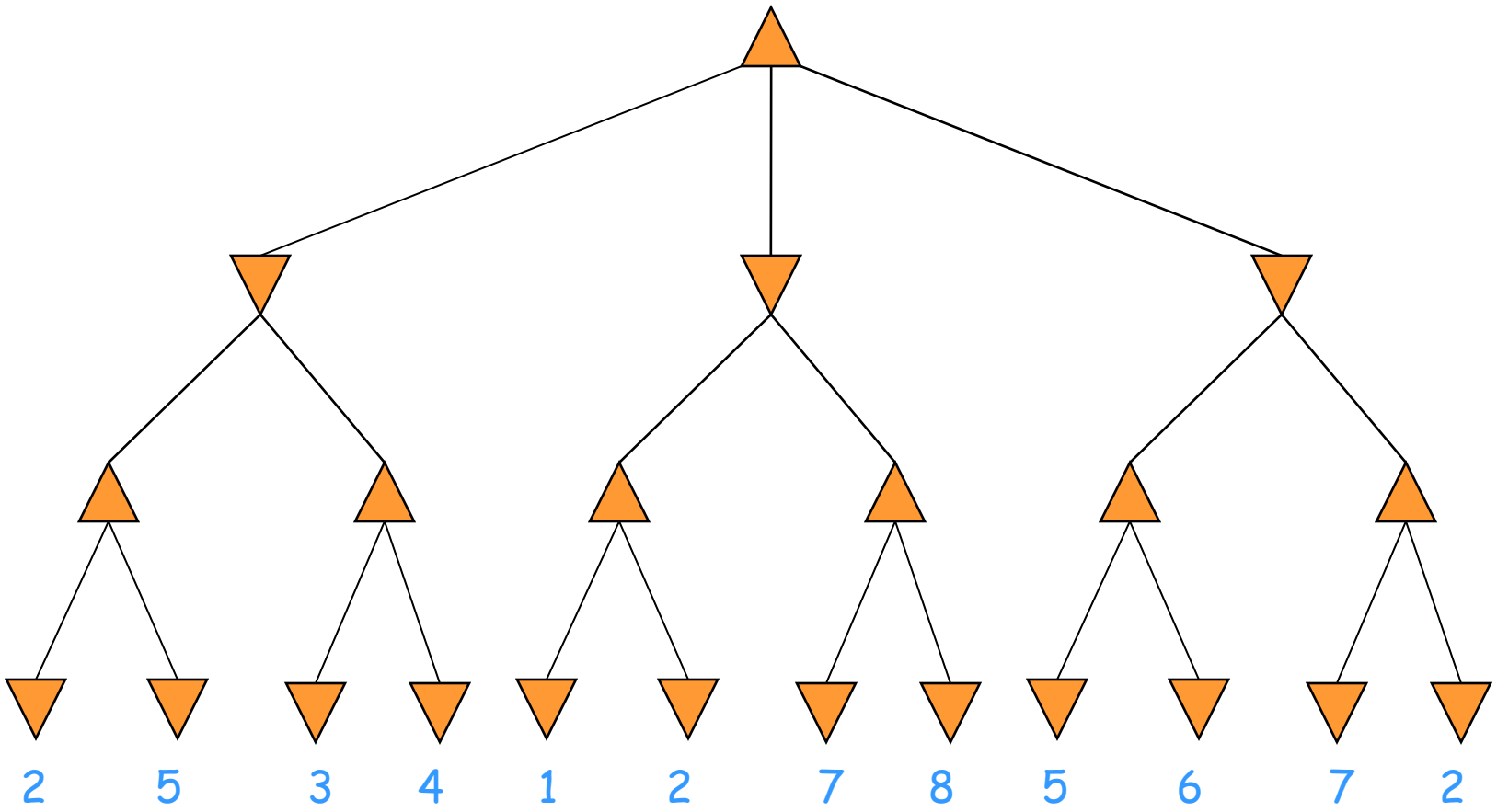


Poda alfa-beta

Los dos parámetros **alfa** y **beta** describen los límites sobre los valores que aparecen a lo largo del camino:

- α: el valor de la mejor opción (el más alto) que se ha encontrado hasta el momento en cualquier punto del camino, para MAX
- β: el valor de la mejor opción (el más bajo) que se ha encontrado hasta el momento en cualquier punto del camino, para MIN

