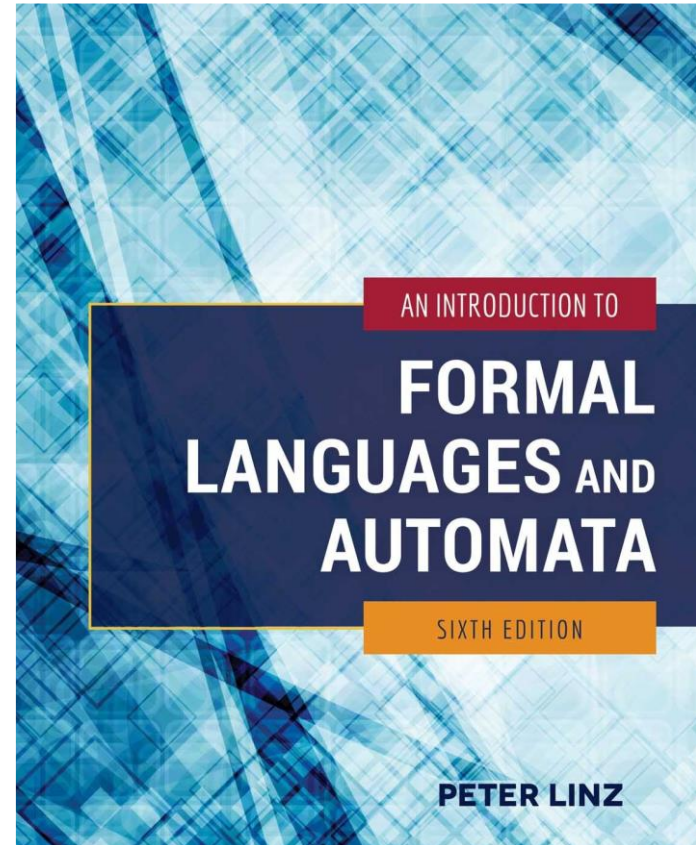
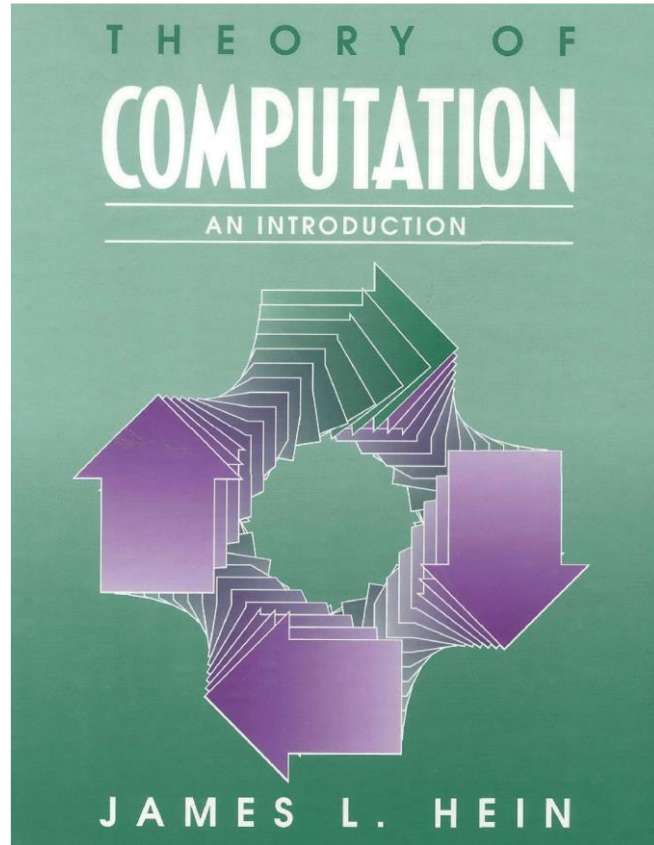


Automata and Formal Languages

Lecture 05

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 icons for Google, a book, RG, LinkedIn, Facebook, Twitter, Google+, YouTube, WordPress, a camera, a globe, a question mark, and an edit icon.

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

(edit)

Regular Expressions & NFA

Agenda

- Regular Expressions
- The Operations Priority
- Languages Associated with Regular Expressions
- RE to NFA
- Examples
- Regular Expression to Finite Automaton
- 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^* .

The Operations Priority

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

•

+ lowest (do it last).

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

Languages Associated with Regular Expressions

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

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

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

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

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

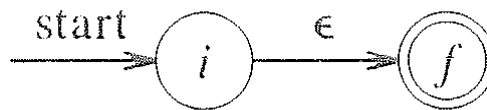
$$L(R^*) = L(R)^* \quad (\text{language closure}).$$

RE to NFA

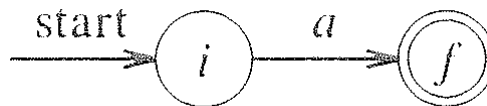
First parse r into its constituent sub expressions.

