Fundamentos Teóricos da Computação

– Autômatos Finitos (Parte 01) –

Zenilton Kleber Gonçalves do Patrocínio Jr.

Ciência da Computação – PUC Minas

Belo Horizonte, Brasil

2025

PUC Minas

IM SCIENCE

Sumário

Introdução a Autômatos Finitos

Arquitetura de AF

Principais componentes da arquitetura de um AF:

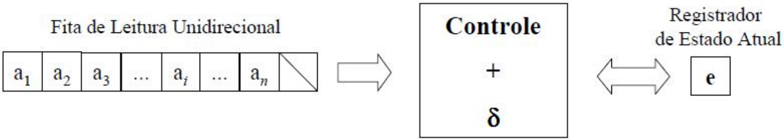
- Fita de Leitura Unidirecional
- Controle + Função de Transição
- Registrador de Estado Atual

Introdução a Autômatos Finitos

Arquitetura de AF

Principais componentes da arquitetura de um AF:

- Fita de Leitura Unidirecional
- Controle + Função de Transição
- Registrador de Estado Atual



Introdução a Autômatos Finitos

Exemplo N.01

Circuito de uma lâmpada

- Dois estados: acesso e apagado
- Duas ações (transições): acender e apagar

Exemplo N.02

Máquina de vender jornal

- Custo do jornal: 0,30
- Moedas aceitas: 0,05 / 0,10 / 0,25
- Não há circuito subtrator, nem memória

Introdução a Autômatos Finitos

Exemplo N.01

Circuito de uma lâmpada

- Dois estados: acesso e apagado
- Duas ações (transições): acender e apagar

Exemplo N.02

Máquina de vender jornal

- Custo do jornal: 0,30
- Moedas aceitas: 0,05 / 0,10 / 0,25
- Não há circuito subtrator, nem memória

Introdução a Autômatos Finitos

Exemplo N.01

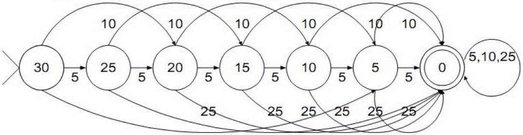
Circuito de uma lâmpada

- Dois estados: acesso e apagado
- Duas ações (transições): acender e apagar

Exemplo N.02

Máquina de vender jornal

- Custo do jornal: 0,30
- Moedas aceitas: 0,05 / 0,10 / 0,25
- Não há circuito subtrator, nem memória



Autômatos Finitos Determinísticos

Definição

Um **Autômato Finito Determinístico** (AFD) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto finito não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ função de transição (função total):

$$\delta : E \times \Sigma \mapsto E$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Determinísticos

Definição

Um **Autômato Finito Determinístico** (AFD) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto finito não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ função de transição (função total):

$$\delta : E \times \Sigma \mapsto E$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Determinísticos

Definição

Um **Autômato Finito Determinístico** (AFD) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto finito não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ função de transição (função total):

$$\delta : E \times \Sigma \mapsto E$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Determinísticos

Definição

Um **Autômato Finito Determinístico** (AFD) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto finito não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ função de transição (função total):

$$\delta : E \times \Sigma \mapsto E$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Determinísticos

Definição

Um **Autômato Finito Determinístico** (AFD) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto finito não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ função de transição (função total):

$$\delta : E \times \Sigma \mapsto E$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Determinísticos

Definição

Um **Autômato Finito Determinístico** (AFD) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

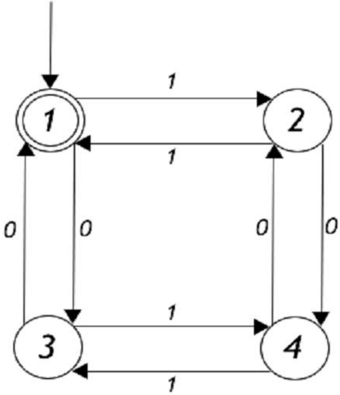
- $E \equiv$ conjunto finito não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ função de transição (função total):

$$\delta : E \times \Sigma \mapsto E$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Determinísticos

Representação Gráfica de um AFD



Estados : $E = \{1, 2, 3, 4\}$

Alfabeto : $\Sigma = \{0, 1\}$

Estado inicial : $i = 1$

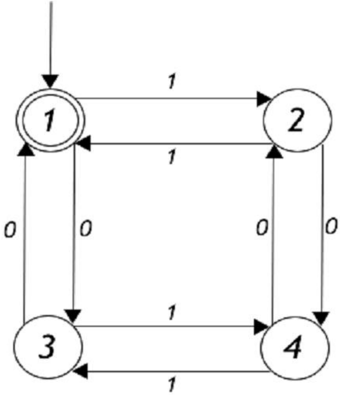
Estados finais : $F = \{1\}$

Função de transição : δ

Estado	Símbolo	
	0	1
1	3	2
2	4	1
3	1	4
4	2	3

Autômatos Finitos Determinísticos

Representação Gráfica de um AFD



Estados : $E = \{1, 2, 3, 4\}$

Alfabeto : $\Sigma = \{0, 1\}$

Estado inicial : $i = 1$

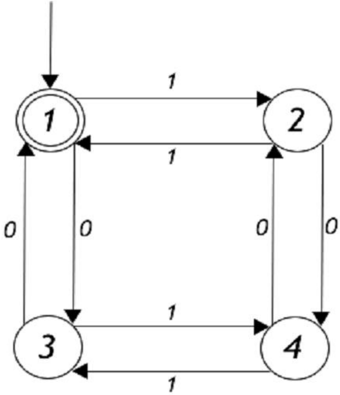
Estados finais : $F = \{1\}$

Função de transição : δ

Estado	Símbolo	
	0	1
1	3	2
2	4	1
3	1	4
4	2	3

Autômatos Finitos Determinísticos

Representação Gráfica de um AFD



Estados : $E = \{1, 2, 3, 4\}$

Alfabeto : $\Sigma = \{0, 1\}$

Estado inicial : $i = 1$

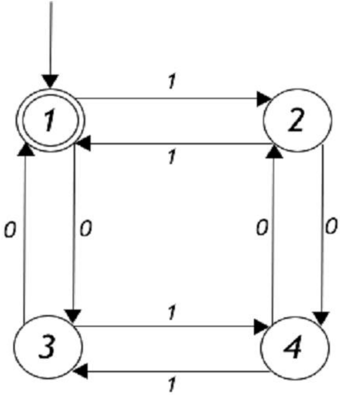
Estados finais : $F = \{1\}$

Função de transição : δ

Estado	Símbolo	
	0	1
1	3	2
2	4	1
3	1	4
4	2	3

Autômatos Finitos Determinísticos

Representação Gráfica de um AFD



Estados : $E = \{1, 2, 3, 4\}$

Alfabeto : $\Sigma = \{0, 1\}$

Estado inicial : $i = 1$

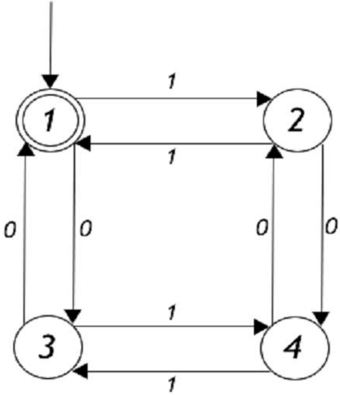
Estados finais : $F = \{1\}$

Função de transição : δ

Estado	Símbolo 0	Símbolo 1
1	3	2
2	4	1
3	1	4
4	2	3

Autômatos Finitos Determinísticos

Representação Gráfica de um AFD



Estados : $E = \{1, 2, 3, 4\}$

Alfabeto : $\Sigma = \{0, 1\}$

Estado inicial : $i = 1$

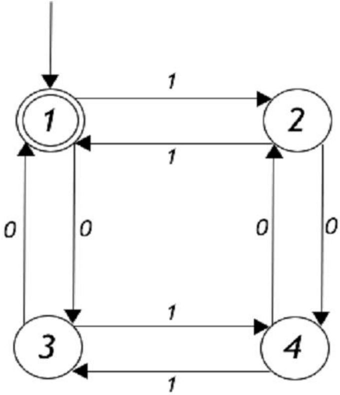
Estados finais : $F = \{1\}$

Função de transição : δ

Estado	Símbolo	
	0	1
1	3	2
2	4	1
3	1	4
4	2	3

Autômatos Finitos Determinísticos

Representação Gráfica de um AFD



Estados : $E = \{1, 2, 3, 4\}$

Alfabeto : $\Sigma = \{0, 1\}$

Estado inicial : $i = 1$

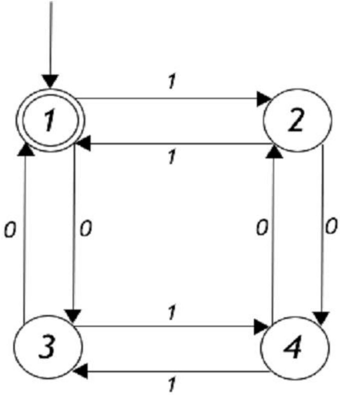
Estados finais : $F = \{1\}$

Função de transição : δ

Estado	Símbolo	
	0	1
1	3	2
2	4	1
3	1	4
4	2	3

Autômatos Finitos Determinísticos

Representação Gráfica de um AFD



Estados : $E = \{1, 2, 3, 4\}$

Alfabeto : $\Sigma = \{0, 1\}$

Estado inicial : $i = 1$

Estados finais : $F = \{1\}$

Função de transição : δ

Estado	Símbolo 0	Símbolo 1
1	3	2
2	4	1
3	1	4
4	2	3

Autômatos Finitos Determinísticos

Propriedades de AFDs

- Determinismo: cada par (estado, símbolo) leva a um único estado, daí a partir do estado inicial é atingido um único estado, para uma dada palavra de entrada
- Função de transição total: para toda palavra de entrada, só é possível se atingir um único estado consumindo-se toda a palavra
- Um único estado inicial: com vários o poder computacional não é maior
- Vários estados finais: com um só, o poder computacional é menor
- Conjunto finito de estados: com conjunto infinito, o poder computacional é maior

Autômatos Finitos Determinísticos

Propriedades de AFDs

- Determinismo: cada par (estado, símbolo) leva a um único estado, daí a partir do estado inicial é atingido um único estado, para uma dada palavra de entrada
- Função de transição total: para toda palavra de entrada, só é possível se atingir um único estado consumindo-se toda a palavra
- Um único estado inicial: com vários o poder computacional não é maior
- Vários estados finais: com um só, o poder computacional é menor
- Conjunto finito de estados: com conjunto infinito, o poder computacional é maior

Autômatos Finitos Determinísticos

Propriedades de AFDs

- Determinismo: cada par (estado, símbolo) leva a um único estado, daí a partir do estado inicial é atingido um único estado, para uma dada palavra de entrada
- Função de transição total: para toda palavra de entrada, só é possível se atingir um único estado consumindo-se toda a palavra
- Um único estado inicial: com vários o poder computacional não é maior
- Vários estados finais: com um só, o poder computacional é menor
- Conjunto finito de estados: com conjunto infinito, o poder computacional é maior

Autômatos Finitos Determinísticos

Propriedades de AFDs

- Determinismo: cada par (estado, símbolo) leva a um único estado, daí a partir do estado inicial é atingido um único estado, para uma dada palavra de entrada
- Função de transição total: para toda palavra de entrada, só é possível se atingir um único estado consumindo-se toda a palavra
- Um único estado inicial: com vários o poder computacional não é maior
- Vários estados finais: com um só, o poder computacional é menor
- Conjunto finito de estados: com conjunto infinito, o poder computacional é maior

Autômatos Finitos Determinísticos

Propriedades de AFDs

- Determinismo: cada par (estado, símbolo) leva a um único estado, daí a partir do estado inicial é atingido um único estado, para uma dada palavra de entrada
- Função de transição total: para toda palavra de entrada, só é possível se atingir um único estado consumindo-se toda a palavra
- Um único estado inicial: com vários o poder computacional não é maior
- Vários estados finais: com um só, o poder computacional é menor
- Conjunto finito de estados: com conjunto infinito, o poder computacional é maior

Autômatos Finitos Determinísticos

Comportamento de um AFD

- Inicialmente o estado atual da máquina é o estado inicial;
- A cada transição do autômato, um símbolo da entrada é lido e o estado atual é atualizado utilizando a função de transição;
- O autômato reconhece o string de entrada se a última transição resultar em um estado final.

Exemplo do comportamento de um AFD

Qual a linguagem aceita pelo seguinte AFD? $(a \cup ba^*b)^*$

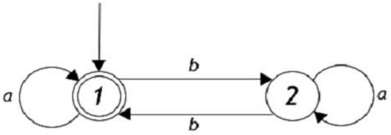
Autômatos Finitos Determinísticos

Comportamento de um AFD

- Inicialmente o estado atual da máquina é o estado inicial;
- A cada transição do autômato, um símbolo da entrada é lido e o estado atual é atualizado utilizando a função de transição;
- O autômato reconhece o string de entrada se a última transição resultar em um estado final.

Exemplo do comportamento de um AFD

Qual a linguagem aceita pelo seguinte AFD? $(a \cup ba^* b)^*$



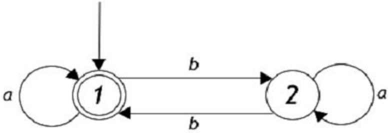
Autômatos Finitos Determinísticos

Comportamento de um AFD

- Inicialmente o estado atual da máquina é o estado inicial;
- A cada transição do autômato, um símbolo da entrada é lido e o estado atual é atualizado utilizando a função de transição;
- O autômato reconhece o string de entrada se a última transição resultar em um estado final.

Exemplo do comportamento de um AFD

Qual a linguagem aceita pelo seguinte AFD? $(a \cup ba^* b)^*$



Autômatos Finitos Determinísticos

Configuração Instantânea de AFD

A configuração instantânea de um AFD é dada pelo par $[e, w]$ em que:

- $e \equiv$ estado atual;
- $w \equiv$ o sufixo ainda não processado da palavra de entrada.

Seja um AFD $M = (E, \Sigma, \delta, i, F)$. A relação $\vdash \subseteq (E \times \Sigma^*)^2$, para M, é tal que todo $e, e' \in E$, $a \in \Sigma$:

$$[e, ay] \vdash [e', y], \forall y \in \Sigma^* \iff \delta(e, a) = e'.$$

OBS.: Usa-se $\vdash^*$ para representar zero ou mais aplicações da relação $\vdash$

Autômatos Finitos Determinísticos

Configuração Instantânea de AFD

A configuração instantânea de um AFD é dada pelo par $[e, w]$ em que:

- $e \equiv$ estado atual;
- $w \equiv$ o sufixo ainda não processado da palavra de entrada.

Seja um AFD $M = (E, \Sigma, \delta, i, F)$. A relação $\vdash \subseteq (E \times \Sigma^*)^2$, para M, é tal que todo $e, e' \in E$, $a \in \Sigma$:

$$[e, ay] \vdash [e', y], \forall y \in \Sigma^* \iff \delta(e, a) = e'.$$

OBS.: Usa-se $\vdash^*$ para representar zero ou mais aplicações da relação $\vdash$

Autômatos Finitos Determinísticos

Configuração Instantânea de AFD

A configuração instantânea de um AFD é dada pelo par $[e, w]$ em que:

- $e \equiv$ estado atual;
- $w \equiv$ o sufixo ainda não processado da palavra de entrada.

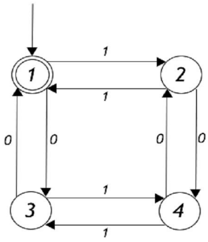
Seja um AFD $M = (E, \Sigma, \delta, i, F)$. A relação $\vdash \subseteq (E \times \Sigma^*)^2$, para M, é tal que todo $e, e' \in E$, $a \in \Sigma$:

$$[e, ay] \vdash [e', y], \forall y \in \Sigma^* \iff \delta(e, a) = e'.$$

OBS.: Usa-se $\vdash^*$ para representar zero ou mais aplicações da relação $\vdash$

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda] \quad \textbf{(Aceita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

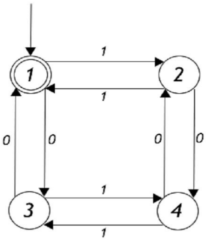
Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda] \quad \textbf{(Rejeita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda] \quad \textbf{(Aceita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

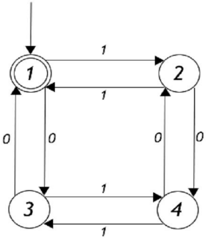
Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda] \quad \textbf{(Rejeita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda] \quad \textbf{(Aceita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

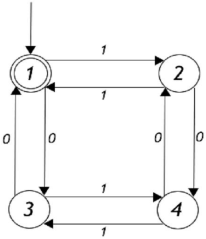
Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda] \quad \textbf{(Rejeita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda] \quad \textbf{(Aceita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

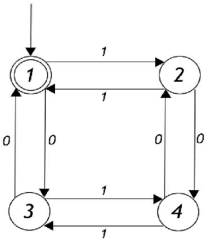
Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda] \quad \textbf{(Rejeita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda] \quad \textbf{(Aceita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

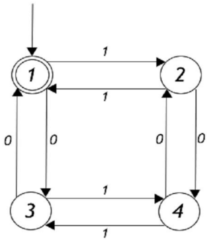
Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda] \quad \textbf{(Rejeita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda] \quad \textbf{(Aceita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

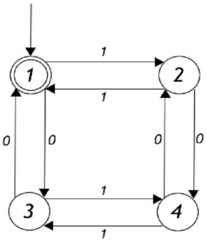
Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda] \quad \textbf{(Rejeita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda] \quad \textbf{(Aceita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

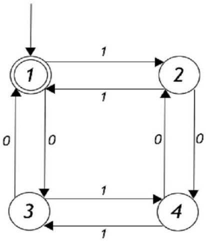
Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda] \quad \textbf{(Rejeita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda] \quad \textbf{(Aceita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

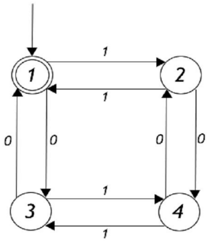
Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda] \quad \textbf{(Rejeita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda] \quad \textbf{(Aceita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

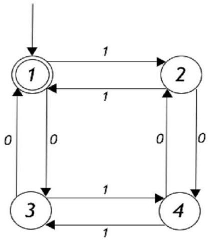
Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda] \quad \textbf{(Rejeita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda] \quad \textbf{(Aceita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

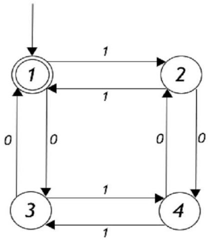
Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda] \quad \textbf{(Rejeita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda]$$ **(Aceita)**

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

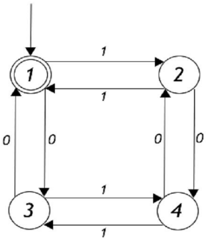
Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda]$$ (Rejeita)

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Reconhecimento de uma sentença por AFD



Pode-se representar o processo de reconhecimento da sentença $w_1 = 0101$ da seguinte forma:

$$[1, 0101] \vdash [3, 101] \vdash [4, 01] \vdash [2, 1] \vdash [1, \lambda] \quad \textbf{(Aceita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado de aceitação com a sentença completamente lida (consumida) !

Por outro lado, o não reconhecimento da sentença $w_2 = 0010$ pode ser visto a seguir:

$$[1, 0010] \vdash [3, 010] \vdash [1, 10] \vdash [2, 0] \vdash [4, \lambda] \quad \textbf{(Rejeita)}$$

OBS: Iniciou-se com a sentença completa no estado inicial e terminou-se em um estado que não é de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Determinísticos

Linguagem de um AFD

Seja um AFD $M = (E, \Sigma, \delta, i, F)$. A linguagem reconhecida por M é

$$L(M) = \{w \in \Sigma^* \mid [i, w] \overset{*}{\vdash} [f, \lambda] \text{ para algum } f \in F\}.$$

Uma palavra w tal que $[i, w] \vdash^* [f, \lambda]$, em que $f \in F$, é dita ser reconhecida (ou aceita) por M.

Equivalência de AFDs

Dois AFDs são equivalentes se eles reconhecem a mesma linguagem.

Autômatos Finitos Determinísticos

Linguagem de um AFD

Seja um AFD $M = (E, \Sigma, \delta, i, F)$. A linguagem reconhecida por M é

$$L(M) = \{w \in \Sigma^* \mid [i, w] \overset{*}{\vdash} [f, \lambda] \text{ para algum } f \in F\}.$$

Uma palavra w tal que $[i, w] \vdash^* [f, \lambda]$, em que $f \in F$, é dita ser reconhecida (ou aceita) por M.

Equivalência de AFDs

Dois AFDs são equivalentes se eles reconhecem a mesma linguagem.

Autômatos Finitos Determinísticos

Determinismo Incompleto

Ocorre quando nem todas as transições são especificadas, isto é, a função de transição δ é uma função parcial.
Quando uma palavra é analisada e não há transição para um próximo estado o autômato pára (**HALT**), rejeitando a mesma.

Exemplo de determinismo incompleto

Qual a linguagem aceita pelo seguinte AFD? $(ab)^*c$

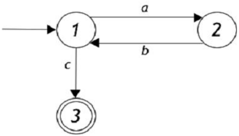
Autômatos Finitos Determinísticos

Determinismo Incompleto

Ocorre quando nem todas as transições são especificadas, isto é, a função de transição δ é uma função parcial.
Quando uma palavra é analisada e não há transição para um próximo estado o autômato pára (**HALT**), rejeitando a mesma.

Exemplo de determinismo incompleto

Qual a linguagem aceita pelo seguinte AFD? $(ab)^*c$



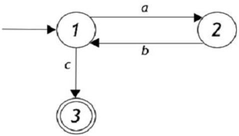
Autômatos Finitos Determinísticos

Determinismo Incompleto

Ocorre quando nem todas as transições são especificadas, isto é, a função de transição δ é uma função parcial.
Quando uma palavra é analisada e não há transição para um próximo estado o autômato pára (**HALT**), rejeitando a mesma.

Exemplo de determinismo incompleto

Qual a linguagem aceita pelo seguinte AFD? $(ab)^*c$



Autômatos Finitos Determinísticos

Simulação de AFD

```
algoritmo
   Entrada:   AFD M = (E, Σ, δ, i, F) e palavra w dada por prox() que
              termina com EOS(="End of Sequence")
   Estado ← i;
   Simbolo ← prox();
   enquanto Simbolo ≠ EOS faça
      se δ(Estado, Simbolo) for indefinido então
         retorne "Entrada não foi aceita"; (HALT)
      fim se;
      Estado ← δ(Estado, Simbolo);  Simbolo ← prox();
   fim enquanto;
   se Estado ∈ F então
      retorne "Entrada foi aceita";
   senão
      retorne "Entrada não foi aceita";
   fim se;
fim algoritmo
```

Autômatos Finitos Determinísticos

Simulação de AFD

```
algoritmo
   Entrada:  AFD M = (E, Σ, δ, i, F) e palavra w dada por prox() que
             termina com EOS(="End of Sequence")
   Estado ← i;
   Simbolo ← prox();
   enquanto Simbolo ≠ EOS faça
      se δ(Estado, Simbolo) for indefinido então
         retorne "Entrada não foi aceita"; (HALT)
      fim se;
      Estado ← δ(Estado, Simbolo); Simbolo ← prox();
   fim enquanto;
   se Estado ∈ F então
      retorne "Entrada foi aceita";
   senão
      retorne "Entrada não foi aceita";
   fim se;
fim algoritmo
```

Autômatos Finitos Determinísticos

Simulação de AFD

```
algoritmo
   Entrada:  AFD M = (E, Σ, δ, i, F) e palavra w dada por prox() que
             termina com EOS(="End of Sequence")
   Estado ← i;
   Simbolo ← prox();
   enquanto Simbolo ≠ EOS faça
      se δ(Estado, Simbolo) for indefinido então
         retorne "Entrada não foi aceita"; (HALT)
      fim se;
      Estado ← δ(Estado, Simbolo); Simbolo ← prox();
   fim enquanto;
   se Estado ∈ F então
      retorne "Entrada foi aceita";
   senão
      retorne "Entrada não foi aceita";
   fim se;
fim algoritmo
```

Autômatos Finitos Determinísticos

Simulação de AFD

```
algoritmo
   Entrada:  AFD M = (E, Σ, δ, i, F) e palavra w dada por prox() que
             termina com EOS(="End of Sequence")
   Estado ← i;
   Simbolo ← prox();
   enquanto Simbolo ≠ EOS faça
      se δ(Estado, Simbolo) for indefinido então
         retorne "Entrada não foi aceita"; (HALT)
      fim se;
      Estado ← δ(Estado, Simbolo); Simbolo ← prox();
   fim enquanto;
   se Estado ∈ F então
      retorne "Entrada foi aceita";
   senão
      retorne "Entrada não foi aceita";
   fim se;
fim algoritmo
```

Autômatos Finitos Determinísticos

Simulação de AFD

```
algoritmo
   Entrada:  AFD M = (E, Σ, δ, i, F) e palavra w dada por prox() que
             termina com EOS(="End of Sequence")
   Estado ← i;
   Simbolo ← prox();
   enquanto Simbolo ≠ EOS faça
      se δ(Estado, Simbolo) for indefinido então
         retorne "Entrada não foi aceita"; (HALT)
      fim se;
      Estado ← δ(Estado, Simbolo); Simbolo ← prox();
   fim enquanto;
   se Estado ∈ F então
      retorne "Entrada foi aceita";
   senão
      retorne "Entrada não foi aceita";
   fim se;
fim algoritmo
```

Autômatos Finitos Determinísticos

Simulação de AFD

algoritmo
  Entrada: AFD $M = (E, \Sigma, \delta, i, F)$ e palavra w dada por $prox()$ que termina com EOS (="End of Sequence")
  $Estado \leftarrow i$;
  $Simbolo \leftarrow prox()$;
  enquanto $Simbolo \neq EOS$ **faça**
    se $\delta(Estado, Simbolo)$ for indefinido **então**
      retorne "Entrada não foi aceita"; (HALT)
    fim se;
    $Estado \leftarrow \delta(Estado, Simbolo)$; $Simbolo \leftarrow prox()$;
  fim enquanto;
  se $Estado \in F$ **então**
    retorne "Entrada foi aceita";
  senão
    retorne "Entrada não foi aceita";
  fim se;
fim algoritmo

Autômatos Finitos Determinísticos

Simulação de AFD

algoritmo
  Entrada: AFD $M = (E, \Sigma, \delta, i, F)$ e palavra w dada por $prox()$ que
  termina com EOS(="End of Sequence")
  $Estado \leftarrow i$;
  $Simbolo \leftarrow prox()$;
  enquanto $Simbolo \neq EOS$ faça
    se $\delta(Estado, Simbolo)$ for indefinido então
      retorne "Entrada não foi aceita"; (HALT)
    fim se;
    $Estado \leftarrow \delta(Estado, Simbolo)$; $Simbolo \leftarrow prox()$;
  fim enquanto;
  se $Estado \in F$ então
    retorne "Entrada foi aceita";
  senão
    retorne "Entrada não foi aceita";
  fim se;
fim algoritmo

Autômatos Finitos Determinísticos

Simulação de AFD

algoritmo
  Entrada: AFD $M = (E, \Sigma, \delta, i, F)$ e palavra w dada por $prox()$ que
     termina com EOS (="End of Sequence")
  $Estado \leftarrow i$;
  $Simbolo \leftarrow prox()$;
  enquanto $Simbolo \neq EOS$ faça
    se $\delta(Estado, Simbolo)$ for indefinido então
      retorne "Entrada não foi aceita"; (HALT)
    fim se;
    $Estado \leftarrow \delta(Estado, Simbolo)$; $Simbolo \leftarrow prox()$;
  fim enquanto;
  se $Estado \in F$ então
    retorne "Entrada foi aceita";
  senão
    retorne "Entrada não foi aceita";
  fim se;
fim algoritmo

Autômatos Finitos Determinísticos

Simulação de AFD

```
algoritmo
   Entrada:  AFD M = (E, Σ, δ, i, F) e palavra w dada por prox() que
             termina com EOS(="End of Sequence")
   Estado ← i;
   Simbolo ← prox();
   enquanto Simbolo ≠ EOS faça
      se δ(Estado, Simbolo) for indefinido então
         retorne "Entrada não foi aceita"; (HALT)
      fim se;
      Estado ← δ(Estado, Simbolo); Simbolo ← prox();
   fim enquanto;
   se Estado ∈ F então
      retorne "Entrada foi aceita";
   senão
      retorne "Entrada não foi aceita";
   fim se;
fim algoritmo
```

Produto de AFDs

Simulação do funcionamento **em paralelo** de dois AFDs

Sejam dois AFDs $M_1 = (E_1, \Sigma, \delta_1, i_1, F_1)$ e $M_2 = (E_2, \Sigma, \delta_2, i_2, F_2)$ pode-se construir $M_3 = (E_3, \Sigma, \delta_3, i_3, F_3)$ em que:

- $E_3 = E_1 \times E_2$
- $i_3 = [i_1, i_2]$
- $\delta_3([e_1, e_2], a) = [\delta_1(e_1, a), \delta_2(e_2, a)], \forall e_1 \in E_1, e_2 \in E_2, a \in \Sigma$
- $F_3 = \begin{cases} F_1 \times F_2 & \text{para } L(M_3) = L(M_1) \cap L(M_2) \\ (F_1 \times E_2) \cup (E_1 \times F_2) & \text{para } L(M_3) = L(M_1) \cup L(M_2) \end{cases}$

Produto de AFDs

Simulação do funcionamento **em paralelo** de dois AFDs

Sejam dois AFDs $M_1 = (E_1, \Sigma, \delta_1, i_1, F_1)$ e $M_2 = (E_2, \Sigma, \delta_2, i_2, F_2)$ pode-se construir $M_3 = (E_3, \Sigma, \delta_3, i_3, F_3)$ em que:

- $E_3 = E_1 \times E_2$
- $i_3 = [i_1, i_2]$
- $\delta_3([e_1, e_2], a) = [\delta_1(e_1, a), \delta_2(e_2, a)], \forall e_1 \in E_1, e_2 \in E_2, a \in \Sigma$
- $F_3 = \begin{cases} F_1 \times F_2 & \text{para } L(M_3) = L(M_1) \cap L(M_2) \\ (F_1 \times E_2) \cup (E_1 \times F_2) & \text{para } L(M_3) = L(M_1) \cup L(M_2) \end{cases}$

Produto de AFDs

Simulação do funcionamento **em paralelo** de dois AFDs

Sejam dois AFDs $M_1 = (E_1, \Sigma, \delta_1, i_1, F_1)$ e $M_2 = (E_2, \Sigma, \delta_2, i_2, F_2)$ pode-se construir $M_3 = (E_3, \Sigma, \delta_3, i_3, F_3)$ em que:

- $E_3 = E_1 \times E_2$
- $i_3 = [i_1, i_2]$
- $\delta_3([e_1, e_2], a) = [\delta_1(e_1, a), \delta_2(e_2, a)], \forall e_1 \in E_1, e_2 \in E_2, a \in \Sigma$
- $F_3 = \begin{cases} F_1 \times F_2 & \text{para } L(M_3) = L(M_1) \cap L(M_2) \\ (F_1 \times E_2) \cup (E_1 \times F_2) & \text{para } L(M_3) = L(M_1) \cup L(M_2) \end{cases}$

Produto de AFDs

Simulação do funcionamento **em paralelo** de dois AFDs

Sejam dois AFDs $M_1 = (E_1, \Sigma, \delta_1, i_1, F_1)$ e $M_2 = (E_2, \Sigma, \delta_2, i_2, F_2)$ pode-se construir $M_3 = (E_3, \Sigma, \delta_3, i_3, F_3)$ em que:

- $E_3 = E_1 \times E_2$
- $i_3 = [i_1, i_2]$
- $\delta_3([e_1, e_2], a) = [\delta_1(e_1, a), \delta_2(e_2, a)], \forall e_1 \in E_1, e_2 \in E_2, a \in \Sigma$
- $F_3 = \begin{cases} F_1 \times F_2 & \text{para } L(M_3) = L(M_1) \cap L(M_2) \\ (F_1 \times E_2) \cup (E_1 \times F_2) & \text{para } L(M_3) = L(M_1) \cup L(M_2) \end{cases}$

Produto de AFDs

Simulação do funcionamento **em paralelo** de dois AFDs

Sejam dois AFDs $M_1 = (E_1, \Sigma, \delta_1, i_1, F_1)$ e $M_2 = (E_2, \Sigma, \delta_2, i_2, F_2)$ pode-se construir $M_3 = (E_3, \Sigma, \delta_3, i_3, F_3)$ em que:

- $E_3 = E_1 \times E_2$
- $i_3 = [i_1, i_2]$
- $\delta_3([e_1, e_2], a) = [\delta_1(e_1, a), \delta_2(e_2, a)], \forall e_1 \in E_1, e_2 \in E_2, a \in \Sigma$
- $F_3 = \begin{cases} F_1 \times F_2 & \text{para } L(M_3) = L(M_1) \cap L(M_2) \\ (F_1 \times E_2) \cup (E_1 \times F_2) & \text{para } L(M_3) = L(M_1) \cup L(M_2) \end{cases}$

Linguagens Finitas

Para toda linguagem finita existe um AFD !

