

**Tema No. 2:** Métodos de solución de problemas.

**Actividad No. 29**

**Tipo de clase:** Conferencia.

**Título:** Búsqueda con adversario.

**Sumario:**

- Búsqueda con adversarios

**Objetivos:**

- Caracterizar los procesos inferenciales de búsquedas con adversario.

**Bibliografía:**

[1] Patrick Henry Winston, Inteligencia Artificial, 3ra edición, 1992.

[2] Elaine Rich & Kevin Knight, Inteligencia Artificial, 2da edición, 1994.

**Introducción:**

En esta conferencia usted conocerá la búsqueda minimax, y la poda alfa-beta, una idea que reduce en gran medida la búsqueda.

Aprenderá también como se pueden representar situaciones de juego en árboles y cómo es que tales árboles se pueden explorar de modo que se haga el movimiento más prometedor.

**Un poco de historia.**

Los juegos provocan una inexplicable fascinación y la idea de que las computadoras puedan jugar existe desde que existen las computadoras:

- Siglo XIX, Babbage, arquitecto de computadoras, pensó en programar su máquina analítica para que jugara al ajedrez.
- '50, Shannon describió los mecanismos que podían usarse en un programa jugara al ajedrez
- '50, Turing describió un programa para jugar al ajedrez pero no lo construyó.
- '60, Samuel construyó el primer programa de juegos importante y operativo, el cual jugaba a las damas y podía aprender de sus errores para mejorar su comportamiento

Los juegos proporcionan una tarea estructurada en la que es muy fácil medir el éxito o el fracaso. En comparación con otras aplicaciones de inteligencia artificial, por ejemplo comprensión del lenguaje, los juegos no necesitan grandes cantidades de conocimiento.

En un primer momento se pensó que se podrían resolver por búsqueda exhaustiva en el **árbol del juego**, es decir, un árbol que contenga todos los movimientos posibles de ambos jugadores. Considerando por ejemplo el juego de ajedrez, en una partida cada jugador realiza una media de 50 movimientos, con un factor de ramificación medio de 35 posibilidades, por lo tanto para examinar el árbol de juego completamente se tendrían que examinar 35100 posibilidades.

Resulta evidente que una simple búsqueda directa no es factible realizarla en la práctica, y por lo tanto es necesario algún tipo de procedimiento de búsqueda heurística.

Todos los procedimientos de búsqueda pueden verse como procedimientos de generación y prueba, en donde la comprobación se realiza después de distintas cantidades de trabajo del generador. En un extremo el generador proporciona una solución completa a evaluar por el comprobador; en el otro extremo, el generador genera movimientos individuales cada uno de los cuales se evalúa por el comprobador.

Para mejorar la efectividad de un programa resolutor de problemas, en particular de juegos, es importante:

- Mejorar el **procedimiento de generación** de manera de que se generen sólo movimientos

buenos.

- Mejorar el **procedimiento de prueba** para que sólo se reconozcan los mejores movimientos.

Dado que estos dos procedimientos no son perfectos, se incorpora la búsqueda o exploración de distintos niveles del juego para ver que puede ocurrir una serie de movimientos más adelante. Resulta evidente que la elección del camino a seguir será más acertada cuantas más capas se exploren antes de tomar la decisión. Se utiliza una función de evaluación estática (heurística) para elegir el movimiento más prometedor. Por ejemplo,

- Turing usó la sencilla función B/N (piezas blancas / piezas negras) para evaluar una posición dada de un tablero de ajedrez.
- Shannon utilizó una función lineal de funciones de evaluación simples para evaluar un tablero de damas:

$c1 * \text{ventaja piezas} + c2 * \text{avance} + c3 * \text{amenazas dobles} + \dots$

En la función anterior, con un mecanismo de aprendizaje, los pesos o ponderaciones se incrementan o disminuyen siempre que sus componentes sugieran movimientos que conducen a la victoria o al fracaso respectivamente. Este es un problema denominado de asignación de crédito, es decir, decidir de entre una serie de acciones cuál es la responsable del resultado (Aprendizaje por inducción).

En juegos y otros dominios la solución se encuentra mediante un proceso de búsqueda que debe combinarse de ser posible con una técnica directa. Por ejemplo, en el ajedrez, las aperturas y los finales están ya estudiados y se almacenan en una base de datos.

### Desarrollo:

Vamos a ocuparnos tan solo de juegos de dos jugadores, con información perfecta (es decir, en los que no interviene el azar y en los que cada jugador "ve" toda la información del otro) y de suma cero (sólo se puede GANAR, PERDER o EMPATAR). Ejemplos son las damas, el ajedrez, el tres en raya.

