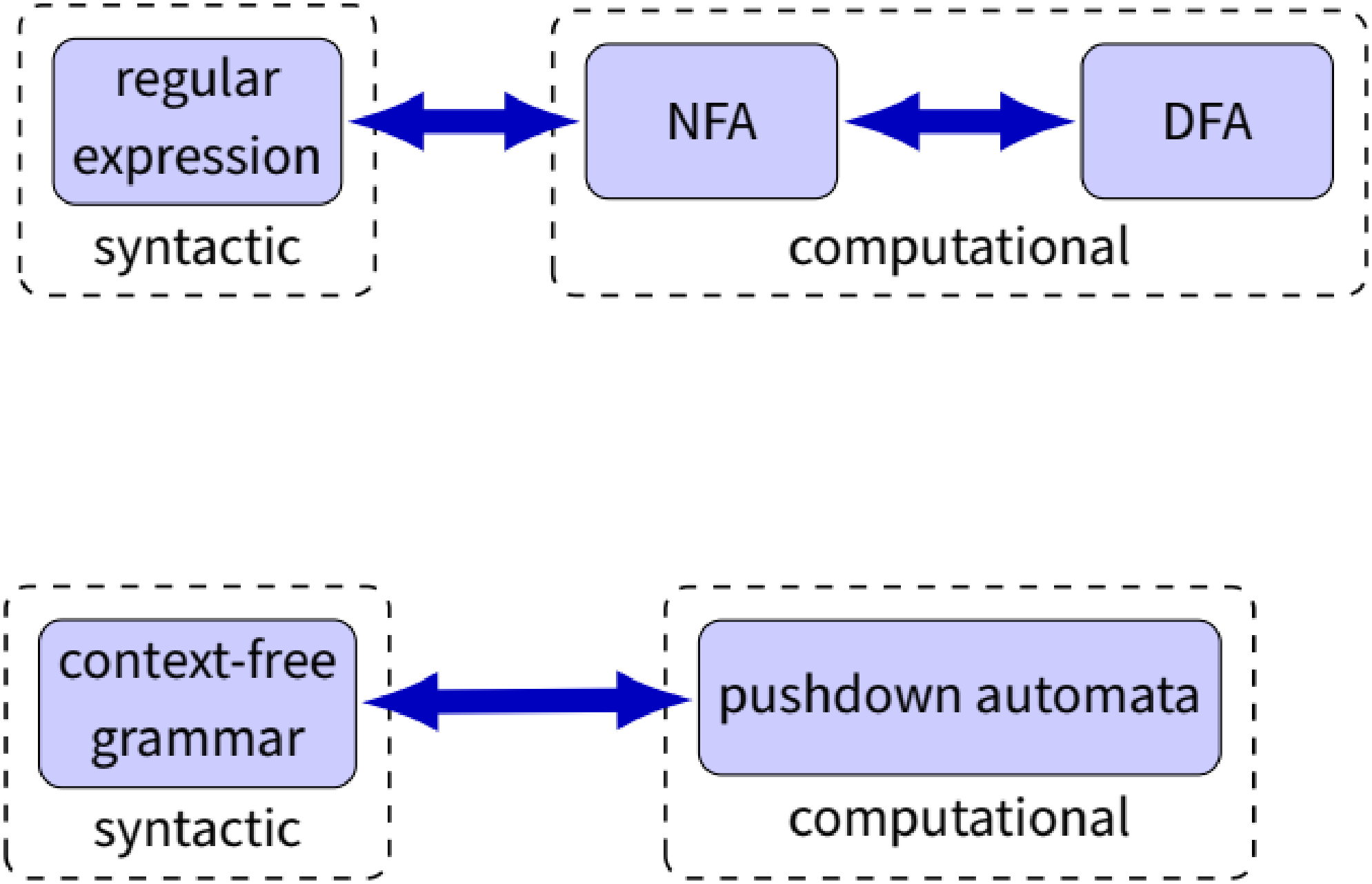
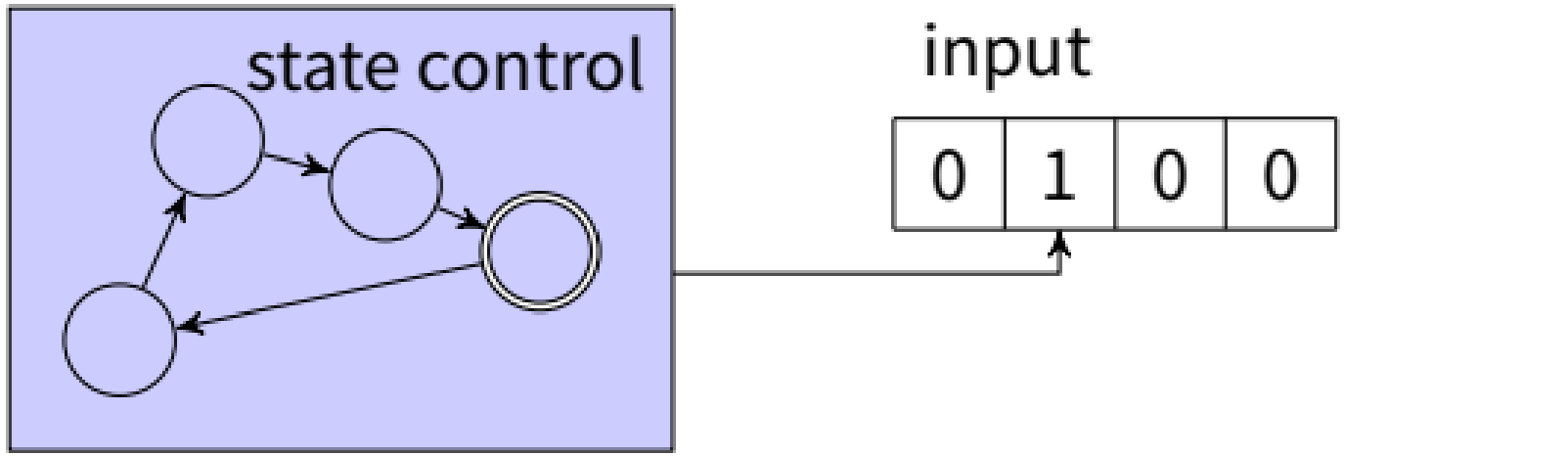
