Aula 05 – Linguagens Regulares

Equivalência de AFNDs AFDs

Prof. Rogério Aparecido Gonçalves[1]
rogerioag@utfpr.edu.br

[1]Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento de Computação (DACOM)
Campo Mourão - Paraná - Brasil

Bacharelado em Ciência da Computação
Ciência da Computação
BCC34B - Linguagens Formais, Autômatos e Computabilidade

Agenda i

1. Equivalência de AFNDs AFDs
2. Transformação de AFND para AFD com setas ϵ ou λ
3. Transformação de AFND para AFD sem setas ϵ ou λ
4. Fecho sobre Operações Regulares
5. Próximas Aulas
6. Referências

Equivalência de AFNDs AFDs

Equivalência de AFNDs AFDs i

- Equivalência entre Autômato Finito Não Determinístico (AFND) e Autômato Finito Determinístico (AFD).
- Autômatos são equivalentes, pois reconhecem a mesma linguagem.

Teorema

Se L é um conjunto/linguagem aceito por um Autômato Finito Não-Determinístico, então existe um Autômato Finito Determinístico que aceita L

- Seja $N = (Q, \Sigma, \delta, q_0, F)$ o AFND que reconhece alguma linguagem A. Construímos um AFD $M = (Q', \Sigma, \delta', q_0{}', F')$ que reconhece A. Antes de realizarmos a construção completa, consideremos primeiro o caso mais fácil no qual N não tem setas ϵ. Mais adiante levamos as setas ϵ em consideração.

Equivalência de AFNDs AFDs ii

1. $Q' = \mathcal{P}(Q)$ Todo estado de M é um conjunto de estados de N. Lembre-se que $\mathcal{P}(Q)$ é conjunto de subconjuntos de Q.

2. Para $R \in Q'$ e $a \in \Sigma$ seja $\delta'(R, a) = \{q \in Q \mid q \in \delta(r, a)$ para algum $r \in R\}$ Se R é um estado de M, é também um conjunto de estados de N. Quando M lê um símbolo a no estado R, ele mostra para onde a leva cada estado em R. Dado que cada estado pode ir para um conjunto de estados, tomamos a **união de todos esses conjuntos**. Outra maneira de escrever essa expressão é:

   $$\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$$

   A notação significa a união dos conjuntos $\delta(r, a)$ para cada $r \in R$.

3. ${q_0}' = \{q_0\}$ M começa no estado correspondente à coleção contendo somente o estado inicial de N. O estado inicial continua o mesmo.

4. $F' = \{R \in Q' \mid R$ contém um estado de aceitação de $N\}$. A máquina M aceita se um dos possíveis estados nos quais N poderia estar nesse ponto é um estado de aceitação.

- Consideremos as setas ϵ. Para fazer isso fixamos um pouco mais a notação. Para qualquer estado R de M, definimos $E(R)$ como a **coleção de estados que podem ser atingidos a partir de R indo somente ao longo de setas ϵ, incluindo os próprios membros de R.**

- Formalmente, para $R \subseteq Q$ seja.

  $E(R) = \{q \mid q$ *pode ser atingido a partir de* R *viajando-se ao longo de* 0 *ou mais setas* $\epsilon\}$.

- Então modificamos a função de transição de M para colocar dedos adicionais (alcançarmos) sobre todos os estados que podem ser atingidos indo ao longo de setas ϵ após cada passo. Substituindo $\delta(r, a)$ por $E(\delta(r, a))$ temos esse efeito.

- Conseqüentemente, $\delta'(R, a) = \{q \in Q \mid q \in E(\delta(r, a))$ para algum $r \in R\}$

Equivalência de AFNDs AFDs v

- Adicionalmente precisamos modificar o estado inicial de M para mover dedos inicialmente para todos os estados possíveis que podem ser atingidos a partir do estado inicial de N ao longo de setas ϵ. Mudando ${q_0}'$ para $E(\{q_0\})$ dá esse efeito. Agora completamos a construção do `AFD` M que simula o `AFND` N.
- A construção de M obviamente funciona corretamente. Em todo passo na computação de M sobre uma entrada, ela claramente entra num estado que corresponde ao subconjunto de estados nos quais N poderia estar nesse ponto. Assim, nossa prova está completa.

