Linguagem Regular e Autômatos Finitos

Linguagens Regulares

- Classe das linguagens mais simples.
- Reconhecimento de cadeias com pouca complexidade.
- As palavras de uma linguagem regular é reconhecida por um AF.
- Formalismos:
  - AF: reconhecedor
  - Expressão Regular: geradora

As linguagens regulares pertencem à classe das linguagens mais simples. As cadeias (palavras) da linguagem regular podem ser reconhecidas através de algoritmos com pouco complexidade. A expressão regular é considerada geradora pois pode-se inferir como construir ("gerar") as palavras da linguagem.

Expressão Regular

Uma ER sobre Σ pode ser definida como:

- $\emptyset$ é uma ER e denota a linguagem vazia
- ε é uma ER e denota a linguagem $\{\varepsilon\}$
- $\mathtt{x} \in \Sigma$ é uma ER e denota a linguagem $\{\mathtt{x}\}$
- Se `r` e `s` são ER e denotam as linguagens R e S
  - `(r+s)` é ER $\Rightarrow R \cup S$
  - `(rs)` é ER $\Rightarrow RS = \{uv \mid u \in Rev \in S\}$
  - $(\mathtt{r}^*)$ é ER $\Rightarrow R^*$ – `(r|s)` é ER $\Rightarrow R$ ou S

Qualquer símbolo $\mathtt{x} \in \Sigma$ ou ε é uma ER e denota a linguagem contendo exclusivamente a palavra x ou a palavra vazia respectivamente. Também podemos definir as ER através de união e concatenação como visto no slide. A união representa de duas linguagens R e S é o conjunto de cadeias que estão em R ou em S ou em ambas. O fechamento (ou estrela ou fechamento de Kleene) de uma linguagem é denotado L^* e representa o conjunto de cadeias que podem ser formados tomando-se qualquer número de cadeias de L.

Expressões Regulares (ER)

- Toda linguagem regular pode ser descrita por uma expressão regular.
- Definida através de:
  - conjuntos básicos
  - concatenação
  - união

Uma expressão regular é definida a partir de conjuntos (linguagens) básicos e operações de concatenação e união e oferece um modo declarativo de expressar as cadeias que queremos aceitar. As expressões regulares são consideradas adequadas para a comunicação homem × homem e, principalmente, para a comunicação homem × máquina.

Convenções

- concatenação sucessiva tem precedência sobre a concatenação e a união
  `abc+a` $\Rightarrow$ `(((ab)c)+(a))`
- a concatenação tem precedência sobre a união.
  `ab+a` $\Rightarrow$ `((ab)+(a))`

A omissão de parênteses em uma ER é usual, respeitando as convenções citadas no slide acima.

Exemplos de ER

ER	Linguagem Representada
aa	somente a palavra aa
ba^*	iniciam com b seguido por 0 ou mais a
$(a+b)^*$	todas as palavras sobre { a, b }

Qual a linguagem correspondente as ER abaixo?

1. $(a+b)^*aa(a+b)^*$
2. $a^*ba^*ba^*$
3. $(a+b)^*(aa+bb)$
4. $(a+\varepsilon)(b+ba)^*$

As linguagem representadas pelas ER citadas no slide acima são:

1. todas as palavras contendo aa como subpalavra
2. todas as palavras contendo exatamente dois b
3. todas as palavras que terminam com aa ou bb
4. todas as palavras que não possuem dois a consecutivos

Sistema de Estados Finitos

- Modelo matemático com entradas e saídas.
- Possui um número finito pré-definido de estados.
- Cada estado resume somente as informações do passado necessárias para determinar as ações para a próxima entrada.
- Exemplos: elevador, analisadores léxicos, processadores de texto.

Nesta disciplina, devemos definir modelos de computação progressivamente poderosos e dispositivos cada vez mais sofisticados para aceitar e gerar linguagens.

Um sistema de estados finitos é um modelo matemático de um sistema com entradas e saídas. Um exemplo clássico é um elevador. Trata-se de um sistema que não memoriza instruções. Cada estado sumariza as informações: andar corrente e direção do movimento. As entradas para o sistema são requisições pendentes.