AFD com diagrama de estados simplificado **sem ciclos** corresponde a uma árvore em que o estado inicial é a raiz e cada palavra é corresponde a um caminho da raiz até um de seus descendentes.

Palavras que possuam um prefixo em comum compartilham um caminho correspondente ao prefixo.

$\Longrightarrow$ Semelhante a uma TRIE !!!

OBS: Esse AFD pode ser não mínimo !

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.

Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
    a. $(0 \cup 1)^*$
    b. $0(0 \cup 1)^*1$
    c. $(0 \cup 1)^*0(0 \cup 1)^*$
    d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w$ termina com 00 $\}$
3. $L = \{w \in \Sigma^* \mid w$ possui 000 como subsentença $\}$
4. $L = \{w \in \Sigma^* \mid$ o 2^o símbolo de w, da esquerda para direita, é 1 $\}$
5. $L = \{w \in \Sigma^* \mid w$ não contém 101 como subsentença $\}$
6. $L = \{w \in \Sigma^* \mid w$ representa um número binário múltiplo de 3 $\}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0\}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N}\}$

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.

Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
   a. $(0 \cup 1)^*$
   b. $0(0 \cup 1)^*1$
   c. $(0 \cup 1)^*0(0 \cup 1)^*$
   d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w \text{ termina com } 00\}$
3. $L = \{w \in \Sigma^* \mid w \text{ possui } 000 \text{ como subsentença}\}$
4. $L = \{w \in \Sigma^* \mid \text{ o } 2^o \text{ símbolo de } w\text{, da esquerda para direita, é } 1\}$
5. $L = \{w \in \Sigma^* \mid w \text{ não contém } 101 \text{ como subsentença}\}$
6. $L = \{w \in \Sigma^* \mid w \text{ representa um número binário múltiplo de } 3\}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0\}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N}\}$

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.

Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
   a. $(0 \cup 1)^*$
   b. $0(0 \cup 1)^*1$
   c. $(0 \cup 1)^*0(0 \cup 1)^*$
   d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w \text{ termina com } 00\}$
3. $L = \{w \in \Sigma^* \mid w \text{ possui } 000 \text{ como subsentença}\}$
4. $L = \{w \in \Sigma^* \mid \text{o } 2^o \text{ símbolo de } w\text{, da esquerda para direita, é } 1\}$
5. $L = \{w \in \Sigma^* \mid w \text{ não contém } 101 \text{ como subsentença}\}$
6. $L = \{w \in \Sigma^* \mid w \text{ representa um número binário múltiplo de } 3\}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0\}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N}\}$

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.

Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
   a. $(0 \cup 1)^*$
   b. $0(0 \cup 1)^*1$
   c. $(0 \cup 1)^*0(0 \cup 1)^*$
   d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w \text{ termina com } 00\}$
3. $L = \{w \in \Sigma^* \mid w \text{ possui } 000 \text{ como subsentença}\}$
4. $L = \{w \in \Sigma^* \mid \text{ o } 2^o \text{ símbolo de } w\text{, da esquerda para direita, é } 1\}$
5. $L = \{w \in \Sigma^* \mid w \text{ não contém } 101 \text{ como subsentença}\}$
6. $L = \{w \in \Sigma^* \mid w \text{ representa um número binário múltiplo de } 3\}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0\}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N}\}$

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.
Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
    a. $(0 \cup 1)^*$
    b. $0(0 \cup 1)^*1$
    c. $(0 \cup 1)^*0(0 \cup 1)^*$
    d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w \text{ termina com } 00 \}$
3. $L = \{w \in \Sigma^* \mid w \text{ possui } 000 \text{ como subsentença } \}$
4. $L = \{w \in \Sigma^* \mid \text{ o } 2^o \text{ símbolo de } w\text{, da esquerda para direita, é } 1 \}$
5. $L = \{w \in \Sigma^* \mid w \text{ não contém } 101 \text{ como subsentença } \}$
6. $L = \{w \in \Sigma^* \mid w \text{ representa um número binário múltiplo de } 3 \}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0 \}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N} \}$

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.

Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
   a. $(0 \cup 1)^*$
   b. $0(0 \cup 1)^*1$
   c. $(0 \cup 1)^*0(0 \cup 1)^*$
   d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w \text{ termina com } 00\ \}$
3. $L = \{w \in \Sigma^* \mid w \text{ possui } 000 \text{ como subsentença}\ \}$
4. $L = \{w \in \Sigma^* \mid \text{ o } 2^o \text{ símbolo de } w\text{, da esquerda para direita, é } 1\ \}$
5. $L = \{w \in \Sigma^* \mid w \text{ não contém } 101 \text{ como subsentença}\ \}$
6. $L = \{w \in \Sigma^* \mid w \text{ representa um número binário múltiplo de } 3\ \}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0\ \}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N}\ \}$

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.

Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
   a. $(0 \cup 1)^*$
   b. $0(0 \cup 1)^*1$
   c. $(0 \cup 1)^*0(0 \cup 1)^*$
   d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w \text{ termina com } 00 \}$
3. $L = \{w \in \Sigma^* \mid w \text{ possui } 000 \text{ como subsentença } \}$
4. $L = \{w \in \Sigma^* \mid \text{ o } 2^o \text{ símbolo de } w\text{, da esquerda para direita, é } 1 \}$
5. $L = \{w \in \Sigma^* \mid w \text{ não contém } 101 \text{ como subsentença } \}$
6. $L = \{w \in \Sigma^* \mid w \text{ representa um número binário múltiplo de } 3 \}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0 \}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N} \}$

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.

Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
   a. $(0 \cup 1)^*$
   b. $0(0 \cup 1)^*1$
   c. $(0 \cup 1)^*0(0 \cup 1)^*$
   d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w$ termina com 00 $\}$
3. $L = \{w \in \Sigma^* \mid w$ possui 000 como subsentença $\}$
4. $L = \{w \in \Sigma^* \mid$ o 2^o símbolo de w, da esquerda para direita, é 1 $\}$
5. $L = \{w \in \Sigma^* \mid w$ não contém 101 como subsentença $\}$
6. $L = \{w \in \Sigma^* \mid w$ representa um número binário múltiplo de 3 $\}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0 \}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N} \}$

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.

Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
   a. $(0 \cup 1)^*$
   b. $0(0 \cup 1)^*1$
   c. $(0 \cup 1)^*0(0 \cup 1)^*$
   d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w \text{ termina com } 00\}$
3. $L = \{w \in \Sigma^* \mid w \text{ possui } 000 \text{ como subsentença}\}$
4. $L = \{w \in \Sigma^* \mid \text{ o } 2^o \text{ símbolo de } w\text{, da esquerda para direita, é } 1\}$
5. $L = \{w \in \Sigma^* \mid w \text{ não contém } 101 \text{ como subsentença}\}$
6. $L = \{w \in \Sigma^* \mid w \text{ representa um número binário múltiplo de } 3\}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0\}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N}\}$

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.

Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
   a. $(0 \cup 1)^*$
   b. $0(0 \cup 1)^*1$
   c. $(0 \cup 1)^*0(0 \cup 1)^*$
   d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w$ termina com 00 $\}$
3. $L = \{w \in \Sigma^* \mid w$ possui 000 como subsentença $\}$
4. $L = \{w \in \Sigma^* \mid$ o 2^o símbolo de w, da esquerda para direita, é 1 $\}$
5. $L = \{w \in \Sigma^* \mid w$ não contém 101 como subsentença $\}$
6. $L = \{w \in \Sigma^* \mid w$ representa um número binário múltiplo de 3 $\}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0 \}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N} \}$

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.

Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
   a. $(0 \cup 1)^*$
   b. $0(0 \cup 1)^*1$
   c. $(0 \cup 1)^*0(0 \cup 1)^*$
   d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w$ termina com 00 $\}$
3. $L = \{w \in \Sigma^* \mid w$ possui 000 como subsentença $\}$
4. $L = \{w \in \Sigma^* \mid$ o 2^o símbolo de w, da esquerda para direita, é 1 $\}$
5. $L = \{w \in \Sigma^* \mid w$ não contém 101 como subsentença $\}$
6. $L = \{w \in \Sigma^* \mid w$ representa um número binário múltiplo de 3 $\}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0 \}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N} \}$

Autômatos Finitos Determinísticos

Autômatos Finitos Determinísticos

Exercício.

Construa AFDs para as seguintes linguagens sobre o alfabeto $\Sigma = \{0, 1\}$

1. Para a linguagem denotadas por:
   a. $(0 \cup 1)^*$
   b. $0(0 \cup 1)^*1$
   c. $(0 \cup 1)^*0(0 \cup 1)^*$
   d. $(1 \cup 01 \cup 001)^*(\lambda \cup 0 \cup 00)$
2. $L = \{w \in \Sigma^* \mid w \text{ termina com } 00\}$
3. $L = \{w \in \Sigma^* \mid w \text{ possui } 000 \text{ como subsentença}\}$
4. $L = \{w \in \Sigma^* \mid \text{ o } 2^o \text{ símbolo de } w\text{, da esquerda para direita, é } 1\}$
5. $L = \{w \in \Sigma^* \mid w \text{ não contém } 101 \text{ como subsentença}\}$
6. $L = \{w \in \Sigma^* \mid w \text{ representa um número binário múltiplo de } 3\}$
7. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \geq 0\}$
8. $L = \{w \in \Sigma^* \mid w = 0^n1^n, n \leq k, k \in \mathbb{N}\}$

Autômatos Finitos Não Determinísticos

Não Determinismo

Autômatos que permitem mais de uma transição partindo de um estado para um mesmo símbolo do alfabeto de entrada.

Ex.: $\delta(e_i, a) = \{e_j, e_k, \ldots, e_m\}$

Definição – AFN

Um **Autômato Finitos Não Determinístico** (AFN) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- E = conjunto não vazio de estados
- Σ = alfabeto de entrada
- δ = função de transição:
$$\delta : E \times \Sigma \mapsto \mathcal{P}(E)$$
- i = estado inicial ($i \in E$)
- F = conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Não Determinismo

Autômatos que permitem mais de uma transição partindo de um estado para um mesmo símbolo do alfabeto de entrada.

Ex.: $\delta(e_i, a) = \{e_j, e_k, \ldots, e_m\}$

Definição – AFN

Um **Autômato Finitos Não Determinístico** (AFN) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**

  $$\delta : E \times \Sigma \mapsto \mathcal{P}(E)$$

- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Não Determinismo

Autômatos que permitem mais de uma transição partindo de um estado para um mesmo símbolo do alfabeto de entrada.

Ex.: $\delta(e_i, a) = \{e_j, e_k, \ldots, e_m\}$

Definição – AFN

Um **Autômato Finitos Não Determinístico** (AFN) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $$\delta : E \times \Sigma \mapsto \mathcal{P}(E)$$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Não Determinismo

Autômatos que permitem mais de uma transição partindo de um estado para um mesmo símbolo do alfabeto de entrada.
Ex.: $\delta(e_i, a) = \{e_j, e_k, \ldots, e_m\}$

Definição – AFN

Um **Autômato Finitos Não Determinístico** (AFN) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $$\delta : E \times \Sigma \mapsto \mathcal{P}(E)$$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Não Determinismo

Autômatos que permitem mais de uma transição partindo de um estado para um mesmo símbolo do alfabeto de entrada.
Ex.: $\delta(e_i, a) = \{e_j, e_k, \ldots, e_m\}$

Definição – AFN

Um **Autômato Finitos Não Determinístico** (AFN) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $$\delta : E \times \Sigma \mapsto \mathcal{P}(E)$$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Não Determinismo

Autômatos que permitem mais de uma transição partindo de um estado para um mesmo símbolo do alfabeto de entrada.
Ex.: $\delta(e_i, a) = \{e_j, e_k, \ldots, e_m\}$

Definição – AFN

Um **Autômato Finitos Não Determinístico** (AFN) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $$\delta : E \times \Sigma \mapsto \mathcal{P}(E)$$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Não Determinismo

Autômatos que permitem mais de uma transição partindo de um estado para um mesmo símbolo do alfabeto de entrada.
Ex.: $\delta(e_i, a) = \{e_j, e_k, \ldots, e_m\}$

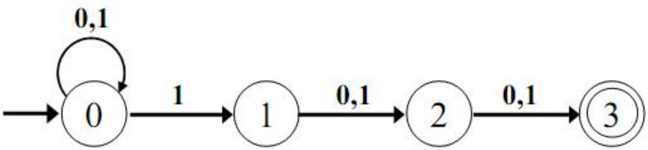
Definição – AFN

Um **Autômato Finitos Não Determinístico** (AFN) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $$\delta : E \times \Sigma \mapsto \mathcal{P}(E)$$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Exemplo de AFN



Uma palavra pode gerar diferentes computações !!!

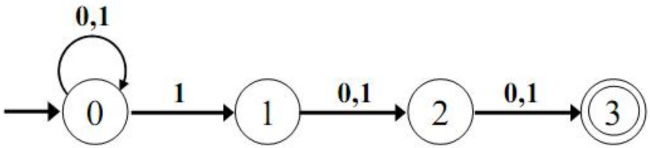
Linguagem de um AFN

Seja um AFN $M = (E, \Sigma, \delta, i, F)$. A linguagem reconhecida por M é

$$L(M) = \{w \in \Sigma^* \mid \exists \text{ computação } [i, w] \overset{*}{\vdash} [f, \lambda] \text{ para algum } f \in F\}.$$

Autômatos Finitos Não Determinísticos

Exemplo de AFN



Uma palavra pode gerar diferentes computações !!!

