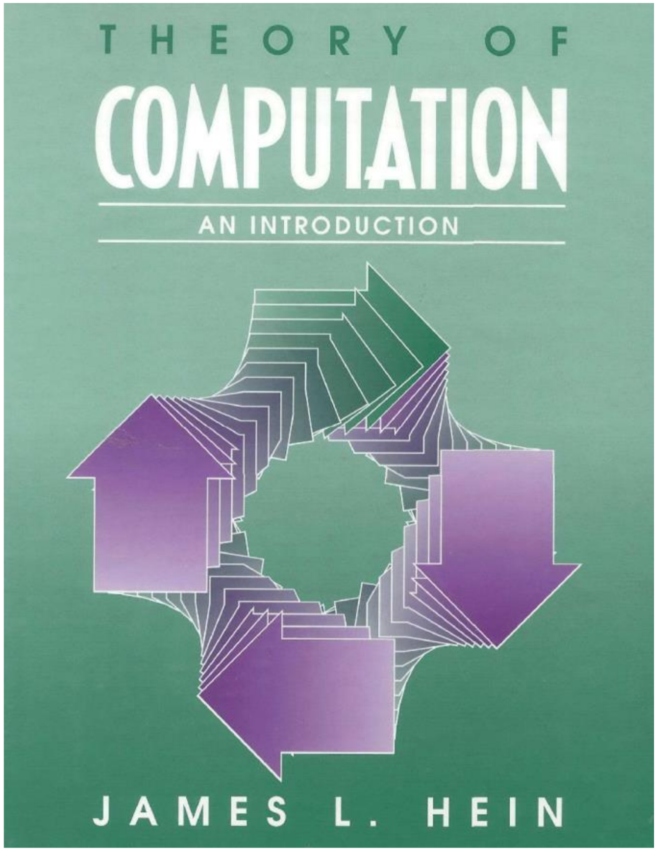
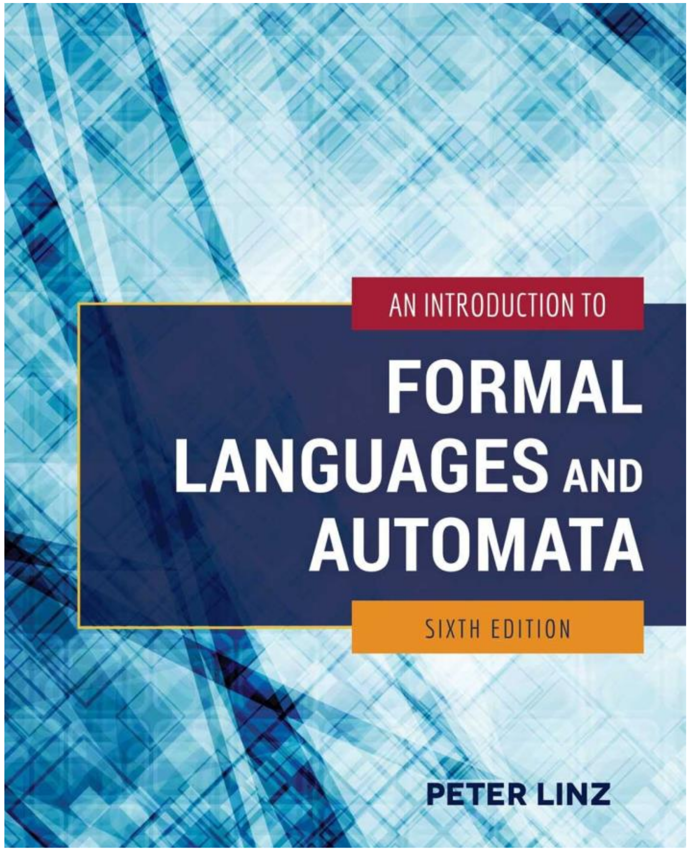
Automata and Formal Languages

