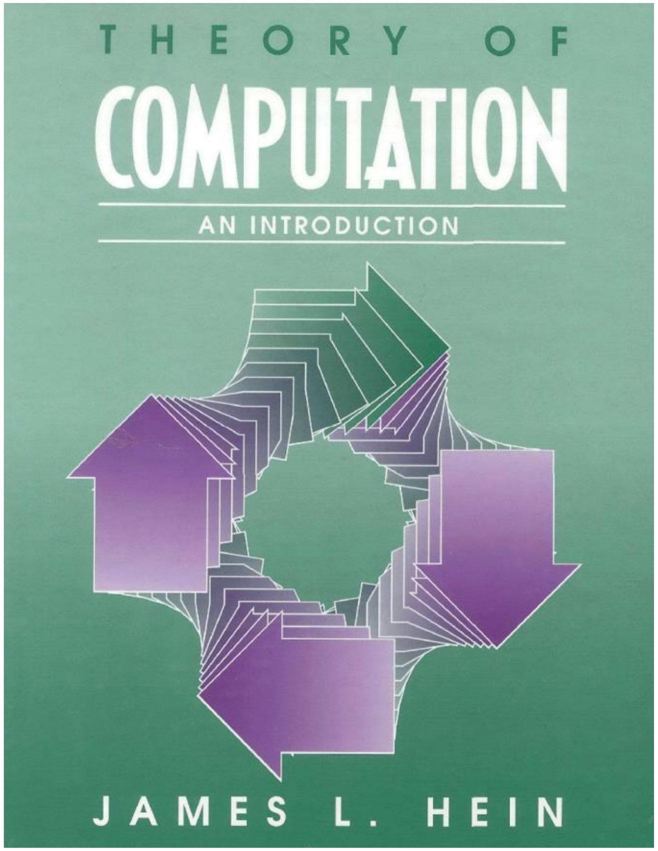
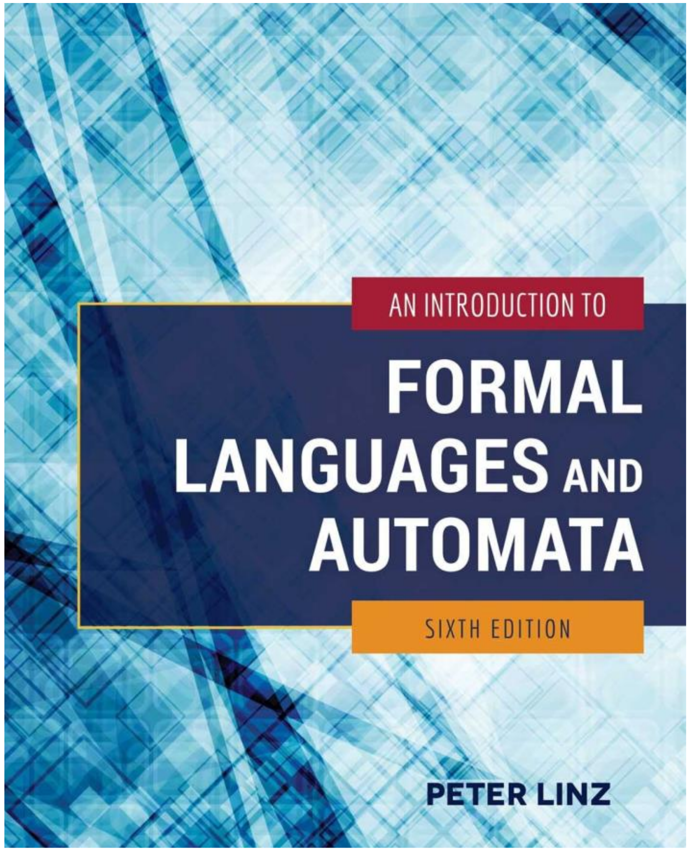
Automata and Formal Languages

