



Escuela de Computadores
Lenguajes, Compiladores E Intérpretes
Grupo 1

Tarea Programada 2

Restaurantec

Estudiantes:

Diego Vega Mora 2020043916
Fabian Castillo Cerdas 2020202938
Kevin Carranza Blanco 2020163275

Profesor: Marcos Rivera Meneses

Fecha de entrega: 5 de junio del 2023

1° Semestre 2023

Índice

Índice	2
Descripción de los hechos y reglas implementadas	3
Descripción de las estructuras de datos desarrolladas	4
Descripción del algoritmo general de solución	5
Problemas sin solución	6
Plan de Actividades	7
Problemas solucionados	8
Conclusiones	9
Recomendaciones	10
Bibliografía	11
Minutas	12
Bitácoras	13
Kevin:	13

Descripción de los hechos y reglas implementadas

El programa implementa un sistema básico de procesamiento del lenguaje natural en Prolog para validar la gramática de oraciones en español. A continuación, se describen los hechos y las reglas utilizadas:

1. Pronombres:

- Se definen pronombres masculinos, femeninos y neutros utilizando las listas [el|S], [ella|S] y [nosotros|S], respectivamente.
- Los pronombres se utilizan para identificar el sujeto de una oración.

2. Sustantivos:

- Se definen sustantivos utilizando la regla `sustantivo([_|S],S)`.
- Los sustantivos se utilizan para completar el sujeto de una oración.

3. Verbos:

- Se definen verbos utilizando listas de palabras como [apetece|S], [comer|S], [tomar|S], etc.
- Los verbos se utilizan para identificar la acción o la relación en una oración.

4. Reglas gramaticales:

- `oracion(A,B)` valida una oración en español.
- `sujeto(A,B)` extrae el sujeto de una oración eliminando el pronombre y el predicado.
- `predicado(A,B)` extrae el predicado de una oración eliminando el verbo.

5. Validación gramatical:

- `validacion_de_gramatica(Oracion)` verifica si una oración es gramaticalmente correcta.
- Utiliza las reglas anteriores para validar la estructura gramatical de la oración.

Descripción de las estructuras de datos desarrolladas

En el programa, se utilizan principalmente listas de palabras para representar las diferentes estructuras gramaticales y elementos del lenguaje natural. A continuación, se describe la estructura de datos principal y cómo se utiliza en el programa:

1. Listas de palabras:

- Las listas de palabras se utilizan para representar oraciones, sujetos, predicados, pronombres, sustantivos y verbos.
- Cada elemento de la lista representa una palabra o un conjunto de palabras que forman parte de una oración o una estructura gramatical.
- Por ejemplo, una lista [quiero, comer] representa la oración "quiero comer", donde "quiero" es el verbo y "comer" es el sustantivo.
- También se utilizan listas de palabras para representar pronombres, como [el] para el pronombre masculino "él".

2. Estructuras de datos compuestas:

- Además de las listas de palabras individuales, se utilizan estructuras de datos compuestas en forma de listas anidadas para representar sugerencias o posibles opciones en ciertas reglas.
- Por ejemplo, [querer, [comer, tomar, beber]] representa una sugerencia de verbos relacionados con el verbo "querer".
- Las estructuras de datos compuestas se utilizan para ampliar las posibilidades y dar opciones al usuario durante la interacción con el programa.

Las listas de palabras son la principal estructura de datos utilizada en el programa para representar las diferentes partes del lenguaje natural. Se construyen y manipulan a través de patrones de coincidencia en las reglas gramaticales para validar la estructura gramatical de las oraciones y proporcionar respuestas adecuadas al usuario.

Descripción del algoritmo general de solución

El algoritmo general de solución implementado en el programa se basa en el análisis gramatical de las oraciones ingresadas por el usuario y la aplicación de reglas predefinidas para comprender y responder de manera adecuada a las solicitudes. A continuación se describe el flujo de trabajo del algoritmo:

1. Lectura de la oración:

- El programa recibe una oración ingresada por el usuario en forma de una lista de palabras.
- Esta oración se pasa como entrada al algoritmo para su procesamiento.

2. Validación gramatical:

- Se aplica una validación gramatical para verificar si la estructura de la oración cumple con las reglas gramaticales definidas en el programa.
- Utilizando reglas gramaticales basadas en patrones de coincidencia, se comprueba si la oración contiene un sujeto, un predicado y una estructura gramatical coherente.
- Si la validación gramatical no es exitosa, el programa solicita al usuario que vuelva a ingresar la oración de manera comprensible.

3. Análisis semántico:

- Una vez que la oración pasa la validación gramatical, se procede al análisis semántico para comprender el significado de la oración.
- El programa utiliza reglas semánticas y patrones de coincidencia para extraer la intención o la solicitud implícita en la oración.
- Por ejemplo, identificar si el usuario quiere comer, beber o realizar alguna acción específica.