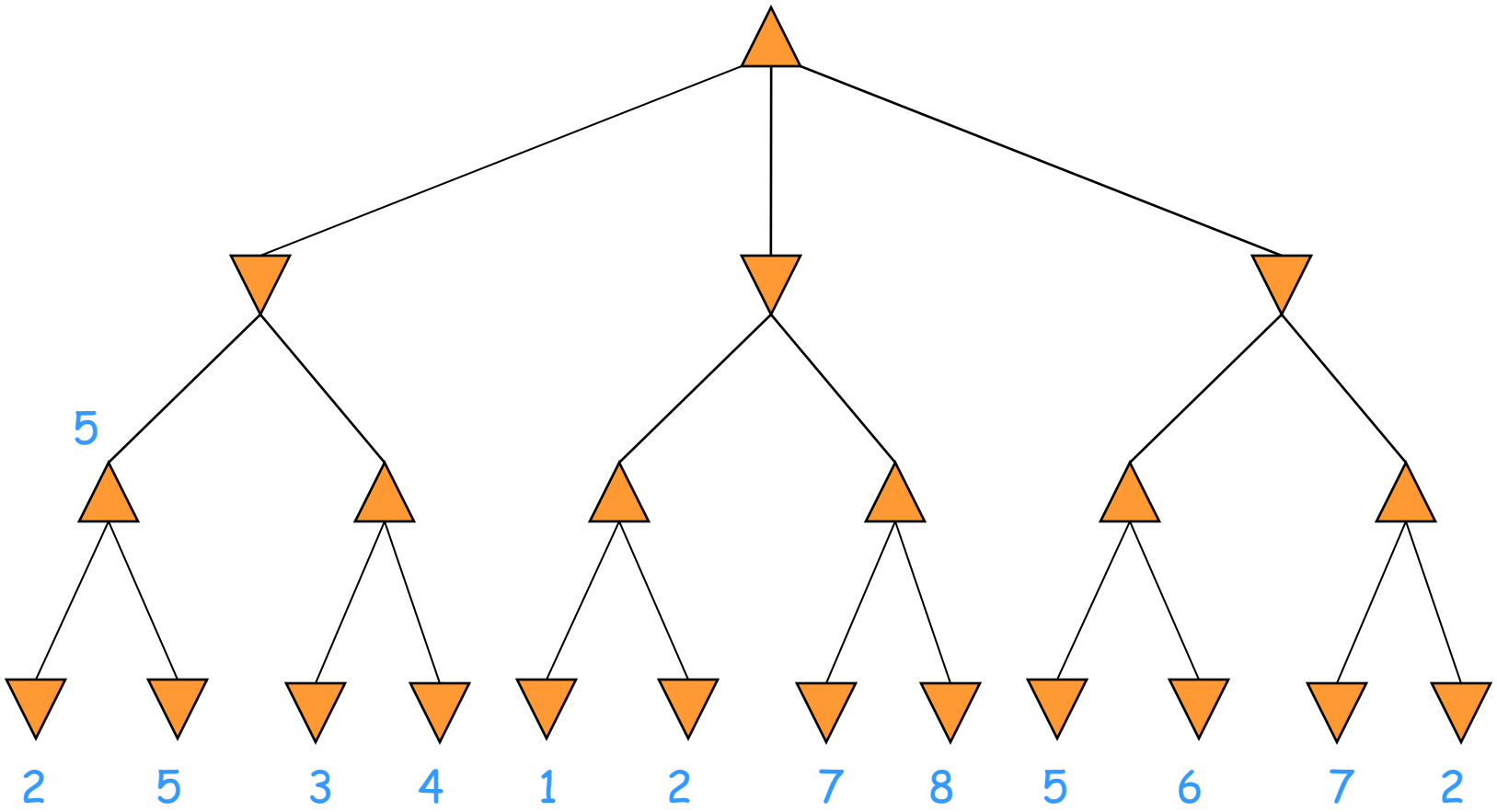
Juegos



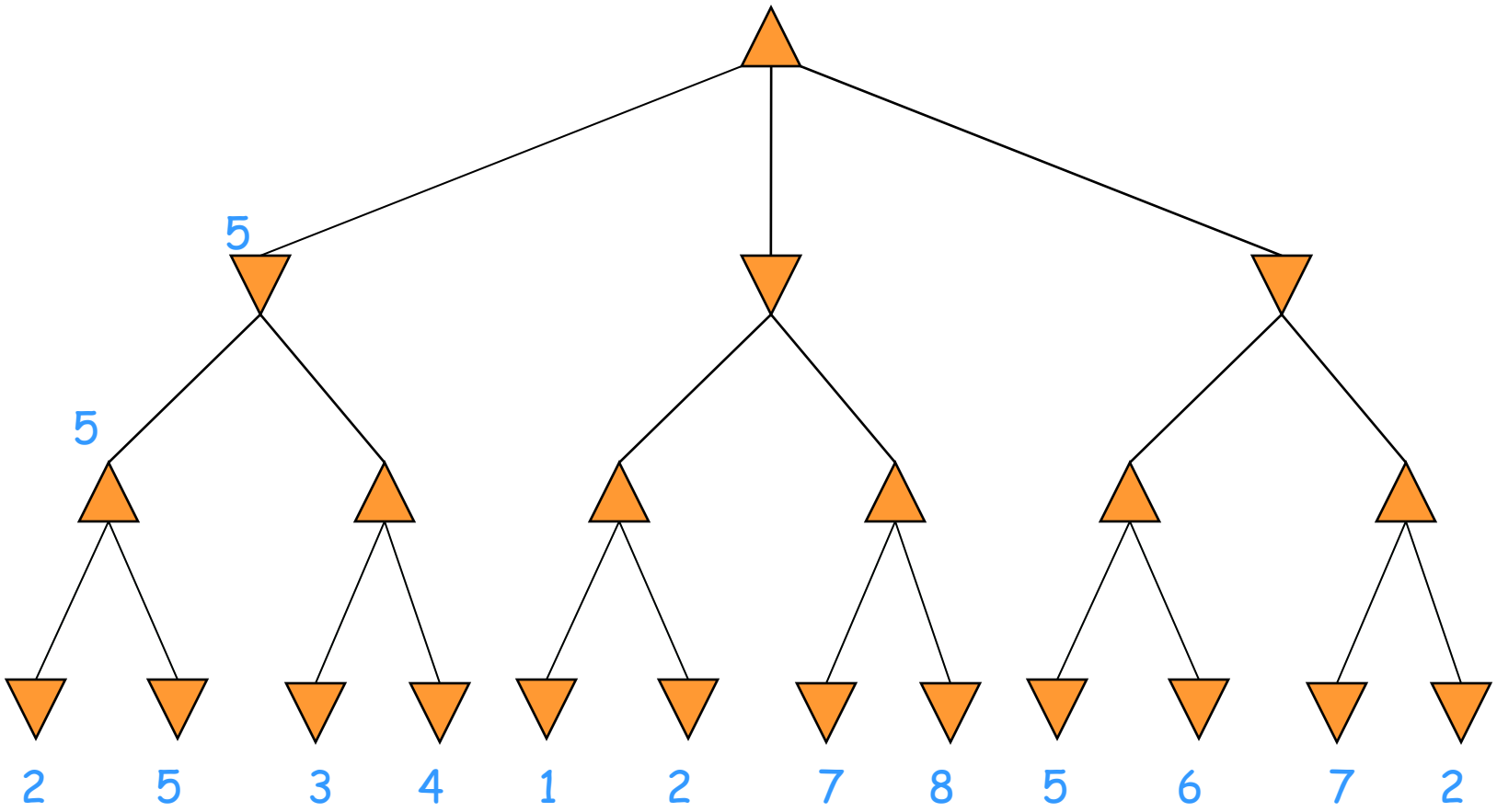
Indique qué nodos se podan

*

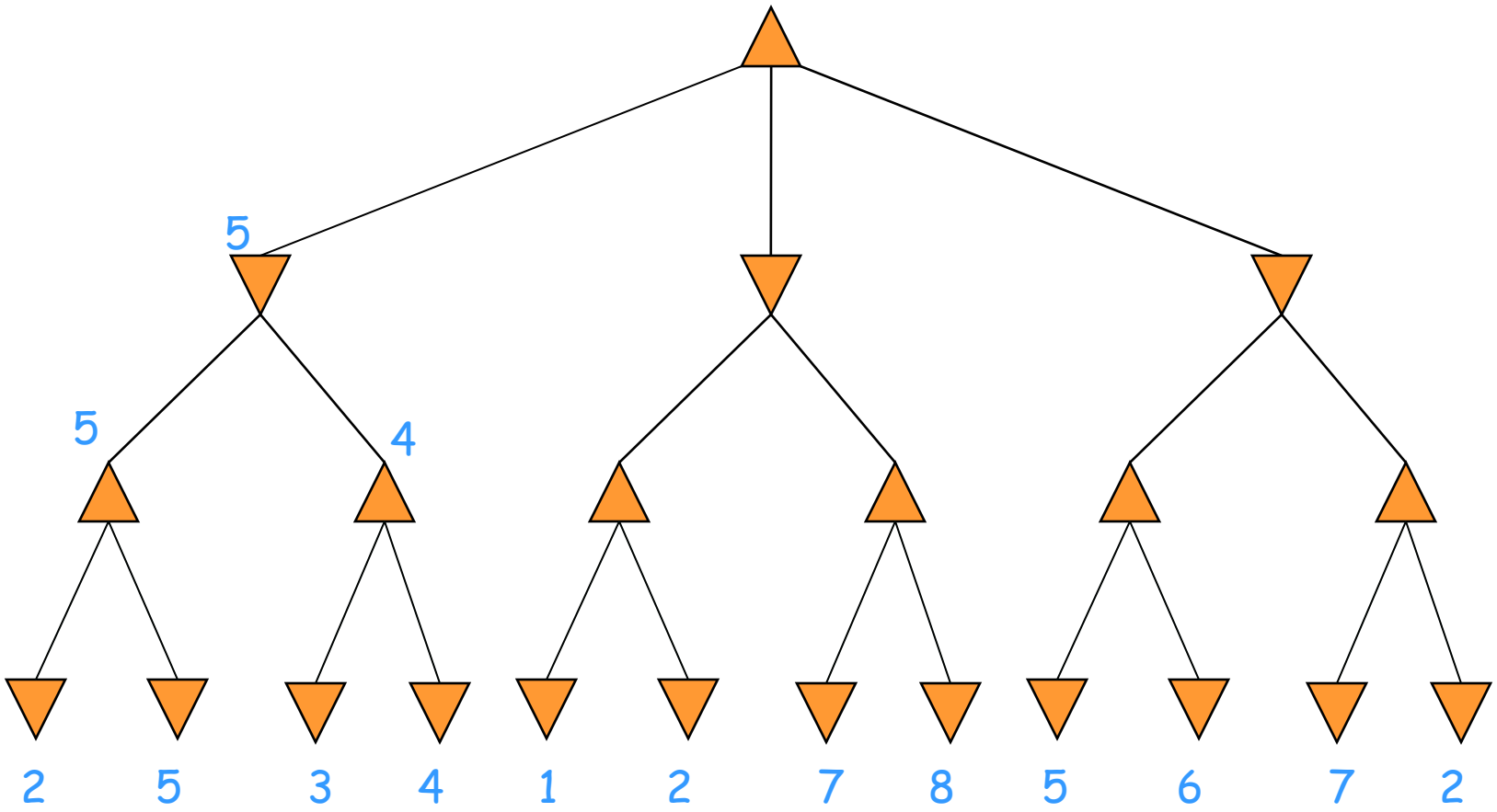
Juegos



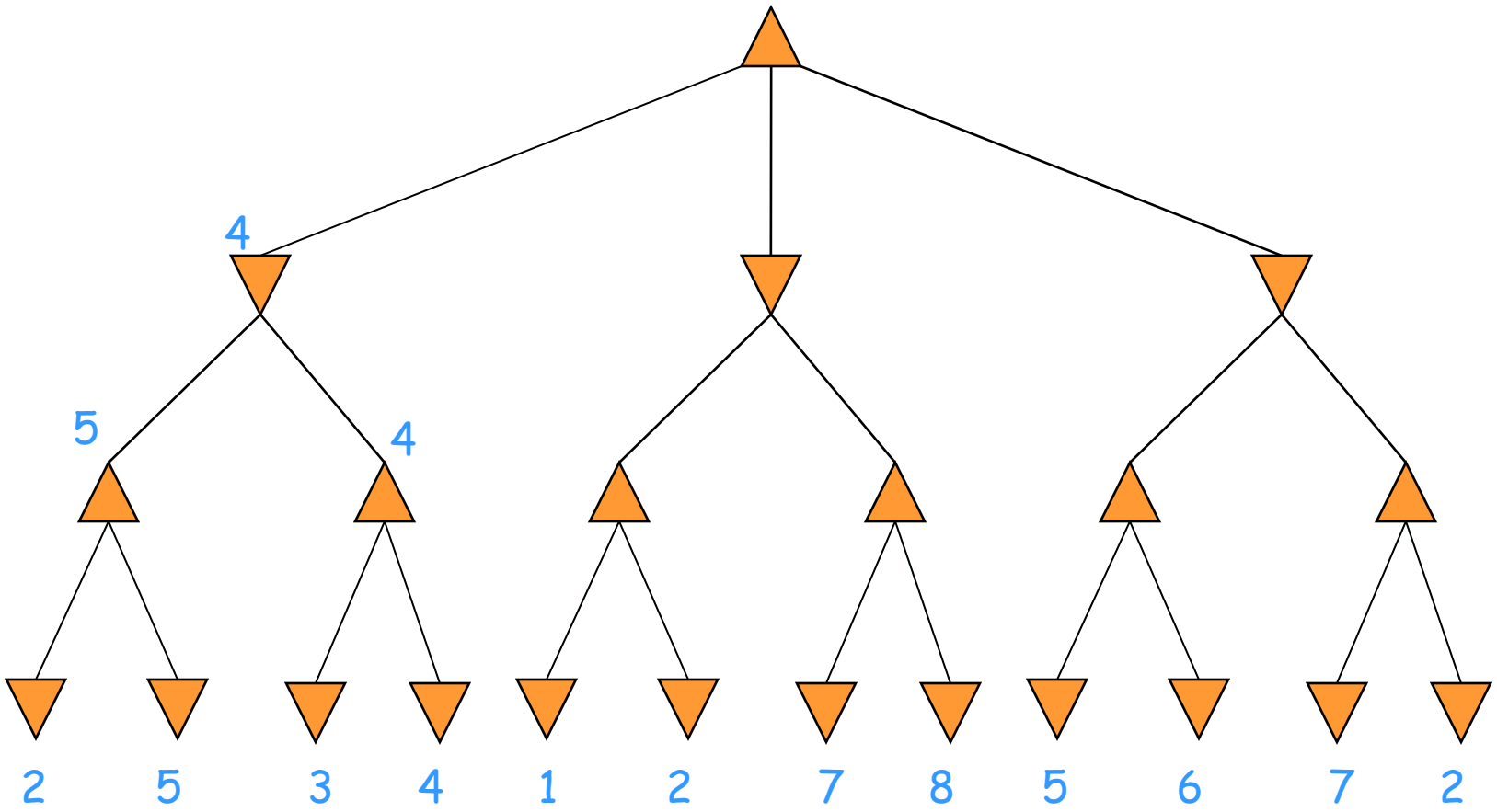
Juegos



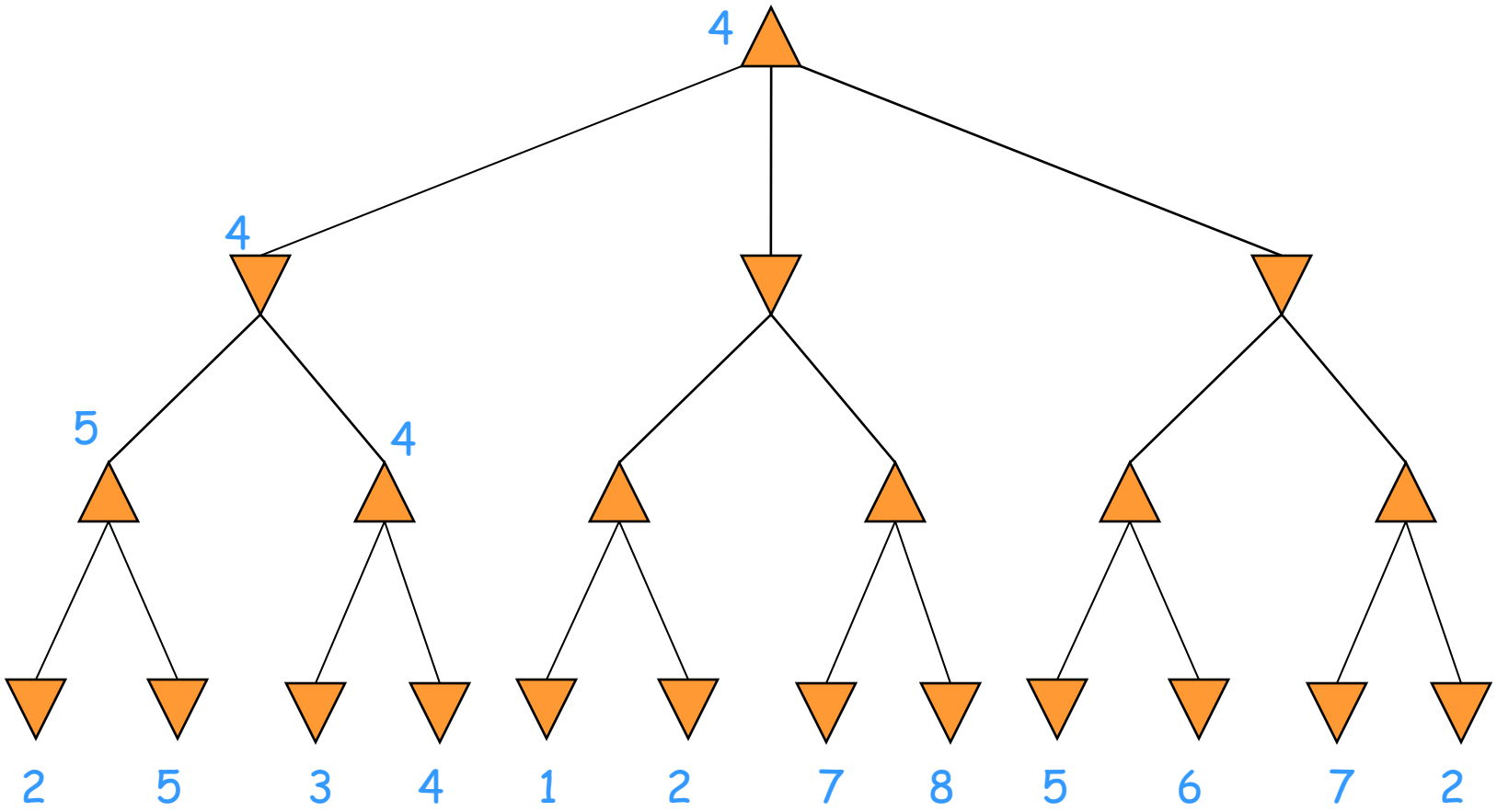
Juegos



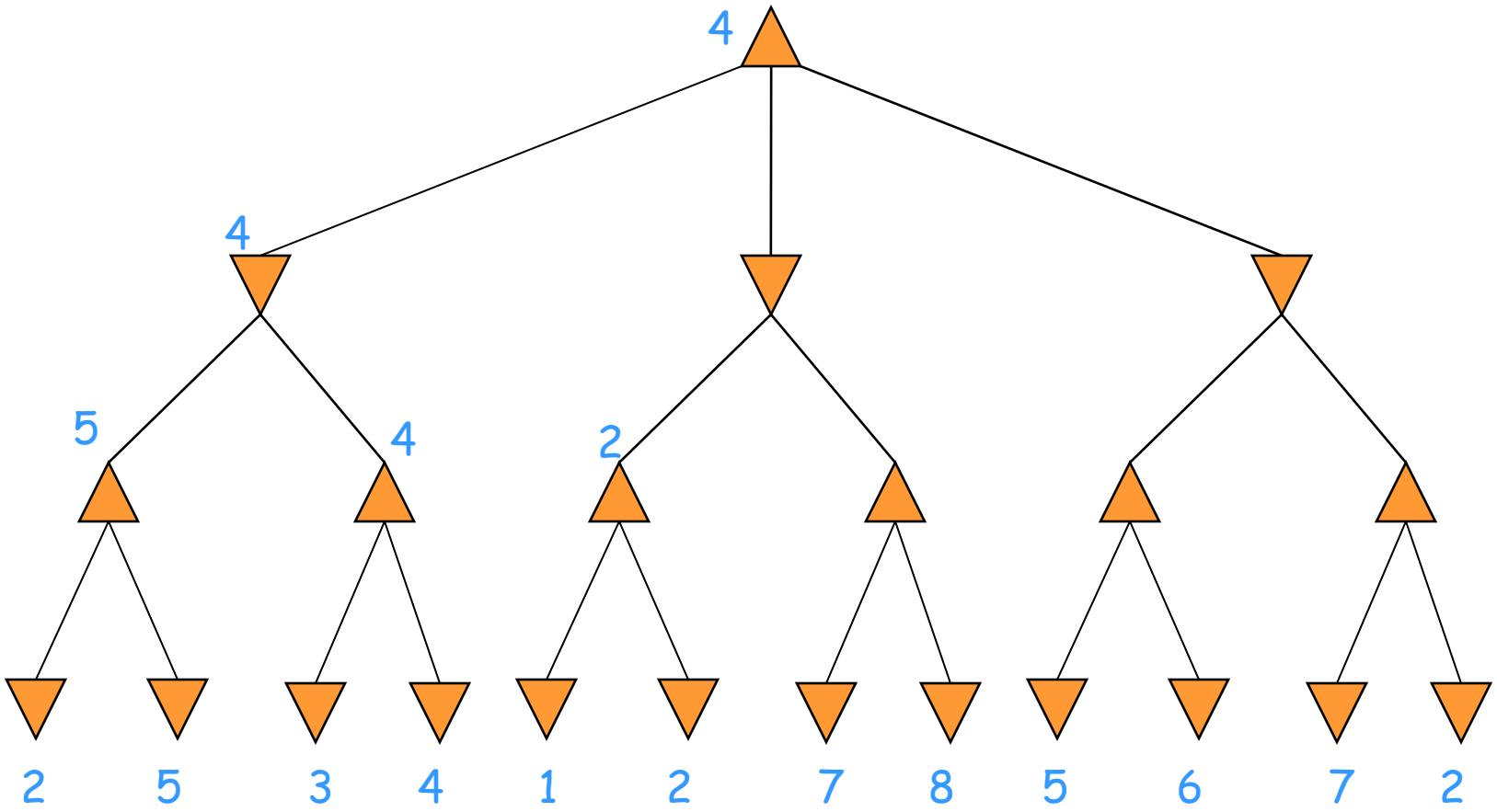
Juegos



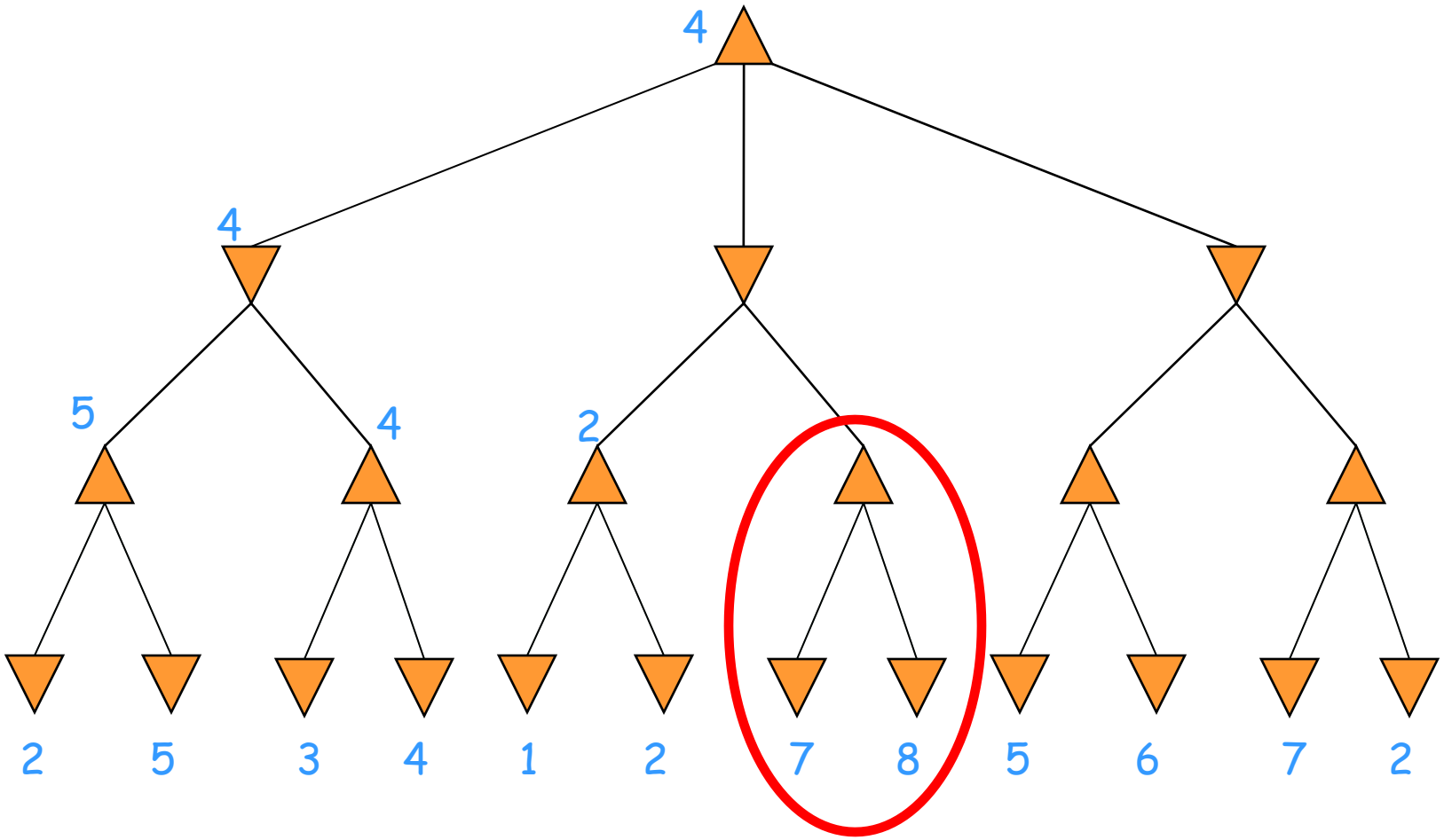
Juegos



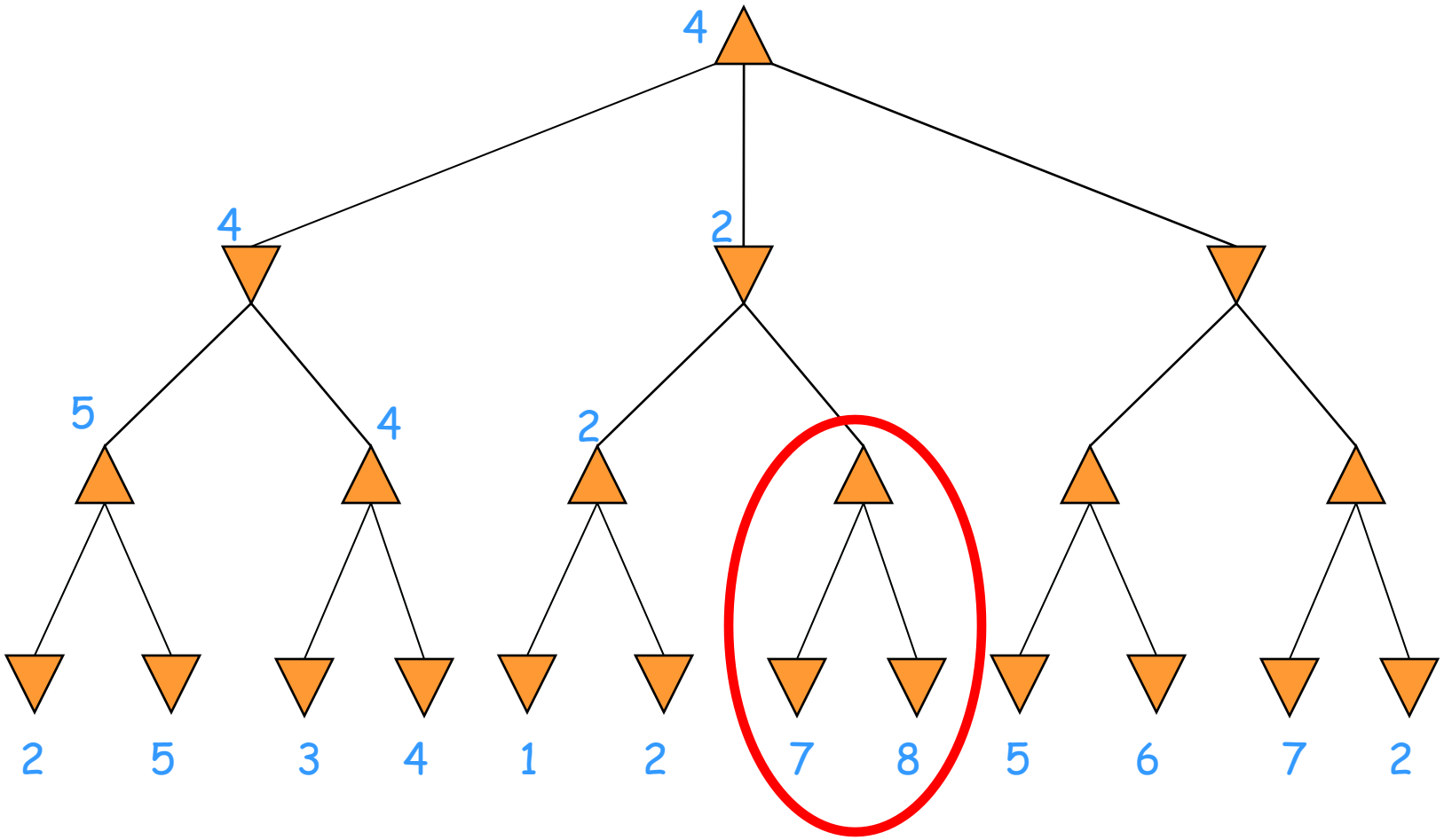
Juegos



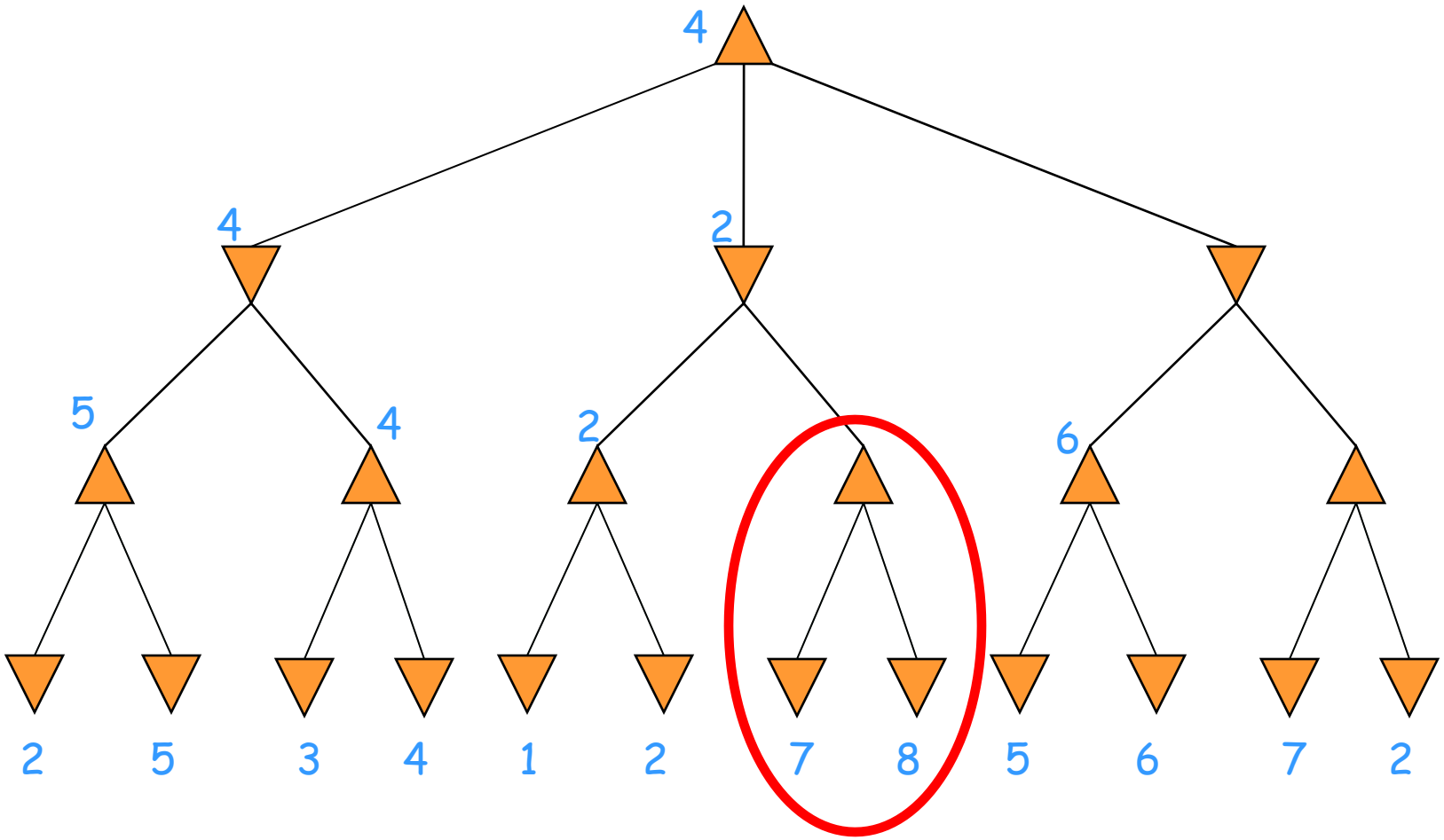
Juegos



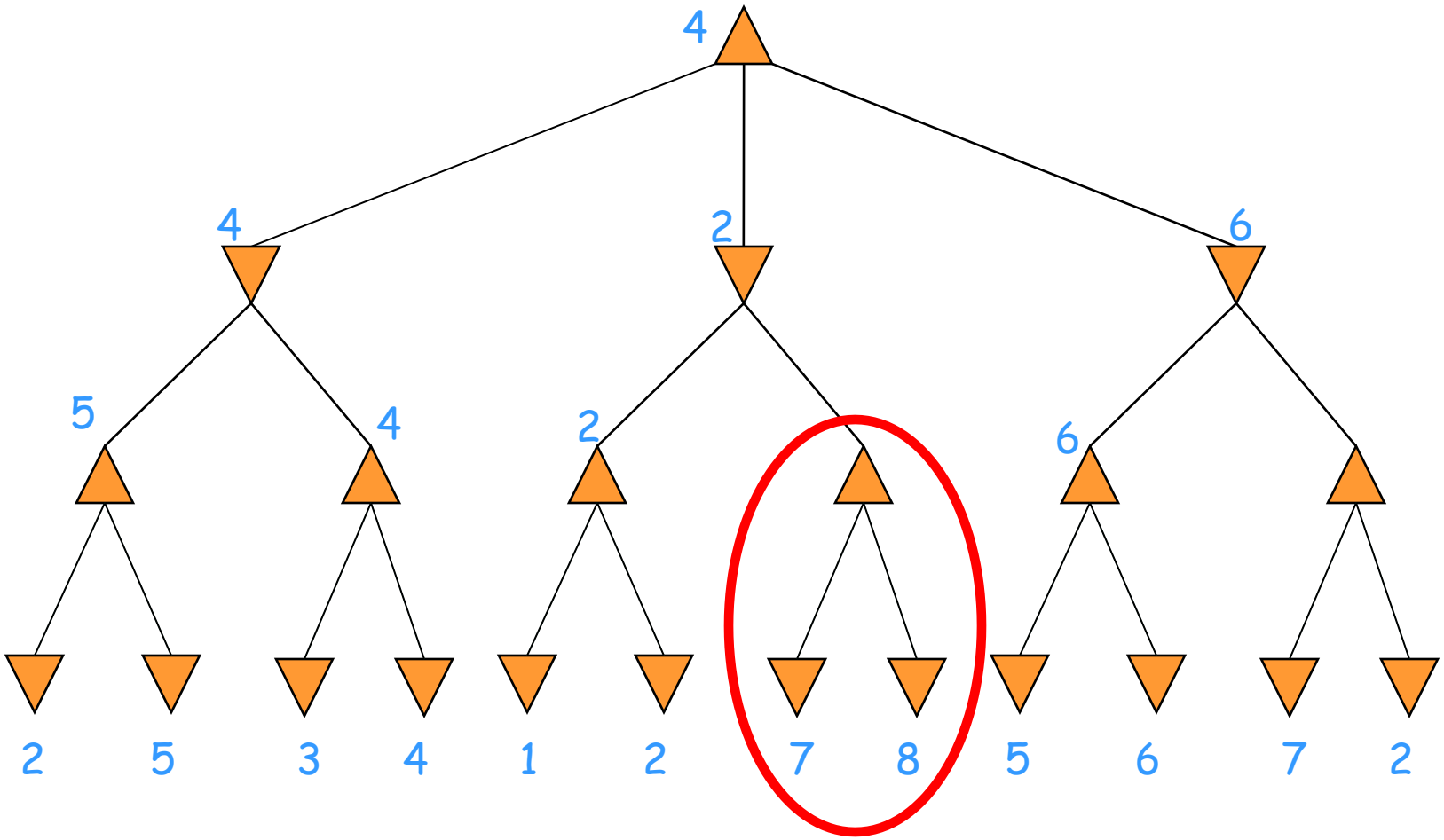
Juegos



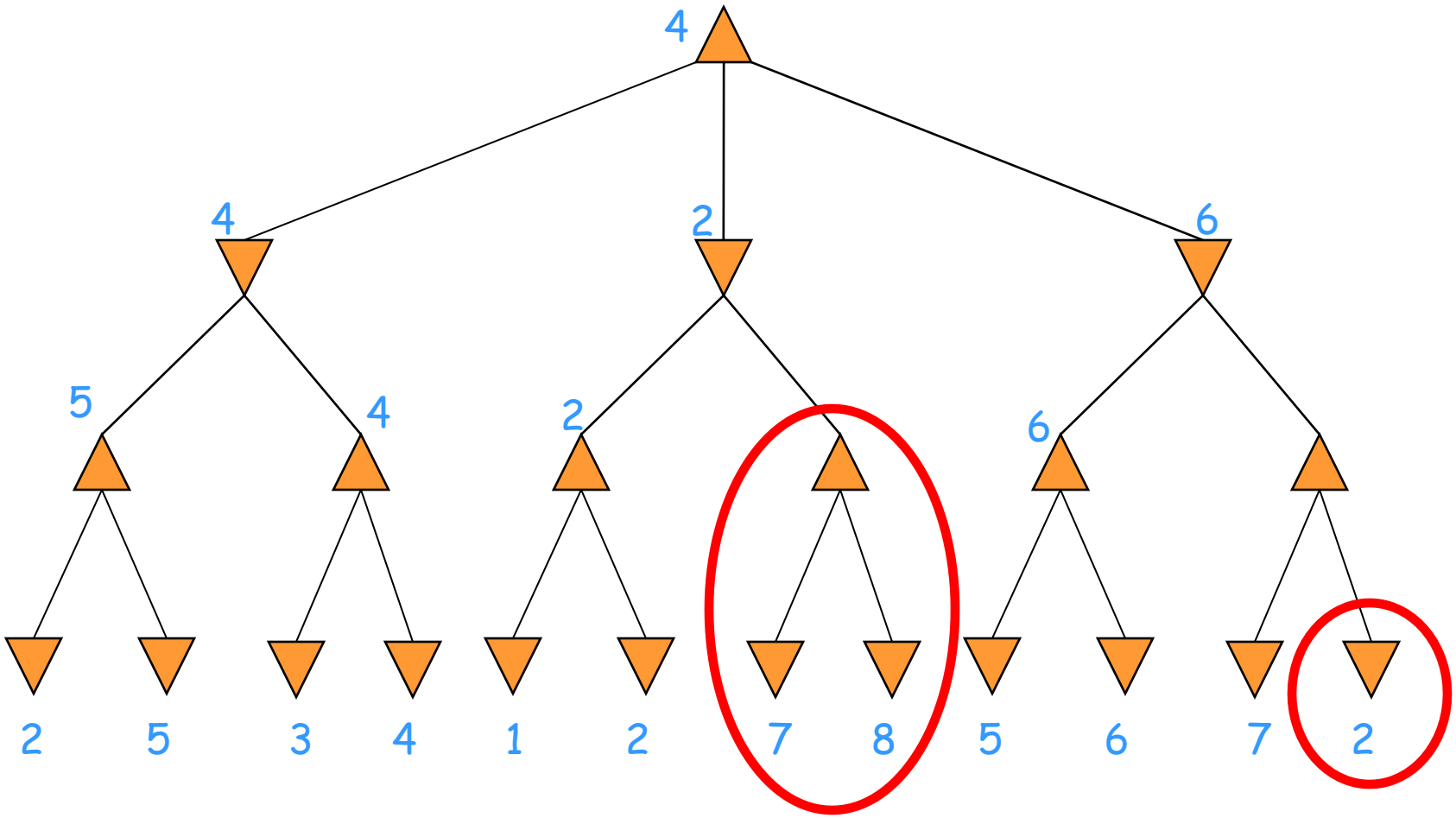
Juegos



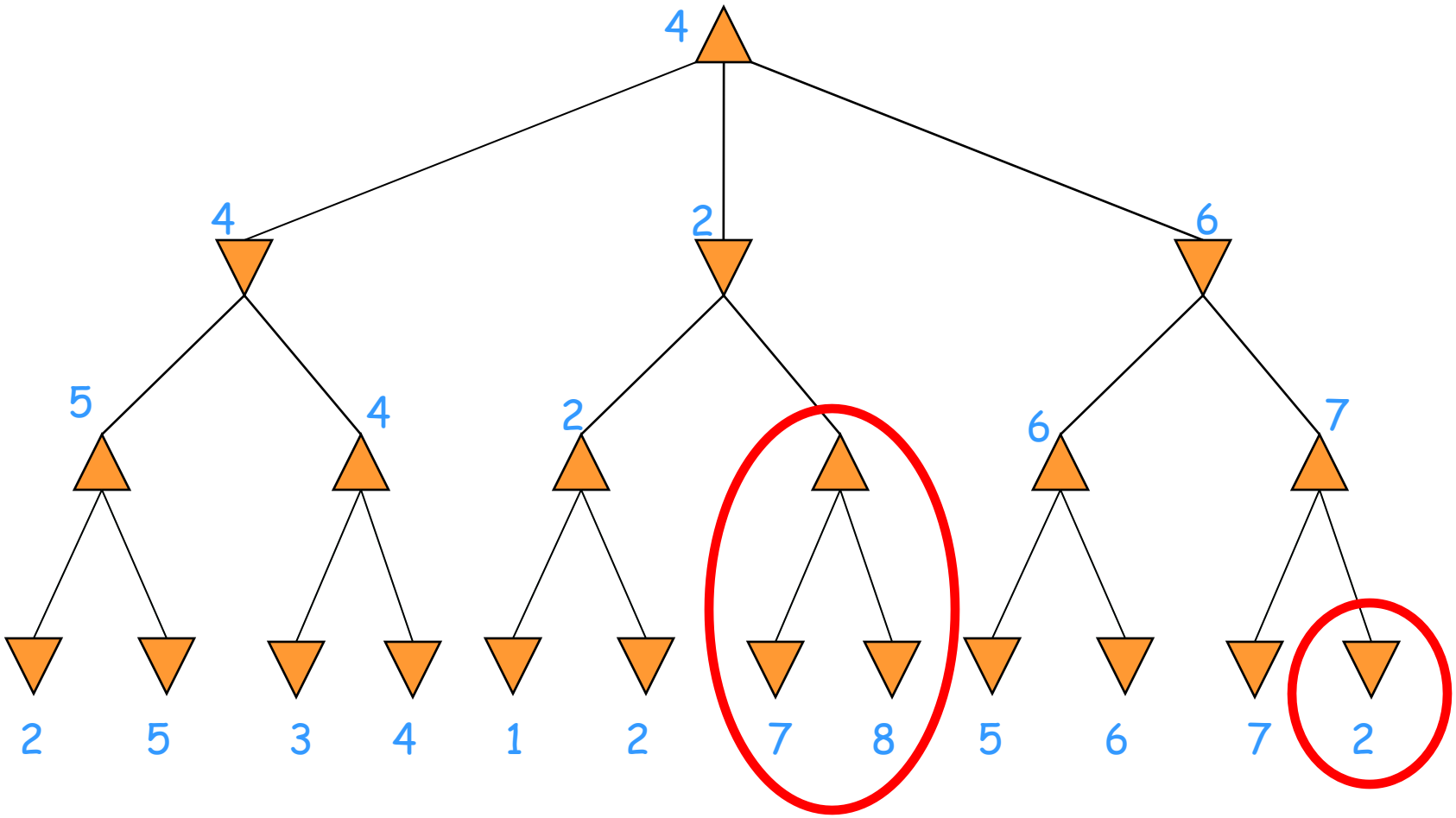
Juegos



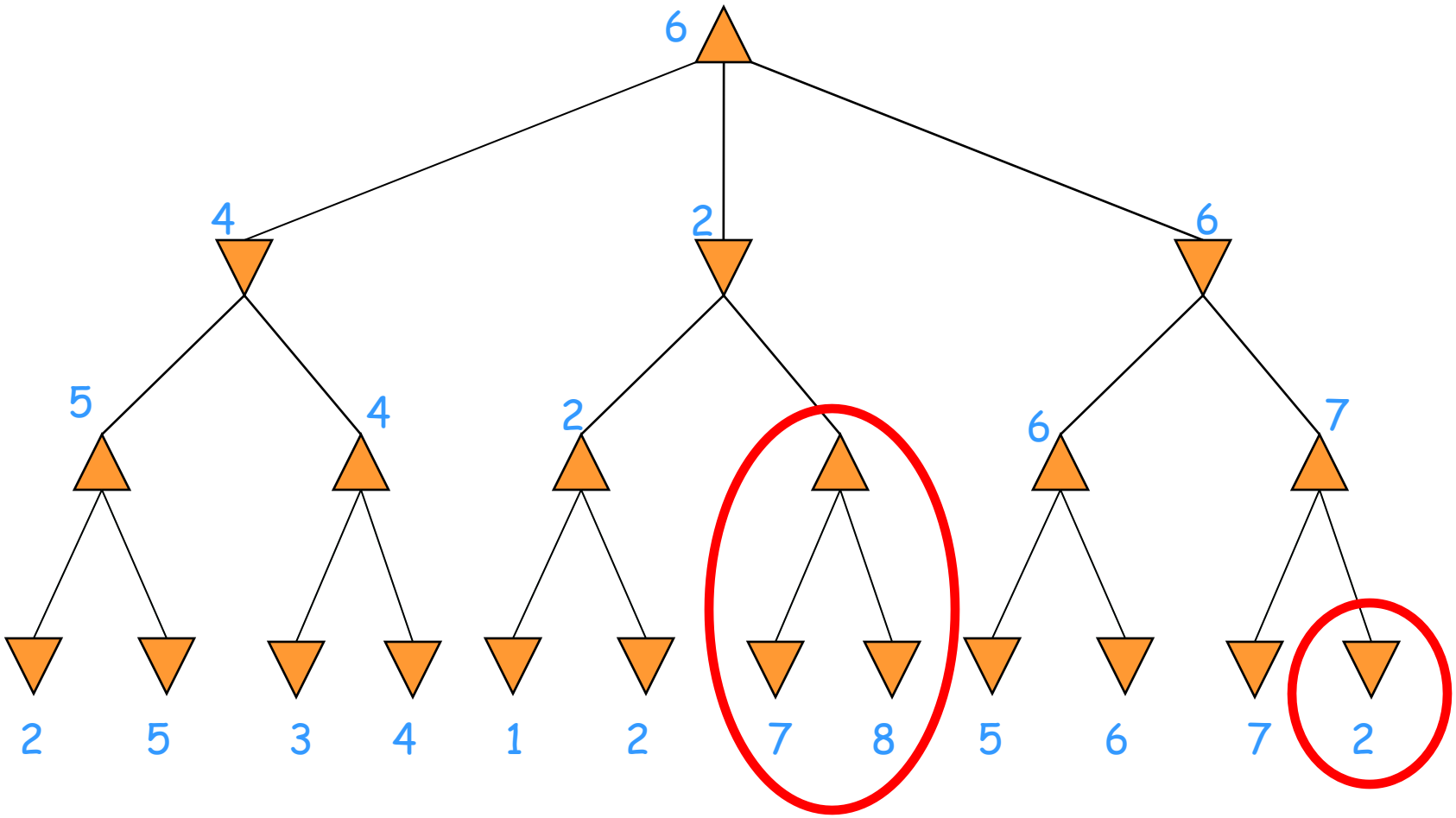
Juegos



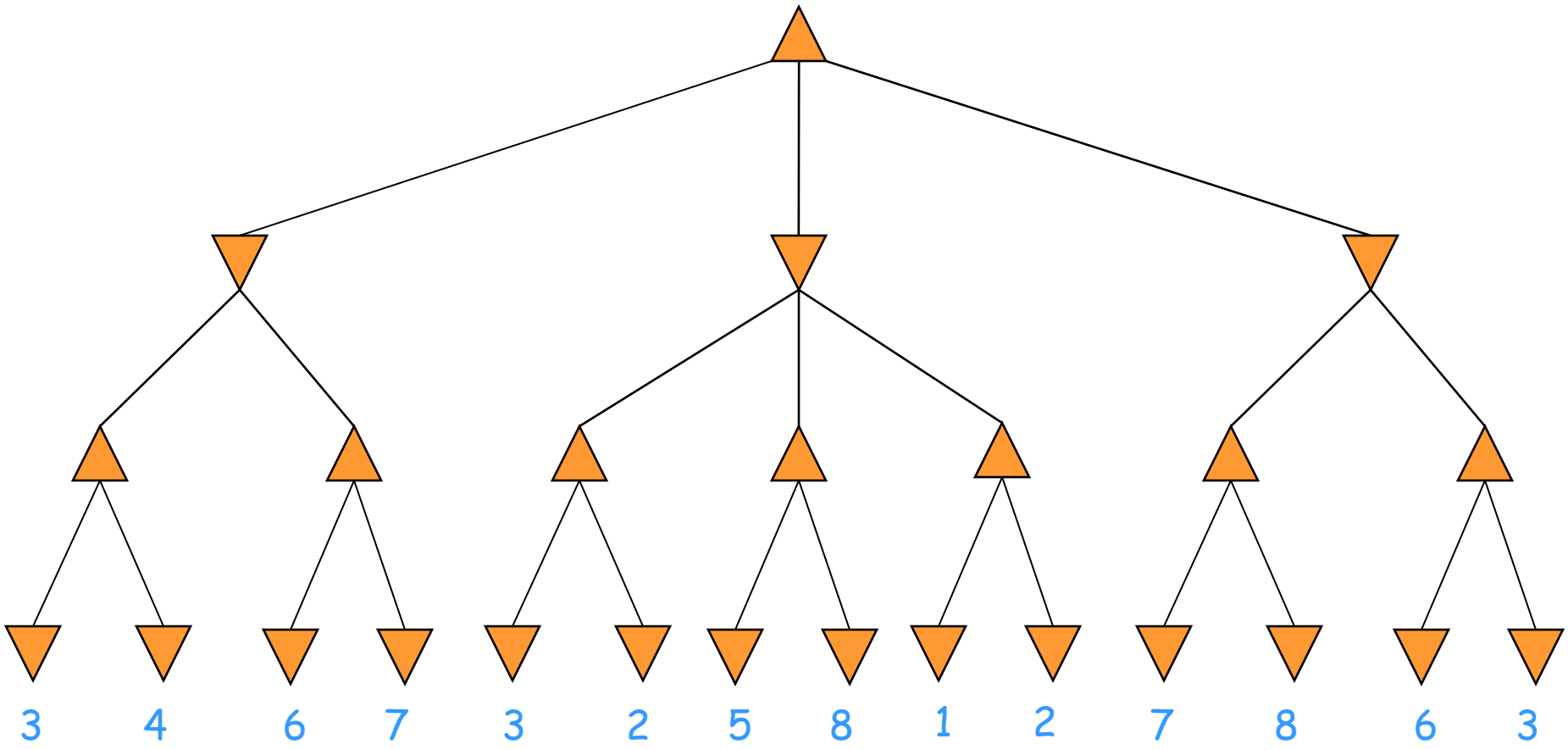
Juegos



Juegos

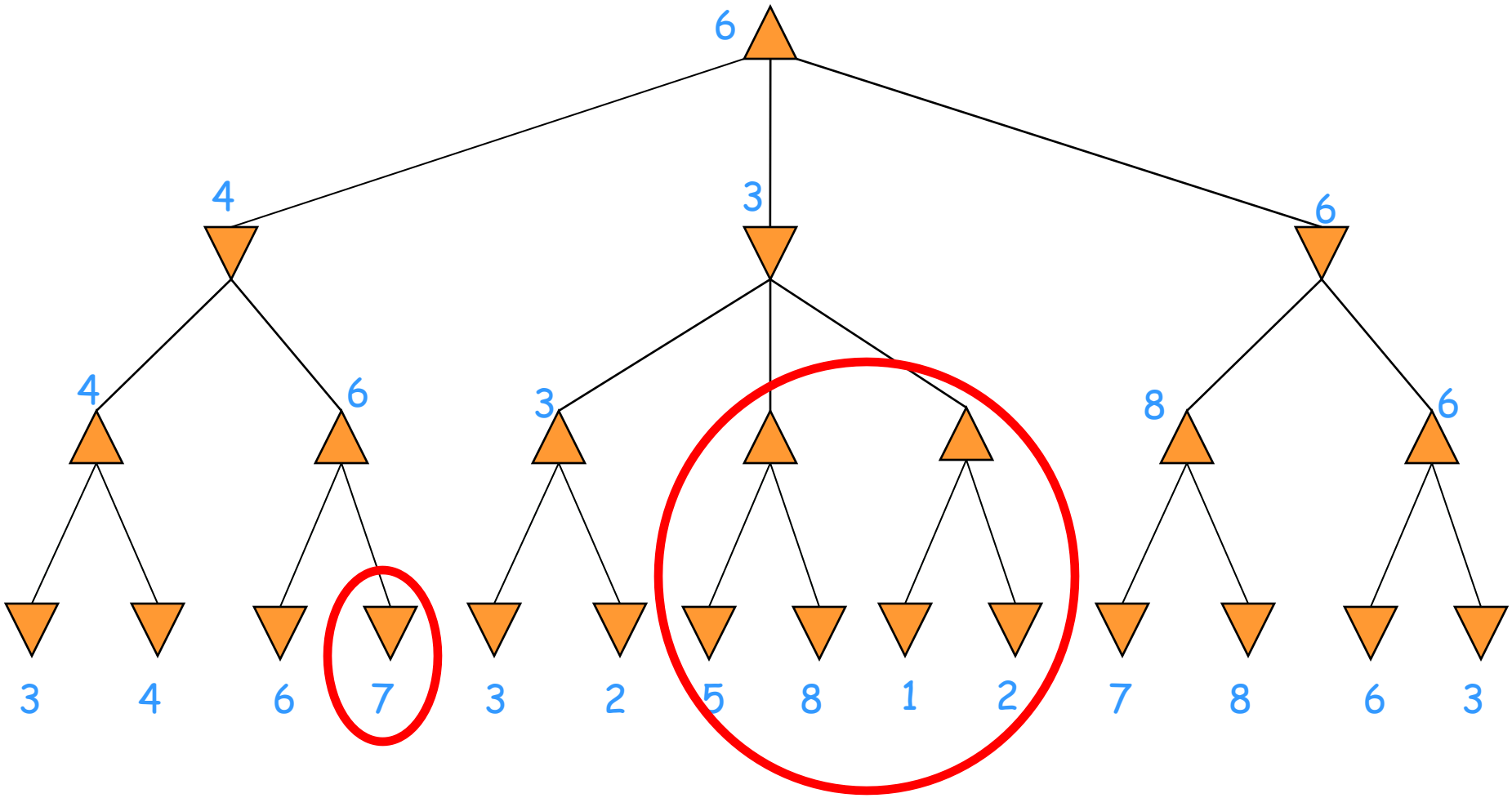


Juegos

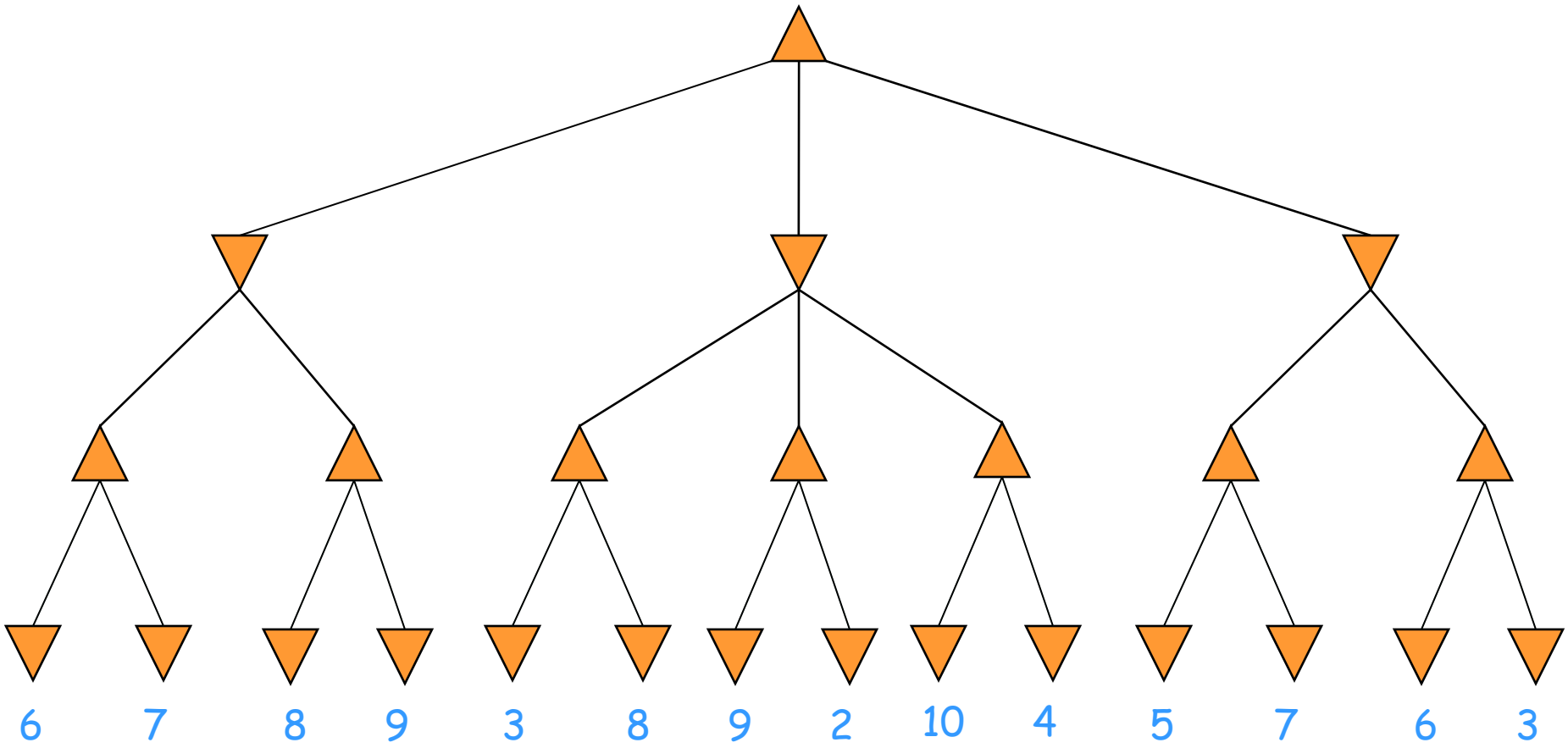


Indique qué nodos se podan

Juegos

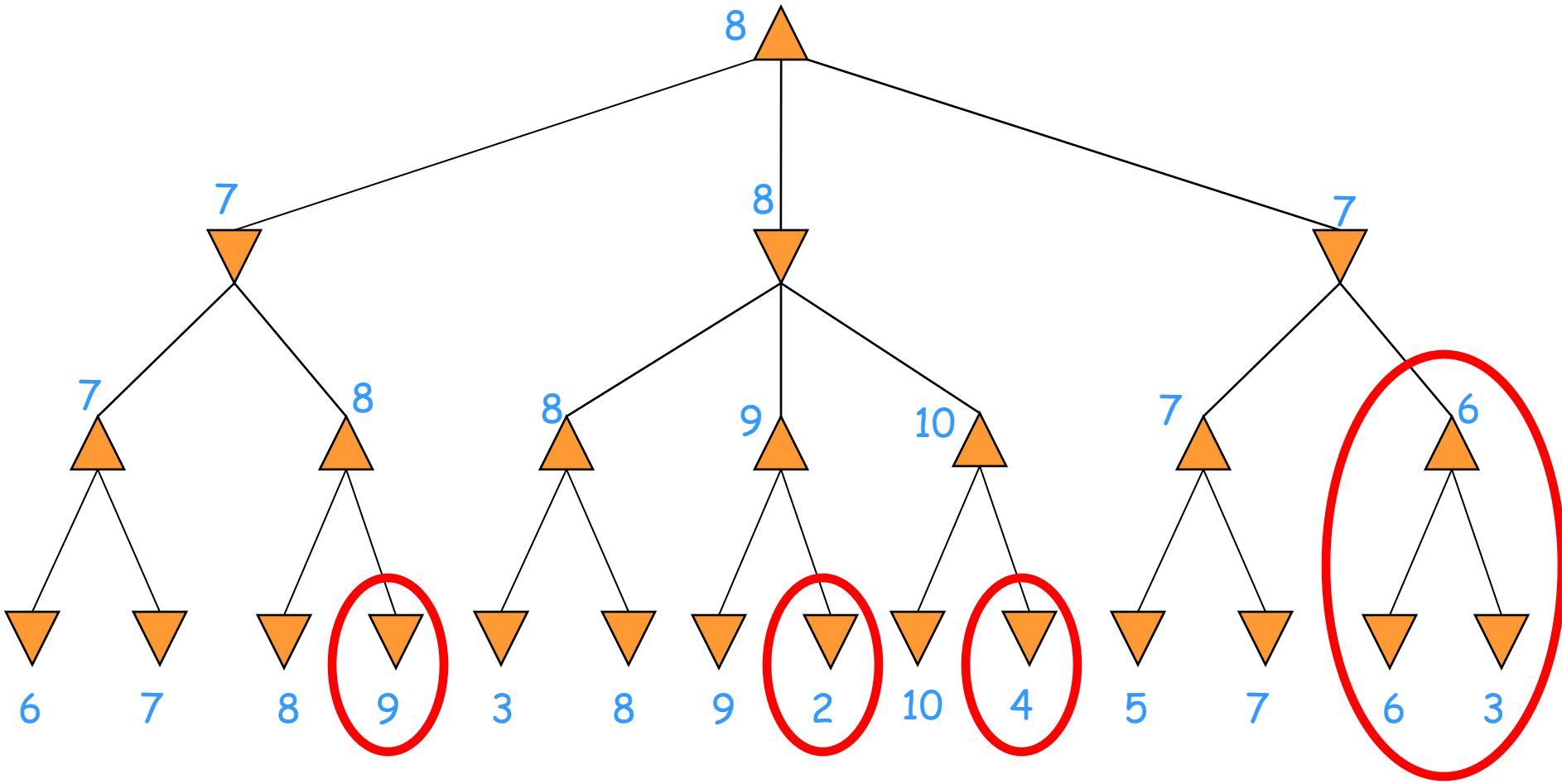


Juegos

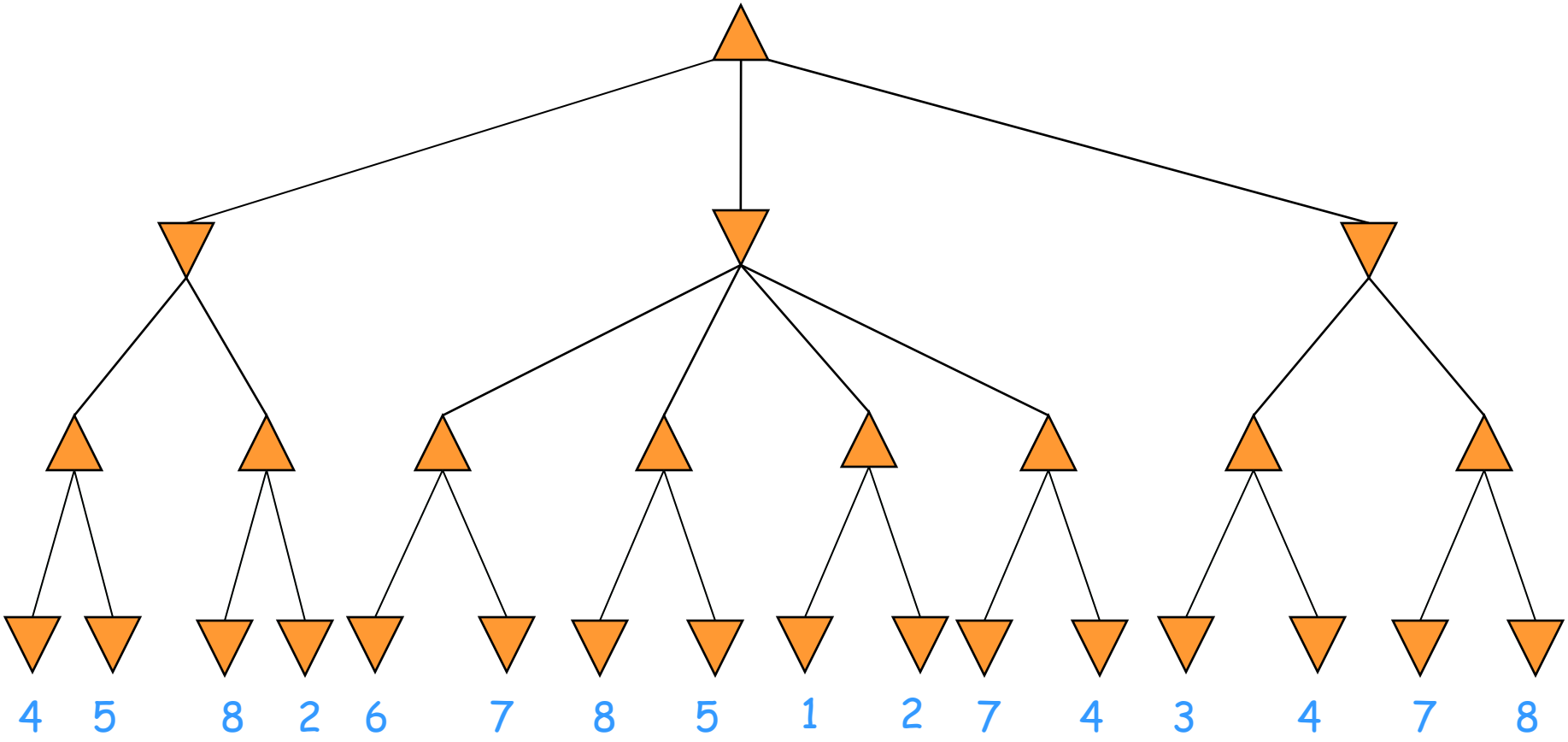


Indique qué nodos se podan

Juegos

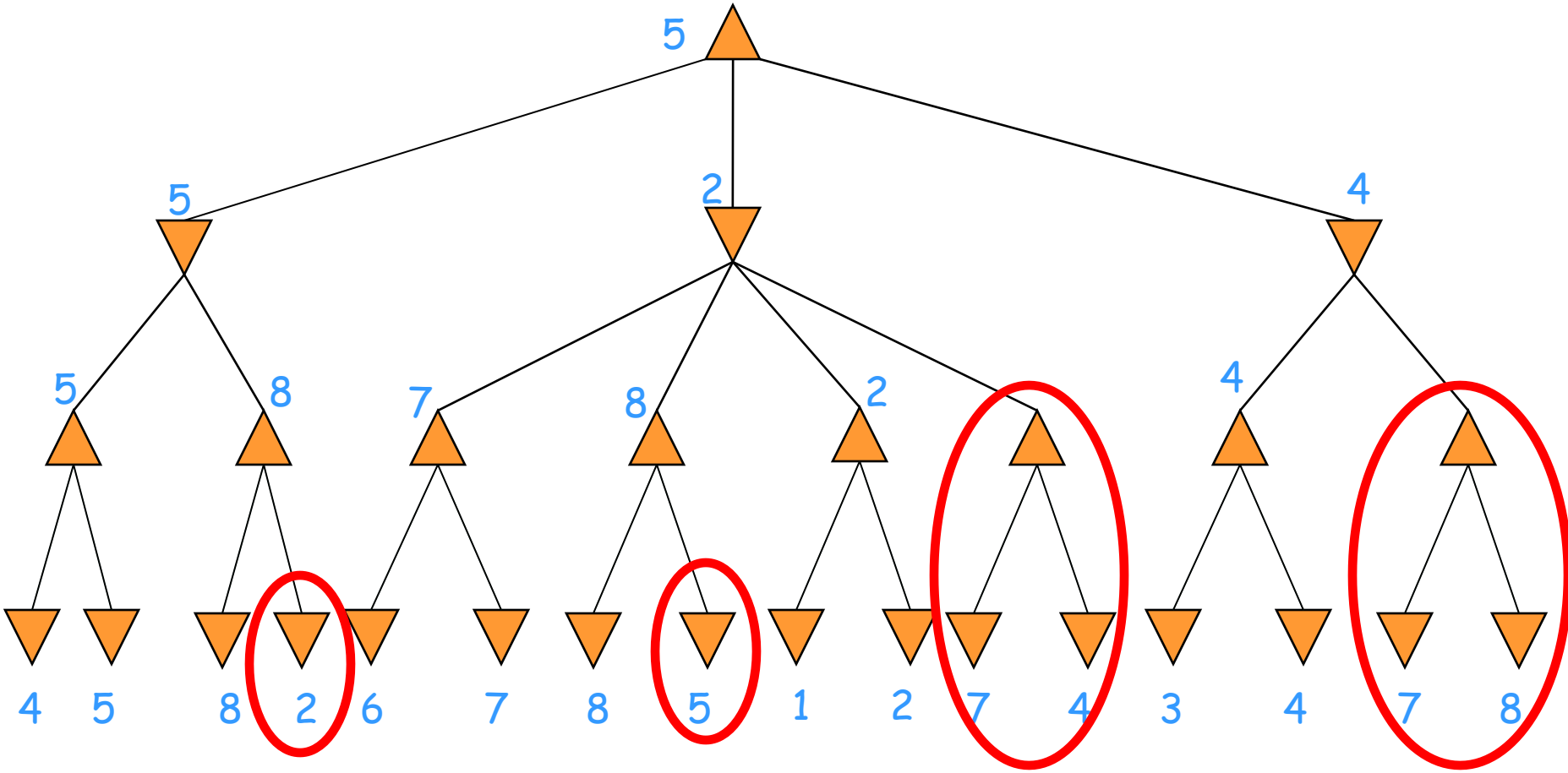


Juegos

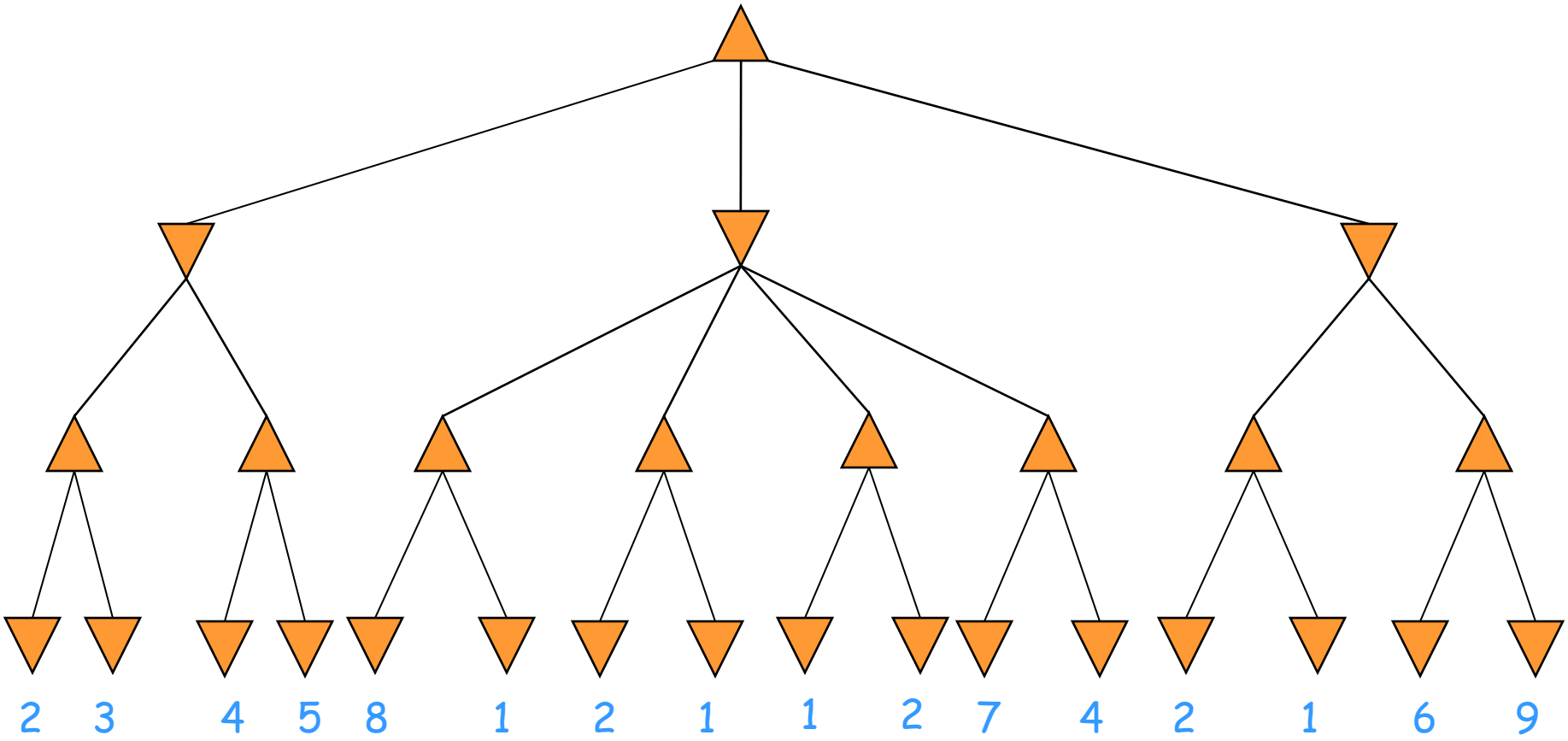


Indique qué nodos se podan

Juegos

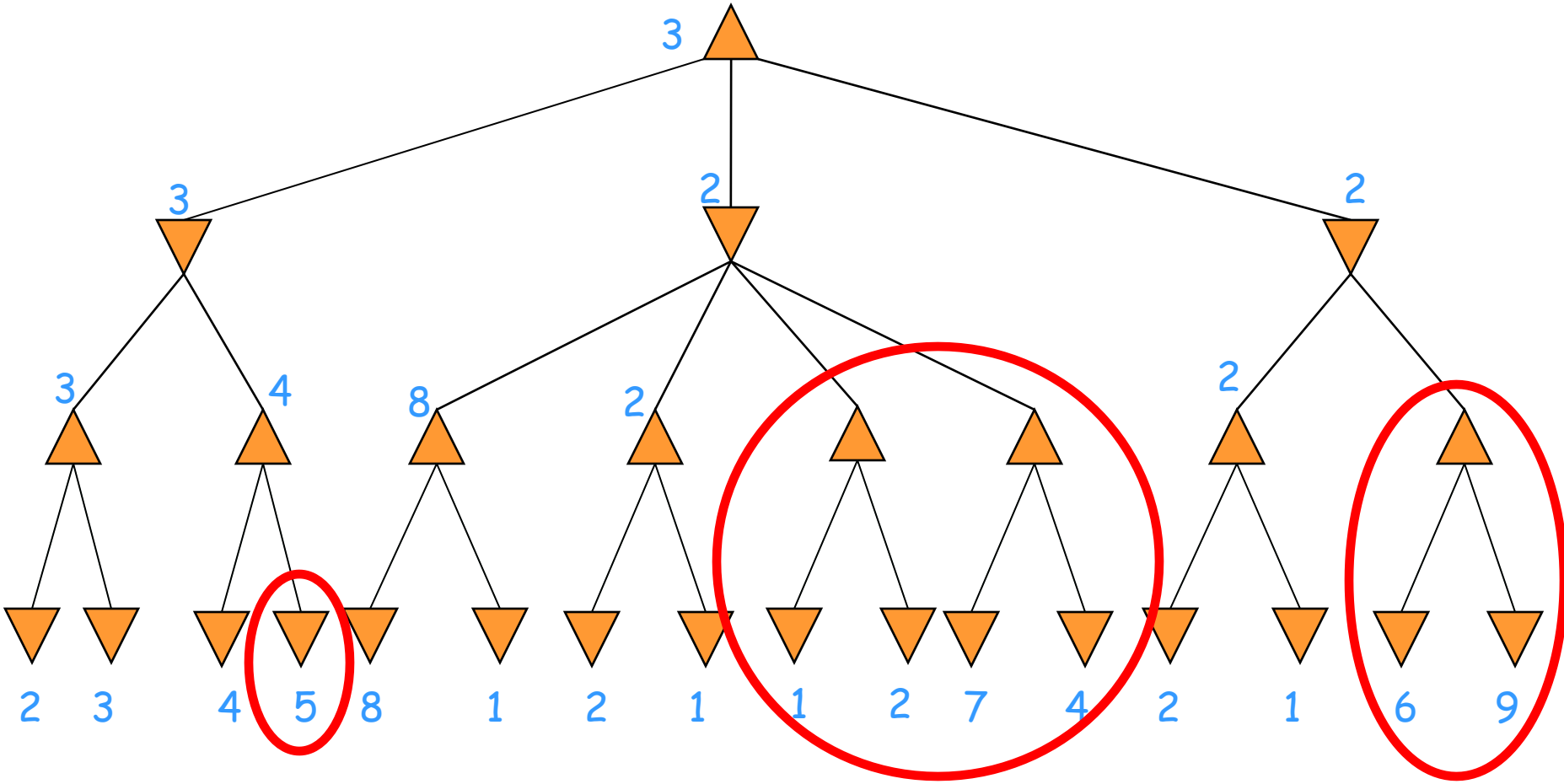


Juegos

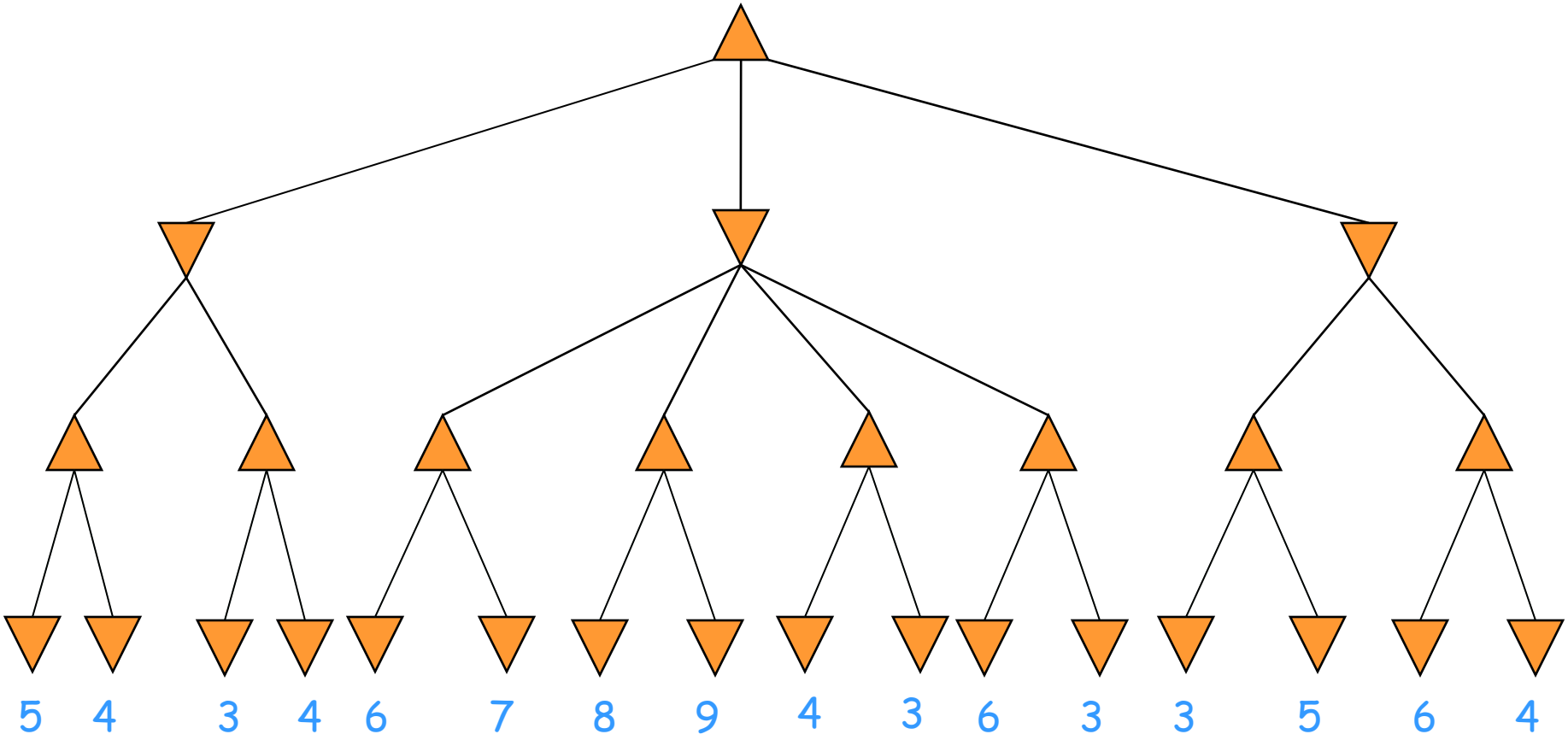


Indique qué nodos se podan

Juegos

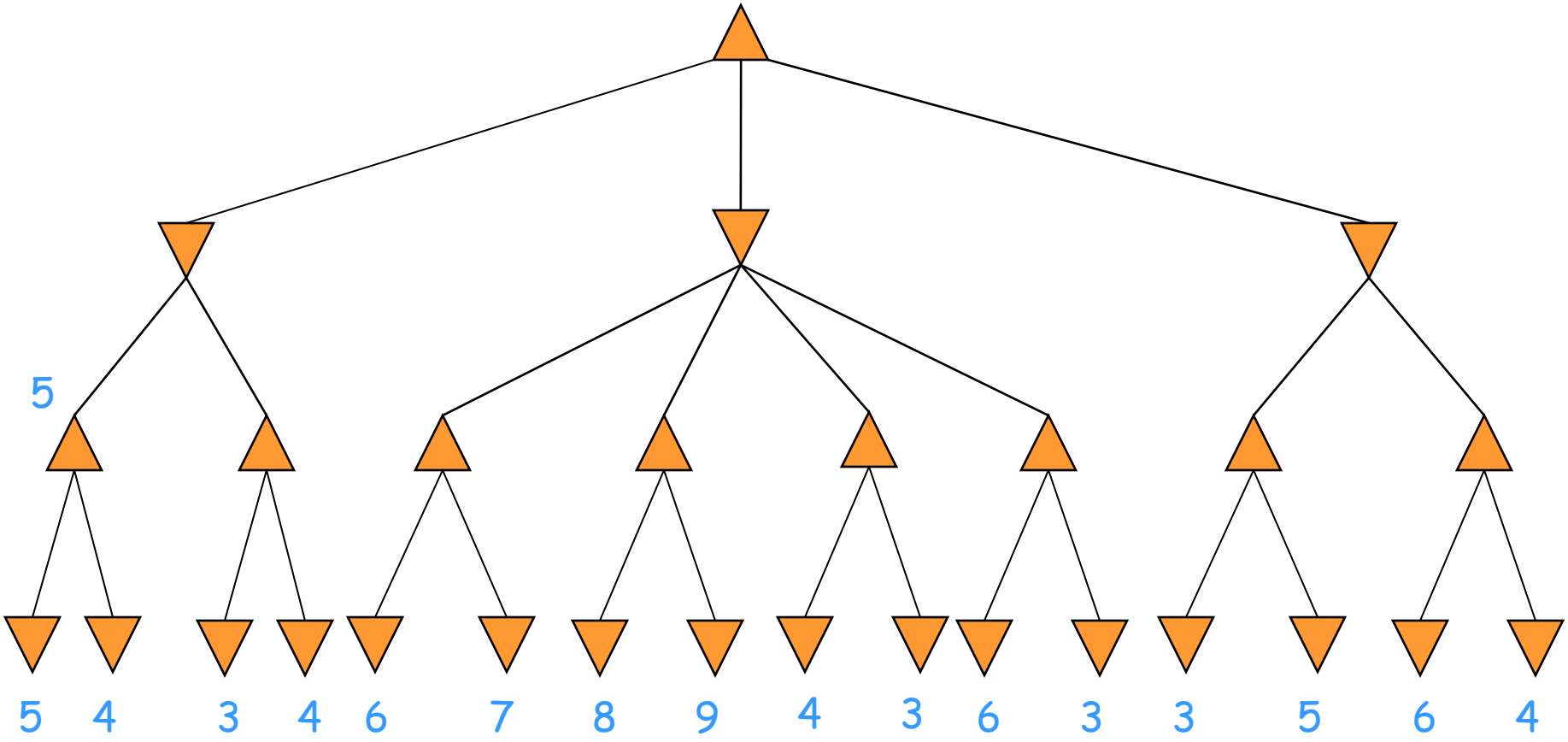


