

Linguagens Formais e Autômatos

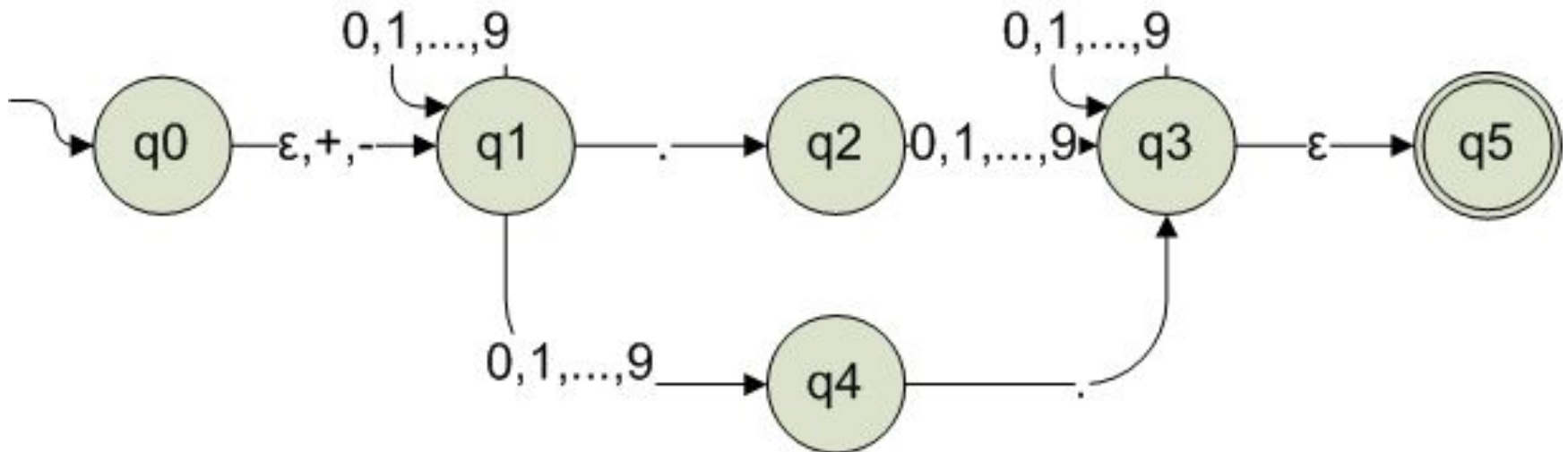
Aula 06 - NFA com transições vazias

Referências bibliográficas

- **Introdução à teoria dos autômatos, linguagens e computação / John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman** ; tradução da 2.ed. original de Vandenberg D. de Souza. - Rio de Janeiro : Elsevier, 2002 (Tradução de: Introduction to automata theory, languages, and computation - ISBN 85-352-1072-5)
 - Capítulo 2 - Seção 2.5
- **Introdução à teoria da computação / Michael Sipser** ; tradução técnica Ruy José Guerra Barretto de Queiroz ; revisão técnica Newton José Vieira. -- São Paulo : Thomson Learning, 2007 (Título original : Introduction to the theory of computation. "Tradução da segunda edição norte-americana" - ISBN 978-85-221-0499-4)
 - Capítulo 1 - Seção 1.2

