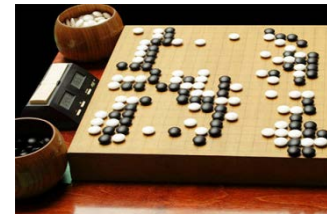


Tema 4: Búsqueda con adversario: juegos

- Generalidades sobre Juegos
- Juegos bipersonales con información perfecta
- Árboles de exploración de juegos
- Algoritmo Minimax
- Poda alfa-beta
- Juegos en los que interviene un elemento aleatorio



Objetivos

- Adquirir las habilidades básicas para construir sistemas capaces de resolver problemas mediante técnicas de IA.
- Entender que la resolución de problemas en IA implica definir una representación del problema y un proceso de búsqueda de la solución.
- Analizar las características de un problema dado y determinar si es susceptible de ser resuelto mediante técnicas de búsqueda. Decidir en base a criterios racionales la técnica más apropiada para resolverlo y saber aplicarla.
- **Conocer las técnicas básicas de búsqueda con adversario (minimax, poda alfa-beta) y su relación con los juegos.**

Estudia el tema en ...

- Nils J. Nilsson, “*Inteligencia Artificial: Una nueva síntesis*”, Ed. Mc Graw Hill, 2000. pp. 175-192
- S. Russell, P. Norvig, “*Inteligencia Artificial: Un enfoque moderno*”, Ed. Prentice Hall, 2ª edición, 2004.

Interés de los Juegos

- Laboratorios perfectos para investigar en técnicas de resolución de problemas.
- Es fácil medir el éxito o el fracaso.
- Fascinación para cierta gente.
- Aspecto comercial.
- Aplicaciones en ámbitos empresariales.

Entornos con múltiples agentes

- Hasta el momento, hemos considerado entornos donde hay un único agente.
- Ahora vamos a considerar entornos en los que dos o más agentes intervienen simultáneamente, de modo que las acciones de un agente influyen, en mayor o menor medida, en la percepción y/o toma de decisiones del resto de agentes.
- Hay una cierta diferencia de matiz entre **entornos con múltiples agentes** y **sistemas multiagente**: “coexistencia” vs. “divide y vencerás”

Entornos con múltiples agentes

- Existen dos situaciones básicas:
 - **Entornos cooperativos**, donde los agentes colaboran para alcanzar un mismo fin (ejemplo: Inteligencia de enjambres).
 - **Entornos competitivos**, donde los agentes tienen metas contrapuestas.

Entornos con múltiples agentes

En un **entorno cooperativo**, los agentes pueden comunicarse o no entre ellos, tener las mismas o distintas habilidades, tener las mismas o diferentes metas, etc.

Ejemplo de entorno multiagente cooperativo: **RoboCup**.



Entornos con múltiples agentes

- En un **entorno competitivo**, los agentes son adversarios que se enfrentan por conseguir su meta particular.
- Estas situaciones se estudian en el ámbito de la **Teoría de Juegos** desarrollada como tal por **John von Neumann** y por **Oskar Morgenstern** en 1944 como una aproximación a la Economía
- La investigación y resultados de **John F. Nash** (el protagonista de ***Una Mente Maravillosa***) aún se utilizan hoy día como parte clave y central de la teoría de juegos (Teoría del Equilibrio).

Juegos

- ¿Qué es un juego?

Es cualquier situación de decisión con varios agentes, (jugadores) gobernada por un conjunto de reglas y con un resultado bien definido, caracterizada por que ninguno de los jugadores con su sola actuación puede determinar el resultado (interdependencia estratégica).

Características de los juegos

- ❑ Número de jugadores
- ❑ Juegos de **información perfecta** vs. Juegos de **información imperfecta: Ajedrez vs. Poquer**
- ❑ Existencia de movimientos de azar,
- ❑ Orden de actuación de los jugadores
- ❑ Existencia o no de pagos colaterales (equilibrio Nash)
- ❑ Juegos de **suma nula** vs. Juegos de **suma no nula**

Juegos bipersonales

- **Ejemplo 1: El dilema del prisionero**

- Dos individuos son detenidos por la policía debido a que cometieron cierto delito. Ambos son encerrados en celdas diferentes y son interrogados de forma individual. Ambos tienen dos alternativas: no confesar o delatar al compañero.
- Saben que si ninguno confiesa, ambos irán a la cárcel por 2 años, pero si uno delata a su compañero y el otro no, entonces al que confiesa le absuelven y al otro le encierran por 10 años. Si ambos confesasen, entonces la pena se repartiría y ambos irían a prisión por 5 años.

Juegos bipersonales con información perfecta

		Prisionero 1	
		No delatar	Delatar
Prisionero 2	No delatar	$(-2, -2)$	$(0, -10)$
	Delatar	$(-10, 0)$	$(-5, -5)$

- **¿Qué harán los prisioneros?** Con toda lógica: Cooperar. Sin embargo, la tentación de hacer la promesa de no delatar, para después traicionar al compañero es muy grande.
- El juego no es de suma nula y tiene una estructura no cooperativa.

Juegos bipersonales de suma nula

- **Ejemplo : El juego de NIM**

- (una versión simple)**

- Inicialmente, hay n palillos sobre la mesa, y dos jugadores A y B. El jugador A comienza el juego quitando 1, 2 ó 3 palillos. Le sigue el jugador B, que también podrá quitar 1, 2 ó 3 palillos. El turno vuelve al jugador A, y estas acciones se repiten hasta que quede un único palillo en la mesa. Aquel que quite este último palillo pierde el juego.



Generalidades sobre juegos

- En un juego, cada jugador intenta conseguir el mayor beneficio para sus intereses:
- La solución de un juego es la determinación de una sucesión de actuaciones que indican a cada jugador qué resultado puede esperar y cómo alcanzarlo.

