

Construção de Compiladores

Capítulo 2

Análise Léxica

José Romildo Malaquias

Departamento de Computação
Universidade Federal de Ouro Preto

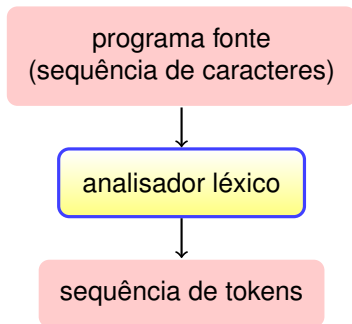
2014.1

1 Análise Léxica

1 Análise Léxica

- **Análise léxica** é o processo onde a sequência de caracteres que constitui o programa fonte é lida da esquerda para a direita e agrupada em **tokens**.
- **Token** é uma sequência de caracteres com um significado coletivo.
- Usualmente há apenas um pequeno número de tokens para uma linguagem de programação:
 - constantes (inteiros, ponto-flutuante, caracteres, strings, etc.)
 - operadores (aritméticos, relacionais, lógicos)
 - sinais de pontuação
 - palavras reservadas

- O **analizador léxico** recebe um programa fonte como entrada e produz uma sequência de tokens como saída.



Exemplo de análise léxica

```
while (cont > 0)
    cont = cont - 2;
```

análise
léxica

```
T_While
T_LParen
T_Identifier "cont"
T_Gt
T_Integer 0
T_RParen
T_Identifier "cont"
T_Eq
T_Identifier "cont"
T_Minus
T_Integer 2
T_Semicolon
```

- O analisador léxico pode reconhecer instâncias particulares de tokens, tais como
 - 3 ou 255 para um token constante inteira
 - "Fred" ou "Wilma" para um token constante string
 - numTicketes ou queue para um token nome de variável
- Instâncias específicas de um token são chamadas de **lexemas**.
- Um lexema é sequência de caracteres que forma o token.
- Um token é uma classe geral à qual o lexema pertence.
- Alguns tokens tem exatamente um lexema (exemplo: o caracter >)
- Outros tokens tem vários lexemas (como as constantes inteiras).

O analisador léxico:

- Deve **agrupar** os caracteres da entrada em tokens válidos.
- Tem uma **visão muito localizada** do programa fonte, sem qualquer contexto.
- Logo **não pode** verificar por exemplo se:
 - a sequência dos tokens está correta
 - um token ocorre em uma posição incorreta
 - um identificador foi declarado
 - uma palavra-chave foi escrita erradamente
- **Pode** relatar por exemplo:
 - caracter inválidp, a partir do qual não é possível formar nenhum token
 - ocorrências inválidas de caracteres em constantes strings
 - strings não terminadas
 - comentários não terminados

- Pode descartar os caracteres brancos e comentários que aparecem entre os tokens.
- Pode realizar substituição de macros, compilação condicional ou inclusão de arquivos, embora estas tarefas possam ser mais apropriadas a um **preprocessador** que filtra a entrada antes de passá-la ao compilador.

- Expressões regulares podem ser implementadas usando **autômatos finitos**.
- Há dois tipos de autômato finito:
 - **NFAs** (autômato finito **não-determinístico**), e
 - **DFA**s (autômato finito **determinístico**).

- Um **autômato finito não determinístico** (NFA) é uma quintupla

$$(S, \Sigma, \delta, s_0, F)$$

onde

- S é um conjunto finito de estados
 - Σ é um alfabeto, um conjunto finito de símbolos de entrada
 - δ é uma função de transição que dá para cada estado $s \in S$ e para cada símbolo a em $\Sigma \cup \{\epsilon\}$, um conjunto de estados seguintes $\delta(s, a) \in \mathcal{P}(S)$
 - $s_0 \in S$ é o estado inicial
 - $F \subseteq S$ é o conjunto de estados finais, que são distinguidos como estados de aceitação
- O símbolo ϵ representando a cadeia vazia não é um elemento do alfabeto.