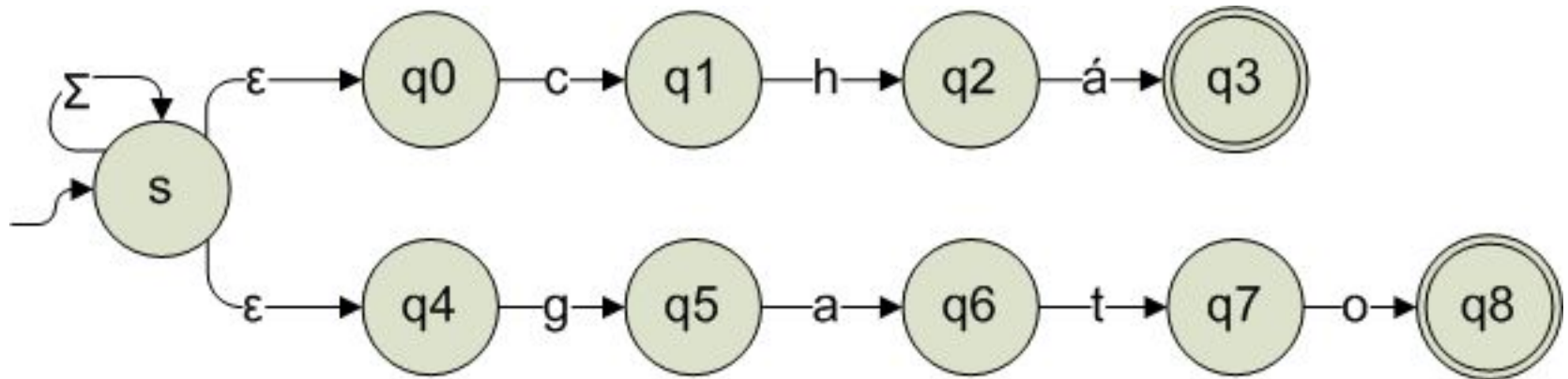
NFA com transições vazias

- Transições espontâneas
 - Isto é, sem nenhuma entrada
- É uma forma de não-determinismo
 - Facilita a “programação”
- Ex: números decimais



NFA com transições vazias

- Ex: busca por palavras-chave



NFA com transições vazias

- Definição formal
 - A mesma que NFA
 - Muda somente a função de transição
 - $\delta: Q \times \Sigma \cup \{\epsilon\} \rightarrow Q^*$
 - Uma coluna extra na tabela, ou transições vazias no diagrama

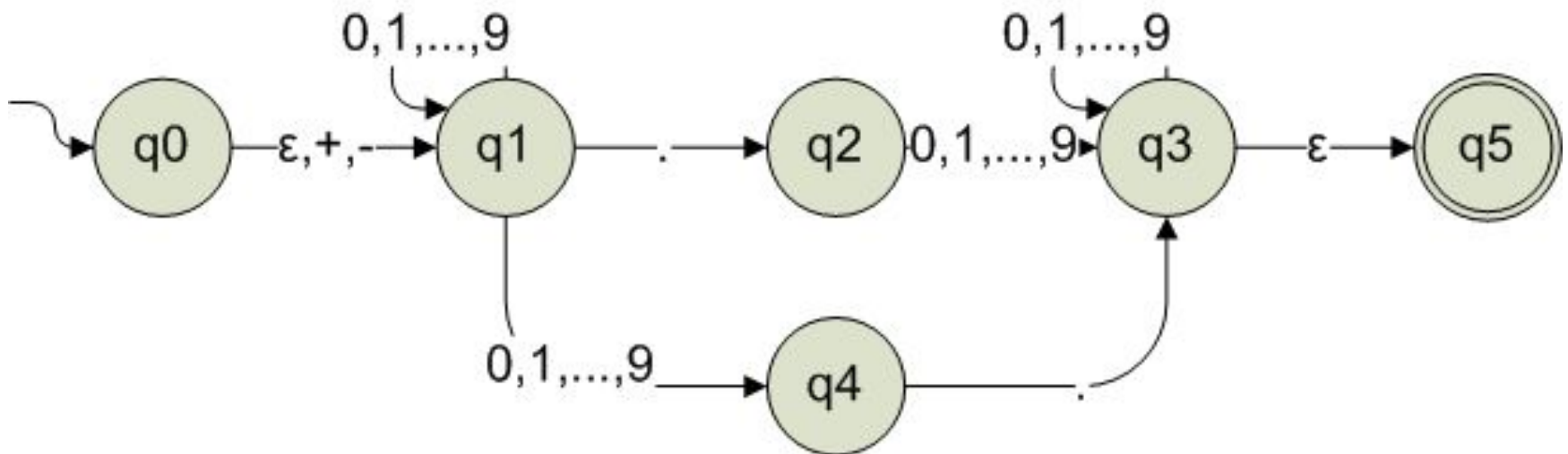
	ϵ	$+, -$	$.$	$0, 1, \dots, 9$
$\rightarrow q_0$	$\{q_1\}$	$\{q_1\}$	\emptyset	\emptyset
q_1	\emptyset	\emptyset	$\{q_2\}$	$\{q_1, q_4\}$
q_2	\emptyset	\emptyset	\emptyset	$\{q_3\}$
q_3	$\{q_5\}$	\emptyset	\emptyset	$\{q_3\}$
q_4	\emptyset	\emptyset	$\{q_3\}$	\emptyset
$* q_5$	\emptyset	\emptyset	\emptyset	\emptyset