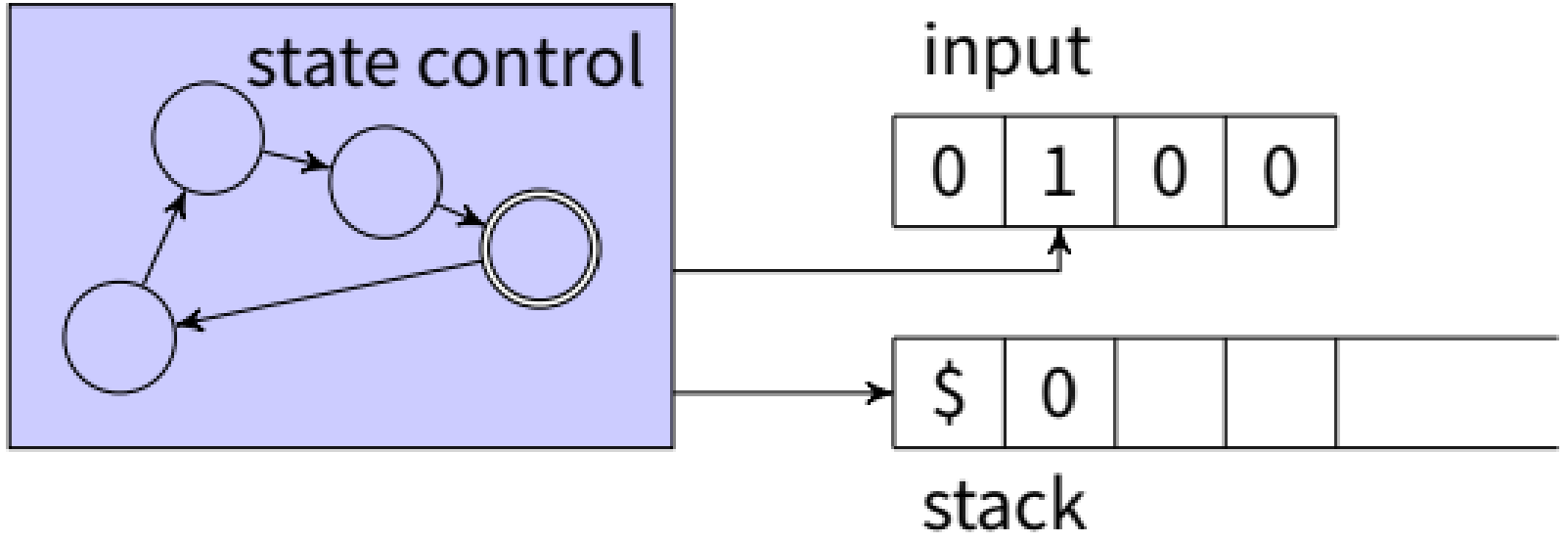
Pushdown Automata

