# 7. Análisis Sintáctico

# 7 Análisis sintáctico

- 7.1 Analizadores sintagmáticos
- 7.2 Gramática libre de contexto
- 7.3 Gramática de dependencias
- 7.4 Técnicas de análisis sintáctico

# Análisis Sintáctico

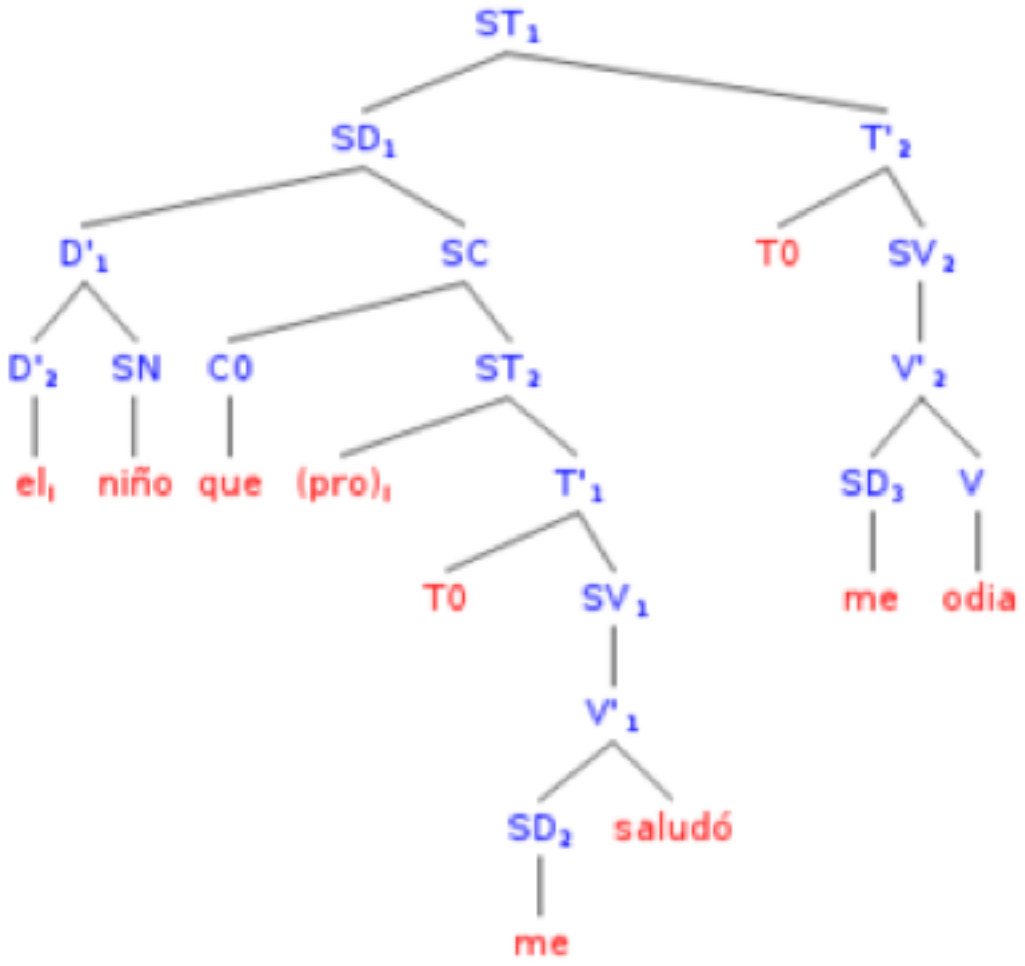
¿Qué es la sintaxis?

La sintaxis es la rama de la lingüística que se encarga de analizar el orden y la relación existente entre las palabras que conforman una oración. Dentro del PLN tiene como objetivo detectar la corrección de una frase o visualizar la estructura de las relaciones entre los constituyentes sintácticos, lo cual sirve para desarrollar diversas tareas y realizar otro tipo de análisis.

# Análisis Sintáctico

Con el análisis sintáctico podemos conocer:

- unidades mínimas de la oración (palabras)
- tipo de palabras (categoría gramatical)
- y cómo se combinan las palabras para formar oraciones y textos




# Análisis sintáctico

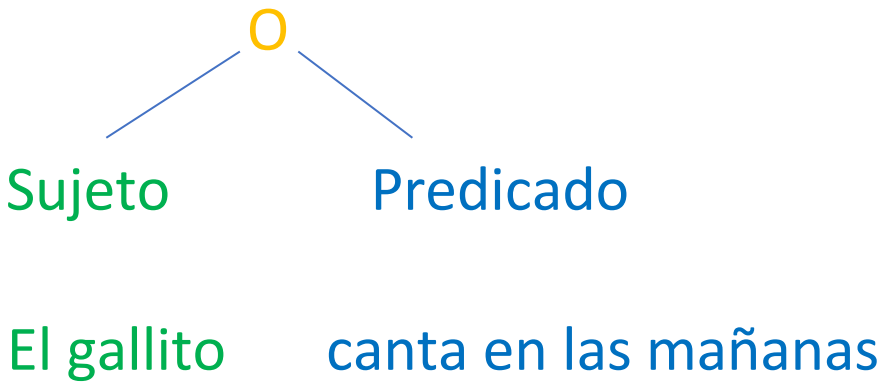
La estructura de la lengua natural no consiste en las palabras como eslabones que forman una cadena (aunque así suela abordarse en términos computacionales), sino que las palabras están combinadas para formar frases que se unirán a otras para así formar oraciones cada vez más grandes y complejas, como las ramas de un árbol.