NFA com transições vazias

- Essa nova característica não aumenta o poder do NFA
 - Ainda reconhece linguagens regulares
- Conceito de épsilon-fechamento
 - $\text{ECLOSE}(q)$
 - Conjunto de todos os estados alcançáveis espontaneamente a partir de q
 - Incluindo os vizinhos diretos e indiretos
 - Analisando-se os arcos rotulados com ϵ
- Função de transição estendida
 - Deve considerar sempre o ECLOSE

Interpretando ϵ -NFAs

- Dado o seguinte autômato (que reconhece números decimais)



Interpretando ϵ -NFAs

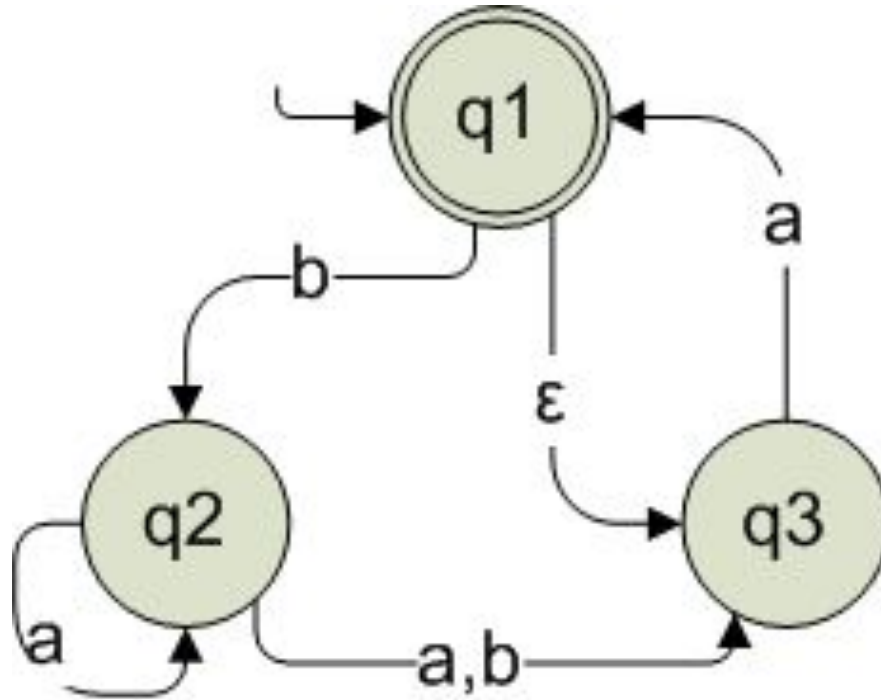
- Calcule a função estendida e mostre, passo a passo, as configurações instantâneas para a seguinte cadeia:
 - 5.6
 - 67.1.2
 - (Use notação de árvore ou conjuntos)
- Passo 1: calcule $ECLOSE(q_0)$
- Passos seguintes: mesmo que NFA
 - Porém, para cada estado alcançado, inclua o seu $ECLOSE$ também

Interpretando ϵ -NFAs

- Resposta:
 - $\{q_0, q_1\}5.6$
 - $5\{q_1, q_4\}.6$
 - $5.\{q_2, q_3, q_5\}6$
 - $5.6\{q_3, q_5\}$
- Conjunto de estados finais contém q_5 , portanto o autômato aceita a cadeia