Exemplo de autômato finito

Autômato que reconhece $\mathcal{L}((a|b) \star abb)$:

$$A = (\Sigma, S, \delta, s_0, F)$$

onde

$$\Sigma = \{a, b\}$$

alfabeto

$$S = \{0, 1, 2, 3\}$$

conjunto de estados

$$\delta = \{(0, a) \mapsto \{0, 1\},$$

função de transição

$$(0, b) \mapsto \{b\},$$

$$(0, \epsilon) \mapsto \{\},$$

$$(1, a) \mapsto \{\},$$

$$(1, b) \mapsto \{2\},$$

$$(1, \epsilon) \mapsto \{\},$$

$$(2, a) \mapsto \{\},$$

$$(2, b) \mapsto \{3\},$$

$$(2, \epsilon) \mapsto \{\} \quad \}$$

$$s_0 = 0$$

estado inicial

$$F = \{3\}$$

conjunto de estados finais

- Um autômato finito pode ser representado por um **grafo de transição** onde
 - os **nós** representam os estados
 - as **arestas rotuladas** representam a função de transição
 - o estado inicial é indicado por uma **aresta sem origem**
 - os estados finais são indicados por **nós com bordas duplas**

Tabela de transição

- Um autômato finito pode ser representado por uma **tabela de transição** onde
 - as **linhas** correspondem aos estados
 - as **colunas** correspondem aos símbolos de entrada e ϵ
 - a **entrada** da tabela para um determinado estado e um símbolo da entrada (ou ϵ) representa o valor da função de transição aplicada a esses argumentos
- **Vantagem:** as transições são facilmente encontradas.
- **Desvantagem:** ocupa muito espaço quando o alfabeto de entrada é grande, mesmo que muitos estados não tenham transições com a maioria dos símbolos da entrada (e ϵ).

Exemplo de autômato finito

Autômato que reconhece
 $\mathcal{L}((a|b) \star abb)$:

$$A = (\Sigma, S, \delta, s_0, F)$$

$$\Sigma = \{a, b\}$$

$$S = \{0, 1, 2, 3\}$$

$$\delta = \{ (0, a) \mapsto \{0, 1\}, \\ (0, b) \mapsto \{b\}, \\ (0, \epsilon) \mapsto \{\}, \\ (1, a) \mapsto \{\}, \\ (1, b) \mapsto \{2\}, \\ (1, \epsilon) \mapsto \{\}, \\ (2, a) \mapsto \{\}, \\ (2, b) \mapsto \{3\}, \\ (2, \epsilon) \mapsto \{\} \}$$

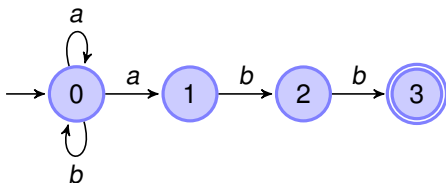
$$s_0 = 0$$

$$F = \{3\}$$

Tabela de transições:

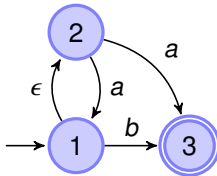
estado	a	b	ϵ
0	{0, 1}	{0}	{}
1	{}	{2}	{}
2	{}	{3}	{}
3	{}	{}	{}

Grafo de transições:



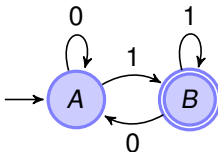
Exemplo de autômato finito

Autômato que reconhece $\mathcal{L}(a * (a|b))$:



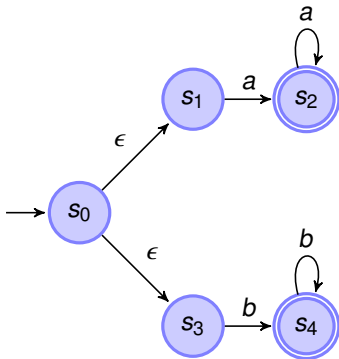
Exemplo de autômato finito

Autômato que reconhece $\mathcal{L}((0|1)^* 1)$:



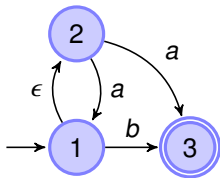
Exemplo de autômato finito

Autômato que reconhece $\mathcal{L}(\textcolor{red}{aa} * | \textcolor{red}{bb} *)$:



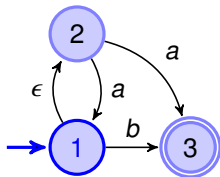
Aceitação de cadeias pelo autômato finito

- Um autômato finito **aceita** uma cadeia de entrada x se e somente se houver algum caminho no grafo de transição do estado inicial para um dos estados finais, de modo que os símbolos ao longo do caminho componham x .
- Exemplo: o autômato a seguir aceita a cadeia $aaab$:



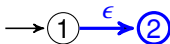
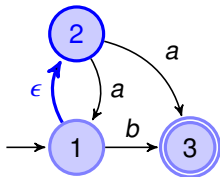
Aceitação de cadeias pelo autômato finito

- Um autômato finito **aceita** uma cadeia de entrada x se e somente se houver algum caminho no grafo de transição do estado inicial para um dos estados finais, de modo que os símbolos ao longo do caminho componham x .
- Exemplo: o autômato a seguir aceita a cadeia $aaab$:



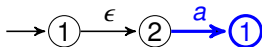
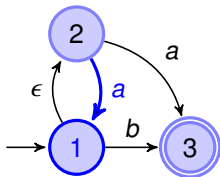
Aceitação de cadeias pelo autômato finito

- Um autômato finito **aceita** uma cadeia de entrada x se e somente se houver algum caminho no grafo de transição do estado inicial para um dos estados finais, de modo que os símbolos ao longo do caminho componham x .
- Exemplo: o autômato a seguir aceita a cadeia $aaab$:



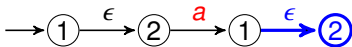
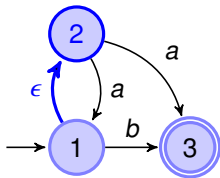
Aceitação de cadeias pelo autômato finito

- Um autômato finito **aceita** uma cadeia de entrada x se e somente se houver algum caminho no grafo de transição do estado inicial para um dos estados finais, de modo que os símbolos ao longo do caminho componham x .
- Exemplo: o autômato a seguir aceita a cadeia $aaab$:



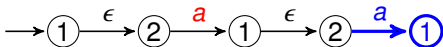
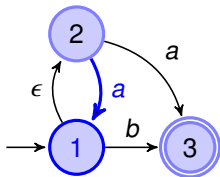
Aceitação de cadeias pelo autômato finito

- Um autômato finito **aceita** uma cadeia de entrada x se e somente se houver algum caminho no grafo de transição do estado inicial para um dos estados finais, de modo que os símbolos ao longo do caminho componham x .
- Exemplo: o autômato a seguir aceita a cadeia $aaab$:



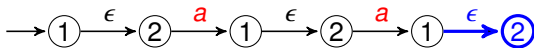
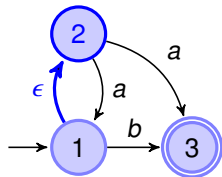
Aceitação de cadeias pelo autômato finito

- Um autômato finito **aceita** uma cadeia de entrada x se e somente se houver algum caminho no grafo de transição do estado inicial para um dos estados finais, de modo que os símbolos ao longo do caminho componham x .
- Exemplo: o autômato a seguir aceita a cadeia $aaab$:



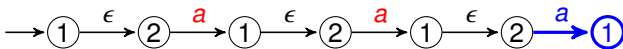
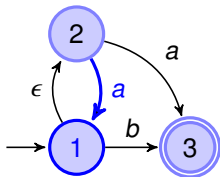
Aceitação de cadeias pelo autômato finito

- Um autômato finito **aceita** uma cadeia de entrada x se e somente se houver algum caminho no grafo de transição do estado inicial para um dos estados finais, de modo que os símbolos ao longo do caminho componham x .
- Exemplo: o autômato a seguir aceita a cadeia $aaab$:



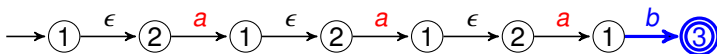
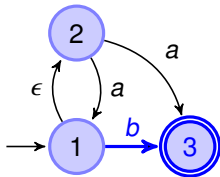
Aceitação de cadeias pelo autômato finito

