Teoria da Computação

AFN$_\mathcal{E}$ e Expressões Regulares

Aula 03

Prof. Felipe A. Louza

UFU

Roteiro

Roteiro

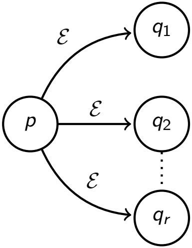
1. Autômatos finitos com Movimentos Vazios (AFN$_{\mathcal{E}}$s)
   - Movimento Vazio
   - Formalização de um AFN$_{\mathcal{E}}$
   - Equivalência entre AFN e AFN$_{\mathcal{E}}$
2. Expressões Regulares (ERs)
   - Formalização de uma ER
   - Exemplos de ERs
   - Equivalência entre ER e AFN$_{\mathcal{E}}$
3. Referências

Movimento vazio

Podemos estender o modelo de AFN para incluir transições vazias $\mathcal{E}$

Função de transição (mapeamento)

- $\delta(p, \mathcal{E}) = \{q_1, q_2, \ldots q_r\}$



AFN$_{\mathcal{E}}$

	$\mathcal{E}$
p	$\{q_1, q_2, \ldots, q_r\}$

Tabela de transições

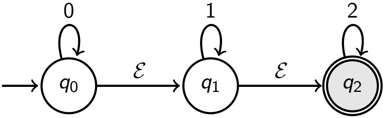
O AFN$_{\mathcal{E}}$ pode ir de $p \rightarrow q_i$ sem consumir w_i (transição espontânea).

Movimento vazio

Funcionamento:

- Se um estado com arcos-$\mathcal{E}$ for encontrado, de forma semelhante aos AFNs, a máquina *"se divide"* em múltiplas cópias (origem e destinos).
- Transição sem leitura de símbolo algum de w

Exemplo:



$$L(N_4) = \{w \mid w \in \{0,1,2\}^* \text{ e } w = 0^*1^*2^*\}$$

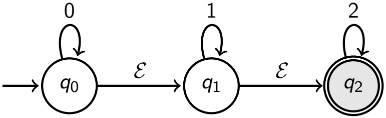
- Ao processar $w = 002$, temos q_0, q_0, q_1, q_2, q_2

Movimento vazio

Funcionamento:

- Se um estado com arcos-$\mathcal{E}$ for encontrado, de forma semelhante aos AFNs, a máquina *"se divide"* em múltiplas cópias (origem e destinos).
- Transição sem leitura de símbolo algum de w

Exemplo:



$$L(N_4) = \{w \mid w \in \{0,1,2\}^* \text{ e } w = 0^*1^*2^*\}$$

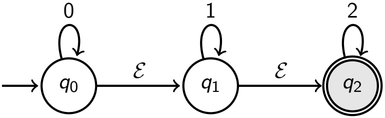
- Ao processar $w = 002$, temos q_0, q_0, q_1, q_2, q_2

Movimento vazio

Funcionamento:

- Se um estado com arcos-$\mathcal{E}$ for encontrado, de forma semelhante aos AFNs, a máquina *"se divide"* em múltiplas cópias (origem e destinos).
- Transição sem leitura de símbolo algum de w

Exemplo:



$$L(N_4) = \{w \mid w \in \{0,1,2\}^* \text{ e } w = 0^*1^*2^*\}$$

- Ao processar $w = 002$, temos q_0, q_0, q_1, q_2, q_2

Roteiro

1. Autômatos finitos com Movimentos Vazios (AFN$_{\mathcal{E}}$s)
   - Movimento Vazio
   - Formalização de um AFN$_{\mathcal{E}}$
   - Equivalência entre AFN e AFN$_{\mathcal{E}}$
2. Expressões Regulares (ERs)
   - Formalização de uma ER
   - Exemplos de ERs
   - Equivalência entre ER e AFN$_{\mathcal{E}}$
3. Referências

Definição formal de um AFN$_\mathcal{E}$

Definição

Um autômato finito não-determinístico com arcos-$\mathcal{E}$ (AFN$_\mathcal{E}$) é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$, em que

1. Q, Σ, q_0 e F como definimos anteriormente;
2. $\delta : Q \times (\Sigma \cup \{\mathcal{E}\}) \rightarrow 2^Q$ é a função de transição
   - $\delta(p, a) = \{q_1, q_2, \ldots, q_r\}$, com $a \in \Sigma \cup \{\mathcal{E}\}$

Relembrando: 2^Q é o conjunto das partes de Q (o conjunto de todos os subconjuntos).

Função de transição estendida

Vamos estender a função de transição para AFN$_{\mathcal{E}}$:

$$\hat{\delta}^* : Q \times \Sigma^* \to 2^Q$$

de forma que

$$\hat{\delta}^*(q, w) = \{q_i, q_j, \ldots, q_r\}$$

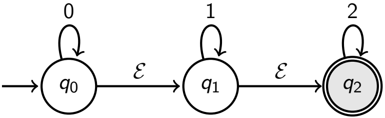
serão todos os estados alcançáveis por q ao processar w, possivelmente incluindo arcos-$\mathcal{E}$ (movimentos vazios).

$FECHO_{\mathcal{E}}(q)$

Antes, vamos definir a computação vazia:

- $FECHO_{\mathcal{E}}(q)$: o conjunto de estados atingíveis a partir de q somente com movimentos vazios (incluindo q).

No exemplo anterior:



$$FECHO_{\mathcal{E}}(q_0) = \{q_0, q_1, q_2\}$$

$$FECHO_{\mathcal{E}}(q_1) = \{q_1, q_2\}$$

$$FECHO_{\mathcal{E}}(q_2) = \{q_2\}$$

$FECHO_{\mathcal{E}}(q)$

Definição

A computação vazia ou $FECHO_{\mathcal{E}}(q)$ denota os estados atingíveis a partir de q somente ao longo de movimentos vazios (incluindo q):

$$FECHO_{\mathcal{E}}(q) = \begin{cases} \{q\} & \text{se } \delta(q,\mathcal{E}) = \varnothing \\ \{q\} \cup \delta(q,\mathcal{E}) \cup \left(\bigcup_{p \,\in\, \delta(q,\mathcal{E})} FECHO_{\mathcal{E}}(q) \right) & \text{caso contrário} \end{cases}$$

Além disso, definimos a computação vazia para conjuntos de estados:

$$FECHO_{\mathcal{E}}(P) = \bigcup_{q \,\in\, P} FECHO_{\mathcal{E}}(q)$$

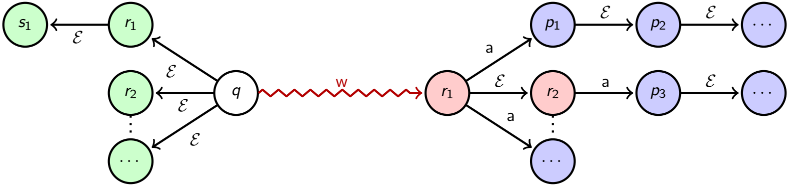
Definição

A Função de transição estendida (função programa) é denotada por

$$\hat{\delta}^* : Q \times \Sigma^* \to 2^Q, \text{ tal que:}$$

1. $\hat{\delta}^*(q, \mathcal{E}) = FECHO_{\mathcal{E}}(q)$
2. $\hat{\delta}^*(q, wa) = FECHO_{\mathcal{E}}(P)$, onde

$$P = \{p \mid \text{para algum } r \in \hat{\delta}^*(q, w), p \in FECHO_{\mathcal{E}}(\delta(r, a))\}$$



Ou seja, $\hat{\delta}^*(q, wa)$ inclui todos os estados alcançáveis ao processar wa

Função de transição estendida

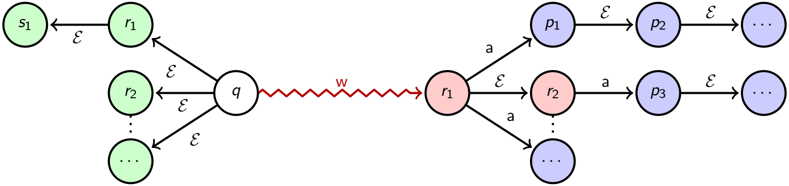
Definição

A Função de transição estendida (função programa) é denotada por

$$\hat{\delta}^* : Q \times \Sigma^* \to 2^Q, \text{ tal que:}$$

1. $\hat{\delta}^*(q, \mathcal{E}) = FECHO_{\mathcal{E}}(q)$
2. $\hat{\delta}^*(q, wa) = FECHO_{\mathcal{E}}(P)$, onde

$$P = \{p \mid \text{para algum } r \in \hat{\delta}^*(q, w), p \in FECHO_{\mathcal{E}}(\delta(r, a))\}$$



