

PROCESADORES DE LENGUAJES

TEMA IV.- ANÁLISIS SINTÁCTICO DESCENDENTE

Prof. Dr. Nicolás Luis Fernández García

Departamento de Informática y Análisis Numérico
Escuela Politécnica Superior de Córdoba
Universidad de Córdoba

Programa

- Tema I.- Introducción
- Tema II.- Análisis Lexicográfico
- Tema III.- Fundamentos Teóricos del Análisis Sintáctico
- Tema IV.- Análisis Sintáctico Descendente
- Tema V.- Análisis Sintáctico Ascendente

Programa

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Programa

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Programa

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Programa

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Contenido del tema

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Contenido de la sección

- 1 Introducción
 - Objetivo
 - Tipos de análisis sintáctico descendente
 - Limitación general
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Introducción

Objetivo

Estrategias

- Comprobar si una gramática **genera** una cadena de entrada utilizando alguna de las siguientes **estrategias**:
 - Construcción de una **derivación por la izquierda** de la cadena de entrada.
 - Construcción de un **árbol sintáctico** de forma **descendente** desde la raíz hasta las hojas.

Introducción

Objetivo

Ejemplo (Gramática de expresiones aritméticas)

1 / 15

$$P = \{$$

- (1) $S \rightarrow \text{identificador} = E$
- (2) $E \rightarrow T E'$
- (3) $E' \rightarrow + T E'$
- (4) $E' \rightarrow \epsilon$
- (5) $T \rightarrow F T'$
- (6) $T' \rightarrow * F T'$
- (7) $T' \rightarrow \epsilon$
- (8) $F \rightarrow (E)$
- (9) $F \rightarrow \text{identificador}$
- (10) $F \rightarrow \text{número}$

$$\}$$

Introducción

Objetivo

Ejemplo (Gramática de expresiones aritméticas 2 / 15)

- *Sentencia de asignación*

identificador = identificador + número * identificador

Introducción

Objetivo

Ejemplo (Gramática de expresiones aritméticas

3 / 15)

Sentencia: $id = id + n * id$

Derivación por la izquierda

S

Árbol sintáctico

S

Introducción

Objetivo

Ejemplo (Gramática de expresiones aritméticas)

4 / 15

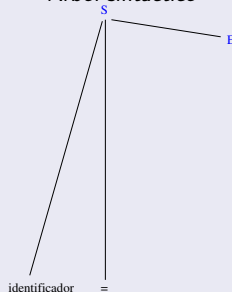
Sentencia: $id = id + n * id$

Derivación por la izquierda

$S \xrightarrow{1} id = E$

Emparejamientos: $id, =$

Árbol sintáctico



Introducción

Objetivo

Ejemplo (Gramática de expresiones aritméticas)

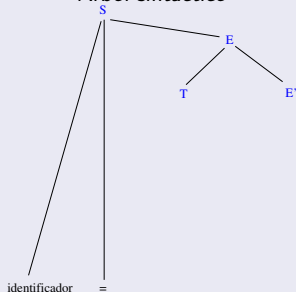
5 / 15

Sentencia: $id = id + n * id$

Derivación por la izquierda

$$\begin{array}{l}
 S \Rightarrow id = E \\
 \xrightarrow{1} \\
 \xrightarrow{2} id = \underline{TE'}
 \end{array}$$

Árbol sintáctico



Introducción

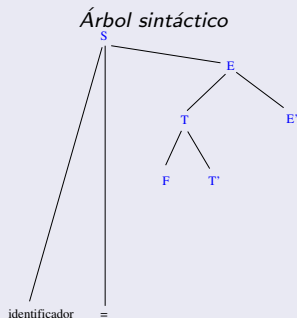
Objetivo

Ejemplo (Gramática de expresiones aritméticas)

6 / 15

Sentencia: $id = id + n * id$

Derivación por la izquierda

$$\begin{array}{l}
 S \Rightarrow id = E \\
 \xrightarrow{1} \\
 \Rightarrow id = TE' \\
 \xrightarrow{2} \\
 \Rightarrow id = \underline{FT'E'} \\
 \xrightarrow{5}
 \end{array}$$


Introducción

Objetivo

Ejemplo (Gramática de expresiones aritméticas)

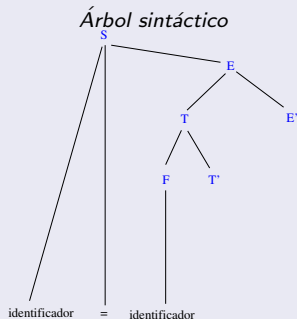
7 / 15

Sentencia: $id = id + n * id$

Derivación por la izquierda

$$\begin{array}{l}
 S \Rightarrow id = E \\
 \xrightarrow{1} \\
 \Rightarrow id = TE' \\
 \xrightarrow{2} \\
 \Rightarrow id = FT'E' \\
 \xrightarrow{5} \\
 \Rightarrow id = \underline{id}T'E' \\
 \xrightarrow{9}
 \end{array}$$

Emparejamiento de id



Introducción

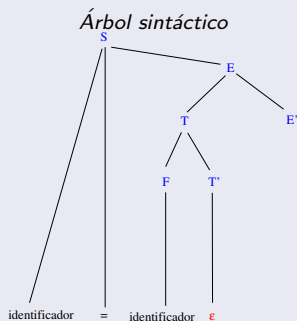
Objetivo

Ejemplo (Gramática de expresiones aritméticas)

8 / 15

Sentencia: $id = id + n * id$

Derivación por la izquierda

$$\begin{array}{l}
 S \Rightarrow id = E \\
 \xRightarrow{1} id = TE' \\
 \xRightarrow{2} id = FT'E' \\
 \xRightarrow{5} id = id T'E' \\
 \xRightarrow{9} id = id \underline{\epsilon} E' \\
 \xRightarrow{7}
 \end{array}$$


Introducción

Objetivo

Ejemplo (Gramática de expresiones aritméticas)

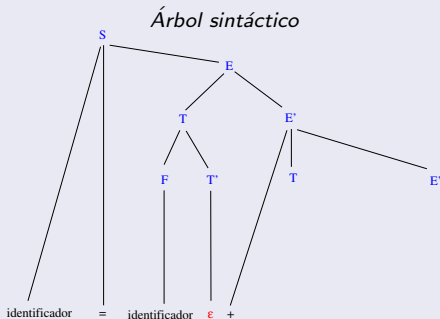
9 / 15)

Sentencia: $id = id + n * id$

Derivación por la izquierda

$$\begin{array}{l}
 S \Rightarrow id = E \\
 \xrightarrow{1} \\
 \Rightarrow id = TE' \\
 \xrightarrow{2} \\
 \Rightarrow id = FT'E' \\
 \xrightarrow{5} \\
 \Rightarrow id = id T'E' \\
 \xrightarrow{9} \\
 \Rightarrow id = id \epsilon E' \\
 \xrightarrow{7} \\
 \Rightarrow id = id \epsilon +TE' \\
 \xrightarrow{3}
 \end{array}$$

Emparejamiento de +



Introducción

Objetivo

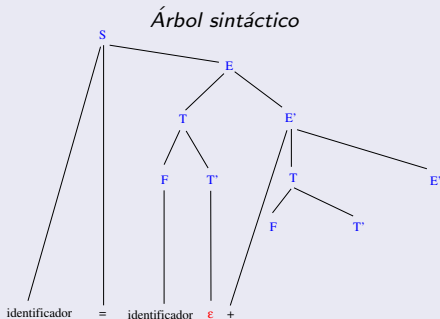
Ejemplo (Gramática de expresiones aritméticas)

10 / 15)

Sentencia: $id = id + n * id$

Derivación por la izquierda

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5}$



Introducción

Objetivo

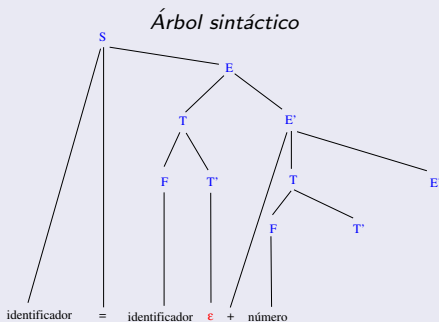
Ejemplo (Gramática de expresiones aritméticas)

11 / 15)

Sentencia: $id = id + n * id$

Derivación por la izquierda

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id \epsilon + nT'E'$
 $\xrightarrow{10}$

Emparejamiento de n 

Introducción

Objetivo

Ejemplo (Gramática de expresiones aritméticas)

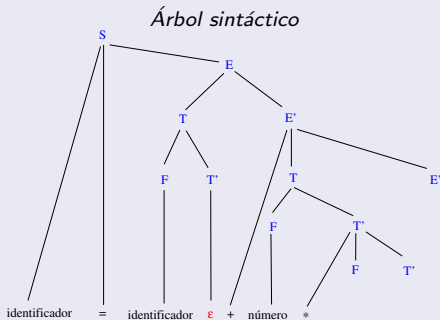
12 / 15)

Sentencia: $id = id + n * id$

Derivación por la izquierda

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id \epsilon + nT'E'$
 $\xrightarrow{10}$
 $\Rightarrow id = id \epsilon + n * FT'E'$
 $\xrightarrow{6}$

Emparejamiento de *



Introducción

Objetivo

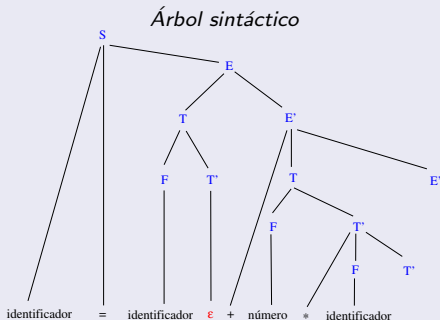
Ejemplo (Gramática de expresiones aritméticas)

13 / 15

Sentencia: $id = id + n * id$

Derivación por la izquierda

$S \Rightarrow id = E$
 $\Rightarrow id = TE'$
 $\Rightarrow id = FT'E'$
 $\Rightarrow id = idT'E'$
 $\Rightarrow id = id \epsilon E'$
 $\Rightarrow id = id \epsilon + TE'$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\Rightarrow id = id \epsilon + n T'E'$
 $\Rightarrow id = id \epsilon + n * FT'E'$
 $\Rightarrow id = id \epsilon + n * idT'E'$

Emparejamiento de id 

Introducción

Objetivo

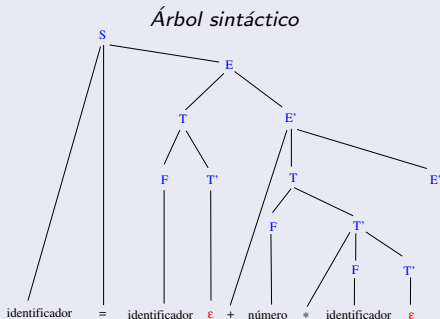
Ejemplo (Gramática de expresiones aritméticas)

14 / 15

Sentencia: $id = id + n * id$

Derivación por la izquierda

$S \Rightarrow id = E$
 $\Rightarrow id = TE'$
 $\Rightarrow id = FT'E'$
 $\Rightarrow id = idT'E'$
 $\Rightarrow id = id \epsilon E'$
 $\Rightarrow id = id \epsilon + TE'$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\Rightarrow id = id \epsilon + n T'E'$
 $\Rightarrow id = id \epsilon + n * FT'E'$
 $\Rightarrow id = id \epsilon + n * id T'E'$
 $\Rightarrow id = id \epsilon + n * id \epsilon E'$



Introducción

Objetivo

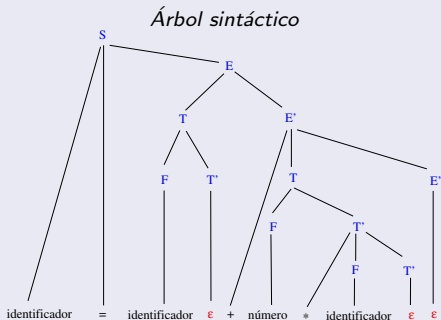
Ejemplo (Gramática de expresiones aritméticas)

15 / 15)

Sentencia: $id = id + n * id$

Derivación por la izquierda

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id \epsilon + n T'E'$
 $\xrightarrow{10}$
 $\Rightarrow id = id \epsilon + n * FT'E'$
 $\xrightarrow{6}$
 $\Rightarrow id = id \epsilon + n * id T'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon + n * id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + n * id \epsilon \epsilon$
 $\xrightarrow{4}$



Contenido de la sección

- 1 Introducción
 - Objetivo
 - Tipos de análisis sintáctico descendente
 - Limitación general
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Introducción

Tipos de análisis sintáctico descendente

Tipos de análisis sintáctico descendente

- 1.- Método de descenso recursivo **con** retroceso o *backtracking*.
- 2.- Método de descenso **predictivo**, es decir, **sin** retroceso.

Introducción

Tipos de análisis sintáctico descendente

1.- Método de descenso recursivo **con** retroceso o *backtracking*

- Utiliza **funciones** asociadas a los símbolos **no terminales** de la gramática.
- Cada función intenta **simular** el uso de las **reglas de producción** del símbolo no terminal asociado.
- Las funciones pueden ser **recursivas**.

Introducción

Tipos de análisis sintáctico descendente

2.- Método de descenso predictivo

- Utiliza una **tabla predictiva** para determinar **qué** regla de **producción** se puede usar en cada paso.

Introducción

Tipos de análisis sintáctico descendente

2.- Método de descenso predictivo

- **Tipos de implementación**

- **Método recursivo**

- Codifica una **función** para cada **símbolo no terminal** de la gramática.
 - El **código** de la función está basado en la **tabla predictiva**.
 - Cada función determina cuál es la **única** regla de producción que se puede utilizar en cada paso.
 - Las funciones pueden ser **recursivas**.

- **Método iterativo**

- En cada paso, consulta la **tabla predictiva** para determinar la **única** regla de producción que se puede aplicar.

Contenido de la sección

- 1 Introducción
 - Objetivo
 - Tipos de análisis sintáctico descendente
 - Limitación general
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Introducción

Limitación general

Limitación general

- El análisis **descendente** **no** se puede realizar si la gramática posee **recursividad por la izquierda**.
(Se explicará en las siguientes secciones).

Contenido del tema

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
 - Descripción
 - Implementación
 - Limitaciones o inconvenientes
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Descenso recursivo con retroceso o *backtracking*

Descripción

Características

- Método **exhaustivo** o de *fuerza bruta*:
 - Método de **ensayo y error**: para reconocer una cadena, intenta aplicar (**simular**) todas las reglas de producción que pueda emplear en su derivación.
- Método **con retroceso**:
 - Si la simulación de una regla de producción de $A \in V_N$ es correcta, el análisis continúa.
 - En caso contrario, hay dos posibilidades:
 - **Retroceso**: intenta probar con otra regla de A , si existe, **retrocediendo** en la cadena de entrada.
 - **Error**: la simulación **fracasa** si A **no** posee más reglas de producción.

Descenso recursivo con retroceso o *backtracking*

Descripción

Simulación de la regla $A \rightarrow X_1 X_2 \dots X_n \in P$

Para i desde 1 hasta n hacer

- Si $X_i \in V_N$ entonces se elige una regla de producción de X_i y se simula su funcionamiento.
 - Si tiene éxito, continúa el análisis.
 - En caso contrario, hay dos posibilidades:
 - **Retroceso**: se elige otra regla de X_i , si existe.
 - **Error**: la simulación **fracasa** si X_i no posee más reglas de producción.
- Si $X_i \in V_T$ entonces se intenta **emparejar** con el componente léxico actual de la cadena de entrada.
 - Si son iguales, el análisis continúa.
 - En caso contrario, **fracasa** la simulación de esta regla de A

fin_para

Descenso recursivo con retroceso o *backtracking*

Descripción

Nota (Simulación de la regla $A \rightarrow \epsilon$)

- *La simulación de la regla ϵ siempre tiene éxito.*

Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplos

- 1.- *Gramática simple.*
- 2.- *Gramática que genera sentencias de asignación de expresiones aritméticas.*

Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

1 / 12)

- Gramática

$$P = \{$$

- (1) $S \rightarrow c A d$
- (2) $A \rightarrow a b$
- (3) $A \rightarrow a$

$$\}$$

- Análisis de **c a d**

Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

2 / 12)

- *Gramática*

$$P = \{$$

$$(1) S \rightarrow c A d$$

$$(2) A \rightarrow a b$$

$$(3) A \rightarrow a$$

$$\}$$

- *Análisis de c a d*

- *Símbolo inicial: S*

Árbol sintáctico

s

Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

3 / 12)

- Gramática

$$P = \{$$

$$(1) S \rightarrow c A d$$

$$(2) A \rightarrow a b$$

$$(3) A \rightarrow a$$

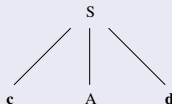
$$\}$$

- Análisis de c a d

- Regla de S

$$S \xRightarrow[1]{} cAd$$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

4 / 12)

- Gramática

$$P = \{$$

$$(1) S \rightarrow c A d$$

$$(2) A \rightarrow a b$$

$$(3) A \rightarrow a$$

$$\}$$

- Análisis de c a d

- Emparejamiento de c

$$S \xRightarrow[1]{} cAd$$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

5 / 12)

- Gramática

$$P = \{$$

$$(1) S \rightarrow c A d$$

$$(2) A \rightarrow a b$$

$$(3) A \rightarrow a$$

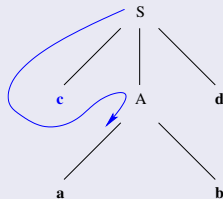
$$\}$$

- Análisis de **c a d**

- Primera regla de **A**

$$S \xRightarrow{1} cAd \xRightarrow{2} cabd$$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

6 / 12)

- Gramática

$$P = \{$$

$$(1) S \rightarrow c A d$$

$$(2) A \rightarrow a b$$

$$(3) A \rightarrow a$$

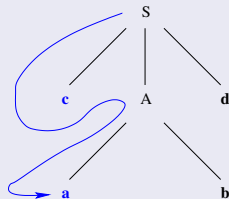
$$\}$$

- Análisis de **c a d**

- Emparejamiento de **a**

$$S \xRightarrow[1]{} cAd \xRightarrow[2]{} cabd$$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

7 / 12)

- Gramática

$$P = \{$$

$$(1) S \rightarrow c A d$$

$$(2) A \rightarrow a b$$

$$(3) A \rightarrow a$$

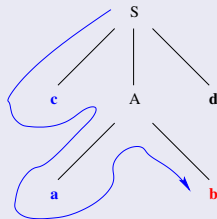
$$\}$$

- Análisis de **c a d**

- Fallo** al emparejar **b**

$$S \xRightarrow{1} cAd \xRightarrow{2} cabd$$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

8 / 12)

- Gramática

$$P = \{$$

$$(1) S \rightarrow c A d$$

$$(2) A \rightarrow a b$$

$$(3) A \rightarrow a$$

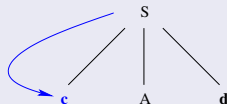
$$\}$$

- Análisis de **c a d**

- Retroceso

$$S \xRightarrow[1]{} cAd$$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

9 / 12)

- Gramática

$$P = \{$$

$$(1) S \rightarrow c A d$$

$$(2) A \rightarrow a b$$

$$(3) A \rightarrow a$$

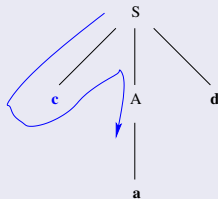
$$\}$$

- Análisis de c a d

- Segunda regla de *A*

$$S \xRightarrow{1} cAd \xRightarrow{3} cad$$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

10 / 12)

- Gramática

$$P = \{$$

$$(1) S \rightarrow c A d$$

$$(2) A \rightarrow a b$$

$$(3) A \rightarrow a$$

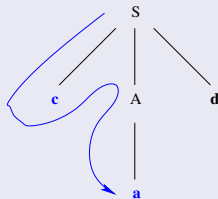
$$\}$$

- Análisis de **c a d**

- Emparejamiento de **a**

$$S \xRightarrow[1]{} cAd \xRightarrow[3]{} cad$$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

11 / 12)

- Gramática

$$P = \{$$

$$(1) S \rightarrow c A d$$

$$(2) A \rightarrow a b$$

$$(3) A \rightarrow a$$

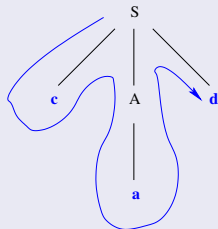
$$\}$$

- Análisis de **c a d**

- Emparejamiento de **d**

$$S \xRightarrow{1} cAd \xRightarrow{3} cad$$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (1.- Gramática simple

12 / 12)

- Gramática

$$P = \{$$

$$(1) S \rightarrow c A d$$

$$(2) A \rightarrow a b$$

$$(3) A \rightarrow a$$

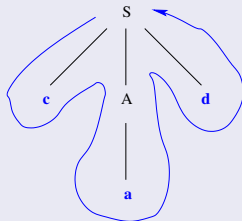
$$\}$$

- Análisis de **c a d**

- Fin del análisis

$$S \xRightarrow{1} cAd \xRightarrow{3} cad$$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 1 / 29)

$$P = \{$$

- (1) $S \rightarrow \text{identificador} = E$
- (2) $E \rightarrow T E'$
- (3) $E' \rightarrow + T E'$
- (4) $E' \rightarrow \epsilon$
- (5) $T \rightarrow F T'$
- (6) $T' \rightarrow * F T'$
- (7) $T' \rightarrow \epsilon$
- (8) $F \rightarrow (E)$
- (9) $F \rightarrow \text{identificador}$
- (10) $F \rightarrow \text{número}$

$$\}$$

Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 2 / 29)

- *Análisis mediante descenso recursivo con retroceso:*

identificador = identificador + número * identificador

Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 3 / 29)

Análisis de $\text{id} = \text{id} + \text{n} * \text{id}$

S

Árbol sintáctico

S

Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (2.- Gramática de expresiones aritméticas

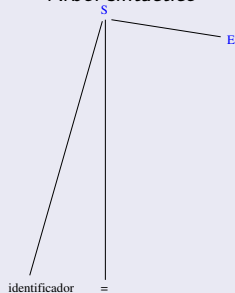
4 / 29)

Análisis de $id = id + n * id$

$S \xRightarrow[1]{} \underline{id = E}$

Emparejamientos: $id, =$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

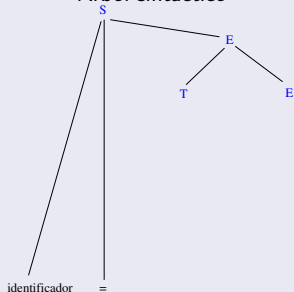
Ejemplo (2.- Gramática de expresiones aritméticas

5 / 29)

Análisis de $\text{id} = \text{id} + \text{n} * \text{id}$

$S \Rightarrow \text{id} = E$
 $\quad \underset{1}{\Rightarrow}$
 $\quad \underset{2}{\Rightarrow} \text{id} = \underline{TE'}$

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

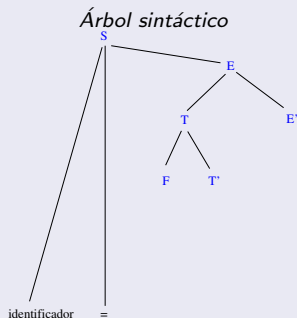
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas

6 / 29)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\quad \underset{1}{\Rightarrow}$
 $\quad id = TE'$
 $\quad \underset{2}{\Rightarrow}$
 $\quad id = \underline{FT'E'}$
 $\quad \underset{5}{\Rightarrow}$



Descenso recursivo con retroceso o *backtracking*

Descripción

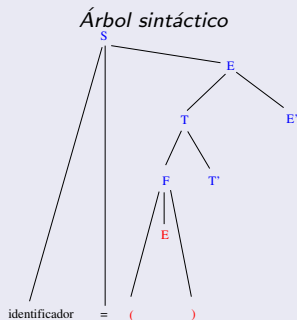
Ejemplo (2.- Gramática de expresiones aritméticas

7 / 29)

Análisis de **id = id + n * id**

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = (E)T'E'$
 $\xrightarrow{8}$

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (2.- Gramática de expresiones aritméticas

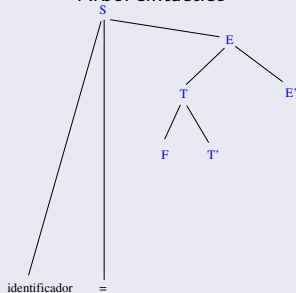
8 / 29)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\quad \underset{1}{\Rightarrow} id = TE'$
 $\quad \underset{2}{\Rightarrow} id = FT'E'$
 $\quad \underset{5}{\Rightarrow}$

Retroceso

Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Descripción

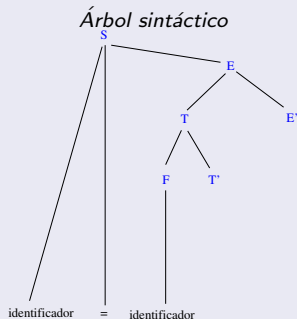
Ejemplo (2.- Gramática de expresiones aritméticas

9 / 29)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$

Emparejamiento de id



Descenso recursivo con retroceso o *backtracking*

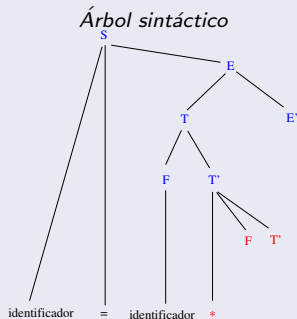
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 10 / 29)

Análisis de **id = id + n * id**

$S \Rightarrow id = E$
 $\quad \Rightarrow id = TE'$
 $\quad \Rightarrow id = FT'E'$
 $\quad \Rightarrow id = id T'E'$
 $\quad \Rightarrow id = id * FT'E'$

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

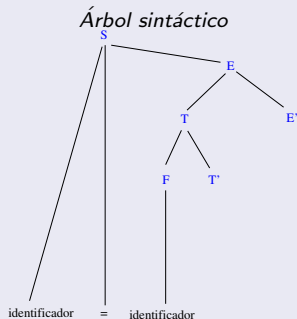
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 11 / 29)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\xrightarrow{2} id = TE'$
 $\xrightarrow{5} id = FT'E'$
 $\xrightarrow{9} id = \underline{id}T'E'$

Retroceso



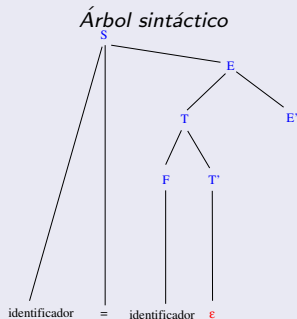
Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 12 / 29)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id T'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$



Descenso recursivo con retroceso o *backtracking*

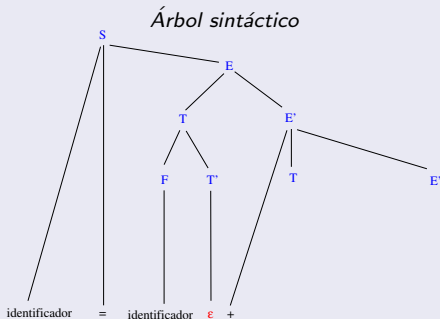
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas) 13 / 29

Análisis de **id = id + n * id**

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id T'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon +TE'$
 $\xrightarrow{3}$

Emparejamiento de +



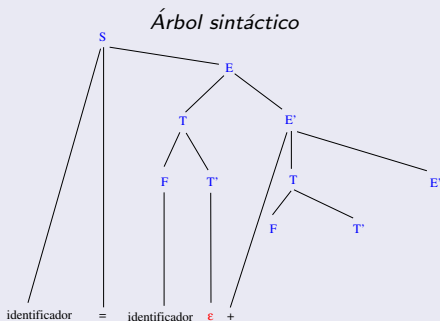
Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 14 / 29)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + \underline{FT'E'}$
 $\xrightarrow{5}$



Descenso recursivo con retroceso o *backtracking*

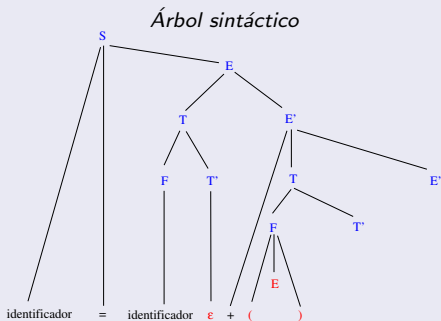
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas) 15 / 29

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\quad 1 \Rightarrow id = TE'$
 $\quad 2 \Rightarrow id = FT'E'$
 $\quad 5 \Rightarrow id = idT'E'$
 $\quad 9 \Rightarrow id = id \epsilon E'$
 $\quad 7 \Rightarrow id = id \epsilon + TE'$
 $\quad 3 \Rightarrow id = id \epsilon + FT'E'$
 $\quad 5 \Rightarrow id = id \epsilon + (E)T'E'$
 $\quad 8$

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

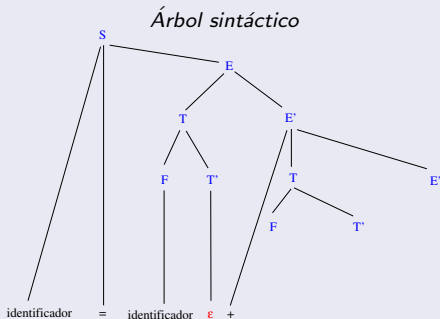
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 16 / 29)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\xrightarrow{5}$

Retroceso



Descenso recursivo con retroceso o *backtracking*

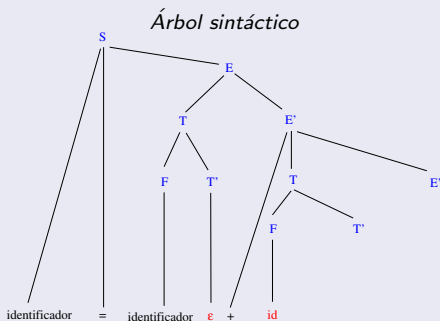
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 17 / 29)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id \epsilon + idT'E'$
 $\xrightarrow{9}$

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

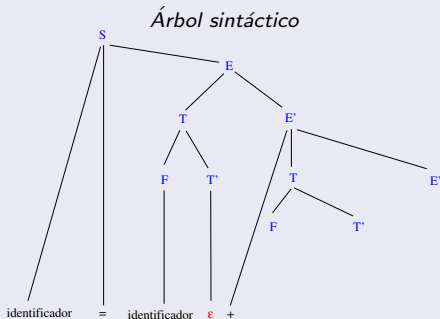
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas) 18 / 29

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + \underline{FT'E'}$
 $\xrightarrow{5}$

Retroceso



Descenso recursivo con retroceso o *backtracking*

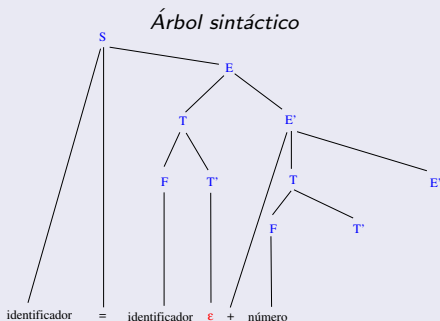
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 19 / 29)

Análisis de $id = id + n * id$

S	\Rightarrow	$id = E$
1	\Rightarrow	$id = TE'$
2	\Rightarrow	$id = FT'E'$
5	\Rightarrow	$id = idT'E'$
9	\Rightarrow	$id = id \epsilon E'$
7	\Rightarrow	$id = id \epsilon + TE'$
3	\Rightarrow	$id = id \epsilon + FT'E'$
5	\Rightarrow	$id = id \epsilon + nT'E'$
10	\Rightarrow	

Emparejamiento de n



Descenso recursivo con retroceso o *backtracking*

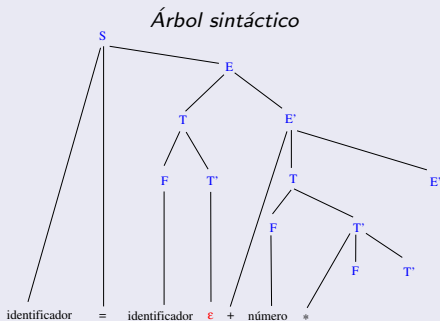
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 20 / 29)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\quad 1 \Rightarrow id = TE'$
 $\quad 2 \Rightarrow id = FT'E'$
 $\quad 5 \Rightarrow id = idT'E'$
 $\quad 9 \Rightarrow id = id \epsilon E'$
 $\quad 7 \Rightarrow id = id \epsilon + TE'$
 $\quad 3 \Rightarrow id = id \epsilon + FT'E'$
 $\quad 5 \Rightarrow id = id \epsilon + nT'E'$
 $\quad 10 \Rightarrow id = id \epsilon + n * FT'E'$
 $\quad 6$

Emparejamiento de *



Descenso recursivo con retroceso o *backtracking*

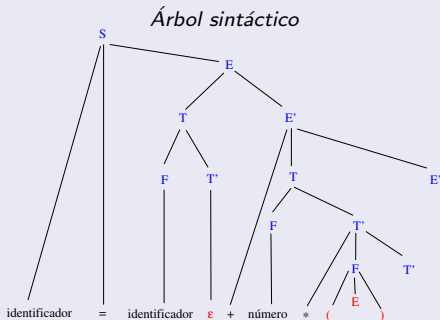
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas) 21 / 29

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 1 $\Rightarrow id = TE'$
 2 $\Rightarrow id = FT'E'$
 5 $\Rightarrow id = idT'E'$
 9 $\Rightarrow id = id \epsilon E'$
 7 $\Rightarrow id = id \epsilon + TE'$
 3 $\Rightarrow id = id \epsilon + FT'E'$
 5 $\Rightarrow id = id \epsilon + nT'E'$
 10 $\Rightarrow id = id \epsilon + n * FT'E'$
 6 $\Rightarrow id = id \epsilon + n * (E)T'E'$
 8

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

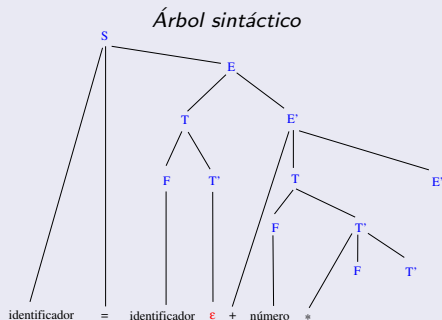
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 22 / 29)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + nT'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6}$

Retroceso



Descenso recursivo con retroceso o *backtracking*

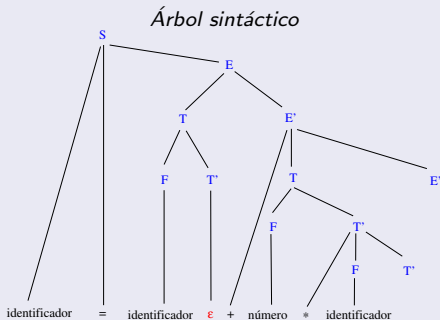
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas) 23 / 29

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * \underline{id}T'E'$
 $\xrightarrow{9}$

Emparejamiento de id



Descenso recursivo con retroceso o *backtracking*

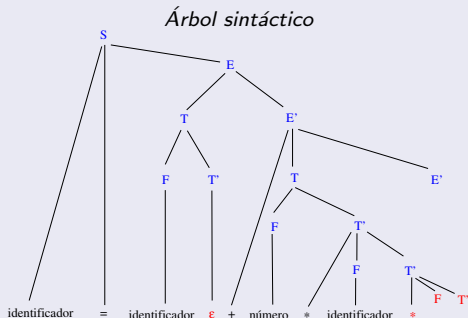
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas 24 / 29)

Análisis de $id = id + n * id$

S $\Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id \epsilon + n T'E'$
 $\xrightarrow{10}$
 $\Rightarrow id = id \epsilon + n * FT'E'$
 $\xrightarrow{6}$
 $\Rightarrow id = id \epsilon + n * id T'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon + n * id * FT'E'$
 $\xrightarrow{6}$

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

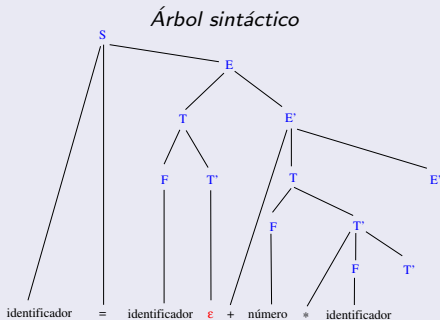
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas) 25 / 29

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * idT'E'$
 $\xrightarrow{9}$

Retroceso



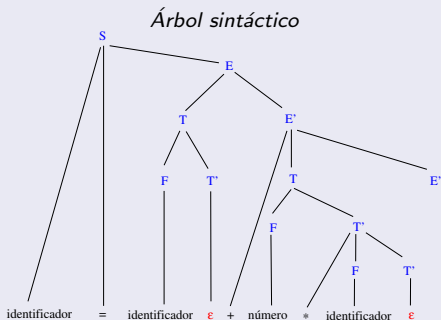
Descenso recursivo con retroceso o *backtracking*

Descripción

Ejemplo (2.- Gramática de expresiones aritméticas) 26 / 29

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * id T'E'$
 $\xrightarrow{9} id = id \epsilon + n * id \epsilon E'$
 $\xrightarrow{7}$



Descenso recursivo con retroceso o *backtracking*

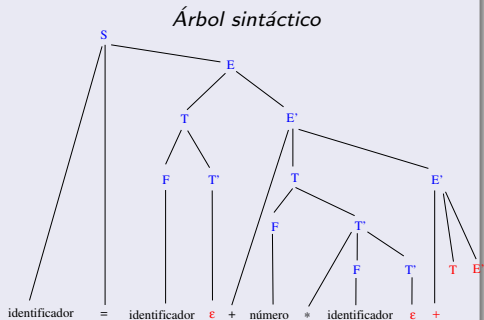
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas) 27 / 29

Análisis de $id = id + n * id$

S $\Rightarrow id = E$
 1 $\Rightarrow id = TE'$
 2 $\Rightarrow id = FT'E'$
 5 $\Rightarrow id = idT'E'$
 9 $\Rightarrow id = id \epsilon E'$
 7 $\Rightarrow id = id \epsilon + TE'$
 3 $\Rightarrow id = id \epsilon + FT'E'$
 5 $\Rightarrow id = id \epsilon + n T'E'$
 10 $\Rightarrow id = id \epsilon + n * FT'E'$
 6 $\Rightarrow id = id \epsilon + n * id T'E'$
 9 $\Rightarrow id = id \epsilon + n * id \epsilon E'$
 7 $\Rightarrow id = id \epsilon + n * id \epsilon + TE'$
 3

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

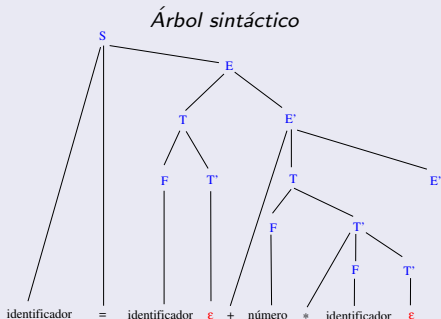
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas) 28 / 29

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\Rightarrow_1 id = TE'$
 $\Rightarrow_2 id = FT'E'$
 $\Rightarrow_5 id = idT'E'$
 $\Rightarrow_9 id = id \epsilon E'$
 $\Rightarrow_7 id = id \epsilon + TE'$
 $\Rightarrow_3 id = id \epsilon + FT'E'$
 $\Rightarrow_5 id = id \epsilon + n T'E'$
 $\Rightarrow_{10} id = id \epsilon + n * FT'E'$
 $\Rightarrow_6 id = id \epsilon + n * id T'E'$
 $\Rightarrow_9 id = id \epsilon + n * id \epsilon E'$
 \Rightarrow_7

Retroceso



Descenso recursivo con retroceso o *backtracking*

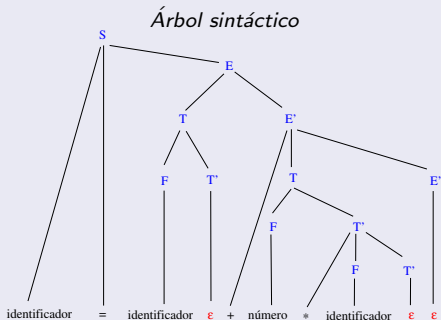
Descripción

Ejemplo (2.- Gramática de expresiones aritméticas) 29 / 29

Análisis de $id = id + n * id$

S $\Rightarrow id = E$
 1 $\Rightarrow id = TE'$
 2 $\Rightarrow id = FT'E'$
 5 $\Rightarrow id = idT'E'$
 9 $\Rightarrow id = id \epsilon E'$
 7 $\Rightarrow id = id \epsilon + TE'$
 3 $\Rightarrow id = id \epsilon + FT'E'$
 5 $\Rightarrow id = id \epsilon + n T'E'$
 10 $\Rightarrow id = id \epsilon + n * FT'E'$
 6 $\Rightarrow id = id \epsilon + n * id T'E'$
 9 $\Rightarrow id = id \epsilon + n * id \epsilon E'$
 7 $\Rightarrow id = id \epsilon + n * id \epsilon \underline{\epsilon}$
 4

Cadena generada



Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
 - Descripción
 - Implementación
 - Limitaciones o inconvenientes
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Descenso recursivo con retroceso o *backtracking*

Implementación

Método

- Se codifica una **función** para cada símbolo $A \in V_N$
- Si A posee alguna regla **recursiva** entonces la función es **recursiva**.
- La función **simula una a una** las reglas de producción de A .
- La función devuelve un valor **lógico**:
 - Si la simulación de una regla es correcta, la función devuelve **Verdadero**
 - En caso contrario, intenta probar con otra regla de A , si existe.
 - Si todas las reglas de A fallan entonces la función devuelve **Falso**.