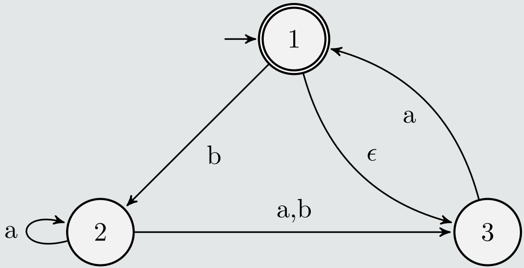
Transformação de AFND para AFD com setas ϵ ou λ

- Autômatos com setas ϵ ou λ.

Exemplo

$N_4 = (Q, \{a, b\}, \delta, 1, \{1\})$

$Q = \{1, 2, 3\}$



- Para construir um AFD D equivalente a N_4:

Exemplo de transformação de AFND para AFD ii

1. Determinar os estados de D. N_4 tem três estados, $Q = \{1, 2, 3\}$. Assim, construímos D com oito estados, um para cada subconjunto de estados de N_4.

$Q' = \mathcal{P}(Q)$ Rotulamos cada um dos estados de D com um subconjunto correspondente.

$$Q' = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$$

2. Determinar os estados inicial e de aceitação de D. O **estado inicial** é $E(\{1\})$, "o conjunto de estados que são atingíveis a partir do estado 1 por setas ϵ, mais o próprio 1." Uma seta ϵ vai de 1 para 3; portanto $E(\{1\}) = \{1, 3\}$. Os novos **estados de aceitação** são aqueles contendo o estado de aceitação de N_4. Assim, $F' = \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 2, 3\}, \}$.
3. Determinar a função de transição de D.

Exemplo de transformação de AFND para AFD iii

3.1 Cada um dos estados de D vai para um lugar dada a entrada a e um lugar quando a entrada é b.

3.2 Em D, o estado $\{2\}$ vai para $\{2, 3\}$ na entrada a, porque em N_4 o estado 2 vai para ambos 2 e 3 na entrada a e não podemos ir mais longe a partir de 2 ou 3 ao longo de setas ϵ.

3.3 O estado $\{2\}$ vai para o estado $\{3\}$ na entrada b e não podemos ir mais longe a partir de 3 com setas ϵ.

3.4 O estado $\{1\}$ vai para $\emptyset$ na entrada a, porque nenhuma seta a sai dele. Ele vai para $\{2\}$ na entrada b.

3.5 Note que o procedimento no Teorema 1.39 especifica que seguimos as sets ϵ depois que cada símbolo de entrada é lido.

3.6 O estado $\{3\}$ vai para $\{1, 3\}$ na entrada a, pois em N_4 o estado 3 vai para 1 na entrada a e 1, por sua vez, vai para 3 com uma seta ϵ.

3.7 O estado $\{3\}$ na entrada b vai para $\emptyset$.

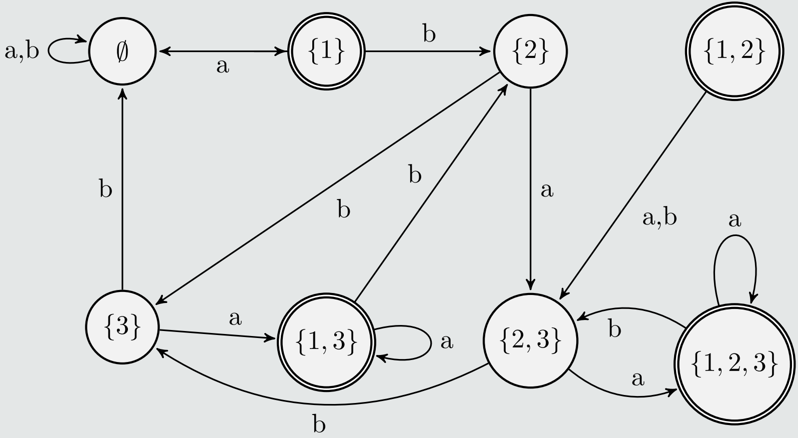
3.8 O estado $\{1, 2\}$ na entrada a vai para $\{2, 3\}$, pois 1 não aponta para nenhum estado com seta a e 2 aponta para ambos 2 e 3 com seta a e nenhum aponta para lugar algum com seta ϵ.

