

PDAs Accept Context-Free Languages

Theorem:

$$\left\{ \begin{array}{c} \text{Context-Free} \\ \text{Languages} \\ \text{(Grammars)} \end{array} \right\} = \left\{ \begin{array}{c} \text{Languages} \\ \text{Accepted by} \\ \text{PDAs} \end{array} \right\}$$

Proof - Step 1:

$$\left\{ \begin{array}{c} \text{Context-Free} \\ \text{Languages} \\ \text{(Grammars)} \end{array} \right\} \subseteq \left\{ \begin{array}{c} \text{Languages} \\ \text{Accepted by} \\ \text{PDAs} \end{array} \right\}$$

Convert any context-free grammar G
to a PDA M with: $L(G) = L(M)$

Proof - Step 2:

$$\left\{ \begin{array}{c} \text{Context-Free} \\ \text{Languages} \\ \text{(Grammars)} \end{array} \right\} \supseteq \left\{ \begin{array}{c} \text{Languages} \\ \text{Accepted by} \\ \text{PDAs} \end{array} \right\}$$

Convert any PDA M to a context-free grammar G with: $L(G) = L(M)$

Proof - step 1

Convert

Context-Free Grammars
to
PDAs

Take an arbitrary context-free grammar G

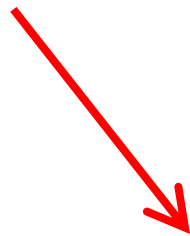
We will convert G to a PDA M such that:

$$L(G) = L(M)$$

Conversion Procedure:

For each
production in G

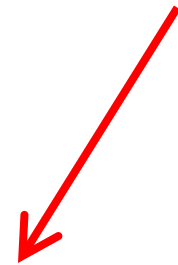
$$A \rightarrow w$$



$$\varepsilon, A \rightarrow w$$

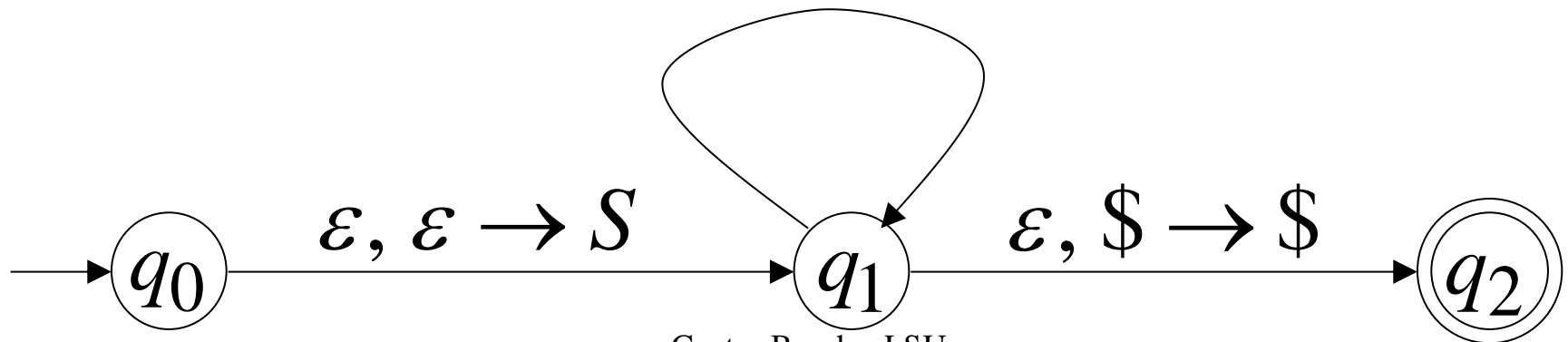
For each
terminal in G

a



$$a, a \rightarrow \varepsilon$$

Add transitions



Example

Grammar

$$S \rightarrow aSTb$$

$$S \rightarrow b$$

$$T \rightarrow Ta$$

$$T \rightarrow \varepsilon$$

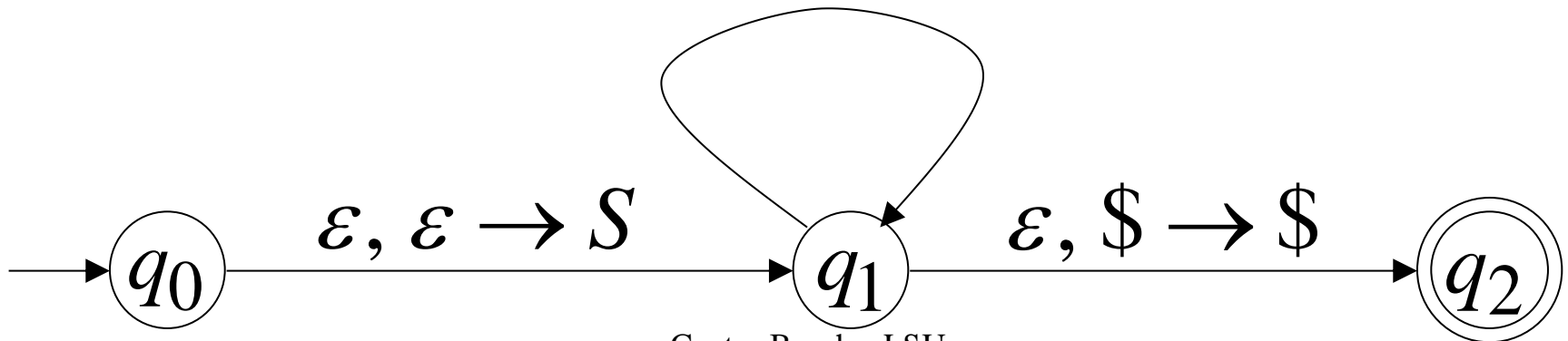
PDA

$$\varepsilon, S \rightarrow aSTb$$

$$\varepsilon, S \rightarrow b$$

$$\varepsilon, T \rightarrow Ta \quad a, a \rightarrow \varepsilon$$

$$\varepsilon, T \rightarrow \varepsilon \quad b, b \rightarrow \varepsilon$$



PDA simulates leftmost derivations

Grammar

Leftmost Derivation

S
 $\Rightarrow \dots$
 $\Rightarrow \sigma_1 \cdots \sigma_k X_1 \cdots X_m$
 $\Rightarrow \dots$
 $\Rightarrow \sigma_1 \cdots \sigma_k \sigma_{k+1} \cdots \sigma_n$

PDA Computation

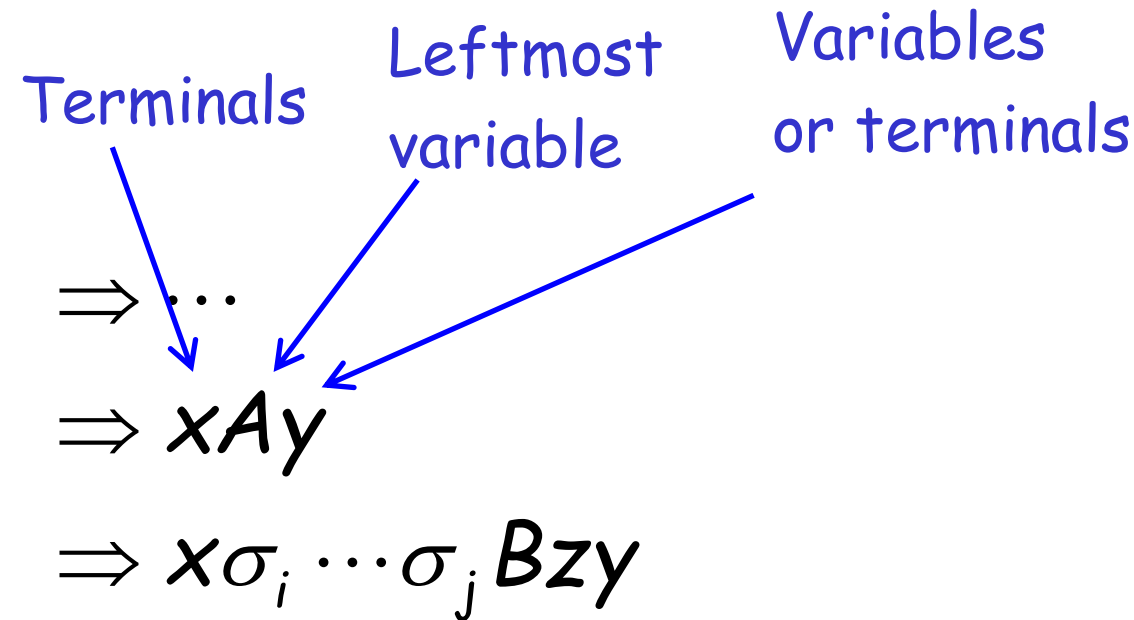
$(q_0, \sigma_1 \cdots \sigma_k \sigma_{k+1} \cdots \sigma_n, \$)$
 $\succ (q_1, \sigma_1 \cdots \sigma_k \sigma_{k+1} \cdots \sigma_n, S\$)$
 $\succ \dots$
 $\succ (q_1, \sigma_{k+1} \cdots \sigma_n, X_1 \cdots X_m \$)$
 $\succ \dots$
 $\succ (q_2, \varepsilon, \$)$

