Ing. de Sistemas e Informática
Matemáticas Discretas
Jorge M. Londoño P.

Introducción al Prolog

Este taller tiene por objetivos:

- Instalar y familiarizarse con el entorno de trabajo del **SWI Prolog**.
- Aprender a definir la base de datos de hechos de un programa.
- Realizar consultas básicas para determinar el valor de verdad de una expresión lógica o determinar si determinada expresión puede ser satisfecha.

1. Instalación y puesta a punto del entorno.

SWI Prolog es una implementación abierta (Open source) de Prolog disponible para sistemas operativos Windows, Linux y OS X. Para realizar los ejercicios del curso se recomienda utilizar la última versión estable, disponible en la siguiente página:

http://www.swi-prolog.org/

Opcionalmente, existe también una versión 'portable' que no requiere privilegios administrativos para su instalación:

Prolog Portable (Versión 8.0.3) (Versión 7.2.3)

2. Definición de la base de datos de hechos.

En Prolog la base de datos contiene *hechos* y *reglas*. Los hechos son los casos verdaderos de los predicados que define el programador. Cualquier caso que no aparezca explícitamente en la base de datos se toma como falso por Prolog. Las reglas en Prolog son una extensión del concepto de regla de inferencia, donde el valor de verdad de un nuevo predicado se puede definir en función de predicados previamente definidos. Durante este taller no se trabajará con reglas.

Pasos para definir la base de datos:

- Entrar al SWI-Prolog, En este momento aparece la interfaz de consultas.
- En el menú "File" seleccionar la opción "New". Al hacer esto aparece el cuadro de dialogo para seleccionar la ruta y el nombre de la base de datos. Asignar el nombre "**<NombreApellido>-<id>-BD3.pl**" y oprimir "Save". Por defecto la base de datos queda guardada con la extensión ".pl".
- A continuación aparece la ventana de edición titulada "Emacs view". Se

trata de un editor de texto plano, donde se pueden ingresar los hechos.

- Los hechos se componen de un predicado y uno o más términos. Por ejemplo, para indicar que `ana` y `maria` son hermanas se define el hecho:

  `hermanos(ana,maria).`

  En este ejemplo, el predicado es `hermanos` y los `términos` (sujetos) son `ana` y `maria`. El nombre del predicado debe empezar por minúscula. Las constantes que identifican elementos del universo se llaman ***átomos*** en Prolog y deben empezar por letra minúscula. Observar el punto al final. Todo hecho (y en general toda cláusula en Prolog) debe terminar en punto.
- Normalmente cada hecho se escribe en una nueva línea.
- Los hechos representan los casos para los cuales el predicado es verdadero. Cualquier cosa que no aparezca en la base de datos, se entiende por falsa en Prolog.
- En Prolog la conjunción de varios predicados se indica por coma (,), la disyunción por punto y coma (;), y cuando un hecho no es probable (no existe una asignación de las variables que lo haga verdadero) por \+.

Ejercicio 1:

Crear una base de datos “tiene pico y placa” con hechos `picoyplaca/2` que relacionan el último digito de la placa con un dia de la semana. Por ejemplo, el 0 tiene pico y placa el lunes. Los distintos dias para un mismo digito deben ser hechos aparte en la base de datos. Definir suficientes casos del predicado para permitir ilustrar las consultas que se solicitan más abajo. Nota: El Prolog permite editar la base de datos, añadir nuevos hechos y recompilarla, en caso de que se necesiten agregar hechos durante la práctica.

Una vez se ha definido la base de datos, esta se debe "compilar". El Prolog almacena todos estos hechos en memoria para permitir su fácil consulta por medio de la interfaz de consulta. La acción de compilación se hace desde el editor por el menú "Compile" -> "Compile buffer".

Ejercicio 2:

Indicar la consulta y el resultado para los siguientes casos:

- Comprobar que un digito tiene pico y placa un dia particular.
- Comprobar que un digito no tiene pico y placa un determinado dia.
- Comprobar que dos digitos tienen pico y placa el mismo dia. (Conjunción)
- Hacer una consulta en forma de conjunción donde uno de los digitos no tenga pico y placa.
- Hacer una consulta en forma de disyunción considerando el caso donde al

menos un digito tiene pico y placa.

- Hacer una consulta en forma de disyunción donde un digito no tenga pico y placa dos dias distintos.
- Hacer una consulta de forma negada: No tiene pico y placa un digito. Hacer la consulta en el caso que si tiene pico y placa.
- Repetir el caso anterior en el caso que el digito no tiene pico y placa.

Indicar tanto la consulta como el resultado en un archivo de texto plano **<NombreApellido>-<id>-consultas3.txt**.

4. Resolución por unificación

Además de evaluar consultas simples, el Prolog está en capacidad de realizar un proceso mucho más complejo llamado resolución. Por medio de este proceso el Prolog busca valores de las variables que hagan verdadera la expresión indicada en la consulta. Las variables en Prolog se indican con letra mayúscula. Por ejemplo, en la siguiente consulta la variable es X y Prolog encontró los hermanos de juan:

```
14 ?- hermano(juan,X).

X = pedro ;

X = lucia ;

X = marcos.
```

Es igualmente válido utilizar expresiones compuestas y varias variables, por ejemplo la consulta:

```
15 ?- hermano(X,juan), padre(X,maria).

X = fernando.
```

arroja uno o varios posibles átomos que hacen la expresión verdadera.

Ejercicio 3:

- Hacer una consulta que encuentre todos los dias que tiene pico y placa un determinado digito.
- Hacer una consulta que encuentre todos los digitos que tienen pico y placa un determinado dia de la semana.
- Hacer una consulta que encuentre dos digitos que tienen pico y placa un mismo dia. (Debe ser una sola expresión lógica)
- Hacer una consulta que encuentre tres digitos que tienen pico y placa un mismo dia. (Debe ser una sola expresión lógica)
- Hacer una consulta que encuentre los dias de la semana que no tiene pico y placa un determinado digito. (indicar porque se presentan dificultades

para que Prolog resuelva este tipo de consulta).

Incluir tanto la consulta como la respuesta de cada uno de estos casos en el archivo **`<NombreApellido>-<id>-consultas3.txt`**.

Informe:

Remitir los 2 archivos (**`<NombreApellido>-<id>-BD3.pl`** y **`<NombreApellido>-<id>-consultas3.txt`**).

No olvidar incluir los comentarios definiendo los predicados utilizados y sus argumentos.

Referencias

WikiBooks, Prolog.

https://en.wikibooks.org/wiki/Prolog

Learn Prolog Now!

http://www.learnprolognow.org/lpnpage.php?pageid=online

An Introduction to Prolog Programming, Ulle Endriss

https://staff.science.uva.nl/u.endriss/teaching/pss/prolog.pdf