Universidad Católica San Pablo

Compiladores

Carlos Eduardo Atencio Torres

Semestre 2020-II

Práctica 3 - Analizador léxico

Objetivo:

Construir un analizador léxico utilizando funciones.

Definiciones:

Un analizador léxico lee los caracteres de la entrada y los agrupa en ***"objetos token"***.

Ejercicios

Tenemos un lenguaje que acepta las siguientes expresiones:

> 1+2
> (4*3)-8
> 60 / (5 + 3) - 0
> A=12
> variable1 =14
> C = A + variable1

1. Escriba una función que reciba una línea para la expresión y retorne una lista de tokens.

```
linea = “variable = tmp0 + 20”
tokens = analizadorLexico( linea )
```

2. Escriba dos funciones que reconozcan un número y el nombre de una variable. Considere lo siguiente:

a) No hay números con punto flotante o expresiones científicas. Sólo números enteros.
b) El número entero solo tiene dígitos [0-9]
c) El nombre de una variable debe empezar por una letra (minúscula o mayúscula). Seguidamente puede venir varios caracteres (números o letras).

3. Las funciones son:

```
def reconoceNumero(linea, idx)
def reconoceVariable(linea, idx)
```

Ellas reconocen respectivamente un número o una variable en linea, a partir de la posición idx. Ellas retornan el token y la nueva posición de idx para continuar con el análisis.

Orientación inicial:

```
def analizadorLexico( linea ) {
  tokens = []
  idx = 0
  while idx<len(linea):
     if linea[idx].isdigit():
        token,idx = reconoceNumero( linea, idx)
        <"1546",E,0>  ,  4
         <"78",E,13> , 15
        tokens.append(token);
     elif linea[idx].isalpha():
        token,idx = reconoceVariable( linea, idx)
        tokens.append(token);
     # Completar: obviar los espacios
     # Completar: tokenizar los operandos +-*/()
   return tokens;
```

4. Un token, además de encapsular a un elemento para su análisis, puede contener diferente información asociada a este. Por ejemplo, para el caso que estamos analizando, además de tener una copia del elemento tendremos información sobre en donde fue encontrado o qué tipo de token es.

Otra información que se utiliza en otros contextos pueden incluir por ejemplo el lema o lexema para el caso de procesamiento de lenguaje natural. Algunas veces no es necesaria una copia sino una referencia.

```
class Token:
  palabra = “”  #almacena una copia de la palabra
  indice = -1 #en donde apareció en la sentencia
  tipo = ‘’ #E (entero), V (variable), O (operador)
  #completar
```

5. Cree una función para imprimir la información de los tokens, así:

```
linea = “variable = tmp0 + 20 )”
tokens = analizadorLexico( linea )
for token in tokens:
  print( token.toString() )
```

El resultado sería:

```
Token[variable]: pos = 0, tipo = V
Token[=]: pos = 9, tipo = O
Token[tmp0]: pos = 11, tipo = V
Token[+]: pos = 16, tipo = O
Token[20]: pos = 18, tipo = E
Token[)]: pos=21, tipo = O
```