Teoria da Computação
Linguagens Regulares (Parte 2)
Minimização de Autômatos Finitos

Prof. Jefferson Magalhães de Morais

Minimização de autômatos

- Um importante resultado da teoria de autômatos é que sabemos da equivalência entre AFD e AFND

- Outro importante resultado: cada conjunto regular é reconhecido por um AFD **mínimo e único**
  - *Mínimo*: não há outro AF com um número inferior de estados
  - *Único*: não há dois AFs mínimos com funções de transição distintas

- Fatores importantes sobre minimização
  1. É valida apenas para a classe de linguagens definidas por AFs
  2. Permite construir reconhecedores sintáticos compactos e eficientes
  3. É possível transforma qualquer AF em uma versão equivalente mínima
  4. O AF é único para cada linguagem regular (equivalência entre linguagens)

Estados equivalentes

- Dois **estados** A e B de um autômato finito são ditos equivalentes se o conjunto de cadeias aceitas em cada um deles for o mesmo

- Considere-se a **linguagem aceita a partir de um estado** X como sendo definida da seguinte forma

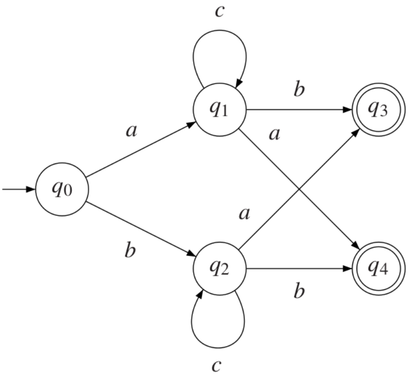
$$L(X) = \{w \in \Sigma^* \mid (X, w) \vdash^* (q_F, \varepsilon), q_F \in F\}$$

- É fácil perceber que $A \equiv B$ se e somente se $L(A) = L(B)$

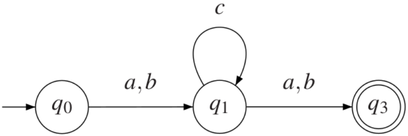
- A **inspeção visual** do autômato pode simplificar a verificação de equivalência entre estados

Uma rápida inspeção visual permite concluir que:

- $L(q_0) = (a \mid b)c^*(a \mid b)$
- $L(q_1) = c^*(a \mid b)$
- $L(q_2) = c^*(a \mid b)$
- $L(q_3) = \varepsilon$
- $L(q_4) = \varepsilon$



Portanto, como $L(q_1) = L(q_2)$ e $L(q_3) = L(q_4)$, então $q_1 \equiv q_2$ e $q_3 \equiv q_4$, e a versão mínima corresponde à apresentada



Método

- O método parte da hipótese de que o autômato a ser minimizado é **determinístico** e portanto, obviamente, isento de transições em vazio

- A minimização é feita em duas etapas
  1. Eliminação de estados inacessíveis e inúteis
  2. Agrupamento e fusão de estados equivalentes

- Em 2, criam-se **classes de equivalência** com base na coincidência do conjunto de entradas aceitas pelos possíveis pares de estados considerados

- Uma vez esgotado o processo, **escolhe-se apenas um estado desse grupo**, descartando-se os demais, uma vez que são equivalentes

Método

- O algoritmo é baseado na análise exaustiva de todos os possíveis pares de estados de um autômato M
- Por isso, torna-se conveniente representar tais pares na forma de uma **matriz**, considerando apenas a diagonal principal (inclusive) para cima, pois o par (q_i, q_j) e o par (q_j, q_i), com $i \neq j$, são idênticos

Notação

$(q_i, q_j) \xrightarrow{\sigma} (q_m, q_n)$:

- $\delta(q_i, \sigma) = q_m$, e
- $\delta(q_j, \sigma) = q_n$

	q_1	q_2	...	q_{n-1}	q_n
q_0	(q_0, q_1)	(q_0, q_2)	...	(q_0, q_{n-1})	(q_0, q_n)
q_1		(q_1, q_2)	...	(q_1, q_{n-1})	(q_1, q_n)
...				...	...
q_{n-2}				(q_{n-2}, q_{n-1})	(q_{n-2}, q_n)
q_{n-1}					(q_{n-1}, q_n)

Algoritmo (parte 1)

