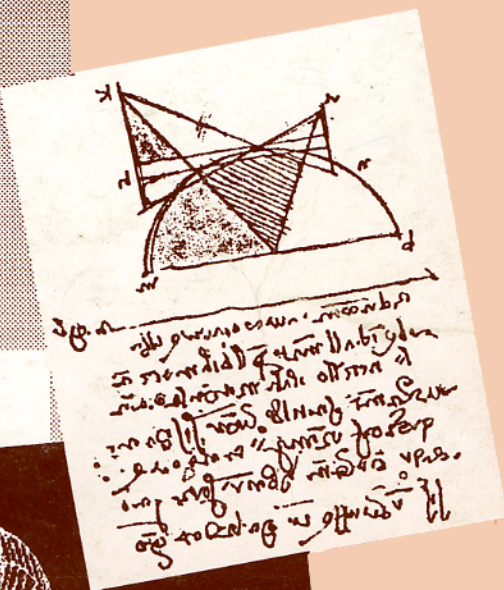
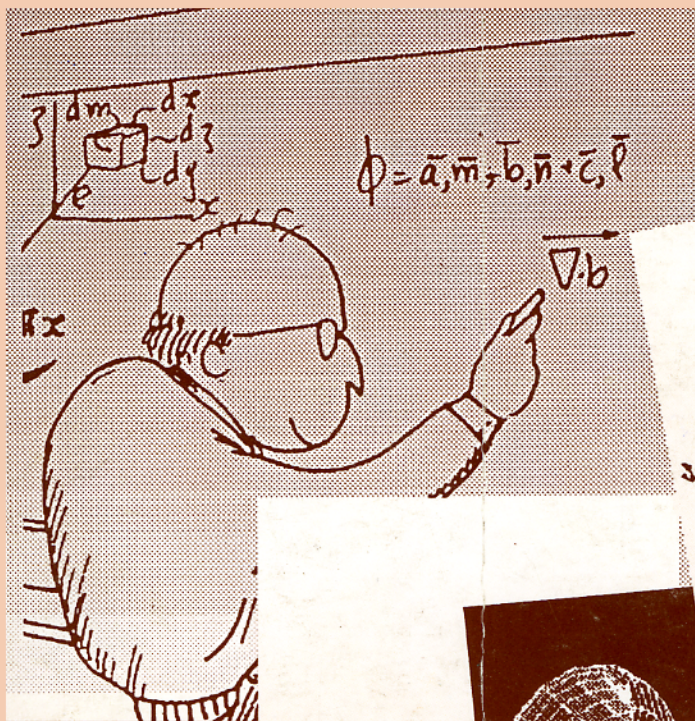


Ingeniería

Revista de la Universidad de Costa Rica
ENERO/JUNIO 1993 VOLUMEN 3 No. 1



UN PROGRAMA PARA LA DEFINICION DEL GENERO GRAMATICAL DE LOS SUSTANTIVOS DE LA LENGUA ESPAÑOLA

*Nicolaos Cosmas **
*Mauricio Molina Delgado **
*Alex Montero Hernández **
*Juan Carlos Porras Valladares **

RESUMEN:

En este artículo se presenta un programa para la definición del género gramatical de los sustantivos. Los datos de entrada son los sustantivos y a la salida se obtiene la definición del género de cada uno de ellos. De hecho, se programa el modelo algebraico lingüístico del género gramatical aplicado a la lengua española, así como fue elaborado en Cosmas, 1992.

SUMMARY:

In this paper a program is given to define grammatical gender of nouns. Nouns constitute the input data, and as output the gender definition of each one of them is obtained. In fact, the algebraic linguistic model of gender applied to Spanish is programmed, in the way in which this model was worked out in Cosmas, 1992.