Lecture 04

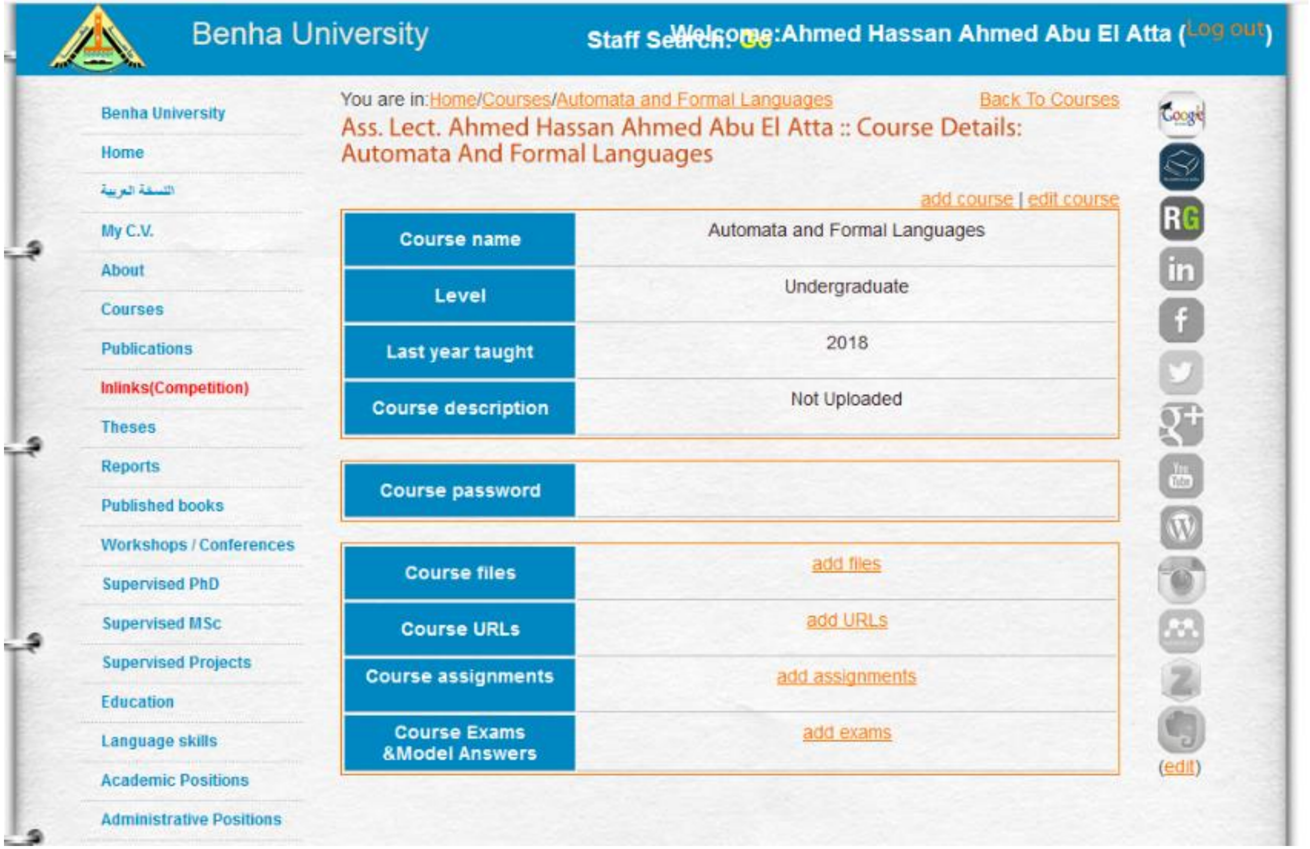
Books


THEORY OF
COMPUTATION
AN INTRODUCTION
JAMES L. HEIN


AN INTRODUCTION TO
FORMAL
LANGUAGES AND
AUTOMATA
SIXTH EDITION
PETER LINZ

PowerPoint

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



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, ø, and a are regular expressions for all a ∈ 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 ø a b

Λ + b

b*

a + (b.a)

(a + b).a

a.b*

a* + b*

The Operations Priority

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

.

+   lowest (do it last).

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

Languages Associated with Regular Expressions

$L(\varnothing) = \varnothing$,

$L(\Lambda) = (\Lambda)$,

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

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

$L(R \bullet S) = L(R)L(S)$ (language product),

$L(R^*) = L(R)^*$ (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:

L(a + bc*) = L(a) ∪ L(bc*)

= L(a) ∪ (L(b).L(c*))

= L(a) ∪ (L(b).L(c)*)

= {a} ∪ ({b} . {c}*)

= {a} ∪ ({b} {Λ, c, c^2, ... , c^n, ...})

= {a} ∪ {b, bc, bc^2 , ... ,bc^n, ... }

= {a, b, bc, bc^2 , ... , bc^n, ... }

Example 02

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

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

Example 03

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

L ((a + b)*.(a + bb))= L ((a+b)*).L (a + bb)

= (L (a+b))*.(L (a) ∪ L (bb))

= ({a, b})*.({a} ∪ L (b)L(b))

= {Λ,a,b,aa,ab,ba,bb ...}{a, bb}

= {a, bb, aa, abb, ba, bbb, ...}

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

- $L(R) = \{w \in \Sigma^* : w$ 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 ∈ {0, 1}* : w has no pair of consecutive zeros}.

