

Sesión de Prácticas 4

Diseño y creación de gramáticas para LN

Objetivos.

En esta práctica vamos a jugar con la creación de gramáticas para analizar oraciones en lenguaje natural y hacer frente a algunos problemas sintácticos. Para ello vamos a utilizar el formalismo gramatical DCG (Gramáticas de Cláusulas Definidas) que se puede implementar directamente en Prolog.

Los dos aspectos que se van a trabajar son:

1. la formalización de una lengua natural y la resolución de aspectos complejos de la sintaxis,
2. las gramáticas DCG y el mecanismo de la unificación.

Ejercicio 1.

Escribir una pequeña gramática siguiendo el formalismo de las Gramáticas de cláusulas Definidas (DCG) y un pequeño léxico que sea capaz de analizar correctamente las siguientes oraciones:

1. “El hermano de Juan trabaja en la oficina”
2. “Juan come”
3. “Juan come pescado”
4. “Juan habla de su casa” (Pero debe dar error para la oración “Juan habla pescado”)
5. “Juan dice la verdad”

Al hacer los cambios pertinentes para cada oración, recuerda comprobar que las oraciones anteriores se siguen analizando correctamente.

Especificaciones:

- El formalismo para las reglas sintácticas se basa en reglas de reescritura entre símbolos no terminales del tipo:

$sn \rightarrow det, n.$

- El símbolo inicial debe ser el de oración.
- Los símbolos terminales serán el del sintagma nominal (sn), sintagma verbal (sv) y sintagma preposicional (sprep). A nivel de categoría gramatical el nombre (n), verbo (v), preposición (prep), determinante (det) y conjunción (cj).
- Las reglas del léxico deben tener la siguiente forma:

$v \rightarrow [corre].$

donde “v” es el símbolo no terminal y la lista “[come]” el símbolo terminal.

- Puedes tomar como ejemplo el fichero gramatical.pro.

- Para que Prolog analice la oración, debes
 - Cargar la gramática (File>consult), y luego lanzar la oración.
 - Dado que el texto no ha sido tokenizado, hay que indicarle a Prolog la separación entre las palabras. Así, la entrada debe ser el símbolo inicial de oración, más dos listas, una con la oración, en las que las palabras queden separadas por comas (sin acentos ni mayúsculas), y la otra lista vacía. Por ejemplo:


```
?- sn([la,casa], []).
```

```
?- S([juan,dice,la,verdad],[]).
```
 - Prolog dirá “Yes” o “No” según si es capaz de analizar la oración con la gramática que habéis creado o no.
- Para ver errores, activa la opción *graphical debugger* de SWI-Prolog y haz una traza.

Ejercicio 2.

Ampliar la gramática para que acepte oraciones con sintagmas nominales coordinados. Por ejemplo:

“El policía y el hermano de Juan atraparon al ladrón en su casa”

Aquí el problema es la recursividad a izquierdas. ¿Qué solución se te ocurre?

Ejercicio 3.

Mediante el mecanismo de la unificación, ampliar la gramática para que sólo acepte como correctos sintagmas nominales en los que haya concordancia de género y número, y oraciones en las que haya concordancia de número.

Debe dar como correcta la oración:

“Los hermanos de Juan comen pescado”

pero como incorrecta las oraciones:

“El hermanos de Juan comen pescado”

“Los hermanos de Juan come pescado”

Para ello, debes introducir la información de género y número en el léxico, y luego ir almacenándola en las reglas de la gramática en forma de variables. Recuerda que en prolog las variables se expresan con la primera letra en mayúscula.

En fichero gramatica2.pro es un ejemplo. Échale un vistazo.

Ejercicio 4

Amplia la gramática para que dé como salida el árbol de derivación sintáctica.
La salida debe ser así:

s(np(det(el),n(perro)),vp(v(come),np(n(pescado))))

Comprobar que funciona para todas las oraciones anteriores.