Para establecer el sentido de una oración es necesario no sólo saber el significado de cada palabra, sino también la relación jerárquica (sintáctica) que existe entre las palabras que forman la oración (O'Grady, 1996).

# Análisis sintáctico

La oración en términos generales podría definirse como una construcción compleja de constituyentes sintácticos que forman un significado. Los constituyentes sintácticos son menores a la oración, pero mayores a la palabra y son muy variados. Los constituyentes básicos de la oración son el sujeto y el predicado.

El sujeto es aquel sobre el que se predica, es decir sobre el que se dice algo, y el predicado es lo que se predica sobre el sujeto, lo que se dice sobre el sujeto. Por ejemplo:



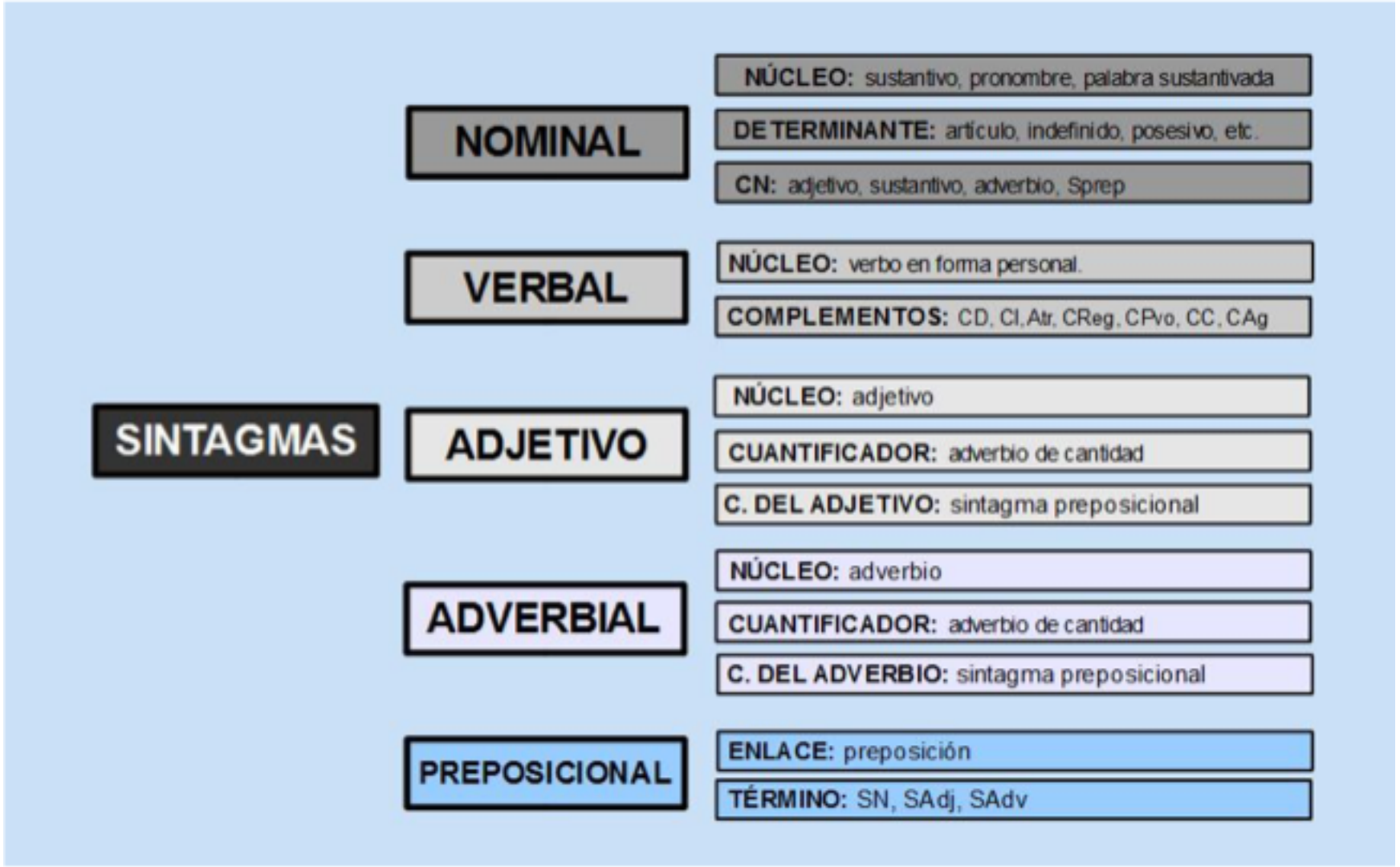
# Análisis sintáctico

Aunque como sabes también existen otras f**unciones sintácticas u oracionales:**

- Suj sujeto
- NV / NP núcleo verbal o núcleo del predicado (verbal o nominal)
- CD complemento directo
- CI complemento indirecto
- CC complemento circunstancial
- Atrib atributo
- Cpredicativo complemento predicativo
- Cagente complemento agente
- Crégimen complemento de régimen verbal, complemento regido, complemento preposicional o suplemento

# Análisis sintáctico

Por otro lado los constituyentes también pueden ser sintagmas, o también llamados frases, que son grupos de elementos léxicos que conforman sub-constituyentes sintácticos. Estos se clasifican como se muestra la imagen, y se nombran según la palabra que rija la construcción.