Scanned
symbols

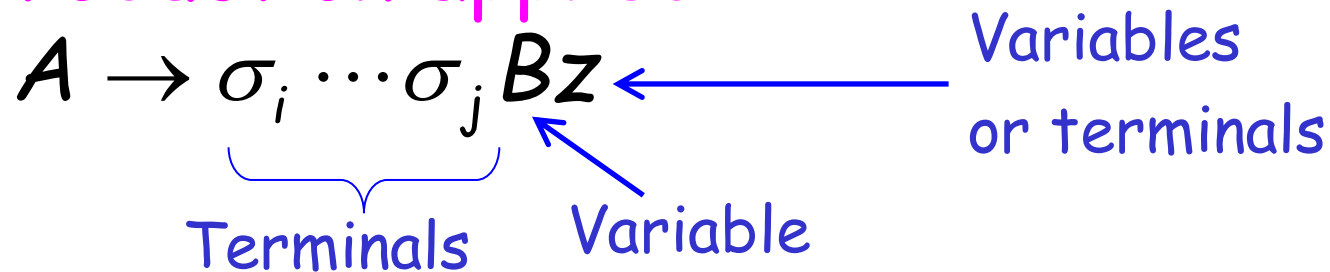
Stack
contents

Grammar

Leftmost Derivation



Production applied



Grammar

Leftmost Derivation

$\Rightarrow \dots$

$\Rightarrow xAy$

$\Rightarrow x\sigma_i \cdots \sigma_j Bzy$

Production applied

$A \rightarrow \sigma_i \cdots \sigma_j Bz$

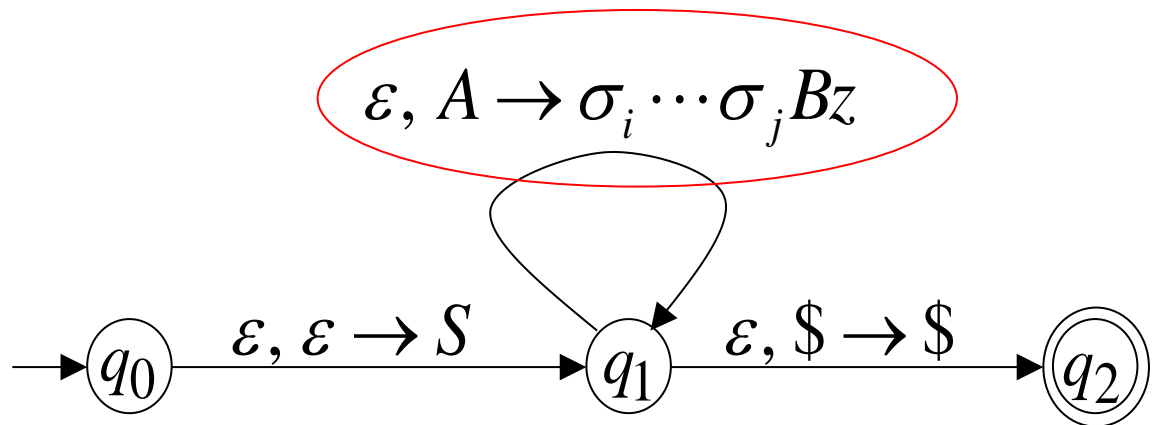
PDA Computation

$\succ \dots$

$\succ (q_1, \sigma_i \cdots \sigma_n, Ay\$)$

$\succ (q_1, \sigma_i \cdots \sigma_n, \sigma_i \cdots \sigma_j Bzy\$)$

Transition applied



Grammar

Leftmost Derivation

$\Rightarrow \dots$

$\Rightarrow xAy$

$\Rightarrow x\sigma_i \cdots \sigma_j Bzy$

PDA Computation

$\succ \dots$

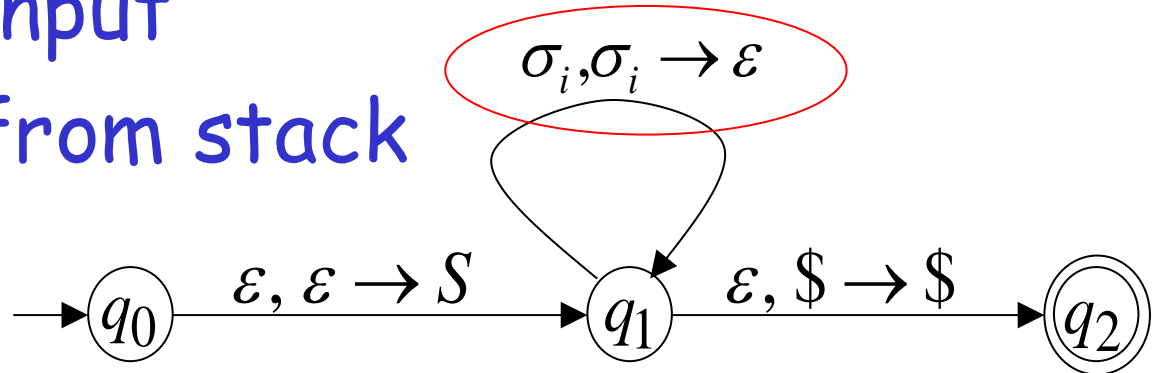
$\succ (q_1, \sigma_i \cdots \sigma_n, Ay\$)$