Juegos

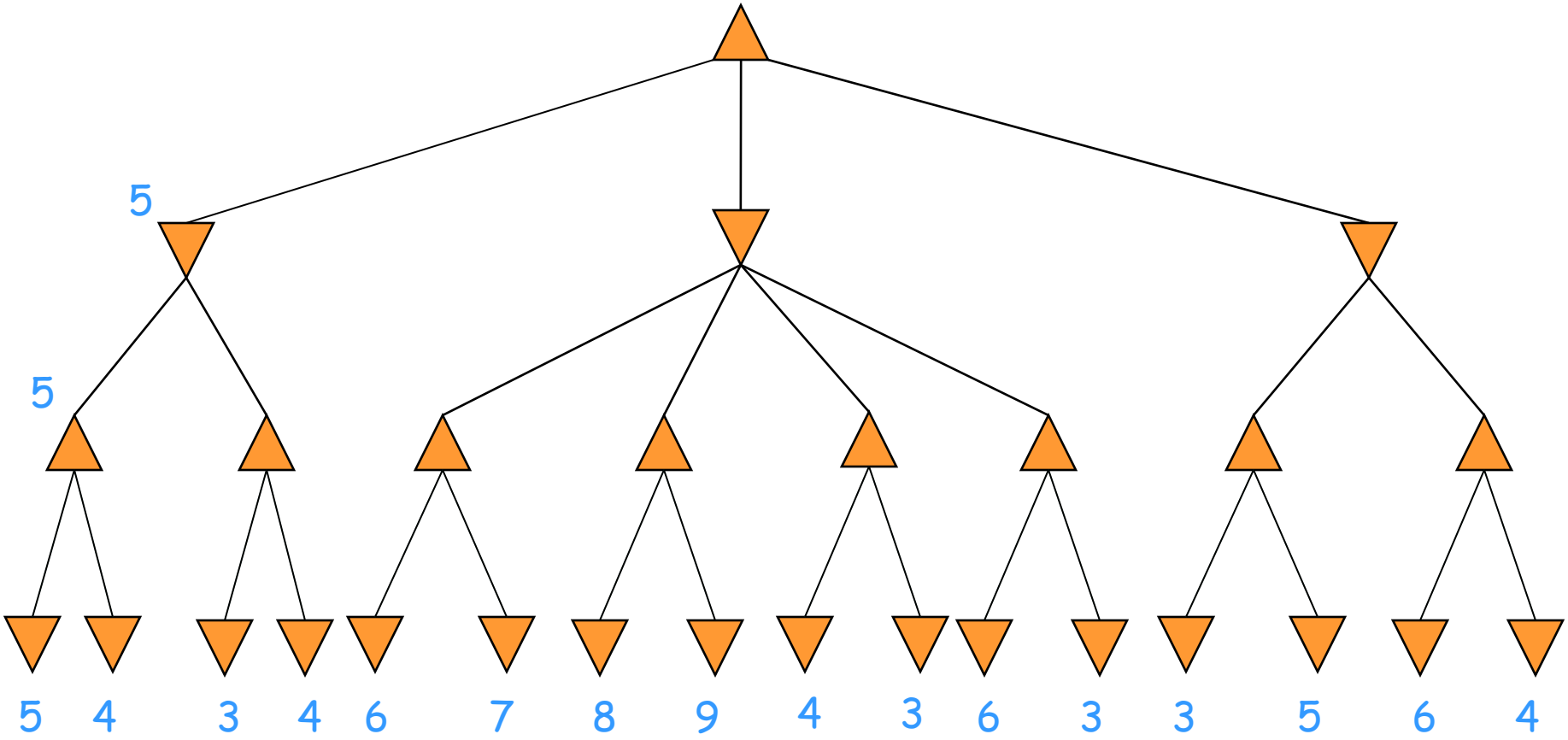


*

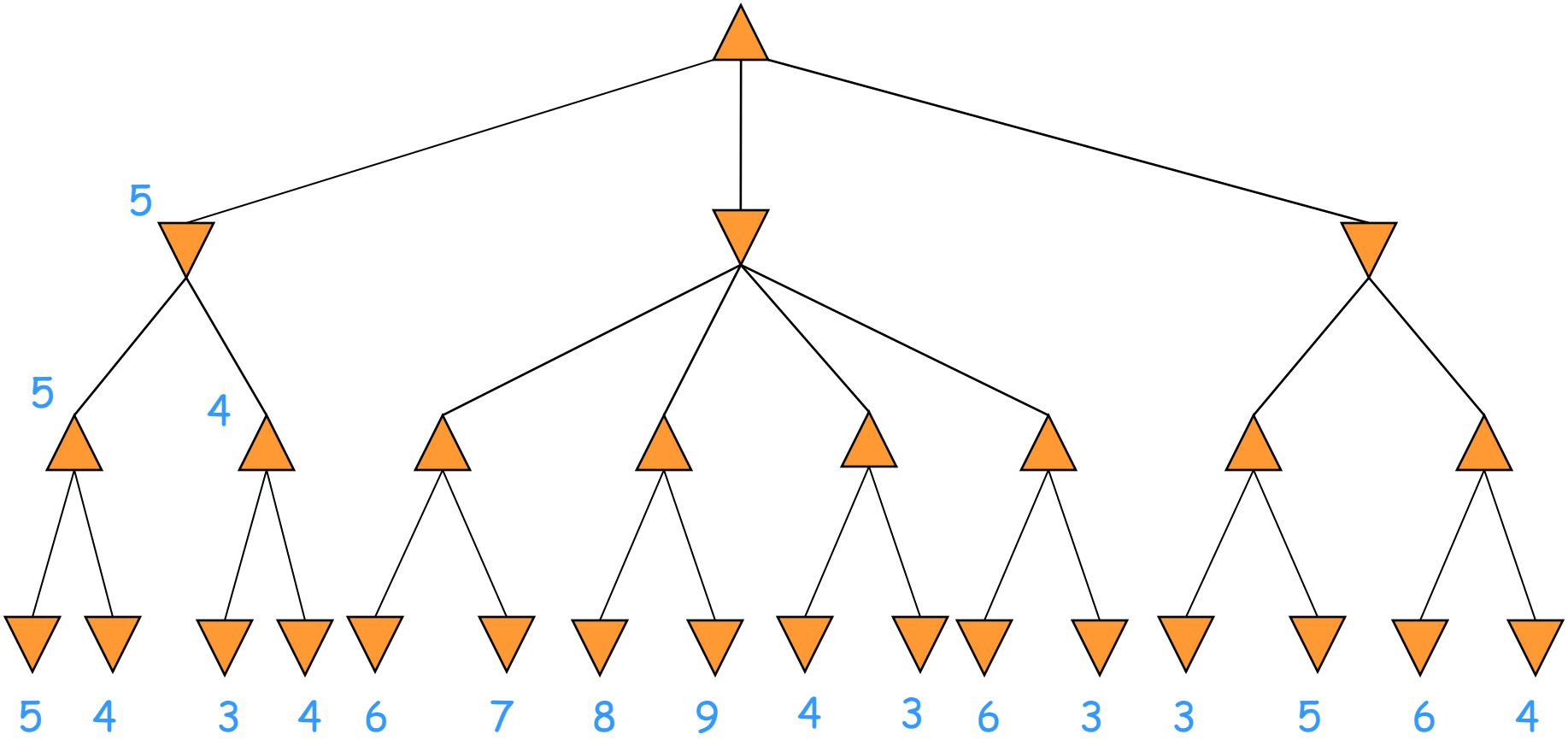
Juegos



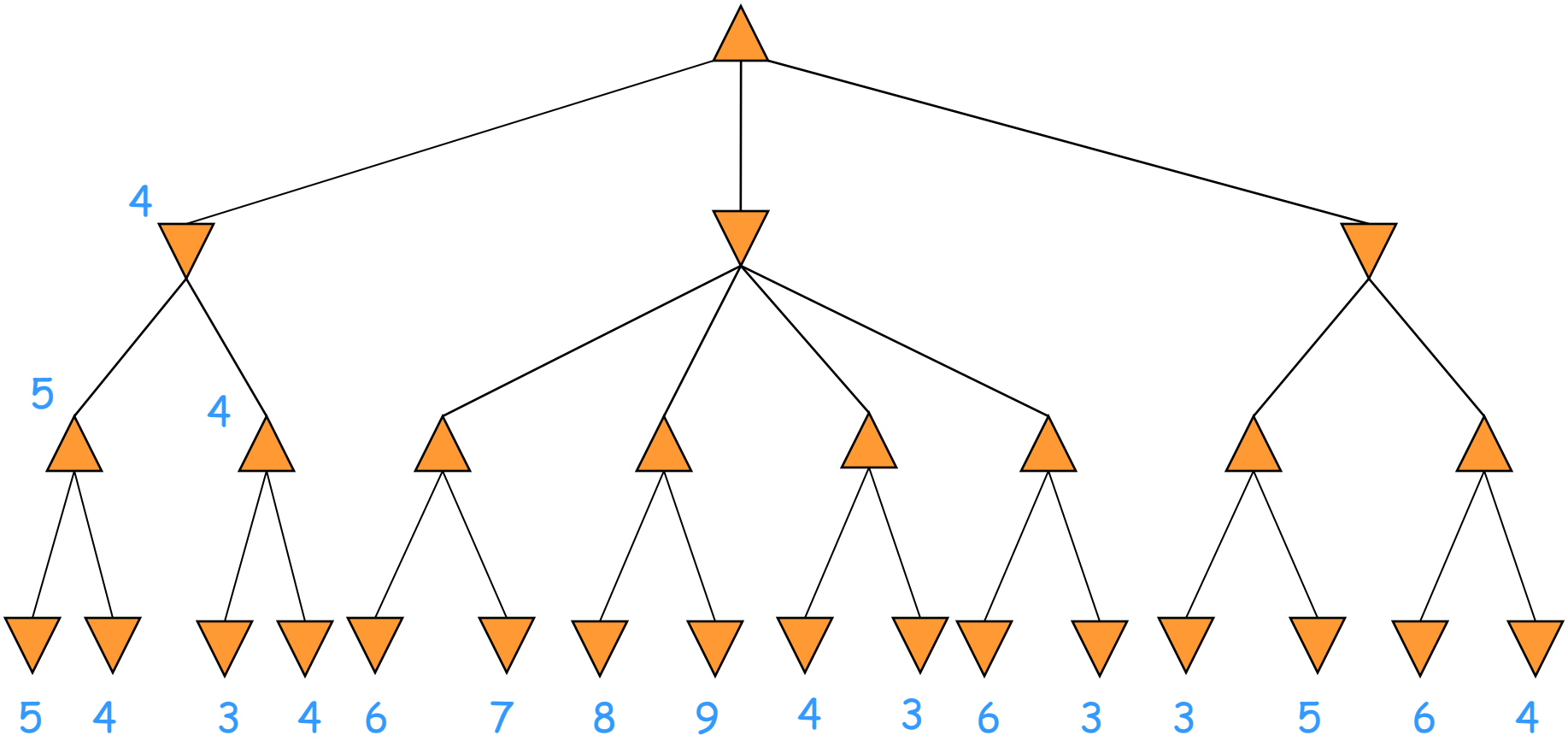
Juegos



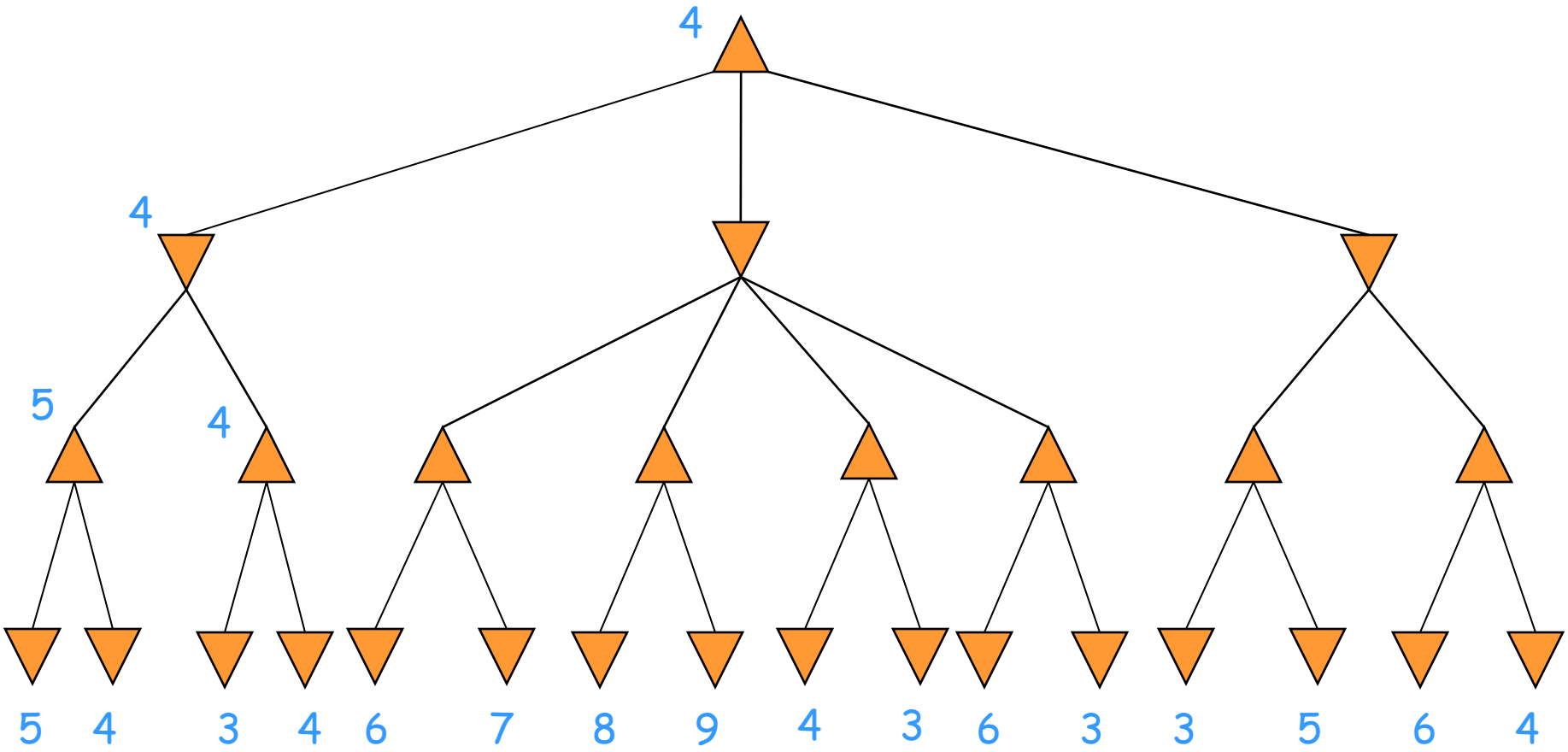
Juegos



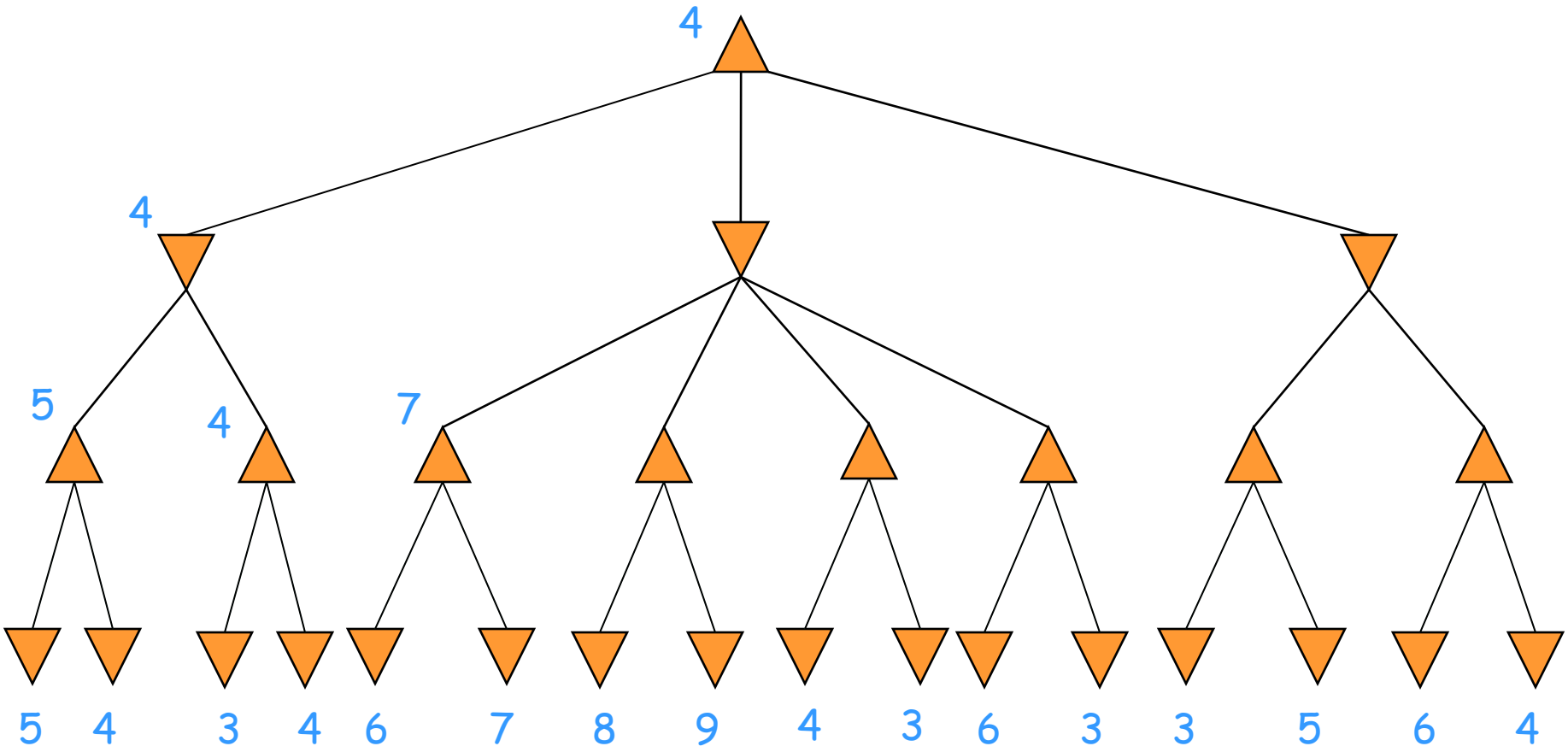
Juegos



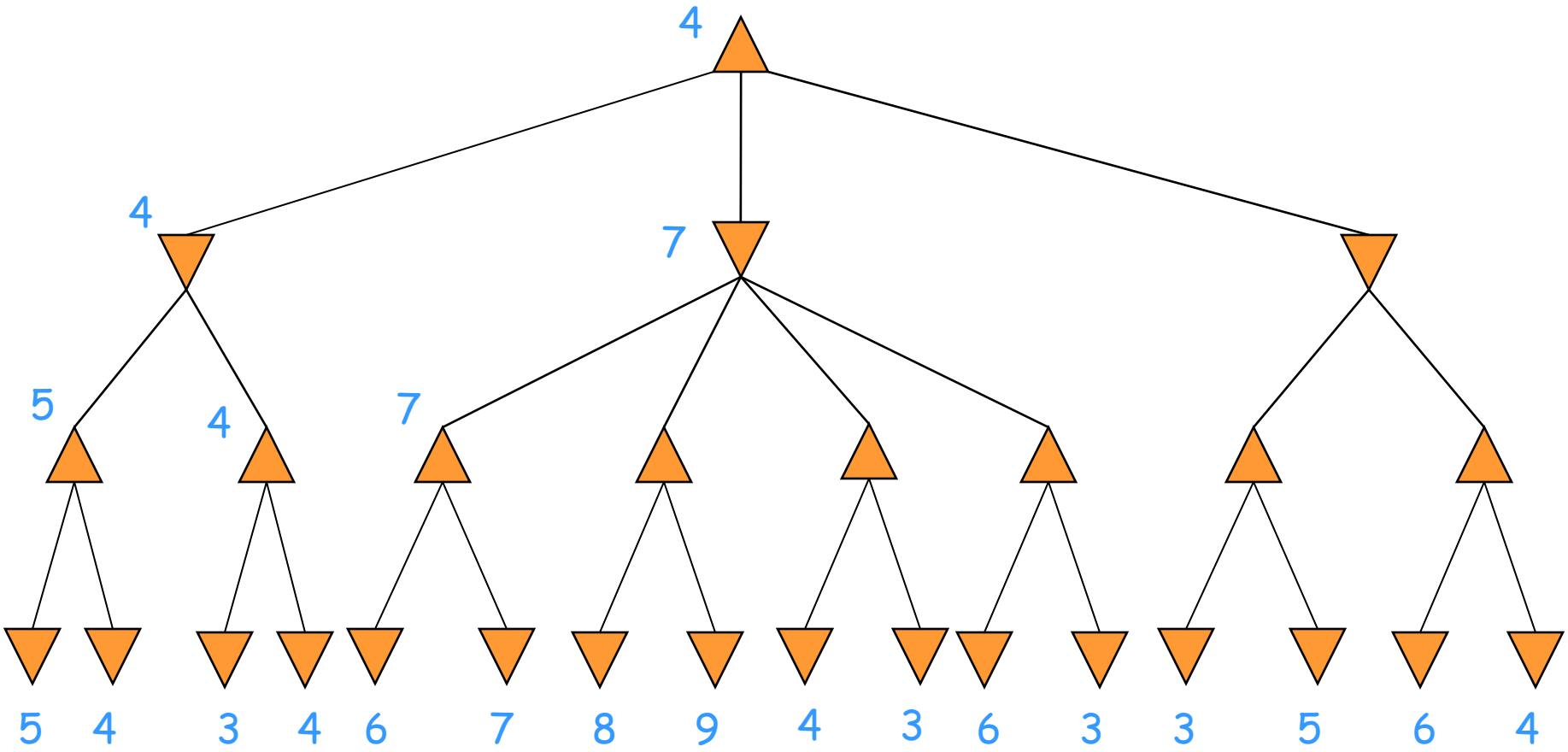
Juegos



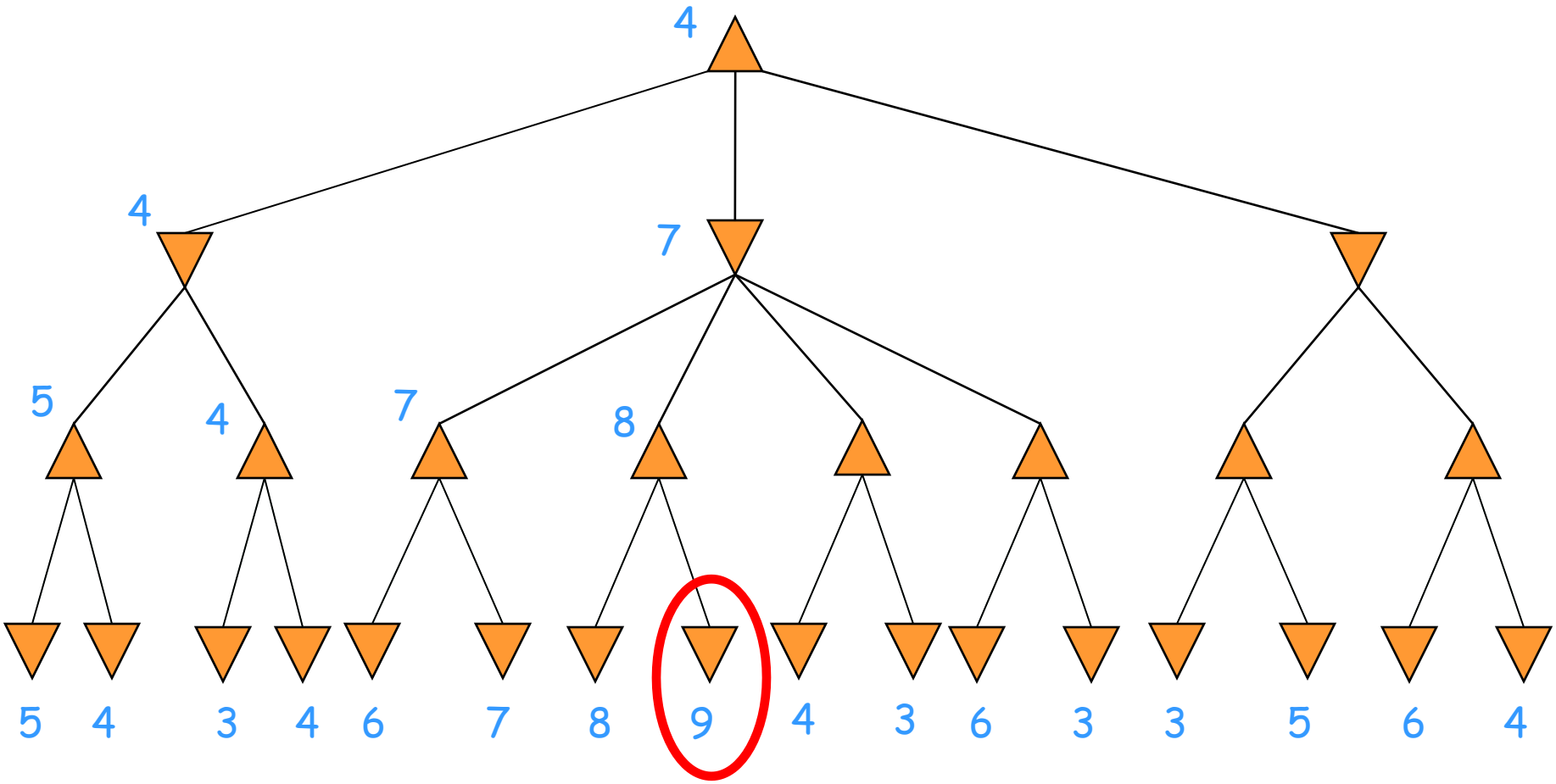
Juegos



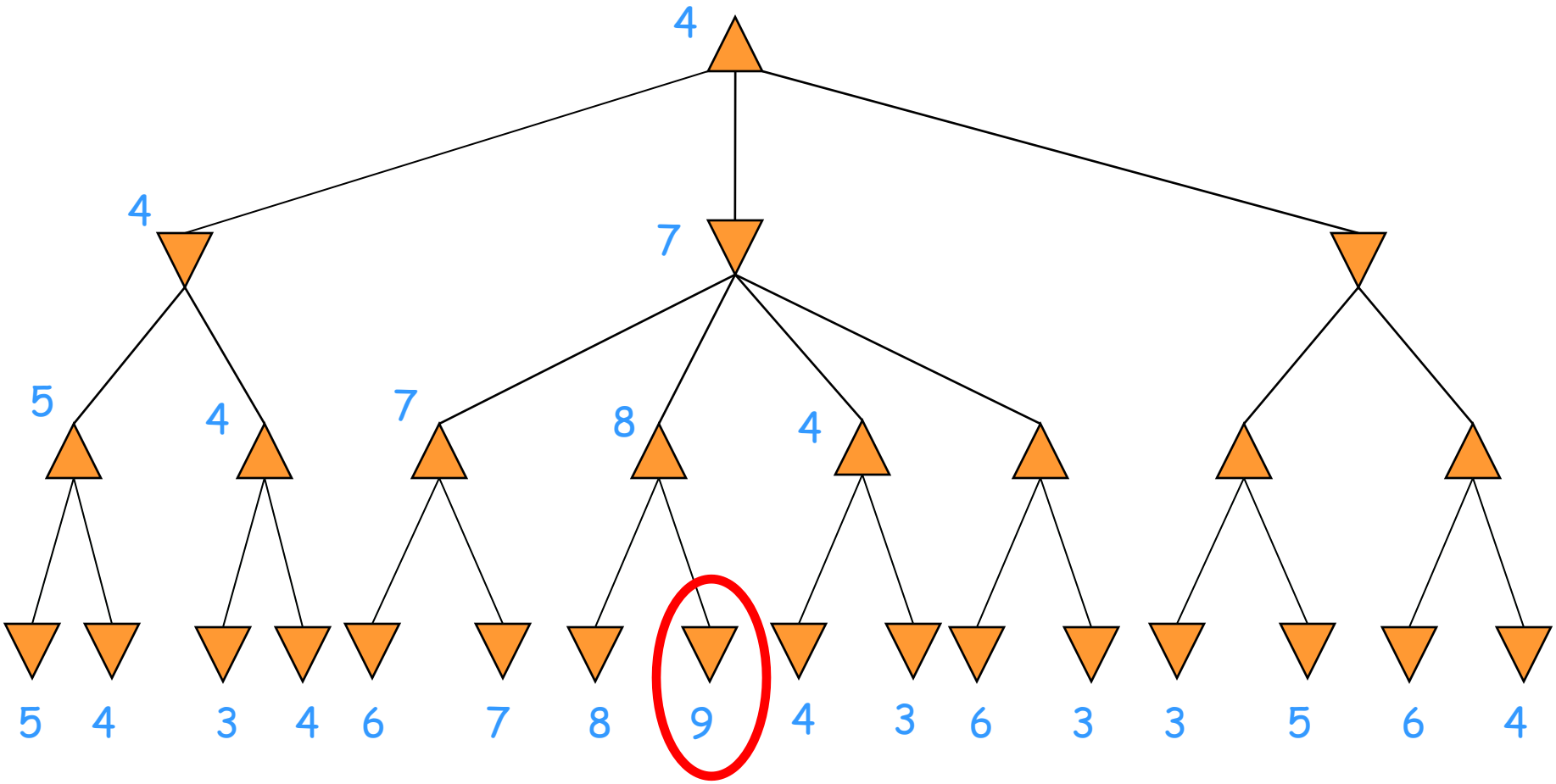
Juegos



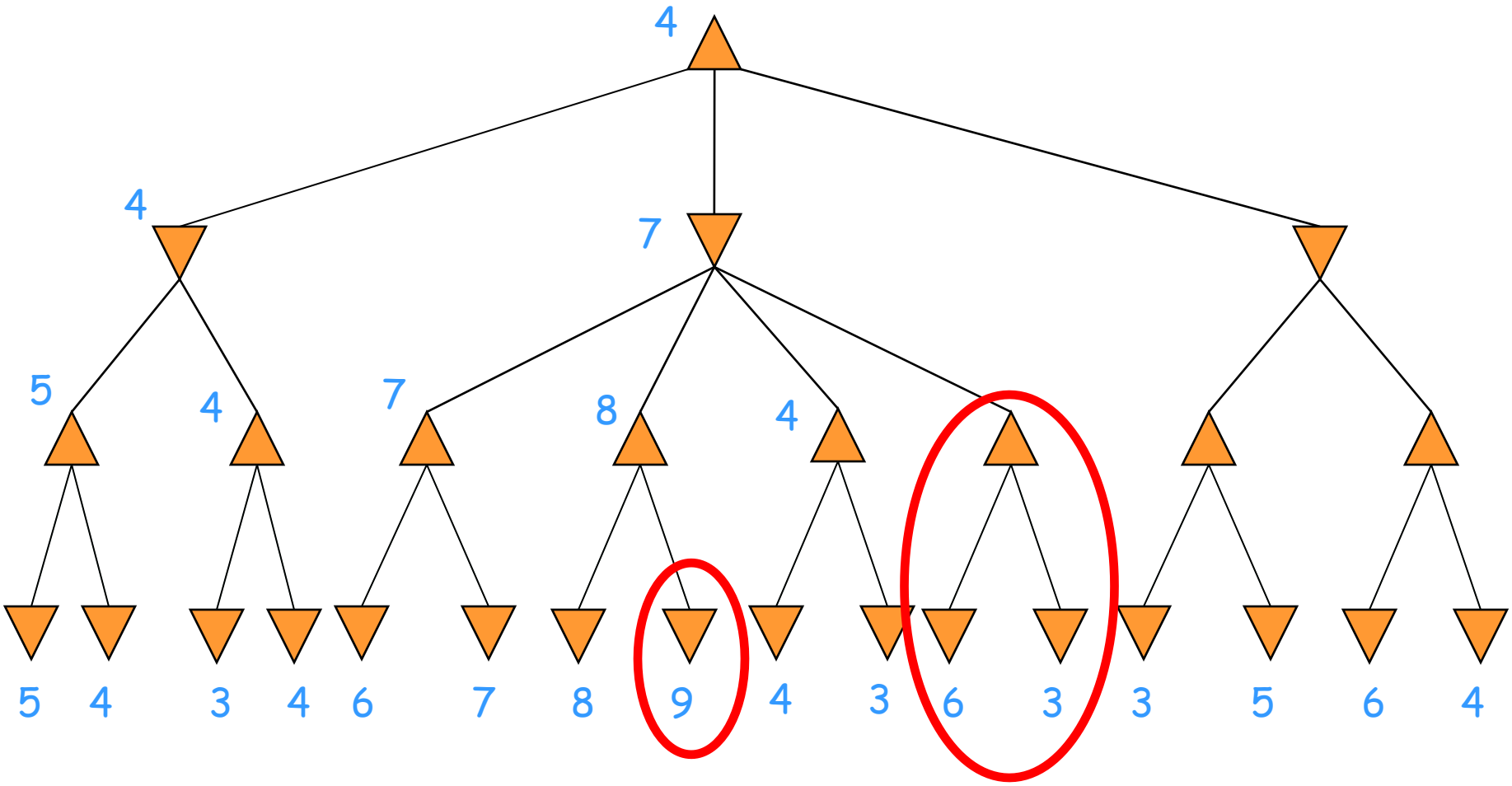
Juegos



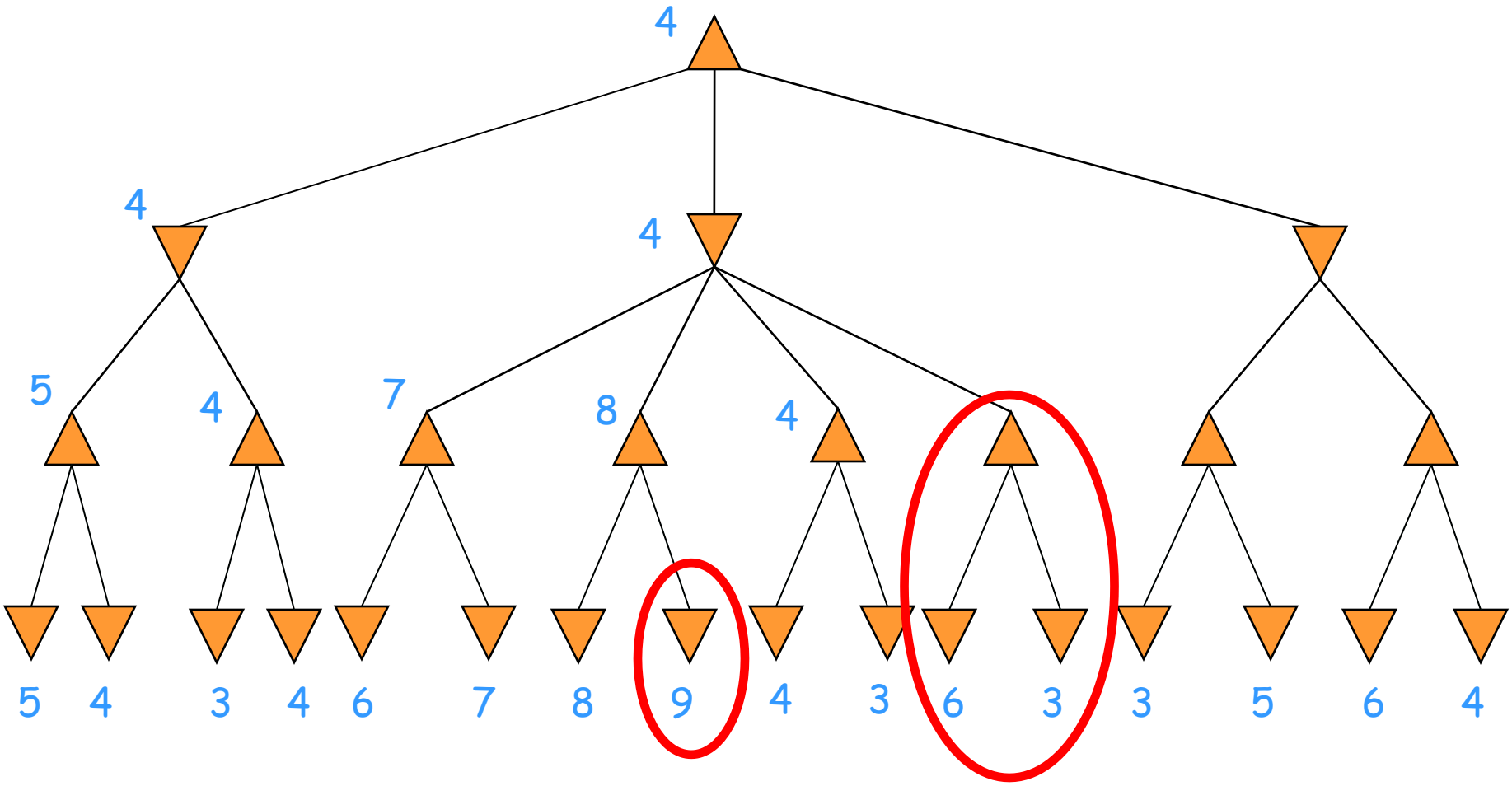
Juegos



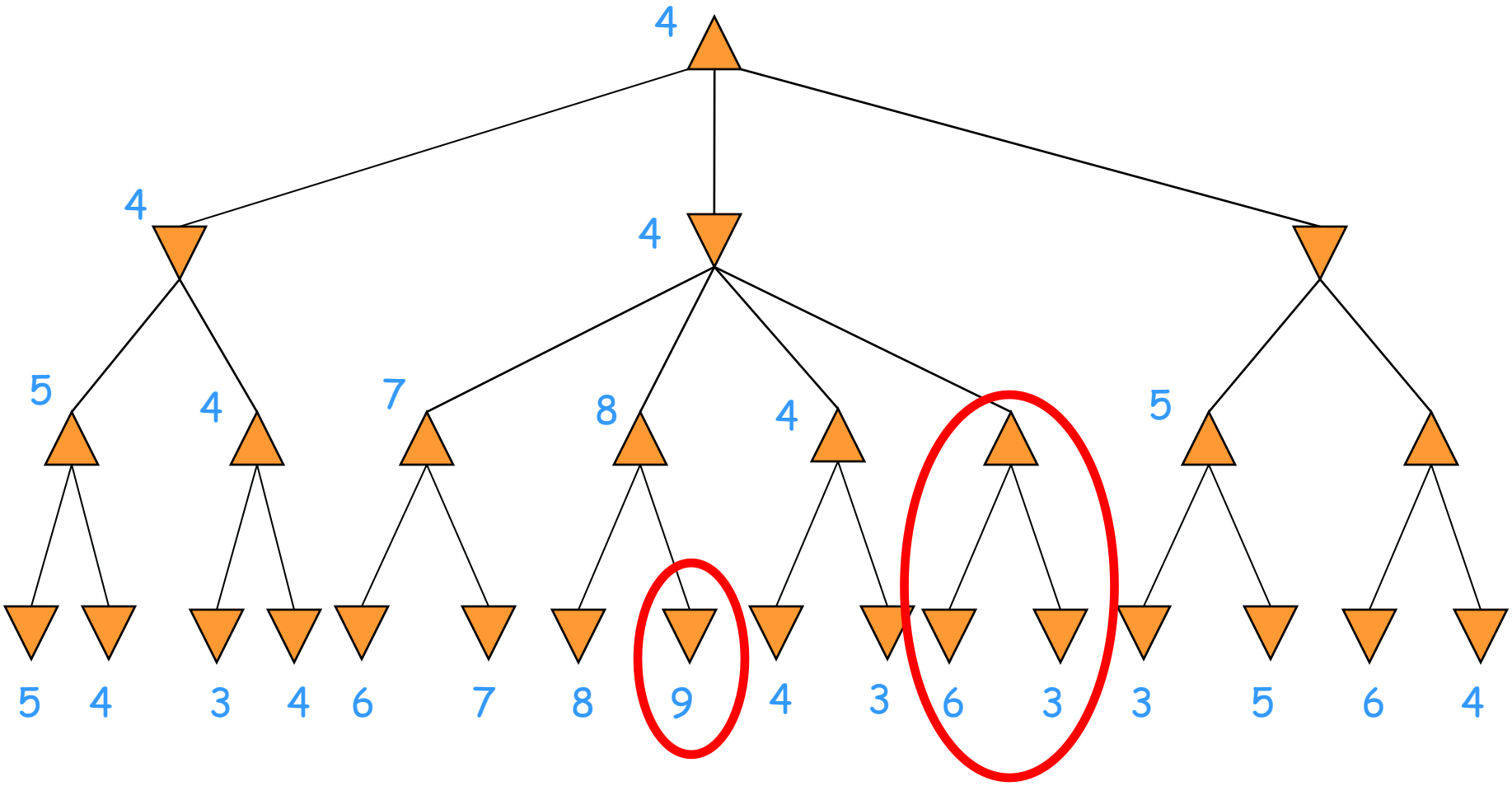
Juegos



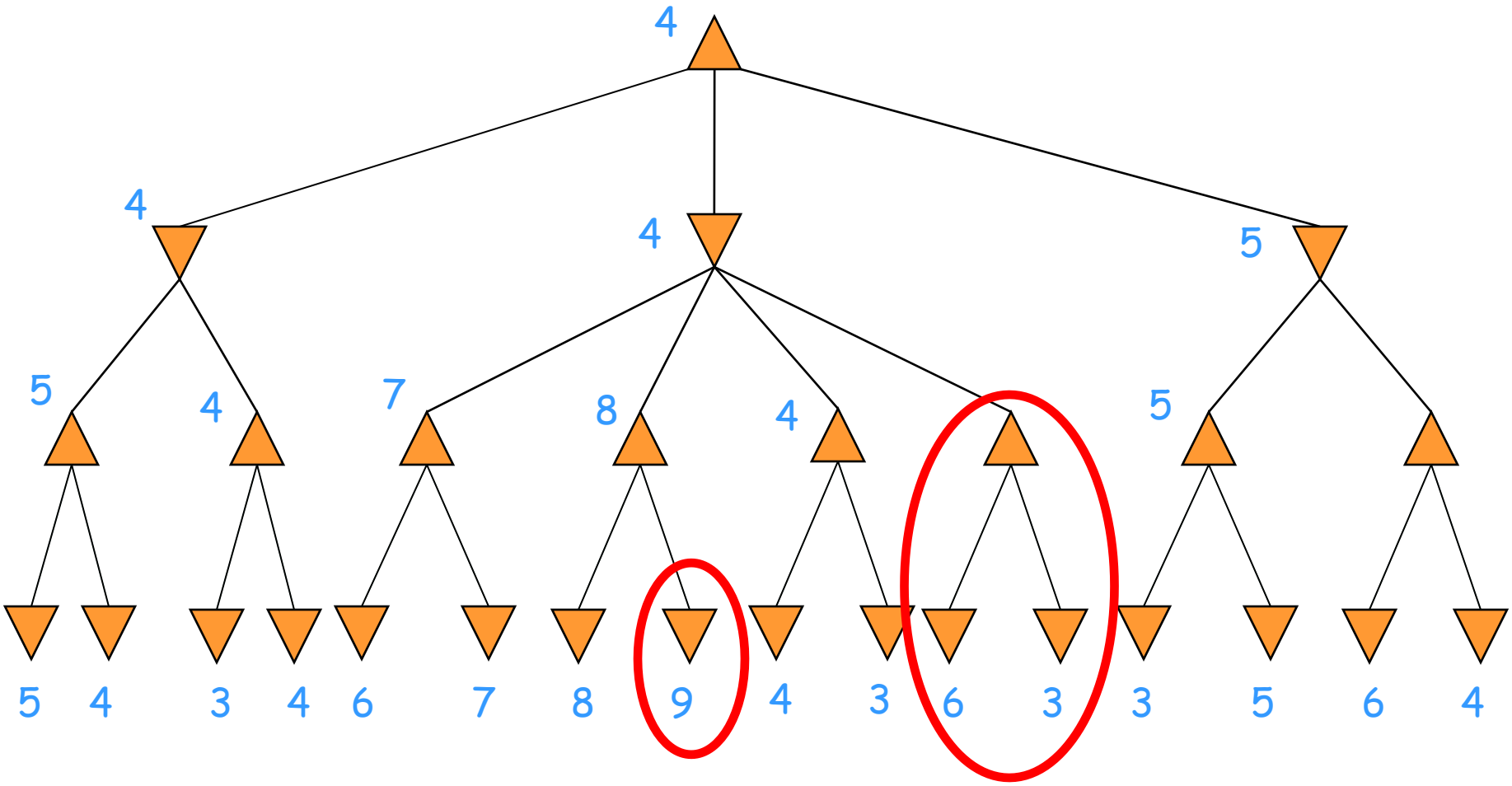
Juegos



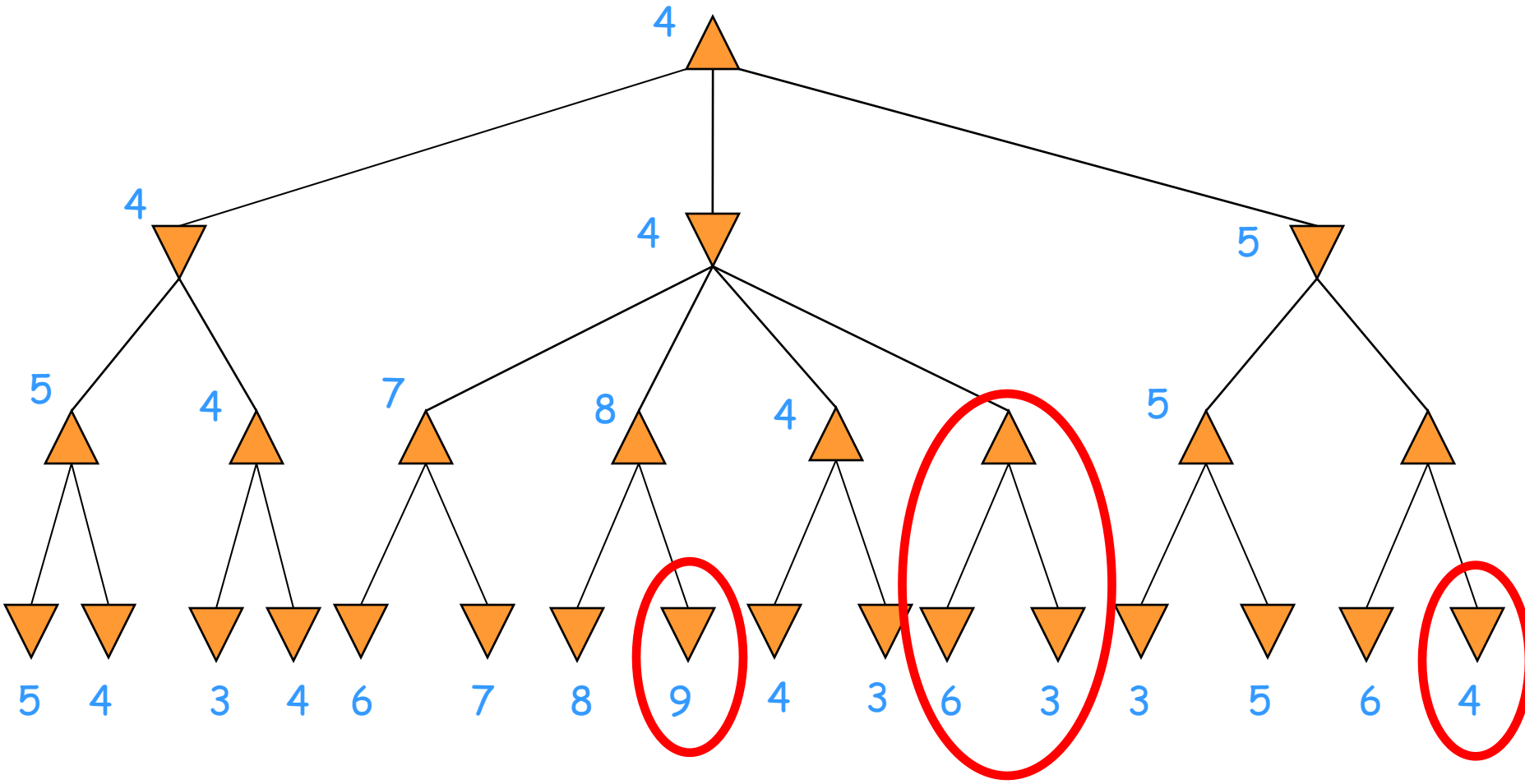
Juegos



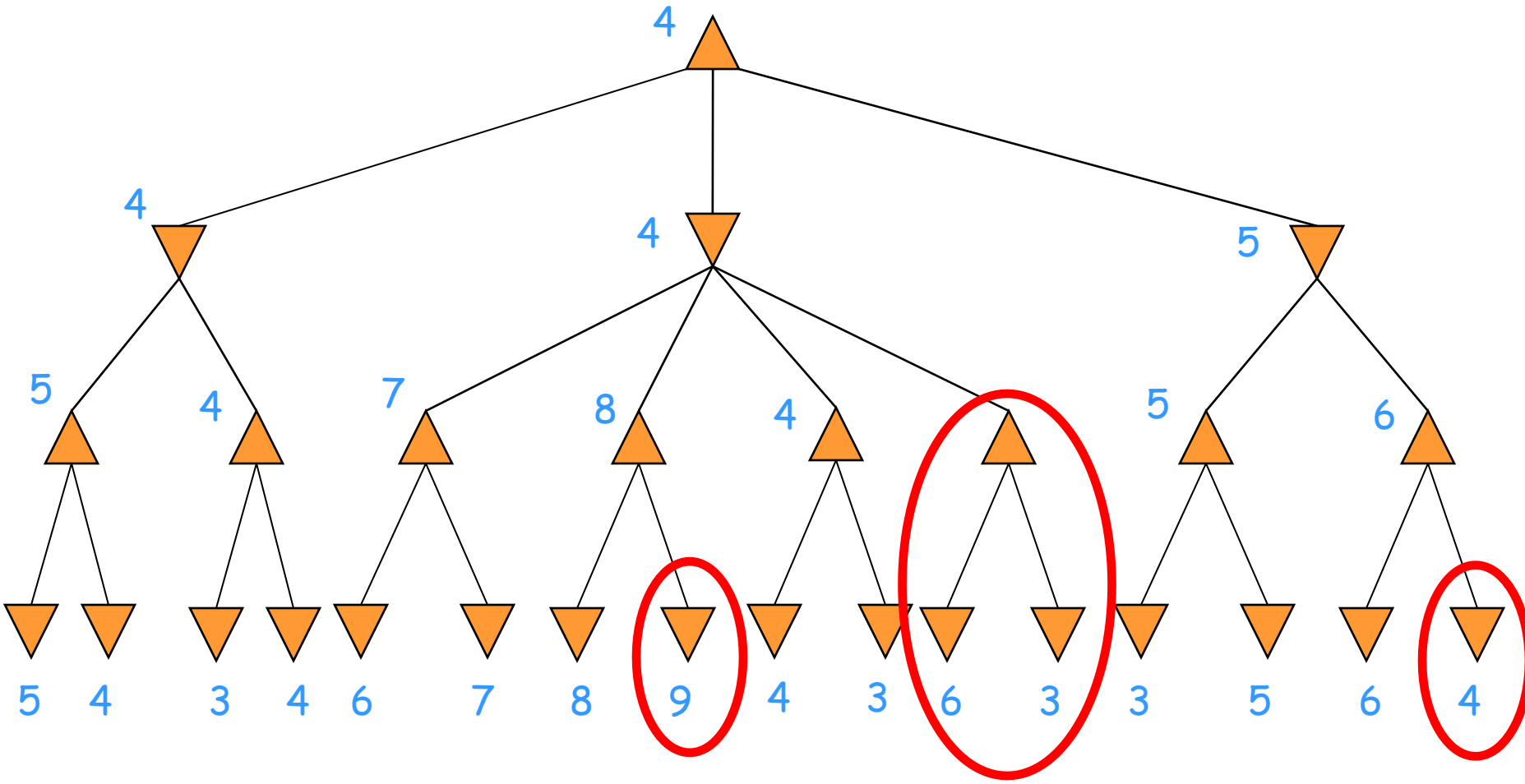
Juegos



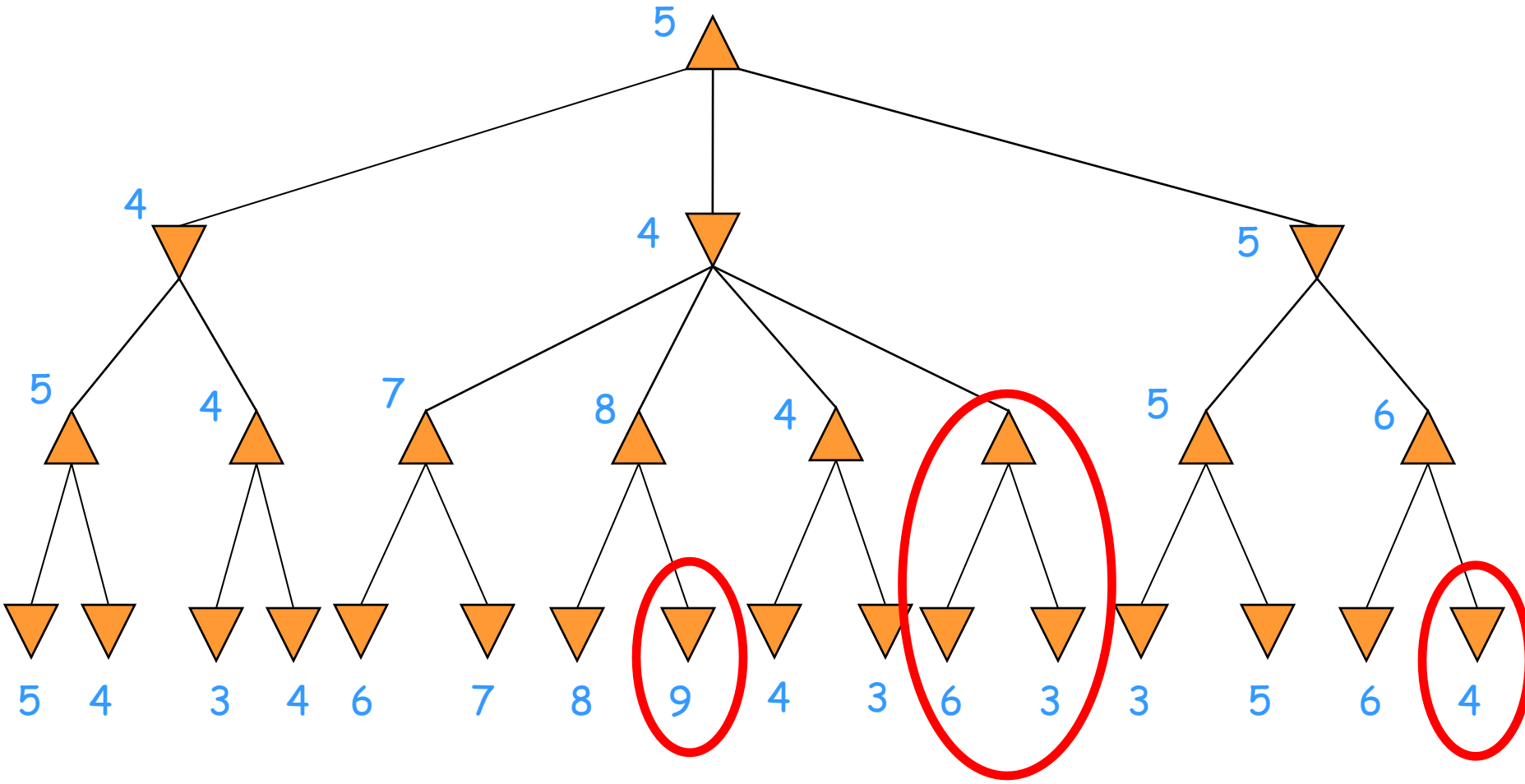
Juegos



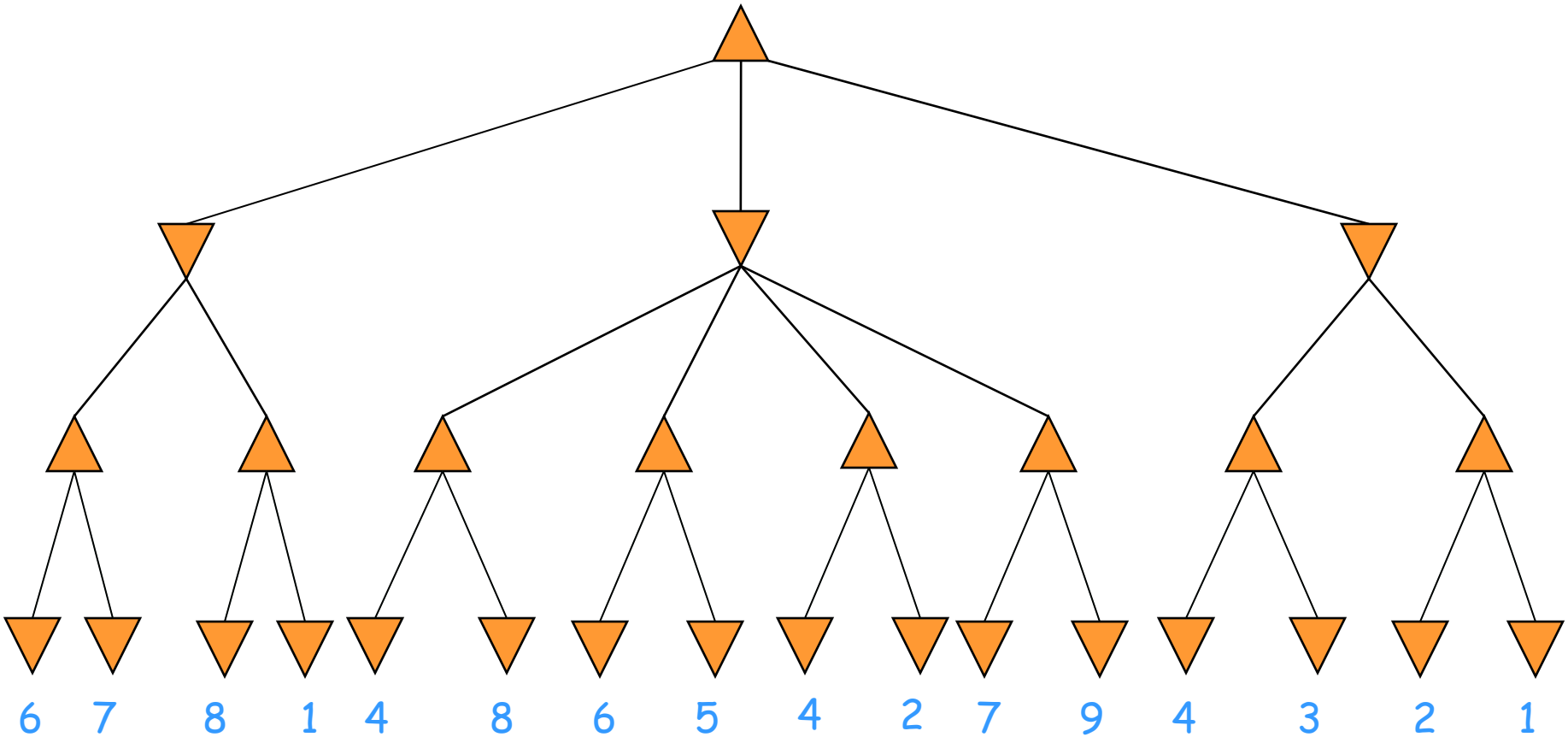
Juegos



Juegos

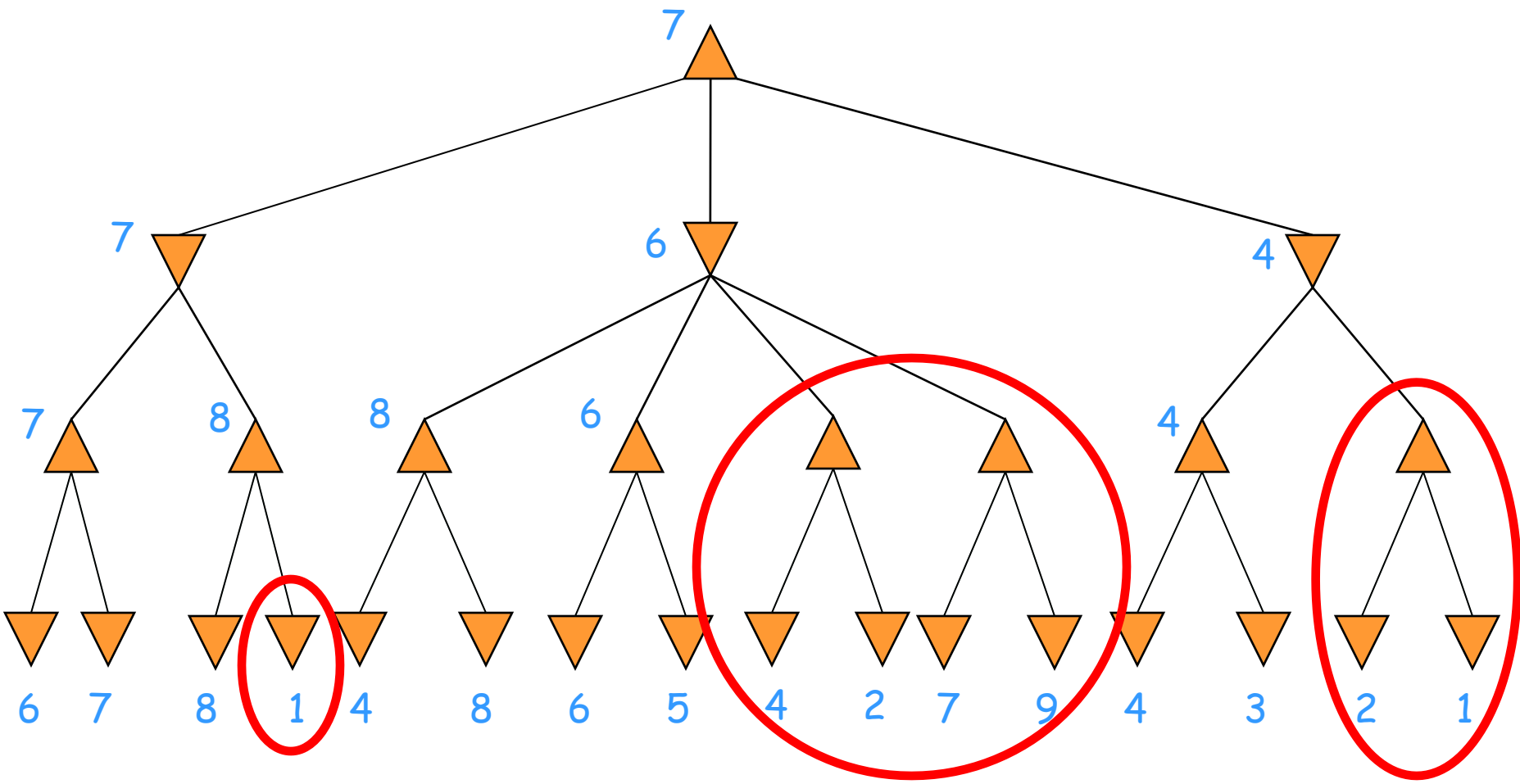


Juegos



Indique qué nodos se podan

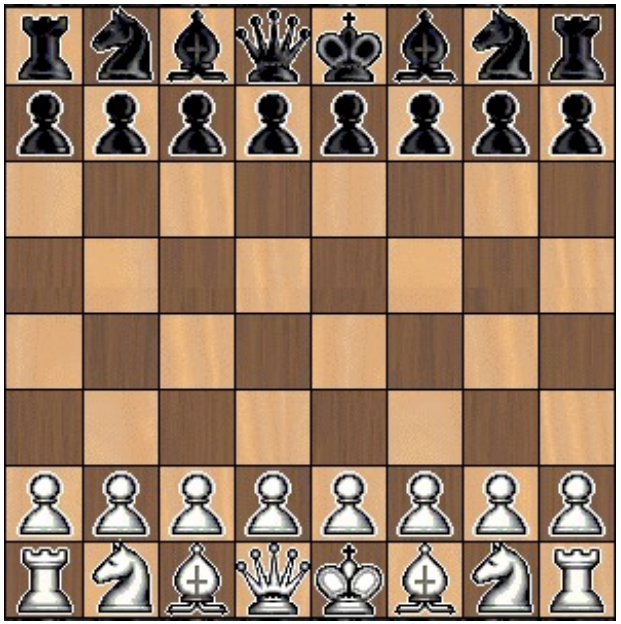
Juegos

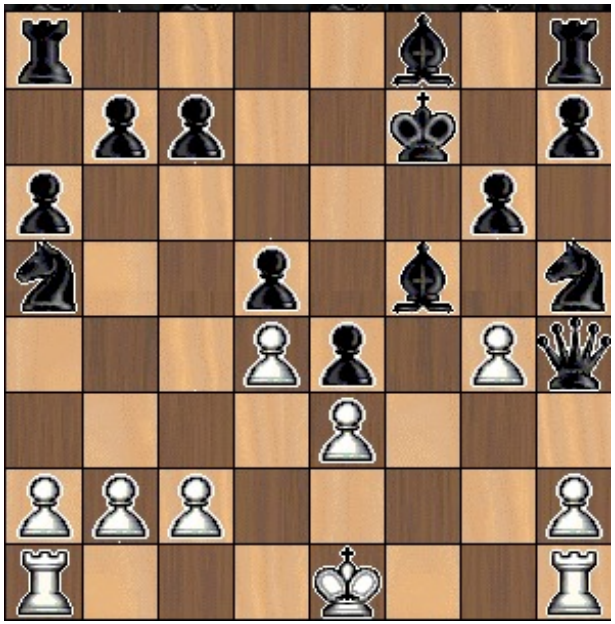


Juegos

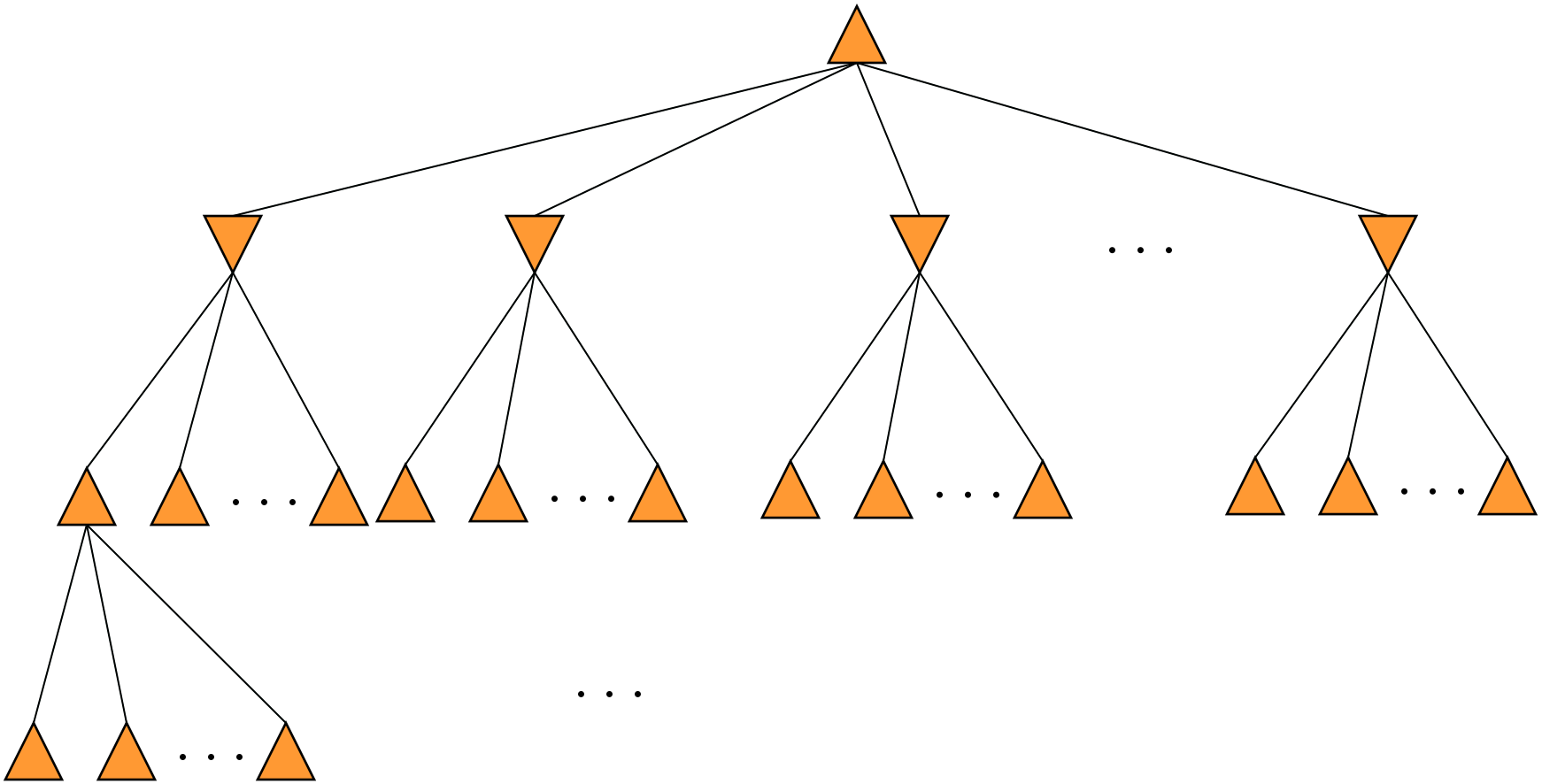
Ajedrez

- Analice el árbol de juego

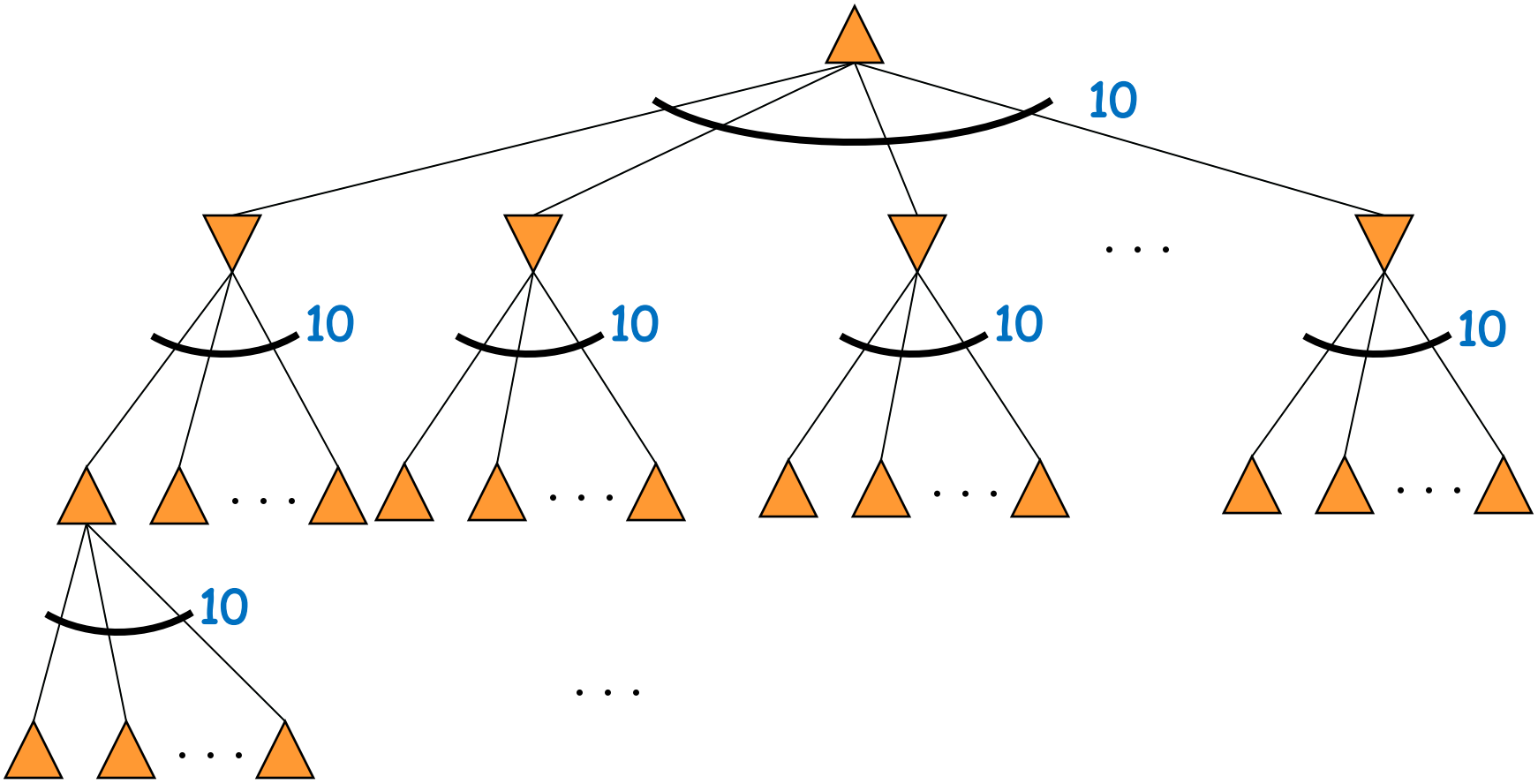




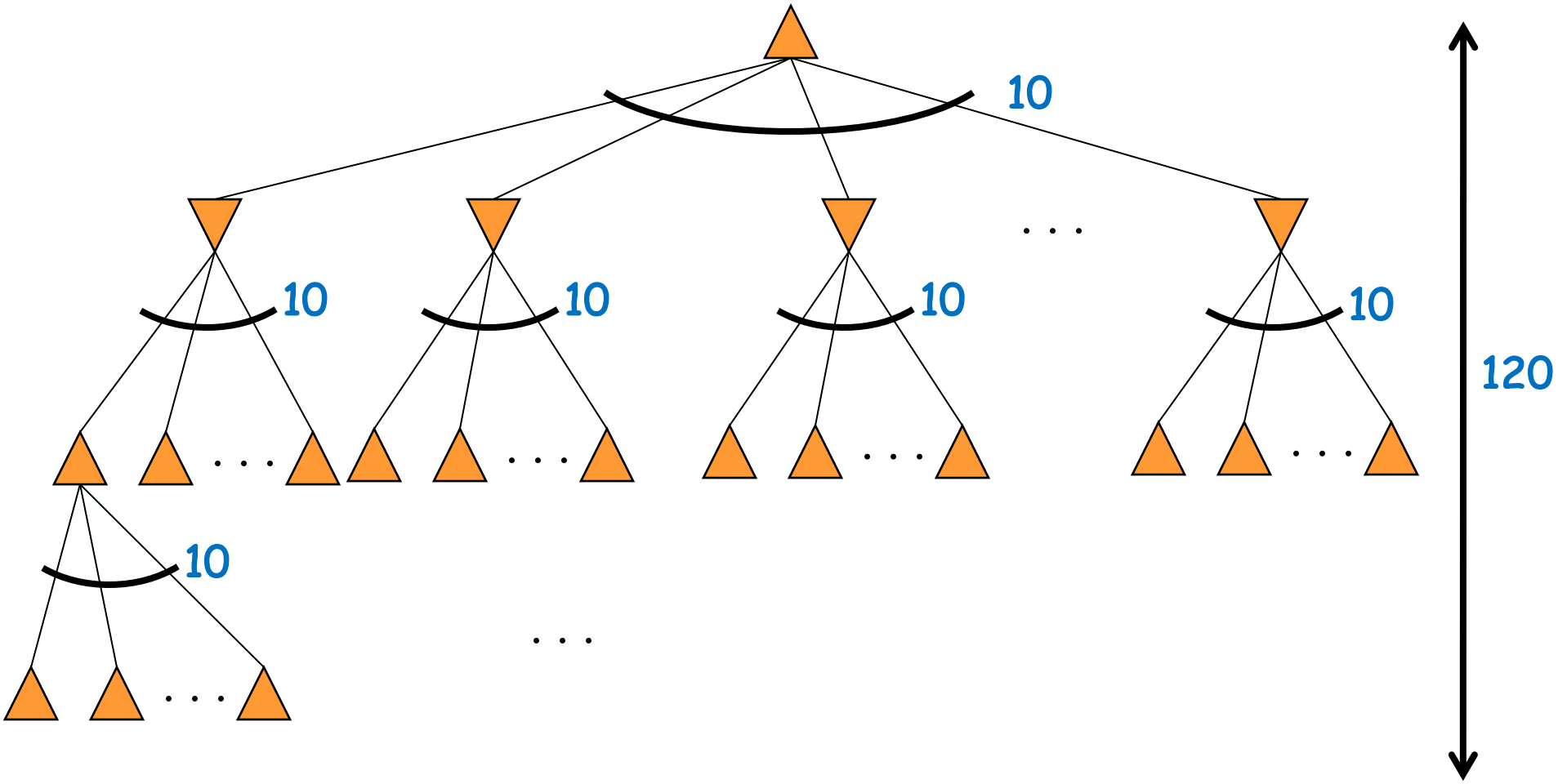
Juegos



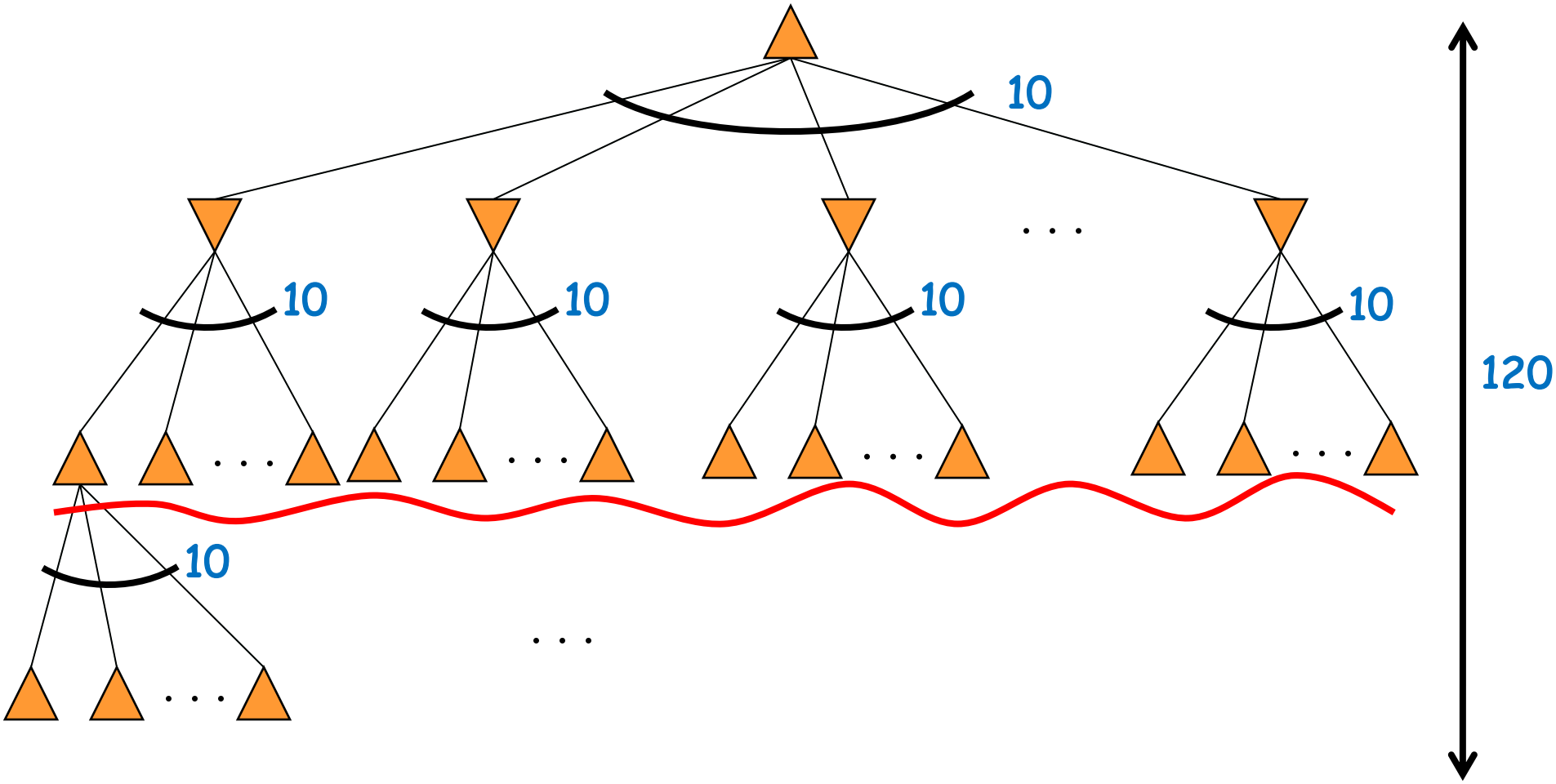
Juegos



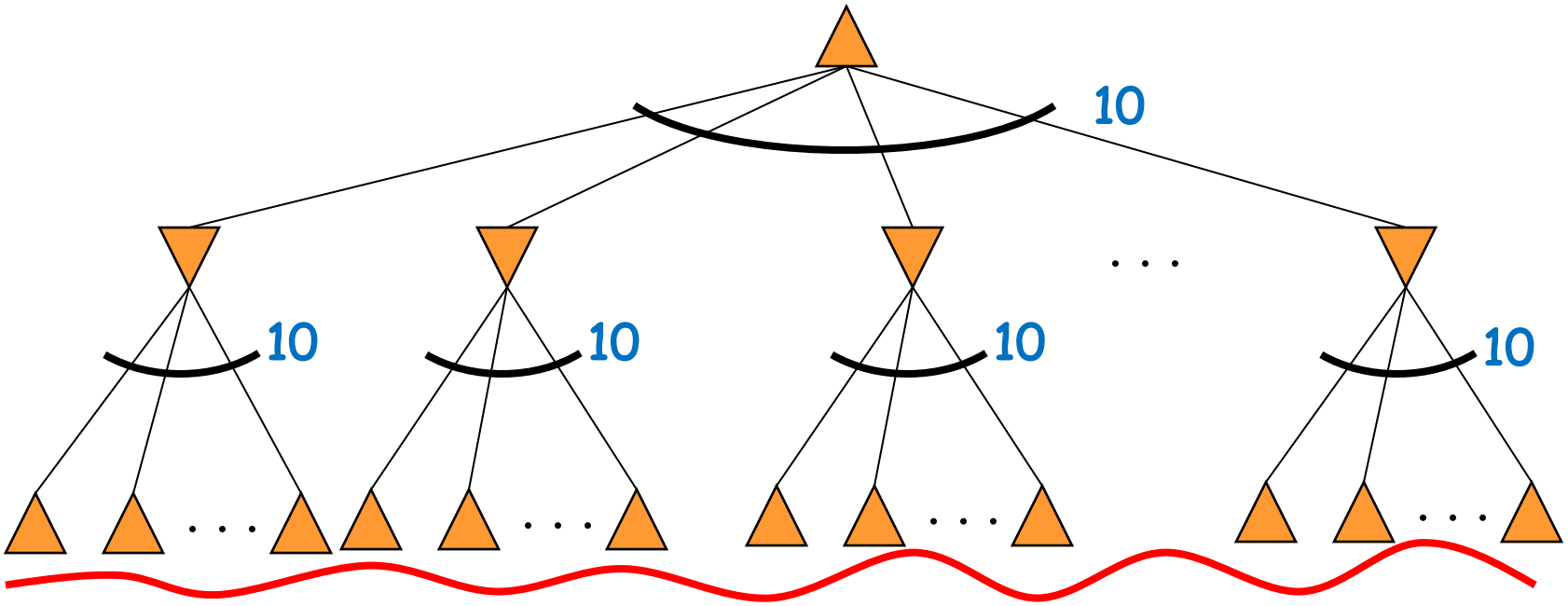
