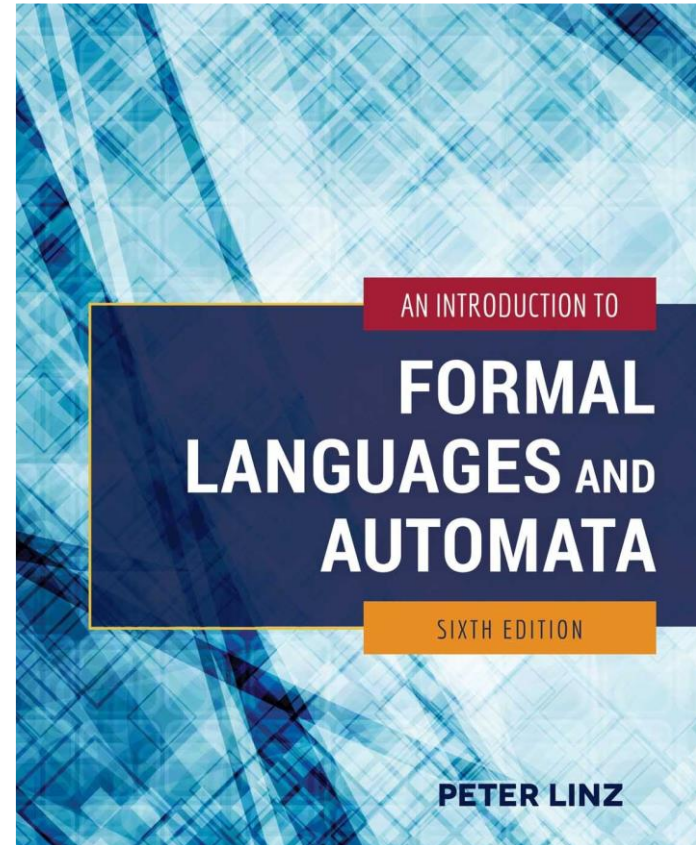
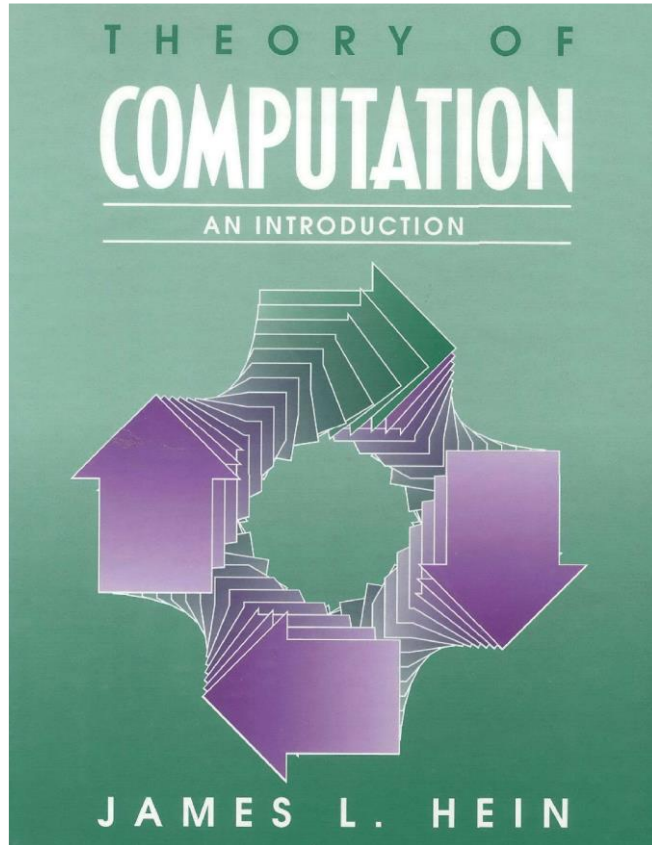


Automata and Formal Languages

Lecture 04

Books



PowerPoint

<http://www.bu.edu.eg/staff/ahmedaboalatah14-courses/14767>

The screenshot shows a web interface for Benha University. At the top, there is a blue header with the university logo, the name 'Benha University', and a welcome message for 'Ahmed Hassan Ahmed Abu El Atta' with a 'Log out' link. Below the header, a navigation menu on the left lists various university services. The main content area displays course details for 'Automata and Formal Languages' by 'Ass. Lect. Ahmed Hassan Ahmed Abu El Atta'. The details are presented in a table with blue headers and white content. A 'Course password' section is also visible. On the right side, there are social media icons and a vertical toolbar with various icons.