# Análisis sintáctico

**Tipos de sintagmas (abreviaturas)**

- SN sintagma nominal
- SV sintagma verbal
- Sprep sintagma preposicional
- Sadj sintagma adjetival
- Sadv sintagma adverbial

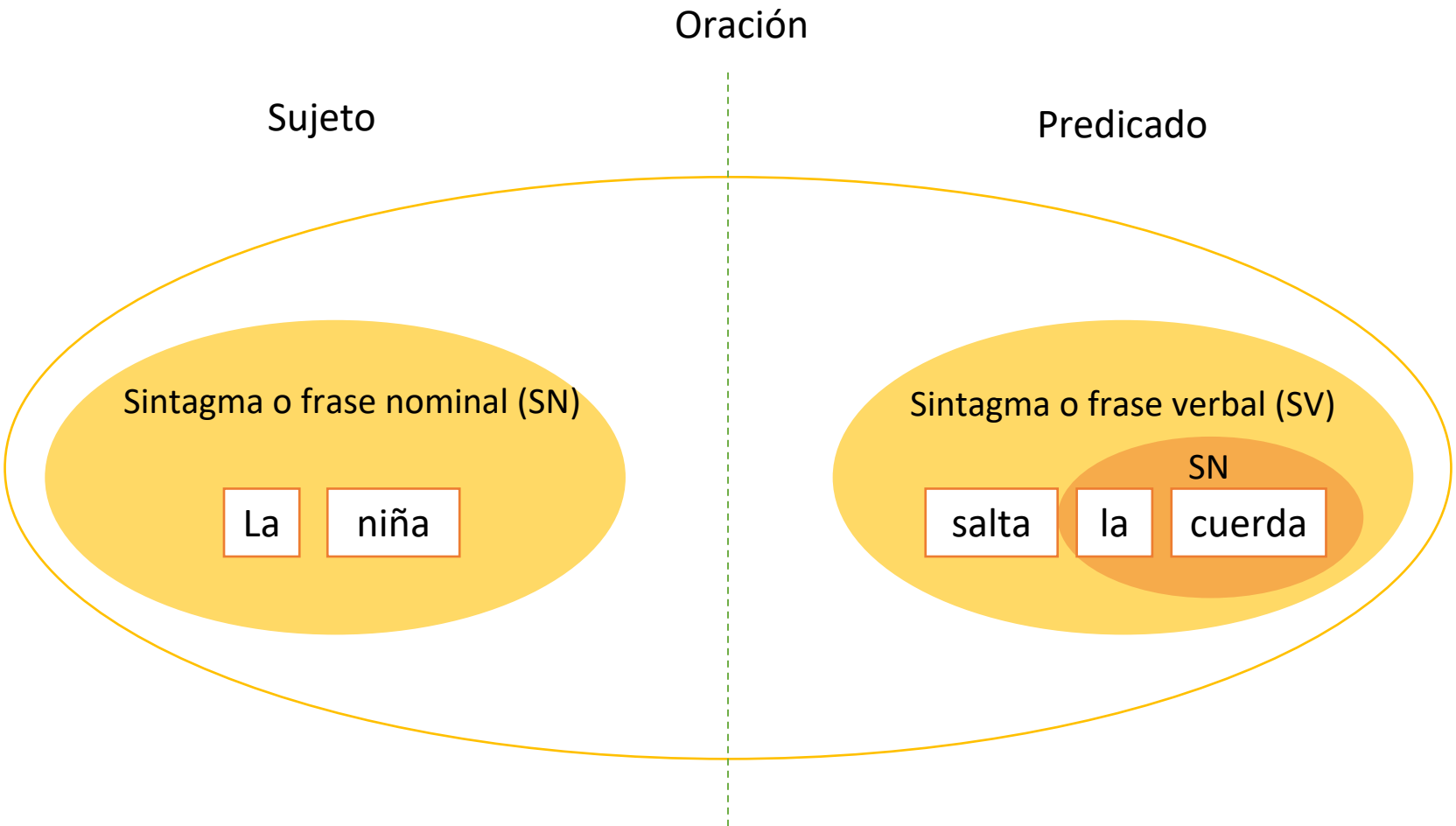
# Análisis sintáctico

Dentro del sintagma cada palabra cumple a su vez una **función sintagmática** (abreviaturas):

- N núcleo
- Det Determinante
- Ady Adyacente
- E Enlace
- CN complemento del nombre
- Cadj complemento del adjetivo
- Cadv complemento de adverbio
- intens o cuant intensificador o cuantificador

# Análisis sintáctico

Ejemplo:



# Análisis sintáctico

Ejemplo:

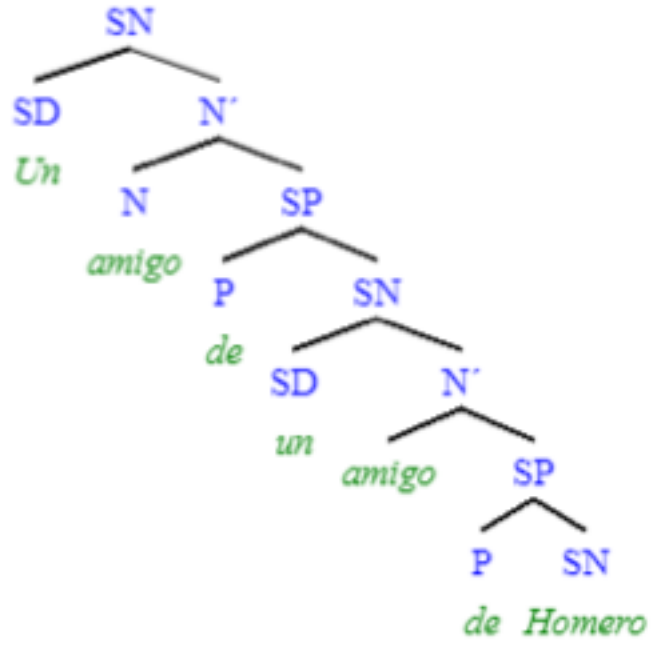
El sintagma preposicional “de tu tía” está compuesto por una preposición y un sintagma nominal.

