Autômatos Finitos Não-determinísticos

Não-determinísmo

Construção de Sub-Conjuntos

Não-determinísmo

- Um *autômato finito não-determinístico* tem o poder de estar em vários estados ao mesmo tempo.
  - Capacidade de "adivinhar" algo sobre sua entrada.
- Função de Transição: recebe um estado e um símbolo de entrada como argumentos, mas retorna um conjunto de zero, um ou mais estados.

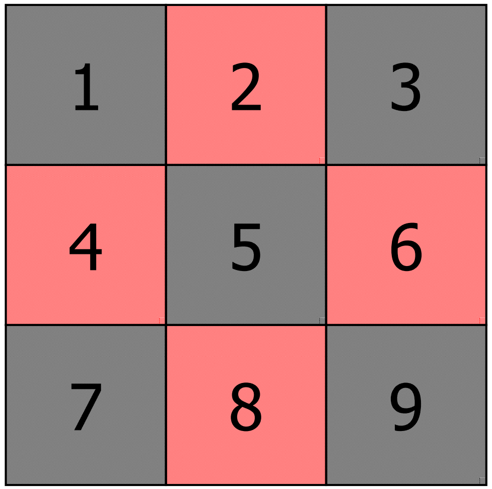
Não-determinísmo – (2)

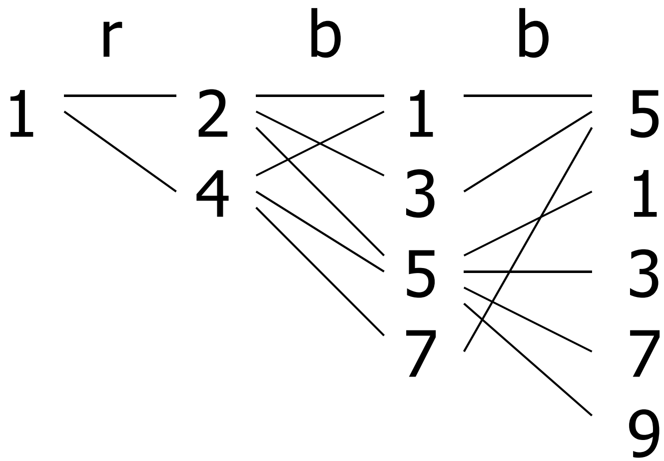
- Começa no estado inicial.
- Aceita se alguma sequência de escolhas conduzir para um estado final.
- Intuitivamente: o NFA sempre "adivinha certo."

Exemplo: Movimentos sobre um tabuleiro de xadrez

- Estados = quadrados.
- Entradas = r (mover para um quadrado adjacente vermelho) e b (mover para um quadrado adjacente preto).
- Estado inicial e estado final estão em cantos opostos.

Exemplo: Tabuleiro – (2)





		r	b
→	1	2,4	5
	2	4,6	1,3,5
	3	2,6	5
	4	2,8	1,5,7
	5	2,4,6,8	1,3,7,9
	6	2,8	3,5,9
	7	4,8	5
	8	4,6	5,7,9
*	9	6,8	5

← Aceitar, pois estado final alcançado

Autômato Finito Não-Determinístico (NFA)

- Um conjunto finito de estados, Q.
- Um conjunto finito de símbolos de entrada, Σ.
- A função de transição, δ.
- Um estado inicial, $q_0 \in Q$
- Um conjunto de estados finais, $F \subseteq Q$.

Função de Transição de um NFA

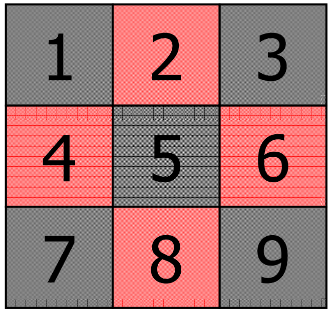
- $\delta(q, a)$ retorna um subconjunto de Q.

- Extender para strings:
- Base: $\hat{\delta}(q, \epsilon) = \{q\}$
- Indução: $\hat{\delta}(q, wa)$ = união sobre todos os estados $p \in \hat{\delta}(q, w)$ de $\delta(p_i, a)$

Linguagem de um NFA

- Uma string w é aceita por um NFA se $\hat{\delta}(q_0, w)$ contêm, <u>pelo menos</u>, um estado final.
- A linguagem de um NFA é o conjunto de strings que ele aceita.
- $L(A) = \{w \mid \hat{\delta}(q_0,w) \cap F \neq \emptyset \}$

Exemplo: Linguagem de um NFA



- Para o NFA Tabuleiro vimos que <u>rbb</u> é aceito.
- Se a entrada consite de somente b's, o conjunto de estados acessíveis alterna entre {5} e {1,3,7,9}.
- Assim somente strings não-vazias de comprimento par de b's são aceitas.