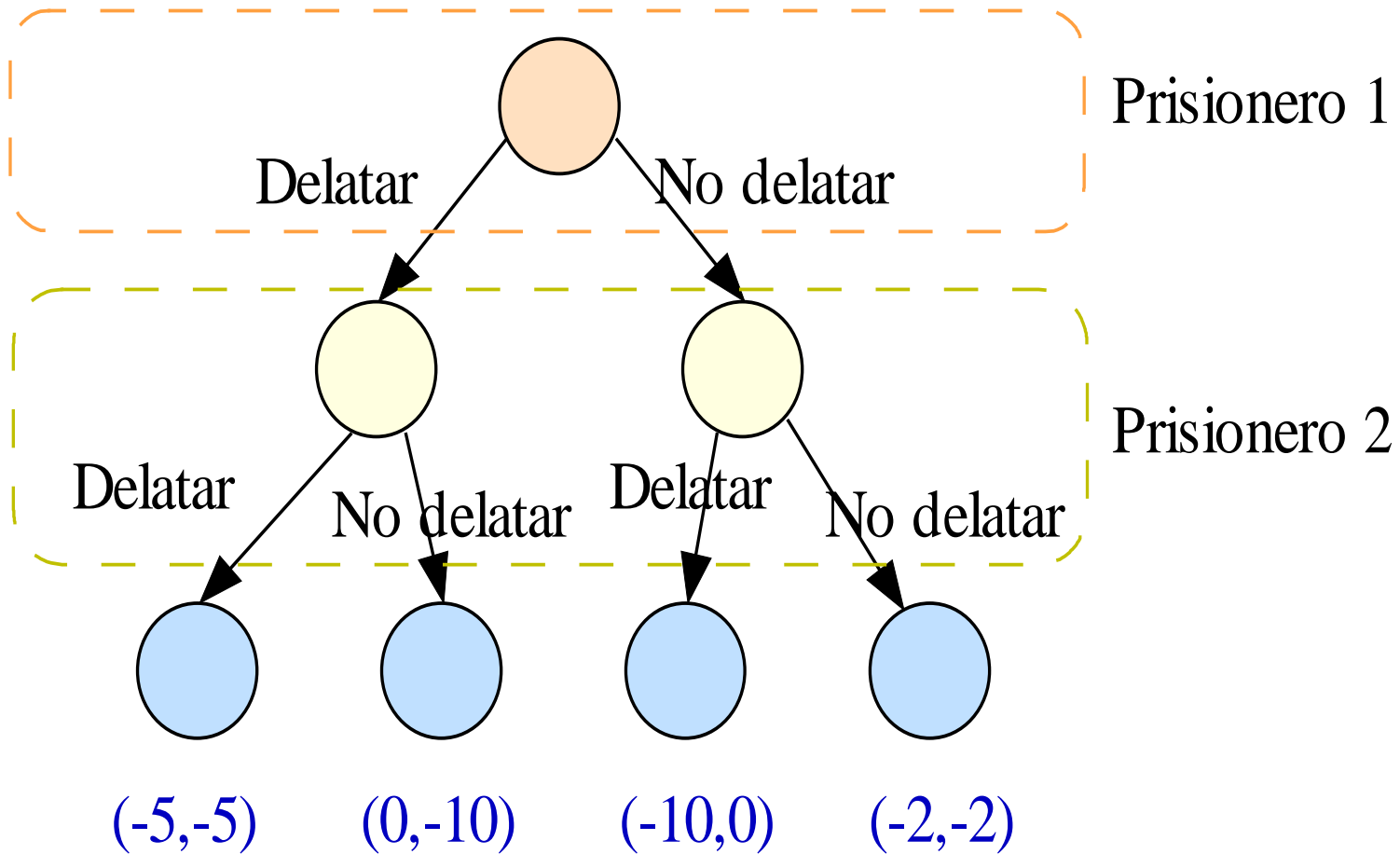
Generalidades

Por tanto, un juego puede plantearse como un **problema de maximización**, aunque finalmente y en muchos casos solo pueda llegarse a una **satisfacción**.

Árboles de exploración de juegos

- Un árbol del juego es una representación explícita de todas las formas de jugar a un juego
- Correspondencia entre árboles de juegos y árboles Y/O

Árbol de un Juego. Estrategias



Árbol de un Juego. Estrategias

- **Ejemplo:** El juego de NIM en la versión expuesta

- ¿Cómo debe jugarse para ganar?

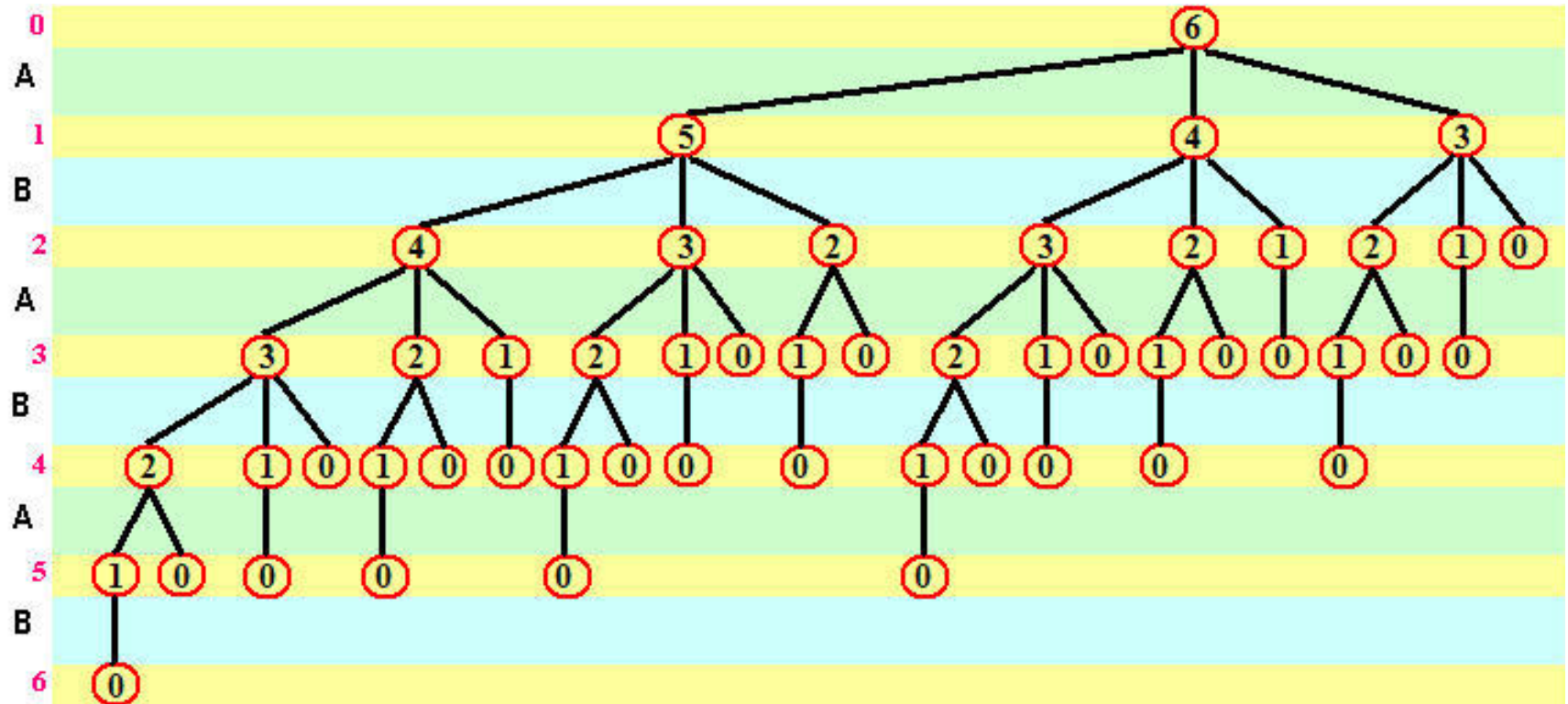


Árbol de un Juego. Estrategias

- Una partida de esta versión del Nim jugada con una sola pila conformada por 9 fichas podría transcurrir del siguiente modo:
- el jugador *A* retira 3 fichas (la pila se reduce a 6), luego el jugador *B* retira 1 ficha (la pila se reduce a 5), luego el jugador *A* retira 2 fichas (la pila se reduce a 3), luego el jugador *B* retira 2 fichas (la pila se reduce a 1 ficha), y luego al jugador *A* obligatoriamente le corresponde tomar la única ficha que quedó, razón por la cual el triunfo es para el jugador *B*

- . Este juego tan simple es eminentemente estratégico, porque según sea el número n de fichas que quedan en la pila, entonces cada jugador debe calcular cuántas fichas k conviene retirar en su turno para forzar al contrincante hacia una situación en la que obligatoriamente al final le corresponda retirar la última ficha que quede de la pila.

Árbol de un Juego. Estrategias



Arbol de Decisiones del Nim jugado con 6 fichas.

