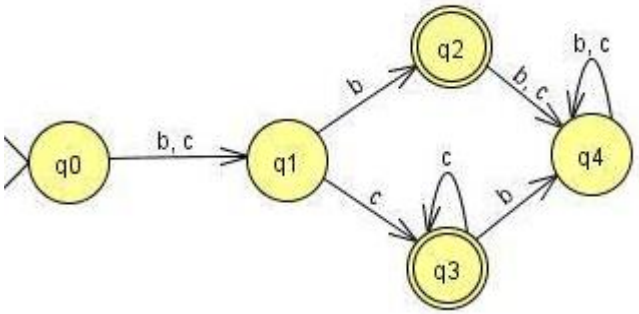
Chapter 17

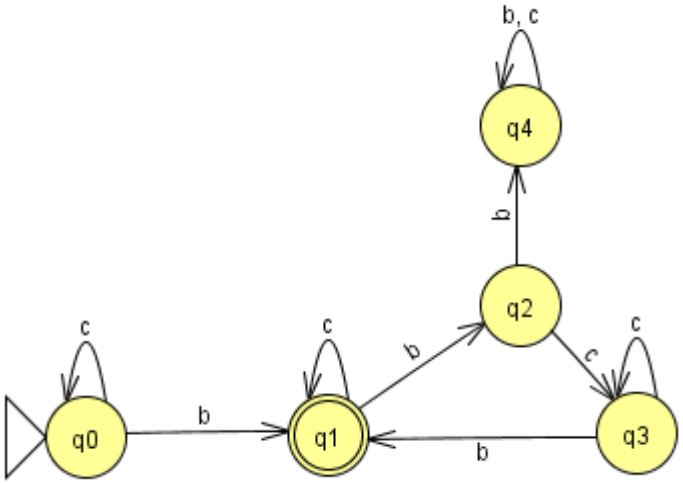
Finite Automata

Deterministic finite automata

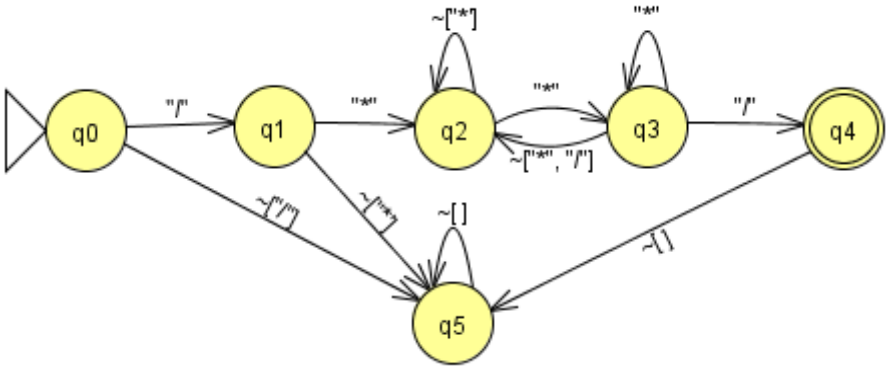


Converting DFA to regular expression

`c*b(c|bcc*b)*`



/*...*/ comments

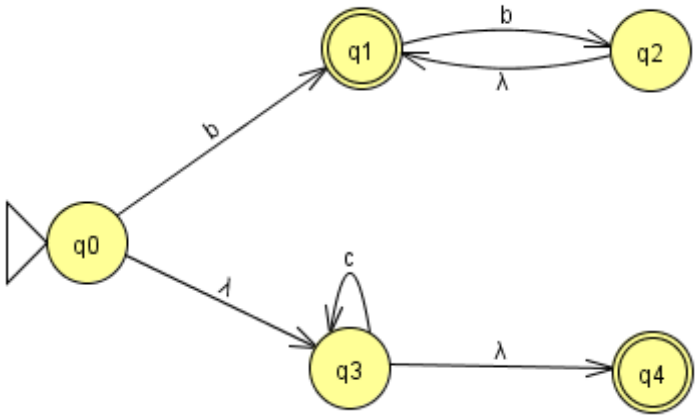


"/*" (~["*"])* "*" ("*" | ~["*","/"](~["*"])* "*")* "/"