Ou seja, $\hat{\delta}^*(q, wa)$ inclui todos os estados alcançáveis ao processar wa

Função de transição estendida

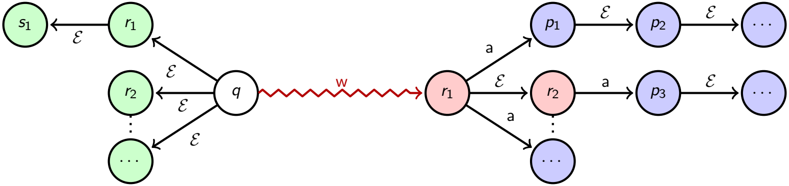
Definição

A Função de transição estendida (função programa) é denotada por

$$\hat{\delta}^{*} : Q \times \Sigma^{*} \rightarrow 2^{Q}, \text{ tal que:}$$

1. $\hat{\delta}^{*}(q, \mathcal{E}) = FECHO_{\mathcal{E}}(q)$
2. $\hat{\delta}^{*}(q, wa) = FECHO_{\mathcal{E}}(P)$, onde

$$P = \{p \mid \text{para algum } r \in \hat{\delta}^{*}(q, w), p \in FECHO_{\mathcal{E}}(\delta(r, a))\}$$



Ou seja, $\hat{\delta}^{*}(q, wa)$ inclui todos os estados alcançáveis ao processar wa

Função de transição estendida

continuação...

É conveniente estender δ e $\hat{\delta}^*$ para conjuntos de estados:

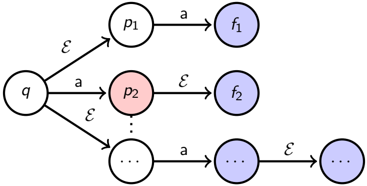
3. $\delta(P, a) = \bigcup_{q \in P} \delta(q, a)$
4. $\hat{\delta}^*(P, w) = \bigcup_{q \in P} \hat{\delta}^*(q, w)$

Função de transição estendida

Notação $\delta \neq \hat{\delta}^*$

Observe que:

- $\hat{\delta}^*(q, a)$ inclui todos os estados alcançáveis a partir de q por caminhos rotulados por 'a' (incluindo caminhos rotulados com $\mathcal{E}$)
- $\delta(q, a)$ inclui apenas estados alcançáveis por 'a'

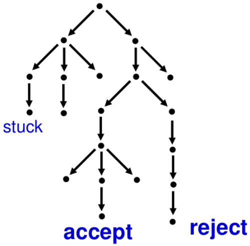


Portanto será necessário distinguir $\hat{\delta}^*$ e δ

Funcionamento de um AFN$_\mathcal{E}$

Condições de parada: análoga à do AFN.

- **aceita** a entrada:
  - se existir pelo menos um caminho $\hat{\delta}^*(q_0, w)$ que leve até um estado de aceitação em F.
- **rejeita** a entrada:
  - se não existir nenhum caminho, partindo de q_0, que leve até um estado de aceitação.



Linguagem reconhecida

Definição

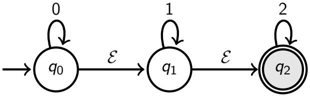
A linguagem reconhecida pelo AFN$_\mathcal{E}$ $N = (Q, \Sigma, \delta, q_0, F)$:

$$L(N) = \{w \mid w \in \Sigma^* \text{ e } \hat{\delta}^*(q_0, w) \text{ contém um estado em } F\}$$

De forma análoga, w é aceita por N se $\hat{\delta}^*(q_0, w) \cap F \neq \varnothing$

Movimento vazio

Considere N_4 e a cadeia de entrada $w = 01$



$$\hat{\delta}^*(q_0, \mathcal{E}) = FECHO_{\mathcal{E}}(q_0) = \{q_0, q_1, q_2\}$$

então

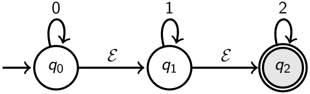
$$\begin{aligned}
\hat{\delta}^*(q_0, 0) &= FECHO_{\mathcal{E}}(\delta(\hat{\delta}^*(q_0, \mathcal{E}), 0)) = FECHO_{\mathcal{E}}(\delta(\{q_0, q_1, q_2\}, 0)) \\
&= FECHO_{\mathcal{E}}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0)) \\
&= FECHO_{\mathcal{E}}(\{q_0\} \cup \varnothing \cup \varnothing) = FECHO_{\mathcal{E}}(\{q_0\}) = \{q_0, q_1, q_2\}
\end{aligned}$$

por fim

$$\begin{aligned}
\hat{\delta}^*(q_0, 01) &= FECHO_{\mathcal{E}}(\delta(\hat{\delta}^*(q_0, 0), 1)) = FECHO_{\mathcal{E}}(\delta(\{q_0, q_1, q_2\}, 1)) \\
&= FECHO_{\mathcal{E}}(\{q_1\}) = \{q_1, q_2\}
\end{aligned}$$

Movimento vazio

Considere N_4 e a cadeia de entrada $w = 01$



$$\hat{\delta}^*(q_0, \mathcal{E}) = FECHO_{\mathcal{E}}(q_0) = \{q_0, q_1, q_2\}$$

então

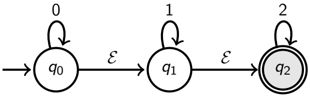
$$\begin{aligned}
\hat{\delta}^*(q_0, 0) &= FECHO_{\mathcal{E}}(\delta(\hat{\delta}^*(q_0, \mathcal{E}), 0)) = FECHO_{\mathcal{E}}(\delta(\{q_0, q_1, q_2\}, 0)) \\
&= FECHO_{\mathcal{E}}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0)) \\
&= FECHO_{\mathcal{E}}(\{q_0\} \cup \varnothing \cup \varnothing) = FECHO_{\mathcal{E}}(\{q_0\}) = \{q_0, q_1, q_2\}
\end{aligned}$$

por fim

$$\begin{aligned}
\hat{\delta}^*(q_0, 01) &= FECHO_{\mathcal{E}}(\delta(\hat{\delta}^*(q_0, 0), 1)) = FECHO_{\mathcal{E}}(\delta(\{q_0, q_1, q_2\}, 1)) \\
&= FECHO_{\mathcal{E}}(\{q_1\}) = \{q_1, q_2\}
\end{aligned}$$

Movimento vazio

Considere N_4 e a cadeia de entrada $w = 01$



$$\hat{\delta}^*(q_0, \mathcal{E}) = FECHO_{\mathcal{E}}(q_0) = \{q_0, q_1, q_2\}$$

então

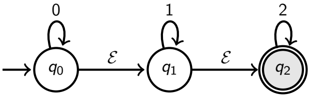
$$\begin{aligned}
\hat{\delta}^*(q_0, 0) &= FECHO_{\mathcal{E}}(\delta(\hat{\delta}^*(q_0, \mathcal{E}), 0)) = FECHO_{\mathcal{E}}(\delta(\{q_0, q_1, q_2\}, 0)) \\
&= FECHO_{\mathcal{E}}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0)) \\
&= FECHO_{\mathcal{E}}(\{q_0\} \cup \varnothing \cup \varnothing) = FECHO_{\mathcal{E}}(\{q_0\}) = \{q_0, q_1, q_2\}
\end{aligned}$$

por fim

$$\begin{aligned}
\hat{\delta}^*(q_0, 01) &= FECHO_{\mathcal{E}}(\delta(\hat{\delta}^*(q_0, 0), 1)) = FECHO_{\mathcal{E}}(\delta(\{q_0, q_1, q_2\}, 1)) \\
&= FECHO_{\mathcal{E}}(\{q_1\}) = \{q_1, q_2\}
\end{aligned}$$

Movimento vazio

Considere N_4 e a cadeia de entrada $w = 01$



$$\hat{\delta}^*(q_0, \mathcal{E}) = FECHO_{\mathcal{E}}(q_0) = \{q_0, q_1, q_2\}$$

então

$$\begin{aligned}
\hat{\delta}^*(q_0, 0) &= FECHO_{\mathcal{E}}(\delta(\hat{\delta}^*(q_0, \mathcal{E}), 0)) = FECHO_{\mathcal{E}}(\delta(\{q_0, q_1, q_2\}, 0)) \\
&= FECHO_{\mathcal{E}}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0)) \\
&= FECHO_{\mathcal{E}}(\{q_0\} \cup \varnothing \cup \varnothing) = FECHO_{\mathcal{E}}(\{q_0\}) = \{q_0, q_1, q_2\}
\end{aligned}$$

por fim

$$\begin{aligned}
\hat{\delta}^*(q_0, 01) &= FECHO_{\mathcal{E}}(\delta(\hat{\delta}^*(q_0, 0), 1)) = FECHO_{\mathcal{E}}(\delta(\{q_0, q_1, q_2\}, 1)) \\
&= FECHO_{\mathcal{E}}(\{q_1\}) = \{q_1, q_2\}
\end{aligned}$$

Roteiro

1. Autômatos finitos com Movimentos Vazios (AFN$_\mathcal{E}$s)
   - Movimento Vazio
   - Formalização de um AFN$_\mathcal{E}$
   - Equivalência entre AFN e AFN$_\mathcal{E}$
2. Expressões Regulares (ERs)
   - Formalização de uma ER
   - Exemplos de ERs
   - Equivalência entre ER e AFN$_\mathcal{E}$
3. Referências

Equivalência entre AFN e AFN$_{\mathcal{E}}$

O movimento vazio não aumenta o poder de reconhecimento de linguagens para os AFNs

- A classe de linguagens aceitas[1] é a mesma.