www.eyeintheskygroup.com

Objetivos

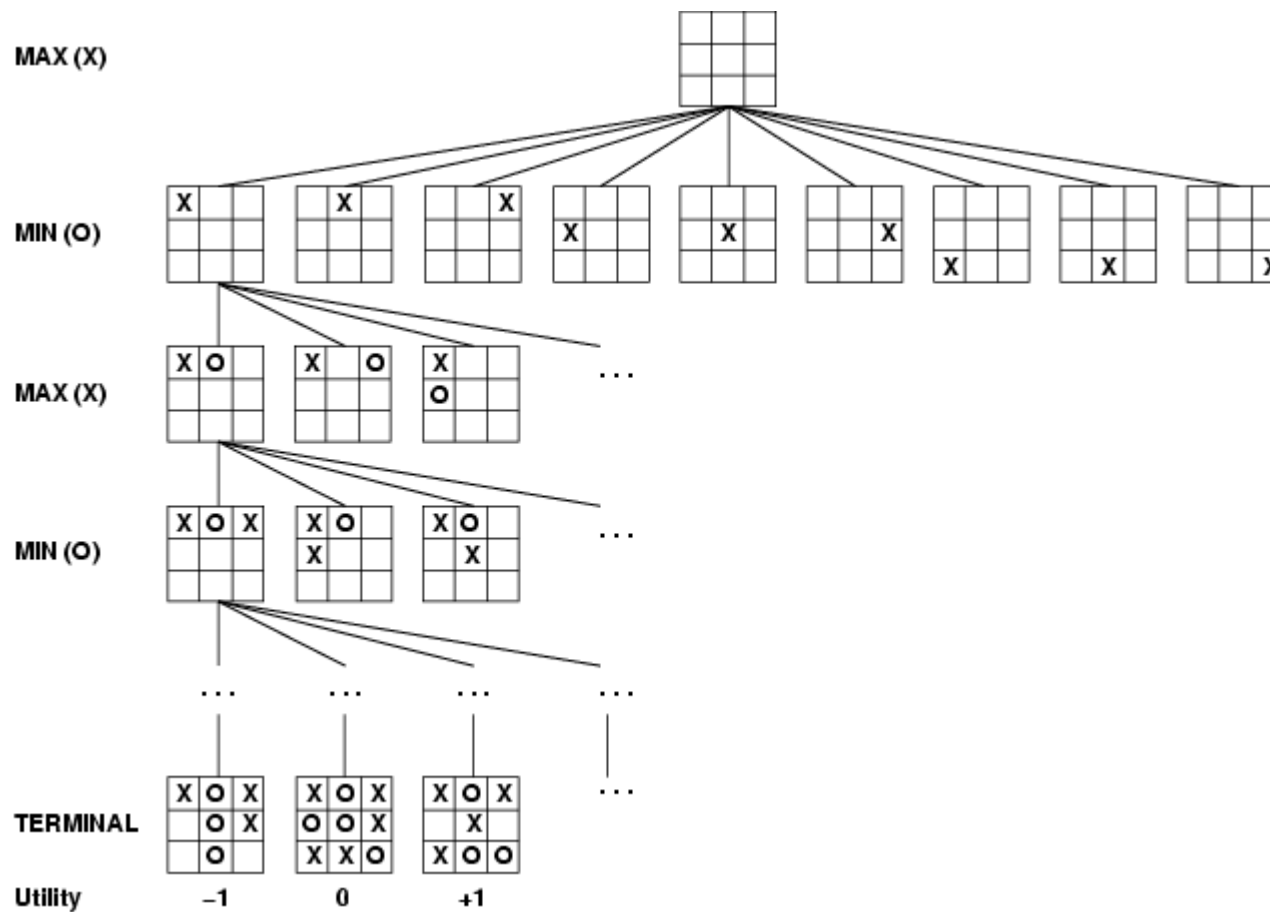
A partir de ahora nos centraremos en:

- Juegos bipersonales,
- Juegos con información perfecta,
- Suma nula

Notación min-max

- MAX: primer jugador
- MIN: segundo jugador
- Nodos MAX y nodos MIN
- Supongamos que los nodos terminales se etiquetan con V (victoria) D (derrota) o E (empate) desde el punto de vista de MAX

Ejemplo (tres en raya)



Algoritmo STATUS

- Si J es un nodo MAX no terminal, entonces STATUS(J)=
 - V si alguno de los sucesores de J tiene STATUS V
 - D si todos los sucesores de J tienen STATUS D
 - E en otro caso
- Si J es un nodo MIN no terminal, entonces STATUS(J)=
 - V si todos los sucesores de J tienen STATUS V
 - D si alguno de los sucesores de J tiene STATUS D
 - E en otro caso

Nuevo modelo de solución

- Los juegos complejos no se pueden resolver ya que es imposible la exploración total hasta la terminación
- Nuevo objetivo: encontrar una buena jugada inmediata
- Importancia de la heurística en el proceso

La regla minimax

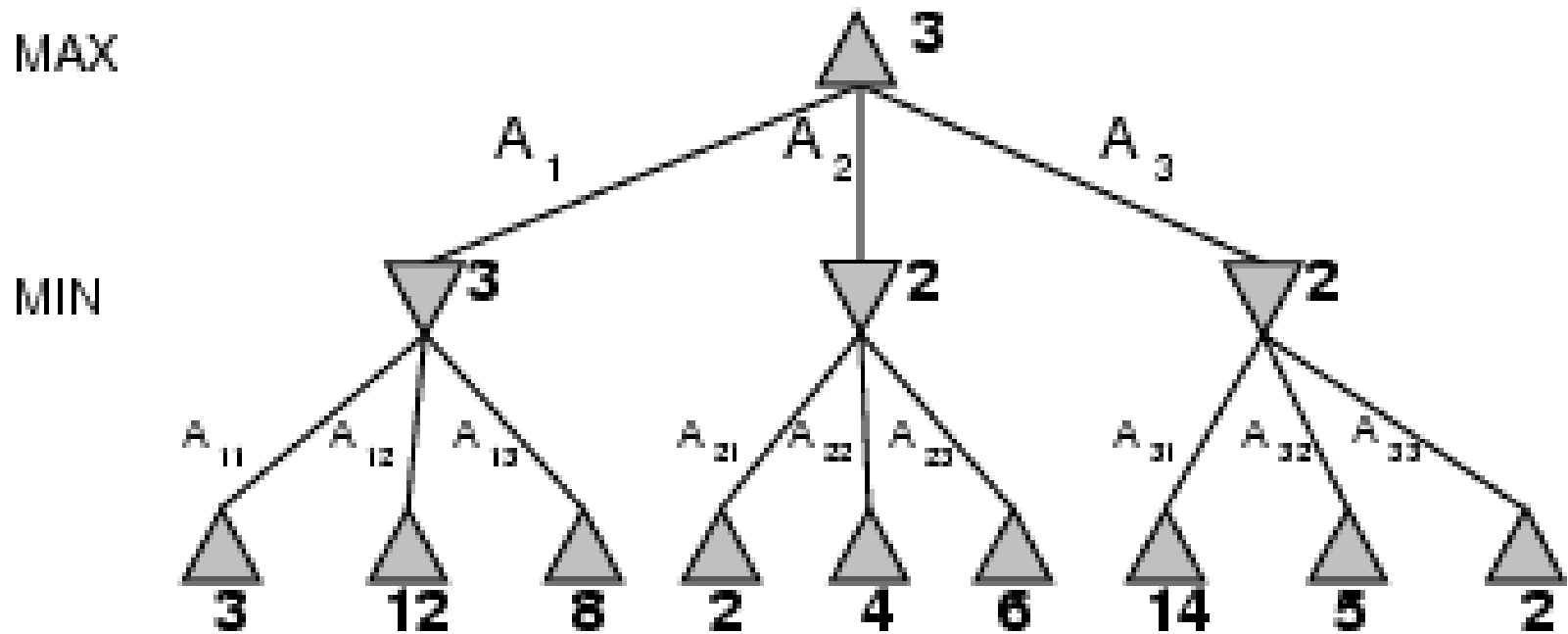
- El valor $V(J)$ de un nodo J de la frontera de búsqueda es igual al de su evaluación estática; en otro caso
- Si J es un nodo MAX, entonces su valor $V(J)$ es igual al máximo de los valores de sus nodos sucesores
- Si J es un nodo MIN, entonces su valor $V(J)$ es igual al mínimo de los valores de sus nodos sucesores.