$\succ (q_1, \sigma_i \cdots \sigma_n, \sigma_i \cdots \sigma_j Bzy\$)$

$\succ (q_1, \sigma_{i+1} \cdots \sigma_n, \sigma_{i+1} \cdots \sigma_j Bzy\$)$

Read σ_i from input
and remove it from stack

Transition applied



Grammar

Leftmost Derivation

$\Rightarrow \dots$

$\Rightarrow xAy$

$\Rightarrow x\sigma_i \cdots \sigma_j Bzy$

All symbols $\sigma_i \cdots \sigma_j$
have been removed
from top of stack

PDA Computation

$\succ \dots$

$\succ (q_1, \sigma_i \cdots \sigma_n, Ay\$)$

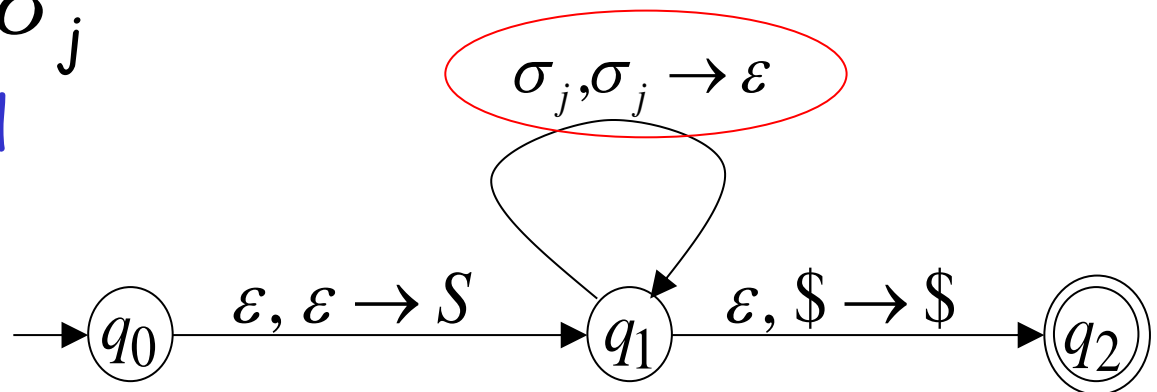
$\succ (q_1, \sigma_i \cdots \sigma_n, \sigma_i \cdots \sigma_j Bzy\$)$

$\succ (q_1, \sigma_{i+1} \cdots \sigma_n, \sigma_{i+1} \cdots \sigma_j Bzy\$)$

$\succ \dots$

$\succ (q_1, \sigma_{j+1} \cdots \sigma_n, Bzy\$)$

Last Transition applied



The process repeats with the next
leftmost variable

$\Rightarrow \dots$

$\Rightarrow xAy$

$\Rightarrow x\sigma_i \cdots \sigma_j Bzy$

$\Rightarrow x\sigma_i \cdots \sigma_j \sigma_{j+1} \cdots \sigma_k Cpzy$

$\succ \dots$

$\succ (q_1, \sigma_{j+1} \cdots \sigma_n, Bzy\$)$

$\succ (q_1, \sigma_{j+1} \cdots \sigma_n, \sigma_{j+1} \cdots \sigma_k Cpzy\$)$

$\succ \dots$

$\succ (q_1, \sigma_{k+1} \cdots \sigma_n, Cpzy\$)$

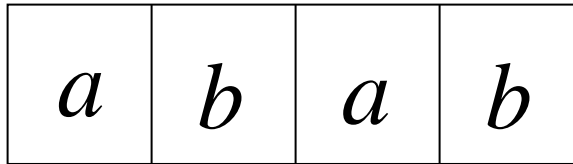
Production applied

$B \rightarrow \sigma_{j+1} \cdots \sigma_k Cp$

And so on.....

