



INSTITUTO TECNOLÓGICO BELTRÁN
Centro de Tecnología e Innovación

Gramáticas Libre del contexto en prolog

Concordancia de género y número

KBpracticaGramaticas.pl

```
1  o --> sn, sv.
2  sn --> det, n.
3  sv --> vt, sn.
4  sv --> vi.
5  det --> [el].
6  det --> [un].
7  det --> [una].
8  n --> [perro].
9  n --> [hueso].
10 vi --> [ladra].
11 vt --> [muerde].
```

KBpracticaGramaticas3.pl

```
1  o(o(SN,SV)) --> sn(SN,_Gen,Num), sv(SV,Num).
2  sn(sn(DET,N),Gen,Num) --> det(DET,Gen,Num), n(N,Gen,Num).
3  sv(sv(VT,SN),Num) --> vt(VT,Num), sn(SN,_Gen,_Num).
4  sv(sv(VI),Num) --> vi(VI,Num).
5  det(det(X),f,sg) --> [X], {member(X,[la,una])}.
6  det(det(X),f,pl) --> [X], {member(X,[las,unas])}.
7  det(det(X),m,sg) --> [X], {member(X,[el,un])}.
8  det(det(X),m,pl) --> [X], {member(X,[los,unos])}.
9  vi(vi(ladra),sg) --> [ladra].
10 vi(vi(ladran),pl) --> [ladran].
11 vt(vt(muerde),sg) --> [muerde].
12 vt(vt(muerden),pl) --> [muerden].
13 n(n(perra),f,sg) --> [perra].
14 n(n(perras),f,pl) --> [perras].
15 n(n(X),m,sg) --> [X], {member(X,[perro,hueso])}.
16 n(n(X),m,pl) --> [X], {member(X,[perros,huesos])}.
```

Alumno = Nicolas Mesquiatti

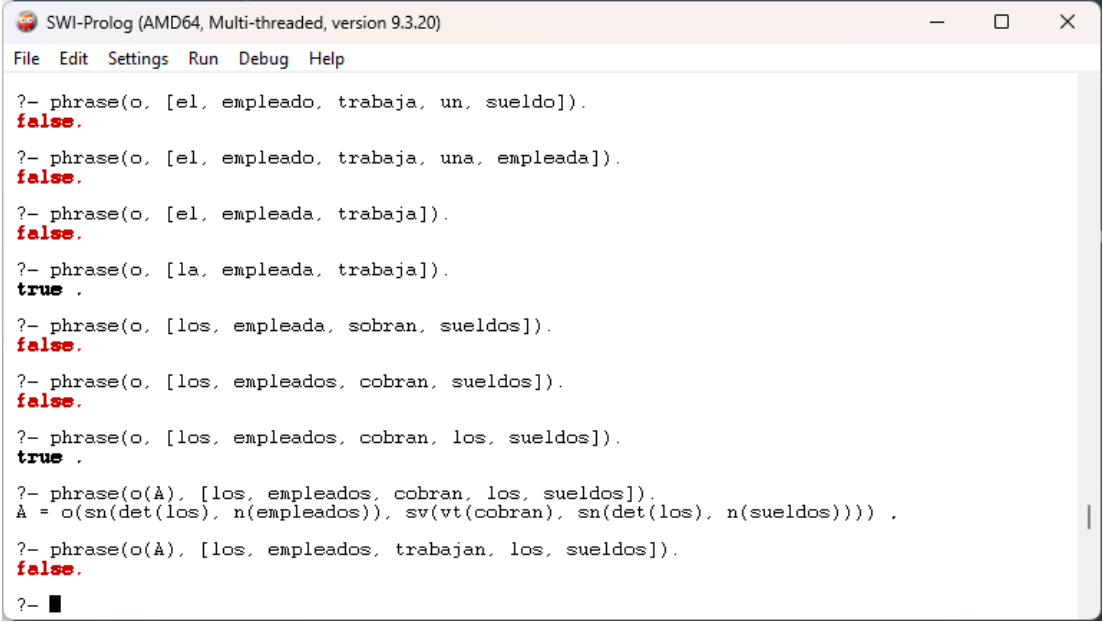
Materia = Técnicas de procesamiento del habla

Práctica:

Realizar programa en Prolog, base de conocimiento y árbol sintáctico con las siguientes palabras:

empleada/o, trabaja/n, cobra/n, sueldo/s

- -Programar en Prolog las gramáticas.
- -Convertir las categorías gramaticales en predicados de Prolog con argumentos que indican
 - el género y el número.
- -Construir el árbol sintáctico.
- -Realizar las siguientes consultas:



```
SWI-Prolog (AMD64, Multi-threaded, version 9.3.20)
File Edit Settings Run Debug Help

?- phrase(o, [el, empleado, trabaja, un, sueldo]).
false.

?- phrase(o, [el, empleado, trabaja, una, empleada]).
false.

?- phrase(o, [el, empleada, trabaja]).
false.

?- phrase(o, [la, empleada, trabaja]).
true.

?- phrase(o, [los, empleada, sobran, sueldos]).
false.

?- phrase(o, [los, empleados, cobran, sueldos]).
false.

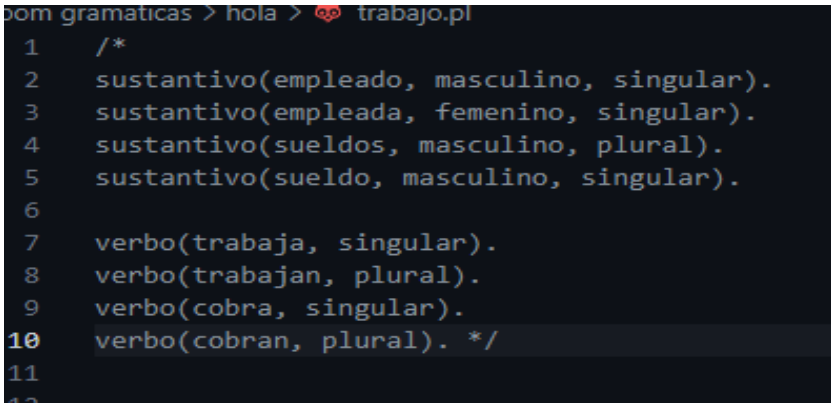
?- phrase(o, [los, empleados, cobran, los, sueldos]).
true.

?- phrase(o(A), [los, empleados, cobran, los, sueldos]).
A = o(sn(det(los), n(empleados)), sv(vt(cobran), sn(det(los), n(sueldos)))) .

?- phrase(o(A), [los, empleados, trabajan, los, sueldos]).
false.

?- 
```

-Programar en Prolog las gramáticas.



```
dom gramaticas > hola > trabajo.pl
1  /*
2  sustantivo(empleado, masculino, singular).
3  sustantivo(empleada, femenino, singular).
4  sustantivo(sueldos, masculino, plural).
5  sustantivo(sueldo, masculino, singular).
6
7  verbo(trabaja, singular).
8  verbo(trabajan, plural).
9  verbo(cobra, singular).
10 verbo(cobran, plural). */
11
12
```

-Convertir las categorías gramaticales en predicados de Prolog con argumentos que indican

```

1  o(o(SN,SV)) --> sn(SN,_Gen,Num), sv(SV,Num). %Una oracion está formada por un SN y un SV que deben concordar en número.
2  sn(sn(DET,N),Gen,Num) --> det(DET,Gen,Num), n(N,Gen,Num). %Un SN tiene un determinante y un nombre con el mismo género y número.
3  sv(sv(VT,SN),Num) --> vt(VT,Num), sn(SN,_Gen,_Num). %Un SV tiene un verbo transitivo y un SN que concuerdan en número.
4  sv(sv(VI),Num) --> vi(VI,Num). %Un SV puede tener un verbo intransitivo que concuerda en número.
5  det(det(X),f,sg) --> [X], {member(X,[la,una])}. %Un determinante femenino singular puede ser "la" o "una".
6  det(det(X),f,pl) --> [X], {member(X,[las,unas])}. %Un determinante femenino plural puede ser "las" o "unas".
7  det(det(X),m,sg) --> [X], {member(X,[el,un])}. %Un determinante masculino singular puede ser "el" o "un".
8  det(det(X),m,pl) --> [X], {member(X,[los,unos])}. %Un determinante masculino plural puede ser "los" o "unos".
9  vi(vi(trabaja),sg) --> [trabaja]. %Un verbo intransitivo singular puede ser "trabaja".
10 vi(vi(trabajan),pl) --> [trabajan]. %Un verbo intransitivo plural puede ser "trabajan".
11 vt(vt(cobra),sg) --> [cobra]. %Un verbo transitivo singular puede ser "cobra".
12 vt(vt(cobran),pl) --> [cobran]. %Un verbo transitivo plural puede ser "cobran".
13 n(n(empleada),f,sg) --> [empleada]. %Un nombre femenino singular puede ser "empleada".
14 n(n(empleadas),f,pl) --> [empleadas]. %Un nombre femenino plural puede ser "empleadas".
15 n(n(X),m,sg) --> [X], {member(X,[empleado,sueldo])}. %Un nombre masculino singular puede ser "empleado" o "sueldo".
16 n(n(X),m,pl) --> [X], {member(X,[empleados,sueldos])}. %Un nombre masculino plural puede ser "empleados" o "sueldos".
17

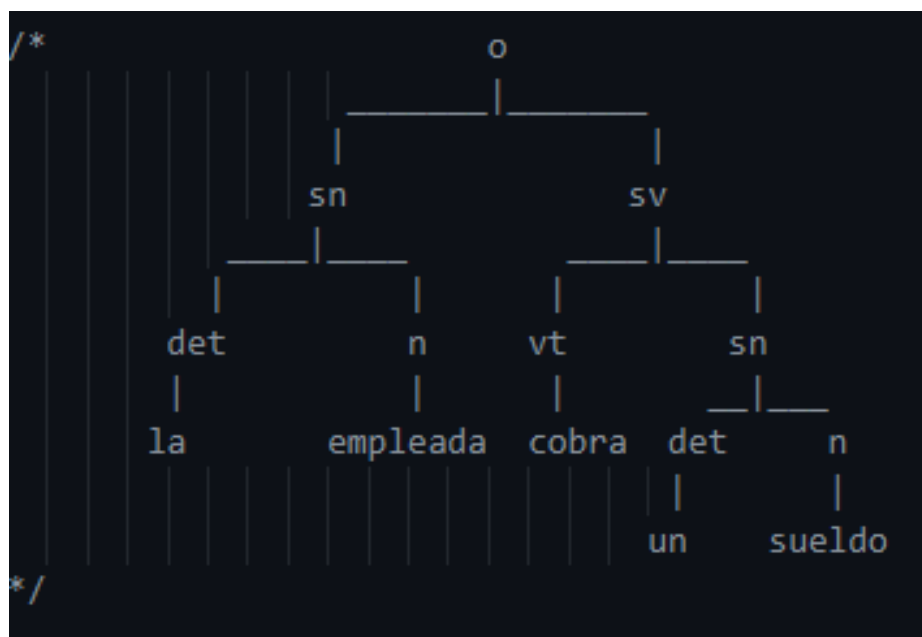
```