Descenso recursivo con retroceso o *backtracking*

Implementación

Nota (Referencia)

- *Al probar con una regla, se debe establecer una **referencia** al componente léxico actual.*
- *Dicha referencia será utilizada si hay **retroceso** para simular otra regla.*

Descenso recursivo con retroceso o *backtracking*

Implementación

Simulación de la regla $A \rightarrow X_1X_2 \dots X_n \in P$

Para i desde 1 hasta n hacer

- Si $X_i \in V_N$ entonces se llama a la función asociada al símbolo X_i
 - Si tiene éxito, continúa el análisis.
 - En caso contrario, **fracasa** la simulación de esta regla de A .
- Si $X_i \in V_T$ entonces se intenta **emparejar** con el componente léxico actual de la cadena de entrada.
 - Si son iguales, el análisis continúa.
 - En caso contrario, **fracasa** la simulación de esta regla de A .

fin_para

Descenso recursivo con retroceso o *backtracking*

Implementación

Nota (Simulación de la regla $A \rightarrow \epsilon$)

- *La simulación de la regla ϵ siempre tiene éxito.*
- *Si un símbolo A posee una regla ϵ ,*
 - *deberá ser simulada en **último lugar***
 - *y la función siempre devolverá el valor *Verdadero*.*

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Gramática de expresiones aritméticas)

- **Primera parte:** *codificación de funciones asociadas a símbolos no terminales de una gramática.*
- **Segunda parte:** *uso de las funciones para analizar una expresión aritmética.*

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Gramática de expresiones aritméticas)

- **Primera parte:** *codificación de funciones asociadas a símbolos no terminales de una gramática.*
- **Segunda parte:** *uso de las funciones para analizar una expresión aritmética.*

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Gramática de expresiones aritméticas)

1 / 9

- $$P = \{$$
- (1) $S \rightarrow \text{identificador} = E$
 - (2) $E \rightarrow T E'$
 - (3) $E' \rightarrow + T E'$
 - (4) $E' \rightarrow \epsilon$
 - (5) $T \rightarrow F T'$
 - (6) $T' \rightarrow * F T'$
 - (7) $T' \rightarrow \epsilon$
 - (8) $F \rightarrow (E)$
 - (9) $F \rightarrow \text{identificador}$
 - (10) $F \rightarrow \text{número}$
- $$\}$$

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 2 / 9)

- *Codificación de las funciones asociadas a los símbolos no terminales: S, E, E', T, T' y F.*

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función S: lógico)

3 / 9

inicio

{ SIMULACIÓN DE (1) $S \rightarrow \text{identificador} = E$ }

si *emparejar*(*identificador*) = VERDADERO entonces

avanzar_entrada

 si *emparejar*("=") = VERDADERO entonces

avanzar_entrada

 si $E = \text{VERDADERO}$ entonces

 DEVOLVER VERDADERO

 si no

 DEVOLVER FALSO

 fin si

...

...

si no { *empareja* "=" }

 DEVOLVER FALSO

fin si

si no { *empareja* *identificador* }

 DEVOLVER FALSO

fin si

fin

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E: lógico)

4 / 9

inicio

{SIMULACIÓN DE (2) $E \rightarrow T E'$ }

si $T = \text{VERDADERO}$ entonces

 si $E' = \text{VERDADERO}$ entonces

 DEVOLVER *VERDADERO*

 si no

 DEVOLVER *FALSO*

 fin si

si no

 DEVOLVER *FALSO*

fin si

fin

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E': lógico)

5 / 9

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (3) $E' \rightarrow + T E'$ }

si *emparejar* ("+") = VERDADERO entonces

avanzar_entrada

si *T* = VERDADERO entonces

si *E'* = VERDADERO entonces

DEVOLVER VERDADERO

si no

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {*empareja* "+" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (4) $E' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T: lógico)

6 / 9

inicio

{SIMULACIÓN DE (5) $T \rightarrow F T$ }

si $F = \text{VERDADERO}$ entonces

 si $T' = \text{verdadero}$ entonces

 DEVOLVER **VERDADERO**

 si no

 DEVOLVER **FALSO**

 fin si

si no

 DEVOLVER **FALSO**

fin si

fin

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

7 / 9

inicio

referencia ← componente léxico actual

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si emparejar ("*") = VERDADERO entonces

avanzar_entrada

si $F = VERDADERO$ entonces

si $T' = VERDADERO$ entonces

devolver VERDADERO

si no

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {empareja "*" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (primera parte))

8 / 9

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (8) $F \rightarrow (E)$ }

si *emparejar* ("(") = VERDADERO entonces

avanzar_entrada

si *E* = VERDADERO entonces

si *emparejar* (")") = VERDADERO entonces

avanzar_entrada

DEVOLVER VERDADERO

si no {*empareja* ")"}

retroceder_entrada(*referencia*)

fin_si

...

...

si no {*es verdadero E*}

retroceder_entrada(*referencia*)

fin_si

si no {*empareja* "("}

retroceder_entrada(*referencia*)

fin si

...

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (segunda parte))

9 / 9

```

...
{SIMULACIÓN DE (9) F → identificador }
si emparejar (identificador) = VERDADERO
    entonces
        avanzar_entrada
        DEVOLVER VERDADERO
    si no {empareja identificador }
        retroceder_entrada(referencia)
fin_si
...

```

```

...
{SIMULACIÓN DE (10) F → número }
si emparejar (número) = VERDADERO
    entonces
        avanzar_entrada
        DEVOLVER VERDADERO
    si no {empareja número }
        DEVOLVER FALSO
fin_si
fin

```

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Gramática de expresiones aritméticas)

- **Primera parte:** *codificación de funciones asociadas a símbolos no terminales de una gramática.*
- **Segunda parte:** *uso de las funciones para analizar una expresión aritmética.*

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Gramática de expresiones aritméticas 1 / 113)

- *Uso de las funciones para analizar la sentencia:*

identificador = identificador + número * identificador

Descenso recursivo con retroceso o *backtracking*

Implementación

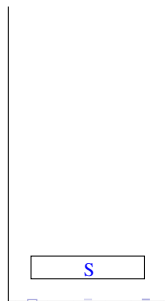
Ejemplo (Gramática de expresiones aritméticas) 2 / 113

- *Llamada a la función asociada al símbolo inicial S*

Árbol de activación

S

Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 3 / 113)

Análisis de **id = id + n * id**

S

Árbol sintáctico

S

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función S: lógico)

4 / 113

inicio

```
{SIMULACIÓN DE (1) S → identificador = E}
```

```
si emparejar(identificador) = VERDADERO entonces
```

```
  avanzar_entrada
```

```
  si emparejar("=") = VERDADERO entonces
```

```
    avanzar_entrada
```

```
    si E = VERDADERO entonces
```

```
      DEVOLVER VERDADERO
```

```
    si no
```

```
      DEVOLVER FALSO
```

```
  fin si
```

```
  ...
```

```
  ...
```

```
  si no {empareja "="}
```

```
    DEVOLVER FALSO
```

```
  fin si
```

```
  si no {empareja identificador }
```

```
    DEVOLVER FALSO
```

```
  fin si
```

```
fin
```

Descenso recursivo con retroceso o *backtracking*

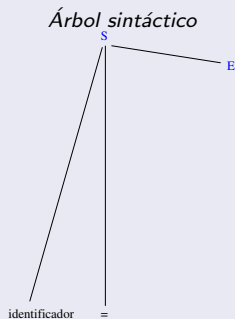
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 5 / 113)

Análisis de **id = id + n * id**

$S \xRightarrow[1]{} \underline{id = E}$

Emparejamientos: *id*, =



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función S: lógico)

6 / 113

inicio

{SIMULACIÓN DE (1) $S \rightarrow \text{identificador} = E$ }

si *emparejar*(*identificador*) = VERDADERO entonces

avanzar_entrada

si *emparejar*("=") = VERDADERO entonces

si *E* = VERDADERO entonces

DEVOLVER VERDADERO

si no

DEVOLVER FALSO

fin si

...

...

si no {*empareja* "="}

DEVOLVER FALSO

fin si

si no {*empareja* *identificador* }

DEVOLVER FALSO

fin si

fin

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función S: lógico)

7 / 113

inicio

{SIMULACIÓN DE (1) $S \rightarrow \text{identificador} = E$ }

si *emparejar*(identificador) = VERDADERO entonces

si *emparejar*("=") = VERDADERO entonces

avanzar_entrada

si $E = \text{VERDADERO}$ entonces

DEVOLVER VERDADERO

si no

DEVOLVER FALSO

fin si

...

...

si no {*empareja* "="}

DEVOLVER FALSO

fin si

si no {*empareja* identificador }

DEVOLVER FALSO

fin si

fin

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función S: lógico)

8 / 113

inicio

{SIMULACIÓN DE (1) $S \rightarrow \text{identificador} = E$ }

si *emparejar*(identificador) = VERDADERO entonces

avanzar_entrada

 si *emparejar*("=") = VERDADERO entonces

avanzar_entrada

 si $E = \text{VERDADERO}$ entonces

 DEVOLVER VERDADERO

 si no

 DEVOLVER FALSO

 fin si

...

...

si no {*empareja* "="}

 DEVOLVER FALSO

fin si

si no {*empareja* identificador }

 DEVOLVER FALSO

fin si

fin

Descenso recursivo con retroceso o *backtracking*

Implementación

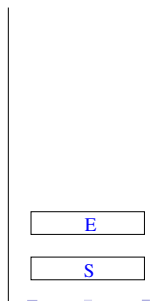
Ejemplo (Gramática de expresiones aritméticas) 9 / 113

- Llamada a la función asociada al símbolo E

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E: lógico

10 / 113)

inicio

{SIMULACIÓN DE (2) $E \rightarrow T E'$ }

si $T = VERDADERO$ entonces

 si $E' = VERDADERO$ entonces

 DEVOLVER *VERDADERO*

 si no

 DEVOLVER *FALSO*

 fin si

si no

 DEVOLVER *FALSO*

fin si

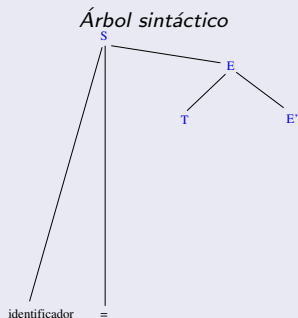
fin

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 11 / 113)

Análisis de **id = id + n * id**

$$S \Rightarrow \begin{matrix} 1 \\ \Rightarrow \\ 2 \end{matrix} \begin{matrix} \text{id} = E \\ \text{id} = \underline{TE'} \end{matrix}$$


Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E: lógico

12 / 113)

inicio

{SIMULACIÓN DE (2) $E \rightarrow T E'$ }

si $T = VERDADERO$ entonces

 si $E' = VERDADERO$ entonces

 DEVOLVER *VERDADERO*

 si no

 DEVOLVER *FALSO*

 fin si

si no

 DEVOLVER *FALSO*

fin si

fin

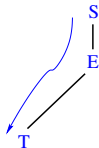
Descenso recursivo con retroceso o *backtracking*

Implementación

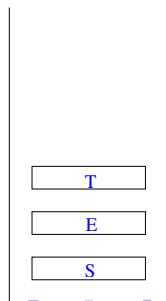
Ejemplo (Gramática de expresiones aritméticas) 13 / 113)

- *Llamada a la función asociada al símbolo T*

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T: lógico

14 / 113)

inicio

{SIMULACIÓN DE (5) $T \rightarrow F T'$ }

si $F = VERDADERO$ entonces

 si $T' = verdadero$ entonces

 DEVOLVER *VERDADERO*

 si no

 DEVOLVER *FALSO*

 fin si

si no

 DEVOLVER *FALSO*

fin si

fin

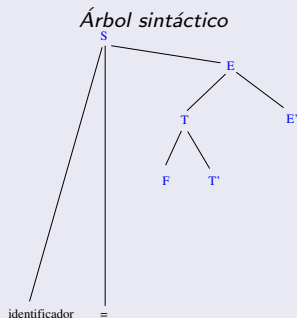
Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 15 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\quad \underset{1}{\Rightarrow}$
 $\quad \Rightarrow id = TE'$
 $\quad \underset{2}{\Rightarrow}$
 $\quad \Rightarrow id = \underline{FT'E'}$
 $\quad \underset{5}{\Rightarrow}$



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T: lógico)

16 / 113)

inicio

{SIMULACIÓN DE (5) $T \rightarrow F T$ }

si $F = \text{VERDADERO}$ entonces

 si $T' = \text{verdadero}$ entonces

 DEVOLVER *VERDADERO*

 si no

 DEVOLVER *FALSO*

 fin si

si no

 DEVOLVER *FALSO*

fin si

fin

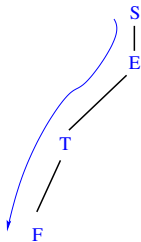
Descenso recursivo con retroceso o *backtracking*

Implementación

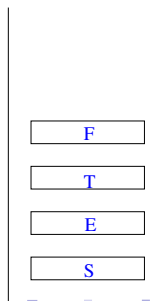
Ejemplo (Gramática de expresiones aritméticas) 17 / 113)

- Llamada a la función asociada al símbolo *F*

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (primera parte))

18 / 113

inicio

```
referencia ← componente léxico actual
```

{SIMULACIÓN DE (8) $F \rightarrow (E)$ }

si emparejar ("(") = VERDADERO entonces

 avanzar_entrada

 si E = VERDADERO entonces

 si emparejar (")") = VERDADERO entonces

 avanzar_entrada

 DEVOLVER VERDADERO

 si no {empareja ")"}

 retroceder_entrada(referencia)

 fin_si

...

...

si no {es verdadero E}

 retroceder_entrada(referencia)

fin_si

si no {empareja "("}

 retroceder_entrada(referencia)

fin_si

...

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (primera parte))

19 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (8) $F \rightarrow (E)$ }

si *emparejar* ("(") = *VERDADERO* **entonces**

avanzar_entrada

si *E* = *VERDADERO* **entonces**

si *emparejar* (")" = *VERDADERO* **entonces**

avanzar_entrada

DEVOLVER *VERDADERO*

si no {*empareja* ")"} }

retroceder_entrada(*referencia*)

fin_si

...

...

si no {*es verdadero* *E*}

retroceder_entrada(*referencia*)

fin_si

si no {*empareja* "("}

retroceder_entrada(*referencia*)

fin si

...

Descenso recursivo con retroceso o *backtracking*

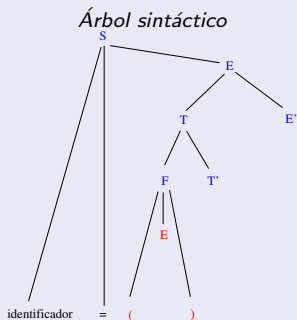
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 20 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\quad \quad \quad \Rightarrow id = TE'$
 $\quad \quad \quad \Rightarrow id = FT'E'$
 $\quad \quad \quad \Rightarrow id = (E)T'E'$

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (primera parte))

21 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (8) $F \rightarrow (E)$ }

si *emparejar* ("(") = VERDADERO entonces

avanzar_entrada

si *E* = VERDADERO entonces

si *emparejar* (")") = VERDADERO entonces

avanzar_entrada

DEVOLVER VERDADERO

si no {*empareja* ")"} }

retroceder_entrada(*referencia*)

fin_si

...

...

si no {*es verdadero* *E*}

retroceder_entrada(*referencia*)

fin_si

si no {*empareja* "("}

retroceder_entrada(*referencia*)

fin si

...

Descenso recursivo con retroceso o *backtracking*

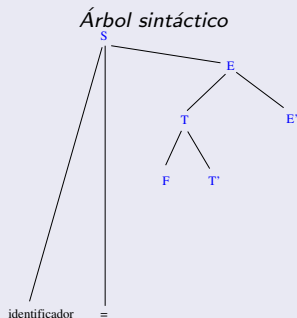
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 22 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\quad \underset{1}{\Rightarrow} id = TE'$
 $\quad \underset{2}{\Rightarrow} id = FT'E'$
 $\quad \underset{5}{\Rightarrow}$

Retroceso



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (segunda parte))

23 / 113

...

```
{ SIMULACIÓN DE (9) F → identificador }
```

```
si emparejar (identificador) = VERDADERO
```

```
    entonces
```

```
        avanzar_entrada
```

```
        DEVOLVER VERDADERO
```

```
    si no { empareja identificador }
```

```
        retroceder_entrada(referencia)
```

```
    fin_si
```

...

...

```
{ SIMULACIÓN DE (10) F → número }
```

```
si emparejar (número) = VERDADERO
```

```
    entonces
```

```
        avanzar_entrada
```

```
        DEVOLVER VERDADERO
```

```
    si no { empareja número }
```

```
        DEVOLVER FALSO
```

```
    fin_si
```

```
fin
```

Descenso recursivo con retroceso o *backtracking*

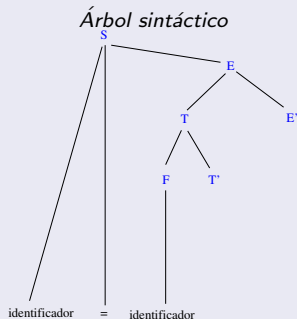
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 24 / 113)

Análisis de **id = id + n * id**

$S \Rightarrow id = E$
 $\quad \underset{1}{\Rightarrow} id = TE'$
 $\quad \underset{2}{\Rightarrow} id = FT'E'$
 $\quad \underset{5}{\Rightarrow} id = \underline{id}T'E'$
 $\quad \underset{9}{\Rightarrow}$

Emparejamiento de id



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (segunda parte))

25 / 113

...

```
{ SIMULACIÓN DE (9) F → identificador }
```

```
si emparejar (identificador) = VERDADERO
```

```
    entonces
```

```
        avanzar_entrada
```

```
        DEVOLVER VERDADERO
```

```
si no { empareja identificador }
```

```
    retroceder_entrada(referencia)
```

```
fin_si
```

...

...

```
{ SIMULACIÓN DE (10) F → número }
```

```
si emparejar (número) = VERDADERO
```

```
    entonces
```

```
        avanzar_entrada
```

```
        DEVOLVER VERDADERO
```

```
si no { empareja número }
```

```
    DEVOLVER FALSO
```

```
fin_si
```

```
fin
```

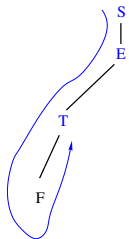

Descenso recursivo con retroceso o *backtracking*

Implementación

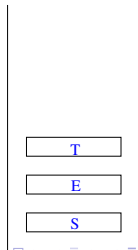
Ejemplo (Gramática de expresiones aritméticas) 26 / 113

- *Fin de la función asociada al símbolo F*
- *Regreso a la función asociada al símbolo T*

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T: lógico)

27 / 113)

inicio

{SIMULACIÓN DE (5) $T \rightarrow F T'$ }

si $F = \text{VERDADERO}$ entonces

si $T' = \text{verdadero}$ entonces

DEVOLVER *VERDADERO*

si no

DEVOLVER *FALSO*

fin si

si no

DEVOLVER *FALSO*

fin si

fin

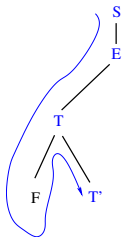
Descenso recursivo con retroceso o *backtracking*

Implementación

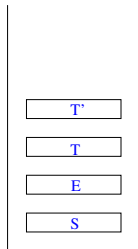
Ejemplo (Gramática de expresiones aritméticas) 28 / 113)

- Llamada a la función asociada al símbolo T'

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

29 / 113)

inicio

```
referencia ← componente léxico actual
```

```
{SIMULACIÓN DE (6) T' → * F T'}
```

```
si emparejar ("*") = VERDADERO entonces
```

```
  avanzar_entrada
```

```
  si F = VERDADERO entonces
```

```
    si T' = VERDADERO entonces
```

```
      devolver VERDADERO
```

```
    si no
```

```
      retroceder_entrada(referencia)
```

```
  fin_si
```

```
  ...
```

```
  ...
```

```
  si no
```

```
    retroceder_entrada(referencia)
```

```
  fin si
```

```
  si no {empareja "*" }
```

```
    retroceder_entrada(referencia)
```

```
  fin si
```

```
{SIMULACIÓN DE (7) T' → ε}
```

```
DEVOLVER VERDADERO
```

```
fin
```

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

30 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si *emparejar* ("*") = VERDADERO entonces

avanzar_entrada

 si *F* = VERDADERO entonces

 si *T'* = VERDADERO entonces

 devolver VERDADERO

 si no

retroceder_entrada(*referencia*)

 fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {*empareja* "*" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

Descenso recursivo con retroceso o *backtracking*

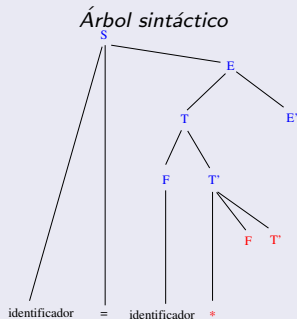
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 31 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 1
 $\Rightarrow id = TE'$
 2
 $\Rightarrow id = FT'E'$
 5
 $\Rightarrow id = id T'E'$
 9
 $\Rightarrow id = id *FT'E'$
 6

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

32 / 113

inicio

referencia ← componente léxico actual

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si emparejar ("*") = VERDADERO entonces

avanzar_entrada

si $F = \text{VERDADERO}$ entonces

si $T' = \text{VERDADERO}$ entonces

devolver VERDADERO

si no

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {empareja "*" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

Descenso recursivo con retroceso o *backtracking*

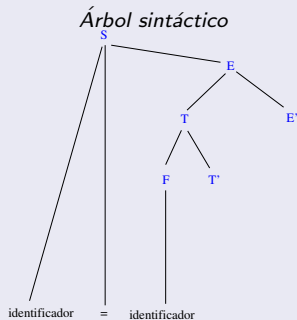
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 33 / 113)

Análisis de **id = id + n * id**

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\xrightarrow{2} id = TE'$
 $\xrightarrow{5} id = FT'E'$
 $\xrightarrow{9} id = \underline{id}T'E'$

Retroceso



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

34 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si *emparejar* ("*") = VERDADERO entonces

avanzar_entrada

si $F = \text{VERDADERO}$ entonces

si $T' = \text{VERDADERO}$ entonces

devolver VERDADERO

si no

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {*empareja* "*" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

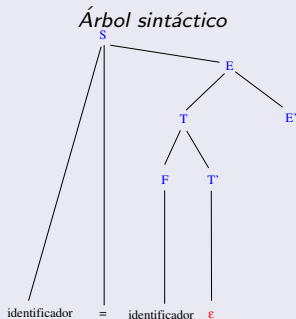
Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 35 / 113)

Análisis de **id = id + n * id**

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id T'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$



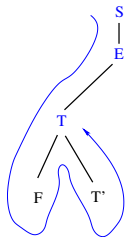
Descenso recursivo con retroceso o *backtracking*

Implementación

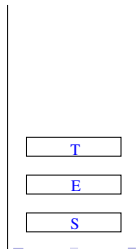
Ejemplo (Gramática de expresiones aritméticas) 36 / 113

- Fin de la función asociada al símbolo T'
- Regreso a la función asociada al símbolo T

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T: lógico)

37 / 113

inicio

{SIMULACIÓN DE (5) $T \rightarrow F T$ }

si $F = \text{VERDADERO}$ entonces

si $T' = \text{verdadero}$ entonces

DEVOLVER VERDADERO

si no

DEVOLVER FALSO

fin si

si no

DEVOLVER FALSO

fin si

fin

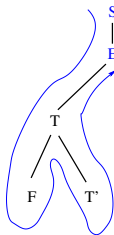
Descenso recursivo con retroceso o *backtracking*

Implementación

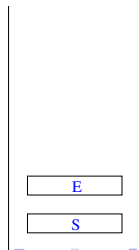
Ejemplo (Gramática de expresiones aritméticas) 38 / 113

- Fin de la función asociada al símbolo T
- Regreso a la función asociada al símbolo E

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E: lógico

39 / 113)

inicio

{SIMULACIÓN DE (2) $E \rightarrow T E'$ }

si $T = \text{VERDADERO}$ entonces

si $E' = \text{VERDADERO}$ entonces

DEVOLVER *VERDADERO*

si no

DEVOLVER *FALSO*

fin si

si no

DEVOLVER *FALSO*

fin si

fin

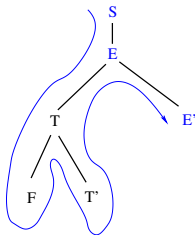
Descenso recursivo con retroceso o *backtracking*

Implementación

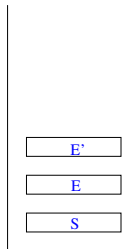
Ejemplo (Gramática de expresiones aritméticas) 40 / 113

- 1ª activación de la función asociada al símbolo E'

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E': lógico)

41 / 113

inicio

```
referencia ← componente léxico actual
```

```
{ SIMULACIÓN DE (3)  $E' \rightarrow + T E'$  }
```

```
si emparejar ("+" ) = VERDADERO entonces
```

```
  avanzar_entrada
```

```
  si T = VERDADERO entonces
```

```
    si E' = VERDADERO entonces
```

```
      DEVOLVER VERDADERO
```

```
    si no { empareja "+" }
```

```
      retroceder_entrada(referencia)
```

```
    fin_si
```

```
  ...
```

```
  ...
```

```
  si no
```

```
    retroceder_entrada(referencia)
```

```
  fin si
```

```
si no
```

```
  retroceder_entrada(referencia)
```

```
fin si
```

```
{ SIMULACIÓN DE (4)  $E' \rightarrow \epsilon$  }
```

```
DEVOLVER VERDADERO
```

```
fin
```


Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E': lógico)

42 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (3) $E' \rightarrow + T E'$ }

si *emparejar* (“+”) = *VERDADERO* entonces

avanzar_entrada

si *T* = *VERDADERO* entonces

si *E'* = *VERDADERO* entonces

DEVOLVER *VERDADERO*

si no {*empareja* “+” }

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (4) $E' \rightarrow \epsilon$ }

DEVOLVER *VERDADERO*

fin

Descenso recursivo con retroceso o *backtracking*

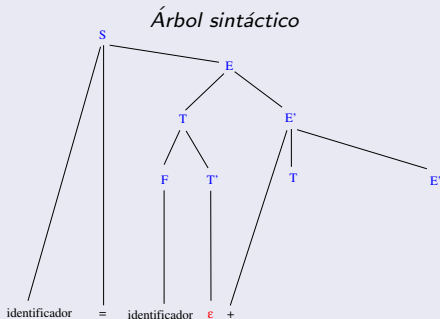
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 43 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id T'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon +TE'$
 $\xrightarrow{3}$

Emparejamiento de +



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E': lógico)

44 / 113

inicio

referencia ← *componente léxico actual*

{ SIMULACIÓN DE (3) $E' \rightarrow + T E'$ }

si emparejar ("+") = VERDADERO entonces

avanzar_entrada

si $T = \text{VERDADERO}$ entonces

si $E' = \text{VERDADERO}$ entonces

DEVOLVER VERDADERO

si no { empareja "+" }

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no

retroceder_entrada(*referencia*)

fin si

{ SIMULACIÓN DE (4) $E' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E': lógico)

45 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (3) $E' \rightarrow + T E'$ }

si *emparejar* ("+") = VERDADERO entonces

avanzar_entrada

si $T = \text{VERDADERO}$ entonces

si $E' = \text{VERDADERO}$ entonces

DEVOLVER VERDADERO

si no {*empareja* "+" }

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (4) $E' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

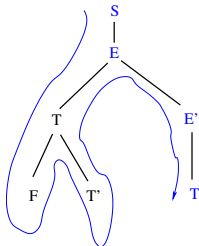
Descenso recursivo con retroceso o *backtracking*

Implementación

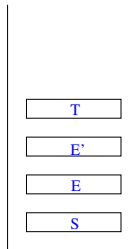
Ejemplo (Gramática de expresiones aritméticas) 46 / 113

- Llamada a la función asociada al símbolo T

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T: lógico

47 / 113)

inicio

{SIMULACIÓN DE (5) $T \rightarrow F T'$ }

si $F = \text{VERDADERO}$ entonces

 si $T' = \text{verdadero}$ entonces

 DEVOLVER *VERDADERO*

 si no

 DEVOLVER *FALSO*

 fin si

si no

 DEVOLVER *FALSO*

fin si

fin

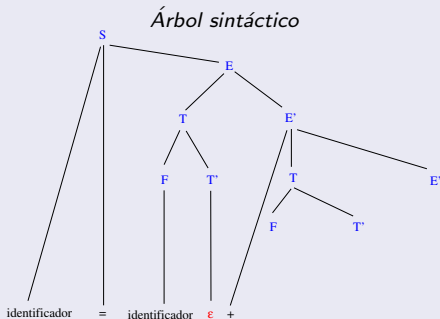
Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 48 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + \underline{FT'E'}$
 $\xrightarrow{5}$



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T: lógico)

49 / 113)

inicio

{SIMULACIÓN DE (5) $T \rightarrow F T$ }

si $F = \text{VERDADERO}$ entonces

 si $T' = \text{verdadero}$ entonces

 DEVOLVER *VERDADERO*

 si no

 DEVOLVER *FALSO*

 fin si

si no

 DEVOLVER *FALSO*

fin si

fin

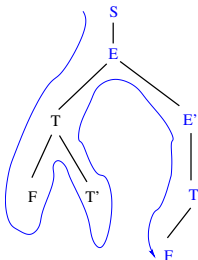
Descenso recursivo con retroceso o *backtracking*

Implementación

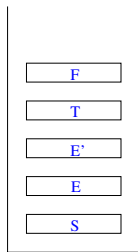
Ejemplo (Gramática de expresiones aritméticas) 50 / 113

- *Llamada a la función asociada al símbolo F*

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (primera parte))

51 / 113

inicio

```
referencia ← componente léxico actual
```

{SIMULACIÓN DE (8) $F \rightarrow (E)$ }

si emparejar ("(") = VERDADERO entonces

 avanzar_entrada

 si E = VERDADERO entonces

 si emparejar (")" = VERDADERO entonces

 avanzar_entrada

 DEVOLVER VERDADERO

 si no {empareja ")"}

 retroceder_entrada(referencia)

 fin_si

...

...

si no {es verdadero E}

 retroceder_entrada(referencia)

fin_si

si no {empareja "("}

 retroceder_entrada(referencia)

fin si

...

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (primera parte))

52 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (8) $F \rightarrow (E)$ }

si *emparejar* ("(") = *VERDADERO* **entonces**

avanzar_entrada

si *E* = *VERDADERO* **entonces**

si *emparejar* (")" = *VERDADERO* **entonces**

avanzar_entrada

DEVOLVER *VERDADERO*

si no {*empareja* ")"} }

retroceder_entrada(*referencia*)

fin_si

...

...

si no {*es verdadero* *E*}

retroceder_entrada(*referencia*)

fin_si

si no {*empareja* "("}

retroceder_entrada(*referencia*)

fin si

...

Descenso recursivo con retroceso o *backtracking*

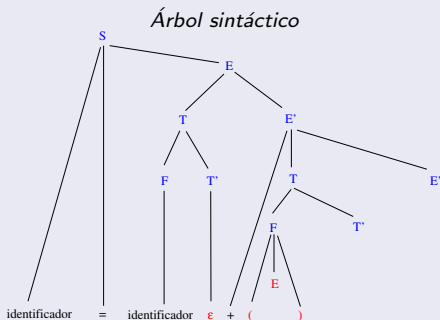
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 53 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\Rightarrow id = TE'$
 $\Rightarrow id = FT'E'$
 $\Rightarrow id = idT'E'$
 $\Rightarrow id = id \epsilon E'$
 $\Rightarrow id = id \epsilon + TE'$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\Rightarrow id = id \epsilon + (E)T'E'$

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (primera parte))

54 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (8) $F \rightarrow (E)$ }

si *emparejar* ("(") = VERDADERO **entonces**

avanzar_entrada

si *E* = VERDADERO **entonces**

si *emparejar* (")") = VERDADERO **entonces**

avanzar_entrada

DEVOLVER VERDADERO

si no {*empareja* ")"} }

retroceder_entrada(*referencia*)

fin_si

...

...

si no {*es verdadero* *E*}

retroceder_entrada(*referencia*)

fin_si

si no {*empareja* "("}

retroceder_entrada(*referencia*)

fin si

...

Descenso recursivo con retroceso o *backtracking*

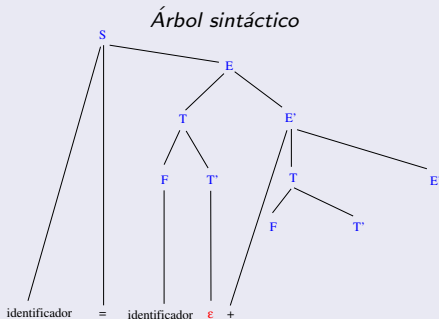
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 55 / 113)

Análisis de **id = id + n * id**

S \Rightarrow id = E
 1 \Rightarrow id = TE'
 2 \Rightarrow id = FT'E'
 5 \Rightarrow id = idT'E'
 9 \Rightarrow id = id ϵ E'
 7 \Rightarrow id = id ϵ + TE'
 3 \Rightarrow id = id ϵ + FT'E'
 5

Retroceso



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (segunda parte))

56 / 113

```

...
{SIMULACIÓN DE (9) F → identificador }
si emparejar (identificador) = VERDADERO
    entonces
        avanzar_entrada
        DEVOLVER VERDADERO
    si no {empareja identificador }
        retroceder_entrada(referencia)
fin_si
...

```

```

...
{SIMULACIÓN DE (10) F → número }
si emparejar (número) = VERDADERO
    entonces
        avanzar_entrada
        DEVOLVER VERDADERO
    si no {empareja número }
        DEVOLVER FALSO
fin_si
fin

```

Descenso recursivo con retroceso o *backtracking*

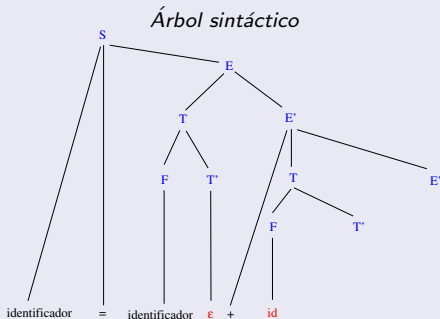
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 57 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id \epsilon + idT'E'$
 $\xrightarrow{9}$

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (segunda parte))

58 / 113

...

```
{ SIMULACIÓN DE (9) F → identificador }
```

```
si emparejar (identificador) = VERDADERO
```

```
entonces
```

```
    avanzar_entrada
```

```
    DEVOLVER VERDADERO
```

```
si no { empareja identificador }
```

```
    retroceder_entrada(referencia)
```

```
fin_si
```

...

...

```
{ SIMULACIÓN DE (10) F → número }
```

```
si emparejar (número) = VERDADERO
```

```
entonces
```

```
    avanzar_entrada
```

```
    DEVOLVER VERDADERO
```

```
si no { empareja número }
```

```
    DEVOLVER FALSO
```

```
fin_si
```

```
fin
```

Descenso recursivo con retroceso o *backtracking*

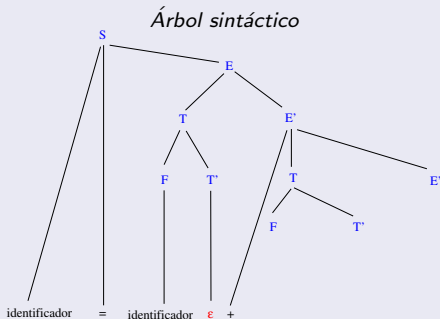
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 59 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + \underline{FT'E'}$
 $\xrightarrow{5}$

Retroceso



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (segunda parte))

60 / 113)

```

...
{SIMULACIÓN DE (9) F → identificador }
si emparejar (identificador) = VERDADERO
    entonces
        avanzar_entrada
        DEVOLVER VERDADERO
    si no {empareja identificador }
        retroceder_entrada(referencia)
fin_si
...

```

```

...
{SIMULACIÓN DE (10) F → número }
    si emparejar (número) = VERDADERO
        entonces
            avanzar_entrada
            DEVOLVER VERDADERO
        si no {empareja número }
            DEVOLVER FALSO
    fin_si
fin

```

Descenso recursivo con retroceso o *backtracking*

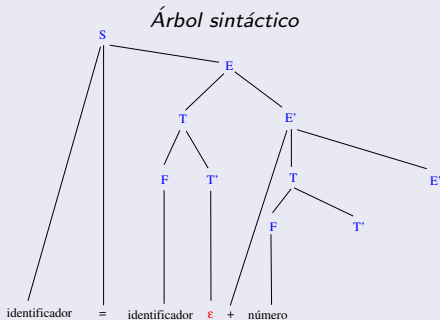
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 61 / 113)

Análisis de **id = id + n * id**

$S \Rightarrow id = E$
 $\quad \downarrow$
 $\quad id = TE'$
 $\quad \downarrow$
 $\quad id = FT'E'$
 $\quad \downarrow$
 $\quad id = idT'E'$
 $\quad \downarrow$
 $\quad id = id \epsilon E'$
 $\quad \downarrow$
 $\quad id = id \epsilon + TE'$
 $\quad \downarrow$
 $\quad id = id \epsilon + FT'E'$
 $\quad \downarrow$
 $\quad id = id \epsilon + \underline{n}T'E'$

Emparejamiento de n



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (segunda parte))

62 / 113

```

...
{SIMULACIÓN DE (9) F → identificador }
si emparejar (identificador) = VERDADERO
  entonces
    avanzar_entrada
    DEVOLVER VERDADERO
  si no {empareja identificador }
    retroceder_entrada(referencia)
fin_si
...

```

```

...
{SIMULACIÓN DE (10) F → número }
si emparejar (número) = VERDADERO
  entonces
    avanzar_entrada
    DEVOLVER VERDADERO
  si no {empareja número }
    DEVOLVER FALSO
fin_si
fin

```

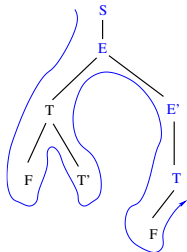
Descenso recursivo con retroceso o *backtracking*

Implementación

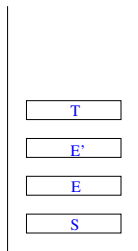
Ejemplo (Gramática de expresiones aritméticas) 63 / 113

- *Fin de la función asociada al símbolo F*
- *Regreso a la función asociada al símbolo T*

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T: lógico)

64 / 113)

inicio

{SIMULACIÓN DE (5) $T \rightarrow F T'$ }

si $F = \text{VERDADERO}$ entonces

si $T' = \text{verdadero}$ entonces

DEVOLVER *VERDADERO*

si no

DEVOLVER *FALSO*

fin si

si no

DEVOLVER *FALSO*

fin si

fin

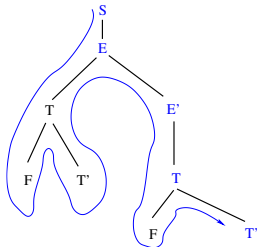
Descenso recursivo con retroceso o *backtracking*

Implementación

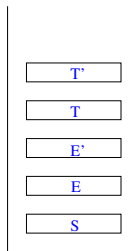
Ejemplo (Gramática de expresiones aritméticas) 65 / 113)

- 1ª activación de la función asociada al símbolo T'

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

66 / 113)

inicio

```
referencia ← componente léxico actual
```

```
{SIMULACIÓN DE (6) T' → * F T'}
```

```
si emparejar ("*") = VERDADERO entonces
```

```
  avanzar_entrada
```

```
  si F = VERDADERO entonces
```

```
    si T' = VERDADERO entonces
```

```
      devolver VERDADERO
```

```
    si no
```

```
      retroceder_entrada(referencia)
```

```
  fin_si
```

```
  ...
```

```
  ...
```

```
  si no
```

```
    retroceder_entrada(referencia)
```

```
  fin si
```

```
  si no {empareja "*" }
```

```
    retroceder_entrada(referencia)
```

```
  fin si
```

```
{SIMULACIÓN DE (7) T' → ε}
```

```
DEVOLVER VERDADERO
```

```
fin
```

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

67 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si *emparejar* ("*") = VERDADERO entonces

avanzar_entrada

 si *F* = VERDADERO entonces

 si *T'* = VERDADERO entonces

 devolver VERDADERO

 si no

retroceder_entrada(*referencia*)

 fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {*empareja* "*" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

Descenso recursivo con retroceso o *backtracking*

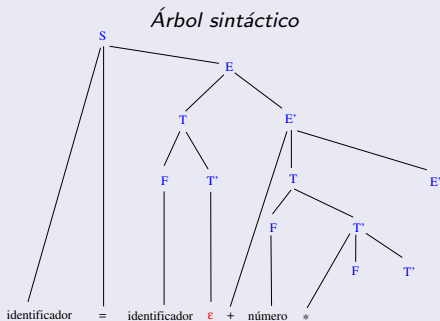
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 68 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id \epsilon + nT'E'$
 $\xrightarrow{10}$
 $\Rightarrow id = id \epsilon + n * FT'E'$
 $\xrightarrow{6}$

Emparejamiento de *



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

69 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si *emparejar* ("*") = VERDADERO entonces

avanzar_entrada

si *F* = VERDADERO entonces

si $T' = VERDADERO$ entonces

devolver VERDADERO

si no

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {*empareja* "*" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T' : lógico)

70 / 113

inicio

referencia \leftarrow *componente léxico actual*

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si emparejar ("***") = *VERDADERO* entonces

avanzar_entrada

si $F = \text{VERDADERO}$ entonces

si $T' = \text{VERDADERO}$ entonces

devolver *VERDADERO*

si no

retroceder_entrada(referencia)

fin_si

...

...

si no

retroceder_entrada(referencia)

fin si

si no {*empareja "*" }*}

retroceder_entrada(referencia)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER *VERDADERO*

fin

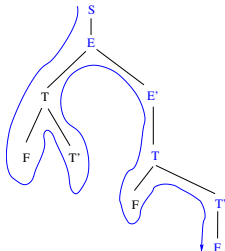
Descenso recursivo con retroceso o *backtracking*

Implementación

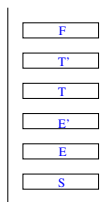
Ejemplo (Gramática de expresiones aritméticas) 71 / 113)

- *Llamada a la función asociada al símbolo F*

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (primera parte))

72 / 113

inicio

```
referencia ← componente léxico actual
```

{SIMULACIÓN DE (8) $F \rightarrow (E)$ }

si emparejar ("(") = VERDADERO entonces

 avanzar_entrada

 si E = VERDADERO entonces

 si emparejar (")" = VERDADERO entonces

 avanzar_entrada

 DEVOLVER VERDADERO

 si no {empareja ")"}

 retroceder_entrada(referencia)

 fin_si

...

...

si no {es verdadero E}

 retroceder_entrada(referencia)

fin_si

si no {empareja "("}

 retroceder_entrada(referencia)

fin si

...

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (primera parte))

73 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (8) $F \rightarrow (E)$ }

si *emparejar* ("(") = *VERDADERO* **entonces**

avanzar_entrada

si *E* = *VERDADERO* **entonces**

si *emparejar* (")" = *VERDADERO* **entonces**

avanzar_entrada

DEVOLVER *VERDADERO*

si no {*empareja* ")"} }

retroceder_entrada(*referencia*)

fin_si

...

...

si no {*es verdadero* *E*}

retroceder_entrada(*referencia*)

fin_si

si no {*empareja* "("}

retroceder_entrada(*referencia*)

fin si

...

