

## Especificación de reglas contextuales

Las reglas a ser aplicadas por el componente para Reglas Contextuales de Lavinia deben especificarse en un archivo de texto.

Las reglas se agrupan en módulos. En un mismo archivo se incluyen todos los módulos con sus reglas. Para cada módulo se incluye una línea que le da nombre al módulo y a continuación todas las reglas que lo componen.

### Definición del módulo

```
modulo# nomModulo.
```

### Sintaxis de las reglas

```
nomRegla# etiquetaAAsignar => contIzq >> cuerpo << contDer // conds .
```

nomRegla	Nombre de la regla.
etiquetaAAsignar	Etiqueta que se le va a asignar al segmento de texto que reconoce la regla, es decir, que concuerda con el cuerpo de la regla.
contIzq	Contexto izquierdo: elementos que se piden a la izquierda del segmento que la regla reconoce (puede ser vacío).
contDer	Contexto derecho: elementos que se piden a la derecha del segmento que la regla reconoce (puede ser vacío).
cuerpo	segmento que se va a etiquetar (NO puede ser vacío).
conds	condiciones que se deben cumplir para algunos valores de algunos elementos de los contextos o del cuerpo.

Notar que tanto la línea que define el módulo como la especificación de las reglas terminan en **punto**.

### Formato del archivo

El archivo completo tendrá la siguiente estructura:

```
modulo# mod1.

regl1# etiqu => contIzq >> cuerpo << contDer // conds .
regl2# etiqu => contIzq >> cuerpo << contDer // conds .
...

modulo# mod2.

reg21# etiqu => contIzq >> cuerpo << contDer // conds .
reg22# etiqu => contIzq >> cuerpo << contDer // conds .
...

...

modulo# modn.

regn1# etiqu => contIzq >> cuerpo << contDer // conds .
regn2# etiqu => contIzq >> cuerpo << contDer // conds .
...
```

## **Definición de etiquetas**

Las reglas pueden asignar etiquetas nuevas o etiquetas ya existentes. Cuando se asigna una etiqueta nueva se le da un nombre (primer elemento de la etiqueta) y atributos (elementos restantes). Ejemplo de etiqueta:

[ 'indicadorFuente', 'positivo', 'verbal' ]

## **Elementos que pueden aparecer en contIzq, contDer y cuerpo**

Cada elemento indica una etiqueta que debe estar presente en el texto, ya sea porque es una de las etiquetas de partida (las que se obtienen a partir de los *tags* de FreeLing), ya sea porque es una etiqueta asignada por alguna regla aplicada con anterioridad a la regla que la usa.

Las etiquetas que tiene el texto en primera instancia provienen de la información de FreeLing. Cada palabra o signo de puntuación del texto está anotada con una etiqueta como la siguiente:

[ categoria, palabra, lema, atr1, atr2, ... ]

categoria	Categoría gramatical de la palabra, se obtiene a partir del primer carácter de la etiqueta asignada por FreeLing.
palabra	Palabra original presente en el texto.
lema	Lema correspondiente a la palabra.
atrI	I-ésimo carácter de la etiqueta asignada a la palabra por FreeLing.

Por ejemplo, para la palabra *árboles*, FreeLing da como lema la palabra *árbol* y asigna la etiqueta NCMS000. El texto que recibe el módulo de aplicación de reglas contextuales tendrá para la palabra *árboles*, la etiqueta:

[ 'N', 'árboles', 'árbol', 'C', 'M', 'S', '0', '0', '0' ]

Para obtener información sobre las etiquetas de FreeLing, leer la documentación de FreeLing en <http://garraf.epsevg.upc.es/freeling/> o bajar el documento con la especificación de las etiquetas de la página del curso.

Sobre las etiquetas pueden aplicarse algunos operadores que aumentan la expresividad de las reglas:

op(et)	la etiqueta et es opcional
el(et)	la aplicación de la regla, además de asignar la etiqueta que aparece del lado izquierdo de la regla (antes del signo =>), tiene como efecto eliminar la etiqueta et.

Para cada etiqueta, la concordancia puede establecerse sobre cualquiera de sus componentes. La notación siguiente:

[ \_, 'v2', \_, \_, 'v5', \_, \_, \_, \_ ]

indica que se pide una etiqueta con el valor v2 en el segundo componente y el valor v5 en el quinto componente. Estos valores se ponen entre comillas simples (aunque en muchos casos estas comillas pueden evitarse, dependiendo de algunas restricciones del lenguaje Prolog). Los lugares ocupados por infra-guiones indican que se admite cualquier valor para esos componentes.