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

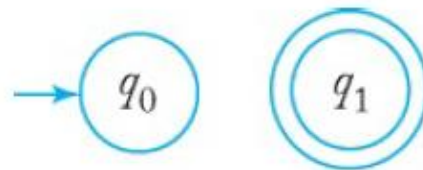
- for ϵ



- for a in Σ

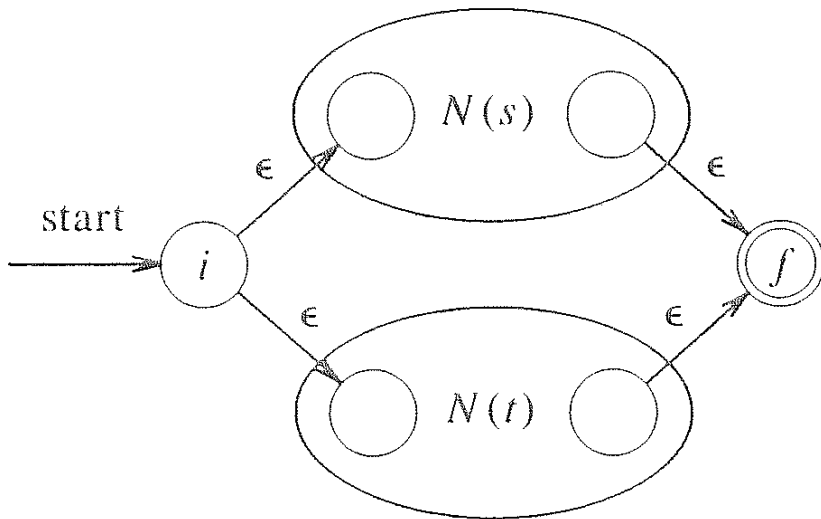


- for \emptyset

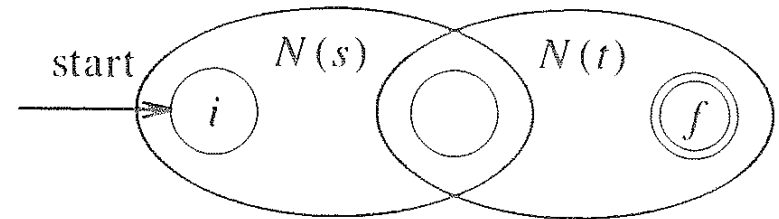


RE to NFA (cont.)

For the regular expression $s+t$,



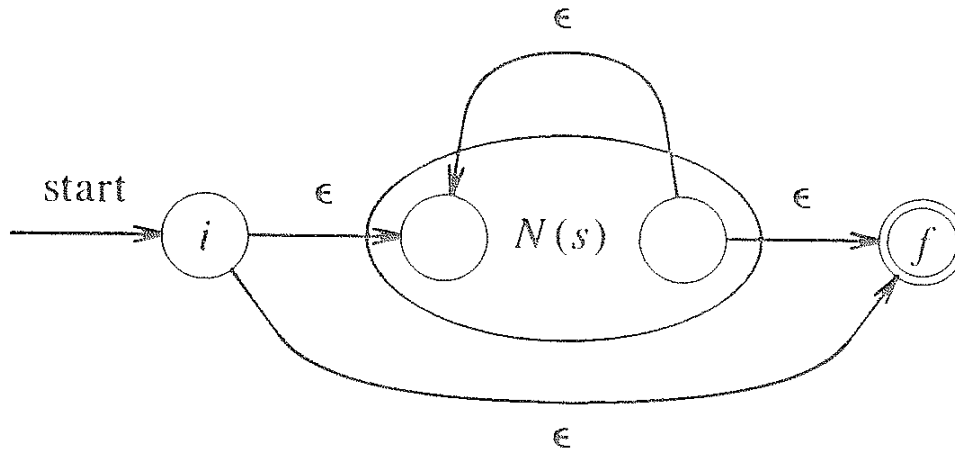
For the regular expression $s.t$,



RE to NFA (cont.)

