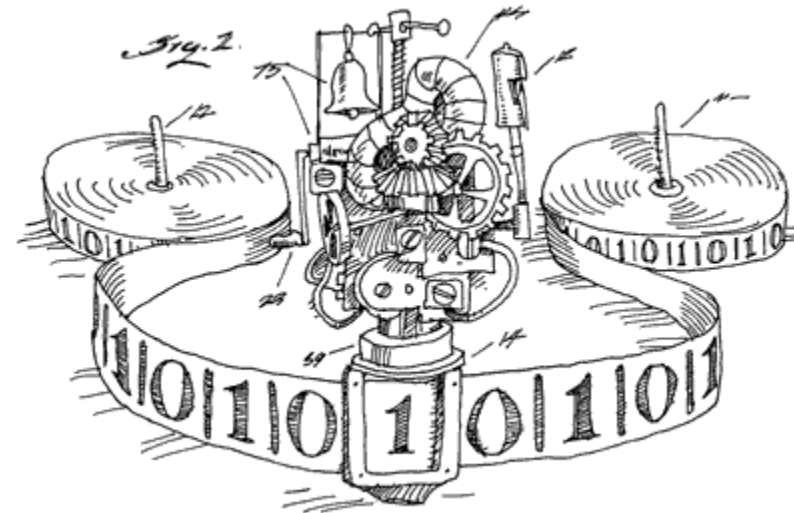


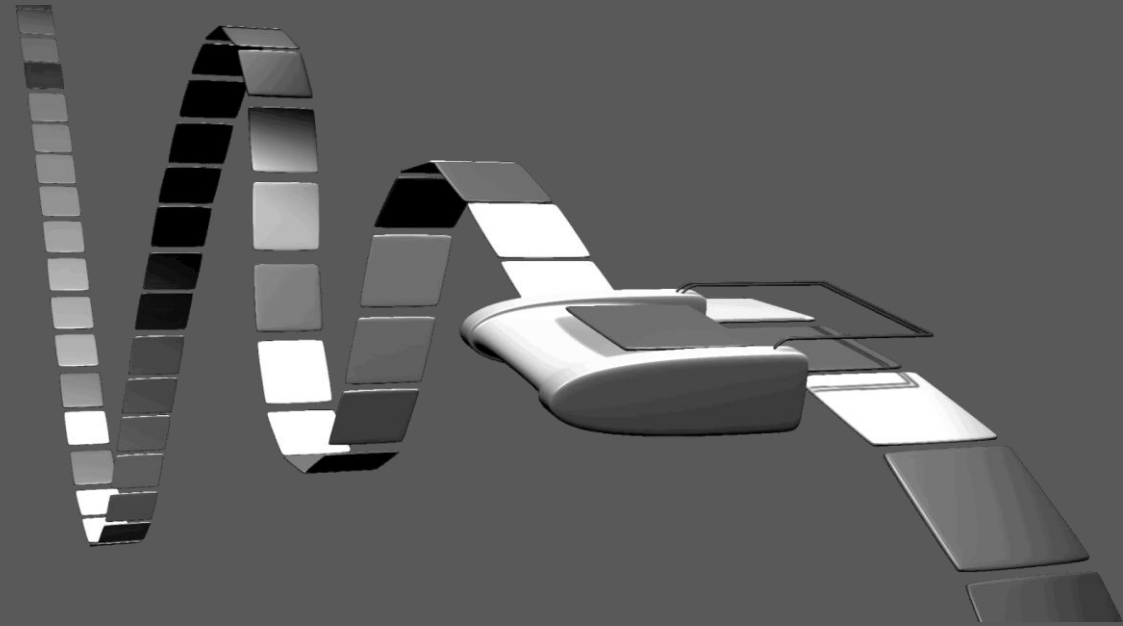
Equivalência entre expressões regulares e ε -AFNs

Introdução à Teoria da Computação

Nicollas Mocelin Sdroievski



Relembrando

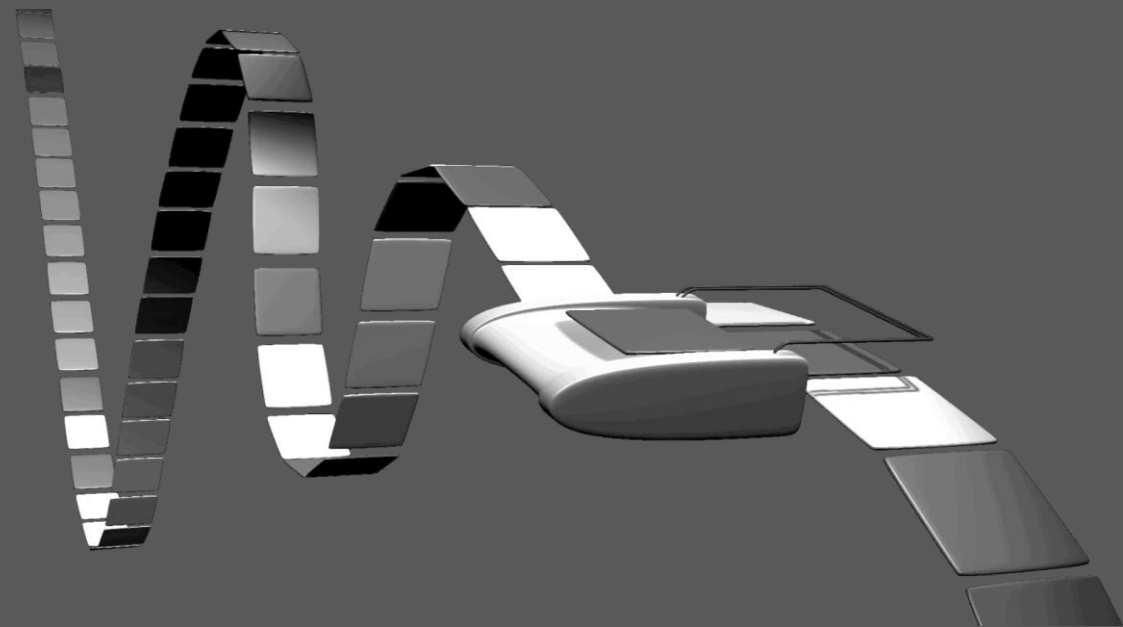


Expressões regulares

- Expressões como $(0 + 1)^*1(0 + 1)(0 + 1)$
 - Fecho
 - Concatenação
 - União

Exercício 3.27 Seja $\Sigma = \{a, b\}$. Apresente a expressão regular para a linguagem das strings sobre Σ que tem tamanho ímpar. ■

