

Práctica 5

Sistemas basados en reglas

Fecha de entrega: 9 de enero del 2019

El primer objetivo de esta práctica consiste en conocer las técnicas de representación de conocimiento basadas en reglas y, más concretamente, en aprender las características básicas del lenguaje CLIPS. El segundo objetivo consiste en analizar distintas posibilidades de aplicación de los sistemas de reglas al diseño de un recomendador de aplicaciones para móviles.

1. Sistema de reglas para empaquetado de artículos

Se quiere construir un sistema de reglas en CLIPS que permita empaquetar artículos en cajas de forma automática. Se partirá de un conjunto de hechos iniciales que describen los productos que tenemos que empaquetar y de las cajas disponibles en el almacén. Dispongo de un número ilimitado de cajas iguales, es decir, con la misma capacidad máxima.

PRODUCTOS	
Nombre	cadena
Tipo	Frágil/Pesado/Normal
Envuelto	Si/No
Volumen	Valor entre 0 y 200

Codificar las reglas necesarias para modelar el siguiente comportamiento:

- Cada artículo se envuelve de manera individual.
- En las cajas sólo podemos empaquetar artículos envueltos.
- Todos los artículos de la misma caja son del mismo tipo, es decir, un artículo frágil no se empaquetará en una caja que ya se ha abierto para meter artículos pesados.
- Si tenemos una caja ya empezada con algún artículo, seguiremos llenándola con artículos del mismo tipo (frágil o pesado) antes de abrir una caja nueva.
- Cada artículo y caja tienen unos volúmenes asociados, de forma que un artículo podrá empaquetarse en una caja si el volumen disponible en ésta es mayor que el volumen del artículo.

Ejecutar algunos ejemplos de prueba y comprobar cómo se modifica la memoria de trabajo.

Comenta cómo afecta el cambio de estrategia de resolución de conflictos en el resultado del sistema.

2. Sistema de reglas para recomendación de aplicaciones

El objetivo de la segunda parte de la práctica es el desarrollo de un recomendador de aplicaciones para Smartphones. El Recomendador utilizará reglas de recomendación que reflejan vuestra propia experiencia (como expertos en el dominio).

El sistema será capaz de recomendar una o varias aplicaciones en función del perfil del cliente, de sus preferencias personales y de la popularidad de las aplicaciones.

2.1. Ingeniería del conocimiento

En el campus se ha compartido un catálogo de aplicaciones del google play store (en formato csv) con distintas características que incluyen por ejemplo la versión de Android necesaria, la fecha de la última actualización, si es de pago o gratuita, el rating medio de los usuarios para esa aplicación, su categoría temática, el número de descargas, edad recomendada,

Columns			Cancel	No changes yet
A String	App	Application name		
A String	Category	Category the app belongs to ss		
# Decimal	Rating	Overall user rating of the app (as when scraped)		
# Integer	Reviews	Number of user reviews for the app (as when scraped)		
A String	Size	Size of the app (as when scraped)		
A String	Installs	Number of user downloads/installs for the app (as when scraped)		
A String	Type	Paid or Free		
A String	Price	Price of the app (as when scraped)		
A String	Content Rating	Age group the app is targeted at - Children / Mature 21+ / Adult		
A String	Genres	An app can belong to multiple genres (apart from its main category). For eg, a musical family game will belong to Music, Game, Family genres.		
📅 Date	Last Updated	Date when the app was last updated on Play Store (as when scraped)		
A String	Current Ver	Current version of the app available on Play Store (as when scraped)		
A String	Android Ver	Min required Android version (as when scraped)		

El Recomendador comienza con información del perfil de usuario que incluye sus datos, preferencias y requisitos, por ejemplo, edad, versión de Android, espacio libre en el móvil, me gustan los deportes, el arte, tengo niños pequeños,... Para evitar pedirle demasiada información de perfil al usuario, sobre todo en lo referente a sus gustos, se puede recopilar la información a través de sus 3 aplicaciones del móvil preferidas e inferir el perfil a partir de sus características: prefiere aplicaciones ligeras, gratuitas, muy populares y de temática juegos.

Según sus características las aplicaciones se pueden clasificar según distintos criterios, por ejemplo, las descargas y las estrellas permiten clasificarlas por **popularidad**. El rating debe tener en cuenta si hay pocas revisiones (no es lo mismo tener 5 estrellas si solo te tenido 5 revisiones que si tengo 5000). El tamaño de la aplicación permite clasificarlas en pesadas, normales, ligeras. El precio me permitirá clasificar las apps en gratis, baratas, caras. Las categorías permiten hacer una agrupación en niveles o se pueden usar directamente.

