



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”

LINGUAGEM

NAJA

ANALISADOR LÉXICO

Douglas Canevarollo

Gabriel Andrey

1. Descrição

A linguagem NAJA foi desenvolvida tendo como base duas das principais linguagens do mercado: Python e JavaScript.

Abaixo estão descritas as características de sua confecção:

1.1. Alfabeto

O alfabeto Σ consiste de letras do alfabeto inglês, números e alguns caracteres especiais:

$$\Sigma = \{a, \dots, z, A, \dots, Z, 0, \dots, 9, -, +, *, /, (,), ., ` , ' , " , \& , | , \% , < , > , = , ! , ; , _\}$$

1.2. Expressões regulares

As expressões regulares que denotam o analisador léxico da linguagem Naja são as seguintes:

- $([a-zA-Z])([a-zA-Z]|\[\emptyset-9_])^*$
 - Responsável pela regra geral. Captura cadeias que representam nomes de variáveis ou comandos;
- $[\emptyset-9]^+$
 - Responsável pela regra numérica;
- $["==" \mid "!=" \mid "<=" \mid ">=" \mid "&&" \mid "||"]$
 - Responsável pela regra de comparações;
- $[";" \mid ", " \mid "{" \mid "}" \mid "=" \mid "'" \mid "`"]$
 - Responsável pela regra dos caracteres especiais;
- $["+ " \mid "- " \mid "/" \mid "*" \mid "(" \mid ")" \mid "." \mid "\%"]$
 - Responsável pela regra dos sinais de aritmética;
- $([\emptyset-1]^+)([a-zA-Z])^*$
 - Responsável por capturar cadeias que iniciem com números e gerar um erro léxico.

As demais cadeias que não forem capturas por essas expressões regulares acima terão um erro léxico lançado na sua leitura.

1.3. Estrutura principal

Um programa em Naja tem sempre, em sua função principal, uma estrutura semelhante a seguir:

```
void init() {  
    ...  
  
    return;  
}
```

2. Manual de uso

Devido a facilidade de instalação e configuração do Flex, sugerimos que o analisador léxico seja executado em um ambiente **Linux**.

No terminal, digite:

```
$ sudo apt-get install flex
```

Com isso, o Flex estará devidamente instalado e configurado na sua máquina.

Ainda pelo terminal, navegue até a pasta do analisador (arquivo compactado baixado) e rode os seguintes comandos:

```
$ lex analex.l  
$ gcc -o analex lex.yy.c -lfl
```

Com isso, um arquivo `analex.exe` será gerado. A execução deste junto a algum arquivo texto exibirá os tokens gerados e/ou as mensagens de erros durante a análise léxica.

Por exemplo, para executar a análise no arquivo texto de operações aritméticas, digite:

```
$ ./analex.exe < tests\aritmetica.txt
```

O mesmo pode ser feito para os arquivos textos `comparações.txt` e `erros.txt`. Este último consiste de algumas cadeias que lançam erros ao serem lidas.

Nota: certifique-se de que todos os comandos estão sendo executados dentro da pasta do analisador.