

Informe Laboratorio N° 2

Paradigma Lógico

Nombre estudiante: Daniel Villarroel Montenegro

Profesor: Gonzalo Martínez

Sección: B-2

Introducción

Como hemos visto anteriormente con el paradigma funcional , donde teníamos que ver todo como una sola “función” donde la salida solo dependía de la entrada , ahora nos adentramos al siguiente paso el paradigma lógico.

En esta ocasión utilizaremos el lenguaje de prolog , imagines que tenemos un robot que no sabe nada, al no ser humano no sabe que es un tren, por ejemplo , así tenemos que enseñarle que un tren es un vehículo que contine carros , vagones y que pertenecen a una línea de metro,etc. Con esta idea el paradigma lógico declara una base de conocimiento a través de hechos y reglas, y a través de consultas , se obtiene una respuesta.

Usando el programa SWI-Prolog se procederá a realizar el laboratorio , se expondrá la problemática de este trabajo junto con su análisis, se explicará el paradigma que se utilizará, se detallarán los conceptos involucrados, se describirá la solución implementada y su posterior evaluación, y finalmente se presentarán las conclusiones correspondientes.

Problema

Las líneas de metros son conocidas por presentar múltiples elementos dentro de la misma red por los cuales se agrupan. Manejar todo requiere de un sistema que pueda administrar distintas tareas, como los vagones de tren, las estaciones, etc.

Para ello se pide la realización de un sistema para una red de líneas de metro donde se permita realizar al usuario las siguientes operaciones: Creación y Gestión de líneas de Metro, Añadir Estaciones, Planificación de Rutas y Conexiones, Gestión de Trenes, Control de Tarifas, Gestión de Personal, Mantenimiento de Infraestructuras, Monitoreo y Control, Interfaz de Usuario. Cada una de estas operaciones requerirá distintos TDA's (Tipo de datos Abstractos) para poder realizar las distintas acciones.

Paradigma

Para la realización, como antes se mencionó, fueron necesario la programación lógica, a través del lenguaje de programación prolog, el cual consiste en programación declarativa y su funcionamiento es a través de la lógica booleana (true, false, or, and) donde los símbolos se le llaman términos (Term) y estos son contenidos en hechos y/o reglas.⁴

- Metas primarias: Especificar las que son consultadas directamente por el intérprete a la base de conocimiento.
- Metas secundarias: Las que son consultadas de forma interna por las metas primarias, pero no desde fuera de la base de conocimiento.
- Hechos: Un hecho es un mecanismo para representar propiedades o relaciones de los objetos que se están representando.
- Los hechos declaran los valores que van a ser verdaderos o afirmativos para un predicado en todo el programa.
- Reglas: Hechos que dependen de la veracidad de otros hechos.
- Recursión: Prolog permite la recursión , declarando un caso base , y haciendo el llamado nuevamente , el funcionamiento de prolog es como si fuera la recursión “en reversa” a comparación de la recursión de otros lenguajes
- Átomo: Nombres de objetos, propiedades, o relaciones. Estos deben empezar en minúscula.
- Consulta: Las consultas nos permitirán acceder a la base de conocimiento , enseñándole a prolog y dando la respuesta a nuestra consulta, dentro de los límites de la base.

Análisis del Problema

Para el problema tendremos ciertos requisitos que cumplir que se mencionan a continuación:

- Creación y Gestión de Líneas de Metro
- Añadir Estaciones
- Planificación de Rutas y Conexiones
- Gestión de Trenes

- Control de Tarifas
- Gestión de Personal
- Mantenimiento de Infraestructuras
- Monitoreo y Control
- Interfaz de Usuario

Para la creación y gestión de los distintos “objetos” (tren, estaciones, etc.) será importante tener TDA, para poder manejar la parte de constructores , selectores, etc.

Diseño de Solución

Dado el enunciado del problema para poder generar las distintas operaciones necesitamos diferentes TDA's que implementar. Para esta ocasión serán necesario los siguientes TDA's

1. TDA-station: Creación de las diferentes estaciones con ID y nombre único, necesario un TDA-constructor, selector, modificador y de pertenencia.
2. TDA-section: Creación de una sección del metro el cual contiene un punto1 y punto2 correspondientes a estaciones, necesario un TDA-constructor, selector, modificador y de pertenencia
3. TDA-line: Creación de una línea con múltiples funciones. Necesario distintos TDA constructor , selector, modificador y pertenencia para poder completarlo
 - Otros Predicados: En otros predicados como “lineLength” se utilizará la recursión proporcionada por prolog y revisada en clase (Clase 07b) donde tendremos un caso base que estará vacío y un llamado al predicado nuevamente con toda la meta secundaria correspondiente, usando principalmente los selectores para poder obtener los distintos valores especificados en el laboratorio.

4. TDA-pcar: Creación de carros para pasajeros que conforman un comboy, para ello se utilizan el dominio indicado y uso de estructuras basada en listas y/o pares
5. TDA-train : Creación de convoy o tren , en la que tiene alojado distintos carros de pasajeros.
6. TDA-driver: Predicado que permite crear un conductor cuya habilitación de conducción depende del fabricante de tren (train-maker)
7. TDA-subway: Predicado que permite crear una red de metro.

Aspectos de implementación

Todo el código fuente fue realizado en la SWI-Prolog version 9.2.5. Para el código se dividió en los diferentes TDA mencionados, todo estarán en un archivo tda_main que contendrá los diferentes “use_module” para poder obtener las “module” de los demás TDA para así hacer las consultas correspondientes.

Instrucciones de uso

Todo el Código fuente se encuentra en la carpeta, donde se ubicarán los distintos TDA , como el tda_main, para la consulta

Consideraciones:

- Para la realización de este laboratorio los tipos de estación , rieles, etc, están definidos como string para evitar complicaciones al momento de las consultas y demás.
- Dentro del código fuente se encuentra el script de prueba del laboratorio

Resultados y Autoevaluación

Para los resultados el programa fue capaz de realizar tareas como la creación modificación de estación sección, trenes y líneas, generando hacia varias líneas de trenes del mundo como es el metro de New York y el metro de Paris.

Para los otros predicados se completó parcialmente, obteniendo dificultades en lineSectionLength, al no obtener los resultados esperados.

Para este laboratorio se alcanzó a realizar hasta el requisito 16 del enunciado.

Conclusión

En este laboratorio tuvimos que estudiar y poner en práctica otro paradigma , en este caso el paradigma lógico a través de prolog.

Pudimos determinar los distintos TDA necesarios para la construcción de nuestra línea de metro , para poder realizar a través del programa SWI-Prolog en inicio de nuestro programa.

El programa como vimos en los resultados consiguió obtener respuesta a distintos requerimientos cumpliendo parcialmente el objetivo, ya que no se pudieron completar distintos requisitos en su totalidad.

Con este laboratorio pudimos aprender distintas formas de abordar el mismo problema de la vez anterior pero ahora desde otro punto de vista programable , para poder avanzar en nuestra meta final.

Referencias

Manual SWI-Prolog. (s. f.). https://www.swi-prolog.org/pldoc/doc_for?object=manual

Clase 07b-Programacion Lógica (Profesor Gonzalo).