Lecture 08

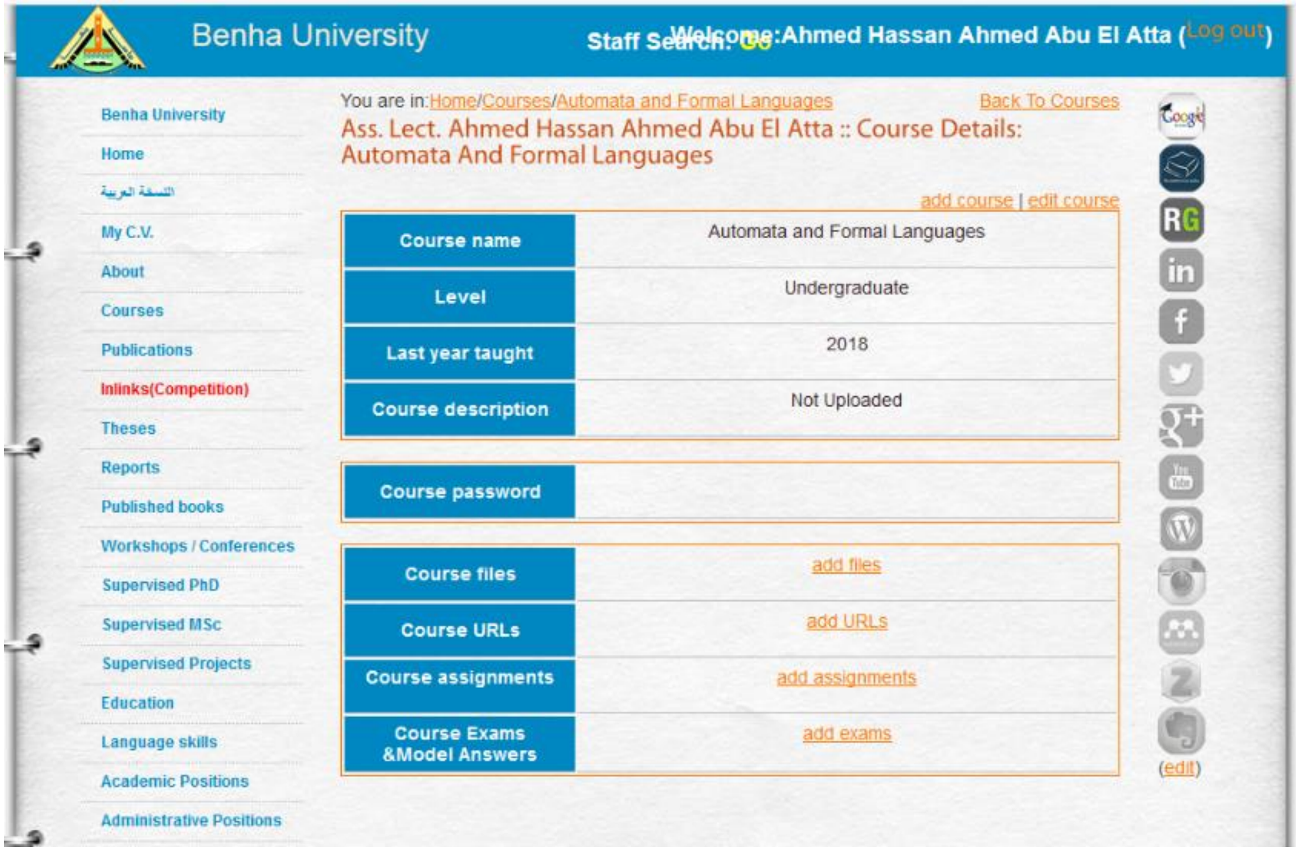
Books


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



Minimum-State DFAs

MINIMUM-STATE DFAS

Agenda

- Minimum-State DFAs
- Example 1
- Minimum-State DFAs Algorithm
- Example 2
- Example 3

Minimum-State DFAs

One way to try and simplify the DFA for some regular expression is to algebraically transform the regular expression into a simpler one before starting construction of the DFA.

$$\lambda + a + aaa^* = a^*.$$

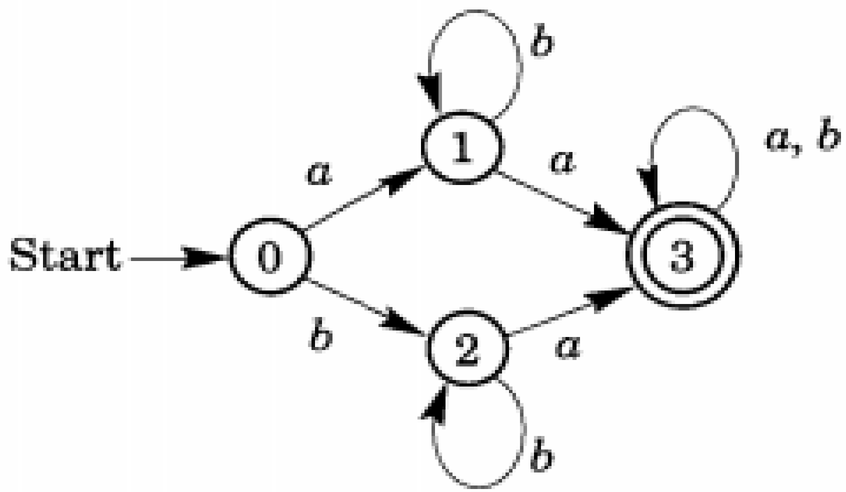
*Every regular expression has a unique minimum-state DFA.

