

Programación Lógica

Gramáticas lógicas



DCG: *Definite clause grammars*

Gramáticas escritas con una sintaxis similar a la usual (en Gramáticas Independientes de Contexto) que se traducen a cláusulas definidas y, por lo tanto, a predicados Prolog.

DCG: *Definite clause grammars*

Ejemplo: Gramática para el lenguaje regular $a^* b^* c^*$

$s \rightarrow a, b, c.$

$a \rightarrow [a], a.$

$a \rightarrow [].$

$b \rightarrow [b], b.$

$b \rightarrow [].$

$c \rightarrow [c], c.$

$c \rightarrow [].$

DCG: *Definite clause grammars*

Sintaxis:

- Los símbolos no terminales se representan usando átomos Prolog.
- Los símbolos terminales se encierran entre [].
- Los símbolos de la parte derecha de la regla van separados por comas.
- Cada regla termina con un punto.

DCG: *Definite clause grammars*

- Las DCGs permiten manejar argumentos para los símbolos no terminales y llamadas a predicados dentro de las reglas.
- Estas extensiones permiten representar lenguajes sensibles al contexto con gramáticas muy similares a las anteriores.

DCG: *Definite clause grammars*

Gramática para el lenguaje sensible al contexto $a^N b^N c^N$

$s \rightarrow a(N), b(N), c(N).$

$a(N) \rightarrow [a], a(N1), \{N \text{ is } N1 + 1\}.$

$a(0) \rightarrow [].$

$b(N) \rightarrow [b], b(N1), \{N \text{ is } N1 + 1\}.$

$b(0) \rightarrow [].$

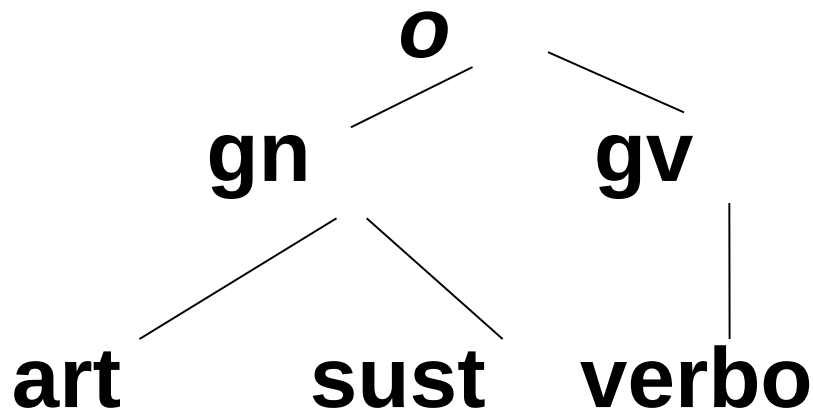
$c(N) \rightarrow [c], c(N1), \{N \text{ is } N1 + 1\}.$

$c(0) \rightarrow [].$

DCG para el lenguaje natural

Gramática para una estructura simple del español:

El niño corre.



DCG para el lenguaje natural

Gramática para la oración:

El niño corre.

oracion --> gn, gv.

gn --> art, sust.

gv --> verbo.

art --> [el].

sust --> [niño].

verbo --> [corre].

DCG para el lenguaje natural

- Esta gramática solamente reconoce la oración del ejemplo.
- Para que reconozca otras oraciones con la misma estructura hay que generar más reglas para art, sust y verbo.

El hombre lee.

El estudiante estudia.

El niño come.

¿Qué pasa si agregamos además : *Los niños leen ?*

DCG para el lenguaje natural

Para que la gramática reconozca la oración:

Los niños corren.

Pero no reconozca:

* *El niños corre.*

* *El niño corren.*

DCG para el lenguaje natural

Para que la gramática reconozca la oración:

Los niños corren.

Pero no reconozca:

* *El niños corre.*

* *El niño corren.*

Hay que manejar un argumento extra que garantice la concordancia de número, por un lado, entre gn y gv, por otro lado, dentro del gn.

DCG para el lenguaje natural

Gramática para oraciones con variación en el número

oracion --> gn(N), gv(N).

gn(N) --> art(N), sust(N).

gv(N) --> verbo(N).

art(sg) --> [el].

art(pl) --> [los].

sust(sg) --> [niño].

sust(pl) --> [niños].

verbo(sg) --> [corre].

verbo(pl) --> [corren].

DCG para el lenguaje natural

- Para controlar la concordancia de género, se trabaja con un nuevo argumento.
- En este caso, sólo se chequea la concordancia entre art y sust.

DCG para el lenguaje natural

Llamadas a predicados para introducir el léxico

- Podemos tener una sola regla para cada categoría léxica:
 - `sust(N) --> [Sust], {es_sust(N, Sust)}.`
- E introducir el léxico como hechos:
 - `es_art(sg, el).`
 - `es_art(pl, los).`
 - `es_sust(sg, niño).`

DCG para el lenguaje natural

Gramática con una sola regla por categoría y llamadas a predicados

oracion --> gn(N), gv(N).

gn(N) --> art(N), sust(N).

gv(N) --> verbo(N).

gv(N) --> verbo(N), gn(N1).

art(N) --> [Art], {es_art(N, Art)}.

sust(N) --> [Sust], {es_sust(N, Sust)}.

verbo(N) --> [Ver], {es_ver(N, Ver)}.

