

Compiladores

Análise Léxica

O que é?

- A tarefa do analisador léxico é quebrar a entrada em símbolos que façam sentido para a definição da linguagem e para o analisador sintático.
- Para identificarmos quais são esses símbolos, basta olharmos na definição da linguagem – sua gramática ou grafo sintático – e identificarmos quais são os seus símbolos terminais.
- No contexto da análise léxica, costumamos chamar esses símbolos de **tokens**.

Linguagens e Alfabeto

- Os símbolos terminais compõem o alfabeto sobre o qual a nossa linguagem é definida.
- Esse conjunto de símbolos pode ser definido como uma linguagem.
- Muito mais simples que a linguagem alvo.
- Modelos simples como AFD e ER.

Autômato Finito Determinístico

Um autômato finito determinístico (AFD) é um modelo para definição de linguagens regulares composto de cinco elementos $\langle \Sigma, S, s_0, \delta, F \rangle$, onde:

Σ é o alfabeto sobre o qual a linguagem é definida;

S é um conjunto finito não vazio de estados;

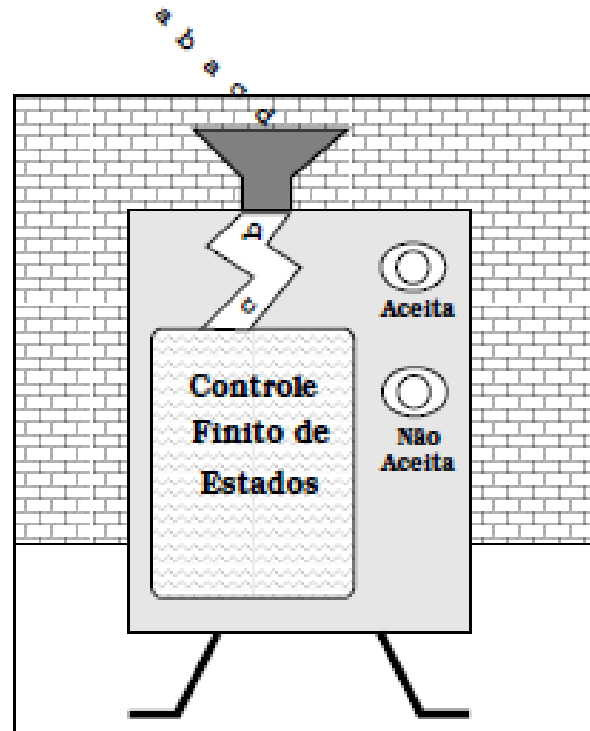
s_0 é o estado inicial, $s_0 \in S$;

δ é a função de transição de estados, $\delta : S \times \Sigma \rightarrow S$;

F é o conjunto de estados finais, $F \subseteq S$.

Máquina de Reconhecer Caracteres

Recebe como entrada uma cadeia e diz se ela pertence ou não à linguagem desejada.



Representações

δ	0	1	2	3	4	5	6	7	8	9
s_0	s_0	s_1	s_2	s_0	s_1	s_2	s_0	s_1	s_2	s_0
s_1	s_1	s_2	s_0	s_1	s_2	s_0	s_1	s_2	s_0	s_1
s_2	s_2	s_0	s_1	s_2	s_0	s_1	s_2	s_0	s_1	s_2