For the regular expression s^* ,

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

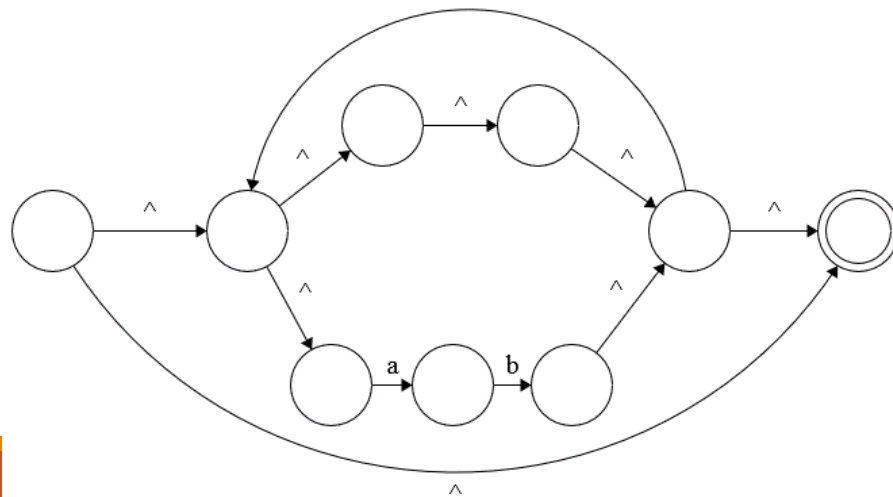
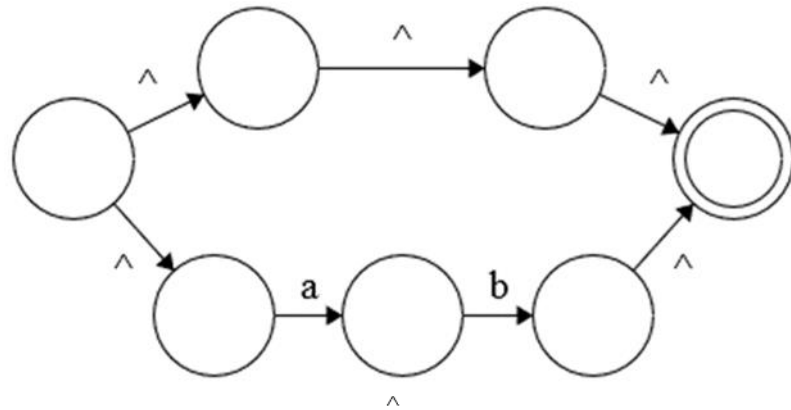
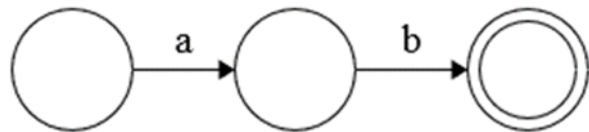
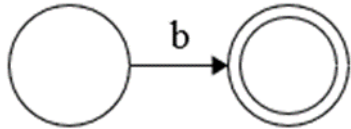
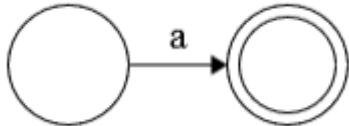
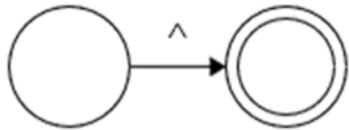


Every time we construct a new state, we give it a distinct name.

Example 08

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

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



Example 09

Find an NFA that accepts each regular Expression

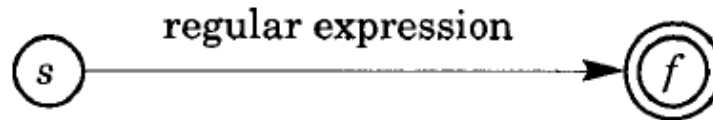
$a^*a + ab$

$(aab)^*ab$

ab^*aa

Regular Expression to Finite Automaton

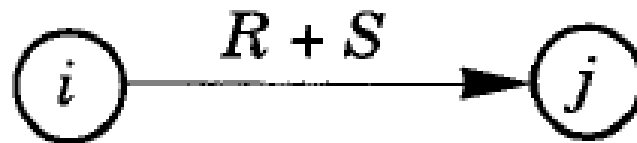
Given a regular expression, we start the algorithm with a machine that has a **start state**, a **single final state**, and an edge labeled with the given regular expression as follows:



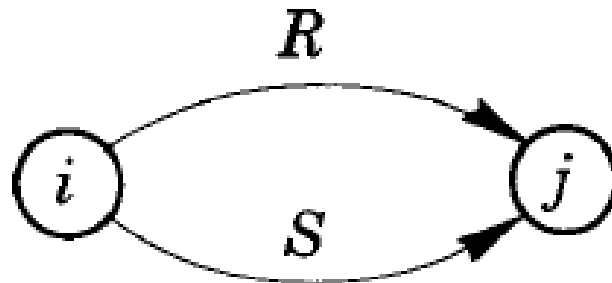
Applying the following rules until all edges are labeled with either a letter or Λ :

Regular Expression to Finite Automaton (cont.)

1. If an edge is labeled with \emptyset , then **erase** the edge.
2. Transform any diagram like



into the diagram



Regular Expression to Finite Automaton (cont.)

