PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

FACULTAD DE LETRAS

Introducción a la lingüística computacional

César Antonio Aguilar

Facultad de Lenguas y Letras

17/08/2017

Cesar.Aguilar72@gmail.com

Síntesis de la clase pasada (1)

En la clase anterior nos dedicamos a instalar tanto el entorno **Anaconda** como la librería **NLTK**, con miras a comenzar a hacer nuestras tareas de procesamiento de textos.

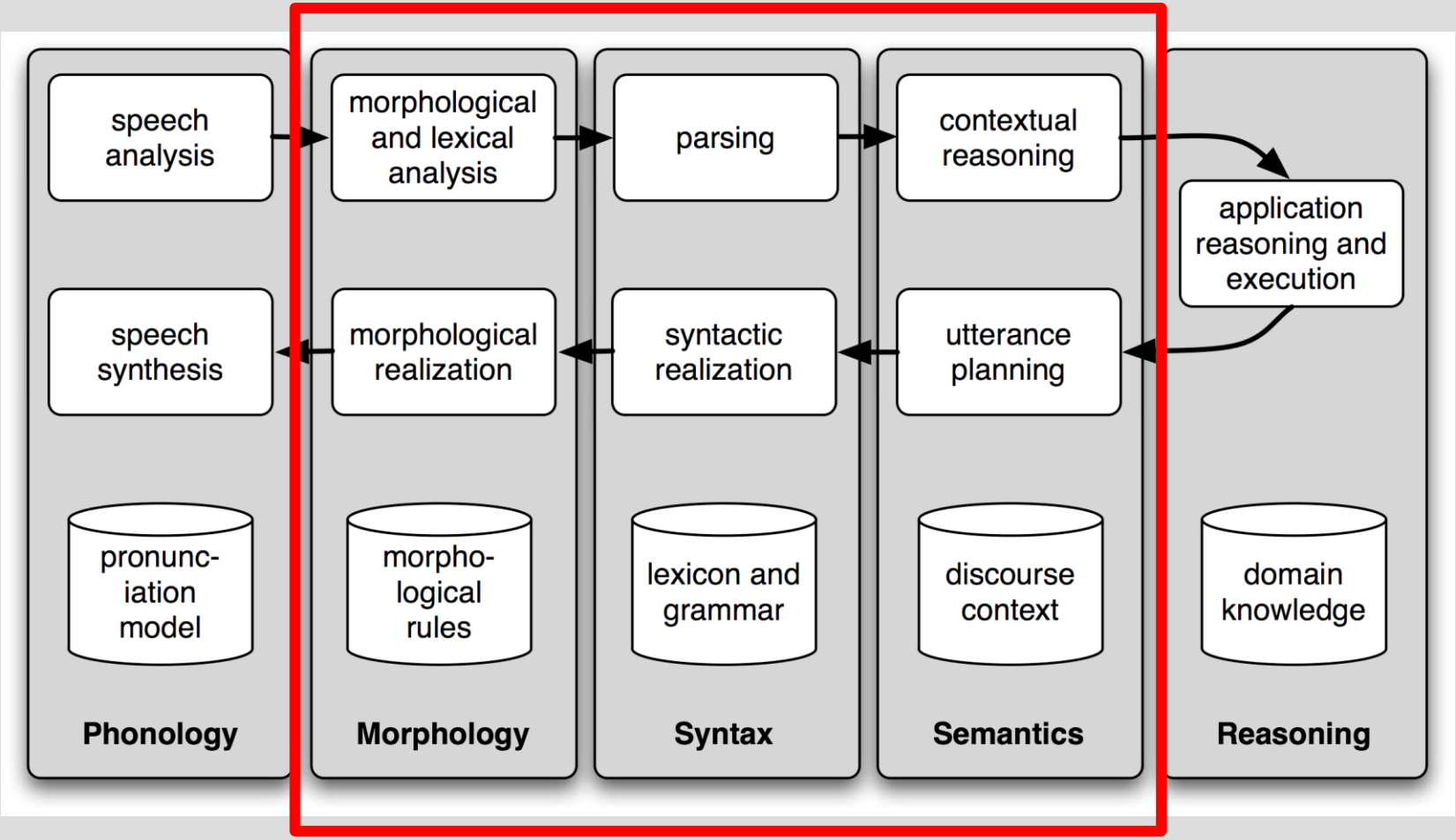
ANACONDA
Powered by Continuum Analytics

Natural Language Toolkit

La idea entonces es: Anaconda es un plataforma o entorno de programación, en tanto que NLTK son una colección de programas que nos permiten hacer tareas de lingüística computacional.

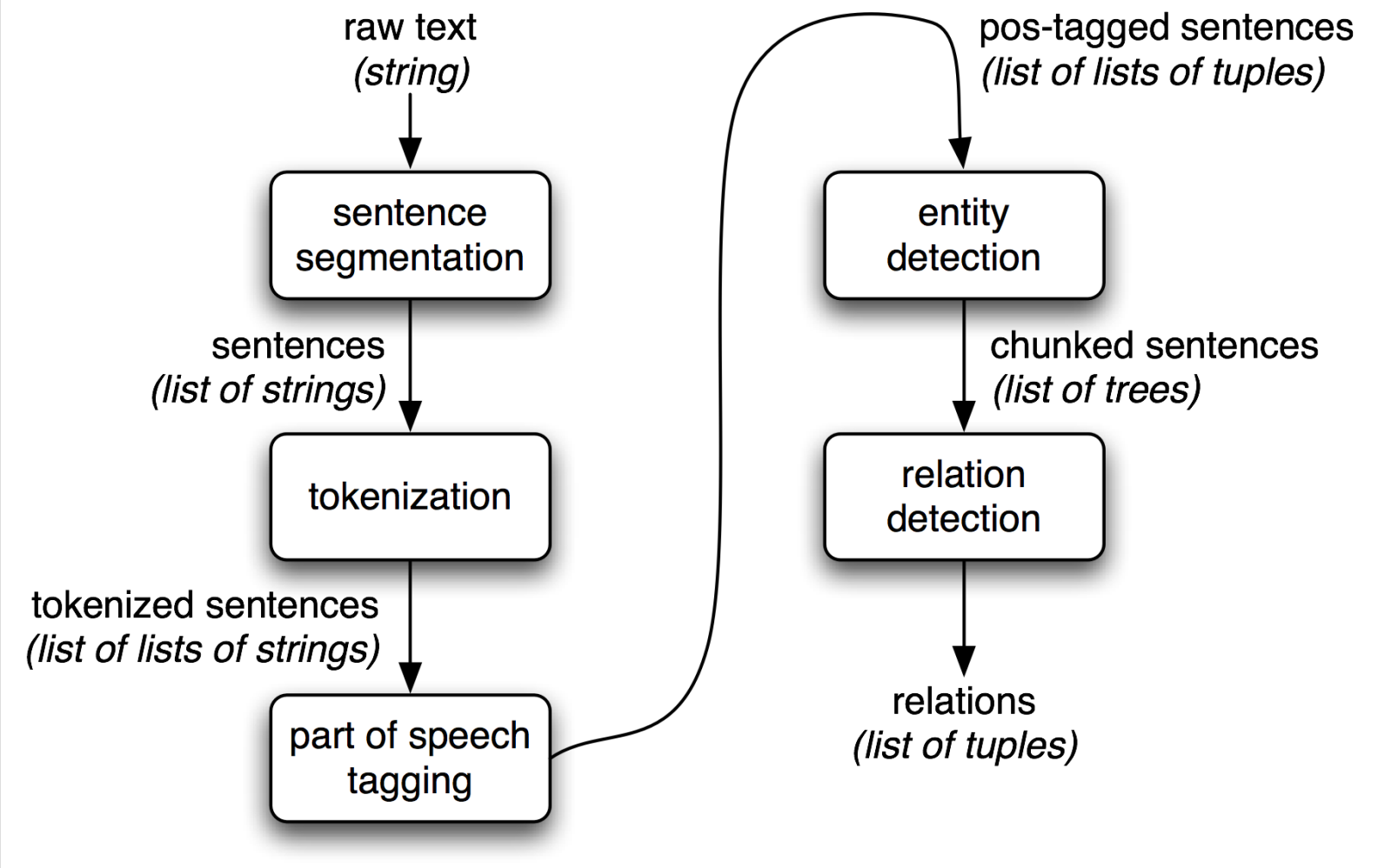
Síntesis de la clase pasada (2)

Como comentamos al inicio del curso, lo que tratamos de establecer es una cadena de procesamiento que, en nuestro caso, tome en cuenta tres niveles del lenguaje natural: morfología, sintaxis y semántica.



Síntesis de la clase pasada (3)

Para lograr lo anterior, nos vamos a concentrar en realizar las siguientes tareas:

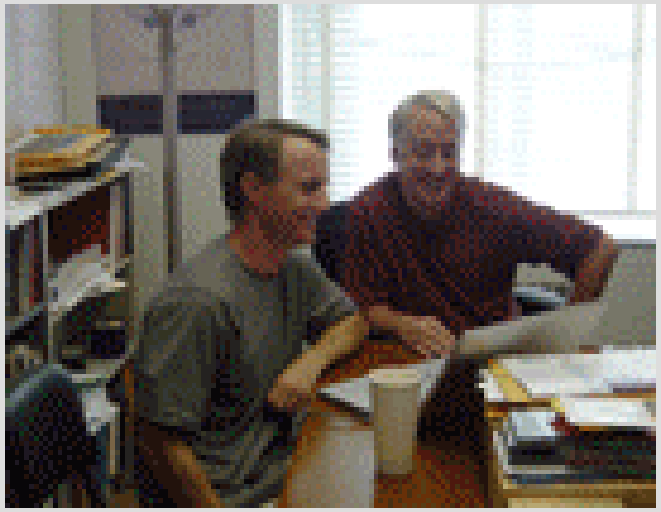


Bases computacionales (1)

Lenguaje natural *versus* lenguaje formal

Lenguaje natural: lenguaje empleado por seres humanos para fines comunicativos. Obedecen, aunque no siempre, a principios de economía y optimidad, los cuales se ven afectados por variables como factores cognitivos, sociales, emocionales, etc.

Lenguaje formal: lenguaje conformado a partir conjuntos finito de caracteres que genera cadenas. Cuentan con reglas de buena formación (gramática) e interpretación semántica (significado). Estos lenguajes es que se ajustan con rigor a su gramática y a su semántica.



Bases computacionales (2)

Ejemplos de lenguajes formales

```
√9874285 | 3142
 9       |-----------
 087     | 61x1=61
  61     | 624x4=2496
  2642   | 6282x2=12564
  2496
  014685
   12564
   02121
```

Matemáticas

Negation Rules

$$\frac{\Delta \vdash B, \Gamma}{\Delta, B^{\perp} \vdash \Gamma}\ (\cdot)^{\perp}\mathrm{L} \qquad \frac{\Delta, B \vdash \Gamma}{\Delta \vdash B^{\perp}, \Gamma}\ (\cdot)^{\perp}\mathrm{R}$$

Lógica

```
>>> x = int(raw_input("Please enter an integer: "))
>>> if x < 0:
...     x = 0
...     print 'Negative changed to zero'
... elif x == 0:
...     print 'Zero'
... elif x == 1:
...     print 'Single'
... else:
...     print 'More'
...
```

Lenguajes de programación

¿Qué es un lenguaje de programación?

Tipo de lenguaje formal que se ocupa para controlar el desempeño de una computadora.

Manejan una sintaxis y una semántica para generar un conjunto infinito de instrucciones para controlar los procesos que ejecute una computadora

Prolog

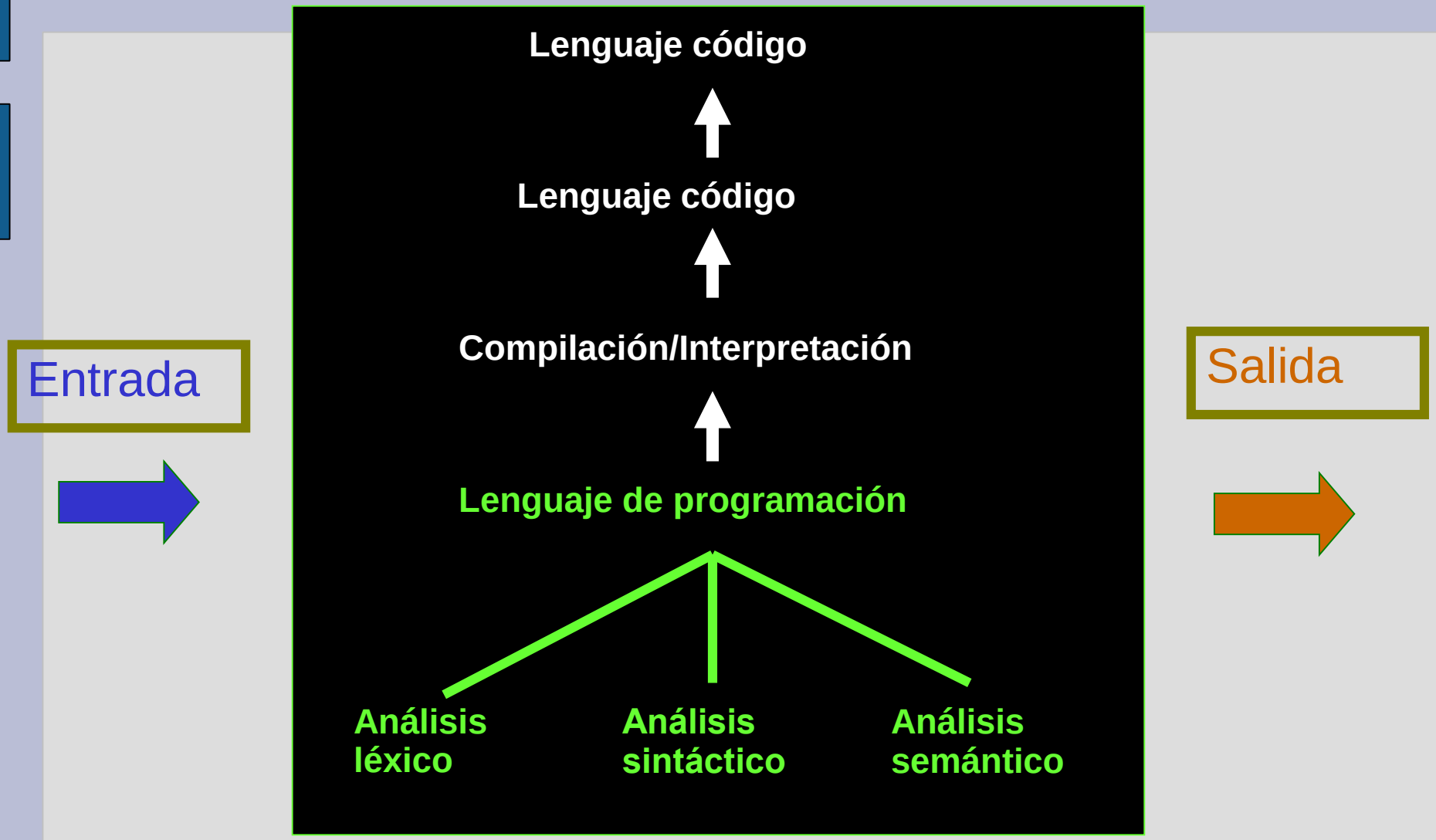
perl

Java

python

¿Cómo funciona?

Caja negra



Veamos el proceso (1)

Lenguaje código: combinaciones binarias de 0/1

```
1 0 1 0 0 1 1 0 1 0
| - | - - | | - | -
x o x o o x x o x o
y n y n n y y n y n
```

Compilación/interpretación: es un proceso por el cual se traduce un programa escrito en un lenguaje a otro, generando un programa equivalente que la máquina será capaz de interpretar. Usualmente el segundo lenguaje es código máquina, pero también puede ser simplemente texto.

Motorola 68000

```
{---------------------------}
procedure Expression;
{ Parse and Translate
  an Expression }
begin
  Term;
  EmitLn('MOVE D0,D1');
  case Look of
   '+': Add;
   '-': Subtract;
   else Expected('Addop');
  end;
end;

{---------------------------}
```

Veamos el proceso (2)

Lenguaje de programación: conjunto de instrucciones codificadas conforme a una sintaxis y a una semántica específicas. Dichas instrucciones asignan una tarea a una computadora, considerando el uso de variables y reglas lógicas.

```
>>> chomsky = ['colorless', 'green', 'ideas', 'sleep', 'furiously']
>>> for word in chomsky:
...     print len(word), word[-1], word
...
9 s colorless
5 n green
5 s ideas
5 p sleep
9 y furiously
```

¿Qué es el análisis léxico?

Proceso que consiste en la lectura de un programa fuente (escrito en un lenguaje de programación), y agrupa en componentes léxicos o **tokens**, esto es, en secuencias de caracteres con significado.

Ejemplos de tokens:

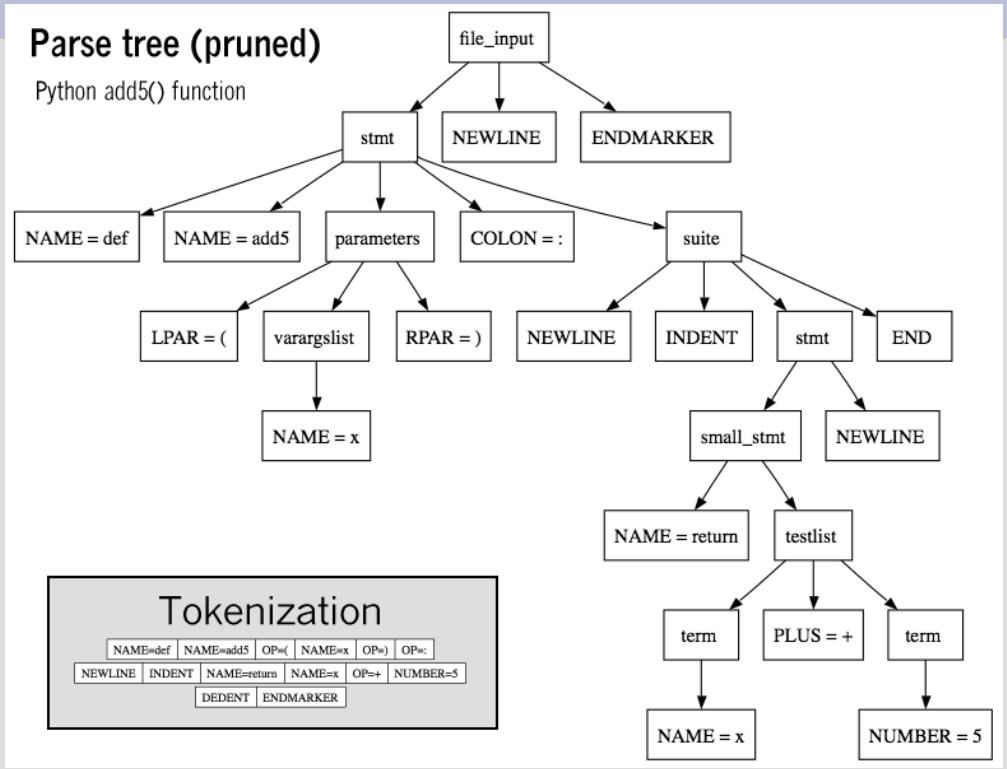
Palabras clave: if, while, in…

Identificadores: Var, String, List…

Números: 0-9.

Signos: α, #, &, *…

Operadores: +, -, ≠, ∑…



¿Qué es el análisis sintáctico?

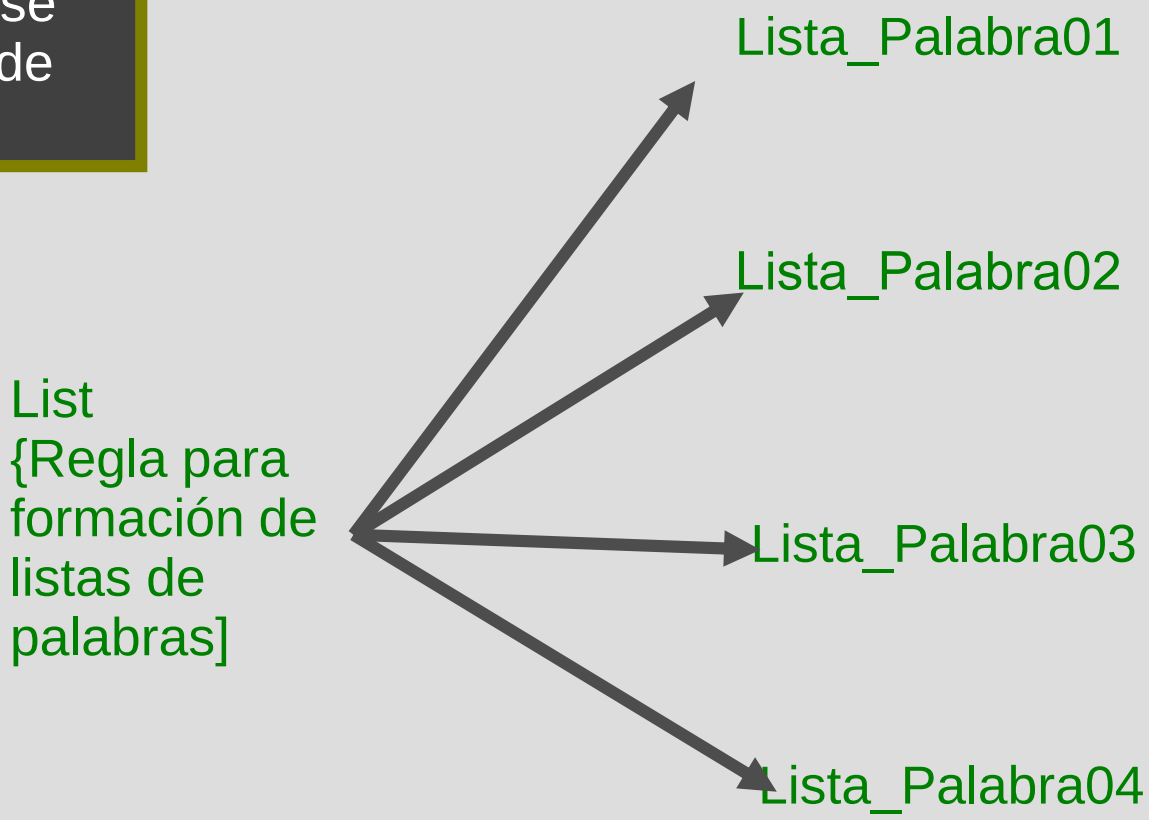
En esta fase los caracteres o componentes léxicos se agrupan jerárquicamente en frases gramaticales. Se comprueba si lo obtenido de la fase anterior es sintácticamente correcto.

```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodename()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print ' %s [label="%s' % (nodename, label),
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s"];' % ast[1]
        else:
            print '"]'
    else:
        print '"];'
        children = []
        for n, child in enumerate(ast[1:]):
            children.append(dotwrite(child))
        print ' %s -> {' % nodename,
        for name in children:
            print '%s' % name,
```

¿Qué es el análisis semántico?

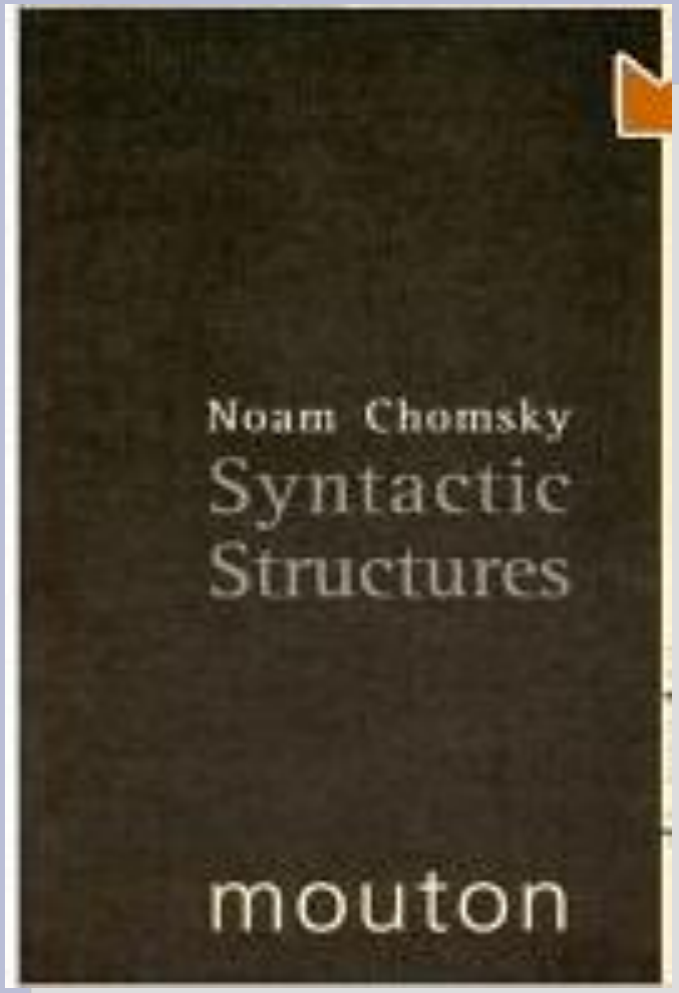
Es la fase donde se revisa el programa para encontrar errores semánticos, reuniendo la información sobre los tipos para la fase posterior de generación de código.



Lenguajes formales (1)

En 1957, un doctor recién graduado en lingüística, Noam Chomsky (1928), escribió *Syntactic Structures*, que era una versión “resumida” de su tesis doctoral. Si tienen curiosidad, la tesis tiene entre 500 y 600 páginas.

Lenguajes formales (2)



Dejando de lado el objeto de estudio, el trabajo se planteó dos preguntas sumamente brillantes:

1. ¿Existe una facultad humana para generar lenguajes?

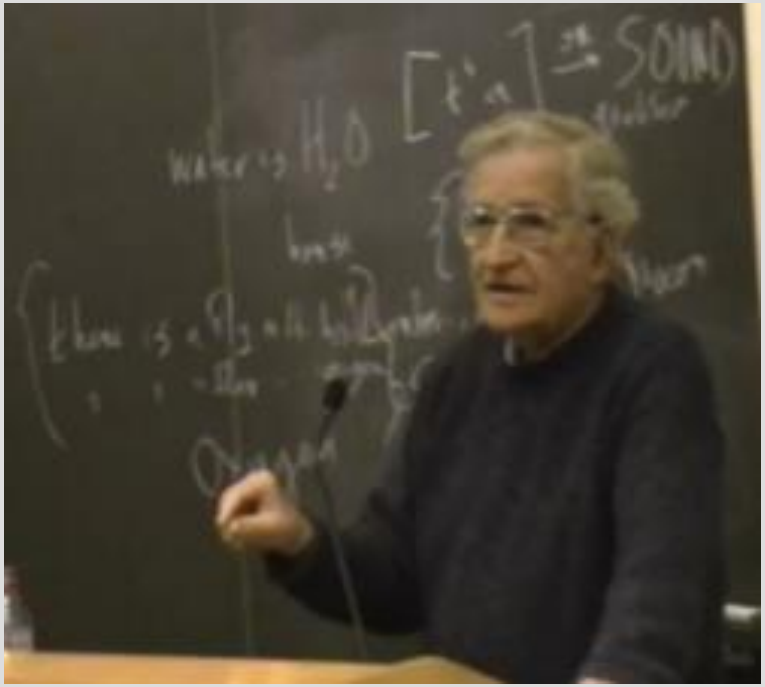
2. ¿Se puede dar una representación formal de dicha facultad?

Lenguajes formales (3)

Para responder a estas preguntas, Chomsky formuló ideas como…

"La sintaxis es el estudio de los principios y procesos en virtud de los cuales son construidas las oraciones en las lenguas particulares".

"La investigación científica de una lengua dada tiene como objetivo la construcción de una gramática, considerada como una especie de mecanismo [o autómata], capaz de generar las oraciones de la lengua que se quiera analizar".



Lenguajes formales (4)

¿Qué clases de gramáticas pueden hacer lo que plantea Chomsky? Según él, por lo menos hay 4, y cada una de estas permite construir un autómata capaz de generar oraciones (o “cadenas sintácticas”):

Tipo	Lenguaje	Autómata	Normas de producción de gramáticas
0	recursivamente enumerable (LRE)	Máquina de Turing (MT)	Sin restricciones
1	dependiente del contexto (LSC)	Autómata linealmente acotado	$\alpha A \beta \rightarrow \alpha \gamma \beta$
2	independiente del contexto (LLC)	Autómata con pila	$A \rightarrow \gamma$
3	regular (RL)	Autómata finito	$A \rightarrow aB$ $A \rightarrow a$

Lenguaje regular (1)

Es un lenguaje que cuenta con un alfabeto finito de caracteres, que sigue una gramática con un conjunto finito de reglas, y genera un número infinito de cadenas regulares.

