

Compiladores

Práctica 3a – Analizadores sintáctico descendente recursivo predictivo (1era parte)

Objetivo:

- Construir un analizador sintáctico descendente predictivo.
- Modelar una gramática.
- Escribir una tabla de análisis sintáctico
- Escribir el algoritmo de reconocimiento de una cadena utilizando nuestro analizador.

Indicaciones:

1. Estableciendo una gramática

Especificar los constructores y sobrecarga de operadores para las siguientes clases:

```
class Produccion:
    izq #Componente de la izquierda
    der #Componente(s) de la derecha

class Gramatica:
    produccion = [] #Lista de producciones
    terminales = {} #Conjunto de terminales
    noterminales = {} #no terminales
```

2. Crear métodos para cargar la gramática a partir de un texto y para obtener una producción.

```
class Gramatica:
    def cargar(texto):
        #cargar un texto en gramatica

    def getProduccion(izq):
        #retornar el componente(s) de la derecha de la
        #produccion apuntada por izq
```

3. Gramática a modelar:

```

E  := T Ep
Ep := + T Ep
Ep := - T Ep
Ep := lambda
T  := F Tp
Tp := * F Tp | / F Tp | lambda
F  := ( E ) | num | id
  
```

*** Implementar el método para imprimir la gramática**

4. Tabla sintáctica de predicción:

	+	-	*	/	()	num	id	\$
E					T Ep		T Ep	T Ep	
Ep	+ T Ep	- T Ep				lambda			lambda
T					F Tp		F Tp	F Tp	
Tp	lambda	lambda	* F Tp	/ F Tp		lambda			lambda
F					(E)		num	id	

Diseñar su solución y crear la tabla

```
tablaSintactica[NoTerminal][Terminal] = producción
```

por ejemplo..

```

def llenarEstaticamente():
    tablaSintactica["Ep"]["+"] = ["+", "T", "Ep"]
    tablaSintactica.insertar( "E", "(", ["T", "Ep"] )
  
```



Universidad Católica
San Pablo

#u otra forma que uds deseen