Y en donde estos juegos pueden ser representados a través de un árbol de estados, en los que cada estado representa un tablero diferente y en donde los lazos de unión entre estados del árbol indican posibles movimientos para pasar de un estado de tablero a otro estado. Lo anterior puede ser esquematizado con la ayuda del siguiente árbol.

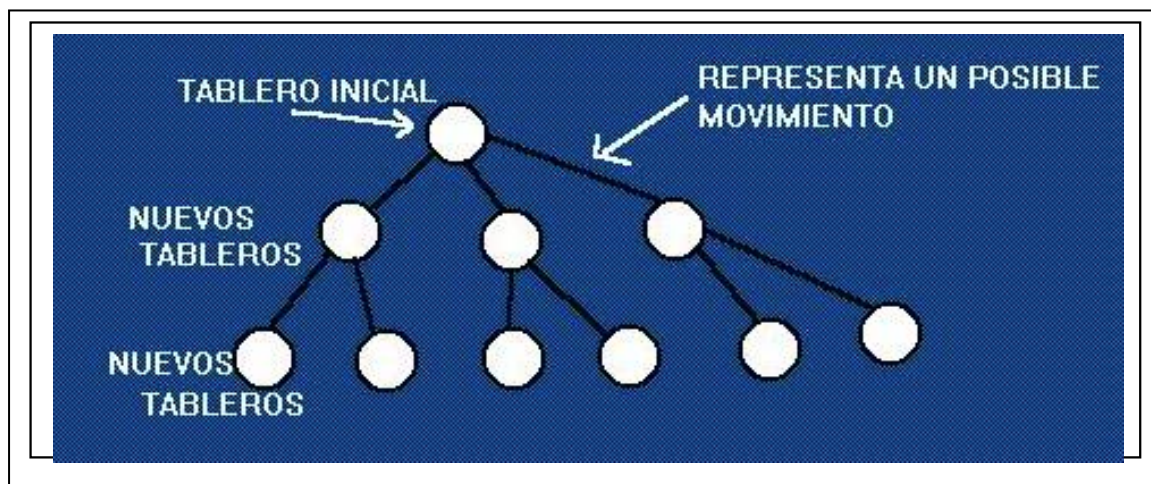


Figura 1. Representación en árbol de un juego de tablero

### **Procedimiento Minimax es un procedimiento de búsqueda hacia delante.**

Bajo la técnica minimax asociado con cada nodo del árbol de juego se tiene un valor que indica la calidad del tablero. (Ejemplo para el caso del ajedrez, podría ser el número de piezas ganadas hasta el momento).

En este método se supone que existe un cierto nivel del árbol para el cual se conoce el valor asociado con cada tablero de ese nivel y en base al cual y a la técnica minimax se descubre el valor de los tableros en los niveles superiores, de tal manera que se puede encontrar cuáles fueron los movimientos que hicieron cada uno de los contendientes.

El proceso de cálculo de un número que refleje la calidad del tablero se conoce como **evaluación estática**. El procedimiento que hace el cálculo es un **evaluador estático** y el número que calcula se conoce como **resultado de la evaluación estática**.

### **Bajo este método existen 2 jugadores**

- uno que es el maximizador, quien buscará siempre hacer un movimiento que le produzca el máximo beneficio.
- el otro que es el minimizador que busca hacer movimientos que genere valores mínimos.

Suponga que tenemos un analizador de situaciones que convierte todos los juicios acerca de las situaciones del tablero en un número simple de calidad global.

Suponga también que los números mayores, por convención, indican la ventaja de un jugador y los menores la ventaja del otro.

El jugador que espera obtener números mayores se conoce como jugador de maximización o maximizador. El otro jugador recibe el nombre de jugador de minimización o minimizador.

El maximizador busca un movimiento que le lleve a un número grande y supone que el minimizador intentará forzar el juego hacia situaciones con evaluaciones estáticas con números pequeños.

Las decisiones del maximizador deben tener conocimiento de las alternativas disponibles para el minimizador en el siguiente nivel inferior. De manera similar, las decisiones del minimizador deben tomar en cuenta las alternativas disponibles para el maximizador en el siguiente nivel.

Sin embargo, en un momento dado, se alcanza el límite de exploración y el evaluador estático proporciona una base directa para la selección de alternativas. En el siguiente ejemplo, las evaluaciones estáticas del fondo determinan que el minimizador puede escoger entre resultados efectivos de 2 y 7 nivel inmediato superior en la rama izquierda. Obviamente, el maximizador elige moverse hacia el nodo en el que el minimizador no pueda hacer otra cosa que mantenerse en su primer movimiento el resultado efectivo de 2. De nuevo, los resultados de un nivel determinan la acción y el resultado efectivo del nivel inmediato superior.