"α puede ser reemplazado por β si α pertenece a los símbolos **No Terminales** y β es uno de estos 3:

- Un símbolo **Terminal** no nulo seguido de un **No Terminal**.
- Un símbolo **No Terminal** seguido de un símbolo **Terminal** no nulo.
- Un símbolo **Terminal** pudiendo ser la cadena vacía."

$$\alpha \rightarrow \beta$$

$$\alpha \in NT$$

$$\beta \in \begin{cases} aB \\ Ba \\ b \end{cases}$$

$$B \in NT$$

$$a \in T^{+}$$

$$b \in T^{*}$$

Lenguaje regular (2)

Un ejemplo en lenguaje natural:

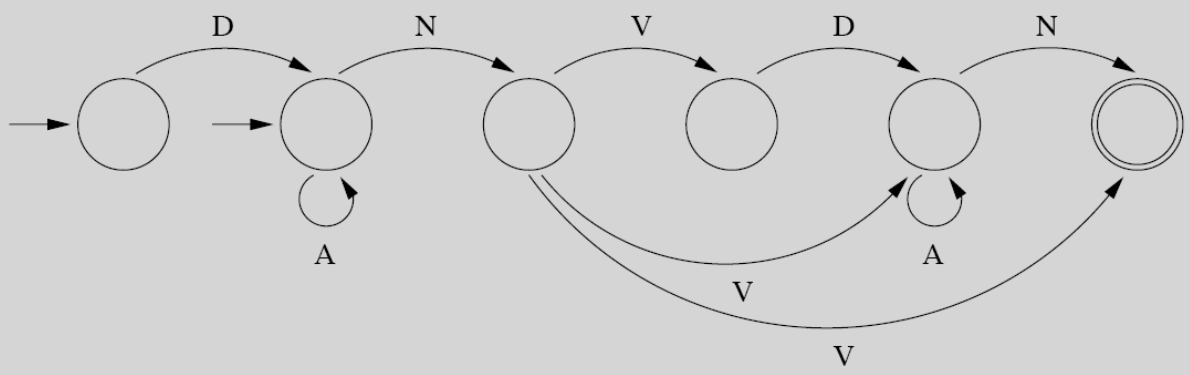
- *the noisy dogs left*

  D A N V
- *the noisy dogs chased the innocent cats*

  D A N V D A N

  dogs chased cats

  the nasty noisy dogs chased the poor little cats
- $a^* = \{\emptyset, a, aa, aaa, aaaa, \ldots\}$
- $a^+ = \{a, aa, aaa, aaaa, \ldots\}$



This is equivalent to (D) A* N V ((D) A* N)

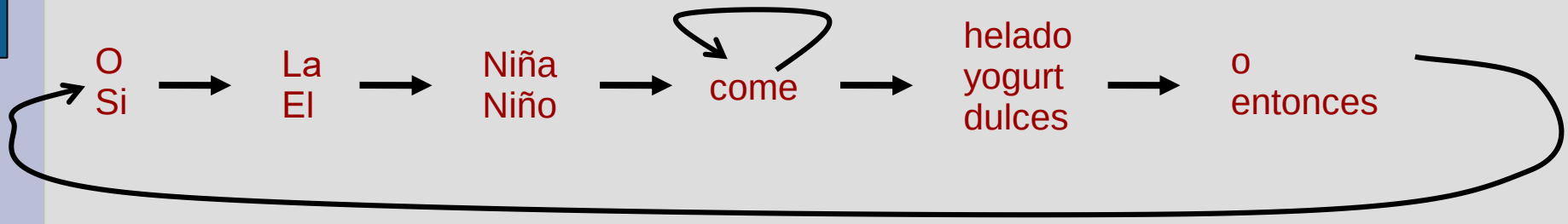
Lenguaje regular (3)

Sin embargo, una gramática regular no es un modelo adecuado para representar el lenguaje humano, porque el autómata trabaja a partir del reconocimiento de patrones regulares, y no propiamente de reglas de generación.

Por ejemplo, según Chomsky, no tenemos modo de generar reglas de producción para evitar producir cadenas sintácticas anómalas.

Lenguaje regular (4)

1. O la niña come helado o come yogurt.
2. Si la niña come helado, entonces el niño come dulces.



El autómata generaría oraciones anómalas como :

1. # O la niña come helado entonces el niño come dulces.
2. # Si la niña come helado o la niña come yogurt
3. (??) O la niña come helado, entonces el niño come dulces, o si la niña come yogurt, el niño come dulces

Lenguajes libres de contexto (1)

Es un lenguaje que cuenta con un alfabeto finito de caracteres, que sigue una gramática con un conjunto finito de reglas, las cuales pueden producir un conjunto infinito de cadenas a partir del uso recursivo de dichas reglas, así como de un proceso de evaluación.

