

Instituto Tecnológico de Costa Rica

Escuela de Computación

Área Académica Ingeniería en Computadores

Curso: CE-3104 Lenguajes, Compiladores e Intérpretes

Grupo #2

Prof. Marco Rivera

Tarea #2: Call Center Log

Grupo de Trabajo #9

Estudiantes:

- Dylan Garbanzo Fallas
- Irene Garzona Moya
- Alejandra Rodríguez Castro

I Semestre

2023

- **Algoritmos de solución desarrollados**

Los sistemas expertos son programas de computadora diseñados para imitar el comportamiento de un experto humano en un campo específico. Según el artículo "Expert Systems: History and Future" publicado en la revista científica IEEE Expert, "los sistemas expertos son programas de computadora que utilizan el conocimiento y la experiencia de un experto humano para tomar decisiones y realizar tareas complejas en un dominio específico".

Los sistemas expertos son utilizados en una amplia variedad de aplicaciones, desde la medicina y la ingeniería hasta la planificación empresarial y la gestión del conocimiento. Los sistemas expertos son una tecnología en constante evolución, y su capacidad para imitar el razonamiento humano ha mejorado significativamente en los últimos años. Según el artículo "Expert Systems for Risk Assessment and Decision Support in Environmental Management" publicado en la revista científica Environmental Science & Technology, "los sistemas expertos pueden ser una herramienta útil para la evaluación del riesgo ambiental y la toma de decisiones, ya que pueden ayudar a los expertos en la materia a identificar y evaluar riesgos complejos y proporcionar recomendaciones precisas para la gestión ambiental".

- Diagrama (s)

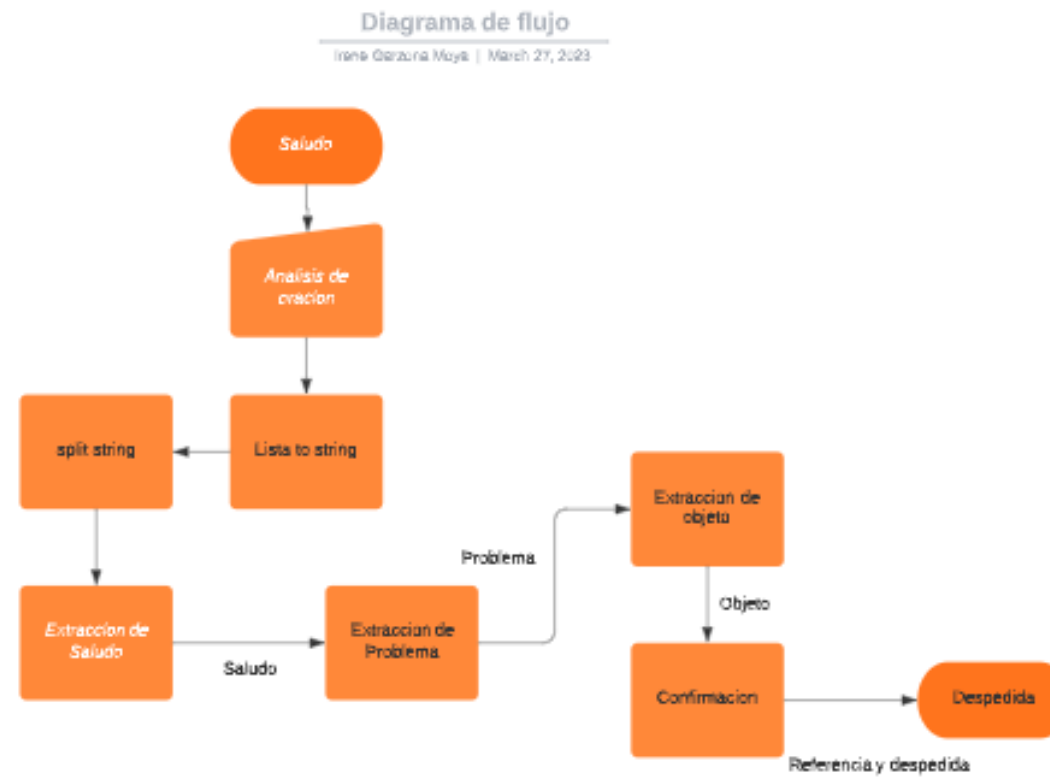


Diagrama 1. Análisis BNF. Fuente propia

- **Funciones implementadas**

El código proporciona una serie de funciones para verificar diferentes dispositivos y solucionar problemas comunes. Cada función solicita al usuario información específica sobre el estado del dispositivo y ofrece soluciones basadas en las respuestas proporcionadas.

Las funciones implementadas es la siguiente:

- verificar_dispositivo(elemento): Verifica el estado del elemento solicitado y ofrece soluciones basadas en la información proporcionada por el usuario, estas soluciones dependen del elemento.

