

Programación Declarativa (Lógica y Restricciones)

Ejercicios de Programación Lógica Pura, Parte 1

(predicados no recursivos)

Fecha de entrega: 3 de marzo

Ejercicio 1 Dada la siguiente información, construir un programa lógico que incluya un predicado `elimina/2` que permite saber quién va a eliminar a quien.

1. Si Corleone controla Manhattan y Brooklyn, eliminará a Solozzo.
2. Si Solozzo controla las drogas y Roth le apoya, eliminará a Corleone.
3. Si Roth apoya a Solozzo, Solozzo controlará el Bronx y Harlem; pero si Roth apoya a Corleone, Corleone controlará Manhattan y Brooklyn.
4. Roth apoyará a cualquiera que le garantice la impunidad.
5. Corleone controla las apuestas y Solozzo controla las drogas.
6. Es necesario controlar a la policía si controlas las apuestas.
7. Controlar a la policía implica garantizar la impunidad de cualquiera.

Para estandarizar las respuestas, representaremos todos los nombres con las siguientes constantes: `corleone`, `solozzo`, `manhattan`, `brooklyn`, `drogas`, `roth`, `bronx`, `harlem`, `apuestas`, `policia`.

Ejercicio 2 La familia Addams está compuesta por Homer, Morticia, Pugsley, Wednesday, Tío Fester, Tío Cosa y Abuela Addams. Homer es hermano del Tío Fester y del Tío Cosa; todos ellos son hijos de la Abuela Addams. Morticia y Homer están casados y tienen dos hijos, Pugsley y Wednesday. Escribir un programa lógico puro que contenga estas relaciones y que pueda ser consultado de la siguiente manera:

- ¿Quién es hermano de quién? (definir `hermano/2`)
- ¿Quién es padre de quién? (definir `padre/2`)
- ¿Quién es hijo/a de quién? (definir `hijo/2`)
- ¿Quién es la abuela de quién? (definir `abuela/2`)
- ¿Quién es tío de quién? (definir `tio/2`)
- ¿Quién es sobrino/sobrina de quién? (usando `tio/2`)
- ¿Quién es cuñado de quién? (definir `cunado/2`)

Para estandarizar las respuestas, representaremos todos los nombres con las siguientes constantes: `homer`, `tio_fester`, `tio_cosa`, `abuela_addams`, `morticia`, `pugsley`, `wednesday`.

Importante:

- Para estos dos ejercicios el archivo `code.pl` debe comenzar con una declaración de módulo de la forma:

```
:- module(_,_,[pure]).
```

El tercer argumento determina las características del lenguaje que se puede utilizar en el fichero `code.pl`. Concretamente `pure` evita cargar cualquier predicado predefinido de Prolog, de modo que sólo se permite programación lógica pura.