Example:

Input



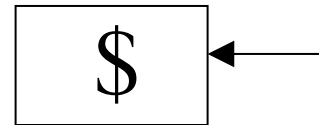
Time 0

$$\varepsilon, S \rightarrow aSTb$$

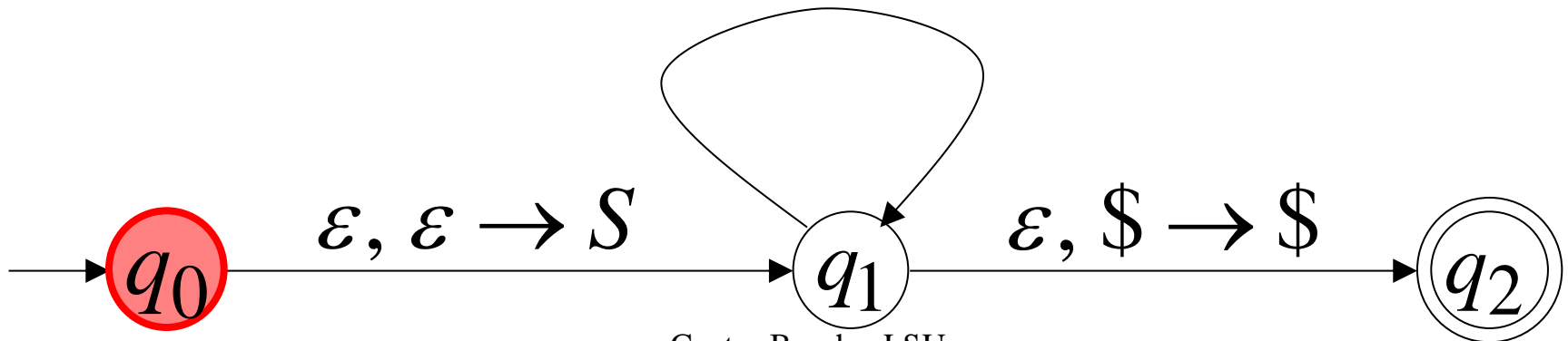
$$\varepsilon, S \rightarrow b$$

$$\varepsilon, T \rightarrow Ta \quad a, a \rightarrow \varepsilon$$

$$\varepsilon, T \rightarrow \varepsilon \quad b, b \rightarrow \varepsilon$$

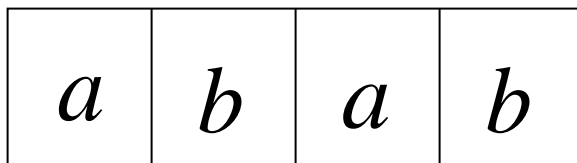


Stack



Derivation: S

Input



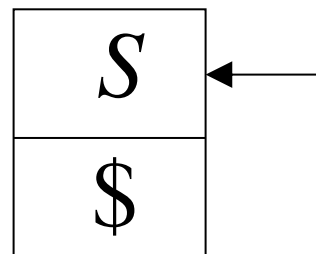
Time 1

$$\varepsilon, S \rightarrow aSTb$$

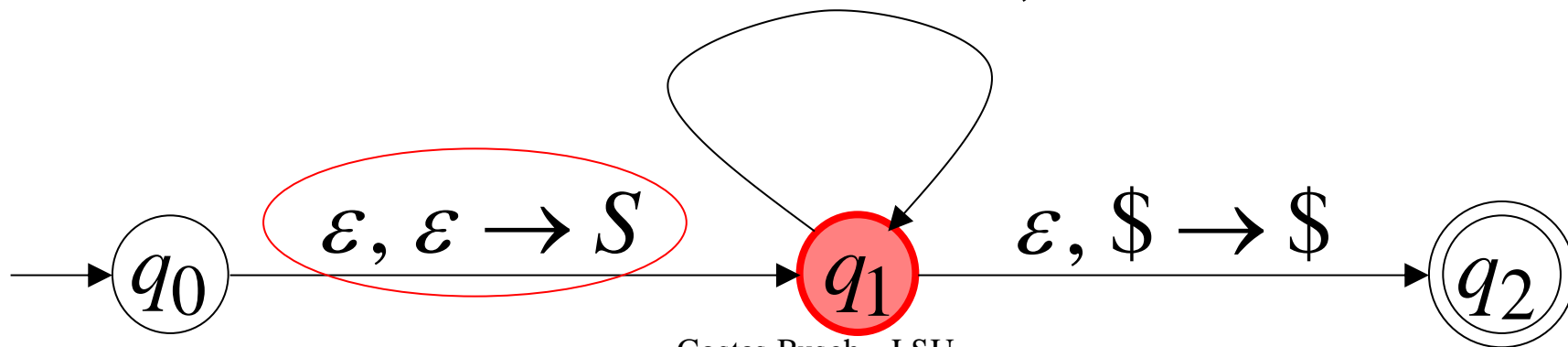
$$\varepsilon, S \rightarrow b$$

$$\varepsilon, T \rightarrow Ta \quad a, a \rightarrow \varepsilon$$