Interpretando ϵ -NFAs

- Dado o seguinte autômato

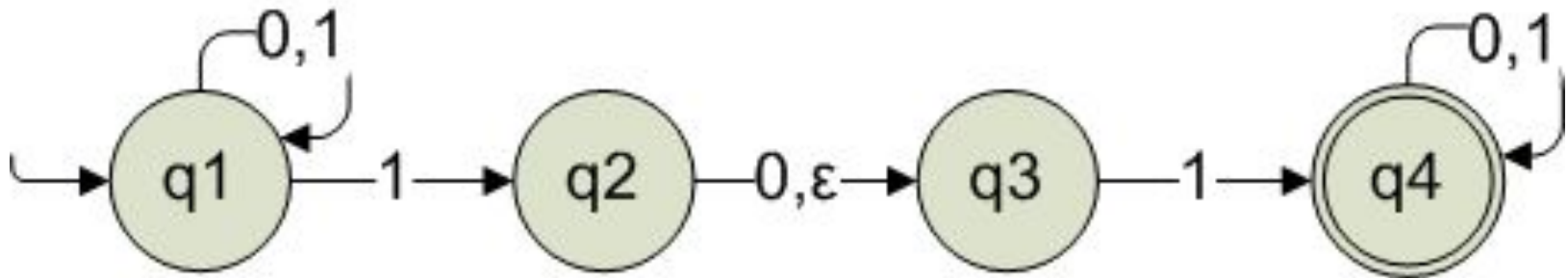


Interpretando ϵ -NFAs

- Calcule a função estendida e mostre, passo a passo, as configurações instantâneas para as seguintes cadeias:
 - ϵ
 - a
 - baba
 - baa
 - b
 - bb
 - babba
 - (Use notação de árvore ou conjuntos)

Interpretando ϵ -NFAs

- Dado o seguinte autômato

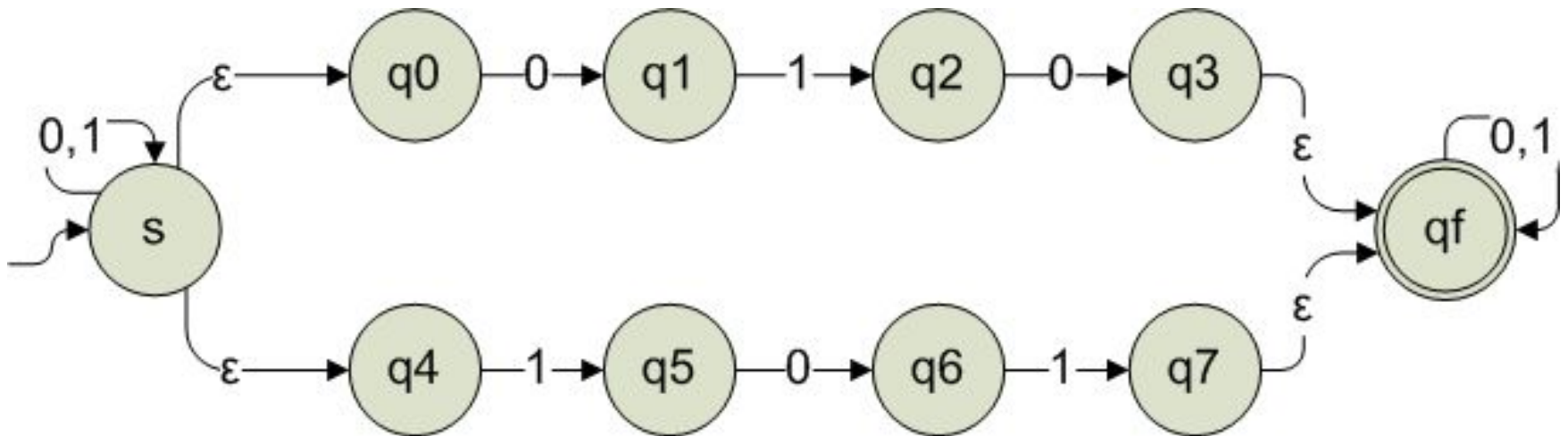


Interpretando ϵ -NFAs

- Calcule a função estendida e mostre, passo a passo, as configurações instantâneas para as seguintes cadeias:
 - 1111
 - 11
 - 101
 - 00010
 - (Use notação de árvore ou conjuntos)