Juegos



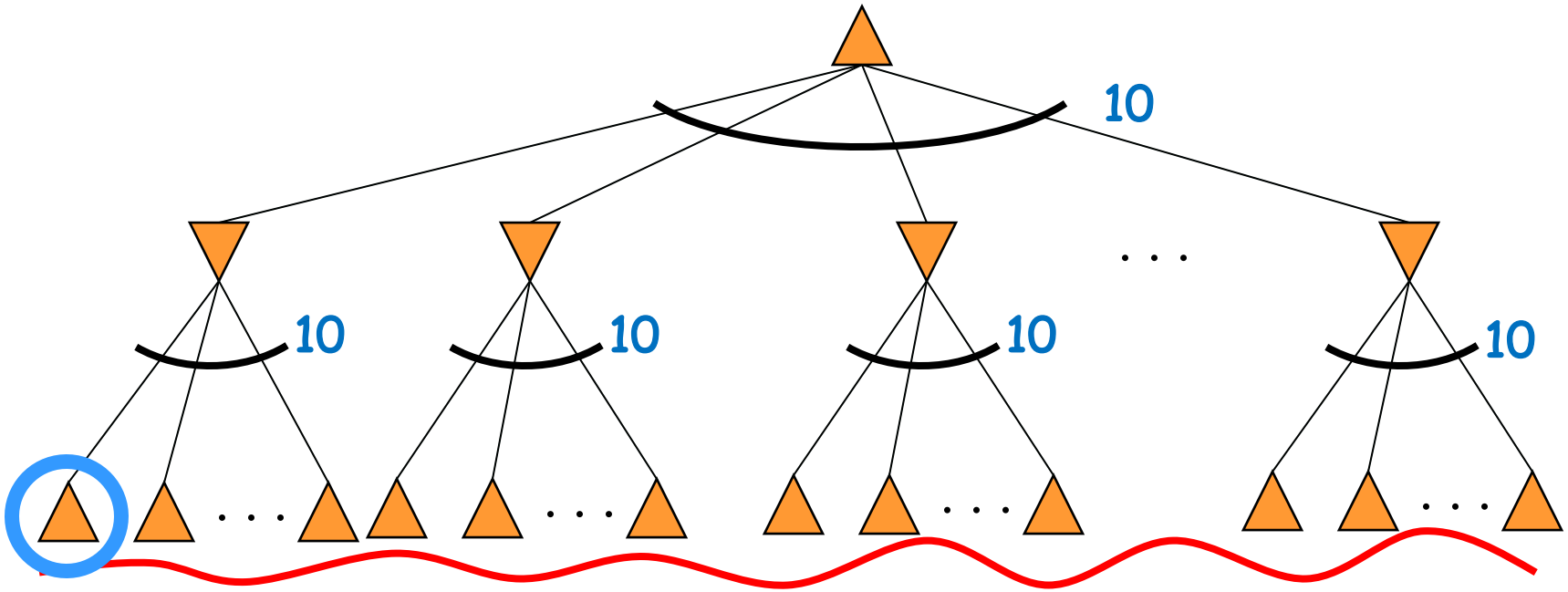
Juegos



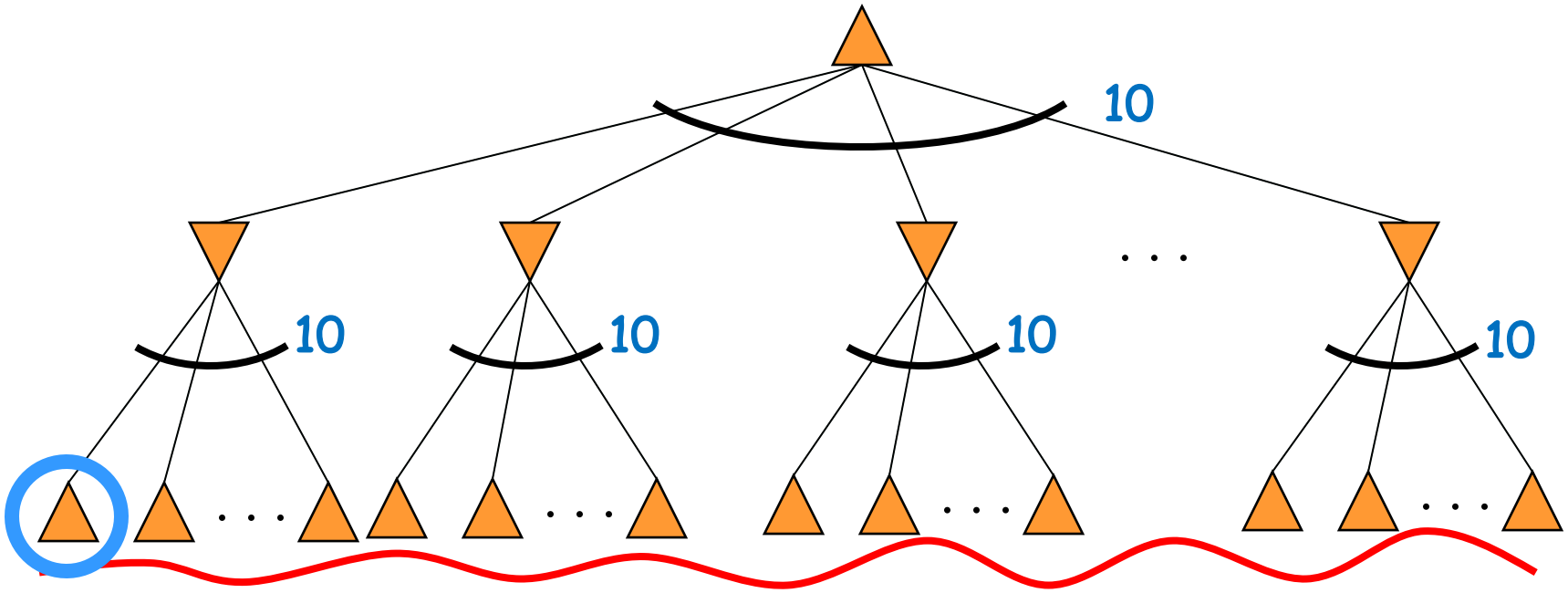
Juegos

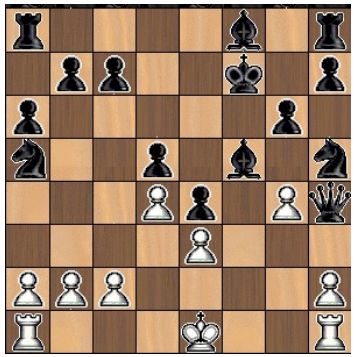


Juegos



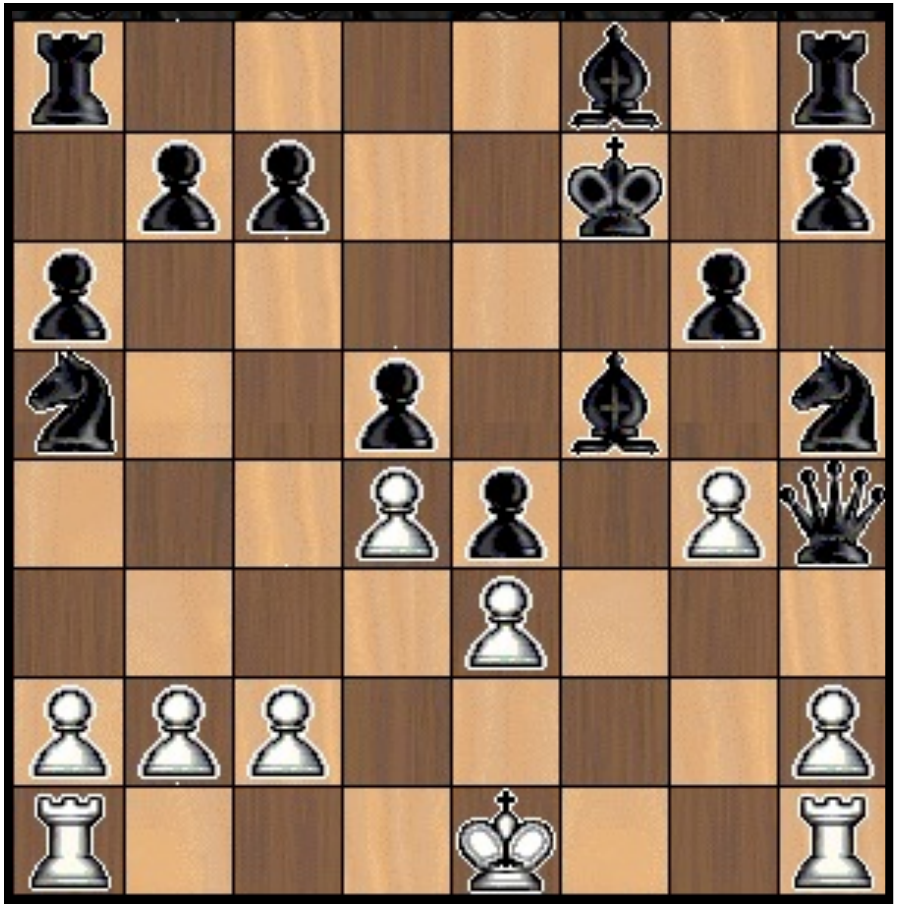
Juegos



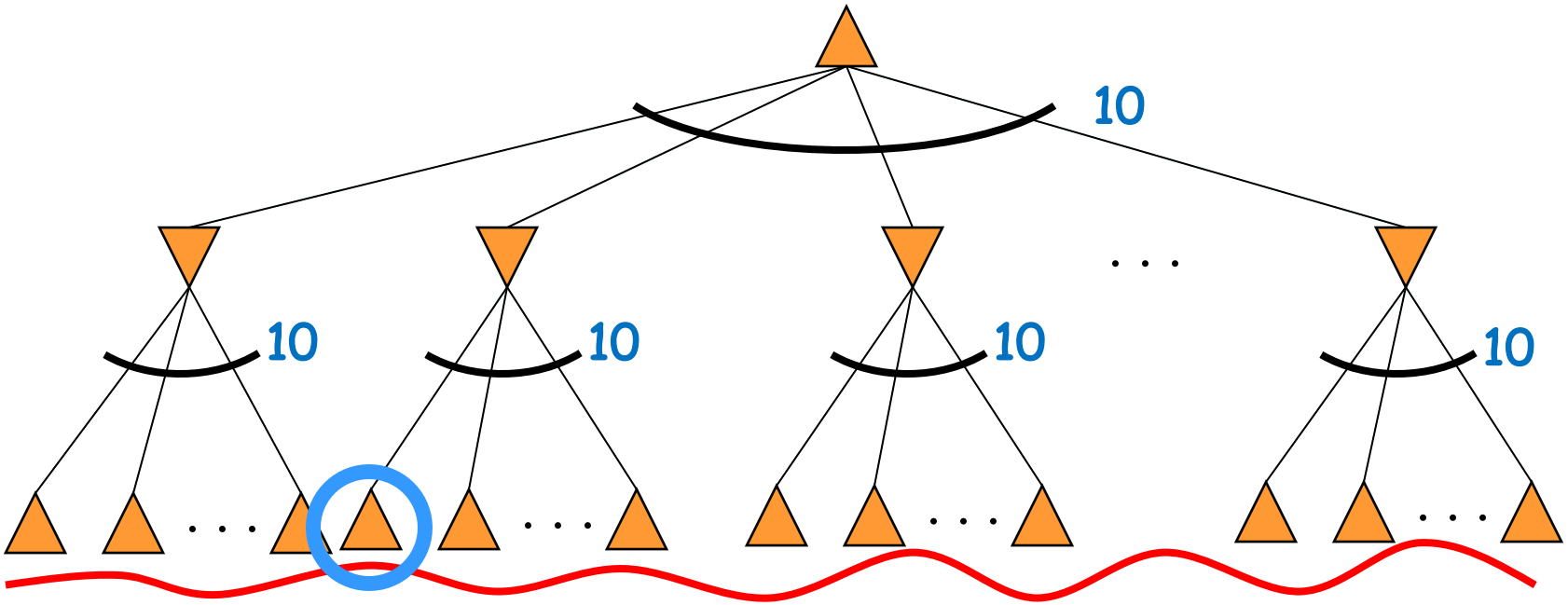


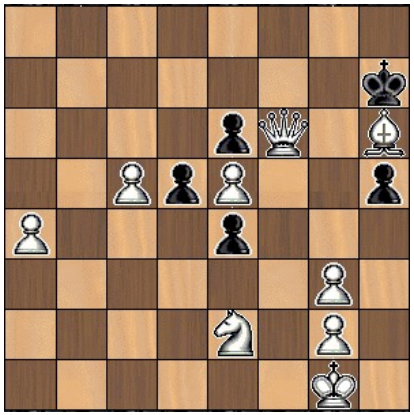
¿Cuál es la **utilidad** de este nodo para **MAX** que juega con blancas?

Juegos



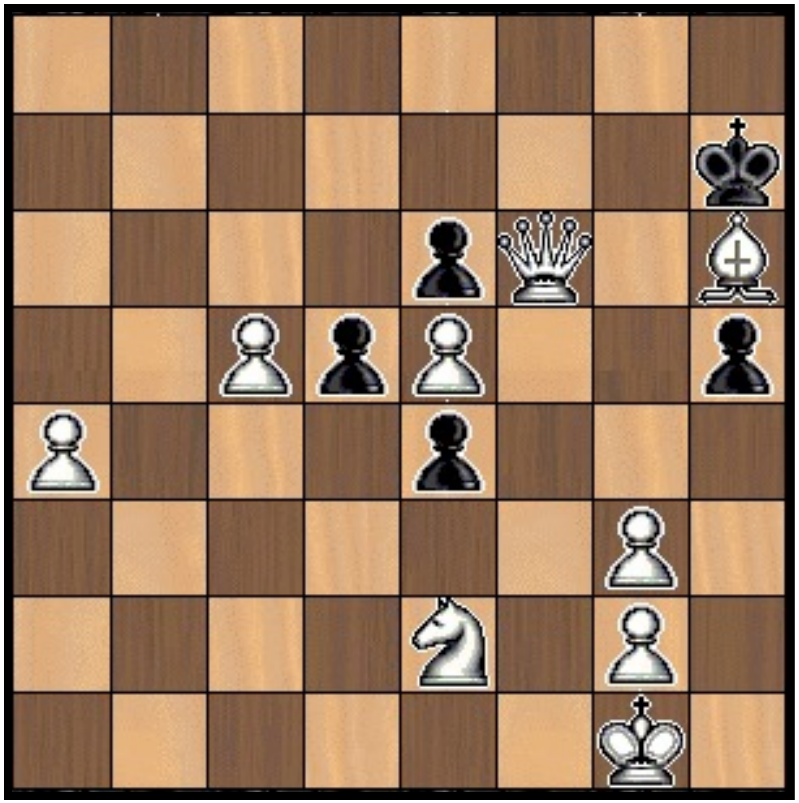
Juegos



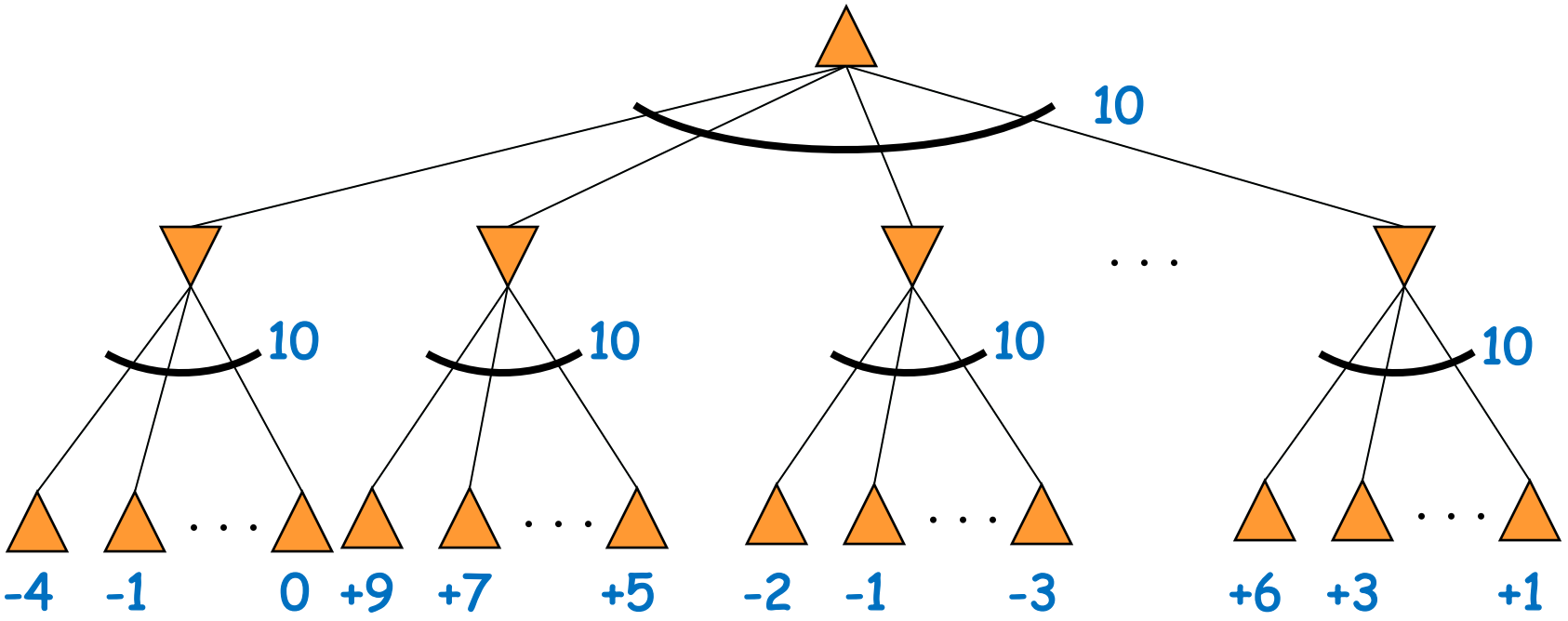


¿Cuál es la **utilidad** de este nodo para **MAX** que juega con blancas?

Juegos

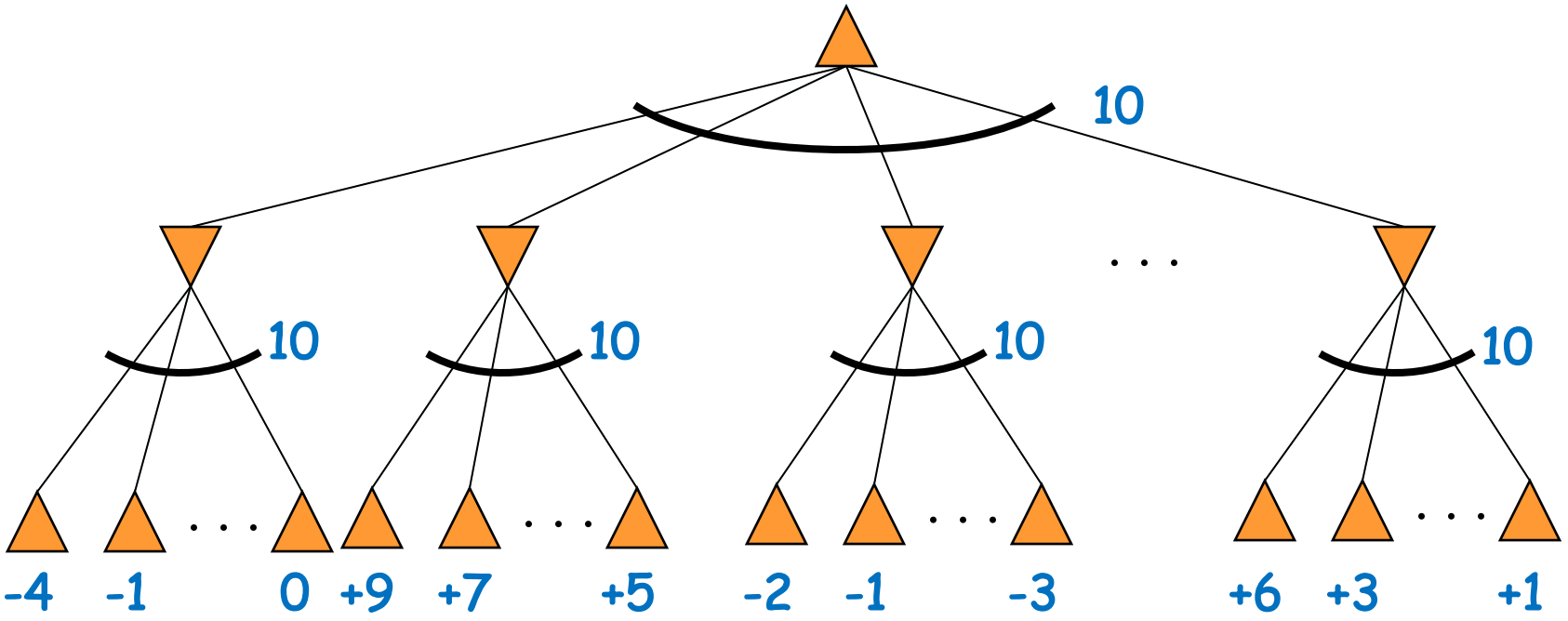


Juegos



Se estiman las utilidades y se toma la **decisión minimax** con base en esos valores

Juegos



Se <u>estiman</u> las utilidades y se toma la **decisión minimax** con base en esos valores

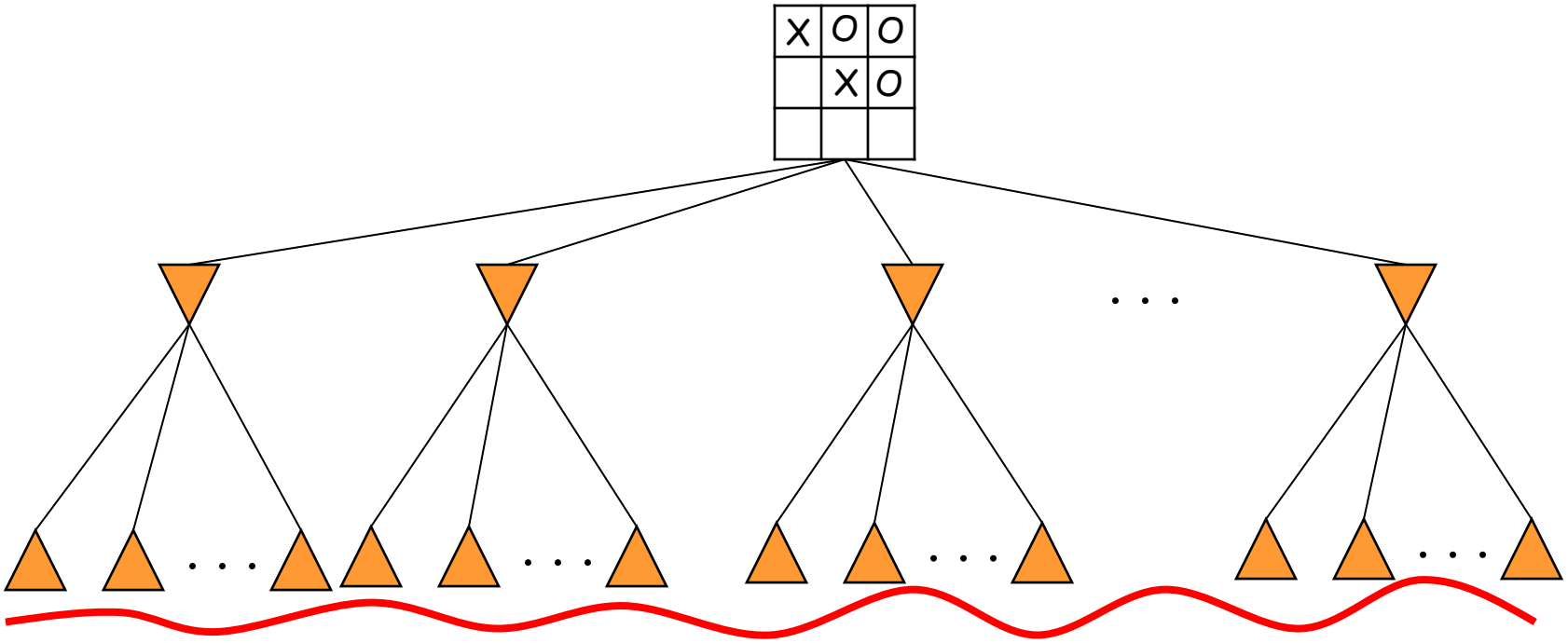
Algoritmo minimax con decisiones imperfectas

Juegos

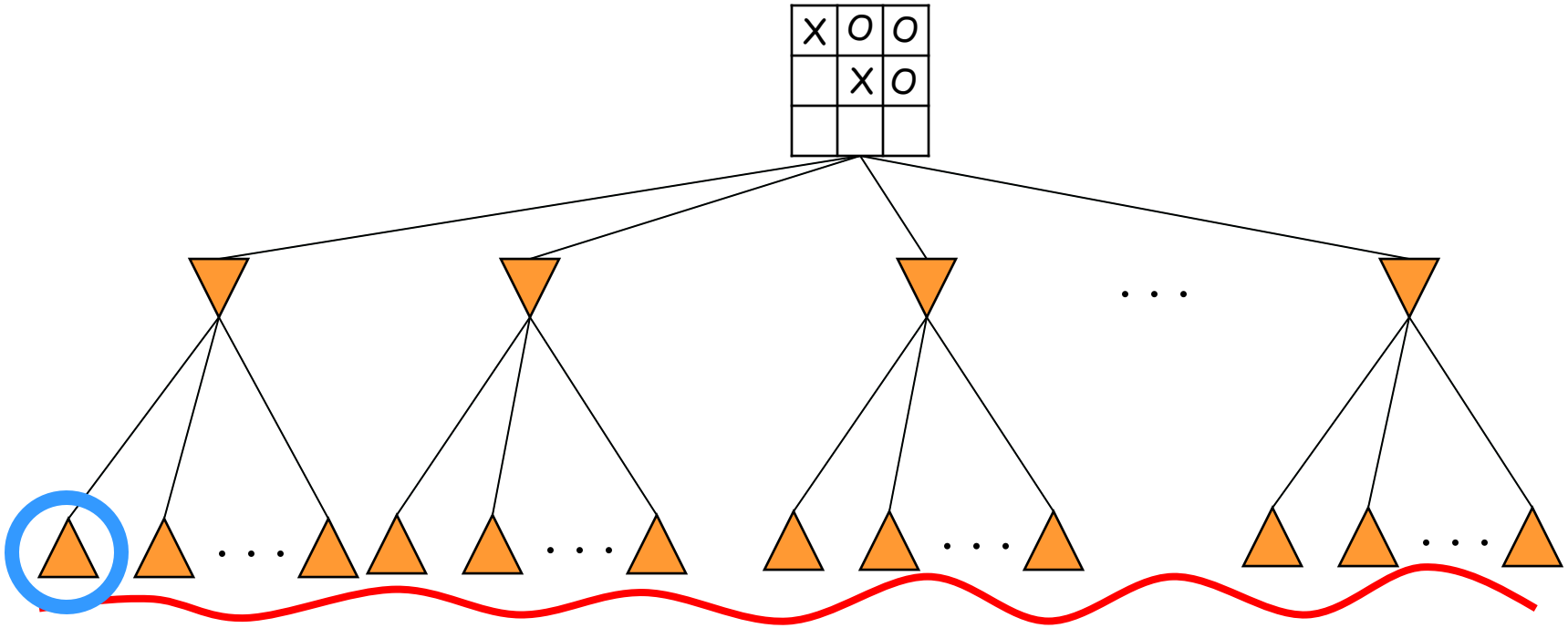
Algoritmo minimax con decisiones imperfectas

- En lugar de llegar hasta los nodos terminales, se **suspende** antes la búsqueda y se aplica sobre estos nodos una **función de utilidad heurística**

Juegos



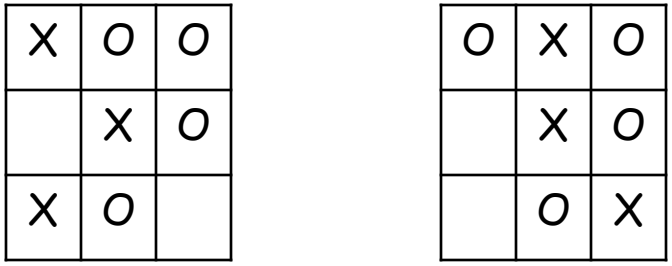
Juegos



X	O	O
	X	O
X	O	

¿Cuál es la **utilidad** de este nodo para **MAX** que juega con X?