Este código define un conjunto de predicados en Prolog para analizar una oración y extraer ciertas partes de ella. Aquí hay un resumen de las funciones:

- split_string(String, Sep, Words) - Regla para dividir una cadena en palabras usando un carácter separador Sep.
- list_to_string(List, String) - Regla para concatenar una lista de cadenas en una sola cadena.
- analyze_sentence(Sentence, Greeting, Problem, Object, Affirmation, Cause, Reference, Goodbye) - La regla principal para analizar una oración y extraer las partes de saludo, problema, objeto, afirmación, causas, referencias y despedida. Llama a otros predicados auxiliares para extraer estas partes.
- extract_greeting(Words, Greeting) - Regla auxiliar para extraer la parte de saludo de la lista de palabras.
- extract_goodbye(Words, Goodbye) - Regla auxiliar para extraer la parte de despedida de la lista de palabras.
- extract_problem(Words, Problem) - Regla auxiliar para extraer la parte del problema de la lista de palabras.
- extract_object(Words, Object) - Regla auxiliar para extraer la parte de objeto de la lista de palabras.
- extract_affirmation(Words, Affirmation) - Regla auxiliar para extraer la parte de afirmación de la lista de palabras.
- verificar_dispositivo(Objeto) - Regla para filtrar el camino en la base y dar las preguntas con respecto al dispositivo solicitado.
- verificar_tipo_mouse(Objeto) - Regla para diferenciar si se solicita ayuda con un mouse inalámbrico o de cable.
- verificar "nombre objeto" "número" (Respuesta) - Diferentes reglas utilizadas para navegar por las preguntas según el dispositivo filtrado por la regla "verificar_dispositivo".
- posibles_causas(Objeto) - Regla que da las diferentes causas por las que un objeto no funciona.

- verificar_referencia(Objeto) - Regla que analiza cual es el proveedor o modelo del dispositivo o software del cual se está solicitando una referencia.
- proporcionar_referencia(Objeto) regla que da la referencia según el proveedor dado por “verificar_referencia”.

- **Ejemplificación de estructuras de datos desarrolladas.**

Las estructuras de datos utilizadas en este código son principalmente listas, que se utilizan para almacenar las palabras de una oración, y cadenas de caracteres, que se utilizan para concatenar las palabras de las listas.

En particular, la estructura de lista se utiliza en las funciones `extract_greeting`, `extract_goodbye`, `extract_problem`, `extract_object`, y `extract_affirmation` para extraer ciertas palabras de una lista de palabras. Además, la estructura de cadena de caracteres se utiliza en la función `list_to_string` para concatenar las palabras de una lista y formar una cadena de caracteres.

- **Problemas sin solución**

Uno no se puede salir del programa en cualquier momento, únicamente detecta la despedida al inicio o cuando pregunta si puede ayudar con otra cosa.

- **Problemas encontrados**

Si uno está probando el programa varias veces seguidas, muchas veces el programa no va a terminar o cortar el procedimiento que usted estaba haciendo anteriormente, entonces incluso si lo compila y lo consulta nuevamente, tiene que terminar las consultas que estaba realizando anteriormente, sino le va a tirar un error y lo puede llegar a confundir.

- **Plan de Actividades**

- Cronograma:

Actividad		Encargados(as)	15/3/2023	16/3/2023	21/3/2023	22/3/2023	23/3/2023	24/3/2023	25/3/2023	26/3/2023	27/3/2023
Leer documento instructivo de la tarea		Todos									
Trabajo Escrito		Todos									
Sistema Experto (SE)	Confeccion de conocimiento sobre los problemas	Dylan									
		Irene									
	Confección de reglas de verificación y filtrado	Dylan									
Gramática de libre contexto (BNF)	Reconocimiento de palabras	Alejandra									
	Análisis de Oraciones	Irene									
		Alejandra									
	Base de Datos	Alejandra									
Unificar trabajo final		Todos									
Revisión de bugs y detalles		Todos									

- Bitácora

- 23-03-2023

Miembros:

- Alejandra Rodriguez Castro
 - Dylan Garbanzo Fallas
 - Irene Garzona Moya

Actividades realizadas:

La estudiante Alejandra Rodriguez inicia con la confección de la base de datos para el BNF y la regla para obtener todas las partes necesarias de la oración, esta última no queda funcionando correctamente.

El estudiante Dylan Garbanzo, inicia la confección de las reglas para el SE.

La estudiante Irene Garzona inicia la confección de la base de la documentación externa del trabajo, además de apoyar con ideas para las reglas que necesita el SE.

