



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

# Análisis sintáctico descendente

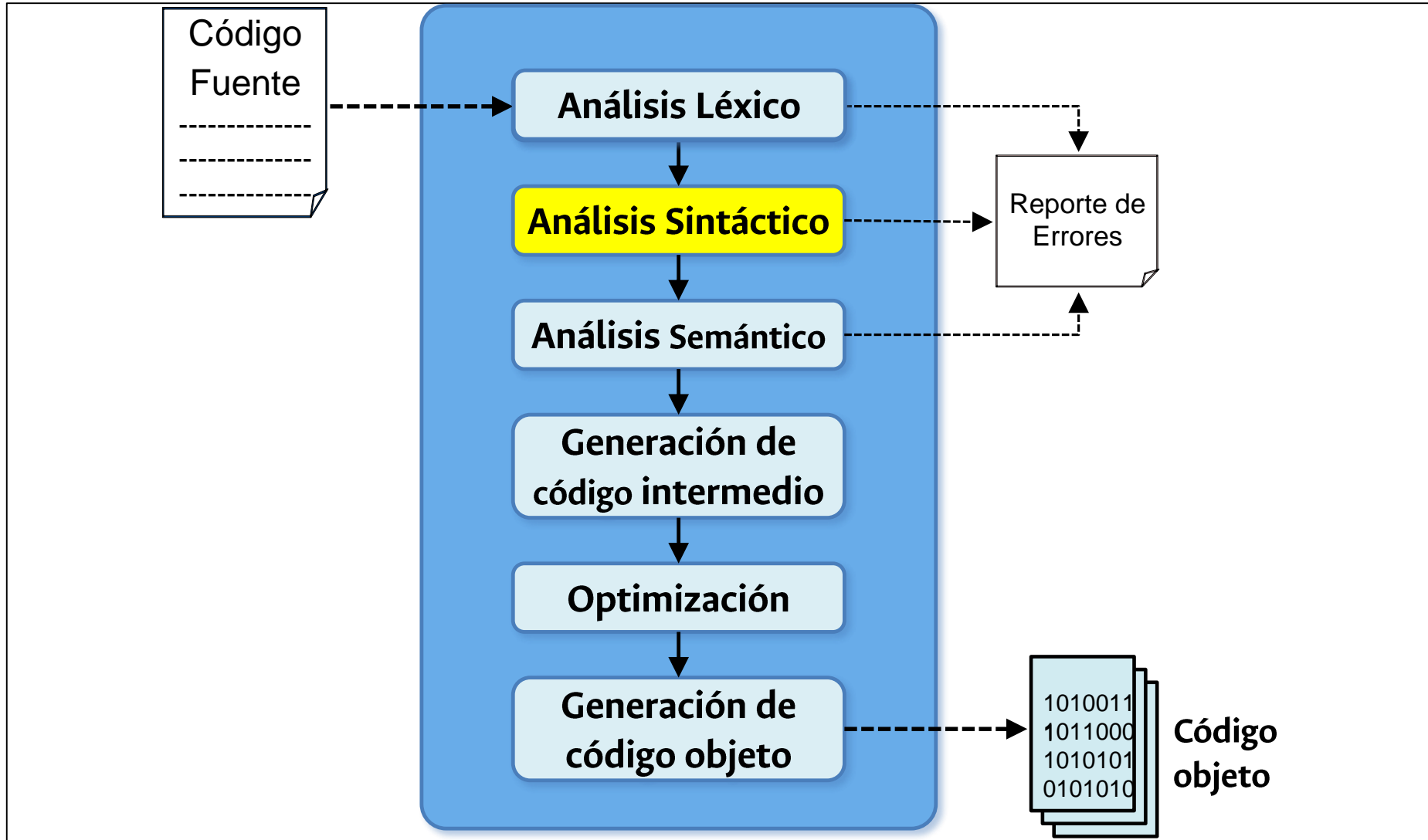
**Felipe Restrepo Calle**

[ferestrepoca@unal.edu.co](mailto:ferestrepoca@unal.edu.co)

Departamento de Ingeniería de Sistemas e Industrial  
Facultad de Ingeniería  
Universidad Nacional de Colombia  
Sede Bogotá



