

Aproximación al tratamiento informático de las perífrasis verbales con infinitivo en español

An Approach to the Computational Treatment of Spanish Verb Periphrasis with Infinitive

Rodolfo Bonino, Walter Koza y Stella Maris Moro

Equipo INFOSUR

Universidad Nacional de Rosario

Rosario, Argentina

rodolfobonino@yahoo.com.ar,

walter_koza@yahoo.com.ar,

smmoro@yahoo.com.ar

Resumen

El presente trabajo se propone la implantación en máquina de reglas para el tratamiento informático de frases verbales núcleo con perífrasis. A tales efectos, se recurre a la herramienta *xfst* (Xerox, 1997), que opera sobre cadenas de caracteres, a las que asigna categorías previamente declaradas, para luego dar lugar a la producción y análisis automáticos de sintagmas bien formados del lenguaje natural. Para ello, es necesario declarar un conjunto de reglas que interactúan determinando combinaciones posibles de categorías.

En función de las exigencias de esta herramienta, se presenta un repertorio de reglas que permiten procesar sintagmas verbales núcleos que incluyen, además del núcleo verbal, auxiliares de tiempo compuesto y de pasiva, y auxiliares modales de articulación directa e indirecta, con sus respectivas negaciones. Se definen además reglas para incorporar el tratamiento de los clíticos, en relación con los distintos lugares que pueden ocupar en el sintagma.

Finalmente, se presentan algunas posibles proyecciones de esta propuesta.

Palabras claves: lingüística computacional, procesamiento automático, frase verbal, perífrasis, español.

Abstract

This paper aims at implanting in a computer system the rules for the computational treatment of nuclei verb syntagma with periphrasis. We use *xfst* (Xerox, 1997), a tool that operates on character chains to which it assigns previously declared categories to render the automatic production and analysis of well-formed syntagmas in a natural language. With this purpose it is necessary to declare a set of interacting rules that determine plausible category combinations.

According to this tool's requirements, we present a rule repertoire that allows for the processing of nuclei verb syntagmas including the verb nucleus, compound tense and passive auxiliaries, and modal auxiliaries of direct and indirect articulation, with their respective negations. We also define rules that incorporate the treatment of clitics in relation to their different occurrence positions in the syntagma.

Finally, some possible implications of this proposal will be presented.

Keywords: Computational Linguistics, Automatic Analysis, Verb Phrase, Periphrasis, Spanish.

1. INTRODUCCION

En este trabajo intentamos acercarnos a la utilización de herramientas informáticas para la generación y análisis de perífrasis verbales con auxiliar flexionado y núcleo infinitivo en español.

En el marco de las gramáticas no computacionales, Gómez Torrego (1999) define las perífrasis verbales del siguiente modo:

Una perífrasis verbal es la unión de dos o más verbos que constituyen un solo 'núcleo' del predicado. El primer verbo, llamado 'auxiliar', comporta las informaciones morfológicas de número y persona, y se conjuga en todas (o en parte de) las formas o tiempos de la conjugación. El segundo verbo, llamado 'principal' o 'auxiliado' debe aparecer en infinitivo, gerundio o participio, es decir, en una forma no personal. Según se trate de una u otras formas, hablamos de perífrasis verbales de infinitivo, de gerundio y de participio. (p. 3325. Comillas en el original)

Nuestra propuesta es establecer hipótesis acerca de las propiedades de este objeto de estudio en el marco del Paradigma 5P, elaborado por Gabriel Bès (1999: 280-281).

El paradigma 5P es una propuesta de formalización para la aplicación del método deductivo-inductivo de las ciencias empíricas al estudio del lenguaje. Según Bès, el método implica los siguientes niveles:

- P1 o P de Protocolos; un Protocolo es la representación de un dato al cual llega un Observador debidamente modelado. Este dato puede haber sido observado o no en un corpus efectivo.
- P2 o P de Propiedades; una Propiedad puede identificarse formalmente con un axioma. Un conjunto finito de Propiedades especifica intencionalmente un conjunto (finito o infinito) de secuencias de expresiones de una lengua determinada. Llamaremos modelos a estas secuencias. Un modelo es una secuencia que satisface un conjunto de Propiedades. Los modelos serán asociados a los enunciados de la lengua descripta.
- P3 o P de Proyecciones; una Proyección es una abstracción de un conjunto de Propiedades de una lengua, que especifica las características comunes a todas estas Propiedades.
- P4 o P de Principios; un Principio –al igual que una Propiedad– puede identificarse formalmente con un axioma, siendo los Principios más abstractos y generales que las propiedades. Los Principios van a introducir restricciones generales, válidas para todas las lenguas o para grupos de lengua, sobre el tipo de Propiedades –y, por lo tanto, de los modelos– que describen las lenguas particulares.
- P5 o P de Procesos; un Proceso es un procedimiento efectivo implantado en máquina o que puede serlo y que permite analizar y/o producir un enunciado de una lengua particular. Este enunciado está asociado a un modelo que satisface las Propiedades (P2) que describen esta lengua particular.

Para este desarrollo de las propiedades, nos centramos en el nivel P2 y nos basamos en los conjuntos de propiedades formuladas por Gabriel Bès (Ib.) para el estudio de la frase verbal con núcleo flexionado del francés, que resultan adecuadas al tratamiento del tipo de secuencias del español que queremos abordar.

Con la finalidad de validar nuestra formalización, recurrimos a la herramienta *xfst*[1]. En este programa, la aplicación se presenta como una implementación de autómatas de estados finitos para producir análisis morfológicos y generación de cadenas. En el lenguaje informático, las expresiones de una lengua se presentan como secuencias de códigos de caracteres pertenecientes a estándares informáticos (ASCII, UNICODE, etc.); a partir de tales secuencias, la aplicación de un sistema algorítmico, basado en hipótesis lingüísticas, permite analizar y generar expresiones del lenguaje natural.

2. LA FRASE VERBAL NÚCLEO FLEXIONADA (fvnf)

A partir de Bès (1999: 273), definimos la Frase Verbal Núcleo Flexionada como una cadena de palabras que puede comenzar por el adverbio de negación ‘no’, por un pronombre clítico, por un auxiliar o por un verbo flexionado, y que finaliza en un verbo, que es el núcleo de la frase verbal, o por un enclítico.

Como se puede observar, esta definición delimita un objeto mayor a la palabra, con propiedades formales estables.

Se puede precisar que la fvnf, en su forma más simple, está compuesta solo por un verbo flexionado. La fvnf puede expandirse por la aparición de auxiliares de tiempo compuesto, de pasiva y modales. Como hemos citado más arriba, los auxiliares serán los portadores de los rasgos morfológicos de Inflexión. El verbo, ahora en su forma no flexionada, continuará funcionando como núcleo de la nueva estructura expandida, en tanto que no deja de seleccionar los complementos ni de asignarles los rasgos de caso.

En función de las propiedades formales (orden de palabras, exclusiones mutuas, etc.), Bès (op.cit.) también incluye los clíticos, en la fvnf.

Además, se trata el adverbio de negación en posición inicial, aunque en etapas posteriores se podría tratar su colocación en otras posiciones de la cadena, así como también otros adverbios.

3. PERÍFRASIS CON INFINITIVOS

El objeto de estudio de este trabajo puede definirse, entonces, como una ampliación de la fvnf, que incluye un infinitivo con articulación directa o indirecta (*puede escribir, empieza a escribir, tiene que escribir*). La tarea que pretendemos desarrollar es establecer cuáles son las propiedades de este tipo particular de fvnf. Tales propiedades deberán dar cuenta también de las fvnf que no contengan perífrasis.

En una primera etapa, intentamos trabajar con las construcciones más simples: las que contienen sólo el modal, el infinitivo y, si corresponde, la preposición o la conjunción (incluyendo sus formas negativas): *puede/ tiene que/ empieza a escribir*.

Para la segunda etapa nos propusimos:

- 1) La forma compuesta del infinitivo, excluyendo *empezar a* que la rechaza (*puede/ tiene que haber escrito*).
- 2) La forma pasiva simple del infinitivo (*puede/ tiene que/ empieza a ser escrito*).
- 3) La forma pasiva compuesta del infinitivo, en los casos posibles (*puede/ tiene que haber sido escrito*).

- 4) Las formas compuestas de los modales (*ha podido escribir*).
- 5) La presencia de clíticos.
- 6) Las formas recursivas de varios auxiliares modales (*puede tener que empezar a escribir*).

Quedarán para una posterior etapa de la investigación, los siguientes temas:

- La negación del infinitivo (*puede no escribir*).
- Los problemas derivados de la variación del género de los participios.
- Los problemas derivados de la transitividad/intransitividad verbal.
- Las combinaciones que surgen de todas esas formas.

4. CATEGORÍAS

Como punto de partida se definen las categorías que pueden aparecer como constituyentes de la fvnf que contiene perífrasis verbales. Por razones mnemónicas, para denominar las los rasgos que caracterizan a las categorías se emplean términos utilizados tradicionalmente por la gramática, pero esto no supone la adhesión a ninguna teoría gramatical (bien podrían utilizarse cifras o letras); lo que justifica una clasificación no es el nombre que se asigna a los rasgos sino el hecho de que dichos rasgos den cuenta de oposiciones relevantes para un objeto determinado (y, en última instancia, para una lengua).

Con la definición de las categorías queda explicitada la primera propiedad de existencia:

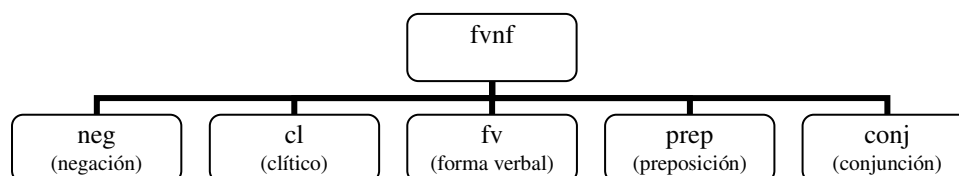
P 1 categ (fvnf, [neg, fv, prep, conj, cl]).

donde “categ” es el nombre del predicado, fvnf es la etiqueta que se asigna a la secuencia trabajada y “neg” (negación), “fv” (forma verbal), “prep” (preposición), “conj” (conjunción) y “cl” (clítico), son los constituyentes.

Vale aclarar que, mientras algunas de estas categorías se definen por un solo rasgo, otras se definen por un conjunto de rasgos. En este último caso decimos que una categoría subsume a otra(s). Por ejemplo, fv subsume “fvv” (forma verbal verbo), “fvser” (forma verbal ser) y “fvaux” (forma verbal auxiliar), esto implica que hay un rasgo común a varias categorías (fv), cada una de las cuales se define por rasgos particulares (v, ser, aux). Llamamos categorías terminales a aquellas que no subsumen a ninguna otra.

En el esquema 1 mostramos los constituyentes de las fvnf que conforman nuestro objeto de estudio:

Esquema 1: Constituyentes de la fvnf



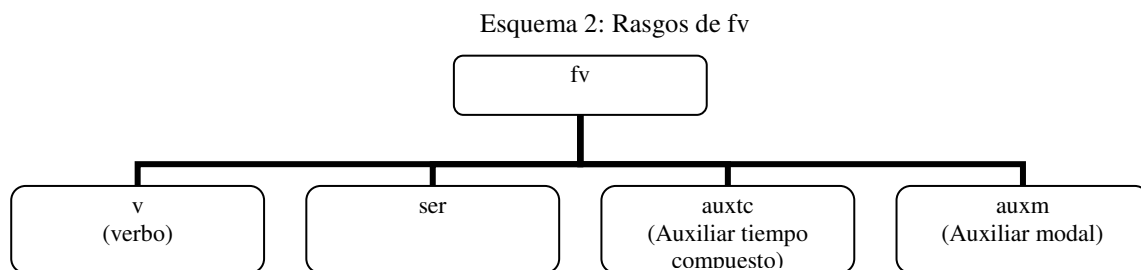
Para nuestro análisis “neg”, “prep” y “conj”, son categorías terminales porque no pretendemos profundizar en las propiedades formales de ninguna de ellas. Si quisiéramos abordar, por ejemplo, el estudio de la negación, además de “no” deberíamos incluir formas como “nadie”, “nunca”, etc., lo que nos obligaría a incorporar distintos rasgos y esto haría que “neg” no fuera una categoría terminal, sino que debería subsumir otros rasgos; lo mismo ocurre con las demás categorías terminales.

En los siguientes esquemas se muestran los rasgos que subsumen las categorías fv y cl, que son las que resultan pertinentes para nuestro objeto de estudio.

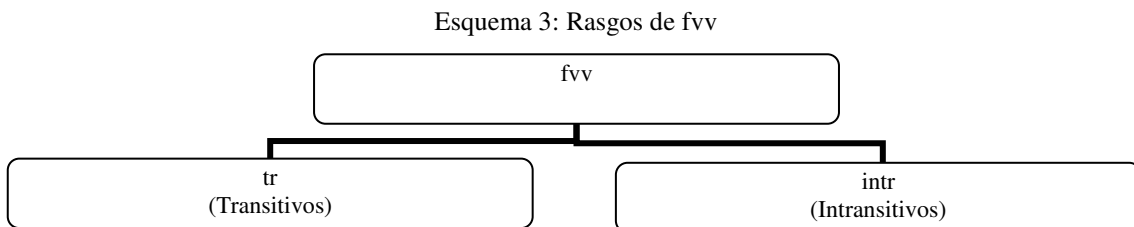
En el esquema 2 se muestra que es necesario precisar varios rasgos distintivos de las formas verbales:

- El rasgo “verbo” caracteriza a las formas verbales que no pueden ser auxiliares.
- El rasgo “ser” se refiere al auxiliar de las perífrasis de pasiva que puede regir participios con variación de género y número (*fue detenido/a, fueron detenidos/as*).
- El rasgo “auxiliar compuesto” corresponde al verbo “haber”, que puede regir participios invariables (*ha/han detenido*).
- El rasgo “auxiliar modal” corresponde a los verbos que rigen infinitivos, con preposición o conjunción o sin ellas (*empezó a escribir, pudo escribir, tuvo que escribir*).

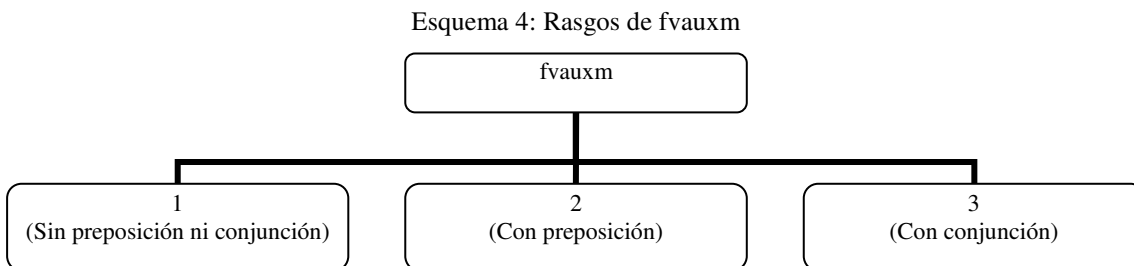
Cabe aclarar que todas las formas verbales que **pueden** ser auxiliares, también pueden ser verbos plenos (*Juan es estudioso; Hubo una fiesta, Todo lo puede, Empezó la función, Tiene dinero*)



En el esquema 3 a la categoría caracterizada por los rasgos “FV” y “V” (“FVV”), se le anexan los rasgos sintácticos y semánticos TR, INTR, que son relevantes para determinar si FVV admite clíticos acusativos y si puede tener como auxiliar a “Ser”.



En el esquema 4 la categoría caracterizada por los rasgos “fv” y “auxm” (“fvauxm”) se subclasifica según cómo se relaciona con los infinitivos (articulación directa, articulación con preposición, articulación con conjunción).



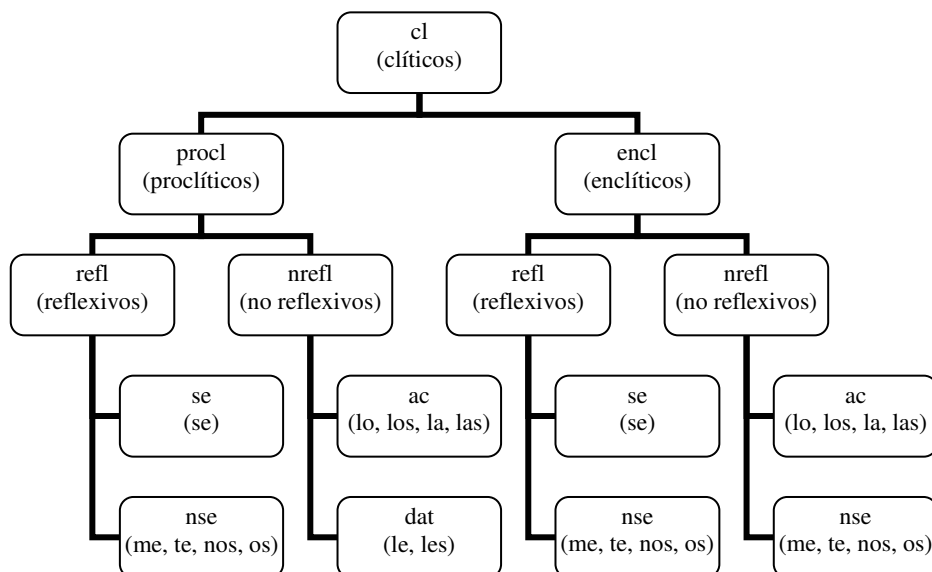
Finalmente se asignan a todas las categorías los rasgos morfológicos que resultan relevantes para el caso, es decir, forma flexionada (f), participio (p), infinitivo (i). Las categorías terminales resultantes de la combinatoria de estos rasgos son las siguientes:

Tabla 1: Categorías Terminales

1. forma verbal (fv), verbo (v), transitivo (tr), flexionado (f) fvvtrf (*escribe*)
2. forma verbal (fv), verbo (v), transitivo (tr), participio (p) fvvtrp (*escrito*)
3. forma verbal (fv), verbo (v), transitivo (tr), infinitivo (i) fvvtri (*escribir*)
4. forma verbal (fv), verbo (v), intransitivo (intr), flexionado (f) fvvintrf (*tose*)
5. forma verbal (fv), verbo (v), intransitivo (intr), participio (p) fvvintrp (*tosido*)
6. forma verbal (fv), verbo (v), intransitivo (intr), infinitivo (i) fvvintr (i) fvvintr (i) (*toser*)
7. forma verbal (fv), ser (ser), flexionado (f) fvserf (*es*)
8. forma verbal (fv), ser (ser), participio (p) fvserp (*sido*)
9. forma verbal (fv), ser (ser), infinitivo (i) fvseri (*ser*)
10. forma verbal (fv), auxiliar tpo. comp. (auxtc), flexionado (f) fvautcf (*ha*)
11. forma verbal (fv), auxiliar tpo. comp. (auxtc), participio (p) fvautcp (*habido*)
12. forma verbal (fv), auxiliar tpo. comp. (auxtc), flexionado (f) fvautci (*haber*)
13. forma verbal (fv), auxiliar modal (auxm), art. directa (1) flexionado (f) fvauxm1f (*puede*)
14. forma verbal (fv), auxiliar modal (auxm), art. directa (1) participio (p) fvauxm1p (*podido*)
15. forma verbal (fv), auxiliar modal (auxm), art. directa (1) infinitivo (i) fvauxm1i (*poder*)
16. forma verbal (fv), auxiliar modal (auxm), c/ prep. (2) flexionado (f) fvauxm2f (*empieza*)
17. forma verbal (fv), auxiliar modal (auxm), c/ prep. (2) participio (p) fvauxm2p (*empezado*)
18. forma verbal (fv), auxiliar modal (auxm), c/ prep. (2) infinitivo (i) fvauxm2i (*empezar*)
19. forma verbal (fv), auxiliar modal (auxm), c/ conj. (3) flexionado (f) fvauxm3f (*tiene*)
20. forma verbal (fv), auxiliar modal (auxm), c/ conj. (3) participio (p) fvauxm3p (*tenido*)
21. forma verbal (fv), auxiliar modal (auxm), c/ conj. (3) infinitivo (i) fvauxm3i (*tener*)

En el esquema 5 se muestran los rasgos que se utilizaron para definir a los clíticos. El criterio general que hemos aplicado es el mismo que utiliza Bès (op. cit), es decir, los clíticos se agrupan por sus propiedades de unicidad: en una frase bien formada del español no puede aparecer más de una categoría terminal. La denominación puede dar lugar a discusiones, que no resultan pertinentes, si se tiene en cuenta lo expresado *supra* en relación con ese tema.

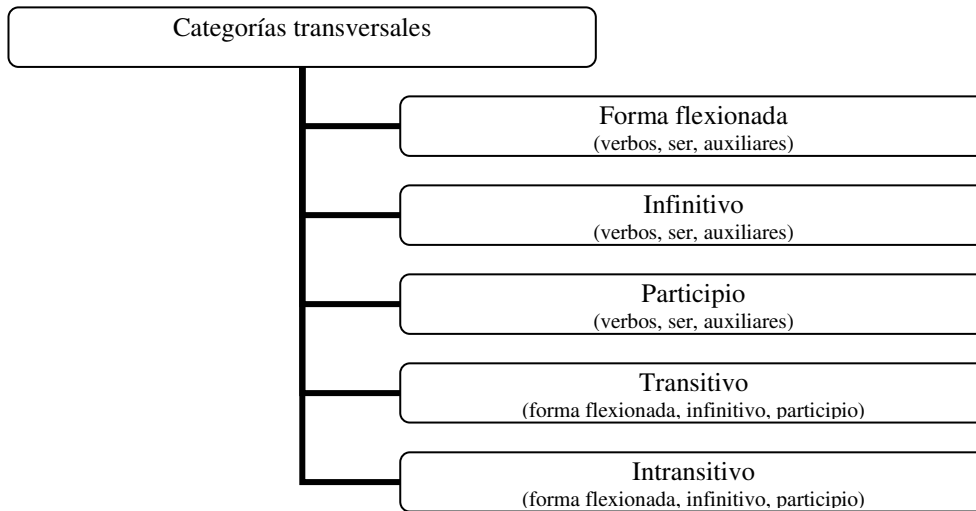
Esquema 5: Rasgos de cl



Además de estas categorías, que pueden asociarse a lo que tradicionalmente se ha llamado “partes de la oración” o “clases de palabras”, resulta empíricamente útil para la formulación de reglas establecer categorías transversales; así, por ejemplo, la definición de la categoría transversal *f* (flexión) era necesaria para luego definir una propiedad de unicidad que determinara una única forma flexionada en la frase verbal.

En el esquema 6, presentamos las categorías transversales que utilizamos en nuestro análisis.

Esquema 6: Categorías Transversales



Las entradas léxicas se etiquetan con las correspondientes categorías terminales. De este modo, una vez implantadas las propiedades, la herramienta informática podrá analizar y generar tanto cadenas de categorías como cadenas del lenguaje natural.

El comando *print labels* da como salida todas las entradas léxicas cargadas en el autómata, con sus correspondientes etiquetas.

Tabla 2: Comando *print labels*. Salida.

```

    <enclnreflac:la> <enclnreflac:las> <enclnreflac:lo> <enclnreflac:los>
    <enclnrefldat:le> <enclnrefldat:les> <enclreflnse:me> <enclreflnse:nos>
    <enclreflnse:te> <enclreflse:se> < conj: que> < fvauxm1f: puede>
    < fvauxm1p: podido> < fvauxm2f: empieza> < fvauxm2p: empezado>
    < fvauxm3f: tiene> < fvauxm3p: tenido> < fvautxcf: ha> < fvautxci: haber>
    < fvserf: es> < fvseri: ser> < fvserp: sido> < fvintrf: tose>
    < fvintrp: toser> < fvintrp: toside> < fvvtrf: escribe>
    < fvvtri: escribir> < fvvtrp: escrito> < neg: no> < prep: a>
    < proclnreflac: la> < proclnreflac: las> < proclnreflac: lo>
    < proclnreflac: los> < proclnrefldat: le> < proclnrefldat: les>
    < proclreflnse: me> < proclreflnse: nos> < proclreflnse: te>
    < proclreflse: se>
  
```

5. FORMULACIÓN DE PROPIEDADES

Si bien el trabajo fue desarrollado en etapas, en este apartado, presentamos las propiedades como quedaron diseñadas al final del proceso.

En primer término, se definieron las propiedades de existencia. Estas propiedades incluyen el alfabeto de categorías necesarias, así como la formulación de reglas que determinan la unicidad, exigencia y exclusión de tales categorías en las cadenas.

En segundo término, se fijaron las propiedades de linealidad (precedencia), que dan cuenta del orden de los constituyentes.

5.1. Propiedades de Unicidad

Determinadas categorías no pueden reiterarse en la cadena. Esto se define a través de propiedades de unicidad (unic):

P2unic (fvnf, [v, ser, neg, prep, conj, i, f, aux1, aux2, aux3, nrefl, nse, se]).

Debe establecerse además qué categoría funcionará como núcleo de la cadena. Esto se define en P3:

P3unicnuc (fvnf, [v, ser, auxm]).

Como puede observarse, esta última propiedad define a la vez, para cualquier secuencia dada:

- la categoría núcleo;
- su unicidad;
- su obligatoriedad.