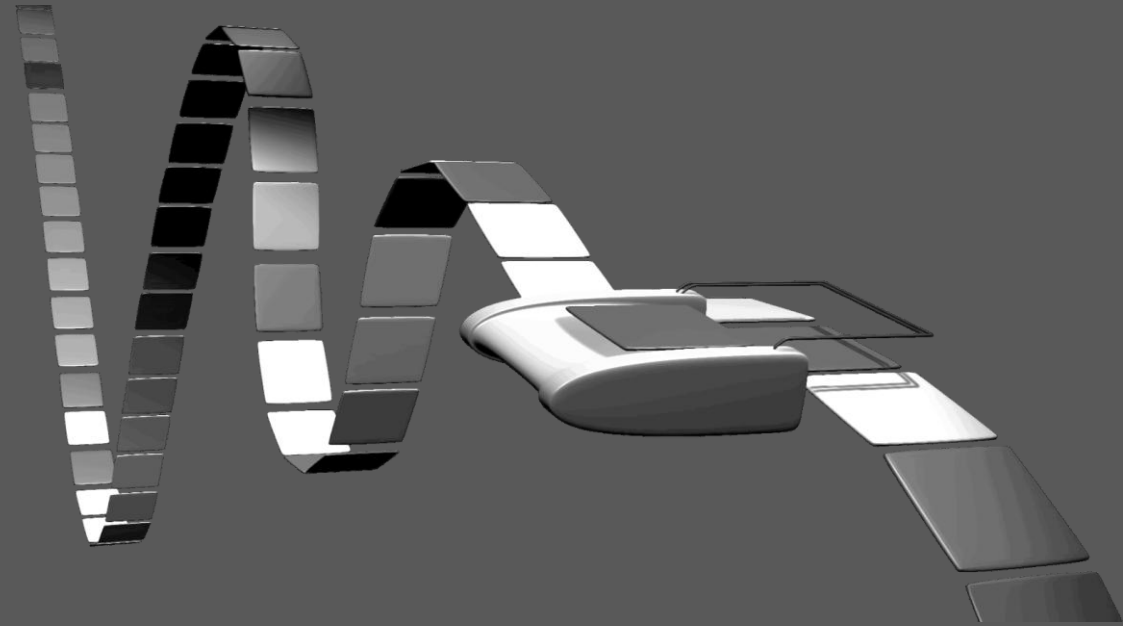
Equivalência



Resultado de hoje

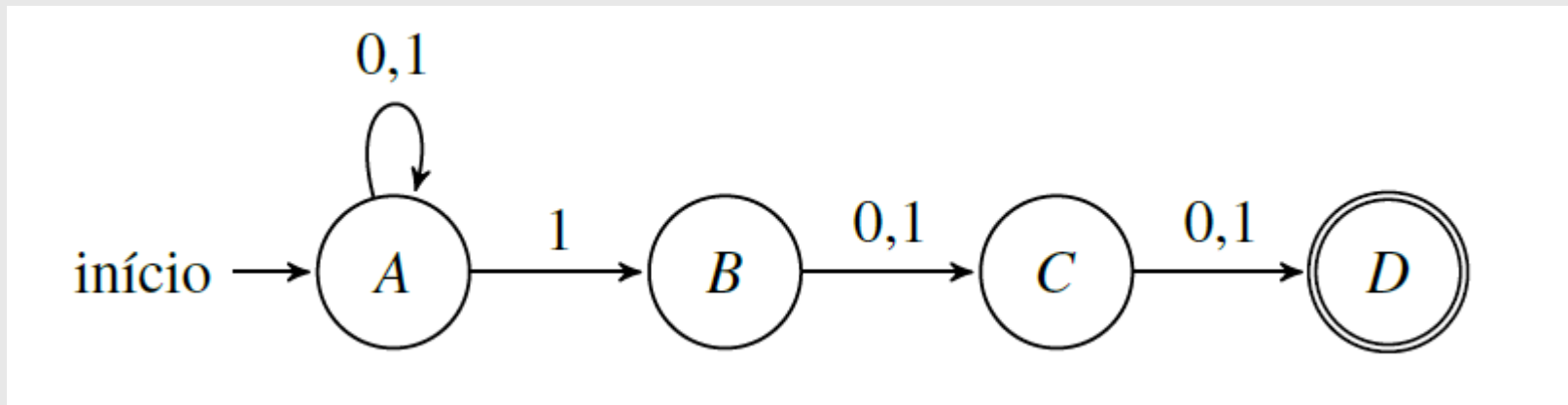
- **Teorema.** Uma linguagem L é regular se e somente se existe uma expressão regular R tal que $L = L(R)$.
- Para demonstrar o teorema, precisamos de duas afirmações
 1. Se L é uma linguagem regular, então existe uma expressão regular R tal que $L = L(R)$
 2. Se R é uma expressão regular qualquer, então $L(R)$ é uma linguagem regular

