

NooJ









Tutorial en español

Rodolfo Bonino
Andrea Rodrigo

Objetivos

Objetivo general: se espera que al finalizar el curso los alumnos sean capaces de utilizar NooJ para efectuar un análisis automático elemental de un texto o de un aspecto de la gramática que sea de su interés.

Objetivos específicos:

-  Conocer la interfaz del programa
-  Cargar textos
-  Interpretar la información que proporciona NooJ antes y después de aplicar diccionarios
-  Declarar propiedades
-  Elaborar diccionarios
-  Elaborar gramáticas morfológicas flexionales y derivacionales
-  Elaborar gramáticas morfológicas productivas
-  Elaborar gramáticas sintácticas

Clase 1

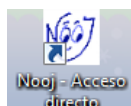
En esta clase aprenderemos a:

- Conocer la interfaz de NooJ
- Cargar un texto
- Interpretar la información que brinda en análisis automático de NooJ antes y después de aplicar diccionarios
- Definir propiedades
- Elaborar un diccionario sencillo

1. El aplicativo NooJ

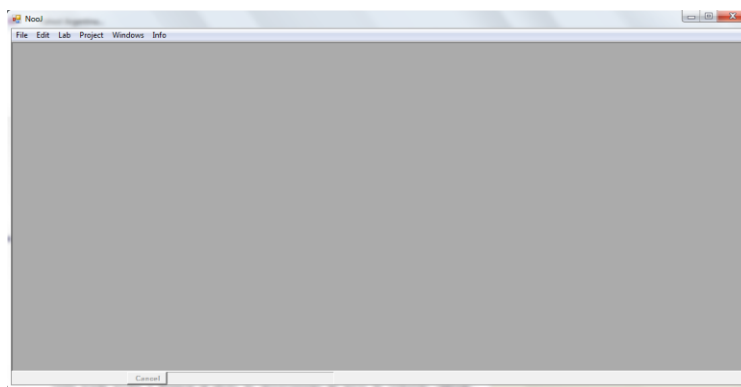
1.1. Abrir NooJ

Para comenzar, debemos abrir el programa ejecutable *Nooj.exe*, que se encuentra dentro de la carpeta *_App* y está representado por el ícono:



Si pensamos trabajar habitualmente con el programa, conviene crear un acceso directo en el escritorio.

Haciendo doble clic el ícono, se visualiza la siguiente pantalla:



1.2. Seleccionar preferencias

Haciendo clic en *Info > Preferences* aparece una nueva ventana con tres solapas *General*, *Lexical Analysis* y *Syntactic Analysis*.

En *General*, inmediatamente debajo de las solapa, tenemos una pequeña ventana desplegable que nos permite seleccionar el idioma con el que vamos a trabajar, en nuestro caso *sp* (español) y hacemos clic en *apply*.

La versión de NooJ que bajamos de Internet puede tener seleccionados por defecto algunos diccionarios o gramáticas, que presenta como modelos. Con la finalidad de empezar el trabajo desde cero, en la pestaña *Lexical Analysis*, hacemos clic sobre los dos botones *Uncheck All*, que se encuentran debajo de cada uno de los paneles en los que se divide la parte superior de la ventana.

1.3. Cargar texto

En primer lugar, cargaremos un pequeño texto, que nos permitirá verificar en forma manual el análisis automático producido por NooJ. Para ello, hacemos clic en *File > New > Text*, verificamos que en la nueva ventana se encuentre seleccionada la opción *sp*, no modificamos los demás valores y hacemos clic sobre *OK* (en la parte inferior derecha de la ventana).

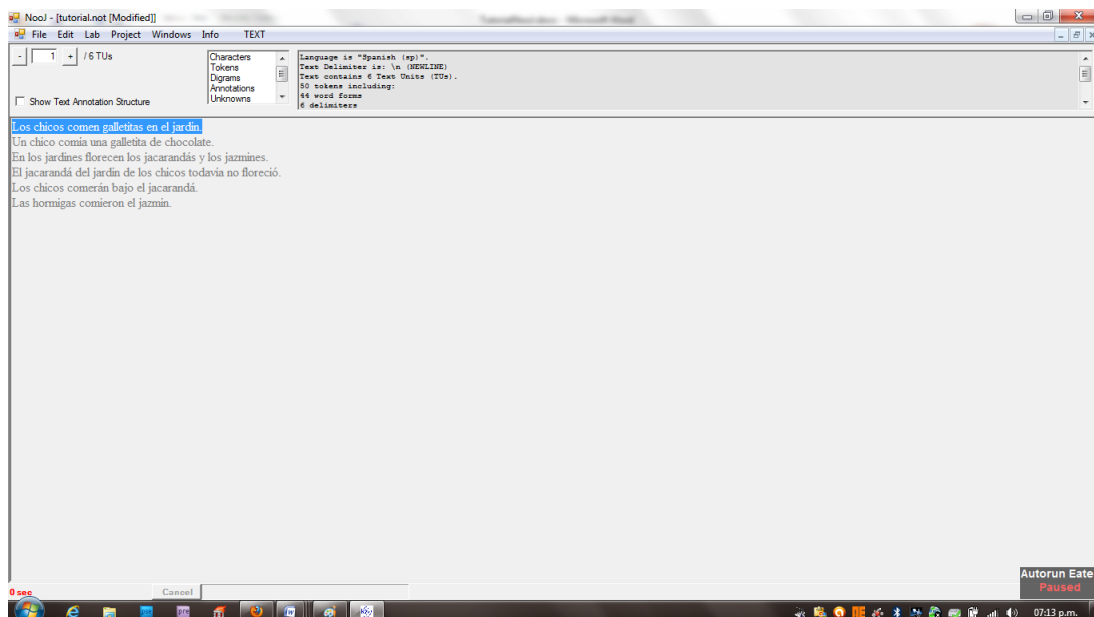
En la siguiente ventana podemos escribir el texto que vamos a utilizar:

*Los chicos comen galletitas en el jardín.
Un chico comía una galletita de chocolate.
En los jardines florecen los jacarandás y los jazmines.
El jacarandá del jardín de los chicos todavía no floreció.
Los chicos comerán bajo el jacarandá.
Las hormigas comieron el jazmín.*

Luego hacemos clic en *File > Save*, le asignamos el nombre *tutorial* (el programa agrega automáticamente la extensión *.not*) y lo guardamos tocando el botón *Guardar*.

1.4. Analizar un texto sin aplicar diccionarios ni gramáticas

Una vez que hayamos guardado el texto, veremos que en la barra de menús aparece el comando *TEXT*; si hacemos clic en *TEXT > Linguistic Analysis*, obtenemos la siguiente ventana:



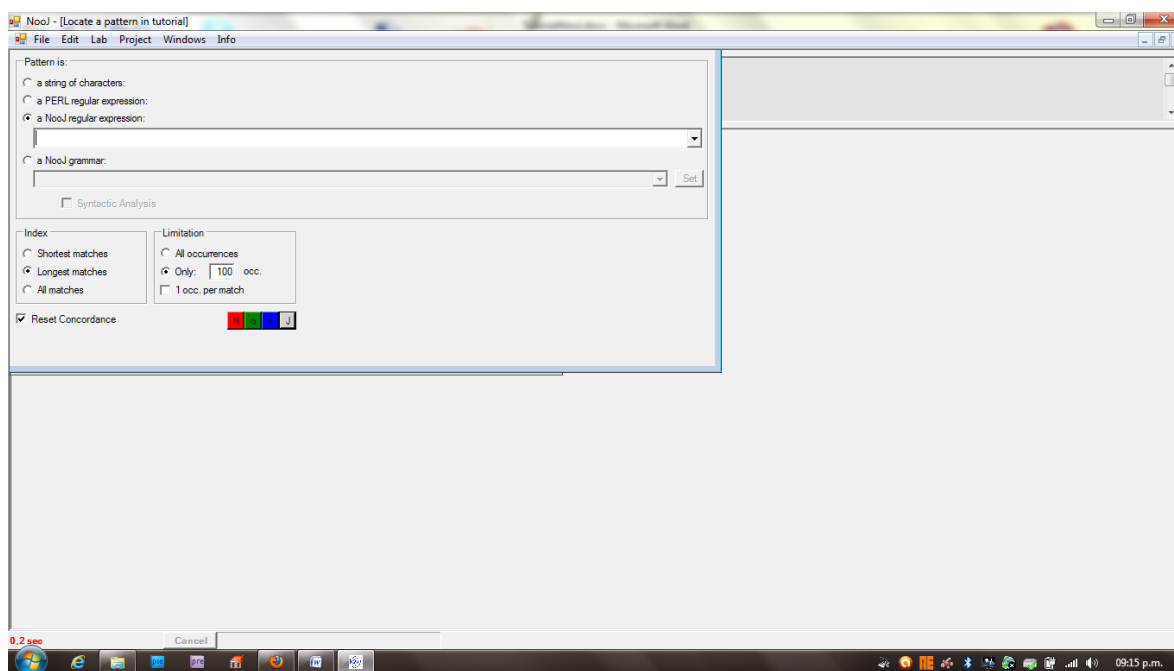
A continuación, podemos observar el análisis automático que produce NooJ:

Inmediatamente debajo de la barra de menús y por encima del texto, se puede ver – de izquierda a derecha– la siguiente información:

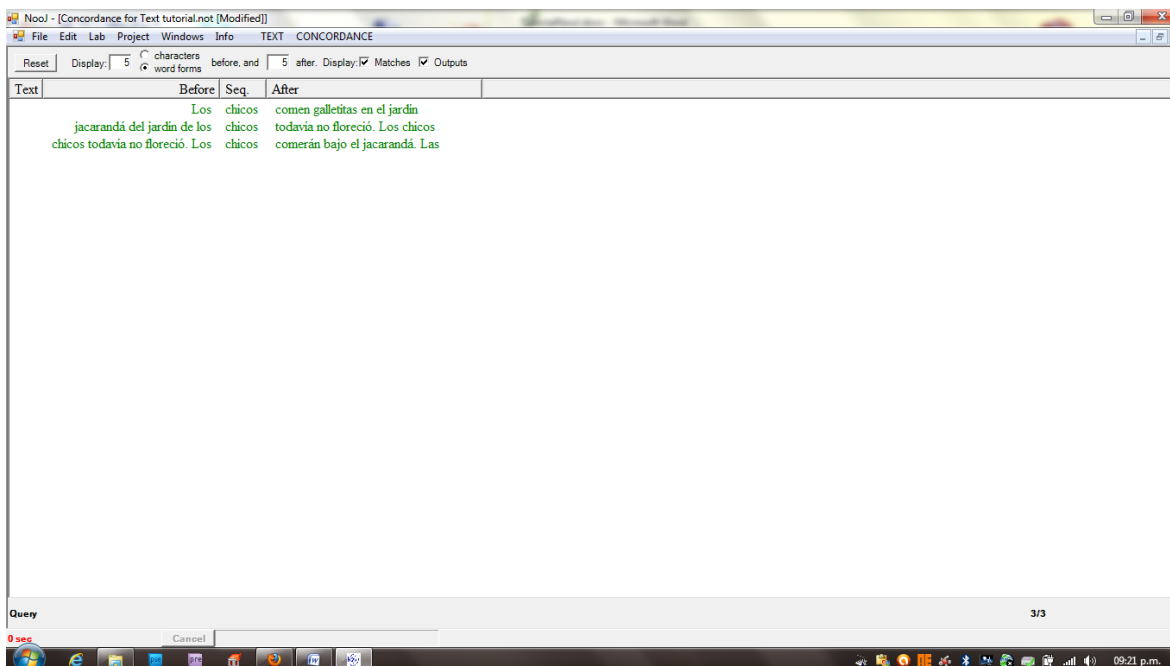
1. Un recuadro con un número (1, en nuestro caso) precedido por el signo – y seguido por *+/6TUs*:
 - a) *6TUs* indica el número de unidades textuales del archivo; en este caso se refiere a las seis párrafos que escribimos (en nuestro caso, los párrafos coinciden con oraciones, los párrafos con más de una oración cuentan como una sola unidad textual), si modificáramos el texto agregando o quitando oraciones, ese número se modificaría.
 - b) El número indica cuál es la unidad textual que se está analizando en un momento determinado. Como el texto no tiene anotaciones, por ahora, esta información es irrelevante.
2. Debajo de ese recuadro, se lee *Show Text Annotation Structure* precedido de un espacio para seleccionar esa opción. Cuando agreguemos diccionarios y gramáticas, nos permitirá ver el resultado de su aplicación en el texto.
3. En la parte central vemos una pequeña ventana con una barra de desplazamiento. Cada una de las líneas de esa ventana, que se pueden seleccionar haciendo doble clic sobre ellas, da alguna información sobre el texto:
 - a) *Characters*: indica la frecuencia con la que aparece cada carácter en el texto, el tipo de carácter –los caracteres pueden ser letras, espacios en blanco o delimitadores (signos de puntuación)– y el código numérico que le corresponde a cada uno de ellos en el sistema Unicode. Haciendo clic en la barra superior se pueden seleccionar distintos criterios para ordenarlos; por ejemplo, si hacemos clic en *Freq* los ordenará de mayor a menor por la frecuencia con que aparecen en

- el texto, en nuestro caso el carácter que aparece con mayor frecuencia es el espacio en blanco, que aparece cuarenta y dos veces.
- b) *Tokens*: indica la frecuencia con la que aparecen conjuntos de caracteres delimitados por espacios en blanco. Como se puede observar, las variaciones morfológicas de cada palabra constituyen distintos *tokens*.
 - c) *Digrams*: marca las agrupaciones de dos *tokens* que se repiten en el texto. En nuestro caso, la única secuencia que se repite dos veces es *los chicos*.
 - d) *Annotations*: abre automáticamente un diccionario en blanco para que se incorporen anotaciones al texto.
 - e) *Unknowns*: marca las palabras que no aparecen en ningún diccionario. En nuestro caso, las selecciona a todas porque no hay diccionarios.
 - f) *Ambiguities*: cuando hay diccionarios, indica cuáles son las palabras que tienen más de una entrada en los diccionarios. Por ejemplo, *canto* puede estar etiquetado como sustantivo y como verbo.
 - g) *Unambiguous Words*: marca las palabras que reciben una sola etiqueta en los diccionarios.

Haciendo clic en *TEXT > Locate* aparece la siguiente pantalla:



Si escribimos cualquier *token* o secuencia de *tokens* del texto y tocamos cualquiera de los botones que se encuentran en la parte inferior de la pantalla, NooJ nos muestra todas las veces que aparece en el texto; por ejemplo, si buscamos la palabra *chicos*, obtenemos el siguiente resultado.



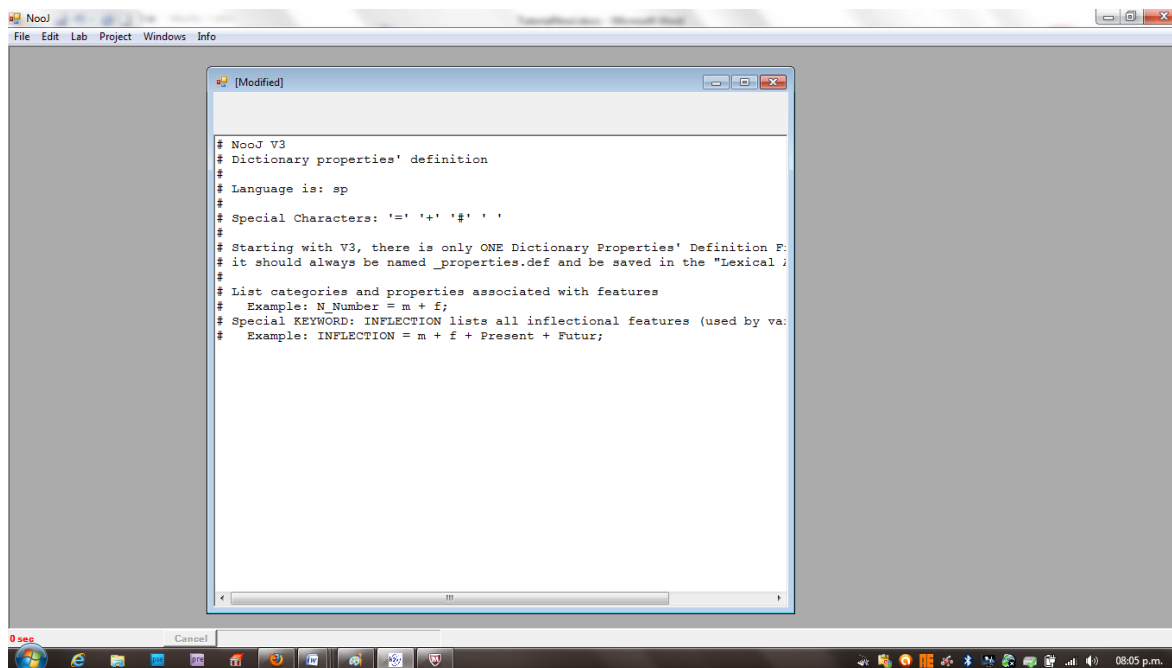
El color de los botones determina el color de los caracteres de salida (la salida que se muestra arriba se obtuvo tocando el botón verde), en la parte superior de la pantalla se pueden seleccionar la cantidad de palabras o caracteres del contexto anterior o posterior que queremos ver. En la parte inferior derecha 3/3 indica que en la pantalla se muestran tres coincidencias con la búsqueda, que es el total de las coincidencias.

1.5. Definir propiedades

Las propiedades son los rasgos que se asignan a las entradas del diccionario. Los más comunes son los rasgos categoriales (N, V, Adj, Adv, Prep, etc.) y los morfológicos (persona, número, género, tiempo, etc.).

El rasgo principal de una entrada es el categorial (V, N, ADV, ADJ, etc.) al que se suman todos los demás, que tienen distintos valores, por ejemplo, al rasgo categorial V (verbos) se le pueden sumar los rasgos *persona* con los valores *1era*, *2da* y *3era*; *número* con los valores *sg* y *pl*, y cualquier otro rasgo que sea relevante para el tema que se está estudiando. Los rasgos se pueden ver como columnas del diccionario. Todos los diccionarios tienen columnas predeterminadas para el rasgo categorial y para el modelo de flexión (que se verá en la próxima clase), el usuario puede definir los demás; para hacerlo debe declarar los rasgos que quiere utilizar y asociarlos a las distintas categorías, el título de la columna es el nombre del rasgo y en cada línea se muestra el valor de rasgo que corresponde a cada entrada.