Vantagens dos AFN$_{\mathcal{E}}$'s:

- Facilitam algumas construções e demonstrações

[1] Linguagens Regulares

Equivalência entre AFN e AFN$_{\mathcal{E}}$

O movimento vazio não aumenta o poder de reconhecimento de linguagens para os AFNs

- A classe de linguagens aceitas[1] é a mesma.

Vantagens dos AFN$_{\mathcal{E}}$'s:

- Facilitam algumas construções e demonstrações

[1] Linguagens Regulares

Equivalência entre AFN e AFN$_\mathcal{E}$

Teorema:

Qualquer linguagem reconhecida por um AFN$_\mathcal{E}$ N também pode ser reconhecida por um AFN N' equivalente.

$$L(N) = L(N')$$

Equivalência entre AFN e AFN$_{\mathcal{E}}$

Equivalência

- AFN $\rightarrow$ AFN$_{\mathcal{E}}$
  - Não precisa ser provado, por definição todo AFN é um AFN$_{\mathcal{E}}$.
- AFN$_{\mathcal{E}}$ $\rightarrow$ AFN
  - Prova por construção (iremos apresentar um procedimento).

A ideia central é mostrar que as funções de transição do AFN$_{\mathcal{E}}$ e do AFN equivalente são iguais.

Equivalência entre AFN e AFN$_\mathcal{E}$

Equivalência

- AFN $\rightarrow$ AFN$_\mathcal{E}$
  - Não precisa ser provado, por definição todo AFN é um AFN$_\mathcal{E}$.
- AFN$_\mathcal{E}$ $\rightarrow$ AFN
  - Prova por construção (iremos apresentar um procedimento).

A ideia central é mostrar que as funções de transição do AFN$_\mathcal{E}$ e do AFN equivalente são iguais.

Equivalência entre AFN e AFN$_\mathcal{E}$

Equivalência

- AFN $\rightarrow$ AFN$_\mathcal{E}$
  - Não precisa ser provado, por definição todo AFN é um AFN$_\mathcal{E}$.
- AFN$_\mathcal{E}$ $\rightarrow$ AFN
  - Prova por construção (iremos apresentar um procedimento).

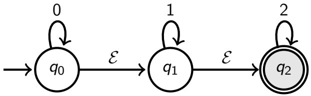
A ideia central é mostrar que as funções de transição do AFN$_\mathcal{E}$ e do AFN equivalente são iguais.

Equivalência entre AFN e AFN$_{\mathcal{E}}$

Vamos definir o AFN$_{\mathcal{E}}$ equivalente $N' = (Q, \Sigma, \delta', q_0, F')$, tal que:

1. $F' = \{q \mid q \in Q$ e $FECHO_{\mathcal{E}}(q)$ contém um estado em $F\}$
2. $\delta'(q, a) = \hat{\delta}^*(q, a) = \{q_i, q_j, \ldots, q_r\}$, para todo $q \in Q$ e $a \in \Sigma$.

Exemplo:



	0	1	2
$\rightarrow q_0$			
q_1			
$\star q_2$			

Tabela: Transições

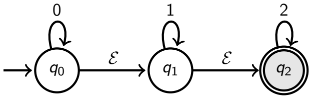
$\hat{\delta}^*(q, a) = \{q_i, q_j, \ldots, q_r\}$ inclui todos os estados alcançáveis por q ao processar a, possivelmente incluindo transições $\mathcal{E}$.

Equivalência entre AFN e AFN$_{\mathcal{E}}$

Vamos definir o AFN$_{\mathcal{E}}$ equivalente $N' = (Q, \Sigma, \delta', q_0, F')$, tal que:

1. $F' = \{q \mid q \in Q$ e $FECHO_{\mathcal{E}}(q)$ contém um estado em $F\}$
2. $\delta'(q, a) = \hat{\delta}^*(q, a) = \{q_i, q_j, \ldots, q_r\}$, para todo $q \in Q$ e $a \in \Sigma$.

Exemplo:



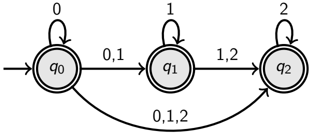
	0	1	2
$\rightarrow q_0$			
q_1			
$\star q_2$			

Tabela: Transições

$\hat{\delta}^*(q, a) = \{q_i, q_j, \ldots, q_r\}$ inclui todos os estados alcançáveis por q ao processar a, possivelmente incluindo transições $\mathcal{E}$.

Equivalência entre AFN e AFN$_\mathcal{E}$

Resultado:



	0	1	2
$\rightarrow \star q_0$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
$\star q_1$	$\varnothing$	$\{q_1, q_2\}$	$\{q_2\}$
$\star q_2$	$\varnothing$	$\varnothing$	$\{q_2\}$

Tabela: Transições

Equivalência entre AFN e AFN$_\mathcal{E}$

O AFN N' simula todas as computações do AFN$_\mathcal{E}$ N?

Prova formal:

É fácil ver por indução no tamanho de $w = w_1 w_2 \dots w_n$ que

$$\delta'^*(q_0, w) = \{q_i, q_j, \dots, q_k\}$$

se e somente se

$$\hat{\delta}^*(q_0, w) = \{q_i, q_j, \dots, q_k\}$$

(dessa vez) para

$$w \neq \mathcal{E}$$

Quando $w = \mathcal{E}$:

Equivalência entre AFN e AFN$_{\mathcal{E}}$

O AFN N' simula todas as computações do AFN$_{\mathcal{E}}$ N?

Prova formal:

É fácil ver por indução no tamanho de $w = w_1 w_2 \dots w_n$ que

$$\delta'^{*}(q_0, w) = \{q_i, q_j, \dots, q_k\}$$

se e somente se

$$\hat{\delta}^{*}(q_0, w) = \{q_i, q_j, \dots, q_k\}$$

(dessa vez) para

$$w \neq \mathcal{E}$$

Quando $w = \mathcal{E}$:

- Segue da definição que no AFN $\delta'^{*}(q_0, \mathcal{E}) = \{q_0\}$ e por construção $q_0 \in F'$ sempre que no AFN$_{\mathcal{E}}$ $FECHO_{\mathcal{E}}(q_0)$ contém um estado em F.

Equivalência entre AFN e AFN$_{\mathcal{E}}$

Base da indução: $|w| = 1$, portanto $w = a$ é um símbolo de Σ, e

$$\delta'(q_0, a) = \hat{\delta}(q_0, a)$$

- Verdadeiro, por definição (ver página 21).

Equivalência entre AFN e AFN$_{\mathcal{E}}$

Hipótese de indução: $|w| = n$ e $n > 1$, suponha que:

$$\delta'^*(q_0, w) = \hat{\delta}^*(q_0, w) = \{q_u, q_v, \ldots, q_w\} = P$$

Passo: $|wa| = n + 1$ e $n \geq 1$:

$$\delta'^*(q_0, wa) = \delta'(\delta'^*(q_0, w), a)$$

Pela HI:

$$\delta'(\delta'^*(q_0, w), a) = \delta'^*(P, a) = \bigcup_{q \in P} \delta'^*(q, a) = \bigcup_{q \in P} \hat{\delta}^*(q, a)$$

como

$$P = \hat{\delta}^*(q_0, w)$$

então

$$\bigcup_{q \in P} \hat{\delta}^*(q, a) = \hat{\delta}^*(q_0, wa)$$

- É verdadeiro pela definição regra (2), página 11.