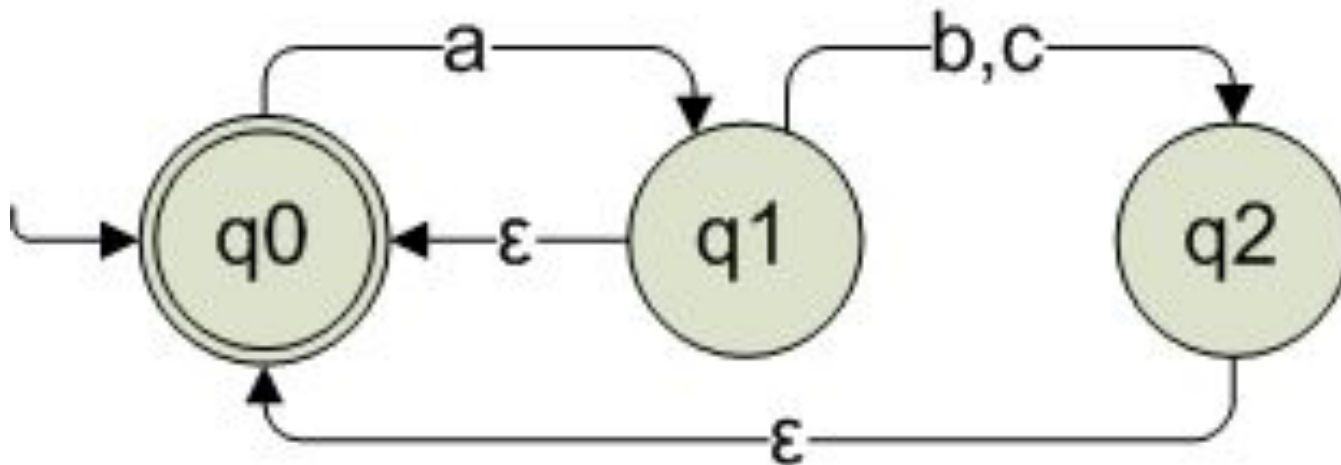
Projetando ϵ -NFAs

- Ex:
 - $\Sigma = \{0,1\}$
 - Linguagem = cadeias que contém a sequência 010 ou 101 como subcadeia



Projetando ϵ -NFAs

- Ex:
 - $\Sigma = \{a,b,c\}$
 - Linguagem = cadeias onde b e c sempre aparecem depois de uma ocorrência de a



Equivalência ϵ -NFAs e DFAs

- Transições vazias não adicionam poder ao autômato
 - Ainda reconhece as mesmas linguagens
 - Linguagens regulares
- Teorema:
 - Uma Linguagem L é aceita por algum DFA se e somente se L é aceita por algum ϵ -NFA
 - Prova por construção dos dois lados:
 - “Se”: um processo que constrói um DFA a partir de um ϵ -NFA
 - “Somente se”: um processo que constrói um ϵ -NFA a partir de um DFA

Conversão ϵ -NFA \rightarrow DFA

- É o mesmo procedimento da construção de subconjuntos dado anteriormente
 - Porém incorporando o cálculo de ϵ -fechamento após cada passo
 - Similar à implementação do ϵ -NFA

Conversão ϵ -NFA \rightarrow DFA

- Passo a passo com exemplo
- Dado o NFA:

	ϵ	a	b	c
\rightarrow p	\emptyset	{p}	{q}	{r}
q	{p}	{q}	{r}	\emptyset
* r	{q}	{r}	\emptyset	{p}

Conversão ϵ -NFA \rightarrow DFA

- Passo 1 (auxiliar): calcule o ECLOSE de todos os estados
- $\text{ECLOSE}(p) = \{p\}$
- $\text{ECLOSE}(q) = \{p, q\}$
- $\text{ECLOSE}(r) = \{p, q, r\}$

	ϵ	a	b	c
$\rightarrow p$	\emptyset	$\{p\}$	$\{q\}$	$\{r\}$
q	$\{p\}$	$\{q\}$	$\{r\}$	\emptyset
* r	$\{q\}$	$\{r\}$	\emptyset	$\{p\}$