Algoritmo Minimax

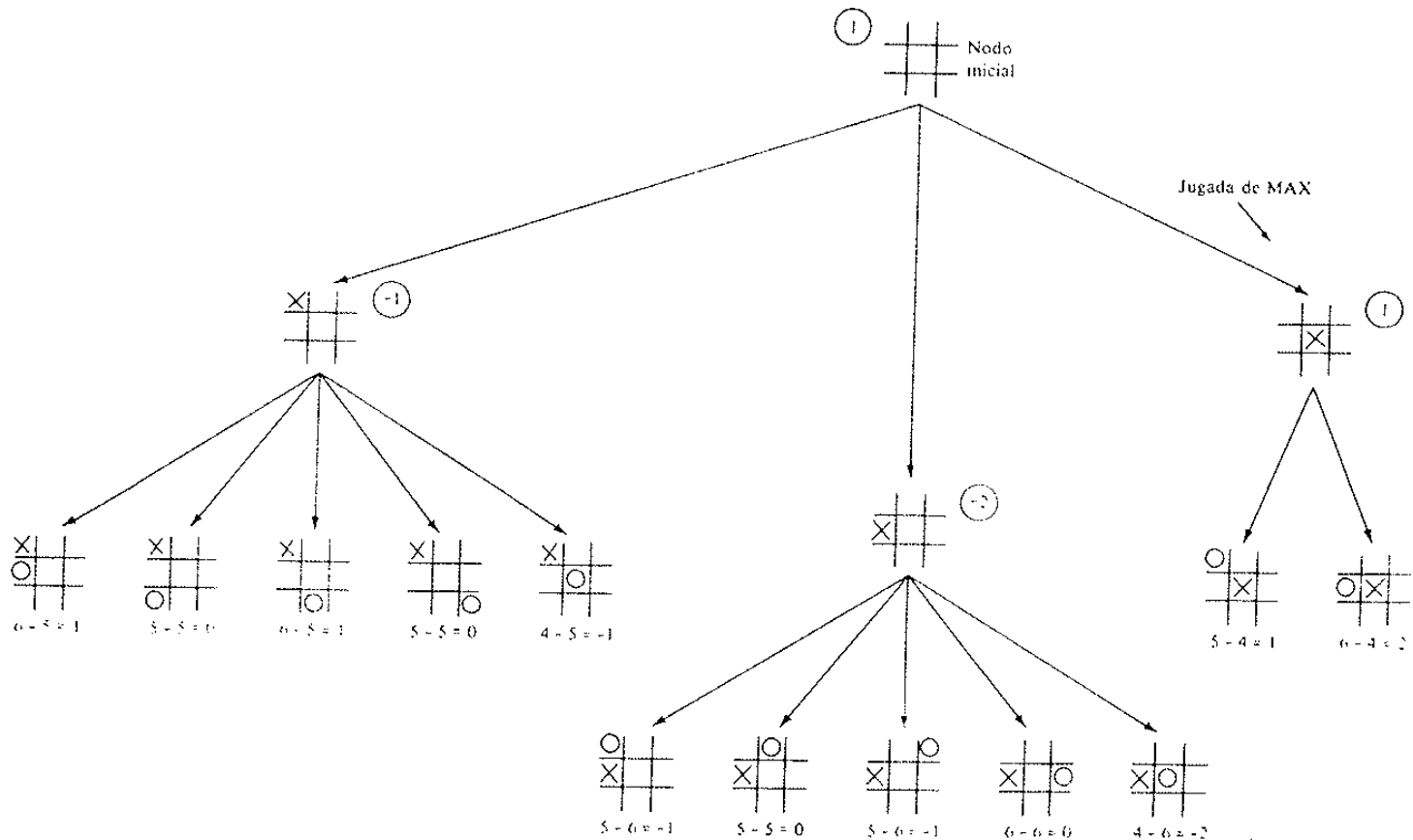
Para determinar el valor minimax, $V(J)$ de un nodo J , hacer lo siguiente:

- Si J es un nodo terminal, devolver $V(J)=f(J)$; en otro caso
- Para $k=1,2,\dots,b$, hacer:
 - Generar J_k , el k -ésimo sucesor de J
 - Calcular $V(J_k)$
 - Si $k=1$, hacer $AV(J) \leftarrow V(J_1)$; en otro caso, para $k \geq 2$,
 - hacer $AV(J) \leftarrow \max\{AV(J), V(J_k)\}$ si J es un nodo MAX o
 - hacer $AV(J) \leftarrow \min\{AV(J), V(J_k)\}$ si J es un nodo MIN
- Devolver $V(J)=AV(J)$