De autômatos
para ERs



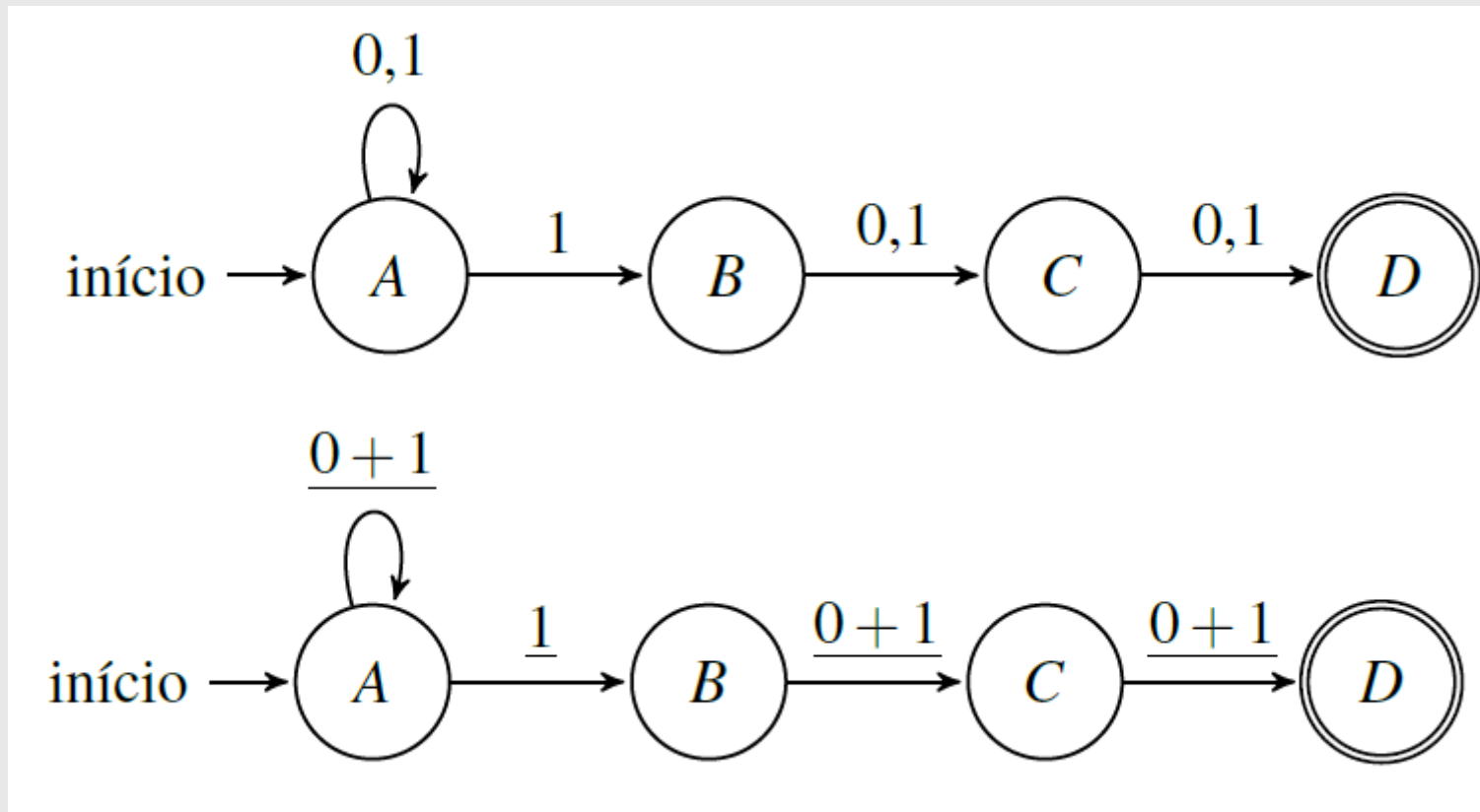
De autômatos para ERs

- **Teorema.** Seja $L(E)$ a linguagem de um ε -AFN E qualquer. Então existe uma expressão regular R tal que $L(R) = L(E)$.
- Vamos construir uma expressão regular a partir do autômato
- **Exemplo**