Conversão ϵ -NFA \rightarrow DFA

- Passo 2: faça uma tabela “vazia” com as entradas (sem a coluna ϵ)

[illegible]

Conversão ϵ -NFA \rightarrow DFA

- Passo 3: Crie um novo estado inicial no DFA, um conjunto que contém o ECLOSE do estado inicial do NFA

[illegible]

Conversão ϵ -NFA \rightarrow DFA

- Passo 4:

- Para cada entrada, insira no DFA um conjunto que contém o ECLOSE da união de todos os resultados da transição NFA daquela entrada para todos os estados do conjunto à esquerda

[illegible]

Conversão ϵ -NFA \rightarrow DFA

- Passo 5:
 - Para cada novo conjunto de estados que aparecer, insira uma nova linha na tabela do DFA e volte para o passo 4

	a	b	c
$\rightarrow \{p\}$	{p}	{p,q}	{p,q,r}
{p,q}	{p,q}	{p,q,r}	{p,q,r}
{p,q,r}	{p,q,r}	{p,q,r}	{p,q,r}

Conversão ϵ -NFA \rightarrow DFA

- Passo 6: Quando não houver mais novos estados, marque como estado de aceitação os conjuntos que contém ao menos um estado de aceitação do NFA

	a	b	c
$\rightarrow \{p\}$	$\{p\}$	$\{p,q\}$	$\{p,q,r\}$
$\{p,q\}$	$\{p,q\}$	$\{p,q,r\}$	$\{p,q,r\}$
$* \{p,q,r\}$	$\{p,q,r\}$	$\{p,q,r\}$	$\{p,q,r\}$

Conversão ϵ -NFA \rightarrow DFA

- Passo 7: “Renomeie” os conjuntos para estados, de forma a facilitar a leitura do DFA

	a	b	c
\rightarrow A	A	B	C
B	B	C	C
* C	C	C	C

Conversão ϵ -NFA \rightarrow DFA

- Dado o seguinte ϵ -NFA
 - Converta para um DFA que aceita a mesma linguagem

	ϵ	a	b	c
$\rightarrow p$	{q,r}	\emptyset	{q}	{r}
q	\emptyset	{p}	{r}	{p,q}
* r	\emptyset	\emptyset	\emptyset	\emptyset

Conversão ε -NFA \rightarrow DFA

- Resposta

	a	b	c
$\rightarrow^* \{p,q,r\}$	$\{p,q,r\}$	$\{q,r\}$	$\{p,q,r\}$
$^* \{q,r\}$	$\{p,q,r\}$	$\{r\}$	$\{p,q,r\}$
$^* \{r\}$	$\{\}$	$\{\}$	$\{\}$
$\{\}$	$\{\}$	$\{\}$	$\{\}$

Conversão ϵ -NFA \rightarrow DFA

- Dado o seguinte ϵ -NFA
 - Converta para um DFA que aceita a mesma linguagem

	ϵ	a	b
$\rightarrow * 1$	$\{3\}$	\emptyset	$\{2\}$
2	\emptyset	$\{2,3\}$	$\{3\}$
3	\emptyset	$\{1\}$	\emptyset

Conversão ϵ -NFA \rightarrow DFA

- Resposta

	a	b
$\rightarrow * \{1,3\}$	$\{1,3\}$	$\{2\}$
$\{2\}$	$\{2,3\}$	$\{3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\{3\}$	$\{1,3\}$	$\{\}$
$* \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{\}$	$\{\}$	$\{\}$