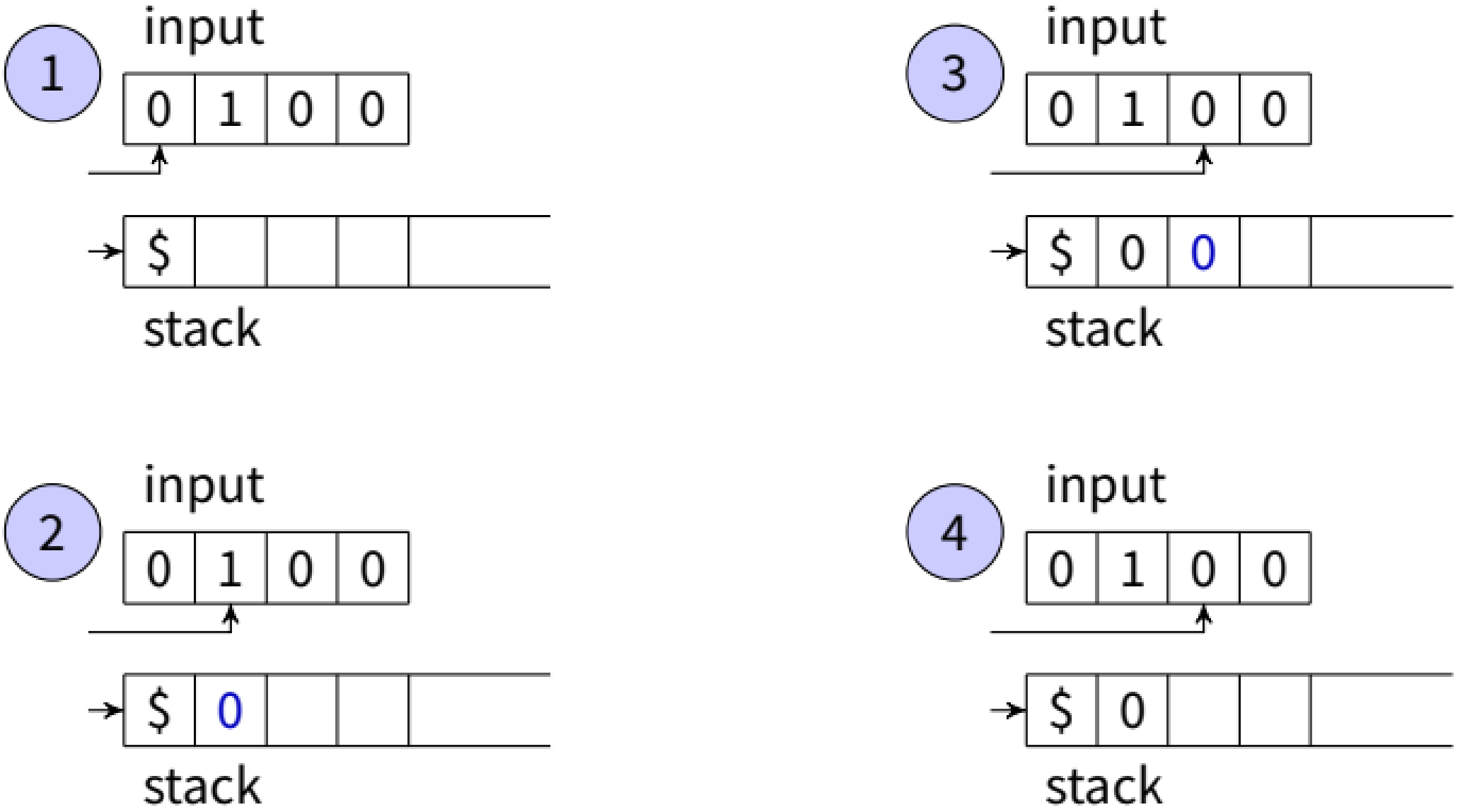

regular expression
syntactic
NFA
DFA
computational
context-free grammar
syntactic
pushdown automata
computational

NFA:

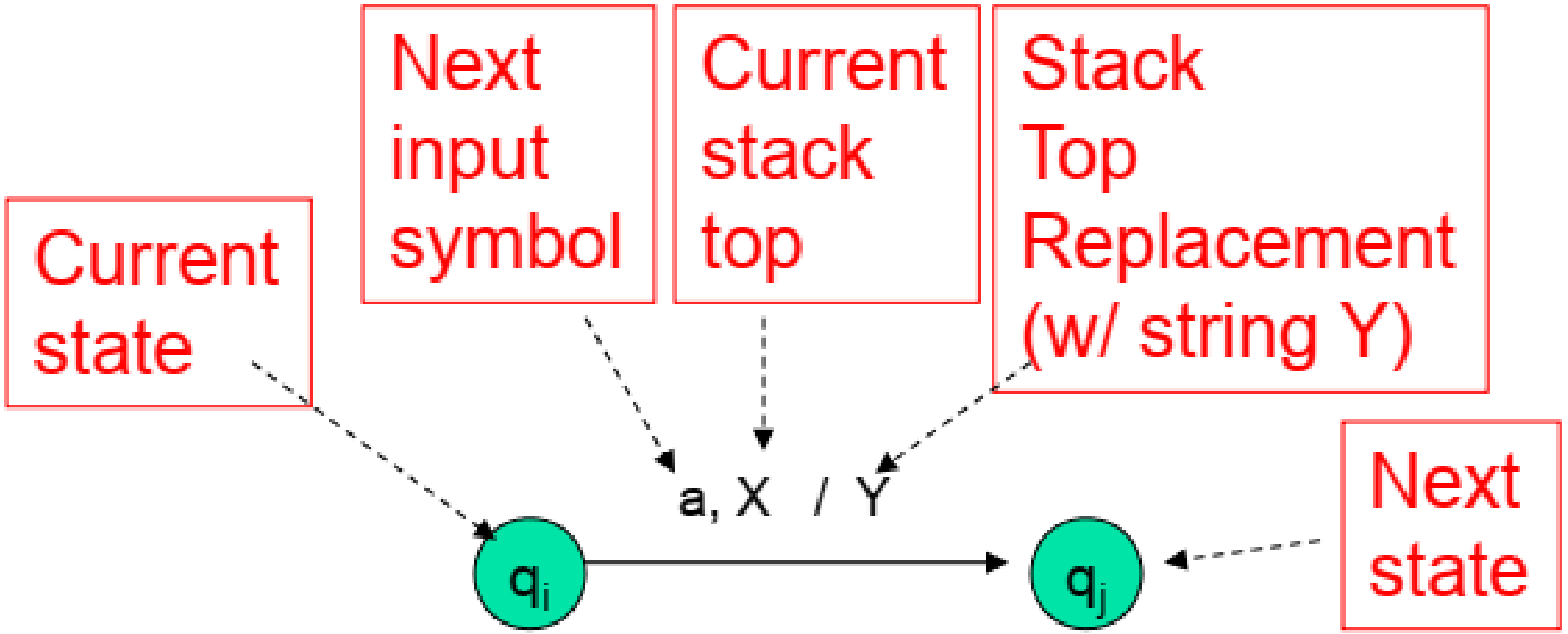


PDA:

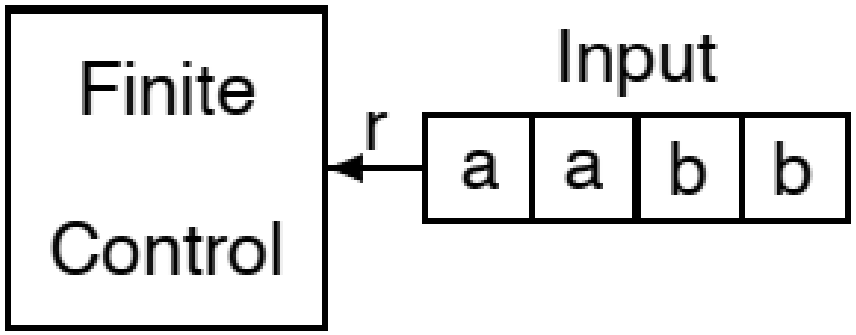



1
input
0 1 0 0
$
stack
2
input
0 1 0 0
$ 0
stack
3
input
0 1 0 0
$ 0 0
stack
4
input
0 1 0 0
$ 0
stack

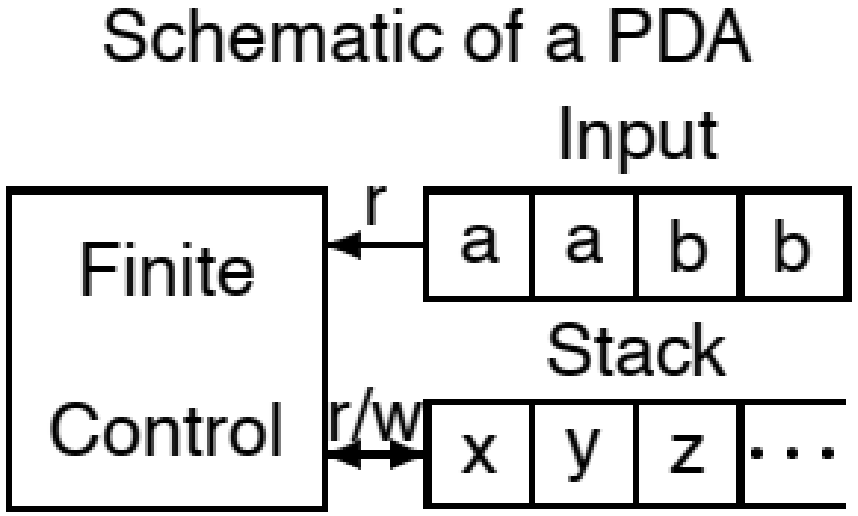
$\delta(q_i, a, X) = \{(q_j, Y)\}$

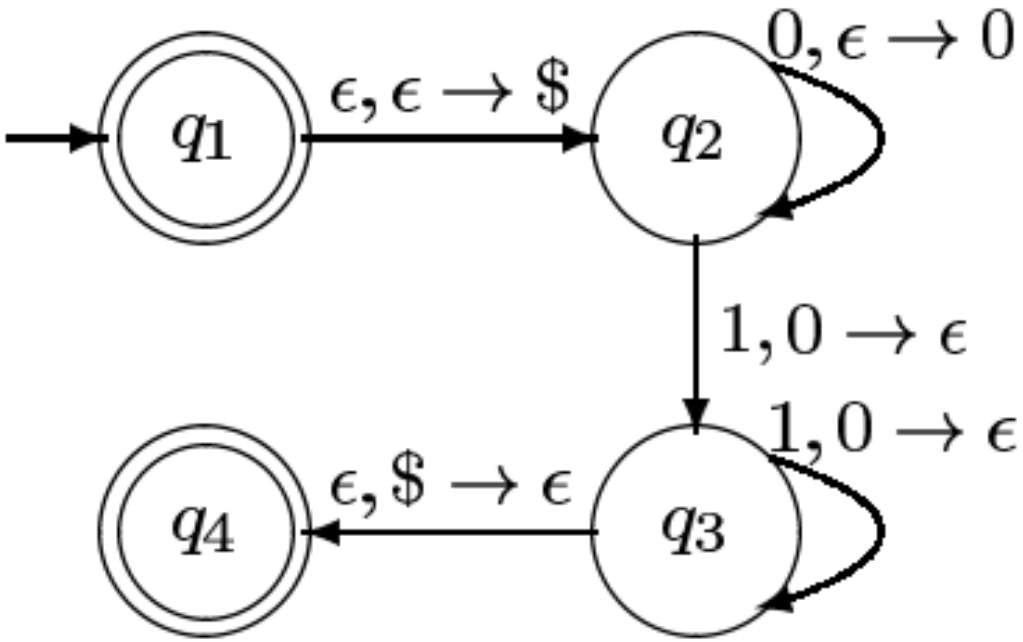


Schematic of a NFA



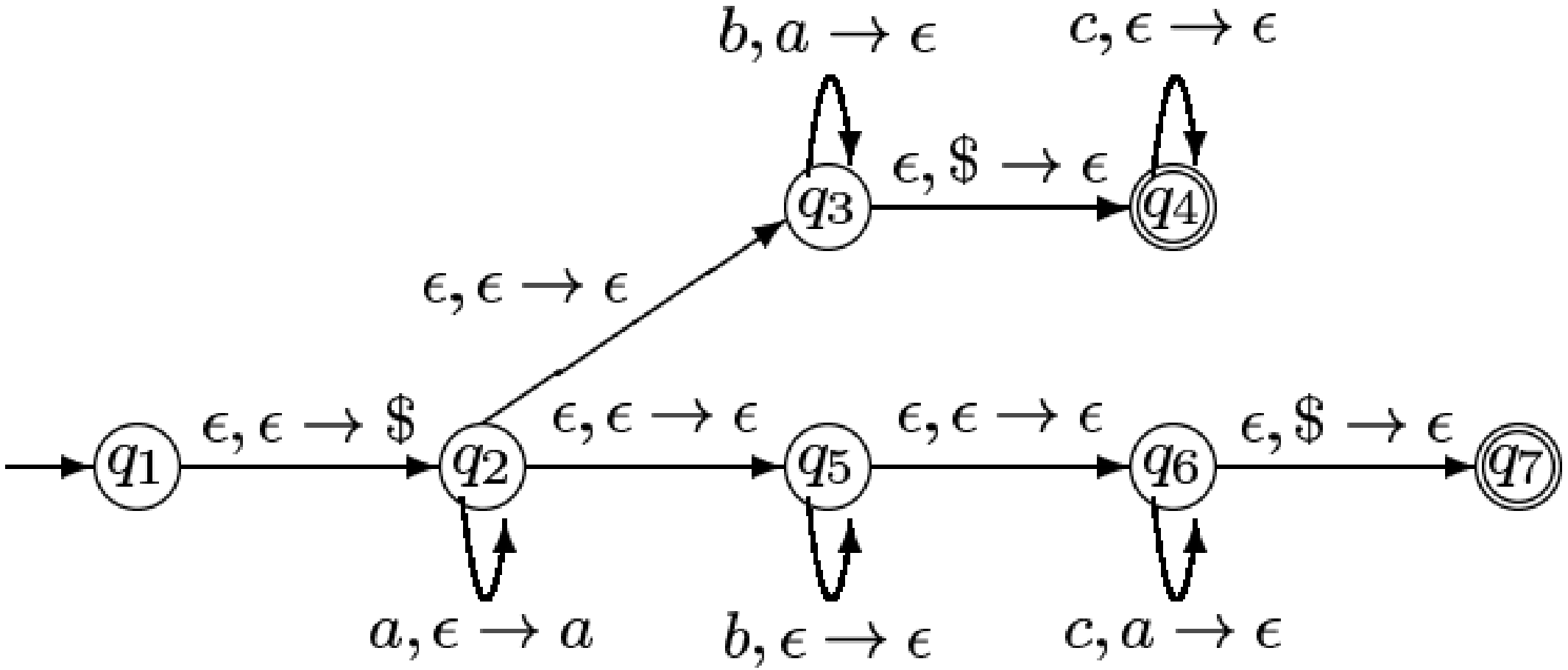
Schematic of a PDA