Juegos



MAX juega con X. ¿En cuál de los dos estados el valor estimado de la utilidad debe ser mayor?

Juegos

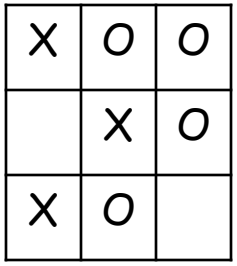
f(n): ???

La función debe **estimar la utilidad** para MAX estando en el nodo n

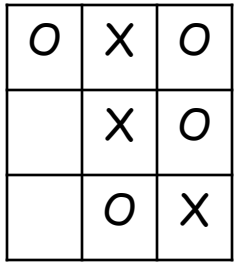
Juegos

f(n): cantidad de filas, columnas y diagonales libres para MAX

Juegos

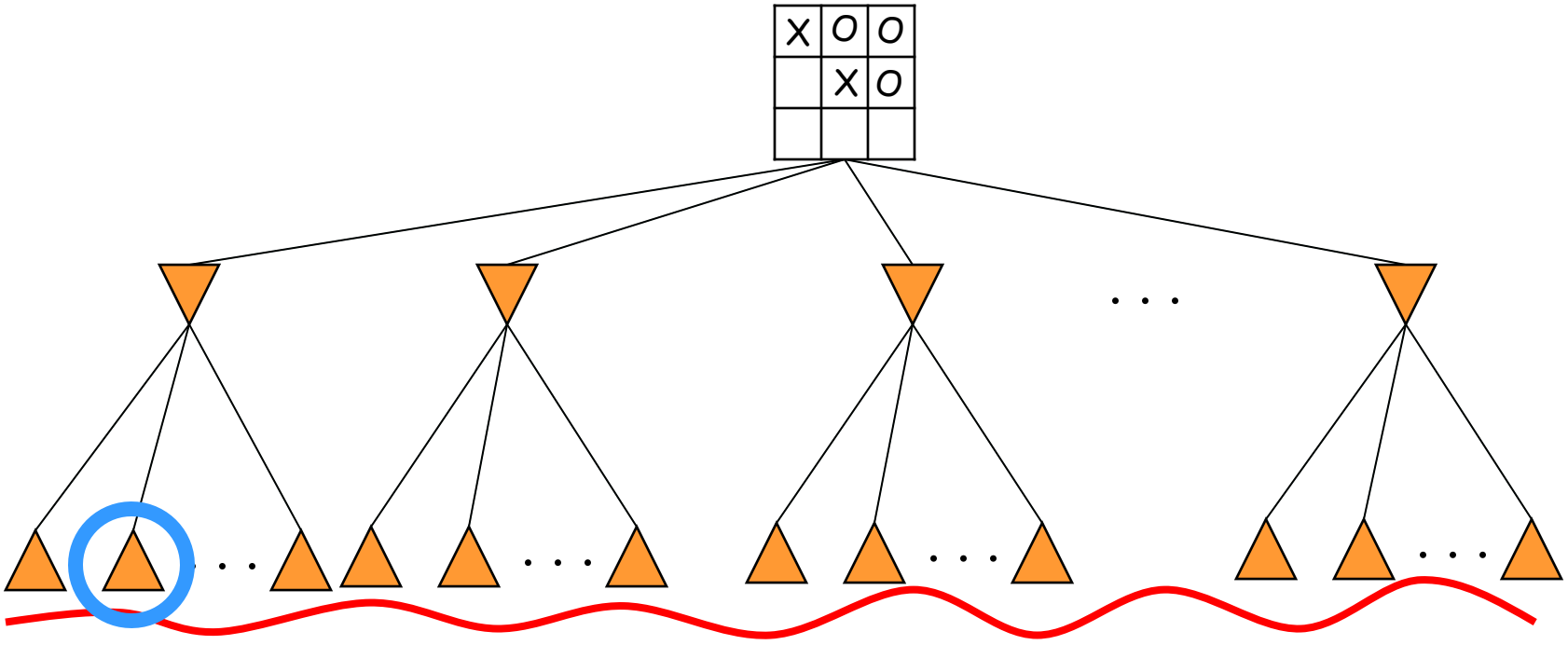


f(n): 2



f(n): 0

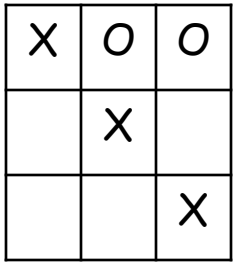
Juegos



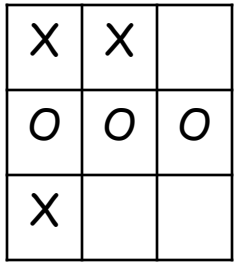
X	O	O
	X	O
X	O	

f(n): 2

Juegos

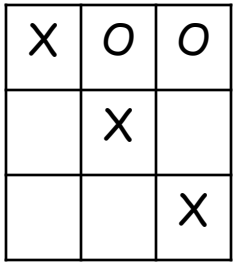


f(n): ?

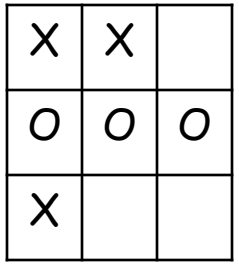


f(n): ?

Juegos

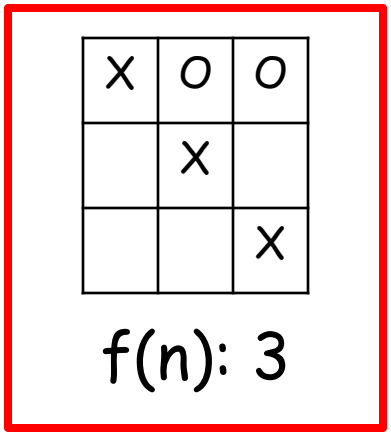


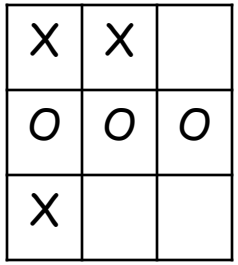
f(n): 3



f(n): 2

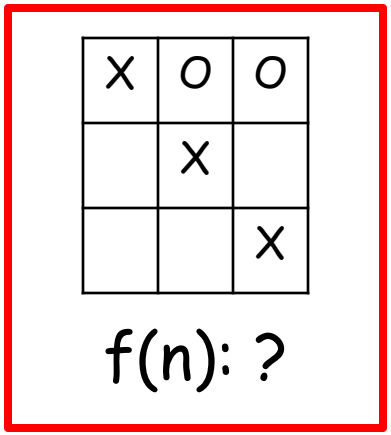
Juegos

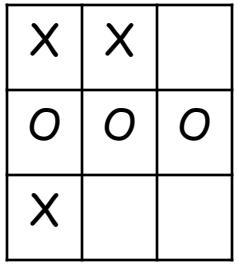




f(n)=3 pero no refleja que MAX ya ganó

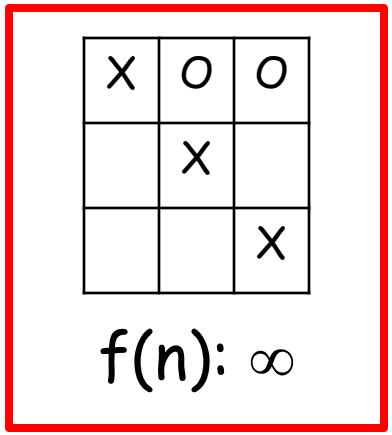
Juegos

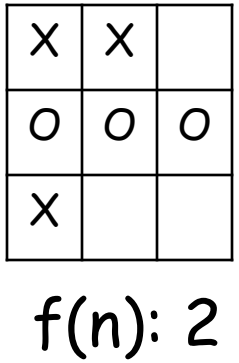




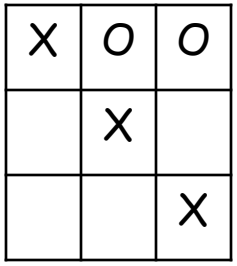
¿En un nodo donde MAX ya ganó,
qué valor se debería colocar a f(n)?

Juegos

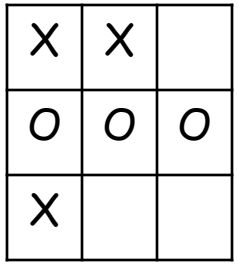




Juegos



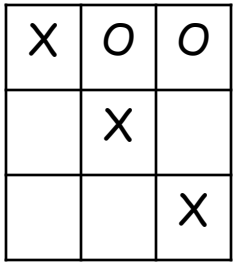
f(n): ∞



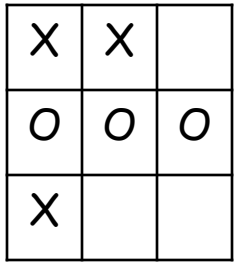
f(n): 2

f(n)=2 pero no refleja que MAX ya perdió

Juegos



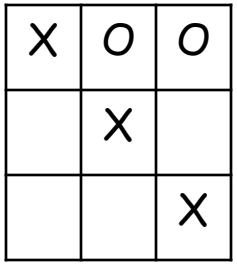
f(n): ∞



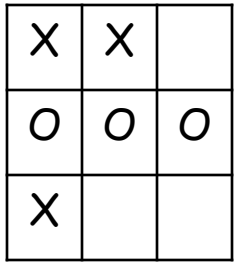
f(n): ?

¿En un nodo donde MAX ya perdió, qué valor se debería colocar a f(n)?

Juegos



f(n): ∞

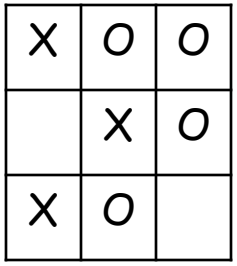


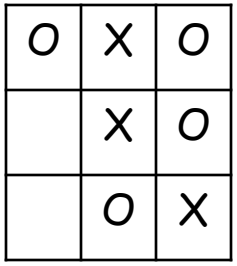
f(n): -∞

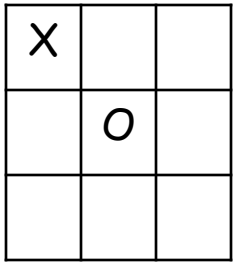
Juegos

$$f(n) = \begin{cases} \text{(número de filas, columnas o diagonales libres para MAX), si el nodo n no es un estado en el que gane alguno de los jugadores} \\ \infty, \text{ si gana MAX} \\ -\infty, \text{ si gana MIN} \end{cases}$$

Juegos

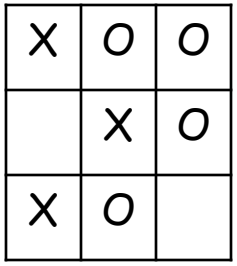


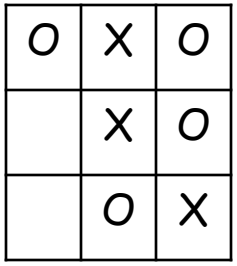


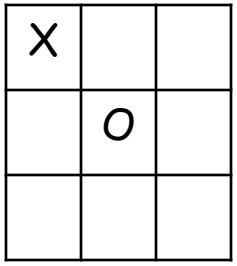


f(n): ?

Juegos

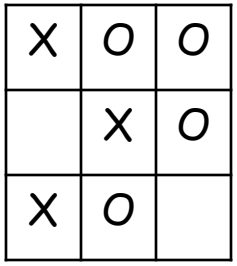


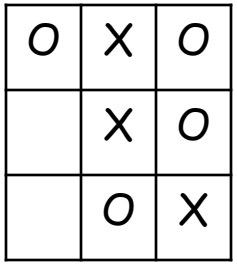


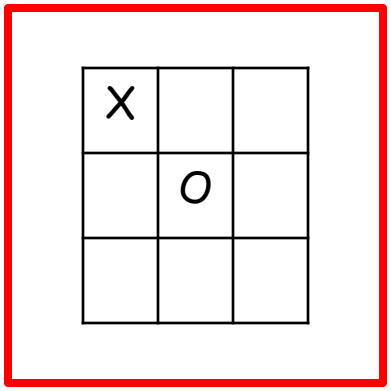


f(n): 4

Juegos







f(n)=4 pero no refleja que MIN tiene más formas de hacer triqui

Juegos

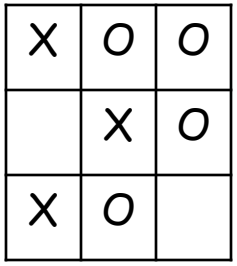
$f(n) =$ {

(número de filas, columnas o diagonales libres para MAX) - (número de filas, columnas o diagonales libres para MIN), si el nodo n no es un estado en el que gane alguno de los jugadores

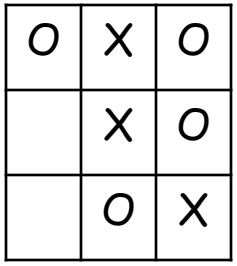
∞, si gana MAX

$-\infty$, si gana MIN

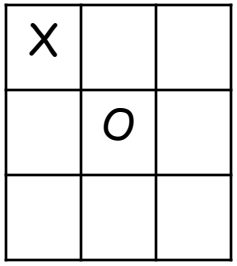
Juegos



f(n)=2-1



f(n)=0-1



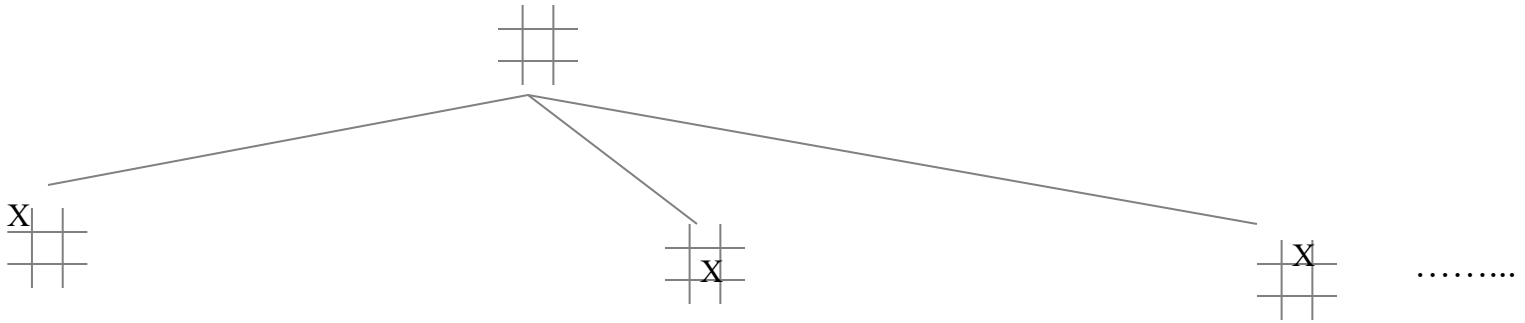
f(n)=4-5

Juegos

- Aplique minimax con profundidad 2

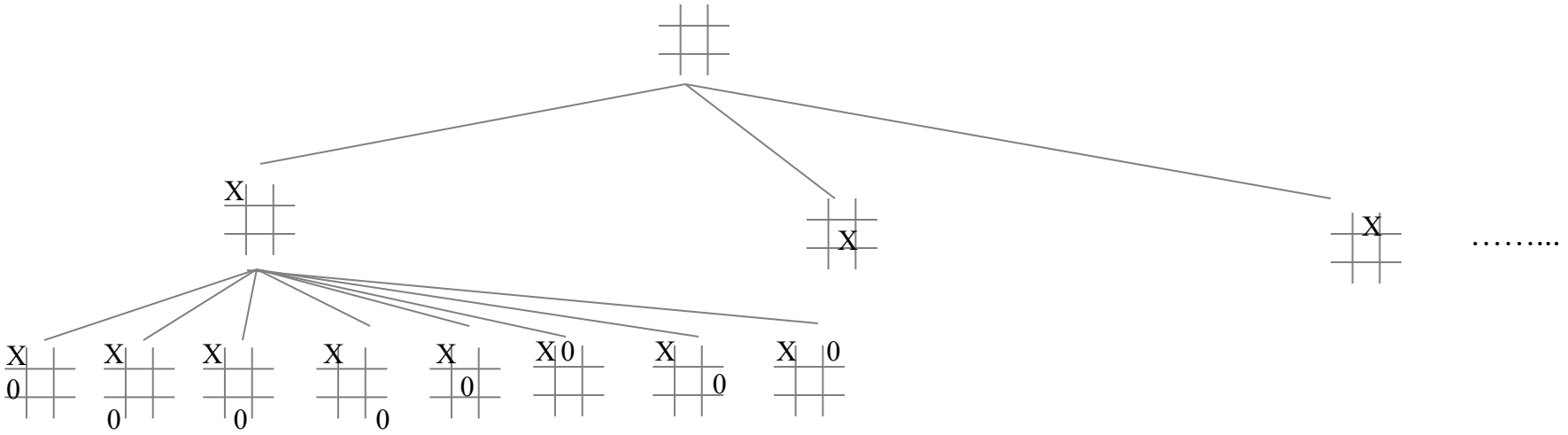
Juegos

- Aplique minimax con profundidad 2



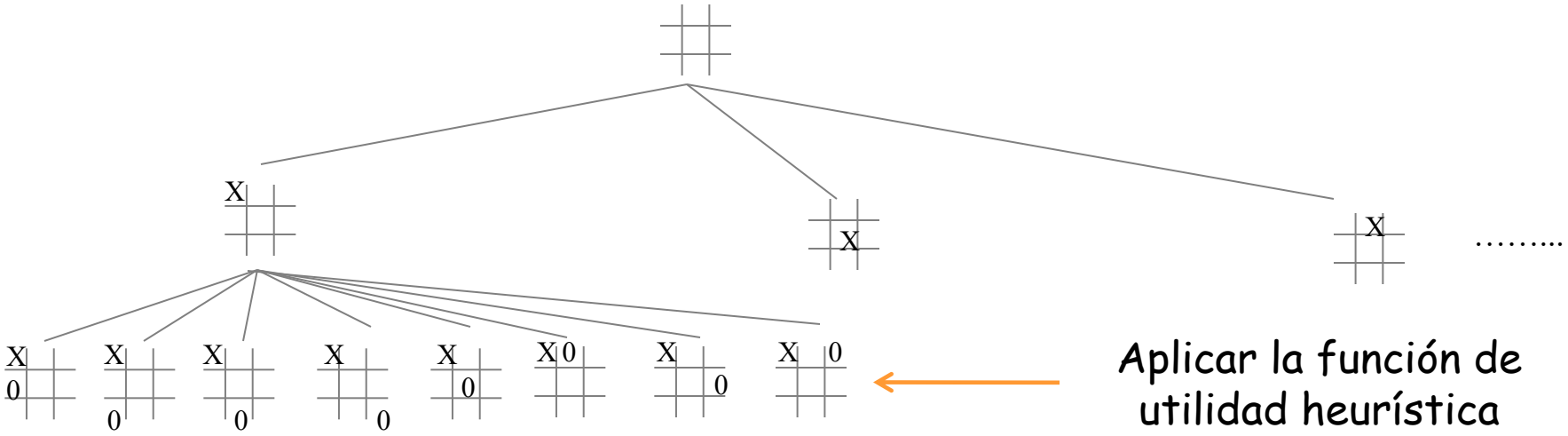
Juegos

- Aplique minimax con profundidad 2



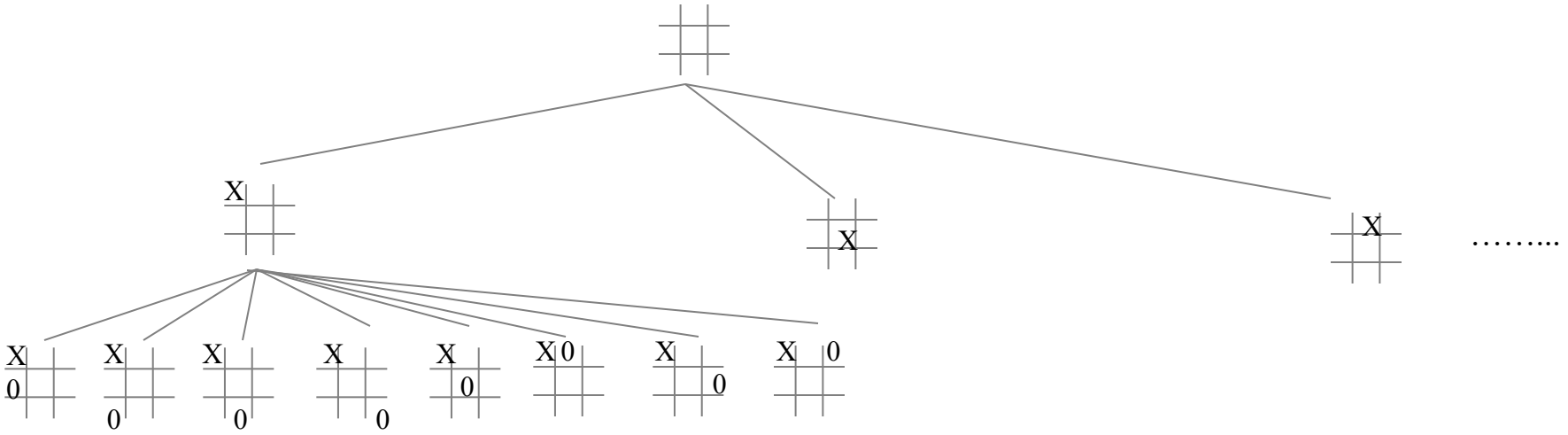
Juegos

- Aplique minimax con profundidad 2



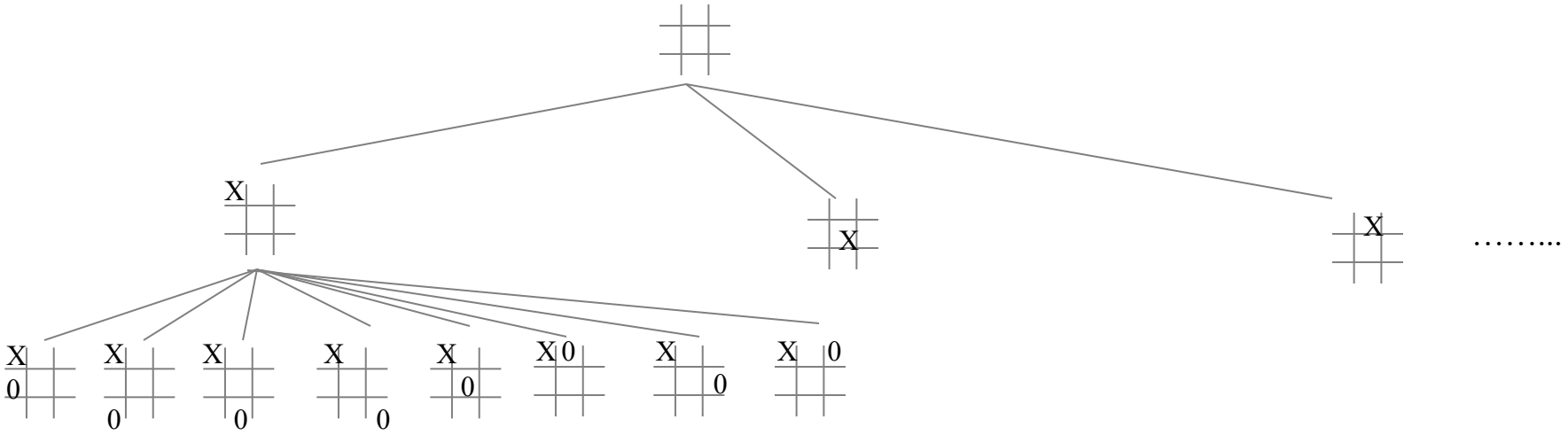
Juegos

- Aplique minimax con profundidad 2



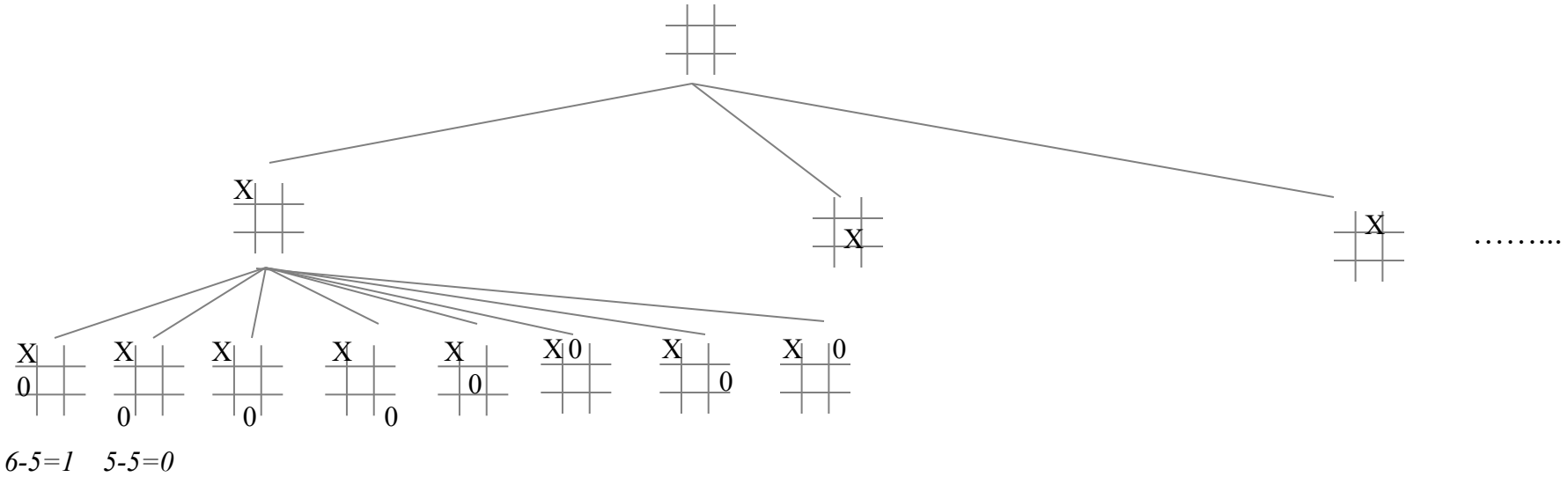
Juegos

- Aplique minimax con profundidad 2



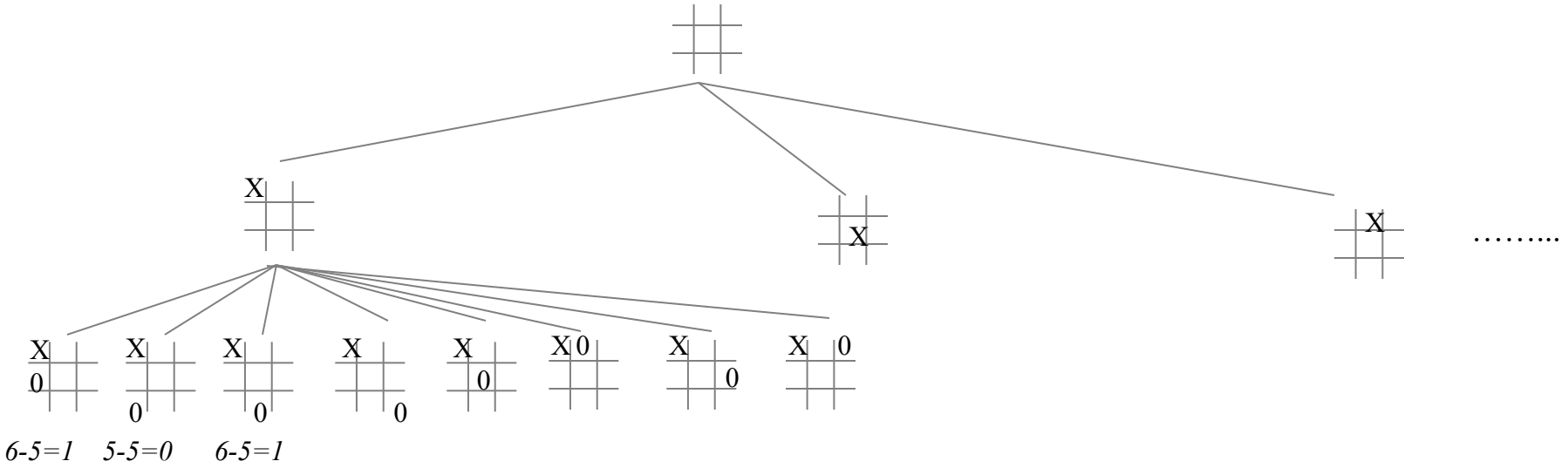
Juegos

- Aplique minimax con profundidad 2



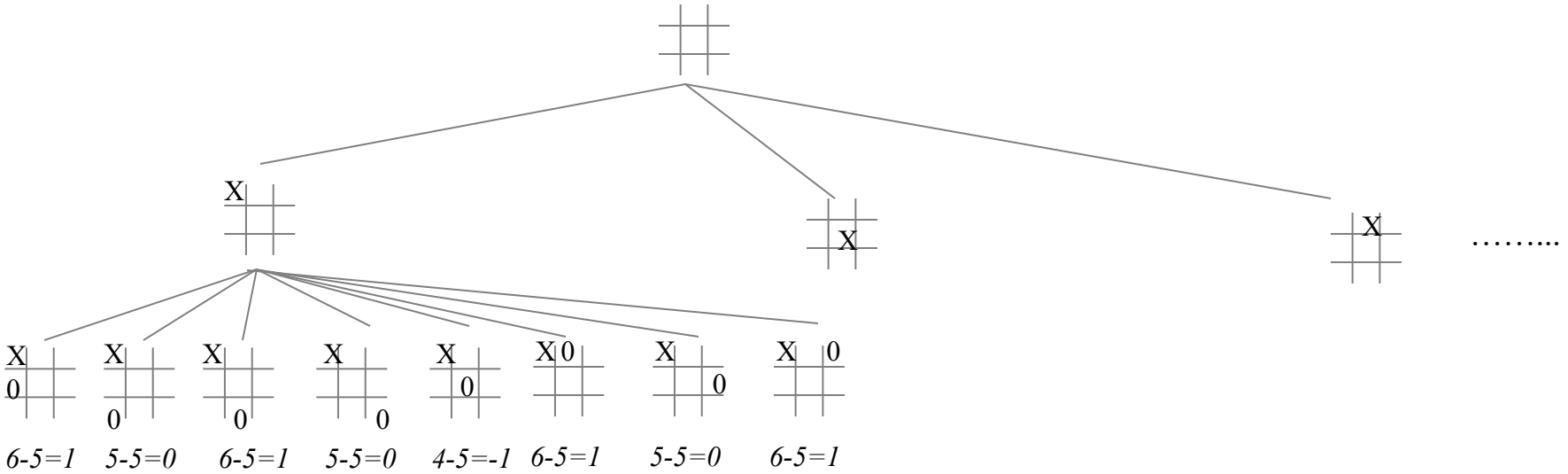
Juegos

- Aplique minimax con profundidad 2



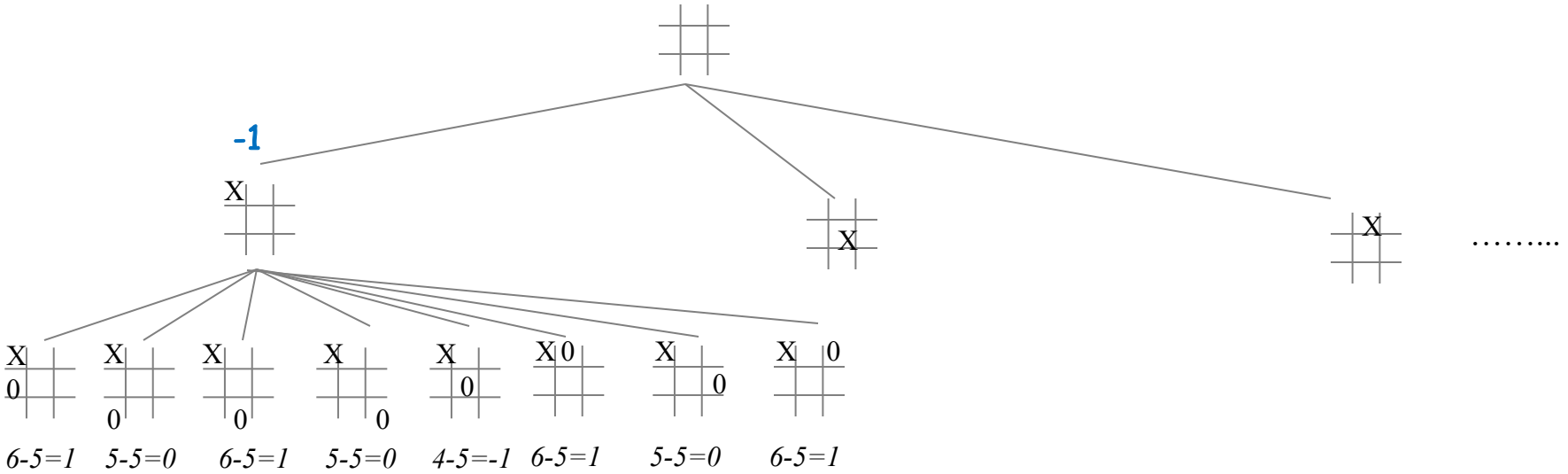
Juegos

- Aplique minimax con profundidad 2



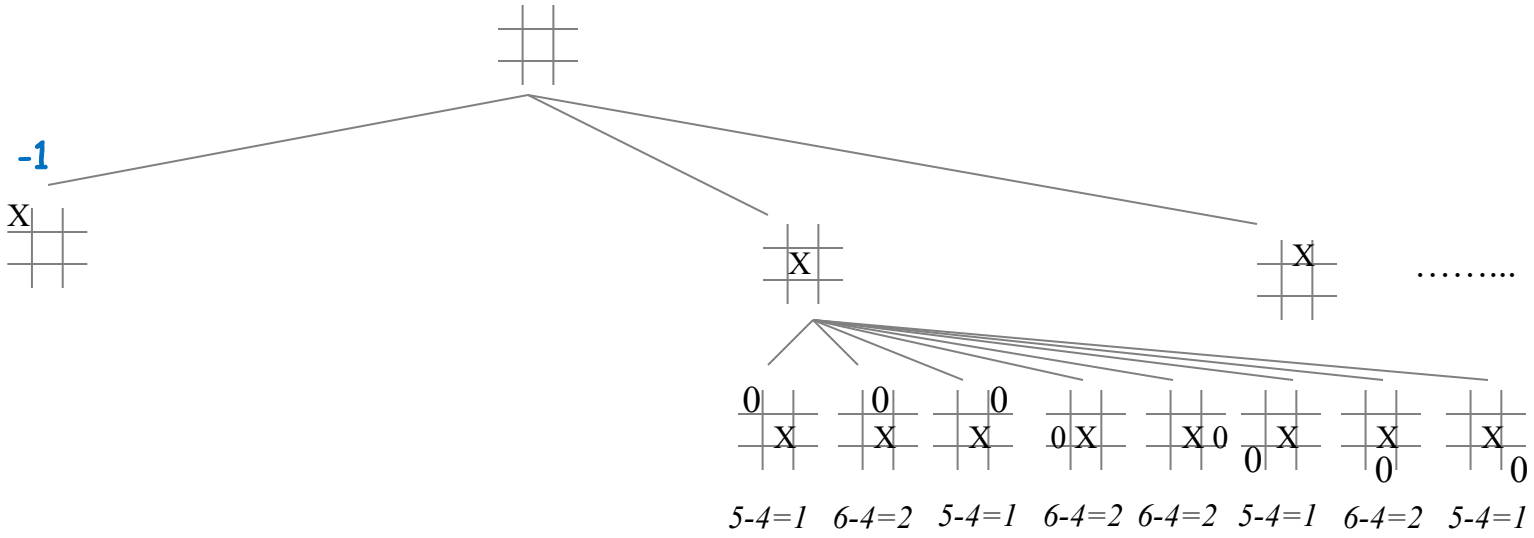
Juegos

- Aplique minimax con profundidad 2



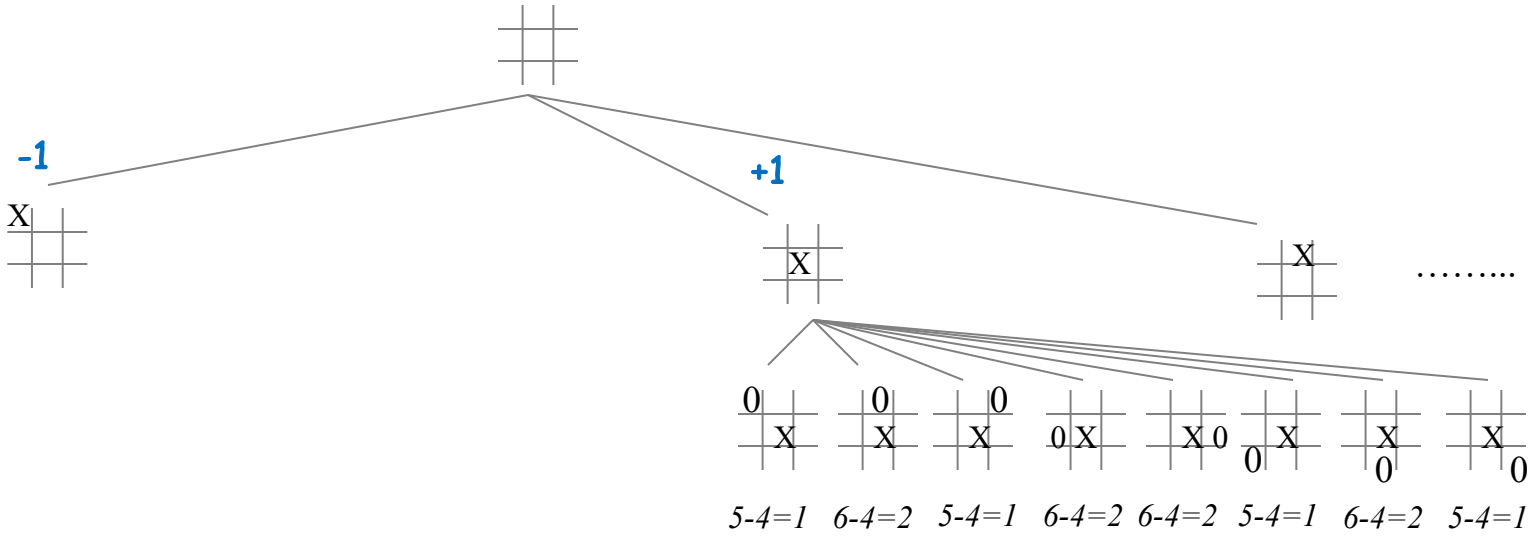
Juegos

- Aplique minimax con profundidad 2



Juegos

- Aplique minimax con profundidad 2

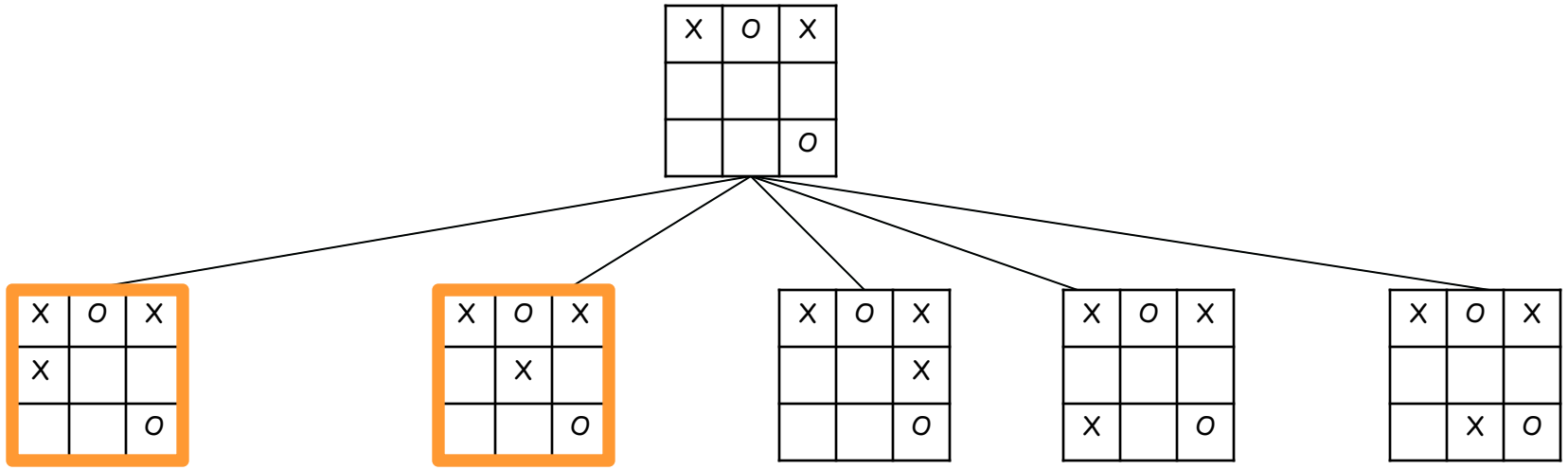


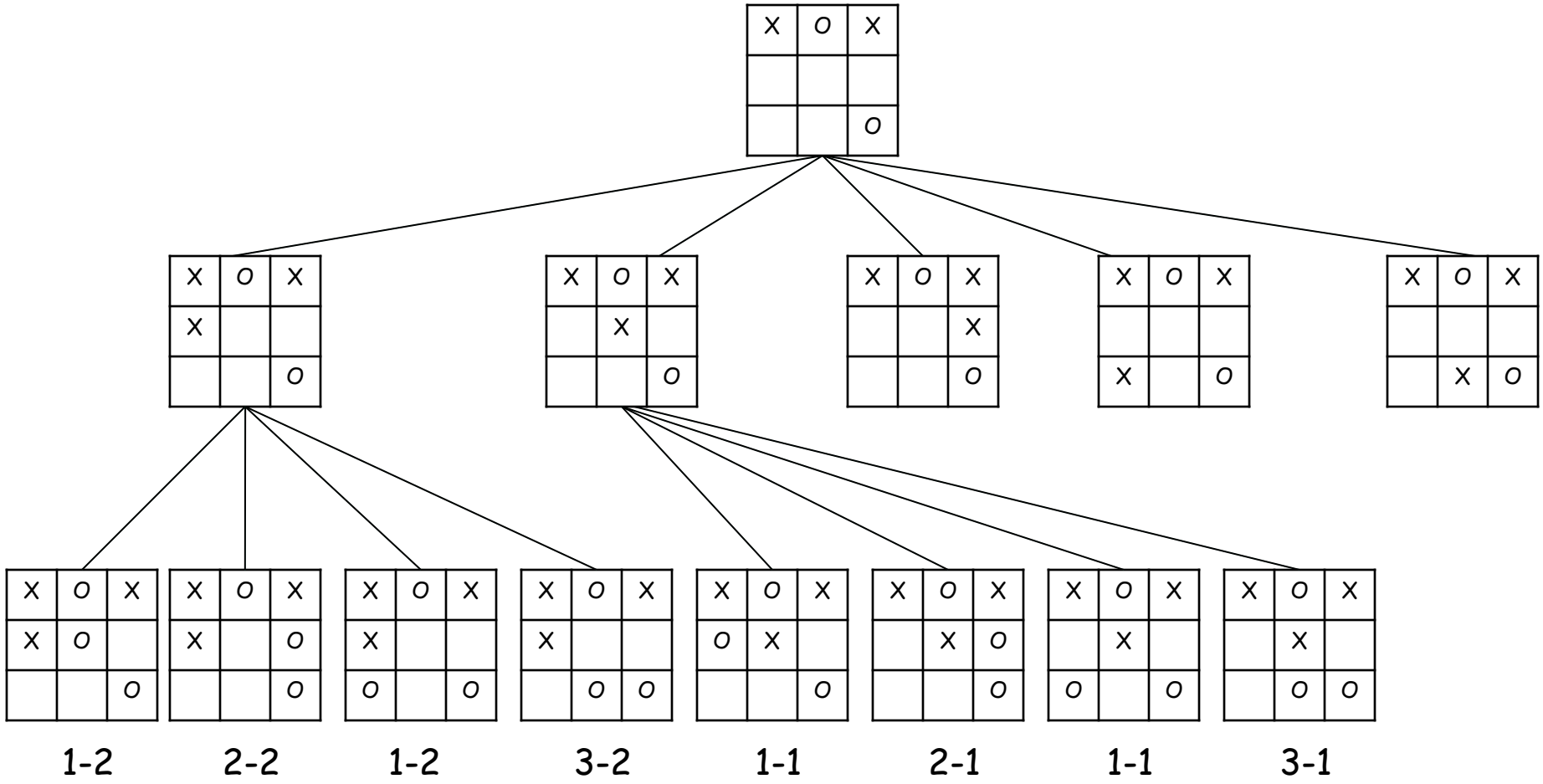
Juegos

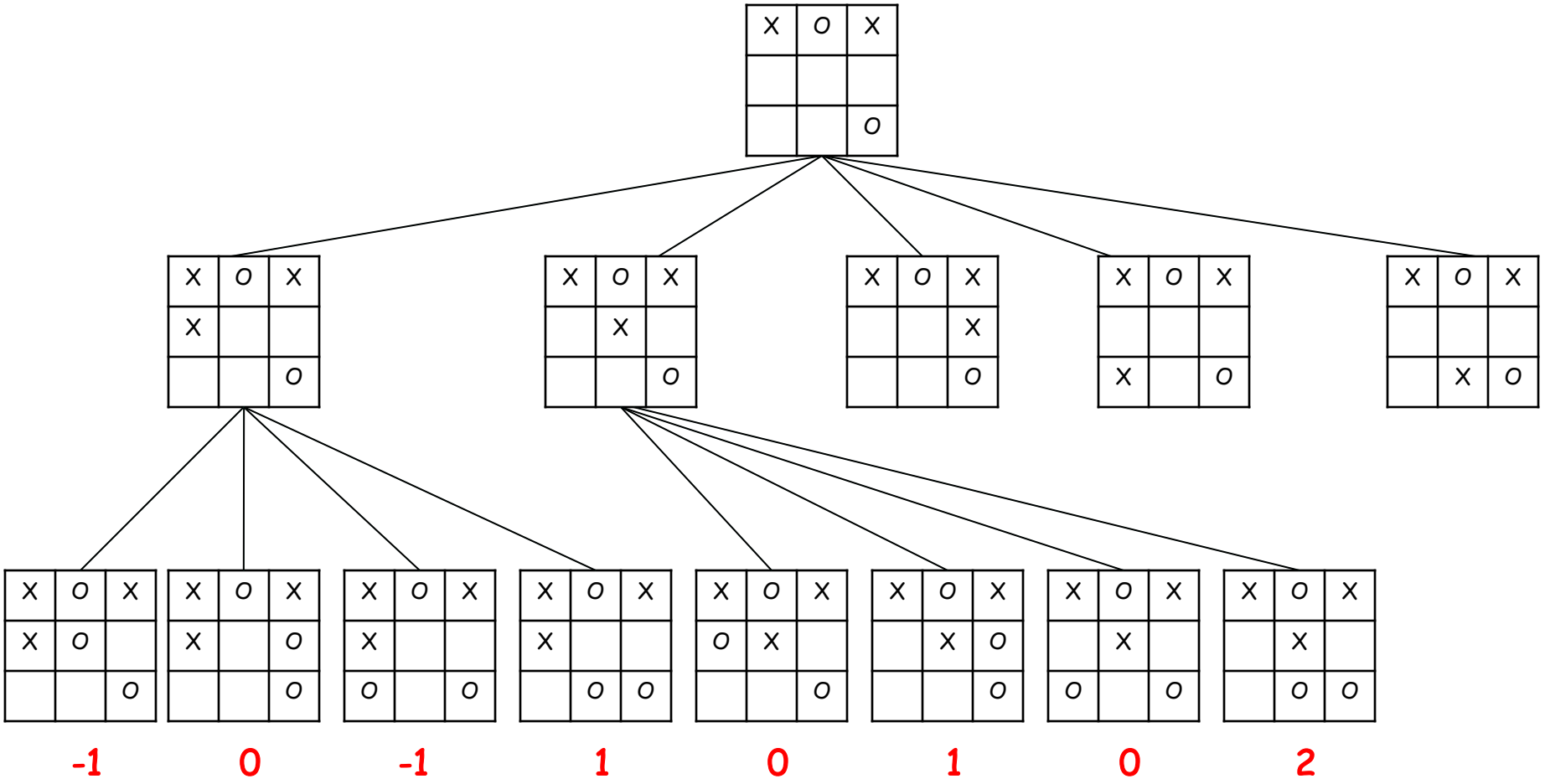
Decisiones imperfectas

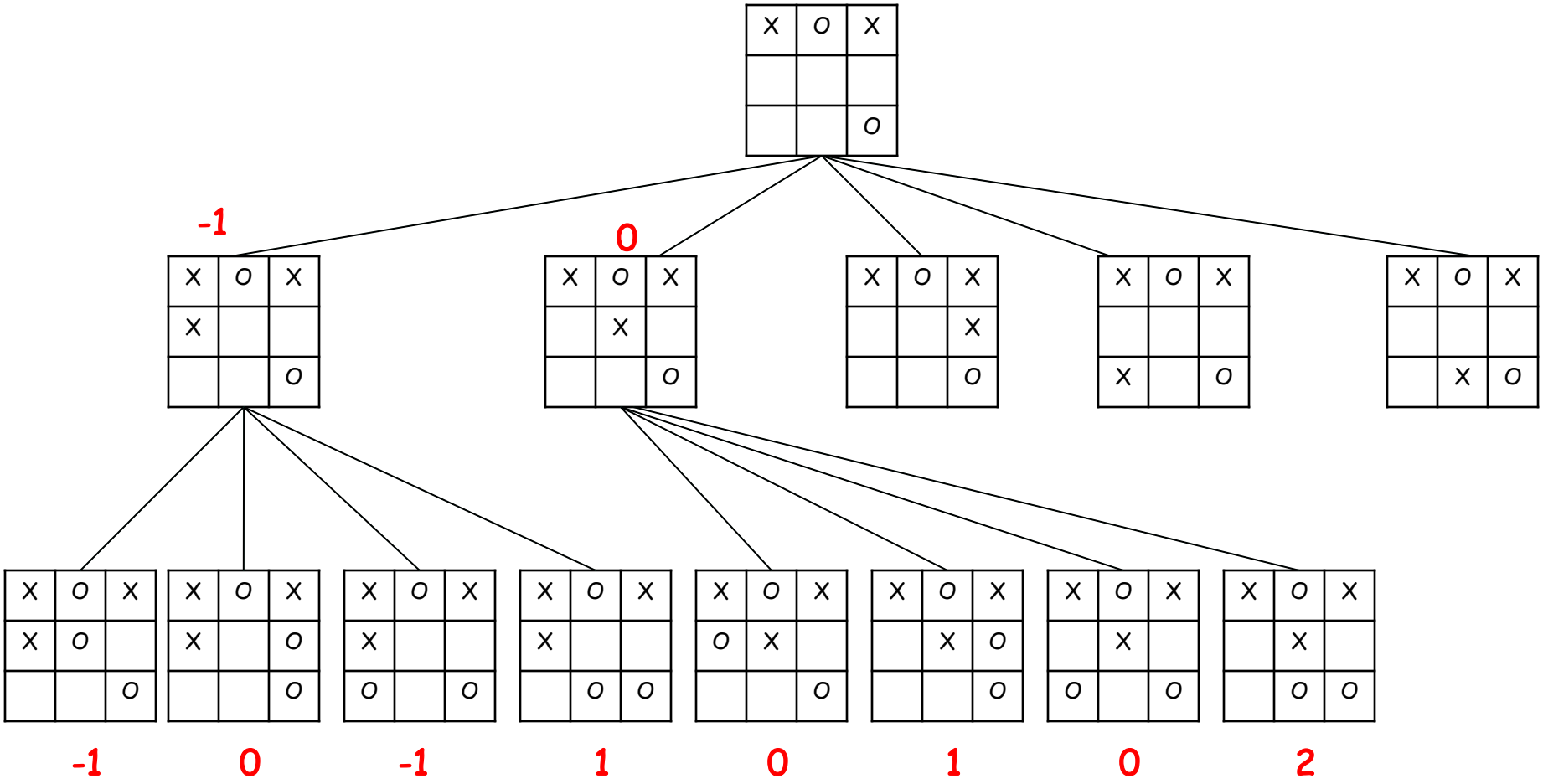
X	O	X
		O

- Aplique minimax con profundidad 2
- La jugada es de MAX(x)