Sintagma preposicional = de + tu tía

SP = Preposición + SN

# Análisis sintáctico

La estructura sintáctica permite la recursividad, esto quiere decir, como ya vimos en los ejemplos anteriores, que se puede incluir una estructura dentro de otra (o un constituyente sintáctico dentro de otro), respetando una jerarquía sintáctica.

# Análisis sintáctico

Ahora bien, mediante un modelo de aprendizaje basado en gramáticas formales se puede hacer la jerarquización de los elementos que constituyen la oración. A continuación recordaremos un poco sobre gramáticas libres y dependientes (las cuales ya vimos en la primera sesión).

# Análisis sintáctico

Gramáticas de estructura de frase, o mejor conocida como libre de contexto, definen de manera formal la lengua natural por medio de sus reglas, las cuales pueden construirse con ayuda de los sintagmas. Estas gramáticas son útiles para representar las relaciones sintácticas complejas de una oración y tienen las restricciones necesarias para desarrollar algoritmos eficaces, entre ellas están GPSG, HPSG, LFG, PATR-II, DCG, etc.

# Análisis sintáctico

Ejemplo:

Si quisiéramos analizar las siguientes oraciones una gramática libre de contexto quedaría así:

a) La niña salta la cuerda

b) La cuerda salta la niña

Q= {Art, NN, V, SN,SV,O}

Σ= {la, niña, salta, cuerda}

R= {

O→ SN+SV

SN→ Art+NN

SV→ V+SN

Art→ *la*

NN→ niña

NN→ cuerda

V→ salta

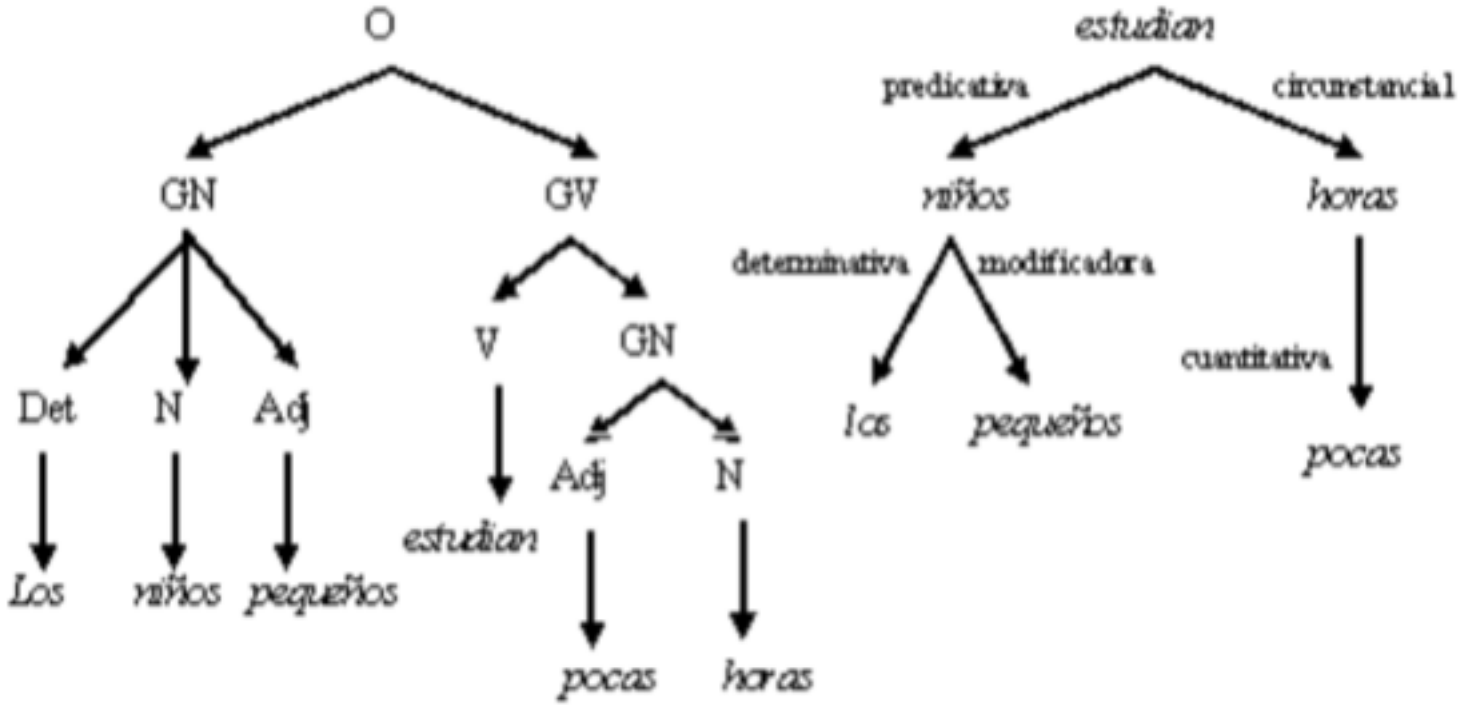
}

Art = artículo
NN= sustantivo o nombre
V= verbo
SN= sintagma nominal
SV= sintagma verbal
O= oración

# Análisis sintáctico

Otro tipo de gramática con la que se puede hacer análisis sintáctico es con las gramáticas de dependencia (igual revisadas en la sesión 1) o también llamadas gramáticas de cláusulas definidas, ilustran la estructura recursiva de las oraciones y aplican el formalismo de la unificación. Expresan esquemas de dependencias del contexto, por lo que incluyen condiciones adicionales en las reglas de la gramática y pueden aumentar los símbolos no terminales con nuevos argumentos.