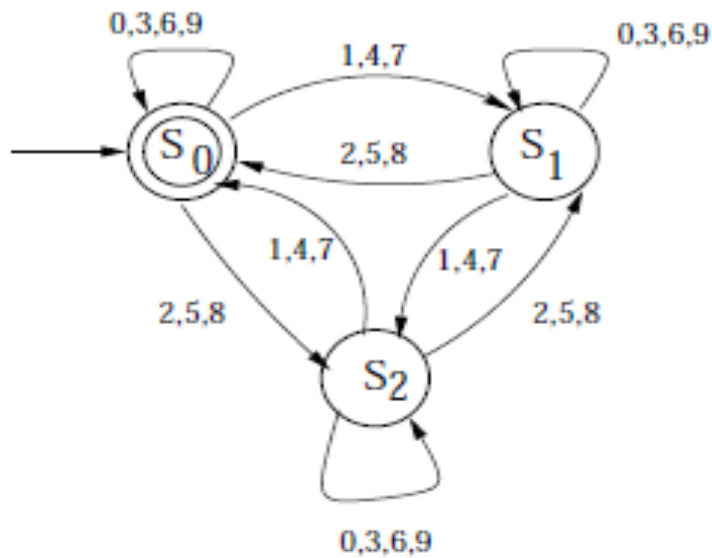
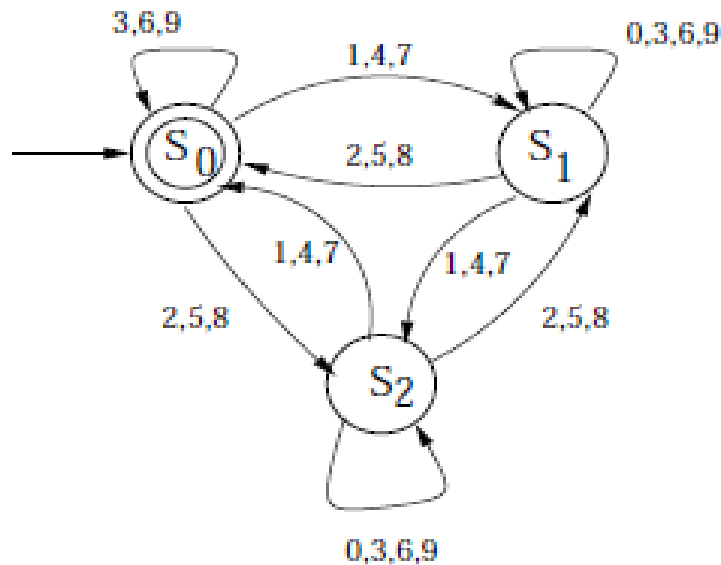


Tabela de Transições

- Uma das vantagens de se utilizar um AFD para definir quais são os tokens a serem reconhecidos pelo AL é que é fácil implementar um AL baseado na tabela de transição de estados.
- Tal analisador deve apenas ler uma letra da entrada e, baseado na tabela, fazer a mudança de estado.
- Se o estado em que o AFD se encontrar for um estado final, então a cadeia lida até aquele ponto é um token válido.

AFD com função parcial

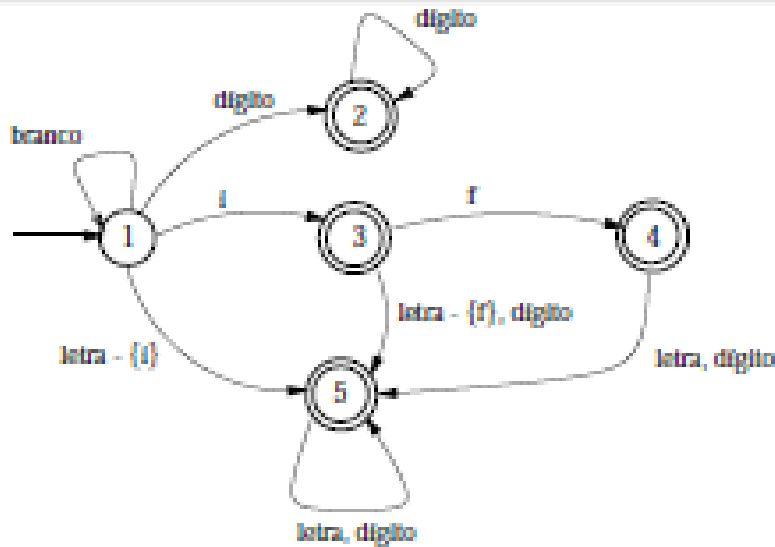
δ	0	1	2	3	4	5	6	7	8	9
s_0	---	s_1	s_2	s_0	s_1	s_2	s_0	s_1	s_2	s_0
s_1	s_1	s_2	s_0	s_1	s_2	s_0	s_1	s_2	s_0	s_1
s_2	s_2	s_0	s_1	s_2	s_0	s_1	s_2	s_0	s_1	s_2