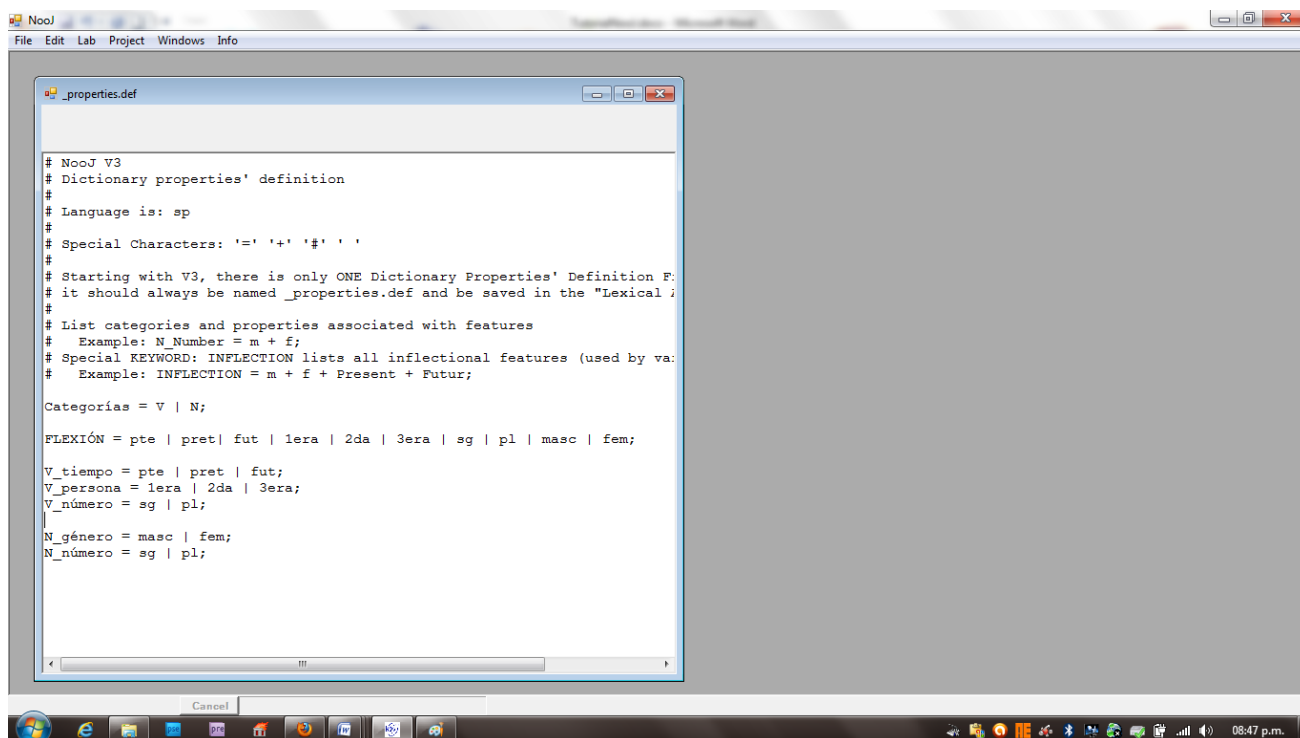
Todos los diccionarios toman los rasgos declarados en un archivo que lleva el nombre de *_properties.def* y se guarda en la carpeta *Lexical Analysis* del idioma seleccionado. Para crearlo haremos clic en *File > New > Properties definition*, seleccionaremos *sp* y veremos la siguiente pantalla:



En nuestro diccionario solo incluiremos verbos (V) y sustantivos (N), de modo que a continuación de la última línea de la pantalla predeterminada escribimos:

Categorías = V | N;

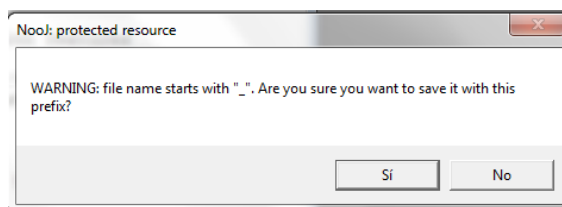
El archivo de propiedades que utilizará nuestro primer diccionario será:



Las etiquetas no debe ser necesariamente la que utilizamos en este ejemplo; en lugar de V, podríamos utilizar VB y en lugar de N, NM. Solo debemos tener cuidado de utilizar siempre la misma etiqueta con el mismo valor.

El carácter “|” es un operador del programa que indica disyunción; dicho en lenguaje no formal, la sentencia “*Categorías = V | N;*” significa “*las categorías que utilizamos son V o N*”.

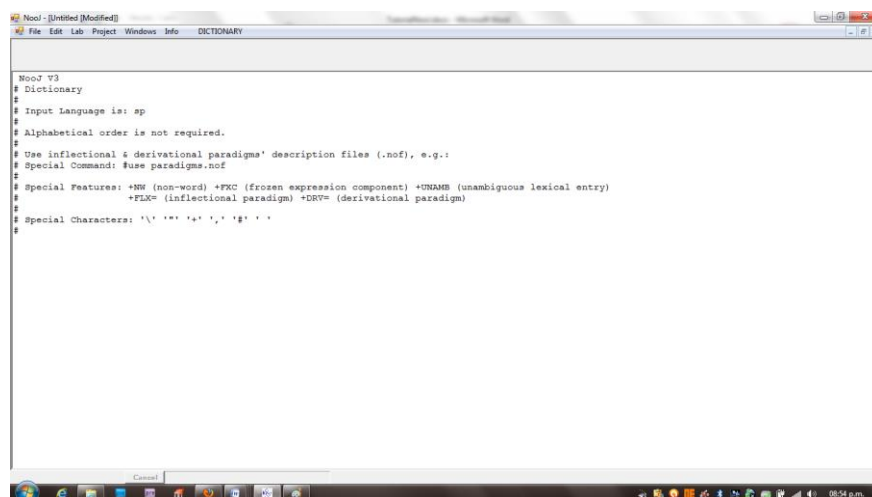
En la línea siguiente declaramos los rasgos morfológicos que se van utilizar en el diccionario y, a continuación, asociamos las categorías con los rasgos flexivos que les corresponden. Al finalizar hacemos clic en *File > Save as*, le damos el nombre *_properties.def* y lo guardamos (NooJ suele tener un archivos con el nombre *_properties.def*, por lo tanto nos pedirá indicación de reemplazarlo; sería conveniente haber guardado previamente el archivo original en otra carpeta, pero, si no se efectuó esta operación, el archivo reemplazado se puede recupera bajando nuevamente el programa). Después de esto se verá un nuevo mensaje:



Esta advertencia aparece porque todos los archivos cuyo nombre se inicia con “_” desaparecen cada vez que descargamos una nueva versión del programa, por ello conviene archivar una copia de la última versión fuera de NooJ.

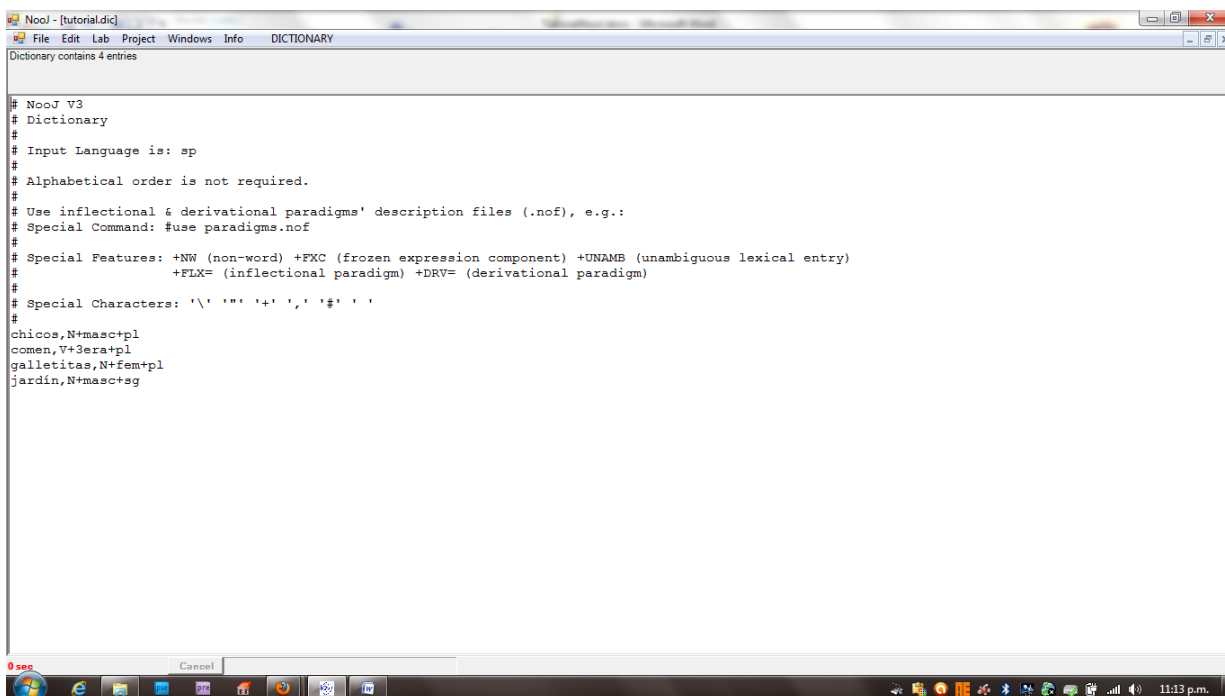
1.6. Crear un diccionario

Para crear un diccionario hacemos clic en *File > New > Dictionary* y seleccionamos *sp* y vemos la siguiente pantalla:

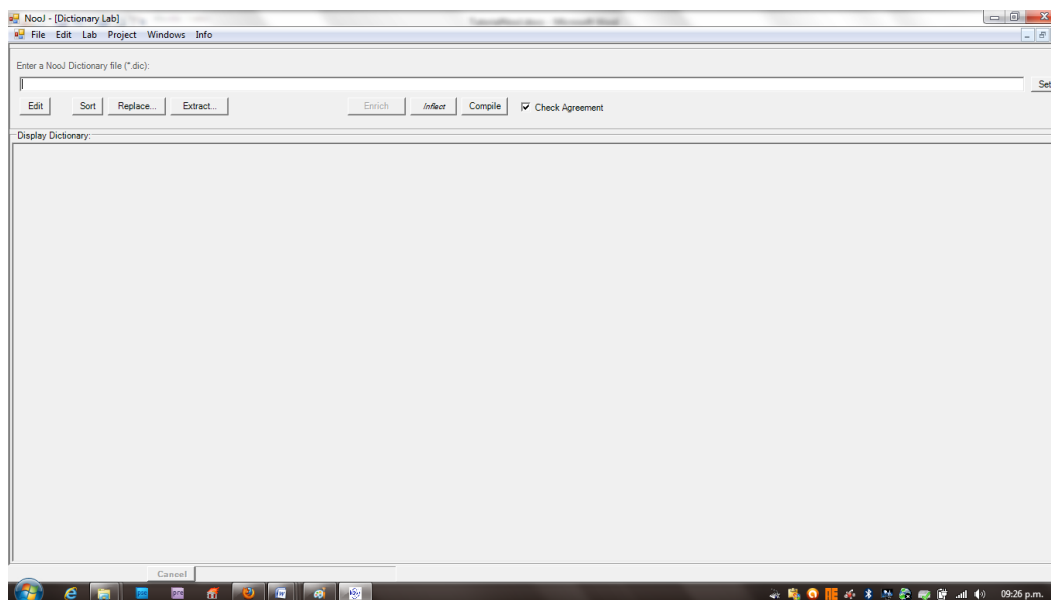


Como nuestro diccionario no utiliza ninguna gramática morfológica (tema que veremos en la próxima clase), simplemente escribimos las entradas “,” la etiqueta categorial “+” los rasgos

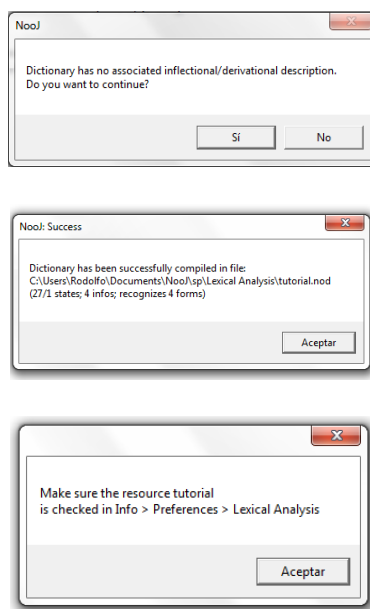
morfológicos. Las entradas se escriben en minúscula, una debajo de la otra; no debe haber espacios en blanco después de la coma ni después del último rasgo. Para comenzar, ingresamos los sustantivos y los verbos de la primera oración; luego hacemos clic en *DICTIONARY > check format* (si nuestro diccionario está correctamente declarado, obtenemos la respuesta *No problem found*) y, para finalizar, *File > Save as* (le damos el nombre *tutorial* al que NooJ agrega automáticamente la extensión *.dic*) > *Guardar*. El diccionario resultante es el siguiente:



Para que el diccionario pueda ser aplicado al análisis del texto, previamente debemos compilarlo; haciendo clic en *Lab > Dictionary* aparece la siguiente pantalla:



Haciendo clic en *Set*, podemos seleccionar los diccionarios disponibles para compilar, hacemos clic sobre el nombre del diccionario y luego en el botón *abrir*; de este modo, el nombre del diccionario queda en el espacio en blanco que precede a *Set*. Luego hacemos clic en *Compile*, aceptamos la advertencia de que el diccionario no está asociado con ninguna gramática flexional y las dos advertencias siguientes:



1.7. Aplicar el diccionario al análisis del texto

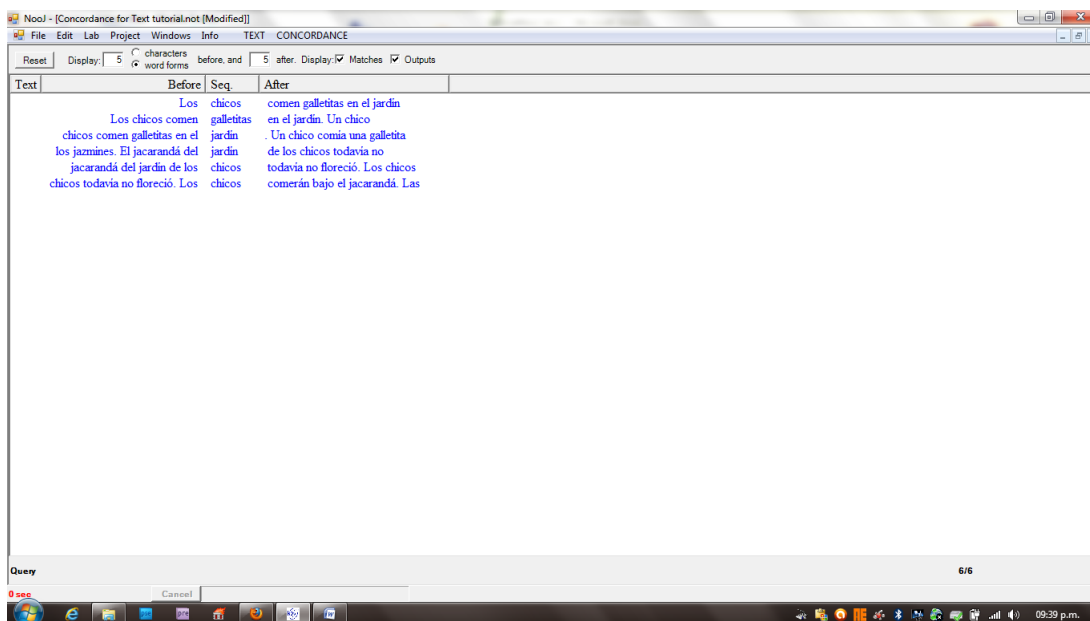
Para aplicar el diccionario al análisis del texto, debemos abrirlo haciendo clic en *File > Open > Text* (el texto se puede modificar haciendo clic en *TEXT > Modify Text*). Luego hacemos clic en *Info > Preferences*; en la pestaña *General*, verificamos que esté seleccionado el idioma *sp*; en la pestaña *Lexical Analysis*, tocamos el botón *Refresh* para que aparezca el diccionario que hemos compilado y tildamos el casillero que se encuentra a su izquierda; finalmente, verificamos que en la ventana de la derecha de la pestaña *Syntactic Analysis* no haya ningún archivo seleccionado; si hubiera alguno hacemos clic sobre el nombre y lo eliminamos con **X** para evitar que aplique gramáticas que no hemos hecho nosotros. Una vez que hayamos seleccionado todos los archivos que queremos aplicar al análisis, hacemos clic en *Apply*.

Para efectuar el análisis, hacemos clic en *TEXT > Linguistic Analysis* y verificamos los resultados.

- Tildando el casillero que está a la izquierda de *Show Text Annotation Structure*, vemos que en la parte inferior de la ventana aparece un análisis de la primera oración, que muestra los sustantivos y verbos que incluimos en el diccionario con los rasgos categoriales y morfológicos que les asignamos.
- Cada vez que hacemos clic en “+” pasamos a la siguiente oración y vemos el análisis de cada una de ellas. Como se puede ver, solo se muestran salidas de análisis en las oraciones 1, 4 y 5, que son las únicas donde se repiten algunos *tokens* de la primera oración. Nótese que no se analizan las variantes flexionales porque, si bien se trata de la misma palabra, son distintos *tokens*.

- c) En la ventana central de la parte superior, se puede obtener información que no estaba disponible antes de aplicar el diccionario: haciendo doble clic en *Annotations* se abre automáticamente el diccionario y en *Unknowns* se ven las palabras que no están incluidas en el diccionario.

Ahora, con la opción *TEXT > Locate*, además de *tokens* o secuencias de *tokens*, podemos buscar categorías solas o con determinados rasgos (siempre que hayan sido incluidos en el diccionario), en este caso el objeto de la búsqueda debe aparecer entre <>; por ejemplo, si queremos buscar sustantivos, pondremos <N>; si queremos buscar sustantivos en singular, pondremos <N+sg>; si queremos buscar sustantivos masculinos en singular, pondremos <N+masc+sg>. No es posible buscar rasgos que no estén asociados a alguna categoría; por lo tanto, si ponemos <+masc+sg> el resultado de la búsqueda es *Concordance is empty*. En la siguiente pantalla se muestra el resultado de la búsqueda <N> en el contexto de las cinco palabras anteriores y las cinco posteriores:



Ejercitación

- 1) Modifique el texto y agregue las oraciones: *El camino del jardín es de grava* y *Camino por el jardín*.
- 2) Agregue al diccionario los sustantivos y los verbos de estas dos oraciones (verifique que no estén ya incluidos y recuerde guardar los cambios)
- 3) Compile el diccionario.
- 4) Analice el texto modificado.
- 5) Indique la frecuencia de caracteres *a, d, c, “.”* y espacios en blanco.
- 6) ¿Cuáles son los *tokens* que aparecen con mayor frecuencia?
- 7) ¿Cuáles son los *digrams* que hay en el texto y con qué frecuencia aparecen?
- 8) ¿Qué función utilizaría para saber qué palabras del texto no están incluidas en el diccionario?
- 9) ¿Qué ambigüedades encuentra NooJ y cómo las analiza?
- 10) ¿Como efectuaría la búsqueda de todos los verbos del texto?, ¿qué verbos obtiene con la aplicación del diccionario modificado?

