Aula 04 – Linguagens Regulares

Autômatos Finitos Não Determinísticos

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

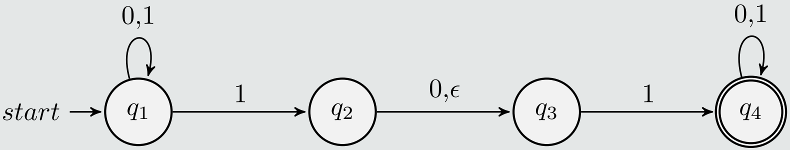
Introdução e Conceitos

Não-Determinismo i

- Não terminismo é conceito útil que tem sido de grande impacto sobre a Teoria da Computação.
- Com AFD quando uma máquina está em um dado estado e lê o próximo símbolo de entrada, sabemos qual será o estado, pois está determinado (Computação determinística).
- Em uma máquina não-determinística, várias escolhas podem existir para o próximo estado em qualquer ponto.
- Não-determinismo é uma generalização de determinismo, portanto, todo AFD é automaticamente um Autômato Finito Não Determinśitico (AFND).

Não-Determinismo ii

Exemplo (Figura 1.27)



- É perceptível a diferença entre AFD e AFND.
- Todo estado de um AFD sempre tem uma seta de transição para cada símbolo do alfabeto.
- Observemos o AFND da Figura 1.27.
- Em geral, um AFND pode ter setas rotuladas com símbolos do alfabeto ou com ϵ. Zero ou mais transições podem sair de cada estado com o rótulo ϵ.

Não-Determinismo iii

- Ainda em nosso exemplo, estando no estado q_1 e o símbolo de entrada sendo 1, após ler o símbolo a máquina se divide em múltiplas cópias de si mesma e segue todas as possibilidades em paralelo.
- Cada uma das cópias (instâncias) segue como antes. Se existirem escolhas subseqüentes, a máquina divide-se novamente.
- Se o próximo símbolo de entrada não aparece sobre qualquer das setas saindo do estado ocupado por uma instância da máquina, essa cópia da máquina morre, junto com o ramo de computação associado a ela.
- Finalmente, se *qualquer uma* das instâncias da máquina estiver em um estado de aceitação no final da entrada, o `AFND` aceita a cadeia de entrada.

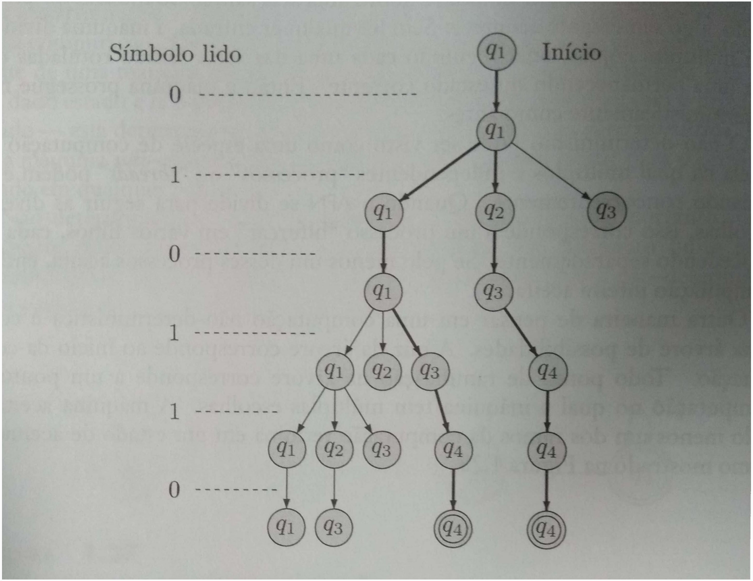
Não-Determinismo iv

- Se um estado com um símbolo ϵ em uma das suas transições for encontrado, algo semelhante acontece. Sem ler qualquer entrada, a máquina divide-se em múltiplas cópias, uma seguindo cada uma das setas saindo rotuladas com ϵ e uma permanecendo no estado corrente.
- Então a máquina prossegue não-deterministicamente como antes.
- O não-determinismo pode ser visto como uma espécie de computação paralela, na qual múltiplos e independentes `processos` ou `threads` podem estar rodando concorrentemente.
- Quando um `AFND` se divide para seguir diversas escolhas, isso corresponde a um processo executar um *fork* (bifurcar) em vários filhos, cada um procedendo separadamente. Da mesma maneira se pelo menos uma dessas instâncias aceita, então a computação inteira aceita.

Não-Determinismo v

- Se pensarmos a computação **não-determinística** como uma árvore de possibilidades. A raiz corresponde ao início da execução da máquina. Todo ponto de ramificação na árvore corresponde a um ponto na computação no qual a máquina tem múltiplas escolhas. A máquina aceita se pelo menos um dos ramos de computação termina em um estado de aceitação.
- A computação do nosso exemplo sobre a entrada `010110` é ilustrada na figura a seguir.