Benha University

Staff Search: **Welcome: Ahmed Hassan Ahmed Abu El Atta (Log out)**

You are in: [Home](#) / [Courses](#) / [Automata and Formal Languages](#) [Back To Courses](#)

Ass. Lect. Ahmed Hassan Ahmed Abu El Atta :: Course Details:
Automata And Formal Languages [add course](#) | [edit course](#)

Course name	Automata and Formal Languages
Level	Undergraduate
Last year taught	2018
Course description	Not Uploaded
Course password	
Course files	add files
Course URLs	add URLs
Course assignments	add assignments
Course Exams & Model Answers	add exams

Course password

Course files [add files](#)

Course URLs [add URLs](#)

Course assignments [add assignments](#)

Course Exams & Model Answers [add exams](#)

(edit)

Regular Expressions & Regular Languages

Agenda

- Regular Expressions
- Example
- The Operations Priority
- Languages Associated with Regular Expressions
- Examples
- Properties of Regular Expressions
- RE to NFA
- Examples

Regular Expressions

The set of regular expressions over an alphabet “A” is defined inductively as follows, where + and • are binary operations and * is a unary operation:

Basis:

- Λ , \emptyset , and a are regular expressions for all $a \in A$.

Induction:

- If R and S are regular expressions, then the following expressions are also regular:

(R) , $R + S$, $R.S$, and R^* .

Example 00

$A = \{a, b\}$

Λ

\emptyset

a

b

$\Lambda + b$

b^*

$a + (b.a)$

$(a + b).a$

$a.b^*$

$a^* + b^*$

The Operations Priority

*** highest (do it first),**

•

+ lowest (do it last).

$$a + b.a^* = (a + (b.(a^*)))$$

Languages Associated with Regular Expressions

$$L(\emptyset) = \emptyset,$$

$$L(\wedge) = (\wedge),$$

$$L(a) = (a) \text{ for each } a \in A,$$

$$L(R + S) = L(R) \cup L(S),$$

$$L(R \bullet S) = L(R)L(S) \quad (\text{language product}),$$

$$L(R^*) = L(R)^* \quad (\text{language closure}).$$

Example 01

Let's find the language of the regular expression $a + bc^*$. We can evaluate the expression $L(a + bc^*)$ as follows:

$$\begin{aligned}L(a + bc^*) &= L(a) \cup L(bc^*) \\ &= L(a) \cup (L(b).L(c^*)) \\ &= L(a) \cup (L(b).L(c)^*) \\ &= \{a\} \cup (\{b\} . \{c\}^*) \\ &= \{a\} \cup (\{b\} \{\Lambda, c, c^2, \dots, c^n, \dots\}) \\ &= \{a\} \cup \{b, bc, bc^2, \dots, bc^n, \dots\} \\ &= \{a, b, bc, bc^2, \dots, bc^n, \dots\}\end{aligned}$$

Example 02

language $L(a^* \cdot (a + b))$ in set notation:

$$\begin{aligned}L(a^* \cdot (a + b)) &= L(a^*) L(a + b) \\ &= (L(a))^* (L(a) \cup L(b)) \\ &= \{\Lambda, a, aa, aaa, \dots\} \{a, b\} \\ &= \{a, aa, aaa, \dots, b, ab, aab, \dots\}.\end{aligned}$$

Example 03

language $r = (a + b)^* (a + bb)$ in set notation:

$$\begin{aligned}L((a + b)^* . (a + bb)) &= L((a+b)^*) . L(a + bb) \\ &= (L(a+b))^* . (L(a) \cup L(bb)) \\ &= (\{a, b\})^* . (\{a\} \cup L(b)L(b)) \\ &= \{\Lambda, a, b, aa, ab, ba, bb \dots\} \{a, bb\} \\ &= \{a, bb, aa, abb, ba, bbb, \dots\}\end{aligned}$$

Example 04

Find language?