Execução do AL

- Cadeia longa de símbolos.
- A cada execução do AFD (analisador léxico), um símbolo é reconhecido.
- O AFD é executado até que não haja mais nenhuma transição possível
- Se o último estado for final, o símbolo é válido.
- Se não, um erro ocorreu.

Tipo de Símbolo



- Associar a cada nó um tipo de token.
- Se execução termina no estado 2, um número foi reconhecido.
- Se termina nos estados 3 ou 5, um identificador foi reconhecido.
- Se termina no estado 4, um *if* foi reconhecido

Tabela de Transição

Estado	Letra: próximo estado				Tipo
1	branco:1	digito:2	i:3	letra-{i}:5	ERRO
2	digito:2				NÚMERO
3	f:4	letra:5	digito:5		IDENT
4	letra:5	digito:5			IF
5	letra:5	digito:5			IDENT

Complexidade do AFD

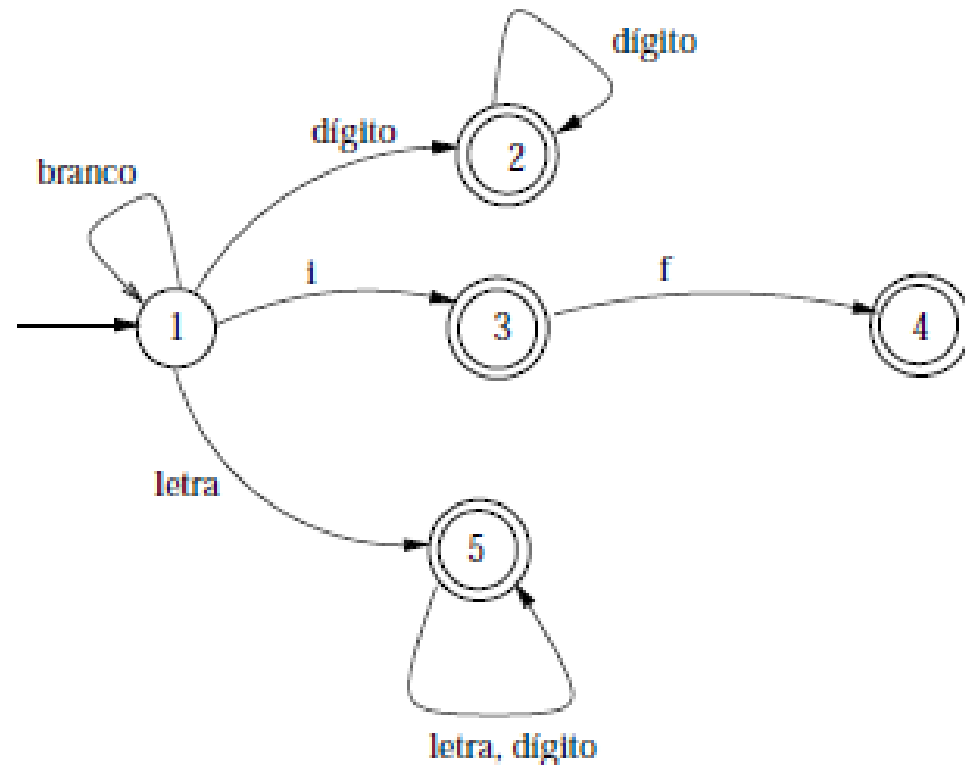
- Na linguagem X++, que é bastante simples, temos nada menos que 39 símbolos terminais.
- A construção do AFD pode se tornar bastante complexa.
- Uma das maneiras de diminuir essa dificuldade é utilizando um autômato finito não determinístico (AFND).