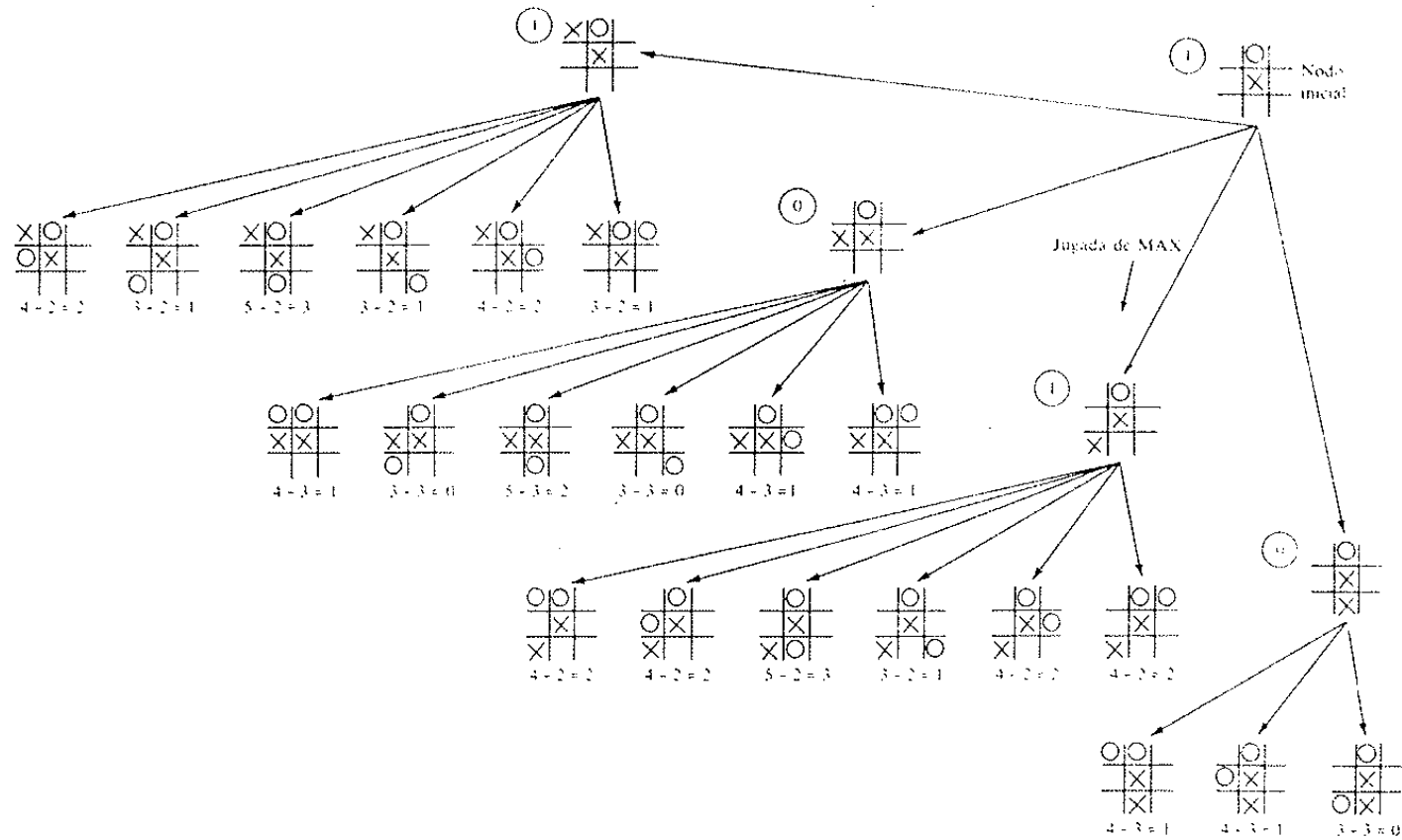
Ejemplo



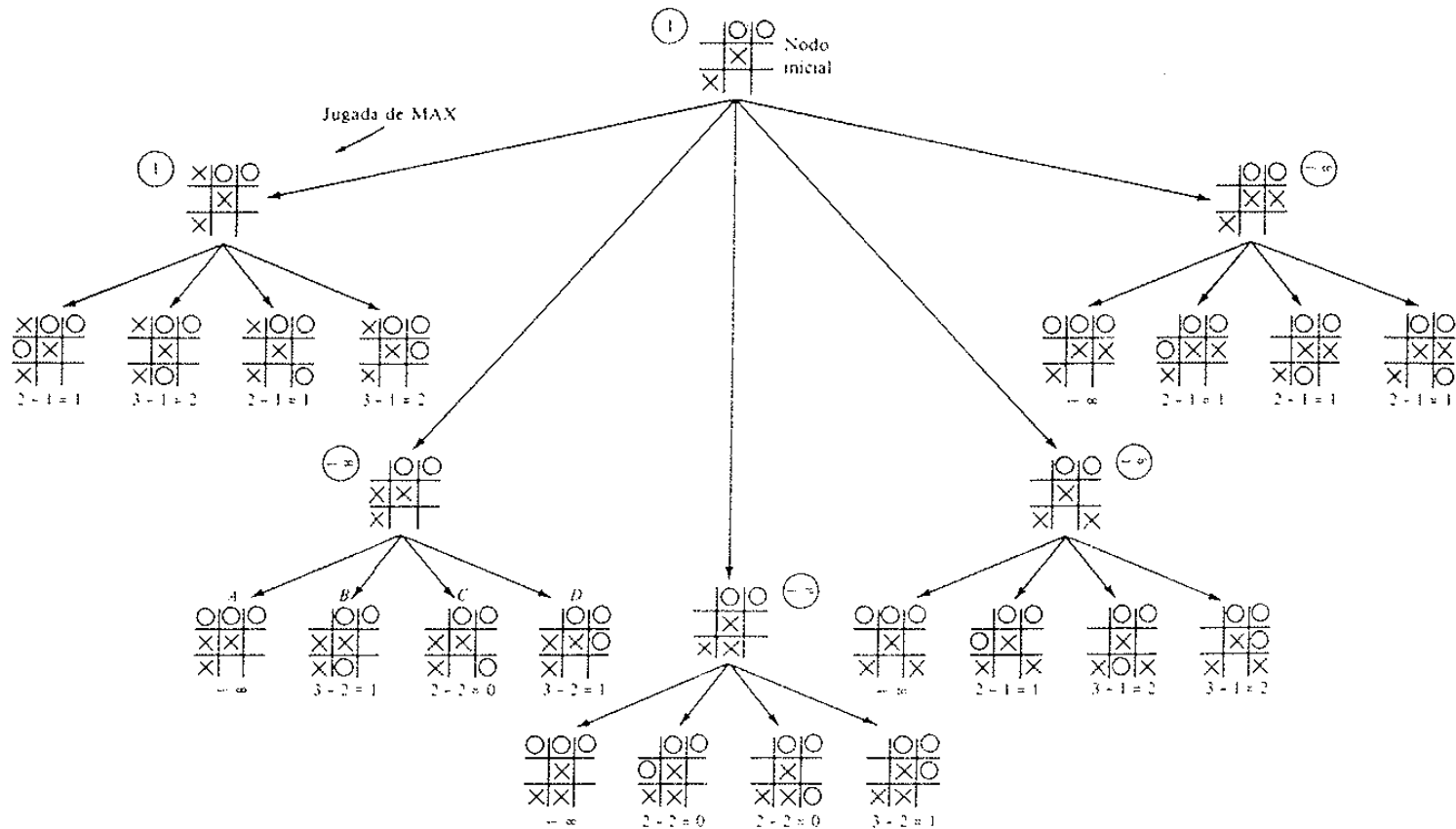
Ejemplo



Ejemplo



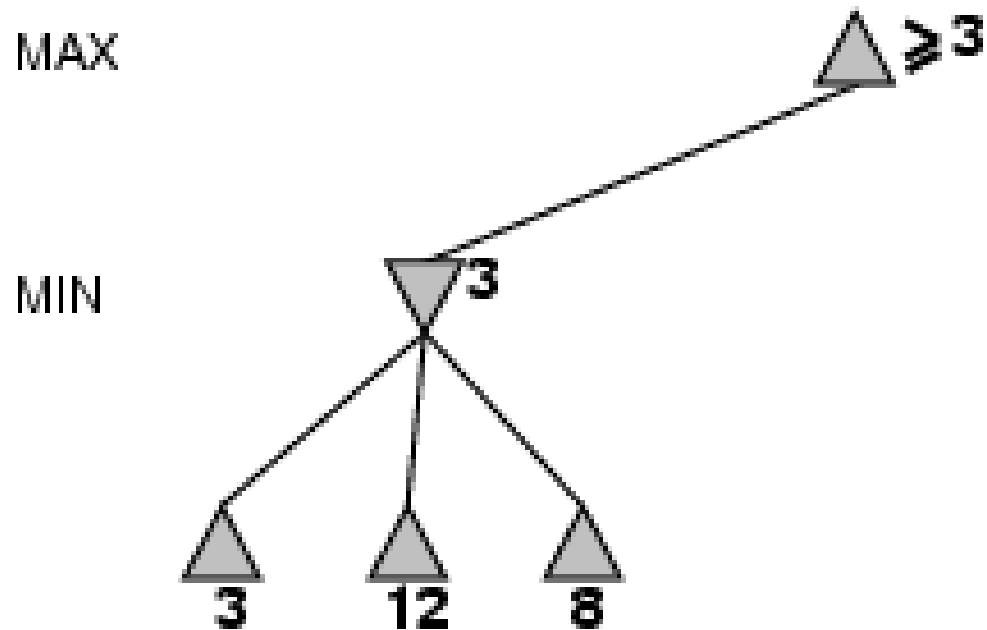
Ejemplo



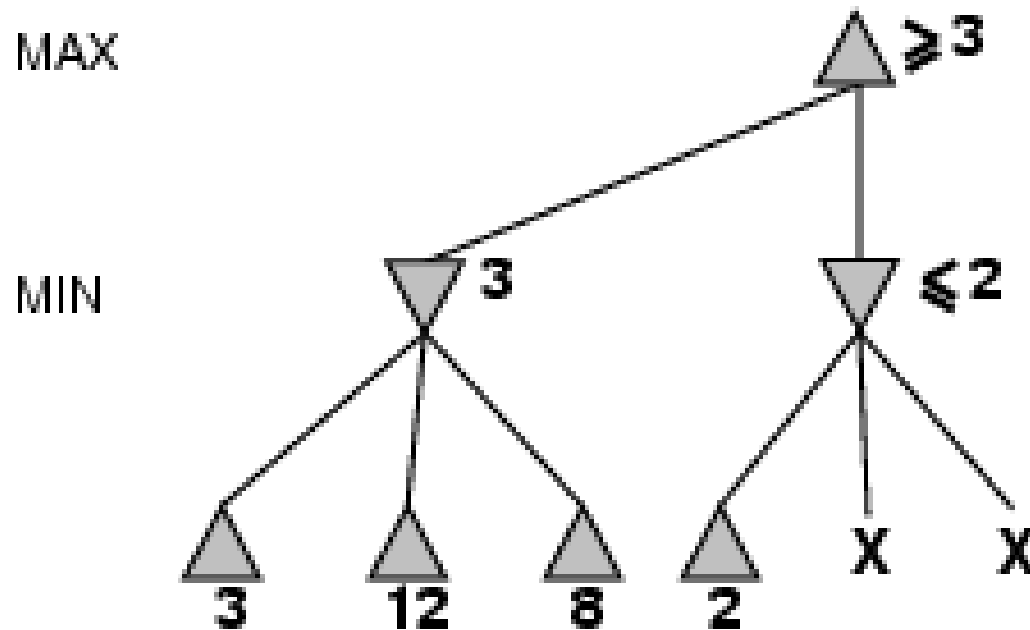
Poda alfa-beta

- ¿podríamos obtener el mismo resultado que el algoritmo minimax con menos esfuerzo computacional?