$\Lambda + b$

b^*

$a + (b.a)$

$(a + b).a$

$(aa)^* (bb)^* b$

Example 05

For $\Sigma = \{0, 1\}$, give a regular expression **R** such that

- $L(R) = \{w \in \Sigma^* : w \text{ has at least one pair of consecutive zeros}\}$.

$$R = (0 + 1)^* 00 (0 + 1)^*$$

*Example 06

Find a regular expression for the language

- $L = \{w \in \{0, 1\}^* : w \text{ has no pair of consecutive zeros}\}.$

$$(1 + 01)^*(\Lambda + 0)$$

*Example 06

$\{\Lambda, 0, 1, 01, 10, 11, 010, 011, 101, 110, 111, 0101, 0110, 1010, 0111, 1110, \dots\}$

$= \{\Lambda, 1, 01, 11, 011, 101, 111, 0101, 0111, \dots\}$

$\cup \{0, 10, 010, 110, 0110, 1010, 1110, \dots\}$

$= \{\Lambda, 1, 01, 11, 011, 101, 111, 0101, 0111, \dots\}$

$\cup \{0, 10, 010, 110, 0110, 1010, 1110, \dots\}$

$= \{\Lambda, 1, 01, 11, 011, 101, 111, 0101, 0111, \dots\}$

$\cup \{\Lambda, 1, 01, 11, 011, 101, 111, \dots\}\{0\}$

$= \{1, 01\}^* \cup \{1, 01\}^* \cdot \{0\}$

$= \{1, 01\}^* \{\Lambda, 0\}$

Example 07

Find a regular expression for

$$L_1 = \{a^n b^m : n \geq 3, m \text{ is odd}\}$$

$$L_2 = \{a^n b^m : (n + m) \text{ is odd}\}$$

$$L_3 = \{a^n b^m, n \geq 3, m \leq 4\}$$

Properties of Regular Expressions

1. (+ properties)

$$R + T = T + R,$$

$$R + \emptyset = \emptyset + R = R,$$

$$R + R = R,$$

$$(R + S) + T = R + (S + T).$$

2. (\cdot properties)

$$R\emptyset = \emptyset R = \emptyset,$$

$$R\Lambda = \Lambda R = R,$$

$$(RS)T = R(ST).$$

3. (Distributive properties)

$$R(S + T) = RS + RT,$$

$$(S + T)R = SR + TR.$$

Properties of Regular Expressions (cont.)

(Closure properties)

4. $\emptyset^* = \Lambda^* = \Lambda$.

5. $R^* = R^*R^* = (R^*)^* = R + R^*$,

$$R^* = \Lambda + R^* = (\Lambda + R)^* = (\Lambda + R)R^* = \Lambda + RR^*,$$

$$R^* = (R + \dots + R^k)^* \quad \text{for any } k \geq 1,$$

$$R^* = \Lambda + R + \dots + R^{k-1} + R^kR^* \quad \text{for any } k \geq 1.$$

6. $R^*R = RR^*$.

7. $(R + S)^* = (R^* + S^*)^* = (R^*S^*)^* = (R^*S)^*R^* = R^*(SR^*)^*$.

8. $R(SR)^* = (RS)^*R$.

9. $(R^*S)^* = \Lambda + (R + S)^*S$,

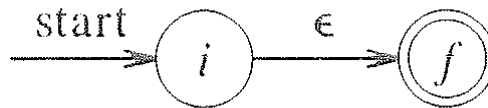
$$(RS^*)^* = \Lambda + R(R + S)^*.$$

RE to NFA

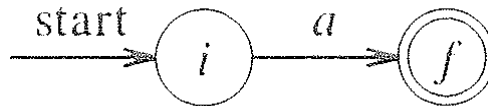
First parse r into its constituent sub expressions.

