

Instructivo PROLOG 1era parte

El lenguaje Prolog es un lenguaje de programación basado en la lógica de predicados.

En los prácticos de Modelos Formales No Transformacionales vamos a usar este lenguaje únicamente para escribir y probar Gramáticas Independientes de Contexto y, más adelante, Gramáticas de Unificación y Rasgos, siguiendo en todos los casos la sintaxis de las llamadas Definite Clause Grammars (DCGs). En este instructivo se especifican algunas formas de instalarlo y de utilizarlo para el reconocimiento y escritura de Gramáticas Independientes de Contexto.

1. Instalación

Toda instalación depende siempre del sistema operativo. Aquí especificaremos solamente la forma de hacerlo para Windows y para Linux (chequeada para Ubuntu 16.04). En caso de poseer otro sistema operativo o de que no funcionen las presentes instrucciones, se recomienda utilizar algún buscador para encontrar la manera más adecuada de hacer la instalación.

Para Windows

Bajarse el ejecutable de Prolog para Windows desde <http://www.swi-prolog.org/>.

Para facilitar la utilización (sobre todo para estudiantes no habituados a manejarse desde la consola), se pueden bajar el editor SWI-Prolog Editor desde la página <http://arbeitsplattform.bildung.hessen.de/fach/informatik/swiprolog/indexe.html> (en ese caso, es sumamente necesario que la versión de PROLOG que se hayan descargado sea la de 32bits, en caso contrario, pueden bajarse la versión de 64bits).

Instalar primero el programa y después el editor.

Para Linux

Abrir la consola (ctrl+alt+t) correr sucesivamente los siguientes comandos (se especifica entre paréntesis y de modo muy informal y resumido qué hace cada comando):

```
sudo apt-add-repository ppa:swi-prolog/stable
```

(Este comando sirve para definir el repositorio desde donde se va a descargar el programa)

```
sudo apt-get update
```

(Este comando sirve para actualizar los paquetes disponibles)

```
sudo apt-get install swi-prolog
```

(Este comando instala swi-prolog)

2. Cómo leer DCGs

A continuación se especifican distintas formas de leer una DCG. Vamos a especificar dos formas para Windows (una con el editor especificado en la instalación y otra sin) y una para Linux.

2.1 Cómo leer DCGs en Prolog alternativa 1 (Windows con editor).

1) Abrir el editor (normalmente se crea un ícono en el escritorio)

2) Abrir desde la pestaña *archivo* “cfgejemplo1”.

3) En la parte de arriba va a aparecer la gramática de ejemplo escrita para Prolog, en la parte de abajo va a aparecer la consola del lenguaje y allí vamos a ver un símbolo de pregunta y un guion. Escribir ahí “[cfgejemplo1].”. No se olviden de poner un punto al final.

1 ?- [cfgejemplo1].

4) Apretan *enter* y entonces, si todo salió bien, les aparece esto:

1 ?- [cfgejemplo1].

true.

5) Ahora le pueden hacer preguntas respecto de si tal o cual oración puede ser generada por su gramática o no.

2 ?- phrase(s, [el, estudiante, piensa]).

Apretan *enter* y si la oración pertenece a su gramática aparece:

true

Si no, aparece

false

6) También pueden preguntar si determinada frase es o no gramatical:

3 ?- phrase(sn, [el, estudiante]).

Apretan *enter* y si la frase pertenece a su gramática aparece:

true

Si no, aparece

false

7) También pueden pedirle al sistema que genere todas las oraciones gramaticales posibles usando el siguiente comando:

4?- phrase(s, Sentence).

Prolog va a generar primero solo una oración. Si apretan la tecla de punto y coma, les generará otra y así sucesivamente hasta agotar todas las oraciones posibles a partir de esa gramática.

2.2 Cómo leer DCGs en Prolog 2 (Windows sin editor)

1) Abrir el archivo cfgejemplo1.pl y el archivo cfgejemplo1.txt. Ambos archivos son idénticos, a excepción de la extensión con la que están guardadas, por lo que consultar el txt les permitirá ver cómo está construida la gramática. En el archivo de formato pl tipear el siguiente comando.

[cfgejemplo1].

Al apretar *enter* debería aparecer true.

2) Ahora le pueden hacer preguntas respecto de si tal o cual oración puede ser generada por su gramática o no.