X O X
O
X O X
X
O
X O X
X
O
X O X
X
O
X O X
X
O
X O X
X
O

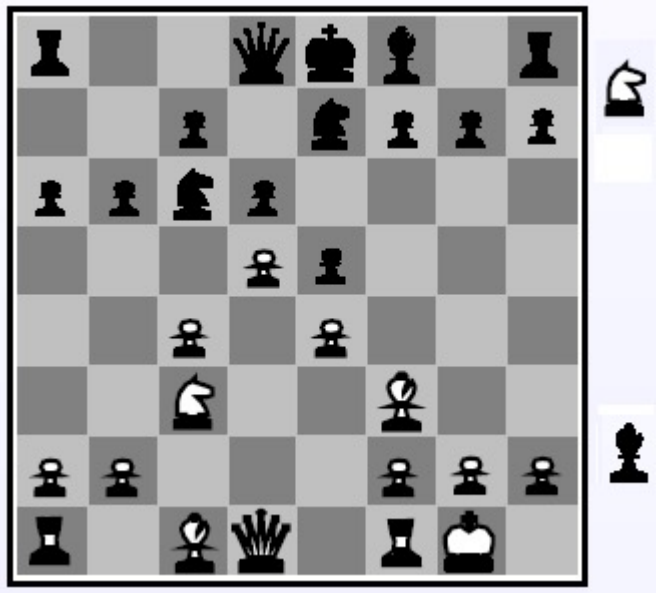

1-2
2-2
1-2
3-2
1-1
2-1
1-1
3-1


-1
0
-1
1
0
1
0
2


-1
0
-1
0
1
0
1
0
2

Juegos

Decisiones imperfectas



Cada pieza tiene un valor:

- **Peón**: 1
- **Caballo**: 3
- **Alfil**: 3
- **Torre**: 5
- **Reina**: 9

Juegos

Decisiones imperfectas

$f_1(n)$: valor material de las piezas que MAX ha tomado de su contrincante

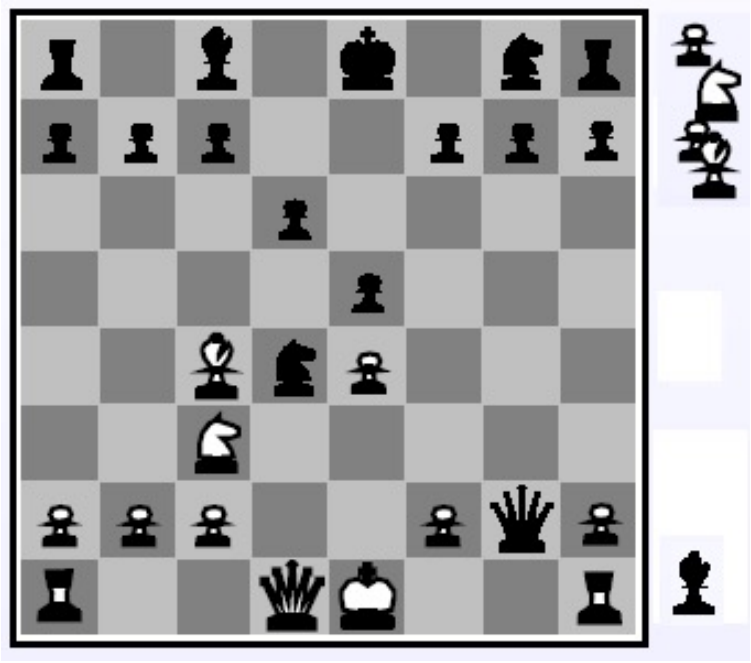
Juegos

Decisiones imperfectas

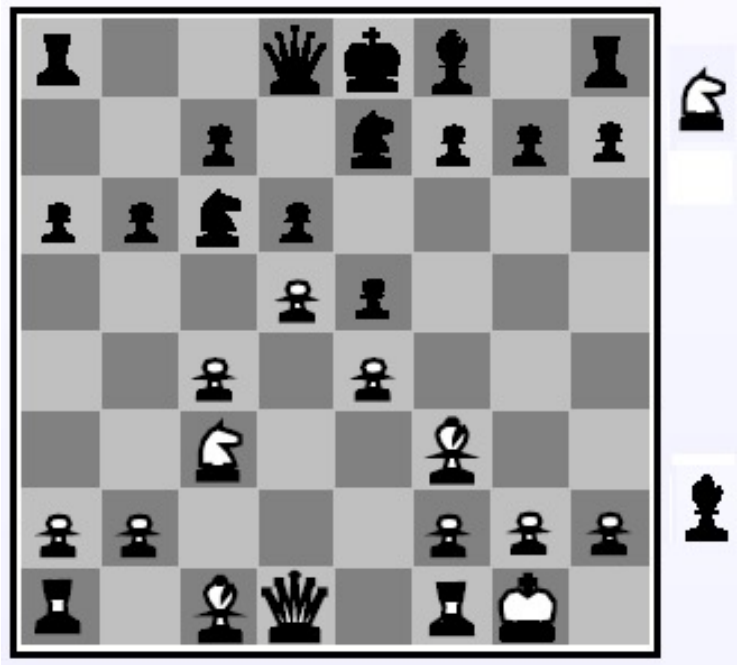
Suponga que MAX juega con blancas

Cada pieza tiene un valor:

Peón:1 **Caballo**:3 **Alfil**:3 **Torre**:5 **Reina**:9



$f_1(n)=?$

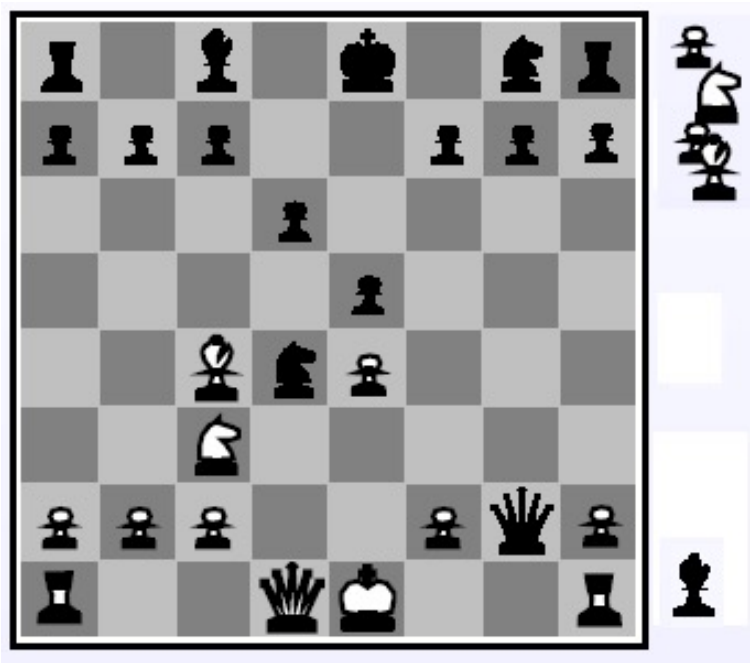


$f_1(n)=?$

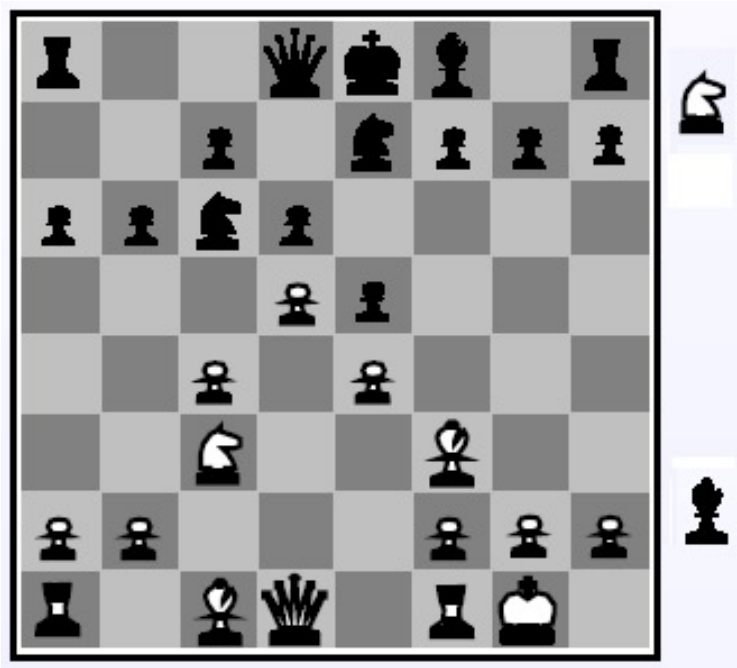
Decisiones imperfectas

Cada pieza tiene un valor:

Peón:1 **Caballo**:3 **Alfil**:3 **Torre**:5 **Reina**:9



$f_1(n)=3$



$f_1(n)=3$

Juegos

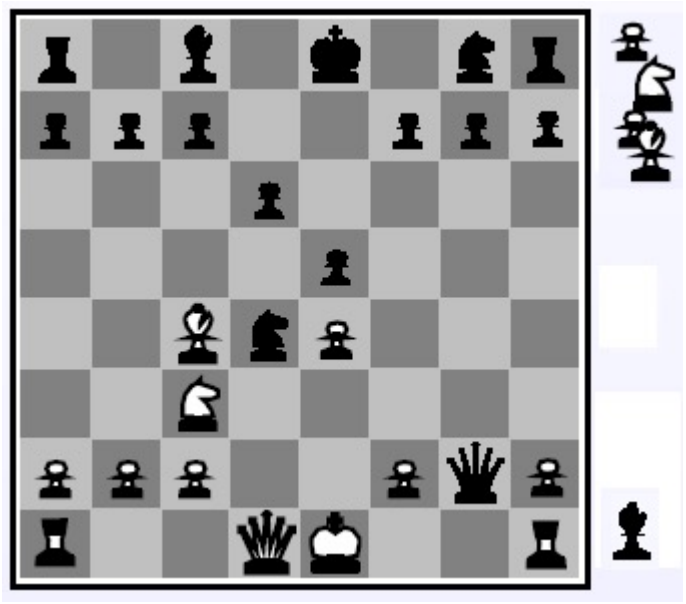
Decisiones imperfectas

$f_2(n)$: valor material de las piezas que MAX ha tomado de su contrincante – valor material de las piezas que MAX ha perdido

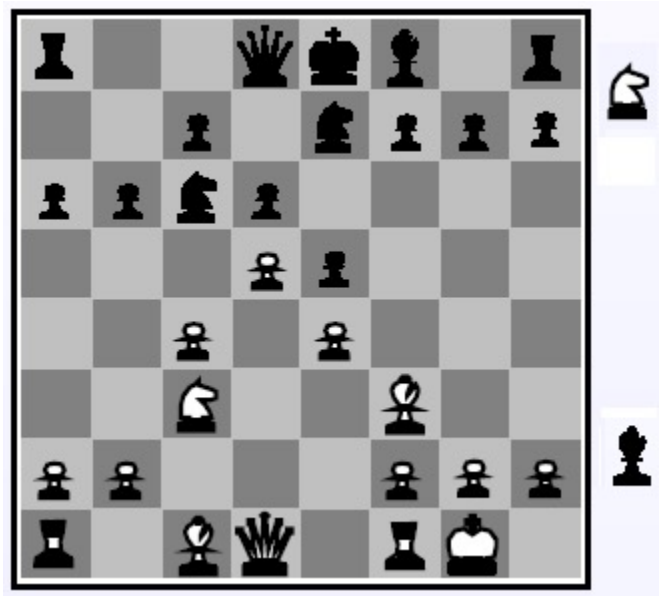
Decisiones imperfectas

Cada pieza tiene un valor:

Peón:1 **Caballo**:3 **Alfil**:3 **Torre**:5 **Reina**:9



$f_2(n)=3-8=-5$

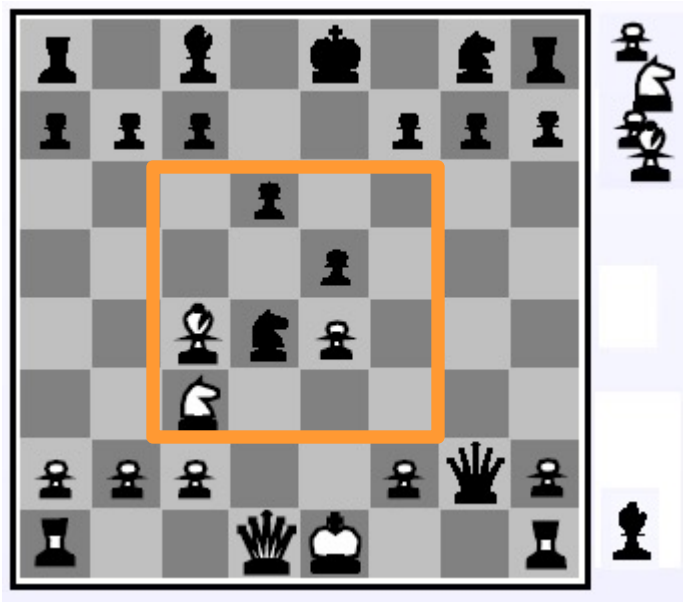


$f_2(n)=3-3=0$

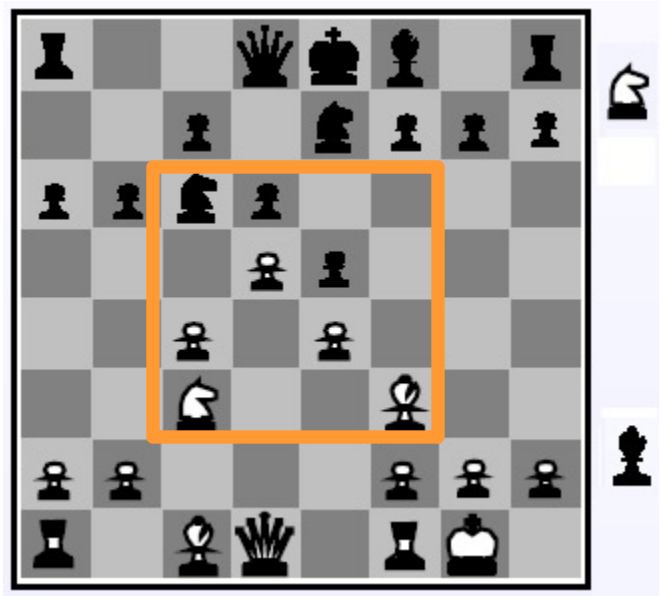
Decisiones imperfectas

Cada pieza tiene un valor:

Peón:1 **Caballo**:3 **Alfil**:3 **Torre**:5 **Reina**:9



$f_2(n)=3-8=-5$

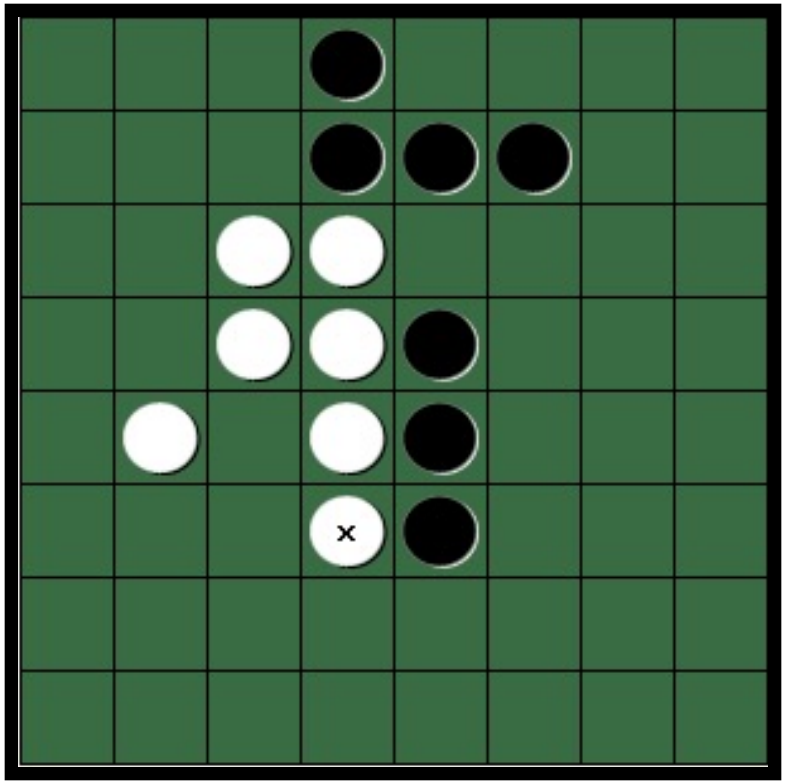


$f_2(n)=3-3=0$

Juegos

Reversi

¿Qué **función de utilidad heurística** utilizaría?

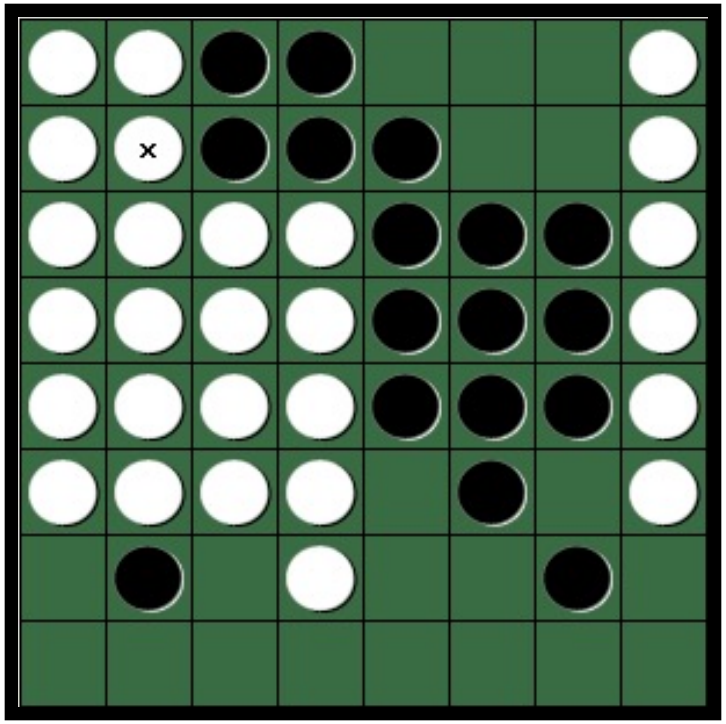


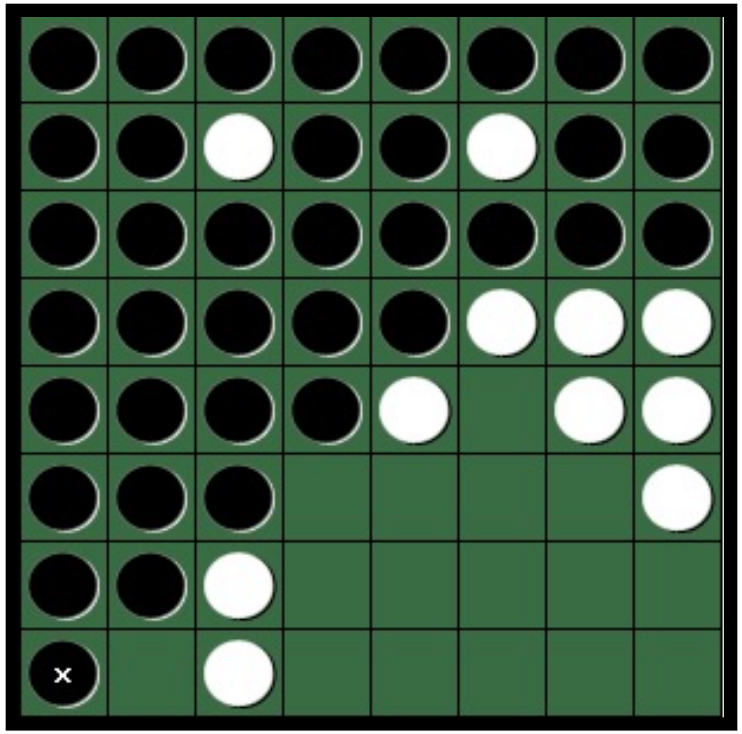
https://playpager.com/jugar-reversi/index.html

https://github.com/eigenfoo/otto-othello

Juegos

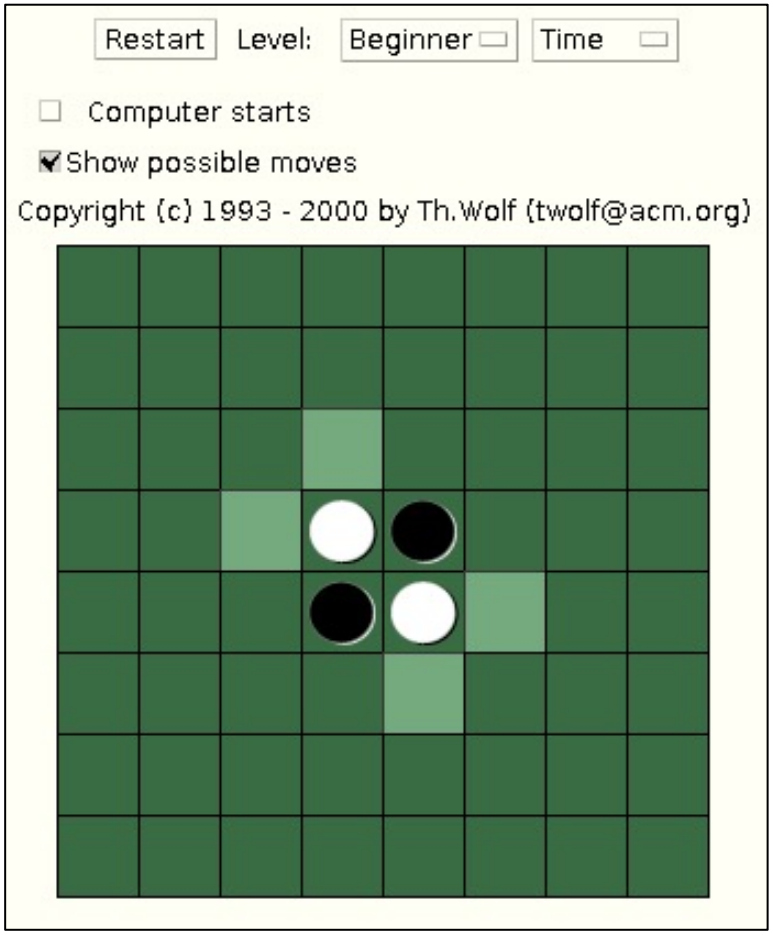
Reversi





Juegos

Reversi



Level "**Easy**" 1-move look-ahead
Level "**Beginner**" 2-move look-ahead
Level "**Amateur**" 4-move look-ahead
Level "**Expert**" 6-move look-ahead
(may take a couple of minutes to compute its next move)

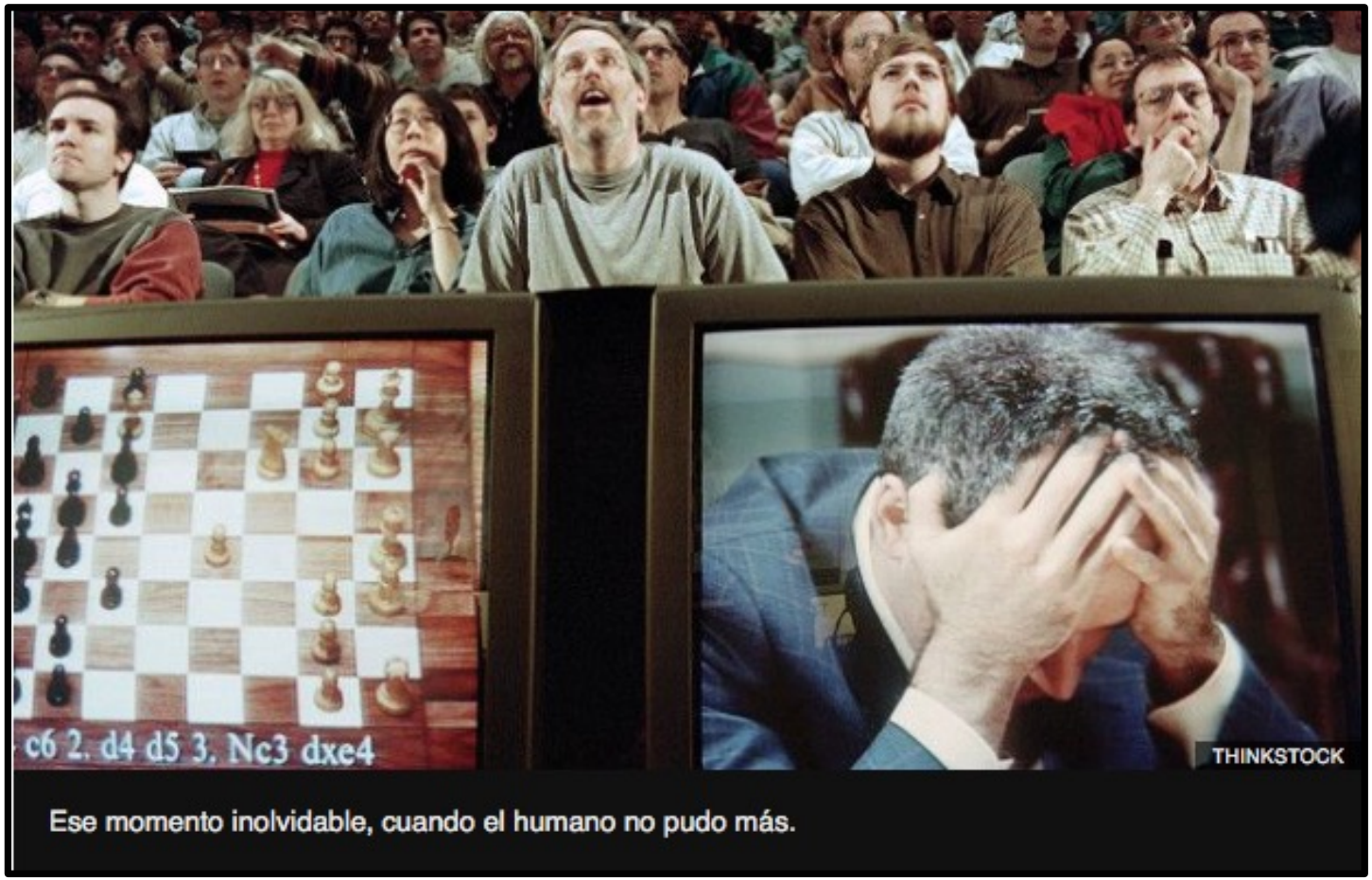
Juegos

Ajedrez

• 1997. El superordenador **IBM Deep Blue** derrotó al campeón mundial Garry Kasparov

Juegos

Juegos


c6 2. d4 d5 3. Nc3 dxe4
THINKSTOCK
Ese momento inolvidable, cuando el humano no pudo más.

Juegos

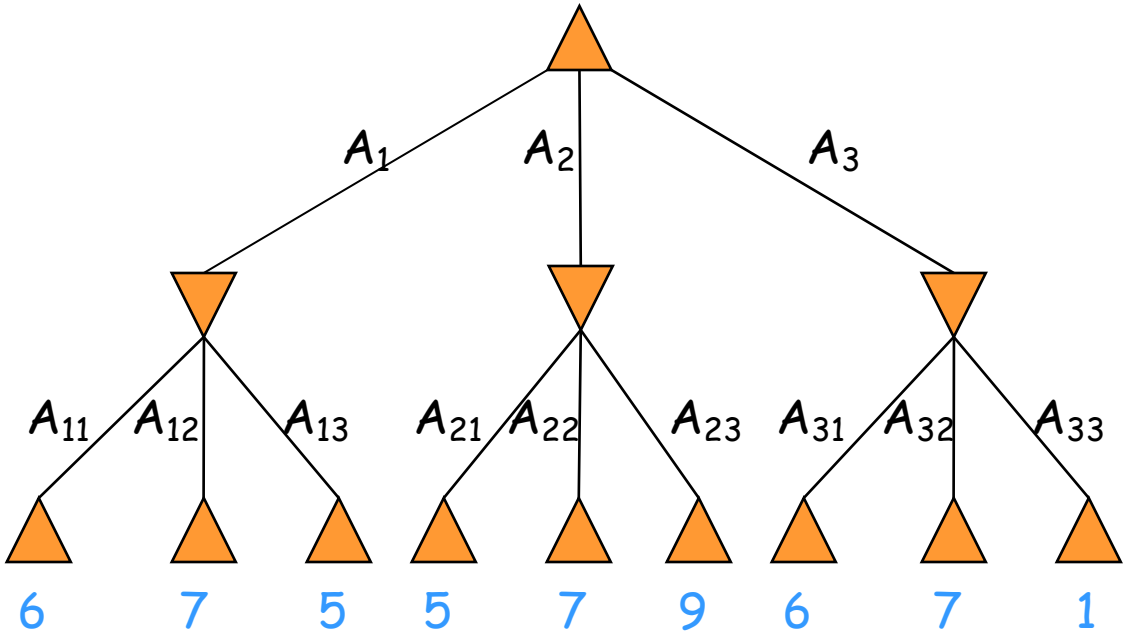
- 2006. El programa Deep Fritz, funcionando en un ordenador personal con procesador Intel Core 2 Duo consiguió derrotar también al actual campeón mundial Vladimir Krámnik por el marcador 4 – 2