Introducción

Aquí describimos en forma sintética las ideas y nociones del modelo algebraico del género gramatical, el cual se basa en dos aspectos principales, el matemático y el lingüístico. (Para los detalles matemáticos de esto ver Cosmas op. cit.). El aspecto lingüístico consiste en el paso del género natural de las palabras al género gramatical. Para hacer esta operación se eligen dos sustantivos que representan el prototipo del género natural, por ejemplo, en español **hombre** y **mujer**. En algunas lenguas esta operación no es tan simple como parece a primera vista, pues hay casos como el de la palabra **das Weib** (mujer) en lengua alemana, en la que el género gramatical no coincide con el género natural. Luego se establece una relación entre los sustantivos masculinos y femeninos (como género gramatical, indiferentemente de su género natural) por una parte y, respectivamente, el prototipo masculino y femenino, por otra. Para obtener esta relación se usan conceptos matemáticos de **célula mixta** y **cadena**.

Así, cada palabra en una lengua natural, se asocia con dos conjuntos de palabras: el conjunto de las formas flexionales y el conjunto de las

palabras que aparecen en el mismo contexto con la palabra respectiva. El primer conjunto representa el **paradigma de la palabra dada**, mientras que el segundo conjunto constituye la **clase de distribución de la palabra**. Esto es la dicotomía lingüística de Ferdinand de Saussure **sintagma/paradigma** en la que las relaciones que unen los términos lingüísticos pueden desarrollarse en dos planos que corresponden a dos formas de la actividad mental.

Las formas flexionales de cada palabra son tomadas como bien determinadas, pues los paradigmas se consideran conocidos. En lo relacionado con las clases de distribución, depende de los sintagmas tomados en cuenta. Por ejemplo el paradigma de la palabra **casa** es {**casa, casas**} y su clase de distribución es {**casa, mesa, cama,...**} en el caso de los sintagmas de tipo **adjetivo cualitativo en grado positivo + sustantivo**. Por ejemplo **la bella + S** donde **S = casa, mesa, cama, etc.**, pues es correcto decir **la bella casa, la bella mesa, etc.**, mientras que no es correcto decir **la bella casas, etc.**

De esta manera, con base en este ejemplo de la lengua española, podemos construir una serie de palabras que, de forma intuitiva, ilustra la

noción de **cadena**. Esta aparece como una combinación, una alternancia del paradigma de la palabra respectiva con su clase de distribución. Por ejemplo **casas - casa - mesa** es una cadena que hace la relación entre **casas** y **mesa**, pues la palabra común a ambos conjuntos antes citados es **casa** y es ella la que permite la unión de las palabras en una cadena. El número de términos de la cadena se llama **longitud**, por consiguiente la longitud de la cadena anterior es igual a 3.

El uso de las cadenas para definir el género gramatical aparece normal, puesto que ellas hacen la relación en los dos aspectos fundamentales de la gramática, el paradigmático y el sintagmático. De este modo, utilizando las nociones de cadena y de longitud, podemos describir el género gramatical masculino y el género gramatical femenino y el paso del género natural al género gramatical. Por otra parte, las reglas conforme a las que se hace esta determinación del género gramatical de los dos sustantivos fueron construidas con base en un gran número de pruebas. A continuación mostraremos que estas reglas tienen también validez en la lengua española.

Antes de presentar el modelo que formaliza estas reglas, daremos su descripción no formal.

Un sustantivo es de **género gramatical masculino** si cada palabra de su paradigma se puede ligar a cualquier palabra del paradigma del prototipo masculino por medio de una cadena de longitud máxima igual a 3.

Un sustantivo es de **género gramatical femenino** si cada palabra de su paradigma se puede ligar a cualquier palabra del paradigma del prototipo femenino por medio de una cadena de longitud máxima igual a 3.

Tautológicamente el prototipo masculino es de género gramatical masculino y el prototipo femenino es de género gramatical femenino.

Un sustantivo es de **género gramatical neutro** si no es ni masculino ni femenino.

Un sustantivo tiene **género gramatical doble** si es tanto masculino como femenino.

Así pues, hasta tanto no exista otra propuesta, el conjunto de las frases que consideremos para definir las clases de distribución será del tipo **pronombre demostrativo + verbo + sustantivo**. Un ejemplo de pronombre demostrativo es **éste**, el cual tiene cinco formas flexionales: **éste, ésta, esto, éstos, éstas**.

