LFA0001 – Linguagens Formais e Autômatos
Aula 09
Minimização de Autômatos Finitos

Karina Girardi Roggia
karina.roggia@udesc.br

Departamento de Ciência da Computação
Centro de Ciências Tecnológicas
Universidade do Estado de Santa Catarina

2016

Sumário

UDESC

Autômatos Finitos *versus* Complexidade

Simulador de AFD

- A cada símbolo lido da palavra de entrada
- Controla o estado atual do autômato
- Tempo de processamento: diretamente proporcional ao tamanho da palavra de entrada

Tempo de Processamento

- Não depende do autômato considerado
- Qualquer AFD que reconheça a linguagem terá a mesma eficiência

Autômatos Finitos *versus* Complexidade

Otimização possível

- minimização do número de estados
- importante para diminuir o tamanho do programa simulador

AFD Mínimo (ou Autômato Finito Mínimo)

- AFD equivalente, com o menor número de estados possível

Aplicações Específicas podem não utilizar ADF mínimo

- Nem sempre o número menor de estados dará o menor custo de implementação
- exemplo: circuitos lógicos
  - pode ser desejável introduzir estados intermediários
  - para melhorar eficiência ou facilitar ligações físicas

Unicidade do Autômato Finito Mínimo

Autômato Finito Mínimo é único

- a menos de isomorfismo
- diferenciação na identificação dos estados

Autômato Finito Mínimo

Definição (Estados Equivalentes)

Dado $M = \langle \Sigma, Q, \delta, q_0, F \rangle$ um AFD qualquer, $q, p \in Q$ são **estados equivalentes** se e somente se, para qualquer $w \in \Sigma^*$

$$\delta^*(q, w) \text{ e } \delta^*(p, w)$$

resultam simultaneamente em estados finais, ou não-finais.

Algoritmo de Minimização

- unifica os estados equivalentes

Autômato Finito Mínimo

Definição (Autômato Finito Mínimo)

Seja $L \subseteq \Sigma^*$ uma linguagem regular. O Autômato Finito Mínimo de L é um AFD

$$M_m = \langle \Sigma, Q_m, \delta_m, q_{0_m}, F_m \rangle$$

tal que:

- ACEITA$(M_m) = L$
- para qualquer AFD $M = \langle \Sigma, Q, \delta, q_0, F \rangle$ tal que ACEITA$(M) = L$

$$|Q| \geq |Q_m|$$

Pré-Requisitos

Antes de executar o Algoritmo de Minimização, o Autômato deverá

- ser determinístico
- ter somente estados alcançáveis a partir do estado inicial
- ter sua função de transição δ total

O que fazer caso um dos pré-requisitos não seja satisfeito?

Algoritmo de Minimização

Ideia de execução

- Identificar os estados equivalentes por exclusão
- Organização de tabela de estados
  - marca estados não-equivalentes
  - ao final, entradas não marcadas indicam estados equivalentes.

Algoritmo de Minimização

Seja $M = \langle \Sigma, Q, \delta, q_0, F \rangle$ um AFD que satisfaça os pré-requisitos do algoritmo de minimização.

Passo 1: Construção da Tabela: relaciona estados distintos.

q_1					
q_2					
$\ldots$					
q_{n-1}					
q_n					
	q_0	q_1	$\ldots$	q_{n-2}	q_{n-1}

Algoritmo de Minimização

Passo 2: Marcação dos Estados Trivialmente Não Equivalentes

Marcar todos os pares do tipo {estado final, estado não final}

Passo 3: Marcação dos Estados Não Equivalentes

Para $\{q_u, q_v\}$ não marcado e $a \in \Sigma$, suponha que

$$\delta(q_u, a) = p_u \text{ e } \delta(q_v, a) = p_v$$

- $p_u = p_v$
  q_u é equivalente a q_v para a: não marcar.
- $p_u \neq p_v$ e $\{p_u, p_v\}$ não está marcado
  $\{q_u, q_v\}$ incluído na lista encabeçada por $\{p_u, p_v\}$
- $p_u \neq p_v$ e $\{p_u, p_v\}$ está marcado
  - $\{q_u, q_v\}$ não é equivalente: marcar
  - se $\{q_u, q_v\}$ encabeça uma lista: marcar todos os pares da lista (e, recursivamente, se algum par da lista encabeça outra lista)

UDESC

Algoritmo de Minimização

Passo 4: Unificação dos Estados Equivalentes

Pares não marcados, após passar por todos os símbolos de Σ, são equivalentes.

- Equivalência de estados é transitiva.
- Pares de estados não finais equivalentes:
  um único estado não final
- Pares de estados finais equivalentes
  um único estado final
- se algum dos estados equivalentes é inicial
  estado unificado é inicial
- transições com origem(destino) em um estado equivalente
  origem(destino) no estado unificado

Algoritmo de Minimização

Passo 5: Exclusão dos Estados Inúteis

q é um estado inútil se

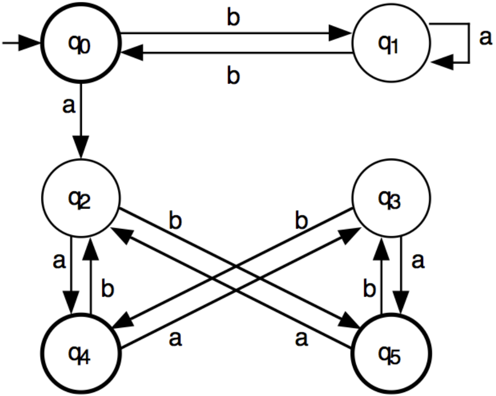
- for não final e
- a partir de q não é possível atingir um estado final

Transições com origem ou destino em estado inútil serão excluídas.