Clase 2

En esta clase aprenderemos a:

- ✚ Crear gramáticas regulares y gráficas, morfológicas y derivacionales
- ✚ Conocer los operadores que se pueden utilizar
- ✚ Elaborar diccionarios que utilicen estas gramáticas
- ✚ Evaluar los diccionarios creados y los resultados de distintas consultas

1. Gramáticas morfológicas

Los diccionarios no electrónicos del español tienen como entrada los infinitivos de los verbos y la forma masculina singular de los adjetivos y sustantivos. Antes de efectuar una búsqueda, el usuario aplica intuitivamente su conocimiento gramatical para asociar las variantes morfológicas de una palabra con la forma que proporciona el diccionario. Esta operación es elemental para la inteligencia humana, pero no para la inteligencia artificial. Por lo tanto, para lograr el análisis automático de un texto, es necesario lograr que el sistema informático reconozca todas las variaciones que puede tener una palabra.

Si pretendiéramos analizar automáticamente un texto con diccionarios diseñados con el criterio que aplicamos en la clase anterior, tendríamos que incluir como entrada cada una de las variantes de cada palabra. Sin embargo, dado que las variaciones morfológicas tienen patrones que presentan cierta regularidad, resulta mucho más sencillo y económico crear gramáticas que contengan modelos que se puedan asociar con distintas palabras, así, por ejemplo el modelo del verbo *cantar*, se puede aplicar a todos los verbos regulares de la primera conjugación y el modelo de *perro* se puede aplicar a todos los sustantivos y adjetivos que tengan masculino terminado en *o*, femenino en *a* y plural en *s*. Estas gramáticas permiten que un autómata efectúe una operación similar a la que aplica un usuario humano al consultar un diccionario tradicional.

1.1. Editor de gramáticas regulares

NooJ ofrece una aplicación para crear gramáticas. En primer lugar crearemos una gramática flexiva regular. Si hacemos clic en *File > New > Grammar*, aparece una ventana dividida en dos paneles superiores –(1) *Select Languages* y (2) *Optional Information*– y uno inferior – *Select Grammar Type*–. En el superior izquierdo (1) seleccionamos el idioma de entrada y de salida; en nuestro caso ambos son *sp* (español); el panel superior derecho permite consignar el autor y la institución que desarrolla la gramática, seleccionar opciones del tamaño del editor gráfico (*Large graph*), y bloquear acceso de la gramática mediante claves (estas funciones son opcionales y no las utilizaremos); en el cuadro inferior seleccionamos el tipo de gramática que vamos a desarrollar. Como indica la pantalla, las gramáticas morfológicas (*Inflection & Derivation* y *Productive Morphology*) reconocen secuencias de letras y las gramáticas sintácticas (*Syntax*) secuencias de palabras.

Dado que lo que queremos crear es una gramática flexiva regular, haremos clic en *rule editor* y luego en *Inflection & Derivation* y aparecerá la siguiente pantalla:

```
# NooJ V3
#
# Inflectional/Derivational description
#
# Language is: sp
#
# Special Characters: '=' '<' '>' '\ ' '!' ':' '+' '/' '#' ';'
#
# Generic Commands:
# <B>: keyboard Backspace
# <C>: change Case
# <D>: Duplicate current char
# <E>: Empty string
# <L>: keyboard Left arrow
# <N>: go to end of Next word form
# <P>: go to end of Previous word form
# <R>: keyboard Right arrow
# <S>: delete/suppress current char
# Arguments for commands <B>, <L>, <N>, <P>, <R>, <S>:
# xx number: repeat xx times
# W: whole word
# Examples
# <R3>: go right 3 times
# <LW>: go to beg. of word
#
# Language-Specific Commands:
# <A>: remove acute accent from last letter aeiou
# <A>: add acute accent to last letter aeiou
# <A>: add grave accent to last letter aeiou
# <A>: add circumflexe to last letter aeiou
# <A>: add dieresis/trema/umlaut to last letter aeiou
# <U>: replace last 'e' with 'ie' or last 'o' with 'ue'
# <V>: replace last 'e' with 'i', 'o' with 'u', 'z' with 'c', 'u' with 'u', 'g' with 'j', 'c' with 'z'
```

En esta pantalla se ven los comandos que se pueden utilizar; para mayor claridad los repetimos traducidos con una breve explicación:

 Borra caracteres de izquierda a derecha. Por ejemplo; PERRO = a/fem+sg, indica que debe borrar la "o" y agregar la "a" con la etiqueta femenino singular.

<C> Cambia minúsculas por mayúsculas y viceversa.

<D> Duplica un carácter.

<E> Agrega un elemento vacío. Se utiliza cuando queremos poner una etiqueta a un elemento que no está representado por ningún carácter. Por ejemplo, CANTAR = <E>/inf, indica que a la cadena se le debe agregar la etiqueta sin agregar ni quitar ningún carácter. En cambio, CANTAR = é/fut+1era+sg, indica que debe agregar el carácter "é" y agregar la etiqueta mencionada.

<L> Mueve el cursor a la izquierda.

<N> Mueve el cursor al final de la siguiente palabra.

<P> Mueve el cursor al final de la palabra anterior.

<R> Mueve el cursor a la derecha.

Los comandos se escriben en mayúscula y entre <> y las etiquetas se escriben precedidas de /; si se omiten estos operadores, NooJ lo interpreta como un carácter y se limita a agregarlos a la cadena anterior; por ejemplo, si escribimos PERRO = Ba fem+sg, obtendremos la cadena "perrBa fem+sg" como si fuera una palabra.

Si se agregan números arábigos después del comando, la operación se repite la cantidad de veces que indica el número. Por ejemplo, si ponemos <B3>, borra tres caracteres de izquierda a derecha; si ponemos <L2> mueve el cursor dos caracteres a la izquierda. Es posible combinar varios

comandos; por ejemplo, LUCIR = <B2><L>z<R>o/pte+1era+sg, produce "luzco" con la etiqueta correspondiente.

El comando <W> indica la palabra completa; por ejemplo <LW> mueve el cursor hacia la izquierda, al principio de la palabra.

<A> Quita la tilde a la primera vocal acentuada que encuentra de izquierda a derecha.

<Á> Agrega tilde aguda a la primera vocal que encuentra de izquierda a derecha.

<À> Agrega tilde grave a la primera vocal que encuentra de izquierda a derecha.

<Â> Agrega tilde circunfleja a la primera vocal que encuentra de izquierda a derecha.

<U> Reemplaza las vocales "e" y "o" por los diptongos "ie" y "ue", respectivamente.

<V> Efectúa varios reemplazos, que se indican en la pantalla.

A continuación de esta pantalla comenzaremos a desarrollar una pequeña gramática que permita reconocer las variaciones morfológicas de las palabras que aparecen en el texto que analizamos en la clase anterior.

En primer lugar, se indica el nombre del modelo, que es la referencia que se utiliza en el diccionario para aplicar las operaciones que se establecen en la gramática. La única restricción para el nombre del modelo es que debe estar escrito en mayúscula; para facilitar su identificación tomamos una palabra; por ejemplo la palabra "CHICO" puede servir de nombre para el modelo de sustantivos que en femenino reemplazan el último carácter por "a" y agregan "s" al plural, y "COMER" como nombre del modelo de los verbos regulares de la segunda conjugación.

En el texto "chicos" aparece solo en masculino singular y plural; si además de estas variaciones, quisiéramos agregar los femeninos tendríamos que hacer alguna de estas reglas:

CHICO = <E>/masc+sg | s/masc+pl | a/fem+sg | as/fem+pl;

CHICO = (<E>/masc | a/fem) s/pl;

Ambas reglas producen el mismo resultado. La diferencia entre ellas es que la primera es más clara y sencilla para el usuario porque efectúa todas las operaciones a partir de la palabra completa, pero es menos económica desde el punto de vista del procesamiento porque agrega dos veces cada etiqueta; mientras la segunda, por el contrario, es más económica para el procesamiento, pero menos clara para el usuario. De cualquier modo, para enunciar una regla se utilizan los siguientes elementos:

1. La barra hacia la derecha "/" indica que lo que sigue es la etiqueta.
2. Cuando a una misma cadena de caracteres se agrega más de una etiqueta, a partir de la segunda, se utiliza el signo "+".
3. La barra recta "|" indica alternativa: en la primera regla decimos que agregue la cadena vacía "<E>" seguida de las etiquetas "masc" y "sing" o que agregue "s" seguida de las etiquetas masc y pl o etc.
4. El espacio en blanco indica concatenación: en la segunda regla decimos que agregue la cadena vacía seguida de la etiqueta "masc" o borre el último carácter y agregue "a" seguida de la etiqueta "fem" y que luego concatene cualquiera de ambos resultados con una "s" seguida de la etiqueta "pl"

5. Los paréntesis agrupan elementos: la ausencia de paréntesis en la segunda regla daría como resultado que "s/pl" se agregara únicamente a la operación "a/fem" y, consecuentemente, no obtendríamos el masculino plural.
6. Cada regla finaliza en punto y coma.

La siguiente gramática permite generar y analizar todas las variantes morfológicas de los sustantivos y adjetivos que aparecen el texto:

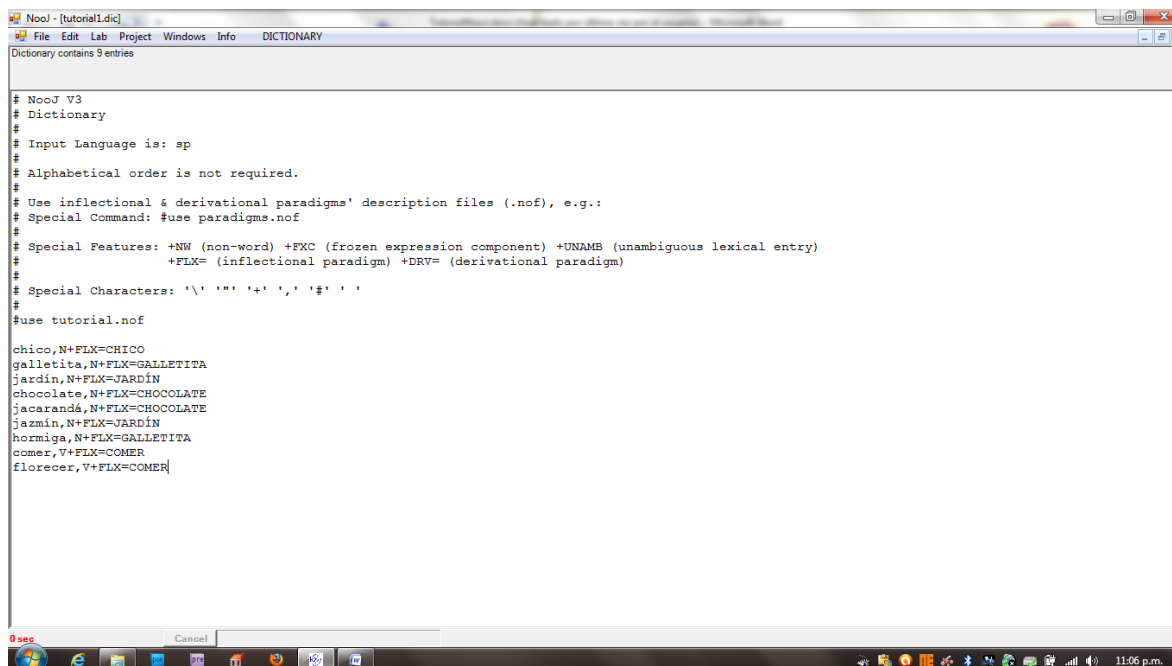
```
# <E>: Empty string
# <L>: keyboard Left arrow
# <N>: go to end of Next word form
# <P>: go to end of Previous word form
# <R>: keyboard Right arrow
# <S>: delete/Suppress current char
# Arguments for commands <B>, <L>, <N>, <P>, <R>, <S>:
# xx number: repeat xx times
# W: whole word
# Examples
# <R3>: go right 3 times
# <LW>: go to beg. of word
#
# Language-Specific Commands:
# <A>: remove accute accent from last letter aeiou
# <A>: add acute accent to last letter aeiou
# <A>: add grave accent to last letter aeiou
# <A>: add circumflexe to last letter aeiou
# <A>: add dieresis/trema/umlaut to last letter aeiou
# <U>: replace last 'e' with 'ie' or last 'o' with 'ue'
# <V>: replace last 'e' with 'i', 'o' with 'u', 'z' with 'c', 'u' with 'u', 'g' with 'j', 'c' with 'z'

CHICO = <E>/masc+sg | s/masc+pl | <B> a/fem+sg | <B> as/fem+pl;
GALLETA = <E>/fem+sg | s/fem+pl;
JARDIN = <E>/masc+sg | <A> es/masc+pl;
CHOCOLATE = <E>/masc+sg | s/masc+pl;
COMER = <E>/inf | <B2>en/pte+3era+pl | <B2>ia/pps+3era+sg | <B>rán/fut+3era+pl | <B2>ieron/pps+3era+pl;
LEGAL = <E>/inv+sg | es/inv+sg;
```

1.2. Aplicación de la gramática al diccionario

El siguiente paso es crear un diccionario asociados con la gramática. Para ello, hacemos clic en *File* > *New* > *Dictionary*, seleccionamos *sp* y hacemos clic en *ok*. En la primera línea del diccionario escribimos *#use tutorial. nof*. y, a continuación, las entradas del diccionario seguidas de coma, la etiqueta categorial "+FLX=...", que expresa que la palabra del diccionario sigue determinado modelo de flexión. En el texto, utilizamos solo las formas regulares del verbo *floreecer*, por lo tanto, podemos decir *floreecer, V+FLX=COMER*, si quisiéramos incluir la conjugación completa del verbo, tendríamos que crear un nuevo modelo porque, de lo contrario, se excluirían las formas irregulares de *floreecer*; por ejemplo, si se aplica el modelo de *comer*, para la tercera persona del presente del subjuntivo obtendríamos la forma **florecan*.

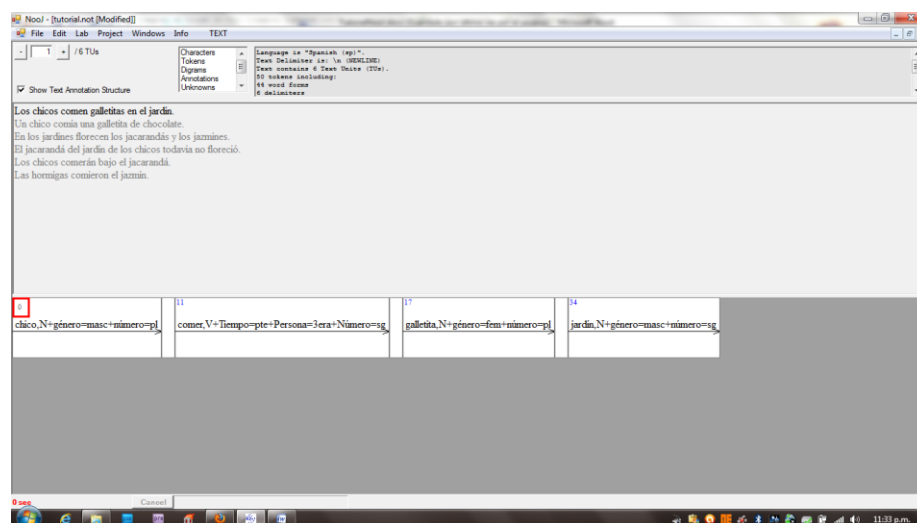
Para crear el nuevo diccionario hacemos clic en *file* > *New* > *Diccionario*, verificamos que esté seleccionado el idioma *sp* y pulsamos en botón *OK*. En la primera línea tipeamos *#use tutorial. nof*, luego incluimos cada una de las palabras del texto seguida de la etiqueta categorial y el modelo de flexión y, por último lo guardamos con el nombre *tutorial1.dic* (recordemos que la extensión *.dic* se agrega automáticamente). El diccionario resultante es:



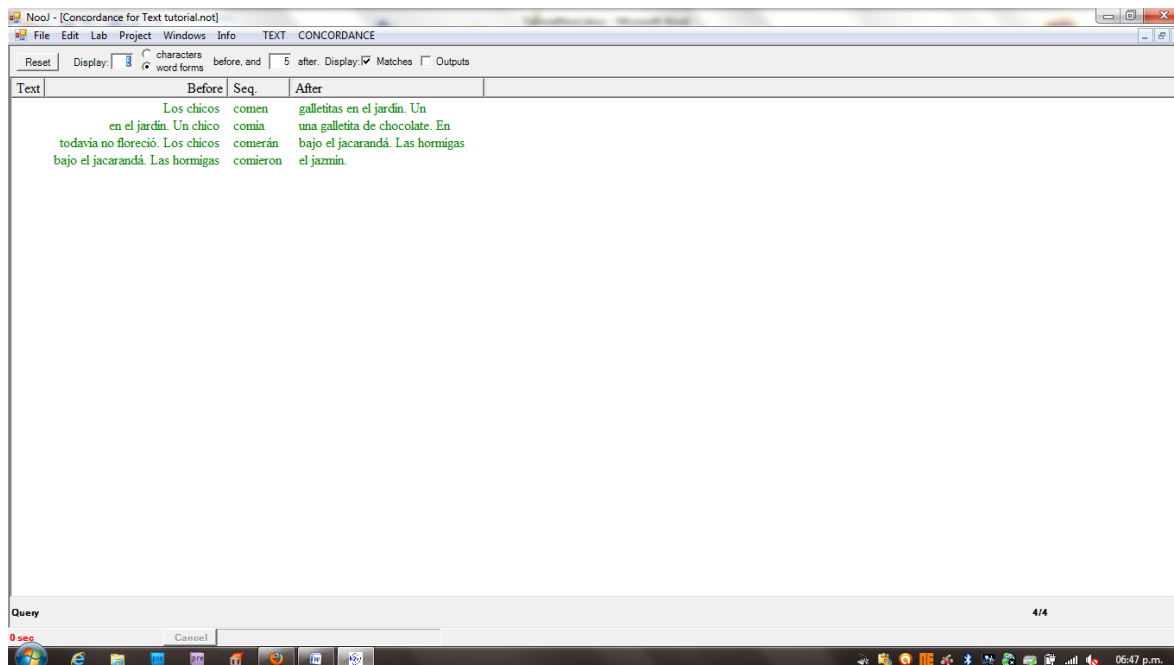
1.3. Resultados del análisis y las consultas

Antes de aplicar el nuevo diccionario al análisis del texto, debemos compilarlo como se explicó en la clase anterior. Una vez que lo tengamos compilado, abrimos el texto *tutorial.not* (*File > Open > Text*); en la barra de menú, tocamos *Info > Preferences*; en la pestaña *General*, seleccionamos *Lenguaje Name sp*; en *Lexical Analysis*, desmarcamos el diccionario *Tutorial.dic*, marcamos *Tutorial1.dic* y hacemos clic sobre *Apply*; y, por último, hacemos clic en *TEXT>Linguistic Analysis*.

