UNIVERSIDAD DE SANTIAGO DE CHILE

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INFORMÁTICA

INFORME CHATBOT

Programación en Prolog

Nombre Alumno:	Francisco Guajardo
Profesores:	Daniel Gacitúa
Ayudantes:	Javier Vasquez
Fecha de Entrega:	21 – 05 - 2018

Tabla de contenido

Introducción

En la actualidad muchas empresas y compañías de bienes y servicios implementan en algún momento la entidad chatbot.
El chatbot es una entidad conversacional artificial. Son básicamente programas computacionales que son capaces de mantener o generar una conversación fluida con un usuario.
En este laboratorio se nos encomendó la tarea de realizar un chatbot que logre los estándares de este, basando nuestro algoritmo en el paradigma lógico usando el lenguaje Prolog.
Este chatbot esta diseñado para mantener una conversación corta con un usuario que desea comprar una película, juego o manga de la tienda X, indicando su precio y pidiendo forma de pago al usuario.
Este programa funciona en SWI-Prolog, interprete que nos facilitara la tarea de generar sentencias y correr nuestro programa.

Marco teórico

- **Lenguaje de programación:** Un lenguaje de programación es un vocabulario, un conjunto de reglas gramaticales que tiene como objetivo dar instrucciones al computador para que este realice tareas especificas.

- **Prolog:** Es un lenguaje de programación basado en el paradigma lógico el cual a través de hechos y reglas intenta, no guiar a la computadora hacia la respuesta, sino delimitar su campo de búsqueda.

- **Paradigma de programación:** Es un estilo de programar, un camino a seguir, una filosofía para diseñar soluciones a problemas relacionados con la programación.

- **Recursión:** Es el proceso de definir una solución a un problema en términos de ella misma. Teniendo elementos mínimos como condición de parada y llamada recursiva. Existen diferentes tipos de recursión, como la recursión natural, recursión arbórea y recursión de cola.

- **Algoritmo:** Un algoritmo es un procedimiento esquemático que comprende un conjunto de pasos secuenciales ordenados, para realizar una actividad especifica

- **TDA:** Un tipo de dato abstracto consiste en la representación de un objeto, proceso o cualquier elemento real o imaginario para que sea operado por la maquina. Para formar un TDA tenemos que tener en consideración su arquitectura de seis elementos los cuales son Operadores, modificadores, selectores, funciones de pertenencia, constructores y destructores. Como mínimo de esa arquitectura se requiere una función constructora, función selectora y funciones para operar el TDA.

- **Hecho:** Un hecho en el Paradigma Lógico es una relación entre objetos.

- **Regla:** Una Regla en el Paradigma Lógico se compone de dos partes, la cabeza y el cuerpo: cabeza:- cuerpo. El cuerpo esta conformado de uno o varios hechos. (la cabeza es verdad si el cuerpo es verdad).

Descripción problema

El problema que nos fue planteado es el de crear un chatbot basado en el paradigma lógico usando el lenguaje de programación Prolog.

Para este problema había que diseñar un contexto o temática general para el chatbot, el cual se usaría para guiar y definir las actitudes que este va a tomar con las respuestas que el usuario le entregue, además este contexto se deberá de mantener durante todos los laboratorios.

El chatbot debería ser capaz de poder responder al usuario de forma empática para lograr una mayor conexión con el, este a su vez deberá de contar con líneas de mensajes protocolares y de estilo libre.

Un ejemplo de conversación podría ser el siguiente:

- Chatbot: ¿Cual de nuestros servicios desea contratar?
- Usuario: cable
- Chatbot: Registrado, ¿Qué hay de internet?
- Usuario: no
- Chatbot: Entendido, ¿y telefonía?
- Usuario: si
- Chatbot: Perfecto, ya he registrado sus servicios

(ejemplo rescatado del enunciado del problema desde USACH Virtual)

Así también, se plantea como problema el transformar nuestro Log o conversación a un formato mas entendible para el usuario, en este caso se solicita que este sea del tipo String.

La forma de programar, en comparación con el laboratorio 1 es la implementación de un nuevo lenguaje, Prolog, en un nuevo ambiente de trabajo, SWI-Prolog, pensando la solución en un nuevo paradigma, utilizar la lógica y todos sus derivados es el desafío para este proyecto.

Paradigma Lógico

El paradigma lógico tiene como característica principal la aplicación de las reglas de la lógica para inferir conclusiones a partir de datos. Conociendo la información y las condiciones del problema, la ejecución de un programa consiste en la búsqueda de un objetivo dentro de las declaraciones realizadas.

Esta forma de tratamiento de la información permite pensar la existencia de “programas inteligentes” que puedan responder, no por tener en la base de datos todos los conocimientos, sino por poder inferirlos a través de la deducción.

