

Maze Reduction

Guion: Reducción de laberintos

Vicente Ferrari

Dpto. de Cs. de la computación e Informática
Universidad de la Frontera
Temuco, Chile
v.ferrari01@ufromail.cl

Sebastián Opazo

Dpto. de Cs. de la computación e Informática
Universidad de la Frontera
Temuco, Chile
s.opazo02@ufromail.cl

I. INTRODUCCIÓN

Primero decimos nuestros nombres y hablamos sobre el problema que elegimos. Personalmente elegí este problema porque me pareció bueno al leerlo, supe al tiro que iba a ser sobre grafos y arboles

II. CONCEPTOS BÁSICOS

A. En qué consiste el problema

La premisa del problema es la siguiente: Jay es dueño de un parque de atracciones y la atracción más popular es un laberinto. Los visitantes comienzan el laberinto en una pieza aleatoria y deben navegar por el laberinto con el objetivo de completar un mapa completo de este. Algo de lo que se percató Jay es qué podría en teoría reducir el laberinto sin que los visitantes se den cuenta. Si es que hay piezas que son efectivamente idénticas se podrían eliminar las que sobran.

B. ¿Qué es un nodo?

Desde ahora en adelante llamaremos a las piezas "nodos" (FOTO). Estos nodos son representados en un diagrama como un círculo con un número adentro para identificarlos. Los jugadores pueden entrar y salir de los nodos libremente pero no pueden dejar nada dentro de estos para identificarlos más tarde. Para los jugadores todos los nodos son exactamente iguales excepto en el número de pasillos que salen de este, hacia donde van dichos pasillos y el orden en el que están, en dirección horaria.

C. ¿Qué es un pasillo?

Los pasillos son la estructura que conecta un nodo a otro. Estos son todos indistinguibles de uno a otro. Son representados en el diagrama (FOTO) como líneas pero en la realidad pueden ser de cualquier forma, mientras sean indistinguibles desde el punto de vista de una persona. Los pasillos que salen de un nodo están dados por las entradas al programa y estos se ordenan en dirección horaria (FOTO).

III. OTROS CONCEPTOS

A. Efectivamente idénticos

Dos nodos son efectivamente idénticos cuando y solo cuando un jugador puede comenzar el laberinto en A o B y no puede distinguir en que parte del laberinto comenzó incluso teniendo el mapa de este de antemano.

IV. ENTRADAS/SALIDAS

A. Posibles entradas y salidas 1

Este es un ejemplo de una entrada al programa y una correspondiente salida. Pusimos primero el ejemplo más aburrido pero lo vamos a volver a ver después. Como pueden ver, nos dan una lista (APUNTAR AL CUADRADO ROJO) de la que sacamos toda la información. El número del nodo está implícito por el número de línea. La primera línea es cuantos nodos hay en el laberinto, y luego van todos los nodos, el primer número en cada nodo nos dice cuantos pasillos tiene el nodo y luego enumera los nodos a los que van cada pasillo, siempre en orden horario. Es interesante notar que por la naturaleza de un círculo, podemos permutar ciclicamente los nodos de un pasillo sin perder información.

B. Posibles entradas y salidas 2

Este ejemplo se pone un poco más interesante. Lo primero que hay que notar, cosa que pasó igual en el ejemplo anterior pero no importa. Lo primero que hay que darse cuenta es que los laberintos pueden tener componentes conexas como se muestra en la pantalla (MOSTRAR COMPONENTES). La componente que es completamente circular tiene una propiedad interesante. Pero ahora ya tenemos una salida. Como pueden ver si aparecemos en los nodos 1 o 2 no tenemos como distinguir entre ellos (APUNTAR A LOS NODOS). Y como les decíamos hay un conjunto máximo de nodos efectivamente idénticos, el 5, 6 y 7 donde no importa en cual empezamos los 3 son indistinguibles de los otros.

C. Posibles entradas y salidas 3

El ejemplo 3 ya se pone un poco más complejo. Hay 13 nodos, 3 componentes conexas y 3 conjuntos maximales en la salida. Como pueden ver, una propiedad entretenida de las componentes circulares es que son todas indistinguibles entre sí. Es decir, si comienzo el laberinto en el nodo 8, no tengo que saber que no estoy en el 7, en el 9, 10, 11, 12 o 13. Por eso hacen un conjunto maximal de nodos efectivamente idénticos, o conjunto maximal de soluciones.