$$\varepsilon, T \rightarrow \varepsilon \quad b, b \rightarrow \varepsilon$$

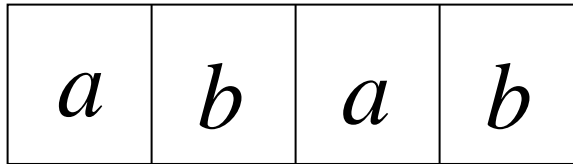


Stack



Derivation: $S \Rightarrow aSTb$

Input



$a b a b$

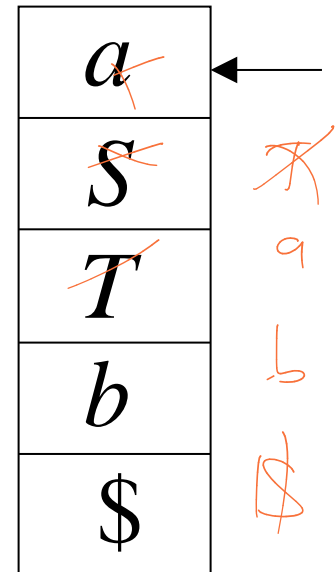
Time 2

$\varepsilon, S \rightarrow aSTb$

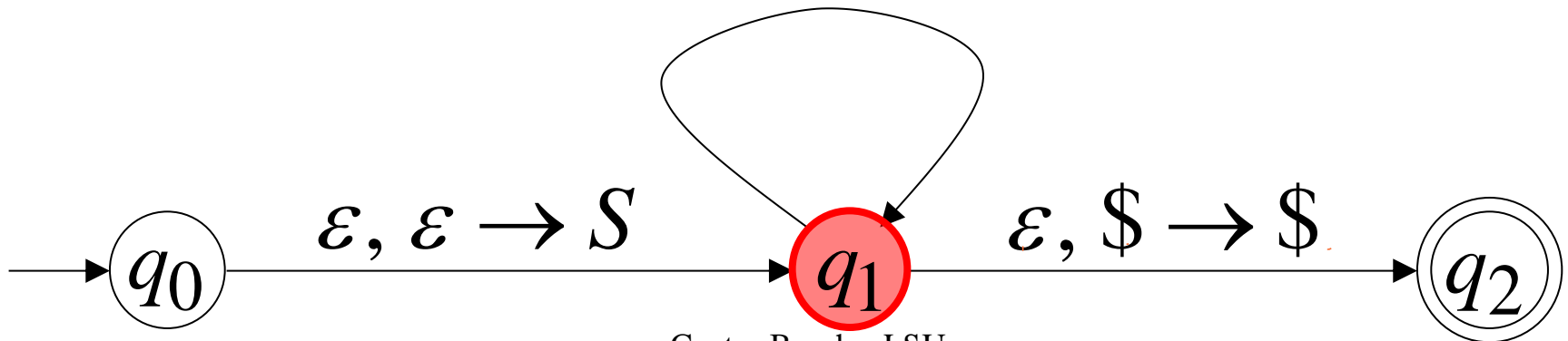
$\varepsilon, S \rightarrow b$

$\varepsilon, T \rightarrow Ta \quad a, a \rightarrow \varepsilon$

$\varepsilon, T \rightarrow \varepsilon \quad b, b \rightarrow \varepsilon$

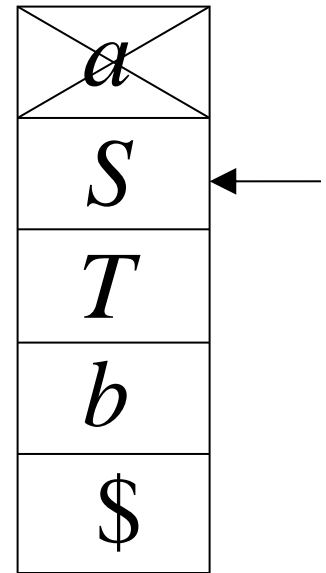
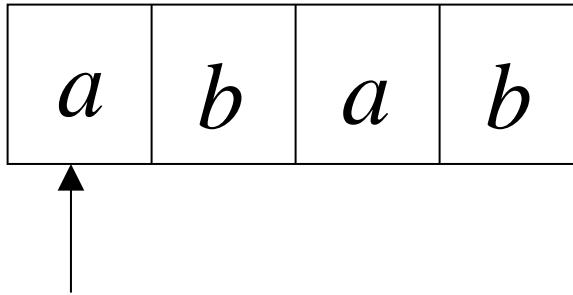


