



**OPPA European Social Fund
Prague & EU: We invest in your future.**

ePAL - Approximate Text Searching

Radek Mařík
Marko Genyk-Berezovskyj

ČVUT FEL, K13133

November 28, 2012



Outline

- 1 Basic Automata
- 2 Non-deterministic Finite Automaton
- 3 Regular Expressions
- 4 Languages
- 5 Hamming distance
- 6 Levenshtein distance
- 7 Dictionary Automata
- 8 Binary Implementation of Searching Automata

Outline

- 1 Basic Automata
- 2 Non-deterministic Finite Automaton
- 3 Regular Expressions
- 4 Languages
- 5 Hamming distance
- 6 Levenshtein distance
- 7 Dictionary Automata
- 8 Binary Implementation of Searching Automata

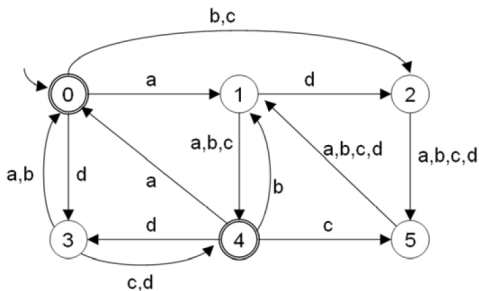
Example 1

Automaton A_1 is given by its transition table. Draw its transition diagram.

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

Example 2

Automaton A_2 is given by its transition diagram. Draw its transition table.



Example 3

Make a decision if automaton A_1 accepts the following words

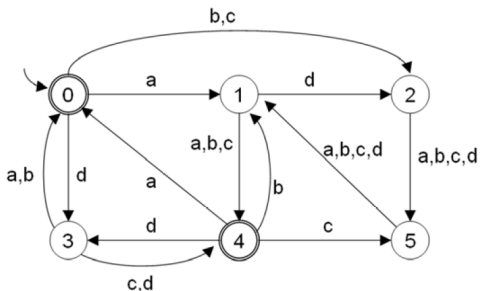
- 1 *addca*
- 2 *bbcca*
- 3 *bbccaba*

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

Example 4

Make a decision if automaton A_2 accepts the following words

- 1 *addca*
- 2 *bbcca*
- 3 *bbccaba*



Example 5

Draw a state diagram of an automaton that accepts just all words over alphabet $\{0, 1\}$ which

- 1 contain subsequence 01,
- 2 do not contain subsequence 01,
- 3 contain a single character 1 and an arbitrary number of characters 0,
- 4 begin and end with symbol 1,
- 5 represent binary representations of numbers 0, 1, 2, 3, 4, 5, 6, 7 in their all 1-, 2- 3- digits sequences.



Outline

- 1 Basic Automata
- 2 Non-deterministic Finite Automaton**
- 3 Regular Expressions
- 4 Languages
- 5 Hamming distance
- 6 Levenshtein distance
- 7 Dictionary Automata
- 8 Binary Implementation of Searching Automata

Example 6

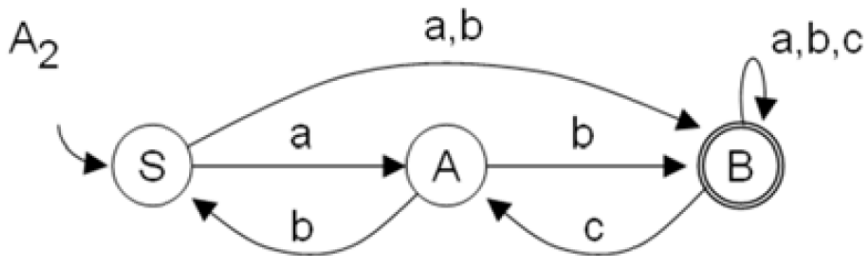
Automaton A_1 is given by its transition table. Determine its equivalent deterministic automaton

A_1

	a	b	c	d	
0	0, 1		2	2	F
1		0, 2			
2	1		1, 2	0, 2	

Example 7

Automaton A_2 is given by its transition table. Determine its equivalent deterministic automaton



Example 8

Create an NFA over alphabet $\{a, b, c\}$ that accepts all words both beginning and ending with chain

- 1 *abc*,
- 2 *acaca*,

Example 9

Create an NFA over alphabet $\{a, b, c\}$ that accepts all words not containing chain

- 1 *abc*,
- 2 *acaca*,

Example 10

Create an NFA and its related DFA that searches for a given exact pattern

- 1 *aaba*,
- 2 *abaa*,

Example 11

Create an NFA and its related DFA that searches for all subchains of a given pattern

- 1 *abcab*,
- 2 *accbc*,

Example 12

Create an NFA and its related DFA that searches for all chains having Hamming distance at most 2 from a given pattern:

- 1 *bbaabb*,
- 2 *acacab*,

Outline

- 1 Basic Automata
- 2 Non-deterministic Finite Automaton
- 3 Regular Expressions**
- 4 Languages
- 5 Hamming distance
- 6 Levenshtein distance
- 7 Dictionary Automata
- 8 Binary Implementation of Searching Automata