Juegos

Campeones mundiales de ajedrez

Juegos

Juegos con elemento aleatorio



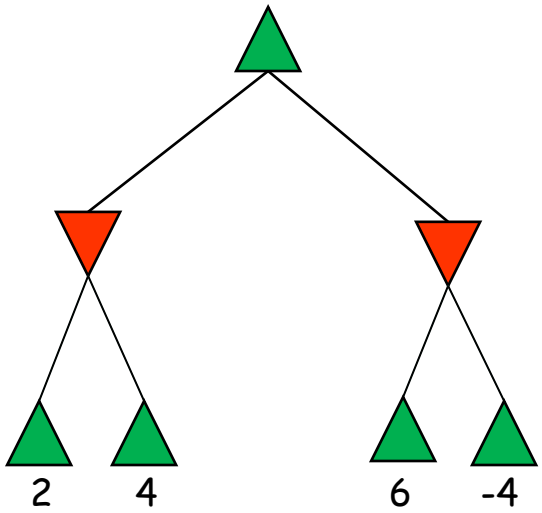
Juegos

Juegos con elemento aleatorio

- **Problema**: cómo aplicar minimax en juegos donde interviene el azar

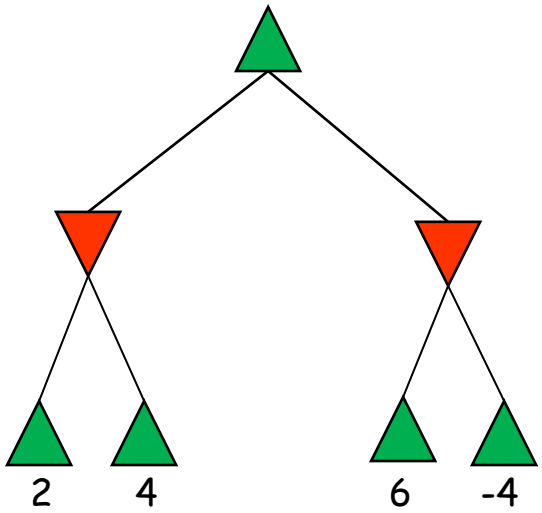
Juegos

Juegos con elemento aleatorio

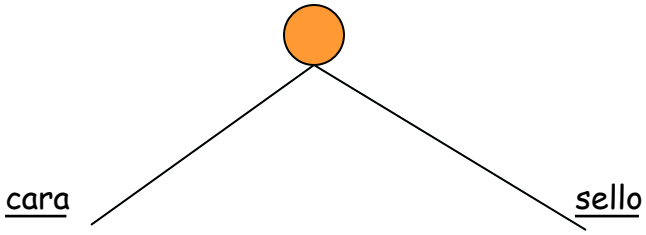


Juegos

Juegos con elemento aleatorio

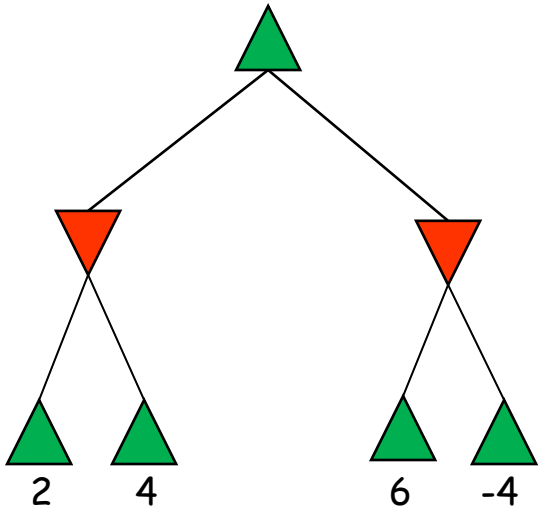


- Un árbol de juego donde influye el azar, debe incluir **nodos aleatorios**

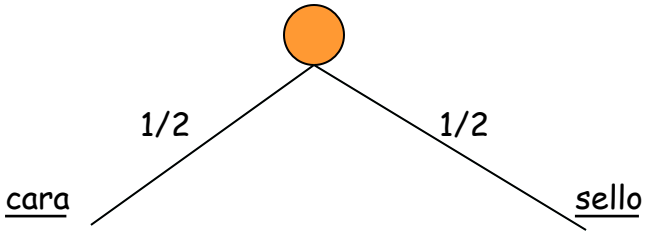


Juegos

Juegos con elemento aleatorio

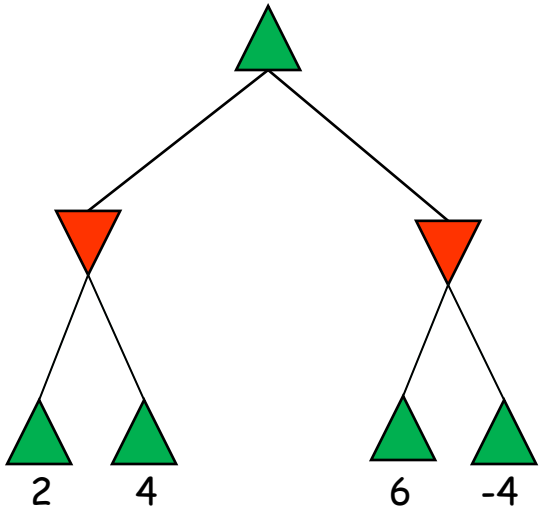


- Un árbol de juego donde influye el azar, debe incluir **nodos aleatorios**

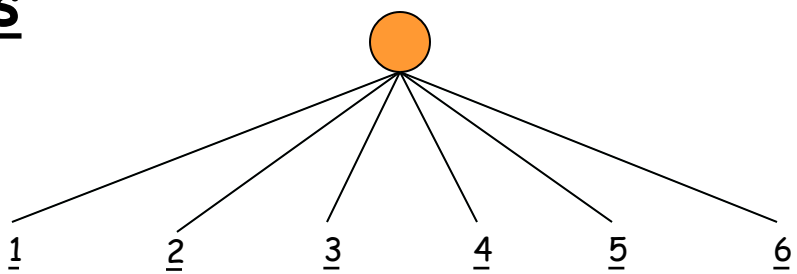


Juegos

Juegos con elemento aleatorio

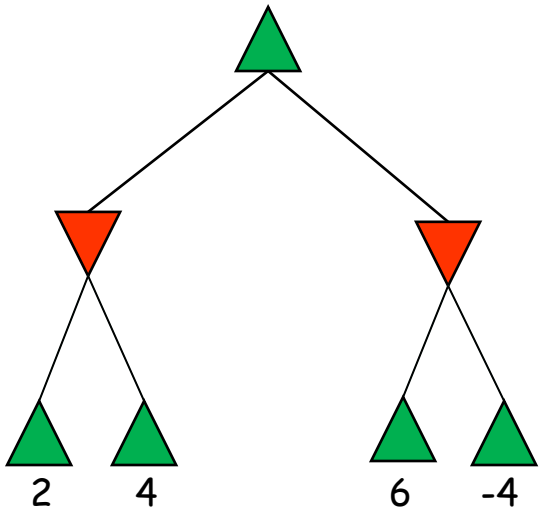


- Un árbol de juego donde influye el azar, debe incluir **nodos aleatorios**

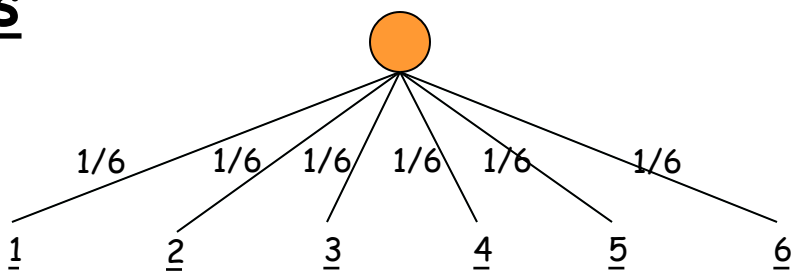


Juegos

Juegos con elemento aleatorio

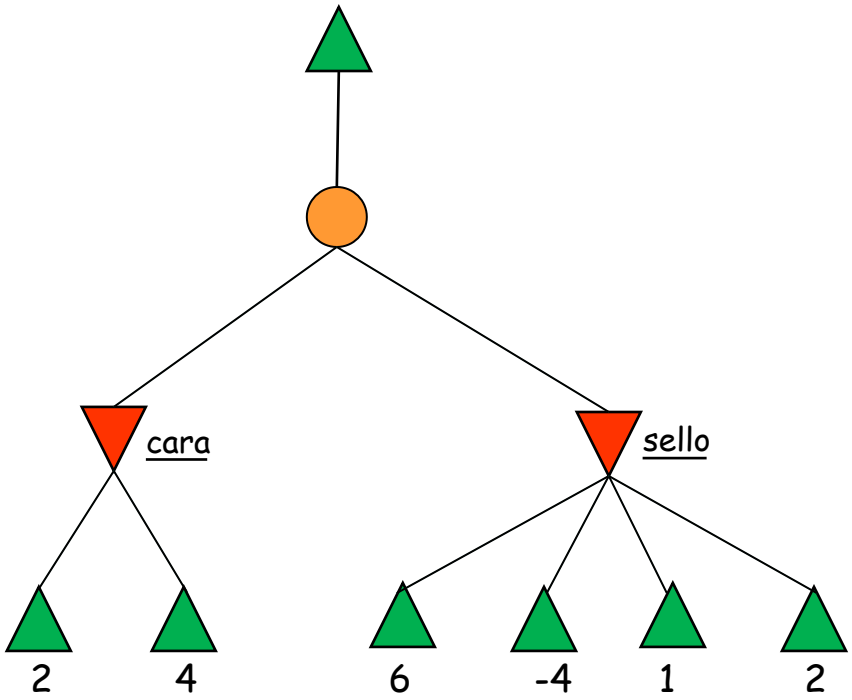


- Un árbol de juego donde influye el azar, debe incluir **nodos aleatorios**



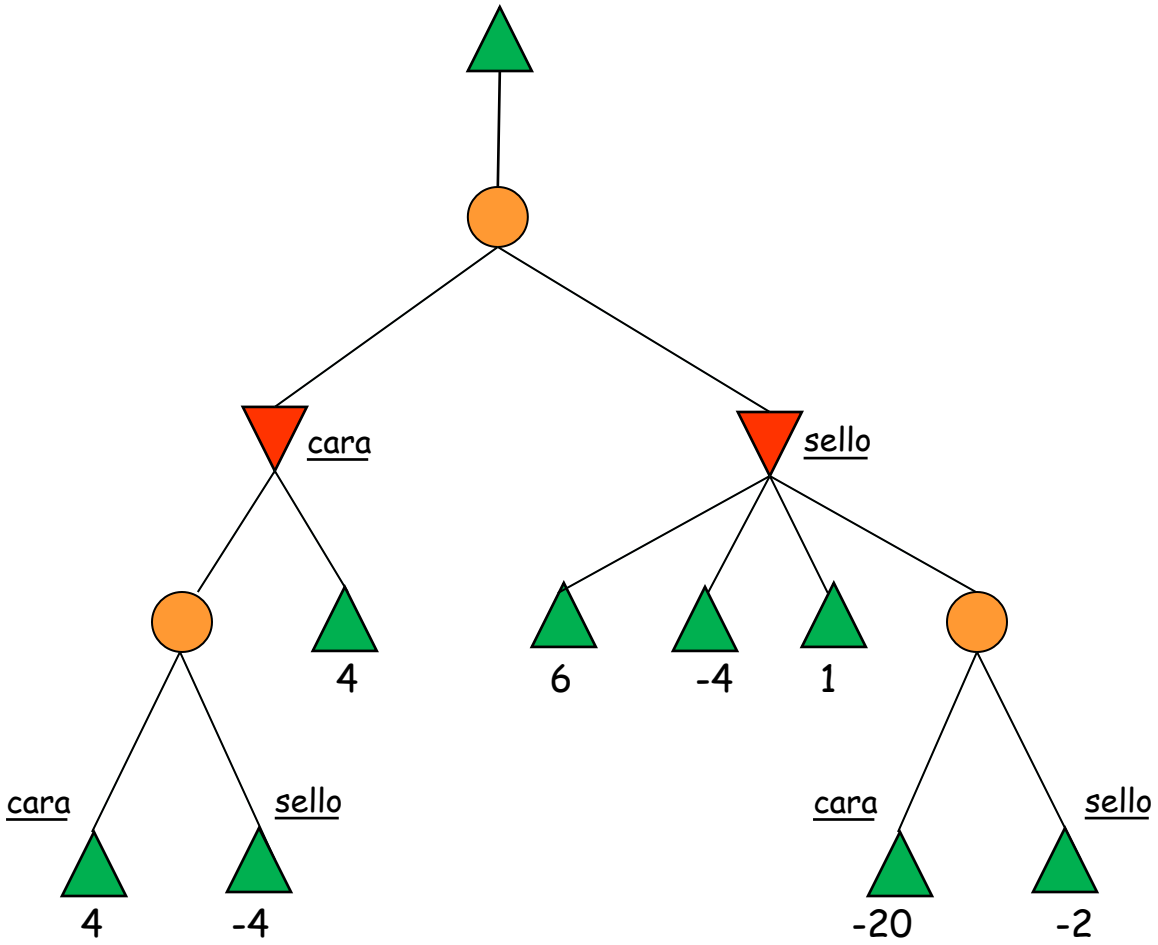
Juegos

Juegos con elemento aleatorio



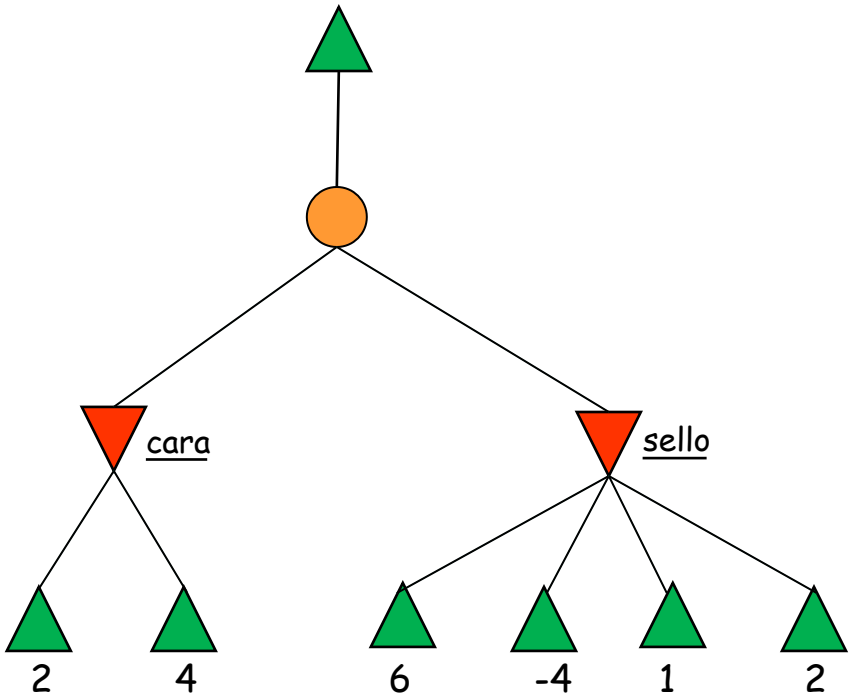
Juegos

Juegos con elemento aleatorio



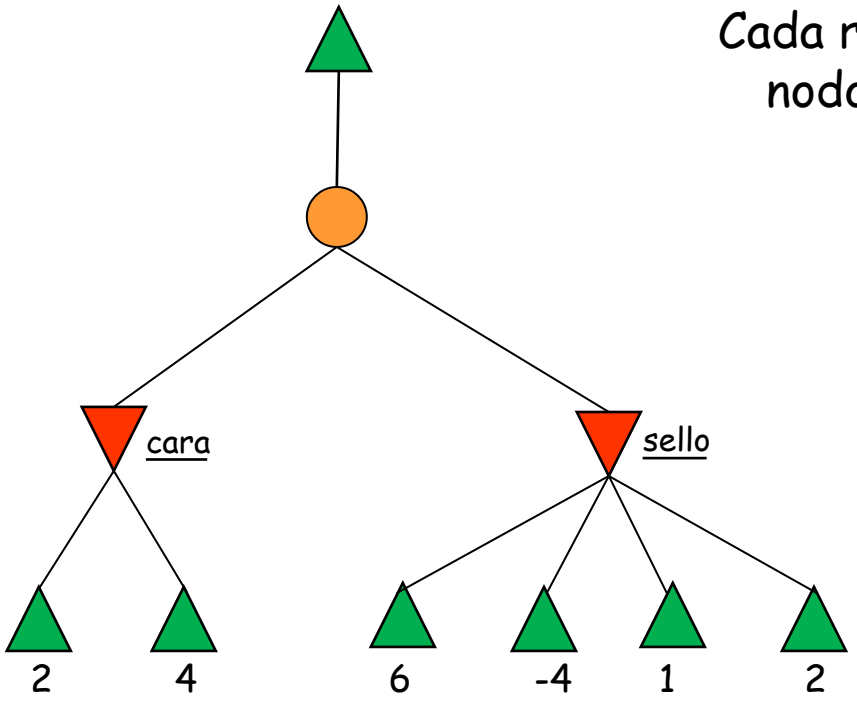
Juegos

Juegos con elemento aleatorio



Juegos

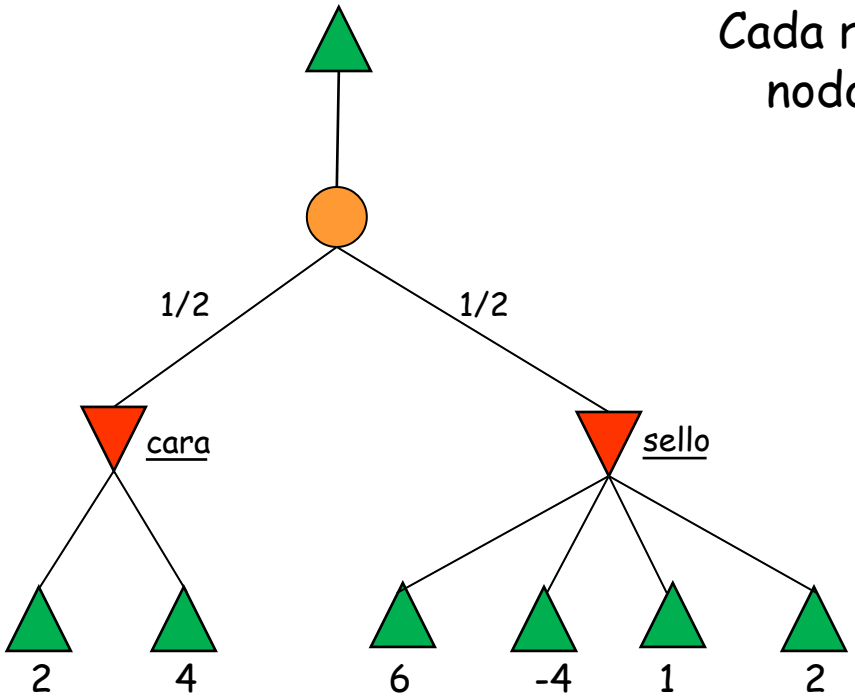
Juegos con elemento aleatorio



Cada rama que sale de un nodo aleatorio indica probabilidad

Juegos

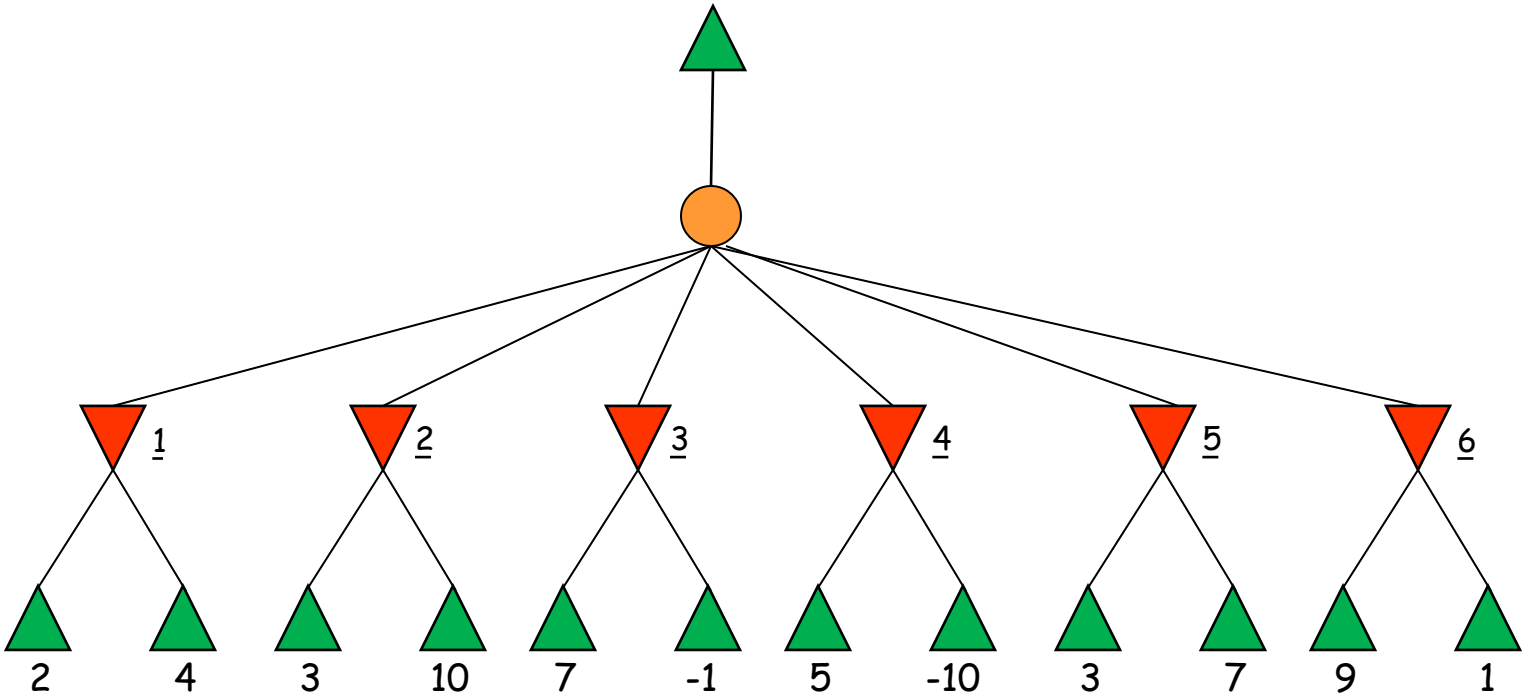
Juegos con elemento aleatorio



Cada rama que sale de un nodo aleatorio indica probabilidad

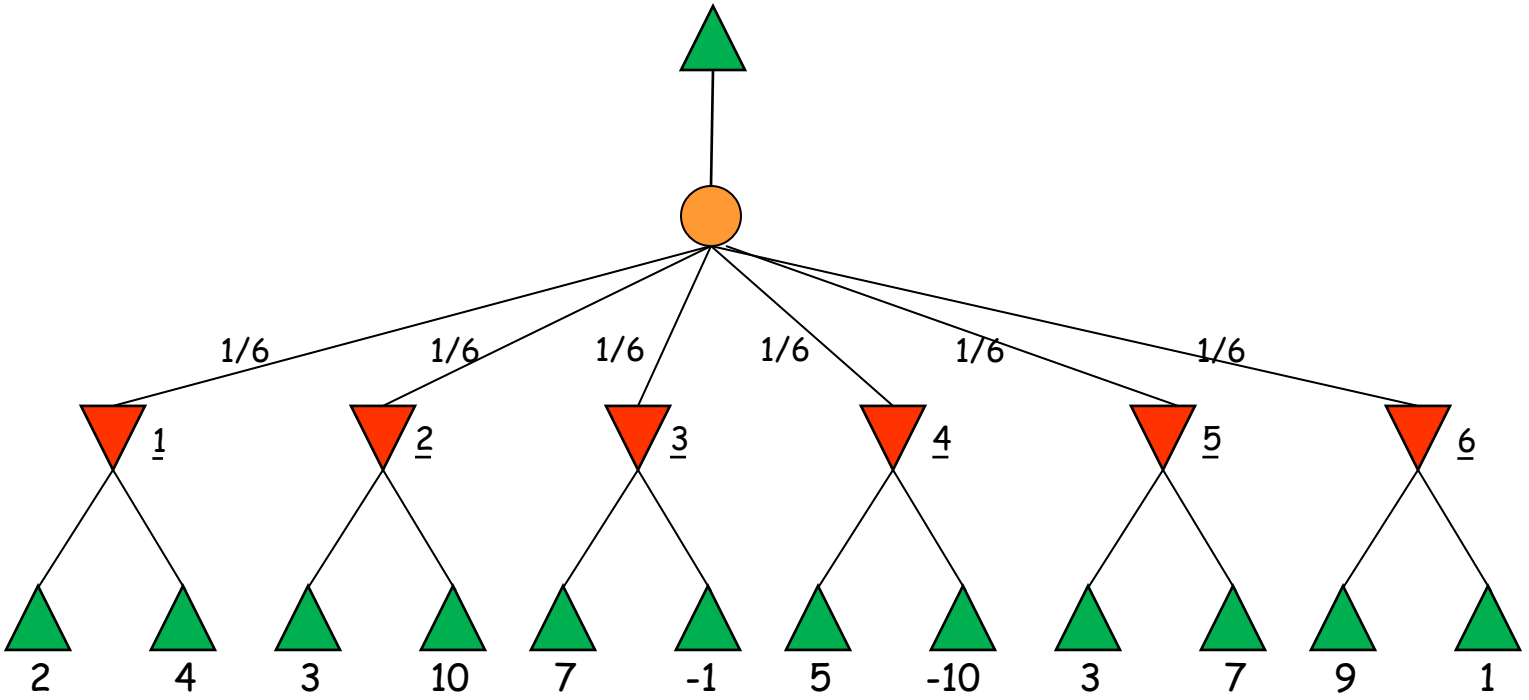
Juegos

Juegos con elemento aleatorio



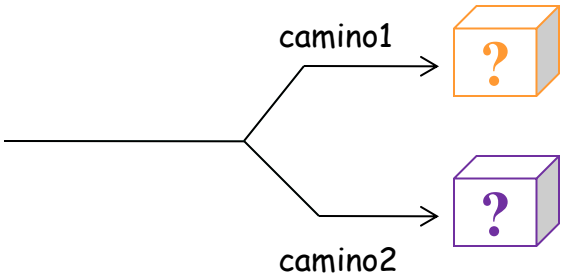
Juegos

Juegos con elemento aleatorio



Juegos con elemento aleatorio

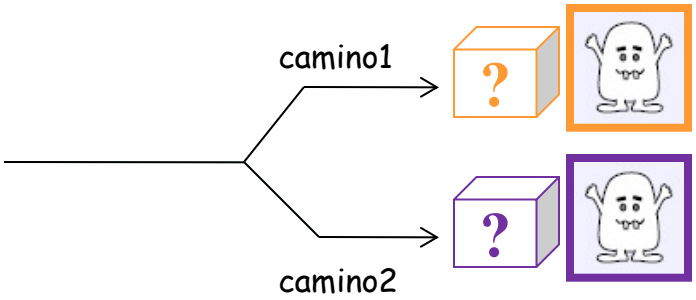
- Ejemplo de juego con lanzamiento de moneda



Juegos

Juegos con elemento aleatorio

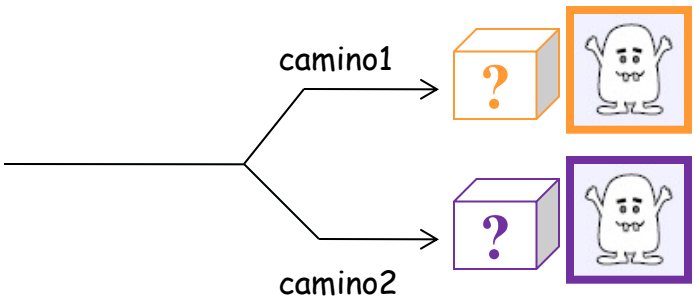
- Ejemplo de juego con lanzamiento de moneda



Juegos

Juegos con elemento aleatorio

- Ejemplo de juego con lanzamiento de moneda



Cara

Ataque1	2
Ataque2	4

Sello

Ataque3	7
Ataque4	4

Cara

Ataque1	6
Ataque2	0

Sello

Ataque3	5
Ataque4	-2

Juegos

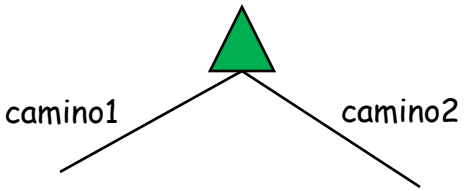
Juegos con elemento aleatorio

- Ejemplo de juego con lanzamiento de moneda

Juegos

Juegos con elemento aleatorio

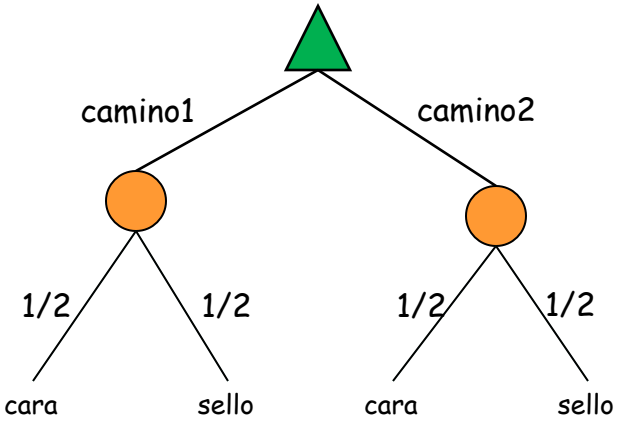
- Ejemplo de juego con lanzamiento de moneda



Juegos

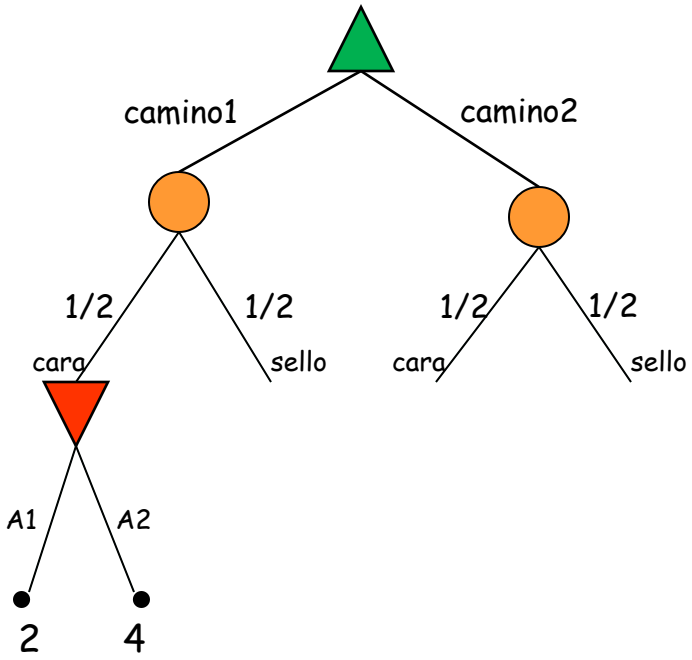
Juegos con elemento aleatorio

- Ejemplo de juego con lanzamiento de moneda



Juegos con elemento aleatorio

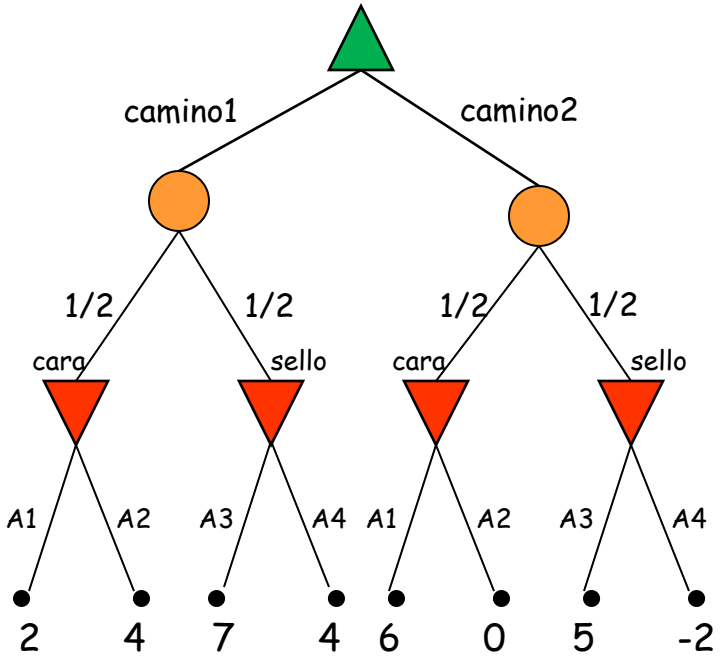
- Ejemplo de juego con lanzamiento de moneda



Juegos

Juegos con elemento aleatorio

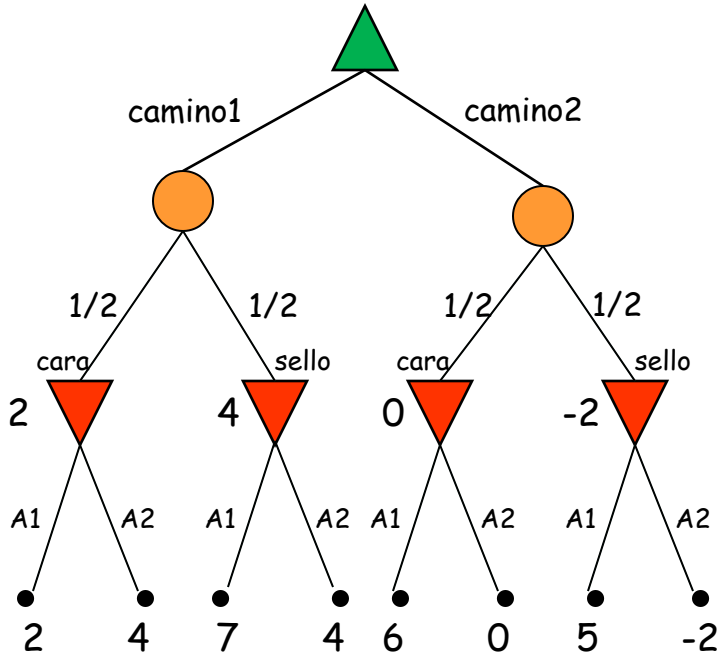
- Ejemplo de juego con lanzamiento de moneda



Juegos

Juegos con elemento aleatorio

- Ejemplo de juego con lanzamiento de moneda



Juegos

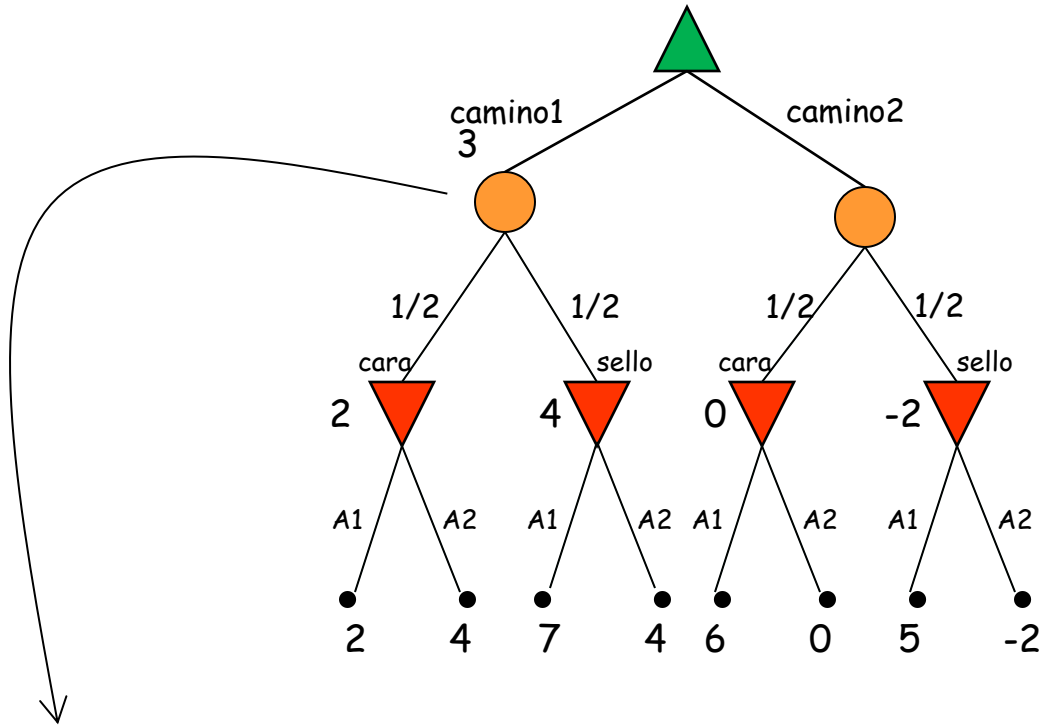
Juegos con elemento aleatorio

- Se calcula el **valor esperado** en **cada** nodo aleatorio

Juegos

Juegos con elemento aleatorio

- Ejemplo de juego con lanzamiento de moneda

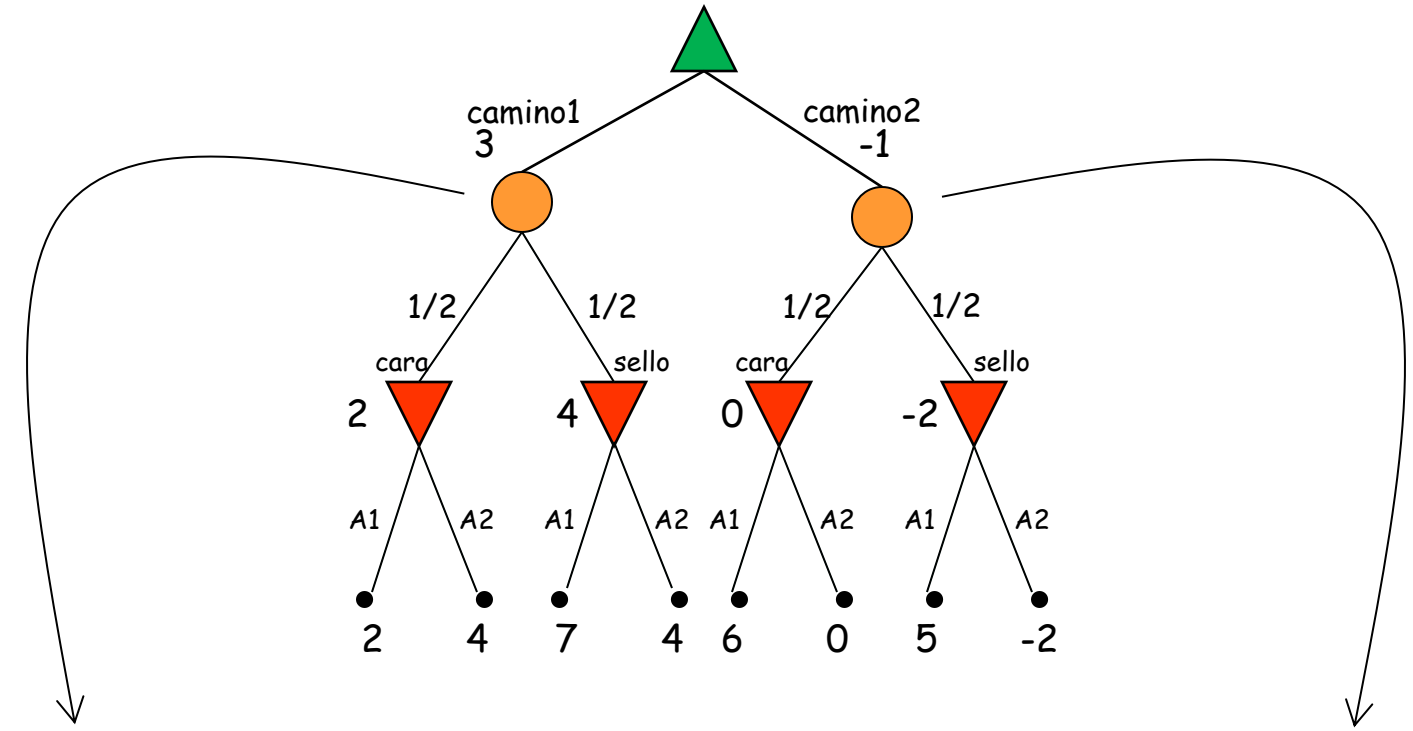


$(\frac{1}{2})*2 + (\frac{1}{2})*4 = 3$

Juegos

Juegos con elemento aleatorio

- Ejemplo de juego con lanzamiento de moneda



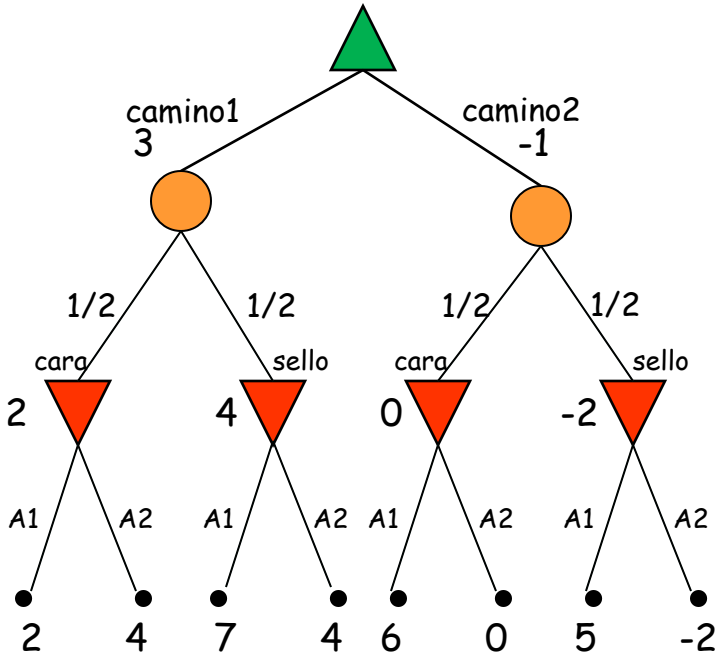
$(\frac{1}{2})*2 + (\frac{1}{2})*4 = 3$

$(\frac{1}{2})*0 + (\frac{1}{2})*(-2) = -1$

Juegos

Juegos con elemento aleatorio

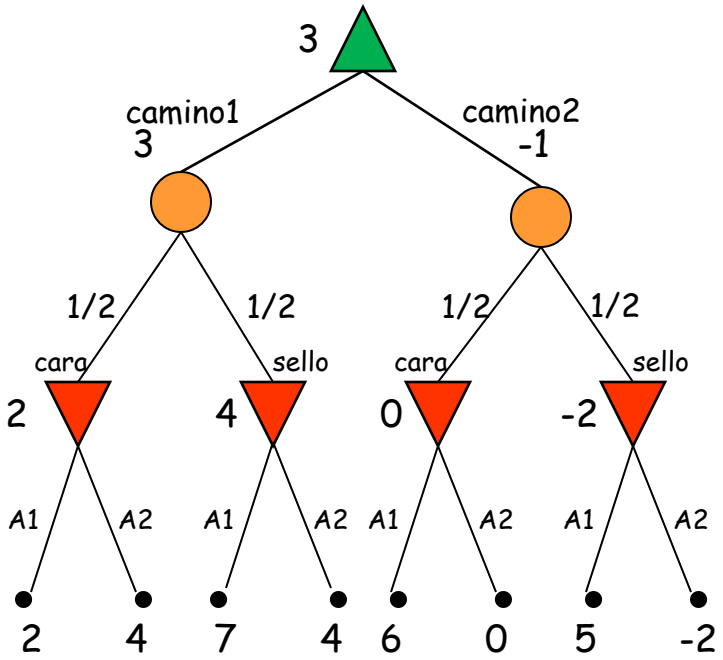
- Ejemplo de juego con lanzamiento de moneda



Juegos

Juegos con elemento aleatorio

- Ejemplo de juego con lanzamiento de moneda

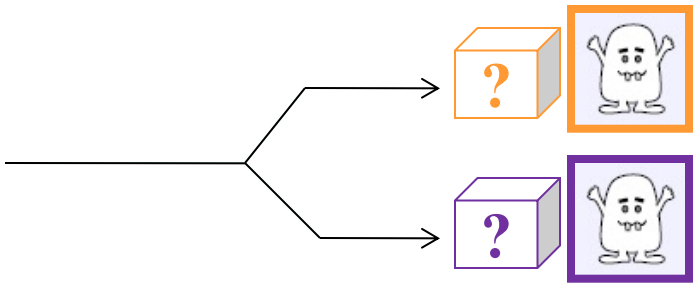


La decision minimax es **tomar el camino1**

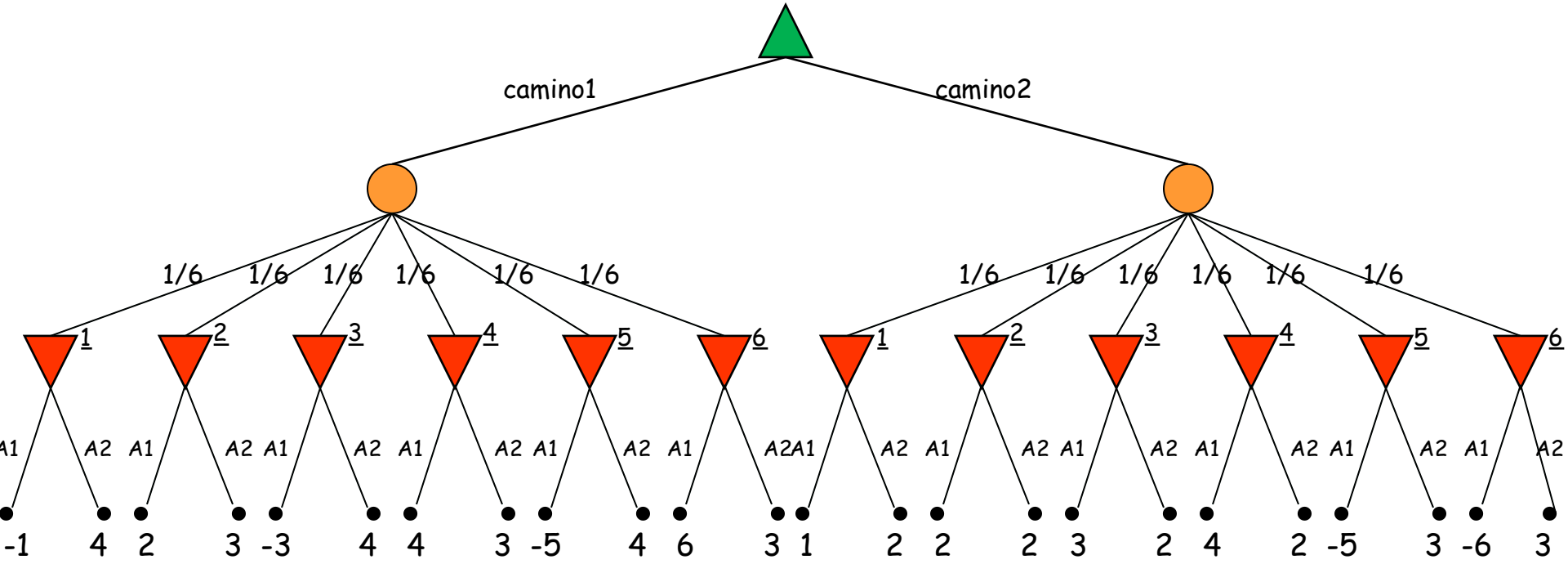
Juegos

Aplique **minimax** considerando un dado en lugar de una moneda:

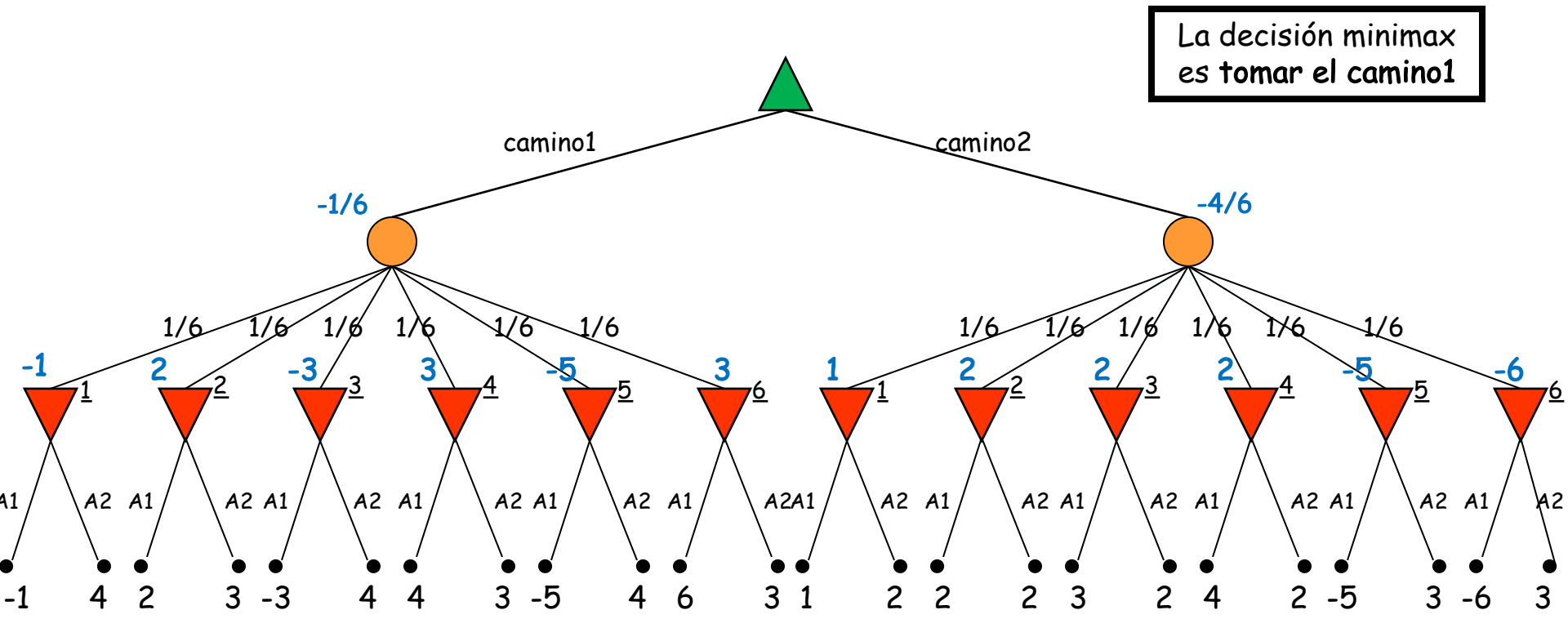
- En el **camino 1**, si usted saca un número par el fantasma puede optar entre dejarlo con una utilidad que es igual al número obtenido o dejarlo en 3, y si saca un número impar lo puede dejar en una utilidad que es igual a menos el número obtenido o 4
- En el **camino 2**, si usted saca un número menor o igual que 4, el fantasma puede optar entre dejarlo con una utilidad que es igual al número obtenido o dejarlo en 2, y en los otros casos (5,6) lo puede dejar en menos el número obtenido o en 3



Juegos


camino1
camino2
1/6
1
2
3
4
5
6
A1
A2
-1
4
2
3
-3
4
4
3
-5
4
6
3
1
2
2
2
3
2
4
2
-5
3
-6
3

Juegos


La decisión minimax es tomar el camino1
camino1
camino2
-1/6
-4/6
1/6
-1
2
-3
3
-5
3
1
2
2
2
-5
-6
A1
A2
-1 4 2 3 -3 4 4 3 -5 4 6 3 1 2 2 2 3 2 4 2 -5 3 -6 3

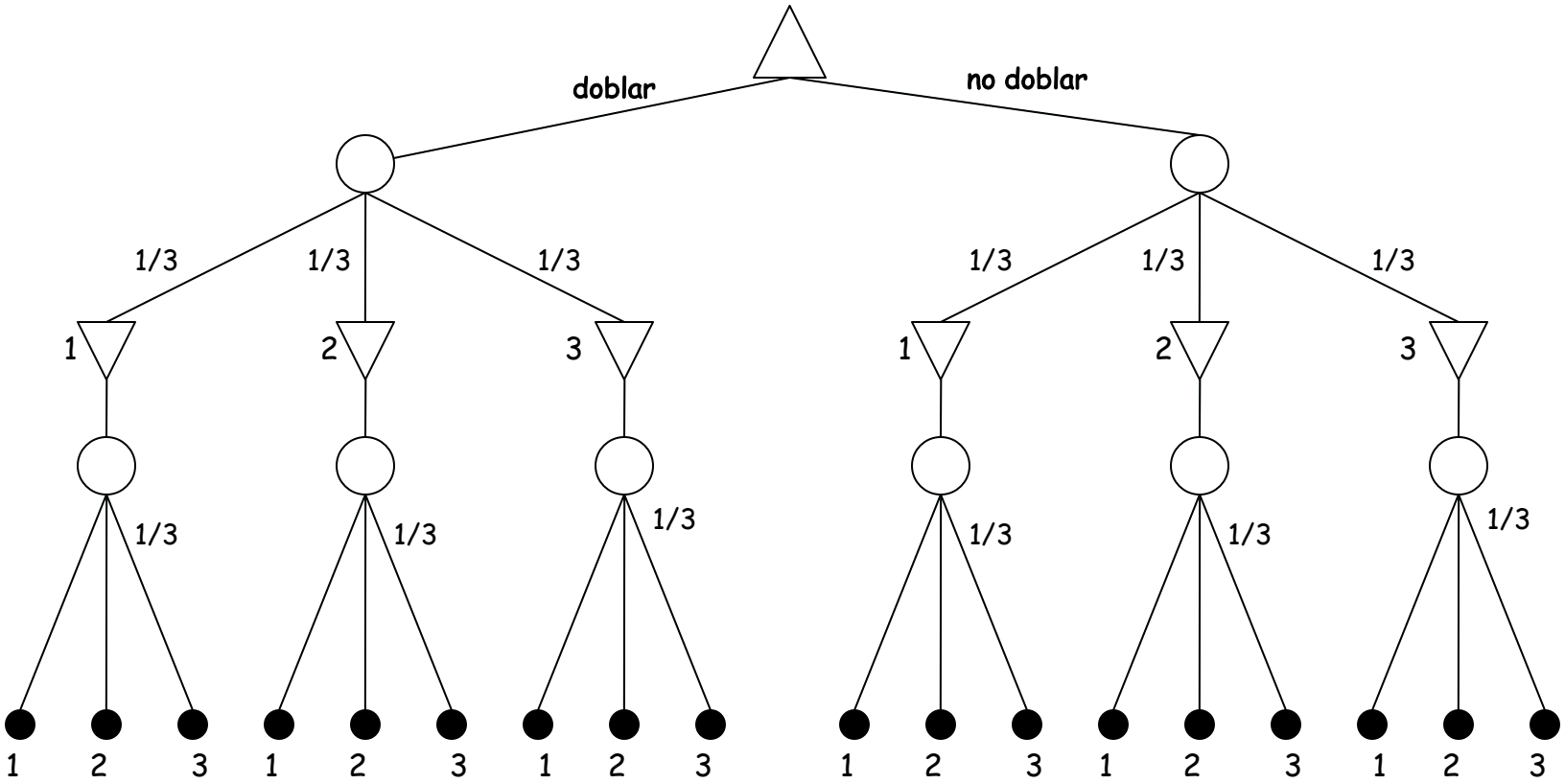
Juegos

Aplique **minimax** en el siguiente juego:

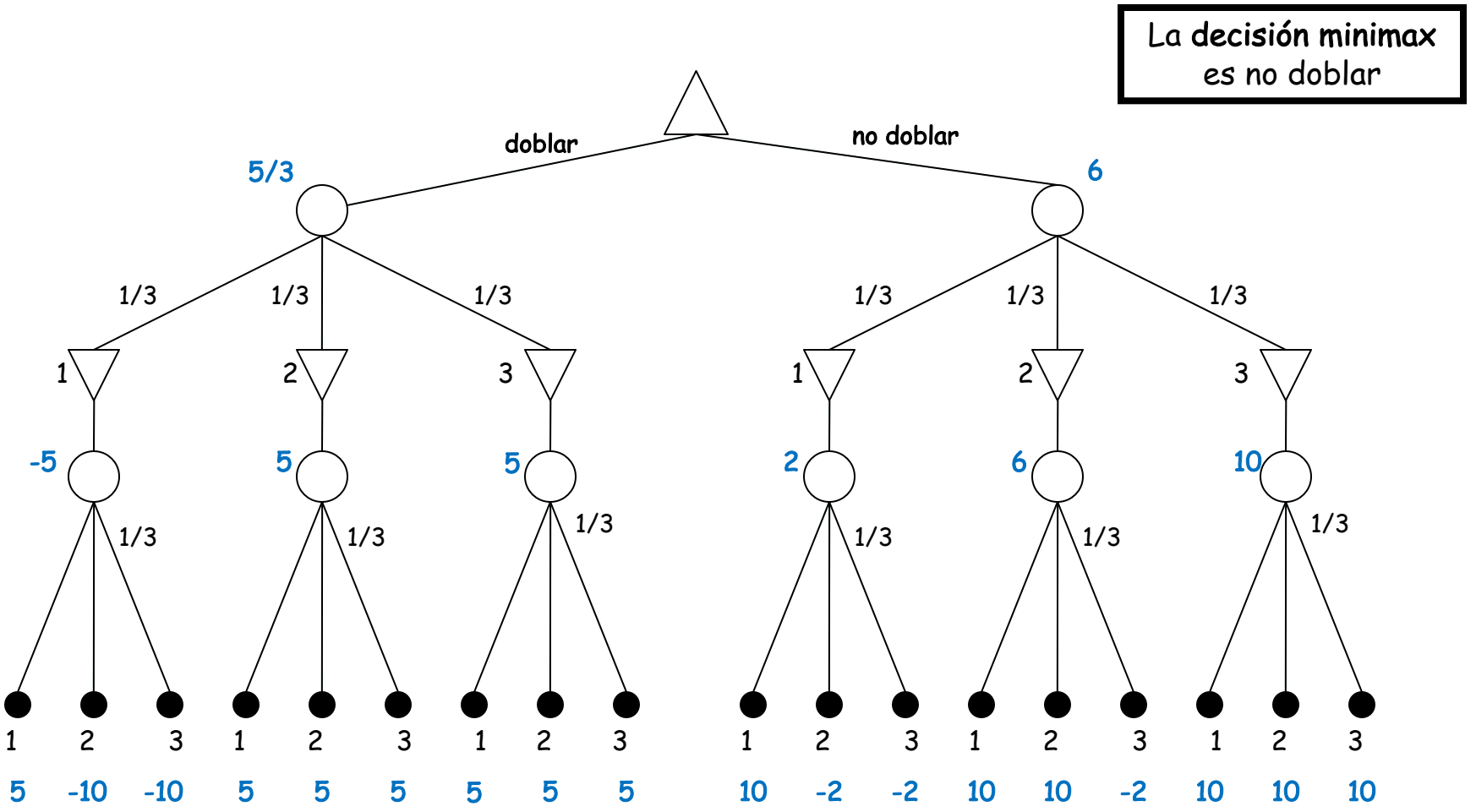
- El juego consiste en que dos personas lanzan un dado* y gana quien obtiene un mayor puntaje. El jugador A puede inicialmente decidir si va a jugar doblando el valor obtenido o no
- Si A dobla su puntaje y obtiene un valor mayor (doblado) que el de B, su utilidad será 5, si obtiene menos o lo mismo su utilidad será -10
- Si A no dobla su puntaje y saca más o lo mismo que B, obtiene utilidad 10, si saca menos obtiene utilidad -2
- B no puede decidir si doble su puntaje
- Primero lanza el dado A y luego B

** Considere el dado de solo números 1, 2 y 3*

Juegos


doblar
no doblar
1/3
1/3
1/3
1/3
1/3
1/3
1
2
3
1
2
3
1/3
1/3
1/3
1/3
1/3
1/3
1
2
3
1
2
3
1
2
3
1
2
3
1
2
3
1
2
3

Juegos


La decisión minimax es no doblar
doblar
no doblar
5/3
6
1/3
1/3
1/3
1/3
1/3
1/3
1
2
3
1
2
3
-5
5
5
2
6
10
1/3
1/3
1/3
1/3
1/3
1/3
1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
5 -10 -10 5 5 5 5 5 5 10 -2 -2 10 10 -2 10 10 10

Juegos

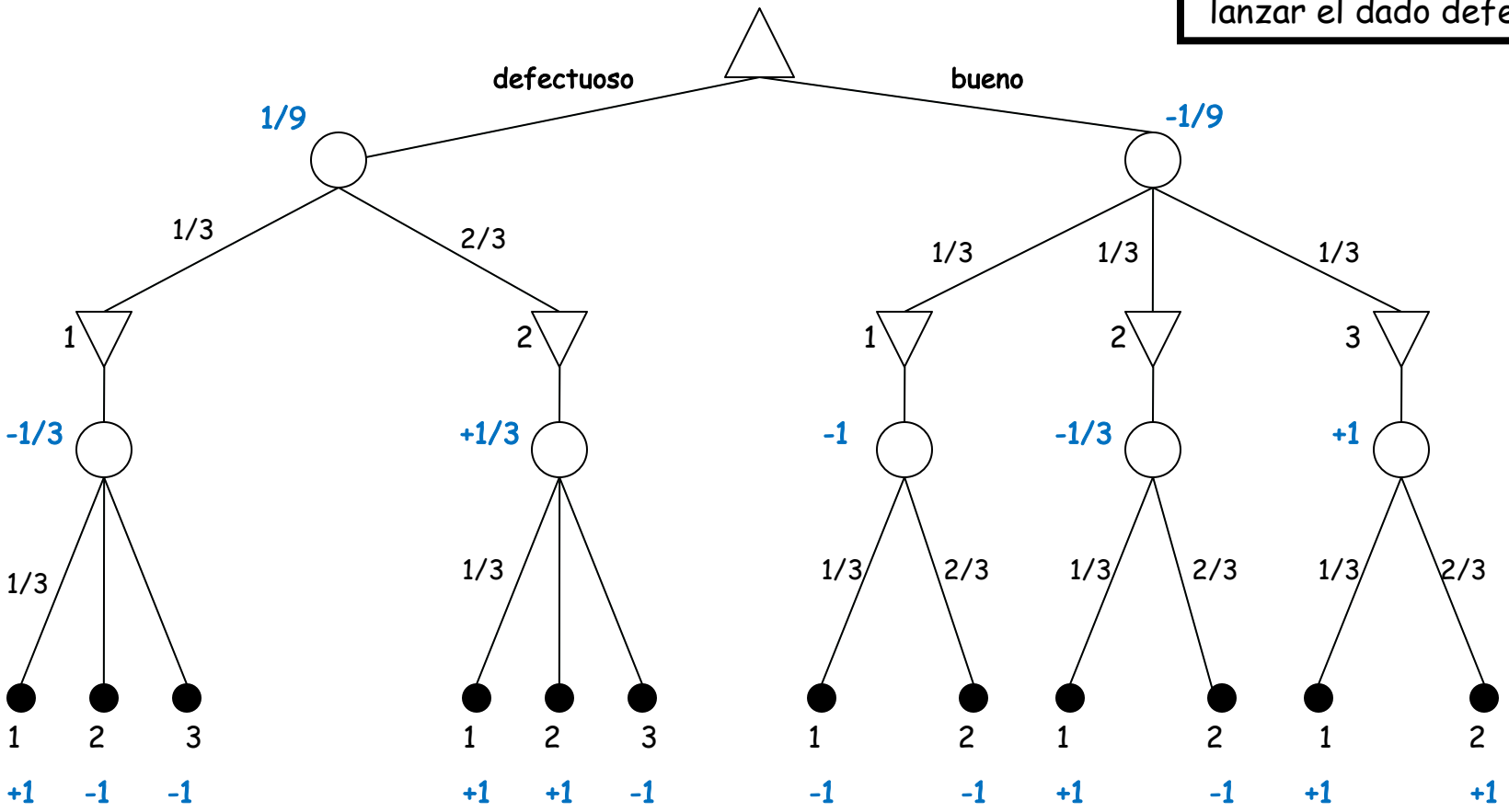
Aplique **minimax** en el siguiente juego:

- En una mesa se tienen dos dados*, uno está defectuoso por lo que al lanzarlo solo se puede obtener 1 ó 2, con probabilidades 1/3 y 2/3 respetivamente. El otro dado funciona normalmente
- El juego consiste en que cada jugador lanza un dado una sola vez y gana quien obtenga el mayor valor
- En caso de empate gana el que lanzó el dado defectuoso
- La decisión de con qué dado se lanza solamente la puede tomar el jugador A quien inicia el juego, es decir, B tendrá que lanzar el dado que no escogió A
- La utilidad cuando gana A es +1 y cuando pierde -1

