
INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 2

PROYECTO 1

201709149 – Percy Juventino Duarte Gálvez

Resumen

Se ha diseñado un programa para la simulación del comportamiento de células para una rejilla correspondiente a una cierta enfermedad de un listado de pacientes dados por una entrada en formato XML.

Una rejilla consiste en un conjunto de celdas representantes de células con dos posibles estados, sanas e infectadas. Según su posición y células vecinas durante n periodos pueden sufrir cambios dichas células. El programa grafica los estados para cada período y crea una salida en formato XML con el diagnóstico. El seguimiento es representado por colores siendo celdas blancas sanas y negras infectadas.

La implementación del programa se ha realizado con estructuras de datos TDA: listas simplemente enlazadas y matrices ortogonales, así como de la librería Graphviz para la realización de las gráficas.

Palabras clave

Coordenada: sistema de referencia para ubicación de posición.

Matriz: arreglo bidimensional de números o datos.

Ortogonal: 90° respecto de una referencia. Para un movimiento se da en dirección norte, sur, este y oeste.

TDA: Tipo de Dato Abstracto (modelo de datos en memoria dinámica).

XML: Xtensible Markup Language (Lenguaje de marcado extensible).

Abstract

A program has been designed to simulate the behavior of cells for a grid corresponding to a certain disease from a list of patients given by an input in XML format.

A grid consists of a set of cells representing cells with two possible states, healthy and infected. Depending on their position and neighboring cells during n periods, these cells can undergo changes. The program graphs the states for each period and creates an output in XML format with the diagnosis. Tracking is represented by colors with white healthy cells and black infected cells.

The implementation of the program has been carried out with TDA data structures: simply linked lists and orthogonal matrices, as well as the Graphviz library for making the graphs.

Keywords

ADT: Abstracta Data Type (dynamic memory data model).

Coordinate: reference system for position location.

Matrix: bidimensional array of numbers or data.

Orthogonal: 90° from a reference. For a movement it occurs in a north, south, east and west direction.

XML: Xtensible Markup Language.

Introducción

Se ha hecho manejo de listas simplemente enlazadas, doblemente enlazadas, matrices ortogonales y lectura de archivos XML principalmente además de la herramienta Graphviz para la representación gráfica de las estructuras manejadas en el programa.

El manejo de las estructuras permite la modificación de las posiciones, atributos y generalización de los comportamientos que puedan tener los nodos que le conforman. Recorriendo la matriz se implementado un análisis de nodo en nodo mientras estos no sean nulos, comparando la posición actual con las posiciones adyacentes (ortogonalmente), así como con una segunda matriz de referencia que contiene el patrón deseado para el análisis libre de toma de decisiones de los cambios que sufren las células durante cada periodo.

a) Estructura del programa

El algoritmo principal desarrollado para la solución consta de un análisis ortogonal en el cual se recorre nodo por nodo verificando las ocho direcciones vecinas siempre que no sean nulas. Se contabiliza la cantidad de vecinos para cada nodo y se verifica si la celda debe cambiar su estado de sana o infectada. De ser así una vez analizadas todas las células, se procede a generar el nuevo patrón.

Una vez obtenido un nuevo patrón, se realizan las mismas verificaciones hasta alcanzar la cantidad máxima correspondiente al periodo a evaluar.

Para cada período se hace a su vez una comparación de patrones buscando concurrencias exactas con el patrón inicial o un patrón previo. De ser encontradas dichas frecuencias se establecen las recurrencias N para N periodos de recurrencia del patrón inicial y N_1 para la recurrencia de patrones distintos.

Finalmente, según los valores de N y N_1 se provee un diagnóstico del tipo de enfermedad, siendo casos leves, graves y mortales para el paciente diagnosticado. Dicha información es proporcionada en la salida del archivo XML generado por el programa.

b) Clases

b.1) Clase menú:

Se encarga de desplegar las opciones iniciales del programa. En él, el usuario solo debe escribir un número correspondiente a la opción deseada y presionar la tecla ENTER para acceder a dicha opción.