Construct NFA's for each of the basic symbols in r .

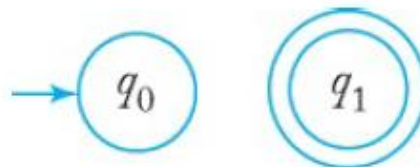
- for ϵ



- for a in Σ

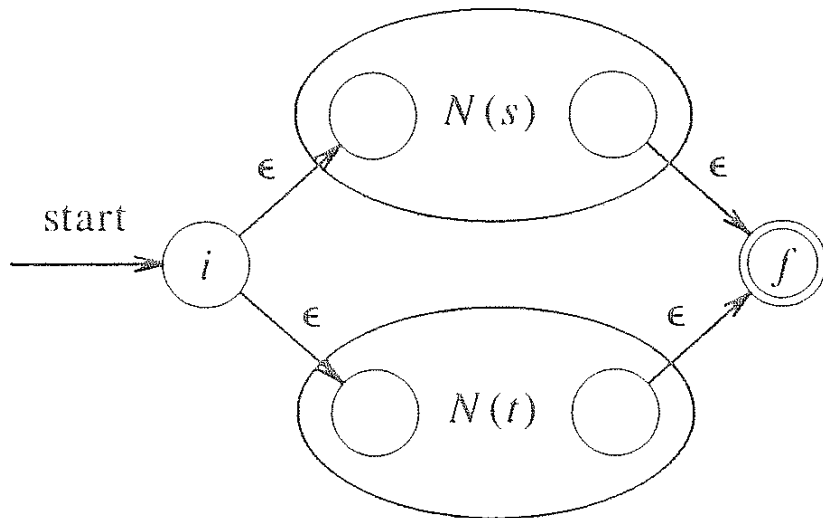


- for \emptyset

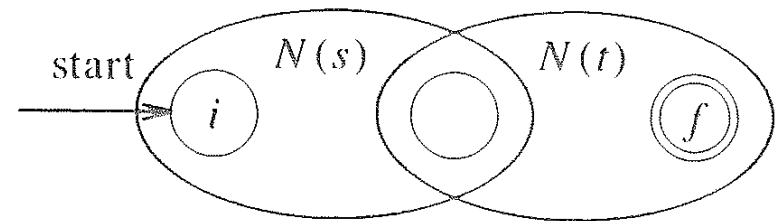


RE to NFA (cont.)

For the regular expression $s+t$,



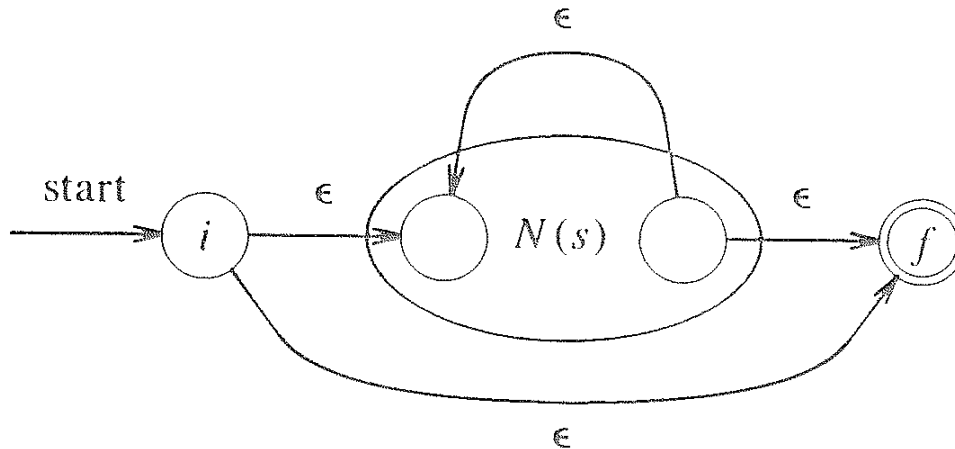
For the regular expression $s.t$,



RE to NFA (cont.)