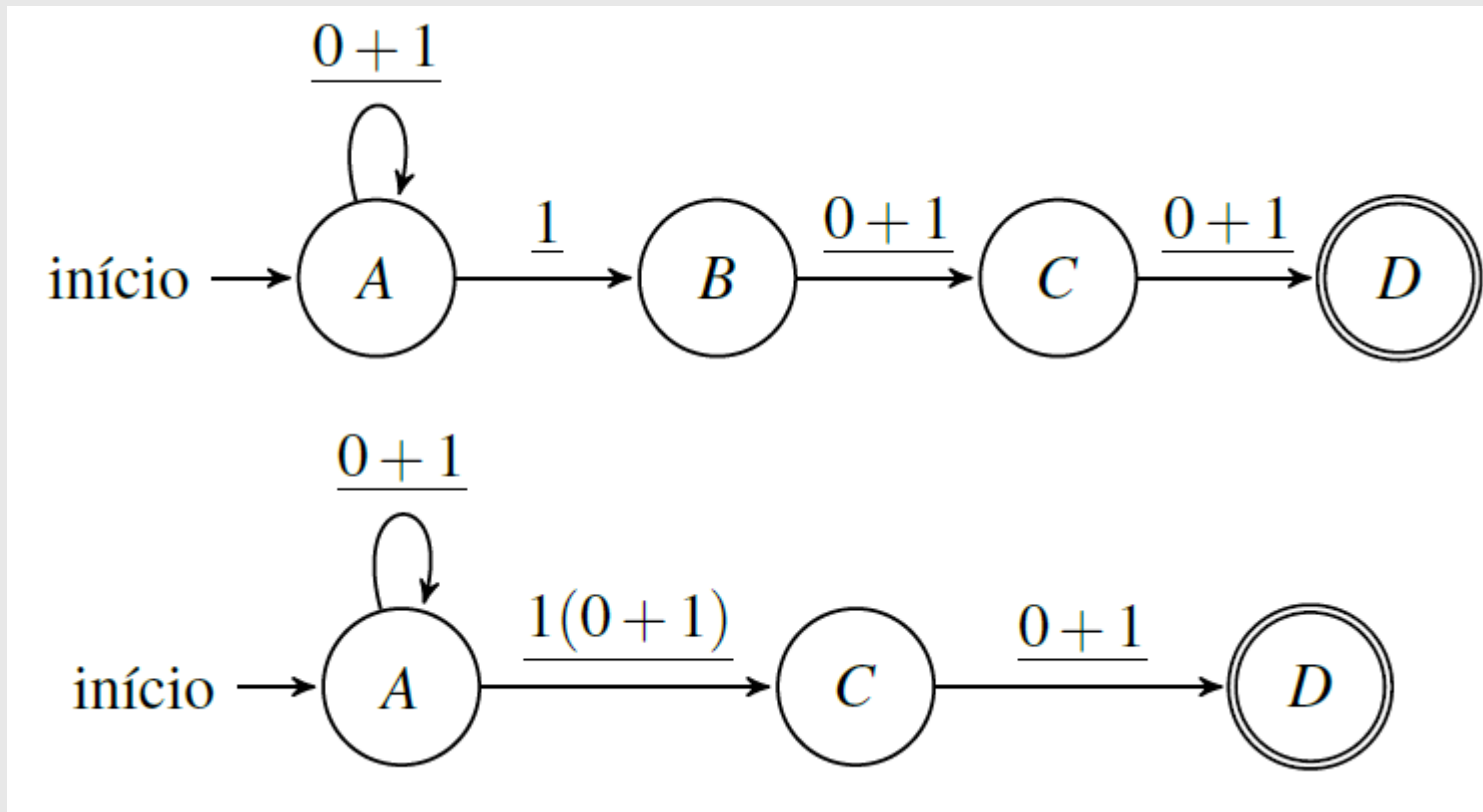
Autômato generalizado

- Primeiro passo: construir um autômato generalizado
 - ERs no lugar dos símbolos nas transições

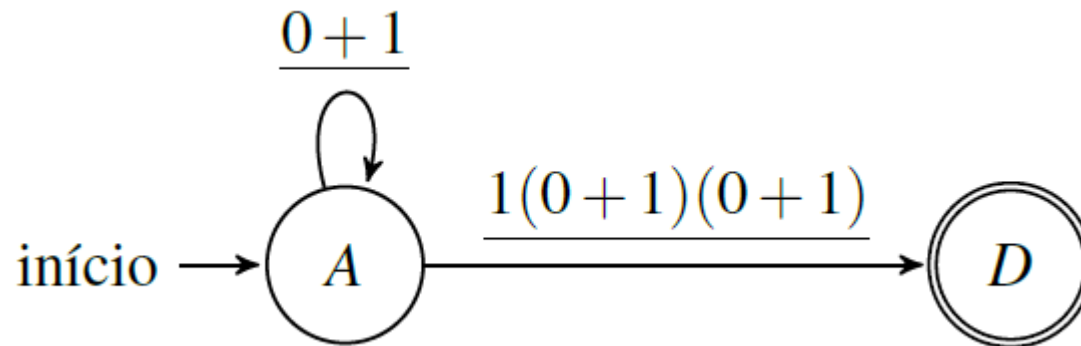
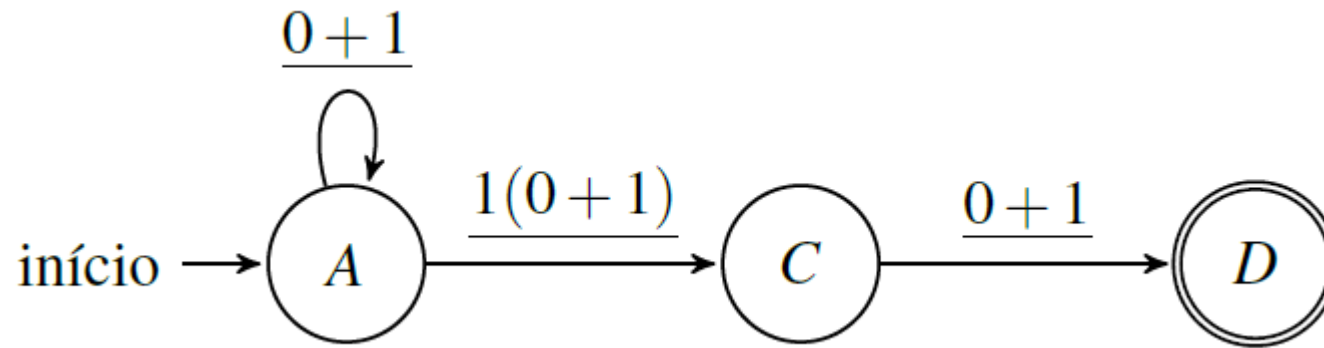


Remoção de estados

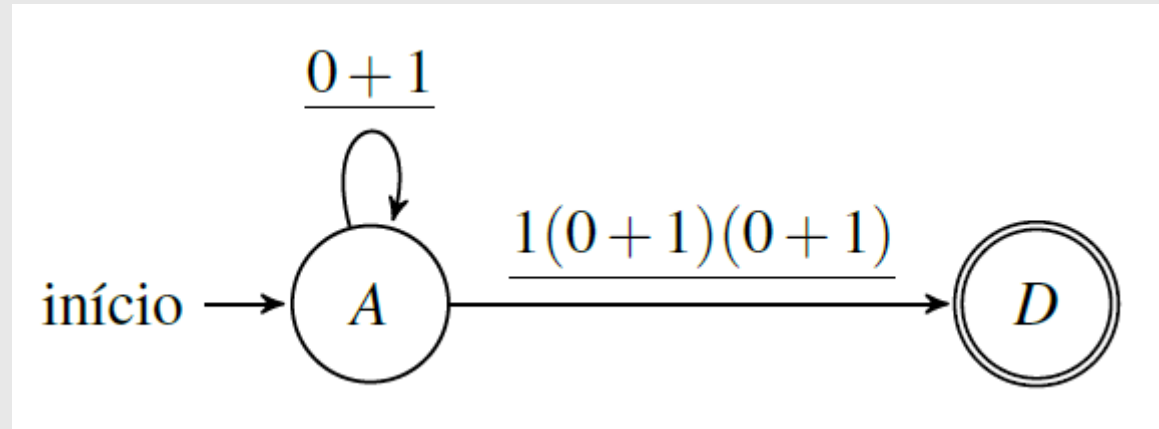
- Próximo passo: remover estados, transformando-os em expressões regulares