Exemplo de transformação de AFND para AFD iv

3.9 O estado $\{1, 2\}$ na entrada b vai para $\{2, 3\}$.
3.10 Continuar até obter o diagrama de D.

Exemplo de transformação de AFND para AFD v

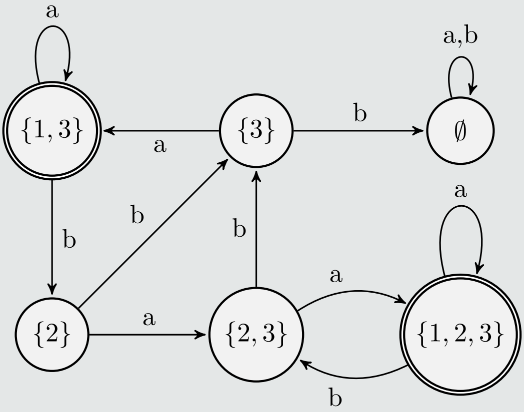
Autômato Completo



Exemplo de transformação de AFND para AFD vi

4. Simplificar o autômato obtido. Nenhuma seta aponta para os estados $\{1\}$ e $\{1, 2\}$, logo eles podem ser removidos sem afetar o desempenho da Máquina.

Chegamos ao diagrama:



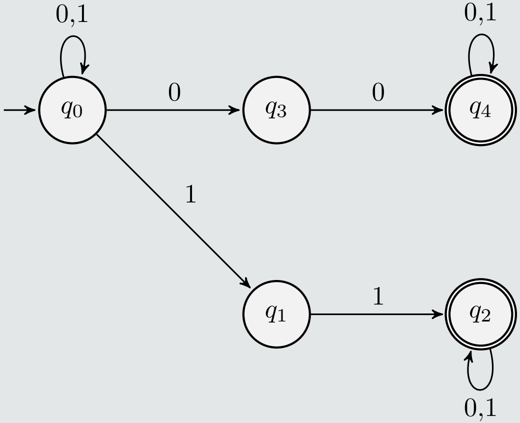
Transformação de AFND para AFD sem setas ϵ ou λ

Exemplo de transformação de AFND para AFD i

- Autômatos sem setas ϵ ou λ, mas com não-determinismo.

Exemplo

AFND que aceita o conjunto de todas as sentenças que contém dois $0's$ ou dois $1's$ consecutivos.



Passo 1: Montar a tabela de transição do AFND.

	δ	0	1
$\rightarrow$	q_0	$\{q_0, q_3\}$	$\{q_0, q_1\}$
	q_1	—	$\{q_2\}$
$\leftarrow$	q_2	$\{q_2\}$	$\{q_2\}$
	q_3	$\{q_4\}$	—
$\leftarrow$	q_4	$\{q_4\}$	$\{q_4\}$

Exemplo de transformação de AFND para AFD iv

Notação:

$\longrightarrow$: marcar o estado inicial.

$\longleftarrow$: marca os estados finais de aceitação.

Passo 2: Construir a tabela de transição do AFD. Identificar o 1o. estado do AFND e criar novos estados etiquetando-os com a concatenação dos estados do AFND alcançáveis pelas mesmas transições. Iniciando por q_0, olhamos para a tabela de transição do AFND e selecionamos a linha q_0 e coluna 0, o valor de $\delta(q_0, 0) = \{q_0, q_3\}$, que nos devolve um conjunto dos estados $alcançáveis com a entrada 0. Com os elementos do conjunto de estados, criamos um novo estado q_0q_3 com a concatenação dos elementos do conjunto. O mesmo ocorre com a entrada 1, um novo estado q_0q_1 é criado.

Exemplo de transformação de AFND para AFD v

Continuamos na construção da nova tabela de transição, seguindo a partir dos novos estados criados, até completarmos todos os estados.