# Análisis Sintáctico Descendente - ASD



- 1. Análisis Sintáctico Descendente**
- 2. ASD predictivo**
- 3. Algoritmos**

## Algoritmos de análisis sintáctico

- Para cualquier GIC:
  - Cocke-Younger-Kasami (CYK), Earley, Tomita, ...  $O(n^3)$
- Si se desea costo temporal lineal  $O(n)$ , es necesario poner restricciones a las GIC. Dos estrategias:
  - **Análisis sintáctico descendente (ASD)**
  - Análisis sintáctico ascendente (ASA)

## Análisis Sintáctico Descendente – ASD

- **Análisis Sintáctico:** permite decidir si una cadena dada pertenece o no a una gramática independiente del contexto (GIC).
- **Descendente:** parte del símbolo inicial (la raíz del árbol de derivación) y trata de llegar a la cadena de terminales dada (las hojas del árbol).
- Lee la cadena de entrada **de izquierda a derecha** (**left-to-right**), para obtener la **derivación válida por la izquierda** de la cadena de entrada (**leftmost derivation**) -> **LL**



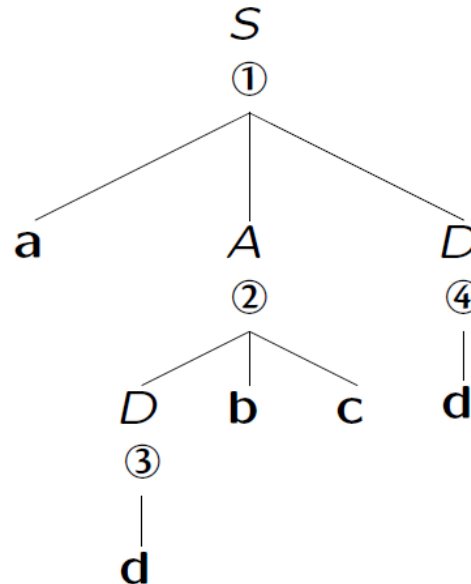
## Ejemplo ASD:

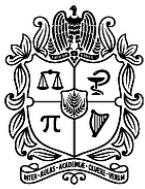
Gramática:

$$S \rightarrow a A D$$
$$A \rightarrow D b c \mid e$$
$$D \rightarrow d$$

Cadena: **a d b c d \$**

Árbol de derivación:





## Análisis Sintáctico Descendente – ASD

- Siempre se tiene que tratar de derivar el no terminal más a la izquierda en la cadena de símbolos.
- Inicialmente la cadena de símbolos sólo contiene el símbolo inicial, pero según se van aplicando reglas contiene terminales y no terminales.

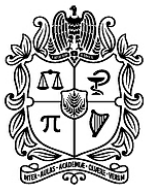
### Ejemplo:

$A \rightarrow a B C$   
 $B \rightarrow b \text{ bas}$   
 $B \rightarrow \text{big } C \text{ boss}$   
 $C \rightarrow \varepsilon$   
 $C \rightarrow c$

Cadena: **a b bas**

**Derivación por la izquierda**

$A \rightarrow a B C$   
 $\rightarrow a b \text{ bas } C$   
 $\rightarrow \dots$



## Condiciones para el ASD en tiempo lineal

- Para realizar un análisis sintáctico lineal, el analizador debe saber en todo momento qué regla debe aplicar, no puede hacer *backtracking* (complejidad **exponencial**).
- Por tanto, debe ser:

**Análisis Sintáctico Descendente Predictivo**



## Condiciones para el ASD predictivo

- A la medida que se van aplicando reglas, hay que comprobar que los símbolos terminales que aparecen (por la izquierda) coinciden (*match*) con los que aparecen en la cadena de entrada.
- Pero, dado un símbolo no terminal (ej: B), ¿cómo se puede predecir qué regla hay que aplicar?