How to transform a DFA into a minimum state DFA?

*The key idea is to define two states s and t to be equivalent if for every string w, the transitions

T(s, w) and T(t, w) are either both final or both nonfinal.

Example 1:

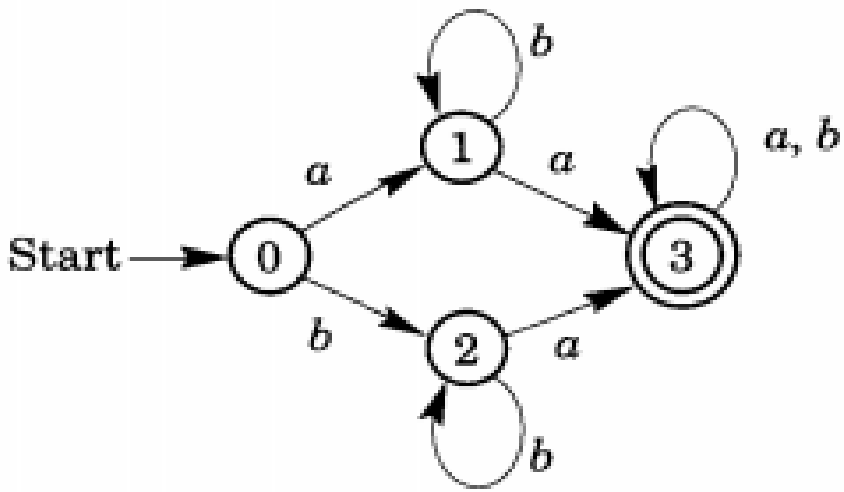


	T	a	b
start	0	1	2
	1	3	1
	2	3	2
final	3	3	3

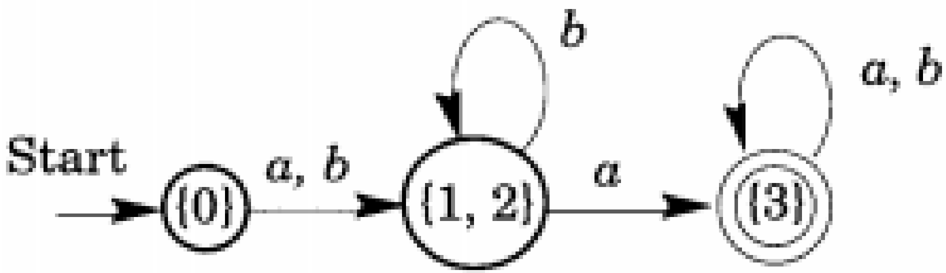
States 1 and 2 are equivalent. So for any string w, both T(1, w) and T(2, w) are either both final or both nonfinal.

{0}, {1, 2}, and {3}.

Example 1:



	T	a	b
start	0	1	2
	1	3	1
	2	3	2
final	3	3	3



	T_{min}	a	b
start	{0}	{1, 2}	{1, 2}
	{1, 2}	{3}	{1, 2}
final	{3}	{3}	{3}

Minimum-State DFAs

Input: A DFA with set of states S and transition table T. Assume

Output: A minimum-state DFA recognizing the same regular language as the input DFA.

1. Construct the equivalent pairs of states by calculating the descending sequence of sets of pairs $E_0 \supset E_1 \supset \ldots$ defined as follows:

2. E_0 = {{s, t) I s and t are distinct and either both states are final or both states are non-final}.

   E_{i+1} = {{s, t} I {s, t} $\in E_i$ and for every a $\in$ A either T(s, a) = T(t, a) or {T(s, a), T(t, a)} $\in E_i$}.