Equivalência entre AFN e AFN$_{\mathcal{E}}$

Hipótese de indução: $|w| = n$ e $n > 1$, suponha que:

$$\delta'^*(q_0, w) = \hat{\delta}^*(q_0, w) = \{q_u, q_v, \ldots, q_w\} = P$$

Passo: $|wa| = n + 1$ e $n \geq 1$:

$$\delta'^*(q_0, wa) = \delta'(\delta'^*(q_0, w), a)$$

Pela HI:

$$\delta'(\delta'^*(q_0, w), a) = \delta'^*(P, a) = \bigcup_{q \in P} \delta'^*(q, a) = \bigcup_{q \in P} \hat{\delta}^*(q, a)$$

como

$$P = \hat{\delta}^*(q_0, w)$$

então

$$\bigcup_{q \in P} \hat{\delta}^*(q, a) = \hat{\delta}^*(q_0, wa)$$

- É verdadeiro pela definição regra (2), página 11.

Equivalência entre AFN e AFN$_\mathcal{E}$

Hipótese de indução: $|w| = n$ e $n > 1$, suponha que:

$$\delta'^*(q_0, w) = \hat{\delta}^*(q_0, w) = \{q_u, q_v, \ldots, q_w\} = P$$

Passo: $|wa| = n + 1$ e $n \geq 1$:

$$\delta'^*(q_0, wa) = \delta'(\delta'^*(q_0, w), a)$$

Pela HI:

$$\delta'(\delta'^*(q_0, w), a) = \delta'^*(P, a) = \bigcup_{q \in P} \delta'^*(q, a) = \bigcup_{q \in P} \hat{\delta}^*(q, a)$$

como

$$P = \hat{\delta}^*(q_0, w)$$

então

$$\bigcup_{q \in P} \hat{\delta}^*(q, a) = \hat{\delta}^*(q_0, wa)$$

- É verdadeiro pela definição regra (2), página 11.

Equivalência entre AFN e AFN$_{\mathcal{E}}$

Hipótese de indução: $|w| = n$ e $n > 1$, suponha que:

$$\delta'^{*}(q_0, w) = \hat{\delta}^{*}(q_0, w) = \{q_u, q_v, \ldots, q_w\} = P$$

Passo: $|wa| = n + 1$ e $n \geq 1$:

$$\delta'^{*}(q_0, wa) = \delta'(\delta'^{*}(q_0, w), a)$$

Pela HI:

$$\delta'(\delta'^{*}(q_0, w), a) = \delta'^{*}(P, a) = \bigcup_{q \in P} \delta'^{*}(q, a) = \bigcup_{q \in P} \hat{\delta}^{*}(q, a)$$

como

$$P = \hat{\delta}^{*}(q_0, w)$$

então

$$\bigcup_{q \in P} \hat{\delta}^{*}(q, a) = \hat{\delta}^{*}(q_0, wa)$$

- É verdadeiro pela definição regra (2), página 11.

Equivalência entre AFN e AFN$_{\mathcal{E}}$

Hipótese de indução: $|w| = n$ e $n > 1$, suponha que:

$$\delta'^{*}(q_0, w) = \hat{\delta}^{*}(q_0, w) = \{q_u, q_v, \ldots, q_w\} = P$$

Passo: $|wa| = n + 1$ e $n \geq 1$:

$$\delta'^{*}(q_0, wa) = \delta'(\delta'^{*}(q_0, w), a)$$

Pela HI:

$$\delta'(\delta'^{*}(q_0, w), a) = \delta'^{*}(P, a) = \bigcup_{q \in P} \delta'^{*}(q, a) = \bigcup_{q \in P} \hat{\delta}^{*}(q, a)$$

como

$$P = \hat{\delta}^{*}(q_0, w)$$

então

$$\bigcup_{q \in P} \hat{\delta}^{*}(q, a) = \hat{\delta}^{*}(q_0, wa)$$

- É verdadeiro pela definição regra (2), página 11.

Equivalência entre AFN e AFN$_{\mathcal{E}}$

Então, $\delta'^*(q_0, wa) = \hat{\delta}^*(q_0, wa)$ $\square$

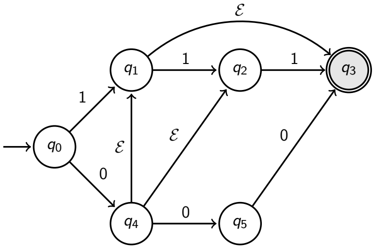
- Portanto,

$$L(N') = L(N)$$

Equivalência entre AFN e AFN$_\mathcal{E}$

Outro exemplo:

- Seja $N_5 = (\{q_0, q_1, \ldots, q_5\}, \{0, 1\}, \delta, q_0, \{q_3\})$ um AFN$_\mathcal{E}$:



- Vamos contruir $N_5' = (Q, \Sigma, \delta', q_0, F')$, tal que $L(N_5) = L(N_5')$

Equivalência entre AFN e AFN$_\mathcal{E}$

1. Calculando F':

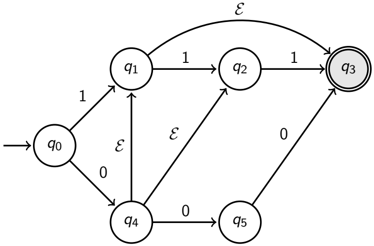
$$FECHO_\mathcal{E}(q_1) = \{q_1, q_3\}$$

$$FECHO_\mathcal{E}(q_4) = \{q_1, q_2, q_3, q_4\}$$

2. $\delta'(q, a) = \hat{\delta}^*(q, a) = \{q_i, q_j, \ldots, q_r\}$ todos os estados alcançáveis por q ao processar a

	0	1
$\rightarrow q_0$		
q_1		
q_2		
$\star q_3$		
q_4		
q_5		

Tabela: Transições



Equivalência entre AFN e AFN$_\mathcal{E}$

1. Calculando F':

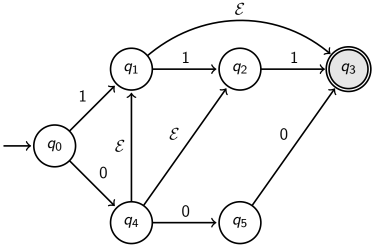
$$FECHO_\mathcal{E}(q_1) = \{q_1, q_3\}$$

$$FECHO_\mathcal{E}(q_4) = \{q_1, q_2, q_3, q_4\}$$

2. $\delta'(q, a) = \hat{\delta}^*(q, a) = \{q_i, q_j, \ldots, q_r\}$ todos os estados alcançáveis por q ao processar a

	0	1
$\rightarrow q_0$		
q_1		
q_2		
$\star q_3$		
q_4		
q_5		

Tabela: Transições

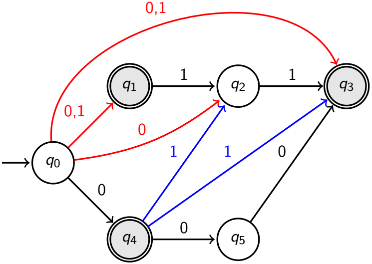


Equivalência entre AFN e AFN$_{\mathcal{E}}$

Resultado:

	0	1
$\rightarrow q_0$	$\{q_4, q_1, q_2, q_3\}$	$\{q_1, q_3\}$
$\star q_1$	$\varnothing$	$\{q_2\}$
q_2	$\varnothing$	$\{q_3\}$
$\star q_3$	$\varnothing$	$\varnothing$
$\star q_4$	$\{q_5\}$	$\{q_2, q_3\}$
q_5	$\{q_3\}$	$\varnothing$

Tabela: Transições



Roteiro

Expressões Regulares

Podemos usar expressões simples, chamadas de Expressões Regulares (**ER**), para descrever linguagens (conjuntos de cadeias):

- **ER** é um formalismo denotacional.
- **ER** explica como gerar palavras.

Exemplos:

$$\mathbf{a} = \{a\}$$

$$\mathbf{a}^* = \{\mathcal{E}, a, aa, aaa, \dots\}$$

$$(\mathbf{ab})^+ = \{ab, abab, ababab, \dots\}$$

Expressões Regulares

Vamos ver que o valor de uma **ER** (conjuntos de cadeias) é sempre uma Linguagem Regular[2].

Dessa forma:

- Toda Linguagem Regular pode ser descrita por uma **ER**

[2] Classe de linguagens reconhecidas por AFs.

Vamos ver que o valor de uma **ER** (conjuntos de cadeias) é sempre uma Linguagem Regular[2].

Dessa forma:

- Toda Linguagem Regular pode ser descrita por uma **ER**

[2] Classe de linguagens reconhecidas por AFs.

Expressões Regulares

ERs são úteis para descrever padrões para busca e reconhecimento em textos:

- Programas como AWK e GREP em sistemas operacionais. Unix-like
- Linguagens de programação: PEARL, PYTHON, ...
- Projeto de compiladores.

Roteiro

Expressões Regulares

Uma **ER** é definida a partir de **conjuntos básicos** (linguagens) e **operações** de concatenação e união.