Si marcamos el casillero *Show Text Annotation Structure*, podemos ver que Nool reconoce todos los sustantivos y verbos, e identifica el lema, que es la entrada que se le dio en el diccionario (sustantivo masculino y verbo infinitivo) y los rasgos morfológicos que lo caracterizan:



Ahora, haciendo clic en *TEXT > Locate*, además de buscar una categoría con un rasgo morfológico determinado, se pueden buscar todas las variantes morfológicas de un lema, para hacerlo basta escribir en minúsculas cualquier variante del lema entre <>. El resultado de la búsqueda <comer> es:

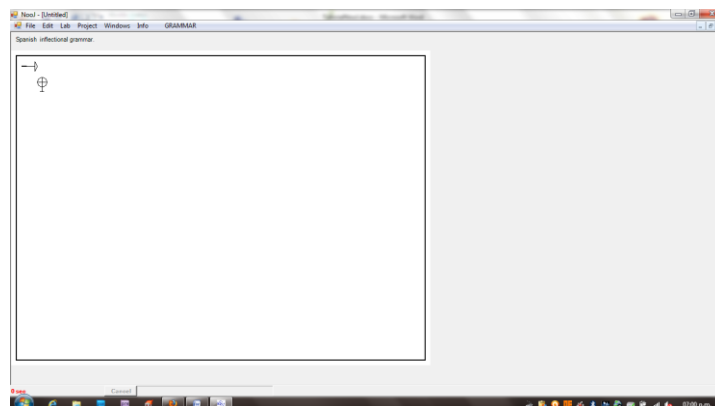


Si ponemos *comer* sin paréntesis angulares, no obtenemos ningún resultado porque, en este caso, NooJ busca la palabra exacta, que no se encuentra en el texto; si ponemos *comen* sin paréntesis angulares obtendremos una sola salida, en cambio, si ponemos <comen> obtendremos todas las variantes. Esta función podría ser de utilidad para personas que estudian el español como segunda lengua porque les permitiría identificar todas las variantes morfológicas que aparecen en un texto.

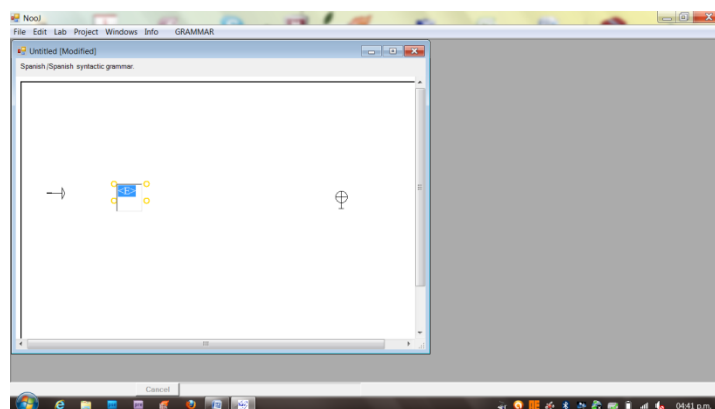
1.4. Editor gráfico de gramáticas

Las gramáticas gráficas utilizan los mismos operadores y producen los mismos resultados que las gramáticas regulares. Pueden resultar más sencillas porque evitan el uso de paréntesis y punto y coma.

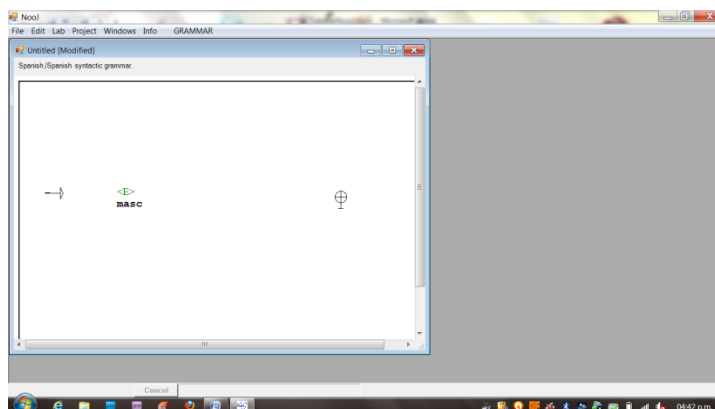
Para desarrollar con el sistema de gráficos la misma gramática morfológica que hicimos con el sistema de reglas, hacemos clic en *File > New > Grammar*, verificamos que se encuentre seleccionado *sp* como idioma de entrada y de salida, marcamos el casillero circular *graphical editor*, luego, tocamos *Infection & Derivation* y aparece la siguiente pantalla:



La flecha representa el estado inicial y el círculo con la cruz el estado final. Entre ambos, debemos crear los estados intermedios; para ello, arrastramos con el *mouse* el círculo hacia abajo a la derecha y la flecha hacia abajo; luego, mientras mantenemos presionada la tecla *ctrl* hacemos clic y se crea un nuevo nodo vacío, representado por <E>.

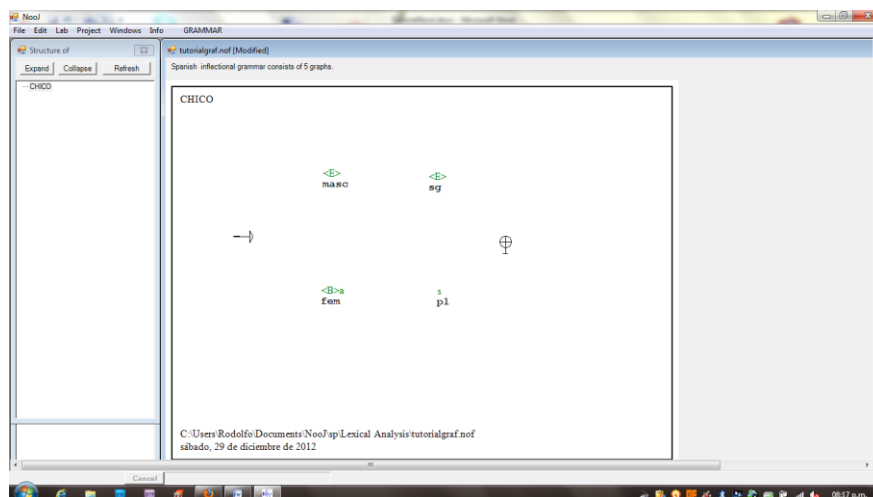


Para crear el gráfico que corresponde a CHICO, en el primer nodo dejamos la categoría vacía y agregamos la etiqueta "/masc" y, mientras mantenemos presionada la tecla *ctrl* presionamos *enter*; luego, hacemos clic en cualquier lugar de la pantalla para deseleccionar ese nodo.



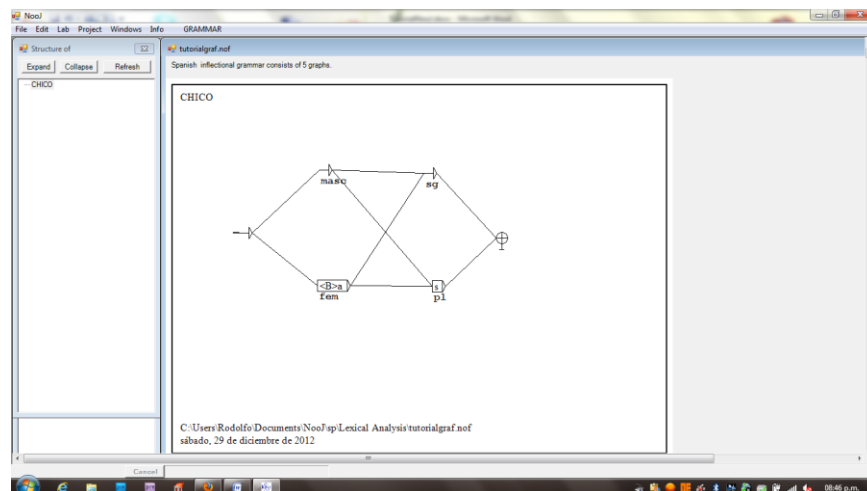
Al lado de este nodo, creamos otro, donde a la categoría vacía le agregamos la etiqueta "/sg"; para el femenino, borramos la categoría vacía y escribimos "a/fem"; para el plural, borramos la

categoría vacía y agregamos "s/pl" (recuerde que los operadores del programa van entre <> y los caracteres (letras) van sin paréntesis angulares). Para poder asociar la entrada del diccionario con el modelo de flexión, debemos asignarle un nombre al modelo; a tal efecto, hacemos clic en *GRAMMAR > Show Structure* y a la izquierda se despliega un nuevo panel con la palabra *main*, hacemos doble clic sobre ella y escribimos CHICO y guardamos la gramática con el nombre *tutorialgraf*

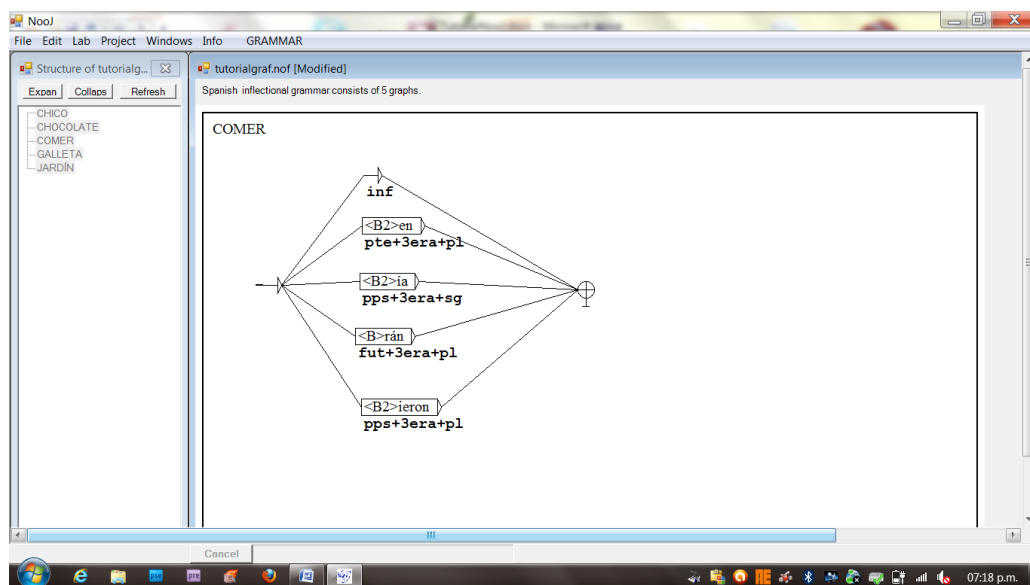
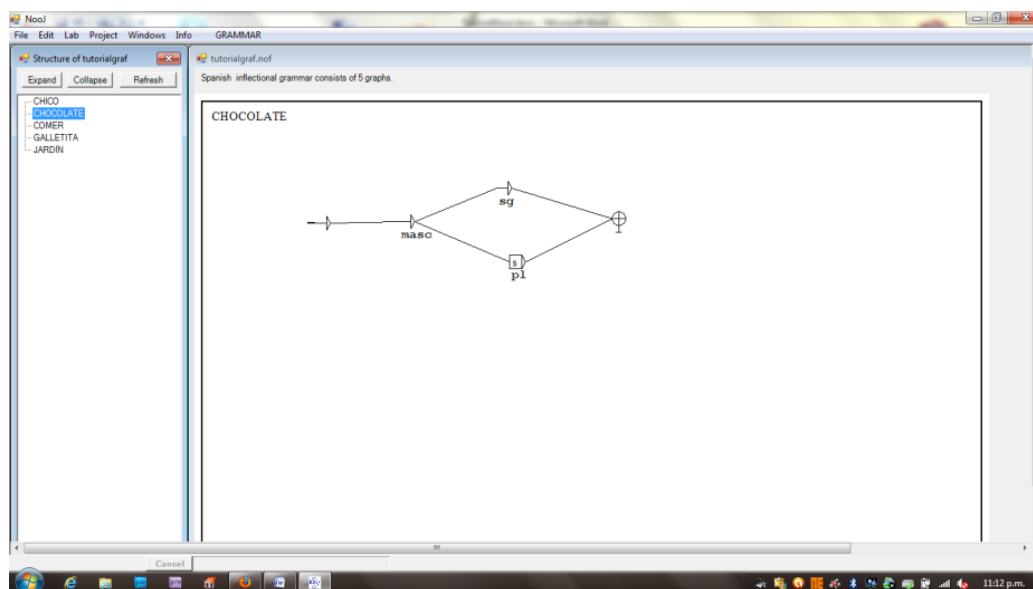


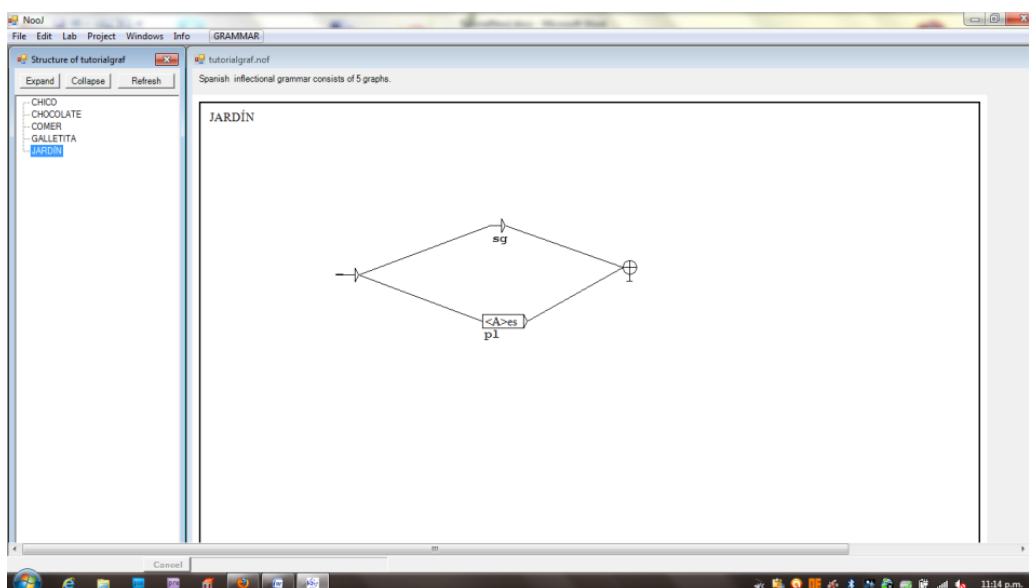
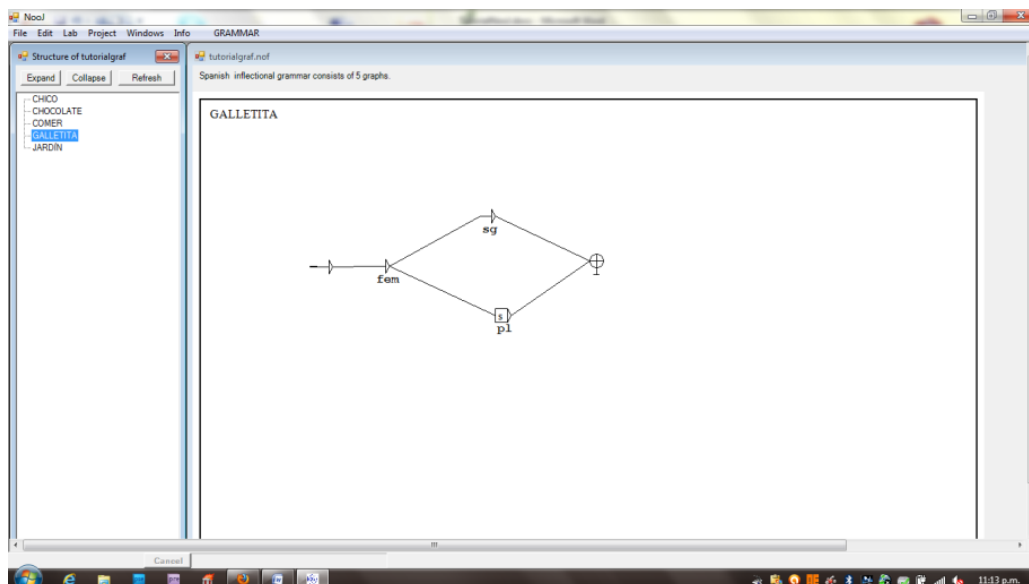
A continuación, es necesario marcar el camino que debe seguir el proceso de NooJ uniendo los nodos. En este caso, el proceso es de izquierda a derecha, pero también es posible hacerlo en sentido inverso o unir un nodo sobre sí mismo para repetirlo. Por esta razón, suele suceder que se establezcan conexiones inesperadas; para borrarlas, es suficiente repetir el procedimiento que se utilizó para crearlas.

Para vincular los nodos, seleccionamos el nodo inicial (la flecha) haciendo clic sobre él (aparecen cuatro círculos de colores en cada uno de sus vértices) y lo unimos con el de arriba (<E>/masc) con otro clic, después repetimos el procedimiento para unir el nodo inicial con el de abajo (a/fem), luego unimos cada uno de ellos con <E>/sg y con s/pl y, a estos con el estado final (el círculo con la cruz). El resultado es el siguiente:



Haciendo clic en *GRAMMAR>New Graph*, en el panel izquierdo aparece una nueva ramificación al mismo nivel de *CHICO* con el nombre *Untitled 0*; haciendo clic sobre *Untitled 0*, cambiamos el nombre por el del modelo que vamos a crear (en nuestro caso, *CHOCOLATE*) y, en el panel derecho creamos, el gráfico que corresponde. A continuación se muestran los gráficos de cada uno de los modelos utilizados en el diccionario, cada uno de ellos se crea haciendo clic en *GRAMMAR>New Graph* y cambiando el nombre *Untitled 0* por el nombre del modelo:





Una vez que hayamos incluido todos los modelos; guardamos la gramática, reabrimos el diccionario *tutorial1.dic*; en la primera línea, reemplazamos *tutorial.nof* por *tutorialgraf.nof* y compilamos nuevamente el diccionario. De este modo, podemos comprobar que la gramática gráfica funciona exactamente igual que la gramática regular. Consecuentemente, se obtendrá el mismo análisis del texto y las consultas arrojarán los mismos resultados.