Code for DFA

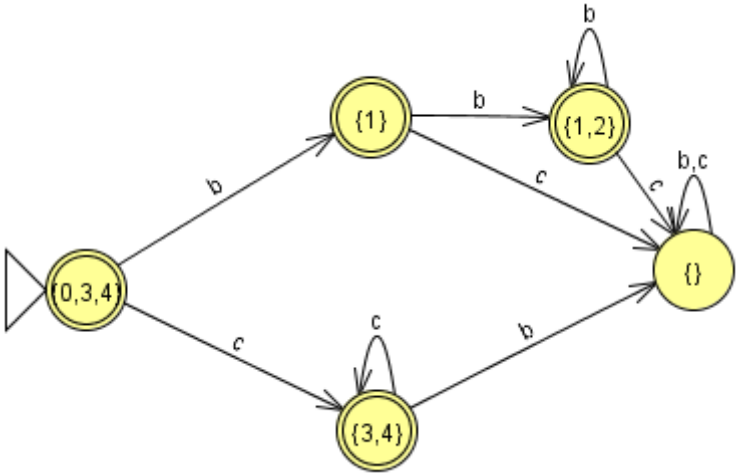
```
36      while (currentToken != '#')
37      {
38        switch(currentState)
39        {
40          case 0:
41            if (currentToken == 'b') currentState = 1;
42            else
43            if (currentToken == 'c') currentState = 1;
44            break;
45          case 1:
46            if (currentToken == 'b') currentState = 2;
47            else
48            if (currentToken == 'c') currentState = 3;
49            break;
50          case 2:
51            if (currentToken == 'b') currentState = 4;
52            else
53            if (currentToken == 'c') currentState = 4;
54            break;
55          case 3:
56            if (currentToken == 'b') currentState = 4;
57            else
58            if (currentToken == 'c') currentState = 3;
59            break;
60          case 4:
61            if (currentToken == 'b') currentState = 4;
62            else
63            if (currentToken == 'c') currentState = 4;
64            break;
65        }
66        advance();
67      }
```

Nondeterministic finite automata

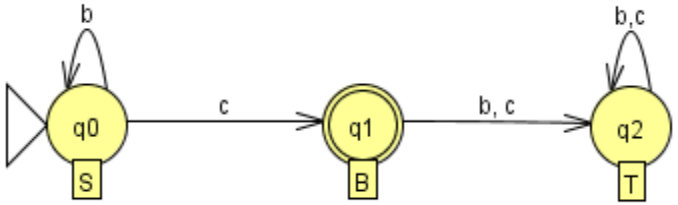


$$\{q0\} \xrightarrow{\lambda} \{q0,q1,q2\} \xrightarrow{b} \{q1\} \xrightarrow{\lambda} \{q1\} \xrightarrow{b} \{q2\} \xrightarrow{\lambda} \{q1,q2\} \xrightarrow{b} \{q2\} \xrightarrow{\lambda} \{q1,q2\}$$

Converting an NFA to a DFA



Converting DFA to regular grammar



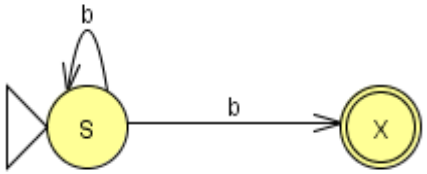
$S \rightarrow bS$
$S \rightarrow cB$
$B \rightarrow bT$
$B \rightarrow cT$
$T \rightarrow bT$
$T \rightarrow cT$
$B \rightarrow \lambda$