4. Generación de respuesta:

- Con base en el análisis semántico, el programa genera una respuesta adecuada que cumple con la solicitud o la intención expresada en la oración.
- Utilizando estructuras de datos predefinidas, como sugerencias de opciones y respuestas predefinidas, el programa selecciona la respuesta más apropiada.
- La respuesta se muestra al usuario, completando así el ciclo del algoritmo.

Problemas sin solución

- Ambigüedad semántica: Aunque se haya resuelto la ambigüedad gramatical, aún se pueden encontrar oraciones que son semánticamente ambiguas. Esto significa que pueden tener múltiples interpretaciones de significado válido. No hay una solución definitiva para abordar la ambigüedad semántica, ya que depende del contexto y del conocimiento del mundo.
- Contexto insuficiente: En algunos casos, la comprensión precisa de una oración puede depender del contexto que la rodea. El proyecto se basa únicamente en el análisis de oraciones individuales sin tener en cuenta el contexto más amplio, puede haber limitaciones en la interpretación y generación de respuestas precisas.
- Escalabilidad limitada: La estructura del proyecto no está muy bien diseñada para ser escalable, es decir, si no puede manejar volúmenes de datos más grandes o crecientes de manera eficiente, esto puede convertirse en un problema a medida que el proyecto se expande o se enfrenta a mayores demandas de rendimiento.
- Falta de manejo de excepciones: Puede haber casos excepcionales en el idioma donde las reglas gramaticales no se apliquen de manera estándar. Por ejemplo, frases idiomáticas, expresiones coloquiales o construcciones gramaticales inusuales podrían no ser reconocidas correctamente por las reglas definidas en el proyecto.
- Preguntas: La funcionalidad de poder realizar preguntas al sistema experto se salió de las manos razón por la que al final se prefirió no incorporar para poder mantener la integridad del código.

Plan de Actividades

Actividad	Descripción	Tiempo (h)	Encargado	Fecha de entrega
Crear base de datos	Usando diferentes hechos crear la base de datos de los restaurantes.	5	Diego	27/05
Ampliar la base de datos	El mínimo de restaurantes es de 10, por lo que es necesario agrandar la base.	2	Diego	27/05
Comenzar con el BNF	La creación de los sintagmas nominales y verbales así como los verbos	6	Fabian	01/06
Crear reglas de búsqueda	Crear el conjunto de reglas necesario para el funcionamiento del programa.	6	Kevin	01/06
Corregir y juntar	Hacer las correcciones necesarias tanto a los hechos como las reglas y enlazar el BNF con la base de datos	2	Todos	04/06
Documentación	Hacer de forma correcta la documentación de la tarea, cumpliendo con los parámetros establecidos	1	Todos	04/06

Problemas solucionados

- Validación gramatical: El algoritmo es capaz de validar la estructura gramatical de las oraciones ingresadas por el usuario. Esto implica verificar si la oración contiene un sujeto, un predicado y una estructura coherente. Si la oración no cumple con las reglas gramaticales predefinidas, el algoritmo solicita al usuario que vuelva a ingresar la oración de manera comprensible.
- Reconocimiento de intenciones: El algoritmo es capaz de analizar semánticamente las oraciones y reconocer la intención o solicitud implícita en ellas. Por ejemplo, es capaz de identificar si el usuario desea comer, beber o realizar alguna acción específica. Esto permite al programa generar una respuesta adecuada a la intención expresada por el usuario.
- Generación de respuestas coherentes: Basado en el análisis semántico, el algoritmo es capaz de generar respuestas coherentes y relevantes a las solicitudes del usuario. Utiliza estructuras de datos predefinidas, como sugerencias de opciones y respuestas predefinidas, para seleccionar la respuesta más apropiada. Esto asegura que las respuestas proporcionadas sean relevantes y útiles para el usuario.
- Manejo de diversas estructuras gramaticales: El algoritmo puede manejar diferentes estructuras gramaticales, incluyendo pronombres masculinos, femeninos y neutros, así como sustantivos y verbos en diferentes formas. Esto permite al usuario expresar sus solicitudes de diversas formas, y el algoritmo es capaz de comprender y responder de manera adecuada.