**Mirando los primeros símbolos de las partes derechas de las reglas de ese no terminal**

**Ejemplo:** B → **b** bas

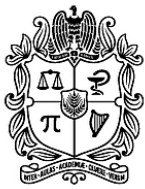
B → **big** C boss



## Condiciones para el ASD predictivo

Ejemplo:

| Gramática |  | Entrada   | Derivación |
|-----------|--|-----------|------------|
| $A$       | $\rightarrow a B C$                      | a b bas c | $A$        |
| $B$       | $\rightarrow b \text{ bas}$              | a b bas c | a $B C$    |
| $B$       | $\rightarrow \text{big } C \text{ boss}$ | b bas c   | $B C$      |
| $C$       | $\rightarrow \epsilon$                   | b bas c   | b bas $C$  |
| $C$       | $\rightarrow c$                          | bas c     | bas $C$    |
|           |  | c         | $C$        |
|           |  | c         | c          |



## Algoritmo de ASD predictivo

1. Inicialmente se tiene la cadena de entrada y el símbolo inicial de la gramática en la derivación.
2. Repetir hasta llegar al final de la cadena de entrada (\$):
  - Si el símbolo más a la izquierda en la derivación es un **terminal**, hay que compararlo con el símbolo de la entrada y avanzar **¿Y si no coinciden?**
  - Si el símbolo más a la izquierda en la derivación es un **no terminal**:
    - **¿Predecir?** Predecir qué regla aplicar en función del símbolo que hay en la entrada
    - Aplicar la regla **¿Y si no hay ninguna regla aplicable?**



## Predicción de la regla a aplicar

Para elegir (“predecir”) qué regla aplicar para un no terminal  $A$ , hay que consultar la parte derecha de las reglas de  $A$ :

|                 |            |     |             |                |
|-----------------|------------|-----|-------------|----------------|
| $A \rightarrow$ | <b>all</b> | $B$ | $C$         | { <b>all</b> } |
| $A \rightarrow$ | <b>bad</b> |     |             | { <b>bad</b> } |
| $B \rightarrow$ | <b>big</b> | $C$ | <b>boss</b> | { <b>big</b> } |
| $B \rightarrow$ | <b>bet</b> |     |             | { <b>bet</b> } |
| $C \rightarrow$ | <b>cat</b> |     |             | { <b>cat</b> } |
| $C \rightarrow$ | <b>cow</b> |     |             | { <b>cow</b> } |

En este caso, es muy fácil, debemos mirar el terminal de la entrada y el **no terminal** a derivar, y se elige la regla a aplicar.



## Predicción de la regla a aplicar

¿y si al principio de la parte derecha hay un no terminal?

|  |              |
|--|--------------|
| $A \rightarrow B C$                        | { big, bet } |
| $A \rightarrow \text{bad}$                 | { bad }      |
| $B \rightarrow \text{big } C \text{ boss}$ | { big }      |
| $B \rightarrow \text{bet}$                 | { bet }      |
| $C \rightarrow \text{cat}$                 | { cat }      |
| $C \rightarrow \text{cow}$                 | { cow }      |

Es necesario tener calculados el conjunto de terminales que aparecen al principio de la parte derecha de las reglas de ese **no terminal**, es decir, el conjunto de terminales que aparecerían al principio de las cadenas generadas por dicho **no terminal**.

## Predicción de la regla a aplicar

El conjunto de símbolos que aparecen al principio de las cadenas generadas por un **no terminal** se conoce con el nombre de **conjunto de PRIMEROS (FIRST)**:

$A \rightarrow B C$

$A \rightarrow \text{bad}$

$B \rightarrow \text{big } C \text{ boss}$

$B \rightarrow \text{bet}$

$C \rightarrow \text{cat}$

$C \rightarrow \text{cow}$

**PRIMEROS(A) = { bad, big, bet }**

**PRIMEROS(B) = { big, bet }**

**PRIMEROS(C) = { cat, cow }**

## Predicción de la regla a aplicar

¿y si un no terminal genera una cadena vacía ( $\epsilon$ )?