Mais uma remoção



Autômato final e expressão regular

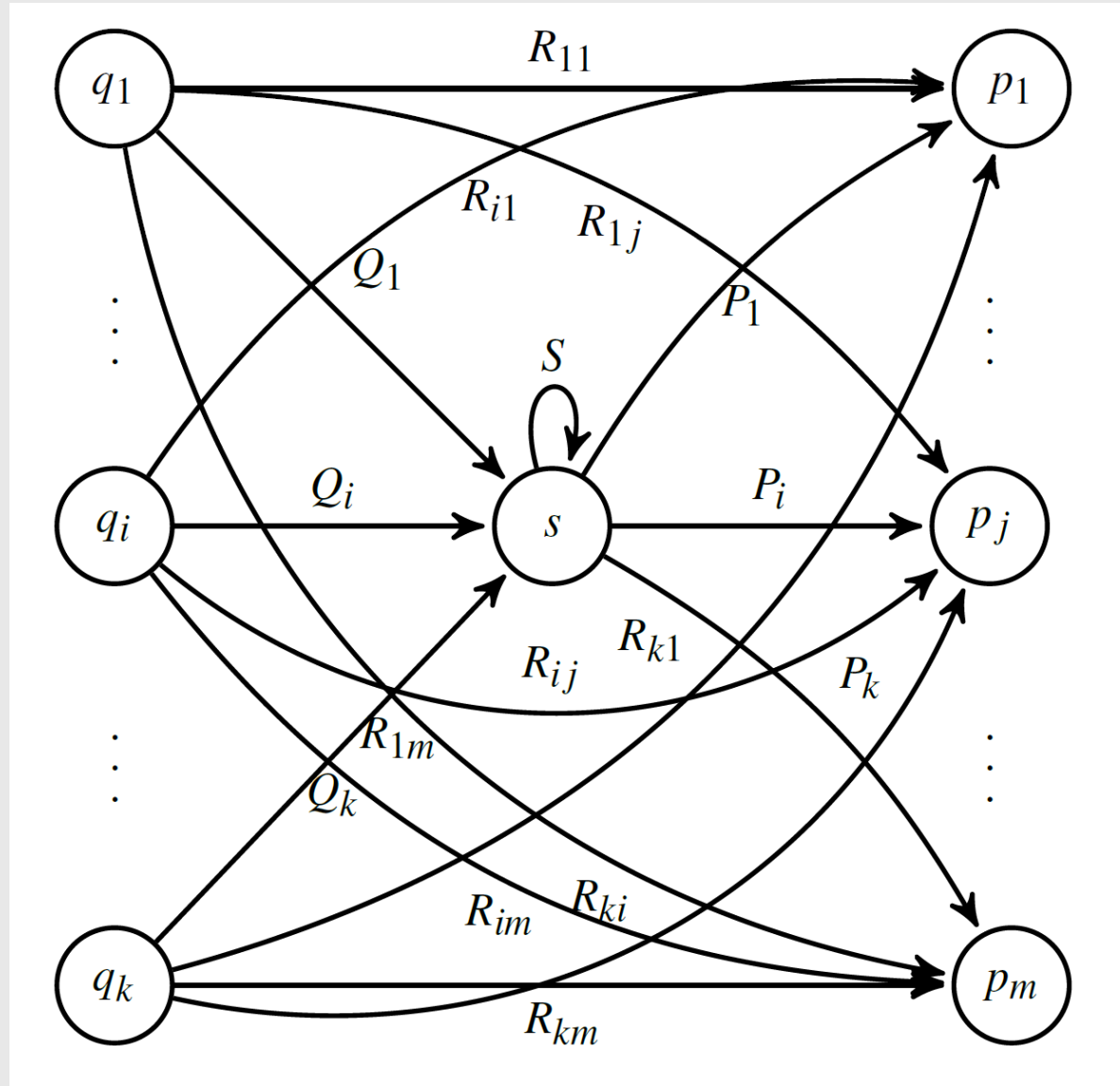


- Expressão regular para a linguagem deste autômato?
- $(0 + 1)^* 1(0 + 1)(0 + 1)$

Caso geral

- Ao remover um estado s do autômato, precisamos observar todo par de estados (q_i, p_j) tal que $q_i \rightarrow s$ e $s \rightarrow p_j$ são transições do autômato
- Suponha que há as seguintes transições
 - Com rótulo/expressão R_{ij} de q_i para p_j
 - Com rótulo/expressão Q_i de q_i para s
 - Com rótulo/expressão S de s para s (loop)
 - Com rótulo/expressão P_i de s para p_j
- A transição final entre q_i e p_j terá o rótulo $R_{ij} + Q_i S^* P_j$

Caso geral: figura



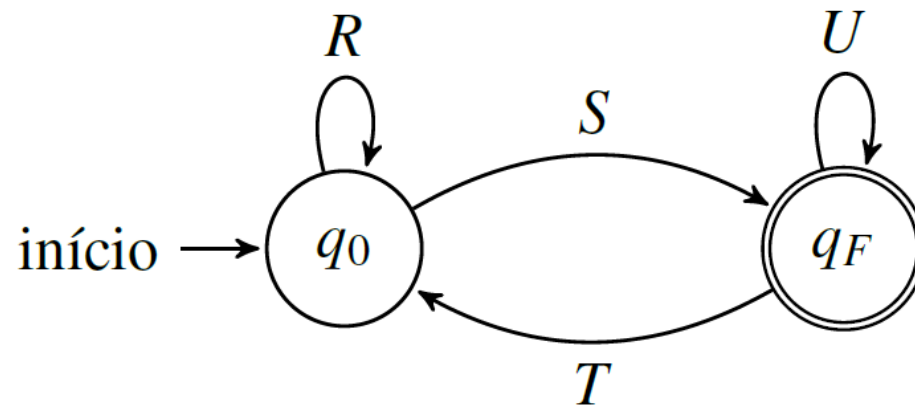
Algoritmos

- Algoritmos para casos diferentes
 1. Há apenas um estado final, que não é estado inicial
 2. Há apenas um estado final, que é também estado inicial
 3. Há vários estados finais (caso geral)