Tener en cuenta que una aplicación puede pertenecer a varias categorías temática. Se pueden consultar la lista completa de categorías en la base de datos: ART_AND_DESIGN, AUTO_AND_VEHICLES, BEAUTY, BOOKS_AND_REFERENCE, BUSINESS, COMICS, COMMUNICATION, DATING, EDUCATION, ENTERTAINMENT, EVENTS, FINANCE, FOOD_AND_DRINK, HEALTH_AND_FITNESS, HOUSE_AND_HOME, LIBRARIES_AND_DEMO, .. SPORTS, PARENTING, VIDEO_PLAYERS, ...

Usando reglas de recomendación hay que relacionar a los perfiles de usuario con las aplicaciones. En primer lugar, deberéis analizar cuál es la información que necesitamos sobre el perfil del cliente. Parte de la práctica es pensar cuales son las reglas de recomendación que propondrías como expertos.

Al menos existirán tres bloques de conocimiento: conocimiento sobre el cliente, conocimiento sobre las aplicaciones y conocimiento de recomendación de aplicaciones.

Los dos primeros son de tipo factual y se representarán mediante hechos en el prototipo. El último es de tipo procedimental (qué recomendar en qué casos) y se representará mediante reglas. No se trata de que haya una correspondencia directa entre clientes y aplicaciones, debería haber reglas que establecieran correspondencias intermedias entre los tipos de clientes y los tipos de aplicaciones.

2.2. Desarrollo de un prototipo en CLIPS

Una vez que tengamos el conocimiento, desarrollaremos un primer prototipo básico que no necesite operaciones de entrada/salida de datos. En este primer prototipo, los datos correspondientes a las respuestas del usuario sobre su perfil y preferencias se establecerán directamente en un deffacts, sin necesidad de realizar operaciones de lectura.

Es aconsejable que el prototipo tenga, al menos, 3 partes correspondientes a distintas etapas de la recomendación. En una primera parte se analizarían las características y gustos del usuario para determinar de qué tipo de usuario se trata y qué tipo de aplicaciones necesita. En la segunda parte las aplicaciones se clasifican en tipos. Y en una tercera parte se generarían las recomendaciones concretas para el usuario que sean adecuadas a sus necesidades.

Los resultados obtenidos por cada uno de los módulos se asertarán en la memoria de trabajo, pudiendo visualizarse mediante el uso de (facts *).

2.3. Mejoras (opcional)

Se incluirán mejoras en el prototipo:

- El cliente establece una consulta donde indica cual es el criterio de recomendación y el prototipo imprimirá las recomendaciones correspondientes a cada cliente, seleccionando las mejores según este criterio elegido por el cliente (por ejemplo, popularidad, precio, temática, ...).
- Las recomendaciones aparecerán ordenadas, por orden decreciente, de acuerdo a dicho criterio.

- Si el número de recomendaciones es alto, se mostrarán sólo las N mejores.

No se utilizarán ifs en la parte CLIPS, ni otras construcciones ajenas a los sistemas de producción, salvo de forma muy puntual en alguna deffunction. Tampoco se utilizarán las prioridades ni se eliminarán reglas para simular un programa secuencial. Se podrán utilizar prioridades, módulos, hechos centinela o valores por defecto para garantizar que se completa una etapa del programa antes de iniciar la siguiente.

Sólo se utilizará test cuando se requiera hacer comparaciones de menor que, mayor que, negaciones o funciones or. Las comparaciones de igualdad se realizarán mediante encaje de patrones con los hechos disponibles en la memoria de trabajo.

3. Memoria

La memoria de la práctica deberá incluir:

Parte 1. Ejemplo de ejecución y comentarios generales.

Parte 2.

- Descripción del conocimiento necesario para el funcionamiento del prototipo. La selección y estructuración del conocimiento será una parte importante de la práctica, debiendo documentarse adecuadamente.
- Ejecución del recomendador para al menos 3 usuarios diferentes. Cada ejecución incluirá los hechos iniciales que se han utilizado y los resultados que se han obtenido.

4. Entrega de la práctica

La práctica se entregará a través del Campus Virtual. Se habilitará una nueva tarea:

Entrega de la Práctica 5 que permitirá subir un zip que contendrá la memoria de la práctica y el fichero clp con las soluciones a la primera y segunda parte de la práctica.

El fichero subido deberá tener el siguiente nombre: **Practica5GXX.zip**, siendo XX el nombre del grupo. Por ejemplo, *Practica5G03.zip*.

Uno sólo de los miembros del grupo será el encargado de subir la práctica.

La fecha límite para entregar la Práctica 5 será el **9 de enero de 2019 a las 23:55**. Se recomienda no dejar la entrega para el último momento para evitar problemas de última hora.

No se corregirá ninguna práctica que no cumpla estos requisitos.