Autômatos Finitos

- recebe como entrada uma string
- não produz saída
- indica se a entrada é aceita ou não
- memória fixa e limitada
- também denominado Autômato Finito Determinístico (AFD)

O autômato finito compartilha com um computador real o fato de que ele tem uma "unidade central de processamento" de capacidade fixa, finita. Recebe sua entrada como uma string e não produz nenhuma saída, exceto uma indicação informando se a entrada foi ou não considerada aceitável.

Autômato Finito

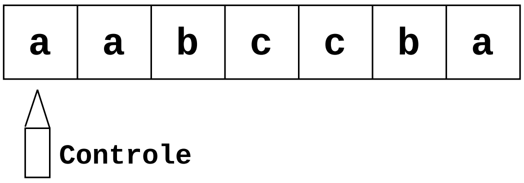
Máquina composta por três partes:

1. Fita – dispositivo de entrada que contém a informação a ser processada
2. Unidade de controle – reflete o estado corrente da máquina. Possui uma unidade de leitura (cabeça da fita) acessando uma célula de cada vez e movimentando-se sempre para a direita
3. Programa ou Função de Transição – comanda as leituras e define o estado da máquina

A fita é finita à esquerda e à direita, sendo dividida em células, onde cada uma armazena um símbolo. Os símbolos pertencem a um alfabeto de entrada. Não é possível gravar sobre a fita e não existe uma memória auxiliar. Inicialmente, a palavra a ser processada ocupa toda a fita.

Autômato Finito

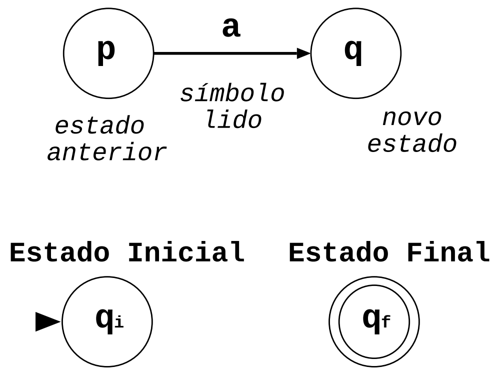
- início → cabeça na célula mais à esquerda
- após leitura → movimento p/ direita
- estado corrente e símbolo lido determinam o novo estado



A unidade de controle possui um número finito e pré-definido de estados. A unidade de leitura lê o símbolo de uma célula a cada vez. Após a leitura, a cabeça da fita move-se uma célula para a direita. Inicialmente a cabeça da fita está posicionada na célula mais a esquerda.

O programa é uma função que dependendo do estado corrente e do símbolo lido, determina o novo estado do autômato.

AFD



Um autômato finito pode ser representado através de um grafo (diagrama de estados) como mostra o slide acima. Note na representação dos estados iniciais e finais.

AFD - Definição

- $M = (\Sigma, Q, \delta, q_0, F)$
  - Σ – alfabeto de símbolos de entrada
  - Q – conj. de estados possíveis do autômato o qual é finito
  - δ – função de transição ($\delta : Q \times \Sigma \rightarrow Q$)
  - q_0 – estado inicial ($q_0 \in Q$)
  - F – conjunto dos estados finais tal que F está contido em Q

Um autômato finito determinístico (AFD), ou simplesmente autômato finito M, é determinado pela quíntupla:

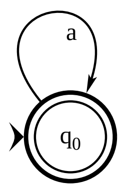
$$M = (\Sigma, Q, \delta, q_0, F)$$

Exemplo 1

$L = \{w \mid w$ possui $\mathtt{a}^*$ como subpalavra$\}$

$M = (\Sigma, Q, \delta, q_0, F)$
$\Sigma = \{\mathtt{a}\}$
$Q = \{q_0\}$
$F = \{q_0\}$

δ	a
q_0	q_0

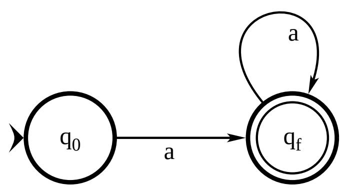


Exemplo 2

$L = \{w \mid w$ possui $\mathtt{a}^+$ como subpalavra$\}$

$M = (\Sigma, Q, \delta, q_0, F)$
$\Sigma = \{\mathtt{a}\}$
$Q = \{q_0, q_f\}$
$F = \{q_f\}$