Primeiro caso

Caso 1: Algoritmo para obter ER a partir de um ε -AFN $(Q, \Sigma, \delta, q_0, \{q_F\})$, com $q_0 \neq q_F$:

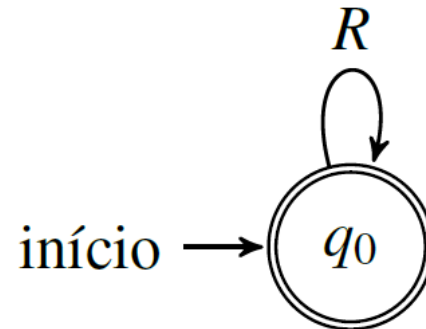
1. Troque os rótulos das transições de E por expressões regulares equivalentes de forma que agora E seja um autômato generalizado
2. A cada passo remova um estado $q \in Q \setminus \{q_0, q_F\}$ e atualize as transições e as expressões regulares dos rótulos do autômato
3. Quando restar apenas o estado inicial q_0 e o estado final q_F , como o da figura abaixo, a expressão regular final é $(R + SU^*T)^*SU^*$



Segundo caso

Caso 2: Algoritmo para obter ER a partir de ε -AFN $(Q, \Sigma, \delta, q_0, F)$ com $|F| = 1$ e $q_0 \in F$:

1. Troque os rótulos das transições de E por expressões regulares equivalentes de forma que agora E seja um autômato generalizado
2. A cada passo remova um estado $q \in Q \setminus \{q_0\}$ e atualize as transições e as expressões regulares dos rótulos do autômato
3. Quando restar apenas o estado inicial (e final) q_0 , como o da figura abaixo, a expressão regular final é R^* .



Ideia do caso geral

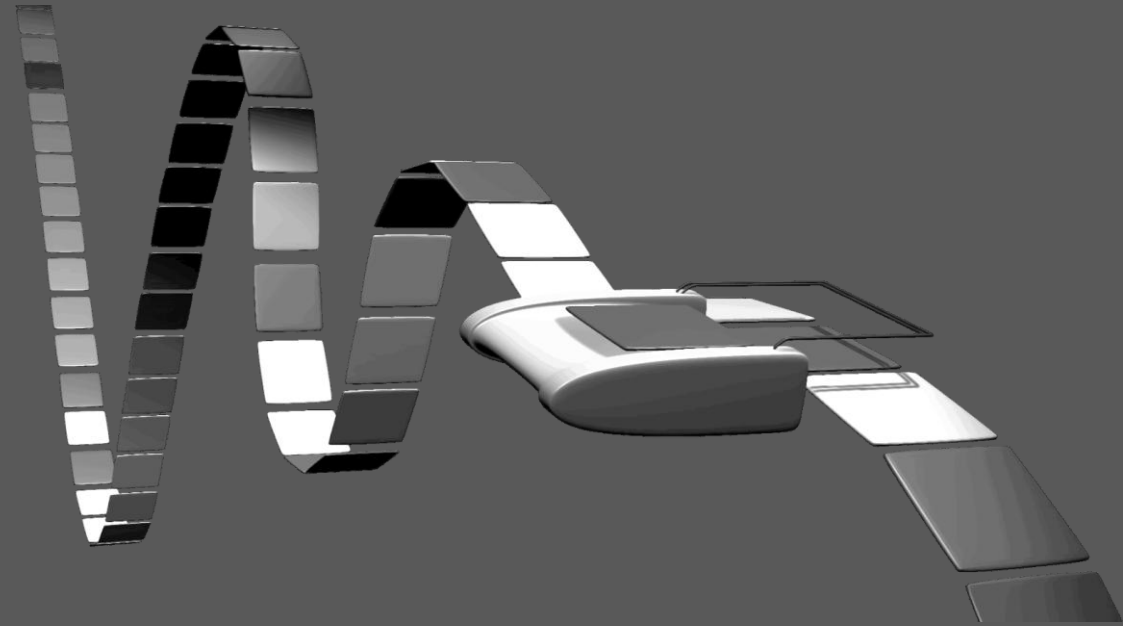
- Seja $E = (Q, \Sigma, \delta, q_0, F = \{f_1, f_2, \dots, f_k\})$ um ε -AFN
- Tome uma string $w \in L(E)$ e considere o estado f_i (com i mínimo) que é atingido pelo autômato, e construa um autômato que tem somente esse estado final
 - $E_i = (Q, \Sigma, \delta, q_0, \{f_i\})$
 - Esse autômato também aceita w
 - E não aceita nenhuma string que não seja aceita por E
- Ideia no caso geral: considerar um estado final por vez, e fazer a união (+) das expressões obtidas

Caso geral

Caso geral: Algoritmo para obter ER a partir de ε -AFN $E = (Q, \Sigma, \delta, q_0, F)$ qualquer

- A partir de E , crie k autômatos $E_1, E_2, \dots, E_k$, onde E_i é o o autômato $(Q, \Sigma, \delta, q_0, \{q_i\})$
- Encontre as expressões regulares $R_1, R_2, \dots, R_k$ para cada autômato E_i usando os algoritmos para os casos especiais de autômatos com apenas um estado final
- A partir daí, e ER para E é $R_1 + R_2 + \dots + R_k$

De ERs para
autômatos

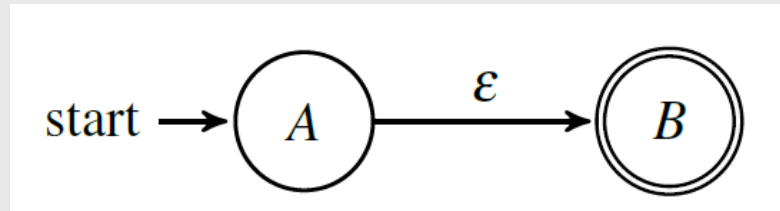


De ERs para autômatos

- **Teorema.** Seja $L(R)$ a linguagem representada por uma expressão regular R qualquer. Então existe um ε -AFN E tal que $L(E) = L(R)$.
- Assim como antes, demonstração construtiva
- Usaremos a própria definição recursiva de um ER para construir um ε -AFN equivalente.