# Análisis sintáctico




A) Árbol de constituyentes

B) Árbol de dependencias

# Análisis sintáctico

Estas gramáticas son utilizadas por un analizador sintáctico o *parser,* el cual es básicamente un programa informático que analiza una cadena de símbolos de acuerdo a las reglas de una gramática formal.

La finalidad del análisis sintáctico es asignar una estructura a la entrada. Existen dos tipos de análisis:

- Profundo: determina si una oración es correcta con relación a las reglas sintácticas establecidas.
- Parcial: representa determinados constituyentes sintácticos y no la oración completa. Este análisis sirve para localizar sintagmas y es beneficioso en tareas de RI y QA.

# Análisis sintáctico

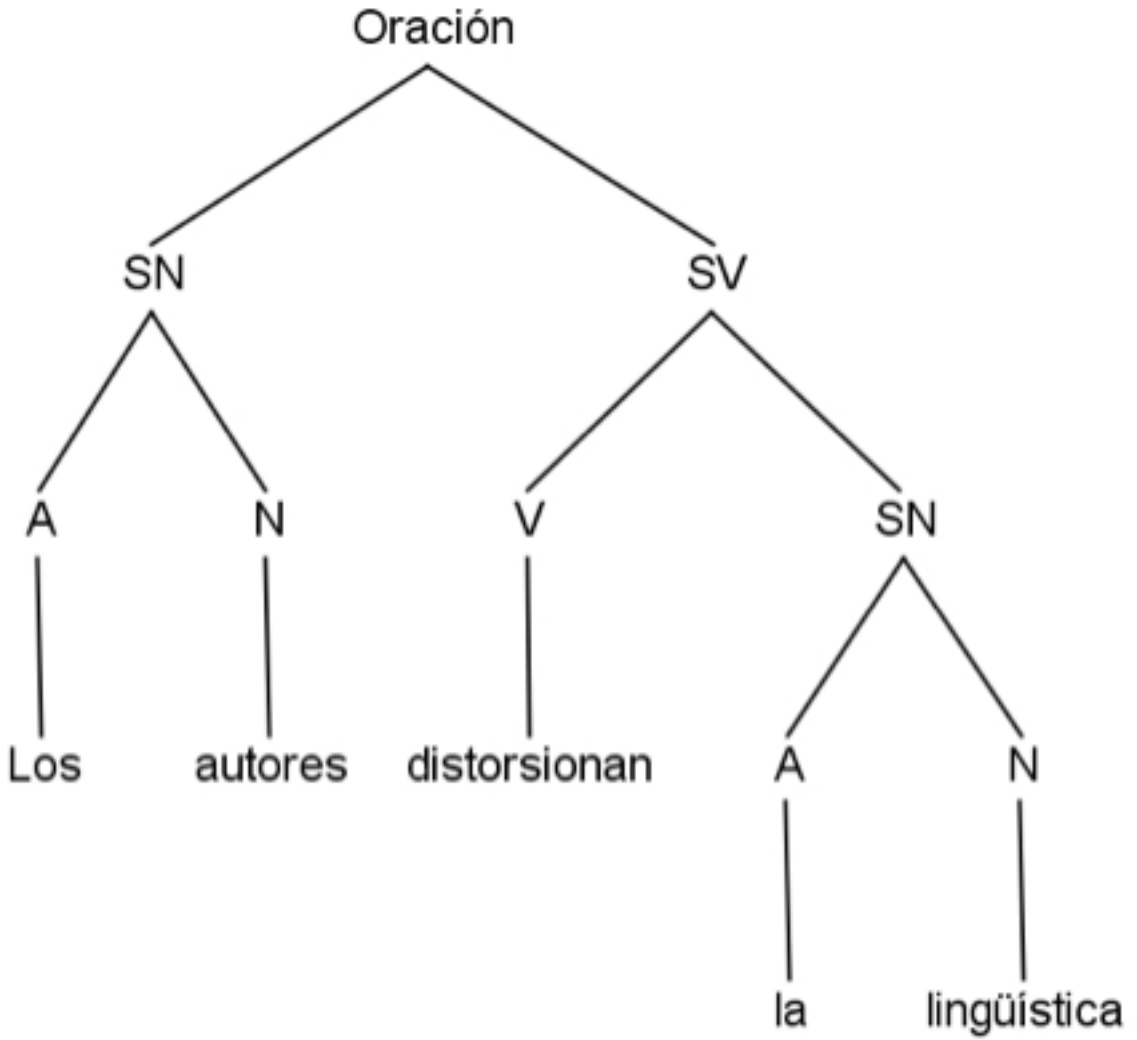
Existen diferentes estrategias de *parsing*,  aquí hablaremos de dos:

- Analizador descendente (Top-down): Lo que hace el analizador es dividir la entrada en partes más pequeñas sucesivamente, el árbol se construye hacia abajo y utiliza más reglas, analiza en profundidad y de izquierda a derecha, puede acabar re-expandiendo muchas veces un símbolo dado, comenzando en la misma palabra si ese símbolo aparece en  contextos diferentes.
- Analizador ascendente (Bottom-up): Determina la construcción de los constituyentes complejos hasta completar la oración, el árbol es de abajo hacia arriba y disminuye el número de reglas mal aplicadas, sin embargo, no puede manejar la ‘recursividad a izquierdas’.