δ	a
q_0	q_f
q_f	q_f



Exercício 1

$L = (\mathtt{a} \cup \mathtt{b})^*$

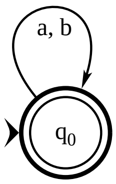
$M = (\Sigma, Q, \delta, q_0, F)$

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

$Q = \{q_0\}$

$F = \{q_0\}$

δ	a	b
q_0	q_0	q_0



Exercício 2

$L = (\mathtt{a} \cup \mathtt{b})^+$

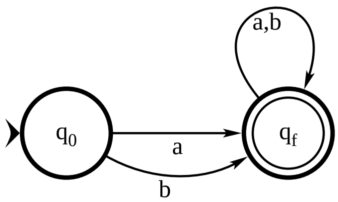
$M = (\Sigma, Q, \delta, q_0, F)$

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

$Q = \{q_0, q_f\}$

$F = \{q_f\}$

δ	a	b
q_0	q_f	q_f
q_f	q_f	q_f



Exercício 3

$L = (\mathtt{a.b})^*$

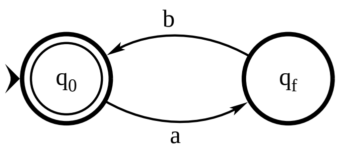
$M = (\Sigma, Q, \delta, q_0, F)$

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

$Q = \{q_0, q_f\}$

$F = \{q_0\}$

δ	a	b
q_0	q_f	-
q_f	-	q_0



Exercício 4

$L = (\mathtt{a.b})^+$

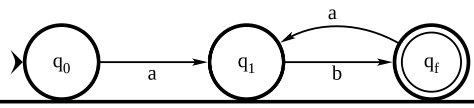
$M = (\Sigma, Q, \delta, q_0, F)$

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

$Q = \{q_0, q_1, q_f\}$

$F = \{q_0\}$

δ	a	b
q_0	q_1	-
q_1	-	q_f
q_f	q_1	-



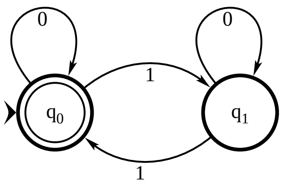
Condições de Parada em AFD

- após processar o último símbolo da fita, assume um estado final: o autômato pára e a entrada é aceita.
- após processar o último símbolo da fita, assume um estado não final: o autômato pára e a entrada é rejeitada
- a função programa é indefinida para o argumento (estado corrente e símbolo lido): o autômato pára e a entrada é rejeitada

Um autômato sempre pára após processar qualquer entrada, pois como toda palavra é finita e como um novo símbolo de entrada é lido a cada aplicação da função programa, não existe a possibilidade de ciclo (loop) infinito. A parada de processamento pode ocorrer de duas maneiras: aceitando ou rejeitando uma entrada `w`

Verificação de Validade da cadeia

Dado o autômato finito `M` abaixo, verifique se $01001011 \in$ L(M)



$q_0 \xrightarrow{0} q_0 \xrightarrow{1} q_1 \xrightarrow{0} q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_0 \xrightarrow{0}$
$q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_0$

A linguagem aceita por um autômato finito $M = (\Sigma, Q, \delta, q_0, F)$ denotada por ACEITA(M), ou L(M) é o conjunto de todas as palavras pertencentes a Σ^* que são aceitas por M.

Analogamente, REJEITA(M) é o conjunto de todas as palavras pertencentes a Σ^* que são rejeitadas por M.

- a intersecção dos conjunto ACEITA(M) e REJEITA(M) é vazio
- a união dos conjuntos ACEITA(M) e REJEITA(M) é Σ^*

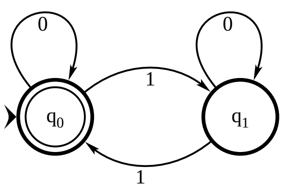
Exercício

Dado o diagrama de estado do AFD, monte-o e verifique se aceita a cadeia aabba.

$M = (\Sigma, Q, \delta, q_0, F)$
$\Sigma = \{a, b\}$
$Q = \{q_0, q_1\}$
$F = \{q_0\}$

δ	a	b
q_0	q_0	q_1
q_1	q_1	q_0

Solução:



$q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{b} q_0 \xrightarrow{a} q_0$

A cadeia aabba é aceita pois é esgotada no estado final do autômato.

