CS310 Automata Theory – 2016-2017

Nutan Limaye

Indian Institute of Technology, Bombay
nutan@cse.iitb.ac.in

Lecture 12: Finite state automata
January 30, 2017

Last class

Myhill-Nerode relations.

Applications of Myhill-Nerode theorem to prove non-regularity.

Decision problems on DFA/NFAs.

The minimization problem for DFAs.

Decision problems on regular languages

Acceptance problem (for fixed Σ)

Given: DFA A, input string $w \in \Sigma^*$

Output: "yes" iff A accepts w.

Construct a graph from an automaton:

> Let $Q = \{q_0, \ldots, q_{m-1}\}$, q_0 be the start state, $F \subseteq Q$ be the set of final states.
>
> Create a layered graph $G_{A,n}$, where $|w| = n$, as follows:
>
> Make $n+1$ copies of Q: $Q_0, Q_1, \ldots, Q_n$, where $Q_i = \{q_{i,0}, \ldots, q_{i,m-1}\}$.
>
> Add edge $(q_{i,u}, q_{i+1,v})$ with label $a \in \Sigma$ if $\delta(q_u, a) = q_v$.

Lemma

There is a path from $q_{0,0}$ to $q_{n,u}$ labelled by a string w in $G_{A,|w|}$ if and only if $\tilde{\delta}(q_0, w) = q_u$ in A.

Decision problems on regular languages

Nonemptiness problem (for fixed Σ)

Given: DFA A

Output: "yes" iff $\exists w : A$ accepts w.

Lemma

If a DFA $A = (Q, \Sigma, \delta, q_0, F)$ accepts some string then it accepts a string of length $\leq |Q|$.

Minimization problem

Minimization problem (for fixed Σ)

Given:	DFA A
Output:	DFA B s.t. $L(A) = L(B)$ and B has the smallest number of states possible for recognizing $L(A)$

Definition

Let $A = (Q, \Sigma, \delta, q_0, F)$. We call states p, q indistinguishable if $\forall w \in \Sigma^*$, $\tilde{\delta}(p, w) \Leftrightarrow \tilde{\delta}(q, w)$.

Minimization algorithm.

Identify indistinguishable states.

Collapse them.

Minimization problem

Minimization problem (for fixed Σ)

Given:	DFA A
Output:	DFA B s.t. $L(A) = L(B)$ and B has the smallest number of states possible for recognizing $L(A)$

Example

	0	1	2	3	4	5
a	1	2	3	4	5	0

(Red color indicates final states.)

0					
–	1				
–	–	2			
–	–	–	3		
–	–	–	–	4	
–	–	–	–	–	5

Minimization problem

Minimization problem (for fixed Σ)

Given: DFA A

Output: DFA B s.t. $L(A) = L(B)$ and B has the smallest number of states possible for recognizing $L(A)$

Example

	0	1	2	3	4	5
a	1	2	3	4	5	0

(Red color indicates final states.)

0					
–	1				
–	–	2			
–	–	–	3		
–	–	–	–	4	
–	–	–	–	–	5

0					
✓	1				
–	✓	2			
–	✓	–	3		
✓	–	✓	✓	4	
–	✓	–	–	✓	5

Minimization problem

Minimization problem (for fixed Σ)

Given:	DFA A
Output:	DFA B s.t. $L(A) = L(B)$ and B has the smallest number of states possible for recognizing $L(A)$

Example

	0	1	2	3	4	5
a	1	2	3	4	5	0

(Red color indicates final states.)

0					
–	1				
–	–	2			
–	–	–	3		
–	–	–	–	4	
–	–	–	–	–	5

0					
✓	1				
✓	✓	2			
–	✓	–	3		
✓	–	✓	✓	4	
–	✓	–	–	✓	5

Minimization problem

Minimization problem (for fixed Σ)

Given:	DFA A
Output:	DFA B s.t. $L(A) = L(B)$ and B has the smallest number of states possible for recognizing $L(A)$

Example

	0	1	2	3	4	5
a	1	2	3	4	5	0

(Red color indicates final states.)

