

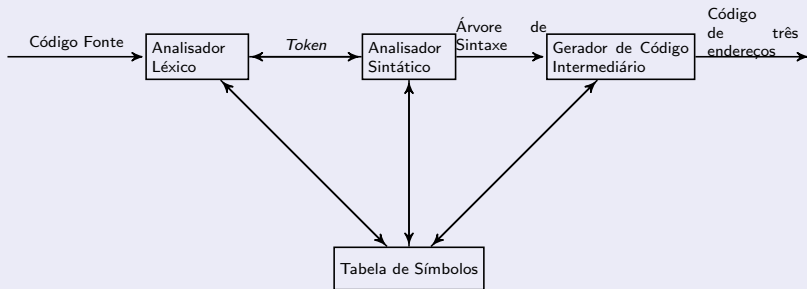
Compiladores

Análise Léxica

Bruno Lopes

- Lida com a linguagem de entrada
- Teste de pertinência: código fonte \in linguagem fonte?
- Programa está bem formado?
 - Sintaticamente?
 - Semanticamente?
- Cria um código intermediário

Front-end



Overview

- 1 Escrever expressão regular para representar a linguagem
- 2 Construir NFA para reconhecer a linguagem, a partir das ERs
- 3 Transformar NFA em DFA
- 4 Minimizar DFA
- 5 Construir o scanner a partir do DFA

Reconhecimento de *tokens*

- Como reconhecer de forma automática os *tokens* de uma linguagem?
- Como automatizar a construção de reconhecedores de ERs de forma eficiente?

Reconhecimento de *tokens*

- Como reconhecer de forma automática os *tokens* de uma linguagem?
- Como automatizar a construção de reconhecedores de ERs de forma eficiente?

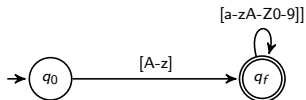
Autômatos Finitos

- Reconhecer identificadores
- Reconhecedor: DFA



Autômatos Finitos

- Reconhecer identificadores
- Reconhecedor: DFA



Números

```
<num> ::= <signedNat>.<nat> | <signedNat>.<nat>E<signedNat>  
<signedNat> ::= +<nat> | -<nat>  
<nat> ::= <digito> | <digito><nat>  
<digito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

- Expressão regular?
- DFA?

Números

```
<num> ::= <signedNat>.<nat> | <signedNat>.<nat>E<signedNat>  
<signedNat> ::= +<nat> | -<nat>  
<nat> ::= <digito> | <digito><nat>  
<digito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

- Expressão regular?
- DFA?

Números

```
<num> ::= <signedNat>.<nat> | <signedNat>.<nat>E<signedNat>  
<signedNat> ::= +<nat> | -<nat>  
<nat> ::= <digito> | <digito><nat>  
<digito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

- Expressão regular?
- DFA?

- Comentários?
- Símbolos?

- (i) Expressão Regular
- (ii) NFA
- (iii) DFA
- (iv) DFA Mínimo