Descenso recursivo con retroceso o *backtracking*

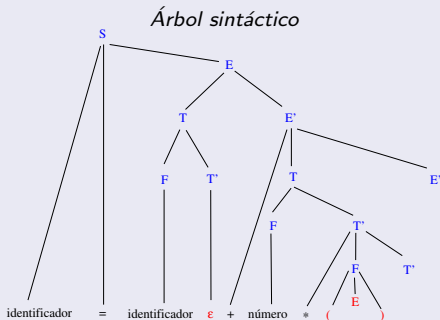
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 74 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 1 $\Rightarrow id = TE'$
 2 $\Rightarrow id = FT'E'$
 5 $\Rightarrow id = idT'E'$
 9 $\Rightarrow id = id \epsilon E'$
 7 $\Rightarrow id = id \epsilon + TE'$
 3 $\Rightarrow id = id \epsilon + FT'E'$
 5 $\Rightarrow id = id \epsilon + nT'E'$
 10 $\Rightarrow id = id \epsilon + n * FT'E'$
 6 $\Rightarrow id = id \epsilon + n * (E)T'E'$
 8

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (primera parte))

75 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (8) $F \rightarrow (E)$ }

si *emparejar* ("(") = VERDADERO entonces

avanzar_entrada

si *E* = VERDADERO entonces

si *emparejar* (")") = VERDADERO entonces

avanzar_entrada

DEVOLVER VERDADERO

si no {*empareja* ")"} }

retroceder_entrada(*referencia*)

fin_si

...

...

si no {*es verdadero* *E*}

retroceder_entrada(*referencia*)

fin_si

si no {*empareja* "("}

retroceder_entrada(*referencia*)

fin si

...

Descenso recursivo con retroceso o *backtracking*

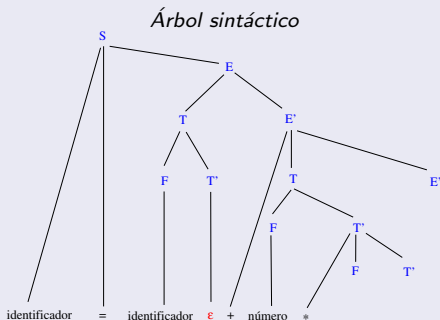
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 76 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\quad \quad \quad \downarrow$
 $\quad \quad \quad id = TE'$
 $\quad \quad \quad \downarrow$
 $\quad \quad \quad id = FT'E'$
 $\quad \quad \quad \downarrow$
 $\quad \quad \quad id = idT'E'$
 $\quad \quad \quad \downarrow$
 $\quad \quad \quad id = id \epsilon E'$
 $\quad \quad \quad \downarrow$
 $\quad \quad \quad id = id \epsilon + TE'$
 $\quad \quad \quad \downarrow$
 $\quad \quad \quad id = id \epsilon + FT'E'$
 $\quad \quad \quad \downarrow$
 $\quad \quad \quad id = id \epsilon + nT'E'$
 $\quad \quad \quad \downarrow$
 $\quad \quad \quad id = id \epsilon + n * FT'E'$

Retroceso



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (segunda parte))

77 / 113

...

```
{ SIMULACIÓN DE (9) F → identificador }
```

```
si emparejar (identificador) = VERDADERO
```

```
    entonces
```

```
        avanzar_entrada
```

```
        DEVOLVER VERDADERO
```

```
    si no { empareja identificador }
```

```
        retroceder_entrada(referencia)
```

```
    fin_si
```

...

...

```
{ SIMULACIÓN DE (10) F → número }
```

```
si emparejar (número) = VERDADERO
```

```
    entonces
```

```
        avanzar_entrada
```

```
        DEVOLVER VERDADERO
```

```
    si no { empareja número }
```

```
        DEVOLVER FALSO
```

```
    fin_si
```

```
fin
```

Descenso recursivo con retroceso o *backtracking*

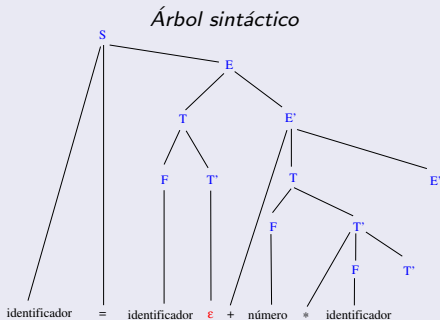
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 78 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * \underline{id}T'E'$
 $\xrightarrow{9}$

Emparejamiento de id



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función F: lógico (segunda parte))

79 / 113)

```

...
{ SIMULACIÓN DE (9) F → identificador }
si emparejar (identificador) = VERDADERO
    entonces
        avanzar_entrada
        DEVOLVER VERDADERO
    si no { empareja identificador }
        retroceder_entrada(referencia)
fin_si
...

```

```

...
{ SIMULACIÓN DE (10) F → número }
si emparejar (número) = VERDADERO
    entonces
        avanzar_entrada
        DEVOLVER VERDADERO
    si no { empareja número }
        DEVOLVER FALSO
fin_si
fin

```

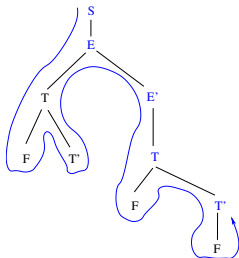
Descenso recursivo con retroceso o *backtracking*

Implementación

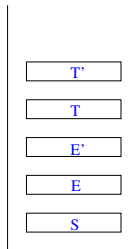
Ejemplo (Gramática de expresiones aritméticas) 80 / 113

- Fin de la función asociada al símbolo F
- Regreso a la 1ª activación de la función asociada al símbolo T'

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T' : lógico

81 / 113)

inicio

referencia \leftarrow componente léxico actual

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si emparejar ("*") = VERDADERO entonces

avanzar_entrada

si $F = \text{VERDADERO}$ entonces

si $T' = \text{VERDADERO}$ entonces

devolver VERDADERO

si no

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {empareja "*" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

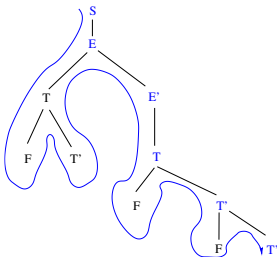
Descenso recursivo con retroceso o *backtracking*

Implementación

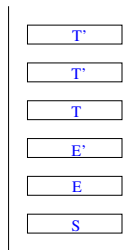
Ejemplo (Gramática de expresiones aritméticas) 82 / 113

- 2ª activación de la función asociada al símbolo T'

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

83 / 113

inicio

```
referencia ← componente léxico actual
```

```
{SIMULACIÓN DE (6) T' → * F T'}
```

```
si emparejar ("*") = VERDADERO entonces
```

```
  avanzar_entrada
```

```
  si F = VERDADERO entonces
```

```
    si T' = VERDADERO entonces
```

```
      devolver VERDADERO
```

```
    si no
```

```
      retroceder_entrada(referencia)
```

```
  fin_si
```

```
  ...
```

```
  ...
```

```
si no
```

```
  retroceder_entrada(referencia)
```

```
fin si
```

```
si no {empareja "*" }
```

```
  retroceder_entrada(referencia)
```

```
fin si
```

```
{SIMULACIÓN DE (7) T' → ε}
```

```
DEVOLVER VERDADERO
```

```
fin
```

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

84 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si *emparejar* ("*") = *VERDADERO* entonces

avanzar_entrada

si *F* = *VERDADERO* entonces

si *T'* = *VERDADERO* entonces

devolver *VERDADERO*

si no

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {*empareja* "*" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER *VERDADERO*

fin

Descenso recursivo con retroceso o *backtracking*

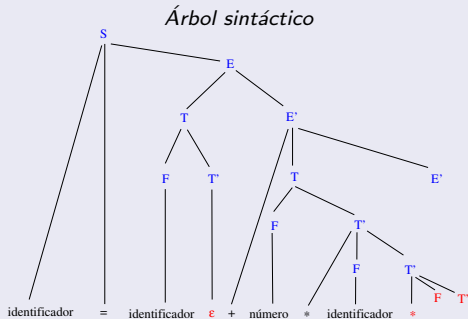
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 85 / 113)

Análisis de $id = id + n * id$

- S \Rightarrow $id = E$
- \Rightarrow $id = TE'$
- \Rightarrow $id = FT'E'$
- \Rightarrow $id = idT'E'$
- \Rightarrow $id = id \epsilon E'$
- \Rightarrow $id = id \epsilon + TE'$
- \Rightarrow $id = id \epsilon + FT'E'$
- \Rightarrow $id = id \epsilon + n T'E'$
- \Rightarrow $id = id \epsilon + n * FT'E'$
- \Rightarrow $id = id \epsilon + n * id T'E'$
- \Rightarrow $id = id \epsilon + n * id * FT'E'$

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

86 / 113)

inicio

referencia ← componente léxico actual

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si emparejar ("*") = VERDADERO entonces

avanzar_entrada

si $F = \text{VERDADERO}$ entonces

si $T' = \text{VERDADERO}$ entonces

devolver VERDADERO

si no

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {empareja "*" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

Descenso recursivo con retroceso o *backtracking*

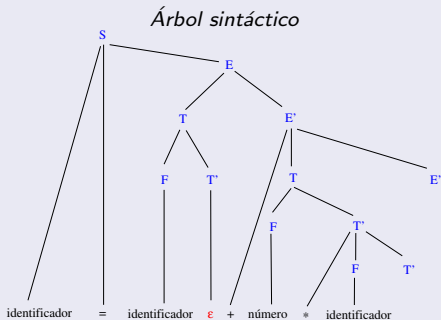
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 87 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * idT'E'$
 $\xrightarrow{9}$

Retroceso



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

88 / 113)

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si emparejar ("*") = VERDADERO entonces

avanzar_entrada

si $F = \text{VERDADERO}$ entonces

si $T' = \text{VERDADERO}$ entonces

devolver VERDADERO

si no

retroceder_entrada(referencia)

fin_si

...

...

si no

retroceder_entrada(referencia)

fin si

si no {*empareja "*" }*

retroceder_entrada(referencia)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

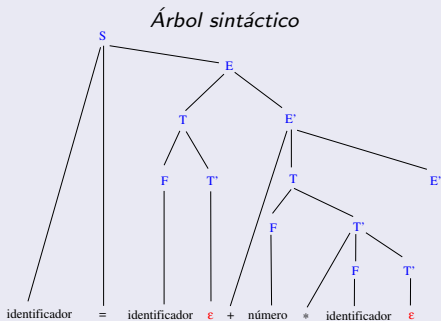
Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 89 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * id T'E'$
 $\xrightarrow{9} id = id \epsilon + n * id \epsilon E'$
 $\xrightarrow{7}$



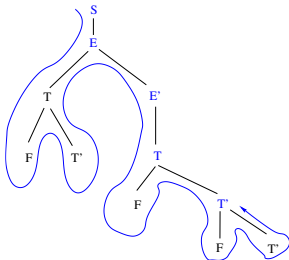
Descenso recursivo con retroceso o *backtracking*

Implementación

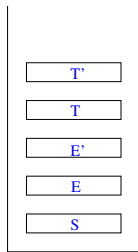
Ejemplo (Gramática de expresiones aritméticas 90 / 113)

- Fin de la 2ª activación de la función asociada al símbolo T'
- Regreso a la 1ª activación de la función asociada al símbolo T'

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T': lógico)

91 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (6) $T' \rightarrow * F T'$ }

si *emparejar* ("*") = VERDADERO entonces

avanzar_entrada

 si *F* = VERDADERO entonces

 si *T'* = VERDADERO entonces

 devolver VERDADERO

 si no

retroceder_entrada(*referencia*)

 fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {*empareja* "*" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (7) $T' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

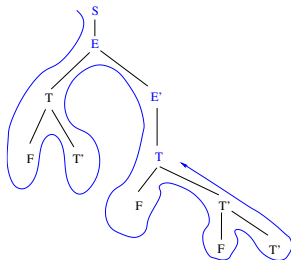
Descenso recursivo con retroceso o *backtracking*

Implementación

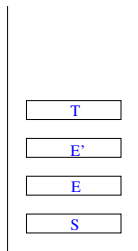
Ejemplo (Gramática de expresiones aritméticas 92 / 113)

- Fin de la 1ª activación de la función asociada al símbolo T'
- Regreso a la función asociada al símbolo T

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función T: lógico)

93 / 113)

inicio

{SIMULACIÓN DE (5) $T \rightarrow F T$ }

si $F = \text{VERDADERO}$ entonces

si $T' = \text{verdadero}$ entonces

DEVOLVER *VERDADERO*

si no

DEVOLVER *FALSO*

fin si

si no

DEVOLVER *FALSO*

fin si

fin

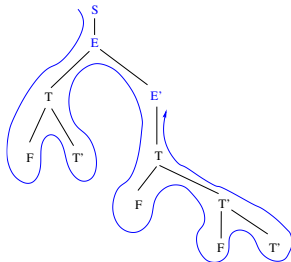
Descenso recursivo con retroceso o *backtracking*

Implementación

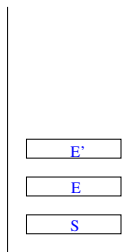
Ejemplo (Gramática de expresiones aritméticas) 94 / 113)

- Fin de la llamada a la función asociada al símbolo T
- Regreso a la 1ª activación de la función asociada al símbolo E'

Árbol de activación



Pila de activación



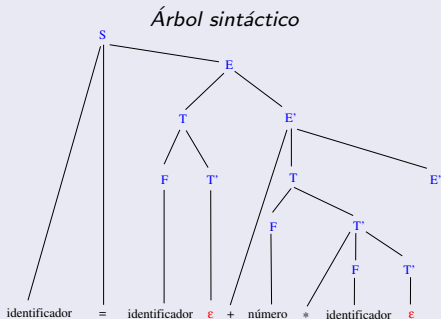
Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 95 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * id T'E'$
 $\xrightarrow{9} id = id \epsilon + n * id \epsilon E'$
 $\xrightarrow{7}$



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E': lógico)

96 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (3) $E' \rightarrow + T E'$ }

si *emparejar* ("+") = VERDADERO entonces

avanzar_entrada

si $T = \text{VERDADERO}$ entonces

si $E' = \text{VERDADERO}$ entonces

DEVOLVER VERDADERO

si no

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {*empareja* "+" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (4) $E' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

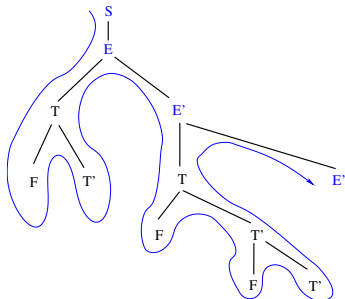
Descenso recursivo con retroceso o *backtracking*

Implementación

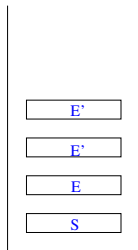
Ejemplo (Gramática de expresiones aritméticas) 97 / 113

- 2ª activación de la función asociada al símbolo E'

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E': lógico)

98 / 113

inicio

```
referencia ← componente léxico actual
```

```
{ SIMULACIÓN DE (3)  $E' \rightarrow + T E'$  }
```

```
si emparejar ("+" ) = VERDADERO entonces
```

```
  avanzar_entrada
```

```
  si T = VERDADERO entonces
```

```
    si E' = VERDADERO entonces
```

```
      DEVOLVER VERDADERO
```

```
    si no { empareja "+" }
```

```
      retroceder_entrada(referencia)
```

```
    fin_si
```

```
  ...
```

```
  ...
```

```
  si no
```

```
    retroceder_entrada(referencia)
```

```
  fin si
```

```
si no
```

```
  retroceder_entrada(referencia)
```

```
fin si
```

```
{ SIMULACIÓN DE (4)  $E' \rightarrow \epsilon$  }
```

```
DEVOLVER VERDADERO
```

```
fin
```

Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E' : lógico

99 / 113)

inicio

referencia \leftarrow *componente léxico actual*

{ SIMULACIÓN DE (3) $E' \rightarrow + T E'$ }

si *emparejar* (“+”) = *VERDADERO* entonces

avanzar_entrada

si $T = \text{VERDADERO}$ entonces

si $E' = \text{VERDADERO}$ entonces

DEVOLVER *VERDADERO*

si no { *empareja* “+” }

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no

retroceder_entrada(*referencia*)

fin si

{ SIMULACIÓN DE (4) $E' \rightarrow \epsilon$ }

DEVOLVER *VERDADERO*

fin

Descenso recursivo con retroceso o *backtracking*

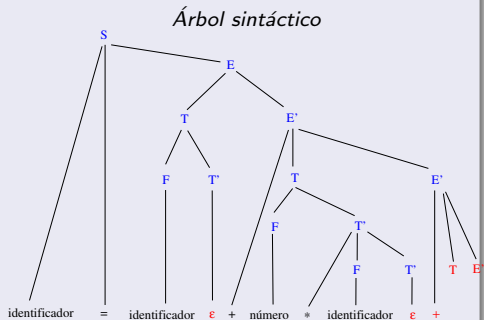
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 100 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * id T'E'$
 $\xrightarrow{9} id = id \epsilon + n * id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + n * id \epsilon + TE'$
 $\xrightarrow{3}$

Error al emparejar



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E': lógico)

101 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (3) $E' \rightarrow + T E'$ }

si emparejar ("+") = VERDADERO entonces

avanzar_entrada

si $T = \text{VERDADERO}$ entonces

si $E' = \text{VERDADERO}$ entonces

DEVOLVER VERDADERO

si no

retroceder_entrada(*referencia*)

fin_si

...

...

si no

retroceder_entrada(*referencia*)

fin si

si no {empareja "+" }

retroceder_entrada(*referencia*)

fin si

{SIMULACIÓN DE (4) $E' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

Descenso recursivo con retroceso o *backtracking*

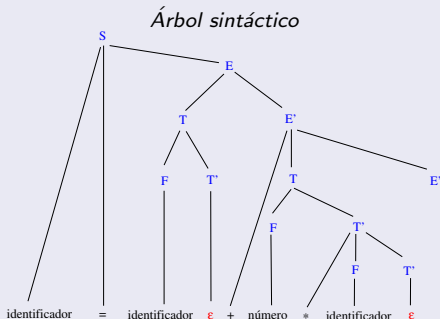
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 102 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\Rightarrow_1 id = TE'$
 $\Rightarrow_2 id = FT'E'$
 $\Rightarrow_5 id = idT'E'$
 $\Rightarrow_9 id = id \epsilon E'$
 $\Rightarrow_7 id = id \epsilon + TE'$
 $\Rightarrow_3 id = id \epsilon + FT'E'$
 $\Rightarrow_5 id = id \epsilon + n T'E'$
 $\Rightarrow_{10} id = id \epsilon + n * FT'E'$
 $\Rightarrow_6 id = id \epsilon + n * id T'E'$
 $\Rightarrow_9 id = id \epsilon + n * id \epsilon E'$
 \Rightarrow_7

Retroceso



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E': lógico)

103 / 113

inicio

referencia ← *componente léxico actual*

{ SIMULACIÓN DE (3) $E' \rightarrow + T E'$ }

si *emparejar* ("+") = VERDADERO entonces

avanzar_entrada

 si *T* = VERDADERO entonces

 si *E'* = VERDADERO entonces

 DEVOLVER VERDADERO

 si no

retroceder_entrada(*referencia*)

 fin_si

 ...

 ...

 si no

retroceder_entrada(*referencia*)

 fin si

si no { *empareja* "+" }

retroceder_entrada(*referencia*)

fin si

{ SIMULACIÓN DE (4) $E' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

Descenso recursivo con retroceso o *backtracking*

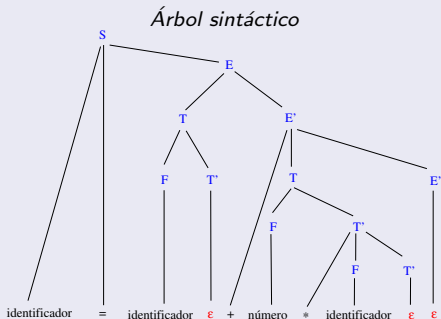
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 104 / 113)

Análisis de $\text{id} = \text{id} + \text{n} * \text{id}$

S

\Rightarrow	1	$\text{id} = E$
\Rightarrow	2	$\text{id} = TE'$
\Rightarrow	5	$\text{id} = FT'E'$
\Rightarrow	9	$\text{id} = \text{id}T'E'$
\Rightarrow	7	$\text{id} = \text{id} \epsilon E'$
\Rightarrow	3	$\text{id} = \text{id} \epsilon + TE'$
\Rightarrow	5	$\text{id} = \text{id} \epsilon + FT'E'$
\Rightarrow	10	$\text{id} = \text{id} \epsilon + \text{n}T'E'$
\Rightarrow	6	$\text{id} = \text{id} \epsilon + \text{n} * FT'E'$
\Rightarrow	9	$\text{id} = \text{id} \epsilon + \text{n} * \text{id}T'E'$
\Rightarrow	7	$\text{id} = \text{id} \epsilon + \text{n} * \text{id} \epsilon E'$
\Rightarrow	4	$\text{id} = \text{id} \epsilon + \text{n} * \text{id} \epsilon \epsilon$



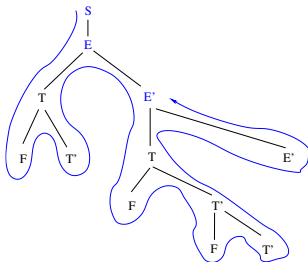
Descenso recursivo con retroceso o *backtracking*

Implementación

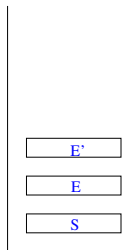
Ejemplo (Gramática de expresiones aritméticas 105 / 113)

- Fin de la 2ª activación de la función asociada al símbolo E'
- Regreso a la 1ª activación de la función asociada al símbolo E'

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E': lógico)

106 / 113

inicio

referencia ← *componente léxico actual*

{SIMULACIÓN DE (3) $E' \rightarrow + T E'$ }

si *emparejar* ("+") = VERDADERO entonces

avanzar_entrada

 si *T* = VERDADERO entonces

 si *E'* = VERDADERO entonces

 DEVOLVER VERDADERO

 si no

retroceder_entrada(*referencia*)

 fin_si

 ...

 ...

 si no

retroceder_entrada(*referencia*)

 fin_si

si no {*empareja* "+" }

retroceder_entrada(*referencia*)

fin_si

{SIMULACIÓN DE (4) $E' \rightarrow \epsilon$ }

DEVOLVER VERDADERO

fin

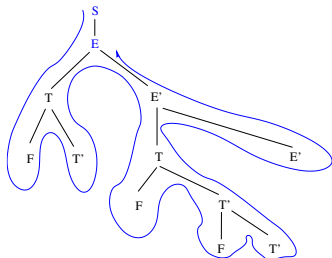
Descenso recursivo con retroceso o *backtracking*

Implementación

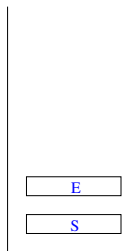
Ejemplo (Gramática de expresiones aritméticas 107 / 113)

- Fin de la 1ª activación de la función asociada al símbolo E'
- Regreso a la llamada a la función asociada al símbolo E

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función E: lógico

108 / 113)

inicio

{SIMULACIÓN DE (2) $E \rightarrow T E'$ }

si $T = \text{VERDADERO}$ entonces

si $E' = \text{VERDADERO}$ entonces

DEVOLVER VERDADERO

si no

DEVOLVER FALSO

fin si

si no

DEVOLVER FALSO

fin si

fin

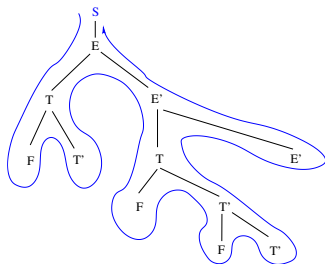
Descenso recursivo con retroceso o *backtracking*

Implementación

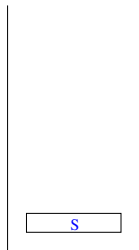
Ejemplo (Gramática de expresiones aritméticas 109 / 113)

- Fin de la llamada a la función asociada al símbolo E
- Regreso a la llamada a la función asociada al símbolo S

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

Implementación

Ejemplo (Función S: lógico)

110 / 113

inicio

{ SIMULACIÓN DE (1) $S \rightarrow \text{identificador} = E$ }

si *emparejar*(identificador) = VERDADERO entonces

avanzar_entrada

 si *emparejar*("=") = VERDADERO entonces

avanzar_entrada

 si $E = \text{VERDADERO}$ entonces

 DEVOLVER VERDADERO

 si no

 DEVOLVER FALSO

 fin si

...

...

si no { *empareja* "=" }

 DEVOLVER FALSO

fin si

si no { *empareja* identificador }

 DEVOLVER FALSO

fin si

fin

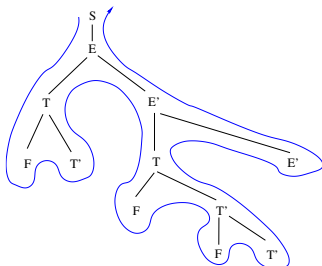
Descenso recursivo con retroceso o *backtracking*

Implementación

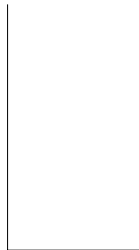
Ejemplo (Gramática de expresiones aritméticas 111 / 113)

- Fin de la llamada a la función asociada al símbolo inicial *S*

Árbol de activación



Pila de activación



Descenso recursivo con retroceso o *backtracking*

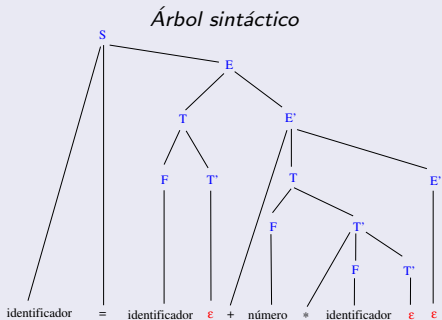
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 112 / 113)

Análisis de $id = id + n * id$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id\epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id\epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id\epsilon + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id\epsilon + nT'E'$
 $\xrightarrow{10}$
 $\Rightarrow id = id\epsilon + n * FT'E'$
 $\xrightarrow{6}$
 $\Rightarrow id = id\epsilon + n * idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id\epsilon + n * id\epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id\epsilon + n * id\epsilon\epsilon$
 $\xrightarrow{4}$

Cadena analizada

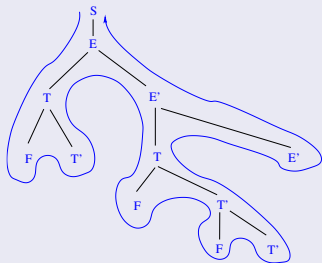


Descenso recursivo con retroceso o *backtracking*

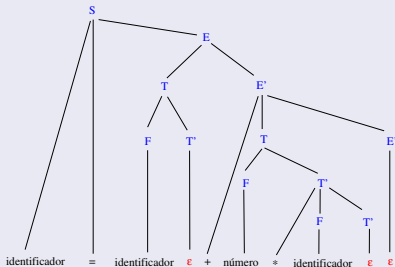
Implementación

Ejemplo (2.- Gramática de expresiones aritméticas 113 / 113)

Árbol de activación



Árbol sintáctico



Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
 - Descripción
 - Implementación
 - Limitaciones o inconvenientes
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Descenso recursivo con retroceso o *backtracking*

Limitaciones o inconvenientes

Limitaciones o inconvenientes

1 / 2

1.- Ineficiencia computacional

- Las **reglas** de la gramática siempre se **simulan** según un **orden preestablecido**, que puede provocar numerosos **retrocesos** para probar otras reglas.
- Si se ha **generado código** durante el análisis entonces dicho código debe ser **descartado** si se produce un **retroceso**.

Descenso recursivo con retroceso o *backtracking*

Limitaciones o inconvenientes

Limitaciones o inconvenientes

2 / 2

2.- No localización de errores

- Si la cadena de entrada es incorrecta, **no** localiza la ubicación del error, ya que siempre termina en la función asociada del símbolo inicial.

3.- Riesgo de recursión infinita

- Las funciones pueden provocar una **recursión infinita** si la gramática posee **recursividad inmediata** o **general** por la **izquierda**.

Descenso recursivo con retroceso o *backtracking*

Limitaciones o inconvenientes

Ejemplos (Gramáticas con recursividad por la izquierda)

- 1.- Gramática con recursividad *inmediata* por la izquierda.
- 2.- Gramática con recursividad *general* por la izquierda.

Descenso recursivo con retroceso o *backtracking*

Limitaciones o inconvenientes

Ejemplo (1.- Recursividad inmediata por la izquierda 1 / 2)

- $$P = \{$$
- (1) $S \rightarrow L E$
 - (2) $L \rightarrow L \text{ identificador} =$
 - (3) $L \rightarrow \text{identificador} =$
 - (4) $E \rightarrow E + T$
 - (5) $E \rightarrow T$
 - (6) $T \rightarrow T * F$
 - (7) $T \rightarrow F$
 - (8) $F \rightarrow (E)$
 - (9) $F \rightarrow \text{identificador}$
 - (10) $F \rightarrow \text{número}$
- $$\}$$

- *Esta gramática genera sentencias de asignación múltiple.*

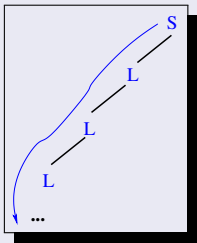
Descenso recursivo con retroceso o *backtracking*

Limitaciones o inconvenientes

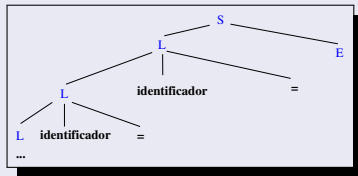
Ejemplo (1.- Recursividad inmediata por la izquierda 2 / 2)

- Se genera una *recursión infinita* al analizar la sentencia:
identificador = identificador = número

Árbol de activación



Árbol sintáctico



Descenso recursivo con retroceso o *backtracking*

Limitaciones o inconvenientes

Ejemplo (2.- Recursividad general por la izquierda 1 / 2)

$$P = \{$$

- (1) $S \rightarrow A a$
- (2) $A \rightarrow S b$
- (3) $A \rightarrow c$

$$\}$$

- Esta gramática genera el lenguaje $L(G) = c(ba)^*a$

Descenso recursivo con retroceso o *backtracking*

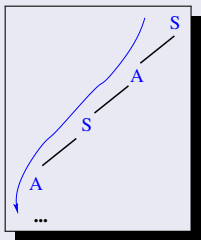
Limitaciones o inconvenientes

Ejemplo (1.- Recursividad inmediata por la izquierda 2 / 2)

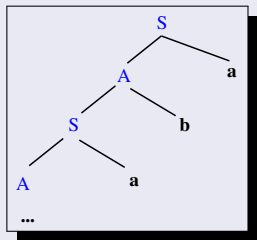
- Se genera una **recursión infinita** al analizar la sentencia:

c b a a

Árbol de activación



Árbol sintáctico



Contenido del tema

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo**
- 4 Detección y recuperación de errores

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
 - Descripción
 - Gramáticas LL(k)
 - Fases
 - Conjunto Primero
 - Conjunto Siguierte
 - Construcción de la tabla predictiva
 - Conflictos en la tabla predictiva
 - Implementación recursiva
 - Implementación iterativa

Análisis sintáctico descendente predictivo

Descripción

Objetivo

- Comprobar si la gramática puede **generar** la cadena de entrada mediante
 - la generación de una **derivación por la izquierda**
 - o la construcción del **árbol sintáctico** de forma **descendente**.
- Utiliza una **tabla predictiva** para determinar **qué regla de producción** (si existe) se puede usar en cada paso

Análisis sintáctico descendente predictivo

Descripción

Notas (Tabla predictiva)

- La **tabla predictiva** indica cuál es la *única* regla de producción (si existe) que permite continuar el análisis.
- Si la tabla predictiva no indica *ninguna* regla entonces la cadena de entrada **no** puede ser generada por la gramática.

Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

1 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3		4		4
T	5				5	5	
T'			7	6	7		7
F	9				8	10	

Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas 2 / 27)

- *Análisis mediante descenso recursivo con retroceso:*

identificador = identificador + número * identificador \$

Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas

3 / 27)

Análisis de **id = id + n * id \$**

S

Árbol sintáctico

S

Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

4 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3		4		4
T	5				5	5	
T'			7	6	7		7
F	9				8	10	

Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

5 / 27

Análisis de **id = id + n * id \$**

$S \xRightarrow{1} \underline{id = E}$

Emparejamientos: *id*, =

Árbol sintáctico



Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

6 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2		2
E'			3			4	4
T	5				5		5
T'			7	6		7	7
F	9				8		10

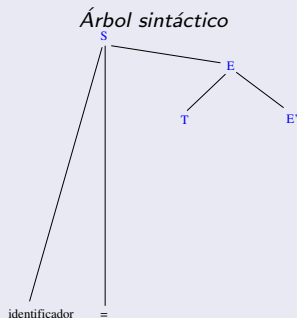
Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

7 / 27

Análisis de **id = id + n * id \$**

$$\begin{array}{l}
 S \Rightarrow \text{id} = E \\
 \quad \underset{1}{\Rightarrow} \\
 \quad \underset{2}{\Rightarrow} \text{id} = \underline{TE'}
 \end{array}$$


Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

8 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3		4		4
T	5				5	5	
T'			7	6	7		7
F	9				8	10	

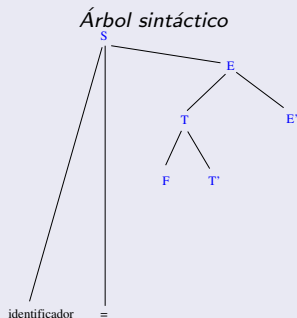
Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

9 / 27

Análisis de **id = id + n * id \$**

$$\begin{array}{l}
 S \Rightarrow id = E \\
 \quad \underset{1}{\Rightarrow} \\
 \quad \Rightarrow id = TE' \\
 \quad \quad \underset{2}{\Rightarrow} \\
 \quad \quad \Rightarrow id = \underline{FT'E'} \\
 \quad \quad \quad \underset{5}{\Rightarrow}
 \end{array}$$


Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

10 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2		2
E'			3			4	4
T	5				5		5
T'			7	6		7	7
F	9				8		10

Análisis sintáctico descendente predictivo

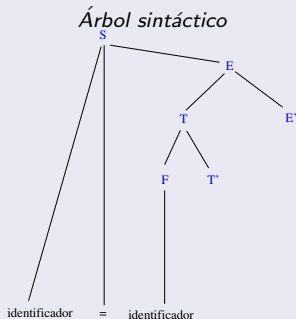
Descripción

Ejemplo (Gramática de expresiones aritméticas)

11 / 27

Análisis de **id = id + n * id \$**

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$

Emparejamiento de *id*

Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

12 / 27)

 $P = \{$

(1) $S \rightarrow \text{identificador} = E$

(2) $E \rightarrow T E'$

(3) $E' \rightarrow + T E'$

(4) $E' \rightarrow \epsilon$

(5) $T \rightarrow F T'$

(6) $T' \rightarrow * F T'$

(7) $T' \rightarrow \epsilon$

(8) $F \rightarrow (E)$

(9) $F \rightarrow \text{identificador}$

(10) $F \rightarrow \text{número}$

 $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3			4	4
T	5				5	5	
T'			7	6		7	7
F	9				8	10	

Análisis sintáctico descendente predictivo

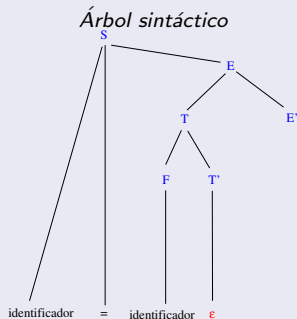
Descripción

Ejemplo (Gramática de expresiones aritméticas)

13 / 27

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $id = TE'$
 $\xrightarrow{2}$
 $id = FT'E'$
 $\xrightarrow{5}$
 $id = id T'E'$
 $\xrightarrow{9}$
 $id = id \epsilon E'$
 $\xrightarrow{7}$



Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

14 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3			4	4
T	5				5	5	
T'			7	6		7	7
F	9				8	10	

Análisis sintáctico descendente predictivo

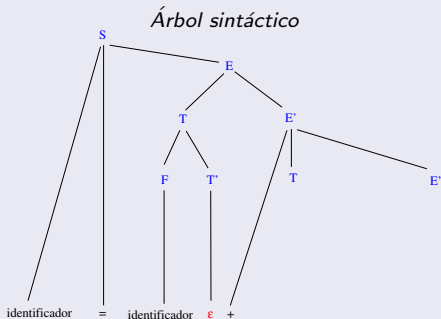
Descripción

Ejemplo (Gramática de expresiones aritméticas)

15 / 27

Análisis de $\text{id} = \text{id} + \text{n} * \text{id} \$$

$S \Rightarrow \text{id} = E$
 $\xrightarrow{1}$
 $\Rightarrow \text{id} = TE'$
 $\xrightarrow{2}$
 $\Rightarrow \text{id} = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow \text{id} = \text{id} T'E'$
 $\xrightarrow{9}$
 $\Rightarrow \text{id} = \text{id} \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow \text{id} = \text{id} \epsilon \underline{+TE'}$
 $\xrightarrow{3}$

Emparejamiento de $+$ 

Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

16 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3			4	4
T	5				5	5	
T'			7	6		7	7
F	9				8	10	

Análisis sintáctico descendente predictivo

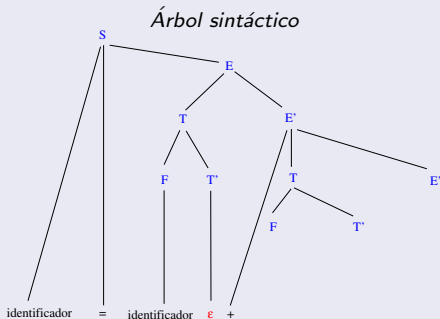
Descripción

Ejemplo (Gramática de expresiones aritméticas)

17 / 27

Análisis de $\text{id} = \text{id} + \text{n} * \text{id} \$$

$S \Rightarrow \text{id} = E$
 $\xrightarrow{1}$
 $\Rightarrow \text{id} = TE'$
 $\xrightarrow{2}$
 $\Rightarrow \text{id} = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow \text{id} = \text{id}T'E'$
 $\xrightarrow{9}$
 $\Rightarrow \text{id} = \text{id} \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow \text{id} = \text{id} \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow \text{id} = \text{id} \epsilon + \underline{FT'E'}$
 $\xrightarrow{5}$



Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

18 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2		2
E'			3			4	4
T	5				5		5
T'			7	6		7	7
F	9				8		10

Análisis sintáctico descendente predictivo

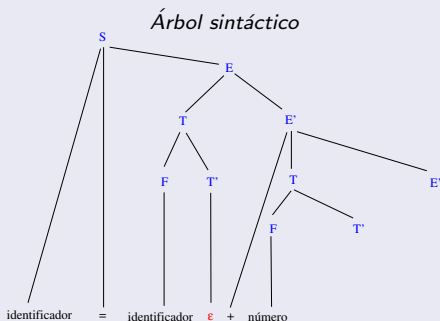
Descripción

Ejemplo (Gramática de expresiones aritméticas)

19 / 27

Análisis de $id = id + n * id \$$

S	\Rightarrow	$id = E$
1	\Rightarrow	$id = TE'$
2	\Rightarrow	$id = FT'E'$
5	\Rightarrow	$id = idT'E'$
9	\Rightarrow	$id = id \epsilon E'$
7	\Rightarrow	$id = id \epsilon + TE'$
3	\Rightarrow	$id = id \epsilon + FT'E'$
5	\Rightarrow	$id = id \epsilon + nT'E'$
10	\Rightarrow	$id = id \epsilon + nT'E'$

Emparejamiento de n 

Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

20 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada							
	id	=	+	*	()	número	\$	
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F	9				8		10	

Análisis sintáctico descendente predictivo

Descripción

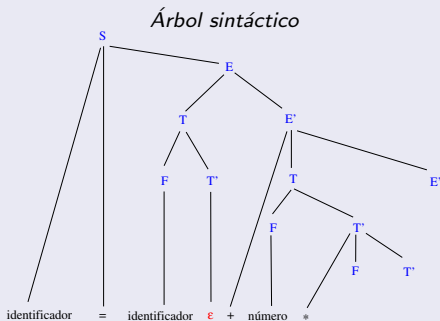
Ejemplo (Gramática de expresiones aritméticas)

21 / 27

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id \epsilon + nT'E'$
 $\xrightarrow{10}$
 $\Rightarrow id = id \epsilon + n * FT'E'$
 $\xrightarrow{6}$

Emparejamiento de *



Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

22 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2		2
E'			3			4	4
T	5				5		5
T'			7	6		7	7
F	9				8		10

Análisis sintáctico descendente predictivo

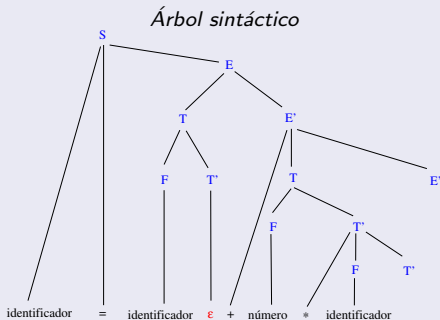
Descripción

Ejemplo (Gramática de expresiones aritméticas)

23 / 27)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * \underline{id}T'E'$
 $\xrightarrow{9}$

Emparejamiento de id 

Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

24 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3			4	4
T	5				5	5	
T'			7	6		7	7
F	9				8	10	

Análisis sintáctico descendente predictivo

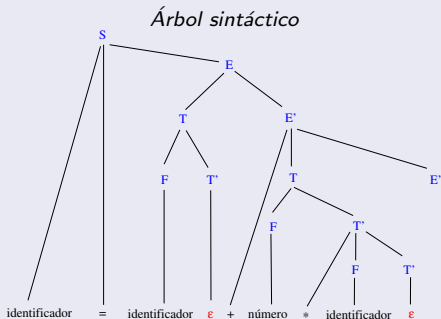
Descripción

Ejemplo (Gramática de expresiones aritméticas)

25 / 27)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * id T'E'$
 $\xrightarrow{9} id = id \epsilon + n * id \epsilon E'$
 $\xrightarrow{7}$



Análisis sintáctico descendente predictivo

Descripción

Ejemplo (Gramática de expresiones aritméticas)

26 / 27)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3		4		4
T	5				5	5	
T'			7	6	7		7
F	9				8	10	

Análisis sintáctico descendente predictivo

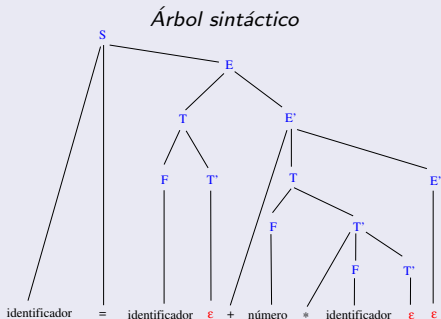
Descripción

Ejemplo (Gramática de expresiones aritméticas)

27 / 27

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \in E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \in + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \in + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id \in + nT'E'$
 $\xrightarrow{10}$
 $\Rightarrow id = id \in + n * FT'E'$
 $\xrightarrow{6}$
 $\Rightarrow id = id \in + n * idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \in + n * id \in E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \in + n * id \in \in$
 $\xrightarrow{4}$



Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
 - Descripción
 - Gramáticas LL(k)
 - Fases
 - Conjunto Primero
 - Conjunto Siguierte
 - Construcción de la tabla predictiva
 - Conflictos en la tabla predictiva
 - Implementación recursiva
 - Implementación iterativa

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Definición (Gramática LL(k))

- Una *gramática LL(k)* es una gramática de contexto libre que **admite** un análisis descendente *predictivo*

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Significado

- LL(k): procesa la **cadena de entrada** de **izquierda a derecha**.
- LL(k): genera una **derivación** por la **izquierda**.
- LL(k): **número** de componentes léxicos que explora de la cadena de entrada para determinar qué regla se va a procesar en cada paso de la derivación.