Como prototipo del género natural hemos optado para el masculino **el padre** y para el femenino **la madre**. Sus paradigmas son respectivamente (el padre, los padres) y (la madre, las madres).

Aplicando las reglas del modelo, se construyen cadenas con el prototipo masculino y el prototipo femenino y según la longitud de esas cadenas se define el género respectivo.

Por ejemplo si consideramos el sustantivo **el sol** cuyo paradigma comprende dos formas flexionales: (el sol, los soles), como consecuencia de que también el paradigma del prototipo consta de dos formas flexionales, resultan las siguientes $2 \times 2 = 4$ cadenas:

1. el sol - el padre.
2. el sol - el padre - los padres.
3. los soles - el sol - el padre.
4. los soles - los padres.

Constatamos que estas cadenas tienen una longitud menor o igual a 3 y por lo tanto el sol es de género gramatical masculino. No tienen género doble (ambiguo y común) pues no existe una cadena que una palabras de este paradigma con las del paradigma femenino. Así pues, no es de género doble (ambiguo y común).

El programa, utilizando estas nociones, crea cadenas cuya longitud determina el género gramatical de las palabras.

La programación en PROLOG del modelo del género gramatical

El lenguaje usado es el PROLOG en la versión 1.0 de Arity Standard. El ambiente del desarrollo del programa es el sistema operativo MS DOS. En caso de que el usuario utilice otro ambiente, deberá hacer referencia a las convenciones válidas para éste.

El programa está constituido por tres partes fundamentales: una parte está destinada a la introducción de los datos, en este caso el de los sustantivos y a su almacenamiento para crear una base de datos. En dicha base de datos están incluidas las particiones en paradigmas y en clases de distribución. La segunda parte, que constituye la más importante, programa el modelo matemático del género gramatical y es aquí donde se define el género gramatical de los sustantivos. Una última parte es de utilidad y sirve para borrar sustantivos innecesarios, para revisar palabras y para dar una visión completa de la base de datos o una parte de ella.

En la primera parte se introducen los sustantivos y se crea la base de datos.

Esta parte del programa está destinada a la lectura de oraciones del lenguaje natural de tipo "este es el ...", "esta es la ...", "estos son los ..." Para lo cual se definieron operadores convenientes que son este, esta, esto, estos, estas, el, la, los, las, lo. De todo modo, la entrada de datos se lleva a cabo mediante el predicado "entre", de la siguiente forma, utilizando oraciones bien formadas en español:

entre(este es el 'sustantivo en singular', estos son los 'sustantivo en plural')

entre(esta es la 'sustantivo en singular', estos son los 'sustantivo en plural')

entre(esto es 'adjetivo sustantivado'). Se nota que en dicho caso no se incluye el plural puesto que no tiene sentido dentro del género neutro. (Se trata aquí una variación de lo expuesto anteriormente en

el sentido de que el neutro no se determina como una palabra que no es ni masculina ni femenina, sino según algunas propiedades de la descripción matemática [Cosmas, 1992].

Estas entradas se darán iterativamente en el intérprete de Prolog y siempre tendrán éxito a menos que la palabra que se intenta incluir ya exista en la base de datos. Estas palabras son almacenadas en un archivo llamado "bdatos2.arp" que en principio (y en todo momento) debe tener almacenados los prototipos de cada género: padre para el masculino, madre para el femenino y "lo" paterno, para el neutro, así como sus plurales, excepto el del neutro.

Al recibir una oración de este tipo, se realiza el siguiente algoritmo:

-Si las palabras para ser introducidas ya están almacenadas, fracasa el intento de almacenamiento.

-Si las palabras no están en la base de datos y el predicado utilizado invoca oraciones de tipo "este es el A, estos son los A'", se introduce A en la clase de distribución S(padre) del prototipo masculino, A' en la clase de distribución S(padres) del plural del prototipo masculino y se agrega un nuevo paradigma P(A, A'). Se guarda físicamente el archivo modificado bdatos2.arp. (Donde A y A' indican los sustantivos en singular y plural respectivamente).

-Si una o ambas palabras que el usuario intenta incluir en la base de datos ya están almacenadas con género femenino o neutro (son los casos de doble género o de error) y el predicado invocado es de tipo entre('este es el A, estos son los A'), se introduce una palabra A modificada en la clase de distribución del prototipo masculino S(padre), se introduce A' modificada en la clase de distribución del plural del prototipo masculino S(padres) y se agrega un nuevo paradigma P(A, A'). Se guarda físicamente el archivo modificado bdatos2.arp.

(En la práctica, al almacenar una palabra modificada se usó la palabra original junto a -1. En los ejemplos se puede ver con más claridad esto, aunque resulta transparente al usuario).

- Si las palabras no están en la base de datos, y el predicado utilizado invoca oraciones de tipo "esta es la B, estas son las B'", se introduce B en la clase de distribución del prototipo femenino S(madre), B' en la clase de distribución del plural del prototipo femenino S(madres) y se agrega un nuevo paradigma P(B, B'). Se guarda físicamente el archivo modificado bdatos2.arp. (Donde B y B' indican los sustantivos en singular y plural respectivamente).
- Si una o ambas palabras que el usuario intenta incluir en la base de datos ya están almacenadas con género masculino o neutro y el predicado invocado es de tipo entre("esta es la B, estas son las B'), se introduce una palabra B modificada en la clase de distribución del prototipo femenino S(madre), se introduce B' modificada en la clase de distribución del plural del prototipo femenino S(madres) y se agrega un nuevo paradigma P(B, B'). Se guarda físicamente el archivo modificado bdatos2.arp.
- Si las palabras no están en la base de datos, y el predicado utilizado invoca oraciones de tipo "esto es lo C", se introduce C en la clase de distribución del prototipo neutro S("lo" paterno) y se agrega un nuevo paradigma P(C). Se guarda físicamente el archivo modificado bdatos2.arp. (Donde C indica el adjetivo sustantivado en singular).
- Si la palabra que el usuario intenta incluir en la base de datos ya está almacenada con género masculino o femenino y el predicado invocado es de tipo entre(estos es lo C), se introduce una palabra C modificada en la clase de distribución del prototipo neutro S("lo" paterno) y se agrega un nuevo paradigma P(C). Se guarda físicamente el archivo modificado bdatos2.arp.

En la segunda parte del programa se hace la consulta del género de una palabra previamente incluida en la base de datos. El programa utiliza el algoritmo del modelo matemático del género buscando cadenas entre el prototipo de cada género y el sustantivo introducido. Para conocer el género de un sustantivo se digita:

genero(sustantivo,X).

después de lo cual el programa:

1. Intenta una cadena de longitud menor o igual a tres entre la palabra y el prototipo masculino (padre). Si esto se verifica da como solución **X=masculino**.
2. Intenta una cadena de longitud menor o igual a tres entre la palabra y el prototipo femenino (madre). Si esto se verifica da como solución **X=femenino**.
3. Intenta una cadena de longitud menor o igual a tres entre la palabra y el prototipo del género neutro("lo" paterno). Si esto se verifica da como solución **X=neutro**.

Debe notarse que es posible que una palabra dé como solución varios géneros. Por ejemplo, una palabra de género doble, debe dar como solución masculino y femenino. En este sentido en el PROLOG hay la posibilidad de varias respuestas. Después de la primera, el usuario puede solicitar otra por medio de un punto y coma ';' y el programa responderá el otro género, o bien responderá un 'no' si este no existiera.

Las cadenas entre las palabras se definen recursivamente. Existe una cadena entre dos palabras cuando ambas están en el mismo paradigma, en la misma clase de distribución o cuando existe una tercera palabra que sirve de eslabón y unión entre dos palabras que están en diferente paradigma y diferente clase de distribución.

La última parte del programa es de utilidad para el usuario, pues le permite borrar y revisar. Con la indicación:

borre(sustantivo).

puede borrar cualquier palabra excepto los tres prototipos. El borrar una palabra implica borrar también su plural. Cuando se desea borrar una palabra con género doble, al darle la instrucción solamente se pierde uno de los dos géneros que tenía. Para borrar el otro género se le debe dar la instrucción:

borre(sustantivo-1).