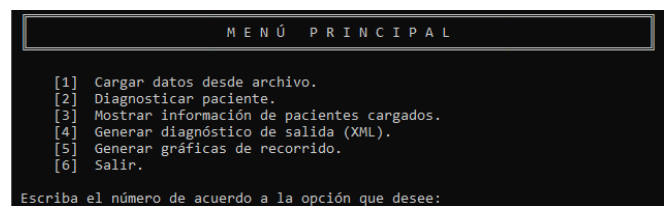


Figura 1. Menú principal

Fuente: elaboración propia.

b.1.1) [1] Cargar archivo

Permite al usuario elegir entre escribir directamente la ruta de carga del archivo o abrir el explorador de archivos del sistema para seleccionar el archivo de entrada. El archivo de entrada debe poseer la extensión “.xml”. Se encarga también internamente de procesar toda la información de la entrada y almacenarla en memoria para la ejecución del resto de tareas del programa.

b.1.2) [2] Diagnosticar paciente

Ofrece al usuario la elección de un paciente para realizar su diagnóstico. Detalla la información del paciente y muestra las rejillas para cada periodo de ejecución. Finalmente provee un diagnóstico del tipo

de enfermedad correspondiente a la rejilla del paciente seleccionado.

```
Se diagnosticará al siguiente paciente:
-----
Nombre: Denny
Edad: 24
Dimensión: 10 x 10
Periodos: 5
Rejilla:
    Células sanas: 76
    Células infectadas: 24

[1][1][0][0][0][0][0][0][1][1]
[1][1][0][0][0][0][0][0][1][1]
[0][0][0][0][0][0][0][0][0][0]
[0][0][0][0][0][0][0][0][0][0]
[0][1][1][0][0][0][0][1][1][0]
[0][1][1][0][0][0][0][1][1][0]
[0][0][0][0][0][0][0][0][0][0]
[0][0][0][0][0][0][0][0][0][0]
[1][1][0][0][0][0][0][0][1][1]
[1][1][0][0][0][0][0][0][1][1]
-----

Se ha iniciado un diagnóstico...
Estado de la rejilla tras 1 periodo(s):
Células sanas: 76
```

Figura 2. Menú 2: despliegue de datos de diagnóstico.
Fuente: elaboración propia.

b.1.3) [3] Mostrar información de pacientes cargados.

Muestra más detalladamente a los pacientes cargados en memoria, sus datos y rejilla inicial.

```
Nombre: Mildred
Edad: 11
Dimensión: 10 x 10
Periodos: 15
Rejilla:
    Células sanas: 92
    Células infectadas: 8

[1][1][0][0][0][0][0][0][0][0]
[1][0][0][0][0][0][0][0][0][0]
[0][1][0][0][0][0][0][0][0][0]
[0][0][0][0][0][0][0][0][0][0]
[0][0][0][0][0][0][0][0][0][0]
[0][0][0][0][0][1][0][0][0][0]
[0][0][0][0][0][1][0][0][0][0]
[0][0][0][0][0][0][0][0][0][0]
[0][0][0][0][0][0][1][0][0][0]
[0][0][0][0][0][0][1][0][0][0]
-----

Nombre: Abril
Edad: 17
Dimensión: 10 x 10
Periodos: 15
Rejilla:
    Células sanas: 88
    Células infectadas: 12
```

Figura 3. Ejemplo información de pacientes.
Fuente: elaboración propia.

b.1.4) [4] Generar diagnóstico de salida (XML).

Genera un archivo en formato XML con la información de todos los pacientes y su diagnóstico de tipo de enfermedad determinada.

b.1.4) [5] Generar gráficas de recorrido.

Permite al usuario seleccionar un paciente para graficar todo el recorrido para la cantidad de periodos propia del paciente.