Base:

1. $\emptyset$ é uma **ER** que denota o conjunto vazio
2. $\mathcal{E}$ é uma **ER** que denota $\{\mathcal{E}\}$
3. Para cada $a \in \Sigma$, **a** é a **ER** para $\{a\}$

Operações:

- Sejam $\mathbf{r_1}$ e $\mathbf{r_2}$ **ER**s que denotam as linguagens R_1 e R_2:
  1. União: $\mathbf{r_1} + \mathbf{r_2}$ denota a linguagem $R_1 \cup R_2$
  2. Concatenação: $\mathbf{r_1 r_2}$ denota $RS = \{w = uv \mid u \in R_1 \text{ e } v \in R_2\}$
  3. Operação estrela: $\mathbf{r_1^*}$ denota R_1^*

Expressões Regulares

Uma **ER** é definida a partir de **conjuntos básicos** (linguagens) e **operações** de concatenação e união.

Base:

1. $\emptyset$ é uma **ER** que denota o conjunto vazio
2. $\mathcal{E}$ é uma **ER** que denota $\{\mathcal{E}\}$
3. Para cada $a \in \Sigma$, **a** é a **ER** para $\{a\}$

Operações:

- Sejam $\mathbf{r_1}$ e $\mathbf{r_2}$ **ER**s que denotam as linguagens R_1 e R_2:

Expressões Regulares

Uma **ER** é definida a partir de **conjuntos básicos** (linguagens) e **operações** de concatenação e união.

Base:

1. $\emptyset$ é uma **ER** que denota o conjunto vazio
2. $\mathcal{E}$ é uma **ER** que denota $\{\mathcal{E}\}$
3. Para cada $a \in \Sigma$, **a** é a **ER** para $\{a\}$

Operações:

- Sejam $\mathbf{r_1}$ e $\mathbf{r_2}$ **ER**s que denotam as linguagens R_1 e R_2:
  1. União: $\mathbf{r_1} + \mathbf{r_2}$ denota a linguagem $R_1 \cup R_2$
  2. Concatenação: $\mathbf{r_1 r_2}$ denota $RS = \{w = uv \mid u \in R_1 \text{ e } v \in R_2\}$
  3. Operação estrela: $\mathbf{r_1^*}$ denota R_1^*

Expressões Regulares

Uma **ER** é definida a partir de **conjuntos básicos** (linguagens) e **operações** de concatenação e união.

Base:

1. $\emptyset$ é uma **ER** que denota o conjunto vazio
2. $\mathcal{E}$ é uma **ER** que denota $\{\mathcal{E}\}$
3. Para cada $a \in \Sigma$, **a** é a **ER** para $\{a\}$

Operações:

- Sejam $\mathbf{r_1}$ e $\mathbf{r_2}$ **ER**s que denotam as linguagens R_1 e R_2:
  1. União: $\mathbf{r_1} + \mathbf{r_2}$ denota a linguagem $R_1 \cup R_2$
  2. Concatenação: $\mathbf{r_1 r_2}$ denota $RS = \{w = uv \mid u \in R_1 \text{ e } v \in R_2\}$
  3. Operação estrela: $\mathbf{r_1^*}$ denota R_1^*

Expressões Regulares

Uma **ER** é definida a partir de **conjuntos básicos** (linguagens) e **operações** de concatenação e união.

Base:

1. $\emptyset$ é uma **ER** que denota o conjunto vazio
2. $\mathcal{E}$ é uma **ER** que denota $\{\mathcal{E}\}$
3. Para cada $a \in \Sigma$, **a** é a **ER** para $\{a\}$

Operações:

- Sejam $\mathbf{r_1}$ e $\mathbf{r_2}$ **ER**s que denotam as linguagens R_1 e R_2:
  1. União: $\mathbf{r_1} + \mathbf{r_2}$ denota a linguagem $R_1 \cup R_2$
  2. Concatenação: $\mathbf{r_1 r_2}$ denota $RS = \{w = uv \mid u \in R_1 \text{ e } v \in R_2\}$
  3. Operação estrela: $\mathbf{r_1^*}$ denota R_1^*

Expressões Regulares

Linguagem gerada

A linguagem denotada por uma **ER r** também pode ser chamada de:

- Linguagem gerada por **r**

$$L(\mathbf{r})$$

Exemplos:

$$L(\mathbf{a}^*) = \{a^n \mid n \geq 0\}$$

$$L((\mathbf{a}+\mathbf{b})^*) = \{w \mid w \in \{a, b\}^*\}$$

$$L(\mathbf{a}^* + \mathbf{b}^*) = \{w \mid w = a^n \text{ ou } w = b^n, n \geq 0\}$$

Usualmente, podemos omitir os parênteses nas **ER**s, respeitamos as convenções de precedência de operadores.

$$\text{Operação estrela} >> \text{Concatenação} >> \text{União}$$

então

$$\mathbf{a} + \mathbf{bc}^* = (\mathbf{a} + (\mathbf{b}(\mathbf{c}^*)))$$

Operação estrela:

- $\mathbf{r}^*$ gera todas as palavras que são zero ou mais concatenações de cadeias de **r**
- $\mathbf{r}^+$ gera todas as palavras que são uma ou mais concatenações de cadeias de **r**
- Dessa forma,

$$\mathbf{r}^+ = \mathbf{r}\mathbf{r}^*$$

$$(\mathbf{r}^+ + \mathcal{E}) = \mathbf{r}^*$$

Operação estrela:

- $\mathbf{r}^*$ gera todas as palavras que são zero ou mais concatenações de cadeias de **r**
- $\mathbf{r}^+$ gera todas as palavras que são uma ou mais concatenações de cadeias de **r**
- Dessa forma,

$$\mathbf{r}^+ = \mathbf{r}\mathbf{r}^*$$

$$(\mathbf{r}^+ + \mathcal{E}) = \mathbf{r}^*$$

Além disso, seja Σ um alfabeto qualquer:

- Σ^* é a linguagem com todas as cadeias sobre Σ
  - Sendo $\Sigma = \{a, b\}$

$$\boldsymbol{\Sigma}^* = ((\mathbf{a} + \mathbf{b})^+ + \mathcal{E}) = (\mathbf{a} + \mathbf{b})^*$$

Roteiro

Exemplos de ER

Alguns exemplos: considere $\Sigma = \{a, b\}$.

ER	Linguagem gerada
$\mathbf{aa}$	Somente a palavra 'aa'
$\mathbf{ba}^*$	Palavras que começam com 'b' seguidas por zero ou mais 'a's
$\mathbf{a}^*\mathbf{ba}^*$	Palavras que contém um único 'b'
$\boldsymbol{\Sigma}^*\mathbf{b}\boldsymbol{\Sigma}^*$	Palavras que tem, pelo menos, um símbolo 'b'
$\boldsymbol{\Sigma}^*\mathbf{aba}\boldsymbol{\Sigma}^*$	Palavras que contém a subcadeia 'aba'
$\mathbf{b}^*(\mathbf{ab}^+)^*$	Palavras em que todo 'a' é seguido por pelo menos um 'b'
$(\boldsymbol{\Sigma\Sigma})^*$	Palavras de comprimento par

Exemplos de ER

Alguns exemplos: considere $\Sigma = \{a, b\}$.

ER	Linguagem gerada
$\mathbf{aa}$	Somente a palavra 'aa'
$\mathbf{ba}^*$	Palavras que começam com 'b' seguidas por zero ou mais 'a's
$\mathbf{a}^*\mathbf{ba}^*$	Palavras que contém um único 'b'
$\mathbf{\Sigma}^*\mathbf{b}\mathbf{\Sigma}^*$	Palavras que tem, pelo menos, um símbolo 'b'
$\mathbf{\Sigma}^*\mathbf{aba}\mathbf{\Sigma}^*$	Palavras que contém a subcadeia 'aba'
$\mathbf{b}^*(\mathbf{ab}^+)^*$	Palavras em que todo 'a' é seguido por pelo menos um 'b'
$(\mathbf{\Sigma\Sigma})^*$	Palavras de comprimento par

Exemplos de ER

Alguns exemplos: considere $\Sigma = \{a, b\}$.

ER	Linguagem gerada
$\mathbf{aa}$	Somente a palavra 'aa'
$\mathbf{ba}^*$	Palavras que começam com 'b' seguidas por zero ou mais 'a's
$\mathbf{a}^*\mathbf{ba}^*$	Palavras que contém um único 'b'
$\mathbf{\Sigma}^*\mathbf{b}\mathbf{\Sigma}^*$	Palavras que tem, pelo menos, um símbolo 'b'
$\mathbf{\Sigma}^*\mathbf{aba}\mathbf{\Sigma}^*$	Palavras que contém a subcadeia 'aba'
$\mathbf{b}^*(\mathbf{ab}^+)^*$	Palavras em que todo 'a' é seguido por pelo menos um 'b'
$(\mathbf{\Sigma\Sigma})^*$	Palavras de comprimento par

Exemplos de ER

Alguns exemplos: considere $\Sigma = \{a, b\}$.