Se borra primero el primer género que se introdujo. Borrar una palabra equivale a sacarla (a ella y a su plural) de su paradigma y de su clase de distribución.

Para revisar cuáles palabras están en la base de datos puede usarse la instrucción 'ver', junto con uno de los tres parámetros **todo**, **paradigma**, **distribución**. La instrucción:

ver(todo).

despliega toda la base de datos, la partición paradigmática y clases de distribución. Con la instrucción:

ver(paradigma).

solamente se muestra la partición paradigmática. Con la instrucción:

ver(distribución).

se obtienen las palabras en sus respectivas clases de distribución.

El código fuente del programa

```
-op(100,fx,este).
-op(100,fx,esta).
-op(100,fx,esto).
-op(100,fx,estos).
-op(100,fx,estas).
-op(60,fx,es).
-op(60,fx,son).
-op(50,fx,la).
-op(50,fx,el).
-op(50,fx,lo).
-op(50,fx,los).
-op(50,fx,las).
```

```
member(A,[A|B]).
```

```
member(A,[B|C]) :-
    member(A,C).
```

```
append([],A,A).
append([A|B],C,D) :-
    append(B,C,E),
    [A|E] = D.
```

```
incluir(A) :-
    incluido(A),
    !.
```

```
incluir(A) :-
    nom_ext(A,B),
    [B],
    assert(incluido(A)).
```

```
f('bdatos2.arp').
:-consult('bdatos2.arp').
```

```
saca(P,[],[]):-!.
saca(P,[P|L],L):-!.
saca(P,[B|L],[B|L2]):-saca(P,L,L2).
```

```
borre(P):-prototipo(X,P),
    !,fail.
```

```
borre(P):-p(X),
    member(P,X),
    member(Q,X),
    not(Q=P),
    retract(p(X)),
    s(Y),
    member(P,Y),
    saca(P,Y,Z),
    retract(s(Y)),
    assert(s(Z)),
    s(W),
    member(Q,W),
    saca(Q,W,V),
    retract(s(W)),
    assert(s(V)),
    f(G),
    tell(G),
    listing(p),
    listing(s),
    told,
    tell(user),!.
```

```
borre(P):-p(X),
  member(P,X),
  retract(p(X)),
  s(Y),
  member(P,Y),
  saca(P,Y,Z),
  retract(s(Y)),
  assert(s(Z)),
  f(G),
  tell(G),
  listing(p),
  listing(s),
  told,
  tell(user),!.
```

```
no_esta(A):-s(X),
  member(A,X),!,fail.
no_esta(A).
```

```
entre(este es el A,estos son los B):-
  genero(A,masculino),
  genero(B,masculino),
  !,fail.
```

```
entre(este es el A,estos son los B) :-
  no_esta(A),
  no_esta(B),
  s([padre|C]),
  append([padre|C],[A],D),
  retract(s([padre|C])),
  assertz(s(D)),
  s([padres|E]),
  append([padres|E],[B],F),
  retract(s([padres|E])),
  assertz(s(F)),
  assertz(p([A,B])),
  f(G),
  tell(G),
  listing(p),
  listing(s),
  told,
  tell(user),
  !.
```

```
entre(este es el A,estos son los B) :-
  s([padre|C]),
  append([padre|C],[A-1],D),
  retract(s([padre|C])),
  assertz(s(D)),
  s([padres|E]),
```

```
append([padres|E],[B-1],F),
  retract(s([padres|E])),
  assertz(s(F)),
  assertz(p([A-1,B-1])),
  f(G),
  tell(G),
  listing(p),
  listing(s),
  told,
  tell(user),
  !.
```

```
entre(esta es la A,estas son las B):-
  genero(A,femenino),
  genero(B,femenino),
  !,fail.
```

```
entre(esta es la A,estas son las B) :-
  no_esta(A),
  no_esta(B),
  s([madre|C]),
  append([madre|C],[A],D),
  retract(s([madre|C])),
  assertz(s(D)),
  s([madres|E]),
  append([madres|E],[B],F),
  retract(s([madres|E])),
  assertz(s(F)),
  assertz(p([A,B])),
  f(G),
  tell(G),
  listing(p),
  listing(s),
  told,
  tell(user),
  !.
```

```
entre(esta es la A,estas son las B) :-
  s([madre|C]),
  append([madre|C],[A-1],D),
  retract(s([madre|C])),
  assertz(s(D)),
  s([madres|E]),
  append([madres|E],[B-1],F),
  retract(s([madres|E])),
  assertz(s(F)),
  assertz(p([A-1,B-1])),
  f(G),
  tell(G),
  listing(p),
```

```
listing(s),
told,
tell(user),
!.
entre(esto es lo A):-genero(A,neutro),
!.fail.
```

```
entre(esto es lo A) :-
no_esta(A),
s([B|C]),
not B = padre,
not B = madre,
not B = padres,
not B = madres,
append([B|C],[A],D),
retract(s([B|C])),
assertz(s(D)),
assertz(p([A])),
f(E),
tell(E),
listing(p),
listing(s),
told,
tell(user),!.
```

```
entre(esto es lo A) :-
genero(A,G),
not(G=neutro),
s([B|C]),
not B = padre,
not B = madre,
not B = padres,
not B = madres,
append([B|C],[A-1],D),
retract(s([B|C])),
assertz(s(D)),
assertz(p([A-1])),
f(E),
tell(E),
listing(p),
listing(s),
told,
tell(user).
```

```
prototipo(masculino,padre).
prototipo(femenino,madre).
prototipo(neutro,paterno).
```

```
cadena(A,B,1,C) :-
distribucion(A,B),
!.
```

```
cadena(A,B,1,C) :-
paradigma(A,B),
!.
```

```
cadena(A,B,C,D) :-
D < 2,
distribucion(A,E),
not A = E,
cadena(E,B,F,D + 1),
C is F + 1.
```

```
cadena(A,B,C,D) :-
D < 2,
paradigma(A,E),
not A = E,
cadena(E,B,F,D + 1),
C is F + 1.
```

```
distribucion(A,B) :-
s(C),
miembro(A,C),
miembro(B,C).
```

```
paradigma(A,B) :-
p(C),
miembro(A,C),
miembro(B,C).
```

```
miembro(A,[A|B]).
miembro(A,[B|C]) :-
miembro(A,C).
```

```
menor([]) : fail , !.
```

```
menor([A|B]) :-
A < 4,
!.
```

```
menor([A|B]) :-
menor(B).
```

```
genero(A,masculino) :-
findall(B,cadena(A,padre,B,0),C),
menor(C),
not(C=[]).
```

```
genero(A,femenino) :-
findall(B,cadena(A,madre,B,0),C),
```