   The computation stops when $E_k = E_{k+1}$ for some index k.

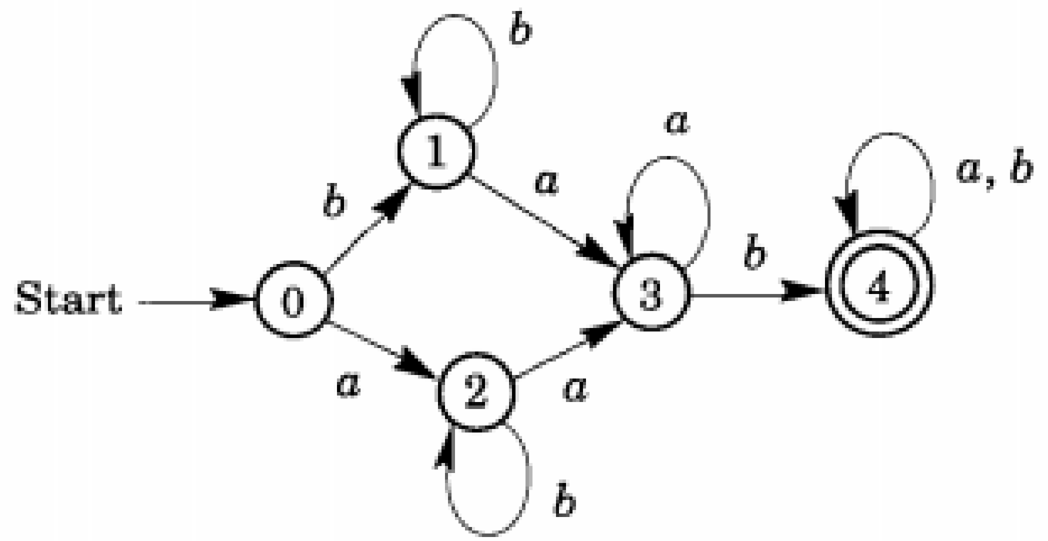
3. The start state is the equivalence class containing the start state.

4. A final state is any equivalence class containing a final state .

5. The transition table T_{min} for the minimum-state DFA is defined as follows, where [s] denotes the equivalence class containing s and a is any letter: $T_{min}([s], a) = [T(s, a)]$.

End of Algorithm

Example 2



T	a	b
0	2	1
1	3	1
2	3	2
3	3	**4**
4	**4**	**4**

Example 2

T	a	b
0	2	1
1	3	1
2	3	2
3	3	**4**
4	**4**	**4**

E0 = {{0, 1}, {0, 2}, {0, 3}, {1, 2}, {1, 3}, {2, 3}}.

{T(s, x),T(t, x)} ϵ E0 or T(s, x) =T(t, x).

{T(0, a),T(1, a)} ={2,3} ϵ E0.

T(0, b)=T(1, b) .

{T(0, a),T(3, a)} ={2,3} ϵ E0.

T(0, b)={1}

T(3, b)={4} . Eliminate { 0,3}

E1 = {{0, 1}, {0, 2}, {1, 2}}.

E2= {{1, 2}}. , E3=E2= {{1, 2}}.

Example 2

{0}, {1, 2}, {3}, {4}

Tmin	a	b
{0}	{1,2}	{1,2}
{1,2}	{3}	{1,2}
{3}	{3}	**{4}**
{4}	**{4}**	**{4}**

Example 3

compute the minimum-state DFA for the DFA given by the following transition table:

T	a	b
0	1	2
1	**4**	1
2	**4**	3
3	**4**	3
4	**4**	**5**
5	**5**	**5**

Example 3

E0 = {{0, 1}, {0, 2}, {0, 3}, {1, 2}, {1, 3}, {2, 3},{4,5}}.

E1 = { {1, 2}, {1, 3}, {2, 3},{4,5}}.

E2 = E1 = { {1, 2}, {1, 3}, {2, 3},{4,5}}.

Three classes {0}, {1,2, 3},{4,5}

T	a	b
0	1	2
1	**4**	1
2	**4**	3
3	**4**	3
4	**4**	**5**
5	**5**	**5**

Example 3

Three classes {0}, {1,2, 3},{4,5}

T	a	b
{0}	{1,2, 3}	{1,2, 3}
{1,2, 3}	**{4,5}**	{1,2, 3}
{4,5}	**{4,5}**	**{4,5}**