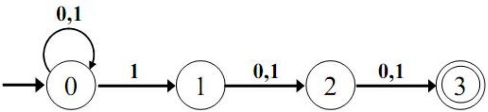
Linguagem de um AFN

Seja um AFN $M = (E, \Sigma, \delta, i, F)$. A linguagem reconhecida por M é

$$L(M) = \{w \in \Sigma^* \mid \exists \text{ computação } [i, w] \overset{*}{\vdash} [f, \lambda] \text{ para algum } f \in F\}.$$

Autômatos Finitos Não Determinísticos

Comportamento de AFN



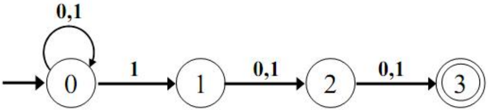
Pode-se representar o processo de reconhecimento da sentença $w_1 = 0100$ da seguinte forma:

$$[0, 0100] \vdash [0, 100] \vdash \begin{cases} [1, 00] \vdash [2, 0] \vdash [3, \lambda] & \text{(Aceita)} \\ [0, 00] \vdash [0, 0] \vdash [0, \lambda] & \text{(Rejeita)} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



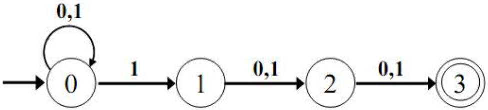
Pode-se representar o processo de reconhecimento da sentença $w_1 = 0100$ da seguinte forma:

$$[0, 0100] \vdash [0, 100] \vdash \begin{cases} [1, 00] \\ [0, 00] \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



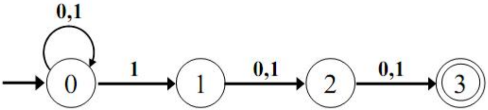
Pode-se representar o processo de reconhecimento da sentença $w_1 = 0100$ da seguinte forma:

$$[0, 0100] \vdash [0, 100] \vdash \begin{cases} [1, 00] \vdash [2, 0] \vdash [3, \lambda] & \textbf{(Aceita)} \\ [0, 00] \vdash [0, 0] \vdash [0, \lambda] & \textbf{(Rejeita)} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



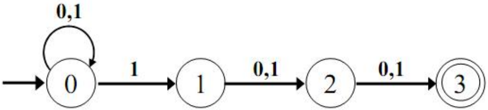
Pode-se representar o processo de reconhecimento da sentença $w_1 = 0100$ da seguinte forma:

$$[0, 0100] \vdash [0, 100] \vdash \begin{cases} [1, 00] \vdash [2, 0] \vdash [3, \lambda] & \textbf{(Aceita)} \\ [0, 00] \vdash [0, 0] \vdash [0, \lambda] & \textbf{(Rejeita)} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



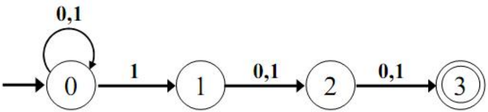
Pode-se representar o processo de reconhecimento da sentença $w_1 = 0100$ da seguinte forma:

$$[0, 0100] \vdash [0, 100] \vdash \begin{cases} [1, 00] \vdash [2, 0] \vdash [3, \lambda] & \textbf{(Aceita)} \\ [0, 00] \vdash [0, 0] \vdash [0, \lambda] & \textbf{(Rejeita)} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



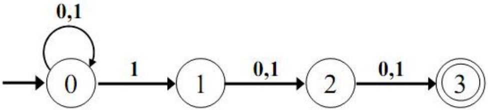
Pode-se representar o processo de reconhecimento da sentença $w_1 = 0100$ da seguinte forma:

$$[0, 0100] \vdash [0, 100] \vdash \begin{cases} [1, 00] \vdash [2, 0] \vdash [3, \lambda] & \textbf{(Aceita)} \\ [0, 00] \vdash [0, 0] \vdash [0, \lambda] & \textbf{(Rejeita)} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



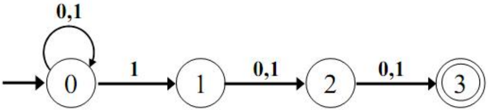
Pode-se representar o processo de reconhecimento da sentença $w_1 = 0100$ da seguinte forma:

$$[0, 0100] \vdash [0, 100] \vdash \begin{cases} [1, 00] \vdash [2, 0] \vdash [3, \lambda] & \textbf{(Aceita)} \\ [0, 00] \vdash [0, 0] \vdash [0, \lambda] & \textbf{(Rejeita)} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



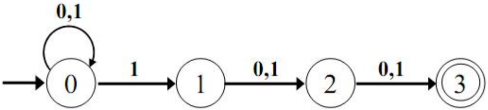
Pode-se representar o processo de reconhecimento da sentença $w_1 = 0100$ da seguinte forma:

$$[0, 0100] \vdash [0, 100] \vdash \begin{cases} [1, 00] \vdash [2, 0] \vdash [3, \lambda] & \textbf{(Aceita)} \\ [0, 00] \vdash [0, 0] \vdash [0, \lambda] & \textbf{(Rejeita)} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



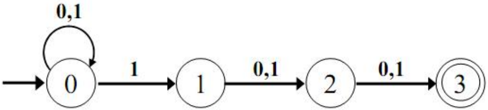
Pode-se representar o processo de reconhecimento da sentença $w_1 = 0100$ da seguinte forma:

$$[0, 0100] \vdash [0, 100] \vdash \begin{cases} [1, 00] \vdash [2, 0] \vdash [3, \lambda] & \textbf{(Aceita)} \\ [0, 00] \vdash [0, 0] \vdash [0, \lambda] & \text{(Rejeita)} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



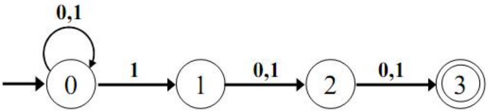
Pode-se representar o processo de reconhecimento da sentença $w_1 = 0100$ da seguinte forma:

$$[0, 0100] \vdash [0, 100] \vdash \begin{cases} [1, 00] \vdash [2, 0] \vdash [3, \lambda] & (\textbf{Aceita}) \\ [0, 00] \vdash [0, 0] \vdash [0, \lambda] & (\textbf{Rejeita}) \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



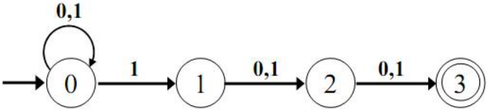
Pode-se representar o processo de reconhecimento da sentença $w_1 = 0100$ da seguinte forma:

$$[0, 0100] \vdash [0, 100] \vdash \begin{cases} [1, 00] \vdash [2, 0] \vdash [3, \lambda] & (\textbf{Aceita}) \\ [0, 00] \vdash [0, 0] \vdash [0, \lambda] & (\textbf{Rejeita}) \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



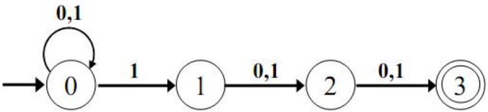
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0, 1111] \vdash \begin{cases} [1, 111] \vdash [2, 11] \vdash [3, 1] \\ [0, 111] \vdash \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



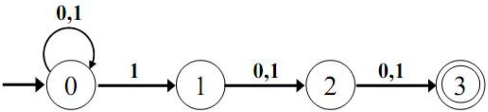
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0, 1111] \vdash \left\{ \begin{array}{llll} [1, 111] \vdash & [2, 11] \vdash & [3, 1] & \textbf{(Rejeita)} \\ & & & \textbf{(Aceita)} \\ [0, 111] \vdash & & & \textbf{(Rejeita)} \\ & & & \textbf{(Rejeita)} \\ & & & \textbf{(Rejeita)} \end{array} \right.$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



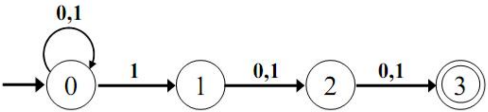
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0, 1111] \vdash \begin{cases} [1, 111] \vdash \quad [2, 11] \vdash \quad [3, 1] & \textbf{(Rejeita)} \\ & \textbf{(Aceita)} \\ & \textbf{(Rejeita)} \\ [0, 111] \vdash & \textbf{(Rejeita)} \\ & \textbf{(Rejeita)} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



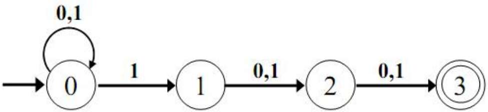
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0,1111] \vdash \begin{cases} [1,111] \vdash \quad [2,11] \vdash \quad [3,1] & \textbf{(Rejeita)} \\ [0,111] \vdash \begin{cases} [1,11] \vdash \quad [2,1] \vdash \quad [3,\lambda] & \textbf{(Aceita)} \\ [0,11] \vdash \begin{cases} & \textbf{(Rejeita)} \\ & \textbf{(Rejeita)} \\ & \textbf{(Rejeita)} \end{cases} \end{cases} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

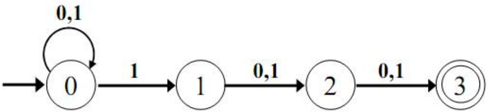
$[0, 1111] \vdash$

- $[1, 111] \vdash \quad [2, 11] \vdash \quad [3, 1]$ (Rejeita)
- $[0, 111] \vdash$
  - $[1, 11] \vdash \quad [2, 1] \vdash \quad [3, \lambda]$ (Aceita)
  - $[0, 11] \vdash$
    - (Rejeita)
    - (Rejeita)
    - (Rejeita)

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN

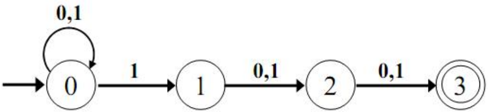


Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0, 1111] \vdash \begin{cases} [1, 111] \vdash \quad [2, 11] \vdash \quad [3, 1] & \textbf{(Rejeita)} \\ \\ [0, 111] \vdash & \end{cases}$$

Autômatos Finitos Não Determinísticos

Comportamento de AFN



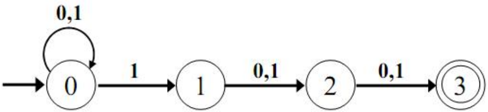
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$[0, 1111] \vdash$
- $[1, 111] \vdash \quad [2, 11] \vdash \quad [3, 1]$ **(Rejeita)**
- $[0, 111] \vdash$
  - $[1, 11] \vdash \quad [2, 1] \vdash \quad [3, \lambda]$ **(Aceita)**
  - $[0, 11] \vdash$ ... **(Rejeita)** **(Rejeita)** **(Rejeita)**

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



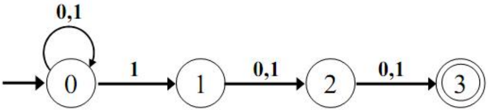
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0,1111] \vdash \begin{cases} [1,111] \vdash \quad [2,11] \vdash \quad [3,1] & \textbf{(Rejeita)} \\ [0,111] \vdash \begin{cases} [1,11] \vdash \quad [2,1] \vdash \quad [3,\lambda] & \textbf{(Aceita)} \\ [0,11] \vdash \begin{cases} [1,1] \vdash \quad [2,\lambda] & \textbf{(Rejeita)} \\ [0,1] \vdash & \textbf{(Rejeita)} \\ & \textbf{(Rejeita)} \end{cases} \end{cases} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



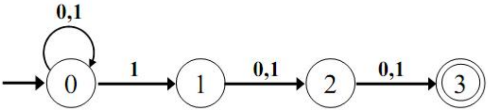
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0,1111] \vdash \begin{cases} [1,111] \vdash \quad [2,11] \vdash \quad [3,1] & \textbf{(Rejeita)} \\ [0,111] \vdash \begin{cases} [1,11] \vdash \quad [2,1] \vdash \quad [3,\lambda] & \textbf{(Aceita)} \\ & \textbf{(Rejeita)} \\ [0,11] \vdash & \textbf{(Rejeita)} \\ & \textbf{(Rejeita)} \end{cases} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

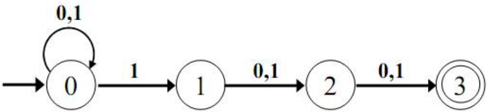
$$[0,1111] \vdash \left\{ \begin{array}{l} [1,111] \vdash \quad [2,11] \vdash \quad [3,1] \\ [0,111] \vdash \left\{ \begin{array}{l} [1,11] \vdash \quad [2,1] \vdash \quad [3,\lambda] \\ [0,11] \vdash \left\{ \begin{array}{l} [1,1] \vdash \quad [2,\lambda] \\ [0,1] \vdash \left\{ \begin{array}{l} \\ \end{array} \right. \end{array} \right. \end{array} \right. \end{array} \right.$$

(Rejeita)
(Aceita)
(Rejeita)
(Rejeita)
(Rejeita)

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



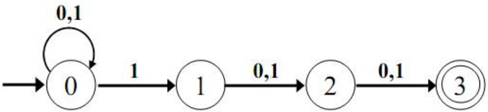
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$
[0,1111] \vdash \begin{cases} [1,111] \vdash \quad [2,11] \vdash \quad [3,1] & \textbf{(Rejeita)} \\ [0,111] \vdash \begin{cases} [1,11] \vdash \quad [2,1] \vdash \quad [3,\lambda] & \textbf{(Aceita)} \\ [0,11] \vdash \begin{cases} [1,1] \vdash \quad [2,\lambda] & \textbf{(Rejeita)} \\ [0,1] \vdash & \textbf{(Rejeita)} \\ & \textbf{(Rejeita)} \end{cases} \end{cases} \end{cases}
$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



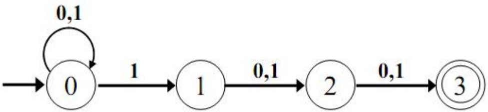
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0,1111] \vdash \begin{cases} [1,111] \vdash \quad [2,11] \vdash \quad [3,1] & \textbf{(Rejeita)} \\ [0,111] \vdash \begin{cases} [1,11] \vdash \quad [2,1] \vdash \quad [3,\lambda] & \textbf{(Aceita)} \\ [0,11] \vdash \begin{cases} [1,1] \vdash \quad [2,\lambda] & \textbf{(Rejeita)} \\ [0,1] \vdash \begin{cases} & \textbf{(Rejeita)} \\ & \textbf{(Rejeita)} \end{cases} \end{cases} \end{cases} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



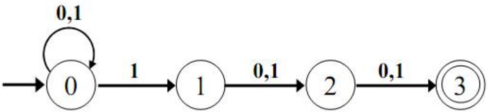
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0,1111] \vdash \begin{cases} [1,111] \vdash [2,11] \vdash [3,1] & \textbf{(Rejeita)} \\ [0,111] \vdash \begin{cases} [1,11] \vdash [2,1] \vdash [3,\lambda] & \textbf{(Aceita)} \\ [0,11] \vdash \begin{cases} [1,1] \vdash [2,\lambda] & \textbf{(Rejeita)} \\ [0,1] \vdash \begin{cases} [1,\lambda] & \textbf{(Rejeita)} \\ [0,\lambda] & \textbf{(Rejeita)} \end{cases} \end{cases} \end{cases} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



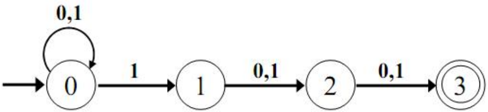
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$
[0,1111] \vdash \left\{ \begin{array}{l}
[1,111] \vdash \quad [2,11] \vdash \quad [3,1] \qquad \textbf{(Rejeita)} \\
[0,111] \vdash \left\{ \begin{array}{l}
[1,11] \vdash \quad [2,1] \vdash \quad [3,\lambda] \qquad \textbf{(Aceita)} \\
[0,11] \vdash \left\{ \begin{array}{l}
[1,1] \vdash \quad [2,\lambda] \qquad \textbf{(Rejeita)} \\
[0,1] \vdash \left\{ \begin{array}{l}
[1,\lambda] \qquad \textbf{(Rejeita)} \\
[0,\lambda] \qquad \textbf{(Rejeita)}
\end{array} \right.
\end{array} \right.
\end{array} \right.
\end{array} \right.
$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



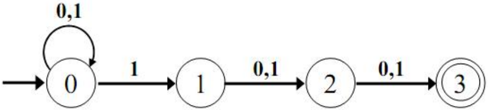
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0,1111] \vdash \begin{cases} [1,111] \vdash [2,11] \vdash [3,1] & \textbf{(Rejeita)} \\ [0,111] \vdash \begin{cases} [1,11] \vdash [2,1] \vdash [3,\lambda] & \textbf{(Aceita)} \\ [0,11] \vdash \begin{cases} [1,1] \vdash [2,\lambda] & \textbf{(Rejeita)} \\ [0,1] \vdash \begin{cases} [1,\lambda] & \textbf{(Rejeita)} \\ [0,\lambda] & \textbf{(Rejeita)} \end{cases} \end{cases} \end{cases} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



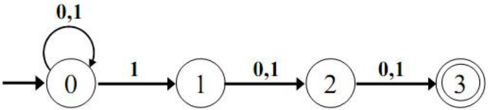
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0,1111] \vdash \begin{cases} [1,111] \vdash [2,11] \vdash [3,1] & \textbf{(Rejeita)} \\ [0,111] \vdash \begin{cases} [1,11] \vdash [2,1] \vdash [3,\lambda] & \textbf{(Aceita)} \\ [0,11] \vdash \begin{cases} [1,1] \vdash [2,\lambda] & \textbf{(Rejeita)} \\ [0,1] \vdash \begin{cases} [1,\lambda] & \textbf{(Rejeita)} \\ [0,\lambda] & \textbf{(Rejeita)} \end{cases} \end{cases} \end{cases} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



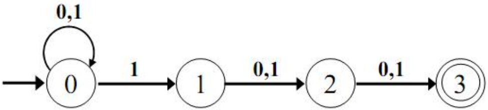
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0,1111] \vdash \begin{cases} [1,111] \vdash [2,11] \vdash [3,1] & \textbf{(Rejeita)} \\ [0,111] \vdash \begin{cases} [1,11] \vdash [2,1] \vdash [3,\lambda] & \textbf{(Aceita)} \\ [0,11] \vdash \begin{cases} [1,1] \vdash [2,\lambda] & \textbf{(Rejeita)} \\ [0,1] \vdash \begin{cases} [1,\lambda] & \textbf{(Rejeita)} \\ [0,\lambda] & \textbf{(Rejeita)} \end{cases} \end{cases} \end{cases} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



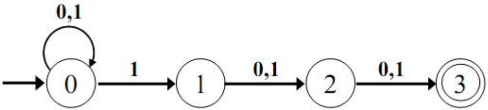
Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

$$[0,1111] \vdash \begin{cases} [1,111] \vdash [2,11] \vdash [3,1] & \textbf{(Rejeita)} \\ [0,111] \vdash \begin{cases} [1,11] \vdash [2,1] \vdash [3,\lambda] & \textbf{(Aceita)} \\ [0,11] \vdash \begin{cases} [1,1] \vdash [2,\lambda] & \textbf{(Rejeita)} \\ [0,1] \vdash \begin{cases} [1,\lambda] & \textbf{(Rejeita)} \\ [0,\lambda] & \textbf{(Rejeita)} \end{cases} \end{cases} \end{cases} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

Comportamento de AFN



Pode-se representar o processo de reconhecimento da sentença $w_2 = 1111$ da seguinte forma:

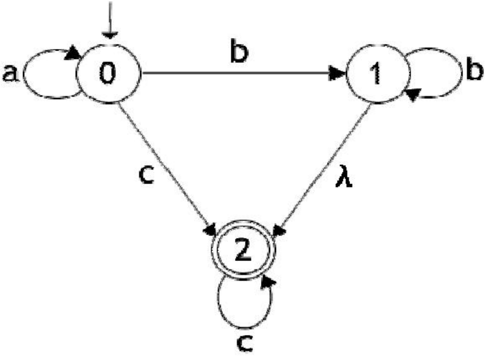
$$[0, 1111] \vdash \begin{cases} [1, 111] \vdash \quad [2, 11] \vdash \quad [3, 1] & \textbf{(Rejeita)} \\ [0, 111] \vdash \begin{cases} [1, 11] \vdash \quad [2, 1] \vdash \quad [3, \lambda] & \textbf{(Aceita)} \\ [0, 11] \vdash \begin{cases} [1, 1] \vdash \quad [2, \lambda] & \textbf{(Rejeita)} \\ [0, 1] \vdash \begin{cases} [1, \lambda] & \textbf{(Rejeita)} \\ [0, \lambda] & \textbf{(Rejeita)} \end{cases} \end{cases} \end{cases} \end{cases}$$

OBS: Iniciou-se com a sentença completa no estado inicial e existe um caminho no qual se terminou em um estado de aceitação com a sentença completamente lida (consumida) !