DCG para el lenguaje natural

Gramática con gv más complejo : *Los niños comen el postre.*

```
oracion --> gn(N), gv(N).  
gn(N) --> art(N), sust(N).  
gv(N) --> verbo(N).  
gv(N) --> verbo(N), gn(_).  
  
art(sg) --> [el].  
art(pl) --> [los].  
sust(sg) --> [niño].  
sust(pl) --> [niños].  
sust(sg) --> [postre].  
sust(pl) --> [postres].  
verbo(sg) --> [come].  
verbo(pl) --> [comen].
```

Se controla concordancia sujeto verbo.

Puede haber anomalías semánticas

DCG para el lenguaje natural

El símbolo inicial puede tener argumentos:

oracion(N) --> gn(N), gv(N).

...

...

DCG: Traducción a predicados

- Traducción de las DCGs a predicados “comunes”.
- El intérprete traduce las DCGs (las reconoce por sintaxis) a predicados Prolog.
- Asumimos que la gramática trabaja sobre una oración representada como lista de palabras. Las palabras son términos ya que no se analiza su estructura interna.
- Cada símbolo no terminal va a generar un predicado.

DCG: Traducción a predicados

Una traducción “natural” sería:

```
oracion(O) :-  
    gn(L1),  
    gv(L2),  
    append(L1, L2, O).
```

```
gn(L) :-  
    art(L1),  
    sust(L2),  
    append(L1, L2, L).
```

```
gv(L) :-  
    verbo(L).  
art([el]).  
sust([niño]).  
verbo([corre]).
```

DCG: Traducción a predicados

Una traducción “natural” sería:

```
oracion(O) :-  
    gn(L1),  
    gv(L2),  
    append(L1, L2, O).
```

```
gn(L) :-  
    art(L1),  
    sust(L2),  
    append(L1, L2, L).
```

```
gv(L) :-  
    verbo(L).  
art([el]).  
sust([niño]).  
verbo([corre]).
```

INCONVENIENTE:

Se consulta en el léxico con variables sin instanciar. El léxico es grande normalmente, muchas opciones antes de ver la palabra que realmente viene.

DCG: Traducción a predicados

Otra traducción “natural” sería:

```
oracion(O) :-  
    append(L1, L2, O),  
    gn(L1),  
    gv(L2).
```

```
gn(L) :-  
    append(L1, L2, L),  
    art(L1),  
    sust(L2).
```

```
gv(L) :-  
    verbo(L).
```

```
art([el]).  
sust([niño]).  
verbo([corre]).
```

DCG: Traducción a predicados

Otra traducción “natural” sería:

```
oracion(O) :-  
    append(L1, L2, O),  
    gn(L1),  
    gv(L2).
```

```
gn(L) :-  
    append(L1, L2, L),  
    art(L1),  
    sust(L2).
```

```
gv(L) :-  
    verbo(L).
```

```
art([el]).  
sust([niño]).  
verbo([corre]).
```

INCONVENIENTE:

Se generan múltiples modos de partir la oración de entrada.

Muchos de ellos son inútiles.

DCG: Traducción a predicados

PROBLEMA: Ineficiencia del predicado *append*.

Para la consulta: ?- oracion([el, niño, corre]).

Se instancia la variable O pero L1 y L2 quedan sin instanciar. Esto genera las siguientes posibilidades:

L1 = []

L2 = [el, niño, corre]

L1 = [el]

L2 = [niño, corre]

L1 = [el, niño]

L2 = [corre]

Recién en el tercer intento la cláusula $gn(L1)$ va a ser verdadera. Ídem para art y sust.

DCG: Traducción a predicados

Solución: Se representan las listas como 2 argumentos:

- La lista completa
- El resto

oracion(O, R) :-

 gn(O, R1),
 gv(R1, R).

DCG: Traducción a predicados

```
[ oracion ] [ resto ]  
[ gn ] [ resto1 ]  
      [ gv ] [ resto ]
```

Si se llama al predicado instanciando R con [], entonces:

```
[ oracion ] []  
[ gn ] [ resto1 ]  
      [ gv ] []
```

DCG: Traducción a predicados

El programa completo queda:

```
oracion(O, R) :-  
    gn(O, R1),  
    gv(R1, R).  
gn(L, R) :-  
    art(L, R1),  
    sust(R1, R).  
gv(L, R) :-  
    verbo(L, R).  
art([e|_], R).  
sust([niño|_], R).  
verbo([corre|_], R).
```

DCG: Traducción a predicados

Si agregamos control de concordancia de número, el predicado *oracion* quedaría:

```
oracion(O, R) :-  
    gn(O, R1, N),  
    gv(R1, R, N).
```

DCG: Traducción a predicados

Consultas:

?- oracion([el,niño, corre],[]).

?- oracion(O, []).

Usando el predicado *phrase/2*:

phrase(oracion, [el niño,corre]).

phrase(oracion, O).

phrase/3.

Similar a *phrase/2*

El 3er argumento es el resto luego de reconocer

DCG: Otros ejemplos

1. Números escritos con letras, del cero al cien

cero, dos, veinticinco, cuarenta y ocho, ...

2. Cambio de formato de fecha

3/2/2004 → 3 de enero de 2004

DCG:

Reconocimiento y generación

Analizar si todos los ejemplos que vimos sirven tanto para reconocer como para generar tiras.

Referencias

- **Principales**

- Clocksin, W. F. and Mellish, C. S., *Programming in Prolog*, 3rd Edition, Springer-Verlag, New York, 1987.
- Sterling, L. and Shapiro, E., *The Art of Prolog*, 2nd Edition, MIT Press, Cambridge, Massachusetts, 1994.

- **Complementarias**

- O'Keefe, R. A., *The Craft of Prolog*, MIT Press, Cambridge, Massachusetts, 1990.