

Universidad de Santiago de Chile
Facultad de ingeniería
Departamento de ingeniería en informática



Informe laboratorio N°1

Paradigma Funcional

Nombre: Jordán Arias Hurtado
Profesor: Víctor Flores Sánchez
Sección: B-2

INDICE

Introducción:	3
Descripción del problema:	3
Descripción del paradigma:	3
Análisis del problema:	3
Diseño de la solución:	4
Consideraciones de la implementación:	4
Instrucciones de uso:	4
Resultados:	5
Conclusiones:	5
Bibliografía:	6
Anexos:	6

Introducción:

En este informe se explicará la implementación de una red de metro, la cual es posible administrar y gestionar. Para su desarrollo se utilizó el lenguaje de programación Scheme, con el IDE Racket, teniendo en cuenta el uso del paradigma funcional para el desarrollo del programa.

Descripción del problema:

En el contexto del alza de formas sustentables y optimas de transporte urbano, surge la necesidad de mas maneras de transportarse de manera eficiente, por lo tanto, se opto el uso de una red de metro para abarcar este desafío.

Esta red consiste en un sistema de transporte subterráneo a través de vehículos motorizados siguiendo rieles a lo largo de túneles ubicados a lo largo y ancho de la ciudad en cuestión, lo cual afecta directamente al descongestionamiento y contaminación en la superficie, además de ser totalmente público y accesible para todos los usuarios.

Descripción del paradigma:

El paradigma funcional trata de ajustarse más a la forma de resolver el problema, que a las construcciones del lenguaje necesarias para cumplir con la ejecución en esta máquina. En la cual las variables no son necesarias, ya que no se considera a la memoria necesaria, pudiéndose entender un programa como una evaluación continua de funciones sobre funciones, y por consecuente evita los estados intermedios y la modificación de datos durante la misma.

Análisis del problema:

Los requisitos específicos de este problema son la creación de una red operativa de metro compuesta por líneas, compuestas por una serie de estaciones. Trenes, compuestos de vagones los cuales circulan por las distintas líneas. Y la administración de esta.

Y las funciones en específico a cubrir son:

- line-length: calcula la distancia total de una línea
- line-section-length: calcula la distancia entre dos estaciones
- line-cost: calcula el costo total de una línea
- line-section-cost: calcula el costo entre dos estaciones
- train-capacity: permite determinar la capacidad máxima de pasajeros del tren
- subway->string: expresar una red de metro en un formato String

- where-is-train: Función que permite determinar dónde está un tren a partir de una hora indicada del día y debe entregar la estación más próxima en la que se encuentre un tren considerando los tramos de la línea
- subway-train-path: Función que permite ir armando el recorrido del tren a partir de una hora especificada

Diseño de la solución:

En primera instancia, se comenzó el desarrollo de los requisitos principales, siendo estos los constructores principales, con los cuales se estableció la forma de las entidades de esta red, mediante el uso exclusivo listas y pares. Posteriormente se desarrollaron los selectores para verificar la consistencia de los mismos constructores, seguido por los selectores para tener acceso a cualquier dato que se requiera, y finalmente los modificadores para poder añadir o eliminar algún elemento para la red.

En la creación de los elementos que componen la red, ósea , los constructores, se siguió un orden ascendente en la complejidad de estos, es decir, se comenzó por los elementos más básicos de cada uno, y a partir de esa base, se fueron creando elementos cada vez más complejos. Y para la creación de los selectores, se optó por tener uno para cada dato que sea necesario de los TDA para trabajar cómodamente, ya que, a pesar de que racket nos brinda funciones como car, y cdr para seleccionar elementos de la lista, se vuelve confuso entender que dato se está obteniendo exactamente de una función que utilice, por ejemplo: `car(cadr(cdr (car list)))` .

Consideraciones de la implementación:

Para este laboratorio, se estructuro la red como la unión de 6 TDA, estos son: station, section, line, pcar, train, y subway. Cada uno comprende su propia representación, con sus respectivos constructores, selectores, funciones de pertenencia, y modificadores respectivamente. Sin embargo solo se pudieron implementar los tres primeros TDA, station, section, y line, para este código

En el desarrollo de este, se utilizó el IDE Racket versión 8.12, en el lenguaje Scheme, y la única biblioteca empleada fue racket

Instrucciones de uso:

Para un correcto uso del programa para crear la red, hasta lo que se logro avanzar, es necesario el correcto uso de las funciones constructoras, ósea partir desde los elementos más básicos, por ejemplo, en el caso de la línea, que se vaya construyendo primero las estaciones, a partir de estas definir secciones como pares de estaciones, para finalmente, unir estas secciones y construir una línea funcional, y cabe mencionar que se puede

verificar el que cada uno de los componentes esté correctamente construido. Lastimosamente, esto todo lo que puede hacer el código hasta el momento.

Resultados:

Como se menciono anteriormente, no se logró la totalidad de la red, y el código se limita a la creación de líneas individuales de la red de metro, lo que implica que no se llegó a implementar el sistema de trenes, ni la gestión y administración de estos en las vías.

Conclusiones:

De manera personal, me apena el hecho de solo haber podido realizar los TDA relacionados a la línea de metro, sin embargo, resalto el hecho del uso del paradigma funcional para la totalidad de este código, pues si bien en un inicio resulta poco natural, ya que desde el inicio de la carrera e incluso antes, siempre hubo manejo de variables para la resolución de problemas, este va siendo bastante simple al momento de entender su lógica detrás, e una interesante forma de abarcar la solución a un problema.

Bibliografía:

- Editorial de la Universidad Católica de Salta. (s. f.). *Vista de la programación funcional: un poderoso paradigma*. Universidad Católica de Salta.
<https://revistas.ucasal.edu.ar/index.php/CI/article/view/182/159>

Anexos: