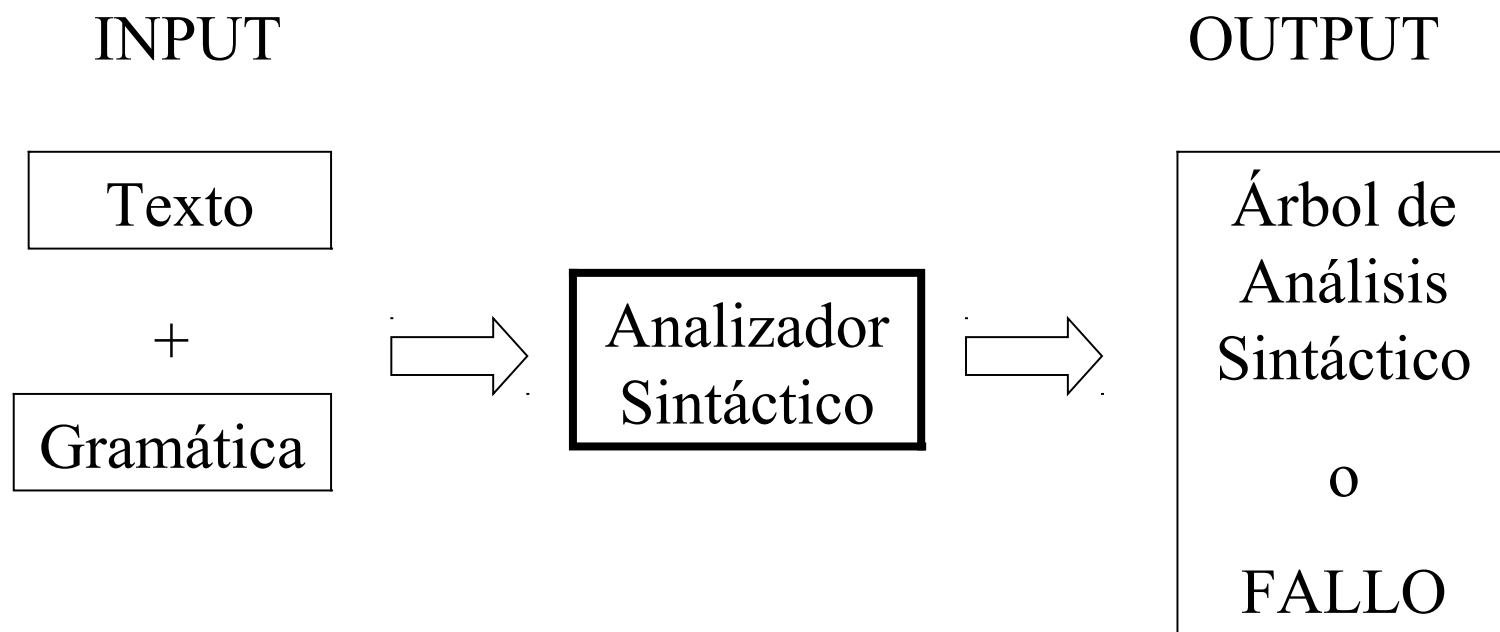


Análisis Sintáctico

Análisis Sintáctico



Esquema general de un analizador sintáctico

Métodos de análisis sintáctico

- Dos grandes clases de análisis sintáctico
 - Descendentes (top-down)
 - Para una secuencia de palabras, construye un árbol de análisis sintáctico desde la raíz a las hojas.
 - Ascendentes (bottom-up)
 - Para una secuencia de palabras, construye un árbol de análisis sintáctico desde las hojas a la raíz.
- Existen combinaciones

Espacio de estados de parsers

- Top-down
 - Formas sentenciales que se obtienen realizando todas las derivaciones posibles a partir de S.
 - El objetivo es llegar a la oración de entrada.
- Bottom-up
 - Formas sentenciales que se obtienen empezando por la oración de entrada y utilizando las reglas “el revés”.
 - El objetivo es llegar al símbolo inicial S.

Ejemplo

Tomo un examen.

Yo tomo un vuelo directo a París.

Tomo un vuelo de Montevideo a París.

Vuelo.

O → GV | GN GV

GV → V | V GN | V GN GP

GN → Det Nom | Det Nom Adj | NomP | GN GP | Pron

GP → Prep GN

V → vuelo | tomo

Det → un

Nom → examen | tomo | vuelo

Pron → yo

Prep → a | de

NomP → París | Montevideo

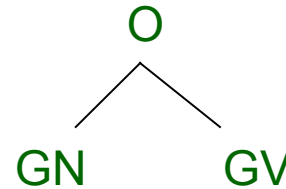
Adj → directo

Ejemplo: top-down

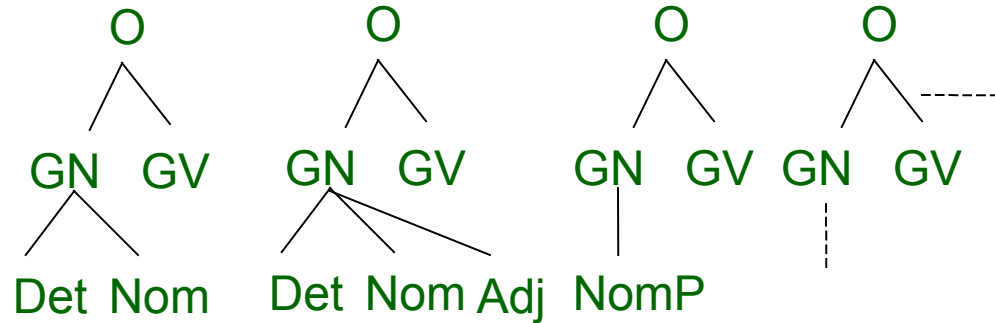
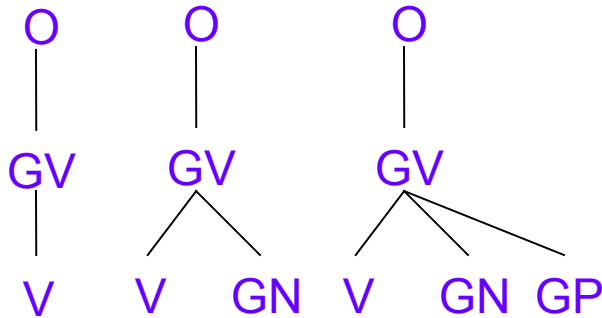
paso 1

O

paso 2

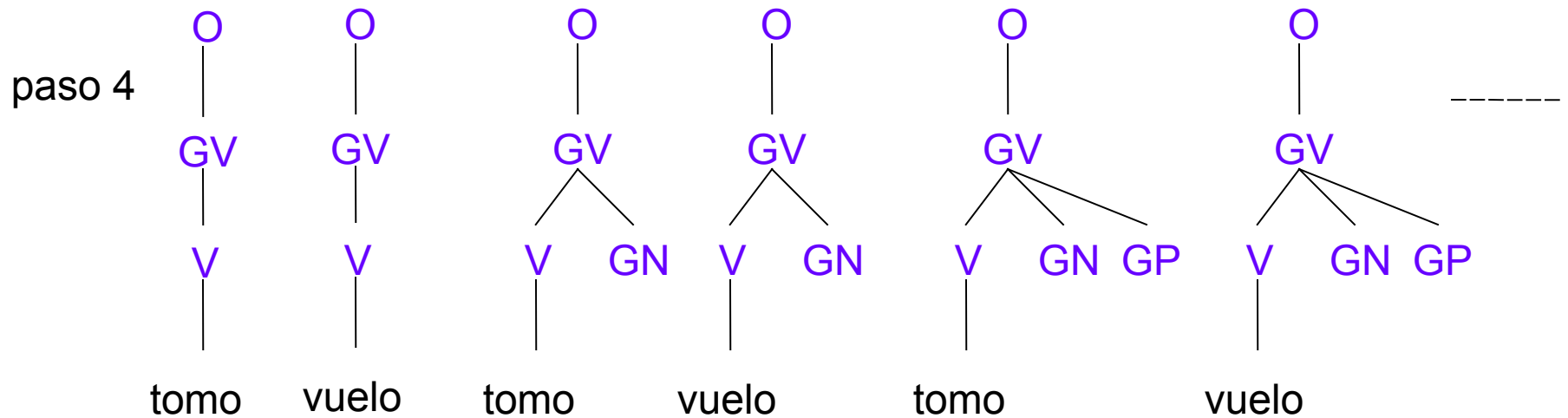


paso 3



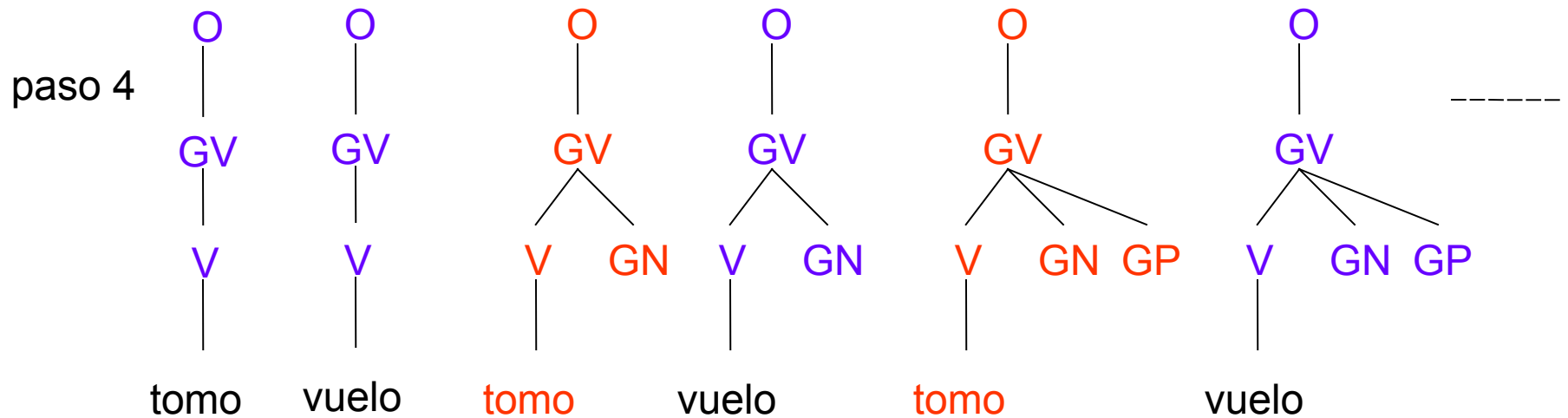
Analizar: “tomo un examen”

Ejemplo: top-down



Analizar: "tomo un examen"

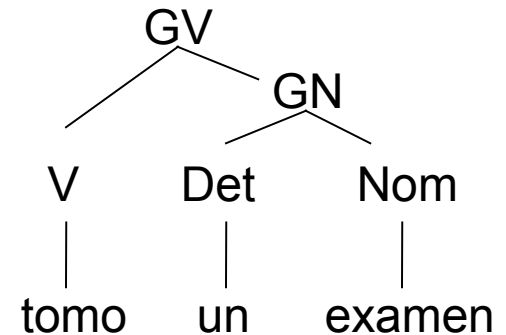
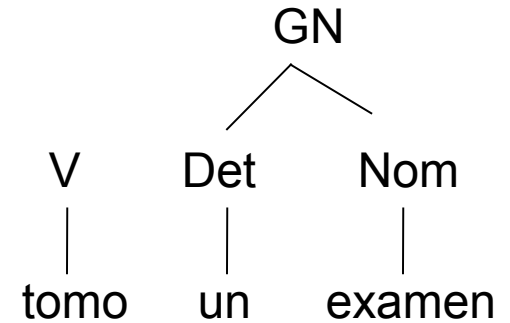
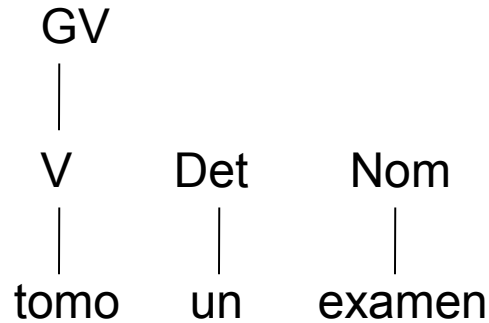
Ejemplo: top-down



Analizar: “tomo un examen”

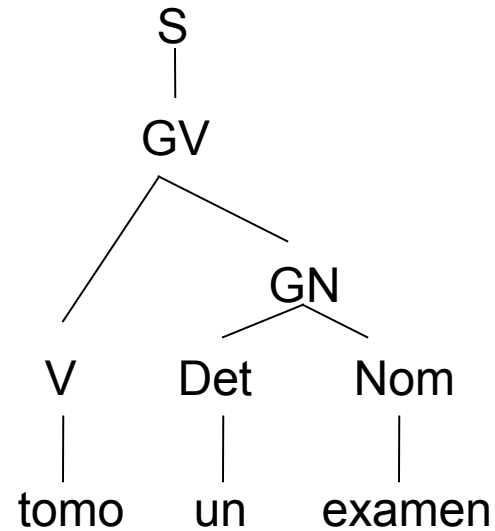
Ejemplo: bottom-up

tomo un examen



Analizar: “tomo un examen”

Ejemplo: bottom-up



Analizar: “tomo un examen”

Comparación de ambos parsers simples

- **Top-Down:**
 - Sólo genera árboles que comienzan en S .
 - Pierde tiempo con árboles que no son consistentes con la entrada.

- **Bottom-up:**
 - Explora árboles que no llegarán a S .
 - Nunca sugieren árboles que no sean al menos localmente consistentes con la entrada.

Parsing con programación dinámica

- Los subárboles de los constituyentes se almacenan en tablas
 - No hay análisis duplicados
 - Ambigüedad: todas las posibilidades son analizadas
- Tres métodos clásicos:
 - Cocke-Kasami-Younger (CKY, 1965)
 - Earley (1968)
 - Chart Parsing (Key, 1967; Kaplan, 1973)

Cocke-Kasami-Younger

- La gramática debe estar escrita acorde a la Forma Normal de Chomsky (FNC):
 - Todas las producciones son de la forma:
 - $X \rightarrow Y_1 Y_2$ (X, Y_1, Y_2 variables)
 - $X \rightarrow w$ (X variable, w terminal)
- Algoritmo para llevar a FNC (G no tiene producciones ϵ):
 - eliminación de reglas que mezclan terminales y no terminales (se agregan variables y reglas)
 - eliminación de reglas unitarias (se modifican reglas)
 - eliminación de reglas de largo mayor a dos en el lado derecho (se agregan variables y reglas)

Gramática transformada a FNC

O → **tomo** | **vuelo** |
V GN |
V **Aux1** |
GN GV

GN → Det Nom |
Det **Aux2** |
París | **Montevideo** |
GN GP |
yo

GV → **tomo** | **vuelo** |
V GN |
V **Aux1**

GP → Prep GN

Aux1 → **GN GP**
Aux2 → **Nom Adj**

Det → un
Nom → tomo | examen | vuelo
Pron → yo
NomP → París | Montevideo
Prep → a | de
V → tomo | vuelo
Adj → directo

CKY: idea del algoritmo

- Técnica bottom-up. A partir de un par de categorías (lado derecho de una regla) generamos el lado izquierdo.
- Como la gramática está en FNC, las combinaciones posibles son binarias.
- Se puede utilizar una matriz bidimensional para representar un árbol de análisis sintáctico.