Não-Determinismo vi



Não-Determinismo vii

- Um AFND são úteis em diversos sentidos.
- Podem ser convertidos em um AFD equivalente. Construir um AFND é às vezes mais fácil que construir um AFD diretamente.
- Podem ser muito menores que AFDs e seu funcionamento pode ser mais fácil de entender.

Autômatos Finitos Não Determinísticos

Autômatos Finitos Não Determinísticos i

Definição

Um **Autômato Finito Não Determinístico** (AFND) é uma quíntupla $A = (Q, \Sigma, \delta, Q_0, F)$ onde:

1. Q é um conjunto finito de estados,
2. Σ é o alfabeto de entrada, finito,
3. δ é a função de transição de estados, tal que $\delta : Q \times \Sigma_\epsilon \rightarrow \mathcal{P}(Q)$
    - $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$,
    - $\mathcal{P}(Q) = 2^Q$: Conjunto das partes de Q, todos os subconjuntos de Q.
4. Q_0 é o **conjunto de estados iniciais**, tal que $Q_0 \subseteq Q$,
5. F é o conjunto de estados finais, tal que $F \subseteq Q$.

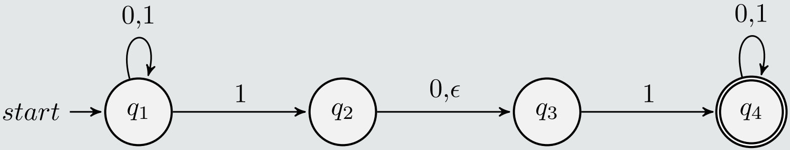
Autômatos Finitos Não Determinísticos ii

Diferenças com relação a AFD

- Agora é um conjunto de estados iniciais.
- Função de transição de estados, para um mesmo símbolo, pode resultar em vários estados.
- Função de transição de estados pode ser para ϵ.
- O símbolo λ também pode ser utilizado para representar símbolo vazio.

Retornemos ao Exemplo

N_1 :



- A descrição formal de N_1 é $(Q, \Sigma, \delta, q_1, F)$, onde:
  1. $Q = \{q_1, q_2, q_3, q_4\}$
  2. $\Sigma = \{0, 1\}$
  3. δ é dado como:

Autômatos Finitos Não Determinísticos iv

	0	1	ϵ
q_1	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
q_2	$\{q_3\}$	$\emptyset$	$\{q_3\}$
q_3	$\emptyset$	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$	$\emptyset$

4. q_1 é o estado inicial, e
5. $F = \{q_4\}$

AFND: Função de transição estendida

Função de transição estendida

Dado o AFND $A = (Q, \Sigma, \delta, Q_0, F)$, a Função de Transição Estendida δ é definida como $\delta : Q \times \Sigma^* \to Q$, tal que:

```
* $\forall s \in Q, \delta(s, \epsilon) = s$
\delta(s, a) = \delta(s, a)$
    * $(\forall s \in Q) \land (\forall a \in \Sigma) \land (
        $\delta(s, ax) = \bigcup_{p \in \delta(s, a)} \delta(
```

AFND: Decisão sobre a pertinência de uma palavra

Pertinência

Dado o AFND $A = (Q, \Sigma, \delta, Q_0, F)$ e uma palavra $x \in \Sigma^*$, temos que:

- A aceita x se e somente se (sse): $\bigcup_{p \in Q_0} \delta(p, x) \cap F \neq \emptyset$
- A rejeita x se e somente se (sse): $\bigcup_{p \in Q_0} \delta(p, x) \cap F = \emptyset$

AFND: Linguagem definida por um AFND

Linguagem de AFND

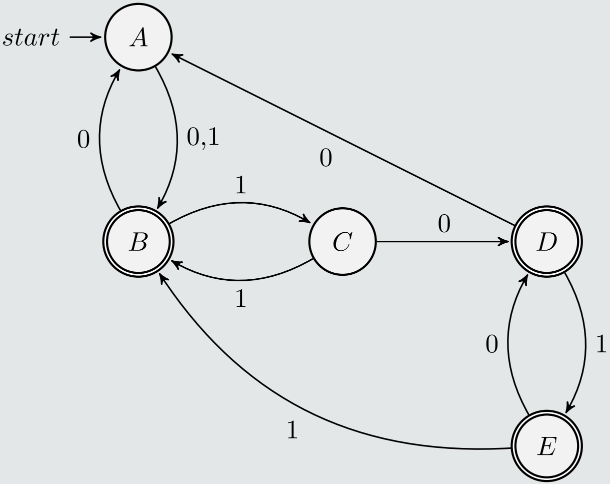
Dado o AFND $A = (Q, \Sigma, \delta, Q_0, F)$, define-se a linguagem $L(A)$, ou seja, a linguagem reconhecida pelo autômato A, como:

$$L(A) = \{x \in \Sigma^* \mid \bigcup_{p \in Q_0} \delta(p, x) \cap F \neq \emptyset\}$$