Nota

Generalmente, k vale 1.

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Condiciones de una gramática LL(1)

- Para cada par de reglas $A \rightarrow \alpha | \beta$ se verifican las siguientes condiciones:
 - 1.- **No** existe ningún símbolo terminal $a \in V_T$ que pueda ser el primer símbolo de una cadena derivable por α y β , es decir, $\text{Primero}(\alpha) \cap \text{Primero}(\beta) = \emptyset$
 - 2.- La palabra vacía ϵ solamente puede ser derivada, **a lo sumo**, o por α o por β , pero **no** por ambas simultáneamente, es decir, si $\epsilon \in \text{Primero}(\alpha)$ entonces $\epsilon \notin \text{Primero}(\beta)$ y viceversa.
 - 3.- Si β (o α) deriva la palabra vacía ϵ entonces α (o β) **no** puede derivar un cadena que comience por un símbolo terminal $a \in V_T$ que pertenezca al conjunto **Siguiente(A)**, es decir, si $\epsilon \in \text{Primero}(\beta)$ entonces $\text{Primero}(\alpha) \cap \text{Siguiente}(A) = \emptyset$

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Primera condición de una gramática LL(1)

1 / 3

- Para cada par de reglas $A \rightarrow \alpha | \beta$ se verifica que
 - 1.- **No** existe ningún símbolo terminal $a \in V_T$ que pueda ser el primer símbolo de una cadena derivable por α y β , es decir, $\text{Primero}(\alpha) \cap \text{Primero}(\beta) = \emptyset$

Si $A \rightarrow \alpha | \beta \in P$

entonces $\nexists a \in V_T$ tal que

$$\begin{cases} A \Rightarrow \alpha \overset{*}{\Rightarrow} a \delta \\ A \Rightarrow \beta \overset{*}{\Rightarrow} a \gamma \end{cases}$$

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Ejemplo (Primera condición de una gramática LL(1) 2 / 3)

- Gramática que genera sentencias condicionales

$$P = \{$$

$$(1) S \rightarrow \text{si } C \text{ entonces } S \text{ si_no } S \text{ fin_si} \quad (S \rightarrow \alpha)$$

$$(2) S \rightarrow \text{si } C \text{ entonces } S \text{ fin_si} \quad (S \rightarrow \beta)$$

$$(3) S \rightarrow \dots$$

$$(4) C \rightarrow \dots$$

$$\}$$

$$\text{Primero}(\alpha) \cap \text{Primero}(\beta) =$$

$$= \text{Primero}(\text{si } C \text{ entonces } S \text{ si_no } S \text{ fin_si}) \cap \text{Primero}(\text{si } C \text{ entonces } S \text{ fin_si})$$

$$= \{\text{si}\} \neq \emptyset$$

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Ejemplo (Primera condición de una gramática LL(1) 3 / 3)

- Gramática que genera sentencias condicionales

$$P = \left\{ \begin{array}{l} (1) S \rightarrow \text{si } C \text{ entonces } S \text{ si_no } S \text{ fin_si} \quad (S \rightarrow \alpha) \\ (2) S \rightarrow \text{si } C \text{ entonces } S \text{ fin_si} \quad (S \rightarrow \beta) \\ (3) S \rightarrow \dots \\ (4) C \rightarrow \dots \\ \} \end{array} \right.$$

- **Conflicto** al analizar una sentencia que empieza por **si**

- Primera opción

$$S \xRightarrow{1} \text{si } C \text{ entonces } S \text{ si_no } S \text{ fin_si}$$

- Segunda opción

$$S \xRightarrow{2} \text{si } C \text{ entonces } S \text{ fin_si}$$

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Segunda condición de una gramática LL(1)

1 / 3

- Para cada par de reglas $A \rightarrow \alpha|\beta$ se verifica que
 - 2.- la palabra vacía ϵ solamente puede ser derivada, **a lo sumo**, o por α o por β , pero no por ambas simultáneamente, es decir, si $\epsilon \in \text{Primero}(\alpha)$ entonces $\epsilon \notin \text{Primero}(\beta)$ y viceversa.

Si $A \rightarrow \alpha|\beta \in P$

entonces

- si $A \Rightarrow \alpha \xrightarrow{*} \epsilon$ entonces $\nexists A \Rightarrow \beta \xrightarrow{*} \epsilon$
- si $A \Rightarrow \beta \xrightarrow{*} \epsilon$ entonces $\nexists A \Rightarrow \alpha \xrightarrow{*} \epsilon$

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Ejemplo (Segunda condición de una gramática LL(1) 2 / 3)

- Gramática

$$P = \{$$

$$(1) S \rightarrow \epsilon \quad (S \rightarrow \alpha)$$

$$(2) S \rightarrow A \quad (S \rightarrow \beta)$$

$$(3) A \rightarrow \epsilon$$

$$(4) A \rightarrow \dots$$

$$\}$$

$$\epsilon \in \text{Primero}(\alpha) \wedge \epsilon \in \text{Primero}(\beta)$$

$$\epsilon \in \text{Primero}(\epsilon) \wedge \epsilon \in \text{Primero}(A)$$

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Ejemplo (Segunda condición de una gramática LL(1) 3 / 3)

- Gramática

$$P = \{$$

$$(1) S \rightarrow \epsilon$$

$$(2) S \rightarrow A$$

$$(3) A \rightarrow \epsilon$$

$$(4) A \rightarrow \dots$$

$$\}$$

- Conflicto** al derivar la palabra vacía ϵ :

- Primera opción

$$S \xRightarrow{1} \epsilon$$

- Segunda opción

$$S \xRightarrow{2} A \xRightarrow{3} \epsilon$$

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Tercera condición de una gramática LL(1)

1 / 3

- Para cada par de reglas $A \rightarrow \alpha|\beta$ se verifica que
 - 3.- Si β (o α) deriva la palabra vacía ϵ entonces α (o β) **no** puede derivar un cadena que comience por un símbolo terminal $\mathbf{a} \in V_T$ que pertenezca al conjunto **Siguiente(A)**, es decir, si $\epsilon \in \text{Primero}(\beta)$ entonces $\text{Primero}(\alpha) \cap \text{Siguiete}(A) = \emptyset$

Si $A \rightarrow \alpha|\beta \in P$ y $\beta \xrightarrow{*} \epsilon$ entonces $\#A \Rightarrow \alpha \xrightarrow{*} \mathbf{a} \gamma$
 donde $\mathbf{a} \in \text{Siguiete}(A)$

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Ejemplo (Tercera condición de una gramática LL(1) 2 / 3)

- Gramática

$$P = \{$$

$$(1) S \rightarrow A a$$

$$(2) A \rightarrow B c \quad (A \rightarrow \alpha)$$

$$(3) A \rightarrow \epsilon \quad (A \rightarrow \beta)$$

$$(4) B \rightarrow a$$

$$\}$$

$$\epsilon \in \text{Primero}(\beta) = \text{Primero}(\epsilon)$$

$$\text{Primero}(Bc) \cap \text{Siguiente}(A) = \{a\} \neq \emptyset$$

Análisis sintáctico descendente predictivo

Gramáticas LL(k)

Ejemplo (Tercera condición de una gramática LL(1)) 3 / 3

- Gramática

$$P = \{$$

$$(1) S \rightarrow A a$$

$$(2) A \rightarrow B c \quad (A \rightarrow \alpha)$$

$$(3) A \rightarrow \epsilon \quad (A \rightarrow \beta)$$

$$(4) B \rightarrow a$$

$$\}$$

- Conflicto** al derivar una cadena que comienza por **a**:

- Primera opción

$$S \xRightarrow{1} A a \xRightarrow{3} a$$

- Segunda opción

$$S \xRightarrow{1} A a \xRightarrow{2} B c a \xRightarrow{4} a c a$$

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
 - Descripción
 - Gramáticas LL(k)
 - Fases
 - Conjunto Primero
 - Conjunto Siguierte
 - Construcción de la tabla predictiva
 - Conflictos en la tabla predictiva
 - Implementación recursiva
 - Implementación iterativa

Análisis sintáctico descendente predictivo

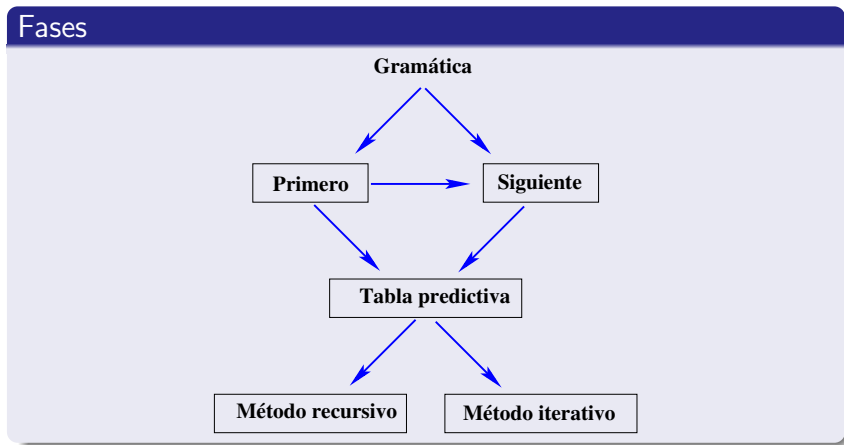
Fases

Fases

- 1.- Construcción del conjunto **Primero**.
- 2.- Construcción del conjunto **Siguiente**.
- 3.- Construcción de la **tabla predictiva**.
- 4.- Uso de la **tabla predictiva** para aplicar un **método** de análisis descendente predictivo
 - **Método recursivo**: basado en funciones recursivas, pero sin retroceso en la cadena de entrada.
 - **Método iterativo**.

Análisis sintáctico descendente predictivo

Fases



Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
 - Descripción
 - Gramáticas LL(k)
 - Fases
 - **Conjunto Primero**
 - Conjunto Siguierte
 - Construcción de la tabla predictiva
 - Conflictos en la tabla predictiva
 - Implementación recursiva
 - Implementación iterativa

Análisis sintáctico descendente predictivo

Conjunto Primero

Definición (Primero(α))

- Si $\alpha \in V^* = (V_N \cup V_T)^*$ entonces

$$\text{Primero}(\alpha) = \{\mathbf{a} \mid \mathbf{a} \in V_T \wedge \alpha \xRightarrow{*} \mathbf{a} \gamma\} \cup \{\epsilon \mid \alpha \xRightarrow{*} \epsilon\}$$

Análisis sintáctico descendente predictivo

Conjunto Primero

Nota ($\text{Primero}(\alpha)$)

- $\text{Primero}(\alpha)$ va a estar compuesto por
 - los símbolos **terminales** que aparecen en la *primera* posición de una cadena que es derivable a partir de α
 - y la palabra vacía ϵ , si α la deriva.

Análisis sintáctico descendente predictivo

Conjunto Primero

Algoritmo (Construcción del conjunto Primero

1 / 5)

- *Caso básico 1:* $\alpha = \epsilon$
- *Caso básico 2:* $\alpha = \mathbf{a} \in V_T$
- *Caso básico 3:* $\alpha = A \in V_N$
- *Caso general:* $\alpha = X_1 X_2 \dots X_n \in V^+ = (V_N \cup V_T)^+$

Análisis sintáctico descendente predictivo

Conjunto Primero

Algoritmo (Construcción del conjunto Primero)

2 / 5

- **Caso básico 1:** $\alpha = \epsilon$

$$\text{Primero}(\alpha) = \text{Primero}(\epsilon) = \{\epsilon\}$$

$$\text{porque } \alpha = \epsilon \xrightarrow{0} \epsilon$$

Análisis sintáctico descendente predictivo

Conjunto Primero

Algoritmo (Construcción del conjunto Primero

3 / 5)

- **Caso básico 2:** $\alpha = \mathbf{a} \in V_T$
 $\text{Primero}(\alpha) = \text{Primero}(\mathbf{a}) = \{\mathbf{a}\}$
porque $\alpha = \mathbf{a} \xrightarrow{0} \mathbf{a}$

Análisis sintáctico descendente predictivo

Conjunto Primero

Algoritmo (Construcción del conjunto Primero)

4 / 5

- **Caso básico 3:** $\alpha = A \in V_N$
 - a) Si $A \rightarrow \epsilon \in P$ entonces $\epsilon \in \text{Primero}(A)$
 - b) Si $A \rightarrow X_1 X_2 \dots X_n \in P$
 - entonces $\text{Primero}(X_1) - \{\epsilon\} \subseteq \text{Primero}(A)$
 - Si $\epsilon \in \text{Primero}(X_i) \forall i \in \{1, 2, \dots, k-1\}$ entonces $\text{Primero}(X_k) - \{\epsilon\} \subseteq \text{Primero}(A)$
 - Si $\epsilon \in \text{Primero}(X_i) \forall i \in \{1, 2, \dots, n\}$ entonces $\epsilon \in \text{Primero}(A)$

Análisis sintáctico descendente predictivo

Conjunto Primero

Algoritmo (Construcción del conjunto Primero)

5 / 5

- **Caso general:** $\alpha = X_1 X_2 \dots X_n \in V^+ = (V_N \cup V_T)^+$
 - $\text{Primero}(X_1) - \{\epsilon\} \subseteq \text{Primero}(\alpha)$
 - Si $\epsilon \in \text{Primero}(X_i) \forall i \in \{1, 2, \dots, k-1\}$ entonces $\text{Primero}(X_k) - \{\epsilon\} \subseteq \text{Primero}(\alpha)$
 - Si $\epsilon \in \text{Primero}(X_i) \forall i \in \{1, 2, \dots, n\}$ entonces $\epsilon \in \text{Primero}(\alpha)$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplos (Construcción del conjunto Primero)

- 1.- *Gramática de las expresiones aritméticas.*
- 2.- *Gramática de las declaraciones.*

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (1.- Gramática de las expresiones aritméticas 1 / 2)

- $$P = \{$$
- (1) $S \rightarrow \text{identificador} = E$
 - (2) $E \rightarrow T E'$
 - (3) $E' \rightarrow + T E'$
 - (4) $E' \rightarrow \epsilon$
 - (5) $T \rightarrow F T'$
 - (6) $T' \rightarrow * F T'$
 - (7) $T' \rightarrow \epsilon$
 - (8) $F \rightarrow (E)$
 - (9) $F \rightarrow \text{identificador}$
 - (10) $F \rightarrow \text{número}$
- $$\}$$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 2)

• $\text{id} \in \text{Primero}(S)$ por (1) $S \rightarrow \text{identificador} = E$

• $\text{Primero}(T) = \{ \epsilon \} \subseteq \text{Primero}(E)$

por (2) $E \rightarrow T E'$

• $+ \in \text{Primero}(E')$ por (3) $E' \rightarrow + T E'$

• $\epsilon \in \text{Primero}(E')$ por (4) $E' \rightarrow \epsilon$

• $\text{Primero}(F) = \{ \epsilon \} \subseteq \text{Primero}(T)$

por (5) $T \rightarrow F T'$

• $* \in \text{Primero}(T')$ por (6) $T' \rightarrow * F T'$

• $\epsilon \in \text{Primero}(T')$ por (7) $T' \rightarrow \epsilon$

• $"(" \in \text{Primero}(F)$ por (8) $F \rightarrow (E)$

• $\text{id} \in \text{Primero}(F)$ por (9) $F \rightarrow \text{id}$

• $n \in \text{Primero}(F)$ por (10) $F \rightarrow n$

	Primero
S	id
E	"(", id
E'	+, ϵ
T	"(", id, n
T'	*, ϵ
F	"(", id, n

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 2)

• $\text{id} \in \text{Primero}(S)$ por (1) $S \rightarrow \text{identificador} = E$

• $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(E)$

por (2) $E \rightarrow T E'$

• $+$ $\in \text{Primero}(E')$ por (3) $E' \rightarrow + T E'$

• $\epsilon \in \text{Primero}(E')$ por (4) $E' \rightarrow \epsilon$

• $\text{Primero}(F) - \{\epsilon\} \subseteq \text{Primero}(T)$

por (5) $T \rightarrow F T'$

• $*$ $\in \text{Primero}(T')$ por (6) $T' \rightarrow * F T'$

• $\epsilon \in \text{Primero}(T')$ por (7) $T' \rightarrow \epsilon$

• $"(" \in \text{Primero}(F)$ por (8) $F \rightarrow (E)$

• $\text{id} \in \text{Primero}(F)$ por (9) $F \rightarrow \text{id}$

• $n \in \text{Primero}(F)$ por (10) $F \rightarrow n$

	Primero
S	id
E	$"(", \text{id}$
E'	$+, \epsilon$
T	$+, \text{id}, n$
T'	$*, \epsilon$
F	$"(", \text{id}, n$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 2)

- $\text{id} \in \text{Primero}(S)$ por (1) $S \rightarrow \text{identificador} = E$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(E)$
por (2) $E \rightarrow T E'$
- $+\in \text{Primero}(E')$ por (3) $E' \rightarrow + T E'$
- $\epsilon \in \text{Primero}(E')$ por (4) $E' \rightarrow \epsilon$
- $\text{Primero}(F) - \{\epsilon\} \subseteq \text{Primero}(T)$
por (5) $T \rightarrow F T'$
- $* \in \text{Primero}(T')$ por (6) $T' \rightarrow * F T'$
- $\epsilon \in \text{Primero}(T')$ por (7) $T' \rightarrow \epsilon$
- $"(" \in \text{Primero}(F)$ por (8) $F \rightarrow (E)$
- $\text{id} \in \text{Primero}(F)$ por (9) $F \rightarrow \text{id}$
- $n \in \text{Primero}(F)$ por (10) $F \rightarrow n$

	Primero
S	id
E	"(", id
E'	+
T	"(", *, n
T'	"(", *, \epsilon
F	"(", id, n

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 2)

- $\text{id} \in \text{Primero}(S)$ por (1) $S \rightarrow \text{identificador} = E$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(E)$
por (2) $E \rightarrow T E'$
- $+\in \text{Primero}(E')$ por (3) $E' \rightarrow + T E'$
- $\epsilon \in \text{Primero}(E')$ por (4) $E' \rightarrow \epsilon$
- $\text{Primero}(F) - \{\epsilon\} \subseteq \text{Primero}(T)$
por (5) $T \rightarrow F T'$
- $* \in \text{Primero}(T')$ por (6) $T' \rightarrow * F T'$
- $\epsilon \in \text{Primero}(T')$ por (7) $T' \rightarrow \epsilon$
- $"(" \in \text{Primero}(F)$ por (8) $F \rightarrow (E)$
- $\text{id} \in \text{Primero}(F)$ por (9) $F \rightarrow \text{id}$
- $n \in \text{Primero}(F)$ por (10) $F \rightarrow n$

	Primero
S	id
E	"(", id
E'	+, , ϵ
T	+, *, id, n
T'	+, *
F	"(", id, n

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 2)

- $\text{id} \in \text{Primero}(S)$ por (1) $S \rightarrow \text{identificador} = E$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(E)$
por (2) $E \rightarrow T E'$
- $+\in \text{Primero}(E')$ por (3) $E' \rightarrow + T E'$
- $\epsilon \in \text{Primero}(E')$ por (4) $E' \rightarrow \epsilon$
- $\text{Primero}(F) - \{\epsilon\} \subseteq \text{Primero}(T)$
por (5) $T \rightarrow F T'$
- $* \in \text{Primero}(T')$ por (6) $T' \rightarrow * F T'$
- $\epsilon \in \text{Primero}(T')$ por (7) $T' \rightarrow \epsilon$
- $"(" \in \text{Primero}(F)$ por (8) $F \rightarrow (E)$
- $\text{id} \in \text{Primero}(F)$ por (9) $F \rightarrow \text{id}$
- $n \in \text{Primero}(F)$ por (10) $F \rightarrow n$

	Primero
S	id
E	
E'	+ , ϵ
T	
T'	
F	

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 2)

- $\text{id} \in \text{Primero}(S)$ por (1) $S \rightarrow \text{identificador} = E$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(E)$
por (2) $E \rightarrow T E'$
- $+ \in \text{Primero}(E')$ por (3) $E' \rightarrow + T E'$
- $\epsilon \in \text{Primero}(E')$ por (4) $E' \rightarrow \epsilon$
- $\text{Primero}(F) - \{\epsilon\} \subseteq \text{Primero}(T)$
por (5) $T \rightarrow F T'$
- $* \in \text{Primero}(T')$ por (6) $T' \rightarrow * F T'$
- $\epsilon \in \text{Primero}(T')$ por (7) $T' \rightarrow \epsilon$
- $"(" \in \text{Primero}(F)$ por (8) $F \rightarrow (E)$
- $\text{id} \in \text{Primero}(F)$ por (9) $F \rightarrow \text{id}$
- $n \in \text{Primero}(F)$ por (10) $F \rightarrow n$

	Primero
S	id
E	
E'	+ , ϵ
T	
T'	*
F	

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 2)

- $\text{id} \in \text{Primero}(S)$ por (1) $S \rightarrow \text{identificador} = E$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(E)$
por (2) $E \rightarrow T E'$
- $+ \in \text{Primero}(E')$ por (3) $E' \rightarrow + T E'$
- $\epsilon \in \text{Primero}(E')$ por (4) $E' \rightarrow \epsilon$
- $\text{Primero}(F) - \{\epsilon\} \subseteq \text{Primero}(T)$
por (5) $T \rightarrow F T'$
- $* \in \text{Primero}(T')$ por (6) $T' \rightarrow * F T'$
- $\epsilon \in \text{Primero}(T')$ por (7) $T' \rightarrow \epsilon$
- $(\in \text{Primero}(F)$ por (8) $F \rightarrow (E)$
- $\text{id} \in \text{Primero}(F)$ por (9) $F \rightarrow \text{id}$
- $n \in \text{Primero}(F)$ por (10) $F \rightarrow n$

	Primero
S	id
E	
E'	$+, \epsilon$
T	
T'	$*, \epsilon$
F	

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 2)

- $\text{id} \in \text{Primero}(S)$ por (1) $S \rightarrow \text{identificador} = E$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(E)$
por (2) $E \rightarrow T E'$
- $+ \in \text{Primero}(E')$ por (3) $E' \rightarrow + T E'$
- $\epsilon \in \text{Primero}(E')$ por (4) $E' \rightarrow \epsilon$
- $\text{Primero}(F) - \{\epsilon\} \subseteq \text{Primero}(T)$
por (5) $T \rightarrow F T'$
- $* \in \text{Primero}(T')$ por (6) $T' \rightarrow * F T'$
- $\epsilon \in \text{Primero}(T')$ por (7) $T' \rightarrow \epsilon$
- $"(" \in \text{Primero}(F)$ por (8) $F \rightarrow (E)$
- $\text{id} \in \text{Primero}(F)$ por (9) $F \rightarrow \text{id}$
- $n \in \text{Primero}(F)$ por (10) $F \rightarrow n$

	Primero
S	id
E	$"(" \text{ id } , \epsilon$
E'	$+ , \epsilon$
T	$"(" \text{ id } , \epsilon$
T'	$* , \epsilon$
F	$"(" \text{ id } , n$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 2)

- $\text{id} \in \text{Primero}(S)$ por (1) $S \rightarrow \text{identificador} = E$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(E)$
por (2) $E \rightarrow T E'$
- $+ \in \text{Primero}(E')$ por (3) $E' \rightarrow + T E'$
- $\epsilon \in \text{Primero}(E')$ por (4) $E' \rightarrow \epsilon$
- $\text{Primero}(F) - \{\epsilon\} \subseteq \text{Primero}(T)$
por (5) $T \rightarrow F T'$
- $* \in \text{Primero}(T')$ por (6) $T' \rightarrow * F T'$
- $\epsilon \in \text{Primero}(T')$ por (7) $T' \rightarrow \epsilon$
- $"(" \in \text{Primero}(F)$ por (8) $F \rightarrow (E)$
- $\text{id} \in \text{Primero}(F)$ por (9) $F \rightarrow \text{id}$
- $n \in \text{Primero}(F)$ por (10) $F \rightarrow n$

	Primero
S	id
E	$"(", \text{id}$
E'	$+, \epsilon$
T	$"(", \text{id}$
T'	$*, \epsilon$
F	$"(", \text{id}$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 2)

- $\text{id} \in \text{Primero}(S)$ por (1) $S \rightarrow \text{identificador} = E$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(E)$
por (2) $E \rightarrow T E'$
- $+ \in \text{Primero}(E')$ por (3) $E' \rightarrow + T E'$
- $\epsilon \in \text{Primero}(E')$ por (4) $E' \rightarrow \epsilon$
- $\text{Primero}(F) - \{\epsilon\} \subseteq \text{Primero}(T)$
por (5) $T \rightarrow F T'$
- $* \in \text{Primero}(T')$ por (6) $T' \rightarrow * F T'$
- $\epsilon \in \text{Primero}(T')$ por (7) $T' \rightarrow \epsilon$
- $"(" \in \text{Primero}(F)$ por (8) $F \rightarrow (E)$
- $\text{id} \in \text{Primero}(F)$ por (9) $F \rightarrow \text{id}$
- $n \in \text{Primero}(F)$ por (10) $F \rightarrow n$

	Primero
S	id
E	$"(", \text{id}, n$
E'	$+, \epsilon$
T	$"(", \text{id}, n$
T'	$*, \epsilon$
F	$"(", \text{id}, n$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (2.- Gramática de declaraciones

1 / 2)

$$P = \{$$

- (1) $S \rightarrow D S$
- (2) $S \rightarrow \epsilon$
- (3) $D \rightarrow T L ;$
- (4) $T \rightarrow \mathbf{int}$
- (5) $T \rightarrow \mathbf{float}$
- (6) $L \rightarrow \mathbf{id} L'$
- (7) $L' \rightarrow , \mathbf{id} L'$
- (8) $L' \rightarrow \epsilon$

$$\}$$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (2.- Gramática de declaraciones

2 / 2)

- $\text{Primero}(D) - \{\epsilon\} \subseteq \text{Primero}(S)$
por (1) $S \rightarrow D S$
- $\epsilon \in \text{Primero}(S)$ por (2) $S \rightarrow \epsilon$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(D)$
por (3) $D \rightarrow T L$;
- $\text{int} \in (T)$ por (5) $T \rightarrow \text{int}$
- $\text{float} \in \text{Primero}(T)$ por (5) $T \rightarrow \text{float}$
- $\text{id} \in \text{Primero}(L)$ por (6) $L \rightarrow \text{id } L'$
- $"," \in \text{Primero}(L')$ por (7) $L' \rightarrow , \text{id } L'$
- $\epsilon \in \text{Primero}(L')$ por (8) $L' \rightarrow \epsilon$

	Primero
S	$\epsilon, \text{int}, \text{float}$
D	int, float
T	int, float
L	id
L'	$"," \epsilon$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (2.- Gramática de declaraciones

2 / 2)

- $\text{Primero}(D) - \{\epsilon\} \subseteq \text{Primero}(S)$
por (1) $S \rightarrow D S$
- $\epsilon \in \text{Primero}(S)$ por (2) $S \rightarrow \epsilon$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(D)$
por (3) $D \rightarrow T L$;
- $\text{int} \in (T)$ por (5) $T \rightarrow \text{int}$
- $\text{float} \in \text{Primero}(T)$ por (5) $T \rightarrow \text{float}$
- $\text{id} \in \text{Primero}(L)$ por (6) $L \rightarrow \text{id } L'$
- $',' \in \text{Primero}(L')$ por (7) $L' \rightarrow ',' \text{id } L'$
- $\epsilon \in \text{Primero}(L')$ por (8) $L' \rightarrow \epsilon$

	Primero
S	ϵ int, float
D	int, float
T	int, float
L	id
L'	',' id, ϵ

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (2.- Gramática de declaraciones

2 / 2)

- $\text{Primero}(D) - \{\epsilon\} \subseteq \text{Primero}(S)$
por (1) $S \rightarrow D S$
- $\epsilon \in \text{Primero}(S)$ por (2) $S \rightarrow \epsilon$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(D)$
por (3) $D \rightarrow T L ;$
- $\text{int} \in \text{Primero}(T)$ por (5) $T \rightarrow \text{int}$
- $\text{float} \in \text{Primero}(T)$ por (5) $T \rightarrow \text{float}$
- $\text{id} \in \text{Primero}(L)$ por (6) $L \rightarrow \text{id } L'$
- $"," \in \text{Primero}(L')$ por (7) $L' \rightarrow , \text{id } L'$
- $\epsilon \in \text{Primero}(L')$ por (8) $L' \rightarrow \epsilon$

	Primero
S	ϵ int , float
D	int , float
T	int , float
L	id
L'	"," ϵ

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (2.- Gramática de declaraciones

2 / 2)

- $\text{Primero}(D) - \{\epsilon\} \subseteq \text{Primero}(S)$
por (1) $S \rightarrow D S$
- $\epsilon \in \text{Primero}(S)$ por (2) $S \rightarrow \epsilon$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(D)$
por (3) $D \rightarrow T L ;$
- $\text{int} \in (T)$ por (5) $T \rightarrow \text{int}$
- $\text{float} \in \text{Primero}(T)$ por (5) $T \rightarrow \text{float}$
- $\text{id} \in \text{Primero}(L)$ por (6) $L \rightarrow \text{id } L'$
- $',' \in \text{Primero}(L')$ por (7) $L' \rightarrow , \text{id } L'$
- $\epsilon \in \text{Primero}(L')$ por (8) $L' \rightarrow \epsilon$

	Primero
S	$\epsilon, \text{int}, \text{float}$
D	int, float
T	int, float
L	id
L'	$',' \epsilon$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (2.- Gramática de declaraciones

2 / 2)

- $\text{Primero}(D) - \{\epsilon\} \subseteq \text{Primero}(S)$
por (1) $S \rightarrow D S$
- $\epsilon \in \text{Primero}(S)$ por (2) $S \rightarrow \epsilon$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(D)$
por (3) $D \rightarrow T L ;$
- $\text{int} \in (T)$ por (5) $T \rightarrow \text{int}$
- $\text{float} \in \text{Primero}(T)$ por (5) $T \rightarrow \text{float}$
- $\text{id} \in \text{Primero}(L)$ por (6) $L \rightarrow \text{id } L'$
- $',' \in \text{Primero}(L')$ por (7) $L' \rightarrow ',' \text{id } L'$
- $\epsilon \in \text{Primero}(L')$ por (8) $L' \rightarrow \epsilon$

	Primero
S	$\epsilon, \text{int}, \text{float}$
D	int, float
T	int, float
L	id
L'	$',' \text{id}, \epsilon$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (2.- Gramática de declaraciones

2 / 2)

- $\text{Primero}(D) - \{\epsilon\} \subseteq \text{Primero}(S)$
por (1) $S \rightarrow D S$
- $\epsilon \in \text{Primero}(S)$ por (2) $S \rightarrow \epsilon$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(D)$
por (3) $D \rightarrow T L ;$
- $\text{int} \in (T)$ por (5) $T \rightarrow \text{int}$
- $\text{float} \in \text{Primero}(T)$ por (5) $T \rightarrow \text{float}$
- $\text{id} \in \text{Primero}(L)$ por (6) $L \rightarrow \text{id } L'$
- $',' \in \text{Primero}(L')$ por (7) $L' \rightarrow , \text{id } L'$
- $\epsilon \in \text{Primero}(L')$ por (8) $L' \rightarrow \epsilon$

	Primero
S	$\epsilon, \text{int}, \text{float}$
D	int, float
T	int, float
L	id
L'	$','$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (2.- Gramática de declaraciones

2 / 2)

- $\text{Primero}(D) - \{\epsilon\} \subseteq \text{Primero}(S)$
por (1) $S \rightarrow D S$
- $\epsilon \in \text{Primero}(S)$ por (2) $S \rightarrow \epsilon$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(D)$
por (3) $D \rightarrow T L ;$
- $\text{int} \in (T)$ por (5) $T \rightarrow \text{int}$
- $\text{float} \in \text{Primero}(T)$ por (5) $T \rightarrow \text{float}$
- $\text{id} \in \text{Primero}(L)$ por (6) $L \rightarrow \text{id } L'$
- $“,” \in \text{Primero}(L')$ por (7) $L' \rightarrow , \text{id } L'$
- $\epsilon \in \text{Primero}(L')$ por (8) $L' \rightarrow \epsilon$

	Primero
S	$\epsilon, \text{int}, \text{float}$
D	int, float
T	int, float
L	id
L'	$“,”$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejemplo (2.- Gramática de declaraciones

2 / 2)

- $\text{Primero}(D) - \{\epsilon\} \subseteq \text{Primero}(S)$
por (1) $S \rightarrow D S$
- $\epsilon \in \text{Primero}(S)$ por (2) $S \rightarrow \epsilon$
- $\text{Primero}(T) - \{\epsilon\} \subseteq \text{Primero}(D)$
por (3) $D \rightarrow T L ;$
- $\text{int} \in (T)$ por (5) $T \rightarrow \text{int}$
- $\text{float} \in \text{Primero}(T)$ por (5) $T \rightarrow \text{float}$
- $\text{id} \in \text{Primero}(L)$ por (6) $L \rightarrow \text{id } L'$
- $“,” \in \text{Primero}(L')$ por (7) $L' \rightarrow , \text{id } L'$
- $\epsilon \in \text{Primero}(L')$ por (8) $L' \rightarrow \epsilon$

	Primero
S	$\epsilon, \text{int}, \text{float}$
D	int, float
T	int, float
L	id
L'	$“,” , \epsilon$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejercicios (Construcción del conjunto Primero)

- 1.- Gramática de prototipos
- 2.- Gramática que genera $L(G) = L(a^* b^* c^*)$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejercicio (1.- Gramática de los prototipos

1 / 2)

$$P = \{$$

- (1) $S \rightarrow D S$
- (2) $S \rightarrow \epsilon$
- (3) $D \rightarrow T \text{ identificador } (P) ;$
- (4) $T \rightarrow \text{int}$
- (5) $T \rightarrow \text{float}$
- (6) $P \rightarrow \epsilon$
- (7) $P \rightarrow \text{id } L$
- (8) $L \rightarrow \epsilon$
- (9) $L \rightarrow , \text{id } L$

$$\}$$

Análisis sintáctico descendente predictivo

Conjunto Primero

Ejercicio (2.- Gramática que genera $L(G) = L(a^* b^* c^*)$ 2 / 2)

$$P = \{$$

- (1) $S \rightarrow A B C$
- (2) $A \rightarrow \mathbf{a} A$
- (3) $A \rightarrow \epsilon$
- (4) $B \rightarrow \mathbf{b} B$
- (5) $B \rightarrow \epsilon$
- (6) $C \rightarrow \mathbf{c} C$
- (7) $C \rightarrow \epsilon$

$$\}$$

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
 - Descripción
 - Gramáticas LL(k)
 - Fases
 - Conjunto Primero
 - **Conjunto Siguierte**
 - Construcción de la tabla predictiva
 - Conflictos en la tabla predictiva
 - Implementación recursiva
 - Implementación iterativa

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Definición (Conjunto Siguiente)

- Si $A \in V_N$ entonces

$$\mathbf{Siguiente(A)} = \{\mathbf{a} \mid \mathbf{a} \in V_T \wedge S \xRightarrow{*} \alpha A \mathbf{a} \beta\} \cup \{\mathbf{\$} \mid S \xRightarrow{*} \alpha A\}$$

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Nota (Conjunto Siguiente)

- Si A es un símbolo no terminal entonces **Siguiente(A)** está compuesto por
 - los *símbolos terminales* que pueden aparecer detrás de A en una derivación de la gramática
 - y el símbolo $\$$ (fin de cadena), si A aparece en la última posición durante algún paso de la derivación.

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Nota (Conjunto Siguierte)

- La palabra vacía ϵ *nunca* pertenece al conjunto **Siguierte(A)**.

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Construcción del conjunto Siguiente

- 1.- $\$ \in \text{Siguiente}(S)$,
donde S es el símbolo inicial de la gramática.
- 2.- Si $A \rightarrow \alpha B \beta \in P$
entonces $\text{Primero}(\beta) - \{\epsilon\} \subseteq \text{Siguiente}(B)$
- 3.- Si $A \rightarrow \alpha B \in P \vee (A \rightarrow \alpha B \beta \wedge \epsilon \in \text{Primero}(\beta))$
entonces $\text{Siguiente}(A) \subseteq \text{Siguiente}(B)$

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (Construcción del conjunto Siguiente: tercera regla)

3.- Si $A \rightarrow \alpha B \in P \vee (A \rightarrow \alpha B \beta \wedge \epsilon \in \text{Primero}(\beta))$

entonces $\text{Siguiente}(A) \subseteq \text{Siguiente}(B)$

Los símbolos que siguen al símbolo A también siguen al símbolo B :

$$(a) S \xrightarrow{*} \delta A \gamma \Rightarrow \delta \underline{\alpha B} \gamma$$

$$(b) S \xrightarrow{*} \delta A \gamma \Rightarrow \delta \underline{\alpha B} \beta \gamma \stackrel{\dagger}{\Rightarrow} \delta \alpha B \gamma$$

porque $\epsilon \in \text{Primero}(\beta)$ y, por tanto, $\beta \stackrel{\dagger}{\Rightarrow} \epsilon$

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplos (Construcción del conjunto Siguiente)

- 1.- *Gramática de las expresiones aritméticas.*
- 2.- *Gramática de las declaraciones.*

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejemplo (1.- Gramática de las expresiones aritméticas 1 / 8)

- $$P = \{$$
- (1) $S \rightarrow \text{identificador} = E$
 - (2) $E \rightarrow T E'$
 - (3) $E' \rightarrow + T E'$
 - (4) $E' \rightarrow \epsilon$
 - (5) $T \rightarrow F T'$
 - (6) $T' \rightarrow * F T'$
 - (7) $T' \rightarrow \epsilon$
 - (8) $F \rightarrow (E)$
 - (9) $F \rightarrow \text{identificador}$
 - (10) $F \rightarrow \text{número}$
- $$\}$$

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 8)

- $\$ \in \text{Siguierte}(S)$ (regla 1)
- $\text{Siguierte}(S) \subseteq \text{Siguierte}(E)$
por (1) $S \rightarrow \text{identificador} = E$
(regla 3)
- $\text{Primero}(E') - \{\epsilon\} \subseteq \text{Siguierte}(T)$
por (2) $E \rightarrow T E'$ (regla 2)
- $\text{Siguierte}(E) \subseteq \text{Siguierte}(E')$
por (2) $E \rightarrow T E'$ (regla 3)
- Como $\epsilon \in \text{Primero}(E')$ entonces
 $\text{Siguierte}(E) \subseteq \text{Siguierte}(T)$
por (2) $E \rightarrow T E'$ (regla 3)

	Primero	Siguierte
S	id	\$
E	"(", id, n	
E'	+, ϵ	
T	"(", id, n	+, *
T'	*, ϵ	
F	"(", id, n	

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 8)

- $\$ \in \text{Siguiente}(S)$ (regla 1)
- $\text{Siguiente}(S) \subseteq \text{Siguiente}(E)$
por (1) $S \rightarrow \text{identificador} = E$
(regla 3)
- $\text{Primero}(E') - \{\epsilon\} \subseteq \text{Siguiente}(T)$
por (2) $E \rightarrow T E'$ (regla 2)
- $\text{Siguiente}(E) \subseteq \text{Siguiente}(E')$
por (2) $E \rightarrow T E'$ (regla 3)
- Como $\epsilon \in \text{Primero}(E')$ entonces
 $\text{Siguiente}(E) \subseteq \text{Siguiente}(T)$
por (2) $E \rightarrow T E'$ (regla 3)

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	
T	"(", id, n	+, *
T'	*, ϵ	
F	"(", id, n	

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 8)

- $\$ \in \text{Siguiente}(S)$ (regla 1)
- $\text{Siguiente}(S) \subseteq \text{Siguiente}(E)$
por (1) $S \rightarrow \text{identificador} = E$
(regla 3)
- $\text{Primero}(E') - \{\epsilon\} \subseteq \text{Siguiente}(T)$
por (2) $E \rightarrow T E'$ (regla 2)
- $\text{Siguiente}(E) \subseteq \text{Siguiente}(E')$
por (2) $E \rightarrow T E'$ (regla 3)
- Como $\epsilon \in \text{Primero}(E')$ entonces
 $\text{Siguiente}(E) \subseteq \text{Siguiente}(T)$
por (2) $E \rightarrow T E'$ (regla 3)

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	
T	"(", id, n	+
T'	*, ϵ	
F	"(", id, n	

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 8)

- $\$ \in \text{Siguierte}(S)$ (regla 1)
- $\text{Siguierte}(S) \subseteq \text{Siguierte}(E)$
por (1) $S \rightarrow \text{identificador} = E$
(regla 3)
- $\text{Primero}(E') - \{\epsilon\} \subseteq \text{Siguierte}(T)$
por (2) $E \rightarrow T E'$ (regla 2)
- $\text{Siguierte}(E) \subseteq \text{Siguierte}(E')$
por (2) $E \rightarrow T E'$ (regla 3)
- Como $\epsilon \in \text{Primero}(E')$ entonces
 $\text{Siguierte}(E) \subseteq \text{Siguierte}(T)$
por (2) $E \rightarrow T E'$ (regla 3)

	Primero	Siguierte
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	\$
T	"(", id, n	+
T'	*, ϵ	
F	"(", id, n	

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (1.- Gramática de las expresiones aritméticas 2 / 8)

- $\$ \in \text{Siguiente}(S)$ (regla 1)
- $\text{Siguiente}(S) \subseteq \text{Siguiente}(E)$
por (1) $S \rightarrow \text{identificador} = E$
(regla 3)
- $\text{Primero}(E') - \{\epsilon\} \subseteq \text{Siguiente}(T)$
por (2) $E \rightarrow T E'$ (regla 2)
- $\text{Siguiente}(E) \subseteq \text{Siguiente}(E')$
por (2) $E \rightarrow T E'$ (regla 3)
- Como $\epsilon \in \text{Primero}(E')$ entonces
 $\text{Siguiente}(E) \subseteq \text{Siguiente}(T)$
por (2) $E \rightarrow T E'$ (regla 3)

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	\$
T	"(", id, n	+, \$
T'	*, ϵ	
F	"(", id, n	

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (1.- Gramática de las expresiones aritméticas 3 / 8)

- $\text{Primero}(E') - \{\epsilon\} \subseteq \text{Siguiente}(T)$
(repetido)

por (3) $E' \rightarrow + T E'$ (regla 2)