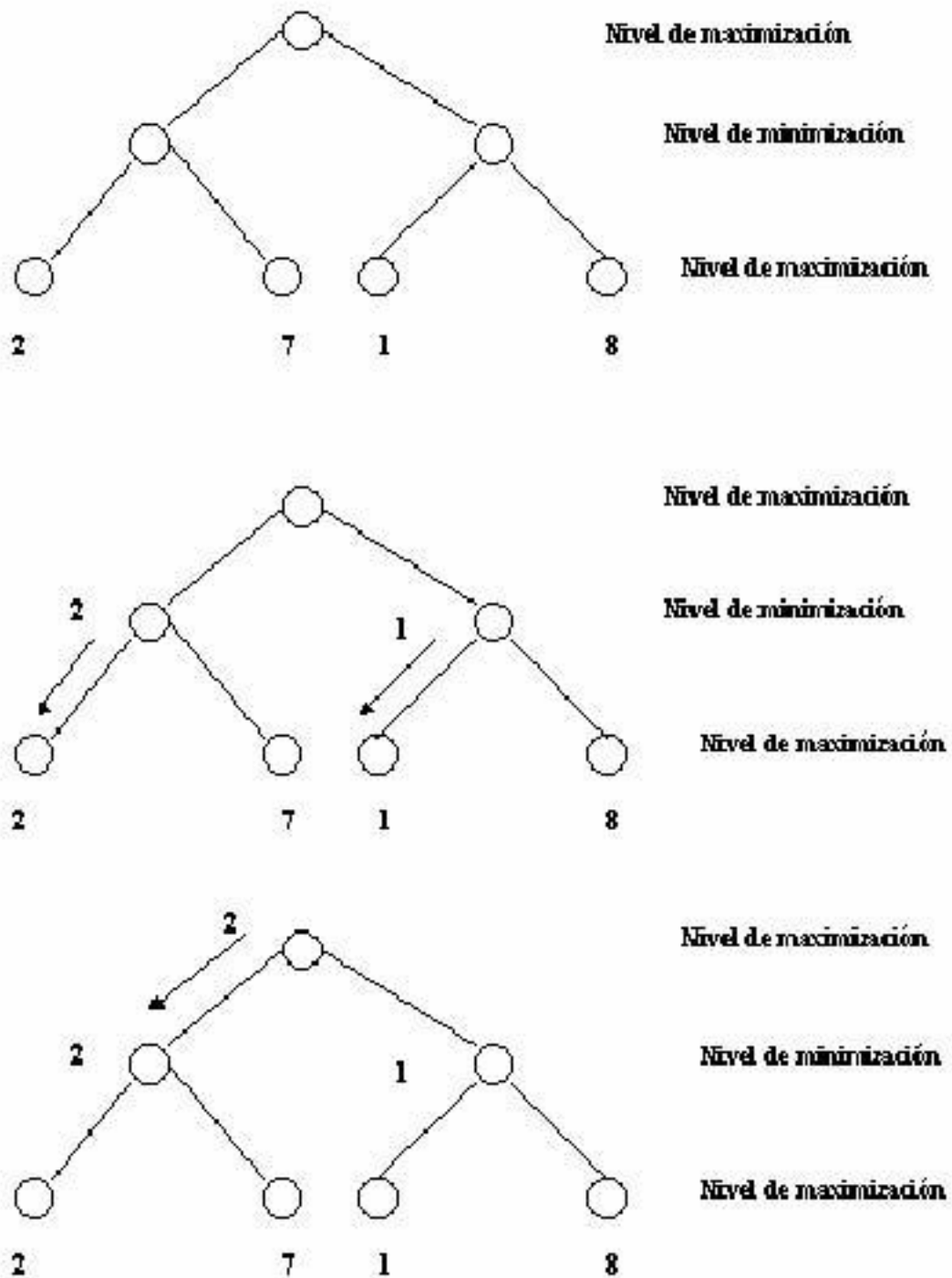


Figura 2 La búsqueda mínima es un método para determinar movimientos. Tal búsqueda emplea un evaluador estático para calcular números de especificación de ventaja para las situaciones del juego desarrollado parcialmente. Un jugador intenta obtener los números más altos, buscando ventaja, mientras que el oponente se esfuerza por obtener los más bajos.

