

Instituto Tecnológico de Costa Rica

Área de Ingeniería en Computadores

CE-3104: Lenguajes, Compiladores e Intérpretes

Profesor:

Marco Rivera Meneses

Tarea 1: WazeLog

Paradigma lógico

Integrantes:

Carlos Adrian Araya Ramirez

Michael Shakime Richards Sparks

José Andrés Solano Mora

Abril, 2021

Tabla de Contenidos

1.	Breve descripción del proyecto	3
1.1.	Descripción detallada de los algoritmos de solución desarrollados	3
1.2.	Descripción de las funciones implementadas	3
1.3.	Descripción de la ejemplificación de las estructuras de datos desarrolladas	4
1.4.	Problemas sin solución	4
1.6.	Plan de Actividades realizadas por estudiante	4
1.7.	Conclusiones	6
1.8.	Recomendaciones	6
1.9.	Bibliografía consultada en todo el proyecto	7
2.	Bitácora en digital, donde se describen las actividades realizadas.	8

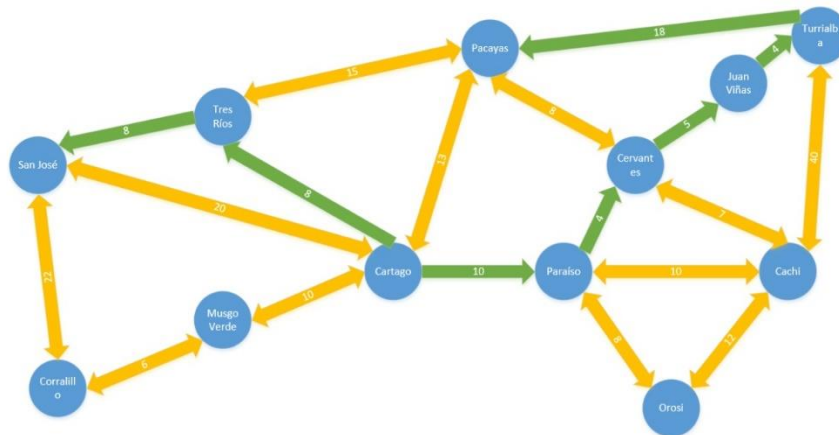
1. Breve descripción del proyecto

WazeLog consiste en un sistema experto que ofrece el servicio de ubicación y guía de tránsito, este sistema funciona a partir grafos y gramáticas libres de contexto para brindar al usuario una interfaz amigable.

1.1. Descripción de los hechos y reglas implementadas.

unions(LugarA,LugarB,Peso).

Para la estructuración del grafo se utilizó el hecho “unions” que definirá el nombre del nodo origen “Lugar A”, así como el nodo destino “Lugar B”, incluyendo a su vez un “Peso” específico de un nodo a otro, cabe recalcar que estas uniones son unidireccionales, o sea que si se desea especificar una unión que funcione en ambos sentidos se deberá crear una regla más con el “Lugar A” y “Lugar B” alternados.



1.2. Descripción de las estructuras de datos desarrolladas.

- **El camino resultante:** ['Turrialba', 'Pacayas', 'Cartago', 'Musgo Verde'].
- **Peso:** X
- **Tiempo:** 2 * X

Esta es la respuesta de la mejor ruta calculada por el algoritmo Dijkstra, donde el primer elemento es donde empieza la ruta y el resto de los elementos en orden son los caminos que debe seguir hasta llegar al destino final. El peso y el tiempo se

manejan por aparte.

1.3. Descripción detallada de los algoritmos desarrollados.

La regla `findminpath(Inicio,Final,Peso,Camino)` se comporta como una variación del algoritmo de **Dijkstra** utilizado para la búsqueda del camino más corto entre dos nodos dentro de un grafo, utilizando `findapath(Inicio,Final,Peso,Camino,Lista)` se realiza una búsqueda de todos los caminos posibles de un nodo a otro, luego estos son almacenados temporalmente en `solution(Peso,Camino)`. A diferencia del algoritmo **Dijkstra**, `findminpath` evalúa cada una de las posibles rutas, sacrificando eficiencia, sin embargo, fue necesario realizarlo de esta manera para asegurarse que la ruta calculada sea en definitiva la más corta, pues `findapath` no muestra las rutas ordenadas y así como la ruta más corta podría ser la primera solución, también podría tratarse de la última.

1.4. Problemas sin solución

- De manera satisfactoria se logró implementar todo lo necesario según la especificación del proyecto.