# Análisis sintáctico

## Top-Down

Empieza en O




## Bottom-Up

Termina en O

# Análisis sintáctico

Para ver algunos ejemplos detallados ingresa a los siguientes links:

- Ejemplo de analizador sintáctico descendente (Top-down)

https://www.infor.uva.es/~teodoro/EJEMPLO-TASD.pdf

- Ejemplo de analizador sintáctico ascendente (Bottom-up)

http://www-lt.ls.fi.upm.es/procesadores/Documentos/AStLR.pdf

# Análisis sintáctico

Parseadores automáticos disponibles:

- NLTK

http://www.nltk.org/

“Es una plataforma líder para construir programas de Python para trabajar con datos de lenguaje humano. Proporciona interfaces fáciles de usar a más de 50 recursos corporales y léxicos como WordNet, junto con un conjunto de bibliotecas de procesamiento de texto para clasificación, tokenización, derivación, etiquetado, análisis y razonamiento semántico, envoltorios para bibliotecas de PNL de fuerza industrial, y un foro de discusión activo.”

# Análisis sintáctico

- Freeling

http://nlp.lsi.upc.edu/freeling/node/1

“FreeLing es una biblioteca en C ++ que proporciona funcionalidades de análisis de lenguaje (análisis morfológico, detección de entidad nombrada, etiquetado PoS, análisis sintáctico, desambiguación de sentido de palabra, etiquetado de función semántica, etc.) para una variedad de idiomas (inglés, español, portugués, italiano, francés, Alemán, ruso, catalán, gallego, croata, esloveno, entre otros).”

# Análisis sintáctico

- Stanford CoreNLP

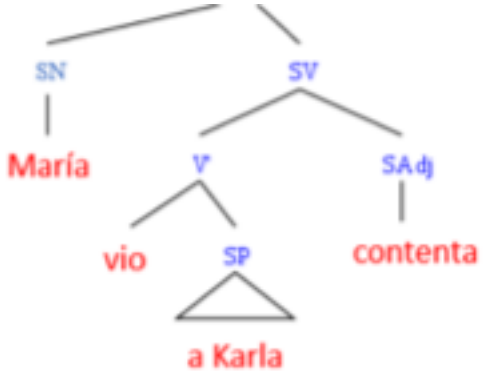
https://stanfordnlp.github.io/CoreNLP/

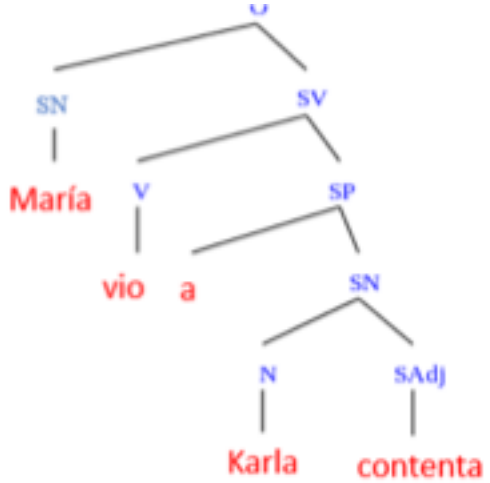
“Stanford CoreNLP provides a set of human language technology tools. It can give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and syntactic dependencies, indicate which noun phrases refer to the same entities, indicate sentiment, extract particular or open-class relations between entity mentions, get the quotes people said, etc.”

# Análisis sintáctico

**Ambigüedad sintáctica**

La ambigüedad estructural o sintáctica es uno de los grandes retos que enfrenta el análisis sintáctico, esta se refiere a la doble representación en árboles sintácticos de una misma expresión. Por ejemplo:

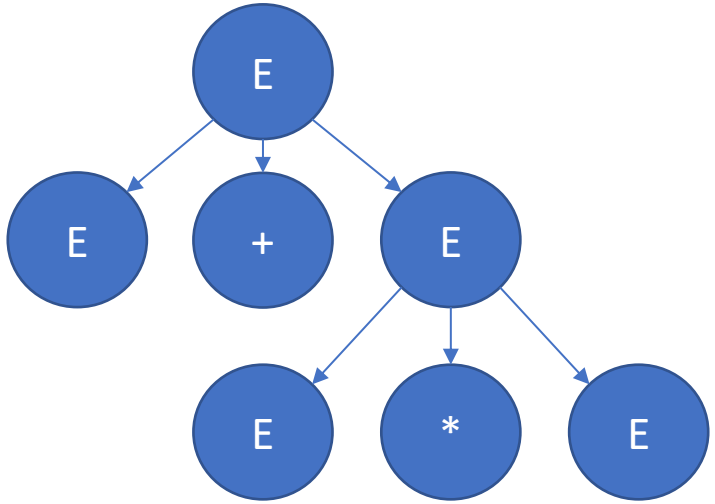


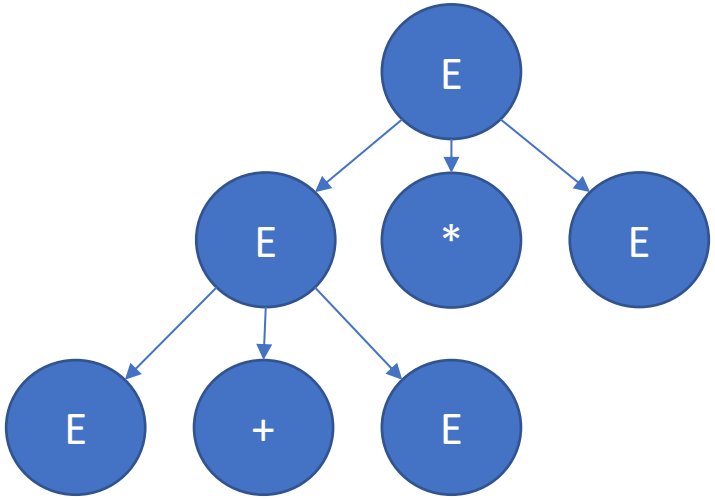




En el primer árbol la contenta es María, mientras que en el segundo es Karla.