Se, originalmente, o autômato não possuía δ total, o estado incluído para cumprir o pré-requisito será um estado inútil.

UDESC

Exemplo



Passo 0 – Verificar os pré-requisitos de minimização

Exemplo

Passo 1 – Construção da tabela

Passo 2 – Marcação dos pares {estado final, estado não final}

q_1	×				
q_2	×				
q_3	×				
q_4		×	×	×	
q_5		×	×	×	
	q_0	q_1	q_2	q_3	q_4

Exemplo

Passo 3 – Análise dos pares de estados não marcados

$\{q_0, q_4\}$

$$\delta(q_0, a) = q_2 \qquad \delta(q_0, b) = q_1$$

$$\delta(q_4, a) = q_3 \qquad \delta(q_4, b) = q_2$$

- $\{q_1, q_2\}$ e $\{q_2, q_3\}$ são não marcados
  - inclui $\{q_0, q_4\}$ nas listas de $\{q_1, q_2\}$ e $\{q_2, q_3\}$

$\{q_0, q_5\}$

$$\delta(q_0, a) = q_2 \qquad \delta(q_0, b) = q_1$$

$$\delta(q_5, a) = q_2 \qquad \delta(q_5, b) = q_3$$

- $\{q_1, q_3\}$ é não marcado ($\{q_2, q_2\}$ é trivialmente equivalente)
  - inclui $\{q_0, q_5\}$ na lista de $\{q_1, q_3\}$

Exemplo

$\{q_1, q_2\}$

$$\delta(q_1, a) = q_1 \qquad \delta(q_1, b) = q_0$$

$$\delta(q_2, a) = q_4 \qquad \delta(q_2, b) = q_5$$

- $\{q_1, q_4\}$ é marcado: marca $\{q_1, q_2\}$
- $\{q_1, q_2\}$ encabeça uma lista: marca $\{q_0, q_4\}$

$\{q_1, q_3\}$

$$\delta(q_1, a) = q_1 \qquad \delta(q_1, b) = q_0$$

$$\delta(q_3, a) = q_5 \qquad \delta(q_3, b) = q_4$$

- $\{q_1, q_5\}$ e $\{q_0, q_4\}$ são marcados: marca $\{q_1, q_3\}$
- $\{q_1, q_3\}$ encabeça uma lista: marca $\{q_0, q_5\}$

UDESC

Exemplo

$\{q_2, q_3\}$

$$\delta(q_2, a) = q_4 \qquad \delta(q_2, b) = q_5$$
$$\delta(q_3, a) = q_5 \qquad \delta(q_3, b) = q_4$$

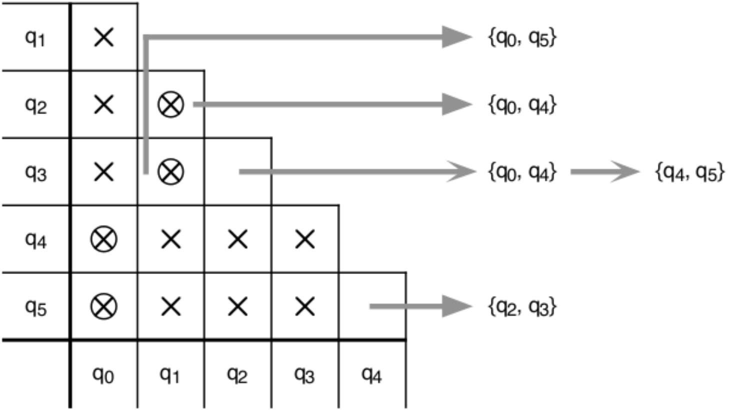
- $\{q_4, q_5\}$ é não marcado: inclui $\{q_2, q_3\}$ na lista de $\{q_4, q_5\}$

$\{q_4, q_5\}$

$$\delta(q_4, a) = q_3 \qquad \delta(q_4, b) = q_2$$
$$\delta(q_5, a) = q_2 \qquad \delta(q_5, b) = q_3$$

- $\{q_2, q_3\}$ é não marcado: inclui $\{q_4, q_5\}$ na lista de $\{q_2, q_3\}$

Exemplo

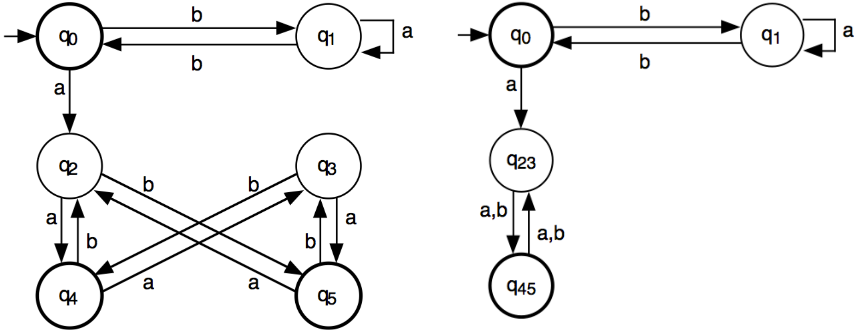


UDESC

Exemplo

Passo 4 – Unificação de estados não marcados

- q_{23}: unificação dos estados q_2 e q_3
- q_{45}: unificação dos estados finais q_4 e q_5



Mínimo e Único

O autômato construído usando o algoritmo de minimização apresentado é o Autômato Finito Mínimo para a linguagem que ele aceita.

Teorema (Unicidade do Autômato Finito Mínimo)

O Autômato Finito Mínimo de uma linguagem é único, a menos de isomorfismo.

Isomorfismo de AFD

Autômatos isomorfos

- diferenciam-se, eventualmente, na identificação dos estados
- definição formal: não será apresentada
- "único a menos de isomorfismo": referimo-nos como **o** autômato mínimo, ao invés de **um** autômato mínimo.