



Instituto Tecnológico de Costa Rica

Unidad Desconcentrada de Computación

Segundo proyecto Lenguajes de Programación

Estudiante:

Bryam Steven López Miranda

Profesor:

Oscar Víquez Acuña.

Campus Tecnológico Local San Carlos

27/04/2022

Índice

1. Introducción	3
2. Descripción del problema	3
3. Solución del problema	4
3.1. El sistema	4
3.1.1. Declaración y llenado de la matriz	4
3.1.2. Selección casillas y e impresión de resultados	5
3.1.3. Tamaño del grupo al que pertenece una casilla	5
3.1.4. Grupos distintos y tamaños	5
3.1.5. Listar todos los grupos y coordenadas	5
4. Análisis de resultados	6
5. Conclusiones y recomendaciones	7
Referencias	7

1. Introducción

La formación técnica de los ingenieros en programación consta del estudio del desarrollo de software y manejo de hardware. Parte de la obtención de este conocimiento es la indagación de paradigmas sistematizados que son fundamentales como principios para tener un entendimiento amplio en distintos temas como la Programación Lógica por ejemplo.

Según Serra (1994) la programación Lógica utiliza el formalismo de la lógica de primer orden para representar el conocimiento sobre un problema y para hacer preguntas que, si se demuestra que se pueden deducir a partir del conocimiento dado en forma de axiomas y de las reglas de deducción estipuladas, se vuelven teoremas. Así se encuentran soluciones a problemas formulados como preguntas. Con base en la información expresada dentro de la lógica de primer orden, se formulan las preguntas sobre el dominio del problema y el intérprete del lenguaje lógico trata de encontrar la respuesta automáticamente.

Por tanto, el entender y conocer la programación Lógica funge como gran herramienta para la resolución de problemas de forma simple comparado a otros paradigmas donde se requeriría un mayor desarrollo del problema.

El proyecto consiste en realizar un programa que sea capaz de realizar consultas sobre una matriz $N \times N$ que contiene en el 30 % de sus celdas círculos que representa que ahí existe un dato.

2. Descripción del problema

El presente documento se basa en la creación de un proyecto programado en lenguaje Lógico PROLOG y el lenguaje orientado a objetos e imperativo Java, pues estos lenguajes son fácilmente conectables. De manera que permita implementar a través de Java la interfaz gráfica de la matriz y por medio de PROLOG las consultas a la base de conocimiento recibida a través de Java, la realización de los objetivos planteados. Estos requisitos tienen como cimientos la implementación del paradigma lógico, para que así, el logro de los objetivos tanto de los requerimientos como de los objetivos de aprendizaje del estudiante autor del proyecto sean cumplidos a cabalidad. El cuadro 1 presenta los objetivos o requisitos base del proyecto.

Cuadro 1:Requerimientos del sistema

Requerimientos
1. Declarar una matriz de tamaño variable dentro de los rangos que defina el programador.
2. Llenar la matriz de forma manual de la forma que el usuario lo defina o de forma aleatoria con no más del 30 % de los posibles puntos llenos.
4. Las selecciones de casillas cuando sea necesario deben hacerse sobre la interfaz y no solicitándolo por teclado.
5. Las impresiones de resultados deben hacerse gráficamente en la matriz, procurando el buen entendimiento de los resultados. El uso de colores por ejemplo podría ayudar a discriminar resultados.
6. Calcule el tamaño del grupo al que pertenece una casilla que se encuentre ocupada cualquiera
7. Calcule el número total de grupos distintos y sus tamaños ordenados por cantidad de elementos
8. Liste todos los grupos (de manera que se muestren tanto la cantidad de elementos como las coordenadas de cada uno de los elementos).

Cabe aclarar que los objetivos mencionados con anterioridad son la base o el punto de partida, pero cada uno de ellos involucra una serie de funciones o métodos que deben llevarse a cabo para que el sistema cumpla su objetivo final.

3. Solución del problema

3.1. El sistema

3.1.1. Declaración y llenado de la matriz

Para la declaración de la matriz se implemento por medio de ciclos for la creación de cada casilla, la casilla con la que se conecta además de si en la casilla hay o no un punto representado por un 1 u 0 respectivamente, para llenar con puntos la matriz se trabajó con una lista donde, se guarda las posiciones de la matriz donde hay puntos, utilizando una regla de 3 respecto al tamaño de la matriz para cerciorarse que no se sobrepasen 30 % de las casillas con puntos,

para enviar la matriz a PROLOG se utilizó la clase Query utilizando hechos dinámicos, en Java la matriz se representa en forma gráfica con la clase JTable de la librería javax.swing.