Example 13

Write all words of length at most 5 of a language described by the following regular expression over alphabet $\{0, 1\}$

- 1 $(01|0) \star 0$
- 2 $0(10|0) \star$

Example 14

Write a regular expression describing a language over alphabet $\{0, 1\}$ such that

- 1 each word contains only zeros,
- 2 each word contains just one 1,
- 3 each word contains at least one 1,
- 4 each word contains at least two 1,
- 5 each word contains an even number of 1,
- 6 each word contains an odd number of 1,



Example 15

Write a regular expression describing a maximum set M of words over alphabet $\{a, b, c\}$ such that

- 1 each word in M starts and ends with symbol b ,
- 2 each word in M contains just one occurrence of symbol c anywhere in the word,
- 3 no word in M contains symbol a on an odd position (positions are indexed from 1).

Example 16

Create an automaton that searches for words described by regular expression R over alphabet A .

- $A = \{a, b, c\}$
- $R = c * (ac + bb)^*$

Example 17

Create an automaton that searches for words described by regular expression R over alphabet A .

- $A = \{0, 1\}$
- $R = 0 * (101 + 11) * 0$

Outline

- 1 Basic Automata
- 2 Non-deterministic Finite Automaton
- 3 Regular Expressions
- 4 Languages**
- 5 Hamming distance
- 6 Levenshtein distance
- 7 Dictionary Automata
- 8 Binary Implementation of Searching Automata

Example 18

We are given two languages L_1 and L_2 over alphabet $\{0, 1\}$. Words of L_1 are described by expression $0 * 1 * 0 * 1 * 0^*$, words of L_2 are described by expression $(01|10)^*$.

- 1 Find the shortest non-empty word of intersection $L_1 \cap L_2$,
- 2 Find the longest word of intersection $L_1 \cap L_2$,
- 3 Find the shortest non-empty word that belongs to L_1 but it is not contained in L_2 ,
- 4 Find the shortest non-empty word that belongs to L_2 but it is not contained in L_1 ,
- 5 Find the shortest non-empty word of union $L_1 \cup L_2$.



Outline

- 1 Basic Automata
- 2 Non-deterministic Finite Automaton
- 3 Regular Expressions
- 4 Languages
- 5 Hamming distance**
- 6 Levenshtein distance
- 7 Dictionary Automata
- 8 Binary Implementation of Searching Automata

Example 19

Find all word occurrences in text T having Hamming distance at most k from pattern P . Use the method of dynamic programming.

- $T = aacacacbaabbbcbccacc$
- $P = abbcba$
- $k = 2$

Example 20

Find all word occurrences in text T having Hamming distance at most k from pattern P . Use the method of dynamic programming.

- $T = 000111011000101010111110$
- $P = 110010$
- $k = 3$

Outline

- 1 Basic Automata
- 2 Non-deterministic Finite Automaton
- 3 Regular Expressions
- 4 Languages
- 5 Hamming distance
- 6 Levenshtein distance**
- 7 Dictionary Automata
- 8 Binary Implementation of Searching Automata

Example 21

Find all words over alphabet $\{a, b, c\}$ having Levenshtein distance k from pattern $P = aba$.

- 1
- 2

Example 23

Find all word occurrences in text T having Levenshtein distance at most k from pattern P .

- $T = \text{aacacacbaabbbcbbcacc}$
- $P = \text{abbcba}$
- $k = 2$

Example 24

Find all word occurrences in text T having Levenshtein distance at most k from pattern P .

- $T = 010011101000010101011100$
- $P = 11100$
- $k = 2$



Outline

- 1 Basic Automata
- 2 Non-deterministic Finite Automaton
- 3 Regular Expressions
- 4 Languages
- 5 Hamming distance
- 6 Levenshtein distance
- 7 Dictionary Automata**
- 8 Binary Implementation of Searching Automata

Example 25

Create a DFA over alphabet A that accepts just words from set M over this alphabet.

- $A = \{a, b, c\}$
- $M = \{a, b, ba, bc, aaa, bab, ccc, abbc, abcc\}$



Example 26

Create a DFA over alphabet A that accepts just words from set M over this alphabet.

- $A = \{0, 1\}$
- $M = \{10, 11, 101, 111, 1011, 1101, 10001, 10011, 10111, 11101, 11111\}$



Outline

- 1 Basic Automata
- 2 Non-deterministic Finite Automaton
- 3 Regular Expressions
- 4 Languages
- 5 Hamming distance
- 6 Levenshtein distance
- 7 Dictionary Automata
- 8 Binary Implementation of Searching Automata**

Example 27

Create a table of a simulation for a searching automata using the method of bitwise paralelism in text T having Hamming distance k from pattern P .

- $T = abcbcaaccbbaa$
- $P = bbac$
- $k = 2$

Example 28

Create a table of a simulation for a searching automata using the method of bitwise paralelism in text T having Hamming distance k from pattern P .

- $T = accbbaaabcba$
- $P = acbb$
- $k = 2$

References I



**OPPA European Social Fund
Prague & EU: We invest in your future.**