Autômatos Finitos Não Determinísticos

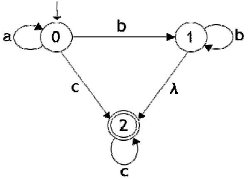
Transição Vazia (ou Transição λ)

Transição λ é aquela realizada sem a necessidade de se utilizar nenhum símbolo de entrada.



Autômatos Finitos Não Determinísticos

Comportamento de AFN-λ

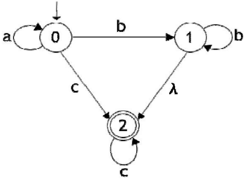


Pode-se representar o processo de reconhecimento da sentença $w = abbcc$ da seguinte forma:

$[0, abbcc] \vdash [0, bbcc] \vdash$

Autômatos Finitos Não Determinísticos

Comportamento de AFN-λ

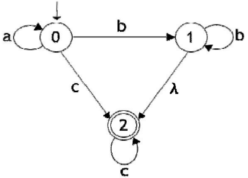


Pode-se representar o processo de reconhecimento da sentença $w = abbcc$ da seguinte forma:

$$[0, abbcc] \vdash [0, bbcc] \vdash \begin{cases} [2, bcc] & \text{(Rejeita)} \\ [1, bcc] \vdash \begin{cases} \\ \\ \end{cases} & \begin{array}{l} \text{(Rejeita)} \\ \text{(Aceita)} \end{array} \end{cases}$$

Autômatos Finitos Não Determinísticos

Comportamento de AFN-λ

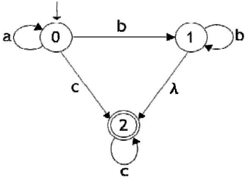


Pode-se representar o processo de reconhecimento da sentença $w = abbcc$ da seguinte forma:

$$[0, abbcc] \vdash [0, bbcc] \vdash \begin{cases} [2, bcc] & \textbf{(Rejeita)} \\ [1, bcc] \vdash \begin{cases} [1, cc] & \textbf{(Rejeita)} \\ [2, cc] \vdash [2, c] \vdash [2, \lambda] & \textbf{(Aceita)} \end{cases} \end{cases}$$

Autômatos Finitos Não Determinísticos

Comportamento de AFN-λ

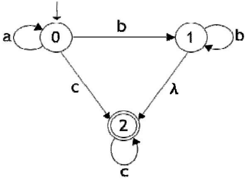


Pode-se representar o processo de reconhecimento da sentença $w = abbcc$ da seguinte forma:

$$[0, abbcc] \vdash [0, bbcc] \vdash \begin{cases} [2, bcc] \\ [1, bcc] \end{cases}$$

Autômatos Finitos Não Determinísticos

Comportamento de AFN-λ

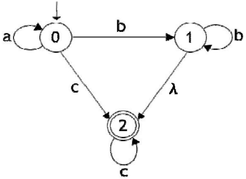


Pode-se representar o processo de reconhecimento da sentença $w = abbcc$ da seguinte forma:

$$[0, abbcc] \vdash [0, bbcc] \vdash \begin{cases} [2, bcc] & & \textbf{(Rejeita)} \\ [1, bcc] \vdash & \begin{cases} [1, cc] \\ [2, cc] \vdash [2, c] \vdash [2, \lambda] \end{cases} & \begin{array}{l} \textbf{(Rejeita)} \\ \textbf{(Aceita)} \end{array} \end{cases}$$

Autômatos Finitos Não Determinísticos

Comportamento de AFN-λ

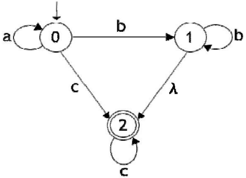


Pode-se representar o processo de reconhecimento da sentença $w = abbcc$ da seguinte forma:

$$[0, abbcc] \vdash [0, bbcc] \vdash \begin{cases} [2, bcc] & \textbf{(Rejeita)} \\ [1, bcc] \vdash \begin{cases} [1, cc] & \textbf{(Rejeita)} \\ [2, cc] \vdash [2, c] \vdash [2, \lambda] & \textbf{(Aceita)} \end{cases} \end{cases}$$

Autômatos Finitos Não Determinísticos

Comportamento de AFN-λ

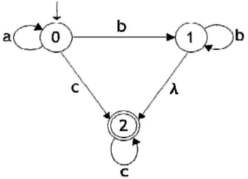


Pode-se representar o processo de reconhecimento da sentença $w = abbcc$ da seguinte forma:

$$[0, abbcc] \vdash [0, bbcc] \vdash \begin{cases} [2, bcc] & \textbf{(Rejeita)} \\ [1, bcc] \vdash \begin{cases} [1, cc] \\ [2, cc] \vdash [2, c] \vdash [2, \lambda] \end{cases} & \begin{matrix} \textbf{(Rejeita)} \\ \textbf{(Aceita)} \end{matrix} \end{cases}$$

Autômatos Finitos Não Determinísticos

Comportamento de AFN-λ

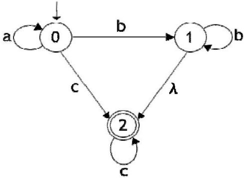


Pode-se representar o processo de reconhecimento da sentença $w = abbcc$ da seguinte forma:

$$[0, abbcc] \vdash [0, bbcc] \vdash \begin{cases} [2, bcc] & \textbf{(Rejeita)} \\ [1, bcc] \vdash \begin{cases} [1, cc] & \textbf{(Rejeita)} \\ [2, cc] \vdash [2, c] \vdash [2, \lambda] & \textbf{(Aceita)} \end{cases} \end{cases}$$

Autômatos Finitos Não Determinísticos

Comportamento de AFN-λ

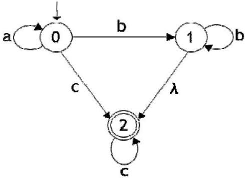


Pode-se representar o processo de reconhecimento da sentença $w = abbcc$ da seguinte forma:

$$[0, abbcc] \vdash [0, bbcc] \vdash \begin{cases} [2, bcc] & \textbf{(Rejeita)} \\ [1, bcc] \vdash \begin{cases} [1, cc] & \textbf{(Rejeita)} \\ [2, cc] \vdash [2, c] \vdash [2, \lambda] & \textbf{(Aceita)} \end{cases} \end{cases}$$

Autômatos Finitos Não Determinísticos

Comportamento de AFN-λ

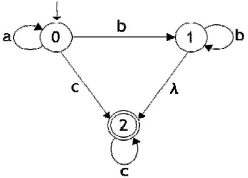


Pode-se representar o processo de reconhecimento da sentença $w = abbcc$ da seguinte forma:

$$[0, abbcc] \vdash [0, bbcc] \vdash \begin{cases} [2, bcc] & \textbf{(Rejeita)} \\ [1, bcc] \vdash \begin{cases} [1, cc] & \textbf{(Rejeita)} \\ [2, cc] \vdash [2, c] \vdash [2, \lambda] & \textbf{(Aceita)} \end{cases} \end{cases}$$

Autômatos Finitos Não Determinísticos

Comportamento de AFN-λ



Pode-se representar o processo de reconhecimento da sentença $w = abbcc$ da seguinte forma:

$$[0, abbcc] \vdash [0, bbcc] \vdash \begin{cases} [2, bcc] & \textbf{(Rejeita)} \\ [1, bcc] \vdash \begin{cases} [1, cc] & \textbf{(Rejeita)} \\ [2, cc] \vdash [2, c] \vdash [2, \lambda] & \textbf{(Aceita)} \end{cases} \end{cases}$$

Autômatos Finitos Não Determinísticos

Definição – AFN-λ

Um **Autômato Finitos Não Determinístico com Transição** λ (AFN-λ) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $\delta : E \times (\Sigma \cup \{\lambda\}) \mapsto \mathcal{P}(E)$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Definição – AFN-λ

Um **Autômato Finitos Não Determinístico com Transição** λ (AFN-λ) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $\delta : E \times (\Sigma \cup \{\lambda\}) \mapsto \mathcal{P}(E)$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Definição – AFN-λ

Um **Autômato Finitos Não Determinístico com Transição** λ (AFN-λ) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $\delta : E \times (\Sigma \cup \{\lambda\}) \mapsto \mathcal{P}(E)$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Definição – AFN-λ

Um **Autômato Finitos Não Determinístico com Transição** λ (AFN-λ) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $\delta : E \times (\Sigma \cup \{\lambda\}) \mapsto \mathcal{P}(E)$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Definição – AFN-λ

Um **Autômato Finitos Não Determinístico com Transição** λ (AFN-λ) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $\delta : E \times (\Sigma \cup \{\lambda\}) \mapsto \mathcal{P}(E)$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Definição – AFN-λ

Um **Autômato Finitos Não Determinístico com Transição** λ (AFN-λ) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $\delta : E \times (\Sigma \cup \{\lambda\}) \mapsto \mathcal{P}(E)$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

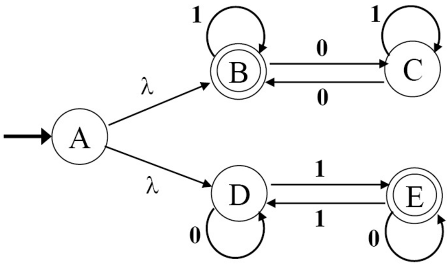
Definição – AFN-λ

Um **Autômato Finitos Não Determinístico com Transição** λ (AFN-λ) é uma quíntupla $M = (E, \Sigma, \delta, i, F)$ em que:

- $E \equiv$ conjunto não vazio de estados
- $\Sigma \equiv$ alfabeto de entrada
- $\delta \equiv$ **função de transição:**
  $\delta : E \times (\Sigma \cup \{\lambda\}) \mapsto \mathcal{P}(E)$
- $i \equiv$ estado inicial ($i \in E$)
- $F \equiv$ conjunto de estados finais ($F \subseteq E$)

Autômatos Finitos Não Determinísticos

Outro Exemplo de AFN-λ

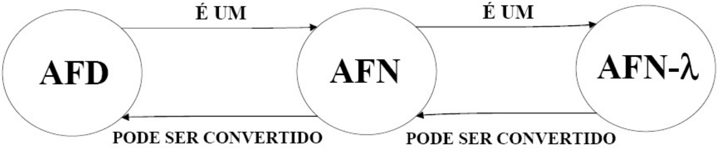


$$L = \{w \in \{0,1\}^* \mid n_0(w) \text{ é par } \textbf{ou } n_1(w) \text{ é ímpar}\},$$

em que $n_s(w)$ representa o número de símbolos s em w.

Equivalência entre Autômatos Finitos

Equivalência



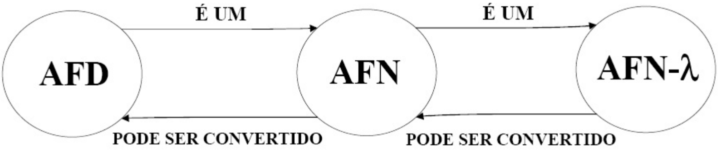
Conversão de AFs

Conversão AFN-λ $\rightarrow$ AFN : Remoção do não-determinismo

Conversão AFN $\rightarrow$ AFD : Simulação determinística

Equivalência entre Autômatos Finitos

Equivalência



Conversão de AFs

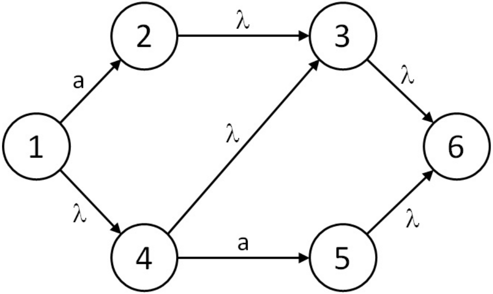
Conversão AFN-λ $\rightarrow$ AFN : Remoção do não-determinismo

Conversão AFN $\rightarrow$ AFD : Simulação determinística

Equivalência entre Autômatos Finitos

Conversão AFN-$\lambda \rightarrow$ AFN

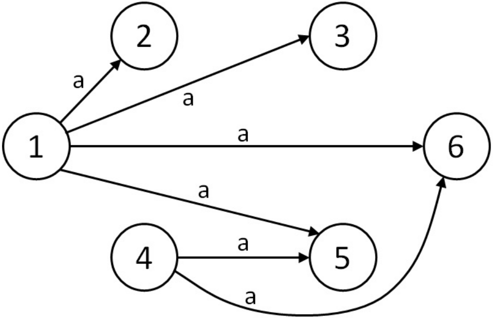
Que estados podem ser alcançados a partir do estado **1** quando *a* é lido na entrada?



Equivalência entre Autômatos Finitos

Conversão AFN-λ $\rightarrow$ AFN

Que estados podem ser alcançados a partir do estado **1** quando *a* é lido na entrada? **RESPOSTA**



Equivalência entre Autômatos Finitos

Fecho-λ

A função fecho-λ de um estado (representada por $f\lambda$) é o conjunto de todos os estados que podem ser alcançados a partir do mesmo sem a leitura de nenhum símbolo da entrada.

Definição recursiva de $f\lambda$

- **Base:** $e_i \in f\lambda(e_i), \forall e_i \in E$
- **Passo recursivo:** Se $e_j \in f\lambda(e_i)$ então $\delta(e_j, \lambda) \subseteq f\lambda(e_i)$

Obs.: É comum se estender $f\lambda$ para conjuntos de estados da seguinte forma:

$$f\lambda(E') = \bigcup_{e \in E'} f\lambda(e), \forall E' \subseteq E$$

Equivalência entre Autômatos Finitos

Fecho-λ

A função fecho-λ de um estado (representada por $f\lambda$) é o conjunto de todos os estados que podem ser alcançados a partir do mesmo sem a leitura de nenhum símbolo da entrada.