La oración completa se divide en dos partes:

- SN (sujeto) y SV (predicado)
- El **SN** está formado por un determinante y un sustantivo, que deben tener el mismo género y número.
- El **SV** puede ser: un verbo intransitivo (ej. "trabaja"), o un verbo transitivo seguido de otro SN (ej. "cobra un sueldo").

La concordancia en número entre SN y SV es obligatoria

-Construir el árbol sintáctico.



-Realizar las siguientes consultas:

```
SWI-Prolog (AMD64, Multi-threaded, version 9.3.20)
File Edit Settings Run Debug Help

?- phrase(o, [el, empleado, trabaja, un, sueldo]).
false.

?- phrase(o, [el, empleado, trabaja, una, empleada]).
false.

?- phrase(o, [el, empleada, trabaja]).
false.

?- phrase(o, [la, empleada, trabaja]).
true.

?- phrase(o, [los, empleada, sobran, sueldos]).
false.

?- phrase(o, [los, empleados, cobran, sueldos]).
false.

?- phrase(o, [los, empleados, cobran, los, sueldos]).
true.

?- phrase(o(A), [los, empleados, cobran, los, sueldos]).
A = o(sn(det(los), n(empleados)), sv(vt(cobran), sn(det(los), n(sueldos)))) .

?- phrase(o(A), [los, empleados, trabajan, los, sueldos]).
false.

?-
```

Consultas realizadas

```
phrase(o_, [la, empleada, trabaja]).
true.

6 ?- phrase(o_), [el, empleado, trabaja, un, sueldo]).
false.

7 ?- phrase(o_), [el, empleado, trabaja, una, empleada]).
false.

8 ?- phrase(o_), [el, empleado, trabaja]).
true.

9 ?- phrase(o_), [el, empleada, trabaja]).
false.

10 ?- phrase(o_), [la, empleada, trabaja]).
true.

11 ?- phrase(o_), [los, empleada, sobran, sueldos]).
false.

12 ?- phrase(o_), [los, empleados, cobran, sueldos]).
false.

13 ?- phrase(o_), [los, empleados, cobran, los, sueldos]).
true.

14 ?- phrase(o(A), [los, empleados, cobran, los, sueldos]).
A = o(sn(det(los), n(empleados)), sv(vt(cobran), sn(det(los), n(sueldos)))) .

15 ?- phrase(o(0), [los, empleados, cobran, los, sueldos]).
0 = o(sn(det(los), n(empleados)), sv(vt(cobran), sn(det(los), n(sueldos)))) .

16 ?- phrase(o(0), [los, empleados, cobran, un, sueldos]).
false.

17 ?- phrase(o(0), [los, empleados, cobran, un, sueldo]).
0 = o(sn(det(los), n(empleados)), sv(vt(cobran), sn(det(un), n(sueldo)))) .
```

Diferencias entre usar (_) o (O)/(A)

- **Cuando usas _**, le estás diciendo a Prolog: *"No me interesa el resultado estructurado, solo dime si la oración es válida."*
- **Cuando usas O/A** en este caso, le pedimos a Prolog: *"Dame la estructura sintáctica de la oración."*

Conclusión

La implementación de gramáticas en Prolog ha permitido modelar estructuras lingüísticas de manera eficiente, convirtiendo categorías gramaticales en predicados con parámetros de género y número. A través del uso de DCG, se ha logrado desarrollar un sistema capaz de analizar y generar oraciones válidas dentro del marco definido.

```
oracion --> sintagma_nominal, sintagma_verbal.  
  
sintagma_nominal --> determinante, sustantivo.  
sintagma_verbal --> verbo.  
  
determinante --> [el].  
sustantivo --> [empleado].  
verbo --> [trabaja].
```

Además, la construcción del árbol sintáctico ha brindado una representación clara de la relación entre los distintos elementos de la oración, facilitando la comprensión estructural del lenguaje