Equivalência entre DFA's e NFA's

- Um DFA pode ser transformado em um NFA que aceita a mesma linguagem.
- Se $\delta_D(q, a) = p$, faça o NFA ter $\delta_N(q, a) = \{p\}$
- O NFA é sempre um conjunto contendo exatamente um estado.

Equivalência – (2)

- Importante: Para qualquer NFA existe um DFA que aceita a mesma linguagem.
- Prova: *construção de subconjuntos*.
- O DFA pode ter exponencialmente mais estados que o NFA (raro!).
- NFA's aceitam as linguagens regulares.

Construção de Subconjuntos

- Dado um NFA com estados Q, entrada Σ, função de transição δ_N, estado incial q_0, e estados finais F, construir um DFA equivalente com:
  - Estados 2^Q (conjunto de subconjuntos de Q).
  - Entrada Σ.
  - Estado inicial $\{q_0\}$.
  - Estados finais = todos aqueles com pelo menos um estado final de F.

Ponto Crítico

- Os estados do DFA têm *nomes* que são conjuntos de estados do NFA.
- Mas como estado do DFA, uma expressão {p,q} precisa ser lida como um símbolo simples, não como um conjunto.

Construção de Subconjuntos – (2)

- A função de transição δ_D é definida por: $\delta_D(\{q_1,\ldots,q_k\}, a)$ é a união sobre todos $i = 1,\ldots,k$ de $\delta_N(q_i, a)$.
- Exercício: Construir o DFA equivalente do NFA do "tabuleiro" .

Exemplo: Construção de subconjuntos

		r	b
→	1	2,4	5
	2	4,6	1,3,5
	3	2,6	5
	4	2,8	1,5,7
	5	2,4,6,8	1,3,7,9
	6	2,8	3,5,9
	7	4,8	5
	8	4,6	5,7,9
*	9	6,8	5

		r	b
→	{1}	{2,4}	{5}
	{2,4}		
	{5}		

Alerta: O que estamos fazendo aqui é a forma *preguiçosa* de construção de DFA, onde apenas construímos um estado se formos forçados a isso.

Exemplo: Construção de subconjuntos

		r	b
→	1	2,4	5
	2	4,6	1,3,5
	3	2,6	5
	4	2,8	1,5,7
	5	2,4,6,8	1,3,7,9
	6	2,8	3,5,9
	7	4,8	5
	8	4,6	5,7,9
*	9	6,8	5

	r	b
→ {1}	{2,4}	{5}
{2,4}	{2,4,6,8}	{1,3,5,7}
{5}		
{2,4,6,8}		
{1,3,5,7}		

Exemplo: Construção de subconjuntos

		r	b
→	1	2,4	5
	2	4,6	1,3,5
	3	2,6	5
	4	2,8	1,5,7
	5	2,4,6,8	1,3,7,9
	6	2,8	3,5,9
	7	4,8	5
	8	4,6	5,7,9
*	9	6,8	5

		r	b
→	{1}	{2,4}	{5}
	{2,4}	{2,4,6,8}	{1,3,5,7}
	{5}	{2,4,6,8}	{1,3,7,9}
	{2,4,6,8}		
	{1,3,5,7}		
*	{1,3,7,9}		

Exemplo: Construção de subconjuntos

		r	b
→	1	2,4	5
	2	4,6	1,3,5
	3	2,6	5
	4	2,8	1,5,7
	5	2,4,6,8	1,3,7,9
	6	2,8	3,5,9
	7	4,8	5
	8	4,6	5,7,9
*	9	6,8	5

		r	b
→	{1}	{2,4}	{5}
	{2,4}	{2,4,6,8}	{1,3,5,7}
	{5}	{2,4,6,8}	{1,3,7,9}
	{2,4,6,8}	{2,4,6,8}	{1,3,5,7,9}
	{1,3,5,7}		
*	{1,3,7,9}		
*	{1,3,5,7,9}		

Exemplo: Construção de subconjuntos

		r	b
→	1	2,4	5
	2	4,6	1,3,5
	3	2,6	5
	4	2,8	1,5,7
	5	2,4,6,8	1,3,7,9
	6	2,8	3,5,9
	7	4,8	5
	8	4,6	5,7,9
*	9	6,8	5

		r	b
→	{1}	{2,4}	{5}
	{2,4}	{2,4,6,8}	{1,3,5,7}
	{5}	{2,4,6,8}	{1,3,7,9}
	{2,4,6,8}	{2,4,6,8}	{1,3,5,7,9}
	{1,3,5,7}	{2,4,6,8}	{1,3,5,7,9}
*	{1,3,7,9}		
*	{1,3,5,7,9}		