2. Gramáticas derivacionales

Las gramáticas derivacionales utilizan los mismos operadores que las gramáticas morfológicas y, también permiten optar entre utilizar sistemas de reglas o gráficos. En las gramáticas y el diccionario que elaboramos, *galletita* se trató como una palabra simple; pero, en realidad, se trata

de un diminutivo, es decir, de una palabra derivada. Si queremos tratarla como tal, deberemos reemplazar en las gramáticas morfológicas (*tutorial.nof* o *tutorialgraf.nof*) el modelo *GALLETITA* por *GALLETA* y en el diccionario (*tutorial.dic*) modificaremos la entrada *galletita,N+FLX=GALLETITA* por *galleta,N+FLX=GALLETA* (se deben cambiar todas las palabras que siguen el modelo *GALLETITA*), luego crearemos una gramática derivacional. Tenga en cuenta que el diccionario utiliza solo la gramática que declaramos en la primera línea, no obstante, sugerimos cambiar las dos para conservar la equivalencia)

A continuación, vamos a elaborar una gramática derivacional que genere y reconozca los diminutivos de los sustantivos del texto. Hacemos clic en *File>New > Grammar*, seleccionamos *sp*, *rule editor* y hacemos clic en *Inflection & Derivation*. En esta nueva gramática vamos a crear los modelos para los diminutivos, que guardaremos con el nombre *derivación* (NooJ agrega la extensión *.nof*):

IT = <L>it/N+dim; (*galletita*: va un carácter hacia atrás, agrega *it* y, luego, la etiqueta categorial N con la especificación +dim)

ITO= ito/N+dim; (*chocolatito*: borra la *e* final, agrega *ito* y, luego, la etiqueta categorial N con la especificación +dim)

QUIT= <L>quit/N+dim; (*chiquito*: va un carácter hacia atrás, borra la *c*, agrega *quit* y, luego, la etiqueta categorial N con la especificación +dim)

GUIT= <L>uit/N+dim; (*hormiguita*: va un carácter hacia atrás, agrega *guit* y, luego, la etiqueta categorial N con la especificación +dim)

CITO= <A>cito/N+dim; (*jardincito*, *jazmincito* y *jacarandacito*: quita el acento de la vocal, agrega *cito* y, luego, la etiqueta categorial N con la especificación +dim).

Para que la etiqueta del tipo de derivación *dim* se pueda ver como una columna del diccionario, deberemos modificar el archivo de propiedades (*_properties.def*) y agregar debajo de la línea *FLEXIÓN* una línea *DERIVACIÓN = dim*; y, debajo de las propiedades de N, *N_derivación = dim*; (no omite guardar)

Si se utilizan otros rasgos para caracterizar otros tipos derivación (aumentativa, despectiva, etc.), en la misma línea, antes del punto y coma, agregaremos precedido de la barra recta "|" en la lista de rasgos derivacionales y, luego, como una propiedad derivacional de la categoría que corresponda.

Una vez concluida la gramática derivacional, debemos indicarle al diccionario que la utilice, para ello en la segunda línea (después de *#use tutorialgraf.nof*) agregamos *#use derivación.nof*.

También se deberá indicar el modelo derivativo que se asigna a cada sustantivo agregando *+DRV* seguido del nombre del modelo de flexión; el diccionario resultante es el siguiente:


```
# Nool - [tutorial2.dic]
File Edit Lab Project Windows Info DICTIONARY
Dictionary contains 9 entries

# Nool V3
# Dictionary
#
# Input Language is: sp
#
# Alphabetical order is not required.
#
# Use inflectional & derivational paradigms' description files (.nof), e.g.:
# Special Command: $use paradigms.nof
#
# Special Features: +NW (non-word) +FXC (frozen expression component) +UNAMB (unambiguous lexical entry)
#                  +FLX= (inflectional paradigm) +DRV= (derivational paradigm)
#
# Special Characters: '\' '\'' '+' ',' '#' '.'
#
# use tutorial2.nof
# use diminutivos.nof

chico,N+FLX=CHICO+DRV=QUIT
galleta,N+FLX=GALLETA+DRV=IT
jardin,N+FLX=JARDIN+DRV=CITO
chocolate,N+FLX=CHOCOLATE+DRV=ITO
jacaranda,N+FLX=CHOCOLATE+DRV=CITO
jazmin,N+FLX=JARDIN+DRV=CITO
hormiga,N+FLX=GALLETA+DRV=QUIT
comer,V+FLX=COMER
florecer,V+FLX=COMER
```

Luego compilamos el diccionario: *Lab>Dictionary>set*, seleccionamos el diccionario *tutorial.doc > compile*, después *inflect*, que genera automáticamente *tutorial-flx.dic*. Si abrimos este archivo (*File>Open>Dictionary*), obtenemos las siguientes salidas:

chico, chico, N+FLX=CHICO+DRV=QUIT+masc+sg
chicos, chico, N+FLX=CHICO+DRV=QUIT+masc+pl
chica, chico, N+FLX=CHICO+DRV=QUIT+fem+sg
chicas, chico, N+FLX=CHICO+DRV=QUIT+fem+pl
chiquito, chico, N+dim+FLX=CHICO+DRV=QUIT+masc+sg
chiquitos, chico, N+dim+FLX=CHICO+DRV=QUIT+masc+pl
chiquita, chico, N+dim+FLX=CHICO+DRV=QUIT+fem+sg
chiquitas, chico, N+dim+FLX=CHICO+DRV=QUIT+fem+pl
galleta, galleta, N+FLX=GALLETA+DRV=IT+fem+sg
galletas, galleta, N+FLX=GALLETA+DRV=IT+fem+pl
galletita, galleta, N+dim+FLX=GALLETA+DRV=IT+fem+sg
galletitas, galleta, N+dim+FLX=GALLETA+DRV=IT+fem+pl
jardín, jardín, N+FLX=JARDÍN+DRV=CITO+masc+sg
jardines, jardín, N+FLX=JARDÍN+DRV=CITO+masc+pl
jardincito, jardín, N+dim+FLX=JARDÍN+DRV=CITO+masc+sg
jardincitoes, jardín, N+dim+FLX=JARDÍN+DRV=CITO+masc+pl
chocolate, chocolate, N+FLX=CHOCOLATE+DRV=ITO+masc+sg
chocolates, chocolate, N+FLX=CHOCOLATE+DRV=ITO+masc+pl
chocolatito, chocolate, N+dim+FLX=CHOCOLATE+DRV=ITO+masc+sg
chocolatitos, chocolate, N+dim+FLX=CHOCOLATE+DRV=ITO+masc+pl
jacarandá, jacarandá, N+FLX=CHOCOLATE+DRV=CITO+masc+sg
jacarandás, jacarandá, N+FLX=CHOCOLATE+DRV=CITO+masc+pl
jacarandacito, jacarandá, N+dim+FLX=CHOCOLATE+DRV=CITO+masc+sg
jacarandacitos, jacarandá, N+dim+FLX=CHOCOLATE+DRV=CITO+masc+pl
jazzmín, jazzmín, N+FLX=JARDÍN+DRV=CITO+masc+sg
jazzmines, jazzmín, N+FLX=JARDÍN+DRV=CITO+masc+pl

jazmincito,jazmín,N+dim+FLX=JARDÍN+DRV=CITO+masc+sg
jazmincitoes,jazmín,N+dim+FLX=JARDÍN+DRV=CITO+masc+pl
 hormiga,hormiga,N+FLX=GALLETA+DRV=GIT+fem+sg
 hormigas,hormiga,N+FLX=GALLETA+DRV=GIT+fem+pl
 hormiguita,hormiga,N+dim+FLX=GALLETA+DRV=GIT+fem+sg
 hormiguitas,hormiga,N+dim+FLX=GALLETA+DRV=GIT+fem+pl
 florecer,florecer,V+FLX=COMER+inf
 florecerán,florecer,V+FLX=COMER+fut+3era+pl
 florecen,florecer,V+FLX=COMER+pte+3era+sg
 florecía,florecer,V+FLX=COMER+pps+3era+sg
 florecieron,florecer,V+FLX=COMER+pps+3era+pl

Como puede verse en las líneas resaltadas, el plural del diminutivo de *jardín* y *jazmín* es incorrecto. Esto sucede a causa de que, salvo que haya alguna otra indicación, NooJ asigna al derivado el mismo modelo flexivo que a la palabra simple. En los otros casos no se produce ningún error porque la palabra derivada tiene el mismo modelo flexivo que la simple, por ejemplo, en *chico* el masculino singular coincide con la entrada, es decir, no se agrega ni se quita ningún carácter para generar esta forma flexiva, el diminutivo es idéntico: *chiquito* es masculino singular sin agregar ni quitar ningún carácter; si borramos la *o* y agregamos *a*, generamos el singular tanto en la variación *chico* / *chica* como en *chiquito* / *chiquita* y, si agregamos *s*, transformamos las formas del singular y del plural en ambos casos (*chicos* / *chicas*, *chiquitos* / *chiquitas*). En cambio, *jardín* y *jazmín* tienen plural en *-es* mientras *jazmincito* y *jardincito* tienen plural en *-s*.

Cuando el modelo flexivo del derivado no coincide con el modelo de la palabra simple de la que deriva es posible asignarle otro modelo de flexión agregando ":" y el nombre del nuevo modelo. En el caso de *jardincito* y *jazmincito*, son palabras masculinas invariables con plural en *-s*, es decir, siguen el modelo flexivo de *chocolate*. Por lo tanto, para salvar el error, en el diccionario, al modelo derivacional debemos agregar el nuevo modelo flexivo:

jazmín,N+FLX=JARDÍN+DRV=CITO:CHOCOLATE
 jardín,N+FLX=JARDÍN+DRV=CITO:CHOCOLATE

Si hacemos esta modificación en nuestro diccionario y volvemos a compilarlo, podremos observar que los errores desaparecen.

Además de modificar el modelo flexivo, es posible modificar la etiqueta categorial del derivado. Por ejemplo *comedor* es un sustantivo derivado del verbo *comer*; en un diccionario podemos optar por tratar estas nominalizaciones como palabras simples y darles entrada como sustantivo, o bien, tratarlas como derivaciones. Para ello, debemos crear en la gramática un modelo como el siguiente:

DOR= dor/N+nom;

Si a cualquier verbo de un diccionario agregamos la marca *+DRV=DOR:JARDÍN*, por indicación de la gramática derivacional, el sistema borra un carácter, agrega *dor*, lo etiqueta como *N* seguida por el rasgo *nom* (nominalización) (no olvidemos agregar el rasgo en el archivo *_properties.def*) y, por indicación de la gramática flexiva, lo flexiona como *JARDÍN*. El inconveniente de no introducir

comedor directamente en el diccionario es que, por tratarse de una palabra derivada, no es posible agregar una nueva derivación, como sería la de diminutivo.

Nótese que el modelo flexivo *JARDÍN* tiene el operador <A>, que indica que quite el acento de la vocal; como el derivado *comedor* no tiene ninguna vocal acentuada, el operador no encuentra su argumento y se neutraliza, consecuentemente, no genera error.

Ejercitación

1) Modifique el texto y agregue las oraciones:

Las hormiguitas hicieron un caminito del jacarandá al jazmín.

El comedor da a un jardincito.

Las hormigas comieron el jazmincito y el jacarandacito del jardín.

2) Analice el texto utilizando el diccionario *tutorial2.dic*

3) Consulte en *TEXT > Locate* cuántos diminutivos y cuántas nominalizaciones aparecen en el texto y anote los resultados.

4) Elabore una gramática derivacional gráfica con los mismos modelos que la gramática regular, denomínela *derivacióngraf* y aplíquela al diccionario *tutorial.dic* (debe reemplazar la gramática *derivación.nof* por esta nueva gramática en la segunda línea del diccionario).

5) Compile el diccionario, vuelva a analizar el texto y efectúe la consulta para verificar que se localicen la misma cantidad de diminutivos y nominalizaciones que en la consulta anterior.

6) Agregue la oración: *La nueva ley prevé que el consumo de marihuana sea legal, pero el tráfico seguirá siendo ilegal.*

7) En la gramática flexiva, agregue los modelos de flexión faltantes para las palabras simples (*ver, ser, legal, traficar, seguir*)

8) En una de las gramáticas derivacionales (regular o gráfica) agregue los modelos de las palabras derivadas (*prever, tráfico, ilegal*). Recuerde que el operador <LW> se utiliza para llevar el cursor al principio de la palabra.

9) En *_properties.def*, agregue los rasgos flexivos y derivacionales que faltan.

10) Agregue al diccionario las nuevas palabras con su etiqueta categorial y el modelo flexivo y derivacional que les corresponda (asegúrese de que el diccionario utilice las gramáticas modificadas).

11) Compile el diccionario, genere nuevamente *tutorial-flex.dic* y verifique que todas las entradas sean correctas.

12) Analice el texto y efectúe consultas.

13) Agregue al diccionario palabras que sigan los modelos flexivos y derivacionales existentes, compílelo y verifique que todas las formas generadas sean correctas.

Clase 3

En esta clase aprenderemos a:

- Crear gramáticas productivas
- Realizar consultas aplicando los diccionarios y gramáticas productivas
- Comparar la eficiencia de ambos
- Crear gramáticas sintácticas gráficas simples
- Utilizar gráficos subyacentes
- Aplicar las gramáticas sintácticas al análisis del texto

1. Gramáticas productivas

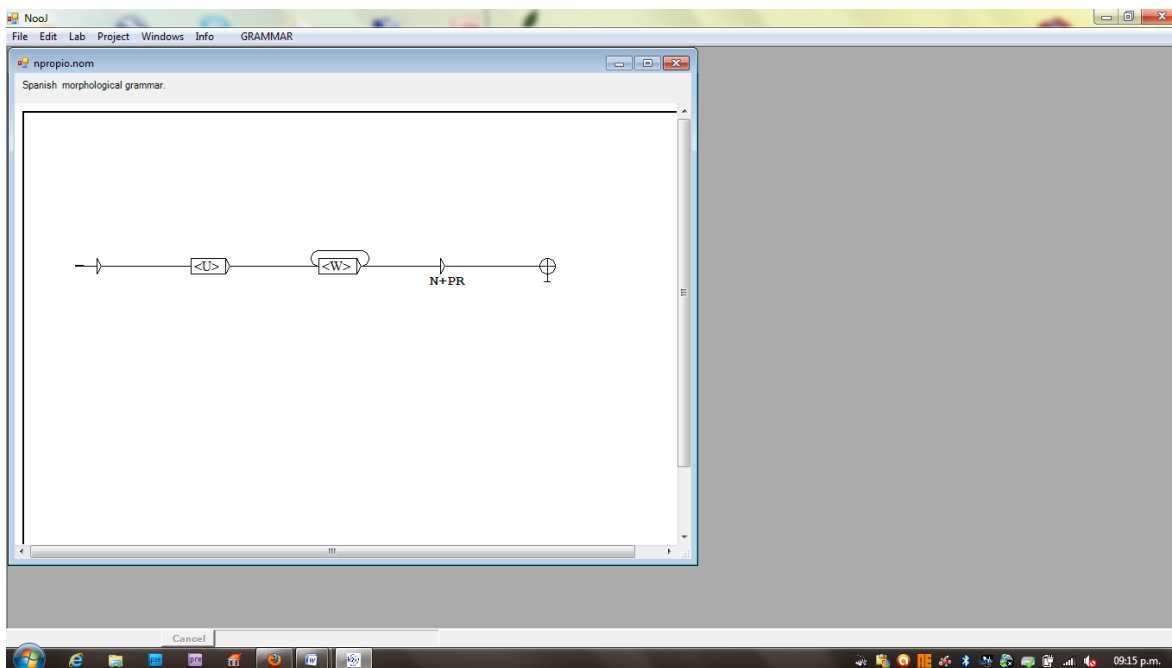
Las gramáticas productivas se utilizan para reconocer palabras que no están incluidas en los diccionarios, pero que tienen determinados rasgos formales comunes. Por ejemplo, sería prácticamente imposible crear un diccionario con todos los nombres propios que pueden aparecer en un gran corpus de textos periodísticos; en ese caso, resultaría de cierta utilidad crear una gramática productiva que etiquetara las palabras que comienzan en mayúscula como nombres propios.

Haciendo clic en *File > New > Grammar*; seleccionamos *sp* como lenguaje de entrada y salida, *graphical editor*; y hacemos clic en *Productive Morphology*; se despliega una pantalla idéntica a la del editor gráfico de gramáticas flexivas y derivacionales.