- $\text{Siguiente}(E') \subseteq \text{Siguiente}(E')$
(superflua)

por (3) $E' \rightarrow + T E'$ (regla 3)

- Como $\epsilon \in \text{Primero}(E')$ entonces
 $\text{Siguiente}(E') \subseteq \text{Siguiente}(T)$
por (3) $E' \rightarrow + T E'$ (regla 3)

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	\$
T	"(", id, n	+, \$
T'	*, ϵ	
F	"(", id, n	

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (1.- Gramática de las expresiones aritméticas 3 / 8)

- $\text{Primero}(E') - \{\epsilon\} \subseteq \text{Siguiente}(T)$
(repetido)

por (3) $E' \rightarrow + T E'$ (regla 2)

- $\text{Siguiente}(E') \subseteq \text{Siguiente}(E')$
(superflua)

por (3) $E' \rightarrow + T E'$ (regla 3)

- Como $\epsilon \in \text{Primero}(E')$ entonces
 $\text{Siguiente}(E') \subseteq \text{Siguiente}(T)$
por (3) $E' \rightarrow + T E'$ (regla 3)

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	\$
T	"(", id, n	+, \$
T'	*, ϵ	
F	"(", id, n	

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (1.- Gramática de las expresiones aritméticas 3 / 8)

- $\text{Primero}(E') - \{\epsilon\} \subseteq \text{Siguiente}(T)$
(repetido)

por (3) $E' \rightarrow + T E'$ (regla 2)

- $\text{Siguiente}(E') \subseteq \text{Siguiente}(E')$
(superflua)

por (3) $E' \rightarrow + T E'$ (regla 3)

- Como $\epsilon \in \text{Primero}(E')$ entonces
 $\text{Siguiente}(E') \subseteq \text{Siguiente}(T)$

por (3) $E' \rightarrow + T E'$ (regla 3)

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	\$
T	"(", id, n	+, \$
T'	*, ϵ	
F	"(", id, n	

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (1.- Gramática de las expresiones aritméticas 4 / 8)

- $\text{Primero}(T') - \{\epsilon\} \subseteq \text{Siguiente}(F)$
por (5) $T \rightarrow F T'$ (regla 2)
- $\text{Siguiente}(T) \subseteq \text{Siguiente}(T')$
por (5) $T \rightarrow F T'$ (regla 3)
- Como $\epsilon \in \text{Primero}(T')$ entonces
 $\text{Siguiente}(T) \subseteq \text{Siguiente}(F)$
por (5) $T \rightarrow F T'$ (regla 3)

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	\$
T	"(", id, n	+, \$
T'	*, ϵ	+, \$
F	"(", id, n	*

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejemplo (1.- Gramática de las expresiones aritméticas 4 / 8)

- $\text{Primero}(T') - \{\epsilon\} \subseteq \text{Siguierte}(F)$
por (5) $T \rightarrow F T'$ (regla 2)
- $\text{Siguierte}(T) \subseteq \text{Siguierte}(T')$
por (5) $T \rightarrow F T'$ (regla 3)
- Como $\epsilon \in \text{Primero}(T')$ entonces
 $\text{Siguierte}(T) \subseteq \text{Siguierte}(F)$
por (5) $T \rightarrow F T'$ (regla 3)

	Primero	Siguierte
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	\$
T	"(", id, n	+, \$
T'	*, ϵ	+, \$
F	"(", id, n	*

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (1.- Gramática de las expresiones aritméticas 4 / 8)

- $\text{Primero}(T') - \{\epsilon\} \subseteq \text{Siguiente}(F)$
por (5) $T \rightarrow F T'$ (regla 2)
- $\text{Siguiente}(T) \subseteq \text{Siguiente}(T')$
por (5) $T \rightarrow F T'$ (regla 3)
- Como $\epsilon \in \text{Primero}(T')$ entonces
 $\text{Siguiente}(T) \subseteq \text{Siguiente}(F)$
por (5) $T \rightarrow F T'$ (regla 3)

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	\$
T	"(", id, n	+, \$
T'	*, ϵ	+, \$
F	"(", id, n	*, +, \$

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejemplo (1.- Gramática de las expresiones aritméticas 5 / 8)

- $\text{Primero}(T') - \{\epsilon\} \subseteq \text{Siguierte}(F)$
(repetido)

por (6) $T' \rightarrow * F T'$ (regla 2)

- $\text{Siguierte}(T') \subseteq \text{Siguierte}(T')$
(superflua)

por (6) $T' \rightarrow * F T'$ (regla 3)

- Como $\epsilon \in \text{Primero}(T')$ entonces
 $\text{Siguierte}(T') \subseteq \text{Siguierte}(F)$
por (6) $T' \rightarrow * F T'$ (regla 3)

	Primero	Siguierte
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	\$
T	"(", id, n	+, \$
T'	*, ϵ	+, \$
F	"(", id, n	*, +, \$

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (1.- Gramática de las expresiones aritméticas 5 / 8)

- $\text{Primero}(T') - \{\epsilon\} \subseteq \text{Siguiente}(F)$
(repetido)

por (6) $T' \rightarrow * F T'$ (regla 2)

- $\text{Siguiente}(T') \subseteq \text{Siguiente}(T')$
(superflua)

por (6) $T' \rightarrow * F T'$ (regla 3)

- Como $\epsilon \in \text{Primero}(T')$ entonces
 $\text{Siguiente}(T') \subseteq \text{Siguiente}(F)$
por (6) $T' \rightarrow * F T'$ (regla 3)

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	\$
T	"(", id, n	+, \$
T'	*, ϵ	+, \$
F	"(", id, n	*, +, \$

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (1.- Gramática de las expresiones aritméticas 5 / 8)

- $\text{Primero}(T') - \{\epsilon\} \subseteq \text{Siguiente}(F)$
(repetido)

por (6) $T' \rightarrow * F T'$ (regla 2)

- $\text{Siguiente}(T') \subseteq \text{Siguiente}(T')$
(superflua)

por (6) $T' \rightarrow * F T'$ (regla 3)

- Como $\epsilon \in \text{Primero}(T')$ entonces
 $\text{Siguiente}(T') \subseteq \text{Siguiente}(F)$

por (6) $T' \rightarrow * F T'$ (regla 3)

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$
E'	+, ϵ	\$
T	"(", id, n	+, \$
T'	*, ϵ	+, \$
F	"(", id, n	*, +, \$

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejemplo (1.- Gramática de las expresiones aritméticas 6 / 8)

- "(" \in *Siguiente*(*E*)
por (8) $F \rightarrow (E)$ (regla 2)

	Primero	Siguiente
<i>S</i>	id	\$
<i>E</i>	"(", id, n	\$, "("
<i>E'</i>	+, ϵ	\$, "("
<i>T</i>	"(", id, n	+, \$, "("
<i>T'</i>	*, ϵ	+, \$, "("
<i>F</i>	"(", id, n	*, +, \$, "("

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Nota (1.- Gramática de las expresiones aritméticas 7 / 8)

- Las reglas de producción que no contienen **ningún símbolo no terminal** en su parte derecha no influyen en el conjunto

Siguiente

$$(4) E' \rightarrow \epsilon$$

$$(7) T' \rightarrow \epsilon$$

$$(9) F \rightarrow \text{identificador}$$

$$(10) F \rightarrow \text{número}$$

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejemplo (1.- Gramática de las expresiones aritméticas 8 / 8)

$$P = \{$$

- (1) $S \rightarrow \text{identificador} = E$
 - (2) $E \rightarrow T E'$
 - (3) $E' \rightarrow + T E'$
 - (4) $E' \rightarrow \epsilon$
 - (5) $T \rightarrow F T'$
 - (6) $T' \rightarrow * F T'$
 - (7) $T' \rightarrow \epsilon$
 - (8) $F \rightarrow (E)$
 - (9) $F \rightarrow \text{identificador}$
 - (10) $F \rightarrow \text{número}$
- $$\}$$

	Primero	Siguierte
S	id	\$
E	"(", id, n	\$, ")"
E'	+, ϵ	\$, ")"
T	"(", id, n	+, \$, ")"
T'	*, ϵ	+, \$, ")"
F	"(", id, n	*, +, \$, ")"

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (2.- Gramática de las declaraciones

1 / 2)

$$P = \{$$

- (1) $S \rightarrow D S$
- (2) $S \rightarrow \epsilon$
- (3) $D \rightarrow T L ;$
- (4) $T \rightarrow \mathbf{int}$
- (5) $T \rightarrow \mathbf{float}$
- (6) $L \rightarrow \mathbf{id} L'$
- (7) $L' \rightarrow , \mathbf{id} L'$
- (8) $L' \rightarrow \epsilon$

$$\}$$

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejemplo (2.- Gramática de las declaraciones

2 / 2)

- $\$ \in \text{Siguierte}(S)$ (regla 1)
- $\text{Primero}(S) = \{\epsilon\} \subseteq \text{Siguierte}(D)$
por (1) $S \rightarrow D S$ (regla 2)
- Como $\epsilon \in \text{Primero}(S)$ entonces
 $\text{Siguierte}(S) \in \text{Siguierte}(D)$
por (1) $S \rightarrow D S$ (regla 3)
- $\text{Primero}(L) = \{\epsilon\} \subseteq \text{Siguierte}(T)$
por (3) $D \rightarrow T L$; (regla 2)
- $;\in \text{Siguierte}(L)$
por (3) $D \rightarrow T L$; (regla 2)
- $\text{Siguierte}(L) \in \text{Siguierte}(L')$
por (6) $L \rightarrow \text{id } L'$ (regla 3)

	Primero	Siguierte
S	$\epsilon, \text{int}, \text{float}$	$\$$
D	int, float	$\text{int}, \text{float}, \$$
T	int, float	id
L	id	$;$
L'	$"", , \epsilon$	$;$

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejemplo (2.- Gramática de las declaraciones

2 / 2)

- $\$ \in \text{Siguierte}(S)$ (regla 1)
- $\text{Primero}(S) - \{\epsilon\} \subseteq \text{Siguierte}(D)$
por (1) $S \rightarrow D S$ (regla 2)
- Como $\epsilon \in \text{Primero}(S)$ entonces
 $\text{Siguierte}(S) \in \text{Siguierte}(D)$
por (1) $S \rightarrow D S$ (regla 3)
- $\text{Primero}(L) - \{\epsilon\} \subseteq \text{Siguierte}(T)$
por (3) $D \rightarrow T L$; (regla 2)
- $;\in \text{Siguierte}(L)$
por (3) $D \rightarrow T L$; (regla 2)
- $\text{Siguierte}(L) \in \text{Siguierte}(L')$
por (6) $L \rightarrow \text{id } L'$ (regla 3)

	Primero	Siguierte
S	$\epsilon, \text{int}, \text{float}$	$\$$
D	int, float	int, float
T	int, float	
L	id	
L'	$“,” , \epsilon$	

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (2.- Gramática de las declaraciones

2 / 2)

- $\$ \in \text{Siguiente}(S)$ (regla 1)
- $\text{Primero}(S) - \{\epsilon\} \subseteq \text{Siguiente}(D)$
por (1) $S \rightarrow D S$ (regla 2)
- Como $\epsilon \in \text{Primero}(S)$ entonces
 $\text{Siguiente}(S) \in \text{Siguiente}(D)$
por (1) $S \rightarrow D S$ (regla 3)
- $\text{Primero}(L) - \{\epsilon\} \subseteq \text{Siguiente}(T)$
por (3) $D \rightarrow T L$; (regla 2)
- $;\in \text{Siguiente}(L)$
por (3) $D \rightarrow T L$; (regla 2)
- $\text{Siguiente}(L) \in \text{Siguiente}(L')$
por (6) $L \rightarrow \text{id } L'$ (regla 3)

	Primero	Siguiente
S	$\epsilon, \text{int}, \text{float}$	$\$$
D	int, float	$\text{int}, \text{float}, \text{ ;}, \$$
T	int, float	
L	id	
L'	$\text{ ;}, \epsilon$	

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejemplo (2.- Gramática de las declaraciones

2 / 2)

- $\$ \in \text{Siguierte}(S)$ (regla 1)
- $\text{Primero}(S) - \{\epsilon\} \subseteq \text{Siguierte}(D)$
por (1) $S \rightarrow D S$ (regla 2)
- Como $\epsilon \in \text{Primero}(S)$ entonces
 $\text{Siguierte}(S) \in \text{Siguierte}(D)$
por (1) $S \rightarrow D S$ (regla 3)
- $\text{Primero}(L;) - \{\epsilon\} \subseteq \text{Siguierte}(T)$
por (3) $D \rightarrow T L ;$ (regla 2)
- $; \in \text{Siguierte}(L)$
por (3) $D \rightarrow T L ;$ (regla 2)
- $\text{Siguierte}(L) \in \text{Siguierte}(L')$
por (6) $L \rightarrow \text{id } L'$ (regla 3)

	Primero	Siguierte
S	$\epsilon, \text{int}, \text{float}$	$\$$
D	int, float	$\text{int}, \text{float}, \text{ ;}, \$$
T	int, float	id
L	id	
L'	$“,” , \epsilon$	

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejemplo (2.- Gramática de las declaraciones

2 / 2)

- $\$ \in \text{Siguiente}(S)$ (regla 1)
- $\text{Primero}(S) - \{\epsilon\} \subseteq \text{Siguiente}(D)$
por (1) $S \rightarrow D S$ (regla 2)
- Como $\epsilon \in \text{Primero}(S)$ entonces
 $\text{Siguiente}(S) \in \text{Siguiente}(D)$
por (1) $S \rightarrow D S$ (regla 3)
- $\text{Primero}(L;) - \{\epsilon\} \subseteq \text{Siguiente}(T)$
por (3) $D \rightarrow T L ;$ (regla 2)
- $;\in \text{Siguiente}(L)$
por (3) $D \rightarrow T L ;$ (regla 2)
- $\text{Siguiente}(L) \in \text{Siguiente}(L')$
por (6) $L \rightarrow \text{id } L'$ (regla 3)

	Primero	Siguiente
S	$\epsilon, \text{int}, \text{float}$	$\$$
D	int, float	$\text{int}, \text{float}, \text{ ;}, \$$
T	int, float	id
L	id	$;$
L'	$"", \text{ ;}, \epsilon$	

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejemplo (2.- Gramática de las declaraciones

2 / 2)

- $\$ \in \text{Siguierte}(S)$ (regla 1)
- $\text{Primero}(S) - \{\epsilon\} \subseteq \text{Siguierte}(D)$
por (1) $S \rightarrow D S$ (regla 2)
- Como $\epsilon \in \text{Primero}(S)$ entonces
 $\text{Siguierte}(S) \in \text{Siguierte}(D)$
por (1) $S \rightarrow D S$ (regla 3)
- $\text{Primero}(L;) - \{\epsilon\} \subseteq \text{Siguierte}(T)$
por (3) $D \rightarrow T L ;$ (regla 2)
- $;\in \text{Siguierte}(L)$
por (3) $D \rightarrow T L ;$ (regla 2)
- $\text{Siguierte}(L) \in \text{Siguierte}(L')$
por (6) $L \rightarrow \text{id } L'$ (regla 3)

	Primero	Siguierte
S	$\epsilon, \text{int}, \text{float}$	$\$$
D	int, float	$\text{int}, \text{float}, \text{ ;}, \$$
T	int, float	id
L	id	$;$
L'	$"", \text{ ;}, \epsilon$	$;$

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Limitaciones del análisis sintáctico descendente predictivo

- **No** se puede realizar si la gramática posee alguna de las siguiente propiedades:
 - Es recursiva por la izquierda.
 - No está factorizada por la izquierda.
 - Es ambigua.

Nota

- *Estas condiciones son **necesarias** pero **no suficientes**.*

Análisis sintáctico descendente predictivo

Conjunto Siguierte

Ejercicios (Construcción del conjunto Siguierte)

- 1.- Gramática de prototipos
- 2.- Gramática que genera $L(G) = L(a^* b^* c^*)$

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejercicio (1.- Gramática de los prototipos

1 / 2)

$$P = \{$$

- (1) $S \rightarrow D S$
- (2) $S \rightarrow \epsilon$
- (3) $D \rightarrow T \text{ identificador } (P) ;$
- (4) $T \rightarrow \text{int}$
- (5) $T \rightarrow \text{float}$
- (6) $P \rightarrow \epsilon$
- (7) $P \rightarrow \text{id } L$
- (8) $L \rightarrow \epsilon$
- (9) $L \rightarrow , \text{id } L$

$$\}$$

Análisis sintáctico descendente predictivo

Conjunto Siguiente

Ejercicio (2.- Gramática que genera $L(G) = L(a^* b^* c^*)$ 2 / 2)

- $$P = \{$$
- (1) $S \rightarrow A B C$
 - (2) $A \rightarrow \mathbf{a} A$
 - (3) $A \rightarrow \epsilon$
 - (4) $B \rightarrow \mathbf{b} B$
 - (5) $B \rightarrow \epsilon$
 - (6) $C \rightarrow \mathbf{c} C$
 - (7) $C \rightarrow \epsilon$
- $$\}$$

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
 - Descripción
 - Gramáticas LL(k)
 - Fases
 - Conjunto Primero
 - Conjunto Siguierte
 - Construcción de la tabla predictiva
 - Conflictos en la tabla predictiva
 - Implementación recursiva
 - Implementación iterativa

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Definición (Tabla predictiva)

M	Símbolo de entrada					
	σ_1	...	σ_j	...	σ_m	\$
S						
A_1						
...						
A_i			<i>Regla k</i>			
...						
A_n						

- $S, A_i \in V_N \forall i \in \{1, 2, \dots, n\}$
- $\sigma_j \in V_T \forall j \in \{1, 2, \dots, m\}$
- **\$**: símbolo que indica el fin de la cadena de entrada
- *Regla K*: regla del símbolo A_i que se debe usar cuando σ_j es el símbolo *actual*.

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Gramática de expresiones aritméticas)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

M	Símbolo de entrada							
	id	=	+	*	()	número	\$	
S	1							
E	2				2	2		
E'			3		4		4	
T	5				5	5		
T'			7	6	7		7	
F	9				8	10		

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Algoritmo (Construcción de la tabla predictiva)

```
[1]   Para cada  $A \rightarrow \alpha \in P$  hacer
[2]       si  $a \in \text{Primero}(\alpha) - \{\epsilon\}$ 
[3]           entonces  $A \rightarrow \alpha \in M[A, a]$ 
[4]       fin_si
[5]
[6]       si  $\epsilon \in \text{Primero}(\alpha)$ 
[7]           entonces
[8]               para cada  $b \in \text{Siguiente}(A)$  hacer
[9]                    $A \rightarrow \alpha \in M[A, b]$ 
[10]              fin_para
[11]       fin_si
[12]   fin_para
```

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Notas (Construcción de la tabla predictiva)

- **M**: *tabla predictiva*
- Las **celdas vacías** de **M** representan situaciones de **error** que han de ser tratadas.
- Si **M** posee alguna **celda con valores** múltiples entonces la gramática tiene un **conflicto** y **no admite** un análisis descendente predictivo.

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva)

1 / 12)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$ *Paso previo*

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$, "("
E'	+, ϵ	\$, "("
T	"(", id, n	+, \$, "("
T'	*, ϵ	+, \$, "("
F	"(", id, n	*, +, \$, "("

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva

2 / 12)

M	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	I							
E								
E'								
T								
T'								
F								

$$(1) \underbrace{S}_{A} \rightarrow \underbrace{\text{identificador} = E}_{\alpha}$$

$$\bullet \text{Primero}(\alpha) - \{\epsilon\} = \text{Primero}(\text{identificador} = E) = \{ \text{identificador} \}$$

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva)

3 / 12)

Tabla predictiva								
M	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'								
T								
T'								
F								

$$(2) \underbrace{E}_A \rightarrow \underbrace{TE'}_\alpha$$

$$\bullet \text{Primero}(\alpha) - \{\epsilon\} = \text{Primero}(TE') = \{ "(", \text{id}, n \}$$

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva)

4 / 12)

Tabla predictiva								
M	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'			3					
T								
T'								
F								

$$(3) \underbrace{E'}_A \rightarrow + \underbrace{T E'}_\alpha$$

$$\bullet \text{Primero}(\alpha) - \{\epsilon\} = \text{Primero}(+ T E') = \{ + \}$$

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva)

5 / 12)

Tabla predictiva								
M	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'			3			4		4
T								
T'								
F								

$$(4) \underbrace{E'}_A \rightarrow \underbrace{\epsilon}_\alpha$$

- Como $\epsilon \in \text{Primero}(\alpha)$, entonces se añade la regla 4 en las celdas $M[E', \mathbf{b}]$ donde $\mathbf{b} \in \text{Siguiente}(E') = \{ \text{""}, \$ \}$

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva)

6 / 12)

Tabla predictiva								
M	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'								
F								

$$(5) \underbrace{T}_A \rightarrow \underbrace{F T'}_\alpha$$

$$\bullet \text{Primero}(\alpha) - \{\epsilon\} = \text{Primero}(F T') = \{ "(", \text{id}, n \}$$

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva)

7 / 12)

Tabla predictiva								
M	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'				6				
F								

$$(6) \underbrace{T'}_A \rightarrow * \underbrace{T E'}_\alpha$$

$$\bullet \text{Primero}(\alpha) - \{\epsilon\} = \text{Primero}(* F T') = \{ * \}$$

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva)

8 / 12)

Tabla predictiva								
M	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F								

$$(7) \underbrace{T'}_A \rightarrow \underbrace{\epsilon}_\alpha$$

- Como $\epsilon \in \text{Primero}(\alpha)$, entonces se añade la regla 7 en las celdas $M[T', \mathbf{b}]$ donde $\mathbf{b} \in \text{Siguiete}(T') = \{ +, "(", \$ \}$

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva)

9 / 12)

Tabla predictiva								
M	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F					8			

$$(8) \underbrace{F}_A \rightarrow \underbrace{(E)}_\alpha$$

$$\bullet \text{Primero}(\alpha) - \{\epsilon\} = \text{Primero}((E)) = \{ "(" \}$$

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva)

10 / 12)

Tabla predictiva								
M	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F	9				8			

$$(9) \underbrace{F}_A \rightarrow \underbrace{\text{identificador}}_\alpha$$

$$\bullet \text{Primero}(\alpha) - \{\epsilon\} = \text{Primero}(\text{identificador}) = \{\text{identificador}\}$$

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva)

11 / 12)

Tabla predictiva								
M	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F	9				8		10	

$$(10) \underbrace{F}_A \rightarrow \underbrace{\text{número}}_\alpha$$

$$\bullet \text{Primero}(\alpha) - \{\epsilon\} = \text{Primero}(\text{número}) = \{\text{número}\}$$

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejemplo (Construcción de la tabla predictiva)

12 / 12)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

M	Símbolo de entrada							
	id	=	+	*	()	número	\$	
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F	9				8		10	

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejercicios (Construcción de la tabla predictiva)

- 1.- *Gramática de declaraciones*
- 2.- *Gramática de prototipos*
- 3.- *Gramática que genera $L(G) = L(a^* b^* c^*)$*

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejercicio (1.- Gramática de declaraciones

1a / 1)

 $P = \{$ (1) $S \rightarrow D S$ (2) $S \rightarrow \epsilon$ (3) $D \rightarrow T L ;$ (4) $T \rightarrow \text{int}$ (5) $T \rightarrow \text{float}$ (6) $L \rightarrow \text{id } L'$ (7) $L' \rightarrow , \text{id } L'$ (8) $L' \rightarrow \epsilon$ $\}$ *Paso previo*

	Primero	Siguiente
S	$\epsilon, \text{int}, \text{float}$	$\$$
D	int, float	$\text{int}, \text{float}, \$$
T	int, float	id
L	id	$;$
L'	$“, ”, \epsilon$	$;$

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejercicio (1.- Gramática de declaraciones

1b / 1)

 $P = \{$ (1) $S \rightarrow D S$ (2) $S \rightarrow \epsilon$ (3) $D \rightarrow T L ;$ (4) $T \rightarrow \text{int}$ (5) $T \rightarrow \text{float}$ (6) $L \rightarrow \text{id } L'$ (7) $L' \rightarrow , \text{id } L'$ (8) $L' \rightarrow \epsilon$ $\}$

Tabla predictiva

M	Símbolo de entrada					
	;	int	float	id	,	\$
S		1	1			2
D		3	3			
T		4	5			
L				6		
L'	8				7	

Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejercicio (2.- Gramática de los prototipos

2 / 3)

```
P = {  
  (1) S → D S  
  (2) S → ε  
  (3) D → T identificador ( P ) ;  
  (4) T → int  
  (5) T → float  
  (6) P → ε  
  (7) P → id L  
  (8) L → ε  
  (9) L → , id L  
}
```


Análisis sintáctico descendente predictivo

Construcción de la tabla predictiva

Ejercicio (3.- Gramática que genera $L(G) = L(a^* b^* c^*)$ 3 / 3)

$$P = \{$$

- (1) $S \rightarrow A B C$
- (2) $A \rightarrow \mathbf{a} A$
- (3) $A \rightarrow \epsilon$
- (4) $B \rightarrow \mathbf{b} B$
- (5) $B \rightarrow \epsilon$
- (6) $C \rightarrow \mathbf{c} C$
- (7) $C \rightarrow \epsilon$

$$\}$$

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
 - Descripción
 - Gramáticas LL(k)
 - Fases
 - Conjunto Primero
 - Conjunto Siguierte
 - Construcción de la tabla predictiva
 - **Conflictos en la tabla predictiva**
 - Implementación recursiva
 - Implementación iterativa

Análisis sintáctico descendente predictivo

Conflictos en la tabla predictiva

Definición (Conflicto en la tabla predictiva)

- Se produce un **conflicto** en una tabla predictiva cuando hay una celda $M[A_i, \sigma_j]$ que tiene asignadas **dos o más** reglas de producción.

Tabla predictiva (Conflicto en la tabla predictiva)

M	Símbolo de entrada					
	σ_1	...	σ_j	...	σ_m	$\$$
S						
A_1						
...						
A_i			k, k'			
...						
A_n						

Análisis sintáctico descendente predictivo

Conflictos en la tabla predictiva

Definición (Conflicto en la tabla predictiva)

- Se produce un **conflicto** en una tabla predictiva cuando hay una celda $M[A_i, \sigma_j]$ que tiene asignadas **dos o más** reglas de producción.

Ejemplo (Conflicto en la tabla predictiva)

M	Símbolo de entrada					
	σ_1	...	σ_j	...	σ_m	\$
S						
A_1						
...						
A_j			k, k'			
...						
A_n						

Análisis sintáctico descendente predictivo

Conflictos en la tabla predictiva

Nota (Conflicto en la tabla predictiva)

- Si una gramática posee al menos un **conflicto** entonces
 - **no admite** un análisis descendente predictivo.
 - **no es una gramática LL**

Análisis sintáctico descendente predictivo

Conflictos en la tabla predictiva

Nota (Conflicto en la tabla predictiva)

- Si una gramática posee *recursividad por la izquierda* o *no está factorizada por la izquierda* entonces tendrá *conflictos* en su tabla predictiva
- Estas condiciones son *necesarias* pero *no* suficientes:
 - Existen gramáticas que no poseen recursividad por la izquierda y que están factorizadas por la izquierda que también presentan conflictos en su tabla predictiva.

Análisis sintáctico descendente predictivo

Conflictos en la tabla predictiva

Ejemplos (Conflicto en la tabla predictiva)

- 1.- *Gramática con recursividad por la izquierda.*
- 2.- *Gramática no factorizada por la izquierda.*
- 3.- *Gramática sin recursividad por la izquierda y factorizada por la izquierda.*

Análisis sintáctico descendente predictivo

Conflictos en la tabla predictiva

Ejemplo (1.- Gramática con recursividad por la izquierda)

 $P = \{$ (1) $S \rightarrow S D$ (2) $S \rightarrow D$ (3) $D \rightarrow T L ;$ (4) $T \rightarrow \text{int}$ (5) $T \rightarrow \text{float}$ (6) $L \rightarrow L , id$ (7) $L' \rightarrow id$ $\}$ Tabla predictiva con **conflictos**

M	Símbolo de entrada					
	;	int	float	,	id	\$
S		1, 2	1, 2			
D						
T						
L						

(1) $S \rightarrow \underbrace{S D}_{\alpha}$ $\text{Primer}(\alpha) = \text{Primer}(S D) = \{ \text{int}, \text{float} \}$ (2) $S \rightarrow \underbrace{D}_{\alpha}$ $\text{Primer}(\alpha) = \text{Primer}(D) = \{ \text{int}, \text{float} \}$

Análisis sintáctico descendente predictivo

Conflictos en la tabla predictiva

Ejemplo (2.- Gramática no factorizada por la izquierda)

 $P = \{$ (1) $S \rightarrow a A$ (2) $A \rightarrow a A$ (3) $A \rightarrow a B$ (4) $B \rightarrow b B$ (5) $B \rightarrow b C$ (6) $C \rightarrow c C$ (7) $C \rightarrow c$ $\}$

Tabla predictiva con conflicto

M	Símbolo de entrada			
	a	b	c	\$
S				
A	2, 3			
B				
C				

(2) $A \rightarrow \underbrace{a A}_{\alpha}$ $Primer(\alpha) = Primer(a A) = \{a\}$ (3) $A \rightarrow \underbrace{a B}_{\alpha}$ $Primer(\alpha) = Primer(a B) = \{a\}$

Análisis sintáctico descendente predictivo

Conflictos en la tabla predictiva

Ejemplo (3.- Gramática sin recursividad por la izquierda y factorizada por la izquierda)

$$P = \{$$

$$(1) S \rightarrow a S$$

$$(2) S \rightarrow A B$$

$$(3) A \rightarrow a A$$

$$(4) A \rightarrow c$$

$$(5) B \rightarrow b B$$

$$(6) B \rightarrow d$$

$$\}$$
Tabla predictiva con **conflictos**

M	Símbolo de entrada				
	a	b	c	d	\$
S	1, 2		2		
A					
B					

$$(1) S \rightarrow \underbrace{a S}_{\alpha}$$

$$\text{Primero}(\alpha) = \text{Primero}(a S) = \{a\}$$

$$(2) S \rightarrow \underbrace{A B}_{\alpha}$$

$$\text{Primero}(\alpha) = \text{Primero}(A B) = \{a, c\}$$

Análisis sintáctico descendente predictivo

Conflictos en la tabla predictiva

Ejercicio (Conflicto de la tabla predictiva)

- Comprueba que la siguiente gramática del *else* danzante genera un **conflicto** en la tabla predictiva:

$$P = \{$$

- (1) $S \rightarrow \text{if } C \text{ then } S S'$
- (2) $S \rightarrow \text{instrucción}$
- (3) $S' \rightarrow \text{else } S$
- (4) $S' \rightarrow \epsilon$
- (5) $C \rightarrow \text{condición}$

$$\}$$

Análisis sintáctico descendente predictivo

Conflictos en la tabla predictiva

Gramática ideal que no genera conflictos

- Una gramática puede generar una tabla predictiva **sin** conflictos de forma **inmediata** si cumple dos condiciones:
 - Está en la **Forma Normal de Greibach**:

$$A \rightarrow \sigma \alpha \in P$$

donde $\sigma \in V_T \wedge \alpha \in V_N^*$

- Para cada símbolo **no** terminal $A \in V_N$, sus alternativas comienzan por un símbolo **terminal diferente**:

$$\text{Si } A \rightarrow \sigma \alpha \mid \sigma' \alpha' \in P \text{ entonces } \sigma \neq \sigma'$$

Análisis sintáctico descendente predictivo

Conflictos en la tabla predictiva

Ejemplo (Gramática ideal que no genera conflictos)

$P = \{$

(1) $S \rightarrow a A B C$

(2) $A \rightarrow a B D$

(3) $A \rightarrow b D C$

(4) $B \rightarrow b B A D$

(5) $B \rightarrow c$

(6) $C \rightarrow d D$

(7) $D \rightarrow c A B$

(8) $D \rightarrow d$

$\}$

Tabla predictiva sin conflictos

M	Símbolo de entrada				
	a	b	c	d	\$
S	1				
A	2	3			
B		4	5		
C				6	
D			7	8	

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
 - Descripción
 - Gramáticas LL(k)
 - Fases
 - Conjunto Primero
 - Conjunto Siguierte
 - Construcción de la tabla predictiva
 - Conflictos en la tabla predictiva
 - Implementación recursiva
 - Implementación iterativa

Análisis sintáctico descendente predictivo

Implementación recursiva

Descripción

- Codifica un **procedimiento** para cada **símbolo no terminal** de la gramática.
- El **código** del procedimiento está basado en la **tabla predictiva**.
- Cada procedimiento determina cuál es la **única** regla de producción que se puede utilizar en cada paso.
- Los procedimientos pueden ser **recursivos** si hay reglas de producción recursivas.

Análisis sintáctico descendente predictivo

Implementación recursiva

Descripción

- Se codifica un **procedimiento** para cada símbolo $A \in V_N$
- Se utiliza la **tabla predictiva** para determinar cuál es la **única regla** de A que se puede **simular** teniendo en cuenta el **símbolo actual** de la cadena de entrada:
 - Si la simulación de una regla es correcta, el procedimiento termina correctamente.
 - En caso contrario, el procedimiento lanza un **error**.
- Si A posee alguna regla **recursiva** entonces el procedimiento es **recursivo**.

Análisis sintáctico descendente predictivo

Implementación recursiva

Nota (Método recursivo sin retroceso)

- *Si la simulación de una regla falla entonces no se prueba con ninguna otra regla.*

Análisis sintáctico descendente predictivo

Implementación recursiva

Simulación de la regla $A \rightarrow X_1X_2 \dots X_n \in P$

Para i desde 1 hasta n hacer

- Si $X_i = B \in V_N$ entonces se llama a la función asociada al símbolo B
 - Si tiene éxito, continúa el análisis.
 - En caso contrario, **fracasa** la simulación de esta regla de A .
- Si $X_i = \sigma \in V_T$ entonces se intenta **emparejar** σ con el componente léxico actual de la cadena de entrada.
 - Si son iguales, el análisis continúa.
 - En caso contrario, **fracasa** la simulación de esta regla de A .

fin_para

Análisis sintáctico descendente predictivo

Implementación recursiva

Nota (Simulación de la regla $A \rightarrow \epsilon$)

- *La simulación de la regla ϵ siempre tiene éxito, ya que se ejecuta una **sentencia nula**.*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplos

- 1.- *Codificación de los procedimientos asociados a los símbolos no terminales de la gramática*
- 2.- *Uso de los procedimientos para analizar una sentencia.*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

1 / 15)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada							
	id	=	+	*	()	número	\$	
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F	9				8		10	

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos) 2 / 15

- Codificación de los **procedimientos** asociados a los símbolos **no terminales**: S , E , E' , T , T' y F .

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

3 / 15)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2		2
E'			3			4	4
T	5				5		5
T'			7	6		7	7
F	9				8		10

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

4 / 15)

Procedimiento S

inicio

*{Regla (1) $S \rightarrow \text{identificador} = E$ }*si (*token = identificador*) entonces *avanzar_entrada(token)*si (*token = "="*) entonces *avanzar_entrada(token)* *Llamada a E*si_no { *token es distinto de "="* } *ERROR*

fin si

...

si_no { *token es otro símbolo* } *ERROR*

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

5 / 15)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F	9				8		10	

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

6 / 15)

Procedimiento E

inicio

{ *Regla (2) $E \rightarrow T E'$* }

si (*token = identificador*) o (*token = número*) o (*token = "("*)

entonces

Llamada a T

Llamada a E'

si_no { *token es otro símbolo* }

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

7 / 15)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada							
	id	=	+	*	()	número	\$	
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F	9				8		10	

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

8 / 15)

Procedimiento E'

inicio

{ Regla (3) $E' \rightarrow + T E'$ }

si (token = "+") entonces

avanzar_entrada(token) *Llamada a T* *Llamada a E'*

si_no { token es distinto de "+" }

...

{ Regla (4) $E' \rightarrow \epsilon$ }

si (token = "\$") o (token = "\$") entonces

{ Sentencia nula }

si_no { token es otro símbolo }

ERROR

fin si

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

9 / 15)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada							
	id	=	+	*	()	número	\$	
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F	9				8		10	

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos)

10 / 15

Procedimiento T

inicio

{Regla (5) $T \rightarrow F T'$ }

si (*token = **identificador***) **o** (*token = **número***) **o** (*token = **"("***)

entonces

Llamada a F

Llamada a T'

si_no

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

11 / 15)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3			4	4
T	5				5	5	
T'			7	6		7	7
F	9				8	10	

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

12 / 15)

Procedimiento T'

inicio

*{ Regla (5) $T' \rightarrow * F T'$ }*si (*token = "*"*)

entonces

avanzar_entrada(token) *Llamada a F* *Llamada a T'*si_no { *token es distinto de "*"* }

...

*{ Regla (6) $T' \rightarrow \epsilon$ }*si (*token = "+"*) o (*token = ")"*) o (*token = \$*) entonces *{ Sentencia nula }*si_no { *token es otro símbolo* }**ERROR**

fin si

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

13 / 15)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada							
	id	=	+	*	()	número	\$	
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F	9				8		10	

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos)

14 / 15

Procedimiento *F*

inicio

{ *Regla (8)* $F \rightarrow (E)$ }

si (*token* = "(") **entonces**

avanzar_entrada(token)

Llamada a E

si (*token* = ")") **entonces**

avanzar_entrada(token)

si_no { *token es distinto de ")"* }

ERROR

fin_si

...

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (Codificación de los procedimientos

15 / 15)

Procedimiento *F* (continuación)

```

si_no { token es distinto de "(" }
    { Regla (9) F → identificador }
    si (token = "id") entonces
        avanzar_entrada(token)
    si_no { token es distinto de "id" }
        { Regla (10) F → número }
        si (token = "número")
            entonces
                avanzar_entrada(token)

```

```

...
si_no { token es otro símbolo }
    ERROR
fin_si

```

```

fin_si
fin

```

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

1 / 75)

- *Análisis de la sentencia:*

identificador = identificador + número * identificador \$

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

2 / 75)

- *Llamada al procedimiento asociado al símbolo inicial S*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

3 / 75)

Análisis de **id** = id + n * id \$

S

Árbol sintáctico

S

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

4 / 75)

Procedimiento S

inicio

{ Regla (1) S → identificador = E }

si (token = identificador) entonces

avanzar_entrada(token)

si (token = "=") entonces

avanzar_entrada(token)

Llamada a E

si_no { token es distinto de "=" }

ERROR

fin si

...

si_no { token es otro símbolo }

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

5 / 75)

Procedimiento S

inicio

`{ Regla (1) S → identificador = E }`si (token = `identificador`) entonces

avanzar_entrada(token)

si (token = "=") entonces

avanzar_entrada(token)

Llamada a E

si_no { token es distinto de "=" }

ERROR

fin si

...

si_no { token es otro símbolo }

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

6 / 75)

Procedimiento S

inicio

{ Regla (1) $S \rightarrow \text{identificador} = E$ }

si (token = **identificador**) entonces

avanzar_entrada(token)

si (token = "=") entonces

avanzar_entrada(token)

Llamada a E

si_no { token es distinto de "=" }

ERROR

fin si

...

si_no { token es otro símbolo }

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

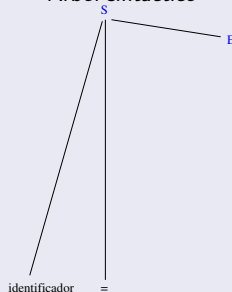
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

7 / 75)

Análisis de $id = id + n * id \$$ $S \xRightarrow{1} \underline{id = E}$ Emparejamientos: $id, =$

Árbol sintáctico



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

8 / 75)

- *Llamada al procedimiento asociado al símbolo E*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

9 / 75)

Procedimiento E

inicio

 $\{ \text{Regla (2) } E \rightarrow T E' \}$ si (token = **identificador**) o (token = **número**) o (token = "(")

entonces

Llamada a T

Llamada a E'

si.no { token es otro símbolo }

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

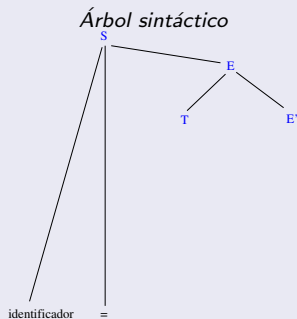
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

10 / 75)

Análisis de $id = id + n * id \$$

$$S \Rightarrow id = E$$

$$\begin{matrix} \Rightarrow \\ 1 \\ \Rightarrow \\ 2 \end{matrix} id = \underline{TE'}$$


Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

11 / 75)

- *Llamada al procedimiento asociado al símbolo T*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

12 / 75)

Procedimiento T

inicio

 $\{ \text{Regla (5) } T \rightarrow F T' \}$ si ($\text{token} = \text{identificador}$) o ($\text{token} = \text{número}$) o ($\text{token} = "("$)

entonces

Llamada a F Llamada a T'

si_no

ERROR

fin si

fin

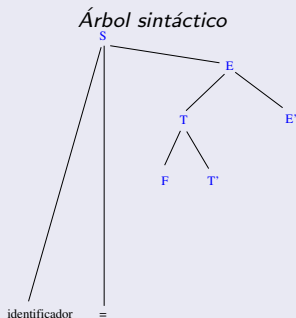
Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

13 / 75)

Análisis de **id = id + n * id \$**

$$\begin{array}{l}
 S \Rightarrow \text{id} = E \\
 \quad \underset{1}{\Rightarrow} \\
 \quad \Rightarrow \text{id} = TE' \\
 \quad \quad \underset{2}{\Rightarrow} \\
 \quad \quad \Rightarrow \text{id} = \underline{FT'E'} \\
 \quad \quad \quad \underset{5}{\Rightarrow}
 \end{array}$$


Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

14 / 75)

- *Llamada al procedimiento asociado al símbolo F*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

15 / 75)

Procedimiento F

inicio $\{ \text{Regla (8) } F \rightarrow (E) \}$ **si** ($\text{token} = "("$) **entonces** $\text{avanzar_entrada}(\text{token})$ $\text{Llamada a } E$ **si** ($\text{token} = ")"$) **entonces** $\text{avanzar_entrada}(\text{token})$ **si_no** { $\text{token es distinto de ")"}$ }**ERROR****fin_si**

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

16 / 75)

Procedimiento F (continuación)

```
si_no { token es distinto de "(" }
```

```
{ Regla (9) F → identificador }
```

```
si (token = "id") entonces
```

```
  avanzar_entrada(token)
```

```
si_no { token es distinto de "id" }
```

```
  { Regla (10) F → número }
```

```
  si (token = "número")
```

```
    entonces
```

```
      avanzar_entrada(token)
```

```
...
```

```
si_no { token es otro símbolo }
```

```
ERROR
```

```
fin_si
```

```
fin_si
```

```
fin
```

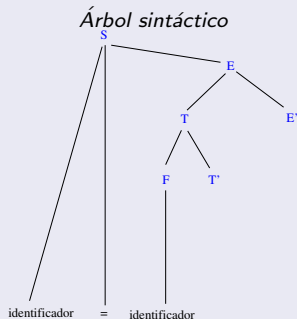
Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

17 / 75)

Análisis de $id = id + n * id \$$

$$\begin{array}{l}
 S \Rightarrow id = E \\
 \xrightarrow{1} \\
 \Rightarrow id = TE' \\
 \xrightarrow{2} \\
 \Rightarrow id = FT'E' \\
 \xrightarrow{5} \\
 \Rightarrow id = \underline{id}T'E' \\
 \xrightarrow{9}
 \end{array}$$
Emparejamiento de id 

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

18 / 75)

- *Fin de la llamada al procedimiento asociado al símbolo F*
- *Se devuelve el control al procedimiento asociado al símbolo T*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

19 / 75)

Procedimiento T

inicio

 $\{ \text{Regla (5) } T \rightarrow F T' \}$ si $(\text{token} = \text{identificador})$ o $(\text{token} = \text{número})$ o $(\text{token} = "(")$

entonces

Llamada a F Llamada a T'

si_no

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

20 / 75)

- *Llamada al procedimiento asociado al símbolo T'*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

21 / 75)

Procedimiento T'

inicio

*{ Regla (5) $T' \rightarrow * F T'$ }*

si (token = "*")

entonces

avanzar_entrada(token)

Llamada a F Llamada a T'

si_no { token es distinto de "*" }

...