5.2. Propiedades de Exigencia

En lo que concierne a las reglas de exigencia, se determinaron los elementos que se requerían para obtener fvnf bien formados. Así por ejemplo:

P+1 exig (svnf [P], [auxtc, ser])

Esto significa que el participio (P) exige la presencia de un auxiliar de tiempo compuesto, o bien el verbo “ser”. De la misma manera, se interpretan las siguientes:

P+2 exig (svnf [serp], [auxtc])

P+3 exig (svnf [i], [auxm])

P+4 exig (svnf [prep, conj, auxm], [i])

P+5 exig (svnf [auxm2], [prep])

P+6 exig (svnf [auxm3], [conj])

P+7 exig (svnf [cl], [fv])

P+8 exig (svnf [encl], [i])

5.3. Propiedades de Exclusión

Las propiedades de exigencia se complementan con las de exclusión. A lo largo del trabajo se pudieron determinar las siguientes:

P~1 excl (svnf, [auxm1], [conj, prep])
 P~2 excl (svnf, [auxm2], [conj])
 P~3 excl (svnf, [auxm3], [prep])
 P~4 excl (svnf, [auxmp], [serf, serp])
 P~5 excl (svnf, [auxtcf], [auxtci])
 P~6 excl (svnf, [auxm], [fvf])
 P~7 excl (svnf, [encl], [procl])
 P~8 excl (svnf, [ser], [clac])
 P~9 excl (svnf, [fvvintr], [ser, cl])
 P~10 excl (svnf, [dat], [clnse, ac])
 P~11 excl (svnf, [N ser, N aux], [v])
 P~12 excl (svnf, [N aux], [ser])

donde el auxiliar modal 1 (puede) excluye conjunción y preposición, y así sucesivamente.

5.4. Propiedades de Precedencia

Por último se definen las propiedades de precedencia, que determinan el orden de los constituyentes de la cadena.

P≈1 preced (svnf, [f], [p, i])
 P≈2 preced (svnf, [neg], [procl, fv])
 P≈3 preced (svnf, [ser], [v])
 P≈4 preced (svnf, [auxm], [prep, conj])
 P≈5 preced (svnf, [prep, conj], [i])
 P≈6 preced (svnf, [auxmp], [seri, auxtci])
 P≈7 preced (svnf, [auxtc], [auxm2])
 P≈8 preced (svnf, [auxm], [v])
 P≈9 preced (svnf, [procl], [fv])
 P≈10 preced (svnf, [i], [encl])
 P≈11 preced (svnf, [clse], [nrefl, clnse])
 P≈12 preced (svnf, [clnse], [nrefl])
 P≈13 preced (svnf, [auxtci], [p])
 P≈14 preced (svnf, [fvauxtci], [p, auxm2])

donde la forma flexionada precede al participio y al infinitivo, etc.

6. ENTRADAS Y EVALUACIÓN DE LA EFICACIA DEL MODELO PROPUESTO

El autómata así definido podrá operar en una doble dirección: por una parte, la herramienta *xfst* dispone de comandos que permiten generar cadenas categoriales o léxicas; por otra parte, dada una cadena léxica puede analizarla en términos categoriales.

En lo que respecta a la **generación**, la función de los comandos *print words* y *print lower words* es producir, respectivamente, todas las combinaciones posibles de **categorías** o **entradas** permitidas por las propiedades declaradas.

En los recuadros siguientes se presenta parte de los archivos de salida obtenidos con cada uno de estos comandos:

Tabla 3: Comando *print words*. Salida. [2]

```
N< fvserf: es>
N< fvvintrf: tose>
N< fvvtrf: escribe>
< proclnrefldat: le>< fvautctf: ha> N< fvvintrp: toside>
< proclnrefldat: le>< fvautctf: ha> N< fvvtrp: escrito>
< proclnrefldat: le> N< fvvintrf: tose>
< proclnrefldat: le> N< fvvtrf: escribe>
< proclnrefldat: les>< fvauxm2f: empieza>< prep: a> N< fvvintrf: toser>
< proclnrefldat: les>< fvauxm2f: empieza>< prep: a> N< fvvtrf: escribir>
< proclnrefldat: les>< fvauxm3f: tiene>< conj: que> N< fvvintrf: toser>
< proclnrefldat: les>< fvauxm3f: tiene>< conj: que> N< fvvtrf: escribir>
```

Tabla 4: Comando *print lower words*. Salida.

```
N es
N tose
N escribe
le ha N toside
le ha N escrito
le N tose
le N escribe
les empieza a N toser
les empieza a N escribir
les tiene que N toser
les tiene que N escribir
```

Estas funciones son de gran utilidad para la evaluación de la eficacia del modelo propuesto: tanto si genera cadenas agramaticales, como si no genera determinadas secuencias de caracteres bien formadas, esto deberá resolverse reajustando las propiedades.

