Universidad Católica San Pablo

Compiladores

Práctica 9 – Interpretador

Objetivos:

- Generar el árbol sintáctico con las clases que utilizan el patrón *interpreter*.
- Crear la interpretación de un pequeño ejemplo de suma y multiplicación.

1. Interpretación de los nodos no-terminales.

Una pequeña gramática

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

Llegamos a las siguientes conclusiones:

- **E,T** son operaciones conformadas por un valor atómico y una operación parcial.
- **F** constituye a un valor atómico.
- **Ep, Tp** son operaciones parciales. Es más:
  - Ep es la operación parcial de la suma.
  - TP es la operación parcial de multiplicación.

Llamaremos a valor atómico a un número o a una agrupación con paréntesis.

Operación parcial sería una expresión del tipo ***operación - valor atómico.***

Pregunta de revisión: ¿Cómo se hace la precedencia de operadores?

2. Formación de clases

Cada clase sintáctica corresponderá a un nodo terminal o nodo no terminal de nuestra gramática y deberá contener los componentes dentro de su clase, así:

```
class E(NNTerminal):
   def __init__(self):
      super().__init__('E')
      self.prod['T'] = None
      self.prod['Ep'] = None

   ...
```

Un caso especial es Ep ya que tiene dos posibles producciones. Para eso no existe una regla que indique cómo debería resolverse. Entonces una forma sería hacer una forma unida de las producciones y dejarle la responsabilidad de identificar la correcta producción a *interpret*.

```
class Ep(NNTerminal):
   def __init__(self):
      super().__init__('Ep')
      self.prod['+'] = 'None'
      self.prod['-']='None'
      self.prod['T'] = 'None'
      self.prod['Ep'] = 'None'
      self.prod['lambda'] = 'None'

   def interpret(self, anterior ):
      valorT = self.prod['T'].interpret()

      if self.prod['+'] != None:
         suma = anterior + valorT
         return self.prod['Ep'].interpret(suma)

      elif self.prod['-'] != None:
          resta = anterior - valorT
          return self.prod['Ep'].interpret(resta)
```

3. **Actividades:**

1. Crear el árbol sintáctico con las clases correctas.
2. Hacer las definiciones de todas las clases para esta gramática de ejemplo.
3. Probar con 1+1 =)