2 ?- phrase(s, [el, estudiante, piensa]).

Apretan *enter* y si la oración pertenece a su gramática aparece:

true

Si no, aparece

false

3) También pueden preguntar si determinada frase es o no gramatical:

3 ?- phrase(sn, [el, estudiante]).

Apretan *enter* y si la frase pertenece a su gramática aparece:

true

Si no, aparece

false

4) También pueden pedirle al sistema que genere todas las oraciones gramaticales posibles usando el siguiente comando:

4?- phrase(s, Sentence).

Prolog va a generar primero solo una oración. Si apretan la tecla de punto y coma, les generará otra y así sucesivamente hasta agotar todas las oraciones posibles a partir de esa gramática.

2.3 Cómo leer DCGs en Prolog 3 (Linux)

1) Hagan click derecho en la carpeta descomprimida “implementaciones-clases-mfnt-en-prolog” y seleccionen allí “Abrir en terminal”

o alternativamente

1') abran la terminal y desde allí muévase hasta el directorio “implementaciones-clases-mfnt-en-prolog” (o a donde sea que estén todos los archivos descargados del presente repositorio.

2) Escribir en la terminal el comando:

swipl

Si todo sale bien, en la consola va a aparecer la siguiente información (la información precisa puede cambiar según la versión que hayan descargado):

Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)

SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.

Please run ?- license. for legal details.

For online help and background, visit <http://www.swi-prolog.org>

For built-in help, use ?- help(Topic). or ?- apropos(Word).

3) Abran el archivo cfgejemplo1 en el editor que les sea cómodo (por ejemplo, gedit, sublime, etc.). Esto les permitirá ver cómo está escrita la gramática.

4) Escriban en la consola el siguiente comando para abrir el archivo cfgejemplo1:

[cfgejemplo1].

Al apretar enter debería aparecer true.

5) Ahora le pueden hacer a la consola preguntas respecto de si tal o cual oración puede ser generada por su gramática o no.

2 ?- phrase(s, [el, estudiante, piensa]).

Apresan enter y si la oración pertenece a su gramática aparece:

true

Si no, aparece

false

6) También pueden preguntar si determinada frase es o no gramatical:

3 ?- phrase(sn, [el, estudiante]).

Apresan enter y si la frase pertenece a su gramática aparece:

true

Si no, aparece

false

7) También pueden pedirle al sistema que genere todas las oraciones gramaticales posibles usando el siguiente comando:

4?- phrase(s, Sentence).

Prolog va a generar primero solo una oración. Si apretan la tecla de punto y coma, les generará otra y así sucesivamente hasta agotar todas las oraciones posibles a partir de esa gramática.

3. Escritura de una DCG.

Para escribir una CFG en Prolog hay que tener en cuenta las siguientes condiciones:

Para escribir en Prolog una gramática deben tener en cuenta las siguientes indicaciones:

- Los símbolos no terminales van en minúscula.
- Los símbolos terminales van entre corchetes: Ej: [una]
- Cada símbolo va separado por una coma.
- Las flechas se representan -->
- La disyunción se expresa con ;
- Cada regla de reescritura debe terminar en punto.

Puede verse una introducción más completa a cómo escribir una gramática en Prolog en:

<http://personal.us.es/fsoler/papers/05capgram.pdf>

Pueden crear su CFG en un archivo en Prolog usando alguna de las siguientes instrucciones:

- Si descargaron el editor y trabajan desde Windows, pueden clicar en el editor “archivo” y luego “nuevo”
- Si están trabajando en Windows usando el archivo pl, pueden crear cualquier txt, escribir la gramática en él y cambiarle luego de guardar todos los cambios la extensión txt por la extensión pl y abrir ese archivo.
- Si están trabajando desde la consola de Linux, pueden simplemente crear un archivo de extensión pl, abrirlo con el editor que tengan disponible (gedit, sublime, etc.), escribir allí la gramática, guardarla y luego cargarla desde la consola.

Tarea: Armar en Prolog la gramática independiente de contexto para generar las siguientes frases nominales (ejercicio adaptado de Müller 2016: 63). Los nombres propios están en minúscula adrede.

una persona

una persona que conocemos

una persona de brandsen

una persona inteligente

una persona inteligente que conocemos

una persona inteligente de brandsen

una persona inteligente de brandsen que conocemos