- 24-03-2023

Miembros:

- Alejandra Rodriguez Castro
 - Dylan Garbanzo Fallas
 - Irene Garzona Moya

Actividades realizadas:

La estudiante Alejandra Rodriguez logra la funcionalidad de la regla para obtener todas las pautas necesarias de la oración.

El estudiante Dylan Garbanzo empieza a confeccionar las filtración y verificación de las reglas para la consulta de preguntas que él SE le hará al usuario.

Todos los estudiantes se reúnen para especificar la base de datos y todas las palabras que consideran necesarias para el funcionamiento adecuado del SE.

- 26-03-2023

Miembros:

- Alejandra Rodriguez Castro

Actividades realizadas:

La estudiante Alejandra Rodriguez crea un tercer archivo 'principal' donde se va a conectar tanto el SE como el BNF, además crea las reglas para el funcionamiento adecuado del sistema. También se arreglan bugs en el SE, se hace funcional el poder proporcionar las referencias y las causas de los problemas, finalmente agrega las últimas palabras claves necesarias en la base de datos para su completo funcionamiento.

- 11-03-2023

Miembros

- Irene Garzona Moya
 - Dylan Garbanzo Fallas
 - Alejandra Rodriguez Castro

Actividades realizadas y planes pendientes:
Últimas revisiones y entrega del trabajo.

- **Conclusiones**

Prolog es un lenguaje poderoso para desarrollar sistemas expertos. Las capacidades procesamiento de lenguaje natural de Prolog, su razonamiento basado en reglas y capacidad de trabajar con estructuras de datos complejas lo convierte en un lenguaje adecuado para crear sistemas expertos. También resaltar que tanto Prolog como los sistemas expertos dependen de la conexión de patrones y la inferencia.

Los sistemas expertos proporcionan beneficios significativos a diversas áreas de trabajo. Pueden dar una forma de capturar y almacenar conocimiento y datos al que pueden acceder personas con poco conocimiento en la programación para tomar decisiones al usar el sistema. Mejora la eficiencia y crea mayor productividad.

Los sistemas expertos tienen limitaciones que tomar en cuenta, como el hecho que están limitados por el conocimiento y experiencia de sus desarrolladores, así como manejar situaciones nuevas e inesperadas. Requieren una inversión significativa de tiempo y recursos para mantenerlos vigentes.

En Prolog, al utilizar variables no se pueden sobrescribir (cambiar de valor), dos veces dentro de la misma regla, ya que por la naturaleza de Prolog, cada variable se toma como un solo valor.

El sistema de Debug de Prolog muchas veces es confuso y poco intuitivo.

Cuando uno va bajando el código, el puntero donde estaba escribiendo no se queda estático donde lo dejó la última vez que estaba escribiendo.

- **Recomendaciones**

Si está interesado en desarrollar sistemas expertos, se recomienda que aprenda Prolog. Este lenguaje tiene muchos recursos disponibles en línea e impresos para aprender. También se recomienda practicar el crear sistemas expertos para generar experiencia y desarrollar habilidad en esta área.

Si una organización está considerando el uso de sistemas expertos, se recomienda que se evalúe con cautela sus necesidades y capacidades antes de implementar uno. Primero se debe identificar el problema, y adquirir por medio de investigación todo el conocimiento técnico para su desarrollo y mantenimiento. También considerar el costo y beneficios de mantenerlo en pie.

Se debe considerar que en el uso de sistemas expertos tiene muchas limitaciones, ya que depende del conocimiento del desarrollador y capacidad de manejar situaciones nuevas. Considere agregar al proceso de desarrollo el uso de técnicas de aprendizaje automático para mejorar la capacidad y reducir el tiempo de consulta. Y también se recomienda revisiones periódicas para su mantenimiento.

Al utilizar variables, se debe mencionar únicamente dos veces en cada regla, una para definirla y otra para ser utilizada como parámetro en otra regla.

Verse un vídeo previamente de cómo utilizar el sistema de Debug de Prolog.

Siempre revisar dónde está el puntero antes de escribir, ya que puede llegar a romper el código sin darse cuenta.

Referencias bibliográficas

Jackson, P. (1999). Expert Systems: History and Future. IEEE Expert, 14(3), 23-29.
<https://doi.org/10.1109/64.772281>

Mazzarotta, B., & Brescia, V. M. (2019). Expert Systems for Risk Assessment and Decision Support in Environmental Management. Journal of Environmental Science and Engineering, 63(4), 298-306. <https://doi.org/10.17265/2162-5298/2019.04.005>

Monferrer, T. E., Toledo, F., & Pacheco, J. (2001). El lenguaje de programación Prolog. Valencia.

SWI-Prolog documentation. (n.d.). Retrieved March 26, 2023, from
<https://www.swi-prolog.org/pldoc/index.html>