ER	Linguagem gerada
$\mathbf{aa}$	Somente a palavra 'aa'
$\mathbf{ba}^*$	Palavras que começam com 'b' seguidas por zero ou mais 'a's
$\mathbf{a}^*\mathbf{ba}^*$	Palavras que contém um único 'b'
$\mathbf{\Sigma}^*\mathbf{b}\mathbf{\Sigma}^*$	Palavras que tem, pelo menos, um símbolo 'b'
$\mathbf{\Sigma}^*\mathbf{aba}\mathbf{\Sigma}^*$	Palavras que contém a subcadeia 'aba'
$\mathbf{b}^*(\mathbf{ab}^+)^*$	Palavras em que todo 'a' é seguido por pelo menos um 'b'
$(\mathbf{\Sigma\Sigma})^*$	Palavras de comprimento par

Exemplos de ER

Alguns exemplos: considere $\Sigma = \{a, b\}$.

ER	Linguagem gerada
$\mathbf{aa}$	Somente a palavra 'aa'
$\mathbf{ba}^*$	Palavras que começam com 'b' seguidas por zero ou mais 'a's
$\mathbf{a}^*\mathbf{ba}^*$	Palavras que contém um único 'b'
$\boldsymbol{\Sigma}^*\mathbf{b}\boldsymbol{\Sigma}^*$	Palavras que tem, pelo menos, um símbolo 'b'
$\boldsymbol{\Sigma}^*\mathbf{aba}\boldsymbol{\Sigma}^*$	Palavras que contém a subcadeia 'aba'
$\mathbf{b}^*(\mathbf{ab}^+)^*$	Palavras em que todo 'a' é seguido por pelo menos um 'b'
$(\boldsymbol{\Sigma\Sigma})^*$	Palavras de comprimento par

Exemplos de ER

Alguns exemplos: considere $\Sigma = \{a, b\}$.

ER	Linguagem gerada
$\mathbf{aa}$	Somente a palavra ‘aa’
$\mathbf{ba}^*$	Palavras que começam com ‘b’ seguidas por zero ou mais ‘a’s
$\mathbf{a}^*\mathbf{ba}^*$	Palavras que contém um único ‘b’
$\boldsymbol{\Sigma}^*\mathbf{b}\boldsymbol{\Sigma}^*$	Palavras que tem, pelo menos, um símbolo ‘b’
$\boldsymbol{\Sigma}^*\mathbf{aba}\boldsymbol{\Sigma}^*$	Palavras que contém a subcadeia ‘aba’
$\mathbf{b}^*(\mathbf{ab}^+)^*$	Palavras em que todo ‘a’ é seguido por pelo menos um ‘b’
$(\boldsymbol{\Sigma}\boldsymbol{\Sigma})^*$	Palavras de comprimento par

Exemplos de ER

Alguns exemplos: considere $\Sigma = \{a, b\}$.

ER	Linguagem gerada
$\mathbf{aa}$	Somente a palavra 'aa'
$\mathbf{ba}^*$	Palavras que começam com 'b' seguidas por zero ou mais 'a's
$\mathbf{a}^*\mathbf{ba}^*$	Palavras que contém um único 'b'
$\mathbf{\Sigma}^*\mathbf{b}\mathbf{\Sigma}^*$	Palavras que tem, pelo menos, um símbolo 'b'
$\mathbf{\Sigma}^*\mathbf{aba}\mathbf{\Sigma}^*$	Palavras que contém a subcadeia 'aba'
$\mathbf{b}^*(\mathbf{ab}^+)^*$	Palavras em que todo 'a' é seguido por pelo menos um 'b'
$(\mathbf{\Sigma\Sigma})^*$	Palavras de comprimento par

Exemplos de ER

Alguns exemplos: considere $\Sigma = \{a, b\}$.

ER	Linguagem gerada
$\mathbf{aa}$	Somente a palavra 'aa'
$\mathbf{ba}^*$	Palavras que começam com 'b' seguidas por zero ou mais 'a's
$\mathbf{a}^*\mathbf{ba}^*$	Palavras que contém um único 'b'
$\mathbf{\Sigma}^*\mathbf{b\Sigma}^*$	Palavras que tem, pelo menos, um símbolo 'b'
$\mathbf{\Sigma}^*\mathbf{aba\Sigma}^*$	Palavras que contém a subcadeia 'aba'
$\mathbf{b}^*(\mathbf{ab}^+)^*$	Palavras em que todo 'a' é seguido por pelo menos um 'b'
$(\mathbf{\Sigma\Sigma})^*$	Palavras de comprimento par

Exemplos de ER

Mais exemplos: considere $\Sigma = \{a, b\}$.

ER	Linguagem gerada
$\mathbf{ab} + \mathbf{ba}$	Somente as palavras $\{ab, ba\}$
$(\mathbf{a} + \mathbf{b})^*$	Todas as palavras sobre Σ
$(\mathbf{a} + \mathbf{b})^*\mathbf{aa}(\mathbf{a} + \mathbf{b})^*$	Todas as palavras que contém a subcadeia 'aa'
$(\mathbf{a} + \mathbf{b})^*(\mathbf{aa} + \mathbf{bb})$	Todas as palavras que terminam com 'aa' ou 'bb'

Exemplos de ER

Mais exemplos: considere $\Sigma = \{a, b\}$.

ER	**Linguagem gerada**
$\mathbf{ab} + \mathbf{ba}$	Somente as palavras $\{ab, ba\}$
$(\mathbf{a} + \mathbf{b})^*$	Todas as palavras sobre Σ
$(\mathbf{a} + \mathbf{b})^*\mathbf{aa}(\mathbf{a} + \mathbf{b})^*$	Todas as palavras que contém a subcadeia 'aa'
$(\mathbf{a} + \mathbf{b})^*(\mathbf{aa} + \mathbf{bb})$	Todas as palavras que terminam com 'aa' ou 'bb'

Expressões Regulares

Seja $\mathbf{r}$ uma **ER** qualquer:

- $\mathbf{r} + \emptyset = \mathbf{r}$
  - Adicionar a linguagem vazia não modifica $L(\mathbf{r})$
- $\mathbf{r}\mathcal{E} = \mathbf{r}$
  - Concatenar a cadeia vazia a qualquer cadeia não modifica $L(\mathbf{r})$

Entretanto:

- $\mathbf{r} + \mathcal{E}$ pode não ser igual à $\mathbf{r}$
  - Se $\mathbf{r}$ não contiver a cadeia vazia, $\mathcal{E}$ é adicionado à $L(\mathbf{r})$
- $\mathbf{r}\emptyset = \emptyset$
  - A concatenação de $\emptyset$ é sempre $\emptyset$

Expressões Regulares

Seja **r** uma **ER** qualquer:

- $\mathbf{r} + \emptyset = \mathbf{r}$
  - Adicionar a linguagem vazia não modifica $L(\mathbf{r})$
- $\mathbf{r}\mathcal{E} = \mathbf{r}$
  - Concatenar a cadeia vazia a qualquer cadeia não modifica $L(\mathbf{r})$

Entretanto:

- $\mathbf{r} + \mathcal{E}$ pode não ser igual à $\mathbf{r}$
  - Se $\mathbf{r}$ não contiver a cadeia vazia, $\mathcal{E}$ é adicionado à $L(\mathbf{r})$
- $\mathbf{r}\emptyset = \emptyset$
  - A concatenação de $\emptyset$ é sempre $\emptyset$

Expressões Regulares

Seja **r** uma **ER** qualquer:

- $\mathbf{r} + \emptyset = \mathbf{r}$
  - Adicionar a linguagem vazia não modifica $L(\mathbf{r})$
- $\mathbf{r}\mathcal{E} = \mathbf{r}$
  - Concatenar a cadeia vazia a qualquer cadeia não modifica $L(\mathbf{r})$

Entretanto:

- $\mathbf{r} + \mathcal{E}$ pode não ser igual à **r**
  - Se **r** não contiver a cadeia vazia, $\mathcal{E}$ é adicionado à $L(\mathbf{r})$
- $\mathbf{r}\emptyset = \emptyset$
  - A concatenação de $\emptyset$ é sempre $\emptyset$

Expressões Regulares

Seja $\mathbf{r}$ uma **ER** qualquer:

- $\mathbf{r} + \emptyset = \mathbf{r}$
  - Adicionar a linguagem vazia não modifica $L(\mathbf{r})$
- $\mathbf{r}\mathcal{E} = \mathbf{r}$
  - Concatenar a cadeia vazia a qualquer cadeia não modifica $L(\mathbf{r})$

Entretanto:

- $\mathbf{r} + \mathcal{E}$ pode não ser igual à $\mathbf{r}$
  - Se $\mathbf{r}$ não contiver a cadeia vazia, $\mathcal{E}$ é adicionado à $L(\mathbf{r})$
- $\mathbf{r}\emptyset = \emptyset$
  - A concatenação de $\emptyset$ é sempre $\emptyset$

Exemplos de ER

Mais exemplos: considere $\Sigma = \{a, b\}$.

