

UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE CIENCIAS EXACTAS Y NATURALES

DEPARTAMENTO DE COMPUTACIÓN

ALGORITMOS Y ESTRUCTURAS DE DATOS III

Trabajo Práctico 3

Autores:

Nicolás Chehebar, mail: *nicocheh@hotmail.com*, LU: 308/16

Matías Duran, mail: *mato_fede@live.com.ar*, LU: 400/16

Lucas Somacal, mail: *lsomacal@gmail.com*, LU: 249/16

Índice

| | |
|--|----------|
| 1. El Juego | 2 |
| 1.1. Descripción | 2 |
| 1.1.1. Ejemplos | 2 |
| 2. Jugador Óptimo | 2 |
| 2.1. El algoritmo | 2 |
| 2.1.1. El Pseudocódigo | 3 |
| 2.2. La poda alfa-beta | 3 |
| 2.2.1. El algoritmo | 3 |
| 2.2.2. El Pseudocódigo | 3 |
| 2.3. Complejidad | 3 |
| 2.4. Experimentación | 3 |
| 2.4.1. Contexto | 3 |
| 2.4.2. Sin poda | 3 |
| 2.4.3. Con poda | 3 |
| 3. Análisis comparativo con paper | 3 |

1. El Juego

1.1. Descripción

El juego es una generalización del popular 4 en línea ¹. Consiste en una grilla de M filas y N columnas en la cual dos jugadores colocan alternadamente una ficha propia (identificada con rojo las de un jugador y azul las del otro). Las fichas se pueden colocar en cualquier columna de la grilla y una vez elegida la columna, esta determina el movimiento, ya que irá a la fila de "más abajo" (la de numeración más baja) que esté desocupada. El objetivo de un jugador será lograr tener una línea recta (diagonal, vertical u horizontal) de C fichas propias. Cuando esto suceda, el jugador ganará el partido. Además, cada jugador dispone de P fichas. En caso de que ambos se queden sin fichas y ninguno haya ganado, la partida finaliza en empate. También se da un empate si la grilla queda llena (y ninguno había ganado). Se trata de una generalización del 4 en línea ya que si tomamos parametros $M = 6, N = 7, C = 4, P = 21$ se replicarían las condiciones iniciales del juego.

1.1.1. Ejemplos

2. Jugador Óptimo

2.1. El algoritmo

El algoritmo del jugador del punto 1.a brinda un jugador óptimo. Nos asegura que este jugador hará la estrategia ganadora si hubiera. En caso de que esta no exista, realizará una de empate. Y si tampoco existiera esa, jugará indistintamente sabiendo que perderá. Este análisis se realiza jugada a jugada. Para ejemplificar esto, podría suceder que el oponente tenga la estrategia ganadora y en ese caso nuestro jugador hará cualquier movida indistintamente (pues sabe que perderá), pero si en la próxima jugada el oponente no realiza la correspondiente a su estrategia ganadora y da un nuevo estado del tablero en el que esta vez nuestro jugador tiene estrategia ganadora, nuestro jugador jugará y ganará ya que ahora sí tiene estrategia ganadora.

Para lograr esto, utilizamos una técnica algorítmica similar al Backtracking en el sentido de que exploremos todas las soluciones posibles y nos quedamos con la óptima. Pero esta vez tenemos dos jugadores interviniendo en la situación donde lo que uno busca es todo lo contrario a lo que busca el otro. Podemos decir que un tablero finalizado tiene 3 puntajes posibles, 1 si ganamos nosotros, 0 si es empate, -1 si ganó el otro (podríamos sino generalizarlo para todo tablero y que haya un cuarto valor que sea inválido si aún no hemos calculado el valor de dicho tablero). Así, lo que sabemos es que turno a turno, uno quiere maximizar el puntaje y el otro minimizarlo. Es por esto que dicha técnica algorítmica se llama Minimax.

De esta manera, igual que en Backtracking tenemos un árbol de ejecución donde cada nodo es un estado del tablero y la raíz es el tablero vacío. Cada nodo (que no sea hoja) tendrá N hijos donde cada uno representará que la próxima jugada fue en alguna de las N columnas. De esta forma, recorreremos todos los tableros posibles. Según quién comience, en el primer nivel trataremos de maximizar o minimizar, en el siguiente lo contrario y así sucesivamente. Todos los niveles impares minimizarán si empieza el contrincante y maximizarán si empieza nuestro jugador. El que maximiza le asignará a su nodo un puntaje que será el máximo de los puntajes de todos sus hijos. Análogamente el que minimiza le asignará a su nodo un puntaje que será el mínimo de los puntajes de todos sus hijos.

Así, ejecutando dicho algoritmo la raíz tendrá la información de quién tiene la estrategia ganadora, o que ambos pueden asegurar el empate según quién empiece y haya un 1, 0 o -1. Esta es la idea general del algoritmo, lo veremos más claro en el siguiente pseudocódigo

¹<https://es.wikipedia.org/wiki/Conecta4>

- 2.1.1. El Pseudocódigo
- 2.2. La poda alfa-beta
 - 2.2.1. El algoritmo
 - 2.2.2. El Pseudocódigo
- 2.3. Complejidad
- 2.4. Experimentación
 - 2.4.1. Contexto
 - 2.4.2. Sin poda
 - 2.4.3. Con poda
- 3. Análisis comparativo con paper