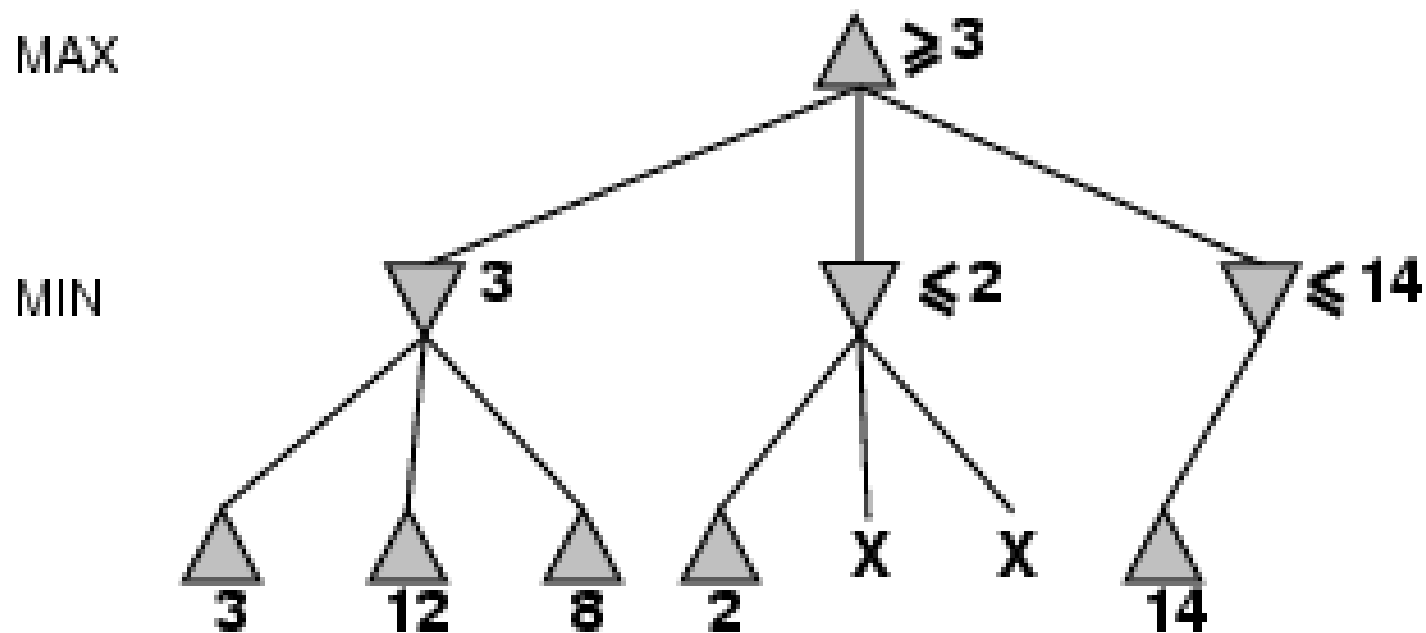
Ejemplo poda



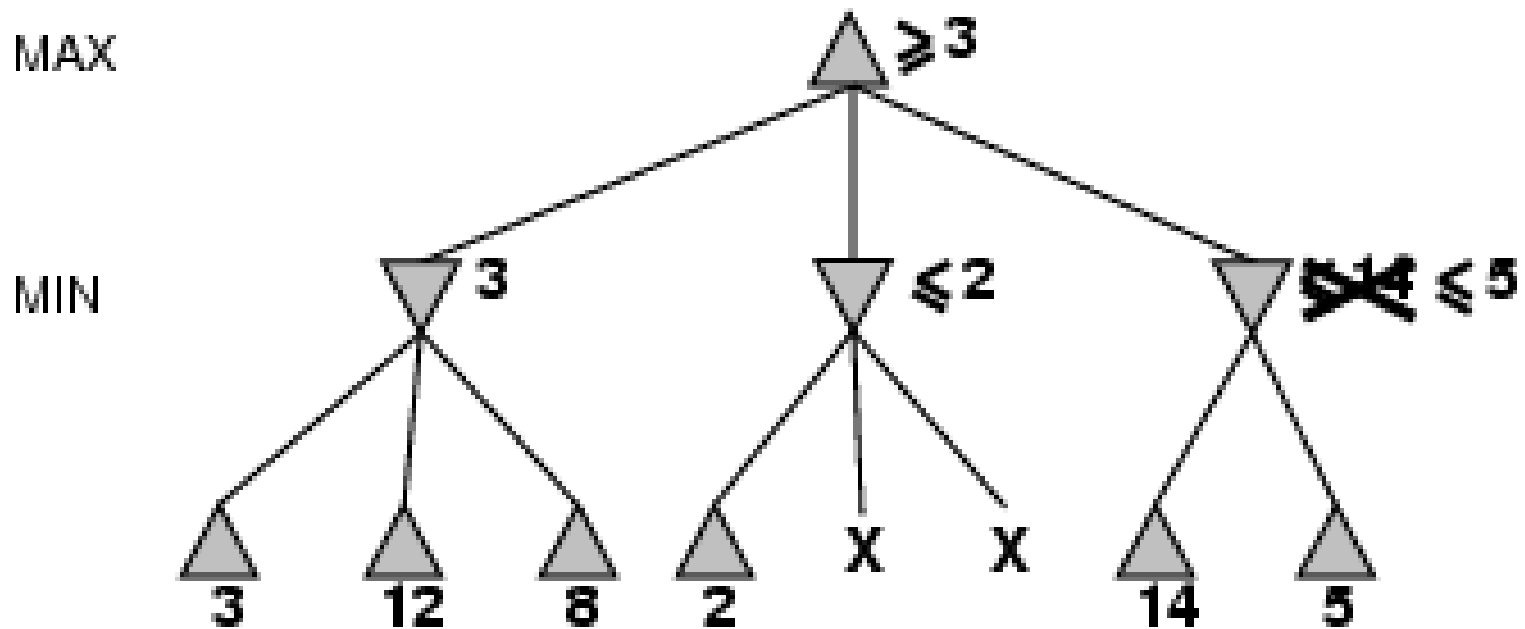
Ejemplo poda



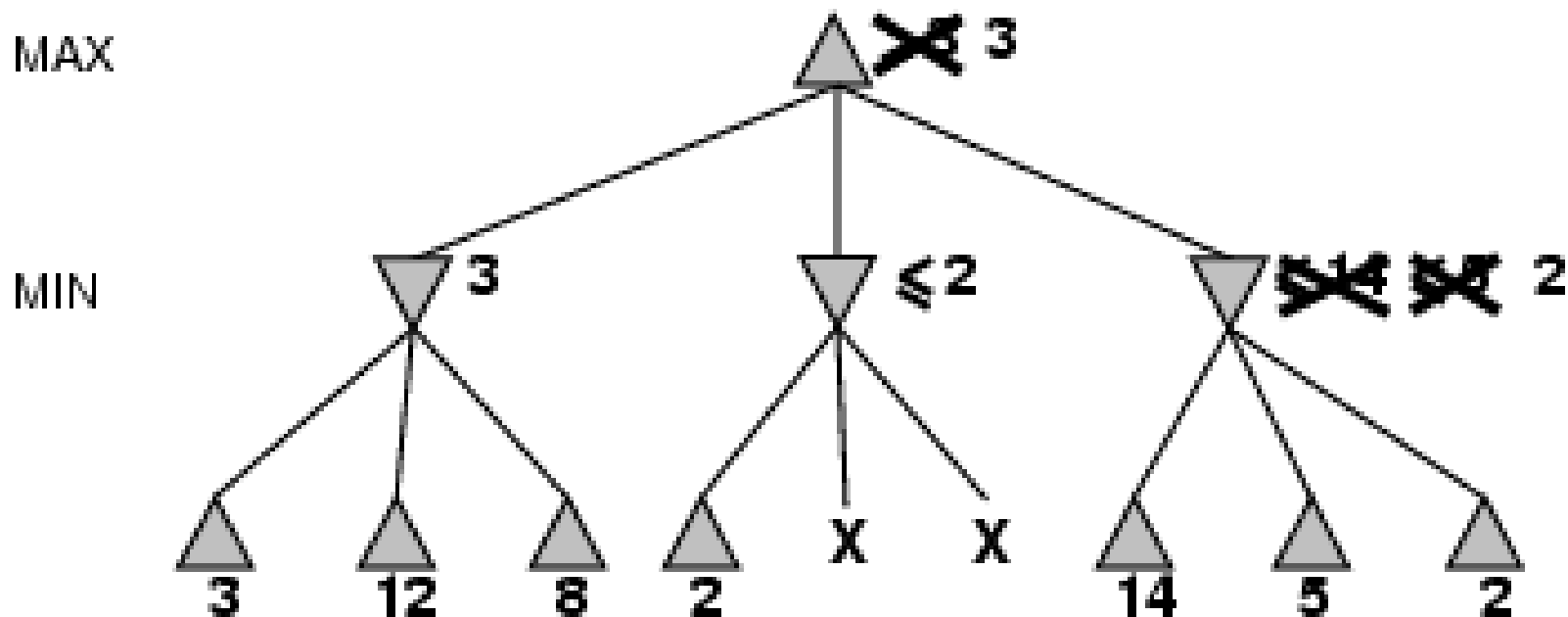
Ejemplo poda



Ejemplo poda



Ejemplo poda



Poda alfa-beta

- Utiliza dos variables a lo largo del proceso de búsqueda en profundidad: alfa y beta.
- Al principio del proceso de búsqueda alfa toma el valor $-\infty$, mientras que beta toma el valor $+\infty$.
- Desde un nodo MAX, iremos actualizando el valor de alfa a medida que vayamos explorando cada arco que cuelgue de ese nodo.

Poda alfa-beta

- Dicha actualización consiste en comparar el valor obtenido por cada arco con el valor actual de alfa. Si el valor obtenido es mayor, ése será el nuevo valor de alfa. Por tanto, alfa almacena el mejor valor que hemos encontrado desde el nodo en que estamos. Sólo nos interesará proseguir la búsqueda desde ese nodo si podemos mejorar alfa.

Poda alfa-beta