Convering regular grammar to NFA

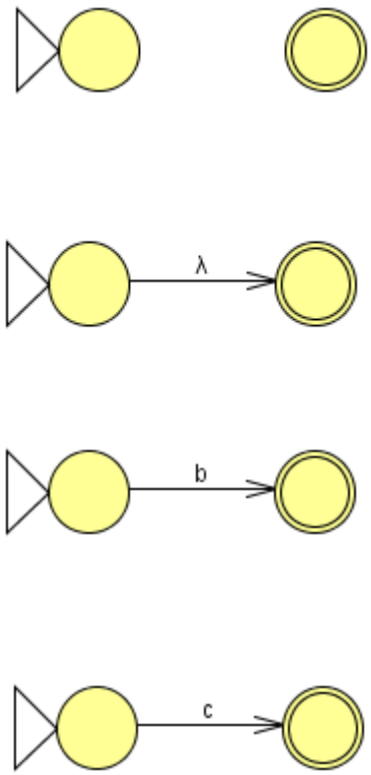
S -> bS
S -> b

Convert to

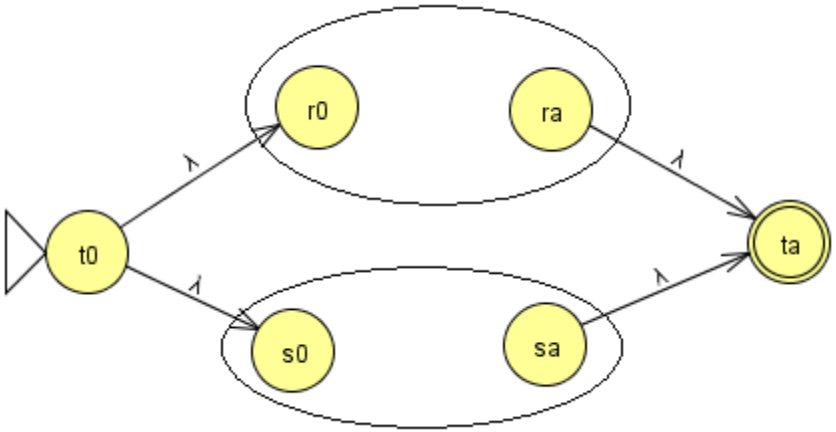
S -> bS
S -> bX
X -> λ



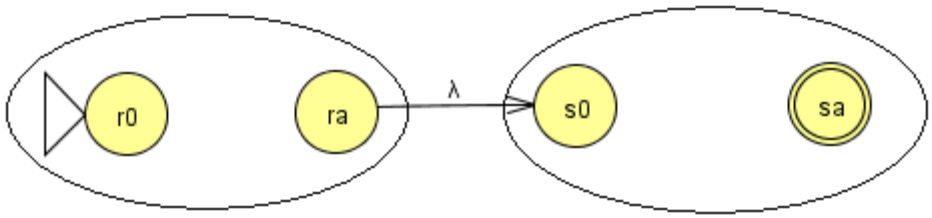
Converting regular expression to NFA



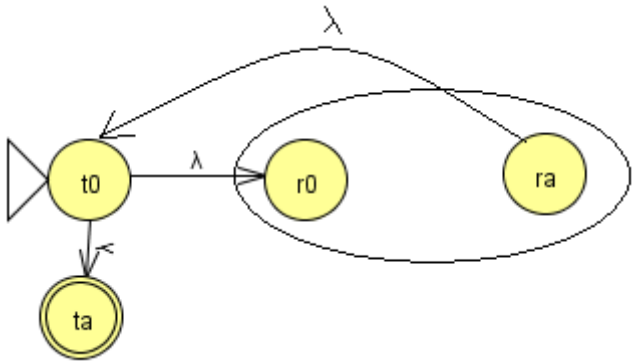
Building more complex expressions



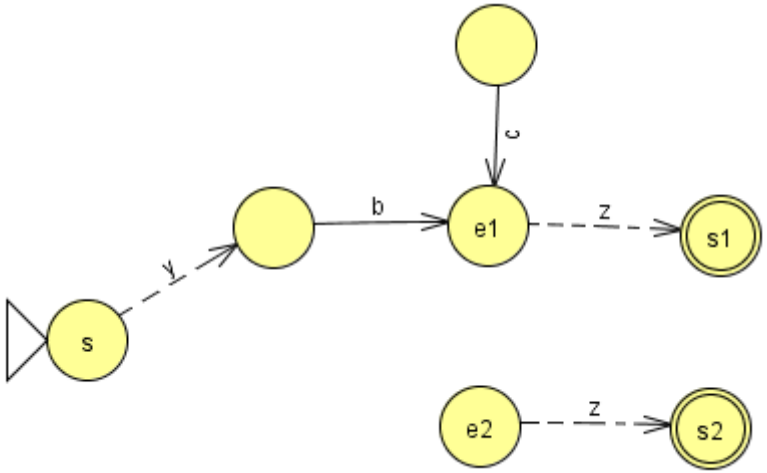
Building more complex expressions



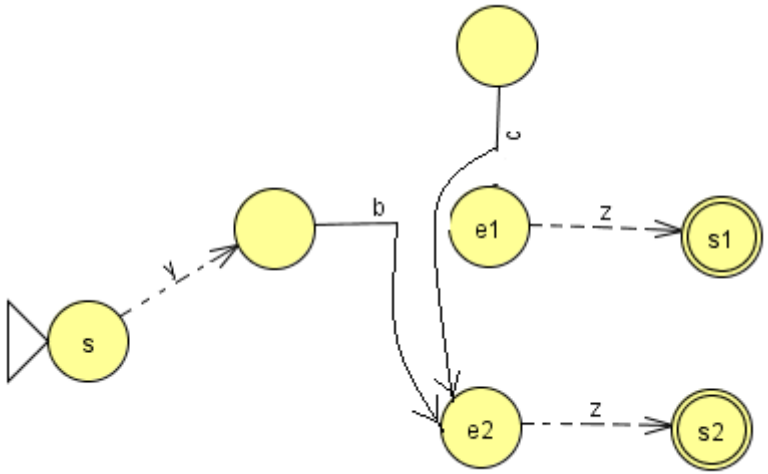
Constructing more complex expressions



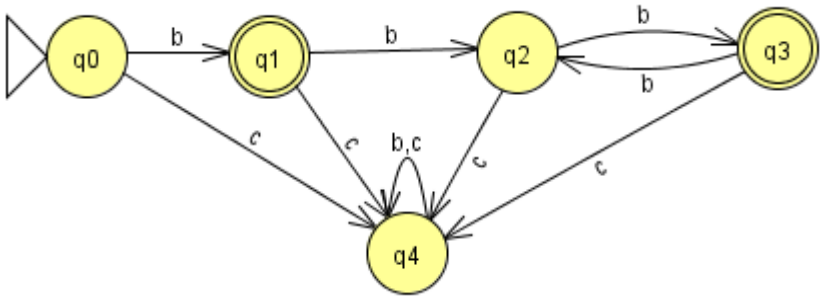
Assume e1 and e2 are equivalent



Redirection does not change language



Computing minimal DFA