{ Regla (6) $T' \rightarrow \epsilon$ }

si (token = "+") o (token = ")")

o (token = \$) entonces

{ Sentencia nula }

si_no { token es otro símbolo }

ERROR

fin si

fin si

fin

Análisis sintáctico descendente predictivo

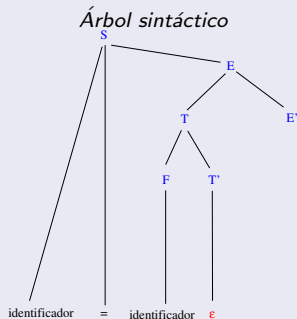
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

22 / 75)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id T'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

23 / 75)

- *Fin de la llamada al procedimiento asociado al símbolo T'*
- *Se devuelve el control al procedimiento asociado al símbolo T*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

24 / 75)

Procedimiento T

inicio

{Regla (5) $T \rightarrow F T'$ }

si ($token = \text{identificador}$) o ($token = \text{número}$) o ($token = "("$)

entonces

Llamada a F

Llamada a T'

si_no

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

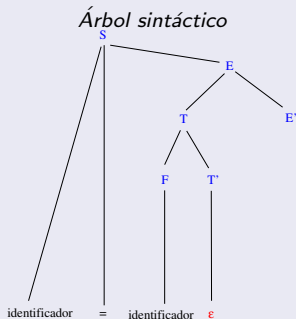
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

25 / 75)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id T'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

26 / 75)

- *Fin de la llamada al procedimiento asociado al símbolo T*
- *Se devuelve el control al procedimiento asociado al símbolo E*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

27 / 75)

Procedimiento E

inicio

 $\{ \text{Regla (2) } E \rightarrow T E' \}$ si (token = **identificador**) o (token = **número**) o (token = "(")

entonces

Llamada a T

Llamada a E'

si_no { token es otro símbolo }

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

28 / 75)

- *Llamada al procedimiento asociado al símbolo E'*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

29 / 75)

Procedimiento E'

inicio

*{ Regla (3) $E' \rightarrow + T E'$ }**si (token = "+") entonces**avanzar_entrada(token)**Llamada a T**Llamada a E'**si_no { token es distinto de "+" }*

...

*{ Regla (4) $E' \rightarrow \epsilon$ }**si (token = "\$") o (token = "\$") entonces**{ Sentencia nula }**si_no { token es otro símbolo }***ERROR***fin si**fin si**fin*

Análisis sintáctico descendente predictivo

Implementación recursiva

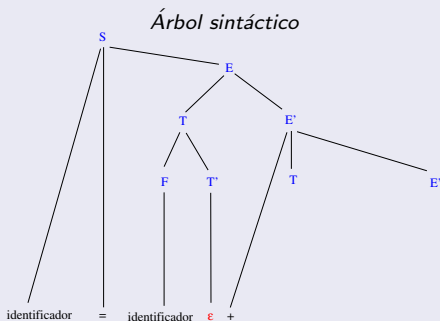
Ejemplo (2.- Análisis de una sentencia

30 / 75)

Análisis de $\text{id} = \text{id} + \text{n} * \text{id} \$$

$S \Rightarrow \text{id} = E$
 $\xrightarrow{1}$
 $\Rightarrow \text{id} = TE'$
 $\xrightarrow{2}$
 $\Rightarrow \text{id} = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow \text{id} = \text{id} T'E'$
 $\xrightarrow{9}$
 $\Rightarrow \text{id} = \text{id} \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow \text{id} = \text{id} \epsilon + TE'$
 $\xrightarrow{3}$

Emparejamiento de +



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

31 / 75)

Procedimiento E'

inicio

{Regla (3) E' → + T E'}

si (token = "+") entonces

avanzar_entrada(token)

*Llamada a T**Llamada a E'*

si_no { token es distinto de "+" }

...

{Regla (4) E' → ε}

si (token = "\$") o (token = "\$") entonces

{Sentencia nula}

si_no { token es otro símbolo }

ERROR

fin si

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

32 / 75)

- *Llamada al procedimiento asociado al símbolo T*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

33 / 75)

Procedimiento T

inicio

 $\{ \text{Regla (5) } T \rightarrow F T' \}$ si ($\text{token} = \text{identificador}$) o ($\text{token} = \text{número}$) o ($\text{token} = "("$)

entonces

Llamada a F Llamada a T'

si_no

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

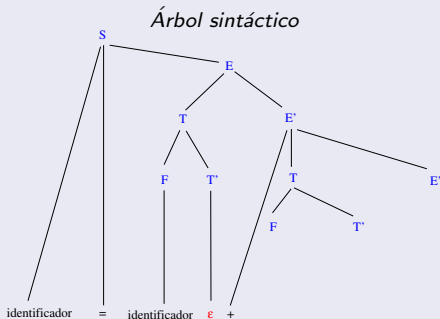
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

34 / 75)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + \underline{FT'E'}$
 $\xrightarrow{5}$



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

35 / 75)

- *Llamada al procedimiento asociado al símbolo F*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

36 / 75)

Procedimiento F

inicio $\{ \text{Regla (8) } F \rightarrow (E) \}$ **si** ($\text{token} = "("$) **entonces***avanzar_entrada(token)**Llamada a E***si** ($\text{token} = ")"$) **entonces***avanzar_entrada(token)***si_no** { *token es distinto de ")"* }**ERROR****fin_si**

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

37 / 75)

Procedimiento F (continuación)**si_no** { token es distinto de "(" }{ Regla (9) $F \rightarrow$ **identificador** }**si** (token = "id") **entonces**

avanzar_entrada(token)

si_no { token es distinto de "id" } { Regla (10) $F \rightarrow$ **número** }**si** (token = "número") **entonces**

avanzar_entrada(token)

...

si_no { token es otro símbolo }**ERROR****fin_si****fin_si****fin**

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

38 / 75)

Procedimiento F (continuación)**si_no** { token es distinto de "(" }{ Regla (9) $F \rightarrow$ **identificador** }

si (token = "id") entonces

avanzar_entrada(token)

si_no { token es distinto de "id" }{ Regla (10) $F \rightarrow$ **número** }

si (token = "número")

entonces

avanzar_entrada(token)

...

si_no { token es otro símbolo }**ERROR****fin_si****fin_si****fin**

Análisis sintáctico descendente predictivo

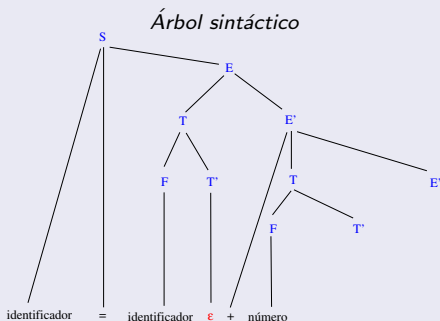
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

39 / 75)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id \epsilon + nT'E'$
 $\xrightarrow{10}$

Emparejamiento de n 

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

40 / 75)

- *Fin de la llamada al procedimiento asociado al símbolo F*
- *Se devuelve el control al procedimiento asociado al símbolo T*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

41 / 75)

Procedimiento T

inicio

 $\{ \text{Regla (5) } T \rightarrow F T' \}$ si $(\text{token} = \text{identificador})$ o $(\text{token} = \text{número})$ o $(\text{token} = "(")$

entonces

Llamada a F Llamada a T'

si_no

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

42 / 75)

- *Llamada al procedimiento asociado al símbolo T'*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

43 / 75)

Procedimiento T'

inicio

*{ Regla (5) $T' \rightarrow * F T'$ }**si (token = "*")**entonces**avanzar_entrada(token)**Llamada a F**Llamada a T'* *si_no { token es distinto de "*" }
...**{ Regla (6) $T' \rightarrow \epsilon$ }**si (token = "+") o (token = " "))**o (token = \$) entonces**{ Sentencia nula }**si_no { token es otro símbolo }**ERROR**fin si**fin si**fin*

Análisis sintáctico descendente predictivo

Implementación recursiva

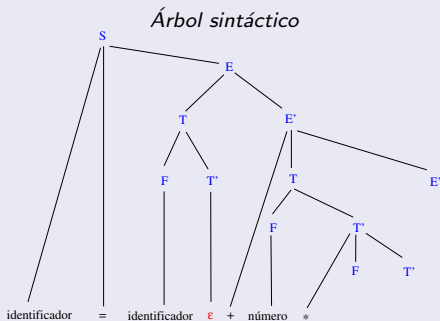
Ejemplo (2.- Análisis de una sentencia

44 / 75)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + nT'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6}$

Emparejamiento de *



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

45 / 75)

Procedimiento T'

inicio

 $\{ \text{Regla (5)} \ T' \rightarrow * F T' \}$

si (token = "*")

entonces

avanzar_entrada(token)

Llamada a F Llamada a T'

si_no { token es distinto de "*" }

...

 $\{ \text{Regla (6)} \ T' \rightarrow \epsilon \}$

si (token = "+") o (token = " "))

o (token = \$) entonces

 $\{ \text{Sentencia nula} \}$

si_no { token es otro símbolo }

ERROR

fin si

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

46 / 75)

- *Llamada al procedimiento asociado al símbolo F*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

47 / 75)

Procedimiento *F*

inicio $\{ \text{Regla (8) } F \rightarrow (E) \}$ **si** (*token* = "(") **entonces***avanzar_entrada(token)**Llamada a E***si** (*token* = ")") **entonces***avanzar_entrada(token)***si_no** { *token es distinto de ")"* }**ERROR****fin_si**

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

48 / 75)

Procedimiento F (continuación)

```
si_no { token es distinto de "(" }
```

```
{ Regla (9) F → identificador }
```

```
si (token = "id") entonces
```

```
  avanzar_entrada(token)
```

```
si_no { token es distinto de "id" }
```

```
  { Regla (10) F → número }
```

```
  si (token = "número")
```

```
    entonces
```

```
      avanzar_entrada(token)
```

```
...
```

```
si_no { token es otro símbolo }
```

```
ERROR
```

```
fin_si
```

```
fin_si
```

```
fin
```

Análisis sintáctico descendente predictivo

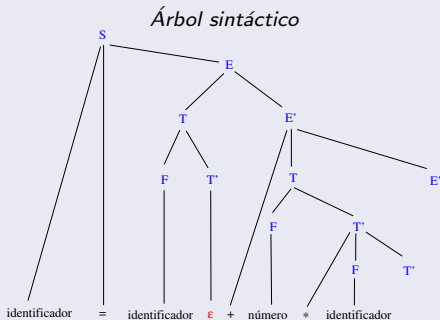
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

49 / 75)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * idT'E'$
 $\xrightarrow{9}$

Emparejamiento de id 

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia 50 / 75)

- *Fin de la llamada al procedimiento asociado al símbolo F*
- *Se devuelve el control al procedimiento asociado al símbolo T'*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

51 / 75)

Procedimiento T'

inicio

*{ Regla (5) $T' \rightarrow * F T'$ }*

si (token = "*")

entonces

avanzar_entrada(token)

Llamada a F Llamada a T'

si_no { token es distinto de "*" }

...

{ Regla (6) $T' \rightarrow \epsilon$ }

si (token = "+") o (token = " ")

o (token = \$) entonces

{ Sentencia nula }

si_no { token es otro símbolo }

ERROR

fin si

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

52 / 75)

- *Llamada recursiva al procedimiento asociado al símbolo T' (segunda activación)*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

53 / 75)

Procedimiento T'

inicio

*{ Regla (5) $T' \rightarrow * F T'$ }*

si (token = "*")

entonces

*avanzar_entrada(token)**Llamada a F* *Llamada a T'*

si_no { token es distinto de "*" }

...

{ Regla (6) $T' \rightarrow \epsilon$ }

si (token = "+") o (token = ")")

o (token = \$) entonces

{ Sentencia nula }

si_no { token es otro símbolo }

ERROR

fin si

fin si

fin

Análisis sintáctico descendente predictivo

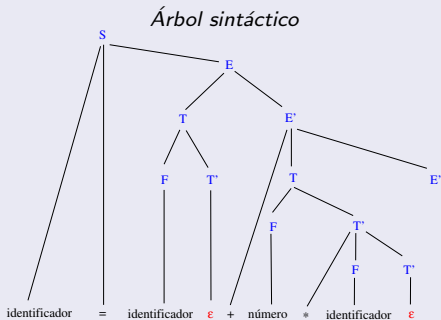
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

54 / 75)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * id T'E'$
 $\xrightarrow{9} id = id \epsilon + n * id \epsilon E'$
 $\xrightarrow{7}$



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

55 / 75)

- *Fin de la segunda activación del procedimiento asociado al símbolo T'*
- *Se devuelve el control a la primera activación del procedimiento asociado al símbolo T'*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

56 / 75)

Procedimiento T'

inicio

*{ Regla (5) $T' \rightarrow * F T'$ }**si (token = "*")*

entonces

*avanzar_entrada(token)**Llamada a F* *Llamada a T'* *si_no { token es distinto de "*" }*

...

*{ Regla (6) $T' \rightarrow \epsilon$ }**si (token = "+") o (token = ")")**o (token = "\$) entonces**{ Sentencia nula }**si_no { token es otro símbolo }***ERROR***fin si**fin si**fin*

Análisis sintáctico descendente predictivo

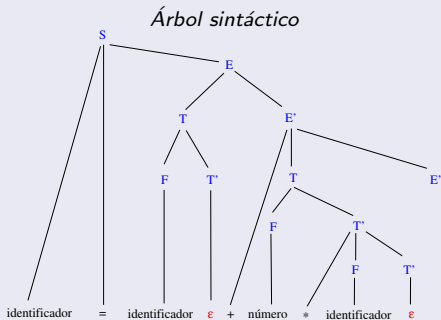
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

57 / 75)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * id T'E'$
 $\xrightarrow{9} id = id \epsilon + n * id \epsilon E'$
 $\xrightarrow{7}$



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

58 / 75)

- *Fin de la primera activación del procedimiento asociado al símbolo T'*
- *Se devuelve el control al procedimiento asociado al símbolo T*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

59 / 75)

Procedimiento T

inicio

 $\{ \text{Regla (5) } T \rightarrow F T' \}$ si ($\text{token} = \text{identificador}$) o ($\text{token} = \text{número}$) o ($\text{token} = "("$)

entonces

Llamada a F *Llamada a T'*

si_no

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

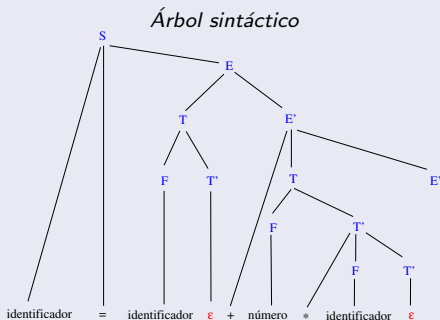
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

60 / 75)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * id T'E'$
 $\xrightarrow{9} id = id \epsilon + n * id \epsilon E'$
 $\xrightarrow{7}$



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

61 / 75)

- *Fin de la ejecución del procedimiento asociado al símbolo T*
- *Se devuelve el control al procedimiento asociado al símbolo E'*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

62 / 75)

Procedimiento E'

inicio

{Regla (3) E' → + T E'}

si (token = "+") entonces

avanzar_entrada(token)

Llamada a T

Llamada a E'

si_no { token es distinto de "+" }

...

{Regla (4) E' → ε}

si (token = "\$") o (token = \$) entonces

{Sentencia nula}

si_no { token es otro símbolo }

ERROR

fin si

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

63 / 75)

- *LLamada recursiva al procedimiento asociado al símbolo E' (segunda activación)*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

64 / 75)

Procedimiento E'

inicio

 $\{ \text{Regla (3) } E' \rightarrow + T E' \}$

si (token = "+") entonces

avanzar_entrada(token)

Llamada a T Llamada a E'

si_no { token es distinto de "+" }

...

 $\{ \text{Regla (4) } E' \rightarrow \epsilon \}$

si (token = "\$") o (token = \$) entonces

 $\{ \text{Sentencia nula} \}$

si_no { token es otro símbolo }

ERROR

fin si

fin si

fin

Análisis sintáctico descendente predictivo

Implementación recursiva

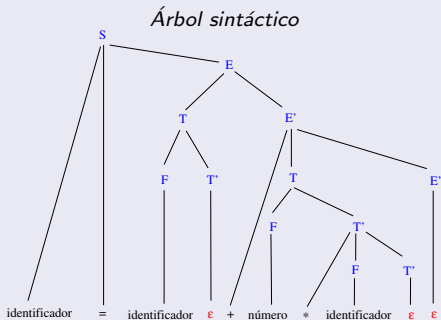
Ejemplo (2.- Análisis de una sentencia

65 / 75)

Análisis de $\text{id} = \text{id} + \text{n} * \text{id} \$$

S

\Rightarrow	1	$\text{id} = E$
\Rightarrow	2	$\text{id} = TE'$
\Rightarrow	5	$\text{id} = FT'E'$
\Rightarrow	9	$\text{id} = \text{id}T'E'$
\Rightarrow	7	$\text{id} = \text{id} \epsilon E'$
\Rightarrow	3	$\text{id} = \text{id} \epsilon + TE'$
\Rightarrow	5	$\text{id} = \text{id} \epsilon + FT'E'$
\Rightarrow	10	$\text{id} = \text{id} \epsilon + \text{n}T'E'$
\Rightarrow	6	$\text{id} = \text{id} \epsilon + \text{n} * FT'E'$
\Rightarrow	9	$\text{id} = \text{id} \epsilon + \text{n} * \text{id}T'E'$
\Rightarrow	7	$\text{id} = \text{id} \epsilon + \text{n} * \text{id} \epsilon E'$
\Rightarrow	4	$\text{id} = \text{id} \epsilon + \text{n} * \text{id} \epsilon \epsilon$



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

66 / 75)

- *Fin de la ejecución del procedimiento asociado al símbolo E' (segunda activación)*
- *Se devuelve el control al procedimiento asociado al símbolo E' (primera activación)*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

67 / 75)

Procedimiento E'

inicio

*{ Regla (3) $E' \rightarrow + T E'$ }**si (token = "+") entonces**avanzar_entrada(token)**Llamada a T**Llamada a E'**si_no { token es distinto de "+" }*

...

*{ Regla (4) $E' \rightarrow \epsilon$ }**si (token = "\$") o (token = \$) entonces**{ Sentencia nula }**si_no { token es otro símbolo }***ERROR***fin si**fin si***fin**

Análisis sintáctico descendente predictivo

Implementación recursiva

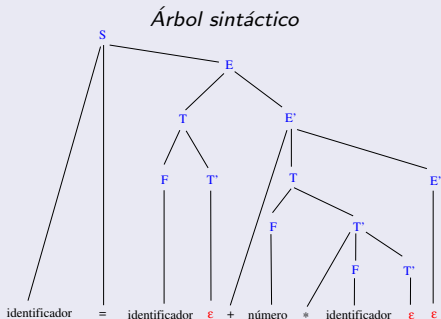
Ejemplo (2.- Análisis de una sentencia

68 / 75)

Análisis de $id = id + n * id \$$

S

\Rightarrow	1	$id = E$
\Rightarrow	2	$id = TE'$
\Rightarrow	5	$id = FT'E'$
\Rightarrow	9	$id = idT'E'$
\Rightarrow	7	$id = id \in E'$
\Rightarrow	3	$id = id \in + TE'$
\Rightarrow	5	$id = id \in + FT'E'$
\Rightarrow	10	$id = id \in + nT'E'$
\Rightarrow	6	$id = id \in + n * FT'E'$
\Rightarrow	9	$id = id \in + n * idT'E'$
\Rightarrow	7	$id = id \in + n * id \in E'$
\Rightarrow	4	$id = id \in + n * id \in \underline{\epsilon}$



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

69 / 75)

- *Fin de la ejecución del procedimiento asociado al símbolo E' (primera activación)*
- *Se devuelve el control al procedimiento asociado al símbolo E*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

70 / 75)

Procedimiento E

inicio

 $\{ \text{Regla (2) } E \rightarrow T E' \}$ si (token = **identificador**) o (token = **número**) o (token = "(")

entonces

Llamada a T

Llamada a E'

si_no { token es otro símbolo }

ERROR

fin si

fin

Análisis sintáctico descendente predictivo

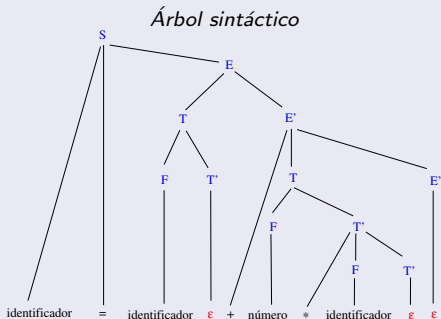
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

71 / 75)

Análisis de $id = id + n * id \$$

S \Rightarrow $id = E$
 $\xrightarrow{1}$
 \Rightarrow $id = TE'$
 $\xrightarrow{2}$
 \Rightarrow $id = FT'E'$
 $\xrightarrow{5}$
 \Rightarrow $id = idT'E'$
 $\xrightarrow{9}$
 \Rightarrow $id = id \in E'$
 $\xrightarrow{7}$
 \Rightarrow $id = id \in + TE'$
 $\xrightarrow{3}$
 \Rightarrow $id = id \in + FT'E'$
 $\xrightarrow{5}$
 \Rightarrow $id = id \in + n T'E'$
 $\xrightarrow{10}$
 \Rightarrow $id = id \in + n * FT'E'$
 $\xrightarrow{6}$
 \Rightarrow $id = id \in + n * id T'E'$
 $\xrightarrow{9}$
 \Rightarrow $id = id \in + n * id \in E'$
 $\xrightarrow{7}$
 \Rightarrow $id = id \in + n * id \in \in$
 $\xrightarrow{4}$



Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

72 / 75)

- *Fin de la ejecución del procedimiento asociado al símbolo E*
- *Se devuelve el control al procedimiento asociado al símbolo S*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

73 / 75)

Procedimiento S

inicio

*{ Regla (1) S → identificador = E }**si (token = identificador) entonces**avanzar_entrada(token)**si (token = "=") entonces**avanzar_entrada(token)**Llamada a E**si_no { token es distinto de "=" }**ERROR**fin si*

...

*si_no { token es otro símbolo }**ERROR**fin si**fin*

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

74 / 75)

- *Fin de la ejecución del procedimiento asociado al símbolo S*

Análisis sintáctico descendente predictivo

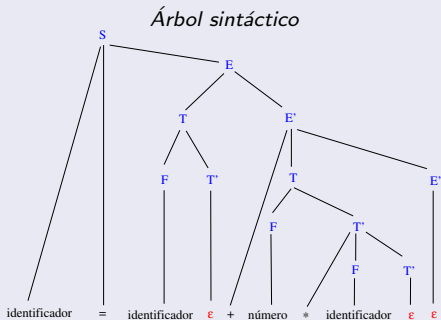
Implementación recursiva

Ejemplo (2.- Análisis de una sentencia

75 / 75)

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * id T'E'$
 $\xrightarrow{9} id = id \epsilon + n * id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + n * id \epsilon \epsilon$
 $\xrightarrow{4}$

Fin del análisis

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejercicios (Método recursivo)

1.- *Gramática de las expresiones.*

id = número * (identificador + identificador) \$

2.- *Gramática de las declaraciones.*

int id, id; float id \$

3.- *Gramática de prototipos*

int id (int id, float id) ; \$

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejercicio (1.- Gramática de expresiones aritméticas)

- *Utiliza el método recursivo para realizar el análisis descendente y predictivo de la sentencia:*

id = número * (identificador + identificador) \$

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejercicio (2.- Gramática de declaraciones)

- *Gramática de declaraciones*

$P = \{$

(1) $S \rightarrow D S$

(2) $S \rightarrow \epsilon$

(3) $D \rightarrow T L ;$

(4) $T \rightarrow \mathbf{int}$

(5) $T \rightarrow \mathbf{float}$

(6) $L \rightarrow \mathbf{id} L'$

(7) $L' \rightarrow , \mathbf{id} L'$

(8) $L' \rightarrow \epsilon$

$\}$

- *Análisis de las declaraciones*

int id, id; float id \$

Análisis sintáctico descendente predictivo

Implementación recursiva

Ejercicio (3.- Gramática de prototipos)

- Gramática de prototipos

$$P = \{$$

(1) $S \rightarrow D S$

(2) $S \rightarrow \epsilon$

(3) $D \rightarrow T \text{ id } (P) ;$

(4) $T \rightarrow \text{int}$

(5) $T \rightarrow \text{float}$

(6) $P \rightarrow \epsilon$

(7) $P \rightarrow T \text{ id } L$

(8) $L \rightarrow , T \text{ id } L$

(9) $L \rightarrow \epsilon$

$$\}$$

- Construcción del conjunto

Primero

- Construcción del conjunto

Siguiente

- Construcción de la **Tabla predictiva**

- Análisis del prototipo

int id (int id, float id) ; \$

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
 - Descripción
 - Gramáticas LL(k)
 - Fases
 - Conjunto Primero
 - Conjunto Siguierte
 - Construcción de la tabla predictiva
 - Conflictos en la tabla predictiva
 - Implementación recursiva
 - Implementación iterativa

Análisis sintáctico descendente predictivo

Implementación iterativa

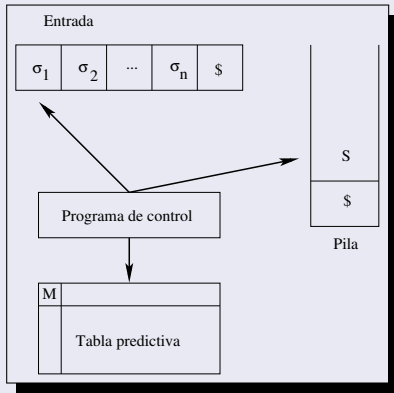
Componentes

- La **implementación iterativa** del análisis sintáctico descendente y predictivo consta de cuatro componentes:
 - 1.- Entrada.
 - 2.- Tabla predictiva.
 - 3.- Pila del análisis sintáctico.
 - 4.- Programa de control.

Análisis sintáctico descendente predictivo

Implementación iterativa

Componentes



Análisis sintáctico descendente predictivo

Implementación iterativa

Componentes: representación tabular

Pila	Entrada	Acción
\$ S	$\sigma_1 \sigma_2 \dots \sigma_n$ \$	
...
\$	\$	Aceptar

Acción (Acción)

Esta columna será utilizada para indicar cada una de las acciones que realizará el Programa de control.

Análisis sintáctico descendente predictivo

Implementación iterativa

Componentes: representación tabular

Pila	Entrada	Acción
\$ S	$\sigma_1 \sigma_2 \dots \sigma_n$ \$	
...
\$	\$	Aceptar

Nota (Acción)

Esta columna será utilizada para indicar cada una de las acciones que realizará el Programa de control.

Análisis sintáctico descendente predictivo

Implementación iterativa

1.- Entrada

1 / 2

- Contiene los **componentes léxicos** o **tokens** que ha reconocido el analizador léxico.
- **\$**: indica el fin de la cadena de entrada.

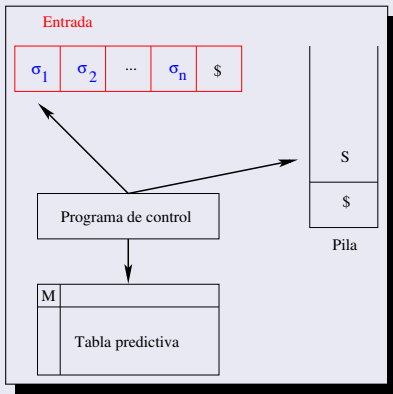
Pila	Entrada	Acción
\$ S	$\sigma_1 \sigma_2 \dots \sigma_n$ \$	
...
\$	\$	Aceptar

Análisis sintáctico descendente predictivo

Implementación iterativa

1.- Entrada

2 / 2



Análisis sintáctico descendente predictivo

Implementación iterativa

2.- Tabla predictiva

1 / 2

- Se genera a partir de la gramática G utilizada en el análisis sintáctico.

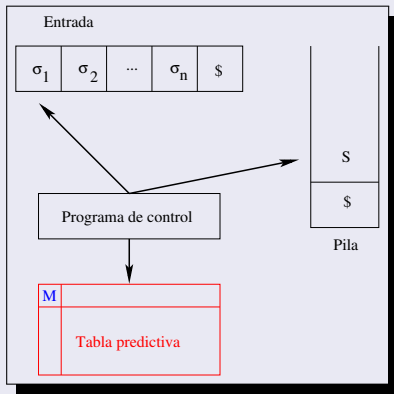
M	Símbolo de entrada					
	σ_1	...	σ_j	...	σ_m	$\$$
S						
A_1						
...						
A_i			Regla k			
...						
A_n						

Análisis sintáctico descendente predictivo

Implementación iterativa

2.- Tabla predictiva

2 / 2



Análisis sintáctico descendente predictivo

Implementación iterativa

3.- Pila del análisis sintáctico

1 / 2

- Puede contener símbolos de la gramática o el símbolo \$
- Configuración inicial:
 - \$: situado en el fondo de la pila
 - Símbolo inicial de la gramática S: situado encima de \$.

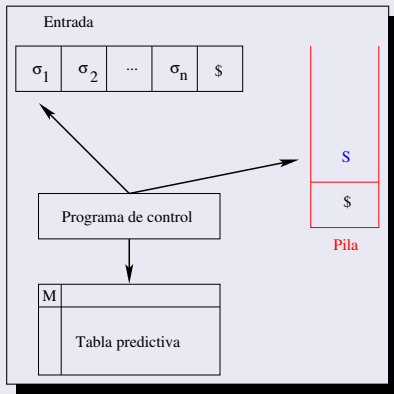
Pila	Entrada	Acción
\$ S	$\sigma_1 \sigma_2 \cdots \sigma_n$ \$	
...

Análisis sintáctico descendente predictivo

Implementación iterativa

3.- Pila del análisis sintáctico

2 / 2



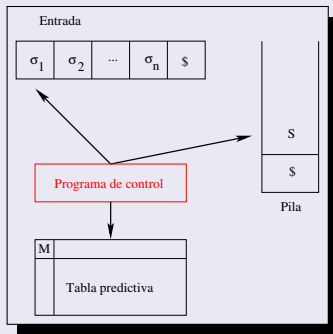
Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

1 / 15

- Guía el proceso de análisis.



Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

2 / 15

- M : tabla predictiva
- $X \in V_N \cup V_T \cup \{\$\}$: símbolo situado en la cima de la pila.
- $\sigma \in V_T \cup \{\$\}$: símbolo actual de la cadena de entrada.

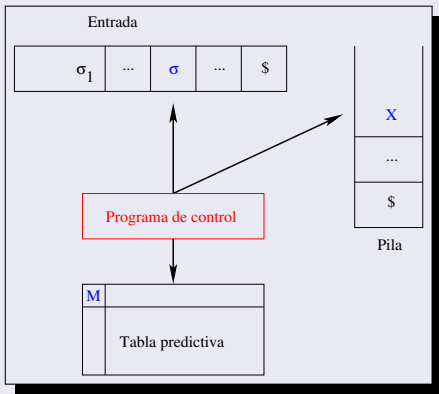
Pila	Entrada	Acción
$\$ S$	$\sigma_1 \sigma_2 \cdots \sigma_n \$$	
...
$\$ \cdots X$	$\sigma \cdots \sigma_n \$$	
...

Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

3 / 15



Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

4 / 15

1.- $X \in V_T \cup \{\$\}$

a) $X = \sigma = \$$

Aceptar: la cadena de entrada es aceptada y finaliza el análisis con éxito.

b) $X = \sigma \neq \$$

Emparejar: se **elimina** el símbolo σ de la cima de la pila y se **avanza** en la entrada.

c) $X = \sigma \wedge \sigma \neq \sigma'$

Error: el símbolo terminal de la cima de la pila **no coincide** con el símbolo terminal de la entrada.

Se debe aplicar un método de recuperación de errores.

Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

5 / 15

2.- $X = A \in V_N$

a) $M[A, \sigma] = A \rightarrow X_1 X_2 \cdots X_N$

Se **extrae** el símbolo A de la pila y se **introducen** los símbolos X_i ($i \in \{1, 2, \dots, N\}$) en orden **inverso**.

b) $M[A, \sigma] = \text{vacía}$

Error: no se puede aplicar ninguna regla de producción.
Se debe aplicar un método de recuperación de errores.

Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

6 / 15

$$1.- X \in V_T \cup \{\$\}$$

$$a) X = \sigma = \$$$

Aceptar: la cadena de entrada es aceptada y finaliza el análisis con éxito.

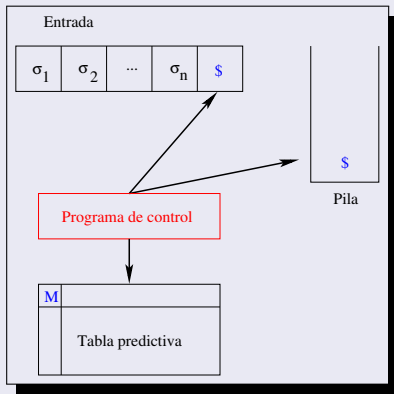
Pila	Entrada	Acción
\$ S	$\sigma_1 \sigma_2 \dots \sigma_n \$$	
...
\$	\$	Aceptar

Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

7 / 15



Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

8 / 15

1.- $X \in V_T \cup \{\$\}$

b) $X = \sigma \neq \$$

Emparejar: se **elimina** el símbolo σ de la cima de la pila y se **avanza** en la entrada.

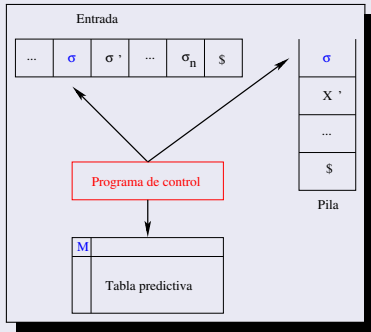
Pila	Entrada	Acción
$\$ \dots X' \sigma$	$\sigma \sigma' \dots \sigma_n \$$	Emparejar
$\$ \dots X'$	$\sigma' \dots \sigma_n \$$	
...

Análisis sintáctico descendente predictivo

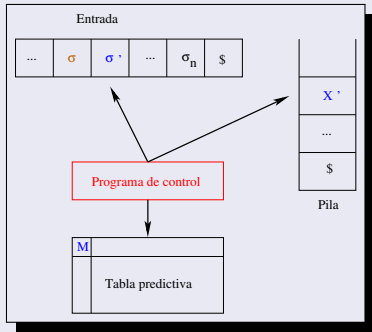
Implementación iterativa

4.- Programa de control

9 / 15



(a)



(b)

Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

10 / 15

1.- $X \in V_T \cup \{\$\}$

c) $X = \sigma \wedge \sigma \neq \sigma'$

Error: no coinciden los símbolos actuales de la entrada y la pila.
Se debe aplicar un **método de recuperación de errores**.

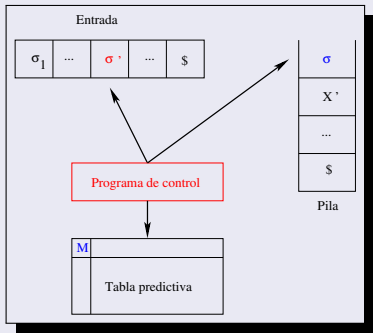
Pila	Entrada	Acción
$\$ \dots X' \sigma$	$\sigma' \dots \sigma_n \$$	Error
...

Análisis sintáctico descendente predictivo

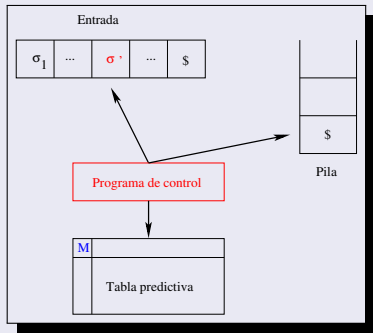
Implementación iterativa

4.- Programa de control

11 / 15



(a)



(b)

Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

12 / 15

2.- $X = A \in V_N$ a) $M[A, \sigma] = \text{regla } k$ donde la **regla** k es $A \rightarrow X_1 X_2 \cdots X_N$ Se **extrae** el símbolo A de la pila y se **introducen** los símbolos X_i ($i \in \{1, 2, \dots, N\}$) en orden **inverso**.

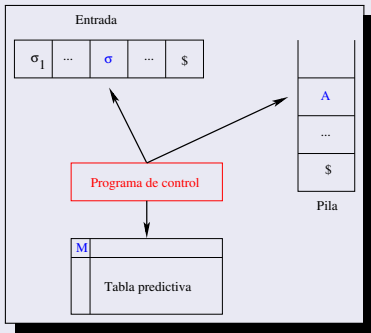
Pila	Entrada	Acción
$\$ \cdots A$	$\sigma \cdots \sigma_n \$$	k) $A \rightarrow X_1 X_2 \cdots X_N$
$\$ \cdots \underline{X_N \cdots X_2 X_1}$	$\sigma \cdots \sigma_n \$$...
...

Análisis sintáctico descendente predictivo

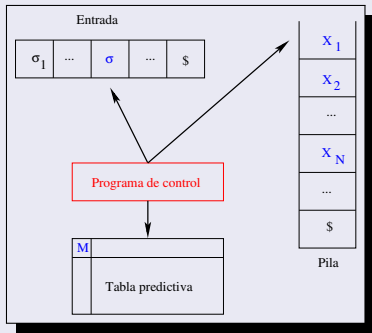
Implementación iterativa

4.- Programa de control

13 / 15



(a)



(b)

Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

14 / 15

2.- $X = A \in V_N$

b) $M[A, \sigma] = \text{vacía}$

Error: no se puede aplicar ninguna regla de producción.
Se debe aplicar un método de **recuperación de errores**.

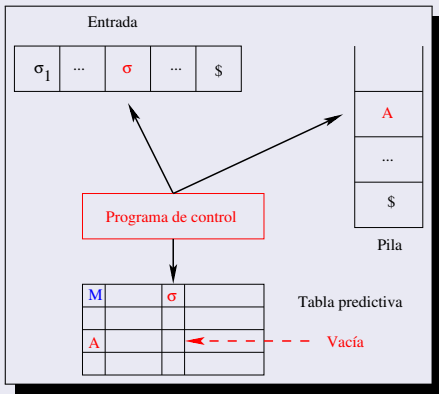
Pila	Entrada	Acción
\$... A	$\sigma \cdots \sigma_n$ \$	Error
...