# Análisis sintáctico

Si una gramática genera más de un árbol sintáctico a partir de la misma raíz y con más de una estructura se logra generar la misma cadena, dicha gramática es ambigua.



# Análisis sintáctico

- Si una gramática es ambigua, posiblemente, pero no necesariamente, existe una gramática **no ambigua** que pueda generar el mismo lenguaje.

- No existe algún algoritmo que pueda definir si una gramática es o no ambigua.

# Análisis sintáctico

Ejemplo:

- Sea G una gramática de contexto libre se tiene:
- G = {Q,Σ,$q_0$,R}
  donde
  - Q = {E}
  - Σ = {+,*,(,),1,…,9}
  - $q_0$ = {E}
  - R = {E→E+E | E→E*E | (E) | 1 | … |9|}

- Una expresión ambigua:
  - E + E * E

- Se pueden hacer dos derivaciones:
  - E → E + E → E + E * E
  - E → E * E → E + E * E

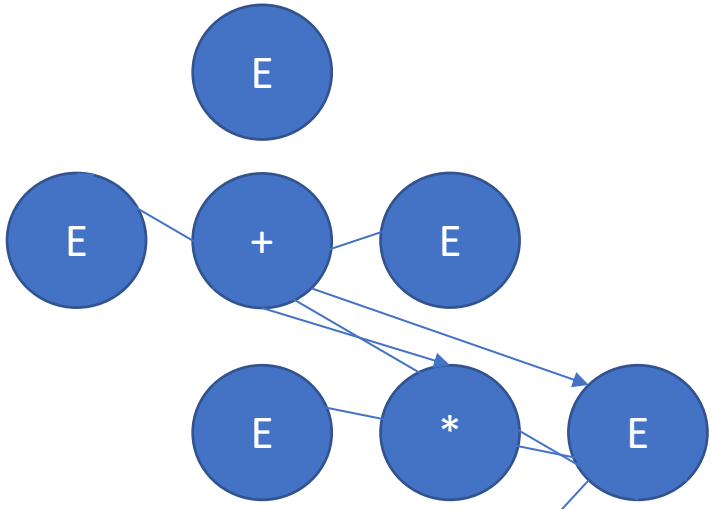
# Análisis sintáctico

- Se observa que la expresión final es la misma:
  - E → E + E * E
  - E → E + E * E

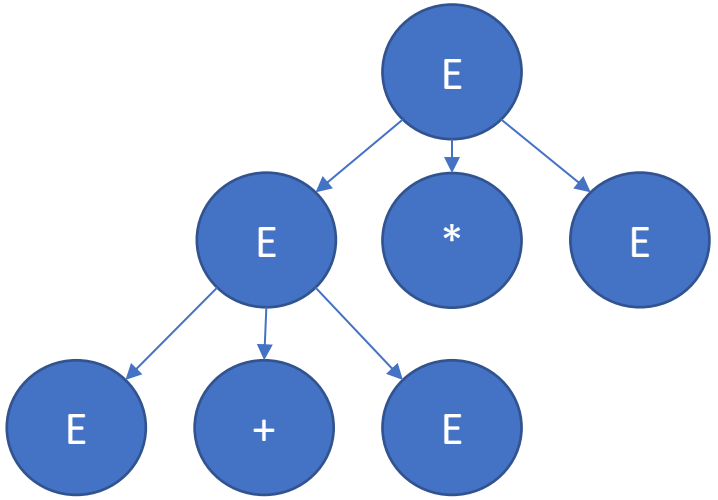
- Pero en sus derivaciones son diferentes:
  - E → E + <u>E</u> → E + <u>E * E</u>
  - E → <u>E</u> * E → <u>E + E</u> * E

# Análisis sintáctico

- A cada derivación le corresponde un árbol sintáctico:
  - Derivaciones diferentes crean la misma cadena.
- Los árboles sintácticos son los siguientes:




E → E + <u>E</u> → E + <u>E * E</u>

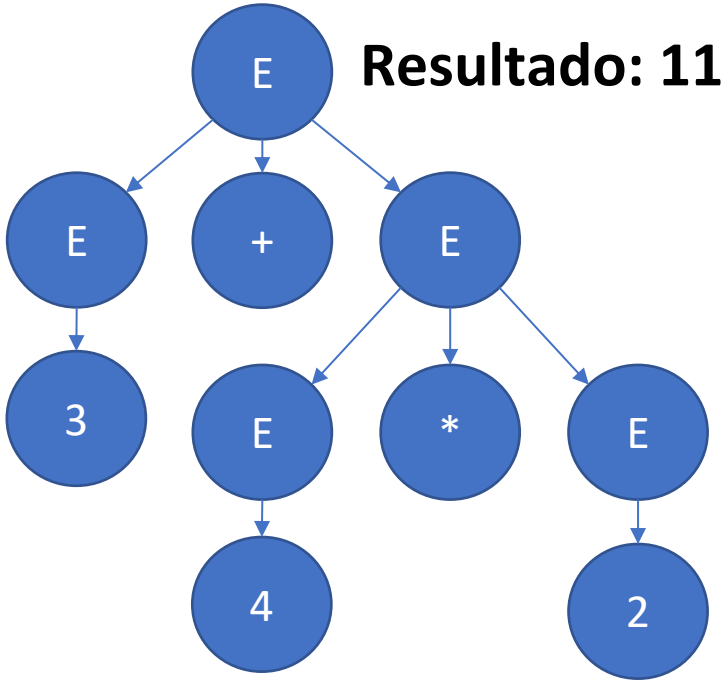



E → <u>E</u> * E → <u>E + E</u> * E

# Análisis sintáctico

¿En qué afecta tener ambigüedad?