- O algoritmo original deve ser aplicado a partir do entendimento
  - Teoremas: *"Estados equivalentes"* e *"Autômato mínimo"*
  - Algoritmos de prova: *"Classe de equivalência"* e *"Minimização de estados"*
- Entretanto, vamos aplicar um algoritmo mais simples que requer um AF
  - Isento de transições em vazio
  - Isento de não-determinismos e
  - Isento de estados inacessíveis

- **Algoritmo**: "*Método prático para a minimização do número de estados de um autômato finito*"
  - *Entrada: Um AFD M, com função de transição total, isento de estados inacessíveis e cujos pares de estados estão dispostos na tabela apresentada*
  - *Saída: Uma partição do conjunto de estados Q de M, correspondente às maiores classes de equivalência encontradas em M*

Algoritmo (parte 2)

1. *Marcar, na tabela, todos os pares do tipo* (q_a, q_b), $q_a \in F$, $q_b \in (Q - F)$ *como não-equivalentes ("*$\not\equiv$*")*
2. *Para cada par restante* (q_a, q_b) *(escolhido arbitrariamente), fazer:*
   - *Se para toda entrada* σ *aceita por* q_a *e* q_b*:*
     - *Se* $\delta(q_a, \sigma) = \delta(q_b, \sigma)$*, ou*
     - *Se* $\delta(q_a, \sigma) \neq \delta(q_b, \sigma)$*, mas* (q_a, σ) *e* (q_b, σ) *forem equivalentes*

     Então marcar o par (q_a, q_b)*, na tabela, como equivalente ("*$\equiv$*"); caso contrário, marcar o par como não-equivalente ("*$\not\equiv$*")*
     Em seguida, deve-se verificar se existem pares cuja relação de equivalência esteja na dependência do resultado obtido e, em caso afirmativo, marcar os respectivos pares na tabela de forma correspondente
     Caso não seja possível concluir pela equivalência (ou não) de um par de estados, prosseguir com a análise de outros pares, deixando o par corrente na dependência dos resultados que forem obtidos para os demais pares
3. Marcar os pares restantes, se houver, como equivalentes ("$\equiv$")
4. A inspeção dos pares marcados indica as classes de equivalência obtidas

Considere-se o AFD com a função de transição total

	δ	a	b
$\rightarrow$	q_0	q_1	q_6
	q_1	q_2	q_3
$\leftarrow$	q_2	q_2	q_3
	q_3	q_4	q_2
$\leftarrow$	q_4	q_2	q_3
$\leftarrow$	q_5	q_4	q_5
	q_6	q_4	q_4

A tabela após a etapa 1 do método

	q_1	q_2	q_3	q_4	q_5	q_6
q_0		$\neq$		$\neq$	$\neq$	
q_1	-	$\neq$		$\neq$	$\neq$	
q_2	-	-	$\neq$			$\neq$
q_3	-	-	-	$\neq$	$\neq$	
q_4	-	-	-	-		$\neq$
q_5	-	-	-	-	-	$\neq$

Exemplos

Considera-se cada um dos pares não marcados dessa tabela (escolhidos arbitrariamente)

- $(q_0, q_1) \xrightarrow{a} (q_1, q_2) \;\not\equiv$
  q_1 e q_2 não são equivalentes (ver na tabela), portanto (q_0, q_1) é "$\not\equiv$"

- $(q_0, q_3) \xrightarrow{a} (q_1, q_4) \;\not\equiv$
  Similar ao caso anterior, portanto (q_0, q_3) é "$\not\equiv$"

- $(q_0, q_6) \xrightarrow{a} (q_1, q_4) \;\not\equiv$
  Similar ao primeiro caso, portanto (q_0, q_6) é "$\not\equiv$"

- $(q_1, q_3) \xrightarrow{a} (q_2, q_4)$?
  $(q_1, q_3) \xrightarrow{b} (q_3, q_2) \;\not\equiv$
  Apesar de não ter informação sobre (q_2, q_4), (q_3, q_2) já foi determinado, portanto (q_1, q_3) é "$\not\equiv$"

Exemplo

- $(q_1, q_6) \xrightarrow{a} (q_2, q_4)$?
  $(q_1, q_6) \xrightarrow{b} (q_3, q_4) \quad \not\equiv$
  Portanto (q_1, q_6) é "$\not\equiv$"

- $(q_2, q_4) \xrightarrow{a} (q_2, q_2) \quad \equiv$
  $(q_2, q_4) \xrightarrow{b} (q_3, q_3) \quad \equiv$
  Portanto (q_2, q_4) é "$\equiv$"

