

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 8.2.1](#)) ([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 "**ApellidoNombre-id-BD5.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 (Prolog no puede determinar una particularización de las variables que lo haga verdadero) por `\+`.

Ejercicio 1:

Crear una base de datos “autores de libros” con hechos `autorLibro/2` que relacionan un autor con un libro que él haya escrito. Por ejemplo, ‘hawking’ escribió ‘el gran diseño’. Cada relación de un autor con un libro debe ser un hecho distinto 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 una persona es autor de un libro.
- Comprobar que para una persona que no es autor de un libro, la consulta es falsa.
- Comprobar que una persona es autor de dos libros. (Conjunción)
- Hacer una consulta en forma de conjunción donde la persona no sea autor de uno de los libros.
- Hacer una consulta en forma de disyunción considerando el caso donde la persona es autor de uno de los libros.
- Hacer una consulta en forma de disyunción donde la persona no sea autor de ninguno de los libros.

- Hacer una consulta de forma negada: No es verdad que cierta persona es autor de un libro? Hacer un caso en que la persona si es el autor y un caso en el que no.

Indicar tanto la consulta como el resultado en un archivo de texto plano **ApellidoNombre-id-consultas5.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 libros de un autor.
- Hacer una consulta que encuentre que autores has escrito dos o más libros. (Debe ser una sola expresión lógica)
- Hacer una consulta que encuentre tres libros distintos escritos por un determinado autor. (Debe ser una sola expresión lógica)
- Hacer una consulta que encuentre los libros que no fueron escritos por un autor. Puede el Prolog resolver este tipo de consulta?

Incluir tanto la consulta como la respuesta de cada uno de estos casos en el archivo **ApellidoNombre-id-consultas5.txt**.

Informe:

Remitir los 2 archivos (**ApellidoNombre-id-BD5.pl** y **ApellidoNombre-id-consultas5.txt**). Subir ambos archivos por separado (no comprimir)

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/lpnpag.php?pageid=online>

An Introduction to Prolog Programming, Ulle Endriss

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