Es posible abreviar el final de la etiqueta, cuando no se necesita poner allí ningún valor específico, utilizando la barra vertical y un infra-guión. A continuación se muestra la abreviación de la etiqueta anterior.

[\_, 'v2', \_, \_, 'v5' | \_]

### **Ejemplos de elementos que pueden ocurrir en contIzq, contDer y cuerpo**

Notación	Tipo de palabra	Ejemplos
[_, 'de'   _]	únicamente la palabra <i>de</i>	<i>de</i>
['S', _, _, 'P'   _]	cualquier preposición	<i>de, a, del, al, contra, desde.</i>
['V'   _]	cualquier forma verbal	<i>ir, voy, cantar, cantaran, vivido</i>
['V', _, 'caminar'   _]	cualquier forma del verbo <i>caminar</i>	<i>camino, caminamos, caminarían, caminado, caminando</i>
['V', _, _, _, 'N'   _]	cualquier verbo en infinitivo	<i>cantar, comer, servir, limpiar</i>
['N', _, _, _, 'S'   _]	cualquier nombre singular	<i>árbol, temor, doctora, lámpara</i>

Sobre cualquiera de los elementos de la tabla pueden aplicarse los operadores **op** y **el**, antes descriptos. Tener en cuenta que, en la tabla, sólo se pusieron ejemplos de las etiquetas iniciales, es decir, las que contienen la información de Freeling.

### **Ejemplos de reglas sin condiciones**

```
reg1# [noRealis,advNeg] => >> [_, 'no' | _], ['V' | _] << .
reg2# [noRealis,advImp] => [_, 'imposible' | _] >> ['V', _, _, _, 'N' | _] << .
```

En reg1 los dos contextos son vacíos y el cuerpo tiene dos elementos (separados por coma): la palabra *no* y un verbo (en cualquier forma). Esto quiere decir que la regla reg1 asigna la etiqueta noRealis con el valor advNeg para el primer atributo a todo segmento de texto compuesto por *no* y un verbo. En reg2 el contexto izquierdo debe contener la palabra *imposible* y el cuerpo debe ser un verbo en infinitivo.

Por ejemplo, para:

Manuel no quiere ir a la escuela.

se agrega la etiqueta:

Manuel <noRealis atr1="advNeg"> no quiere </noRealis> ir a la escuela.

### **Variables**

Se pueden usar variables para tomar el valor de algún atributo de alguna etiqueta del lado derecho de la regla (contIzq, cuerpo o contDer) para ponerlo como valor del algún atributo de la etiqueta que la regla asigna (lado izquierdo de la regla). Cualquier palabra

comenzada en letra mayúscula, sin comillas, es una variable. A continuación se muestra un ejemplo del uso de variables.

```
reg1# [noRealis,advNeg,Modo] => >> [_,'no'|_], ['V',_,_,_,Modo|_] << .
```

La aplicación de esta regla al mismo ejemplo que se dio, da como resultado el texto anotado siguiente:

Ej. 5 Manuel <noRealis atr1="advImp" atr2="I"> no quiere </noRealis> ir a la escuela.

(La letra "I" indica modo Indicativo.)

### **Condiciones**

Se pueden imponer condiciones que deben cumplir algunos elementos de las etiquetas. Al final de la regla, luego del símbolo //, se indica la condición que debe cumplir el elemento, dado como variable. Por ejemplo, en la regla siguiente se pide que el lema del nombre que aparece después del determinante sea un título de persona (*señor, doctor, presidente*), esta restricción se expresa mediante la condición "tit" aplicada al lema.

```
reg3# [titNP] => >> [_,'el'|_], ['N',_,Lema|_], ['N',_,_, 'P'|_] << // tit(Lema) .
```

La lista de elementos que cumple con dicha condición se especifica en un archivo distinto del archivo de reglas, que debe tener la extensión "pl". Por ejemplo, se puede crear el archivo conds.pl con el siguiente contenido:

```
tit('señor').
tit('doctor').
tit('presidente').
tit('senador').
tit('ingeniero').
```

Esto significa que la regla reg3 reconocerá todos los ejemplos siguientes:

*El señor Juan Pérez será llamado ...*

*La doctora Rodríguez vendrá ....*

*Estamos con la ingeniera Sara López ....*

(Notar que FreeLing reconoce nombres propios compuestos por varias palabras, como *Juan Pérez*, y les asigna la etiqueta NP00000. De modo que, inicialmente, el segmento de texto *Juan Pérez* llevará la etiqueta ['N','Juan\_Pérez','Juan\_Pérez','P','0','0','0','0','0']).