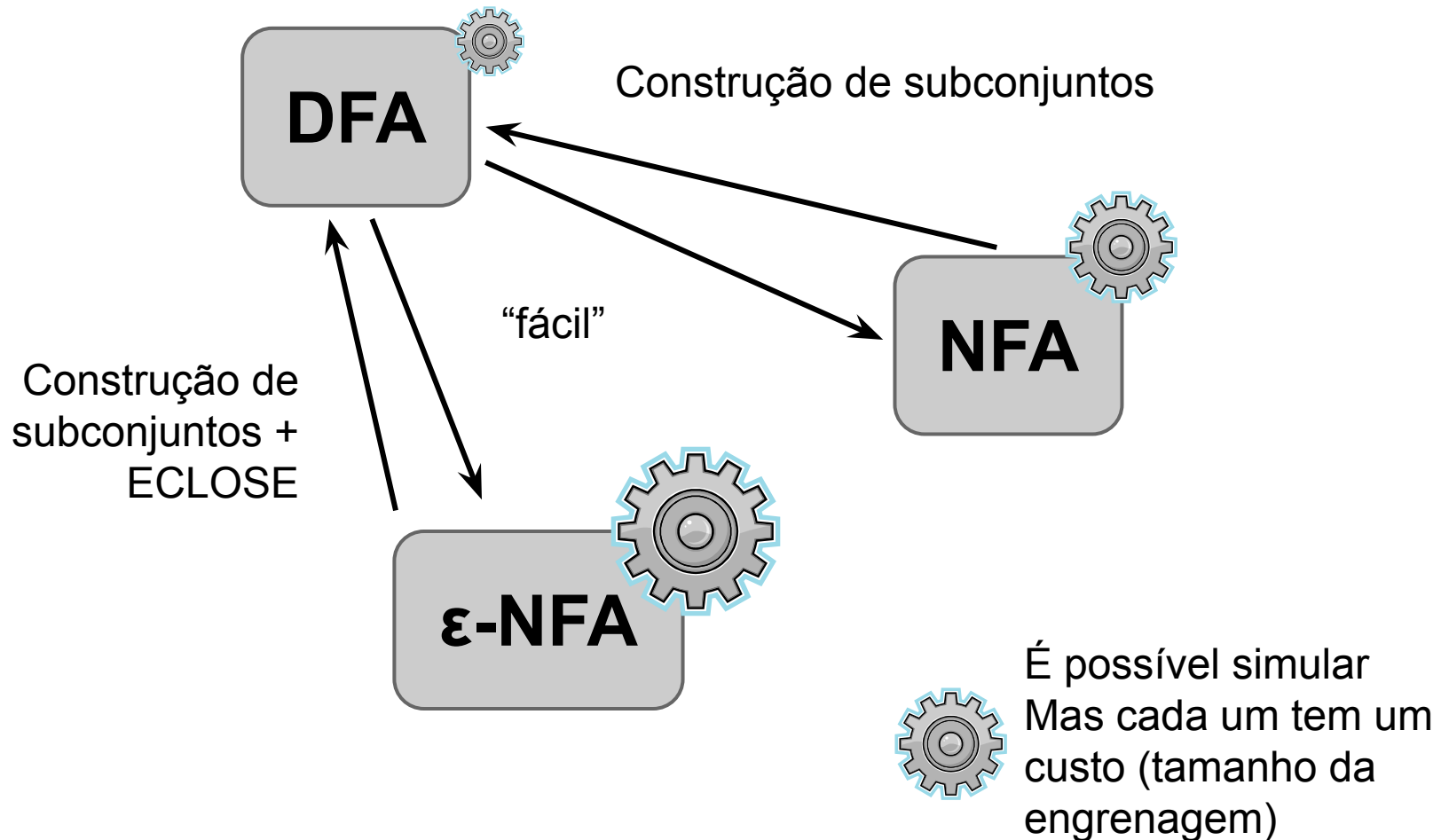
Conversão DFA \rightarrow ϵ -NFA

- “Resto” da prova
- Parte fácil
 - Mesmo caso da conversão de DFA para NFA
 - Mas fazendo com que $\delta(q, \epsilon) = \emptyset$ para todo estado q do DFA
 - Ou seja, não existem transições espontâneas (mas poderia ter)

Resumo

- Definições X Linguagens X Problemas
- Autômatos Finitos
 - DFA
 - NFA
 - ϵ -NFA
- Aceitam as mesmas linguagens (regulares)

Resumo



Resumo

- Simular direto vs converter para DFA
 - Depende da aplicação
 - Padrões de busca voláteis (ex: grep)
 - Melhor simular NFA
 - Padrões fixos (ex: análise léxica)
 - Melhor converter para DFA

Fim

Aula 06 - NFA com transições vazias