

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Inteligencia Artificial 1
Vacaciones de Junio 2023

Práctica 2
Creación de Sistema Experto en Prolog

Carlos Ojani Ng Valladares
201801434

INTRODUCCIÓN

Para la práctica 2 "Creación de Sistema Experto en Prolog" realizada para el laboratorio de Inteligencia Artificial 1, se trabajó para una solución para la selección de una mejor opción de hospedaje utilizando Prolog, el cuál es un lenguaje de programación lógico. El propósito de la práctica es ofrecerle al cliente una solución en formato conversacional utilizando Prolog para que, con base a preguntas, vamos almacenando la información que irá introduciendo y al final poder generarle al cliente una conclusión basada en dicha información.

La información en la cuál compara el programa con las respuestas del cliente viene de una base de datos en formato Excel. Con base a ello, se generaron las reglas suficientes para poder inferir una respuesta razonable y coherente.

OBJETIVOS

Objetivo General

- Determinar la solución de dicha problemática con base a programación lógica.

Objetivos Específicos

- Aplicar conocimientos adquiridos con el lenguaje de programación lógico Prolog.
- Determinar la mejor opción para hacer uso de la información proporcionada por el usuario.
- Determinar las ventajas de la programación predictiva con base a la solución brindada al usuario.

USO DE LA SOLUCIÓN

A continuación, se detalla cómo se presenta la solución ante el usuario y como fue generada la solución mediante el lenguaje de programación lógico Prolog.

Primeramente, el usuario cuenta con una interfaz amigable para el sistema experto en hoteles, el cuál consta de una serie de consultas, a modo de realizar el análisis de una forma conversacional. Al inicio, el usuario verá el siguiente mensaje, seguido de algunas opciones de interés.

```
?- inicio.  
==== Bienvenido al Sistema Experto en Hoteles ====  
Ingrese el criterio de seleccion que desea  
1. Presupuesto  
2. Clima  
3. Idioma/lengua  
4. Cantidad Estrellas  
5. Reportes  
Elija una opcion  
|:
```

De la mano del sistema experto, el usuario podrá obtener recomendaciones bajo ciertos criterios. En este caso por los siguientes:

- Presupuesto.
- Clima.
- Idioma/lengua local.
- Cantidad de estrellas del Hotel.

Adicional a ello, el usuario tendrá la opción de visualizar ciertos reportes, más adelante se detalla cómo se llevaron a cabo.

A continuación, se muestra cómo se visualiza la recolección de información para un futuro análisis (figura del lado derecho).

Como se puede observar, el usuario ha escogido realizar la búsqueda de un hotel que se acomode a sus necesidades con bases a su presupuesto como principal criterio. Posteriormente, siguen una secuencia de preguntas que van a definir nuestra conclusión para el usuario. Ya que influye esta información con respecto a los costos, distancia, calidad del hotel, lugar y la cantidad de personas a hospedarse.

Has escogido por presupuesto
 Ingresar tu presupuesto en quetzales
 |: 6500.

Gracias escogiste un plan caro
 Ingresar el tipo de clima que prefieras
 -. calor
 -. frio
 -. templado
 -. tropical
 |: calor.

Cantidad de distancia a recorrer en km
 |: 100.

Escoge la cantidad de estrallas del hotel
 |: 4.

Escoge el tipo de habitacion simple/doble
 |: simple.

Indique cuantos dias se hospedara
 |: 2.

Tiempos de comida a tomar 0 a 3
 |: 3.

Numero de personas a hospedarse
 |: 2.

Indique si llevara vehiculo s/n
 |: s.

Para el presupuesto se definió cierto rango de dinero para llevar a cabo la búsqueda, sigue el siguiente criterio:

Barato	Moderado	Alto
2500 <=	2500 < x < 5000	< 5000

Un presupuesto barato va a ser menos de Q2,500, uno moderado entre Q2,500 y Q5,000, y uno alto por arriba de los Q5,000.

Con base a este criterio, realizamos las siguientes preguntas al usuario.

Preguntas	Barato	Moderado	Alto
Cantidad estrellas			x
Tipo habitación (Simple/Doble)	x	x	x
Clima		x	x
Distancia (inicia km. 0)	x	x	x
¿Lleva vehículo? (12 Q/km)	x	x	x
Días estadía	x	x	x
Número de personas hospedar	x	x	x
Tiempos de comida (0 a 3)	x	x	x

Dependiendo del presupuesto del usuario se tendrán valores por defecto, o se le dará la libertad de escoger (Por ejemplo, la calidad del hotel y el clima).