δ	0	1
q_0	q_0q_3	q_0q_1
q_0q_3	$q_0q_3q_4$	q_0q_1
q_0q_1	q_0q_3	$q_0q_1q_2$
$q_0q_3q_4$	$q_0q_3q_4$	$q_0q_1q_4$
$q_0q_1q_2$	$q_0q_2q_3$	$q_0q_1q_2$
$q_0q_1q_4$	$q_0q_3q_4$	$q_0q_1q_2q_4$
$q_0q_2q_3$	$q_0q_2q_3q_4$	$q_0q_1q_2$
$q_0q_1q_2q_4$	$q_0q_2q_3q_4$	$q_0q_1q_2q_4$
$q_0q_2q_3q_4$	$q_0q_2q_3q_4$	$q_0q_1q_2q_4$

Exemplo de transformação de AFND para AFD vi

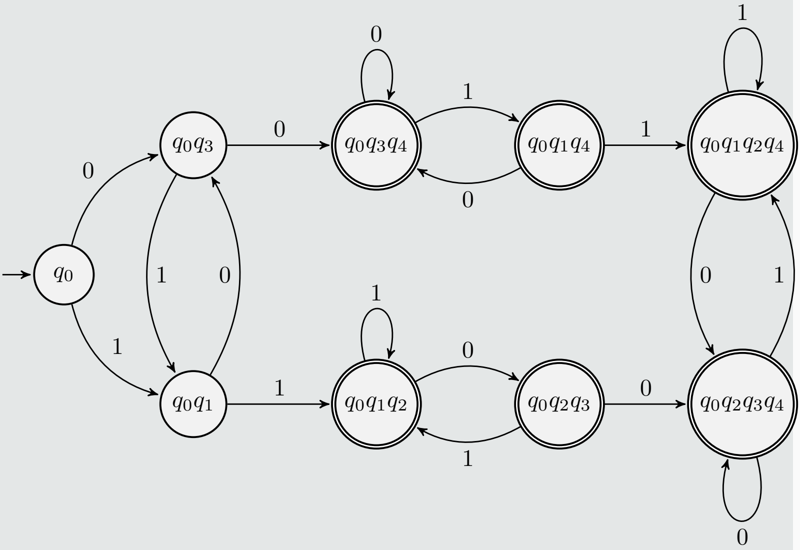
Passo 3: Para cada novo estado criado, para cada símbolo do alfabeto preencher a nova tabela de transições criando novos estados quando preciso.

Por exemplo: para o novo estado q_0q_3, segunda linha da nova tabela de transição, selecionamos as linhas da tabela de transição original para q_0 e q_3, os conjuntos $\{q_0, q_3\}$ e $\{q_4\}$ são obtidos para a entrada 0, resultando da união um novo conjunto $\{q_0, q_3, q_4\}$ que irá criar um novo estado $q_0q_3q_4$, $\delta'(q_0q_3, 0) = \{q_0, q_3, q_4\}$.

Passo 4: Todos os estados que contenham os estados finais de aceitação do AFND original, neste exemplo q_2 e q_4, serão estados de aceitação no novo AFD.

Exemplo de transformação de AFND para AFD vii

AFD Completo



Fecho sobre Operações Regulares

Fecho sobre Operações Regulares i

Teorema

A classe de Linguagens Regulares é fechada sob a operação de União.

- O fechamento quer dizer que o resultado da aplicação da operação resulta também em uma linguagem regular.
- Temos as linguagens regulares A_1 e A_2 e desejamos provar que $A_1 \cup A_2$ é regular. A idéia é tornar os dois `AFND`, N_1 e N_2 para A_1 e A_2, e combiná-los em um novo `AFND`, N.

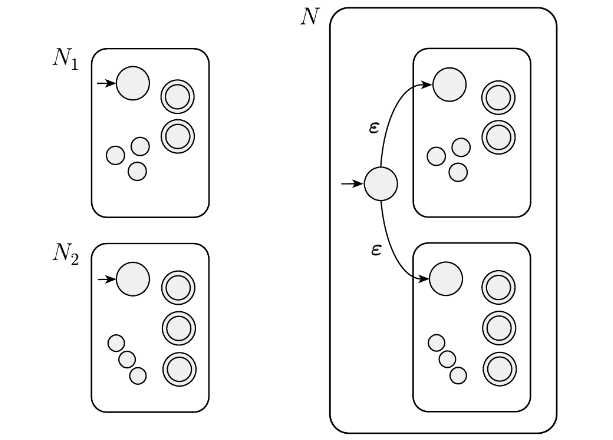


Figura 1: Construção de N para reconhecer $A_1 \cup A_2$

Prova: Suponha que $N_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ reconheça A_1 e que $N_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ que reconheça A_2.

Construa $N = (Q, \Sigma, \delta, q_0, F)$ para reconhecer $A_1 \cup A_2$.