Análisis sintáctico descendente predictivo

Implementación iterativa

4.- Programa de control

15 / 15



Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

1 / 40

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

• Análisis de la sentencia:

 $\text{id} = \text{id} + \text{n} * \text{id} \$$

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

2 / 40

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3			4	4
T	5				5	5	
T'			7	6		7	7
F	9				8	10	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas 3 / 40)

Pila	Entrada	Acción
\$ S	id = id + n * id \$	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas

4 / 40)

Análisis de **id = id + n * id \$**

S

Árbol sintáctico

S

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

5 / 40

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3			4	4
T	5				5	5	
T'			7	6		7	7
F	9				8	10	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

6 / 40

Pila	Entrada	Acción
\$ S	id = id + n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = id + n * id \$	<i>Emparejar</i>
\$ E =	= id + n * id \$	<i>Emparejar</i>
\$ E	id + n * id \$	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

6 / 40

Pila	Entrada	Acción
\$ S	id = id + n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = id + n * id \$	<i>Emparejar</i>
\$ E =	= id + n * id \$	<i>Emparejar</i>
\$ E =	id + n * id \$	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

6 / 40

Pila	Entrada	Acción
\$ S	id = id + n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = id + n * id \$	<i>Emparejar</i>
\$ E =	= id + n * id \$	<i>Emparejar</i>
\$ E	id + n * id \$	

Análisis sintáctico descendente predictivo

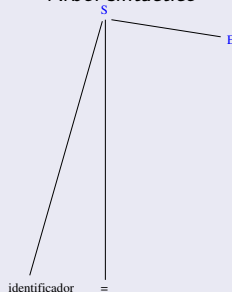
Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

7 / 40

Análisis de `id = id + n * id $` $S \xRightarrow{1} \underline{id = E}$ Emparejamientos: `id, =`

Árbol sintáctico



Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

8 / 40

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2		2
E'			3			4	4
T	5				5		5
T'			7	6		7	7
F	9				8		10

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

9 / 40)

Pila	Entrada	Acción
\$ S	id = id + n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = id + n * id \$	<i>Emparejar</i>
\$ E =	= id + n * id \$	<i>Emparejar</i>
\$ E	id + n * id \$	2) $E \rightarrow T E'$
\$ E' T	id + n * id \$	

Análisis sintáctico descendente predictivo

Implementación iterativa

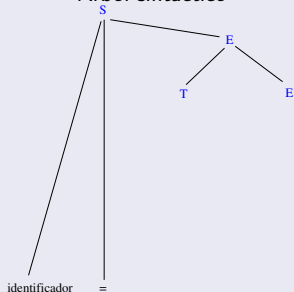
Ejemplo (Gramática de expresiones aritméticas)

10 / 40

Análisis de **id = id + n * id \$**

$$\begin{array}{l}
 S \Rightarrow \text{id} = E \\
 \quad \underset{1}{\Rightarrow} \\
 \quad \underset{2}{\Rightarrow} \text{id} = \underline{TE'}
 \end{array}$$

Árbol sintáctico



Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

11 / 40

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3			4	4
T	5				5	5	
T'			7	6		7	7
F	9				8	10	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

12 / 40)

Pila	Entrada	Acción
\$ <u>S</u>	id = id + n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = id + n * id \$	<i>Emparejar</i>
\$ <u>E =</u>	= id + n * id \$	<i>Emparejar</i>
\$ <u>E</u>	id + n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id + n * id \$	5) $T \rightarrow F T'$
\$ <u>E' T' F</u>	id + n * id \$	

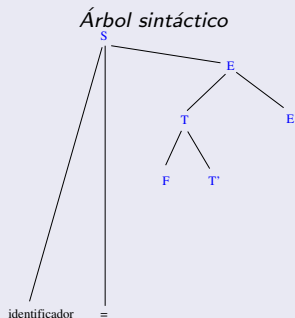
Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

13 / 40

Análisis de **id = id + n * id \$**

$$\begin{array}{l}
 S \Rightarrow \text{id} = E \\
 \quad \underset{1}{\Rightarrow} \\
 \quad \Rightarrow \text{id} = TE' \\
 \quad \quad \underset{2}{\Rightarrow} \\
 \quad \quad \Rightarrow \text{id} = \underline{FT'E'} \\
 \quad \quad \quad \underset{5}{\Rightarrow}
 \end{array}$$


Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

14 / 40

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2		2
E'			3			4	4
T	5				5		5
T'			7	6		7	7
F	9				8		10

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

15 / 40

Pila	Entrada	Acción
\$ <u>S</u>	id = id + n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = id + n * id \$	<i>Emparejar</i>
\$ <u>E =</u>	= id + n * id \$	<i>Emparejar</i>
\$ <u>E</u>	id + n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id + n * id \$	5) $T \rightarrow F T'$
\$ <u>E' T' F</u>	id + n * id \$	9) $F \rightarrow \text{identificador}$
\$ <u>E' T' id</u>	id + n * id \$	<i>Emparejar</i>
\$ <u>E' T'</u>	+ n * id \$	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

15 / 40

Pila	Entrada	Acción
\$ <u>S</u>	id = id + n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = id + n * id \$	<i>Emparejar</i>
\$ <u>E =</u>	= id + n * id \$	<i>Emparejar</i>
\$ <u>E</u>	id + n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id + n * id \$	5) $T \rightarrow F T'$
\$ <u>E' T' F</u>	id + n * id \$	9) $F \rightarrow \text{identificador}$
\$ <u>E' T' id</u>	id + n * id \$	<i>Emparejar</i>
\$ <u>E' T'</u>	+ n * id \$	

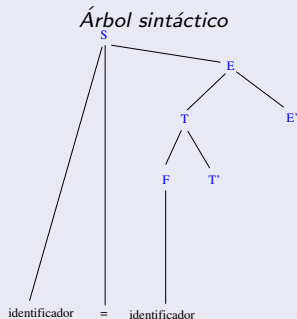
Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

16 / 40

Análisis de $id = id + n * id \$$

$$\begin{array}{l}
 S \Rightarrow id = E \\
 \xrightarrow{1} \\
 \Rightarrow id = TE' \\
 \xrightarrow{2} \\
 \Rightarrow id = FT'E' \\
 \xrightarrow{5} \\
 \Rightarrow id = \underline{id}T'E' \\
 \xrightarrow{9}
 \end{array}$$
Emparejamiento de id 

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

17 / 40

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3			4	4
T	5				5	5	
T'			7	6		7	7
F	9				8	10	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

18 / 40)

Pila	Entrada	Acción
\$ S	$id = id + n * id \$$	1) $S \rightarrow \text{identificador} = E$
\$ $E = id$	$id = id + n * id \$$	<i>Emparejar</i>
\$ $E =$	$= id + n * id \$$	<i>Emparejar</i>
\$ E	$id + n * id \$$	2) $E \rightarrow T E'$
\$ $E' T$	$id + n * id \$$	5) $T \rightarrow F T'$
\$ $E' T' F$	$id + n * id \$$	9) $F \rightarrow \text{identificador}$
\$ $E' T' id$	$id + n * id \$$	<i>Emparejar</i>
\$ $E' T'$	$+ n * id \$$	7) $T' \rightarrow \epsilon$
\$ E'	$+ n * id \$$	

Análisis sintáctico descendente predictivo

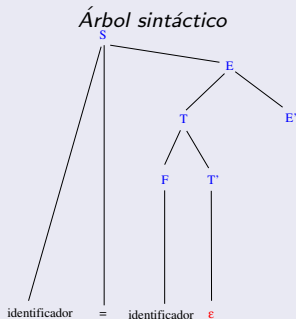
Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

19 / 40

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $id = TE'$
 $\xrightarrow{2}$
 $id = FT'E'$
 $\xrightarrow{5}$
 $id = id T'E'$
 $\xrightarrow{9}$
 $id = id \epsilon E'$
 $\xrightarrow{7}$



Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

20 / 40)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3			4	4
T	5				5	5	
T'			7	6		7	7
F	9				8	10	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

21 / 40

Pila	Entrada	Acción
\$ <u>S</u>	id = id + n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = id + n * id \$	<i>Emparejar</i>
\$ <u>E =</u>	= id + n * id \$	<i>Emparejar</i>
\$ <u>E</u>	id + n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id + n * id \$	5) $T \rightarrow F T'$
\$ <u>E' T' F</u>	id + n * id \$	9) $F \rightarrow \text{identificador}$
\$ <u>E' T' id</u>	id + n * id \$	<i>Emparejar</i>
\$ <u>E' T'</u>	+ n * id \$	7) $T' \rightarrow \epsilon$
\$ <u>E'</u>	+ n * id \$	3) $E' \rightarrow + T E'$
\$ <u>E' T +</u>	+ n * id \$	<i>Emparejar</i>
\$ <u>E'</u>	n * id \$	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

21 / 40

Pila	Entrada	Acción
\$ <u>S</u>	id = id + n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = id + n * id \$	<i>Emparejar</i>
\$ <u>E =</u>	= id + n * id \$	<i>Emparejar</i>
\$ <u>E</u>	id + n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id + n * id \$	5) $T \rightarrow F T'$
\$ <u>E' T' F</u>	id + n * id \$	9) $F \rightarrow \text{identificador}$
\$ <u>E' T' id</u>	id + n * id \$	<i>Emparejar</i>
\$ <u>E' T'</u>	+ n * id \$	7) $T' \rightarrow \epsilon$
\$ <u>E'</u>	+ n * id \$	3) $E' \rightarrow + T E'$
\$ <u>E' T +</u>	+ n * id \$	<i>Emparejar</i>
\$ <u>E' T</u>	n * id \$	

Análisis sintáctico descendente predictivo

Implementación iterativa

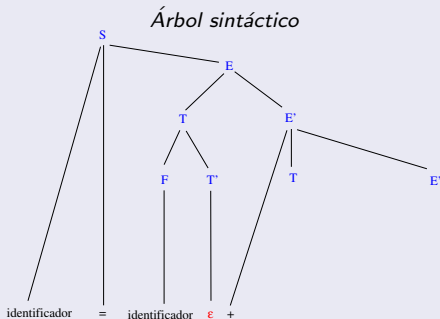
Ejemplo (Gramática de expresiones aritméticas)

22 / 40

Análisis de $\text{id} = \text{id} + \text{id} * \text{id} \$$

$S \Rightarrow \text{id} = E$
 $\xrightarrow{1}$
 $\Rightarrow \text{id} = TE'$
 $\xrightarrow{2}$
 $\Rightarrow \text{id} = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow \text{id} = \text{id} T'E'$
 $\xrightarrow{9}$
 $\Rightarrow \text{id} = \text{id} \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow \text{id} = \text{id} \epsilon + TE'$
 $\xrightarrow{3}$

Emparejamiento de +



Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

23 / 40)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3		4		4
T	5				5	5	
T'			7	6	7		7
F	9				8	10	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

24 / 40)

Pila	Entrada	Acción
...
\$ <u>E'</u> T +	+ n * id \$	<i>Emparejar</i>
\$ E' <u>T</u>	n * id \$	5) $T \rightarrow F T'$
\$ E' T' <u>F</u>	n * id \$	

Análisis sintáctico descendente predictivo

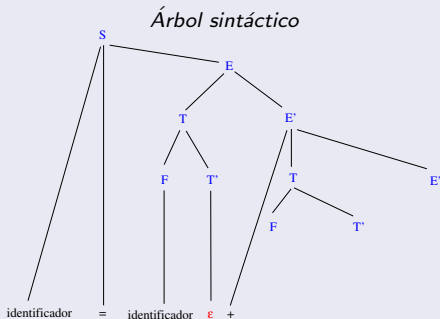
Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

25 / 40

Análisis de $\text{id} = \text{id} + \text{n} * \text{id} \$$

$S \Rightarrow \text{id} = E$
 $\xrightarrow{1}$
 $\Rightarrow \text{id} = TE'$
 $\xrightarrow{2}$
 $\Rightarrow \text{id} = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow \text{id} = \text{id}T'E'$
 $\xrightarrow{9}$
 $\Rightarrow \text{id} = \text{id} \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow \text{id} = \text{id} \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow \text{id} = \text{id} \epsilon + \underline{FT'E'}$
 $\xrightarrow{5}$



Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

26 / 40)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2		2
E'			3			4	4
T	5				5		5
T'			7	6		7	7
F	9				8		10

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

27 / 40)

Pila	Entrada	Acción
...
\$ <u>$E' T +$</u>	$+ n * id \$$	<i>Emparejar</i>
\$ <u>$E' T$</u>	$n * id \$$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	$n * id \$$	10) $F \rightarrow \text{número}$
\$ <u>$E' T' n$</u>	$n * id \$$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	$* id \$$	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

27 / 40)

Pila	Entrada	Acción
...
\$ $E' T +$	$+ n * id \$$	<i>Emparejar</i>
\$ $E' T$	$n * id \$$	5) $T \rightarrow F T'$
\$ $E' T' F$	$n * id \$$	10) $F \rightarrow \text{número}$
\$ $E' T' n$	$n * id \$$	<i>Emparejar</i>
\$ $E' T'$	$* id \$$	

Análisis sintáctico descendente predictivo

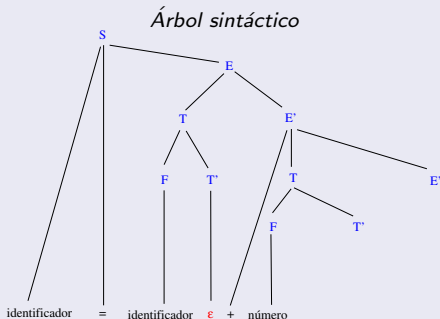
Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

28 / 40

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1}$
 $\Rightarrow id = TE'$
 $\xrightarrow{2}$
 $\Rightarrow id = FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = idT'E'$
 $\xrightarrow{9}$
 $\Rightarrow id = id \epsilon E'$
 $\xrightarrow{7}$
 $\Rightarrow id = id \epsilon + TE'$
 $\xrightarrow{3}$
 $\Rightarrow id = id \epsilon + FT'E'$
 $\xrightarrow{5}$
 $\Rightarrow id = id \epsilon + nT'E'$
 $\xrightarrow{10}$

Emparejamiento de n 

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

29 / 40)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada							
	id	=	+	*	()	número	\$	
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F	9				8		10	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

30 / 40

Pila	Entrada	Acción
...
\$ <u>$E' T +$</u>	$+ n * id \$$	<i>Emparejar</i>
\$ <u>$E' T$</u>	$n * id \$$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	$n * id \$$	10) $F \rightarrow \text{número}$
\$ <u>$E' T' n$</u>	$n * id \$$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	$* id \$$	6) $T' \rightarrow * F T'$
\$ <u>$E' T' F *$</u>	$* id \$$	<i>Emparejar</i>
\$ <u>$E' T' F$</u>	$id \$$	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

30 / 40

Pila	Entrada	Acción
...
\$ <u>$E' T +$</u>	$+ n * id \$$	<i>Emparejar</i>
\$ <u>$E' T$</u>	$n * id \$$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	$n * id \$$	10) $F \rightarrow \text{número}$
\$ <u>$E' T' n$</u>	$n * id \$$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	$* id \$$	6) $T' \rightarrow * F T'$
\$ <u>$E' T' F *$</u>	$* id \$$	<i>Emparejar</i>
\$ <u>$E' T' F$</u>	$id \$$	

Análisis sintáctico descendente predictivo

Implementación iterativa

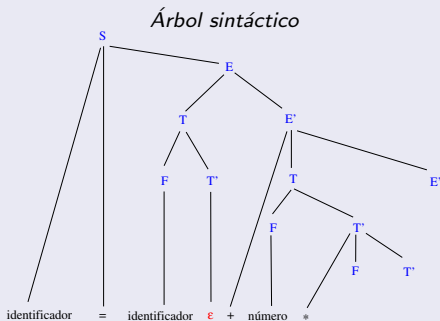
Ejemplo (Gramática de expresiones aritméticas)

31 / 40

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + nT'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6}$

Emparejamiento de *



Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

32 / 40)

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2		2
E'			3			4	4
T	5				5		5
T'			7	6		7	7
F	9				8		10

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

33 / 40

Pila	Entrada	Acción
...
\$ <u>$E' T +$</u>	$+ n * id \$$	<i>Emparejar</i>
\$ <u>$E' T$</u>	$n * id \$$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	$n * id \$$	10) $F \rightarrow \text{número}$
\$ <u>$E' T' n$</u>	$n * id \$$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	$* id \$$	6) $T' \rightarrow * F T'$
\$ <u>$E' T' F *$</u>	$* id \$$	<i>Emparejar</i>
\$ <u>$E' T' F$</u>	$id \$$	9) $F \rightarrow \text{identificador}$
\$ <u>$E' T' id$</u>	$id \$$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	$\$$	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

33 / 40

Pila	Entrada	Acción
...
\$ <u>$E' T +$</u>	$+ n * id \$$	<i>Emparejar</i>
\$ <u>$E' T$</u>	$n * id \$$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	$n * id \$$	10) $F \rightarrow \text{número}$
\$ <u>$E' T' n$</u>	$n * id \$$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	$* id \$$	6) $T' \rightarrow * F T'$
\$ <u>$E' T' F *$</u>	$* id \$$	<i>Emparejar</i>
\$ <u>$E' T' F$</u>	$id \$$	9) $F \rightarrow \text{identificador}$
\$ <u>$E' T' id$</u>	$id \$$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	$\$$	

Análisis sintáctico descendente predictivo

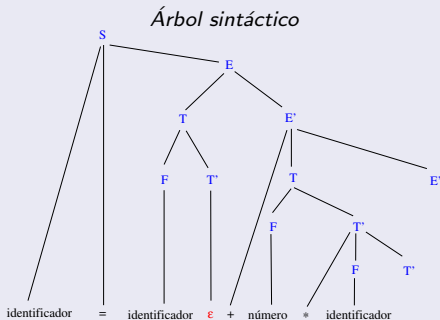
Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

34 / 40

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * \underline{id}T'E'$
 $\xrightarrow{9}$

Emparejamiento de id 

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

35 / 40

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3		4		4
T	5				5	5	
T'			7	6	7		7
F	9				8	10	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

36 / 40

Pila	Entrada	Acción
...
\$ <u>$E' T +$</u>	+ n * id \$	<i>Emparejar</i>
\$ <u>$E' T$</u>	n * id \$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	n * id \$	10) $F \rightarrow \text{número}$
\$ <u>$E' T' n$</u>	n * id \$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	* id \$	6) $T' \rightarrow * F T'$
\$ <u>$E' T' F *$</u>	* id \$	<i>Emparejar</i>
\$ <u>$E' T' F$</u>	id \$	9) $F \rightarrow \text{identificador}$
\$ <u>$E' T' id$</u>	id \$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	\$	7) $T' \rightarrow \epsilon$
\$ <u>E'</u>	\$	

Análisis sintáctico descendente predictivo

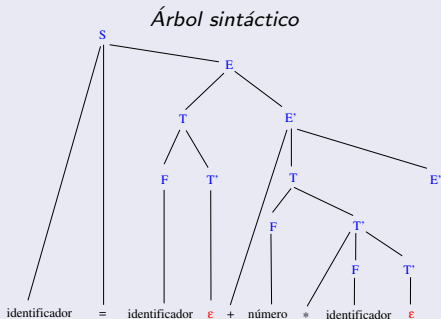
Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

37 / 40

Análisis de $id = id + n * id \$$

$S \Rightarrow id = E$
 $\xrightarrow{1} id = TE'$
 $\xrightarrow{2} id = FT'E'$
 $\xrightarrow{5} id = idT'E'$
 $\xrightarrow{9} id = id \epsilon E'$
 $\xrightarrow{7} id = id \epsilon + TE'$
 $\xrightarrow{3} id = id \epsilon + FT'E'$
 $\xrightarrow{5} id = id \epsilon + n T'E'$
 $\xrightarrow{10} id = id \epsilon + n * FT'E'$
 $\xrightarrow{6} id = id \epsilon + n * id T'E'$
 $\xrightarrow{9} id = id \epsilon + n * id \epsilon E'$
 $\xrightarrow{7}$



Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

38 / 40

 $P = \{$ (1) $S \rightarrow \text{identificador} = E$ (2) $E \rightarrow T E'$ (3) $E' \rightarrow + T E'$ (4) $E' \rightarrow \epsilon$ (5) $T \rightarrow F T'$ (6) $T' \rightarrow * F T'$ (7) $T' \rightarrow \epsilon$ (8) $F \rightarrow (E)$ (9) $F \rightarrow \text{identificador}$ (10) $F \rightarrow \text{número}$ $\}$

Tabla predictiva

	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						
E	2				2	2	
E'			3		4		4
T	5				5	5	
T'			7	6	7		7
F	9				8	10	

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

39 / 40

Pila	Entrada	Acción
...
\$ <u>$E' T +$</u>	$+ n * id \$$	<i>Emparejar</i>
\$ <u>$E' T$</u>	$n * id \$$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	$n * id \$$	10) $F \rightarrow \text{número}$
\$ <u>$E' T' n$</u>	$n * id \$$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	$* id \$$	6) $T' \rightarrow * F T'$
\$ <u>$E' T' F *$</u>	$* id \$$	<i>Emparejar</i>
\$ <u>$E' T' F$</u>	$id \$$	9) $F \rightarrow \text{identificador}$
\$ <u>$E' T' id$</u>	$id \$$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	$\$$	7) $T' \rightarrow \epsilon$
\$ <u>E'</u>	$\$$	4) $E' \rightarrow \epsilon$
\$	$\$$	<i>Aceptar</i>

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejemplo (Gramática de expresiones aritméticas)

39 / 40

Pila	Entrada	Acción
...
\$ <u>$E' T +$</u>	$+ n * id \$$	<i>Emparejar</i>
\$ <u>$E' T$</u>	$n * id \$$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	$n * id \$$	10) $F \rightarrow \text{número}$
\$ <u>$E' T' n$</u>	$n * id \$$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	$* id \$$	6) $T' \rightarrow * F T'$
\$ <u>$E' T' F *$</u>	$* id \$$	<i>Emparejar</i>
\$ <u>$E' T' F$</u>	$id \$$	9) $F \rightarrow \text{identificador}$
\$ <u>$E' T' id$</u>	$id \$$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	$\$$	7) $T' \rightarrow \epsilon$
\$ <u>E'</u>	$\$$	4) $E' \rightarrow \epsilon$
\$	$\$$	<i>Aceptar</i>

Análisis sintáctico descendente predictivo

Implementación iterativa

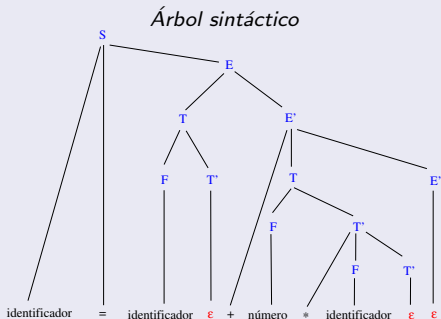
Ejemplo (Gramática de expresiones aritméticas)

40 / 40

Análisis de $id = id + n * id \$$

S

\Rightarrow	1	$id = E$
\Rightarrow	2	$id = TE'$
\Rightarrow	5	$id = FT'E'$
\Rightarrow	9	$id = idT'E'$
\Rightarrow	7	$id = id \in E'$
\Rightarrow	3	$id = id \in + TE'$
\Rightarrow	5	$id = id \in + FT'E'$
\Rightarrow	10	$id = id \in + nT'E'$
\Rightarrow	6	$id = id \in + n * FT'E'$
\Rightarrow	9	$id = id \in + n * idT'E'$
\Rightarrow	7	$id = id \in + n * id \in E'$
\Rightarrow	4	$id = id \in + n * id \in \in$



Análisis sintáctico descendente predictivo

Implementación iterativa

Ejercicios (Método iterativo)

1.- *Gramática de las expresiones.*

id = número * (identificador + identificador) \$

2.- *Gramática de las declaraciones.*

int id, id; float id \$

3.- *Gramática de prototipos*

int id (int id, float id) ; \$

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejercicio (1.- Gramática de expresiones aritméticas)

- *Utiliza el método iterativo para realizar el análisis descendente y predictivo de la sentencia:*

id = número * (identificador + identificador) \$

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejercicio (2.- Gramática de declaraciones)

- *Gramática de declaraciones*

$P = \{$

(1) $S \rightarrow D S$

(2) $S \rightarrow \epsilon$

(3) $D \rightarrow T L ;$

(4) $T \rightarrow \mathbf{int}$

(5) $T \rightarrow \mathbf{float}$

(6) $L \rightarrow \mathbf{id} L'$

(7) $L' \rightarrow , \mathbf{id} L'$

(8) $L' \rightarrow \epsilon$

$\}$

- *Análisis de las declaraciones*

int id, id; float id \$

Análisis sintáctico descendente predictivo

Implementación iterativa

Ejercicio (3.- Gramática de prototipos)

- Gramática de prototipos

$$P = \{$$

(1) $S \rightarrow D S$

(2) $S \rightarrow \epsilon$

(3) $D \rightarrow T \text{ id } (P) ;$

(4) $T \rightarrow \text{int}$

(5) $T \rightarrow \text{float}$

(6) $P \rightarrow \epsilon$

(7) $P \rightarrow T \text{ id } L$

(8) $L \rightarrow , T \text{ id } L$

(9) $L \rightarrow \epsilon$

$$\}$$

- Construcción del conjunto

Primero

- Construcción del conjunto

Siguiente

- Construcción de la **Tabla predictiva**

- Análisis del prototipo

int id (int id, float id) ; \$

Contenido del tema

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores
 - Introducción
 - Detección de errores
 - Recuperación de errores
 - Modo de pánico
 - Método de nivel de frase

Detección y recuperación de errores

Introducción

Objetivos

- **Fase de detección**
 - Debe detectar el **mayor** número de errores posibles.
- **Fase de recuperación**
 - Debe proponer una **solución aceptable** que permita que el análisis **continúe** para detectar más errores.
 - Es **responsabilidad del programador** la elección de la **solución más adecuada**.

Detección y recuperación de errores

Introducción

Ejemplo (Detección y recuperación)

- *Considérese la siguiente sentencia **errónea**:*

id = = id _ n * id

- *Se podría proponer la siguiente solución*

id = id + n * id

- *pero las siguientes soluciones también son correctas*

id = id * n * id

id = n * id

id = id * id

Detección y recuperación de errores

Introducción

Nota (Responsabilidad del programador)

- Al detectar un *error*, el *analizador* solamente *propone* una solución para *continuar* el análisis.
- El *programador* deberá comprobar si la *solución* propuesta es o no la más *adecuada*.

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores
 - Introducción
 - Detección de errores
 - Recuperación de errores
 - Modo de pánico
 - Método de nivel de frase

Detección y recuperación de errores

Detección de errores

Detección de errores

1 / 7

- **Tipos de errores detectados** por el análisis sintáctico descendente predictivo:
 - 1.- El símbolo situado en la cima de la pila es un símbolo terminal σ o \$ que **no coincide** con el símbolo de la entrada.
 - 2.- La celda $M[A,\sigma]$ está **vacía**, donde A es el símbolo situado en la cima de la pila y σ es el símbolo actual de la entrada.

Detección y recuperación de errores

Detección de errores

Detección de errores

2 / 7

- 1.- El símbolo situado en la cima de la pila es un símbolo terminal σ o $\$$ que **no coincide** con el símbolo de la entrada.

a)

Pila	Entrada	Acción
$\$ \dots \sigma$	$\sigma' \dots \sigma_n \$$	Error
...

b)

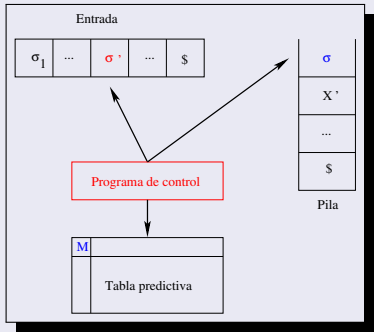
Pila	Entrada	Acción
$\$$	$\sigma' \dots \sigma_n \$$	Error
...

Detección y recuperación de errores

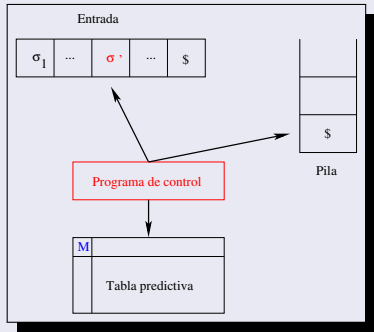
Detección de errores

Detección de errores

3 / 7



(a)



(b)

Detección y recuperación de errores

Detección de errores

Detección de errores

4 / 7

2.- La celda $M[A, \sigma]$ está **vacía**, donde A es el símbolo situado en la cima de la pila y σ es el símbolo actual de la entrada.

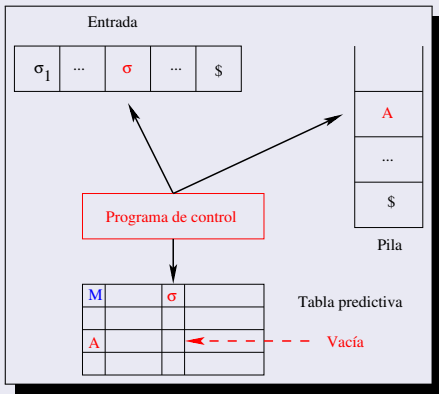
Pila	Entrada	Acción
$\$ \dots A$	$\sigma \dots \sigma_n \$$	Error
\dots	\dots	\dots

Detección y recuperación de errores

Detección de errores

Detección de errores

5 / 7



Detección y recuperación de errores

Detección de errores

Nota (Detección de errores)

6 / 7

- *Se ha realizado la descripción de la detección de errores que se produce si se utiliza el método **predictivo e iterativo**.*
- *La descripción para el método **predictivo y recursivo** sería **similar**.*

Detección y recuperación de errores

Detección de errores

Nota (Detección de errores)

7 / 7

- El método de *descenso recursivo con retroceso o backtracking* **no permite** detectar la **ubicación** del error, porque siempre termina en la función asociada al símbolo inicial de la gramática.

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores
 - Introducción
 - Detección de errores
 - **Recuperación de errores**
 - Modo de pánico
 - Método de nivel de frase

Detección y recuperación de errores

Recuperación de errores

Métodos de recuperación de errores

1 / 5

- 1.- Modo de pánico.
- 2.- Método de nivel de frase.
- 3.- Regla de producción de control de errores.
- 4.- Corrección global.

Detección y recuperación de errores

Recuperación de errores

Métodos de recuperación de errores

2 / 5

1.- Modo de pánico

- Es el método más fácil de aplicar.
- Busca símbolos de sincronización para **continuar** el análisis.

Detección y recuperación de errores

Recuperación de errores

Métodos de recuperación de errores

3 / 5

2.- Método de nivel de frase

- Realiza **transformaciones** en la cadena de entrada o en la pila para continuar el análisis.

Detección y recuperación de errores

Recuperación de errores

Métodos de recuperación de errores

4 / 5

3.- Regla de producción de control de errores

- Se **amplia** la gramática con **nuevas reglas de producción** que permiten generar los errores más frecuentes.
- Si se **utilizan** dichas reglas de producción entonces el analizador activa un método de recuperación de errores.

Este método será descrito en las clases de prácticas.

Detección y recuperación de errores

Recuperación de errores

Métodos de recuperación de errores

4 / 5

3.- Regla de producción de control de errores

- Se **amplia** la gramática con **nuevas reglas de producción** que permiten generar los errores más frecuentes.
- Si se **utilizan** dichas reglas de producción entonces el analizador activa un método de recuperación de errores.

Nota

Este método será descrito en las clases de prácticas.

Detección y recuperación de errores

Recuperación de errores

Métodos de recuperación de errores

5 / 5

4.- Corrección global

- Método **teórico** basado en los anteriores, pero, especialmente, en el método de nivel de frase.
- Trata de obtener un programa **correcto** (sin errores) realizando el **menor número de transformaciones** de la cadena de entrada.
- Las transformaciones no están basadas solamente en el entorno local, sino en el programa completo.

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores
 - Introducción
 - Detección de errores
 - Recuperación de errores
 - **Modo de pánico**
 - Método de nivel de frase

Detección y recuperación de errores

Modo de pánico

Estrategia del método de modo de pánico

Al detectar un error,

- comienza a **eliminar** símbolos de la cadena de entrada
- hasta que encuentre un **componente léxico** o **token**
- que pertenezca a un **conjunto de sincronización**
- que permita **continuar el análisis**.

Detección y recuperación de errores

Modo de pánico

Métodos de construcción del conjunto de sincronización

- 1.- Método basado en el conjunto **Siguiente**.
- 2.- Método que incorpora **símbolos** que indican el comienzo de una sentencia de **mayor jerarquía**.
- 3.- Método basado en el conjunto **Primero**.
- 4.- Método basado en las reglas de producción ϵ .
- 5.- Caso especial para los **símbolos terminales**.
- 6.- Caso especial para el símbolo **\$**.

Detección y recuperación de errores

Modo de pánico

Métodos de construcción del conjunto de sincronización

1.- Método basado en el conjunto **Siguiente**.

Si $M[A, \sigma] = \text{vacía}$ entonces

- se **eliminan** símbolos de la entrada hasta que se **encuentra** un símbolo $\sigma' \in \text{Siguiente}(A)$
- y se **extrae** el símbolo **A** de la pila y el análisis **continúa**.

Pila	Entrada	Acción
\$... X' A	$\sigma \dots \sigma' \dots \sigma_n$ \$	Error: extraer σ de la entrada
...
\$... X' A	$\sigma' \dots \sigma_n$ \$	Sincronización: extraer A de la pila
\$... X'	$\sigma' \dots \sigma_n$ \$	Continúa el análisis

Detección y recuperación de errores

Modo de pánico

Ejemplo (1.- Método basado en el conjunto **Siguiente** 1 / 2)

$P = \{$

- (1) $S \rightarrow \text{identificador} = E$
 - (2) $E \rightarrow T E'$
 - (3) $E' \rightarrow + T E'$
 - (4) $E' \rightarrow \epsilon$
 - (5) $T \rightarrow F T'$
 - (6) $T' \rightarrow * F T'$
 - (7) $T' \rightarrow \epsilon$
 - (8) $F \rightarrow (E)$
 - (9) $F \rightarrow \text{identificador}$
 - (10) $F \rightarrow \text{número}$
- $\}$

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$, ")"
E'	+, ϵ	\$, ")"
T	"(", id, n	+, \$, ")"
T'	*, ϵ	+, \$, ")"
F	"(", id, n	*, +, \$, ")"

Detección y recuperación de errores

Modo de pánico

Ejemplo (1.- Método basado en el conjunto **Siguiente** 2 / 2)

Tabla predictiva							
	Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1						<i>Sincr.</i>
E	2				2	<i>Sincr.</i>	2
E'			3		4		4
T	5		<i>Sincr.</i>		5	<i>Sincr.</i>	5
T'			7	6	7		7
F	9		<i>Sincr.</i>	<i>Sincr.</i>	8	<i>Sincr.</i>	10

Detección y recuperación de errores

Modo de pánico

Métodos de construcción del conjunto de sincronización

2.- Método que incorpora **símbolos** que indican el comienzo de una sentencia de **mayor jerarquía**.

Si $M[A, \sigma] = \text{vacía}$ entonces

- se **eliminan** símbolos de la entrada hasta que se **encuentra** un símbolo $\sigma' \in \text{Siguiente}(A)$ o que indica el comienzo de una sentencia de **mayor jerarquía**.
- y se **extrae** el símbolo A de la pila y el análisis **continúa**.

Pila	Entrada	Acción
\$... X' A	$\sigma \dots \sigma' \dots \sigma_n$ \$	Error: extraer σ de la entrada
...
\$... X' A	$\sigma' \dots \sigma_n$ \$	Sincronización: extraer A de la pila
\$... X'	$\sigma' \dots \sigma_n$ \$	Continúa el análisis

Detección y recuperación de errores

Modo de pánico

Ejemplo (2.- Símbolos que indican el comienzo de una sentencia de mayor jerarquía)

- *Considérese el siguiente código*

```
dato = __ * 3 ;
```

```
...
```

```
if ...
```

*en el que **falta** el primer factor del producto.*

*El análisis podría **continuar** a partir de la palabra clave **if** que indica el comienzo de una **sentencia condicional**.*

Detección y recuperación de errores

Modo de pánico

Métodos de construcción del conjunto de sincronización

3.- Método basado en el conjunto **Primero**.

Si $M[A, \sigma] = \text{vacía}$ entonces

- se **eliminan** símbolos de la entrada hasta que se **encuentra** un símbolo $\sigma' \in \text{Primero}(A)$
- y el análisis **continúa** **sin** extraer el símbolo A de la pila.

Pila	Entrada	Acción
\$... X' A	$\sigma \dots \sigma' \dots \sigma_n \$$	Error: extraer σ de la entrada
...
\$... X' A	$\sigma' \dots \sigma_n \$$	Sincronización: continúa el análisis

Detección y recuperación de errores

Modo de pánico

Ejemplo (1.- Método basado en el conjunto **Primero**)

$P = \{$

- (1) $S \rightarrow \text{identificador} = E$
 - (2) $E \rightarrow T E'$
 - (3) $E' \rightarrow + T E'$
 - (4) $E' \rightarrow \epsilon$
 - (5) $T \rightarrow F T'$
 - (6) $T' \rightarrow * F T'$
 - (7) $T' \rightarrow \epsilon$
 - (8) $F \rightarrow (E)$
 - (9) $F \rightarrow \text{identificador}$
 - (10) $F \rightarrow \text{número}$
- $\}$

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$, ")"
E'	+, ϵ	\$, ")"
T	"(", id, n	+, \$, ")"
T'	*, ϵ	+, \$, ")"
F	"(", id, n	*, +, \$, ")"

Detección y recuperación de errores

Modo de pánico

Métodos de construcción del conjunto de sincronización

4.- Método basado en las reglas de producción ϵ .

- a) Si $\epsilon \in \text{Primero}(A)$ entonces se completan todas las celdas vacías de A en la tabla predictiva con la regla $A \rightarrow \epsilon$, **aunque** dicha regla no pertenezca al conjunto inicial de producciones de la gramática.
- b) Si $M[A, \sigma] = \text{vacía}$ entonces se **extrae** el símbolo A de la pila y el análisis **continúa**.

Pila	Entrada	Acción
$\$ \dots X' A$	$\sigma \dots \sigma_n \$$	Extraer A de la pila
$\$ \dots X'$	$\sigma \dots \sigma_n \$$	Continúa el análisis

Detección y recuperación de errores

Modo de pánico

Nota (4.- Método basado en las reglas de producción ◦)

- Se **reduce** el número de errores que se deben controlar.
- Este método **posterga o retrasa** la detección del error para que sea tratado por *otro símbolo* al que se le haya asignado el conjunto de sincronización.

Detección y recuperación de errores

Modo de pánico

Ejemplo (4.- Método basado en las reglas de producción ϵ 1 / 2)

$P = \{$

- (1) $S \rightarrow \text{identificador} = E$
 - (2) $E \rightarrow T E'$
 - (3) $E' \rightarrow + T E'$
 - (4) $E' \rightarrow \epsilon$
 - (5) $T \rightarrow F T'$
 - (6) $T' \rightarrow * F T'$
 - (7) $T' \rightarrow \epsilon$
 - (8) $F \rightarrow (E)$
 - (9) $F \rightarrow \text{identificador}$
 - (10) $F \rightarrow \text{número}$
- $\}$

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$, ")"
E'	+, ϵ	\$, ")"
T	"(", id, n	+, \$, ")"
T'	*, ϵ	+, \$, ")"
F	"(", id, n	*, +, \$, ")"

Detección y recuperación de errores

Modo de pánico

Ejemplo (4.- Método basado en las reglas de producción ϵ 2 / 2)

$P = \{$

- (1) $S \rightarrow \text{identificador} = E$
 - (2) $E \rightarrow T E'$
 - (3) $E' \rightarrow + T E'$
 - (4) $E' \rightarrow \epsilon$
 - (5) $T \rightarrow F T'$
 - (6) $T' \rightarrow * F T'$
 - (7) $T' \rightarrow \epsilon$
 - (8) $F \rightarrow (E)$
 - (9) $F \rightarrow \text{identificador}$
 - (10) $F \rightarrow \text{número}$
- $\}$

Tabla predictiva

	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'	4*	4*	3	4*	4*	4	4*	4
T	5				5		5	
T'	7*	7*	7	6	7*	7	7*	7
F	9				8		10	

Detección y recuperación de errores

Modo de pánico

Métodos de construcción del conjunto de sincronización

5.- Caso especial para los símbolos terminales.

- Si σ está situado en la cima de la pila, σ' es el símbolo actual de la entrada y $\sigma \neq \sigma'$ entonces
 - a) Se indica que falta el símbolo σ en la entrada
 - b) Se elimina el símbolo σ de la **pila** y el análisis **continúa**.

Pila	Entrada	Acción
$\$ \dots X' \sigma$	$\sigma' \dots \sigma_n \$$	Falta σ en la entrada. Extraer σ de la pila
$\$ \dots X'$	$\sigma' \dots \sigma_n \$$	Continúa el análisis

Detección y recuperación de errores

Modo de pánico

Métodos de construcción del conjunto de sincronización

5.- Caso especial para los símbolos terminales.

- Si σ está situado en la cima de la pila, σ' es el símbolo actual de la entrada y $\sigma \neq \sigma'$ entonces
 - a) Se indica que falta el símbolo σ en la entrada
 - b) Se elimina el símbolo σ' en la entrada y el análisis **continúa**.

Nota

- *El conjunto de sincronización de un símbolo terminal σ está compuesto por todos los demás símbolos terminales y el símbolo \$:*

$$\text{sincronización}(\sigma) = (V_T \cup \{\$\}) - \{\sigma\}$$

Detección y recuperación de errores

Modo de pánico

Métodos de construcción del conjunto de sincronización

6.- Caso especial para el símbolo \$.

- Si \$ está situado en la cima de la pila, σ es el símbolo actual de la entrada y $\$ \neq \sigma$ entonces
 - a) Se indica que σ es un símbolo **inesperado**,
 - b) se **elimina** de la entrada y el análisis **continúa**.

Pila	Entrada	Acción
\$	$\sigma \sigma' \dots \sigma_n \$$	Símbolo inesperado Extraer σ de la entrada
\$	$\sigma' \dots \sigma_n \$$	Continúa el análisis

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

1 / 5

$P = \{$

- (1) $S \rightarrow \text{identificador} = E$
 - (2) $E \rightarrow T E'$
 - (3) $E' \rightarrow + T E'$
 - (4) $E' \rightarrow \epsilon$
 - (5) $T \rightarrow F T'$
 - (6) $T' \rightarrow * F T'$
 - (7) $T' \rightarrow \epsilon$
 - (8) $F \rightarrow (E)$
 - (9) $F \rightarrow \text{identificador}$
 - (10) $F \rightarrow \text{número}$
- $\}$

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$, ")"
E'	+, ϵ	\$, ")"
T	"(", id, n	+, \$, ")"
T'	*, ϵ	+, \$, ")"
F	"(", id, n	*, +, \$, ")"

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

2 / 5)

Tabla predictiva								
Símbolo de entrada								
	id	=	+	*	()	número	\$
S	1							<i>Sincr.</i>
E	2				2	<i>Sincr.</i>	2	<i>Sincr.</i>
E'	4*	4*	3	4*	4*	4	4*	4
T	5		<i>Sincr.</i>		5	<i>Sincr.</i>	5	<i>Sincr.</i>
T'	7*	7*	7	6	7*	7	7*	7
F	9		<i>Sincr.</i>	<i>Sincr.</i>	8	<i>Sincr.</i>	10	<i>Sincr.</i>

Detección y recuperación de errores

Modo de pánico

Nota (Aplicación del modo de pánico)

3 / 5

- *Reglas de producción ϵ*
 - Se han usado para **completar** las celdas *vacías* de los símbolos que contienen a ϵ en su conjunto **Primero**.
- *Sincronización (Sincr.)*
 - Se ha usado el conjunto *Siguiente* como conjunto de **sincronización**.
 - Al alcanzar un *símbolo de sincronización* en la **entrada**, se *eliminará* el símbolo no terminal de la **pila** y el *análisis continuará*.