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

*Example 06

{Λ , 0, 1, 01, 10, 11, 010, 011, 101, 110, 111, 0101, 0110, 1010, 0111, 1110,…….}

= {Λ , 1, 01, 11, 011, 101, 111, 0101, 0111,….}

∪ { 0, 10, 010, 110, 0110, 1010, 1110, ….}

= {Λ , 1, 01, 11, 011, 101, 111, 0101, 0111,….}

∪ { 0, 10, 010, 110, 0110, 1010, 1110, ….}

= {Λ , 1, 01, 11, 011, 101, 111, 0101, 0111,….}

∪ {Λ, 1, 01, 11, 011, 101, 111, ….}{0}

={1, 01}* ∪ {1, 01}*.{0}

={1, 01}*{Λ, 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)

$$R + T = T + R,$$
$$R + \varnothing = \varnothing + R = R,$$
$$R + R = R,$$
$$(R + S) + T = R + (S + T).$$

2. (· properties)

$$R\varnothing = \varnothing R = \varnothing,$$
$$R\Lambda = \Lambda R = R,$$
$$(RS)T = R(ST).$$

3. (Distributive properties)

$$R(S + T) = RS + RT,$$
$$(S + T)R = SR + TR.$$

Properties of Regular Expressions (cont.)

(Closure properties)

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

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

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

   $R^* = (R + \ldots + R^k)^*$ for any $k \geq 1,$

   $R^* = \Lambda + R + \ldots + R^{k-1} + R^kR^*$ 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. $(R^*S)^* = \Lambda + (R + S)^*S,$

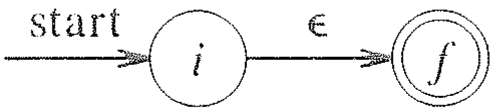
   $(RS^*)^* = \Lambda + R\,(R + S)^*.$

RE to NFA

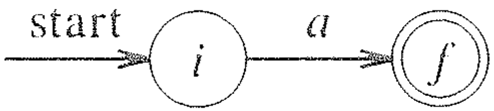
First parse *r* into its constituent sub expressions.

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

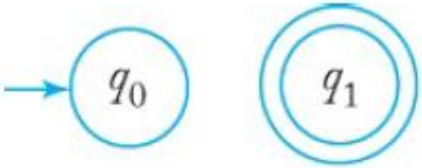
- for ε



- for *a* in Σ

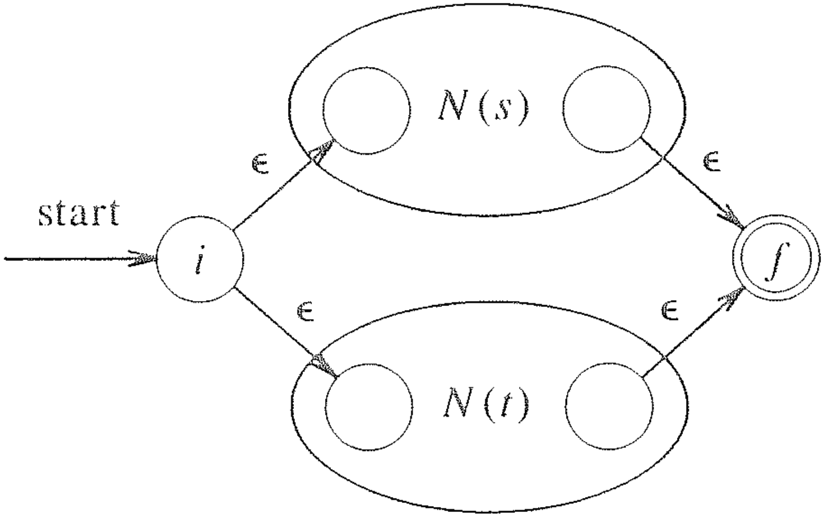


- for *ø*

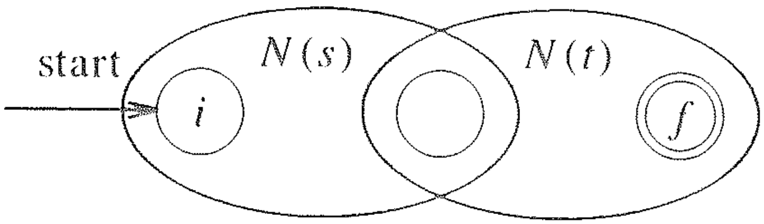


RE to NFA (cont.)