- Ejemplos: usar CKY para analizar: “tomo un examen”, “tomo un vuelo a París”

Tabla CKY 0 a 1

	1	2	3	4	5
0	V Nom GV O				
1					
2					
3					
4					

0 *Tomo* 1 *un* 2 *vuelo* 3 *a* 4 *París* 5

Tabla CKY 1 a 2

	1	2	3	4	5
0	V Nom GV O				
1		Det			
2					
3					
4					

0 **Tomo** 1 **un** 2 **vuelo** 3 **a** 4 **París** 5

Tabla CKY 2 a 3

	1	2	3	4	5
0	V Nom GV O				
1		Det			
2			V Nom GV O		
3					
4					

0 *Tomo* 1 *un* 2 *vuelo* 3 *a* 4 *París* 5

Tabla CKY 3 a 4

	1	2	3	4	5
0	V Nom GV O				
1		Det			
2			V Nom GV O		
3				Prep	
4					

0 *Tomo* 1 *un* 2 *vuelo* 3 *a* 4 *París* 5

Tabla CKY 4 a 5

	1	2	3	4	5	
0	V Nom GV O					
1		Det				
2			V Nom GV O			
3				Prep		
4					NomP GN	
	0 <i>Tomo</i>	1 <i>un</i>	2 <i>vuelo</i>	3 <i>a</i>	4 <i>París</i>	5

Tabla CKY 0 a 2
largo de span = 2

	1	2	3	4	5
0	V Nom GV O				
1		Det			
2			V Nom GV O		
3				Prep	
4					NomP GN

0 *Tomo* 1 *un* 2 *vuelo* 3 *a* 4 *París* 5

Tabla CKY 1 a 3

largo de span = 2

GN → Det Nom

	1	2	3	4	5
0	V Nom GV O				
1		Det	GN		
2			V Nom GV O		
3				Prep	
4					NomP GN

0 **Tomo** 1 **un** 2 **vuelo** 3 **a** 4 **París** 5

Tabla CKY 2 a 4
largo de span = 2

	1	2	3	4	5
0	V Nom GV O				
1		Det	GN		
2			V Nom GV O		
3				Prep	
4					NomP GN

0 **Tomo** 1 **un** 2 **vuelo** 3 **a** 4 **París** 5

Tabla CKY 3 a 5

largo de span = 2

GP → Prep GN

	1	2	3	4	5
0	V Nom GV O				
1		Det	GN		
2			V Nom GV O		
3				Prep	GP
4					NomP GN

0 *Tomo* 1 *un* 2 *vuelo* 3 *a* 4 *París* 5

Tabla CKY 0 a 3
largo de span = 3

O → V GN

	1	2	3	4	5
0	V Nom GV O		O		
1		Det	GN		
2			V Nom GV O		
3				Prep	GP
4					NomP GN

0 *Tomo* 1 *un* 2 *vuelo* 3 *a* 4 *París* 5

Tabla CKY 1 a 4
largo de span = 3

	1	2	3	4	5
0	V Nom GV O		O		
1		Det	GN		
2			V Nom GV O		
3				Prep	GP
4					NomP GN

0 **Tomo** 1 **un** 2 **vuelo** 3 **a** 4 **París** 5

Tabla CKY 2 a 5
largo de span = 3

	1	2	3	4	5
0	V Nom GV O		O		
1		Det	GN		
2			V Nom GV O		
3				Prep	GP
4					NomP GN