menor(C),
not(C=[]).

genero(A,neutro) :-
p([A]).

genero(A,G):-p(X),
miembro(A-1,X),
genero(A-1,G).

ver(paradigma):-listing(p).
ver(distribucion):-listing(s).
ver(todo):-listing(p),
listing(s).

Nota

Se programó aquí según la descripción matemática (definiciones y teoremas) [Cosmas, 1992] del modelo. En este sentido el neutro se define según la longitud de las cadenas respecto a un prototipo y no según el aspecto lingüístico del modelo que define al neutro como aquella palabra que no es ni masculina ni femenina.

Para un estudio análogo, sobre el género gramatical en la lengua portuguesa, ver Zumbado (1992), donde el programa se presenta de una forma mejorada y muy accesible al usuario a través de un menú desde el cual se pueden manejar todas las opciones posibles.

BIBLIOGRAFÍA

Cosmas, N. "El modelo lingüístico matemático del género gramatical y su aplicación en la lengua española". (Investigación n. 606-91-310, Universidad de Costa Rica, 1991). Revista de Filología y Lingüística, Universidad de Costa Rica, XVIII, 2, 1992

Cosmas, N. & S. Marcus. *Lingüística matemática y lingüística general*. Por publicarse.

Zumbado L. , Avila E. , "El modelo matemático del género gramatical aplicado a la lengua portuguesa". Programa de Posgrado en Ciencias Cognoscitivas, Universidad de Costa Rica, 1992. Por publicarse.