3. Transform any diagram like

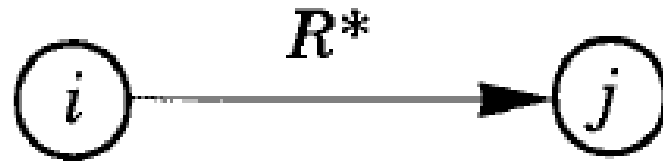


into the diagram

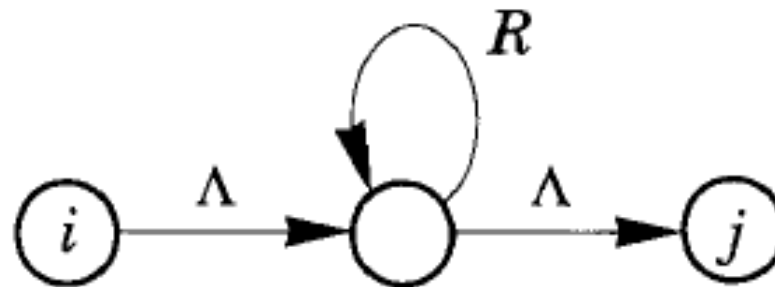


Regular Expression to Finite Automaton (cont.)

4. Transform any diagram like



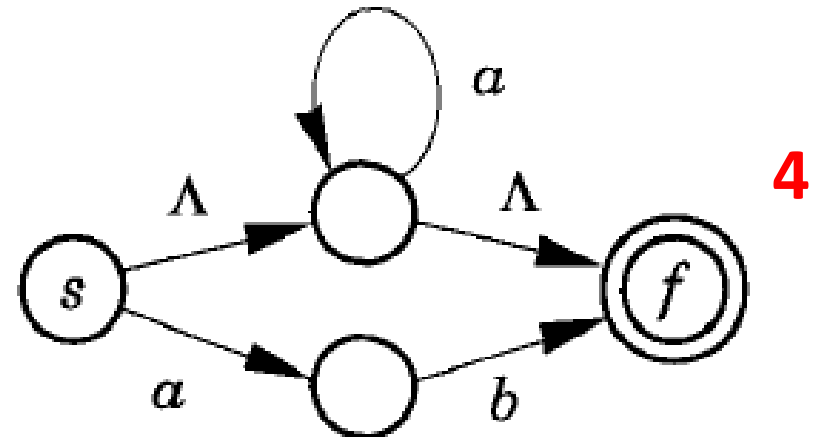
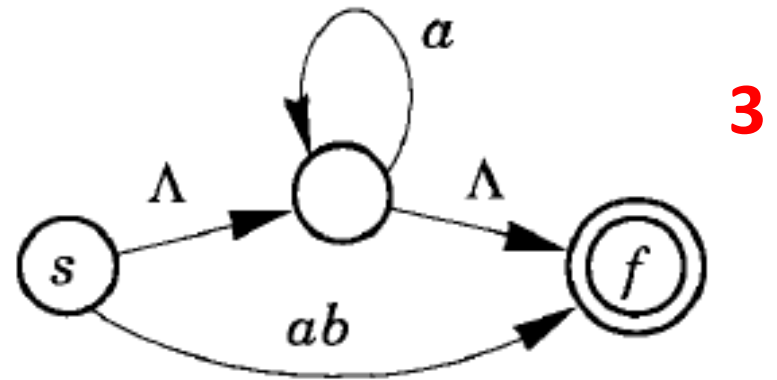
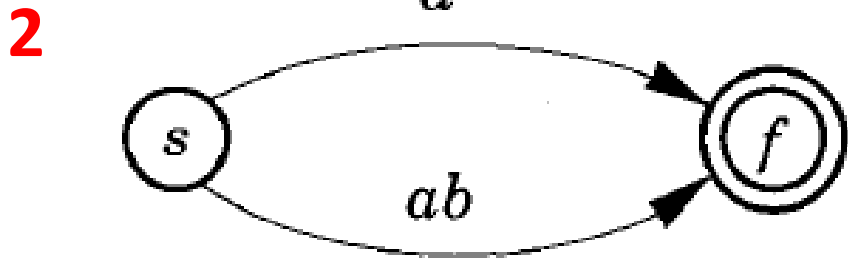
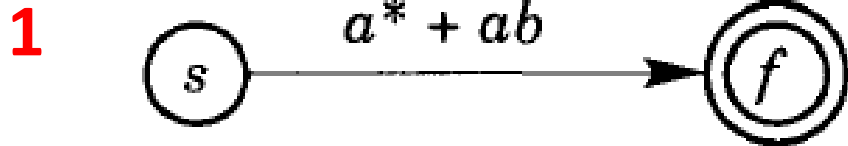
into the diagram



End of Algorithm

Example 10

Construct an NFA for $a^* + ab$



Example 11

Find an NFA that accepts each regular Expression

$\Lambda + b$

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

$a^*a + ab$

$(aab)^*ab$

ab^*aa