1.5. Plan de Actividades realizadas por estudiante

Tarea	Tiempo estimado	Responsable	Fecha de entrega
Iniciar el plan de trabajo y la documentación	1h	Todos	22/03/2021
Encargarse de las bitácoras	3h	José	09/04/2021

Investigar los grafos dirigidos y mixtos en PROLOG	2h	José y Adrián	23/03/2021
Implementar el grafo	2 días	Todos	24/03/2021
Investigar los sistemas expertos en PROLOG	2h	Shakime y José	24/03/2021
Investigar cómo obtener y analizar el input del usuario	2h	Adrián	26/03/2021
Implementar la interfaz de usuario con gramáticas libres de contexto	2 días	Todos	1/04/2021
Realizar todas las pruebas necesarias de la interfaz	1 día	Adrián	3/04/2021
Hacer el manual de usuario	2h	Shakime	5/04/2021

1.6. Problemas solucionados.

- No se sabía cómo volver a pedir los inputs cuando el usuario ingresaba un valor invalido, investigando un poco la documentación de prolog, se encontró la función "repeat" la cual permite hacer backtraking hasta donde se puso esta función, ya sea por un falso involuntario o uno forzado con fail.
- Cuando se ingresa un lugar que no existe en la base de datos al volver a intentarlo no se logra validar, aunque la entrada esté bien, si se ingresa desde un inicio bien no dará

problema. Es cuando se ingresa un lugar que no existe y una oración que si existe. Se solucionó, de la siguiente manera, se hizo una función aparte para verificar la entrada recibido y ver si es una ciudad, lugar o establecimiento, esta misma contemplaba una llamada para validar si la entrada corresponde a una oración o no.

1.7. Conclusiones

- i.** Se concluye que desarrollar una aplicación con el comportamiento experto utilizando Prolog, es una opción factible y favorable, debido a las propiedades y ventajas que brinda un paradigma lógico.
- ii.** Se aplicaron los conceptos y habilidades aprendidas durante el estudio del paradigma de programación lógico, como la declaración de hechos y reglas para satisfacer el objetivo.
- iii.** Se logró modelar el problema mediante la implementación de diversos hechos y reglas para manipular las listas con las que fue modelada la base de datos de los restaurantes. Entre ellas, la función miembro, la obtención de las cabezas de las listas, entre otras.
- iv.** Se logró implementar un sistema experto que se encarga de recomendar al usuario restaurantes a partir de las preferencias brindadas, analizando su gramática y realizando deducciones a partir de la base de datos.

1.8. Recomendaciones

- a.** Prolog brinda características favorables para el desarrollo de sistemas expertos utilizando gramáticas libres de contexto, por lo que se recomienda la implementación de aplicaciones que involucren este tipo de sistemas utilizando un paradigma lógico.
- b.** Para la utilización de la aplicación, se recomienda leer el manual de usuario antes de utilizar la aplicación, esto con el fin de comprender las gramáticas válidas y las distintas partes que componen el programa.
- c.** Recordar utilizar comillas a la hora de escribir el input al programa, ya que de no utilizarlas el programa no funciona correctamente.
- d.** Se recomienda el uso de oraciones claras, que contengan estructuras con verbos y predicados definidos, esto con el fin de facilitar la interacción con el Sistema Experto .

1.9. Bibliografía consultada en todo el proyecto

Soler, F (2020). Programación de gramáticas clausales en Prolog. Recuperado el 4 de Octubre de 2020, de <https://personal.us.es/fsoler/papers/05capgram.pdf>

Valverde, J. (2020). Gramáticas de cláusulas definidas - JARV's blog. Recuperado el 4 de Octubre de 2020, de <http://jariaza.es/blog/post/prolog/gramaticas-de-clausulas-definidas.md>

SWI-Prolog -- split_string/4. (2014). Converting a user input string to list. Recuperado el 5 de Octubre de 2020, de https://www.swi-prolog.org/pldoc/man?predicate=split_string/4

SWI-Prolog -- atom_string/2. (2020). Recuperado el 13 de Octubre de 2020, de https://www.swi-prolog.org/pldoc/man?predicate=atom_string/2

2. Bitácora en digital, donde se describen las actividades realizadas.

Fecha	Estudiante(s)	Actividad
22/03/2021	Todos	Primera reunión para comprender, compartir ideas y organización: División del trabajo correspondiente a cada uno de los integrantes de acuerdo con la dificultad estimada de este.
23/03/2021	José y Adrián	Investigación e inicio de implementación inicial de los hechos y reglas para el grafo.
24/03/2021	José y Shakime	Se termina la implementación del grafo.
25/03/2021	Shakime y Adrián	Definición de hechos y reglas para el BNF. Para iniciar este desarrollo, fue necesario consultar el punto uno y dos de la bibliografía, para reforzar los conceptos y comprender el funcionamiento de las gramáticas libres de contexto.
27/03/2021	Shakime y José	Implementación del análisis del input del usuario. Actualización del BNF. Para este análisis se necesitó investigar en la página web del punto tres de la bibliografía.
1/04/2021	Todos	Reunión de organización y actualización del trabajo realizado hasta el momento.
2/04/2021	Todos	Reunión para terminar de definir la búsqueda, reglas y hechos para el comportamiento experto de la aplicación.
5/04/2021	Shakime y Adrián	Implementación del análisis gramático y obtención de palabras claves para oraciones relacionadas con ubicaciones.