Definição recursiva de $f\lambda$

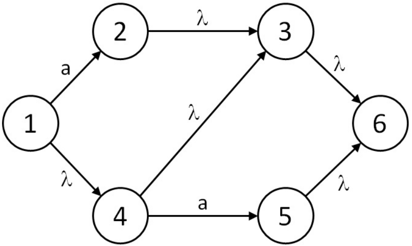
- **Base:** $e_i \in f\lambda(e_i), \forall e_i \in E$
- **Passo recursivo:** Se $e_j \in f\lambda(e_i)$ então $\delta(e_j, \lambda) \subseteq f\lambda(e_i)$

Obs.: É comum se estender $f\lambda$ para conjuntos de estados da seguinte forma:

$$f\lambda(E') = \bigcup_{e \in E'} f\lambda(e), \forall E' \subseteq E$$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



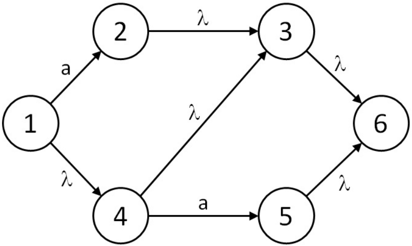
$f\lambda(1) = \{1, 4, 3, 6\}$ $\quad$ $f\lambda(4) = \{4, 3, 6\}$

$f\lambda(2) = \{2, 3, 6\}$ $\quad$ $f\lambda(5) = \{5, 6\}$

$f\lambda(3) = \{3, 6\}$ $\quad$ $f\lambda(6) = \{6\}$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



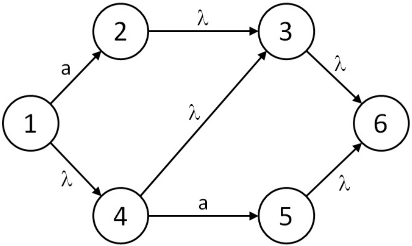
$f\lambda(1) = \{1, 4, 3, 6\}$ $\quad$ $f\lambda(4) = \{4, 3, 6\}$

$f\lambda(2) = \{2, 3, 6\}$ $\quad$ $f\lambda(5) = \{5, 6\}$

$f\lambda(3) = \{3, 6\}$ $\quad$ $f\lambda(6) = \{6\}$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



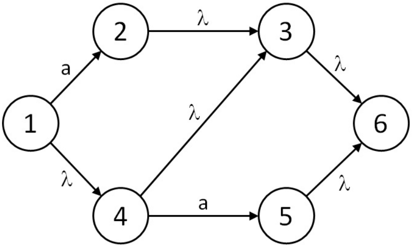
$f\lambda(1) = \{1, 4, 3, 6\}$ $f\lambda(4) = \{4, 3, 6\}$

$f\lambda(2) = \{2, 3, 6\}$ $f\lambda(5) = \{5, 6\}$

$f\lambda(3) = \{3, 6\}$ $f\lambda(6) = \{6\}$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



$f\lambda(1) = \{1, 4, 3, 6\}$

$f\lambda(2) = \{2, 3, 6\}$

$f\lambda(3) = \{3, 6\}$

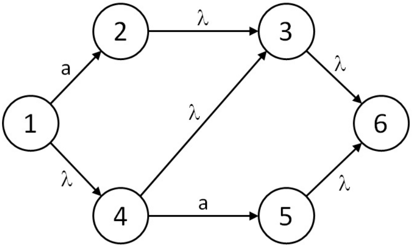
$f\lambda(4) = \{4, 3, 6\}$

$f\lambda(5) = \{5, 6\}$

$f\lambda(6) = \{6\}$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



$f\lambda(1) = \{1, 4, 3, 6\}$

$f\lambda(2) = \{2, 3, 6\}$

$f\lambda(3) = \{3, 6\}$

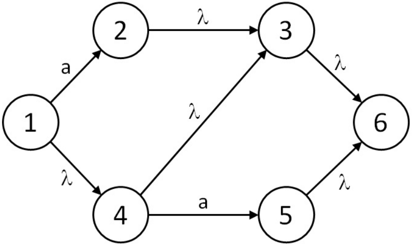
$f\lambda(4) = \{4, 3, 6\}$

$f\lambda(5) = \{5, 6\}$

$f\lambda(6) = \{6\}$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



$f\lambda(1) = \{1, 4, 3, 6\}$

$f\lambda(2) = \{2, 3, 6\}$

$f\lambda(3) = \{3, 6\}$

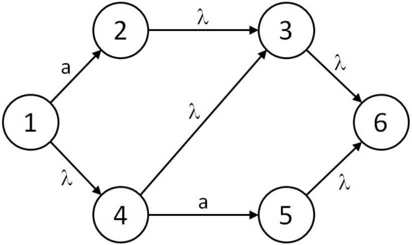
$f\lambda(4) = \{4, 3, 6\}$

$f\lambda(5) = \{5, 6\}$

$f\lambda(6) = \{6\}$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



$f\lambda(1) = \{1, 4, 3, 6\}$

$f\lambda(2) = \{2, 3, 6\}$

$f\lambda(3) = \{3, 6\}$

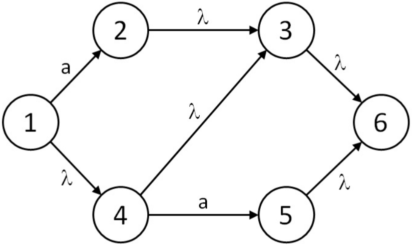
$f\lambda(4) = \{4, 3, 6\}$

$f\lambda(5) = \{5, 6\}$

$f\lambda(6) = \{6\}$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



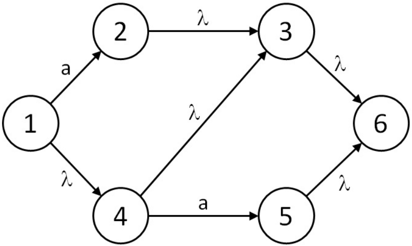
$$f\lambda(1) = \{1, 4, 3, 6\} \qquad f\lambda(4) = \{4, 3, 6\}$$

$$f\lambda(2) = \{2, 3, 6\} \qquad f\lambda(5) = \{5, 6\}$$

$$f\lambda(3) = \{3, 6\} \qquad f\lambda(6) = \{6\}$$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



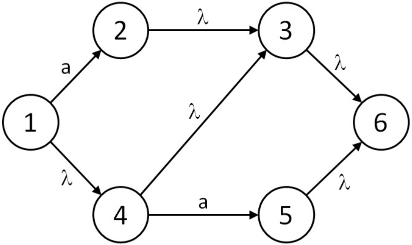
$$f\lambda(1) = \{1, 4, 3, 6\} \qquad f\lambda(4) = \{4, 3, 6\}$$

$$f\lambda(2) = \{2, 3, 6\} \qquad f\lambda(5) = \{5, 6\}$$

$$f\lambda(3) = \{3, 6\} \qquad f\lambda(6) = \{6\}$$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



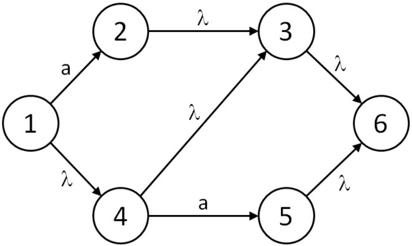
$f\lambda(1) = \{1, 4, 3, 6\}$ $\quad$ $f\lambda(4) = \{4, 3, 6\}$

$f\lambda(2) = \{2, 3, 6\}$ $\quad$ $f\lambda(5) = \{5, 6\}$

$f\lambda(3) = \{3, 6\}$ $\quad$ $f\lambda(6) = \{6\}$

Equivalência entre Autômatos Finitos

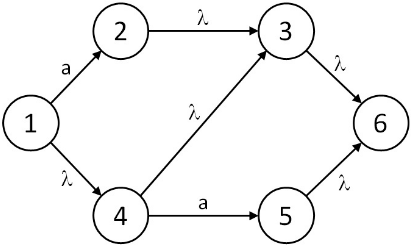
Exemplo de Cálculo de Fecho-λ



$$f\lambda(1) = \{1, 4, 3, 6\}$$
$$f\lambda(2) = \{2, 3, 6\}$$
$$f\lambda(3) = \{3, 6\}$$
$$f\lambda(4) = \{4, 3, 6\}$$
$$f\lambda(5) = \{5, 6\}$$
$$f\lambda(6) = \{6\}$$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



$f\lambda(1) = \{1, 4, 3, 6\}$

$f\lambda(2) = \{2, 3, 6\}$

$f\lambda(3) = \{3, 6\}$

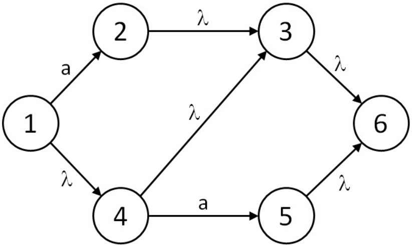
$f\lambda(4) = \{4, 3, 6\}$

$f\lambda(5) = \{5, 6\}$

$f\lambda(6) = \{6\}$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



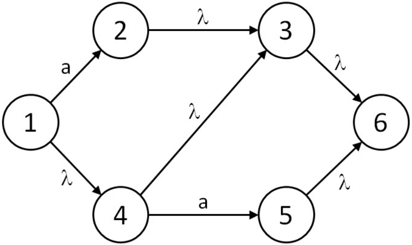
$$f\lambda(1) = \{1, 4, 3, 6\} \qquad f\lambda(4) = \{4, 3, 6\}$$

$$f\lambda(2) = \{2, 3, 6\} \qquad f\lambda(5) = \{5, 6\}$$

$$f\lambda(3) = \{3, 6\} \qquad f\lambda(6) = \{6\}$$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



$f\lambda(1) = \{1, 4, 3, 6\}$

$f\lambda(2) = \{2, 3, 6\}$

$f\lambda(3) = \{3, 6\}$

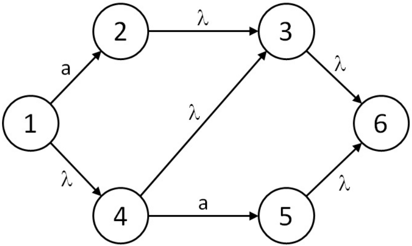
$f\lambda(4) = \{4, 3, 6\}$

$f\lambda(5) = \{5, 6\}$

$f\lambda(6) = \{6\}$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



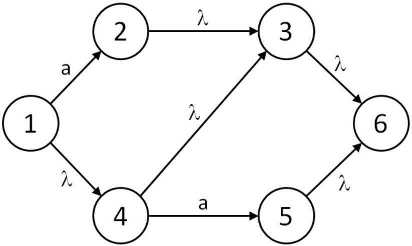
$f\lambda(1) = \{1, 4, 3, 6\}$ $f\lambda(4) = \{4, 3, 6\}$

$f\lambda(2) = \{2, 3, 6\}$ $f\lambda(5) = \{5, 6\}$

$f\lambda(3) = \{3, 6\}$ $f\lambda(6) = \{6\}$

Equivalência entre Autômatos Finitos

Exemplo de Cálculo de Fecho-λ



$$f\lambda(1) = \{1, 4, 3, 6\} \qquad f\lambda(4) = \{4, 3, 6\}$$
$$f\lambda(2) = \{2, 3, 6\} \qquad f\lambda(5) = \{5, 6\}$$
$$f\lambda(3) = \{3, 6\} \qquad f\lambda(6) = \{6\}$$

Equivalência entre Autômatos Finitos

Conversão AFN-λ $\rightarrow$ AFN

Dado um AFN-λ $M = (E, \Sigma, \delta, i, F)$, um AFN equivalente $M' = (E, \Sigma, \delta', i, F')$ pode ser construído da seguinte forma:

$$\delta'(e_i, a) = \bigcup_{r \in f\lambda(e_i)} f\lambda(\delta(r, a)) \quad , \forall e_i \in E, \forall a \in \Sigma$$

$$F' = \begin{cases} F \cup \{i\} & \text{, se } f\lambda(i) \cap F \neq \emptyset \\ F & \text{, caso contrário} \end{cases}$$

Exercício de Conversão AFN-λ $\rightarrow$ AFN

Equivalência entre Autômatos Finitos

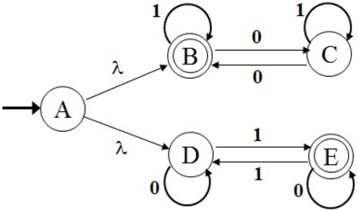
Conversão AFN-λ $\rightarrow$ AFN

Dado um AFN-λ $M = (E, \Sigma, \delta, i, F)$, um AFN equivalente $M' = (E, \Sigma, \delta', i, F')$ pode ser construído da seguinte forma:

$$\delta'(e_i, a) = \bigcup_{r \in f\lambda(e_i)} f\lambda(\delta(r, a)) \quad , \forall e_i \in E, \forall a \in \Sigma$$

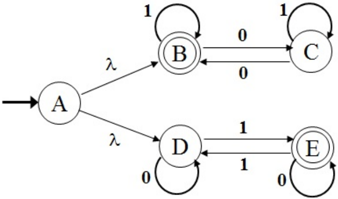
$$F' = \begin{cases} F \cup \{i\} & \text{, se } f\lambda(i) \cap F \neq \emptyset \\ F & \text{, caso contrário} \end{cases}$$

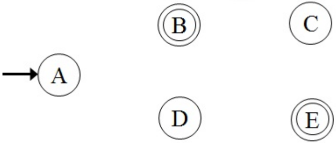
Exercício de Conversão AFN-λ $\rightarrow$ AFN



Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN

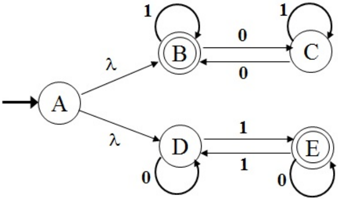


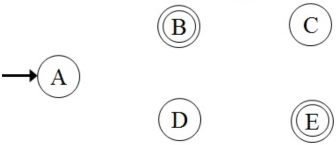


$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN

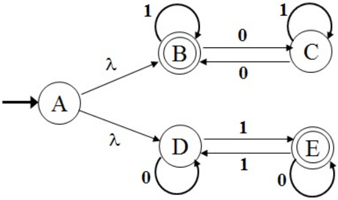


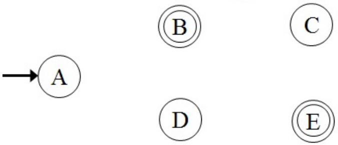


$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN



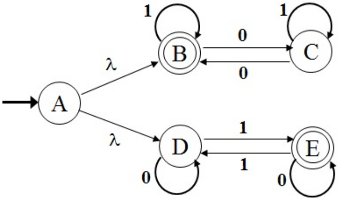


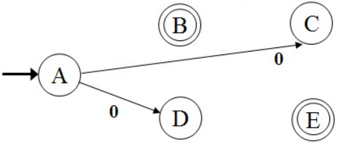
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

$$\begin{aligned}
\delta'(A, 0) &= \bigcup_{r \in f\lambda(A)} f\lambda(\delta(r, 0)) = \\
&= f\lambda(\delta(A, 0)) \cup f\lambda(\delta(B, 0)) \cup f\lambda(\delta(D, 0)) = \\
&= f\lambda(\emptyset) \cup f\lambda(\{C\}) \cup f\lambda(\{D\}) = \emptyset \cup \{C\} \cup \{D\} = \\
&= \{C, D\}
\end{aligned}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN



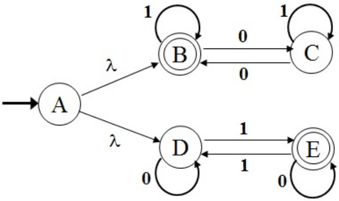


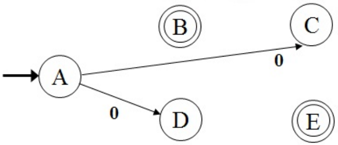
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

$$\begin{aligned}
\delta'(A,0) &= \bigcup_{r \in f\lambda(A)} f\lambda(\delta(r,0)) = \\
&= f\lambda(\delta(A,0)) \cup f\lambda(\delta(B,0)) \cup f\lambda(\delta(D,0)) = \\
&= f\lambda(\emptyset) \cup f\lambda(\{C\}) \cup f\lambda(\{D\}) = \emptyset \cup \{C\} \cup \{D\} = \\
&= \{C, D\}
\end{aligned}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN



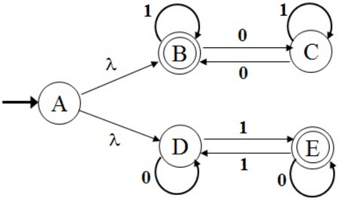


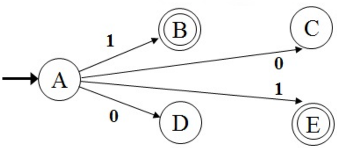
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

$$\begin{aligned}
\delta'(A,1) &= \bigcup_{r \in f\lambda(A)} f\lambda(\delta(r,1)) = \\
&= f\lambda(\delta(A,1)) \cup f\lambda(\delta(B,1)) \cup f\lambda(\delta(D,1)) = \\
&= f\lambda(\emptyset) \cup f\lambda(\{B\}) \cup f\lambda(\{E\}) = \emptyset \cup \{B\} \cup \{E\} = \\
&= \{B, E\}
\end{aligned}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN



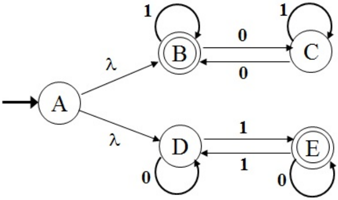


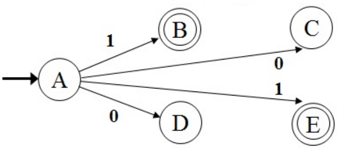
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

$$\begin{aligned}
\delta'(A,1) &= \bigcup_{r \in f\lambda(A)} f\lambda(\delta(r,1)) = \\
&= f\lambda(\delta(A,1)) \cup f\lambda(\delta(B,1)) \cup f\lambda(\delta(D,1)) = \\
&= f\lambda(\emptyset) \cup f\lambda(\{B\}) \cup f\lambda(\{E\}) = \emptyset \cup \{B\} \cup \{E\} = \\
&= \{B, E\}
\end{aligned}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN





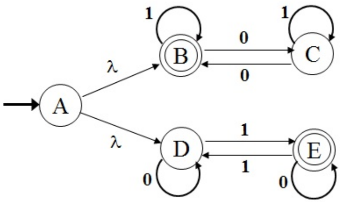
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

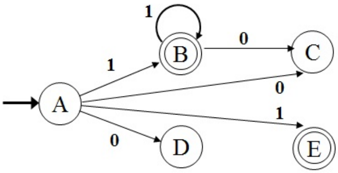
$$\delta'(B, 0) = \bigcup_{r \in f\lambda(B)} f\lambda(\delta(r, 0)) = f\lambda(\delta(B, 0)) = f\lambda(\{C\}) = \{C\}$$

$$\delta'(B, 1) = \bigcup_{r \in f\lambda(B)} f\lambda(\delta(r, 1)) = f\lambda(\delta(B, 1)) = f\lambda(\{B\}) = \{B\}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN





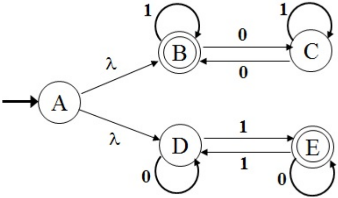
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

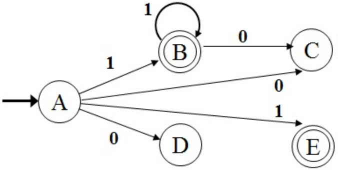
$$\delta'(B,0) = \bigcup_{r \in f\lambda(B)} f\lambda(\delta(r,0)) = f\lambda(\delta(B,0)) = f\lambda(\{C\}) = \{C\}$$

$$\delta'(B,1) = \bigcup_{r \in f\lambda(B)} f\lambda(\delta(r,1)) = f\lambda(\delta(B,1)) = f\lambda(\{B\}) = \{B\}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN





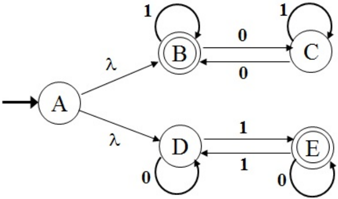
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

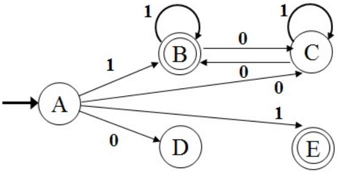
$$\delta'(C,0) = \bigcup_{r \in f\lambda(C)} f\lambda(\delta(r,0)) = f\lambda(\delta(C,0)) = f\lambda(\{B\}) = \{B\}$$

$$\delta'(C,1) = \bigcup_{r \in f\lambda(C)} f\lambda(\delta(r,1)) = f\lambda(\delta(C,1)) = f\lambda(\{C\}) = \{C\}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-$\lambda \rightarrow$ AFN





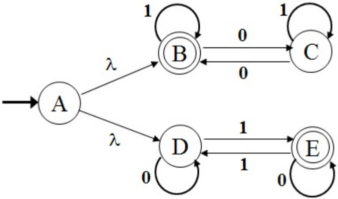
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

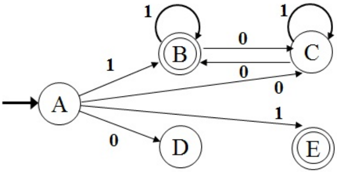
$$\delta'(C, 0) = \bigcup_{r \in f\lambda(C)} f\lambda(\delta(r, 0)) = f\lambda(\delta(C, 0)) = f\lambda(\{B\}) = \{B\}$$

$$\delta'(C, 1) = \bigcup_{r \in f\lambda(C)} f\lambda(\delta(r, 1)) = f\lambda(\delta(C, 1)) = f\lambda(\{C\}) = \{C\}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN





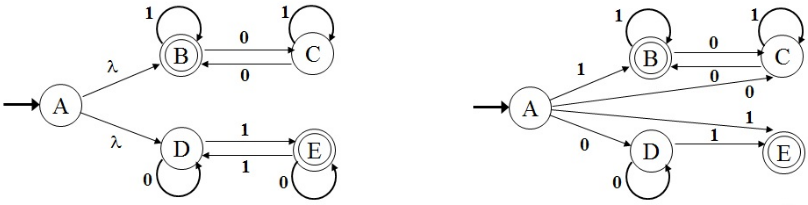
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

$$\delta'(D, 0) = \bigcup_{r \in f\lambda(D)} f\lambda(\delta(r, 0)) = f\lambda(\delta(D, 0)) = f\lambda(\{D\}) = \{D\}$$

$$\delta'(D, 1) = \bigcup_{r \in f\lambda(D)} f\lambda(\delta(r, 1)) = f\lambda(\delta(D, 1)) = f\lambda(\{E\}) = \{E\}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN



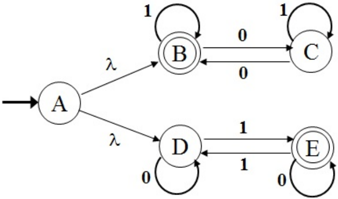
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

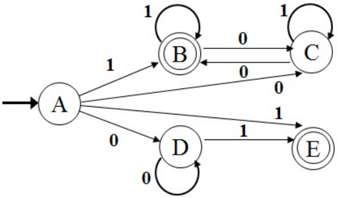
$$\delta'(D, 0) = \bigcup_{r \in f\lambda(D)} f\lambda(\delta(r, 0)) = f\lambda(\delta(D, 0)) = f\lambda(\{D\}) = \{D\}$$

$$\delta'(D, 1) = \bigcup_{r \in f\lambda(D)} f\lambda(\delta(r, 1)) = f\lambda(\delta(D, 1)) = f\lambda(\{E\}) = \{E\}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN





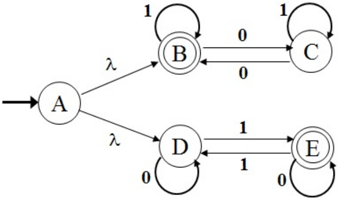
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

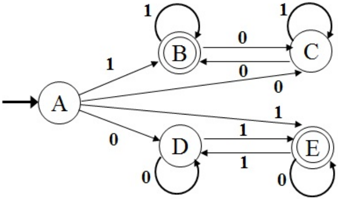
$$\delta'(E, 0) = \bigcup_{r \in f\lambda(E)} f\lambda(\delta(r, 0)) = f\lambda(\delta(E, 0)) = f\lambda(\{E\}) = \{E\}$$

$$\delta'(E, 1) = \bigcup_{r \in f\lambda(E)} f\lambda(\delta(r, 1)) = f\lambda(\delta(E, 1)) = f\lambda(\{D\}) = \{D\}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN





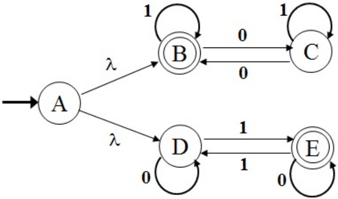
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

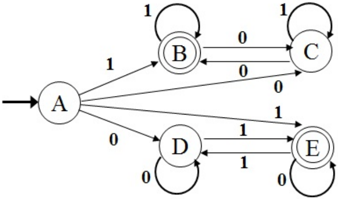
$$\delta'(E,0) = \bigcup_{r \in f\lambda(E)} f\lambda(\delta(r,0)) = f\lambda(\delta(E,0)) = f\lambda(\{E\}) = \{E\}$$

$$\delta'(E,1) = \bigcup_{r \in f\lambda(E)} f\lambda(\delta(r,1)) = f\lambda(\delta(E,1)) = f\lambda(\{D\}) = \{D\}$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-λ $\rightarrow$ AFN





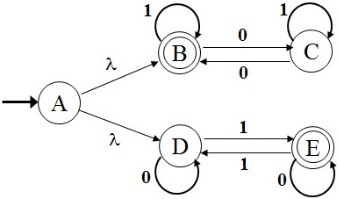
$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

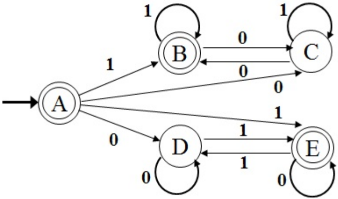
$$\begin{aligned} f\lambda(i) \cap F &= f\lambda(A) \cap F = \\ &= \{A, B, D\} \cap \{B, E\} = \\ &= \{B\} \end{aligned}$$

$$f\lambda(i) \cap F \neq \emptyset$$

Equivalência entre Autômatos Finitos

Exercício de Conversão AFN-$\lambda \rightarrow$ AFN





$$f\lambda(A) = \{A, B, D\}, f\lambda(B) = \{B\}, f\lambda(C) = \{C\}, f\lambda(D) = \{D\}, f\lambda(E) = \{E\}$$

$$\begin{aligned} f\lambda(i) \cap F &= f\lambda(A) \cap F = \\ &= \{A, B, D\} \cap \{B, E\} = \\ &= \{B\} \end{aligned}$$

$$f\lambda(i) \cap F \neq \emptyset$$

Equivalência entre Autômatos Finitos

Conversão AFN → AFD

Dado um AFN $M = (E, \Sigma, \delta, i, F)$, um AFD equivalente $M' = (E', \Sigma, \delta', i', F')$ pode ser construído da seguinte forma:

$$\begin{aligned} E' &= \mathcal{P}(E) \\ i' &= \{i\} \\ F' &= \{R \in E' \mid R \cap F \neq \emptyset\} \\ \delta'(R, a) &= \bigcup_{r \in R} \delta(r, a) \qquad , \forall R \in E', \forall a \in \Sigma \end{aligned}$$

Exercício de Conversão AFN → AFD

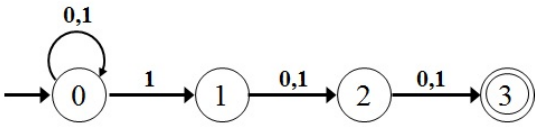
Equivalência entre Autômatos Finitos

Conversão AFN → AFD

Dado um AFN $M = (E, \Sigma, \delta, i, F)$, um AFD equivalente $M' = (E', \Sigma, \delta', i', F')$ pode ser construído da seguinte forma:

$$
\begin{aligned}
E' &= \mathcal{P}(E) \\
i' &= \{i\} \\
F' &= \{R \in E' \mid R \cap F \neq \emptyset\} \\
\delta'(R, a) &= \textstyle\bigcup_{r \in R} \delta(r, a) \qquad , \forall R \in E', \forall a \in \Sigma
\end{aligned}
$$

Exercício de Conversão AFN → AFD



Equivalência entre Autômatos Finitos

Algoritmo Iterativo para Conversão AFN $\rightarrow$ AFD

```
algoritmo
   Entrada:  AFN M = (E, Σ, δ, i, F)
   Saída   :  AFD M' = (E',Σ, δ', i', F') equivalente a M
   i' ← {i};                                   // cria estado inicial
   inserir i' em E';                           // inicia conj estados
   para todo X ∈ E' faça
      para todo a ∈ Σ faça
          Y ← ⋃_{r∈X} δ(r, a)
          se Y ∉ E' então
                 inserir Y em E';
          fim se;
          inserir [X, a, Y] em δ';             //isto é, δ'(X, a) = Y
      fim para;
   fim para;
fim algoritmo
```

Equivalência entre Autômatos Finitos

Algoritmo Iterativo para Conversão AFN $\rightarrow$ AFD

algoritmo
    Entrada: AFN $M = (E, \Sigma, \delta, i, F)$
    Saída : AFD $M' = (E', \Sigma, \delta', i', F')$ equivalente a M
    $i' \leftarrow \{i\}$; // cria estado inicial
    inserir i' em E'; // inicia conj estados
    para todo $X \in E'$ faça
        para todo $a \in \Sigma$ faça
            $Y \leftarrow \bigcup_{r \in X} \delta(r, a)$
            se $Y \notin E'$ então
                inserir Y em E';
            fim se;
            inserir $[X, a, Y]$ em δ'; //isto é, $\delta'(X, a) = Y$
        fim para;
    fim para;
fim algoritmo

Equivalência entre Autômatos Finitos

Algoritmo Iterativo para Conversão AFN $\rightarrow$ AFD

```
algoritmo
   Entrada:  AFN M = (E, Σ, δ, i, F)
   Saída  :  AFD M' = (E',Σ, δ', i', F') equivalente a M
   i' ← {i};                                  // cria estado inicial
   inserir i' em E';                          // inicia conj estados
   para todo X ∈ E' faça
      para todo a ∈ Σ faça
          Y ← ⋃_{r∈X} δ(r, a)
          se Y ∉ E' então
                 inserir Y em E';
          fim se;
          inserir [X, a, Y] em δ';            //isto é, δ'(X, a) = Y
      fim para;
   fim para;
fim algoritmo
```

Equivalência entre Autômatos Finitos

Algoritmo Iterativo para Conversão AFN → AFD

```
algoritmo
   Entrada:  AFN M = (E, Σ, δ, i, F)
   Saída  :  AFD M′ = (E′, Σ, δ′, i′, F′) equivalente a M
   i′ ← {i};                               // cria estado inicial
   inserir i′ em E′;                       // inicia conj estados
   para todo X ∈ E′ faça
      para todo a ∈ Σ faça
         Y ← ⋃_{r∈X} δ(r, a)
         se Y ∉ E′ então
               inserir Y em E′;
         fim se;
         inserir [X, a, Y] em δ′;          //isto é, δ′(X, a) = Y
      fim para;
   fim para;
fim algoritmo
```

Equivalência entre Autômatos Finitos

Algoritmo Iterativo para Conversão AFN $\rightarrow$ AFD

```
algoritmo
   Entrada:  AFN M = (E, Σ, δ, i, F)
   Saída  :  AFD M' = (E',Σ, δ', i', F') equivalente a M
   i' ← {i};                                 // cria estado inicial
   inserir i' em E';                         // inicia conj estados
   para todo X ∈ E' faça
      para todo a ∈ Σ faça
         Y ← ⋃_{r∈X} δ(r, a)
         se Y ∉ E' então
               inserir Y em E';
         fim se;
         inserir [X, a, Y] em δ';            //isto é, δ'(X, a) = Y
      fim para;
   fim para;
fim algoritmo
```

Equivalência entre Autômatos Finitos

Algoritmo Iterativo para Conversão AFN $\rightarrow$ AFD

```
algoritmo
   Entrada:  AFN M = (E, Σ, δ, i, F)
   Saída  :  AFD M' = (E', Σ, δ', i', F') equivalente a M
   i' ← {i};                                // cria estado inicial
   inserir i' em E';                        // inicia conj estados
   para todo X ∈ E' faça
      para todo a ∈ Σ faça
           Y ← ⋃_{r∈X} δ(r, a)
           se Y ∉ E' então
                  inserir Y em E';
           fim se;
           inserir [X, a, Y] em δ';          //isto é, δ'(X, a) = Y
      fim para;
   fim para;
fim algoritmo
```

Equivalência entre Autômatos Finitos

Algoritmo Iterativo para Conversão AFN $\rightarrow$ AFD

```
algoritmo
   Entrada:  AFN M = (E, Σ, δ, i, F)
   Saída   :  AFD M' = (E', Σ, δ', i', F') equivalente a M
   i' ← {i};                                  // cria estado inicial
   inserir i' em E';                          // inicia conj estados
   para todo X ∈ E' faça
      para todo a ∈ Σ faça
         Y ← ⋃_{r∈X} δ(r, a)
         se Y ∉ E' então
               inserir Y em E';
         fim se;
         inserir [X, a, Y] em δ';          //isto é, δ'(X, a) = Y
      fim para;
   fim para;
fim algoritmo
```

