

Proyecto 1 - Consultas en lenguaje natural a una base de conocimiento sobre genealogía.

1. Descripción de la Actividad

La base de conocimiento contiene hechos de cuatro tipos:

persona(A,G): Indicando que A es una persona (G puede ser *masc* o *fem* indicando su género masculino o femenino respectivamente).

padre(A,B): Indicando que A es el padre de B.

madre(A,B): Indicando que A es la madre de B.

esposo(A,B): Indicando que el hombre A está casado con la mujer B.

Para estar relacionado con **padre(...)**, **madre(...)** o **esposo(...)** hace falta que el correspondiente hecho **persona(...)** esté definido.

Definir a partir de estos hechos las relaciones

esposa(X,Y): Y es esposa de X.

hermano(X,Y) (resp. **hermana(X,Y)**): X es una persona de género masculino (resp. femenino) y Y es una persona que tiene el mismo padre o la misma madre que Y.

abuelo(X,Y) (resp. **abuela(X,Y)**): X es padre (resp. madre) del padre o de la madre de Y.

hijo(X,Y) (resp. **hija(X,Y)**): X es una persona de género masculino (resp. femenino) cuyo padre o madre es Y.

nieto(X,Y) (resp. **nieta(X,Y)**): X es hijo (resp. hija) de un hijo o de una hija de Y.

tio(X,Y) (resp. **tia(X,Y)**): X es hermano (resp. hermana) del padre o de la madre de Y.

sobrino(X,Y) (resp. **sobrina(X,Y)**): X es hijo (resp. hija) de un hermano o hermana de Y.

cuñado(X,Y) (resp. **cuñada(X,Y)**): X está casado (resp. casada) con un hermano o una hermana de Y. O bien X es hermano o hermana del esposo o de la esposa de Y.

suegro(X,Y) (resp. **suegra(X,Y)**): X es padre (resp. madre) del esposo o de la esposa de Y.

Escribir un programa que reciba preguntas en castellano (con un vocabulario preestablecido) y responda a estas preguntas. Las preguntas son del tipo siguiente:

- *¿Quién es artículo relación persona?*
- *¿Quiénes son artículo relación persona?*

- ¿Es verdad que persona es artículo relación persona?

Ejemplos de preguntas:

- ¿Quién es el padre de Alberto?
- ¿Quiénes son los hijos del padre Luis?
- ¿Quién es la madre del tío del cuñado de Ramona?
- ¿Es verdad que José es el tío del nieto de María?

Las relaciones simples que pueden aparecer son de la forma *tío de*, *hermano de*, *abuela de*, etc. Para cada una de las relaciones definidas a partir de los hechos dados. Algunas relaciones pueden estar escritas en plural, por ejemplo tiene sentido hablar de *los hermanos*, de *los abuelos*, etc, pero no de *las madres*.

Las relaciones compuestas son de la forma relación del relación o relación de la relación

Una relación está bien formada si cumple con la concordancia en género y número entre los artículos y las relaciones que la componen.

La respuesta a una pregunta de la forma *¿Quién es ...* requiere de un solo resultado.

Las respuestas a preguntas de la forma *¿Quiénes son ...* requiere de una lista de resultados.

Las respuestas a preguntas de la forma *¿Es verdad que ...* requieren de una respuesta afirmativa o negativa.

Las frases serán introducidas con todos sus elementos ya separados en unidades lexicales. Por ejemplo, Para formular la pregunta *¿Quién es el padre de Alberto?* Se procesa la lista ['¿', 'Quién', 'és', 'él', 'padre', 'de', 'Álberto', '??']

Los nombres de personas siempre son una unidad lexical y no se hace ningún tipo de tratamiento de apellidos.

El predicado principal de la aplicación se llama **preguntar** y tiene dos componentes, la frase de pregunta y la respuesta.

Por ejemplo:

```
preguntar(['¿', 'Quién', 'es', 'el', 'padre', 'de', 'Alberto', '??'], ['Luis'])
```

```
preguntar(['¿', 'Quiénes', 'son', 'los', 'sobrinos', 'de', 'Donald', '??'],
          ['Luis', 'Hugo', 'Paco'])
```

```
preguntar(['¿', 'Es', 'verdad', 'que', 'José', 'es', 'el', 'tío', 'del', 'nieto',
          'de', 'María', '??'], _)
```

(puede producir éxito o fallar sin tener que instanciar el segundo elemento de la relación).

Sugerencias de implementación:

- Utilizar gramáticas de metamorfosis (definite clause grammars) para reconocer la frase.
- Construir una lista de relaciones encadenadas como representación de la frase, por ejemplo, representar

```
[',', 'Quién', 'es', 'el', 'tío', 'del', 'nieto', 'de', 'María', '?']
```

como el par

```
(X,[esTio(X, Y), esNieto(Y, 'María')])
```

- Buscar resolver la lista de relaciones a partir del conocimiento almacenado sobre la genealogía de esa familia.

2. Entrega de la implementación

- Debe entregar un archivo **.tar.gz** o **.tar.bz2** (no puede ser **.rar**, ni **.zip**) que contenga el archivo llamado **interpretador.pro**. Este archivo debe poseer el código en prolog que implemente la actividad descrita anteriormente.
- Debe entregar el código de su proyecto al encargado de su sección vía email.
- La fecha de entrega del proyecto es el jueves 11 de febrero del 2010 (semana 5), hasta las 6:30 pm.
- Se evaluará el estilo correcto de programación, debe emplear una indentación adecuada y consistente en cualquier editor. Ver la guía de estilo que se encuentra en la página web del curso.
- El valor del proyecto es 33 puntos.