



UNIVERSIDAD
DE SANTIAGO
DE CHILE

Informe de Laboratorio

Emilio Viñals
Paradigmas de programación
Gonzalo Martinez
4 de Junio del 2023

Introducción

En este informe se describirá el desarrollo de un programa en el lenguaje de programación Prolog, utilizando la herramienta SWI-Prolog, con el objetivo de aplicar los conocimientos adquiridos en el paradigma lógico de programación.

El propósito de este programa es realizar una simulación de un sistema de archivos utilizado en los computadores. A través de la implementación de reglas y relaciones lógicas, se podrá interactuar con este sistema de archivos, obteniendo información y realizando modificaciones según sea necesario.

El paradigma lógico de programación se basa en la lógica formal y la resolución de problemas a través de reglas y relaciones lógicas. Prolog, como lenguaje de programación lógica, proporciona un enfoque declarativo para representar el conocimiento y resolver problemas.

En este informe, se presentarán los conceptos teóricos necesarios para comprender el programa desarrollado, así como una explicación detallada de su implementación. Se describe la estructura del sistema de archivos simulado, las reglas y relaciones utilizadas, y las funcionalidades implementadas para interactuar con el sistema.

Además, se realizarán pruebas y se presentarán ejemplos para ilustrar el funcionamiento del programa y demostrar su utilidad en la simulación de un sistema de archivos.

Descripción breve del problema

El problema planteado era el mismo del primer laboratorio, generar un sistema de manejo de archivo, similar al de los sistemas operativos como windows, usados en el cmd, para esto se debió crear un sistema de listas y strings para poder representarlo.

Descripción del paradigma

El paradigma lógico es una forma de la programación para resolver problemas a través de reglas que uno le va enseñando al programa.

En general, este paradigma se enfoca a la relación que tiene entre cada uno de sus elementos incorporados en el programa mediante términos y funciones

Análisis del problema respecto de los requisitos específicos que deben cubrir

El problema más grande en prolog será el guardar la información actual del sistema que se va a crear, ya que se tiene que ir modificando constantemente y en un orden específico para que el sistema pueda continuar sus funciones dando respuestas positivas que uno busca.

Diseño de la solución

Para la solución del problema, se decidió trabajar con listas, en las cuales las funciones tomaban las archivos o nombres que necesitaba y las unía en una lista, en algunos casos, como los usuarios, eran otra lista, en estos casos, como por ejemplo, saber la dirección en la que uno se encontraba, se tomaba el primer elemento de la lista, osea, el elemento 0, y de esta manera poder ordenar la información.

Aspectos de implementación

Para el desarrollo de este proyecto, se usó el compilador online de SWISH – SWI-PROLOG y sus librerías por defecto

Instrucciones de uso

Para el uso de la herramienta de prolog llamada SWISH, se debe ingresar a la página web:

<https://swish.swi-prolog.org/> luego clicar en la parte de crear un programa, el código se trabajó siempre en un archivo de texto (.txt), ya que los avances en el programa web no se guardan, por lo tanto, la información escrita en el archivo de texto debe pasarse luego al programa de SWISH, para finalmente en la esquina inferior derecha, llamar a las funciones en el orden que se quiere ejecutar, separadas por comas (,) y la última en un punto (.), a continuación un ejemplo del llamado de funciones:

```
system("newSystem", S1), systemAddDrive(S1, "C", "OS", 10000000000, S2),  
systemRegister(S2, "user1", S3), systemRegister(S3, "user2", S4), systemLogin(S4,  
"user1", S5), systemSwitchDrive(S5, "C", S6).
```

Resultados y autoevaluación

El trabajo en sí fue bastante complicado en el momento de intentar cambiar de paradigma, ya que se venía desde otro pensamiento, en cuanto se regula y se ajusta la forma de pensar para prolog, todo pareciera ser más claro, pero en el proceso surgieron bastantes fallos.

Al momento de desarrollar el laboratorio, las cantidades de fallos eran mayores a los éxitos, ya que los métodos dentro del programa son demasiado específicos algunas veces, lo que provoca que para poder darle una solución, sea necesario generar más de una vuelta por distintos otros métodos, los cuales no siempre eran compatibles y había que hacer nuevamente un método que se ajustara a los requerimientos del problema.

Conclusiones del trabajo

En conclusión, el paradigma lógico funciona de manera más adecuada en otros ambientes, como por ejemplo en la busca rápida de respuesta, más que en guardar y modificar información, ya que para eso hay que estar constantemente enseñándole al programa como funciona cada una de sus opciones y operadores