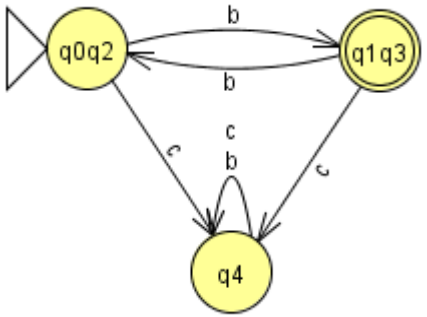


		b	c
Block I states	q0	II	I
	q2	II	I
	q4	I	I
Block II states	q1	I	I
	q3	I	I

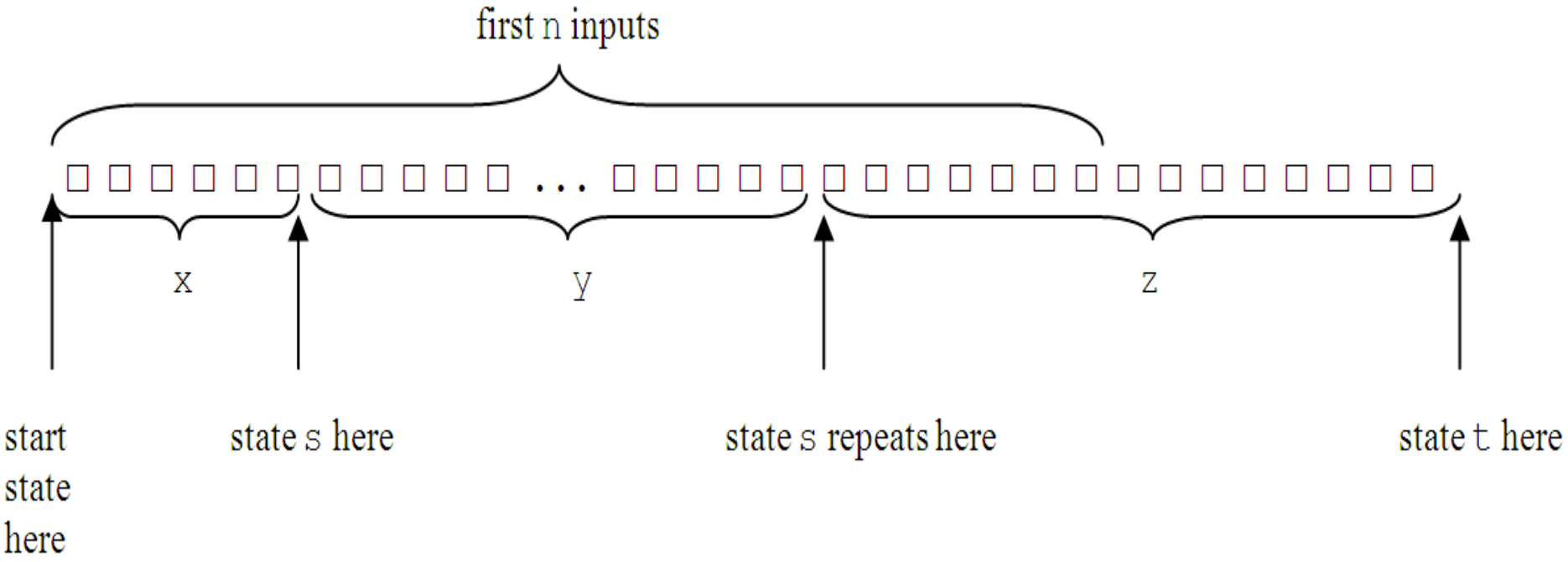
Computing minimal DFA

b)

		b	c
Block 1a states	q0	II	Ib
	q2	II	Ib
Block 1b states	q4	Ib	Ib
Block II states	q1	I	Ib
	q3	I	Ib



Pumping lemma for regular languages



Pumping lemma

Let L be a regular language. There exists an n such that if u is in L and $|u| \geq n$, then $u = xyz$ where

(i) $|y| > 0$
(ii) $|xy| \leq n$
(iii) $xy^iz \in L$ *for all* $i \geq 0$

$PAIRED = \{b^ic^i : i \geq 0\} = \{\lambda,\ bc,\ bbcc,\ bbbccc,\ bbbbcccc,\ \ldots\}$