Para los demás criterios de búsqueda utilizamos las mismas preguntas y se varía qué se quiere preguntar, ya que no se necesita segmentar el análisis, se muestra un ejemplo para una búsqueda por **clima**:

Preguntas (Búsqueda por Clima)	Clima
Cantidad estrellas	
Tipo habitación (Simple/Doble)	x
Clima	
Distancia (inicia km. 0)	x
¿Lleva vehículo? (12 Q/km)	x
Días estadía	x
Número de personas hospedar	x
Tiempos de comida (0 a 3)	x

DETALLE DE LA SOLUCIÓN

Como se mencionó anteriormente, se utilizó Prolog para realizar la solución. Se utiliza una base de datos basada en Excel, a partir de allí, extraemos la información que vamos a utilizar para posteriormente generar los hechos. El conocimiento se representa mediante cláusulas. A esto denominamos los hechos o reglas. Los hechos de la base de datos se manejan de la siguiente forma:

```
departamento(1, peten, 8, espanol, tropical, 600).
hotel(1, 'Camino Real', '14 calle zona 10', 4, 100, 200, 50, 14, 7).
cliente(1, jose, morales, guatemala, 27, soltero, vacaciones).
registro(1, 10, 30, '10/01/2021', 4, 3).
trabajador(1, 'Jose Hernandez', 'Chef', 1).
```

De la siguiente forma se lleva a cabo el menú de selección. Así es cómo se manejan las condicionales en Prolog. De la misma forma para tomar decisiones se utiliza esta sintaxis.

```
seleccion(OPCION):-
    OPCION == 1 -> presupuesto();
    OPCION == 2 -> clima();
    OPCION == 3 -> idioma();
    OPCION == 4 -> estrellas();
    OPCION == 5 -> reportes().

% ---- BUSQUEDA POR PRESUPUESTO ----
presupuesto():- write("Has escogido por presupuesto"),
    nl,
    write("Ingresa tu presupuesto en quetzales"),
    nl,
    read(PRESUPUESTO),
    nl,
    analisispresupuesto(PRESUPUESTO).
```

Para obtener la información del usuario, se emula una conversación, este almacena la información que el usuario ingresa por pregunta, por medio de la funcionalidad **read** de Prolog. Posteriormente, se envía dicha información a un análisis que involucra cálculos y condicionales para, finalmente, llegar a una conclusión.

```
barato(PRESUPUESTO):- write("Gracias escogiste un plan barato"),
    nl,
    CLIMA = 'calor',
    nl,
    write('Cantidad de distancia a recorrer en km'),
    nl,
    read(DISTANCIA),
    nl,
    TIPOHABITACION = 'simple',
    nl,
    write('Indique cuantos dias se hospedara'),
    nl,
    read(DIASHOSPEDAJE),
    nl,
    write('Tiempos de comida a tomar 0 a 3'),
    nl,
    read(TIEMPOSCOMIDA),
    nl,
    write('Numero de personas a hospedarse'),
    nl,
    read(NOPERSONAS),
    nl,
    write('Indique si llevara vehiculo s/n'),
    nl,
    read(CARRO),
    CANTIDADESTRELLAS is 2,
    analisisbarato(PRESUPUESTO, CLIMA, DISTANCIA, CANTIDADESTRELLAS, TIPOHABITACION,
    DIASHOSPEDAJE, TIEMPOSCOMIDA, NOPERSONAS, CARRO),
    nl.
```


Para cada situación (o camino) se presenta una inferencia. Es decir, luego del proceso de recolección de información, se muestran las conclusiones. En este caso, se muestra una inferencia por presupuesto, el usuario posee carro, se brinda el número de días de estadía, el tipo de habitación, el hotel, etc. Al final se muestra un total estimado con base al presupuesto y cuánto tendría restante. Esto se realiza de la siguiente manera:

```
inferencia1(PRESUPUESTO,
NombreHotel,NombreDepartamento,Direccion,CostoGasolina,CostoHabitacion,CostoComida,Sum
atoria,Restante):-
    nl,
    write('===== Conclusion
(Presupuesto)====='),
    nl,
    format('Su presupuesto de Q. ~a se acomoda a las siguientes opciones...
\n    Hotel: ~a
\n    Departamento: ~a
\n    Direccion del Hotel: ~a
\n    Costo de Habitacion(es): ~a
\n    Costo de Gasolina: ~a
\n    Costo de Alimentacion: ~a
\n    Total Estimado: ~a
\n    Presupuesto Restante: ~a',[PRESUPUESTO,
NombreHotel,NombreDepartamento,Direccion,CostoHabitacion,
CostoGasolina,CostoComida,Sumatoria,Restante]),
    nl,

write('====='),
    fail.
```