- $(q_2, q_5) \xrightarrow{a} (q_2, q_4) \quad \equiv$
  $(q_2, q_5) \xrightarrow{b} (q_3, q_5) \quad \not\equiv$
  Portanto (q_2, q_5) é "$\not\equiv$"

- $(q_3, q_6) \xrightarrow{a} (q_4, q_4) \quad \equiv$
  $(q_3, q_6) \xrightarrow{b} (q_2, q_4) \quad \equiv$
  Portanto (q_3, q_6) é "$\equiv$"

- $(q_4, q_5) \xrightarrow{a} (q_2, q_4) \quad \equiv$
  $(q_4, q_5) \xrightarrow{b} (q_3, q_5) \quad \not\equiv$
  Portanto (q_4, q_5) é "$\not\equiv$"

Exemplo

As classes de equivalência desse autômato são: $\{q_0\}, \{q_1\}, \{q_2, q_4\}, \{q_3, q_6\}$ e $\{q_5\}$. O autômato resultante possui cinco estados: $[q_0], [q_1], [q_2, q_4], [q_3, q_6]$ e $[q_5]$, e corresponde à versão mínima do autômato

	q_1	q_2	q_3	q_4	q_5	q_6
q_0	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$
q_1	-	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$	$\not\equiv$
q_2	-	-	$\not\equiv$	$\equiv$	$\not\equiv$	$\not\equiv$
q_3	-	-	-	$\not\equiv$	$\not\equiv$	$\equiv$
q_4	-	-	-	-	$\not\equiv$	$\not\equiv$
q_5	-	-	-	-	-	$\not\equiv$

	δ'	a	b
$\rightarrow$	$[q_0]$	$[q_1]$	$[q_3, q_6]$
	$[q_1]$	$[q_2, q_4]$	$[q_3, q_6]$
$\leftarrow$	$[q_2, q_4]$	$[q_2, q_4]$	$[q_3, q_6]$
	$[q_3, q_6]$	$[q_2, q_4]$	$[q_2, q_4]$
$\leftarrow$	$[q_5]$	$[q_2, q_4]$	$[q_5]$

Exemplo

As classes de equivalência desse autômato são: $\{q_0\}, \{q_1\}, \{q_2, q_4\}, \{q_3, q_6\}$ e $\{q_5\}$. O autômato resultante possui cinco estados: $[q_0], [q_1], [q_2, q_4], [q_3, q_6]$ e $[q_5]$, e corresponde à versão mínima do autômato

	δ	a	b
$\rightarrow$	q_0	q_1	q_6
	q_1	q_2	q_3
$\leftarrow$	q_2	q_2	q_3
	q_3	q_4	q_2
$\leftarrow$	q_4	q_2	q_3
$\leftarrow$	q_5	q_4	q_5
	q_6	q_4	q_4

	δ'	a	b
$\rightarrow$	$[q_0]$	$[q_1]$	$[q_3, q_6]$
	$[q_1]$	$[q_2, q_4]$	$[q_3, q_6]$
$\leftarrow$	$[q_2, q_4]$	$[q_2, q_4]$	$[q_3, q_6]$
	$[q_3, q_6]$	$[q_2, q_4]$	$[q_2, q_4]$
$\leftarrow$	$[q_5]$	$[q_2, q_4]$	$[q_5]$

Função total

- O algoritmo exige que o AF a ser minimizado possua δ total para garantir que todos os pares de estados possam sempre ser comparados em relação a todas as entradas

- Tornar total a δ que é parcial implica na incorporação de **um estado inútil ao autômato**, o qual acaba sendo agrupado com outros estados inúteis eventualmente existentes no autômato original e preservado na versão mínima correspondente

- O AF mínimo obtido é tal que δ é total

- É possível eliminar estados inúteis em um AF minimizado, resultando em um AF' com δ parcial e com menos um estado (no máximo)

Conclusões

- Dada uma linguagem regular L qualquer, então:
  1. Existe um **autômato finito mínimo** que aceita L. Em outras palavras, não existe nenhum outro autômato, com um número inferior de estados, que aceite L
  2. O autômato finito mínimo que aceita L é **único**. Isso significa que não existem dois autômatos finitos com o mesmo número de estados, porém com funções de transição distintas, que aceitam a linguagem L