** Considere el dado de solo números 1, 2 y 3*

Juegos

La **decisión minimax** es lanzar el dado defectuoso



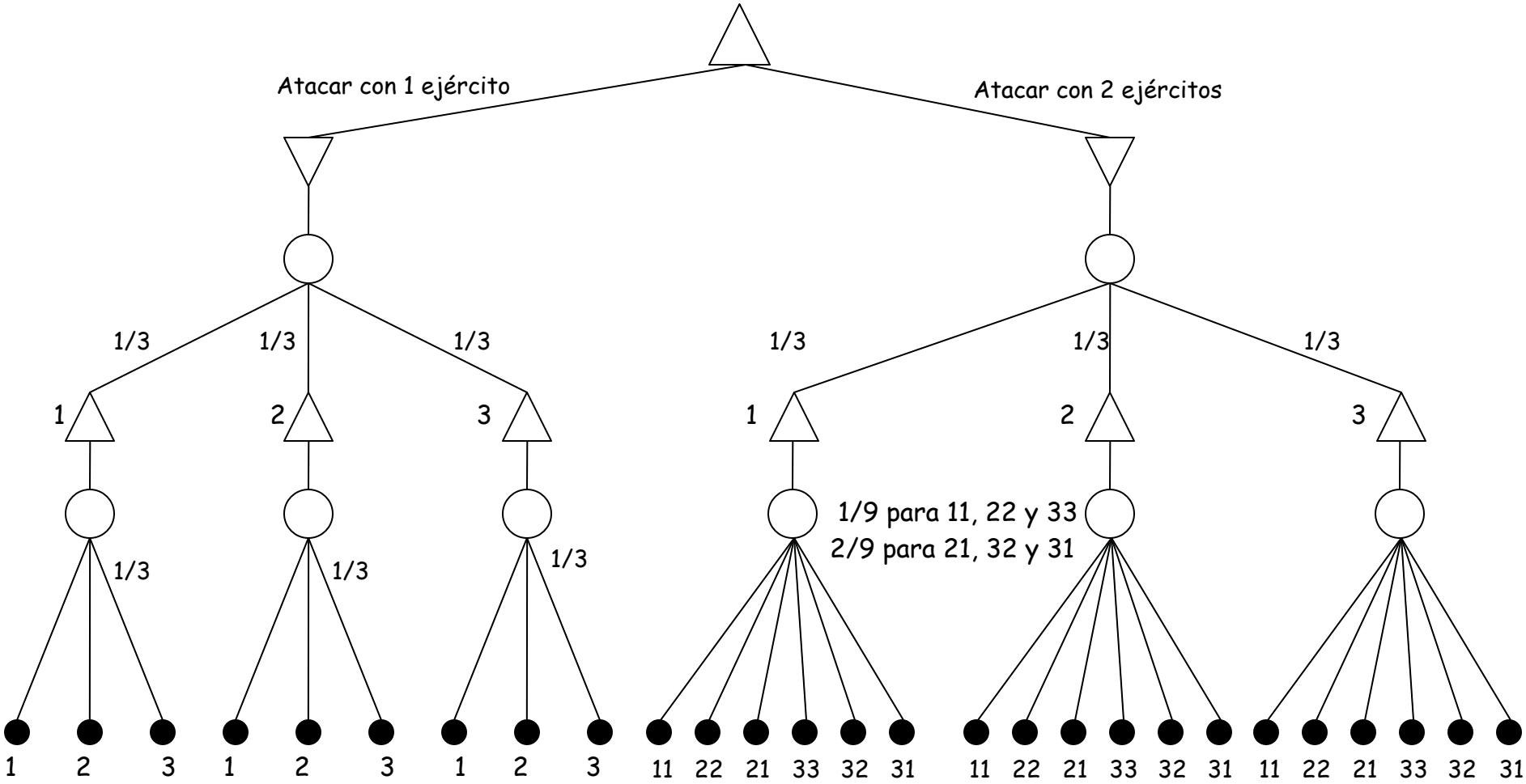
Juegos

Juego de los ejércitos

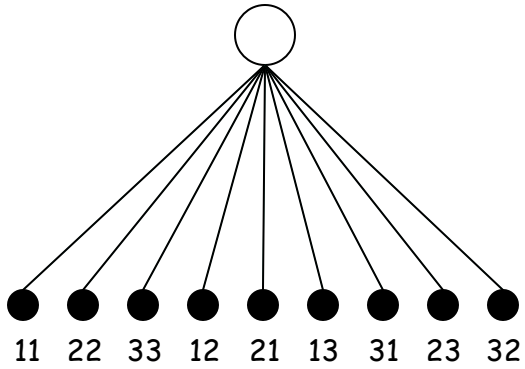
- Un jugador que vaya a atacar puede hacerlo con 1 o 2 ejércitos
- El éxito de un ataque se representa por el valor obtenido en un dado*
- Lanza primero el defensor
- El atacante tiene que sacar un número mayor que el defensor para ganar, si se ataca con 2 ejércitos el atacante podrá lanzar 2 veces el dado

Suponga que el dado solo tiene los números 1, 2 y 3

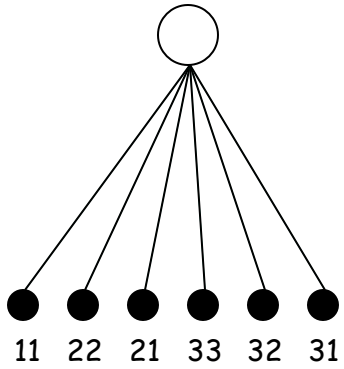
Juegos


Atacar con 1 ejército
Atacar con 2 ejércitos
1/3
1/3
1/3
1/3
1/3
1/3
1
2
3
1
2
3
1/3
1/3
1/3
1/9 para 11, 22 y 33
2/9 para 21, 32 y 31
1
2
3
1
2
3
1
2
3
11
22
21
33
32
31
11
22
21
33
32
31
11
22
21
33
32
31

Juegos



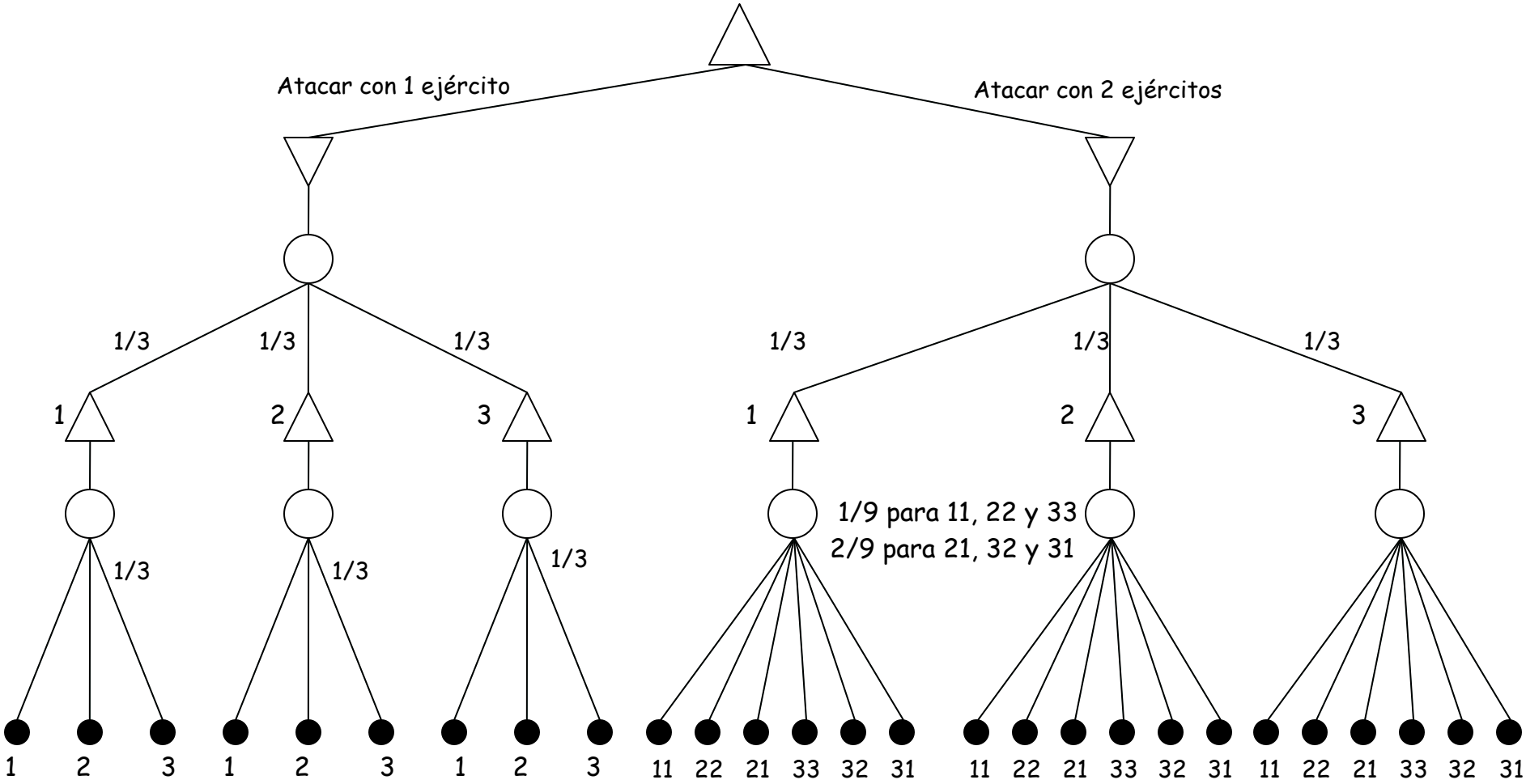
1/9 para cada posibilidad



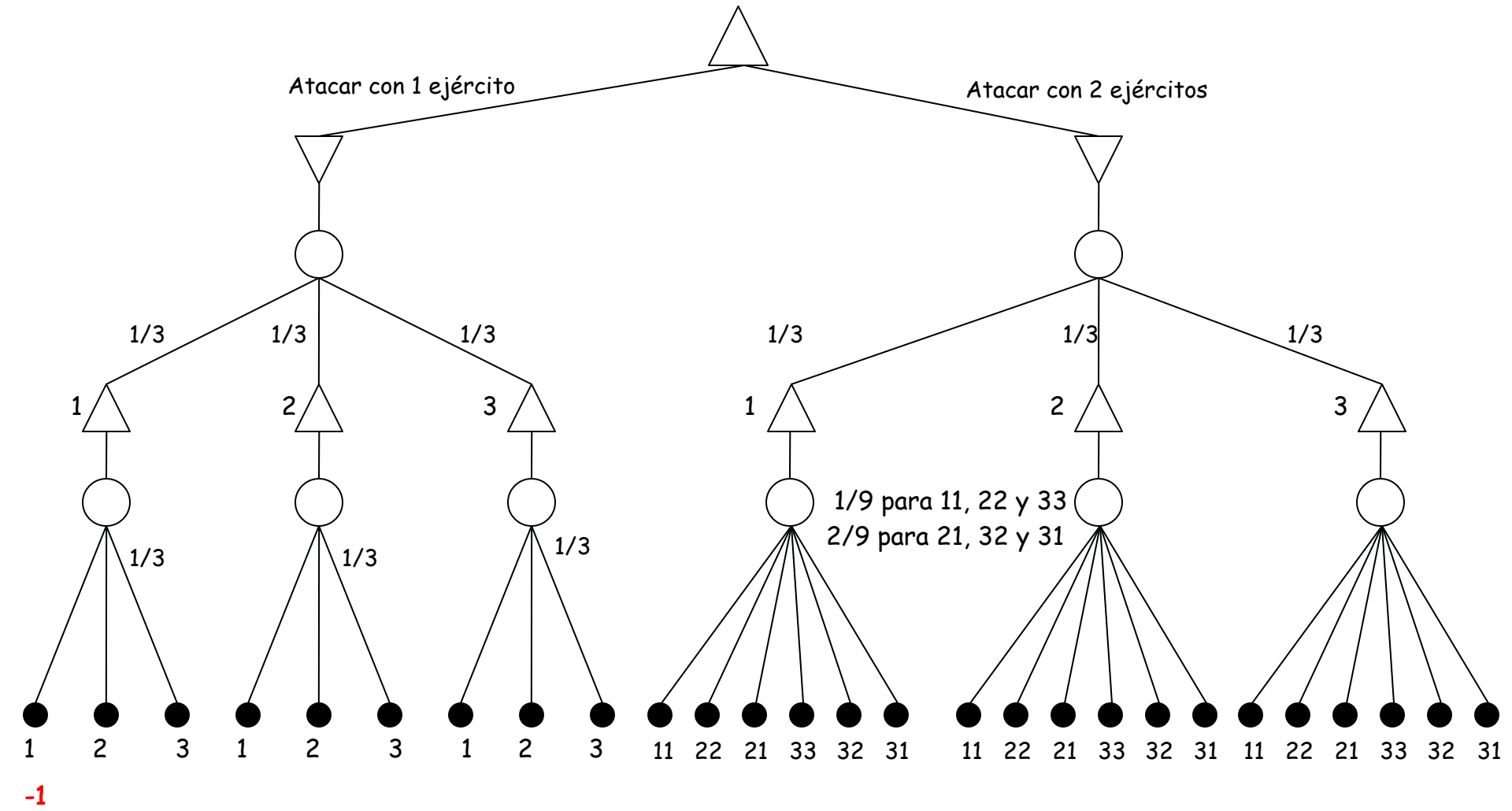
1/9 para 11, 22 y 33

2/9 para 21, 32 y 31

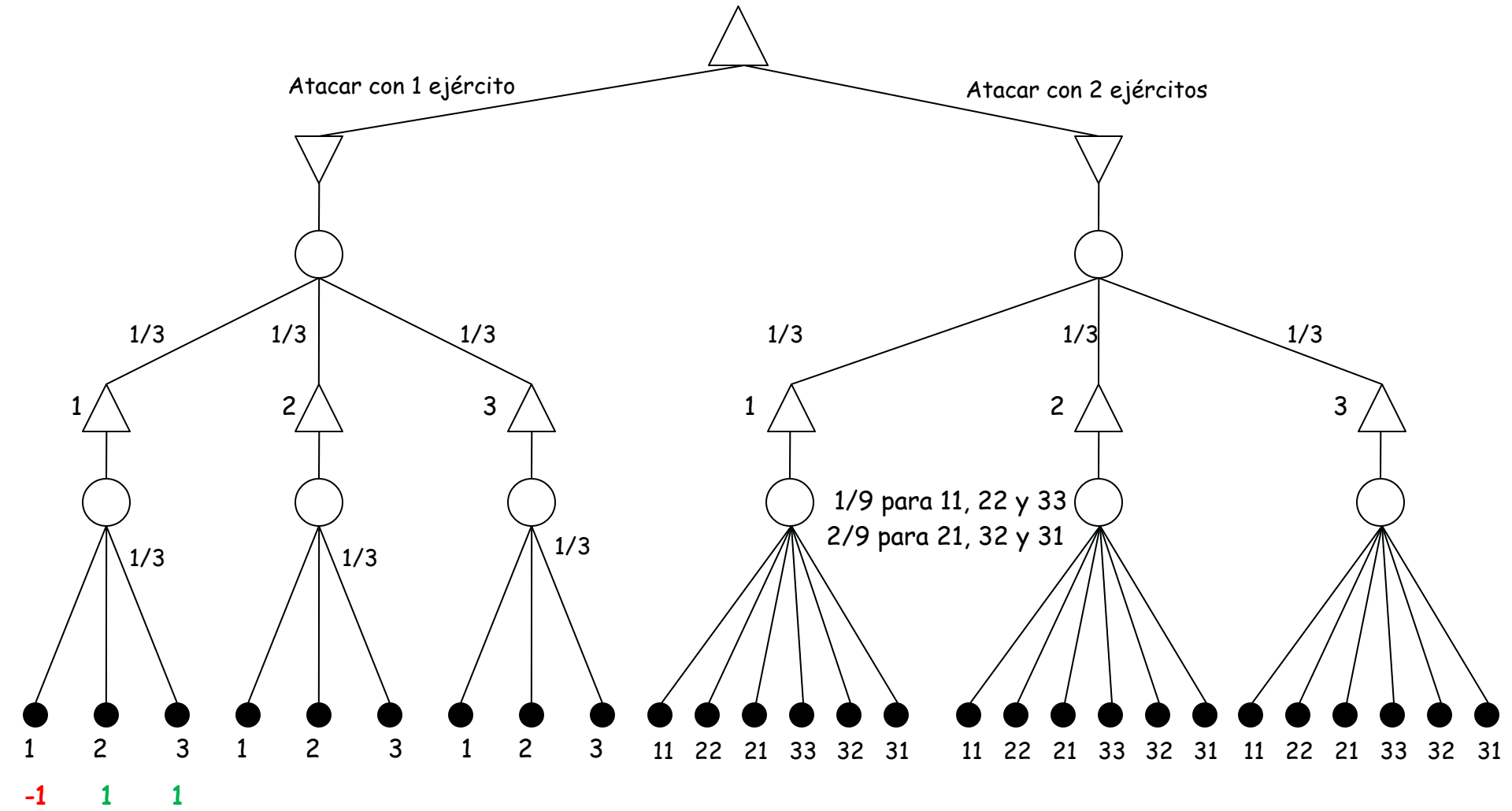
Juegos


Atacar con 1 ejército
Atacar con 2 ejércitos
1/3
1/3
1/3
1/3
1/3
1/3
1
2
3
1
2
3
1/3
1/3
1/3
1/9 para 11, 22 y 33
2/9 para 21, 32 y 31
1
2
3
1
2
3
1
2
3
11
22
21
33
32
31
11
22
21
33
32
31
11
22
21
33
32
31

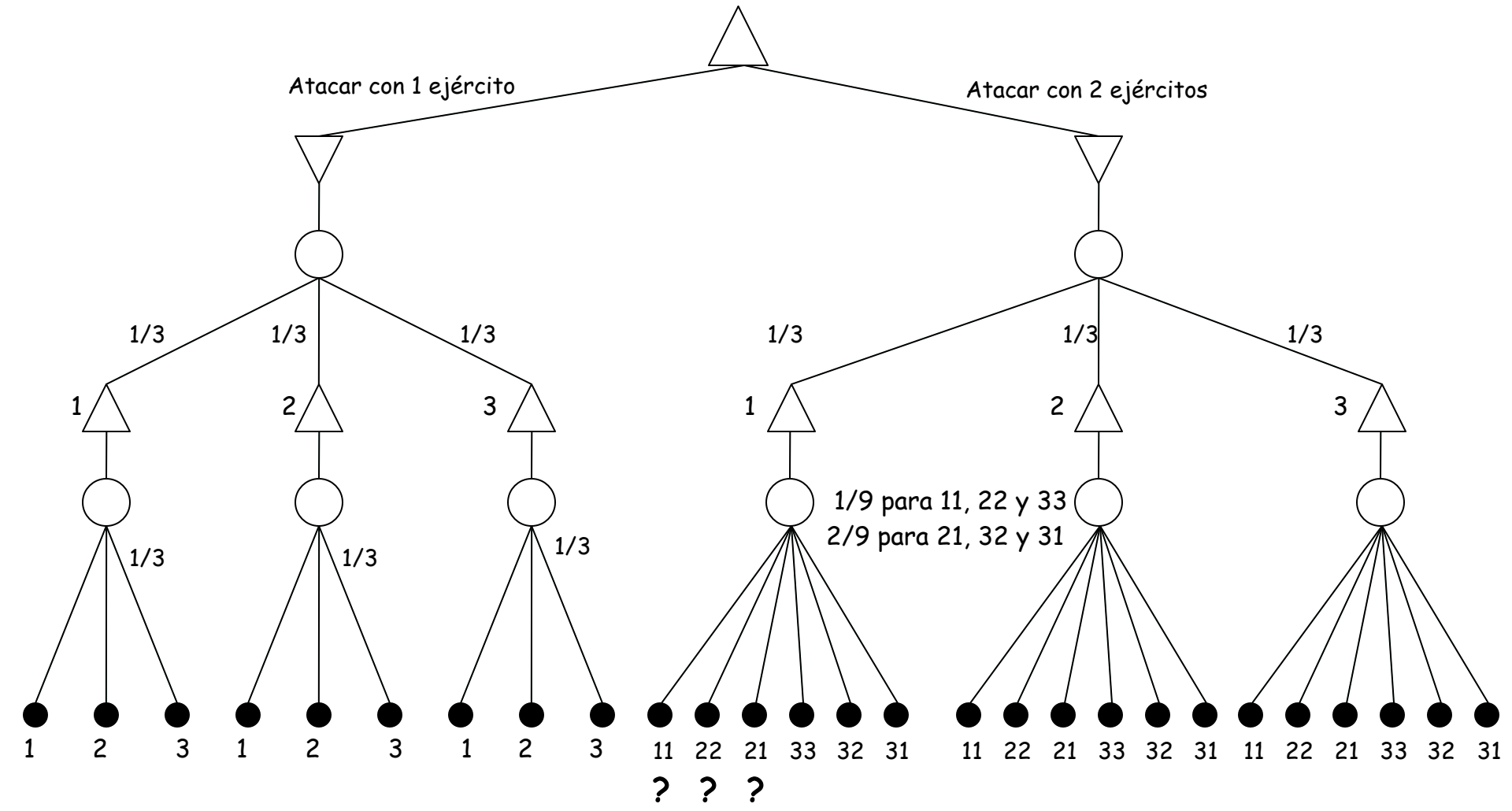
Juegos


Atacar con 1 ejército
Atacar con 2 ejércitos
1/3
1/3
1/3
1/3
1/3
1/3
1
2
3
1
2
3
1/3
1/3
1/3
1/9 para 11, 22 y 33
2/9 para 21, 32 y 31
1
2
3
1
2
3
1
2
3
11
22
21
33
32
31
11
22
21
33
32
31
11
22
21
33
32
31
-1

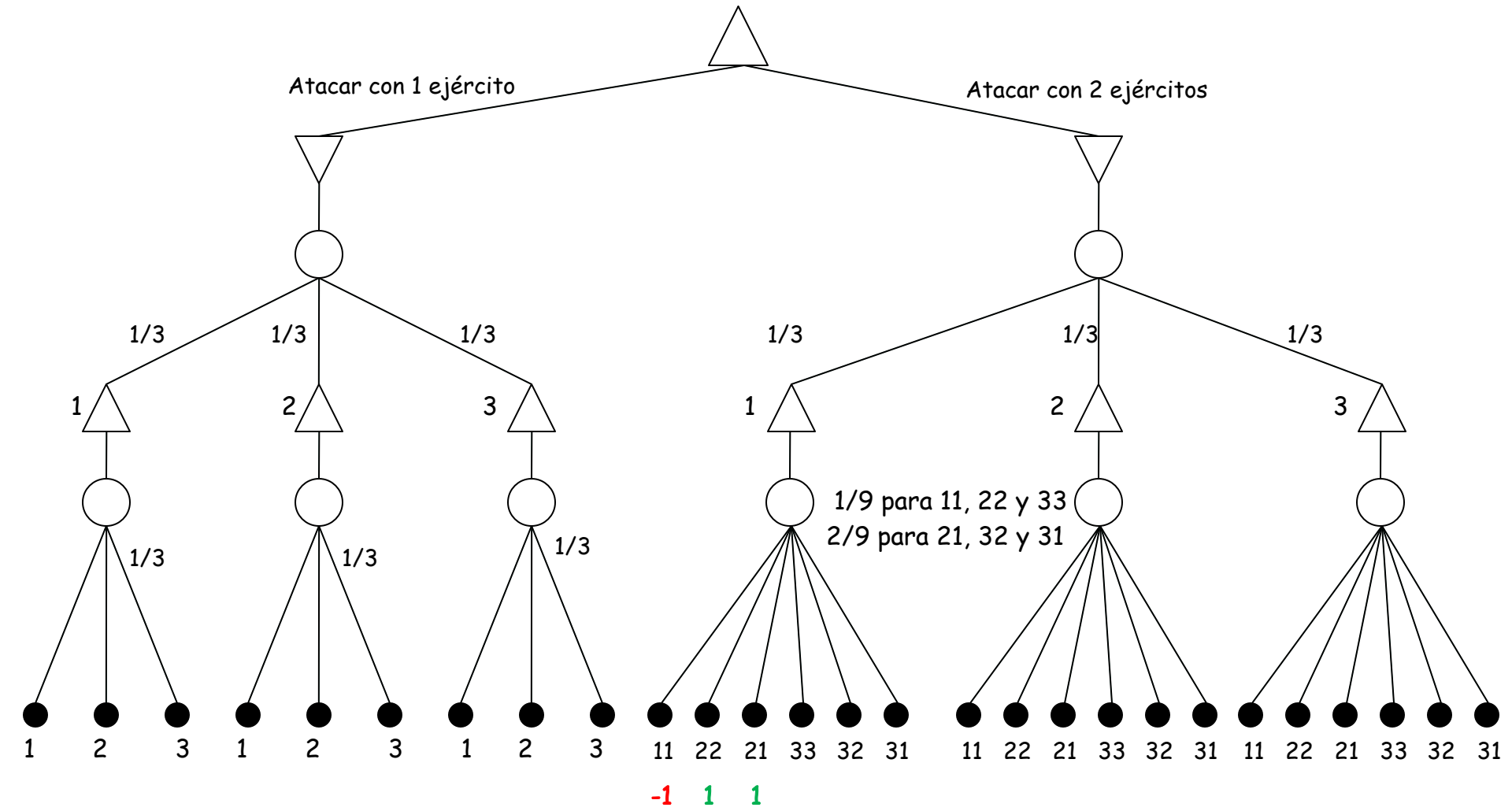
Juegos


Atacar con 1 ejército
Atacar con 2 ejércitos
1/3
1/3
1/3
1/3
1/3
1/3
1
2
3
1
2
3
1/3
1/3
1/3
1/9 para 11, 22 y 33
2/9 para 21, 32 y 31
1
2
3
11
22
21
33
32
31
11
22
21
33
32
31
11
22
21
33
32
31
-1
1
1

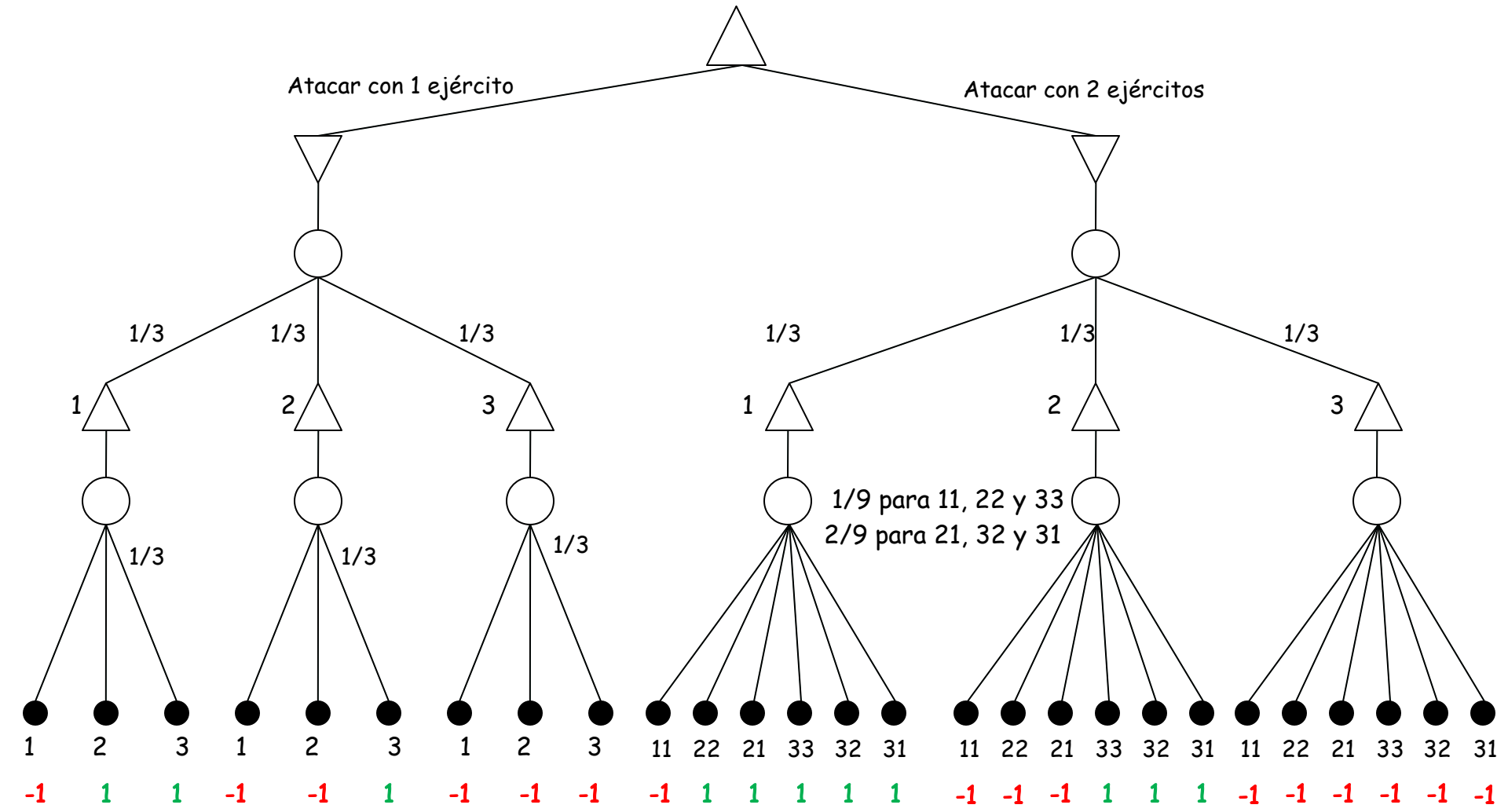
Juegos


Atacar con 1 ejército
Atacar con 2 ejércitos
1/3
1/3
1/3
1/3
1/3
1/3
1
2
3
1
2
3
1/3
1/3
1/3
1/9 para 11, 22 y 33
2/9 para 21, 32 y 31
1 2 3 1 2 3 1 2 3 11 22 21 33 32 31 11 22 21 33 32 31 11 22 21 33 32 31
? ? ?

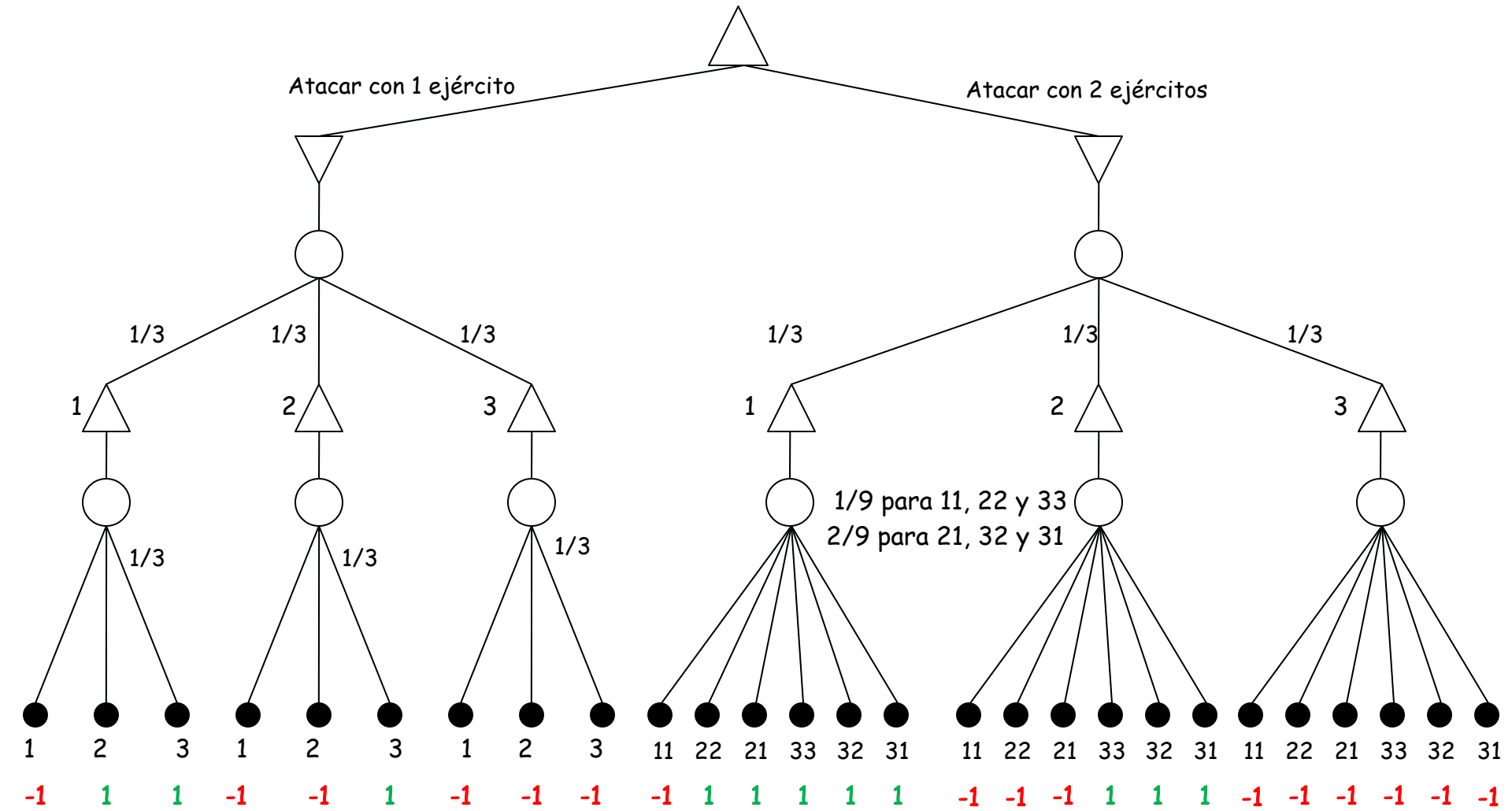
Juegos


Atacar con 1 ejército
Atacar con 2 ejércitos
1/3
1/3
1/3
1/3
1/3
1/3
1
2
3
1
2
3
1/3
1/3
1/3
1/9 para 11, 22 y 33
2/9 para 21, 32 y 31
1 2 3 1 2 3 1 2 3 11 22 21 33 32 31 11 22 21 33 32 31 11 22 21 33 32 31
-1 1 1

Juegos

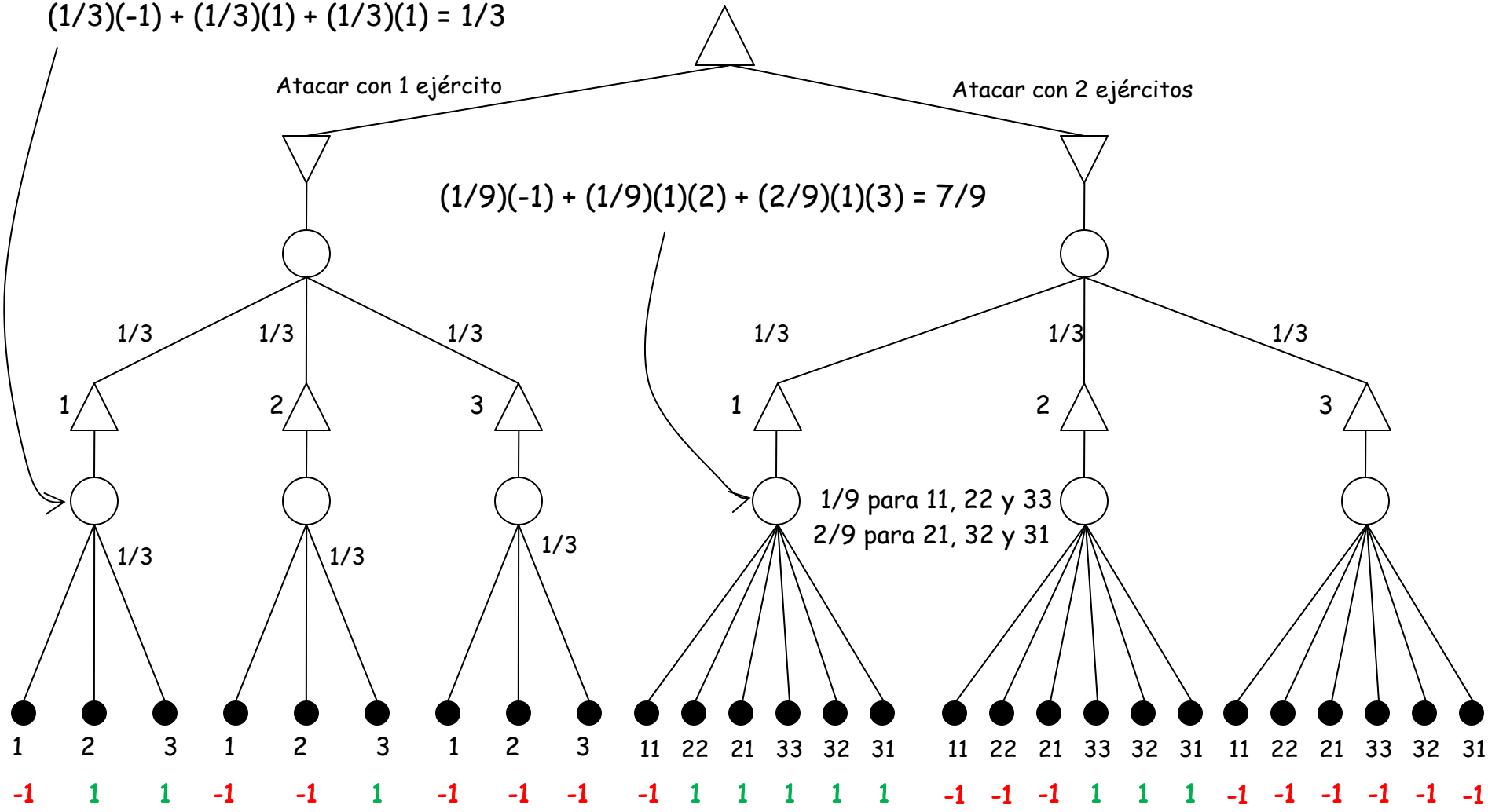

Atacar con 1 ejército
Atacar con 2 ejércitos
1/3
1/3
1/3
1/3
1/3
1/3
1
2
3
1
2
3
1/3
1/3
1/3
1/9 para 11, 22 y 33
2/9 para 21, 32 y 31
1 2 3 1 2 3 1 2 3 11 22 21 33 32 31 11 22 21 33 32 31 11 22 21 33 32 31
-1 1 1 -1 -1 1 -1 -1 -1 -1 1 1 1 1 1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 -1

Juegos

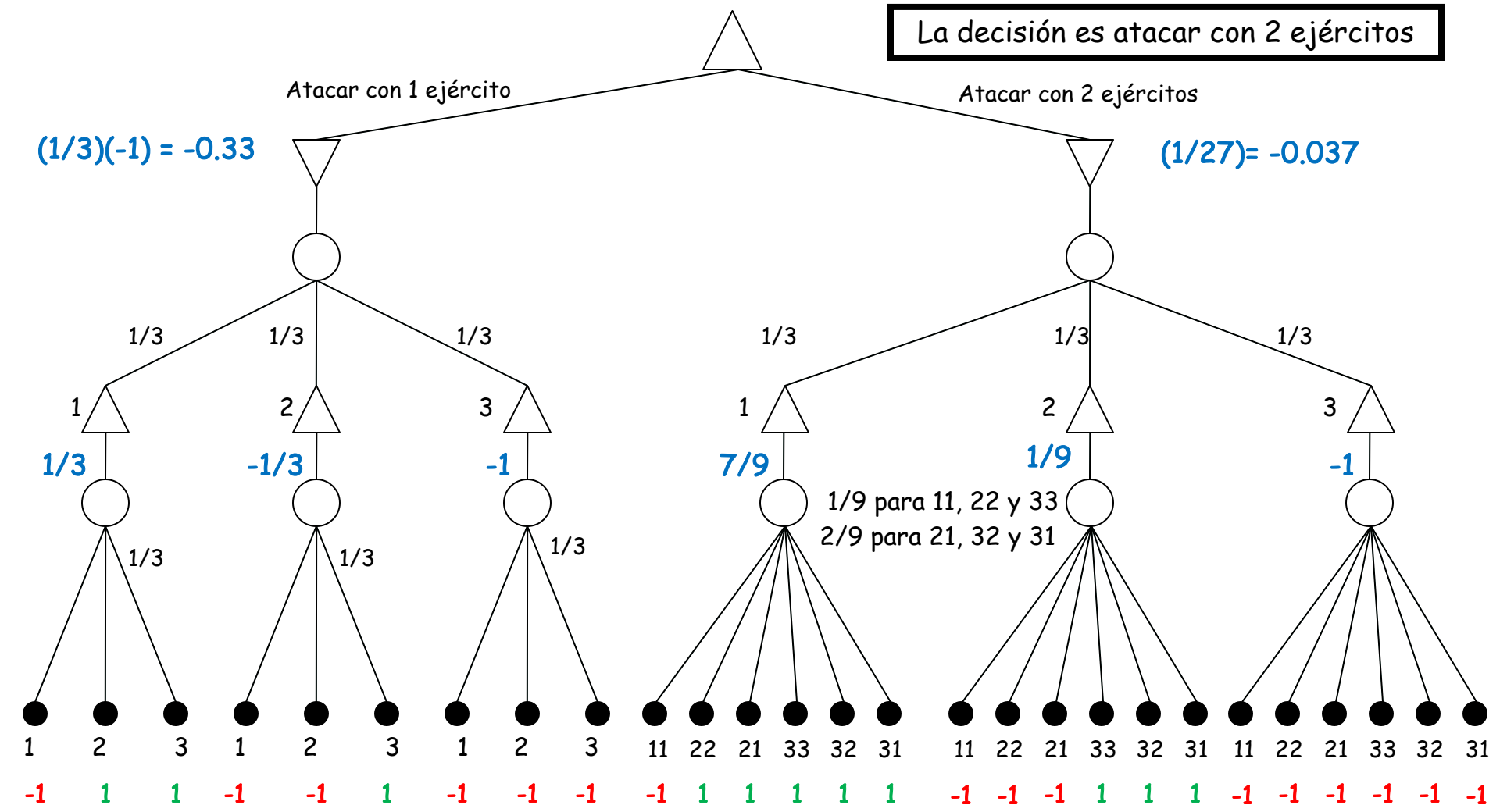


Indique la decisión minimax

Juegos


(1/3)(-1) + (1/3)(1) + (1/3)(1) = 1/3
Atacar con 1 ejército
Atacar con 2 ejércitos
(1/9)(-1) + (1/9)(1)(2) + (2/9)(1)(3) = 7/9
1/3
1/3
1/3
1
2
3
1/3
1/3
1/3
1/9 para 11, 22 y 33
2/9 para 21, 32 y 31
1 2 3 1 2 3 1 2 3 11 22 21 33 32 31 11 22 21 33 32 31 11 22 21 33 32 31
-1 1 1 -1 -1 1 -1 -1 -1 -1 1 1 1 1 1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 -1

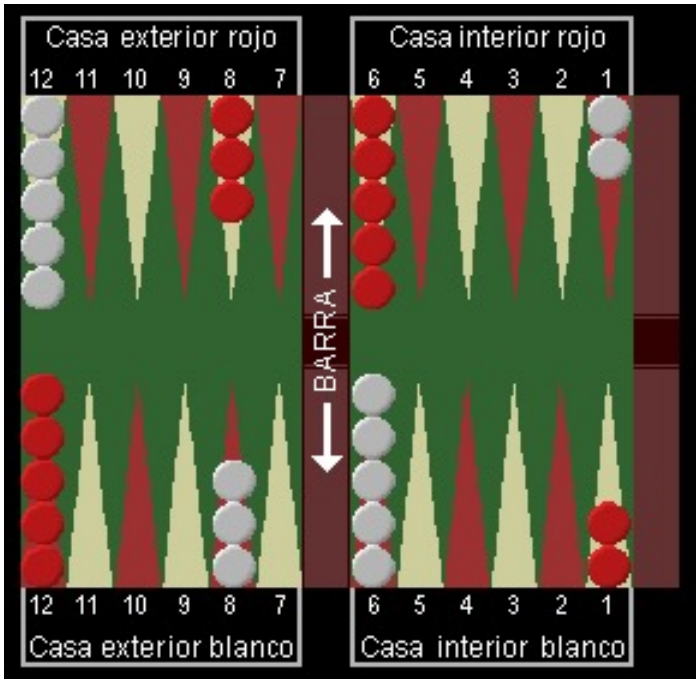
Juegos


La decisión es atacar con 2 ejércitos
Atacar con 1 ejército
Atacar con 2 ejércitos
(1/3)(-1) = -0.33
(1/27)= -0.037
1/3
1/3
1/3
1/3
1/3
1/3
1
2
3
1
2
3
1/3
-1/3
-1
7/9
1/9
-1
1/9 para 11, 22 y 33
2/9 para 21, 32 y 31
1/3
1/3
1/3
1 2 3 1 2 3 1 2 3 11 22 21 33 32 31 11 22 21 33 32 31 11 22 21 33 32 31
-1 1 1 -1 -1 1 -1 -1 -1 -1 1 1 1 1 1 -1 -1 -1 1 1 1 -1 -1 -1 -1 -1 -1

Juegos

Juegos con elemento aleatorio

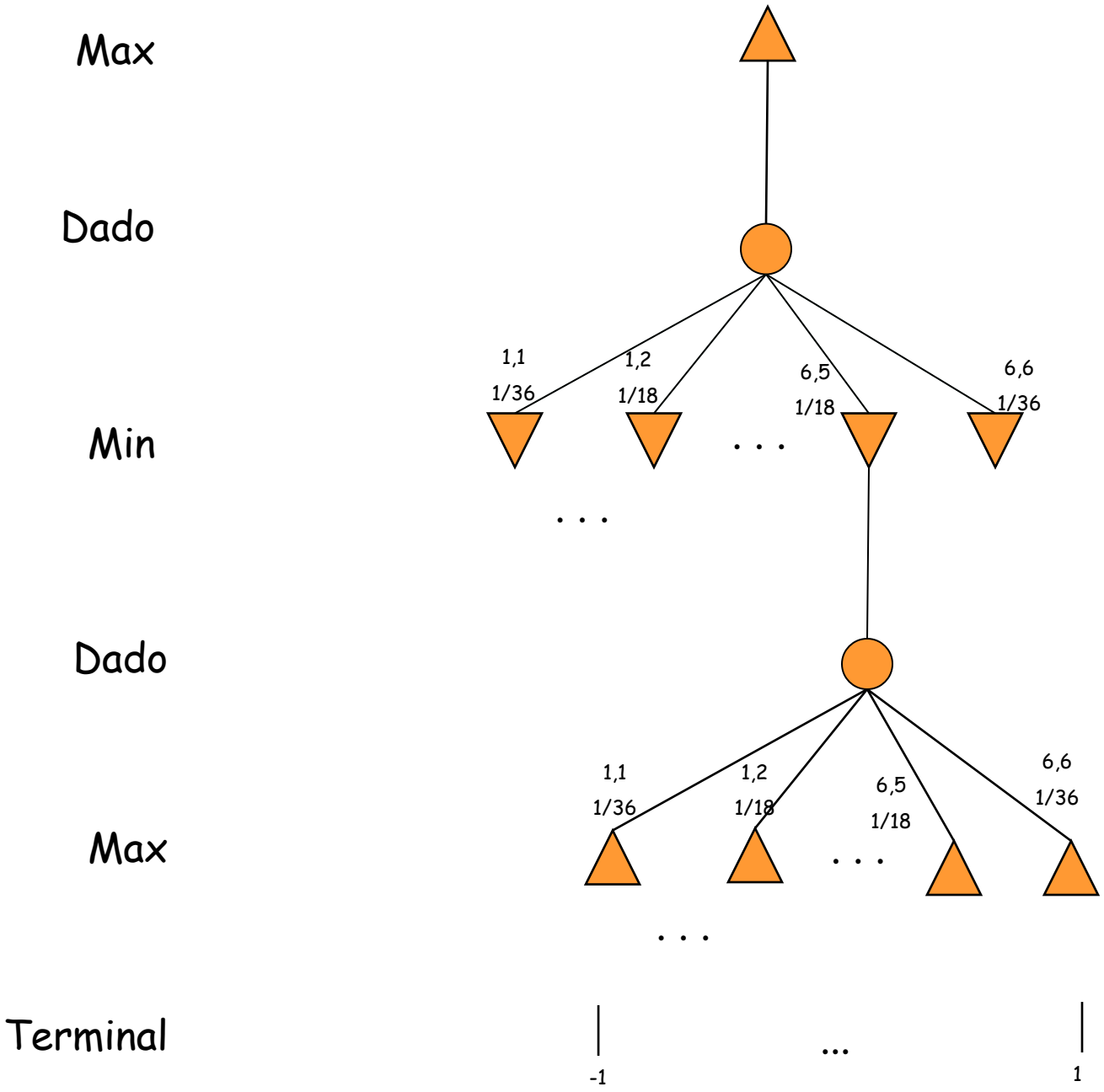
- El backgammon combina estrategia y suerte



Juegos

Juegos con elemento aleatorio

- Aunque el jugador de las fichas blancas sabe sus jugadas permitidas, ignora qué valores obtendrá su contrincante al lanzar los dados
- Un árbol de juego en el backgammon debe incluir **nodos aleatorios**, además de los nodos MAX y MIN


Max
Dado
Min
Dado
Max
Terminal
1,1
1/36
1,2
1/18
6,5
1/18
6,6
1/36
. . .
. . .
1,1
1/36
1,2
1/18
6,5
1/18
6,6
1/36
. . .
. . .
...
-1
1

Juegos

Backgammon

- Primer programa fue BKG que utilizaba una función de evaluación heurística
- En 1980 derrotó al campeón mundial por 5-1
- Como interviene la suerte, BKG puede tener buenas y malas partidas