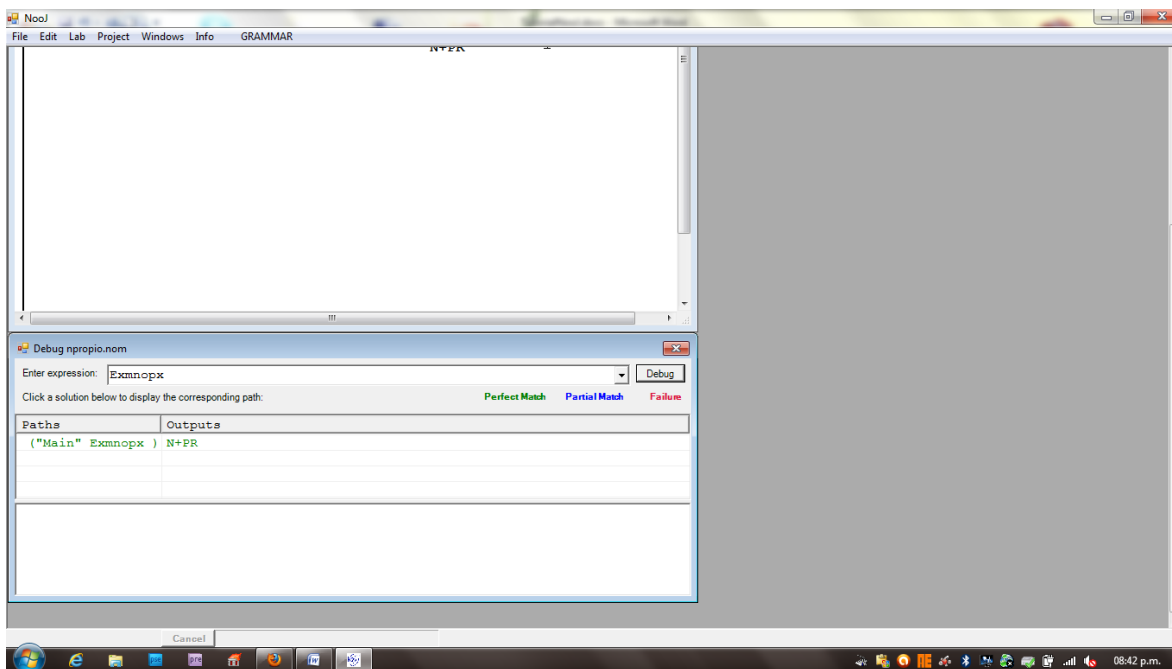
El módulo morfológico de Nooj usa los siguientes símbolos especiales:

- <L> cualquier letra
- <U> cualquier letra mayúscula
- <W> cualquier letra minúscula
- <A> cualquier letra acentuada
- <N> cualquier letra sin acento

Para crear una gramática que etiquete cualquier palabra en mayúscula como nombre propio, ubicamos el estado inicial y el estado final, y creamos un nuevo nodo haciendo *ctrl* clic, reemplazamos el nodo vacío <E>, que aparece automáticamente cuando creamos un nodo, por <U> (cualquier letra mayúscula); hacemos clic en cualquier parte de la pantalla para deseleccionar ese nodo; creamos un nuevo nodo <W> (cualquier letra minúscula), volvemos a hacer clic en cualquier parte de la pantalla para deseleccionarlo y creamos un nuevo nodo donde a la cadena vacía <E> le agregamos, precedida de barra, la etiqueta N+PR. Luego unimos los nodos de izquierda a derecha, hacemos doble clic sobre el nodo <W> para que se una sobre sí mismo, lo que indica un número indeterminado de letras minúsculas, y guardamos con el nombre *npropio*, al que se le agrega automáticamente la extensión *.nom*. La gramática resultante expresa que cualquier secuencia que empieza con una letra mayúscula y sigue con un número indeterminado de minúsculas se debe etiquetar como N+Pr:



Si hacemos clic en la barra de menú *GRAMMAR > Debug...* y bajamos la barra de desplazamiento de la ventana principal, vemos un panel donde podemos introducir distintas cadenas para verificar cuáles son las cadena incluidas y excluidas por esta gramática. Dado que la gramática es inespecífica, reconocerá como nombre propio una cadena como *Exmnopx*, en cambio no reconocerá un nombre propio escrito en minúscula:



Antes de comprobar el funcionamiento de esta gramática en el texto, en el archivo de propiedades agregamos el rasgo

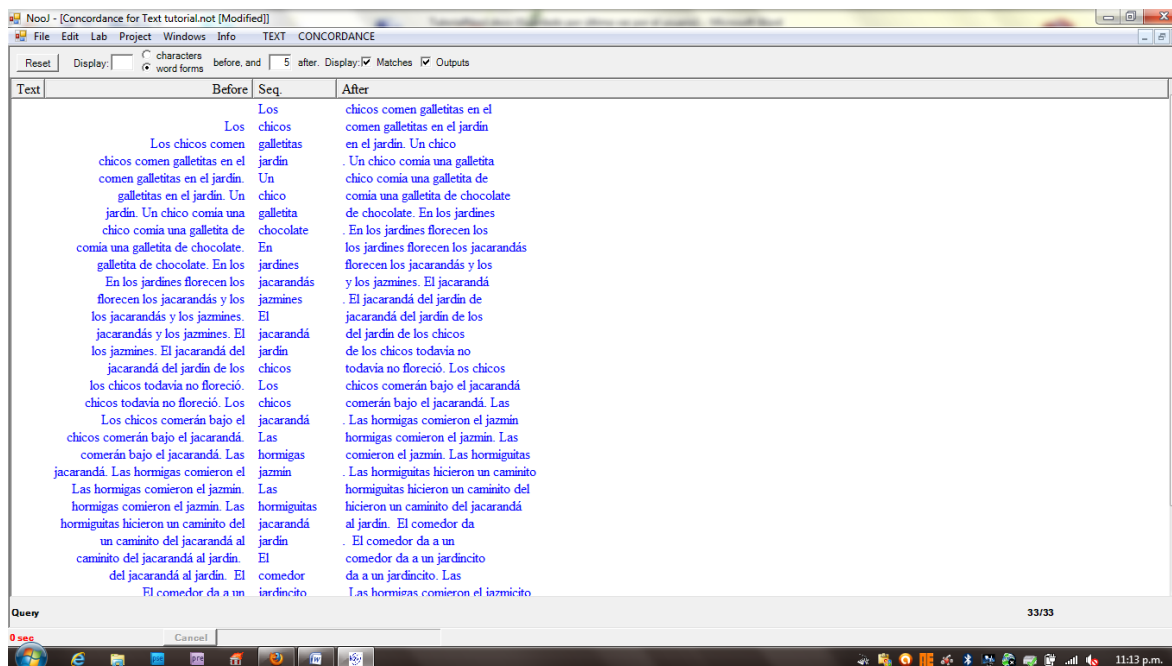
$N_tipo = PR;$

Luego abrimos el texto, en *Info > Preferences*, en el panel superior derecho de la pestaña *Lexical Analysis*, marcamos la gramática *npropio.nom* y, luego, *Apply*. Si hacemos un nuevo análisis del texto, veremos que todas las palabras mayúsculas de comienzo de oración resultan etiquetadas como N+PR, lo cual es incorrecto.

Con la finalidad de seguir analizando el funcionamiento del sistema, vamos a agregar en el archivo de propiedades las etiquetas correspondientes a estas palabras (*ART* y *PREP*) y los rasgos que determinan al artículo:

ART_tipo = det | indet;
ART_género = masc | fem;
ART_número = sg | pl;

Luego, incluimos en el diccionario *tutorial.dic* los artículos con sus rasgos de género y número, y las preposiciones que aparecen en el texto, y compilamos. En un nuevo análisis del texto, vemos que cuando estas palabras aparecen en mayúscula NooJ aplica una doble (*ART* y *N+Pr*). Si en *TEXT > Locate* hacemos la búsqueda de la expresión regular <N> (sustantivo), aparecerán erróneamente todos los artículos y las preposiciones que inician oración, como se muestra en la siguiente pantalla.



Para eludir el doble análisis y el etiquetado erróneo, debemos hacer clic en *Info > Preferences*; en la parte superior derecha de la pestaña, vemos el título "*Priority Level:*" seguido de tres botones con letras rojas: *High*, *Regular*, *Low*. Seleccionando con un clic las gramáticas productivas, mediante el uso de estos botones podemos asignarle distintos niveles de prioridad; si asignamos bajo nivel (*Low*), evitamos que aparezca la doble etiqueta; en este caso, la gramática se aplica

únicamente a palabras que no están en el diccionario, consecuentemente, el resultado de la búsqueda de la expresión regular <N> también será correcto.

En síntesis, las gramáticas productivas son de utilidad si se les asigna bajo nivel de prioridad y si los diccionarios que utilizamos son completos; de lo contrario, arrojarán demasiados resultados erróneos, que habrá que revisar manualmente.

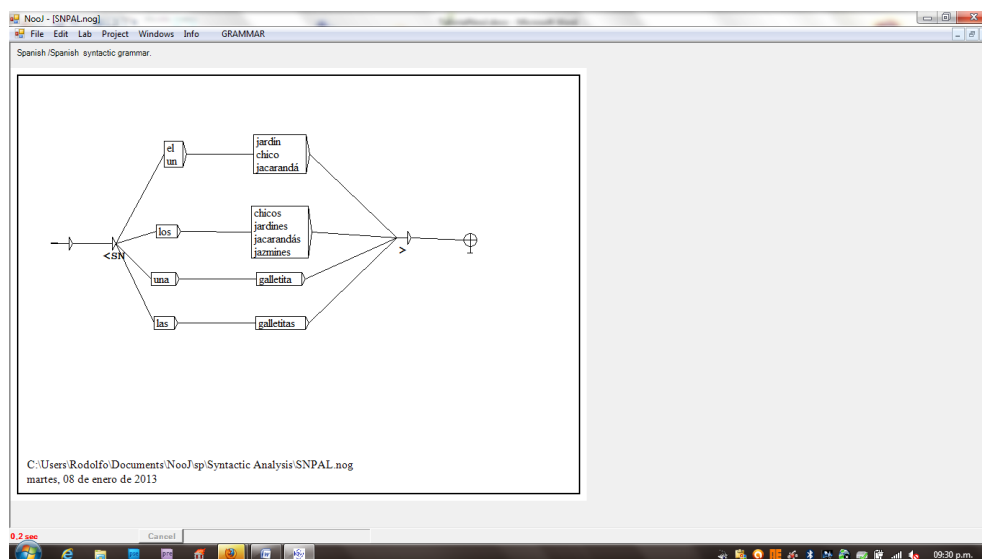
2. Gramáticas sintácticas

Mientras las unidades mínimas de las gramáticas flexivas y derivativas, y de las productivas son los caracteres, la unidad mínima de las gramáticas sintácticas son las palabras. Es posible crear gramáticas sintácticas utilizando palabras, por ejemplo, *chico*; lemas, por ejemplo <*chico*>; categorías léxicas, por ejemplo <V>; o categorías acotadas con determinados rasgos, por ejemplo <V+1era>, que incluye cualquier verbo en primera persona. Para poder utilizar lemas o categorías, es necesario que estos estén declarados en los diccionarios.

2.1. Gramáticas sintácticas con palabras

A continuación desarrollaremos una gramática sintáctica que reconozca los sintagmas nominales formados por un determinante y un sustantivo. Hacemos clic en *File > New > Grammar*, verificamos que el lenguaje de entrada y el de salida sea *sp*, marcamos *graphical editor* y hacemos clic sobre el botón *Syntax*. La pantalla que aparece a continuación es idéntica a la de las gramáticas gráficas anteriores, la diferencia es que en las gramáticas sintácticas separan con espacios en blanco las unidades de cada nodo. Los procedimientos de creación y unión de nodos son iguales a las anteriores, por lo tanto, no nos detendremos a explicarlos.

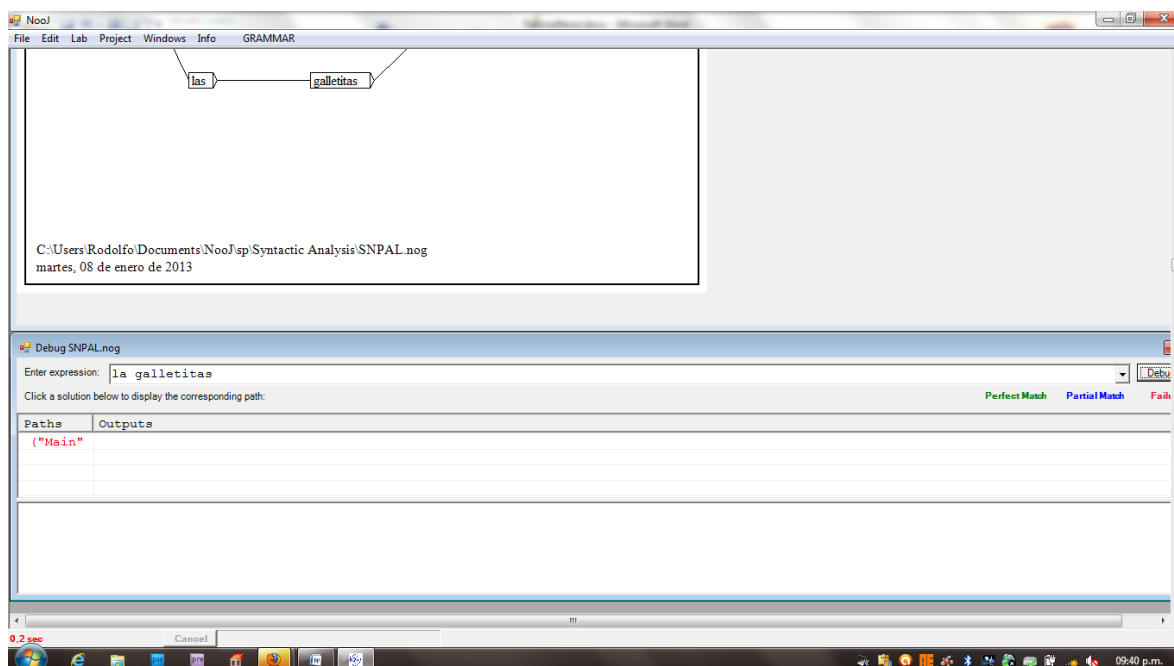
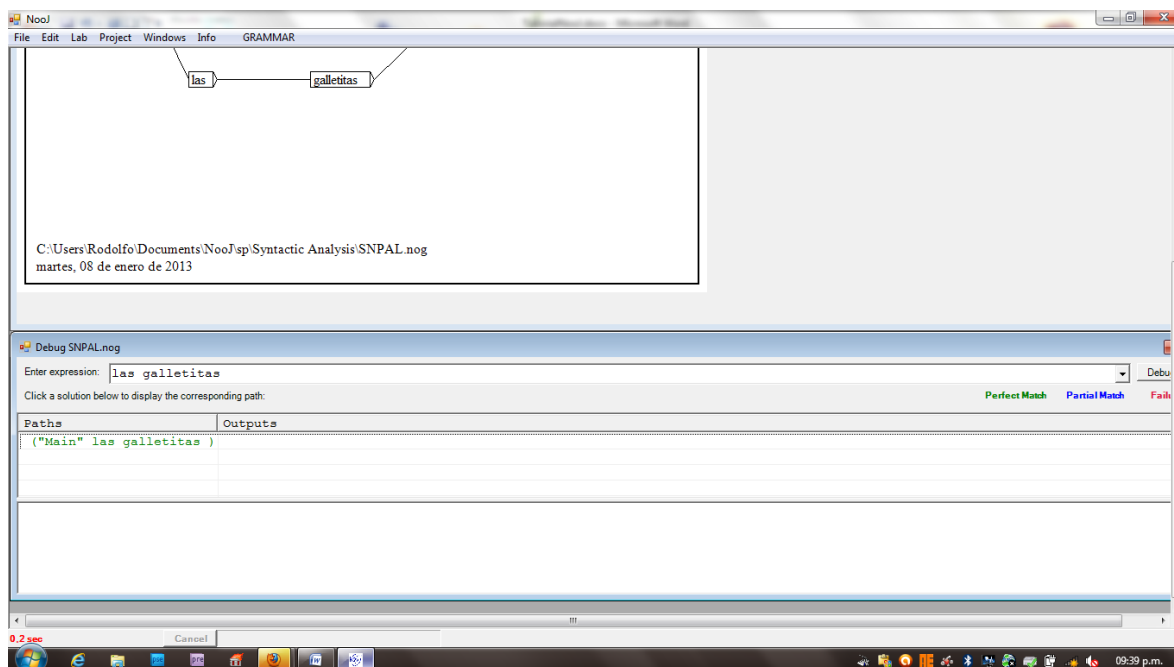
La siguiente, es una gramática del SN desarrollada a partir de palabras, la llamamos SNPAL (NooJ agrega automáticamente la extensión *.nog*):



El nodo que sigue al inicial es un nodo vacío con la etiqueta del sintagma: <E>/<SN>. La etiqueta va precedida de un paréntesis angular de apertura que cierra en el nodo que precede al terminal:

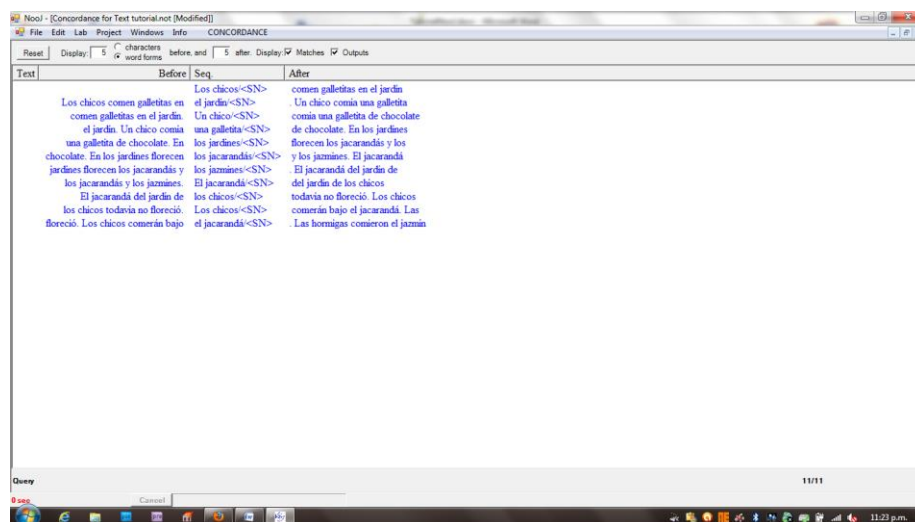
<E>/> esto nodos se incluyen para indicar el comienzo y el final del sintagma. Los artículos y los sustantivos se agrupan en los siguientes nodos por género y número; cuando un nodo tiene más de un elemento después de cada uno de ellos se inserta un *enter*.

Haciendo clic en *GRAMMAR > debug...* y bajando la barra de desplazamiento de la ventana principal, podemos verificar distintas cadenas que la gramática reconoce como válidas o inválidas:



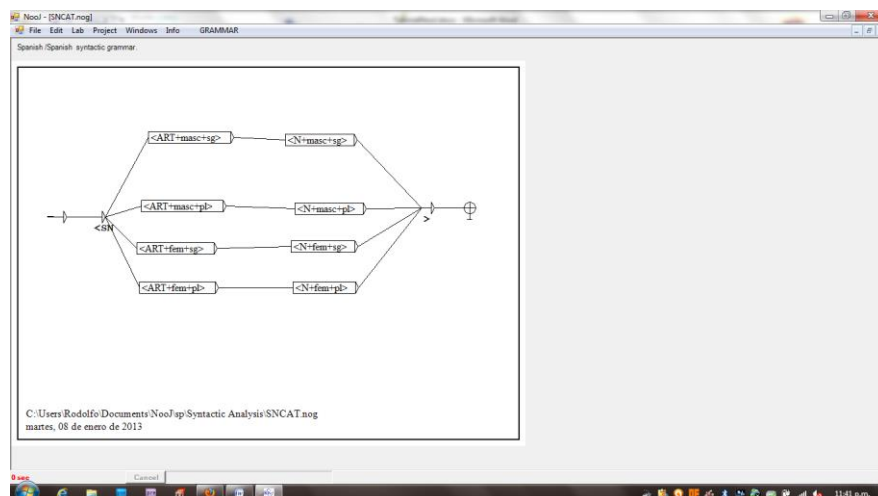
Para analizar el texto con esta gramática; lo abrimos, en la pestaña de *Syntactic Analysis (Info > Preferences)*, hacemos clic en el nombre de la gramática, que se encuentra en panel de la izquierda, lo pasamos al panel de la derecha con el paréntesis angular rojo > que hay entre ambos paneles y hacemos clic en *Apply*. Luego, hacemos clic en *TEXT > Linguistic Analysis* y en *Show Text Annotation Structure*. Así podremos ver que en cada oración los SN se marcan con una línea verde.

