



Linguagens de Programação

Projeto

Celso Olivete Júnior

olivete@fct.unesp.br



Projeto de um analisador léxico

- ❑ Projetar um analisador léxico para uma calculadora simples com números naturais e reais e operações básicas (soma, subtração, multiplicação e divisão)



Projeto de um analisador léxico

- ❑ Questões a considerar:
 1. Que símbolo usar como separador de casa decimais?
 2. A calculadora usa representação monetária?
 3. A calculadora aceita espaços entre os operandos e operadores?
 4. O projetista é quem decide sobre as características desejáveis do compilador ou interpretador. Para a maioria das linguagens de programação existem algumas **convenções** que devem ser respeitadas



Projeto de um analisador léxico

□ **Exemplo** - seja a cadeia $3.2 + (2 * 12.01)$, o analisador léxico teria como saída:

3.2 => número real

+ => operador de soma

(=> abre parênteses

2 => número natural

* => operador de

multiplicação

12.01 => número real

) => fecha parênteses



Projeto de um analisador léxico

1. Definição do Alfabeto

$$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \dots, (,), +, -, *, /, \backslash b\}$$

- **OBS.**: projetista deve considerar TODOS os símbolos que são necessários para formar os padrões



Projeto de um analisador léxico

2. Listagem dos *tokens*

- OPSOMA: operador de soma
- OPSUB: operador de subtração
- OPMUL: operador de multiplicação
- OPDIV: operador de divisão
- AP: abre parênteses
- FP: fecha parênteses
- NUM: número natural/real

- **OBS.:** projetista deve considerar *tokens* especiais e cuidar para que cada *token* seja uma unidade significativa para o problema



Projeto de um analisador léxico

3. Especificação dos *tokens* com definições regulares

- OPSOMA $\rightarrow +$
- OPSUB $\rightarrow -$
- OPMUL $\rightarrow *$
- OPDIV $\rightarrow /$
- AP $\rightarrow ($
- FP $\rightarrow)$
- NUM $\rightarrow [0-9]^+?.[0-9]^*$

❑ OBS.: cuidar para que as definições regulares reconheçam padrões claros, bem formados e definidos



Projeto de um analisador léxico

4. Montar os autômatos para reconhecer cada **token**

- **OBS.:** os autômatos reconhecem **tokens** individuais, mas é o conjunto dos autômatos em um único autômato não-determinístico que determina o analisador léxico de um compilador, por isto, deve ser utilizada uma numeração crescente para os estados.



Projeto de um analisador léxico

- Implementação será a parte 2 do projeto