ERs elementares (casos base)

- Se R é uma expressão regular elementar, então tem umas das seguintes formas
 - $R = \underline{\varepsilon}$, então o autômato abaixo aceita a linguagem $L(\underline{\varepsilon}) = \{\varepsilon\}$.

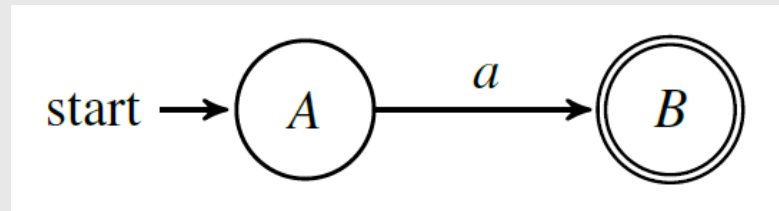


- $R = \underline{\emptyset}$, então o autômato abaixo aceita a linguagem $L(\underline{\emptyset}) = \emptyset$



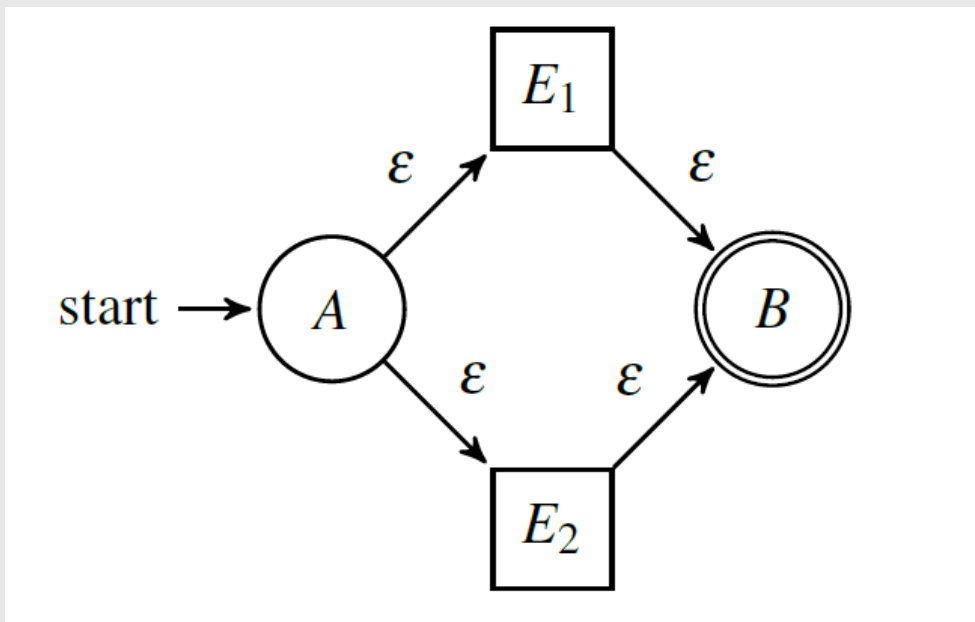
ERs elementares (casos base)

- Se R é uma expressão regular elementar, então tem umas das seguintes formas
 - Se $R = \underline{a}$, então o autômato abaixo aceita a linguagem $L(\underline{a}) = \{a\}$



ERs compostas (caso recursivo)

- **Caso 1.** Suponha que $R = R_1 + R_2$ e que os autômatos E_1 e E_2 aceitam $L(R_1)$ e $L(R_2)$, respectivamente
- O diagrama abaixo representa um autômato que aceita R

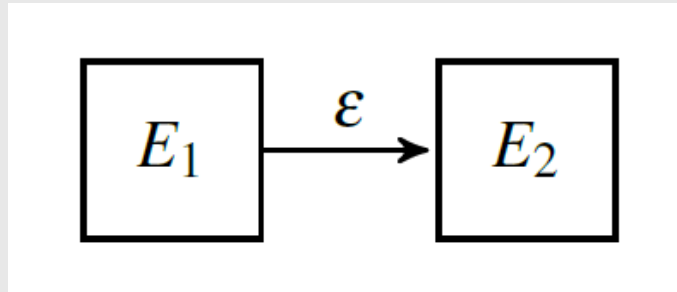


Observações

- As transições que saem de A vão para os antigos estados iniciais de E_1 e E_2
- As transições que entram em B saem dos antigos estados finais de E_1 e E_2

ERs compostas (continuação)

- **Caso 2.** Suponha que $R = R_1R_2$ e que os autômatos E_1 e E_2 aceitam $L(R_1)$ e $L(R_2)$, respectivamente
- O diagrama abaixo representa um autômato que aceita R

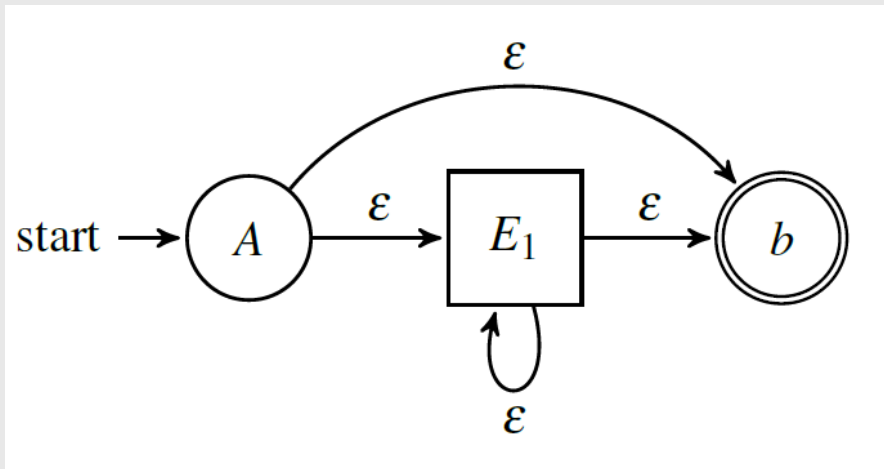


Observações

- A transição de E_1 para E_2 sai do antigo estado final de E_1 e entra no estado inicial de E_2