D. Posibles entradas y salidas 4

Y ahora volvemos al primer ejemplo donde queremos explicar algo entretenido. ¿Cuántos piensan que el 3 y el 4

deberían ser idénticos? (PREGUNTAR AL PÚBLICO) Bueno la verdad es que es bastante simple. Si comienzo del nodo 3 y me voy al 1, voy a tener una rama de profundidad 2 a mi derecha y una rama de profundidad 1 a mi izquierda. Mientras que si comienzo en el 4 y me voy al 1 es al revés, voy a tener una rama de profundidad 1 a mi derecha y una de 2 a mi izquierda. Es así como, teniendo el mapa completo, se podría distinguir fácilmente entre el nodo 3 y 4.

V. POSIBLE SOLUCIÓN

A. ¿Qué es un árbol?

El concepto de "árbol" está presente en varios "campos" (JAJAJA), en una "rama" (AJAJAJA) de las matemáticas llamada teoría de grafos y también en las ciencias de la computación como una estructura de datos fundamental. Pero claramente no tenemos el tiempo así que les daremos un resumen. Un árbol es básicamente lo que hay en pantalla (APUNTAR) comienza de una raíz y se va expandiendo por ramas. Hay árboles binarios donde cada nodo tiene solamente dos hijos o árboles abstractos de sintaxis que se ocupan para hacer análisis gramatical en los lenguajes de programación, et cetera. Esta pequeña pero poderosa estructura nos permitirá pensar en alguna posible solución al problema.

B. Ejemplo

Aquí podemos ver unos de los ejemplos anteriores y algunos árboles de los nodos de este. El primer problema con el que nos encontramos es repetición. Si yo comienzo en el nodo 1 y me muevo al 2, del nodo 2 me puedo volver a mover al 1 y después de nuevo al 2 y así. O puedo moverme del 1 al 2 al 3 al 1, de nuevo al 2, al 3, al 1, et cetera. Esto se ve un poco en el primer árbol (APUNTAR). Para los siguientes dos que tenemos en pantalla ocupamos una restricción donde cada nodo puede estar solamente una vez en cada árbol. Y formamos 2 de estos árboles, como pueden ver formar estos árboles nos permite caracterizar cada nodo con una estructura que luego podemos comparar en algún algoritmo. Si empezamos en el nodo 1 nos podemos mover al 2 y al 3, la rama del 2 termina porque el 3 ya apareció y recuerden que los árboles se van llenando hacia abajo y de izquierda a derecha, por lo que la última posibilidad es moverse del 3 al 4. Si luego hacemos lo mismo empezando del nodo 2 podemos comparar los árboles del 1 y del 2. La única diferencia entre estos es la permutación del 1 y del 2 porque claro, si empezamos del 1 este va a ser la raíz y lo mismo con el 2. Si el árbol fuese más grande lo podríamos ver mejor, pero, si ignoramos esta permutación inicial los árboles son idénticos. Creemos que esta es la clave del problema. Y claramente según la salida el 1 y 2 son un conjunto maximal de nodos efectivamente idénticos.

VI. PLANIFICACIÓN

A.

Aquí tenemos la planificación del proyecto, ya terminamos gran parte del diseño preliminar y la preparación de la

presentación y claramente la presentación estará lista momentáneamente. Luego de esto vamos a tomar feedback de los interesados (compañeros y profesor) para ver si nos faltó algo o que podemos mejorar.

B.

Luego del Feedback vamos a darnos tiempo para programar y luego preparar la siguiente presentación. El resto de la planificación se terminará durante la preparación de la segunda presentación.

VII. RIESGOS

Algunos de los posibles riesgos son que entreguemos el proyecto después del tiempo acordado, algo que no es muy raro en la industria (JAJAJA). Otro riesgo importante de nuestra parte sería dedicarle mucho tiempo al proyecto y dejar de lado otras responsabilidades.

VIII. CONCLUSIÓN

Y con esto terminamos la presentación. ¿Alguna pregunta? (AL PÚBLICO)