En *TEXT>Locate* podemos seleccionar a *NooJ grammar*, buscar la gramática *SNPAL.nog* y ver todas las secuencias del texto que la gramática reconoce como SN haciendo clic en cualquiera de las teclas coloreadas de la parte inferior del panel. Seleccionando el casillero *Outputs* (a la derecha, debajo de la barra de menús) se ve la etiqueta, deseleccionándolo se oculta.



2.2. Gramáticas sintácticas con categorías léxicas

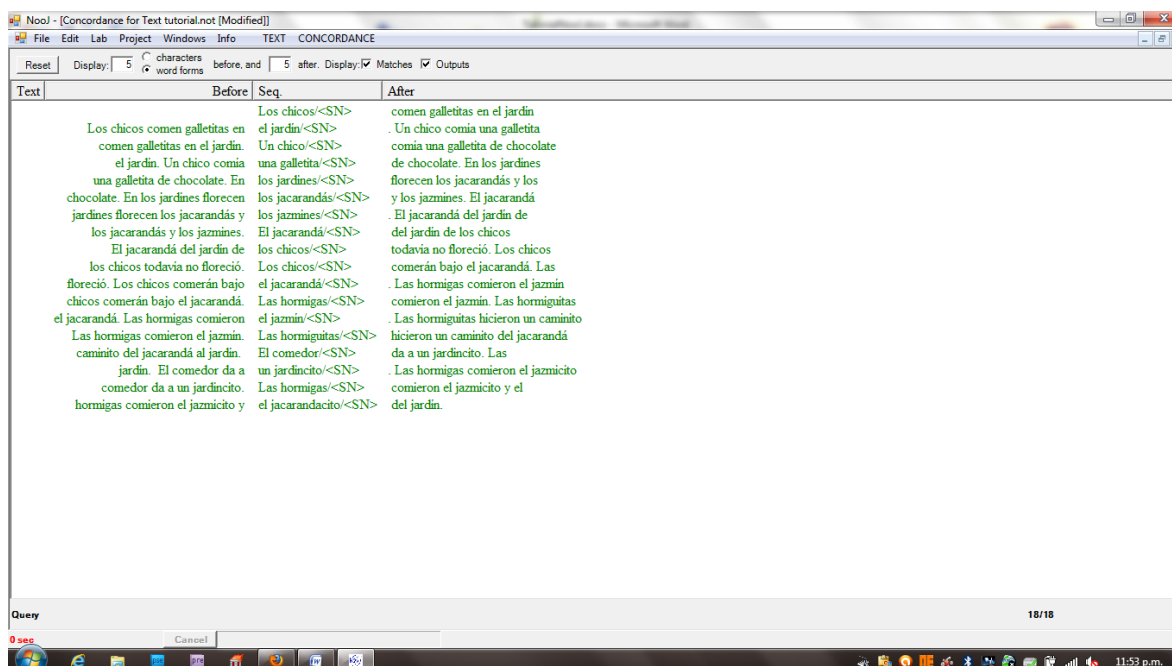
Una gramática que utilice categorías en lugar de palabras sería mucho más eficiente; para desarrollarla, abrimos una nueva gramática, a la que denominaremos *SNCAT*, y reemplazamos las palabras por etiquetas categoriales con sus rasgos de género y número:



Para analizar el texto con esta nueva gramática, en *Syntactic Analysis* (*Info > Preferences*), hacemos clic en el nombre de la gramática *SNCAT.nog*, que se encuentra en panel de la izquierda, lo pasamos al panel de la derecha con el paréntesis angular rojo > que hay entre ambos paneles; en el panel derecho seleccionamos la *SNPAL.nog*, con la X que aparece a la derecha, la eliminamos y hacemos clic en *Apply*.

Luego, hacemos clic en *TEXT > Linguistic Analysis* y en *Show Text Annotation Structure*. Ahora podemos ver que la nueva gramática es mucho más eficiente que la anterior porque reconoce todos los SN sin necesidad de incluir cada una de las palabras.

Si la seleccionamos en *TEXT > Locate > a NooJ grammar*, aparecen las siguientes salidas:



2.3. Gramáticas sintácticas con categorías sintácticas

Además de palabras y categorías léxicas, las gramáticas sintácticas pueden tomar como entradas unidades definidas por otras gramáticas sintácticas. Una gramática sencilla de este tipo es la del sintagma preposicional, formado por una preposición (categoría léxica definida en el diccionario) y un SN (categoría sintáctica definida en una gramática).

Antes de crear una gramática que los reconozca los SP, tenemos que agregar en el diccionario las preposiciones y las contracciones que utilizamos en el texto.

Las preposiciones se agregan directamente con la etiqueta categorial:

en,PREP

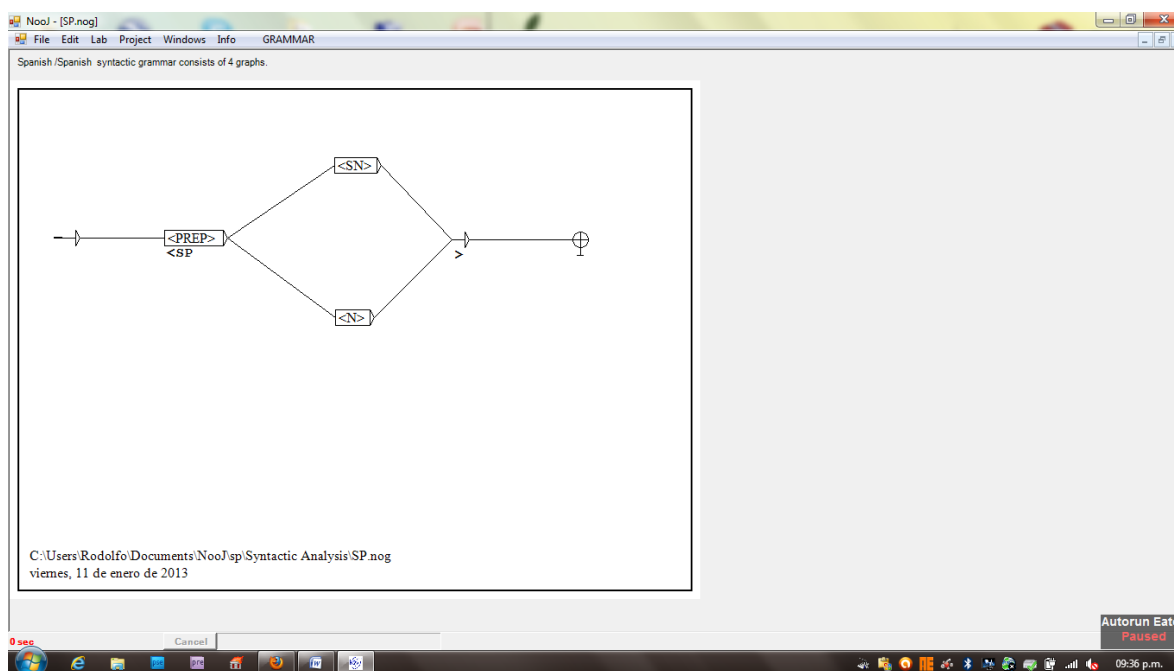
a,PREP

de,PREP
bajo,PREP

Las contracciones deben ser reconocidas como preposición más artículo:

al,<a,PREP><el,ART+masc+sg>
del,<de,PREP><el,ART+masc+sg>

Luego, creamos la siguiente gramática sintáctica, que denominamos *SP.nog*:

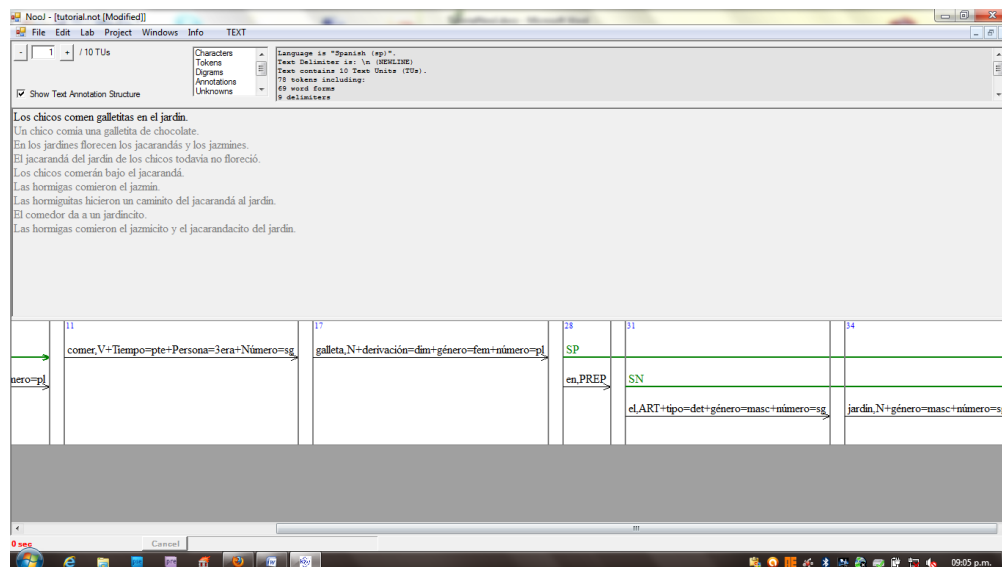


Esta gramática reconoce como SP cualquier secuencia que contenga una preposición y un sustantivo (<N> en la rama inferior del gráfico) o un sintagma nominal (<SN> en la superior); esta última rama presupone la aplicación previa de una gramática sintáctica que haya reconocido el SN.

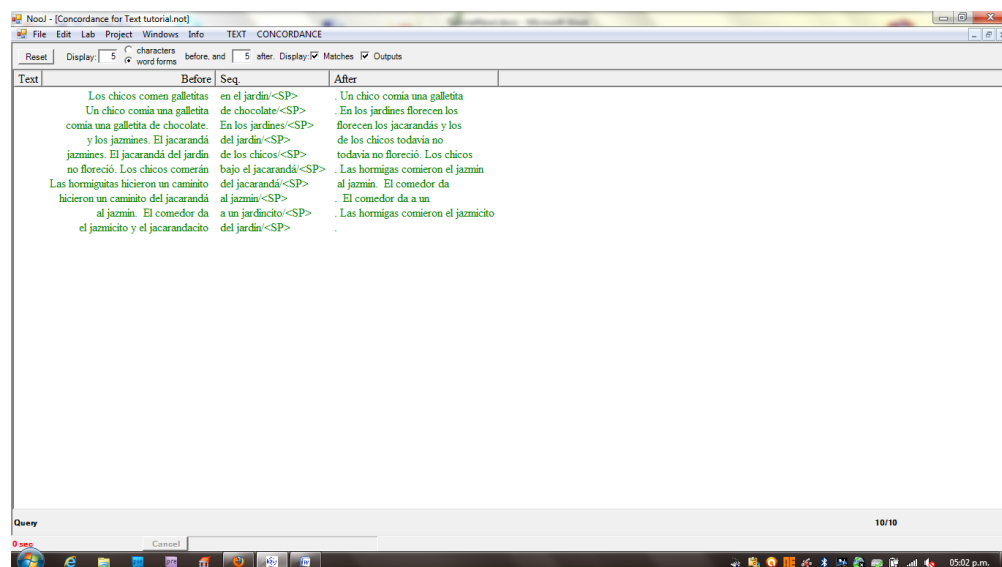
Para aplicar la gramática al análisis del texto, en el panel izquierdo de la pestaña Syntactic Analysis hacemos clic sobre *SP.nog* para seleccionarla y la agregamos al panel derecho mediante >.

Las gramáticas se ordenan en la secuencia en la que las agregamos, por lo tanto, como *SP.nog* se agregó después de *SNCAT.not*, aparece con el número 2. Si por alguna razón quedaran en otro orden, habría que reordenarlas seleccionándolas con un clic y usando las letras rojas **H**, que sube la gramática seleccionada o **L**, que la baja, ya que el orden en el que aparecen en el panel derecho es relevante: *SNCAT.not* debe estar antes que *SP.nog* porque la salida del primero es una entrada del segundo y, si se invierte el orden, no se produce el análisis.

El resultado del análisis es el siguiente:



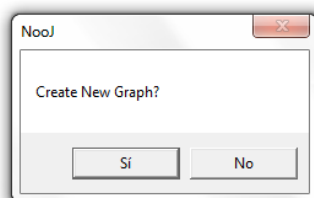
Como se puede ver, el análisis tiene dos niveles: el de SN (ART + N) y el de SP (ART + SN). Si seleccionamos *SP.nog* en *TEXT>Locate >a NooJ grammar*, aparecen las siguientes salidas:



2.4. Gramáticas sintácticas con gráficos subyacentes

En la gramática anterior, el SN es una unidad lingüística independiente, por eso no aparece como un elemento sintáctico del SP en la salida de la consulta *TEXT > Locate > a NooJ grammar*. Si queremos que en esta consulta se visualice que SN es una unidad sintáctica, debemos introducir el SN como un gráfico subyacente.

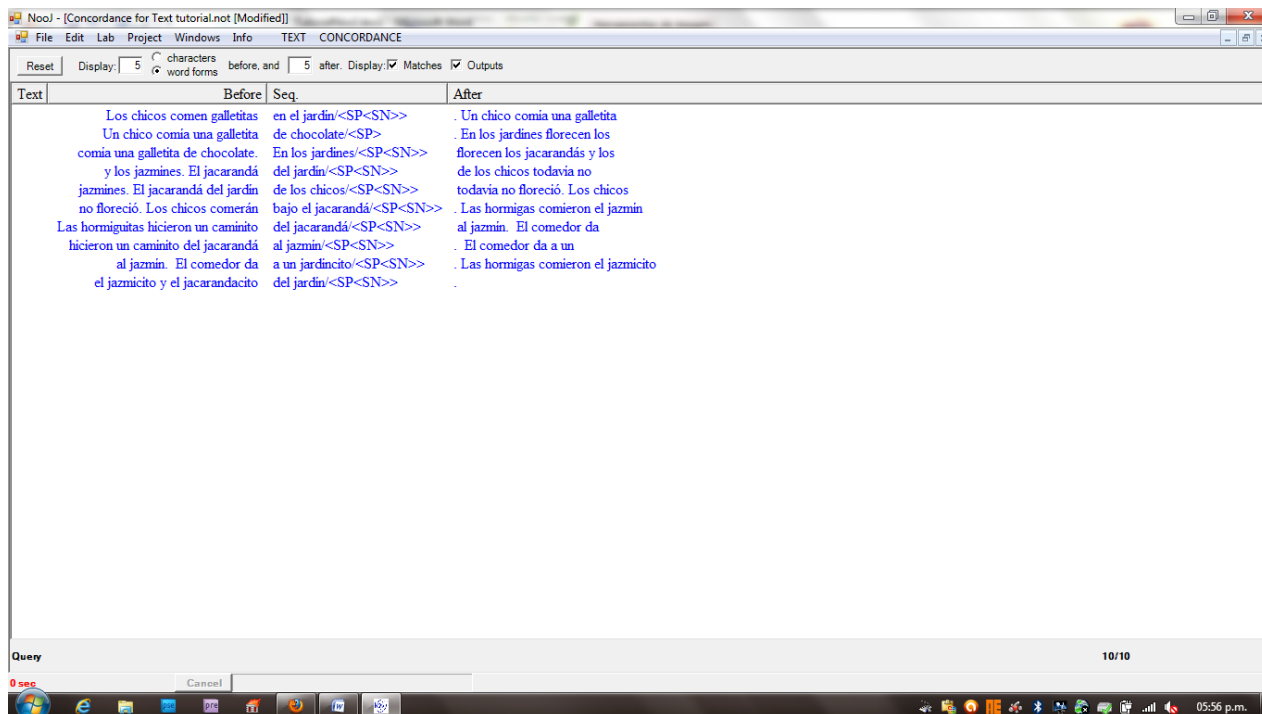
Para lograrlo, hacemos clic en *File > New > Grammar*, seleccionamos *sp* en el idioma de entrada y de salida, *graphical editor*, *Syntax*, ubicamos el estado inicial y final, creamos los nodo *<PREP>/<SP>*, *:SN* y *<E>/>*, y los unimos de izquierda a derecha. Al unirlos, *:SN* se resalta automáticamente dando la posibilidad de crear un gráfico subyacente. Hacemos clic en *Show Structure* y a la izquierda aparece un nuevo panel con una ramificación *SN (X)*, hacemos clic sobre esa ramificación y aparecerá superpuesto el siguiente cartel:



Hacemos clic en *Sí* y se abre la pantalla de un nuevo gráfico, que recibe automáticamente la denominación *SN*. Allí ubicamos el estado inicial y el estado final, creamos los nodos correspondientes al *SN*, los unimos de izquierda a derecha y guardamos la gramática con el nombre *SP1*.

Luego abrimos el texto, en el panel derecho de la pestaña *Syntactic Analysis*, seleccionamos las gramáticas *SNCAT.nog* y *SP.nog*, las eliminamos utilizando el botón *X*, pasamos *SP1.nog* con *>* y volvemos a analizar el texto (*TEXT>Linguistic Analysis*).