- Um autômato finito **aceita** uma cadeia de entrada x se e somente se houver algum caminho no grafo de transição do estado inicial para um dos estados finais, de modo que os símbolos ao longo do caminho componham x .
- Exemplo: o autômato a seguir aceita a cadeia $aaab$:



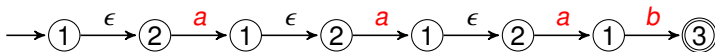
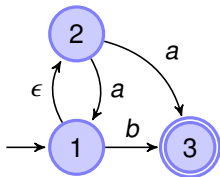
Aceitação de cadeias pelo autômato finito

- Um autômato finito **aceita** uma cadeia de entrada x se e somente se houver algum caminho no grafo de transição do estado inicial para um dos estados finais, de modo que os símbolos ao longo do caminho componham x .
- Exemplo: o autômato a seguir aceita a cadeia $aaab$:



Aceitação de cadeias pelo autômato finito

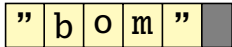
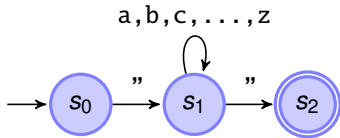
- Um autômato finito **aceita** uma cadeia de entrada x se e somente se houver algum caminho no grafo de transição do estado inicial para um dos estados finais, de modo que os símbolos ao longo do caminho compõem x .
- Exemplo: o autômato a seguir aceita a cadeia $aaab$:



- A **linguagem** $L(A)$ **definida** (ou **aceita**) pelo autômato finito A é o conjunto de todas as cadeias rotulando algum caminho a partir do estado inicial até um estado final de A .

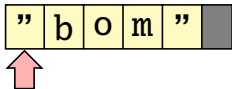
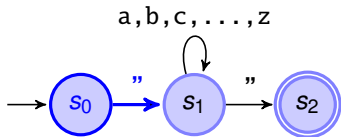
- Mantenha um conjunto de estados atuais, inicialmente dado pelo estado inicial e todos os estados alcançados por transições vazias.
- Para cada caracter da entrada:
 - Mantenha um conjunto dos próximos estados, inicialmente vazio.
 - Para cada estado atual:
 - Siga todas as transições rotuladas com o símbolo atual.
 - Adicione estes estados ao conjunto de novos estados.
 - Adicione todo estado alcançado por transições vazias ao conjunto dos próximos estados.
- Complexidade: $O(mn^2)$ para cadeias de comprimento m e autômatos com n estados.

Exemplo

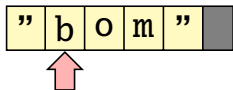
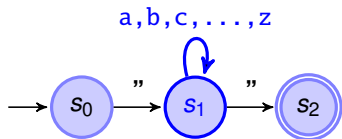


O autômato recebe uma cadeia como entrada e decide aceitá-la ou rejeitá-la

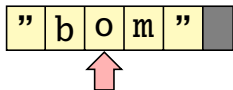
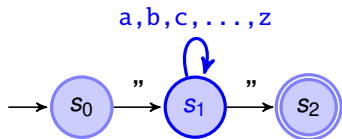
Exemplo



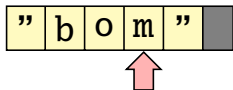
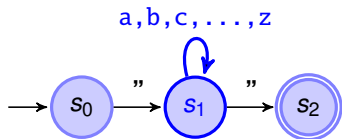
Exemplo



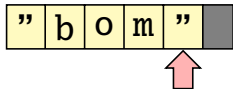
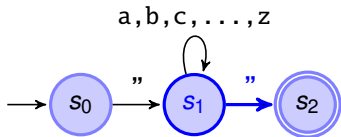
Exemplo



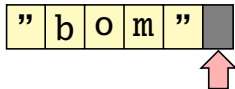
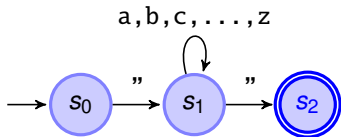
Exemplo



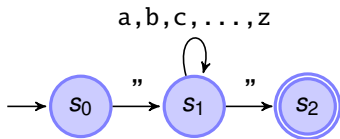
Exemplo



Exemplo



Exemplo



"	b	o	m	"	
---	---	---	---	---	--

 aceita

Fim