Gerador de Analisador Léxico

Ferramenta para gerar o código de analisadores léxicos automaticamente.

Utilizam expressões regulares para definir o padrão do token.

Expressões Regulares

Forma compacta de descrever uma sequência de caracteres.

Exemplo: Sequência de dígitos podendo conter apenas um ponto.

$$([0\text{-}9]^{*}\backslash.)^{*}[0\text{-}9]^{+}$$

Gerador de Analisador Léxico - Exemplos

Lex	– presente no UNIX (em C)
Flex	– presente no Linux (em C)
TPly	– parser em PASCAL
PLY	– parser de Python
JavaCC	– parser em Java
Flex++ ou Flexx	– parser em C++
ANTLR	– parser em diversas linguagens

PLY – Python Lex-Yacc

Ferramenta que ajuda a criar programas em Python para analisar e interpretar um arquivo de entrada.

Possui 2 módulos:

- lex.py : analisador léxico
- yacc.py : analisador sintático

PLY - lex.py

módulo que ajuda a criar programas em Python para analisar e interpretar um arquivo de entrada.

Transforma a entrada em conjunto de tokens através do uso de **expressões regulares.**

Gera a função **token()** que retorna o próximo token válido.

lexer = lex.lex()

PLY – Formato do Arquivo .py

```
import ply.lex as lex

% declaração dos tokens

% expressões regulares simples

% expressões regulares complexas

lexer = lex.lex()
```

PLY – Formato do Arquivo .py - Áreas

Declaração dos tokens

- Definir a variável **tokens** com uma lista dos tokens;
- As ERs que definem os tokens podem ser:
    - **Simples**: sequência de caracteres.
    - **Complexas**: regra definida por uma função.

Ex.: tokens = (*'TS1', 'TS2', 'TC3', 'TC4'*)

PLY – Formato do Arquivo .py - Áreas

Expressões regulares simples

- Definir uma variável para cada **token** com ER simples;

- A variável será iniciada por t_ seguida do nome do token e irá receber a sequência de caracteres que define o token.

PLY – Formato do Arquivo .py - Áreas

Expressões regulares simples

Ex.: considerando os tokens *TS1 e TS2 como tokens com ER simples:*

t_TS1 = r‘texto_ER_token1’
t_TS2 = r‘texto_ER_token2’

PLY – Formato do Arquivo .py - Áreas

Expressões regulares complexas

- Definir uma função para cada **token** com ER complexa;

- A função será iniciada por t_ seguida do nome do token e irá receber como parâmetro o token sendo avaliado.

- O corpo da função define a ER complexa.

PLY – Formato do Arquivo .py - Áreas

Expressões regulares complexas

Ex.: considerando os tokens *TC3 e TC4 como tokens com ER complexas:*

```
def t_TC3(t):
    r‘regra_ER_token3’

def t_TC4(t):
    r‘regra_ER_token4’
```

Lex – Resolver Conflito de Regras

Escolher a regra que consegue casar a maior sequência de caracteres possível.

Quando mais de uma regra casar com a maior sequência de caracteres, escolher aquela que aparece primeiro na seção de regras.

PLY – Exemplo de Arquivo .py

```
import ply.lex as lex
tokens = ( NUMERO, ID, SOMA, SUB, MULT, DIV )
t_SOMA = r'\+'
t_SUB = r'-'
t_MULT = r'\*'
t_DIV = r'/'
def t_NUMERO(t):
        r'{[0-9]}+"."{DIGIT}* '
def t_ID(t):
        r'[a-z][a-z0-9]*'
lexer = lex.lex()
```