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

4 / 5

Pila	Entrada	Acción
\$ S	id == id n * id \$	1) S → identificador = E
\$ E = id	id == id n * id \$	<i>Emparejar</i>
\$ E =	= id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>Error: extraer = de la entrada</i>
\$ E	id n * id \$	<i>Sincronización</i>
\$ E' T	id n * id \$	2) E → T E'
\$ E' T' F	id n * id \$	5) T → F T'
\$ E' T' id	id n * id \$	9) F → identificador
\$ E' T'	n * id \$	<i>Emparejar</i>
\$ E'	n * id \$	7*) T → ε
\$	n * id \$	3*) E → ε
		<i>Símbolo inesperado</i>
		<i>Extraer n de la entrada</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

4 / 5

Pila	Entrada	Acción
\$ S	id == id n * id \$	1) S → identificador = E
\$ <u>E = id</u>	id == id n * id \$	<i>Emparejar</i>
\$ E =	== id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>Error: extraer = de la entrada</i>
\$ E	id n * id \$	<i>Sincronización</i>
\$ E T	id n * id \$	2) E → T E'
\$ E' T E	id n * id \$	5) T → F T'
\$ E' T' id	id n * id \$	9) F → identificador
\$ E' T'	n * id \$	<i>Emparejar</i>
\$ E'	n * id \$	7*) T → ε
\$	n * id \$	3*) E → ε
		<i>Símbolo inesperado</i>
		<i>Extraer n de la entrada</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

4 / 5

Pila	Entrada	Acción
\$ S	id == id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id == id n * id \$	<i>Emparejar</i>
\$ E =	= id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>Error: extraer = de la entrada</i>
\$ E	id n * id \$	<i>Sincronización</i>
\$ E T	id n * id \$	2) $E \rightarrow T E'$
\$ E T E	id n * id \$	5) $T \rightarrow F T'$
\$ E T id	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E T	n * id \$	<i>Emparejar</i>
\$ E	n * id \$	7*) $T \rightarrow \epsilon$
\$	n * id \$	3*) $E \rightarrow \epsilon$
		<i>Símbolo inesperado</i>
		<i>Extraer n de la entrada</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

4 / 5

Pila	Entrada	Acción
\$ S	id == id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id == id n * id \$	<i>Emparejar</i>
\$ E =	= id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>Error: extraer = de la entrada</i>
\$ E	id n * id \$	<i>Sincronización</i>
\$ E T	id n * id \$	2) $E \rightarrow T E'$
\$ E' T E	id n * id \$	5) $T \rightarrow F T'$
\$ E' T' id	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' T'	n * id \$	<i>Emparejar</i>
\$ E'	n * id \$	7*) $T' \rightarrow \epsilon$
\$	n * id \$	3*) $E' \rightarrow \epsilon$
\$	n * id \$	<i>Símbolo inesperado</i>
\$	n * id \$	<i>Extraer n de la entrada</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

4 / 5

Pila	Entrada	Acción
\$ S	id == id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id == id n * id \$	<i>Emparejar</i>
\$ E =	= id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>Error: extraer = de la entrada</i>
\$ E	id n * id \$	<i>Sincronización</i>
\$ E T	id n * id \$	2) $E \rightarrow T E'$
\$ E' T E	id n * id \$	5) $T \rightarrow F T'$
\$ E' T id	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' T	n * id \$	<i>Emparejar</i>
\$ E'	n * id \$	7*) $T \rightarrow \epsilon$
\$	n * id \$	3*) $E \rightarrow \epsilon$
\$	n * id \$	<i>Símbolo inesperado</i>
\$	n * id \$	<i>Extraer n de la entrada</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

4 / 5

Pila	Entrada	Acción
\$ S	id == id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id == id n * id \$	<i>Emparejar</i>
\$ E =	= id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>Error: extraer = de la entrada</i>
\$ E	id n * id \$	<i>Sincronización</i>
\$ <u>E' T</u>	id n * id \$	2) $E \rightarrow T E'$
\$ E' T E	id n * id \$	5) $T \rightarrow F T'$
\$ E' T id	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' T	id n * id \$	<i>Emparejar</i>
\$ E' T	n * id \$	7*) $T \rightarrow \epsilon$
\$ E	n * id \$	3*) $E \rightarrow \epsilon$
\$	n * id \$	<i>Símbolo inesperado</i>
		<i>Extraer n de la entrada</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

4 / 5

Pila	Entrada	Acción
\$ S	id == id n * id \$	1) S → identificador = E
\$ <u>E = id</u>	id == id n * id \$	<i>Emparejar</i>
\$ E =	= id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>Error: extraer = de la entrada</i>
\$ E	id n * id \$	<i>Sincronización</i>
		2) E → T E'
\$ <u>E' T</u>	id n * id \$	5) T → F T'
\$ E' <u>T' F</u>	id n * id \$	9) F → identificador
\$ E' T' id	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	<i>7) T → ε</i>
\$ E	n * id \$	<i>3) E → ε</i>
\$	n * id \$	<i>Símbolo inesperado</i>
		<i>Extraer n de la entrada</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

4 / 5

Pila	Entrada	Acción
\$ S	id == id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id == id n * id \$	<i>Emparejar</i>
\$ E =	= id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>Error: extraer = de la entrada</i>
\$ E	id n * id \$	<i>Sincronización</i>
\$ E' T	id n * id \$	2) $E \rightarrow T E'$
\$ E' T' F	id n * id \$	5) $T \rightarrow F T'$
\$ E' T' id	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' T	n * id \$	<i>Emparejar</i>
\$ E	n * id \$	7) $T \rightarrow \epsilon$
\$	n * id \$	3) $E \rightarrow \epsilon$
\$	n * id \$	<i>Símbolo inesperado</i>
\$	n * id \$	<i>Extraer n de la entrada</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

4 / 5

Pila	Entrada	Acción
\$ <u>S</u>	id == id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id == id n * id \$	<i>Emparejar</i>
\$ <u>E =</u>	= id n * id \$	<i>Emparejar</i>
\$ <u>E</u>	= id n * id \$	<i>Error: extraer = de la entrada</i>
\$ <u>E</u>	id n * id \$	<i>Sincronización</i>
\$ <u>E' T</u>	id n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T' F</u>	id n * id \$	5) $T \rightarrow F T'$
\$ <u>E' T' id</u>	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ <u>E' T'</u>	n * id \$	<i>Emparejar</i>
\$ <u>E</u>	n * id \$	7*) $T \rightarrow \epsilon$
\$	n * id \$	3*) $E \rightarrow \epsilon$
\$	n * id \$	<i>Símbolo inesperado</i>
		<i>Extraer n de la entrada</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

4 / 5

Pila	Entrada	Acción
\$ S	id == id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id == id n * id \$	<i>Emparejar</i>
\$ E =	= id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>Error: extraer = de la entrada</i>
\$ E	id n * id \$	<i>Sincronización</i>
\$ E' T	id n * id \$	2) $E \rightarrow T E'$
\$ E' T' F	id n * id \$	5) $T \rightarrow F T'$
\$ E' T' id	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' T'	n * id \$	<i>Emparejar</i>
\$ E'	n * id \$	7*) $T \rightarrow \epsilon$
\$	n * id \$	3*) $E \rightarrow \epsilon$
\$	n * id \$	<i>Símbolo inesperado</i>
\$	n * id \$	<i>Extraer n de la entrada</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

4 / 5

Pila	Entrada	Acción
\$ S	id == id n * id \$	1) $S \rightarrow$ identificador = E
\$ $E = id$	id == id n * id \$	<i>Emparejar</i>
\$ $E =$	= id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>Error: extraer = de la entrada</i>
\$ E	id n * id \$	<i>Sincronización</i>
\$ $E' T$	id n * id \$	2) $E \rightarrow T E'$
\$ $E' T' F$	id n * id \$	5) $T \rightarrow F T'$
\$ $E' T' id$	id n * id \$	9) $F \rightarrow$ identificador
\$ $E' T'$	n * id \$	<i>Emparejar</i>
\$ E'	n * id \$	7*) $T \rightarrow \epsilon$
\$	n * id \$	3*) $E \rightarrow \epsilon$
		<i>Símbolo inesperado</i>
		<i>Extraer n de la entrada</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

5 / 5

Pila	Entrada	Acción
...
\$	n * id \$	<i>Símbolo inesperado</i> <i>Extraer n de la entrada</i>
\$	* id \$	<i>Símbolo inesperado</i> <i>Extraer * de la entrada</i>
\$	id \$	<i>Símbolo inesperado</i> <i>Extraer id de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

5 / 5)

Pila	Entrada	Acción
...
\$	n * id \$	<i>Símbolo inesperado</i> <i>Extraer n de la entrada</i>
\$	* id \$	<i>Símbolo inesperado</i> <i>Extraer * de la entrada</i>
\$	id \$	<i>Símbolo inesperado</i> <i>Extraer id de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

5 / 5

Pila	Entrada	Acción
...
\$	n * id \$	<i>Símbolo inesperado</i> <i>Extraer n de la entrada</i>
\$	* id \$	<i>Símbolo inesperado</i> <i>Extraer * de la entrada</i>
\$	id \$	<i>Símbolo inesperado</i> <i>Extraer id de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Modo de pánico

Ejemplo (Aplicación del modo de pánico)

5 / 5)

Pila	Entrada	Acción
...
\$	n * id \$	Símbolo <i>inesperado</i> <i>Extraer n</i> de la entrada
\$	* id \$	Símbolo <i>inesperado</i> <i>Extraer *</i> de la entrada
\$	id \$	Símbolo <i>inesperado</i> <i>Extraer id</i> de la entrada
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Modo de pánico

Ventajas e inconveniente

- **Ventajas:**
 - Es fácil de aplicar.
 - Evita caer en bucles infinitos, ya que solamente **elimina** símbolos de la entrada o la pila.
- **Inconveniente:**
 - No es capaz de detectar todos los errores posibles.

Detección y recuperación de errores

Modo de pánico

Ejemplo (No detección de todos los errores)

- Al analizar la sentencia *errónea*:

id = = id _ n * id

el método de modo de pánico ha propuesto la solución

id = id

pero las siguientes soluciones parecen más adecuadas

id = id + n * id

id = id * n * id

id = n * id

id = id * id

Detección y recuperación de errores

Modo de pánico

Ejercicios (Aplicación del modo de pánico)

- 1.- Gramática de las *expresiones aritméticas*.
- 2.- Gramática de las *declaraciones*.
- 3.- Gramática de los *prototipos*.

Detección y recuperación de errores

Modo de pánico

Ejercicio (1.- Gramática de las expresiones aritméticas)

- *Utiliza el modo de pánico para analizar la siguiente sentencia **errónea***

id = n (id id

Detección y recuperación de errores

Modo de pánico

Ejercicio (2.- Gramática de las declaraciones

1 / 3)

- *Utiliza el modo de pánico para analizar la siguiente sentencia errónea*

```
int int id id , id ;
```

Detección y recuperación de errores

Modo de pánico

Ejercicio (2.- Gramática de las declaraciones

2 / 3)

$P = \{$

(1) $S \rightarrow D S$

(2) $S \rightarrow \epsilon$

(3) $D \rightarrow T L ;$

(4) $T \rightarrow \mathbf{int}$

(5) $T \rightarrow \mathbf{float}$

(6) $L \rightarrow \mathbf{id} L'$

(7) $L' \rightarrow , \mathbf{id} L'$

(8) $L' \rightarrow \epsilon$

$\}$

	Primero	Siguiente
S	$\epsilon, \mathbf{int}, \mathbf{float}$	$\$$
D	$\mathbf{int}, \mathbf{float}$	$\mathbf{int}, \mathbf{float}, \$$
T	$\mathbf{int}, \mathbf{float}$	\mathbf{id}
L	\mathbf{id}	$;$
L'	$“,” , \epsilon$	$;$

Detección y recuperación de errores

Modo de pánico

Ejercicio (2.- Gramática de las declaraciones

3 / 3)

 $P = \{$ (1) $S \rightarrow D S$ (2) $S \rightarrow \epsilon$ (3) $D \rightarrow T L ;$ (4) $T \rightarrow \text{int}$ (5) $T \rightarrow \text{float}$ (6) $L \rightarrow \text{id } L'$ (7) $L' \rightarrow , \text{id } L'$ (8) $L' \rightarrow \epsilon$ $\}$

Tabla predictiva

M	Símbolo de entrada					
	;	int	float	id	,	\$
S		1	1			2
D		3	3			
T		4	5			
L				6		
L'	8				7	

Detección y recuperación de errores

Modo de pánico

Ejercicio (3.- Gramática de los prototipos)

$$P = \{$$

(1) $S \rightarrow D S$
 (2) $S \rightarrow \epsilon$
 (3) $D \rightarrow T \text{ id } (P) ;$
 (4) $T \rightarrow \text{int}$
 (5) $T \rightarrow \text{float}$
 (6) $P \rightarrow \epsilon$
 (6) $P \rightarrow T \text{ id } L$
 (7) $L \rightarrow , T \text{ id } L$
 (8) $L \rightarrow \epsilon$

$$\}$$

1.- *Construcción del conjunto*

Primero

2.- *Construcción del conjunto*

Siguiente

3.- *Construcción de la **Tabla** predictiva*

4.- *Análisis del prototipo*

erróneo usando el *modo de pánico*

**int id int id, int float id) ;
\$**

Contenido de la sección

- 1 Introducción
- 2 Descenso recursivo con retroceso o *backtracking*
- 3 Análisis sintáctico descendente predictivo
- 4 Detección y recuperación de errores
 - Introducción
 - Detección de errores
 - Recuperación de errores
 - Modo de pánico
 - Método de nivel de frase

Detección y recuperación de errores

Método de nivel de frase

Descripción

1 / 6

- 1.- La tabla predictiva se **amplía** con una **parte inferior**.
- 2.- Se completan con **acciones** las **celdas** situadas en la **diagonal principal** de la **parte inferior**.
- 3.- Se completan las celdas **vacías** con **funciones de error**
 - **Todas** las celdas **vacías** de la **parte superior**.
 - **Algunas** de las celdas **vacías** de la **parte inferior**.

Detección y recuperación de errores

Método de nivel de frase

Descripción

2 / 6

1.- La tabla predictiva se **amplía** con una **parte inferior**.

- Primer elemento de cada fila: **símbolo terminal** o \$
- Representa la situación en la que el símbolo **actual de la entrada coincide** con el símbolo situado en la **cima de la pila**

Detección y recuperación de errores

Método de nivel de frase

Descripción

3 / 6

M	Símbolo de entrada					
	σ_1	...	σ_j	...	σ_m	\$
S						
A_1						
...						
A_i						
...						
A_n						
σ_1						
...						
σ_j						
...						
σ_m						
\$						

Detección y recuperación de errores

Método de nivel de frase

Descripción: acciones de la parte inferior

4 / 6

2.- Se completan con **acciones** las **celdas** situadas en la **diagonal** principal de la **parte inferior**.

- **Emparejar**
 - El **símbolo terminal actual de la entrada** se corresponde con el símbolo terminal que está en la **cima de la pila**.
- **Aceptar**
 - El símbolo **\$** es el símbolo **actual de la entrada** y el símbolo que aparece en la **cima de la pila**.

Detección y recuperación de errores

Método de nivel de frase

Descripción

3 / 6

M	Símbolo de entrada					
	σ_1	...	σ_j	...	σ_m	\$
S						
A_1						
...						
A_i						
...						
A_n						
σ_1	Emparejar					
...		...				
σ_j			Emparejar			
...				...		
σ_m					Emparejar	
\$						Aceptar

Detección y recuperación de errores

Método de nivel de frase

Descripción

5 / 6

3.- Se completan las celdas **vacías** con funciones de error

- **Celdas vacías** que se han de completar
 - **Todas** las celdas **vacías** de la **parte superior**.
 - **Algunas** de las celdas **vacías** de la **parte inferior**.
- **Método local:**
 - Se debe usar una función **específica** para cada caso.
- **Tipos de funciones:**
 - **Eliminar** un componente léxico de la **entrada**
 - **Insertar** un componente léxico en la **entrada**
 - **Sustituir** un componente léxico de la **entrada**
 - **Eliminar** el símbolo de la **cima** de la **pila**

Detección y recuperación de errores

Método de nivel de frase

Nota (Eliminar un símbolo de la cima de la pila)

- *Esta acción se debe aplicar con **cuidado**.*
- *Puede provocar que la cadena reconocida **no se corresponda** con una cadena que pueda ser **generada** por la gramática.*

Detección y recuperación de errores

Método de nivel de frase

Descripción

6 / 6

3.- Se completan las celdas **vacías** con **funciones de error**

- **Celdas de la parte inferior que se han de completar**

- Celdas **vacías** de las **filas** de los **símbolos terminales** que aparecen en **alguna** regla de producción en un lugar que **no es el primero**.
- En caso contrario, no hay que completar sus celdas vacías con funciones de error: estas celdas **nunca** se consultarán.

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Método de nivel de frase)

1 / 16)

$P = \{$

- (1) $S \rightarrow \text{identificador} = E$
 - (2) $E \rightarrow T E'$
 - (3) $E' \rightarrow + T E'$
 - (4) $E' \rightarrow \epsilon$
 - (5) $T \rightarrow F T'$
 - (6) $T' \rightarrow * F T'$
 - (7) $T' \rightarrow \epsilon$
 - (8) $F \rightarrow (E)$
 - (9) $F \rightarrow \text{identificador}$
 - (10) $F \rightarrow \text{número}$
- $\}$

	Primero	Siguiente
S	id	\$
E	"(", id, n	\$, ")"
E'	+, ϵ	\$, ")"
T	"(", id, n	+, \$, ")"
T'	*, ϵ	+, \$, ")"
F	"(", id, n	*, +, \$, ")"

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Método de nivel de frase: tabla predictiva 2 / 16)

Tabla predictiva								
Símbolo de entrada								
	id	=	+	*	()	número	\$
<i>S</i>	1							
<i>E</i>	2				2		2	
<i>E'</i>			3			4		4
<i>T</i>	5				5		5	
<i>T'</i>			7	6		7		7
<i>F</i>	9				8		10	

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Método de nivel de frase: parte inferior

3 / 16)

Tabla predictiva

	Símbolo de entrada							
	id	=	+	*	()	número	\$
<i>S</i>	1							
<i>E</i>	2				2		2	
<i>E'</i>			3			4		4
<i>T</i>	5				5		5	
<i>T'</i>			7	6		7		7
<i>F</i>	9				8		10	
id								
=								
+								
*								
(
)								
número								
\$								

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Método de nivel de frase: diagonal inferior 4 / 16)

Tabla predictiva								
Símbolo de entrada								
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'			3			4		4
T	5				5		5	
T'			7	6		7		7
F	9				8		10	
id	Emp.							
=		Emp.						
+			Emp.					
*				Emp.				
(Emp.			
)						Emp.		
número							Emp.	
\$								Aceptar

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Método de nivel de frase: reglas ϵ psilon)

5 / 16)

Tabla predictiva								
Símbolo de entrada								
	id	=	+	*	()	número	\$
S	1							
E	2				2		2	
E'	4*	4*	3	4*	4*	4	4*	4
T	5				5		5	
T'	7*	7*	7	6	7*	7	7*	7
F	9				8		10	
id	Emp.							
=		Emp.						
+			Emp.					
*				Emp.				
(Emp.			
)						Emp.		
número							Emp.	
\$								Acceptar

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Método de nivel de frase: parte superior

6 / 16)

Tabla predictiva

	Símbolo de entrada							
	id	=	+	*	()	número	\$
S	1	E1	E2	E2	E2	E2	E2	E3
E	2	E2	E4	E4	2	E2	2	E5
E'	4*	4*	3	4*	4*	4	4*	4
T	5	E2	E4	E4	5	E2	5	E5
T'	7*	7*	7	6	7*	7	7*	7
F	9	E2	E4	E4	8	E2	10	E5
id	Emp.							
=		Emp.						
+			Emp.					
*				Emp.				
(Emp.			
)						Emp.		
número							Emp.	
\$								Aceptar

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Método de nivel de frase: parte inferior

7 / 16)

Tabla predictiva

		Símbolo de entrada						
	id	=	+	*	()	número	\$
S	1	E1	E2	E2	E2	E2	E2	E3
E	2	E2	E4	E4	2	E2	2	E5
E'	4*	4*	3	4*	4*	4	4*	4
T	5	E2	E4	E4	5	E2	5	E5
T'	7*	7*	7	6	7*	7	7*	7
F	9	E2	E4	E4	8	E2	10	E5
id	Emp.							
=	E6	Emp.	E7	E7	E7	E7	E6	E8
+			Emp.					
*				Emp.				
(Emp.			
)	E9	E9	E10	E10	E9	Emp.	E9	E11
número							Emp.	
\$	E2	E2	E2	E2	E2	E2	E2	Aceptar

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Método de nivel de frase: errores

8 / 16)

- **E1**
 - *Símbolo inesperado: falta **identificador***
 - *Insertar **identificador** en la entrada*
- **E2**
 - *Símbolo inesperado*
 - *Eliminar símbolo de la entrada*
- **E3**
 - *Final de entrada inesperada*
 - *Eliminar símbolo de la pila*
- **E4**
 - *Símbolo inesperado: falta operando*
 - *Insertar **identificador** en la entrada*

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Método de nivel de frase: errores

9 /16)

- **E5**
 - *Final de entrada inesperada*
 - *Insertar **identificador** en la entrada*
- **E6**
 - *Símbolo inesperado: falta símbolo = en la entrada*
 - *Insertar = en la entrada*
- **E7**
 - *Símbolo inesperado: falta símbolo = en la entrada*
 - *Eliminar símbolo de la entrada*
- **E8**
 - *Fin de entrada inesperado: falta símbolo = en la entrada*
 - *Insertar = en la entrada*

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Método de nivel de frase: errores

10 / 16)

- **E9**
 - *Símbolo inesperado: falta símbolo) en la entrada*
 - *Eliminar símbolo de la entrada*
- **E10**
 - *Símbolo inesperado: falta símbolo) en la entrada*
 - *Insertar) en la entrada*
- **E11**
 - *Fin de entrada inesperado: falta símbolo) en la entrada*
 - *Insertar) en la entrada*

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Sentencias con errores

11 / 16)

1.- **id = = id n * id**

2.- **id id * + n**

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ E = id	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ E' T	id n * id \$	5) $T \rightarrow F T'$
\$ E' T' F	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' T' id	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	7*) $T \rightarrow \epsilon$
\$ E'	n * id \$	4*) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id n * id \$	5) $T \rightarrow F T'$
\$ <u>E' T' F</u>	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ <u>E' T' id</u>	id n * id \$	<i>Emparejar</i>
\$ <u>E' T'</u>	n * id \$	7*) $T \rightarrow \epsilon$
\$ <u>E'</u>	n * id \$	4*) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E</u> = id	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ E' T	id n * id \$	5) $T \rightarrow F T'$
\$ E' T' F	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' T' id	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	7*) $T' \rightarrow \epsilon$
\$ E'	n * id \$	4*) $E' \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E</u> = id	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E$
\$ E' T	id n * id \$	5) $T \rightarrow F T$
\$ E' T' E	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' T' id	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	7*) $T \rightarrow \epsilon$
\$ E'	n * id \$	4*) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ E' T	id n * id \$	5) $T \rightarrow F T'$
\$ E' T' F	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' T' id	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	7*) $T \rightarrow \epsilon$
\$ E'	n * id \$	4*) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id n * id \$	5) $T \rightarrow F T'$
\$ E' T E	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' T id	id n * id \$	<i>Emparejar</i>
\$ E' T	n * id \$	7) $T \rightarrow \epsilon$
\$ E'	n * id \$	4) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id n * id \$	5) $T \rightarrow F T'$
\$ E' <u>T' F</u>	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' T' id	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	7) $T \rightarrow \epsilon$
\$ E'	n * id \$	4) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id n * id \$	5) $T \rightarrow F T'$
\$ E' <u>T' F</u>	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' <u>T' id</u>	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	<i>7) $T' \rightarrow \epsilon$</i>
\$ E'	n * id \$	<i>4) $E \rightarrow \epsilon$</i>
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id n * id \$	5) $T \rightarrow F T'$
\$ E' <u>T' F</u>	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' <u>T' id</u>	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	7*) $T \rightarrow \epsilon$
\$ E'	n * id \$	4*) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id n * id \$	5) $T \rightarrow F T'$
\$ E' <u>T' F</u>	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' <u>T' id</u>	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	7*) $T \rightarrow \epsilon$
\$ E'	n * id \$	4*) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id n * id \$	5) $T \rightarrow F T'$
\$ E' <u>T' F</u>	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' <u>T' id</u>	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	7*) $T \rightarrow \epsilon$
\$ E'	n * id \$	4*) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id n * id \$	5) $T \rightarrow F T'$
\$ E' <u>T' F</u>	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' <u>T' id</u>	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	7*) $T \rightarrow \epsilon$
\$ E'	n * id \$	4*) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id n * id \$	5) $T \rightarrow F T'$
\$ E' <u>T' F</u>	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ E' <u>T' id</u>	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	7*) $T \rightarrow \epsilon$
\$ E'	n * id \$	4*) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores)

12 / 16)

Pila	Entrada	Acción
\$ S	id = = id n * id \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id = = id n * id \$	<i>Emparejar</i>
\$ E =	= = id n * id \$	<i>Emparejar</i>
\$ E	= id n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$ E	id n * id \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id n * id \$	5) $T \rightarrow F T'$
\$ <u>E' T' F</u>	id n * id \$	9) $F \rightarrow \text{identificador}$
\$ <u>E' T' id</u>	id n * id \$	<i>Emparejar</i>
\$ E' T'	n * id \$	7*) $T \rightarrow \epsilon$
\$ E'	n * id \$	4*) $E \rightarrow \epsilon$
\$	n * id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	* id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	id \$	<i>E2: eliminar símbolo de la entrada</i>
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Primera sentencia con errores

13 / 16)

- *Sentencia de entrada:*

id = = id n * id

- *Sentencia propuesta por el método de nivel de frase:*

id = id

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

14 / 16)

Pila	Entrada	Acción
\$ S	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E</u> = id	id id * + n \$	<i>Emparejar</i>
\$ E =	id * + n \$	<i>E6: insertar =</i>
\$ E =	= id * + n \$	<i>Emparejar</i>
\$ E	id * + n \$	2) $E \rightarrow T E'$
\$ E' <u>T</u>	id * + n \$	5) $T \rightarrow F T'$
\$ E' <u>T</u> F	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ E' <u>T</u> id	id * + n \$	<i>Emparejar</i>
\$ E' T'	* + n \$	6) $T' \rightarrow * F T'$
\$ E' T' F *	* + n \$	<i>Emparejar</i>
\$ E' T' F	+ n \$	<i>E4: insertar id</i>
\$ E' T' F	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

14 / 16)

Pila	Entrada	Acción
\$ S	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id id * + n \$	<i>Emparejar</i>
\$ E =	id * + n \$	<i>E4: insertar =</i>
\$ E =	= id * + n \$	<i>Emparejar</i>
\$ E	id * + n \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id * + n \$	5) $T \rightarrow F T'$
\$ <u>E' T' F</u>	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ <u>E' T' id</u>	id * + n \$	<i>Emparejar</i>
\$ <u>E' T'</u>	* + n \$	6) $T' \rightarrow * F T'$
\$ <u>E' T' F *</u>	* + n \$	<i>Emparejar</i>
\$ <u>E' T' F</u>	+ n \$	<i>E4: insertar id</i>
\$ <u>E' T' F</u>	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

14 / 16)

Pila	Entrada	Acción
\$ S	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id id * + n \$	<i>Emparejar</i>
\$ E =	id * + n \$	<i>E6: insertar =</i>
\$ E =	= id * + n \$	<i>Emparejar</i>
\$ E	id * + n \$	2) $E \rightarrow T E'$
\$ <u>E T</u>	id * + n \$	5) $T \rightarrow F T'$
\$ <u>E T F</u>	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ <u>E T F</u>	id * + n \$	<i>Emparejar</i>
\$ E T'	* + n \$	6) $T' \rightarrow * F T'$
\$ E T' F *	* + n \$	<i>Emparejar</i>
\$ E T' F	+ n \$	<i>E4: insertar id</i>
\$ E T' F	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

14 / 16)

Pila	Entrada	Acción
\$ S	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id id * + n \$	<i>Emparejar</i>
\$ E =	id * + n \$	<i>E6: insertar =</i>
\$ E =	= id * + n \$	<i>Emparejar</i>
\$ E	id * + n \$	2) $E \rightarrow T E'$
\$ <u>E T</u>	id * + n \$	5) $T \rightarrow F T'$
\$ E T F	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ E T F	id * + n \$	<i>Emparejar</i>
\$ E T'	* + n \$	6) $T' \rightarrow * F T'$
\$ E T' F *	* + n \$	<i>Emparejar</i>
\$ E T' F	+ n \$	<i>E4: insertar id</i>
\$ E T' F	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores

14 / 16)

Pila	Entrada	Acción
\$ S	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ E = id	id id * + n \$	<i>Emparejar</i>
\$ E =	id * + n \$	<i>E6: insertar =</i>
\$ E =	= id * + n \$	<i>Emparejar</i>
\$ E	id * + n \$	2) $E \rightarrow T E'$
\$ E T	id * + n \$	3) $T \rightarrow F T'$
\$ E T F	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ E T F id	id * + n \$	<i>Emparejar</i>
\$ E T'	* + n \$	6) $T' \rightarrow * F T'$
\$ E T' F *	* + n \$	<i>Emparejar</i>
\$ E T' F	+ n \$	<i>E4: insertar id</i>
\$ E T' F	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

14 / 16)

Pila	Entrada	Acción
\$ S	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id id * + n \$	<i>Emparejar</i>
\$ E =	id * + n \$	<i>E6: insertar =</i>
\$ E =	= id * + n \$	<i>Emparejar</i>
\$ E	id * + n \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id * + n \$	5) $T \rightarrow F T'$
\$ E' T F	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ E' T id	id * + n \$	<i>Emparejar</i>
\$ E' T'	* + n \$	6) $T' \rightarrow * F T'$
\$ E' T' F *	* + n \$	<i>Emparejar</i>
\$ E' T' F	+ n \$	<i>E4: insertar id</i>
\$ E' T' F	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

14 / 16)

Pila	Entrada	Acción
\$ <u>S</u>	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id id * + n \$	<i>Emparejar</i>
\$ <u>E =</u>	id * + n \$	<i>E6: insertar =</i>
\$ <u>E =</u>	= id * + n \$	<i>Emparejar</i>
\$ <u>E</u>	id * + n \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id * + n \$	5) $T \rightarrow F T'$
\$ <u>E' T' F</u>	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ <u>E' T' id</u>	id * + n \$	<i>Emparejar</i>
\$ <u>E' T'</u>	* + n \$	6) $T' \rightarrow * F T'$
\$ <u>E' T' F *</u>	* + n \$	<i>Emparejar</i>
\$ <u>E' T' F</u>	+ n \$	<i>E4: insertar id</i>
\$ <u>E' T' F</u>	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

14 / 16)

Pila	Entrada	Acción
\$ <u>S</u>	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>E = id</u>	id id * + n \$	<i>Emparejar</i>
\$ <u>E =</u>	id * + n \$	<i>E6: insertar =</i>
\$ <u>E =</u>	= id * + n \$	<i>Emparejar</i>
\$ <u>E</u>	id * + n \$	2) $E \rightarrow T E'$
\$ <u>E' T</u>	id * + n \$	5) $T \rightarrow F T'$
\$ <u>E' T' F</u>	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ <u>E' T' id</u>	id * + n \$	<i>Emparejar</i>
\$ <u>E' T'</u>	* + n \$	6) $T' \rightarrow * F T'$
\$ <u>E' T' F *</u>	* + n \$	<i>Emparejar</i>
\$ <u>E' T' F</u>	+ n \$	<i>E4: insertar id</i>
\$ <u>E' T' F</u>	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

14 / 16)

Pila	Entrada	Acción
\$ <u>S</u>	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>$E = \text{id}$</u>	id id * + n \$	<i>Emparejar</i>
\$ <u>$E =$</u>	id * + n \$	<i>E6: insertar =</i>
\$ <u>$E =$</u>	= id * + n \$	<i>Emparejar</i>
\$ <u>E</u>	id * + n \$	2) $E \rightarrow T E'$
\$ <u>$E' T$</u>	id * + n \$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ <u>$E' T' \text{id}$</u>	id * + n \$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	* + n \$	6) $T' \rightarrow * F T'$
\$ <u>$E' T' F *$</u>	* + n \$	<i>Emparejar</i>
\$ <u>$E' T' F$</u>	+ n \$	<i>E4: insertar id</i>
\$ <u>$E' T' F$</u>	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores

14 / 16)

Pila	Entrada	Acción
\$ <u>S</u>	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>$E = \text{id}$</u>	id id * + n \$	<i>Emparejar</i>
\$ <u>$E =$</u>	id * + n \$	<i>E6: insertar =</i>
\$ <u>$E =$</u>	= id * + n \$	<i>Emparejar</i>
\$ <u>E</u>	id * + n \$	2) $E \rightarrow T E'$
\$ <u>$E' T$</u>	id * + n \$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ <u>$E' T' \text{id}$</u>	id * + n \$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	* + n \$	6) $T' \rightarrow * F T'$
\$ <u>$E' T' F *$</u>	* + n \$	<i>Emparejar</i>
\$ <u>$E' T' F$</u>	+ n \$	<i>E4: insertar id</i>
\$ <u>$E' T' F$</u>	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

14 / 16)

Pila	Entrada	Acción
\$ <u>S</u>	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>$E = \text{id}$</u>	id id * + n \$	<i>Emparejar</i>
\$ <u>$E =$</u>	id * + n \$	<i>E6: insertar =</i>
\$ <u>$E =$</u>	= id * + n \$	<i>Emparejar</i>
\$ <u>E</u>	id * + n \$	2) $E \rightarrow T E'$
\$ <u>$E' T$</u>	id * + n \$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ <u>$E' T' \text{id}$</u>	id * + n \$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	* + n \$	6) $T' \rightarrow * F T'$
\$ <u>$E' T' F *$</u>	* + n \$	<i>Emparejar</i>
\$ <u>$E' T' F$</u>	+ n \$	<i>E4: insertar id</i>
\$ <u>$E' T' F$</u>	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

14 / 16)

Pila	Entrada	Acción
\$ <u>S</u>	id id * + n \$	1) $S \rightarrow \text{identificador} = E$
\$ <u>$E = \text{id}$</u>	id id * + n \$	<i>Emparejar</i>
\$ <u>$E =$</u>	id * + n \$	<i>E6: insertar =</i>
\$ <u>$E =$</u>	= id * + n \$	<i>Emparejar</i>
\$ <u>E</u>	id * + n \$	2) $E \rightarrow T E'$
\$ <u>$E' T$</u>	id * + n \$	5) $T \rightarrow F T'$
\$ <u>$E' T' F$</u>	id * + n \$	9) $F \rightarrow \text{identificador}$
\$ <u>$E' T' \text{id}$</u>	id * + n \$	<i>Emparejar</i>
\$ <u>$E' T'$</u>	* + n \$	6) $T' \rightarrow * F T'$
\$ <u>$E' T' F *$</u>	* + n \$	<i>Emparejar</i>
\$ <u>$E' T' F$</u>	+ n \$	<i>E4: insertar id</i>
\$ <u>$E' T' F$</u>	id + n \$	9) $F \rightarrow \text{identificador}$

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores

15 / 16)

Pila	Entrada	Acción
\$ E' T' F	id + n \$	9) $F \rightarrow \text{identificador}$
\$ E' T' id	id + n \$	<i>Emparejar</i>
\$ E' T'	+ n \$	7) $T' \rightarrow \epsilon$
\$ E'	+ n \$	3) $E' \rightarrow + T E'$
\$ E' T +	+ n \$	<i>Emparejar</i>
\$ E' T	n \$	5) $T \rightarrow F T'$
\$ E' T' F	n \$	10) $F \rightarrow n$
\$ E' T' n	n \$	<i>Emparejar</i>
\$ E' T'	\$	7) $T' \rightarrow \epsilon$
\$ E'	\$	4) $E' \rightarrow \epsilon$
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores

15 / 16)

Pila	Entrada	Acción
\$ E' T' F	id + n \$	9) $F \rightarrow \text{identificador}$
\$ E' T' <u>id</u>	id + n \$	<i>Emparejar</i>
\$ E' T	+ n \$	7) $T \rightarrow \epsilon$
\$ E'	+ n \$	3) $E' \rightarrow + T E'$
\$ E' T +	+ n \$	<i>Emparejar</i>
\$ E' T	n \$	5) $T \rightarrow F T'$
\$ E' T' F	n \$	10) $F \rightarrow n$
\$ E' T' n	n \$	<i>Emparejar</i>
\$ E' T'	\$	7) $T' \rightarrow \epsilon$
\$ E'	\$	4) $E' \rightarrow \epsilon$
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

15 / 16)

Pila	Entrada	Acción
\$ E' T' F	id + n \$	9) $F \rightarrow \text{identificador}$
\$ E' T' <u>id</u>	id + n \$	<i>Emparejar</i>
\$ E' T'	+ n \$	7) $T' \rightarrow \epsilon$
\$ E'	+ n \$	3) $E' \rightarrow + T E'$
\$ E' T +	+ n \$	<i>Emparejar</i>
\$ E' T	n \$	5) $T \rightarrow F T'$
\$ E' T' F	n \$	10) $F \rightarrow n$
\$ E' T' n	n \$	<i>Emparejar</i>
\$ E' T'	\$	7) $T' \rightarrow \epsilon$
\$ E'	\$	4) $E' \rightarrow \epsilon$
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores

15 / 16)

Pila	Entrada	Acción
\$ E' T' F	id + n \$	9) $F \rightarrow \text{identificador}$
\$ E' T' <u>id</u>	id + n \$	<i>Emparejar</i>
\$ E' T'	+ n \$	7) $T' \rightarrow \epsilon$
\$ E'	+ n \$	3) $E' \rightarrow + T E'$
\$ E' T +	+ n \$	<i>Emparejar</i>
\$ E' T	n \$	5) $T \rightarrow F T'$
\$ E' T' F	n \$	10) $F \rightarrow n$
\$ E' T' n	n \$	<i>Emparejar</i>
\$ E' T'	\$	7) $T' \rightarrow \epsilon$
\$ E'	\$	4) $E' \rightarrow \epsilon$
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores

15 / 16)

Pila	Entrada	Acción
\$ E' T' F	id + n \$	9) $F \rightarrow \text{identificador}$
\$ E' T' <u>id</u>	id + n \$	<i>Emparejar</i>
\$ E' T'	+ n \$	7) $T' \rightarrow \epsilon$
\$ E'	+ n \$	3) $E' \rightarrow + T E'$
\$ E' T +	+ n \$	<i>Emparejar</i>
\$ E' T	n \$	5) $T \rightarrow F T'$
\$ E' T' F	n \$	10) $F \rightarrow n$
\$ E' T' n	n \$	<i>Emparejar</i>
\$ E' T'	\$	7) $T' \rightarrow \epsilon$
\$ E'	\$	4) $E' \rightarrow \epsilon$
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores

15 / 16)

Pila	Entrada	Acción
\$ E' T' F	id + n \$	9) $F \rightarrow \text{identificador}$
\$ E' T' <u>id</u>	id + n \$	<i>Emparejar</i>
\$ E' T'	+ n \$	7) $T' \rightarrow \epsilon$
\$ E'	+ n \$	3) $E' \rightarrow + T E'$
\$ E' T +	+ n \$	<i>Emparejar</i>
\$ <u>E'</u> T	n \$	5) $T \rightarrow F T'$
\$ E' T' F	n \$	10) $F \rightarrow n$
\$ E' T' n	n \$	<i>Emparejar</i>
\$ E' T'	\$	7) $T' \rightarrow \epsilon$
\$ E'	\$	4) $E' \rightarrow \epsilon$
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores

15 / 16)

Pila	Entrada	Acción
\$ <u>E' T' F</u>	id + n \$	9) $F \rightarrow \text{identificador}$
\$ E' T' <u>id</u>	id + n \$	<i>Emparejar</i>
\$ E' T'	+ n \$	7) $T' \rightarrow \epsilon$
\$ E'	+ n \$	3) $E' \rightarrow + T E'$
\$ E' T +	+ n \$	<i>Emparejar</i>
\$ <u>E' T</u>	n \$	5) $T \rightarrow F T'$
\$ <u>E' T' F</u>	n \$	10) $F \rightarrow n$
\$ E' T n	n \$	<i>Emparejar</i>
\$ E' T'	\$	7) $T' \rightarrow \epsilon$
\$ E'	\$	4) $E' \rightarrow \epsilon$
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores

15 / 16)

Pila	Entrada	Acción
\$ $E' T' F$	id + n \$	9) $F \rightarrow$ identificador
\$ $E' T' \underline{id}$	id + n \$	<i>Emparejar</i>
\$ $E' T'$	+ n \$	7) $T' \rightarrow \epsilon$
\$ E'	+ n \$	3) $E' \rightarrow + T E'$
\$ $E' T +$	+ n \$	<i>Emparejar</i>
\$ $E' \underline{T}$	n \$	5) $T \rightarrow F T'$
\$ $E' T' F$	n \$	10) $F \rightarrow n$
\$ $E' T' n$	n \$	<i>Emparejar</i>
\$ $E' T'$	\$	7) $T' \rightarrow \epsilon$
\$ E'	\$	4) $E' \rightarrow \epsilon$
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

15 / 16)

Pila	Entrada	Acción
\$ $E' T' F$	id + n \$	9) $F \rightarrow$ identificador
\$ $E' T' \underline{id}$	id + n \$	<i>Emparejar</i>
\$ $E' T'$	+ n \$	7) $T' \rightarrow \epsilon$
\$ E'	+ n \$	3) $E' \rightarrow + T E'$
\$ $E' T +$	+ n \$	<i>Emparejar</i>
\$ $E' \underline{T}$	n \$	5) $T \rightarrow F T'$
\$ $E' T' F$	n \$	10) $F \rightarrow n$
\$ $E' T' n$	n \$	<i>Emparejar</i>
\$ $E' T'$	\$	7) $T' \rightarrow \epsilon$
\$ E'	\$	4) $E' \rightarrow \epsilon$
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

15 / 16)

Pila	Entrada	Acción
\$ $E' T' F$	id + n \$	9) $F \rightarrow$ identificador
\$ $E' T' \underline{id}$	id + n \$	<i>Emparejar</i>
\$ $E' T'$	+ n \$	7) $T' \rightarrow \epsilon$
\$ E'	+ n \$	3) $E' \rightarrow + T E'$
\$ $E' T +$	+ n \$	<i>Emparejar</i>
\$ $E' \underline{T}$	n \$	5) $T \rightarrow F T'$
\$ $E' T' F$	n \$	10) $F \rightarrow n$
\$ $E' T' n$	n \$	<i>Emparejar</i>
\$ $E' T'$	\$	7) $T' \rightarrow \epsilon$
\$ E'	\$	4) $E' \rightarrow \epsilon$
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores)

15 / 16)

Pila	Entrada	Acción
\$ $E' T' F$	id + n \$	9) $F \rightarrow$ identificador
\$ $E' T' \underline{\text{id}}$	id + n \$	<i>Emparejar</i>
\$ $E' T'$	+ n \$	7) $T' \rightarrow \epsilon$
\$ E'	+ n \$	3) $E' \rightarrow + T E'$
\$ $E' T +$	+ n \$	<i>Emparejar</i>
\$ $\underline{E' T}$	n \$	5) $T \rightarrow F T'$
\$ $\underline{E' T' F}$	n \$	10) $F \rightarrow n$
\$ $E' T' n$	n \$	<i>Emparejar</i>
\$ $E' T'$	\$	7) $T' \rightarrow \epsilon$
\$ E'	\$	4) $E' \rightarrow \epsilon$
\$	\$	<i>Aceptar</i>

Detección y recuperación de errores

Método de nivel de frase

Ejemplo (Segunda sentencia con errores

16 / 16)

- *Sentencia de entrada:*

id id * + n

- *Sentencia propuesta por el método de nivel de frase:*

id = id * id + n

PROCESADORES DE LENGUAJES

TEMA IV.- ANÁLISIS SINTÁCTICO DESCENDENTE

Prof. Dr. Nicolás Luis Fernández García

Departamento de Informática y Análisis Numérico
Escuela Politécnica Superior de Córdoba
Universidad de Córdoba