Conclusiones

- Se logró desarrollar una aplicación que cumple con reafirmar el conocimiento del paradigma de programación lógico. La aplicación permite a los usuarios interactuar con un sistema experto implementado en Prolog, lo que les brinda la oportunidad de practicar y aplicar los conceptos de programación lógica de manera práctica.
- La implementación de la aplicación en Prolog demuestra la comprensión y aplicación de los conceptos del paradigma de programación lógica. Se utilizaron elementos como predicados, reglas, estructuras de datos y un conjunto de hechos para construir la lógica del sistema experto.
- La manipulación de listas como estructuras de datos fue un aspecto fundamental en el desarrollo de la aplicación. Se utilizaron listas para representar sugerencias de respuestas, almacenar valores y realizar operaciones de búsqueda y manipulación de datos. Esto demuestra la capacidad para utilizar las estructuras de datos adecuadas en la implementación de soluciones.
- La aplicación cumple con crear un sistema experto que se comporta como un experto en el dominio especificado. El sistema puede recibir consultas en lenguaje natural, validar la gramática y reconocer la intención del usuario para generar respuestas coherentes.
- El desarrollo de la aplicación en Prolog ha brindado una experiencia práctica en la programación lógica, permitiendo consolidar los conocimientos adquiridos en este paradigma. Además, ha proporcionado una visión más profunda sobre la construcción de sistemas expertos y el uso de la lógica para la toma de decisiones.

Recomendaciones

- Mejorar la interfaz de usuario: se puede mejorar la interfaz de usuario de la aplicación para hacerla más intuitiva y atractiva visualmente. Esto ayudará a los usuarios a interactuar de manera más fácil y agradable con el sistema experto.
- Ampliar la base de conocimientos: Para enriquecer la funcionalidad y el alcance del sistema experto, se recomienda ampliar la base de conocimientos. Se pueden agregar más reglas, hechos y respuestas posibles para cubrir una mayor variedad de consultas y escenarios.
- Implementar funcionalidades adicionales: se pueden agregar funcionalidades adicionales que brinden más utilidad y versatilidad al sistema experto. Por ejemplo, se podría incorporar características como la capacidad de almacenar y recuperar sesiones anteriores, guardar consultas frecuentes, proporcionar explicaciones detalladas de las respuestas generadas, etc.
- Mejorar la validación gramatical: la validación gramatical de las consultas se puede refinar, para hacerla más precisa y robusta. Esto ayudará a evitar confusiones y a brindar respuestas más precisas a los usuarios.
- Realizar pruebas exhaustivas: Realizar pruebas exhaustivas de la aplicación para asegurarte de que funcione correctamente en diferentes escenarios e identificar posibles casos de error y realizar correcciones para mejorar la experiencia del usuario.

Bibliografia

- Bratko, I. (2012). Prolog Programming for Artificial Intelligence. Pearson Education.
- Clocksin, W. F., & Mellish, C. S. (2003). Programming in Prolog: Using the ISO Standard. Springer.
- Sterling, L., & Shapiro, E. (1994). The Art of Prolog: Advanced Programming Techniques. MIT Press.
- Cumpston, G. (1998). Prolog Programming: A Tutorial Introduction. IC Press.
- Carlsson, M., & Fasth, T. (2017). Logic Programming with Prolog. Studentlitteratur AB.
- Pereira, L. M., & Warren, D. S. (1987). Definite Clause Grammars for Language Analysis—A Survey of the Formalism and a Comparison with Augmented Transition Networks. *Artificial Intelligence*, 23(1), 69-94.

Minutas

Tema reunión	Fecha	Involucrados
Se realizó la división de los roles de trabajo entre cada uno de los integrantes mediante un grupo de WhatsApp que se tiene destinado a la tarea.	20/5/23	Todos
Se realizó una revisión general de los avances.	24/5/23	Todos
Se documentó de forma interna como externa el trabajo.	05/5/23	Todos

Bitácoras

Kevin:

Fecha: 24/05

Tiempo estimado: 2 horas

Descripción: se investiga y se crean las primeras reglas basado en la base de datos terminada.

Fecha: 27/05

Tiempo estimado: 3 horas

Descripción: se continuó con la creación de las reglas y se logró terminar.

Fecha: 18/05

Tiempo estimado: 1 hora

Descripción: se creó el print inicial y se acoplaron las reglas al BNF

Diego:

Fecha: 21/05

Tiempo estimado: 1 horas

Descripción: se experimentó con formas de hacer la base de datos basada en hechos y se comenzó la creación de 3 restaurantes: Taco Bell, Mac y Burguer.

Fecha: 24/05

Tiempo estimado: 3 horas

Descripción: Se amplió la base de datos agregando 7 nuevos restaurantes, se les añadió como un hecho su capacidad, dirección y tipo de comida. Se eliminaron las “ñ”.

Fecha: 28/05

Tiempo estimado: 1 hora

Descripción: se investigó y realizó un listado de diversos verbos, así como se avanzó en la documentación..

Fabian:

Fecha: 22/05

Tiempo estimado: 2 horas

Descripción: se comenzó la investigación sobre el BNF y de como crear los diferentes sintagmas.

Fecha: 25/05

Tiempo estimado: 4 horas

Descripción: Se comienza y se da un gran avance en la creación de los sintagmas tanto verbal como nominal.

Fecha: 04/06

Tiempo estimado: 1 hora

Descripción: en conjunto con Kevin se acopla en BNF con las reglas.