Para los reportes, se emula la funcionalidad de realizar consultas a los hechos, que vienen siendo las tablas de nuestra base de datos. Con base a ciertas restricciones definidas en el enunciado vamos a filtrar la información. Esto se lleva a cabo de la siguiente manera:

```
reporte1():-
    % --Nombre y nacionalidad de clientes nacidos en Europa con opiniones mayores de 5
    write('== Nombre y nacionalidad de clientes nacidos en Europa con opiniones
mayores de 5 =='),
    nl,
    cliente(IDCliente, NombreCliente, ApellidoCliente, PaisCliente, _, _, _),
    registro(_, IDCliente, _, _,_, Opinion),
    Opinion > 4,
    (PaisCliente=='italia'; PaisCliente=='espanol'; PaisCliente=='espanola';
PaisCliente=='inglaterra'; PaisCliente=='holanda'),
    mostrar_reporte_1(NombreCliente, ApellidoCliente, PaisCliente),
    reportes().

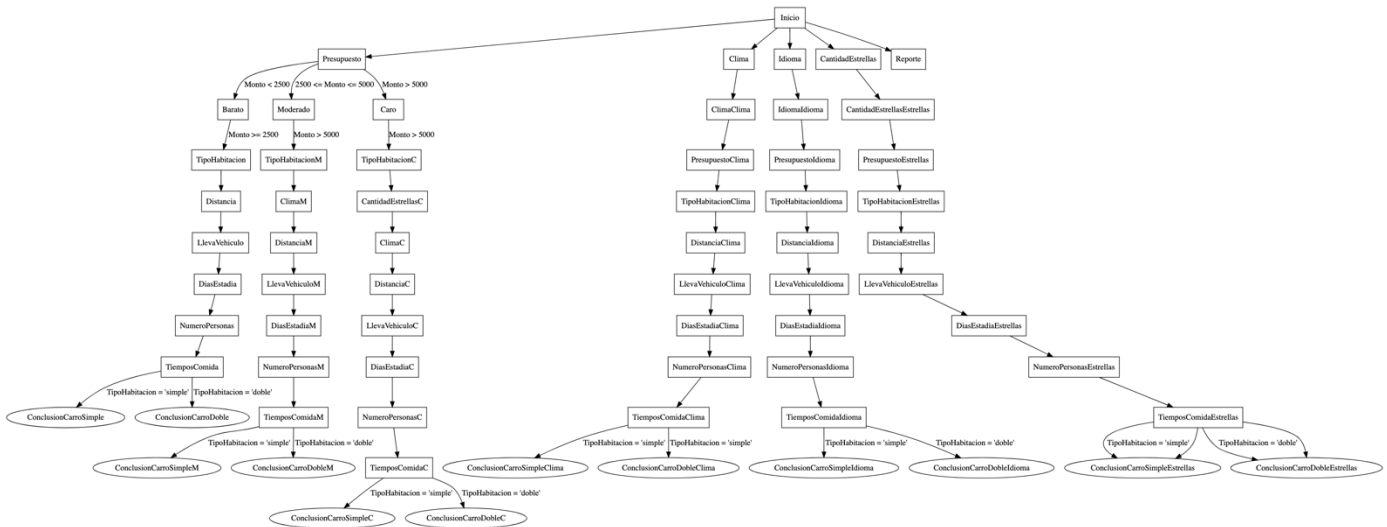
mostrar_reporte_1(NombreCliente, ApellidoCliente, PaisCliente):-
    format('~a ~a | ~a', [NombreCliente, ApellidoCliente, PaisCliente]), nl, fail,
    true.
```

CONCLUSIÓN

Acorde a lo trabajado en la práctica, se concluye que la programación lógica tiene muchas aplicaciones hoy en día. La lógica de primer orden que se ofrece al construir soluciones con Prolog sirve como base para solidificar los conceptos de inteligencia artificial, tales como Machine Learning y el desarrollo de dispositivos capaces de razonar con base a la información con la cual se les alimenta.

Con respecto al lenguaje, con ayuda de la programación lógica es más sencillo escribir código sin conocer del todo el lenguaje. El hecho de que las reglas o especificaciones sean más próximos a entenderse gracias a que se basan en lenguajes formales, lo hace más fácil de entender y aprender. Por otro lado, se concluye también que los motores de inferencia tienden a ser bastante limitados. Esto se ve claramente cuando se retorna un "false" al momento de no cumplir con alguna condición.

DIAGRAMA DE ENCADENAMIENTO



A partir de un grupo de múltiples reglas, o hechos, que se definieron durante la práctica se llevó a cabo una representación gráfica de un encadenamiento. En específico, hacia adelante.

Un encadenamiento hacia adelante realiza una búsqueda o sigue un camino a partir de una problemática, en nuestro caso "búsqueda de un hotel", en el cual se llega a una conclusión con base a la información que el usuario proporciona. La forma de la gráfica describe un encadenamiento hacia adelante, el hecho de ser una secuencia de enlaces entre reglas. Desde los hechos hasta las conclusiones.

En el diagrama los hechos se representan mediante nodos rectangulares y las conclusiones mediante óvalos, que vienen siendo los nodos finales.