THEOREM 7

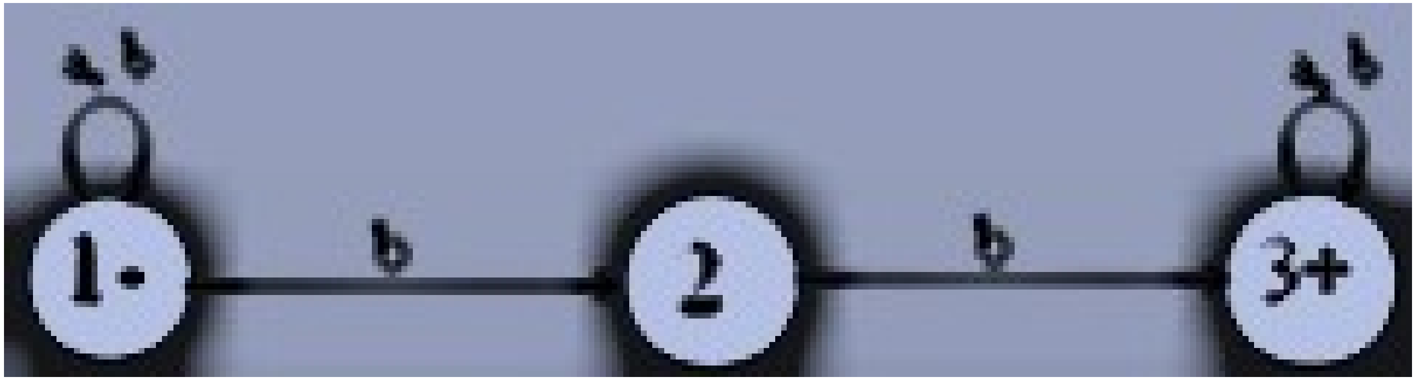
- **for every NFA, there is some FA that accepts exactly the same language.**

THEOREM 7

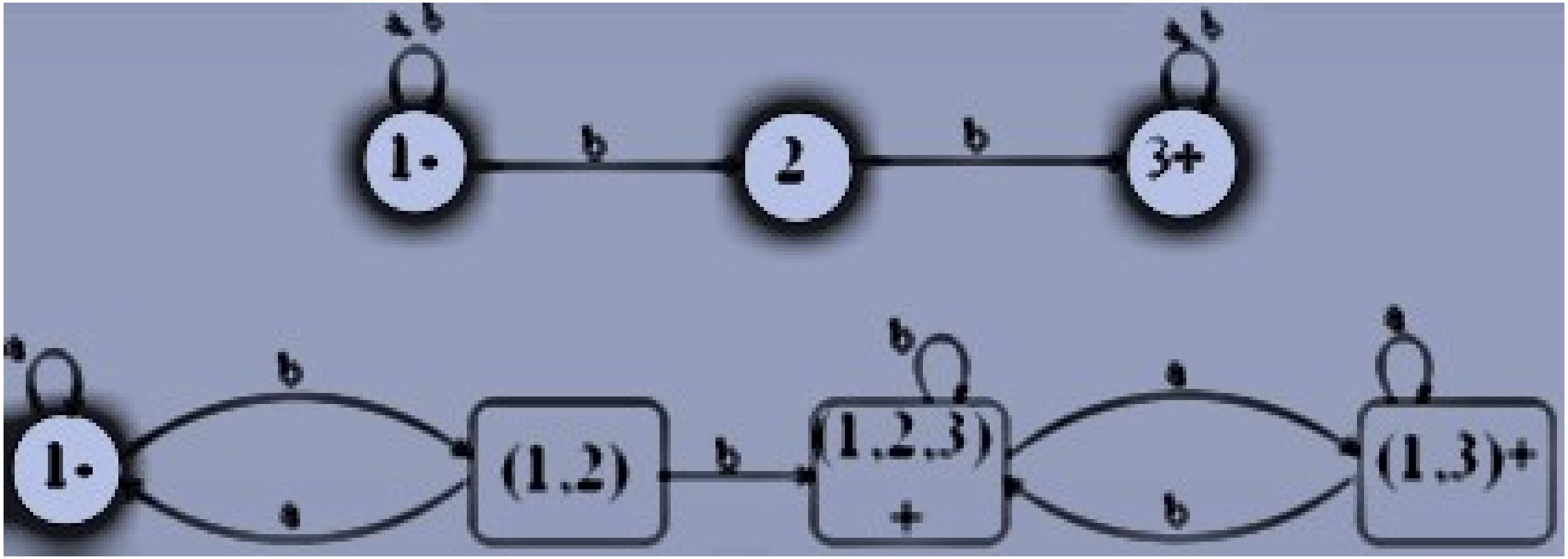
- **for every NFA, there is some FA that accepts exactly the same language.**
- **Proof 1**