Este paradigma, en el caso de Prolog, posee como unidad de programación los hechos y reglas.

Algoritmos de control

Al hacer la comparación con el paradigma imperativo, en este uno programa con un enfoque en lo que se quiere alcanzar y como las metas serán logradas, en cambio en le paradigma lógico, el enfoque va a solo tratar la lógica, el control de algoritmo se le deja a la maquina abstracta.

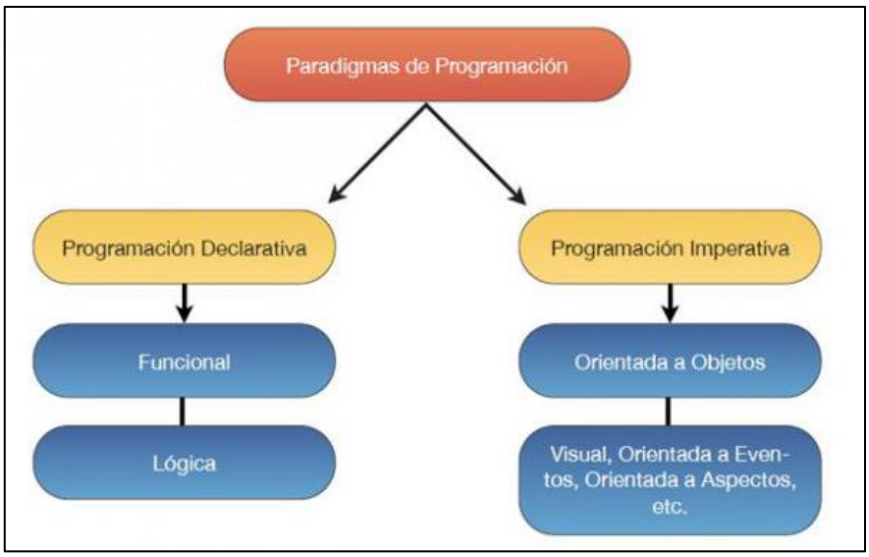

Ilustración 1, paradigmas de programación; mapa conceptual

Análisis del problema

Nos solicitan realizar nuestro chatbot en el ambiente de programación SWI-Prolog, por lo cual la notación a usar debe de corresponder a la implementada en este programa, sin uso de librerías externas ni cosas ajenas a este ambiente.

Como requerimientos nos solicitan trabajar el chatbot en base a una estructura con distintos parámetros, con la materia vista en clases se puede asumir que este chatbot será una estructura de datos abstracta, TDA para simplificar, la cual constara de una arquitectura de 6 niveles.

También se solicita poder manejar un registro histórico de conversación, sin el uso de variables, ya que se trabaja en paradigma lógico, al no poder usar variables actualizables nos queda la opción de solicitar al usuario manualmente ingresar el log obtenido en la conversación a la siguiente función o predicado para poder seguir con el transcurso de la conversación.

La Seed en el programa se supone nos entrega el estado de animo, o las actitudes que debe de tomar el chatbot, con diálogos ya predichos, en este caso basados en prolog y la programación lógica, se tiene que crear, una IA básica que siga un comportamiento lógico.

En el problema, la conversación fluye gracias a tres funciones, las cuales son beginDialog, sendMessage, endDialog. Dos de estas se pueden trabajar de una manera similar, comenzar y terminar, puesto que ambas solo toman el registro de conversación histórico y le agregan un identificador de inicio y un identificador de fin de conversación, respectivamente. SendMessage es mas compleja de tratar puesto que se necesita realizar un predicado dinámico que siga el conducto de una conversación. El trabajo de esta función se basará en el correcto uso del TDA chatbot, el cual consta de una cantidad de hechos que delimitan el conocimiento que este poseerá.

La función test, no implica gran desafío, puesto que es un recopilatorio de todas las funciones y predicados anteriores aplicados a un nuevo TDA, que seria el user, una serie de mensajes ya predichos que terminan por entregar un log con la conversación completa.

El desafío fue en logToStr, puesto que si no se tenia un buen manejo de la recursión o buen manejo de los recursos de prolog, generaría un problema el recorrer una lista para salvar su contenido en un string.

En el informe se explicara los alcances que posee el programa y sus limitaciones.

Diseño y aspectos de implementación de la solución

Para la creación de la entidad chatbot se pensó en el diseño de una base de datos en forma de TDA, de la forma `chatbot(seed,[instancia,Stringrespuestas]).` Teniendo una función constructora, selectora y de pertenencia.