El procedimiento mediante el cual la información acerca de los resultados pasa hacia arriba del árbol de juegos se conoce como procedimiento MINIMAX, debido a que el resultado en cada nodo es o bien el mínimo o el máximo de los resultados de los nodos que están inmediatamente abajo:

Para efectuar una búsqueda mínimas mediante MINIMAX,

- si el límite de búsqueda se ha alcanzado, calcule el valor estático de la posición actual en

relación con el jugador apropiado. Dé a conocer el resultado.

- De otro modo, si el nivel es de minimización, use MINIMAX en los hijos de la posición actual. Dé a conocer el mínimo de los resultados.
- De lo contrario, el nivel es de maximización. Use MINIMAX en los hijos de la posición actual. Notifique el máximo de los resultados.

Note que la idea completa del proceso mínimas se apoya en el hecho de traducir la calidad del tablero en un solo número de resumen. El valor estático. Por desgracia, un número constituye un resumen pobre.

Observe también que el proceso minimax puede ser caro, ya que la generación de trayectorias o la evaluación estática pueden requerir gran cantidad de cálculos. Lo que más cuesta depende de la forma en que el generador de jugadas y el evaluador estático hayan sido instrumentados.

Es importante notar que en este algoritmo la evaluación de una posición no está completa hasta que cada una de sus posiciones sucesoras en el espacio de búsqueda, hasta un cierto nivel, no ha sido evaluada. Se trata por tanto de una estrategia exhaustiva, aunque de profundidad limitada. Veremos en el último apartado que es posible "podar" el espacio de búsqueda sin perder información.

Al principio puede parecer que el evaluador estático debe utilizarse en cada nodo hoja de la base del árbol de juego. Pero, por fortuna no sucede así. Existe un procedimiento que reduce tanto el número de ramas que deben generarse como de evaluaciones estáticas que deben hacerse, acortando así el trabajo global que se debe realizar. Es algo similar a la idea de ramificación y poda en el sentido de que se demuestra que algunas trayectorias son malas aun cuando no se sigan hasta el límite previsto.

Considere la situación que se muestra en la figura 4, en la cual el evaluador estático ya se ha utilizado en las dos primeras situaciones de nodo hoja. El hecho de efectuar el procedimiento minimax en los resultados de 2 y 7 determina que el jugador de minimización tiene garantizado un resultado de 2 si el maximizador toma la rama izquierda del nodo de la cima. Este movimiento a su vez asegura que el maximizador tiene garantizado un resultado de por lo menos 2 en la cima. Esta garantía resulta evidente aun antes de realizar otras evaluaciones estáticas, ya que el maximizador puede elegir indudablemente la rama izquierda siempre que la derecha le conduzca a un resultado mas bajo. Esta situación se indica en el nodo superior de la parte media de la figura 4.

Observe con más detenimiento el árbol. ¿Tiene sentido continuar hasta la situación del tablero del nodo final? ¿Acaso tiene importancia el valor producido en este punto por el evaluador estático? Por extraño que parezca, la respuesta es no. Ya que indudablemente si el maximizador sabe que tiene garantizado un resultado de 2 en caso de seguir la rama izquierda, no necesita saber más acerca de la rama derecha aparte que en esta no puede obtener un resultado mayor que 1. El último nodo evaluado podría ser +100, -100 u otro número cualquiera, sin que ello afecte el resultado. Como se muestra en la parte inferior de la figura 4, resultado del maximizador.