Exemplo: Construção de subconjuntos

		r	b
→	1	2,4	5
	2	4,6	1,3,5
	3	2,6	5
	4	2,8	1,5,7
	5	2,4,6,8	1,3,7,9
	6	2,8	3,5,9
	7	4,8	5
	8	4,6	5,7,9
*	9	6,8	5

		r	b
→	{1}	{2,4}	{5}
	{2,4}	{2,4,6,8}	{1,3,5,7}
	{5}	{2,4,6,8}	{1,3,7,9}
	{2,4,6,8}	{2,4,6,8}	{1,3,5,7,9}
	{1,3,5,7}	{2,4,6,8}	{1,3,5,7,9}
*	{1,3,7,9}	{2,4,6,8}	{5}
*	{1,3,5,7,9}		

Exemplo: Construção de subconjuntos

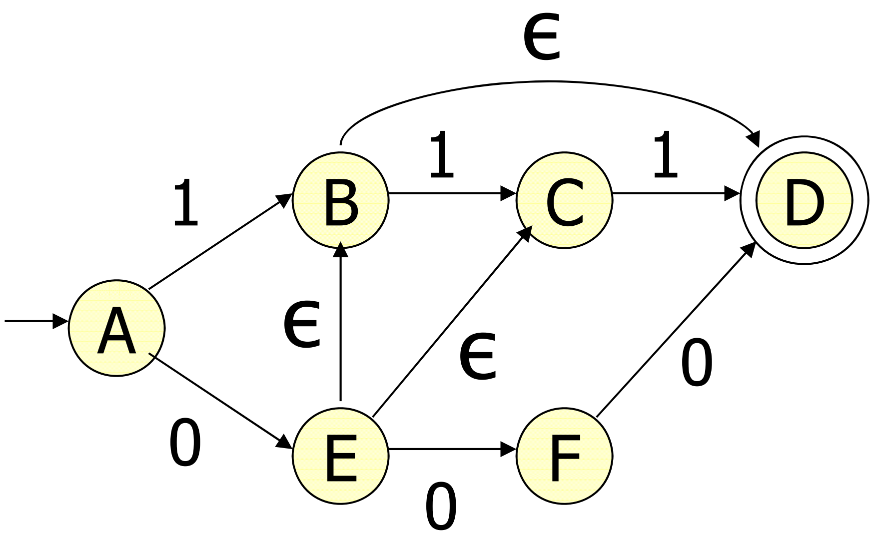
		r	b
→	1	2,4	5
	2	4,6	1,3,5
	3	2,6	5
	4	2,8	1,5,7
	5	2,4,6,8	1,3,7,9
	6	2,8	3,5,9
	7	4,8	5
	8	4,6	5,7,9
*	9	6,8	5

		r	b
→	{1}	{2,4}	{5}
	{2,4}	{2,4,6,8}	{1,3,5,7}
	{5}	{2,4,6,8}	{1,3,7,9}
	{2,4,6,8}	{2,4,6,8}	{1,3,5,7,9}
	{1,3,5,7}	{2,4,6,8}	{1,3,5,7,9}
*	{1,3,7,9}	{2,4,6,8}	{5}
*	{1,3,5,7,9}	{2,4,6,8}	{1,3,5,7,9}

NFA's com ϵ-Transições

- Podemos permitir transições de estado para estado sobre a entrada ϵ
- Essas transições são feitas de forma espontânea, sem receber um símbolo de entrada.
- É conveniente as vezes, mas ainda apenas linguagens regulares são aceitas.

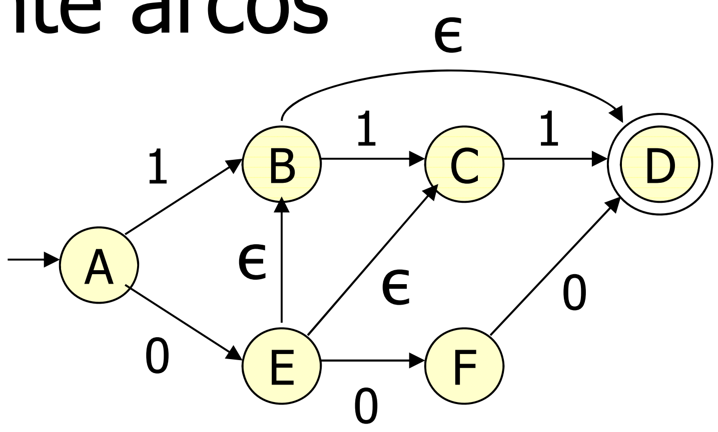
Exemplo: ϵ-NFA



		0	1	ϵ
→	A	{E}	{B}	∅
	B	∅	{C}	{D}
	C	∅	{D}	∅
*	D	∅	∅	∅
	E	{F}	∅	{B, C}
	F	{D}	∅	∅

Fechamento de Estados

- CL(q) = conjunto de estados que podem ser alcançados a partir do estado q seguindo somente arcos rotulados por ϵ.
- Exemplo: CL(A) = {A};
  CL(E) = {B, C, D, E}.