De forma similar se pensó en realizar una pequeña base de datos de los productos ofrecidos por la tienda para tener un registro del precio de estos y si existe stock para hacer un poco mas dinámica la conversación, esta base de datos esta escrita de la forma `stock(Producto,Tipo,Nombre,Precio).` Donde Tipo, Nombre y precio son del tipo string.

Para el desarrollo de algunas funciones se usaron predicados de manejo de listas/string y fecha rescatados de la documentación oficial de SWI-Prolog y algunas funciones rescatadas de las PPTs de la clase, por lo que debería de estar todo en orden con respecto a la implementación del código.

Como el enunciado lo indicaba, el programa corre en el interprete SWI-Prolog sin arrojar error o warnings con relación al uso de variables.

El manejo del log, este se representa como una lista de listas, [[x],[y]] donde X se compone de dos elementos, un string con la fecha y un string con, en el caso del sendMessage, con el mensaje entregado por el usuario y/o por el chatbot; en el caso de beginDialog y endDialog posee un string con la fecha y otro con el delimitador de inicio o fin, como corresponde.

```
OutputLog = [["BeginDialog", "16:53-21/5/2018"], ["Chatbot: Bienvenido a la tienda, que pelicula desea adquirir hoy?",
"16:53-21/5/2018"], ["doctor strange", "16:53-21/5/2018"], ["Chatbot: Desea adquirir la version bd o dvd?",
"16:53-21/5/2018"], ["bd", "16:53-21/5/2018"], ["Chatbot: Producto registrado, medio de pago?", "16:53-21/5/2018"],
["tarjeta", "16:53-21/5/2018"], ["Chatbot: El precio es de $25090 desea algo mas?", "16:53-21/5/2018"], ["No, ya tengo lo
que necesito", "16:53-21/5/2018"], ["Chatbot: su pedido seria (doctor strange/bd) pagando $25090 sin mas que agregar,
finalice el asistente.", "16:53-21/5/2018"], ["Chatbot: Gracias, disfrute la pelicula y hasta la proxima", "16:53-21/5/2018"],
["EndDialog", "16:53-21/5/2018"]] .
```

Ilustración 2, ejemplo de Log

El Log debe de ser entregado por el usuario, de la forma “tradicional”, tiene que copiar y pegar OutputLog de la función previa para poder seguir con la conversación, esto limitado por no existir variables actualizables o una mayor habilidad para crear hechos durante la marcha del programa, si es que fuese posible.

Resultados

Utilizando la función test, para hacer una rápida y limpia ejecución de nuestro programa obtenemos lo siguiente:

```
?- test(user1,Chatbot,[],1,OutputLog).
OutputLog = [["BeginDialog", "19:58-21/5/2018"], ["Chatbot: Bienvenido a la tienda, que pelicula desea adquirir hoy?",
"19:58-21/5/2018"], ["doctor strange", "19:58-21/5/2018"], ["Chatbot: Desea adquirir la version bd o dvd?",
"19:58-21/5/2018"], ["bd", "19:58-21/5/2018"], ["Chatbot: Producto registrado, medio de pago?", "19:58-21/5/2018"],
["tarjeta", "19:58-21/5/2018"], ["Chatbot: El precio es de $25090 desea algo mas?", "19:58-21/5/2018"], ["No, ya tengo lo
que necesito", "19:58-21/5/2018"], ["Chatbot: su pedido seria (doctor strange/bd) pagando $25090 sin mas que agregar,
finalice el asistente.", "19:58-21/5/2018"], ["Chatbot: Gracias, disfrute la pelicula y hasta la proxima", "19:58-21/5/2018"],
["EndDialog", "19:58-21/5/2018"]]
false.

?- test(user2,Chatbot,[],2,OutputLog).
OutputLog = [["BeginDialog", "19:58-21/5/2018"], ["Chatbot: Bienvenido a la tienda, que juego desea adquirir hoy?",
"19:58-21/5/2018"], ["overwatch", "19:58-21/5/2018"], ["Chatbot: Para que consola lo desea? (ps4,xone,switch)",
"19:58-21/5/2018"], ["ps4", "19:58-21/5/2018"], ["Chatbot: Producto registrado, medio de pago?", "19:58-21/5/2018"],
["efectivo", "19:58-21/5/2018"], ["Chatbot: El precio es de $37990 desea algo mas?", "19:58-21/5/2018"], ["No, ya tengo lo
que necesito", "19:58-21/5/2018"], ["Chatbot: su pedido seria (overwatch/ps4) pagando $37990 sin mas que agregar,
finalice el asistente.", "19:58-21/5/2018"], ["Chatbot: Gracias, disfrute su juego y hasta la proxima", "19:58-21/5/2018"],
["EndDialog", "19:58-21/5/2018"]]
false.

?- test(user3,Chatbot,[],3,OutputLog).
OutputLog = [["BeginDialog", "19:59-21/5/2018"], ["Chatbot: Bienvenido a la tienda, que manga desea adquirir hoy?",
"19:59-21/5/2018"], ["attack on titan", "19:59-21/5/2018"], ["Chatbot: En que idioma lo desea, espaniol (es) o ingles (en)?",
"19:59-21/5/2018"], ["en", "19:59-21/5/2018"], ["Chatbot: Producto registrado, medio de pago?", "19:59-21/5/2018"],
["cheques", "19:59-21/5/2018"], ["Chatbot: El precio es de $1 desea algo mas?", "19:59-21/5/2018"], ["No, ya tengo lo que
necesito", "19:59-21/5/2018"], ["Chatbot: su pedido seria (attack on titan/en) pagando $1 sin mas que agregar, finalice el
asistente.", "19:59-21/5/2018"], ["Chatbot: Gracias, disfrute su lectura y hasta la proxima", "19:59-21/5/2018"],
["EndDialog", "19:59-21/5/2018"]]
false.
```