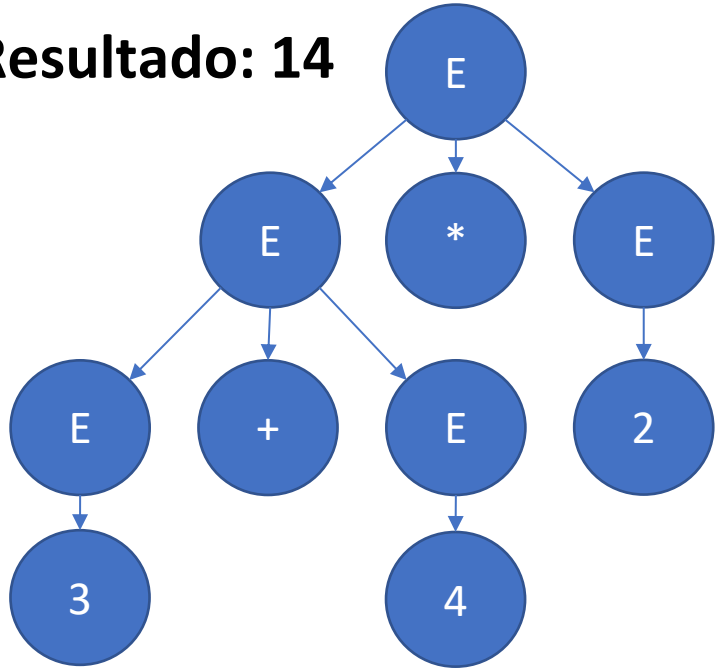
- Sustituyendo valores tenemos:




**Resultado: 11**

E → E + <u>E</u> → E + <u>E * E →</u> 3 + 4 * 2

**Resultado: 14**




E → <u>E</u> * E → <u>E + E</u> * E → 3 + 4 * 2

# Análisis sintáctico

- La ambigüedad surge cuando hay más de un árbol sintáctico para una misma expresión o cadena.

- Sea una GCL G, G = $(Q,\Sigma,q_0,R)$ es ambigua si existe al menos una cadena w en $\Sigma^*$ para la cual hay más de un árbol de parseo con raíz $q_0$.

- En general, no existe un algoritmo para eliminar la ambigüedad.

# Análisis sintáctico

- Eliminar la ambigüedad:
  - En la práctica, por ejemplo en la definición de la GCL para lenguajes de programación, es posible eliminar la ambigüedad.
- Por ejemplo:
  - Si G es una GCL ambigua, tal que L=L(G) y existe una $G_n$ no ambigua tal que L=L(G), se puede eliminar la ambigüedad remplazando G por $G_n$ .
  - En una GCL ambigua es posible elegir uno de los árboles que generan la gramática con base en criterios matemáticos como la precedencia de operadores.

# Análisis Sintáctico

Penn Treebank

Por último, hablemos de Penn Treebank, también conocido como corpus parseado, es un corpus en el que las frases han sido anotadas con su estructura sintáctica, suele emplearse el etiquetado gramatical. Estos corpus sirven para el aprendizaje de gramáticas y analizadores de texto.

# Alphabetical list of part-of-speech tags used in the Penn Treebank Project:

| Number | Tag | Description |
|---|---|---|
| 1. | CC | Coordinating conjunction |
| 2. | CD | Cardinal number |
| 3. | DT | Determiner |
| 4. | EX | Existential *there* |
| 5. | FW | Foreign word |
| 6. | IN | Preposition or subordinating conjunction |
| 7. | JJ | Adjective |
| 8. | JJR | Adjective, comparative |
| 9. | JJS | Adjective, superlative |
| 10. | LS | List item marker |
| 11. | MD | Modal |
| 12. | NN | Noun, singular or mass |
| 13. | NNS | Noun, plural |
| 14. | NNP | Proper noun, singular |
| 15. | NNPS | Proper noun, plural |
| 16. | PDT | Predeterminer |
| 17. | POS | Possessive ending |
| 18. | PRP | Personal pronoun |
| 19. | PRP$ | Possessive pronoun |
| 20. | RB | Adverb |
| 21. | RBR | Adverb, comparative |
| 22. | RBS | Adverb, superlative |
| 23. | RP | Particle |
| 24. | SYM | Symbol |
| 25. | TO | *to* |
| 26. | UH | Interjection |
| 27. | VB | Verb, base form |
| 28. | VBD | Verb, past tense |
| 29. | VBG | Verb, gerund or present participle |
| 30. | VBN | Verb, past participle |
| 31. | VBP | Verb, non-3rd person singular present |
| 32. | VBZ | Verb, 3rd person singular present |
| 33. | WDT | Wh-determiner |
| 34. | WP | Wh-pronoun |
| 35. | WP$ | Possessive wh-pronoun |
| 36. | WRB | Wh-adverb |

https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

# Bibliografía

Haro, S. N. G., & Gelbukh, A. (2007). *Investigaciones en análisis sintáctico para el español*. Instituto Politécnico Nacional, Dirección de Publicaciones.

Jurafsky, Daniel & James H. Martin. (2008). Speech and Language Processing: *An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition* (2nd edition), Prentice Hall.

O'Grady, W., Dobrovolsky, M., & Katamba, F. (Eds.). (1997). *Contemporary linguistics*. St. Martin's.