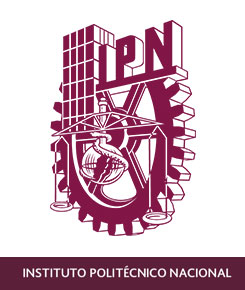
****

**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**

**COMPILADORES**

**GURPO: 5CM4**

**TERCERA GUÍA.**

**PROFESOR:**

**TECLA PARRA ROBERTO**

**ALUMNO:**

**LOPEZ PEREZ ALBERTO ANDREI**

3ª Guía de Compiladores

Falso o Verdadero (F/V)

l.-**Componente léxico** es sinónimo de no terminal (F)

2.-**Token** es sinónimo de no terminal (F)

3.-Una **gramática** **no** es **ambigua** si existe alguna cadena de terminales que pueda obtenerse mediante (F) árboles de análisis sintáctico distintos (dos árboles distintos dan la misma cadena)

4.-Dos **gramáticas** son **equivalentes** si generan el mismo lenguaje. (V*)*

5.-En el **Análisis sintáctico ascendente** el árbol de análisis sintáctico la construcción se inicia en la raíz y (F)

avanza hacia las hojas

6.-En el **Análisis sintáctico descendente** se construye el árbol de análisis sintáctico de la cadena de desde (F)

las hojas y avanza hacia las raíz

7.-el **árbol sintáctico** no es una versión condensada del **árbol de análisis sintáctico** (F)

8.-Un **Esquema de traducción** asocia a cada símbolo de una **GLC** un conjunto de atributos y a cada (F) producción, un conjunto de reglas semánticas para calcular los valores de los atributos asociados con los

símbolos que aparecen en esa producción.

9.-**Definición dirigida por la sintaxis** es una **GLC** en la que se encuentran intercalados, en los lados (F*)*

derechos de las producciones, fragmentos de programa llamados acciones semánticas.

10.-Los valores de los **atributos sintetizados** se calculan a partir de los valores de atributos de su nodo (V) padre o sus nodos hermanos.

11.-Un **atributo** es **heredado** si su valor depende de los valores de los atributos de su **padre** y/o de sus (V) **hermanos**.

12.-El **código de tres direcciones** consiste en una secuencia de instrucciones, cada una de las cuales (V) tiene corno máximo tres operandos.

-En lenguaje C los **parámetros formales** no tienen nombre

13.-En lenguaje C los **parámetros formales** son como **variables locales** que ya fueron inicializadas en el (V) momento de la llamada a la función o procedimiento

14.-En lenguaje C las **variables locales** (no estáticas) se crean cuando se entra a una función y se destruyen (V) cuando se sale de la función.

15.-En hoc los **parámetros usados dentro de la definición de las funciones** no tienen (V) nombre\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

16.-No es posible definir **funciones recursivas** en hoc\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (F)

17.-En hoc no hay **variables locales**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (V)

18.-En hoc cuando una función termina su ejecución se **saca** su **marco** de la **pila de llamadas**. (V) 19.-En hoc los **parámetros reales** son listas de **expresiones** (V)

20.-En hoc el código que ejecuta la **máquina virtual de** pila está en prefijo (considere como se ejecuta una (F) operación de suma)

21.-En hoc el **tipo** de las **variables** es entero (F)

22.-En hoc el **tipo** de las **variables** es doble (V)

23.-En hoc los **parámetros reales** se meten a la pila (V)

24.-Es imposible que la **pila de llamadas** de hoc se **desborde** (Stack Overflow) (F)

25.-EI **código intermedio** debe ser **fácil** de **generar** (V)

26.-La fase de **generación de código intermedio** es cuando se hace la asignación de **variables** a **registros** (F)

1.-El **código de t res direcciones** se usa en

El análisis sintáctico

Generación de código intermedio

El análisis léxico (c)

Generación de código

2.-Es la **fase final** de un **compilador**

a) análisis semántico

c) optimización de código

b) manejo de errores (d)

1. generación de código

# **Análisis Sintáctico Predictivo no Recursivo**

-Use dicho análisis para analizar las cadenas propuestas:

-Muestre el contenido de la pila, la entrada y la salida

-Muestre la derivación por la izquierda

**tabla de análisis sintáctico**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Gramática** |  | No terminal | Símbolo de entrada | | | | | |
| a | b | e | if | t | $ |
| P→iEtPP’| a  P’→eP| ε  E →b |  | P | P→a |  |  | P→ifEtPP’ |  |  |
|  | P’ |  |  | P’→eP  P’→ε |  |  | P’→ ε |
|  | E |  | E→b |  |  |  |  |

cadenas:

**if b t a**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pila** | **Entrada** | **Salida** | **Símbolo no terminal** | **Símbolo terminal** |
| $P | if b t a$ | P→ifEtPP’ | P | if |
| $P’PtEif | if b t a$ |  | if | if |
| $P’PtE | b t a$ | E→b | E | b |
| $P’Ptb | b t a$ |  | b | b |
| $P’Pt | t a$ |  | t | t |
| $P’P | a$ | P→a | P | a |
| $P’a | a$ |  | a | a |
| $P’ | $ | P’→ ε | P’ | $ |
| $ | $ |  | $ | $ |

P→ if E t P P’ → if b t P P’ → if b t a P’ → if b t a

**if b t a e a**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pila** | **Entrada** | **Salida** | **Símbolo no terminal** | **Símbolo terminal** |
| $P | if b t a e a$ | P→ifEtPP’ | P | if |
| $P’PtEif | if b t a e a$ |  | if | if |
| $P’PtE | b t a e a$ | E→b | E | b |
| $P’Ptb | b t a e a$ |  | b | b |
| $P’Pt | t a e a$ |  | t | t |
| $P’P | a e a$ | P→a | P | a |
| $P’a | a e a$ |  | a | a |
| $P’ | e a$ | P’→eP | P’ | e |
| $Pe | e a$ |  | e | e |
| $P | a$ | P→a | P | a |
| $a | a$ |  | a | a |
| $ | $ |  | $ | $ |

P→ if E t P P’ → if b t P P’ → if b t a P’ → if b t a e P → if b t a e a

**if b t if b t a e a**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pila** | **Entrada** | **Salida** | **Símbolo no terminal** | **Símbolo terminal** |
| $P | if b t if b t a e a$ | P→ifEtPP’ | P | if |
| $P’PtEif | if b t if b t a e a$ |  | if | if |
| $P’PtE | b t if b t a e a$ | E→b | E | b |
| $P’Ptb | b t if b t a e a$ |  | b | b |
| $P’Pt | t if b t a e a$ |  | t | t |
| $P’P | if b t a e a$ | P→ifEtPP’ | P | if |
| $P’P’PtEif | if b t a e a$ |  | if | if |
| $P’P’PtE | b t a e a$ | E→b | E | b |
| $P’P’Ptb | b t a e a$ |  | b | b |
| $P’P’Pt | t a e a$ |  | t | t |
| $P’P’P | a e a$ | P→a | P | a |
| $P’P’a | a e a$ |  | a | a |
| $P’P’ | e a$ | P’→eP | P’ | e |
| $P’Pe | e a$ |  | e | e |
| $P’P | a$ | P→a | P | a |
| $P’a | a$ |  | a | a |
| $ P’ | $ | P’→ ε | P’ | $ |
| $ | $ |  | $ | $ |

P → ifEtPP’ → if b t PP’ → if b t if E t PP’P’ → if b t if b t PP’P’ → if b t if b t a P’P’ → if b t if b t a e PP’ → if b t if b t a e a P’ → if b t if b t a e a