1. $Q = \{q_0\} \cup Q_1 \cup Q_2$. Os estados de N são todos os estados de N_1 e N_2, com a adição de um novo estado inicial q_0.
2. O estado q_0 é o estado inicial de N.
3. Os estados de aceitação $F = F_1 \cup F_2$. Os estados de aceitação de N são todos os estados de aceitação de N_1 e N_2. Desta forma, N aceita se N_1 aceita ou N_2 aceita.
4. Defina δ de modo que para qualquer $q \in Q$ e qualquer $a \in \Sigma_\epsilon$,

Fecho sobre Operações Regulares iv

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0 \wedge a = \epsilon \\ \emptyset & q = q_0 \wedge a \neq \epsilon \end{cases}$$

Teorema

A classe de Linguagens Regulares é fechada sob a operação de concatenação.

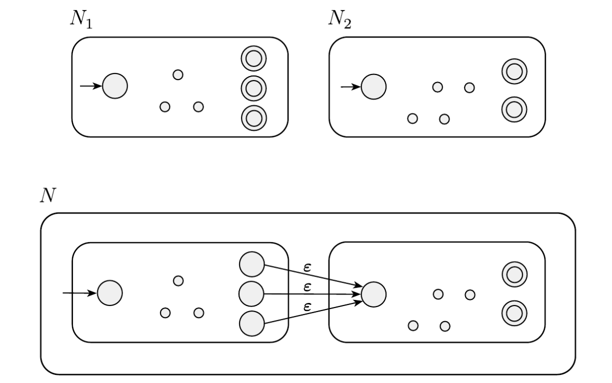


Figura 2: Construção de N para reconhecer $A_1 \circ A_2$

Fecho sobre Operações Regulares vi

Teorema

A classe de Linguagens Regulares é fechada sob a operação estrela.

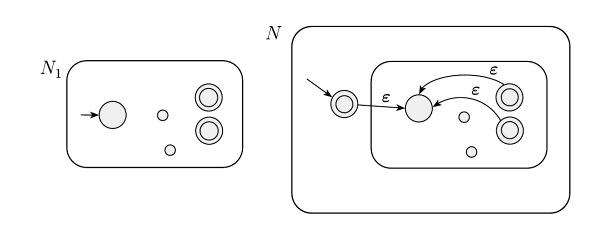


Figura 3: Construção de N para reconhecer A^*

Próximas Aulas

Próximas Aulas

- Linguagens Regulares: Equivalência e Minimização de AFDs.

Livros Texto

MENEZES, P. B. Linguagens formais e autômatos. Porto Alegre: Bookman, 2011. ISBN 9788577807994.

.

SIPSER, M. Introdução à teoria da computação. [s. l.]: Thomson Learning, 2006. ISBN 9788522104994.

Word Cloud


Determinismo
M
reader
abstract
_Word
será
edsmib
Equivalência
allowframebreaks
title
covers
da
l
Bookman
eds
Alegre
Texto
apresentado
Não
ISBN
e
lang
scope
Nesta
computação
Finitos
shib
br
authtype
md
search
Cloud_
sipser
Introdução
column
MENEZES
integrada
login
vitalbook
ramos
aspx
png
Determinísticos
Próximas
pt
minhabiblioteca
n
vbid
fragile
de
wordcloud
Regulares
site
figuras
pageid
Porto
nocite B
Aula
Referências
db
date
https
live
teoria
subtitle
trueebscohost
formais
books
Autômatos
width
o
AFND
P
direct
Thomson
Livro
Learning
AFDs
Linguagens
s
normalsize
blauth
conteúdo

Referências

Referências i

Menezes, Paulo Blauth. 2011. *Linguagens Formais e Autômatos.* Bookman. https://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000000444&lang=pt-br&site=eds-live&scope=site.

Ramos, Marcus Vinícius Midena, Ítalo Santiago Vega, and João José Neto. 2009. *Linguagens Formais: Teoria, Modelagem e Implementação.* Bookman. https://search.ebscohost.com/login.aspx?direct=true&db=cat07269a&AN=utfpr.254492&lang=pt-br&site=eds-live&scope=site.

Sipser, Michael. 2007. *Introdução à Teoria Da Computação.* Cengage Learning. https://search.ebscohost.com/login.aspx?direct=true&db=edsmib&AN=edsmib.000008725&lang=pt-br&site=eds-live&scope=site.