AFD - Exemplo

Exemplo

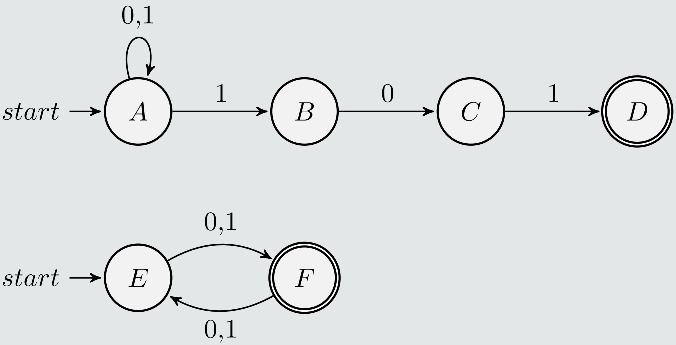
$L = \{xa \in \{0,1\}^* \mid a = 101 \vee |xa| \text{ mod } 2 = 1\}$



AFND - Exemplo

Exemplo

$L = \{xa \in \{0,1\}^* \mid a = 101 \vee |xa| \text{ mod } 2 = 1\}$

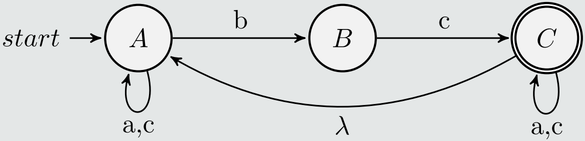


Transições ?

Transição que ocorre sem que nenhum símbolo de entrada seja processado (ou, melhor, é processado o símbolo λ).

Exemplo

$L = \{(x^n y^m z^o)^p \in \{a, b, c\}^* \mid x, z \in \{a, c\} \wedge y = bc\} \wedge p > 0$

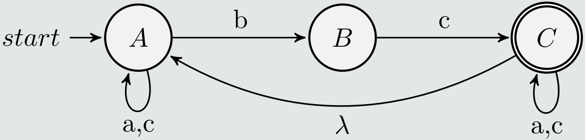


AFND - Closure

Podem ser alcançados a partir de transições λ.

Exemplo

$L = \{(x^n y^m z^o)^p \in \{a,b,c\}^* \mid x,z \in \{a,c\} \wedge y = bc\} \wedge p > 0\}$



- $CL(A) = \{A\}$
- $CL(B) = \{B\}$
- $CL(C) = \{C, A\}$

AFND - Closure: Função de transição estendida

Função de Transição Estendida

Dado o AFND $A = (Q, \Sigma, \delta, Q_0, F)$, a Função de Transição Estendida δ é definida como $\delta : Q \times \Sigma^* \to Q$, tal que:

- $\forall s \in Q, \delta(s, \epsilon) = CL(s)$
- $(\forall s \in Q) \wedge (\forall a \in \Sigma)$: $\delta(s, a) = \bigcup_{p \in CL(s)} \delta(p, a)$
- $(\forall s \in Q) \wedge (\forall a \in \Sigma) \wedge (\forall x \in \Sigma^*)$:
  $\delta(s, ax) = \bigcup_{p \in \delta(s,a)} \delta(p, x)$

Exercício

AFND: Construa um AFND para cada uma das seguintes linguagens

Exercício

- $\{xa \in \{0,1\}^* \mid a = 00\}$ (usar apenas três estados)
    - Tradução: Palavras no alfabeto $\{0,1\}^*$ que terminam com 00.
- $0^*1^*0^+$ (usar apenas três estados)
- $\{x \in \{0,1\}^* \mid$ 1011 ou 111 são subpalavras de x$\}$

Próximas Aulas

Próximas Aulas

- Linguagens Regulares: Autômatos Finitos Não Determinísticos

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


nodes Change
mid
epsilon
thick fill
every state
de
para
gray sets
B
image img
da
e
bend left
appears
que
Q Sigma
em
minimum distance
stealth
shorten
cada
script
q_3
o
edge loop
símbolo
figura
pdf documentclass
state accepting
estado
de transição
shorten 1pt
fill gray
x
c c
column
ou
img pdf
arrow heads
run cmd
lambda
arrow begin
Q_0
node distance
bold node
matrix tikzset
máquina
computação
de entrada
usetikzlibrary automata
tikzpicture stealth'
p
automata positioning
path
distance 7cm
node initial
D
stealth makes
bigcup_ p
delta
document begin
emptyset
distance 3cm
start sets
block
q2
pdflatex jobname
é
end document
Autômatos Finitos
tikz usetikzlibrary
F
1pt auto
edge node
text start
q_4
positioning arrows
node state
podem
Dado
C
como
documentclass tikz
AFD
start arrow
usepackage tikz
tikz standalone
right
q_1
necessary every
https
edges directed
end tikzpicture
AFND
q1
cmd pdflatex
properties
uma
transições
two nodes
delta s
directed stealth
Exemplo
heads bold
style thick
q3
conjunto
initial text
Q
state style
se
edge bend
um
transição estendida
begin document
Função de
usetikzlibrary arrows
tikzpicture end
Sigma delta
standalone usepackage
begin tikzpicture
left node
tikzset makes
auto node

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.

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.