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Gramática** |  |  | ( | ) | ; | x | $ |
| S→(A)  A→CB  B→;A|  C→x|S |  | S | S→(A) |  |  |  |  |
|  | A | A→CB |  |  | A→CB |  |
|  | B |  | B→ε | B→:A |  |  |
|  | C | C→S |  |  | C→x |  |

cadena **(x:(x))**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pila** | **Entrada** | **Salida** | **Símbolo no terminal** | **Símbolo terminal** |
| **$S** | **(x:(x)) $** | S→(A) | S | ( |
| **$)A(** | **(x:(x)) $** |  | ( | ( |
| **$)A** | **x:(x)) $** | A→CB | A | x |
| **$)BC** | **x:(x)) $** | C→x | C | x |
| **$)Bx** | **x:(x)) $** |  | x | x |
| **$)B** | **:(x)) $** | B→:A | B | : |
| **$)A:** | **:(x)) $** |  | : | : |
| **$)A** | **(x)) $** | A→CB | A | ( |
| **$)BC** | **(x)) $** | C→S | C | ( |
| **$)BS** | **(x)) $** | S→(A) | S | ( |
| **$)B)A(** | **(x)) $** | A→CB | ( | ( |
| **$)B)A** | **x)) $** |  | A | x |
| **$)B)BC** | **x)) $** | C→x | C | x |
| **$)B)Bx** | **x))$** |  | x | x |
| **$)B)B** | **))$** | B→ε | B | ) |
| **$)B)** | **))$** |  | ) | ) |
| **$)B** | **)$** | B→ε | B | ) |
| **$)** | **)$** |  | ) | ) |
| **$** | **$** |  | $ | $ |

**S**→(A) →(CB) →(xB) →(x:A) →(x:CB) →(x:SB) →(x:(A)B) →(x:(CB)B) →(x:(xB)B) →(x:(x)B) →(x:(x)).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Gramática |  |  | b | $ |
| S→A  A→bbA  A→ε |  | S | S→A | S→A |
| A | A→bbA | A→ε |

cadenas:

**bb**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pila** | **Entrada** | **Salida** | **Símbolo no terminal** | **Símbolo terminal** |
| $S | bb$ | S→A | S | b |
| $A | bb$ | A→bbA | A | b |
| $Abb | bb$ |  | b | b |
| $Ab | b$ |  | b | b |
| $A | $ | A→ε | A | $ |
| $ | $ |  | $ | $ |

S → A → bbA → bb

**bbbb**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pila** | **Entrada** | **Salida** | **Símbolo no terminal** | **Símbolo terminal** |
| $S | bbbb$ | S→A | S | b |
| $A | bbbb$ | A→bbA | A | b |
| $Abb | bbbb$ |  | b | b |
| $Abb | bbbb$ |  | b | b |
| $Ab | bbb$ |  | b | b |
| $A | bb$ | A→bbA | A | b |
| $Abb | bb$ |  | b | b |
| $Ab | b$ |  | b | b |
| $A | $ | A→ε | A | $ |
| $ | $ |  | $ | $ |

**S** → A → bbA → bbbbA → bbbb

**bbbbbb**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pila** | **Entrada** | **Salida** | **Símbolo no terminal** | **Símbolo terminal** |
| $S | bbbbbb$ | S→A | S | b |
| $A | bbbbbb$ | A→bbA | A | b |
| $Abb | bbbbbb$ |  | b | b |
| $Ab | bbbbb$ |  | b | b |
| $A | bbbb$ | A→bbA | A | b |
| $Abb | bbbb$ |  | b | b |
| $Abb | bbbb$ |  | b | b |
| $Ab | bbb$ |  | b | b |
| $A | bb$ | A→bbA | A | b |
| $Abb | bb$ |  | b | b |
| $Ab | b$ |  | b | b |
| $A | $ | A→ε | A | $ |
| $ | $ |  | $ | $ |

**Análisis LR**

**Problema**. - Considere la siguiente gramática:

|  |  |  |
| --- | --- | --- |
| 1)C→AB | 2)A→a | 3)B→a |

cadena: **aa**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  | | |
|  | a | $ | A | | B | C |
| 0 | d1 |  | 3 | |  | 2 |
| 1 | r2 |  |  | |  |  |
| 2 |  | ac |  | |  |  |
| 3 | d4 |  |  | | 5 |  |
| 4 |  | r3 |  | |  |  |
| 5 |  | r1 |  | |  |  |

-Use análisis LR para analizar la cadena propuesta:

-Muestre el contenido de la pila, la entrada y la salida

-Muestre la derivación por la derecha

Tabla

Descripción generada automáticamente