0 **Tomo** 1 **un** 2 **vuelo** 3 **a** 4 **París** 5

Tabla CKY 0 a 4
largo de span = 4

	1	2	3	4	5
0	V Nom GV O		O		
1		Det	GN		
2			V Nom GV O		
3				Prep	GP
4					NomP GN

0 *Tomo* 1 *un* 2 *vuelo* 3 *a* 4 *París* 5

Tabla CKY 1 a 5
largo de span = 4

GN → GN GP
 Aux1 → GN GP

	1	2	3	4	5
0	V Nom GV O		O		
1		Det	GN		GN Aux1
2			V Nom GV O		
3				Prep	GP
4					NomP GN

0 **Tomo** 1 **un** 2 **vuelo** 3 **a** 4 **París** 5

Tabla CKY 0 a 5

largo de span = 5

O → V GN
 O → V Aux1

	1	2	3	4	5
0	V Nom GV O		O		O O
1		Det	GN		GN Aux1
2			V Nom GV O		
3				Prep	GP
4					NomP GN

0 *Tomo* 1 *un* 2 *vuelo* 3 *a* 4 *París* 5

CKY: algoritmo

Dados :

una tira de largo n , $p(i)$, $i=1,\dots,n$

una gramática G en FNC

para j de 1 a n

$tabla(j-1,j) \leftarrow \{ A \mid A \rightarrow p(j) \text{ está en } G \}$

para d de 2 a n

para (i,j) t.q. $(0 \leq i < j \leq n, j-i=d)$

para $k = i+1$ a $j-1$

$tabla(i,j) \leftarrow tabla(i,j) \cup$
 $\{A \mid A \rightarrow B C \text{ está en } G,$
 $B \text{ está en } tabla(i,k)$
 $C \text{ está en } tabla(k,j)\}$

CKY: algoritmo

Dados :

una tira de largo n , $p(i)$, $i=1,\dots,n$

una gramática G en FNC

para j de 1 a n

$\text{tabla}(j-1,j) \leftarrow \{ A \mid A \rightarrow p(j) \text{ está en } G \}$

para d de 2 a n

para (i,j) t.q. $(0 \leq i < j \leq n, j-i=d)$

para $k = i+1$ a $j-1$

$\text{tabla}(i,j) \leftarrow \text{tabla}(i,j) \cup$
 $\{ A \mid A \rightarrow B C \text{ está en } G,$
 $B \text{ está en } \text{tabla}(i,k)$
 $C \text{ está en } \text{tabla}(k,j) \}$

Inicialización:

reglas pre-léxicas en diagonal principal

Se consideran *spans* de tamaño $d=2,3,\dots,n$

Para cada d , (i,j) t.q. i comienzo posible de 1er *sub-span*; j fin posible de 2do *sub-span*

Para cada *span* se consideran todos los modos de obtenerlo como combinación de 2 *sub-spans* mediante una regla de la gramática

Algoritmo CKY

- Se recorre la tabla (solo la matriz triangular superior derecha) por **diagonales**.
- Es posible recorrer también por **columnas**.
- Cada casillero se llena una sola vez; al momento de llenarlo toda la información necesaria está ya calculada en la tabla.
- Es de orden $\mathbf{o(n^3)}$ (n, largo de la tira, consideramos cte. el tamaño de la gramática).

Algoritmo CKY

- El algoritmo presentado es un reconocedor.
- Para obtener **los análisis** se debe asociar a cada variable punteros a las entradas de las cuales se derivó.
- Para obtener **todos los análisis** se debe registrar la misma variable varias veces si se deduce de más de un modo.
- Si hay ambigüedad, esta se presenta empaquetada.

CKY: problemas

- Los árboles que retorna no son los planeados originalmente.
- Igual capacidad generativa débil (conjunto de tiras reconocidas) pero no hay equivalencia en capacidad generativa fuerte (descripciones estructurales generadas).
- Es posible re-transformar el análisis.