Autômato Finito não-determinístico

A função de transição de estados é definida $\delta : S \times \Sigma \rightarrow$
conjunto de subconjuntos finitos de $\rho(S)$

Pode existir mais de uma transição saindo de um estado
para a mesma letra do alfabeto.

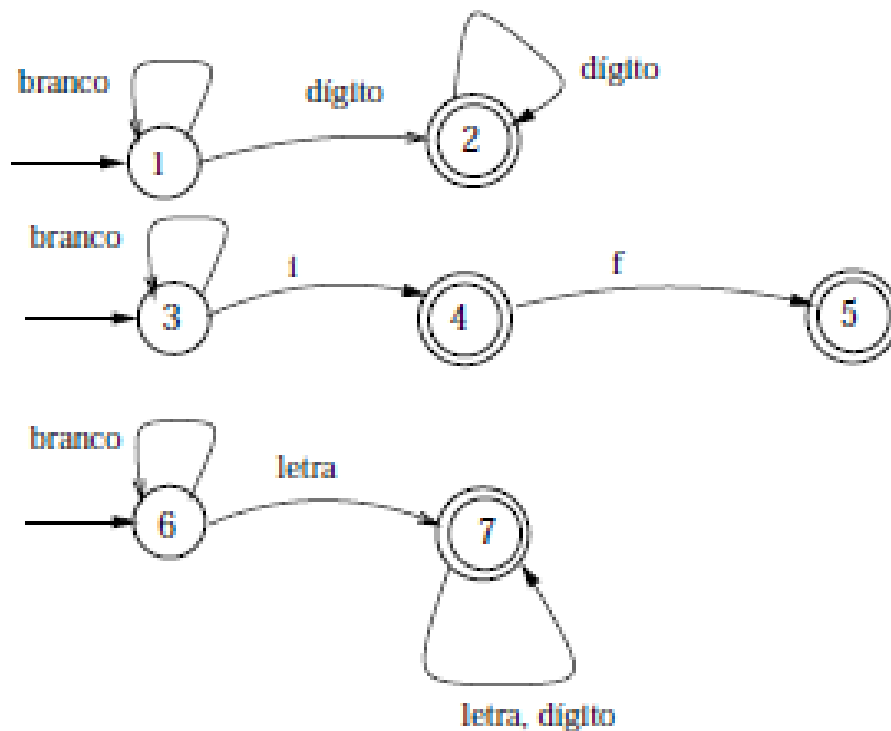
Exemplo de AFND



if12 $\{1\} \xrightarrow{i} \{3, 5\} \xrightarrow{f} \{4, 5\} \xrightarrow{1} \{5\} \xrightarrow{2} \{5\}$

if $\{1\} \xrightarrow{i} \{3, 5\} \xrightarrow{f} \{4, 5\}$

AFND com vários estados iniciais



if12 $\{1, 3, 6\} \xrightarrow{i} \{4, 7\} \xrightarrow{f} \{5, 7\} \xrightarrow{1} \{7\} \xrightarrow{2} \{7\}$

if $\{1, 3, 6\} \xrightarrow{i} \{4, 7\} \xrightarrow{f} \{5, 7\}$

Observações

- AFND facilita a descrição da linguagem reconhecida pelo analisador léxico.
- Por outro lado é difícil de implementar.
- Não adiciona nenhum poder extra aos AFDs
- Portanto podemos sempre transformar um AFND em um AFD.

Expressão Regular

- Cada item léxico pode ser definido através de uma ER.
- NUMERO: $\text{digito} \mid \text{digito}^*$;
- IF: $i \mid f$;
- IDENTIFICADOR: $\text{letra} \mid (\text{letra} \cup \text{digito})^*$.
- Uma (ou diversas) ER pode ser transformada em um AFND.

Exercício

- Identifique cada símbolo terminal na gramática de X++ e construa, para cada um, uma ER e um AFND.