

Segunda práctica: Lógica de Predicados

Carlos Ansótegui, Josep Pon

Diciembre 2018

1 Enunciado

El objetivo de esta práctica es evaluar el conocimiento del alumno sobre lógica de predicados y su uso en programación lógica. En la primera tarea, nos centraremos en la modelización de razonamientos y la demostración de su validez mediante el procedimiento de resolución. En la segunda tarea, se tratará de utilizar una base de hechos para crear un conjunto de reglas en Prolog.

1.1 Modelización

Modeliza en lógica de primer orden y demuestra la validez mediante el procedimiento de resolución, los siguientes razonamientos:

- Algunos franceses son amigos de todos los españoles. Ningún francés es amigo de los aficionados al cricket. Por lo tanto, ningún español es aficionado al cricket.
 - Predicados: $E(x)$: x es español; $F(x)$: x es francés; $C(x)$: x es aficionado al cricket; $A(x, y)$: x es amigo de y .
- Sólo los tontos alimentan a los osos salvajes. Cristina alimenta a Nicolás. Cristina no es tonta. Por lo tanto, Nicolás no es un oso salvaje.
 - Predicados: $T(x)$: x es tonto; $O(x)$: x es un oso salvaje; $A(x, y)$: x alimenta a y .
 - Constantes: a Cristina; b Nicolás.
- Todos los que ayudan a Juan viven en casa de Manolo. Antonio ayuda a todos los que trabajan con él. Juan trabaja con todos los amigos de Carlos. Antonio es amigo de Carlos. Por lo tanto, Antonio vive en casa de Manolo.
 - Predicados: $AY(x, y)$: x ayuda a y ; $V(x, y)$: x vive en casa de y ; $T(x, y)$: x trabaja con y ; $A(x, y)$: x es amigo de y .
 - Constantes: j : Juan; m : Manolo; a : Antonio; c : Carlos.
- Ningún aristócrata debe ser condenado a galeras a menos que sus crímenes sean vergonzosos y lleve una vida licenciosa. Hay aristócratas que han cometido crímenes vergonzosos aunque su forma de vida no es licenciosa. Por lo tanto, hay algún aristócrata que no debe ser condenado a galeras.
 - Predicados: $G(x)$: x debe ser condenado a galeras; $C(x)$: x ha cometido crímenes vergonzosos; $V(x)$: x lleva una vida licenciosa
- Cualquiera que ahorra dinero gana interés. Por lo tanto, si no hay interés, entonces nadie ahorra dinero.
 - Predicados: $A(x, y)$: x ahorra y ; $D(x)$: x es dinero; $I(x)$: x es interés; $G(x, y)$: x gana y .

1.2 Programación Lógica

Supongamos que tenemos la siguiente información genealógica de una familia, indicada mediante instanciaciones de los predicados:

$female(X)$: X es mujer.
 $male(X)$: X es hombre.
 $married(X, Y)$: X esta casado/a con Y .
 $parent(X, Y)$: X es padre/madre de Y .

Por ejemplo:

$female(gabriela)$.
 $male(albert)$.
 $male(edward)$.
 $parent(albert, edward)$.
 $parent(gabriela, edward)$.

Debéis, a partir de una base de hechos como los anteriores (que se os entrega junto al enunciado de la práctica en el archivo *genealogy.pl*), definir los siguientes predicados en Prolog:

1. $father(X, Y)$: evaluará a cierto si X es el padre de Y .
2. $mother(X, Y)$: evaluará a cierto si X es la madre de Y .
3. $son(X, Y)$: evaluará a cierto si X es el hijo de Y .
4. $daughter(X, Y)$: evaluará a cierto si X es la hija de Y .
5. $brother(X, Y)$: evaluará a cierto si X es hermano de Y .
6. $sister(X, Y)$: evaluará a cierto si X es hermana de Y .
7. $uncle(X, Y)$: evaluará a cierto si X es tío de Y .
8. $aunt(X, Y)$: evaluará a cierto si X es tía de Y .
9. $grandparent(X, Y)$: evaluará a cierto si X es abuelo/a de Y .
10. $grandchild(X, Y)$: evaluará a cierto si X es nieto/a de Y .
11. $cousin(X, Y)$: evaluará a cierto si X es primo/a de Y .

Observación 1: para los predicados $uncle(X, Y)$, $aunt(X, Y)$ deberéis contemplar dos casos. Que la relación sea de sangre (X es hermano/a del padre/madre de Y) o que la relación se deba a un matrimonio (X se ha casado con en el hermano/a del padre/madre de Y). Por lo tanto, serán necesarios dos casos para cada uno de estos predicados.

Observación 2: con la base de datos de hechos que hemos definido, los predicados $siblings(X, Y)$, $brother(X, Y)$, $sister(X, Y)$, evaluarán a una persona como su propio hermano/a. Para solucionar este problema debéis utilizar uno de los predicados para hacer comparaciones¹, que ya vienen definidos en Prolog.

2 Material a entregar

El material evaluable de esta práctica es:

- Un informe de nombre *log-prac2.pdf*, que incluya: en una primera parte, las respuestas de la tarea de modelización y, a continuación, otra parte dónde expliquéis vuestra definición de los predicados de la tarea de programación lógica.
- Vuestra definición de los predicados de la tarea de programación lógica en Prolog, dentro del mismo archivo que se os proporción *genealogy.pl*.

log-prac2. [tgz|tar.gz|zip]
├─ log-prac2.pdf
└─ genealogy.pl

¹<http://www.swi-prolog.org/pldoc/man?section=compare>