NFA to FA

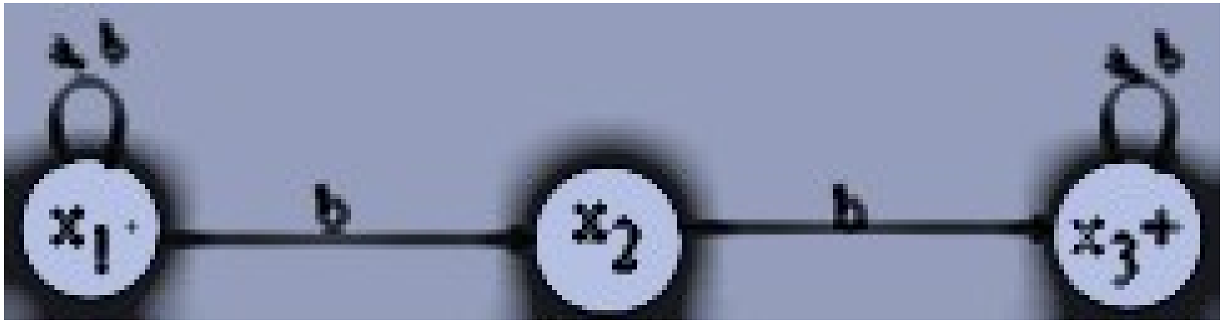
- Build an FA corresponding to the following NFA which accepts the language of strings **containing bb**



NFA to FA cont.



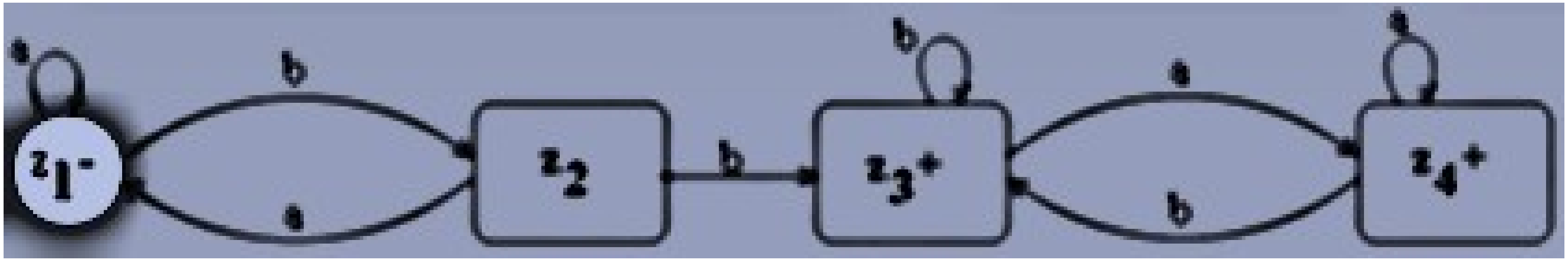
Example continued …



Old states	New states after reading	
	a	b
$Z_1 - \equiv x_1$	$x_1 \equiv z_1$	$(x_1, x_2) \equiv z_2$
$z_2 \equiv (x_1, x_2)$	$(x_1, \Lambda) \equiv x_1 \equiv z_1$	$(x_1, x_2, x_3) \equiv z_3$
$z_3 + \equiv (x_1, x_2, x_3)$	$(x_1,x_3) \equiv z_4$	$(x_1 ,x_2,x_3) \equiv z_3$
$z_4+ \equiv (x_1,x_3)$	$(x_1,x_3) \equiv z_4$	$(x_1 ,x_2,x_3) \equiv z_3$

- The corresponding transition diagram follows as

Example continued …



Alg - RegExp to DFA via GenNFA

- Given a RegExp: ab+ac+ad+ae+af

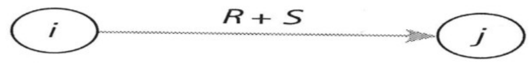
1. Construct a simple GenNFA with two states with one arc between the two states labeled with the regular expression.

2. The source of the arc is the start state

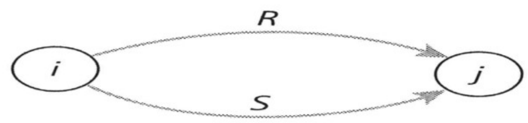
3. The target of the arc is the final

Extend the GenNFA

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

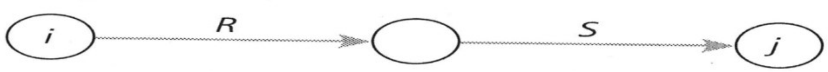


into the diagram

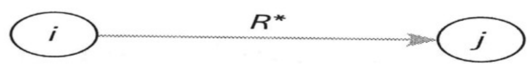


3. Transform any diagram like

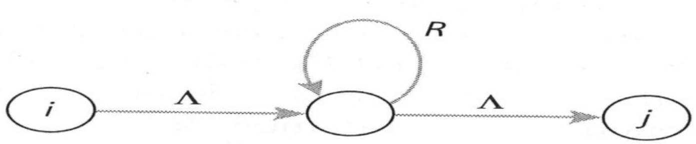
into the diagram



4. Transform any diagram like



into the diagram



(a+b)*cd*