ER	Linguagem gerada
$(a + \mathcal{E})b^*$	$\{w \mid w = ab^* \text{ ou } w = b^*\}$
$(a + \mathcal{E})(b + \mathcal{E})$	$\{\mathcal{E}, a, b, ab\}$
$b^*\emptyset$	$\emptyset$
$\emptyset^*$	$\{\mathcal{E}\}$, (ou seja, juntar qualquer número de cadeias, incluí juntar 0 cadeias, o que gera a cadeia vazia)

Exemplos de ER

Mais exemplos: considere $\Sigma = \{a, b\}$.

ER	**Linguagem gerada**
$(a + \mathcal{E})b^*$	$\{w \mid w = ab^* \text{ ou } w = b^*\}$
$(a + \mathcal{E})(b + \mathcal{E})$	$\{\mathcal{E}, a, b, ab\}$
$b^*\emptyset$	$\emptyset$
$\emptyset^*$	$\{\mathcal{E}\}$, (ou seja, juntar qualquer número de cadeias, incluí juntar 0 cadeias, o que gera a cadeia vazia)

Exemplos de ER

Mais exemplos: considere $\Sigma = \{a, b\}$.

ER	**Linguagem gerada**
$(a + \mathcal{E})b^*$	$\{w \mid w = ab^*$ ou $w = b^*\}$
$(a + \mathcal{E})(b + \mathcal{E})$	$\{\mathcal{E}, a, b, ab\}$
$b^*\emptyset$	$\emptyset$
$\emptyset^*$	$\{\mathcal{E}\}$, (ou seja, juntar qualquer número de cadeias, incluí juntar 0 cadeias, o que gera a cadeia vazia)

Exemplos de ER

Mais exemplos: considere $\Sigma = \{a, b\}$.

ER	**Linguagem gerada**
$(a + \mathcal{E})b^*$	$\{w \mid w = ab^* \text{ ou } w = b^*\}$
$(a + \mathcal{E})(b + \mathcal{E})$	$\{\mathcal{E}, a, b, ab\}$
$b^*\emptyset$	$\emptyset$
$\emptyset^*$	$\{\mathcal{E}\}$, (ou seja, juntar qualquer número de cadeias, incluí juntar 0 cadeias, o que gera a cadeia vazia)

Exemplos de ER

Mais exemplos: considere $\Sigma = \{a, b\}$.

ER	**Linguagem gerada**
$(a + \mathcal{E})b^*$	$\{w \mid w = ab^* \text{ ou } w = b^*\}$
$(a + \mathcal{E})(b + \mathcal{E})$	$\{\mathcal{E}, a, b, ab\}$
$b^*\emptyset$	$\emptyset$
$\emptyset^*$	$\{\mathcal{E}\}$, (ou seja, juntar qualquer número de cadeias, incluí juntar 0 cadeias, o que gera a cadeia vazia)

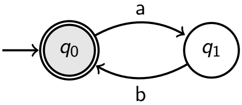
Roteiro

Equivalência entre **ER**s e AFN$_{\mathcal{E}}$s

Expressões Regulares e Autômatos Finitos (AFs) são *equivalentes* em seu poder descritivo.

- Esse fato pode ser surpreendente, já que **ER**s e AFs são aparentemente diferentes.
- Qualquer **ER** pode ser convertida em um AF e vice-versa.

$$(\mathbf{ab})^*$$



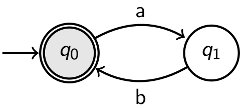
$$L = \{ab, abab, ababab, \ldots\}$$

Equivalência entre **ER**s e AFN$_\mathcal{E}$s

Expressões Regulares e Autômatos Finitos (AFs) são *equivalentes* em seu poder descritivo.

- Esse fato pode ser surpreendente, já que **ER**s e AFs são aparentemente diferentes.
- Qualquer **ER** pode ser convertida em um AF e vice-versa.

$(\mathbf{ab})^*$



$$L = \{ab, abab, ababab, \ldots\}$$

Equivalência entre **ER**s e AFN$_{\mathcal{E}}$s

Equivalência

1. **ER** $\rightarrow$ AFN$_{\mathcal{E}}$.
   - Vamos converter a **ER r** em um AFN$_{\mathcal{E}}$ N (vamos apresentar um procedimento).
2. AFD $\rightarrow$ **ER**.
   - Não iremos mostrar.

Equivalência entre **ER**s e AFN$_\mathcal{E}$s

Equivalência

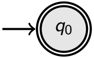
1. **ER** $\rightarrow$ AFN$_\mathcal{E}$.
   – Vamos converter a **ER r** em um AFN$_\mathcal{E}$ N (vamos apresentar um procedimento).
2. AFD $\rightarrow$ **ER**.
   – Não iremos mostrar.

Equivalência entre **ER**s e AFN$_{\mathcal{E}}$s

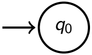
Vamos demonstrar por indução no número de operadores:

- Vamos definir 3 casos base para a **ER** **r** com zero operadores

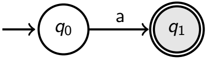
1. $\mathbf{r} = \mathcal{E}$



2. $\mathbf{r} = \emptyset$



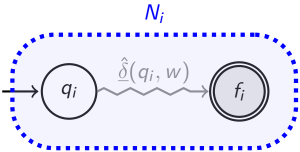
3. $\mathbf{r} = \mathbf{a}$



Equivalência entre **ER**s e AFN$_\mathcal{E}$s

Hipótese da Indução: para uma **ER** $\mathbf{r_i}$ com $n > 0$ operadores

- Vamos supor que o AFN$_\mathcal{E}$ N_i reconhece a linguagem $L(\mathbf{r_i})$, sendo representado por:



Equivalência entre **ER**s e AFN$_{\mathcal{E}}$s

Passo da Indução: para a **ER** **r** com $n+1$ operadores

- A **ER** **r** pode ser representada por:
  1. $\mathbf{r_1} + \mathbf{r_2}$
  2. $\mathbf{r_1}\mathbf{r_2}$
  3. $\mathbf{r_1^*}$
- $\mathbf{r_1}$ e $\mathbf{r_2}$ foram formados por no máximo n operadores

Pela HI: existem AFN$_{\mathcal{E}}$s N_1 e N_2 que aceitam $L(\mathbf{r_1})$ e $L(\mathbf{r_2})$

Equivalência entre **ER**s e AFN$_\mathcal{E}$s

Passo da Indução: para a **ER r** com $n+1$ operadores

- A **ER r** pode ser representada por:
  1. $\mathbf{r_1} + \mathbf{r_2}$
  2. $\mathbf{r_1 r_2}$
  3. $\mathbf{r_1^*}$
- $\mathbf{r_1}$ e $\mathbf{r_2}$ foram formados por no máximo n operadores

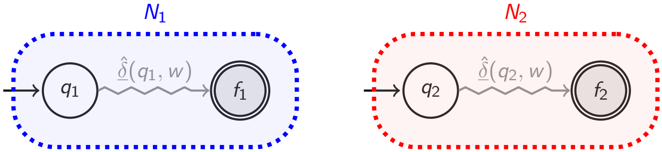
Pela HI: existem AFN$_\mathcal{E}$s N_1 e N_2 que aceitam $L(\mathbf{r_1})$ e $L(\mathbf{r_2})$

Equivalência entre **ER**s e AFN$_\mathcal{E}$s

Passo da Indução: para a **ER** $\mathbf{r}$ com $n+1$ operadores

- A **ER** $\mathbf{r}$ pode ser representada por:
  1. $\mathbf{r_1} + \mathbf{r_2}$
  2. $\mathbf{r_1 r_2}$
  3. $\mathbf{r_1^*}$
- $\mathbf{r_1}$ e $\mathbf{r_2}$ foram formados por no máximo n operadores