- En caso de que estemos en un nodo MIN, iremos actualizando el valor de beta a medida que vayamos explorando cada arco que cuelgue de ese nodo. Dicha actualización consiste en comparar el valor obtenido por cada arco con el valor actual de beta. Si el valor obtenido es menor, ése será el nuevo valor de beta.

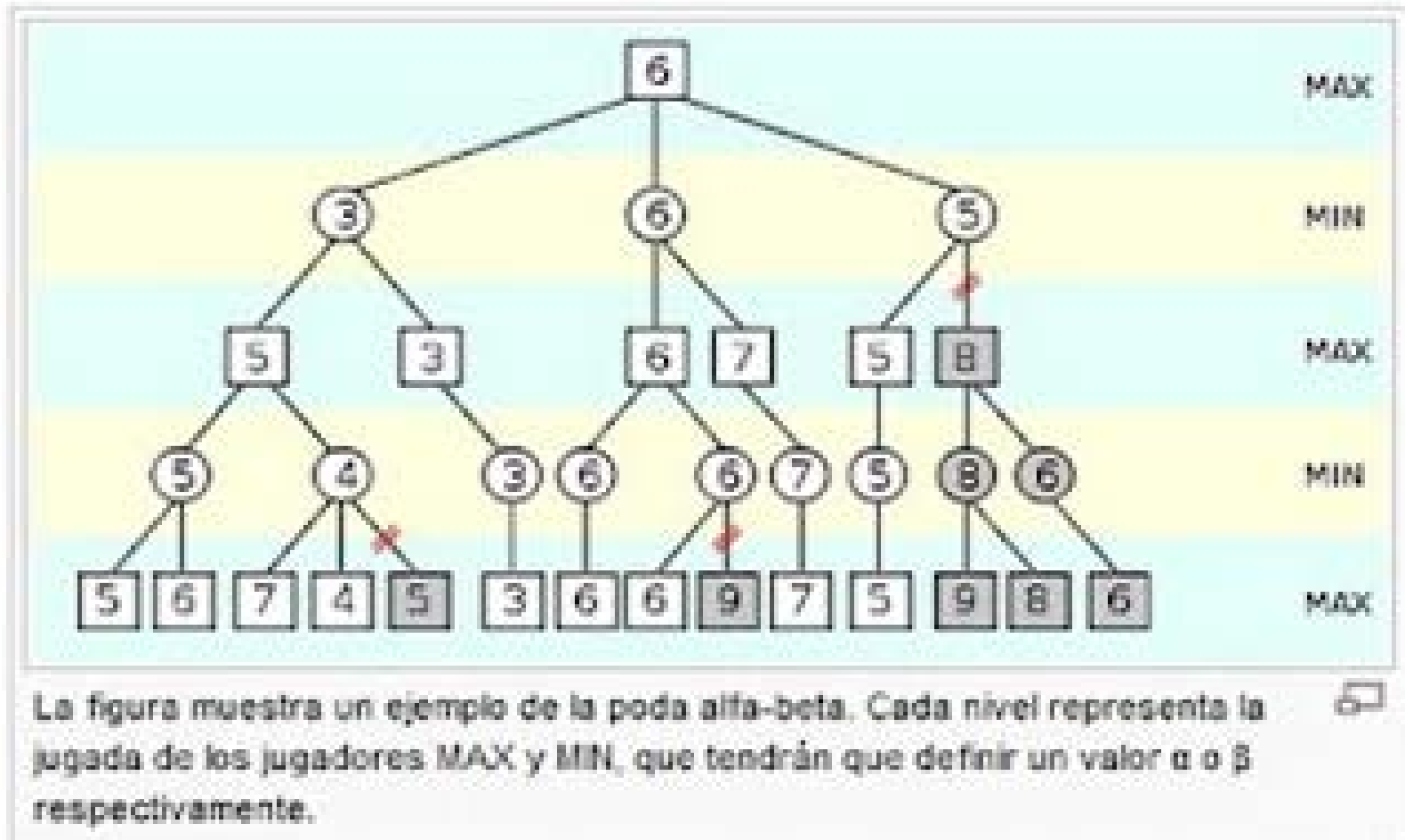
Poda alfa-beta

- Por tanto, beta almacena el peor valor (para MAX) encontrado por MIN hasta el momento. Sólo interesará que la búsqueda prosiga si MIN pudiera reducir beta.

Poda alfa-beta

- El intervalo (alfa,beta) contiene los valores que se puede seguir consiguiendo en el proceso de búsqueda.
- Cuando alfa crece tanto que sobrepasa a beta, o cuando beta disminuye tanto que se hace menor que alfa, se puede hacer una poda y no seguir explorando el subárbol que cuelga desde el nodo en que nos encontramos, prosiguiendo la búsqueda en profundidad desde su nodo padre.

Poda alfa-beta



- El valor alfa de un nodo *MAX* es igual al mayor de todos los valores propagados de sus sucesores obtenidos hasta el momento.
- El valor beta de un nodo *MIN* es igual al menor de todos los valores propagados de sus sucesores obtenidos hasta el momento.