El comando *apply down*, permite obtener, a partir de la introducción de fórmulas categoriales, cadenas de caracteres que corresponden a expresiones lingüísticas del español. Veamos algunos ejemplos:

Tabla 5: Comando *apply down*. Salida.

```
apply down> proclnreflac fvautctf N fvvtrp
lo ha N escrito
la ha N escrito
los ha N escrito
las ha N escrito
apply down>

apply down> proclnreflac fvautctf fvauxm2p prep N fvvtrf
lo ha empezado a N escribir
la ha empezado a N escribir
los ha empezado a N escribir
las ha empezado a N escribir
apply down>
```

Si, por el contrario, lo que se desea es analizar secuencias lingüísticas, es posible recurrir al comando *apply up*, que permite reconocer fórmulas bien formadas e interpretarlas en términos categoriales. A continuación damos algunos ejemplos de las salidas obtenidas de este modo:

Tabla 6: Comando *apply up*. Salida.

```

Comando apply up
xfst[2]: apply up
apply up>

apply up> no ha empezado a N toser
neg fvautxcf fvauxm2p prep N fvintri
apply up>

apply up> puede haberse N escrito
fvauxm1f fvautxcienclreflse N fvvtrp
apply up>

apply up> puede haber sido N escrito
fvauxm1f fvautxci fvserp N fvvtrp
apply up>

apply up> ha empezado a N toserle
fvautxcf fvauxm2p prep N fvintrienclnrefldat
apply up>
    
```

Este comando permite corroborar la confiabilidad del autómata, si se comprueban cadenas no gramaticales de la lengua, que no deben ser reconocidas ni analizadas. En los siguientes ejemplos, la salida del autómata (*apply up*>) permanece vacía, lo que indica que la cadena no ha sido reconocida como válida para la lengua analizada:

Tabla 7: Testeo de confiabilidad.

```

apply up> le la ha sido N escrito
apply up>

apply up> ha empezado a N toserlo
apply up>

apply up> N ha empezado a toser
apply up>
    
```

En el último caso, si bien la cadena ‘ha empezado a toser’ es gramatical, el núcleo se ha asignado de manera errónea al auxiliar de tiempo compuesto, y por ello no es reconocida como válida por el autómata.

7. CONCLUSIONES Y PROYECCIONES

A lo largo del presente trabajo se pudo obtener un autómata adecuado para el análisis y la generación de fvnf con perífrasis verbales del tipo definido en esta propuesta.

En futuros trabajos, se espera desarrollar el campo, incluyendo fvnf más complejas.

Se espera integrar este trabajo con el abordaje de frases de otro tipo (sustantivas, adjetivas, adverbiales y preposicionales), a fin de arribar al análisis de secuencias oracionales completas.

Lograr este objetivo, a largo plazo, implica reformular las conceptualizaciones gramaticales tradicionales desde un punto de vista formal, teniendo como meta la definición de categorías a partir de la interrelación de sus propiedades morfológicas y de sus posibilidades combinatorias.

Una gramática de este tipo puede resultar de gran utilidad en la enseñanza de las lenguas, en tanto que permite que la intuición de aquellos que utilizan el español como primera o segunda lengua pueda confrontarse con las formas analizadas por el autómeta.

NOTAS

[1] *xfst* es un software desarrollado por Xerox y usado por Xerox Research Centre Europe (XRCE, Grenoble, Francia) y Palo Alto Research Center (PARC, California, USA) y otros centros de investigación lingüística en el mundo.

[2] La N insertada en la cadena indica la atribución de la categoría Núcleo al verbo correspondiente.

BIBLIOGRAFÍA

BÈS, Gabriel (1999) “La phrase verbale noyau en français”, en *Recherches sur le Français parlé*, N° 15.

GÓMEZ TORREGO, Leonardo (1999) “Los verbos auxiliares. La perífrasis verbales de infinitivo”, en BOSQUE, Ignacio y DEMONTE, Violeta, *Gramática descriptiva de la lengua española*. Espasa Calpe, S.A., Madrid.

SOLANA, Zulema (2006), *Lingüística Informática y enseñanza de lenguas*, Rosario, Ediciones Juglaría.