Lecture 4 – Appendix

NTIN071 Automata and Grammars

Jakub Bulín (KTIML MFF UK)

Spring 2026

** Adapted from the Czech-lecture slides by Marta Vomlelová with gratitude. The translation, some modifications, and all errors are mine.*

Appendix A: Visit every state

Visit every state (application of closure properties)

Example (visit every state)

Given a DFA A, let L consist of all $w \in \Sigma^*$ that are accepted and, moreover, during the computation every state is visited, i.e.:

- $\delta^*(q_0, w) \in F$
- for every $q \in Q$ there is a prefix x_q of w s.t. $\delta^*(q_0, x_q) = q$

We will show that this language is regular.

Construct L from $M = L(A)$ using operations preserving regularity:

- define an alphabet of 'transitions': $T = \{[paq] \mid \delta(p, a) = q\}$
- define a homomorphism $h([paq]) = a$ for all p, q, a
- $L_1 = h^{-1}(M)$ is regular (inverse homomorphism), consists of accepting sequences of transitions

Visit every state: proof continues

- start at q_0: $L_2 = L_1 \cap L(E_1.T^*)$, $E_1 = \{[q_0 a q] \mid a \in \Sigma, q \in Q\}$
- adjacent states must be equal: define non-matching pairs

  $$E_2 = \{[paq][rbs] \mid q \neq r, p, q, r, s \in Q, a, b \in \Sigma\}$$

  and set $L_3 = L_2 - L(T^*.E_2.T^*)$ (remove if at least one non-matching pair of adjacent states)
- L_3 already ends in accepting state: we started from $M = L(A)$
- all states: for $q \in Q$ let E_q be the RE that is the sum of all the symbols in T not containing q, set $L_4 = L_3 - \bigcup_{q \in Q}\{L(E_q^*)\}$
- from a sequence of transitions back to the word: $L = h(L_4)$

$\square$

Appendix B: Automata with output (Moore and Mealy machines)

Automata with output

Definition (Moore machine)

Moore machine is a six-tuple $A = (Q, \Sigma, Y, \delta, \mu, q_0)$ where

Q non–empty set of states

Σ finite nonempty set of symbols (input alphabet)

Y finite nonempty set of symbols (output alphabet)

δ a mapping $Q \times \Sigma \to Q$ (transition function)

μ a mapping $Q \to Y$ (output function)

$q_0 \in Q$ (initial state)

- the output function may imitate final states
  - $F \subseteq Q$ may be replaced by output function $\mu : Q \to \{0, 1\}$ as follows:

$$\mu(q) = 0 \text{ if } q \notin F,$$
$$\mu(q) = 1 \text{ if } q \in F.$$

Moore Machine Example

Example (Tennis Game Score)

A machine calculates the tenis score.

- Input alphabet: ID of the player who scored a point
- Output alphabet & states: the score ($Q = Y$ and $\mu(q) = q$)

State/output	A	B
00:00	15:00	00:15
15:00	30:00	15:15
15:15	30:15	15:30
00:15	15:15	00:30
30:00	40:00	30:15
30:15	40:15	30:30
30:30	40:30	30:40
15:30	30:30	15:40
00:30	15:30	00:40
40:00	A	40:15
40:15	A	40:30
40:30	A	deuce
30:40	deuce	B
15:40	30:40	B
00:40	15:00	B
deuce	A:40	40:B
A:40	A	deuce
40:B	deuce	B

Mealy machine

Definition (Mealy machine)

Mealy machine is a six–tuple $A = (Q, \Sigma, Y, \delta, \lambda_M, q_0)$ where

Q non–empty set of states

Σ finite nonempty set of symbols (input alphabet)

Y finite nonempty set of symbols (output alphabet)

δ a mapping $Q \times \Sigma \rightarrow Q$ (transition function)

λ_M a mapping $Q \times \Sigma \rightarrow Y$ (output function)

$q_0 \in Q$ (initial state)

- The output is determined by a state and the input symbol
  - Mealy machine is more general then Moore
  - The output function may be replaced as follows

$$\forall x \in \Sigma \ \lambda_M(q, x) = \mu(q)$$

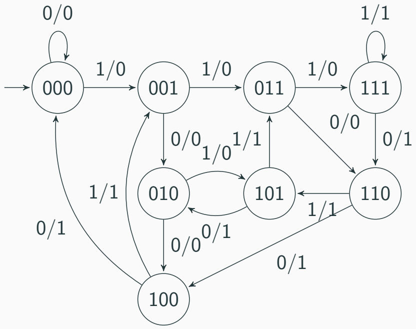
or $$\forall x \in \Sigma \ \lambda_M(q, x) = \mu(\delta(q, x))$$

Mealy Machine Example

The automaton for integer division of the input by 8 (the remainder is discarded).

- Three bit move to the left
- we need to remember last three bits
- three–bit dynamic memory.