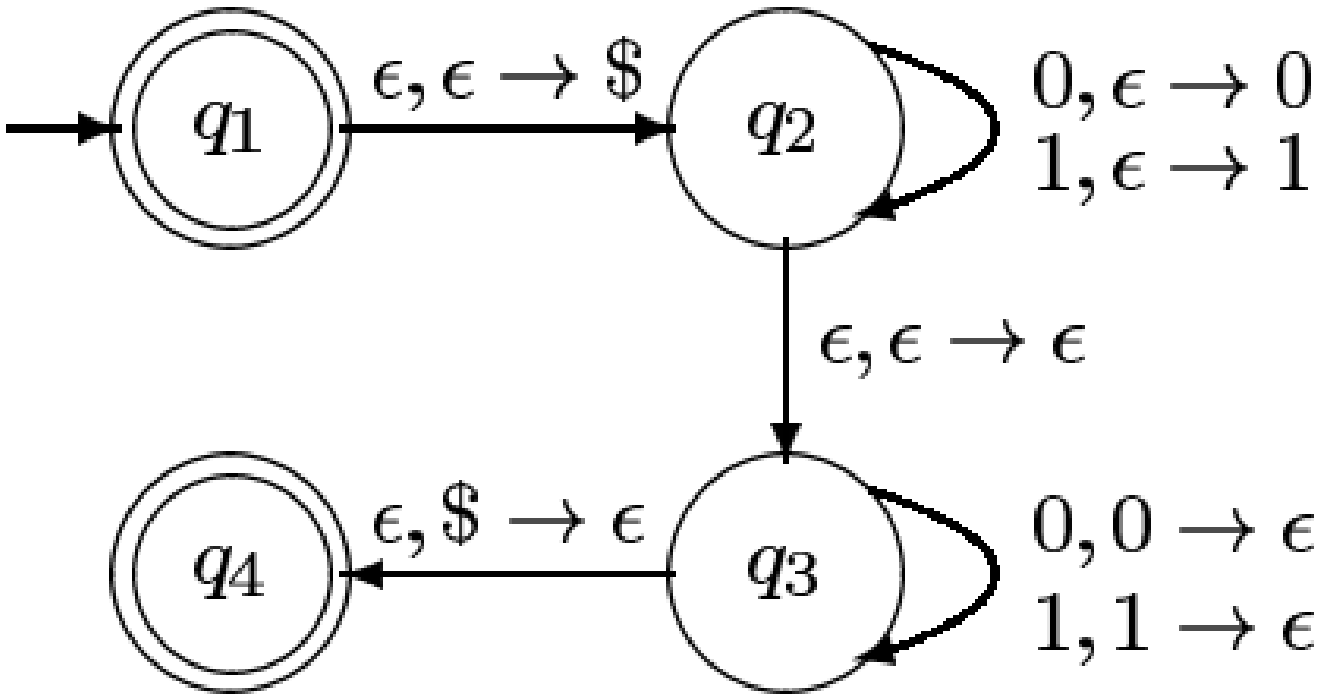


$$\{0^n 1^n | n \geq 0\}$$

$$\{a^i b^j c^k | i, j, k \geq 0 \wedge i = j \vee i = k\}$$



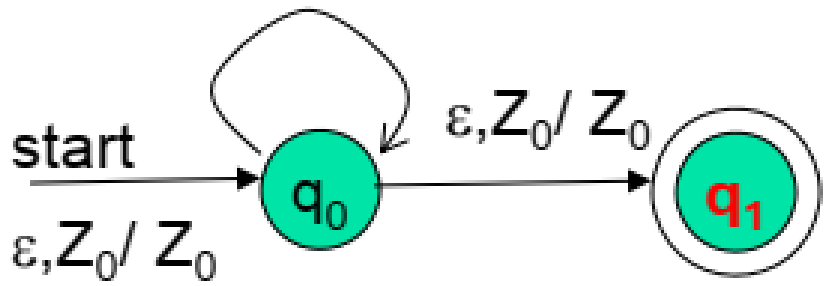
$$\{ww^{\mathcal{R}} | w \in \{0,1\}\}$$



PDA that accepts by final state

P_F:

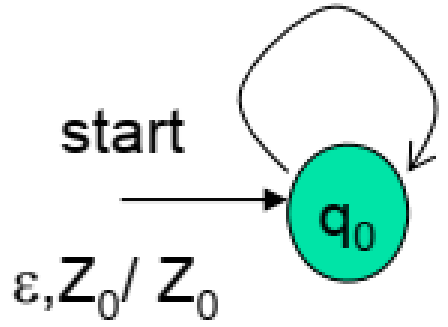
(,Z_0 / (Z_0
(,(/ ((
), (/ ε



An equivalent PDA that accepts by empty stack

P_N:

(,Z_0 / (Z_0
(, (/ ((
), (/ ε
ε,Z_0 / ε



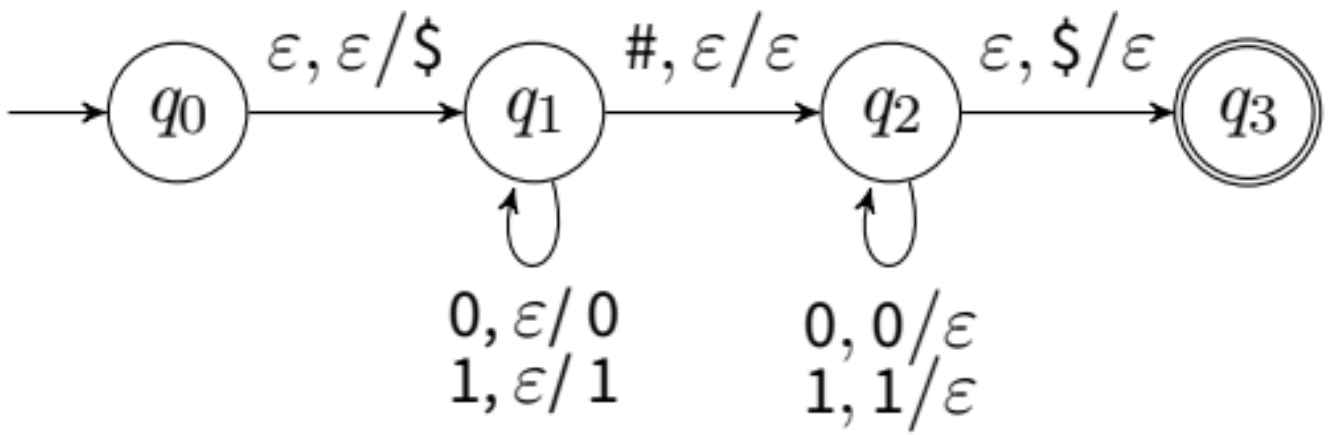
$$L = \{w\#w^R \mid w \in \{0,1\}^*\}$$

$$\Sigma = \{0, 1, \#\}$$

$$\Gamma = \{0, 1, \$\}$$

#, 0#0, 01#10 in L

ε, 01#1, 0##0 not in L



write w on stack

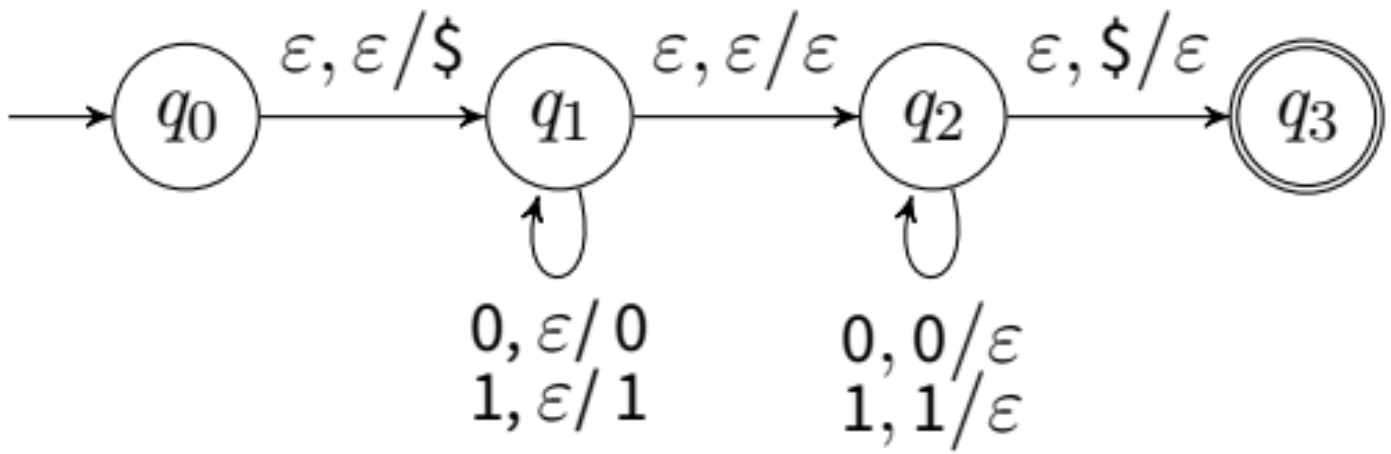
read w from stack

$$L = \{ww^R \mid w \in \Sigma^*\}$$

$\Sigma = \{0, 1\}$