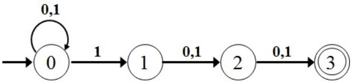
Equivalência entre Autômatos Finitos

Algoritmo Iterativo para Conversão AFN $\rightarrow$ AFD

```
algoritmo
   Entrada:  AFN M = (E, Σ, δ, i, F)
   Saída  :  AFD M' = (E',Σ, δ', i', F') equivalente a M
   i' ← {i};                                  // cria estado inicial
   inserir i' em E';                          // inicia conj estados
   para todo X ∈ E' faça
      para todo a ∈ Σ faça
          Y ← ⋃_{r∈X} δ(r, a)
          se Y ∉ E' então
                 inserir Y em E';
          fim se;
          inserir [X, a, Y] em δ';            //isto é, δ'(X, a) = Y
      fim para;
   fim para;
fim algoritmo
```

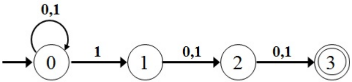
Equivalência entre Autômatos Finitos

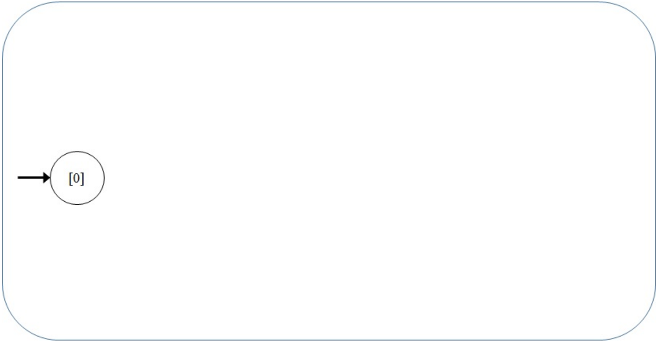
Exercício de Conversão AFN → AFD



Equivalência entre Autômatos Finitos

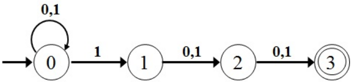
Exercício de Conversão AFN $\to$ AFD

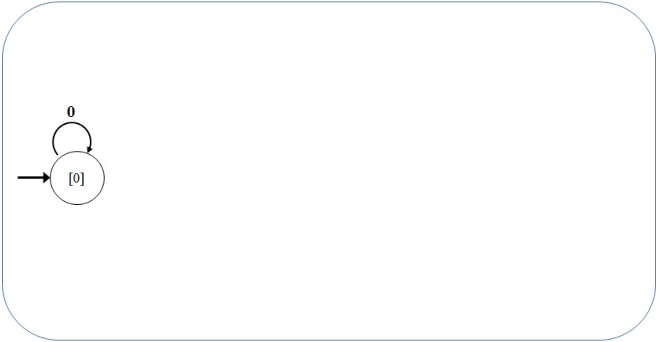




Equivalência entre Autômatos Finitos

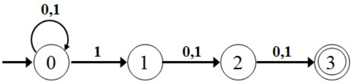
Exercício de Conversão AFN → AFD

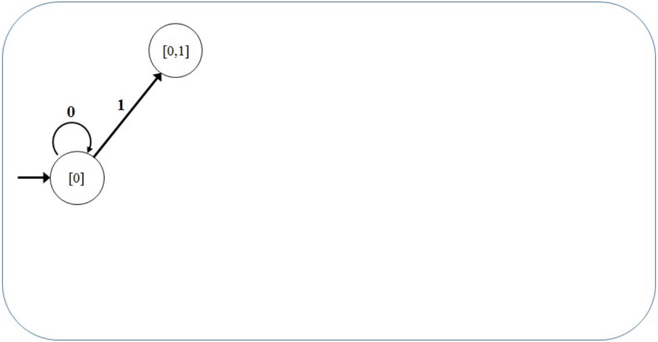




Equivalência entre Autômatos Finitos

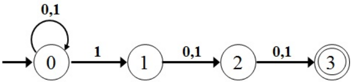
Exercício de Conversão AFN → AFD

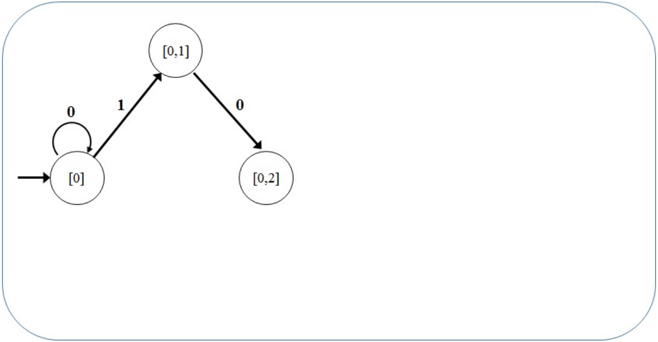




Equivalência entre Autômatos Finitos

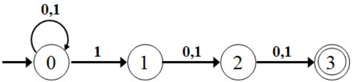
Exercício de Conversão AFN $\rightarrow$ AFD

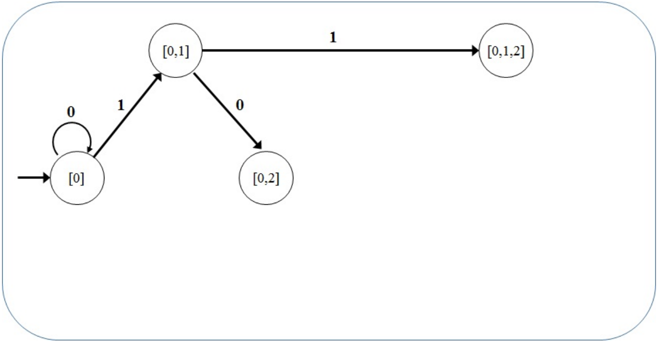




Equivalência entre Autômatos Finitos

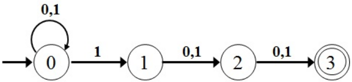
Exercício de Conversão AFN $\rightarrow$ AFD

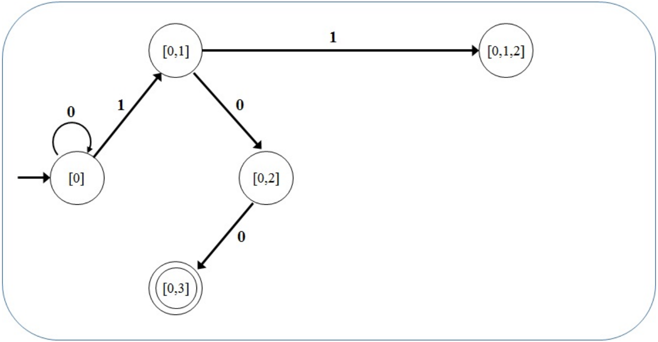




Equivalência entre Autômatos Finitos

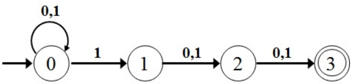
Exercício de Conversão AFN $\rightarrow$ AFD

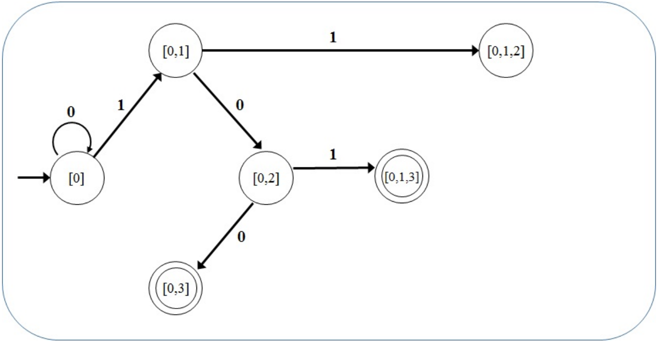




Equivalência entre Autômatos Finitos

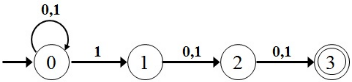
Exercício de Conversão AFN → AFD

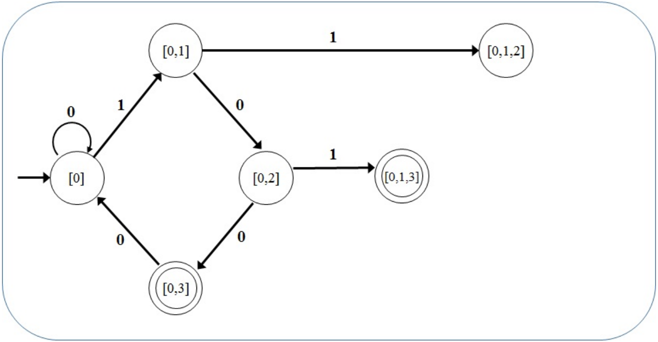




Equivalência entre Autômatos Finitos

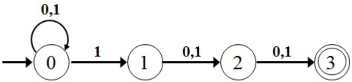
Exercício de Conversão AFN $\rightarrow$ AFD

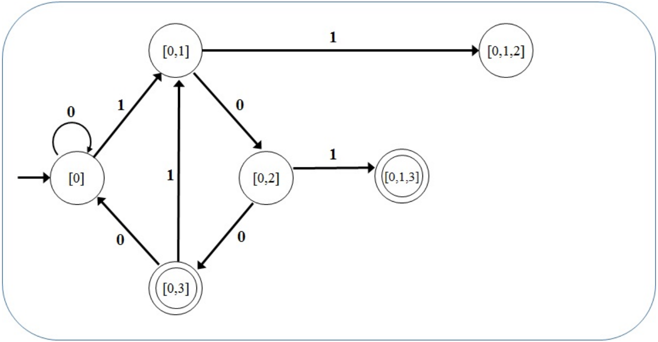




Equivalência entre Autômatos Finitos

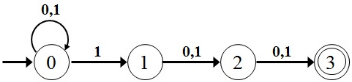
Exercício de Conversão AFN $\rightarrow$ AFD

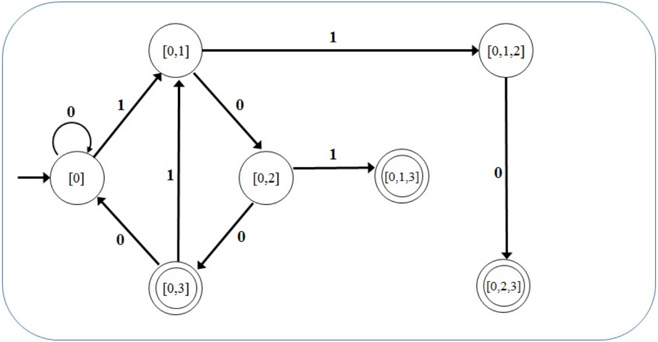




Equivalência entre Autômatos Finitos

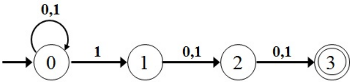
Exercício de Conversão AFN → AFD

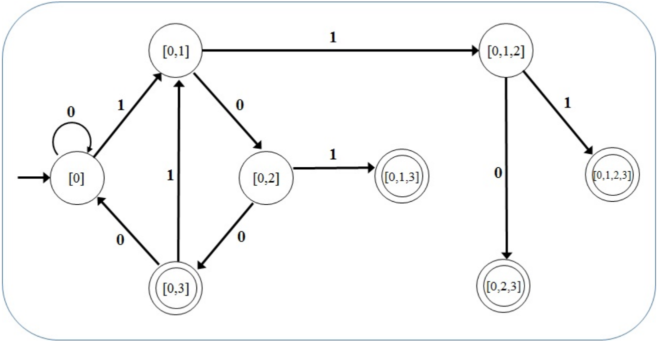




Equivalência entre Autômatos Finitos

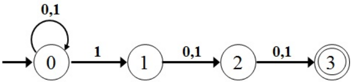
Exercício de Conversão AFN $\rightarrow$ AFD

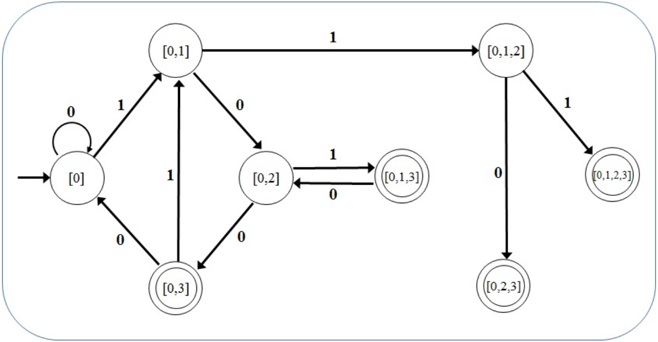




Equivalência entre Autômatos Finitos

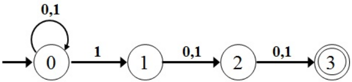
Exercício de Conversão AFN → AFD

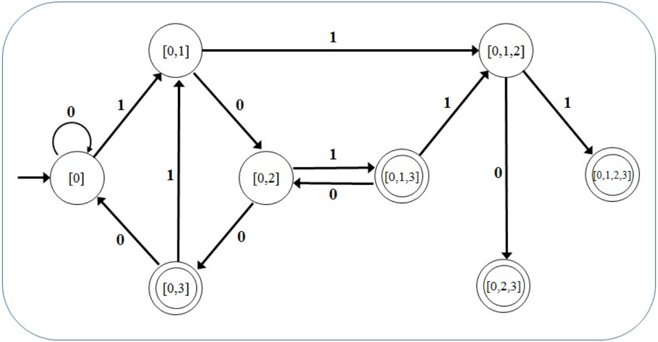




Equivalência entre Autômatos Finitos

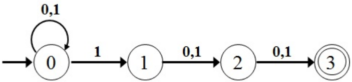
Exercício de Conversão AFN $\rightarrow$ AFD

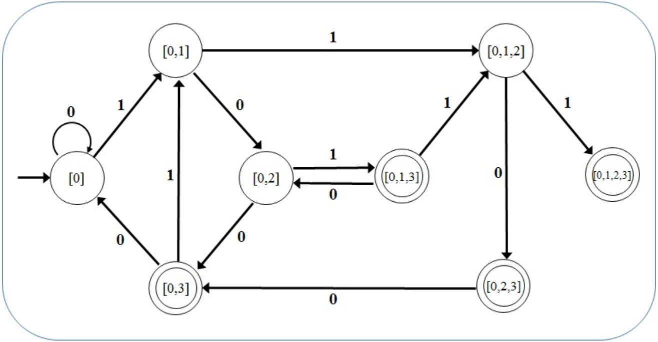




Equivalência entre Autômatos Finitos

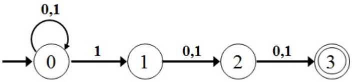
Exercício de Conversão AFN $\rightarrow$ AFD

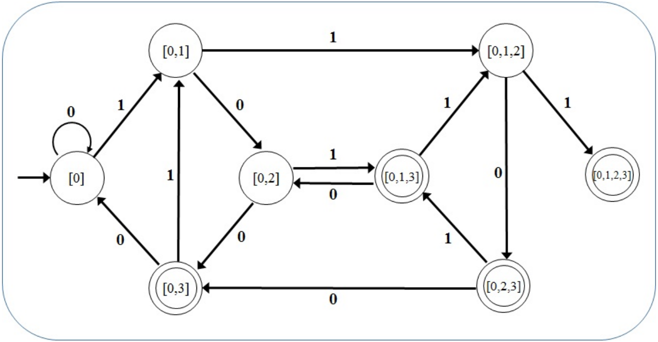




Equivalência entre Autômatos Finitos

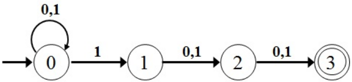
Exercício de Conversão AFN $\rightarrow$ AFD

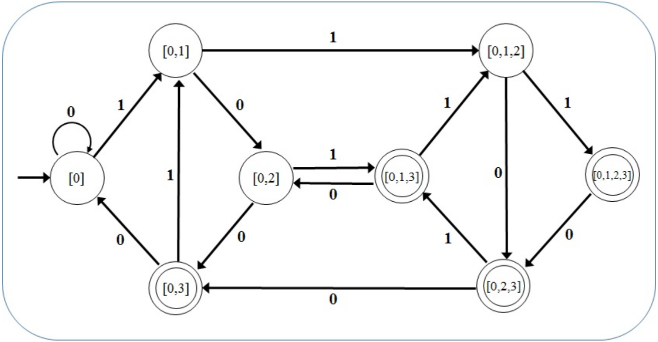




Equivalência entre Autômatos Finitos

Exercício de Conversão AFN $\rightarrow$ AFD





Equivalência entre Autômatos Finitos

Exercício de Conversão AFN → AFD