Laboratorio Lenguajes Formales y de Programación A+

Clase 7

Agenda

- Dudas
- Anuncios
- Introducción a gramáticas tipo 2
- Revisiones

Gramáticas Independientes del Contexto

Recordando un poco...

Este tipo de gramáticas generan los lenguajes libres del contexto.

Son la base para construir la sintaxis de la mayoría de lenguajes.

Son todos los lenguajes que pueden ser conocidos por los autómatas de pila.

EJEMPLO DE SINTAXIS DE DECLARACIONES

Python	`a = 0`
C#	`int a = 0;`
JavaScript	`var a = 0;`

Restricciones

$$A \rightarrow \beta$$

A = es un no terminal y no acepta mas simbolos.

β = es una cadena de terminales y no terminales.

Diferencias Gramáticas Tipo 3 y Tipo 2

A diferencia de las gramáticas regulares, estas gramáticas no tienen restricciones con respecto a la forma del lado derecho de sus reglas de escritura, aunque aún se requiere que el lado izquierdo de cada regla sea un solo no terminal,

Notación

Las gramáticas libres del contexto se escriben frecuentemente utilizando una notación conocida como BNF(Black-Naur Form) que es la tecnica mas comun para definir la sintaxis de los lenguajes de programación.

BNF

En esta notación se deben seguir las siguientes convenciones:

- Los no terminales se escriben entre paréntesis angulares.
- Los terminales se representa con cadenas de caracteres sin paréntesis angulares.
- El lado izquierdo de la regla debe tener un solo no terminal.
- El símbolo ::= que se lee “se define como” o “se reescribe como”

Ejemplo de sintaxis BNF

```
<INICIO>::= Tk_numero <OTRO_NUMERO>
          | Tk_numero <OTRA_CADENA>
```

Ejemplos

Ejemplo 1

Escriba una gramática independiente del contexto que acepte cadenas binarias de longitud 3.

Como se ve para una gramática tipo 3

```
S0 -> 0 S1
S0 -> 1 S1
S1 -> 0 S2
S1 -> 1 S2
S2 -> 0 S3
S2 -> 1 S3
S3 -> epsilon
```

Como se ve para una gramática tipo 2

<INICIO>::= <NUMERO><NUMERO><NUMERO>

<NUMERO>::= 0 | 1

Ejemplo 2

Escriba una gramática tipo 2 que reconozca cadenas binarias con longitud par

Como se ve para una gramática tipo 3

```
S0 -> 0 S1

S0 -> 1 S1

S0 -> epsilon

S1 -> 0 S0

S2 -> 1 S0
```

Como se ve para una gramática tipo 2

```
<INICIO> ::= <NUMERO><NUMERO><INICIO>

         ::= Eplison

<NUMERO>:: 0 | 1
```

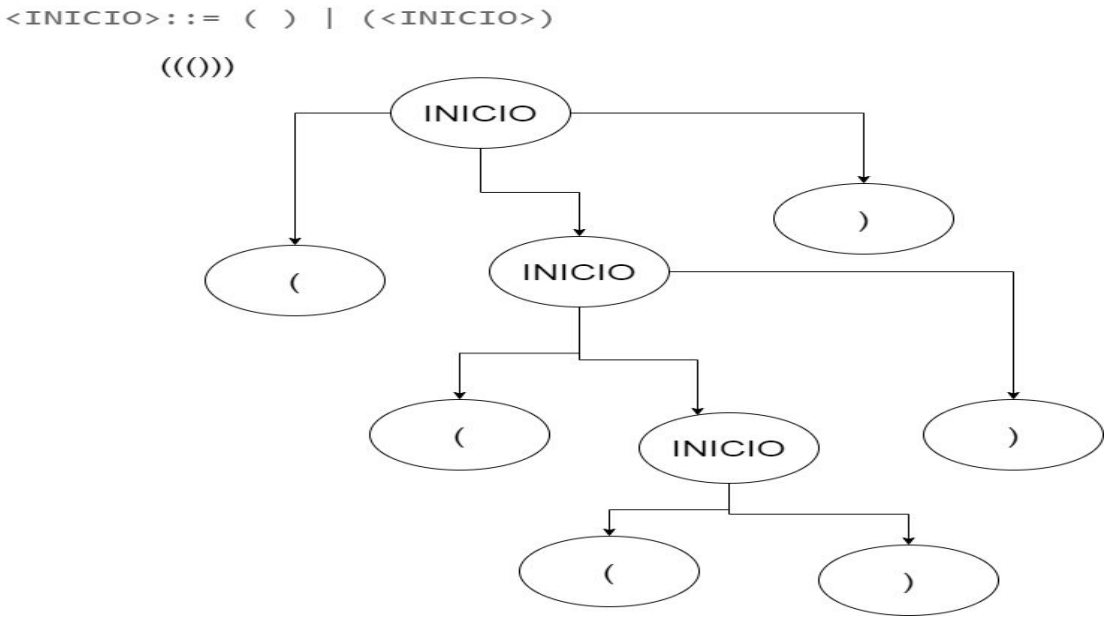
Ejemplo 3

Escriba una gramática tipo 2, que reconozca cadena de paréntesis anidados.

cadena de ejemplo: (), ((()), ((( )))

Solución

<INICIO>::= ( ) | (<INICIO>)



Ejemplo aplicado

Gramática tipo 2 para un while

Estructura básica

```
while (condicion){
    instruccion 1
    instruccion 2
    …
}
```

Solución

- Condición acepte condiciones de tipo igual (==) y que sea con números.

Solución

<WHILE>::= Tk_while Tk_ParA <CONDICION> Tk_ParC Tk_LlaveA

<INTRUCIONES> Tk_LlaveC

<CONDICION>::= Tk_Numero TK_Comparacion Tk_Numero

<INSTRUCIONES>::= <ASGINACIONES>

|<DECLARACIONES>