For the regular expression s^* ,

For the parenthesized regular expression (s) , use $N(s)$ itself as the NFA.

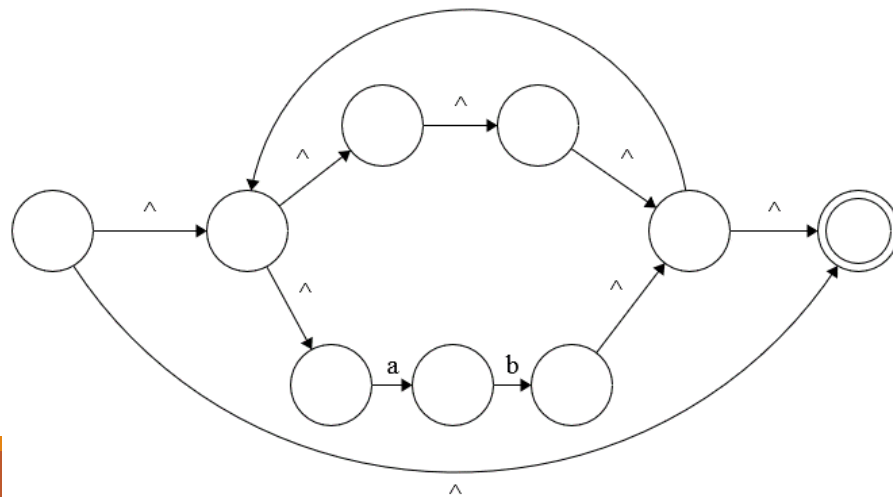
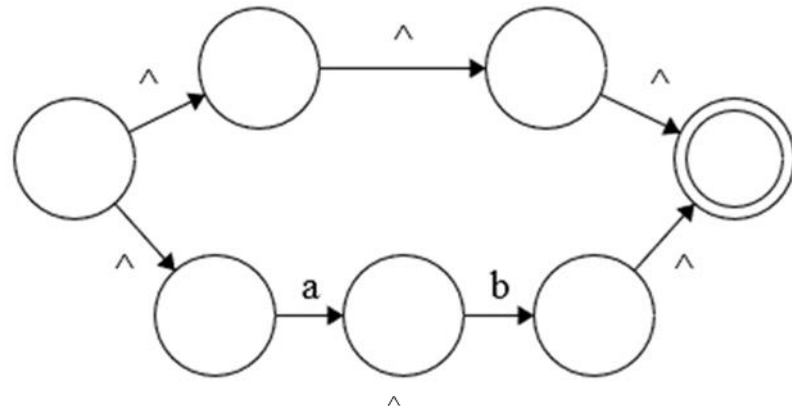
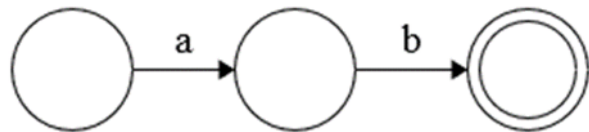
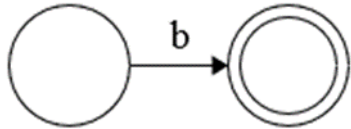
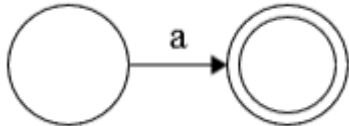
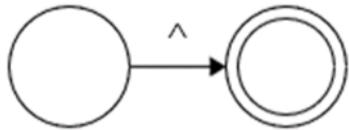


Every time we construct a new state, we give it a distinct name.

Example 08

Find an NFA that accepts $L(r)$, where

$$r = (\Lambda + ab)^*$$



Example 09

Find an NFA that accepts each regular Expression

$a^*a + ab$

$(aab)^*ab$

ab^*aa