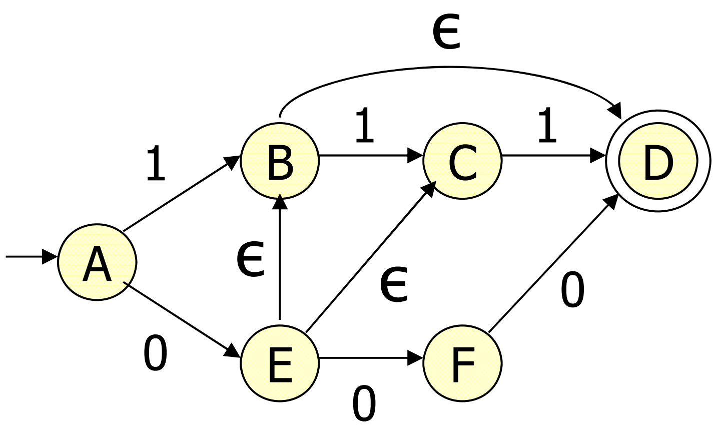


- Fechamento de um conjunto de estados = união do fechamento de cada estado.

Delta Extendido

- Base: $\hat{\delta}(q, \epsilon) = CL(q)$.
- Indução: $\hat{\delta}(q, xa)$ é assim computado:
  1. Comece com $\hat{\delta}(q, x) = S$.
  2. Faça a união de $CL(\delta(p, a))$ para todos p em S.
- Intuição: $\hat{\delta}(q, w)$ é o conjunto de estados que podem ser alcançados de q seguindo um caminho rotulado como w.

Exemplo: Delta Extêndido



- $\hat{\delta}(A, \epsilon) = CL(A) = \{A\}$.
- $\hat{\delta}(A, 0) = CL(\{E\}) = \{B, C, D, E\}$.
- $\hat{\delta}(A, 01) = CL(\{C, D\}) = \{C, D\}$.

- *Linguagem* de um ϵ-NFA é o conjunto de strings w tal que $\hat{\delta}(q_0, w)$ contêm pelo menos um estado final.

Equivalência de ϵ-NFA e DFA

- Dado qualquer ϵ–NFA E, podemos encontrar um DFA D que aceita a mesma linguagem que E.
- Usa a construção de subconjuntos.
- Diferença: precisa incorporar as ϵ–transições de E.

Eliminação de ϵ−transições

- Seja $E = (Q_E, \Sigma, \delta_E, q_E, F_E)$. Então o DFA equivalente

$$D = (Q_D, \Sigma, \delta_D, q_D, F_D)$$

é definido como:

1. Q_D é o conjunto de subconjuntos de Q_E

   Todos os estados acessíveis de D são subconjuntos com ϵ-fechamento de Q_E.

Eliminação de ϵ−transições

- Seja $E = (Q_E, \Sigma, \delta_E, q_E, F_E)$. Então o DFA equivalente

$$D = (Q_D, \Sigma, \delta_D, q_D, F_D)$$

é definido como:

2. $q_D = CL(q_E)$

Eliminação de ϵ−transições

- Seja $E = (Q_E, \Sigma, \delta_E, q_E, F_E)$. Então o DFA equivalente

$$D = (Q_D, \Sigma, \delta_D, q_D, F_D)$$

é definido como:

3. F_D representa os conjuntos de estados que contêm pelo menos um estado de aceitação de E

   $F_D = \{S \mid S \text{ está } Q_D \text{ e } S \cap F_E \neq 0\}$

Eliminação de ϵ−transições

- Seja $E = (Q_E, \Sigma, \delta_E, q_E, F_E)$. Então o DFA equivalente

$$D = (Q_D, \Sigma, \delta_D, q_D, F_D)$$

é definido como:

4. $\delta_D (S,a)$ é calculado por:
   a) Seja $S = \{p_1, p_{2,}...,p_k\}$
   b) Calcule a união $\delta_D (p_i,a) = \{r_1, r_{2,}...,r_m\}$
   c) Então, $\delta_D (S,a) =$ união $CL(r_j)$

Eliminação de ϵ−transições

- Exemplo: Eliminar as ϵ−transições do ϵ−NFA de números decimais.

- Uma linguagem L é aceita por algum ϵ−NFA se e somente se L é aceita por algum DFA.

Resumo

- DFA's, NFA's, e ϵ–NFA's todos aceitam exatamente o mesmo conjunto de linguagens: as linguagens regulares.
- Os NFA's são mais fáceis de projetar e podem ter esponencialmente menos estados que um DFA.
- Mas somente um DFA pode ser implementado!