Tras haber elegido una matriz y haber sido escrito el archivo DOT, se abrirá automáticamente la imagen generada.

b.2) Clase Lector

La clase lector se encarga de leer el archivo de entrada y extraer los datos del formato XML almacenándolos en listas. Para almacenar los paciente, extrae los datos de las etiquetas <paciente> verificando sus propiedades de dimensiones, datos personales y rejilla. Omite si se encuentran irregularidades en el archivo de entrada, de lo contrario almacena todos los pacientes y sus datos en una lista simplemente enlazada.

b.3) Clase Lista_pacientes

Lista simplemente enlazada cuyos nodos son los pacientes obtenidos del archivo de entrada.

b.4) Clase Paciente

Nodo de la clase Lista_pacientes, almacena los datos de todos los pacientes para realizar su posterior diagnóstico.

b.5) Clase Matriz

Estructura de datos ortogonal con apuntadores para cada nodo que la conforma. Se encarga de almacenar a los nodos y de realizar las operaciones lógicas sobre el comportamiento de las células en cada periodo.

b.6) Clase Nodo

Nodo de la clase Matriz que almacena células y su estado para un determinado período.

b.7) Clase Escritor

Se encarga de la escritura del archivo de salida XML y el código en lenguaje DOT necesario para graficar el camino obtenido si la misión fue exitosa. Almacena la matriz elegida y procede a extraer sus datos para colocar la información necesaria para poder graficar. Una vez finalizado el código lo compila y ejecuta de inmediato.

Anexos:

```
<?xml version="1.0" encoding="UTF-8"?>
<pacientes>
  <paciente>
    <datospersonales>
      <nombre>Denny</nombre>
      <edad>24</edad>
    </datospersonales>
    <periodos>5</periodos>
    <m>10</m>
    <resultado>mortal</resultado>
  </paciente>
  <paciente>
    <datospersonales>
      <nombre>Mildred</nombre>
      <edad>11</edad>
    </datospersonales>
    <periodos>15</periodos>
    <m>10</m>
    <resultado>leve</resultado>
  </paciente>
  <paciente>
    <datospersonales>
      <nombre>Abril</nombre>
      <edad>17</edad>
    </datospersonales>
    <periodos>15</periodos>
    <m>10</m>
    <resultado>leve</resultado>
  </paciente>
</pacientes>
```

Figura 4. Ejemplo de salida en formato XML.

Fuente: Elaboración propia.

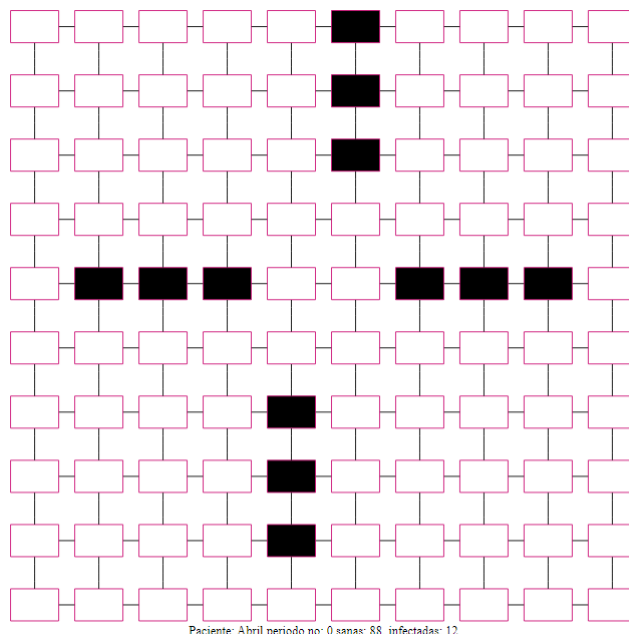


Figura 5. Ejemplo de rejilla graficada.

Fuente: Elaboración propia.

Referencias bibliográficas

Graphviz. *Documentation*, 2010.

Disponible en: graphviz.org

Serrano. M. *Estructura de datos*. Universidad
de Valladolid. Segovia.

Disponible en: www.infor.uva.es