Algoritmo ALFA-BETA

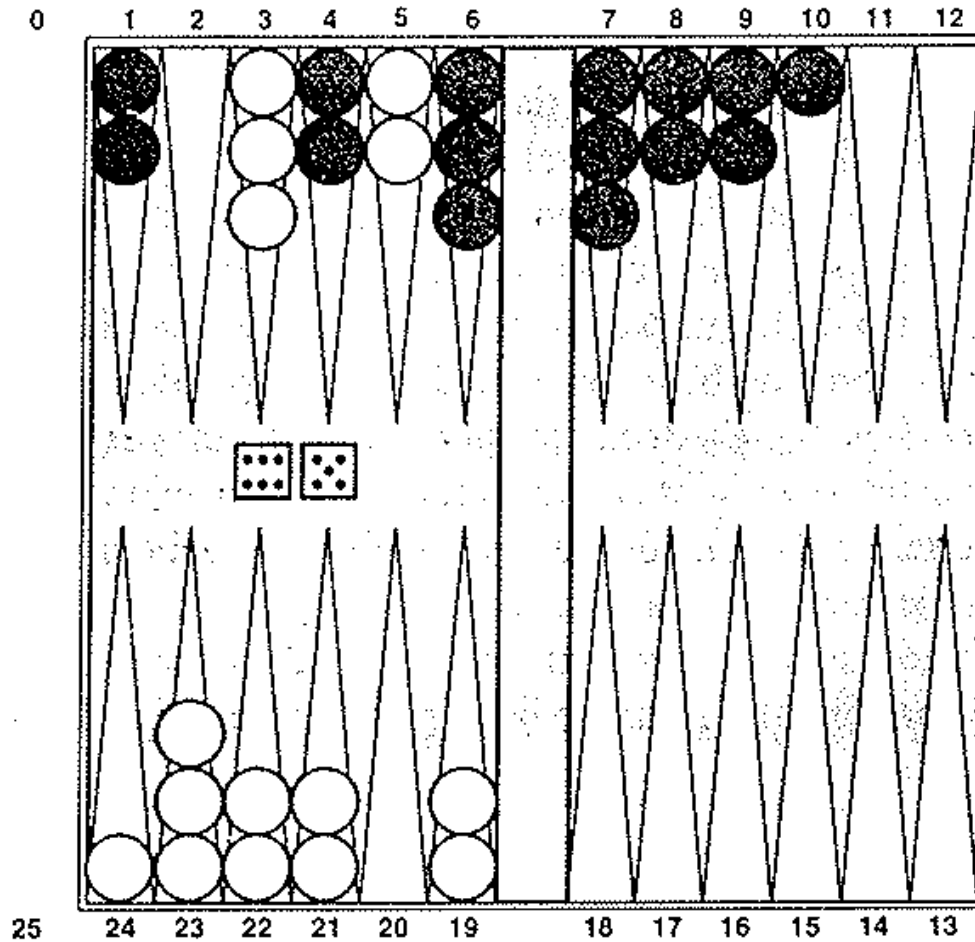
Para calcular el valor $V(J, \alpha, \beta)$, hacer lo siguiente:

1. Si J es un nodo terminal, devolver $V(J)=f(J)$. En otro caso, sean $J_1, \dots, J_k, \dots, J_b$ los sucesores de J . Hacer $k \leftarrow 1$ y, si J es un nodo MAX ir al paso 2; si J es un nodo MIN ir al paso 5.
2. Hacer $\alpha \leftarrow \max(\alpha, V(J_k, \alpha, \beta))$.
3. Si $\alpha \geq \beta$ devolver β ; si no, continuar
4. Si $k=b$, devolver α ; si no, hacer $k \leftarrow k+1$ y volver al paso 2.
5. Hacer $\beta \leftarrow \min(\beta, V(J_k, \alpha, \beta))$.
6. Si $\beta \leq \alpha$ devolver α ; si no, continuar
7. Si $k=b$, devolver β ; si no, hacer $k \leftarrow k+1$ y volver al paso 5.

Poda ALFA-BETA

1. Se puede suspender la exploración por debajo de cualquier nodo *MIN* que tenga un valor beta menor o igual que el valor alfa de cualquiera de sus nodos *MAX* antecesores. El valor de propagación definitivo para este nodo *MIN* se puede hacer igual a su valor beta. Este valor puede que no sea el mismo que el que se obtendría mediante el procedimiento minimax completo, pero conduce a la selección del mismo movimiento.
2. La exploración puede suspenderse por debajo de cualquier nodo *MAX* que tenga un valor alfa mayor o igual que el valor beta de cualquiera de sus nodos *MIN* antecesores. El valor de propagación definitivo para este nodo se puede hacer igual a su valor alfa.

Juegos en los que interviene el azar



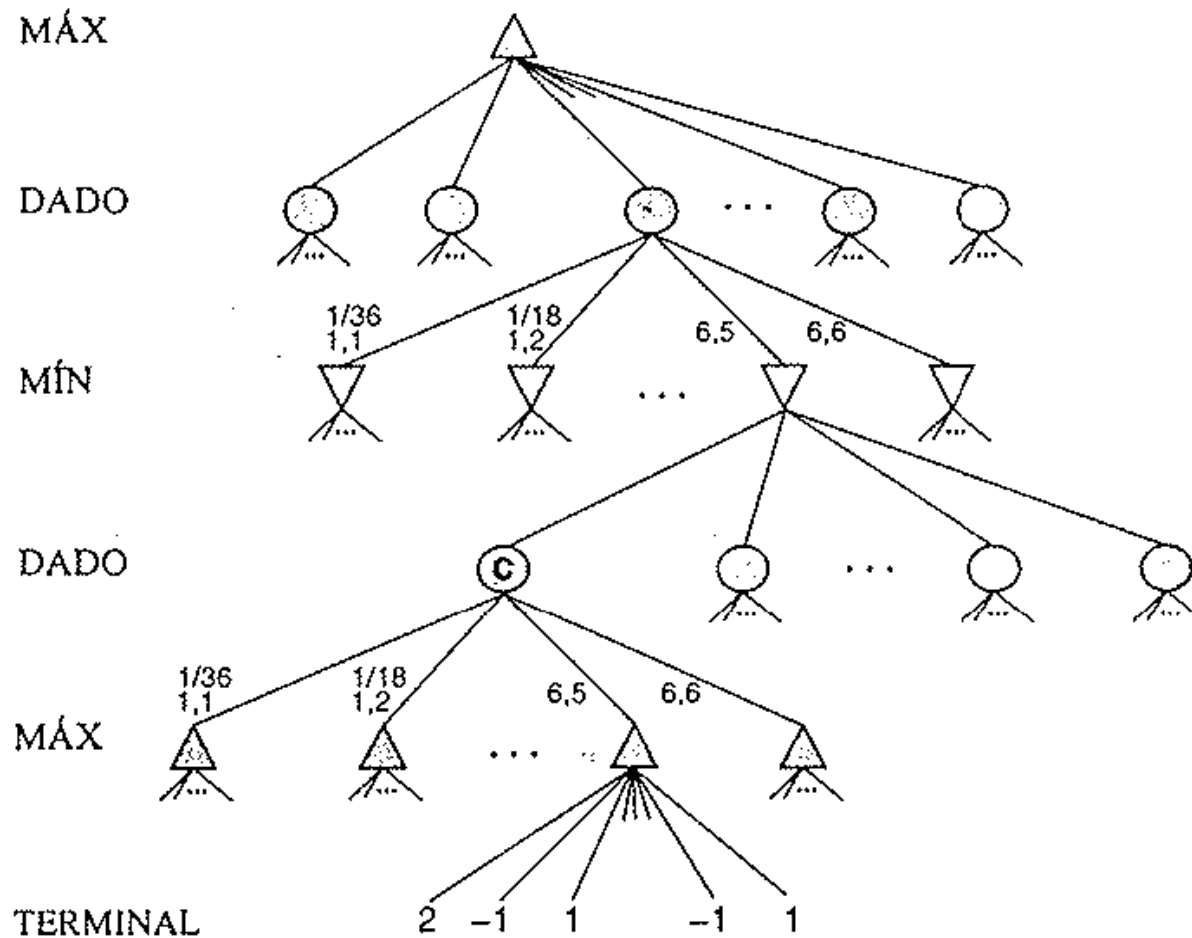
Juegos en los que interviene el azar

MAX y *MIN* van a realizar un movimiento, primero tienen que lanzar un dado. En este caso, podemos imaginar que cada vez que se lance el dado, un jugador ficticio, al que llamaremos *DADO*, realiza un movimiento, que, además, está determinado por el azar. En el caso concreto en el que se disponga de un solo dado, todos los posibles resultados son equiprobables, aunque la componente de azar puede estar sujeta a otras distribuciones de probabilidad.

Juegos en los que interviene el azar

Al igual que en los procedimientos que acabamos de analizar, en los árboles de búsqueda en juegos de azar también podemos aplicar el proceso de propagación de valores, teniendo en cuenta que cuando vayamos a propagar los valores de los nodos que representan un movimiento de azar, debemos propagar el valor *esperado* (media) de los valores de sus sucesores, en vez de un máximo o un mínimo¹. Los números que aparecen junto a los nodos de la Figura 12.8 repre-

Modelo



Algunos problemas