Como podemos comprobar, el análisis que produce *SP1.nog* en la parte inferior de la pantalla es igual al que produce *SP.nog*. Sin embargo, en la consulta *TEXT > Locate > a NooJ grammar* los resultados son diferentes:



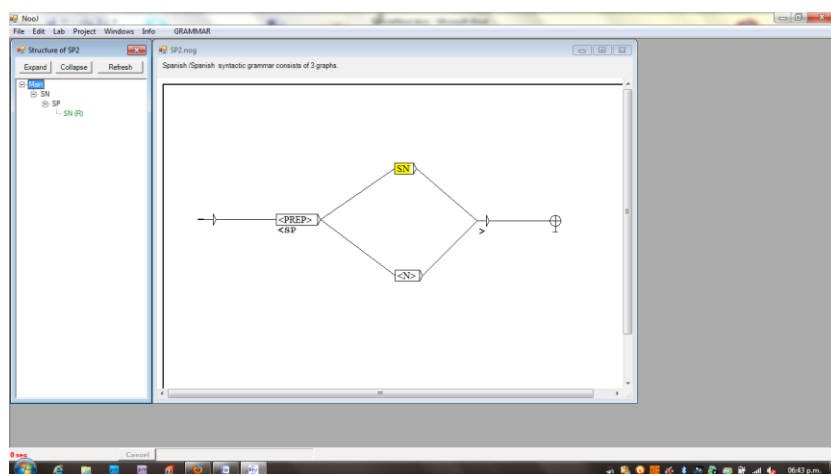
Como se puede ver, acá, el SN es un formante del SP. Esto sucede porque en el primer caso SN no está incluido en SP.nog, por lo tanto, para esta gramática es equivalente a una categoría léxica.

2.4. Recursividad de las gramáticas sintácticas con gráficos subyacentes

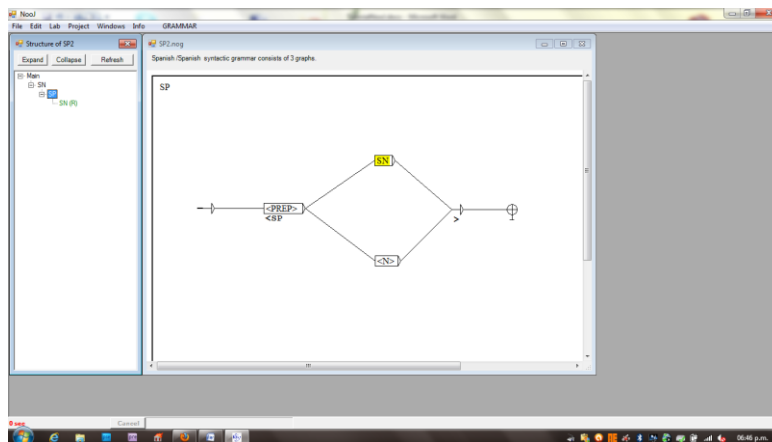
Las gramáticas con gráficos subyacentes, además de distinguir categorías léxicas de categorías sintácticas como constituyentes de una determinada cadena, son recursivas; es decir, permiten utilizar los elementos definidos en nuevas definiciones.

Por ejemplo, en la secuencia *el jacarandá del jardín de los chicos*, la gramática anterior reconoce los dos SP subrayados como independientes; sin embargo, *de los chicos* es un constituyente del SN nominal *el jardín*, que, a su vez, constituye el SP principal. Para analizar este tipo de secuencias se requiere una gramática con gráficos subyacentes, como la que se muestra a continuación:

En la rama principal (*Main*), el SP se define como $\langle PREP \rangle + :SN$.

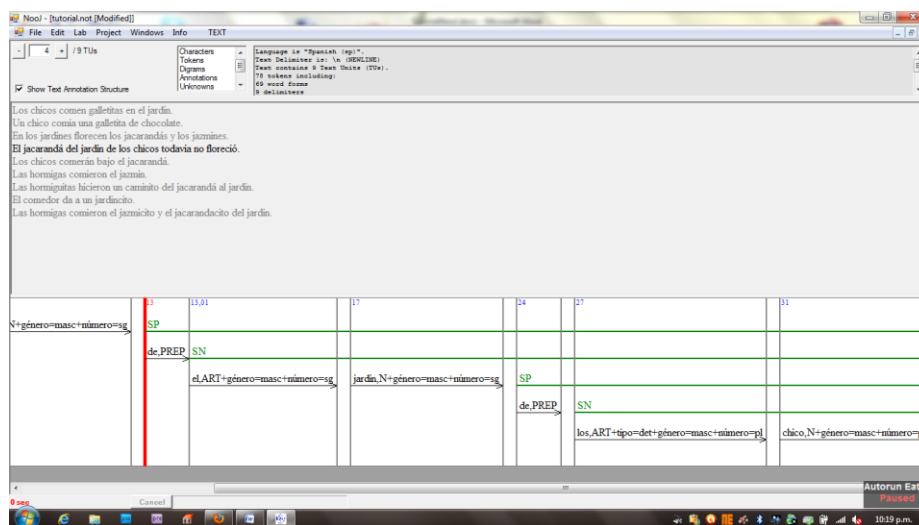


En el primer gráficos subyacente, se define $:SN$ como $\langle ART \rangle + \langle N \rangle$; en la rama superior se introduce un nodo vacío $\langle E \rangle$ y, en la rama inferior, $:SP$; de ese modo, el mismo gráfico tiene dos caminos alternativos: uno para los sintagmas nominales que no tienen SP (*El jardín*) y otro para los que tienen SP (*El jardín de los chicos*).



En el segundo gráfico subyacente se reintroduce el mismo gráfico de *Main* para definir :SP y Nool agrega automáticamente la ramificación **SN(R)**, de este modo, cualquier secuencia que empiece con una preposición puede incrementarse indefinidamente con N o SN más preposición.

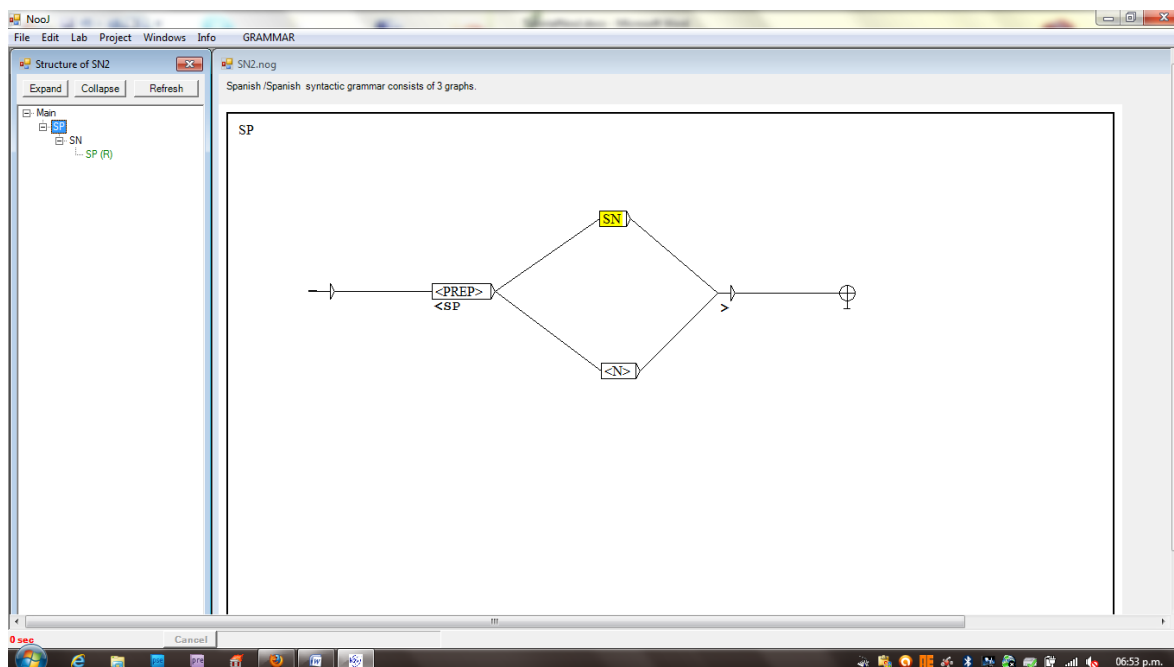
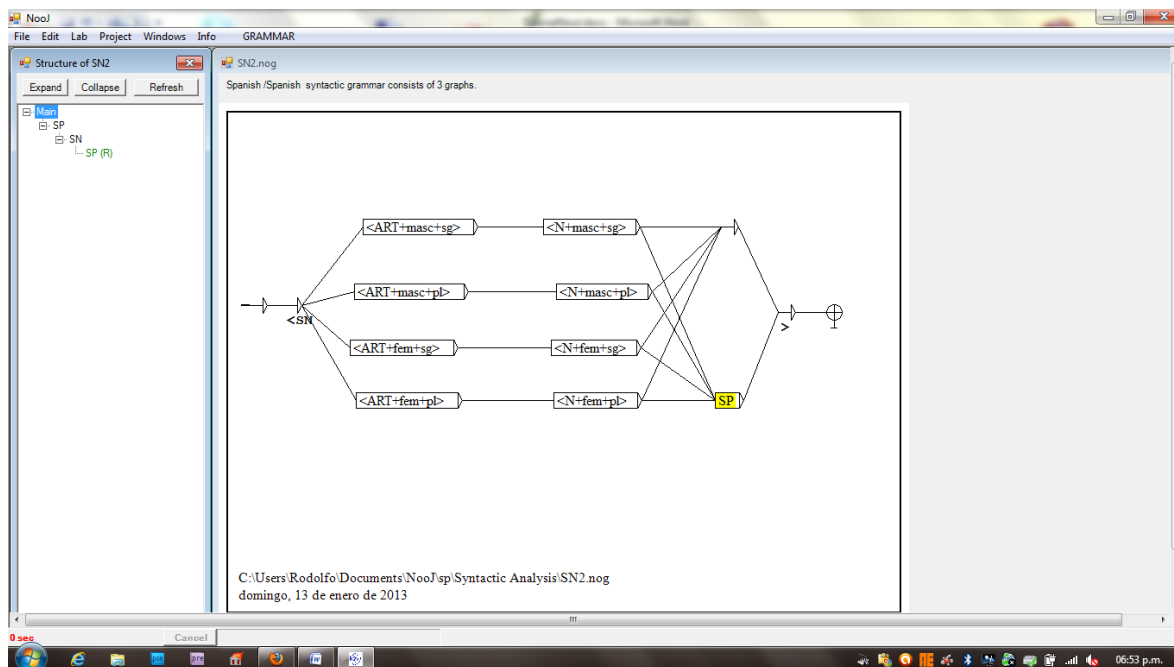
Si aplicamos esta gramática al análisis del texto, obtenemos el siguiente resultado:

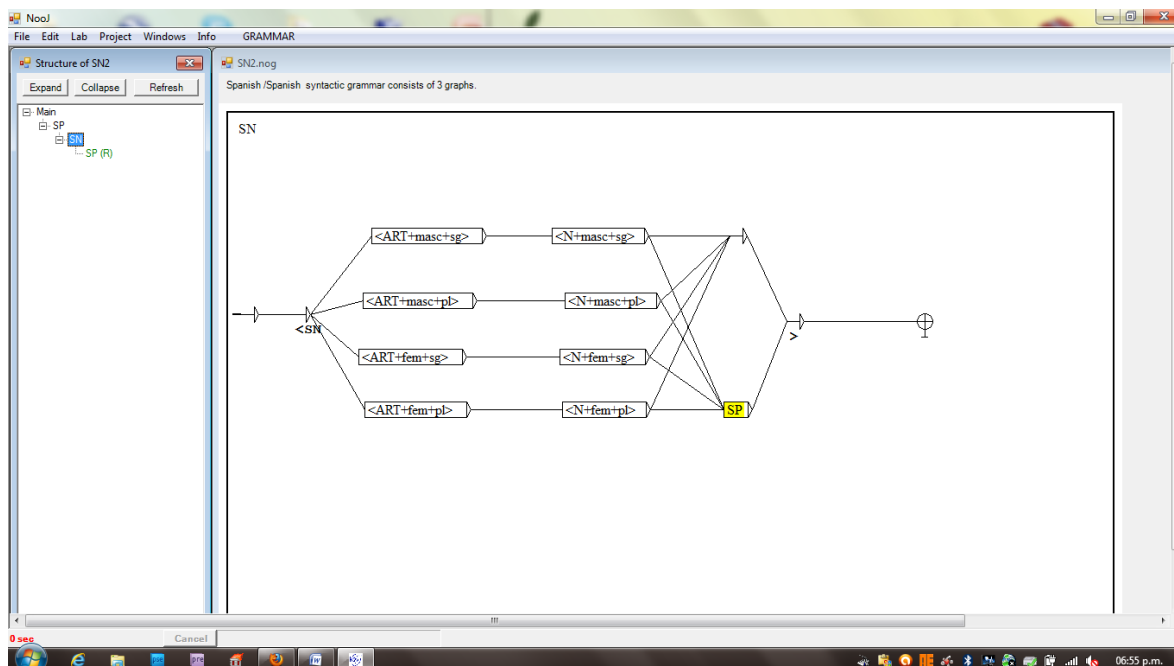


El análisis que produce esta gramática es apropiado para cualquier secuencia de preposición + sustantivo o SN, que pueden o no estar modificados por otro SP; pero no es capaz de reconocer que esa secuencia, a su vez, puede modificar a un SN, como sucede en el ejemplo que estamos tratando, donde *del jardín de los chicos* modifica a *el jacarandá*. En este caso la secuencia empieza con un SN y nuestra gramática indica que la secuencia debe empezar con preposición.

Nool - [Concordance for Text tutorial.not]			
TEXT CONCORDANCE			
Reset	Display: 5 characters before, and 5 after. Display: <input checked="" type="checkbox"/> Matches <input checked="" type="checkbox"/> Outputs		
Text	Before	Seq.	After
Los chicos comen galletitas		en el jardín<SP<SN>>	Un chico comía una galletita
Un chico comía una galletita		de chocolate<SP>	En los jardines florecen los
comía una galletita de chocolate.		En los jardines<SP<SN>>	florece los jacarandás y los
y los jazmines. El jacarandá		del jardín de los chicos<SP<SN>>>>	todavía no floreció. Los chicos
no floreció. Los chicos comerán		bajo el jacarandá<SP<SN>>	Las hormigas comieron el jazmín
Los chicos comieron un caminito		del jacarandá al jazmín<SP<SN>>>>	El comedor da a un
al jazmín. El comedor da		a un jardincito<SP<SN>>	Las hormigas comieron el jazmín
el jazmín y el jacarandá del		del jardín<SP<SN>>	.

Para obtener el análisis de esta secuencia se deberá invertir el orden de los gráficos, como se hace en la gramática *SN2.nog*, donde el gráfico principal (*Main*) es el SN:





En **TEXT > Locate > a NooJ grammar**, obtenemos las siguientes salidas:

Text	Before	Seq.	After
Los chicos comen galletitas en el jardín.	Los chicos/<SN>		comen galletitas en el jardín
el jardín. Un chico comía una galletita de chocolate.	el jardín/<SN>		. Un chico comía una galletita comía una galletita de chocolate
En los jardines florecen los jacarandás y los jazmines.	una galletita de chocolate/<SN<SP>>		. En los jardines florecen los florecen los jacarandás y los y los jazmines. El jacarandá
El jacarandá del jardín de los chicos.	los jardines/<SN>		. El jacarandá del jardín de todavía no floreció. Los chicos comerán bajo el jacarandá. Las
Las hormigas comieron el jazmín.	los jacarandás/<SN>		. Las hormigas comieron el jazmín comieron el jazmín. Las hormiguitas
El comedor da a un jardincito.	los jazmines/<SN>		. Las hormiguitas hicieron un caminito hicieron un caminito del jacarandá
Las hormigas comieron el jazmín.	El jacarandá del jardín de los chicos/<SN<SP<SN<SP<SN>>>>		. El comedor da a un da a un jardincito. Las
El comedor da a un jardincito.	Los chicos/<SN>		. Las hormigas comieron el jazmín comieron el jazmín y el
Las hormigas comieron el jazmín.	el jacarandá/<SN>		
El comedor da a un jardincito.	Las hormigas/<SN>		
Las hormigas comieron el jazmín.	el jazmín/<SN>		
El comedor da a un jardincito.	Las hormiguitas/<SN>		
Las hormigas comieron el jazmín.	un caminito del jacarandá al jazmín/<SN<SP<SN<SP<SN>>>>		
El comedor da a un jardincito.	El comedor/<SN>		
Las hormigas comieron el jazmín.	un jardincito/<SN>		
El comedor da a un jardincito.	Las hormigas/<SN>		
Las hormigas comieron el jazmín.	el jacarandacito del jardín/<SN<SP<SN>>>		

El ejemplo que tomamos como modelo para la gramática *SN2.nog* es el caso de un SN (*el jacarandá del jardín de los chicos*), que contiene un SP (*del jardín de los chicos*), que contiene un SN (*jardín de los chicos*), que contiene un SP (*de los chicos*), que contiene un SN (*los chicos*); pero en un ejemplo como *Las hormigas hicieron un caminito del jacarandá al jazmín* hay tres sintagmas independientes *un caminito* (objeto directo), *del jacarandá* (locativo de origen) y *al jazmín*

(locativo de destino); como puede verse en las salidas, en este caso, el análisis que produce la gramática no es adecuado.

Otra deficiencia que muestra esta gramática es que no reconoce sintagmas preposicionales que no estén precedidos por un SN como sucede en el ejemplo *En los jardines, crecen...*; una alternativa para lograr un análisis más completo sería aplicar simultáneamente las gramáticas *SP2.nog* y *SN2.nog*; pero, es necesario comprobar su eficiencia en un corpus más amplio, ya que la recursividad es un fenómeno que ha sido profusamente estudiado por la lingüística formal, pero resulta muy difícil de tratar automáticamente; por lo tanto, este aspecto exige un tratamiento mucho más exhaustivo.

Ejercitación

- 1) Elabore una gramática productiva para abreviaturas como *aprox.*, *avda.*, *sr.*, *Bs. As.*, *Cía.*, *EE. UU.*, *FF. AA.*, *R.I.P.* (recuerde que una gramática puede tener más de una ramificación).
- 2) Elabore una gramática para siglas como *ONU*, *FMI*, *UNAM*, *ONG*
- 3) Verifique las gramáticas introduciendo ejemplos positivos y negativos en *GRAMMAR> debug...*
- 4) Elabore una gramática sintáctica para oraciones transitivas e intransitivas (*Las hormigas hicieron un caminito. Los chicos juegan*)
- 5) Agregue al texto diferentes oraciones con estas estructuras.
- 6) Agregue al diccionario los nuevos lemas utilizados (no olvide compilarlo).
- 7) Analice el texto con estas gramáticas.
- 8) Agregue a la gramática un camino que incluya a las oraciones transitivas que tienen una oración subordinada como objeto directo (*Juan dijo que Pedro no podía venir*)
- 9) Agregue al texto oraciones que contengan estas estructuras.
- 10) Actualice el diccionario.
- 11) Vuelva a analizar el texto.