Ilustración 3, resultado test/5 en SWI-Prolog

Lo cual nos indica los limites de nuestro chatbot, el cual solo puede pedir que producto comprar, alguna cualidad de este y que medio de pago el usuario escogerá para comprar, luego se despide.
En si es muy limitado puesto que solo se logra una conversación de 5 a 6 líneas, este no es el resultado esperado, pero tampoco se consideraría un producto final.

Tampoco se logro el resultado esperado en logToStr, entregando lo siguiente:

```
% /Users/franciscog/Desktop/Programacion/Paradigmas/Prolog/chatbot compiled 0.01 sec, 1 clauses
?- logToStr([["BeginDialog", "23:33-20/5/2018"], ["Chatbot: Bienvenido a la tienda, que pelicula desea adquirir hoy?",
"23:33-20/5/2018"], ["attack on titan", "23:33-20/5/2018"], ["Chatbot: Desea adquirir la version bd o dvd?",
"23:33-20/5/2018"], ["en", "23:33-20/5/2018"], ["Chatbot: Producto registrado, medio de pago?", "23:33-20/5/2018"],
["cheques", "23:33-20/5/2018"], ["Chatbot: El precio es de $1 desea algo mas?", "23:33-20/5/2018"], ["No, ya tengo lo que
necesito", "23:33-20/5/2018"], ["Chatbot: su pedido seria (attack on titan/en) pagando $1 sin mas que agregar, finalice el
asistente.", "23:33-20/5/2018"], ["Chatbot: Gracias, disfrute la pelicula y hasta la proxima", "23:33-20/5/2018"],
["EndDialog", "23:33-20/5/2018"]],Meh).
ERROR: Arguments are not sufficiently instantiated
ERROR: In:
ERROR:    [9] string_concat(_3866,"23:33-20/5/2018 || BeginDialog",_3870)
ERROR:    [8] logToStr([["BeginDialog"|...],...|...],_3896) at /Users/franciscog/Desktop/Programacion/Paradigmas/Prolog/
chatbot.pl:275
ERROR:    [7] <user>
```

Lo cual indica el mal manejo de la recursión empleada en la función logToStr, dejando mas limitado el programa, puesto que no podrá retornar un string entendible del log para el usuario.

Conclusiones

Como el paradigma en si costo comprenderlo, al principio le había dado un enfoque imperativo a la programación, intentando simular las sentencias usadas en lenguajes donde se desarrolla un algoritmo basado en implementación de variables y esos detalles, al percatar que no se lograba hacer mucho viendo este enfoque se intento ver lo lógico de las sentencias y leer mas a fondo la documentación, logrando comprender el funcionamiento básico.

En este laboratorio, al igual que el anterior, no termino conforme con el resultado final, siento que se pudo haber implementado de mejor forma, pero la comprensión del paradigma y como dirigir la programación fue la que termino retrasando el desarrollo correcto de este laboratorio.

Comparando este paradigma con el anteriormente tratado, el paradigma funcional era mas comprensible para la programación, puesto que era menos abstracto que el actual paradigma.

Referencias

1. https://kevinldp.wordpress.com/4-paradigma-logico/
2. http://hemeroteca.abc.es/nav/Navigate.exe/hemeroteca/madrid/abc/1986/10/12/171.html
3. http://rockeromaniaco.blogspot.es/1408916012/paradigmas-de-programacion/
4. PPTs de la clase, https://www.udesantiagovirtual.cl/wp/
5. http://cs.lmu.edu/~ray/notes/paradigms/
6. https://users.dcc.uchile.cl/~bebustos/apuntes/cc30a/TDA/

Tabla de ilustraciones