ERs compostas (continuação)

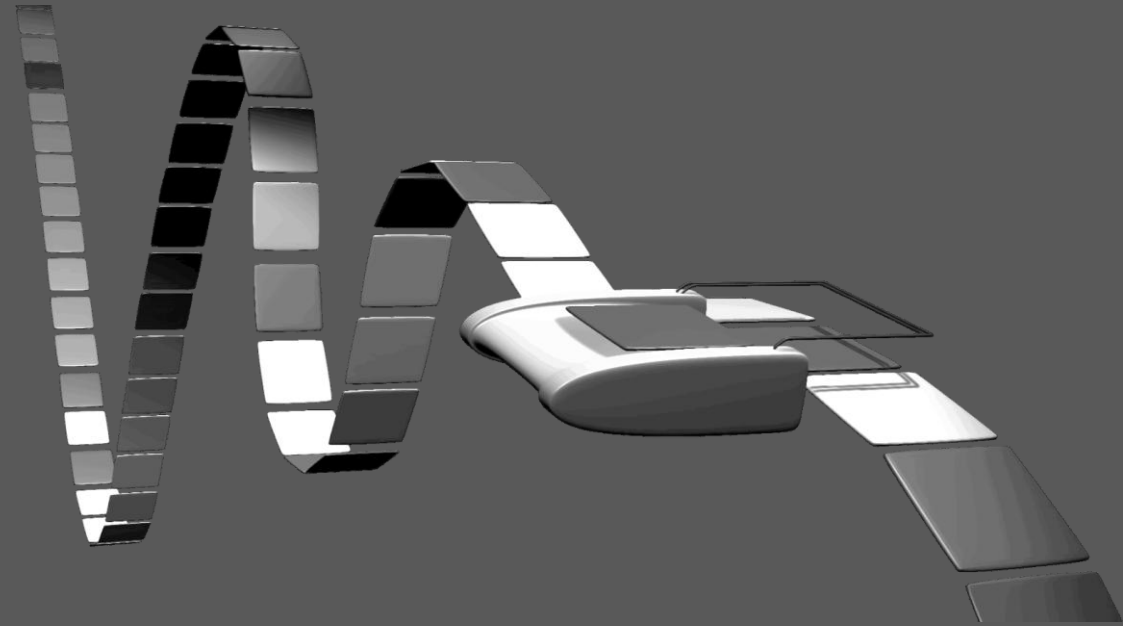
- **Caso 3.** Suponha que $R = R_1^*$ e que o autômato E_1 aceita $L(R_1)$
- O diagrama abaixo descreve o autômato que aceita R



Observações

- A transição de A para E_1 entra no antigo estado inicial de E_1
- A transição de E_1 para si mesmo sai de seu antigo estado final para seu antigo estado inicial
- A transição de E_1 para B sai do antigo estado final de E_1

Exercícios



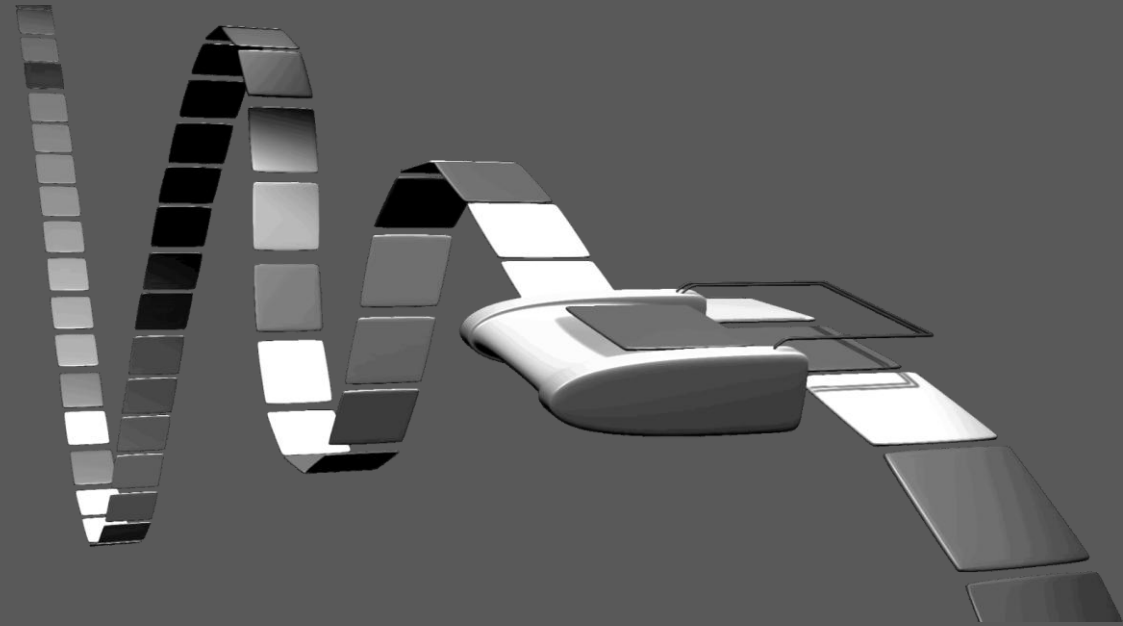
Exercícios

Exercício 3.35 Dado o AFD abaixo, encontre uma expressão regular equivalente. Você deve mostrar o desenvolvimento passo a passo de solução.

	0	1
$\rightarrow^* p$	s	p
q	p	s
r	r	q
s	q	r

Exercício 3.30 (Resolvido) Apresente um autômato para aceitar a linguagem de $R = \underline{(0 + 1)^*}$.

Para fechar



Para fechar

- Hoje
 - Equivalência entre ε -AFNs e ERs
 - AFDs, AFNs, ε -AFNs e ERs caracterizam linguagens regulares
- Próxima aula
 - Além das linguagens regulares
 - Lema do bombeamento

Referências

- [SIL25] – SILVA, M.V.G. Autômatos, Computabilidade e Complexidade Computacional , 2025.