

Práctica Final: CONECTA 4

Alberto Jesús Durán López
Juan Jesús González Mairena
Ana Peña Arnedo

Índice

1. Introducción
2. Objetivos
3. Problema
4. Solución propuesta
 - a. Creación de árboles
 - b. Recorrido de árboles
 - c. Análisis de los tableros

1.Introducción

El objetivo principal de esta práctica es resolver el problema propuesto eligiendo la mejor estructura de datos para las operaciones solicitadas. Con este documento presentamos una memoria de la solución que proponemos a dicho problema, en el que detallamos los pasos seguidos para resolverlo centrándonos en la métrica propuesta.

2.Objetivos

El objetivo de esta práctica final es llevar a cabo el análisis, diseño e implementación del proyecto. Con este fin se espera que abordemos el problema propuesto donde se requieren estructuras de datos que puedan almacenar grandes volúmenes de datos a los que se pueda acceder de la manera más eficiente.

3.Problema

Se nos pide implementar un programa que simule el juego “Conecta 4”. El objetivo de este juego es alinear cuatro fichas en un tablero compuesto por seis filas y siete columnas. Cada jugador dispone de 21 fichas de un color, lo que en nuestro caso sustituiremos por los caracteres “X” y “O”. Por turnos, los jugadores deben introducir una ficha en la columna que deseen, contando con que ésta no esté completa, y dicha ficha caerá en la posición más baja que quede libre de esa columna. Se proclamará ganador de la partida aquel jugador que consiga alinear cuatro fichas consecutivas de un mismo color en horizontal, vertical o diagonal. Si todas las columnas están llenas y nadie ha conseguido una fila válida, se considera un empate entre ambos jugadores.

Para esta práctica debemos implementar una versión de este juego en la que los movimientos de uno de los jugadores se efectúen de forma automática. Esta modalidad, llamada “Player vs. Computer”, requiere implementar un mecanismo de decisión automática basado en la disposición actual del tablero. Para llevar a cabo este mecanismo de decisión automática, hemos representado todos los posibles movimientos a partir del tablero actual y hemos definido una “métrica” que nos permite evaluar cuál de los movimientos posibles deriva un tablero con más probabilidad de éxito para el jugador automático.

4.Solución propuesta

Cronstrucción de árboles

Para la construcción de árboles hemos implementado dos funciones, “ConstruirArbol” y “Construir”. La diferencia principal entre ellas es que “ConstruirArbol” pasa el tablero sin referencia y “Construir” lo pasa por referencia. Esto se debe a que la función recursiva “Construir” va modificando el tablero cambiando el turno según cada nivel de profundidad. En “ConstruirArbol” no lo pasamos por referencia para evitar que “Construir” altere los turnos de forma errónea al crear el árbol.

La creación del árbol se lleva a cabo en la función recursiva “Construir”. Dentro de esta función encontramos un bucle “for” que establece el número de hijos que va a tener cada nodo dependiendo del número de columnas. A su vez, dentro de este “for” tenemos una condición que comprueba que la columna escogida no está completa. Si está completa, no crea ese hijo. Si por el contrario no está completa, crea el hijo y vuelve a llamar dentro de éste a “Construir” que crea los correspondientes hijos. Esta función recursiva termina cuando la variable “nivel” es igual a 0.

Recorrido

Para el recorrido del árbol hemos implementado las funciones “AnalisisArbol” y “AnalisisHijos”. La función “AnalisisArbol” la usamos para decidir el siguiente movimiento a realizar que nos proporcione mayor probabilidad de éxito dependiendo de los posibles tableros. En la función recursiva “AnalisisHijos” encontramos un bucle “for” que va recorriendo todos los hijos del nodo padre para analizarlos haciendo uso de la función “AnalisisTablero”. La función “AnalisisHijos”, dependiendo del turno, dará un valor mayor o menor al hijo, es decir, si el turno es del oponente le asigna un valor negativo al hijo, y si es turno del jugador automático le asigna un valor positivo. Después de analizar el hijo, la función vuelve a llamar a “AnalisisHijos” recursivamente para analizar los hijos de los hijos, y así hasta que el puntero “nodo” apunte a 0 o “NULL”.

Análisis de tableros

Para analizar los tableros implementamos la función "AnálisisTablero". Esta función otorga un valor de bondad a un nodo, que representa a un tablero que se le ha pasado como parámetro. En esta función encontramos llamadas a otras cuatro funciones, "Columna", "Fila", "Diagonal1" y "Diagonal2". Estas funciones recorren el tablero por columnas, filas, diagonal desde arriba a la derecha o diagonal desde arriba a la izquierda respectivamente. La métrica que hemos utilizado en estas funciones consiste en un contador que comprueba cuántas fichas del mismo tipo están juntas clasificándolas en grupos de 2, 3 y 4 fichas. Esta métrica establece un valor por cada grupo de fichas que se encuentre en el tablero que estamos analizando, haciendo que el grupo de 2 fichas tenga menor valor que el grupo de 3, y a su vez el de 3 menor valor que el de 4 fichas. A continuación sumamos todos los resultados de estos grupos y el total será el valor del tablero.