$G = (V_t, V_n, P, S)$ donde

- V_t es un conjunto finito de terminales
- V_n es un conjunto finito de no terminales
- P es un conjunto finito de producciones
- $S \in V_n$ el denominado Símbolo Inicial
- los elementos de P son de la forma

$$V_n \longrightarrow (V_t \cup V_n)^*$$

Lenguajes libres de contexto (2)

- Lexicon: a list of pairs < word, category>
- Set of rules: $C_0 \longrightarrow C_1 \ldots C_n$
- Lexical and nonlexical categories (N, V, S, NP, VP, etc)
- Designated 'initial symbol'

Otro ejemplo en lenguaje natural:

<u>RULES</u>

S $\longrightarrow$ NP VP

NP $\longrightarrow$ (D) A* N PP*

VP $\longrightarrow$ V (NP) (PP)

PP $\longrightarrow$ P NP

<u>LEXICON</u>

D: *the, some*

A: *big, brown, old*

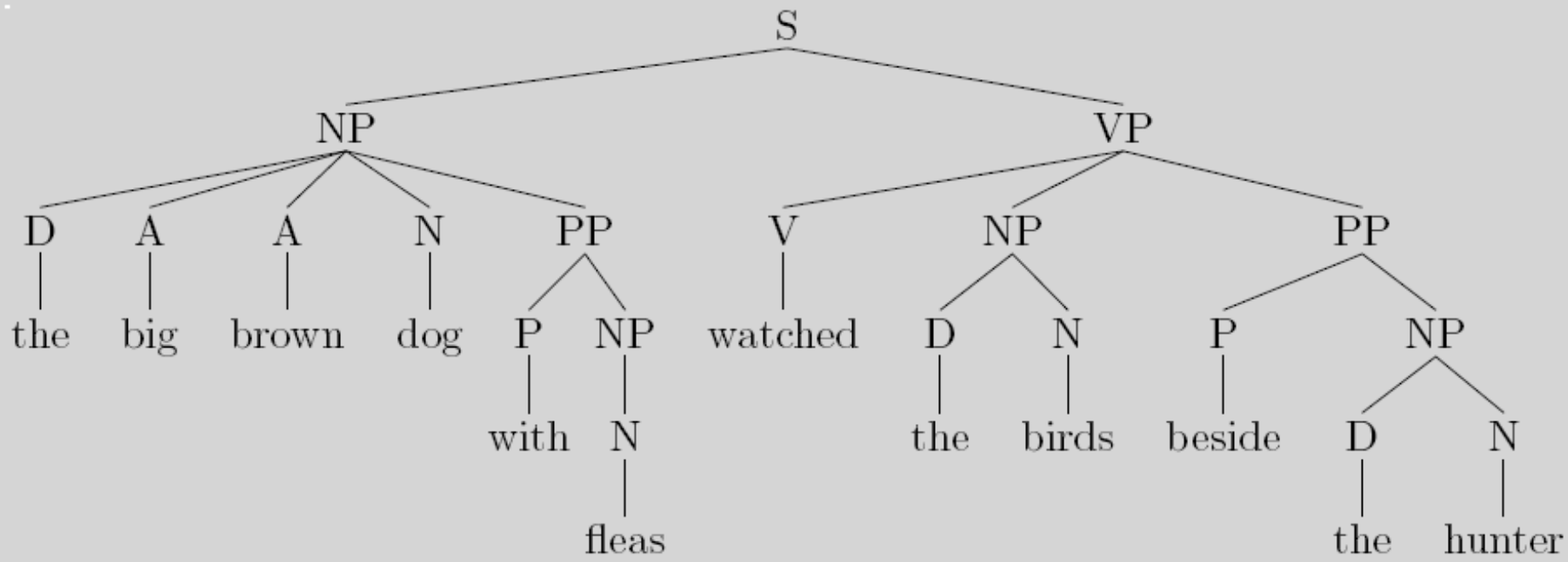
N: *birds, fleas, dog, hunter, I*

V: *attack, ate, watched*

P: *for, beside, with*

Lenguajes libres de contexto (3)

Representación arbórea



Lenguajes libres de contexto (4)

Empero, una gramática de este tipo no me permite resolver cuestiones como, por ejemplo, la concordancia que debe de haber entre artículos, nombres, adjetivos y verbos en dos cadenas sintácticamente similares. Para que mi autómata pueda resolver este problema, debo hacer explícito en mi lexicón todas las opciones posibles.

Lenguajes libres de contexto (5)

S → FN + FV
FN → DET + N
FV → V + FN
DET → {la/el/las/los}
N → {niña/niñas/niño/niños/helado/dulces}
V → {come/comen}

Oración correcta: *el niño come helado*

Oración extraña: *?el niños come helado*

Oración incorrecta: *#la dulces come los niñas*

Gracias por su atención

Blog del curso:
http://cesaraguilar.weebly.com/curso-de-procesamiento-del-lenguaje-natural.html