ε, 00, 0110 in L

011, 010 not in L

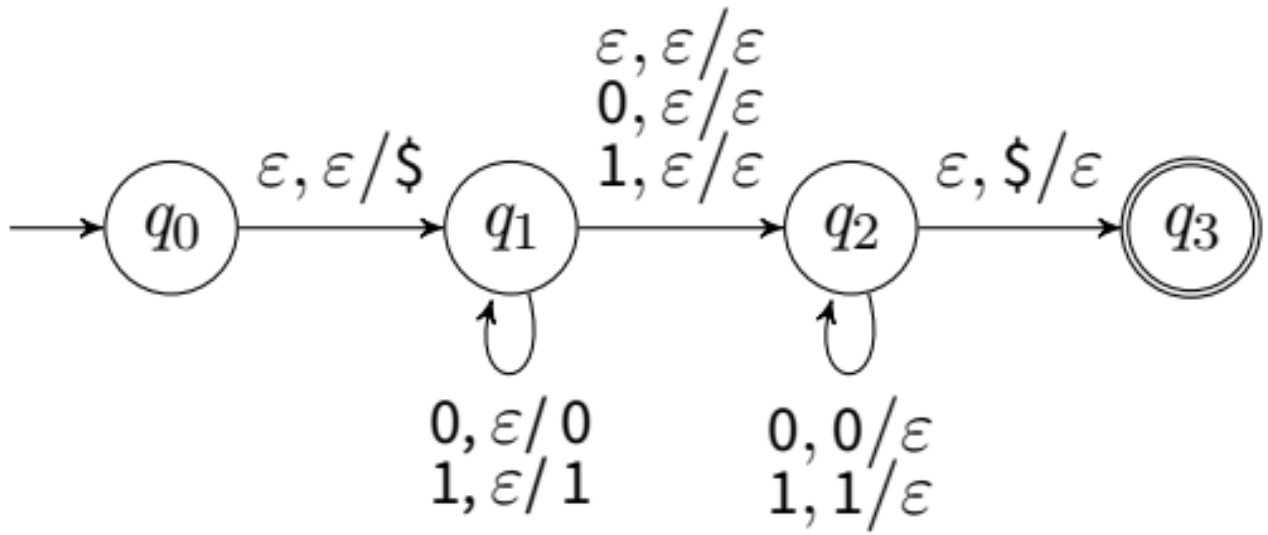


$$L = \{w \in \Sigma^* \mid w = w^R\}$$

$\Sigma = \{0, 1\}$

ε, 00, 010, 0110 in L

011 not in L

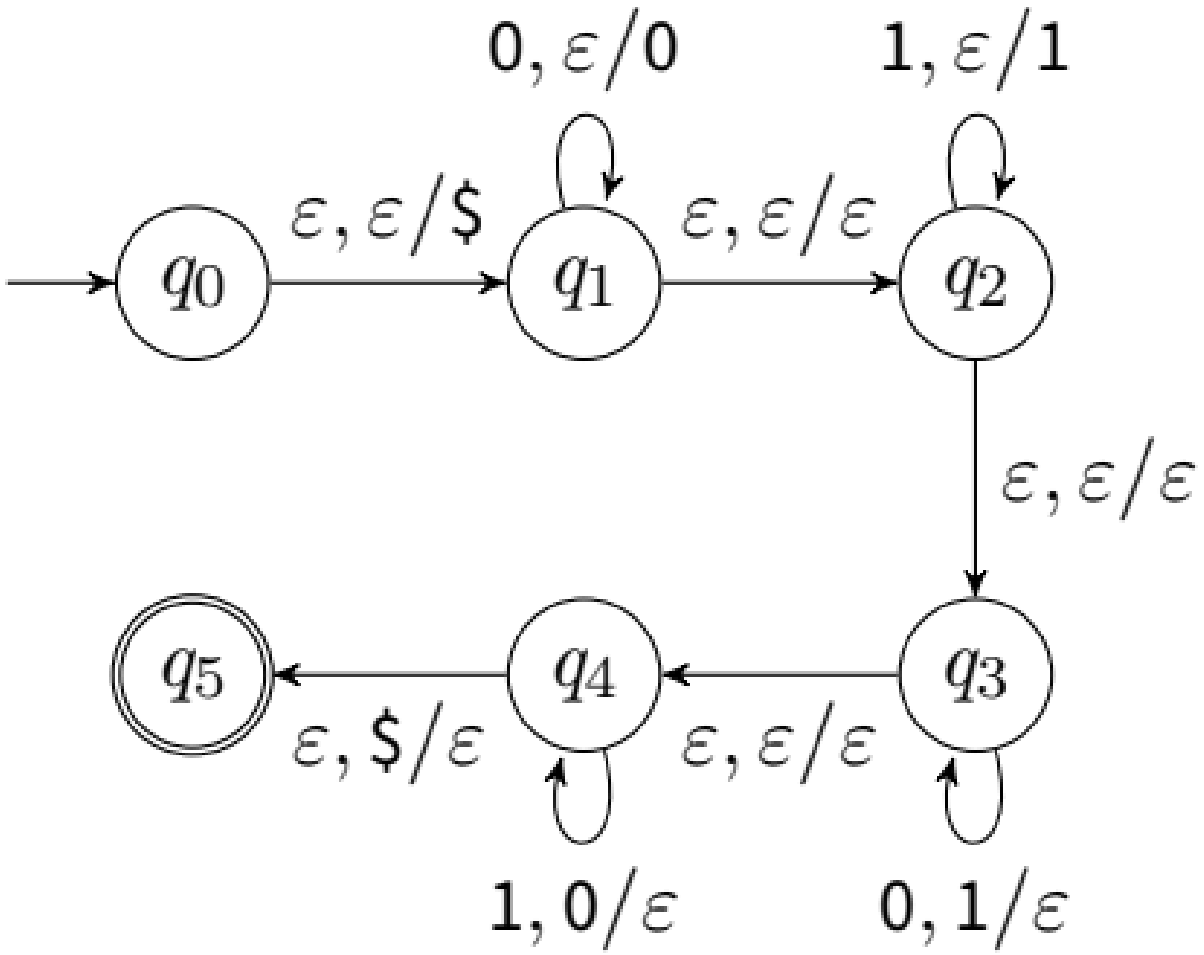


middle symbol can be ε, 0, or 1

$\underbrace{0010}_{x}\underbrace{0100}_{x^R}$ or $\underbrace{0010}_{x}1\underbrace{0100}_{x^R}$