For the regular expression *s+t*,



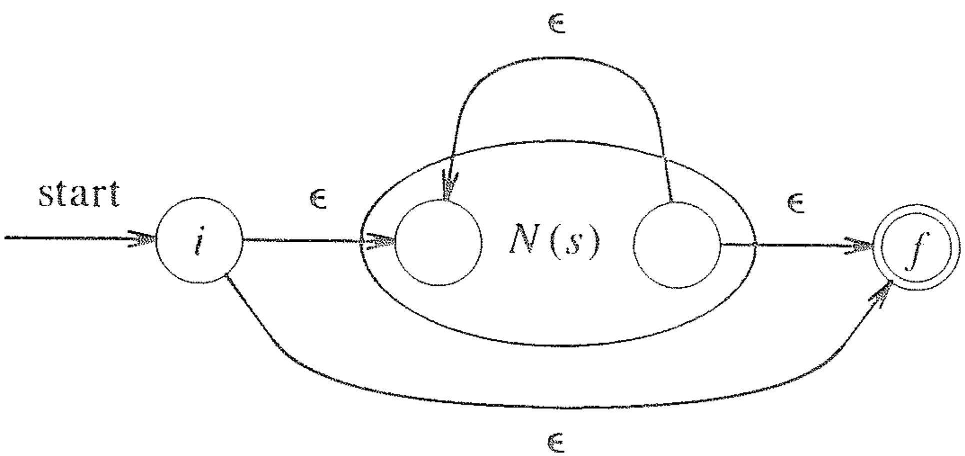
For the regular expression *s.t*,



RE to NFA (cont.)

For the regular expression *s**,

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

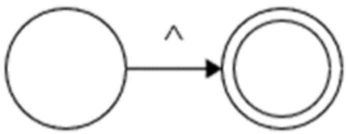


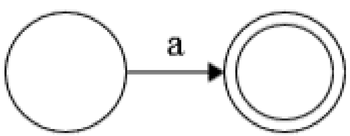
Every time we construct a new state, we give it a distinct name.

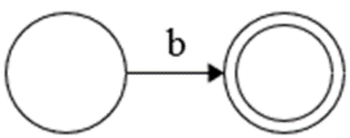
Example 08

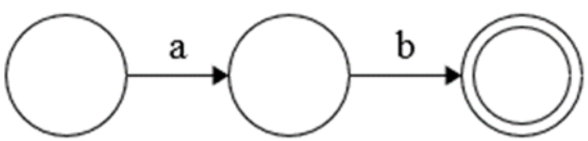
Find an NFA that accepts L (r), where

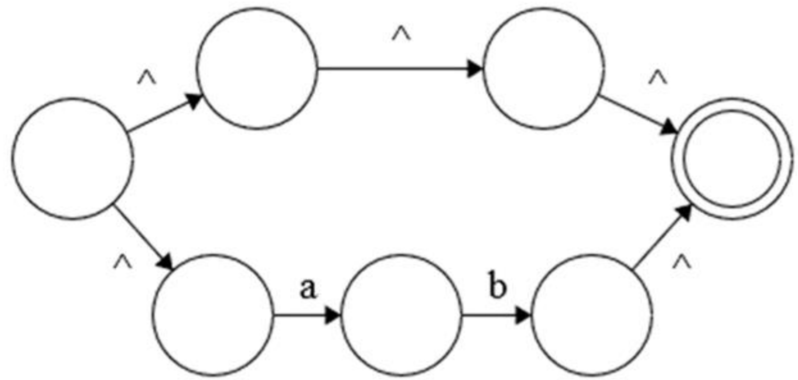
r = (Λ + ab)*

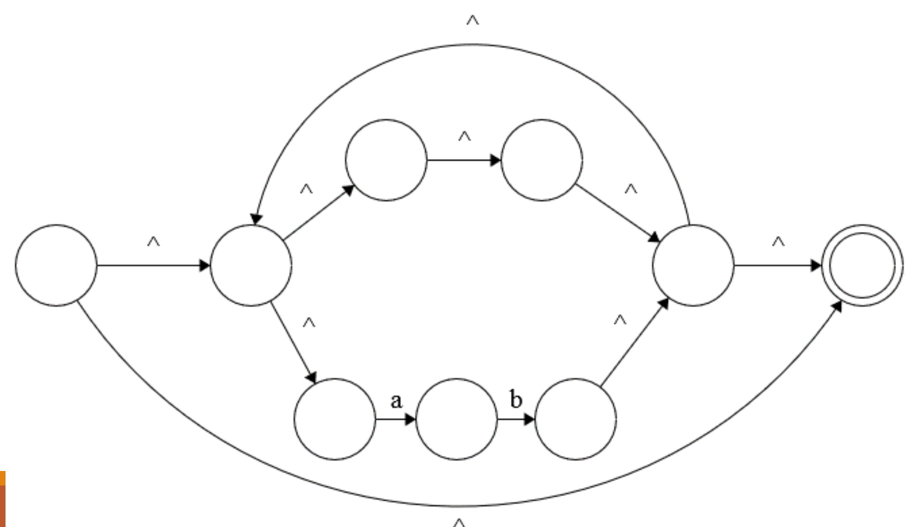












Example 09

Find an NFA that accepts each regular Expression

a*a + ab

(aab)*ab

ab*aa