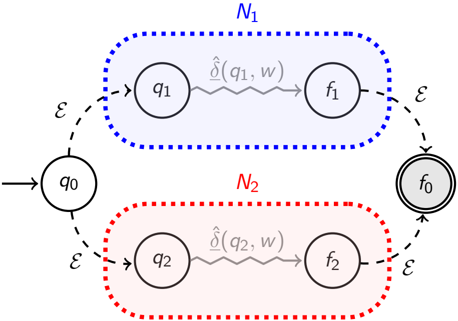
Pela HI: existem AFN$_\mathcal{E}$s N_1 e N_2 que aceitam $L(\mathbf{r_1})$ e $L(\mathbf{r_2})$



Equivalência entre **ER**s e AFN$_\mathcal{E}$s

Dessa forma:

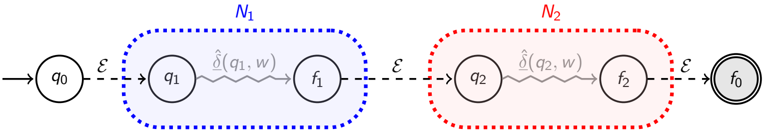
2. Se $\mathbf{r} = \mathbf{r_1} + \mathbf{r_2}$



Equivalência entre **ER**s e AFN$_{\mathcal{E}}$s

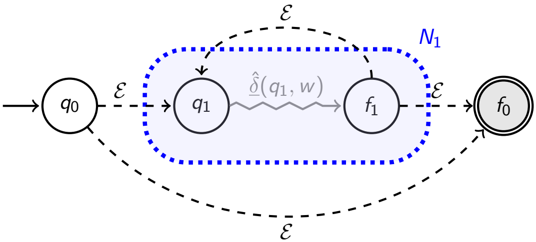
(continuação):

2. Se $\mathbf{r} = \mathbf{r_1 r_2}$



(continuação):

3. Se $\mathbf{r} = \mathbf{r_1^*}$



Equivalência entre **ER**s e AFN$_\mathcal{E}$s

Portanto, toda **ER r** pode ser convertida em um AFN$_\mathcal{E}$ N a partir de dos casos base utilizando os operadores das **ER**s.

$\square$

Exemplo

Seja $\mathbf{r1} = \mathbf{01}^* + \mathbf{1} = (\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$

- Vamos definir:

1. $\mathbf{r}_1 = \mathbf{0}$

2. $\mathbf{r}_2 = \mathbf{1}$

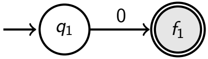
3. $\mathbf{r}_3 = \mathbf{1}^*$

Exemplo

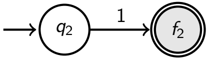
Seja $\mathbf{r1} = \mathbf{01}^* + \mathbf{1} = (\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$

- Vamos definir:

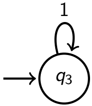
1. $\mathbf{r_1} = \mathbf{0}$



2. $\mathbf{r_2} = \mathbf{1}$



3. $\mathbf{r_3} = \mathbf{1}^*$



Exemplo

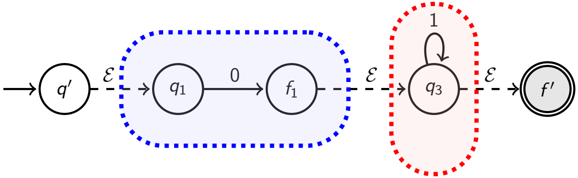
Em seguida, $\mathbf{r1} = \mathbf{01}^* + \mathbf{1} = (\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$

- Para:
  4. $(\mathbf{0}(\mathbf{1}^*))$

Exemplo

Em seguida, $\mathbf{r1} = \mathbf{01}^* + \mathbf{1} = (\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$

- Para:
  4. $(\mathbf{0}(\mathbf{1}^*))$



Exemplo

Por fim, $\mathbf{r1} = \mathbf{01}^* + \mathbf{1} = (\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$

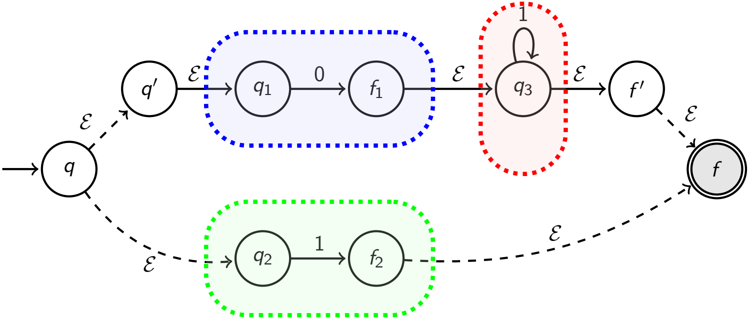
- Para:
  5. $(\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$

Exemplo

Por fim, $\mathbf{r1} = \mathbf{01}^* + \mathbf{1} = (\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$

- Para:

    5. $(\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$



Exemplo

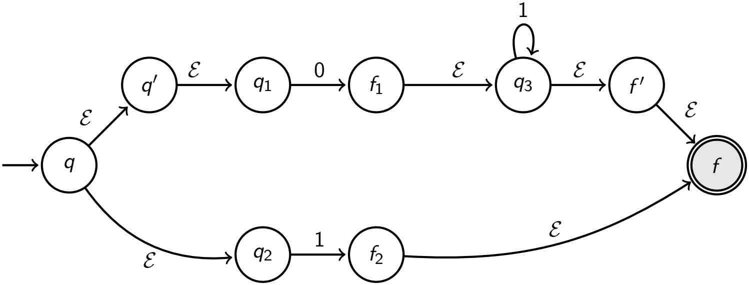
Resultado, $\mathbf{r1} = \mathbf{01}^* + \mathbf{1} = (\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$

- AFN$_\mathcal{E}$ equivalente:

Exemplo

Resultado, $\mathbf{r1} = \mathbf{01}^* + \mathbf{1} = (\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$

- AFN$_\mathcal{E}$ equivalente:



Equivalência entre **ER**s e AFN$_\mathcal{E}$s

Acabamos de mostrar que:

Teorema

Seja L_1 uma linguagem definida por uma **ER r**, então existe um AFN$_\mathcal{E}$ N, tal que

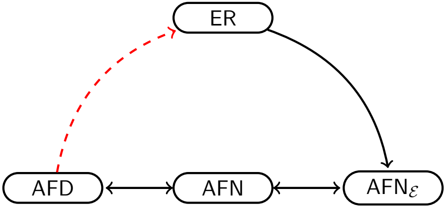
$$L(N) = L_1$$

Portanto, as linguagens são equivalentes e L_1 é uma **Linguagem Regular**.

Formalismos para as Linguagens Regulares

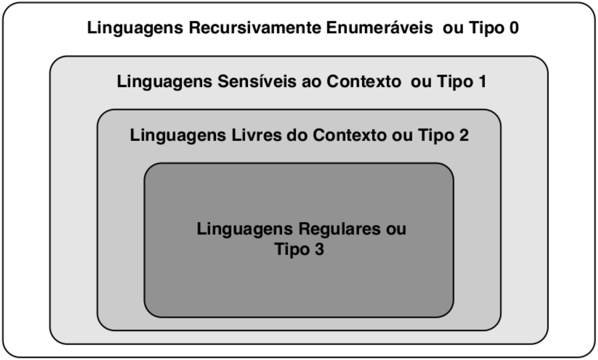
Vimos diferentes formalismos para reconhecer/gerar Linguagens Regulares:

- Todos eles são equivalentes



Hierarquia de Chomsky

Relembrando a Hierárquia de Chomsky:



Quais são as linguagens que não são regulares?

Formalismos para as Linguagens Regulares

Quais linguagens não são regulares?

$$L_1 = \{w \mid w = ab^n \text{ para } n \geq 0\}$$
$$L_2 = \{w \mid w = a^n b \text{ para } n \geq 0\}$$
$$L_3 = \{w \mid w = a^n b^n \text{ para } n \geq 0\}$$

Vamos ver melhor esse assunto na próxima aula ...

Formalismos para as Linguagens Regulares

Quais linguagens não são regulares?

$$L_1 = \{w \mid w = ab^n \text{ para } n \geq 0\}$$
$$L_2 = \{w \mid w = a^n b \text{ para } n \geq 0\}$$
$$L_3 = \{w \mid w = a^n b^n \text{ para } n \geq 0\}$$

Vamos ver melhor esse assunto na próxima aula ...

Fim

Dúvidas?

Roteiro

Referências

Referências:

1. *“Introdução à Teoria da Computação”* de M. Sipser, 2007.
2. *“Introdução à Teoria de Autômatos, Linguagens e Computação”* de J. E. Hopcroft, R. Motwani, e J. D. Ullman, 2003.
3. Materiais adaptados dos slides do Prof. Evandro E. S. Ruiz, da USP.