Exercícios

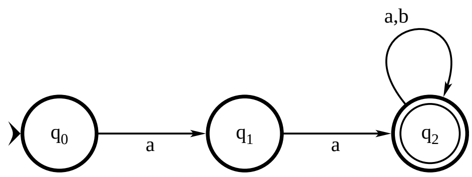
Desenvolver AFDs que reconheçam as seguintes linguagens sobre $\Sigma = \{a, b\}$

1. $L_1 = \{w \mid$ o prefixo de w é aa$\}$
2. $L_2 = \{w \mid w$ possui aa ou bb como subpalavra$\}$
3. $L_3 = \{w \mid w$ possui um número par de a e b$\}$
4. $L_4 = \{w|w$ possui um número ímpar de a$\}$

Construa um autômato finito determinístico dada as linguagens acima. Defina o formalismo do autômato, sua tabela de transição e verifique se uma cadeia que pertença a linguagem é reconhecida pelo autômato. Em todas as linguagens o alfabeto é {a,b}.

1. $M_1 = (\Sigma, Q, \delta, q_0, F)$
   $\Sigma = \{a, b\}$
   $Q = \{q_0, q_1, q_2\}$
   $F = \{q_2\}$

δ	a	b
q_0	q_1	-
q_1	q_2	-
q_2	q_2	q_2

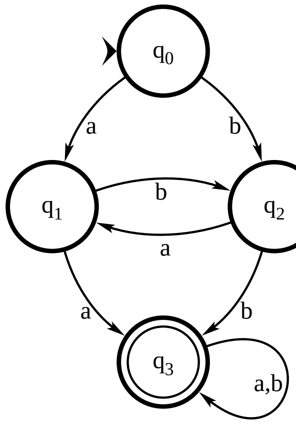


Cadeia: aabba $\rightarrow$ aceita:

$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{b} q_2 \xrightarrow{b} q_2 \xrightarrow{a} q_2$

2. $M_2 = (\Sigma, Q, \delta, q_0, F)$
   $\Sigma = \{a, b\}$
   $Q = \{q_0, q_1, q_2, q_3\}$
   $F = \{q_3\}$

δ	a	b
q_0	q_1	q_2
q_1	q_3	q_2
q_2	q_1	q_3
q_3	q_3	q_3

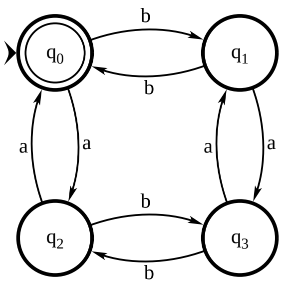


Cadeia: `babaab` $\rightarrow$ aceita:

$q_0 \xrightarrow{b} q_2 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_1 \xrightarrow{a} q_3 \xrightarrow{b} q_3$

3. $M_3 = (\Sigma, Q, \delta, q_0, F)$
   $\Sigma = \{a, b\}$
   $Q = \{q_0, q_1, q_2, q_3\}$
   $F = \{q_0\}$

δ	a	b
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

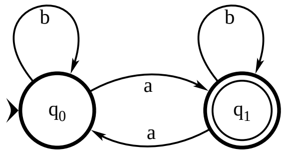


Cadeia: `ababaa` $\rightarrow$ aceita:

$q_0 \xrightarrow{a} q_2 \xrightarrow{b} q_3 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{a} q_2 \xrightarrow{a} q_0$

4. $M_4 = (\Sigma, Q, \delta, q_0, F)$
   $\Sigma = \{a, b\}$
   $Q = \{q_0, q_1\}$
   $F = \{q_1\}$

δ	a	b
q_0	q_1	q_0
q_1	q_0	q_1



Cadeia: `abbabab` $\rightarrow$ aceita:

$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_1 \xrightarrow{b} q_1 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_1$

Exercícios

δ	a	b
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_0	q_3
q_3	q_3	q_3

δ	a	b
q_0	q_2	q_1
q_1	q_1	q_0
q_2	q_4	q_5
q_3	q_5	q_4
q_4	q_3	q_2
q_5	q_2	q_3

Construa um AFD a partir das tabelas de transição acima. Defina o formalismo do AFD e verifique se uma cadeia é reconhecida.