$$L = \{0^n 1^m 0^m 1^n \mid n \geqslant 0, m \geqslant 0\}$$

$$\Sigma = \{0, 1\}$$



input: $0^n 1^m 0^m 1^n$

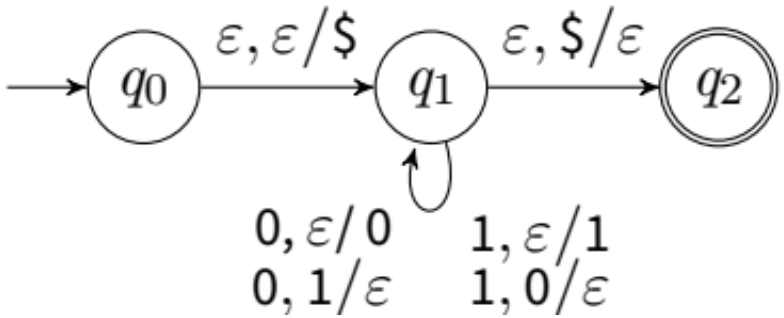
stack: $0^n 1^m$

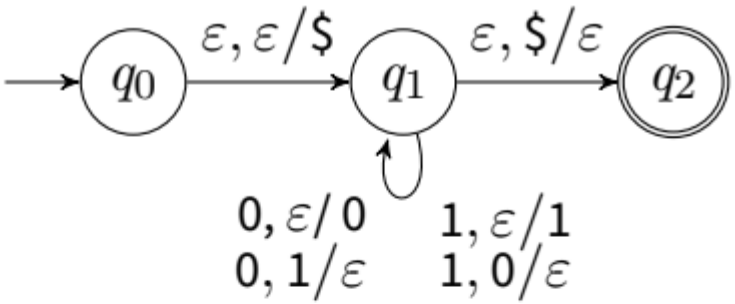
$L =$ same number of 0s and 1s

$\Sigma = \{0, 1\}$

Keep track of excess of 0s or 1s

If at the end, the stack is empty, number is equal





Example input: 001110