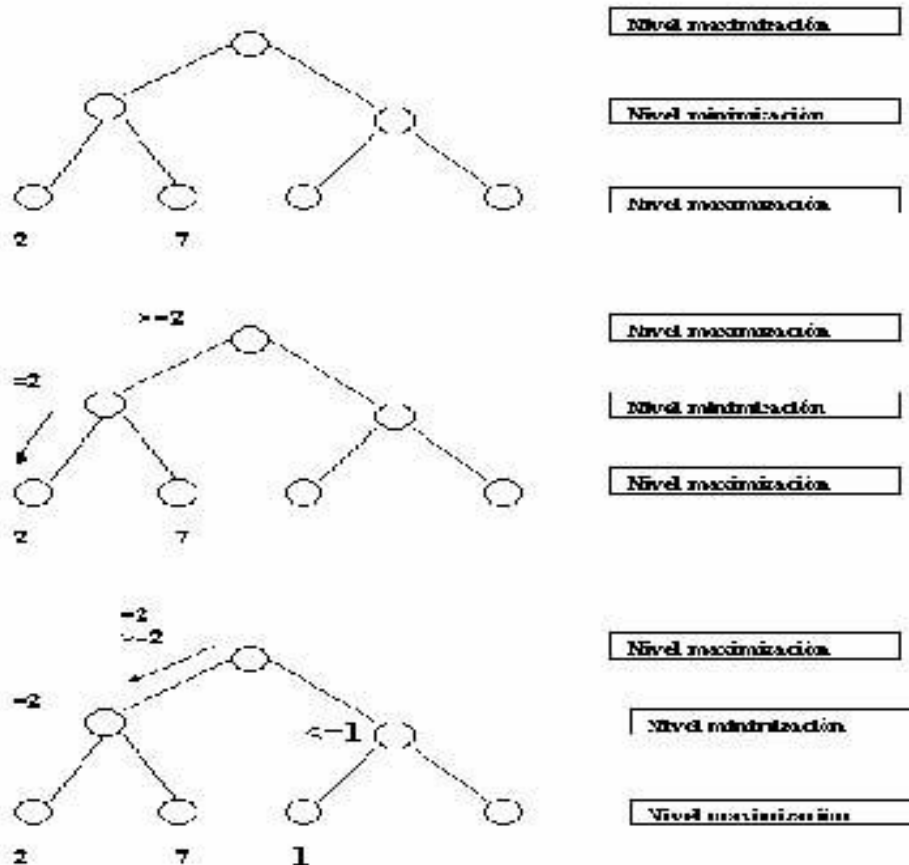


Figura 3. Procedimiento ALFA-BETA en funcionamiento. No hay necesidad de explorar por completo la parte derecha del árbol, ya que no hay forma de que el resultado pueda alterar la decisión acerca del movimiento. Una vez que se ha mostrado que el movimiento de la derecha es peor que el de la izquierda, no hay que ver en que proporción.

Si reflexionamos, resulta claro que si un oponente tiene una respuesta que establece que movimiento potencial es malo, no hay necesidad de revisar cualesquiera otras respuestas ante el movimiento potencial. De forma más general, usted tiene un ejemplo poderoso en la siguiente idea:

### El principio alfa-beta:

- *Si tiene una idea que indudablemente es mala, no se tome el tiempo para constatar que tan mala es.*

Esta idea se conoce como el principio alfa-beta porque, como se vera mas adelante, forma parte del procedimiento ALFA-BETA. Este procedimiento utiliza dos parámetros, tradicionalmente conocidos como alfa y beta, para seguir el rastro de las expectativas.

En el contexto especial de los juegos, el principio alfa-beta establece que, siempre que usted descubra valores parciales alfa/beta para un nodo determinado, deberá revisar los valores parciales alfa/beta del nodo antecesor o padre, pues puede ser que no resulte sensato continuar la búsqueda por la rama actual.

Con el principio alfa-beta traducido a instrucciones para tratar con cambios de resultados, se puede trabajar con un ejemplo más grande. Desgraciadamente, es un tanto difícil ver sobre papel como las evaluaciones estáticas se entremezclan con las conclusiones sobre los valores de nodo. Debemos proceder con números de suceso, encerrados en cajas junto a cada conclusión, que se muestran en el ejemplo de la figura 5, en el que vemos otro árbol estilizado con una profundidad de 3 y un factor de ramificación uniforme también de 3:

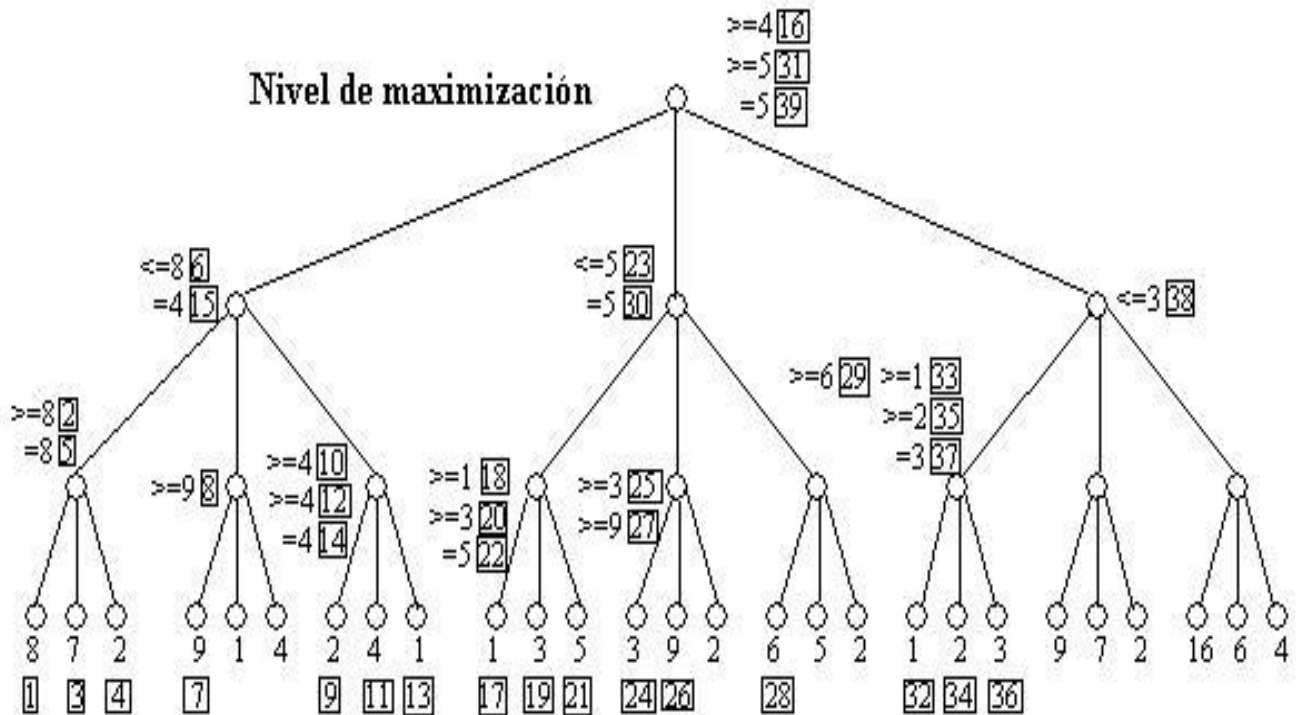


Figura 4. Árbol de juegos con profundidad de 3 y factor de ramificación de 3. Los números encerrados en cajas muestran el orden en que se determinan las conclusiones. Observe que sólo se realizaron 16 evaluaciones estáticas, en lugar de 27 que se requieren sin la poda alfa-beta. Es evidente que la mejor jugada para el maximizador es descender por la rama intermedia.

1-2 Avanzando hacia abajo por la rama izquierda en cada punto de decisión, la búsqueda penetra hasta la base, donde se encuentra un valor estático de 8. Este 8 claramente significa que el maximizador tiene garantizado un resultado de al menos 8 con las tres alternativas disponibles. Una nota a este respecto se coloca junto al paso 2

3-5 Para tener la certeza de que no se va a encontrar un resultado mayor que 8, el maximizador examina los otros dos movimientos que tiene disponibles. Debido a que 7 y 2 indican movimientos inferiores, el maximizador concluye que el resultado más alto que se puede lograr es exactamente 8 y que el movimiento correcto es el primero que se examinó.

6 Si se asegura el resultado del maximizador en el nodo más bajo, usted podrá llegar a una conclusión respecto a lo que el minimizador puede esperar el siguiente nivel superior. Dado que ahora se sabe que un movimiento conduce a una situación en la que el maximizador puede obtener un resultado de 8, usted está conciente de que el minimizador en el nivel inmediato superior puede lograr un resultado de 8 o menor.

7-8 Para ver si el minimizador tiene un mejor desempeño en el segundo nivel, usted debe examinar sus dos movimientos restantes. El primero conduce a una situación en la que el maximizador puede obtener al menos 9. Aquí se representa un corte. Al tomar la rama izquierda, el minimizador fuerza un resultado de 8; pero si toma la rama del centro, el minimizador permite un resultado que ciertamente no es menor que 9 y que será mayor si las otras alternativas del maximizador son mayores. En consecuencia, la rama intermedia es mala para el minimizador; no hay necesidad de continuar para saber cuán mala es y, por tanto, tampoco se requiere tener dos evaluaciones estáticas. No hay cambio de expectativas del peor de los casos del minimizador; aun es de 8.

9-14 El minimizador aun debe investigar su última opción, la de la derecha. Usted necesita ver lo que el maximizador puede hacer ahí. La siguiente serie de pasos oscila entre las evaluaciones estáticas y las conclusiones acerca de la situación del maximizador inmediatamente encima de ellos. La conclusión es que el resultado del maximizador es 4.

15 Al descubrir que la rama derecha conduce a un resultado forzado de 4, el minimizador tomaría la rama derecha, ya que 4 es menor que 8, el resultado bajo anterior.