0					
–	1				
–	–	2			
–	–	–	3		
–	–	–	–	4	
–	–	–	–	–	5

0					
✓	1				
✓	✓	2			
–	✓	–	3		
✓	–	✓	✓	4	
✓	✓	–	–	✓	5

Minimization problem

Minimization problem (for fixed Σ)

Given: DFA A

Output: DFA B s.t. $L(A) = L(B)$ and B has the smallest number of states possible for recognizing $L(A)$

Example

	0	1	2	3	4	5
a	1	2	3	4	5	0

(Red color indicates final states.)

0					
–	1				
–	–	2			
–	–	–	3		
–	–	–	–	4	
–	–	–	–	–	5

0					
✓	1				
✓	✓	2			
–	✓	✓	3		
✓	–	✓	✓	4	
✓	✓	–	–	✓	5

Minimization problem

Minimization problem (for fixed Σ)

Given: DFA A

Output: DFA B s.t. $L(A) = L(B)$ and B has the smallest number of states possible for recognizing $L(A)$

Example

	0	1	2	3	4	5
a	1	2	3	4	5	0

(Red color indicates final states.)

0					
–	1				
–	–	2			
–	–	–	3		
–	–	–	–	4	
–	–	–	–	–	5

0					
✓	1				
✓	✓	2			
–	✓	✓	3		
✓	–	✓	✓	4	
✓	✓	–	✓	✓	5

Minimization problem

Minimization problem (for fixed Σ)

Given:	DFA A
Output:	DFA B s.t. $L(A) = L(B)$ and B has the smallest number of states possible for recognizing $L(A)$

Example

	0	1	2	3	4	5
a	1	3	4	5	5	5
b	2	4	3	5	5	5

(Red color indicates final states.)

0					
–	1				
–	–	2			
–	–	–	3		
–	–	–	–	4	
–	–	–	–	–	5

DIY!

Minimization problem

Minimization problem (for fixed Σ)

Given: DFA A

Output: DFA B s.t. $L(A) = L(B)$ and B has the smallest number of states possible for recognizing $L(A)$

Algorithm

Let $Q = \{q_1, \ldots, q_n\}$.

1. For each $1 \le i < j \le n$, initialize $T(i,j) = --$
2. For each $1 \le i < j \le n$

   If ($q_i \in F$ AND $q_j \notin F$) OR ($q_i \in F$ AND $q_j \notin F$)

   $T(i,j) \leftarrow \checkmark$
3. Repeat

   { For each $1 \le i < j \le n$

   If $\exists a \in \Sigma$, $T(\delta(q_i, a), \delta(q_j, a)) = \checkmark$

   then $T(i,j) \leftarrow \checkmark$

   }

Untill T stays unchanged.

Recap of Module - I

DFA, NFA, Regular expressions and their equivalence.

Closure properties of regular languages.

Non-regular languages and Pigeon Hole Principle.

Pumping lemma and its applications.

Myhill Nerode relation and characterization of regular languages.

Polynomial time algorithms for membership problem, emptiness problem and minimization problem.

Module - II: Different models of computation

What do we plan to do in this module?

2DFA, a variant of a DFA where the input head moves right/left.

Chapter 18, from the text of Dexter Kozen

Pushdown automata, context-free languages(CFLs), context-free grammar(CFG), closure properties of CFLs.

Module - II: Different models of computation

2DFA: Two-way deterministic finite state automata.

$\# \; w_1 \; w_2 \; \ldots \; \ldots \; \ldots \; \ldots \; \ldots \; \ldots \; w_n \; \$$

Input head moves left/right on this tape.

It does not go to the left of $\#$.

It does not go to the right of \$.

Can potentially get stuck in an infinite loop!

Formal definition of 2DFA

Definition

A 2DFA $A = (Q, \Sigma \cup \{\#, \$\}, \delta, q_0, q_{acc}, q_{rej})$, where

Q :	set of states,	Σ:	input alphabet
$\#$:	left endmarker	$\$$:	right endmarker
q_0:	start state		
q_{acc}:	accept state	q_{rej}:	reject state

$\delta : Q \times (\Sigma \cup \{\#, \$\} \to Q \times \{L, R\}$

The following conditions are forced:

$\forall q \in Q$, $\exists q', q'' \in Q$ s.t. $\delta(q, \#) = (q', R)$ and $\delta(q, \$) = (q'', L)$.