$A \rightarrow B C$

$A \rightarrow \text{bad}$

$B \rightarrow \text{big } C \text{ boss}$

$B \rightarrow \epsilon$

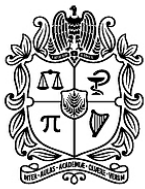
$C \rightarrow \text{cat}$

$C \rightarrow \text{cow}$

$\text{PRIMEROS}(A) = \{ \text{bad}, \text{big}, \text{cat}, \text{cow} \}$

$\text{PRIMEROS}(B) = \{ \text{big}, \epsilon \}$

$\text{PRIMEROS}(C) = \{ \text{cat}, \text{cow} \}$

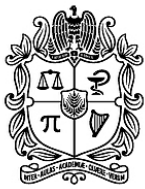


## Predicción de la regla a aplicar

### Resumen:

- En un momento dado del análisis, se tiene que derivar un **no terminal** A.
- Además, se conoce el símbolo que aparece en la cadena de entrada.
- Dado el símbolo de la entrada y el **no terminal** A, se debe elegir qué regla de A hay que aplicar para llegar a un análisis correcto sin *backtracking*.
- Para elegir la regla a aplicar, hay que consultar las partes derechas de las reglas de A:
  - Si la parte derecha de una regla empieza por un **terminal** y ese terminal coincide con el símbolo de la entrada, ésta es la regla que hay que aplicar.
  - Si la parte derecha de una regla empieza por un **no terminal** B, hay que consultar los símbolos que pueden ser generados por B; si el símbolo de la entrada está entre esos símbolos, ésta es la regla que hay que aplicar
    - ¿Y si B genera  $\varepsilon$ ? Hay que mirar los símbolos que aparecen después de B en la regla de A.





## Algoritmo para calcular el conjunto de **PRIMEROS**

El cálculo del conjunto de **PRIMEROS** de una cadena  $\alpha$  de símbolos (terminales y/o no terminales) se define formalmente así: **PRIMEROS**( $\alpha$ )

1. Si  $\alpha$  es  $\varepsilon$ , entonces  $\text{PRIMEROS}(\alpha) = \{ \varepsilon \}$
2. Si  $\alpha = a_1 a_2 \dots a_n$  donde cada  $a_i$  puede ser un terminal o un no terminal de la gramática:
  - a. Si  $a_1$  es un terminal, entonces  $\text{PRIMEROS}(\alpha) = \{ a_1 \}$
  - b. Si  $a_1$  es un no terminal, hay que añadir  $\text{PRIMEROS}(a_1) - \{ \varepsilon \}$  a  $\text{PRIMEROS}(\alpha)$
  - c. Si  $\varepsilon \in \text{PRIMEROS}(a_1)$ :
    - Si  $n = 1$ , es decir,  $\alpha = a_1$ , entonces hay que añadir  $\varepsilon$  a  $\text{PRIMEROS}(\alpha)$
    - Si  $n > 1$  entonces hay que añadir  $\text{PRIMEROS}(a_2 a_3 \dots a_n)$  a  $\text{PRIMEROS}(\alpha)$
  - d. Si  $a_1$  es un no terminal  $A$ ,  $\text{PRIMEROS}(A) = \bigcup_{A \rightarrow \alpha_i} \text{PRIMEROS}(\alpha_i)$



## Predicción de la regla a aplicar

¿Y si ...

- ... todos los símbolos de la parte derecha de la regla de A son no terminales y todos generan  $\epsilon$ ?

$A \rightarrow B C$

$A \rightarrow \dots$

$B \rightarrow \epsilon$

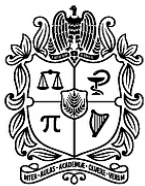
$B \rightarrow \dots$

$C \rightarrow \epsilon$

$C \rightarrow \dots$

- ... la parte derecha de A es directamente  $\epsilon$ ?  $A \rightarrow \epsilon$

**En esos casos, parece que mirando solamente los PRIMEROS no es suficiente para decidir qué regla aplicar.**



## Predicción de la regla a aplicar

Hay dos soluciones al problema de las reglas que generan  $\varepsilon$ :

1. Cuando no se puede aplicar ninguna otra regla, se aplica la regla que genera  $\varepsilon$  (algunos compiladores lo hacen, porque si hay un error en la entrada se detectará más adelante, al emparejar terminales).
2. Antes de decidir si aplicar la regla que genera  $\varepsilon$ , mirar los símbolos que pueden aparecer después de A en una derivación válida. Se denomina conjunto de **SIGUIENTES** de A:

Ejemplo:

$A \rightarrow B C$

$A \rightarrow \text{ant } A \text{ all}$

$B \rightarrow \text{big } C \text{ bad}$

$B \rightarrow \text{bus } A \text{ boss}$

$B \rightarrow \varepsilon$

$C \rightarrow \text{cat}$

$C \rightarrow \varepsilon$

**SIGUIENTES(A) = { all, boss, \$ }**



## Algoritmo para calcular el conjunto de **SIGUIENTES**

El conjunto de **SIGUIENTES** de un no terminal  $A$  se define así:

1. Si  $A$  es el símbolo inicial de la gramática, añadir  $\$$  a  $\text{SIGUIENTES}(A)$
2. Sea una regla de la gramática que contiene en su parte derecha el **no terminal**  $A$ , así:  $B \rightarrow \alpha A \beta$ , donde  $B$  es un **no terminal**, y  $\alpha$  y  $\beta$  son cadenas de terminales y no terminales de la gramática (ambas pueden ser  $\varepsilon$ ):
  - a. Añadir  $\text{PRIMEROS}(\beta) - \{ \varepsilon \}$  a  $\text{SIGUIENTES}(A)$
  - b. Si  $\varepsilon \in \text{PRIMEROS}(\beta)$  (o bien  $\beta = \varepsilon$ ), entonces hay que añadir los  $\text{SIGUIENTES}(B)$  a los  $\text{SIGUIENTES}(A)$
3. Repetir el paso 2 (a y b) hasta que no se puedan añadir más símbolos a  $\text{SIGUIENTES}(A)$

## Algoritmo para calcular el conjunto de predicción (**PRED**)

Dada una gramática  $G$  y un no terminal de la gramática  $A$ , para poder predecir en cualquier derivación qué regla de  $A$  se debe aplicar es necesario calcular el conjunto de predicción de cada regla de  $A$ :

$$\begin{aligned} \text{PRED}(A \rightarrow \alpha) = & \\ & \text{si } \epsilon \in \text{PRIMEROS}(\alpha) \text{ entonces} \\ & \quad (\text{PRIMEROS}(\alpha) - \{ \epsilon \}) \cup \text{SIGUIENTES}(A) \\ & \text{sino} \\ & \quad \text{PRIMEROS}(\alpha) \end{aligned}$$

Dado un no terminal  $A$  y un terminal de la entrada  $t$ , se aplicará la regla de  $A$  que tenga a  $t$  en su conjunto de predicción.

**¿Y si hay más de una regla de  $A$  que tenga a  $t$ ?**



## Ejemplo del cálculo de conjuntos de predicción

$A \rightarrow B C$

$A \rightarrow \text{ant } A \text{ all}$

$B \rightarrow \text{big } C$

$B \rightarrow \text{bus } A \text{ boss}$

$B \rightarrow \epsilon$

$C \rightarrow \text{cat}$

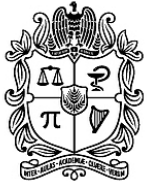
$C \rightarrow \text{cow}$

Para poder calcular el conjunto de PRIMEROS de las partes derechas de las reglas, primero se calculan los PRIMEROS de los no terminales:

$\text{PRIMEROS}(A) = \{ \text{ant, big, bus, cat, cow} \}$

$\text{PRIMEROS}(B) = \{ \text{big, bus, } \epsilon \}$

$\text{PRIMEROS}(C) = \{ \text{cat, cow} \}$



## Ejemplo del cálculo de conjuntos de predicción

$A \rightarrow B C$

$A \rightarrow \text{ant } A \text{ all}$

$B \rightarrow \text{big } C$

$B \rightarrow \text{bus } A \text{ boss}$

$B \rightarrow \epsilon$

$C \rightarrow \text{cat}$

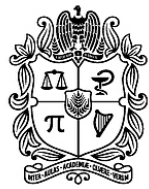
$C \rightarrow \text{cow}$

Como hay una producción que genera  $\epsilon$ , tenemos que calcular los SIGUIENTES de cada no terminal:

$\text{SIGUIENTES}(A) = \{ \text{all, boss, \$} \}$

$\text{SIGUIENTES}(B) = \{ \text{cat, cow} \}$

$\text{SIGUIENTES}(C) = \{ \text{cat, cow, all, boss, \$} \}$



## Ejemplo del cálculo de conjuntos de predicción

Los conjuntos de predicción de cada regla son:

|  |                                   |   |
|--|-----------------------------------|---|
| $A \rightarrow B C$                        | $\{ \text{big, bus, cat, cow} \}$ | $\text{PRIMEROS}(BC)$   |
| $A \rightarrow \text{ant } A \text{ all}$  | $\{ \text{ant} \}$                |   |
| $B \rightarrow \text{big } C$              | $\{ \text{big} \}$                |   |
| $B \rightarrow \text{bus } A \text{ boss}$ | $\{ \text{bus} \}$                |   |
| $B \rightarrow \epsilon$                   | $\{ \text{cat, cow} \}$           | $(\text{PRIM}(\epsilon) - \{ \epsilon \}) \cup \text{SIG}(B)$ |
| $C \rightarrow \text{cat}$                 | $\{ \text{cat} \}$                |   |
| $C \rightarrow \text{cow}$                 | $\{ \text{cow} \}$                |   |



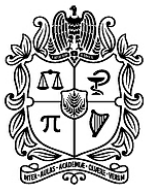


## Ejemplo del cálculo de conjuntos de predicción

Los conjuntos de predicción de cada regla son:

|  |                                   |   |
|--|-----------------------------------|---|
| $A \rightarrow B C$                        | $\{ \text{big, bus, cat, cow} \}$ | PRIMEROS(BC)  |
| $A \rightarrow \text{ant } A \text{ all}$  | $\{ \text{ant} \}$                |   |
| $B \rightarrow \text{big } C$              | $\{ \text{big} \}$                |   |
| $B \rightarrow \text{bus } A \text{ boss}$ | $\{ \text{bus} \}$                |   |
| $B \rightarrow \epsilon$                   | $\{ \text{cat, cow} \}$           | $(\text{PRIM}(\epsilon) - \{ \epsilon \}) \cup \text{SIG}(B)$ |
| $C \rightarrow \text{cat}$                 | $\{ \text{cat} \}$                |   |
| $C \rightarrow \text{cow}$                 | $\{ \text{cow} \}$                |   |

| Derivación              | Entrada        | Regla/acción                              |
|-------------------------|----------------|---|
| $A \$$                  | ant cat all \$ | $A \rightarrow \text{ant } A \text{ all}$ |
| ant $A \text{ all } \$$ | ant cat all \$ | emparejar ant                             |
| $A \text{ all } \$$     | cat all \$     | $A \rightarrow B C$                       |
| $B C \text{ all } \$$   | cat all \$     | $B \rightarrow \epsilon$                  |
| $C \text{ all } \$$     | cat all \$     | $C \rightarrow \text{cat}$                |
| cat all \$              | cat all \$     | emparejar cat                             |
| all \$                  | all \$         | emparejar all                             |
| \$                      | \$             | ??OK!!                                    |



## Ejercicios

1. Calcular los conjuntos de PRIMEROS y SIGUIENTES de los no terminales de la siguiente gramática, y los conjuntos de PREDICCIÓN de las reglas:

$S \rightarrow A \text{ uno } B C$

$S \rightarrow S \text{ dos}$

$A \rightarrow B C D$

$A \rightarrow A \text{ tres}$

$A \rightarrow \epsilon$

$B \rightarrow D \text{ cuatro } C \text{ tres}$

$B \rightarrow \epsilon$

$C \rightarrow \text{cinco } D B$

$C \rightarrow \epsilon$

$D \rightarrow \text{seis}$

$D \rightarrow \epsilon$



## Ejercicios

2. Calcular los conjuntos de PRIMEROS y SIGUIENTES de los no terminales de la siguiente gramática, y los conjuntos de PREDICCIÓN de las reglas:

$S \rightarrow A B$  **uno**

$A \rightarrow$  **dos**  $B$

$A \rightarrow \epsilon$

$B \rightarrow C D$

$B \rightarrow$  **tres**

$B \rightarrow \epsilon$

$C \rightarrow$  **cuatro**  $A B$

$C \rightarrow$  **cinco**

$D \rightarrow$  **seis**

$D \rightarrow \epsilon$