16 Ahora se puede colocar una cota en el nivel superior. El maximizador. al examinar la situación en ese punto, ve que su rama izquierda conduce a un resultado de 4, de modo que ahora sabe que al menos obtendrá un resultado así, y tal vez mejor. Para ver si puede obtener uno mejor, debe examinar su rama intermedia y derecha.

17-22 Para decidir como reaccionara el minimizador al final de la rama central se requiere saber que pasa en la rama izquierda que desciende a partir de ese punto. Aquí, el maximizador entra en acción y descubre que la mejor jugada es hacia a una posición con un resultado de 5.

23 En tanto no se sabía nada definitivo acerca de lo que el maximizador podía hacer, no era posible colocar limites al potencial del minimizador. Sin embargo, el hecho de saber que el maximizador obtiene 5 a lo largo de la rama izquierda significa saber algo definitivo. La conclusión es que le minimizador puede obtener un resultado de por lo menos 5.

24-27 Al encontrar lo que el maximizador puede hacer debajo de la rama central del minimizador, usted descubre parcialmente, mediante el análisis, que el maximizador puede alcanzar un resultado de 9. Pero 9 es una alterativa pobre en relación con la opción conocida del minimizador, que asegura un 5. De nuevo se presenta un corte. . No tiene caso investigar la otra opción del maximizador y de esa forma usted evita una evaluación estática.

28-29 Al examinar rápidamente la rama derecha del minimizador se ve que este, también, da la oportunidad al maximizador de forzar la jugada hacia un resultado más alto que el que puede obtener el minimizador por la rama izquierda. Aquí un corte evita dos evaluaciones estáticas.

30 En vista de que ya no hay más ramas que investigar, el resultado de 5 del minimizador ha dejado ser un simple límite; ahora es el valor real que se puede lograr.

31 Desde la cima el maximizador ve una alternativa que puede conducir a un resultado más alto a través de la rama central; de modo que escoge tentativamente esa rama que puede obtener al menos 5.

32-37 Ahora se debe explorar la alternativa de la rama derecha del maximizador. al internarse en el árbol y deambular un poco, llega a la conclusión de que el minimizador ve una alternativa en la rama izquierda que le asegura un resultado de 3.

38 El minimizador puede concluir que el resultado de la rama izquierda es un límite inferior al resultado que puede obtener.

39 Al saber el minimizador puede forzar una jugada en la que el resultado es de 3, el maximizador en el nivel superior concluye que no tiene caso explorar mas la rama derecha. Después de todo, un resultado de 5 se obtiene después de un movimiento en la rama central. Note que esto permite ahorrar seis evaluaciones estáticas, así como dos generaciones más de movimiento.



No es raro que uno se pierda en esta clase de demostración. Incluso los especialistas experimentados en juegos encuentra algo de mágico en el procedimiento alfa-beta. Cada conclusión individual parece correcta, pero de algún modo el resultado global es extraño y difícil de creer.

Observe que, en el ejemplo, en ningún momento tuvo que examinar más de un nivel hacia arriba para decidir si se detenía o no la exploración. En árboles más profundos, con cuatro o más niveles, se pueden presentar los llamados cortes profundos, que obligarían a una búsqueda más prolongada.

Una forma de estar al tanto de toda la contabilidad es utilizar un procedimiento con parámetros, alfa y beta, que registren todas las observaciones necesarias. El procedimiento ALFA-BETA se inicia en el nodo raíz con un valor alfa de  $-\infty$  y un valor beta de  $+\infty$ ; ALFA-BETA entonces se llama a sí mismo de manera recursiva con un intervalo que se va estrechando entre los valores alfa y beta:

**Para efectuar una búsqueda minimax con el procedimiento ALFA-BETA,**

- Si el nivel es superior sea alfa  $-\infty$  y beta infinito;
- Si se ha alcanzado el límite de búsqueda, calcule el valor estático de la posición actual en relación con el jugador apropiado. Notifique el resultado;
- Si el nivel es de minimización,
  1. hasta que todos los hijos sean examinados con ALFA-BETA o hasta que alfa sea mayor o igual que beta,
  - 1.1 use el procedimiento ALFA-BETA, con los valores actuales de alfa y beta, en un hijo; apunte el valor notificado.
  - 1.2 Compare el valor notificado con el valor beta; si el valor informado es menor, ajuste beta al nuevo valor
  2. Notifique el valor beta.
- De otro modo, el nivel es de maximización:
  1. hasta que todos los hijos sean examinados con ALFA-BETA o alfa sea igual o mayor que beta,
  - 1.1 use el procedimiento ALFA-BETA, con los valores actuales de alfa y beta, en un hijo; apunte el valor notificado.
  - 1.2 Compare el valor notificado con el valor alfa; si el valor es mayor, ajuste alfa al nuevo valor.
  2. notifique el valor alfa.