3.1.2. Selección casillas y e impresión de resultados

La selección de casillas se trabajó con botones donde debe ir cada punto en las coordenadas según la celda que corresponda, esto para facilitar saber cuál celda fue seleccionada obteniendo el eje x e y, la impresión de resultados no se realizó graficamente, se realizó por medio de un JTextArea de la librería javax.swing.

3.1.3. Tamaño del grupo al que pertenece una casilla

Para determinar el tamaño del grupo a que pertenece una casilla se aprovecha el backtracking de PROLOG donde se hace un recorrido de la base de datos, se utiliza un contador dinámico, si se determina que una casilla tiene punto se suma 1 al contador y se pasa a la casilla siguiente para determinar si la misma tiene o no punto, sino tiene se realiza backtracking para buscar otra casilla compañera y seguir buscando, se finaliza cuando no hay más rutas donde seguir, osea se terminan las casillas juntas con puntos.

3.1.4. Grupos distintos y tamaños

Para realizar esta consulta se usa la anterior, se recorre toda la matriz en busca de grupos, se lleva el contador de grupos encontrados, una lista con tamaños y el contador para el tamaño del grupo que se calcula, cuando se termina de identificar un grupo se verifica si el tamaño está en la matriz, si es así se descarta el grupo, de lo contrario de agrega el nuevo tamaño a la lista de tamaños y se suma 1 al contador de grupos encontrado, luego se usa una función para ordenar la lista de tamaños de menos a mayor.

3.1.5. Listar todos los grupos y coordenadas

Esta consulta se trabaja de forma similar a la anterior, pero se guarda la lista de coordenadas de los grupos en una lista y no importa si se encuentra con el mismo tamaño, se agrega a la lista de tamaños, solo se verifica que no se agregue el mismo grupo más de una vez, porque al recorrer todas las casillas se encuentran grupos repetidos, de igual manera que la anterior se lleva un contador de cuántos grupos han ingresado a la lista de tamaños y coordenadas.

4. Análisis de resultados

Este proyecto funciona con éxito de acuerdo con los objetivos, los cuales pueden visualizarse en la tabla 2, excepto por mostrar resultados de forma gráfica en la matriz, en el sistema fue aplicada la programación lógica con éxito.

Durante el desarrollo se encontraron problemas que fueron solventados con un análisis profundo por medio de debug de PROLOG, foros de internet y análisis gráfico del problema en papel

Cuadro 2: Análisis de resultados basado en los requisitos del sistema

Requisito u objetivo	Estado
1. Declarar una matriz de tamaño variable dentro de los rangos que defina el programador.	Logrado
2. Llenar la matriz de forma manual de la forma que el usuario lo defina o de forma aleatoria con no más del 30 % de los posibles puntos llenos.	Logrado
4. Las selecciones de casillas cuando sea necesario deben hacerse sobre la interfaz y no solicitándolo por teclado.	Logrado
5. Las impresiones de resultados deben hacerse gráficamente en la matriz, procurando el buen entendimiento de los resultados.	Inconcluso, se usó JTextArea
6. Calcule el tamaño del grupo al que pertenece una casilla que se encuentre ocupada cualquiera.	Logrado
7. Calcule el número total de grupos distintos y sus tamaños ordenados por cantidad de elementos.	Logrado
8. Liste todos los grupos (de manera que se muestren tanto la cantidad de elementos como las coordenadas de cada uno de los elementos).	Logrado

5. Conclusiones y recomendaciones

El código desarrollado cumple con una serie de funciones basadas en consultas sobre una matriz con interfaz gráfica en Java y guardada como base de datos en PROLOG, el sistema cumple con casi todos los objetivos sin mayores inconvenientes, el aprendizaje obtenido de programación imperativa con el desarrollo del proyecto es muy grande y sin lugar a dudas abre la posibilidad de implementar los conocimientos y experiencia obtenidos en proyectos más escalables y de utilidad.

Referencias

Serra, S. (1994). Programación lógica. <https://labsys.frc.utn.edu.ar/ppr-2011/Unidad%20V%20-%20Paradigma%20L%C3%B3gico/Unidad%20V%20-%20Paradigma%20LOGICO.pdf>.