State\symbol	0	1
→000	000/0	001/0
001	010/0	011/0
010	100/0	101/0
011	110/0	111/0
100	000/1	001/1
101	010/1	011/1
110	100/1	101/1
111	110/1	111/1



Extended Output Function

for any word in the input alphabet $\Sigma^* \rightarrow$ we get a word in the output alphabet Y^*

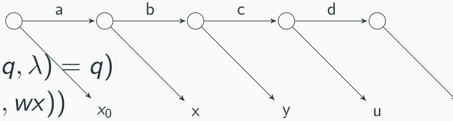
Moore machine

output function $\mu : Q \rightarrow Y$

$\mu^* : Q \times \Sigma^* \rightarrow Y^*$

$\mu^*(q, \lambda) = \lambda$ (sometimes $\mu^*(q, \lambda) = q$)

$\mu^*(q, wx) = \mu^*(q, w).\mu(\delta^*(q, wx))$



Example: μ^*(00:00,AABA)=(00:00 .) 15:00 . 30:00 . 30:15 . 40:15

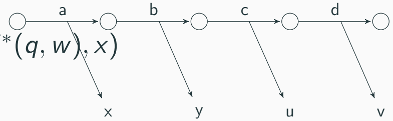
Mealy machine

output function $\lambda_M : Q \times \Sigma \rightarrow Y$

$\lambda_M^* : Q \times \Sigma^* \rightarrow Y^*$

$\lambda_M^*(q, \lambda) = \lambda$

$\lambda_M^*(q, wx) = \lambda_M^*(q, w).\lambda_M(\delta^*(q, w), x)$



Example: λ_M^*(000,1101010)=0001101

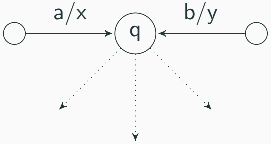
Moore and Mealy Machines Reductions

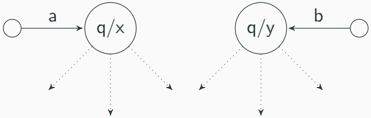
Lemma

For any Moore machine there exists a Mealy machine, and vice versa, mapping each input word to the same output word.

$\Rightarrow$ Mealy machine $B = (Q, \Sigma, Y, \delta, \lambda_M, q_0)$ where $\lambda_M(q, x) = \mu(\delta(q, x))$

$\Leftarrow$ We define states of the Moore machine $Q \times Y$, $\delta^{|}([q, y], x) = [\delta(q, x), \lambda(q, x)]), \mu([q, y]) = y$.





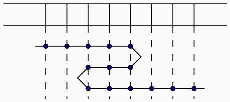
Appendix C: A constructive conversion from two-way DFA to NFA

Constructive proof of two-way automata regularity: two-way DFA to NFA

Transcribe left-right movement to a linear computation.

We are interested in accepting computations only.

We focus on transitions in cuts between input symbols.



The direction of movement repeats (right, left)

The first and the last transitions are to the right

A is deterministic, accepting computations have no cycles.

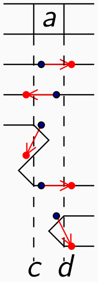
The first and the last cut contain only one state.

- Find all possible cuts = state sequences (finite number).
- Define non-deterministic transition between cuts according to input symbol.
- We re–construct the computation by composing cuts like a puzzle.

Formal reduction two-way DFA to NFA

Let $A = (Q, \Sigma, \delta, q_0, F)$be a two–way DFA. We define an equivalent NFA $B = (Q^|, \Sigma, \delta^|, (q_0), F^|)$ as follows:
$Q^|$ all possible correct transition sequences

- sequences of states $(q^1, \ldots, q^k)$; $q^i \in Q$
- with an odd length $(k = 2m + 1)$
- no state repeats at odd nor at even position $(\forall i \neq j)\ (q^{2i} \neq q^{2j}) \& (\forall i \neq j)\ (q^{2i+1} \neq q^{2j+1})$



$F^| = \{(q) | q \in F\}$ sequences of the length 1
$\delta^|(c, a) = \{d | d \in Q^| \& c \overset{a}{\to} d$ is a locally consistent transition for $a\}$

- $\exists$ bijection: $h : c_{odd} \cup d_{even} \to c_{even} \cup d_{odd}$ so that:
- for $h(q) \in c_{even}$ is $(h(q), -1) = \delta(q, a)$
- for $h(q) \in d_{odd}$ is $(h(q), +1) = \delta(q, a)$

Trajectory in A corresponds to cuts in $B \Rightarrow L(A) = L(B)$

Example Reduction Two-way DFA to NFA

	a	b
$\rightarrow p$	p,+1	q,+1
$*q$	q,+1	r,-1
r	p,+1	r,-1

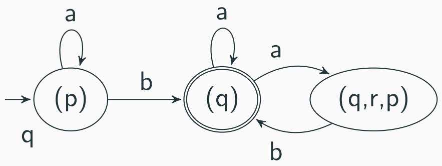
Possible cuts and their transitions

- leftwards only r – all even positions r, that means only one even position
- possible cuts: $(p), (q), (p, r, q), (q, r, p)$.

	a	b
$\rightarrow (p)$	(p)	(q)
$*(q)$	(q),(q,r,p)	
(p,r,q)		
(q,r,p)		(q)

Non-accepting example:

a	a	b	a	a	b	a	a	b	b
p	p	p	q	q	q				
					r				
					p	q	q	q	
								r	
								p	q


a
a
a
b
b
(p)
(q)
(q,r,p)