A continuación un ejemplo donde se puede ver la corrida anterior.

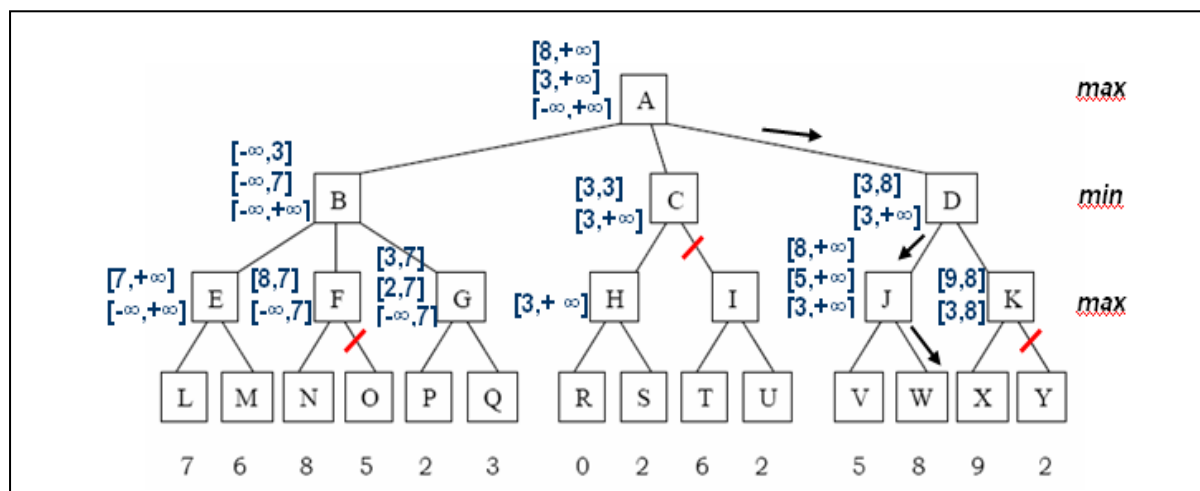


Figura 6. Ejemplo de poda alfa-beta

Esta reducción en el esfuerzo de la búsqueda fue alcanzada manteniendo las pistas de las cotas sobre los valores propagados. En general, como los sucesores de un nodo van produciendo nuevos valores, las cotas pueden ser modificadas, pero:

- los valores *ALFA* de los nodos *MAX* no pueden decrecer, y
- los valores *BETA* de los nodos *MIN* no pueden crecer.

Debido a estas restricciones podemos enunciar las reglas siguientes para podar la búsqueda:

- a) Puede podarse debajo de cualquier nodo *MIN* que tenga un valor *BETA* menor o igual al valor *ALFA* de sus nodos *MAX* padres. El valor final propagado de este nodo *MIN* será entonces el valor *BETA*. Este valor puede no ser el mismo que el obtenido por una búsqueda *MINIMAX* completa, pero su uso trae por consecuencia que se seleccione el mismo mejor movimiento.
- b) Puede podarse debajo de cualquier nodo *MAX* que tenga un valor *ALFA* mayor o igual al valor *BETA* de cualquiera de sus nodos *MIN* padres. El valor final propagado de este nodo *MAX* será entonces el valor *ALFA*. Cuando la búsqueda se interrumpe por la regla a) se dice que ocurrió un corte alfa. Si, por el contrario, fue por la regla b) se dice que ocurrió un corte beta.

Empleando el procedimiento alfa-beta siempre se realiza un movimiento que es tan bueno como podría haber sido encontrado por el procedimiento *MINIMAX*, buscando a la misma profundidad pero realizando mucha menos búsqueda. Para ejecutar una poda *ALFA-BETA*, al menos alguna parte del árbol de búsqueda tiene que ser generado a la profundidad máxima, porque los valores *ALFA* y *BETA* se basan en los valores estáticos de los nodos terminales. Por eso, algún tipo de búsqueda primero en profundidad se emplea usualmente al utilizar este procedimiento.

### Conclusiones:

Los juegos siempre han fascinado a las personas, y la idea de poder jugar con una computadora ha existido desde que existen las computadoras.

Los algoritmos de juegos pudieran ser muy eficientes en la implementación de algoritmos de búsqueda.

Estos algoritmos no solo tienen aplicación en la computación sino también, en problemas de la vida práctica.