Stack



Derivation: $S \Rightarrow aSTb$

Input



Time 3

$\varepsilon, S \rightarrow aSTb$

$\varepsilon, S \rightarrow b$

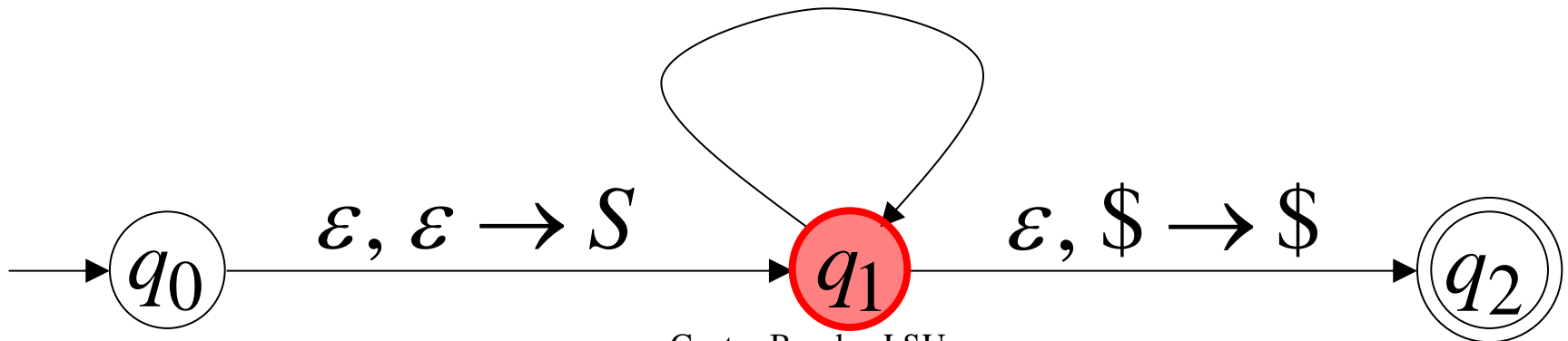
$\varepsilon, T \rightarrow Ta$

$a, a \rightarrow \varepsilon$

$\varepsilon, T \rightarrow \varepsilon$

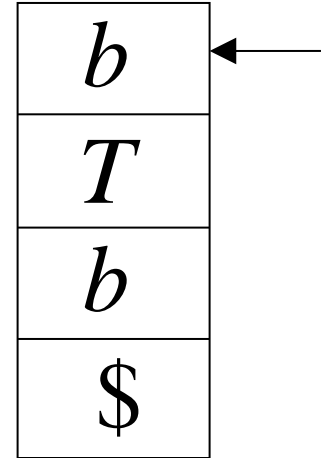
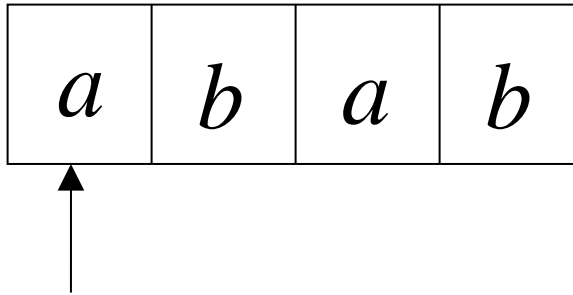
$b, b \rightarrow \varepsilon$

Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb$

Input



Time 4

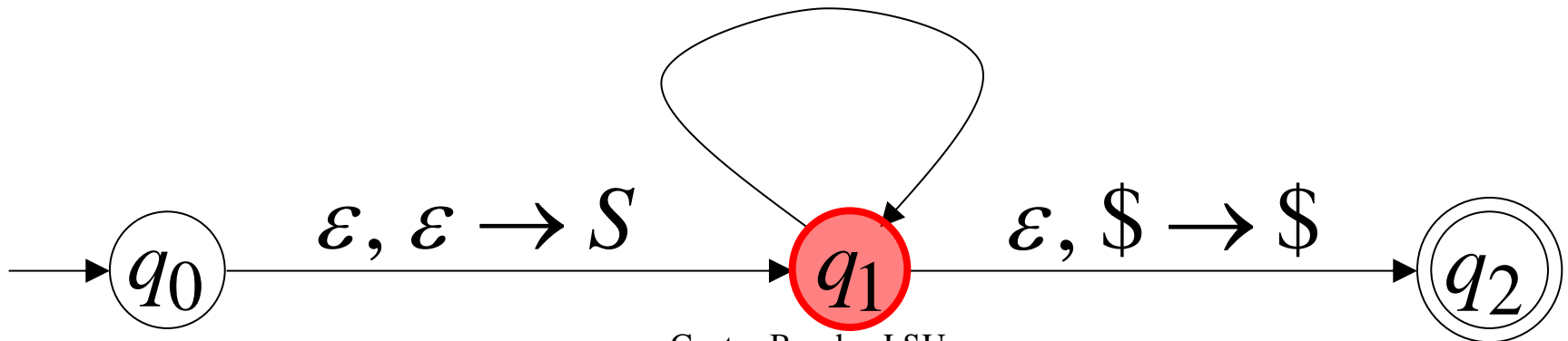
$\varepsilon, S \rightarrow aSTb$

$\varepsilon, S \rightarrow b$

$\varepsilon, T \rightarrow Ta \quad a, a \rightarrow \varepsilon$

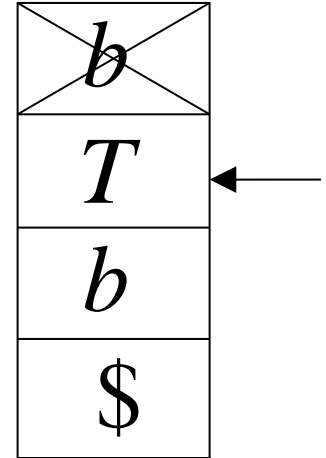
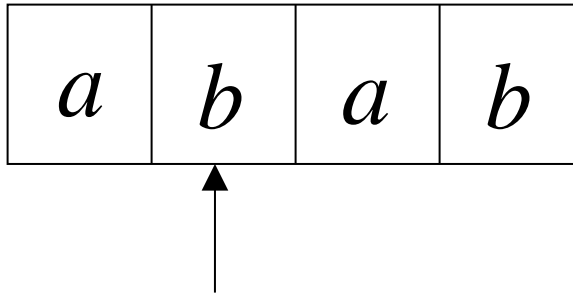
$\varepsilon, T \rightarrow \varepsilon \quad b, b \rightarrow \varepsilon$

Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb$

Input



Time 5

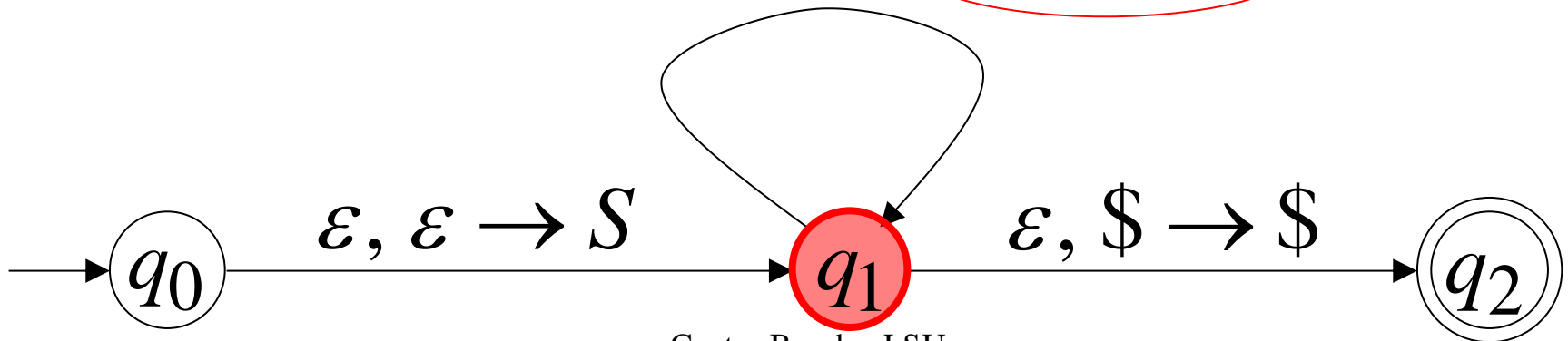
$\varepsilon, S \rightarrow aSTb$

$\varepsilon, S \rightarrow b$

$\varepsilon, T \rightarrow Ta$ $a, a \rightarrow \varepsilon$

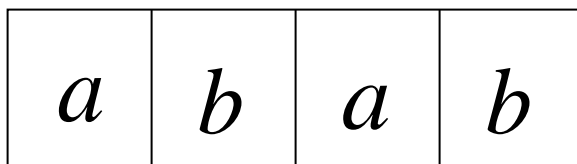
$\varepsilon, T \rightarrow \varepsilon$ $b, b \rightarrow \varepsilon$

Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb \Rightarrow abTab$

Input

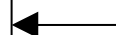
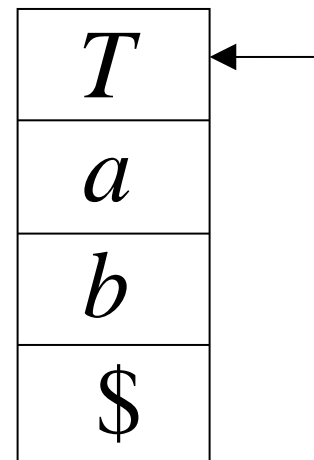


$\varepsilon, S \rightarrow aSTb$

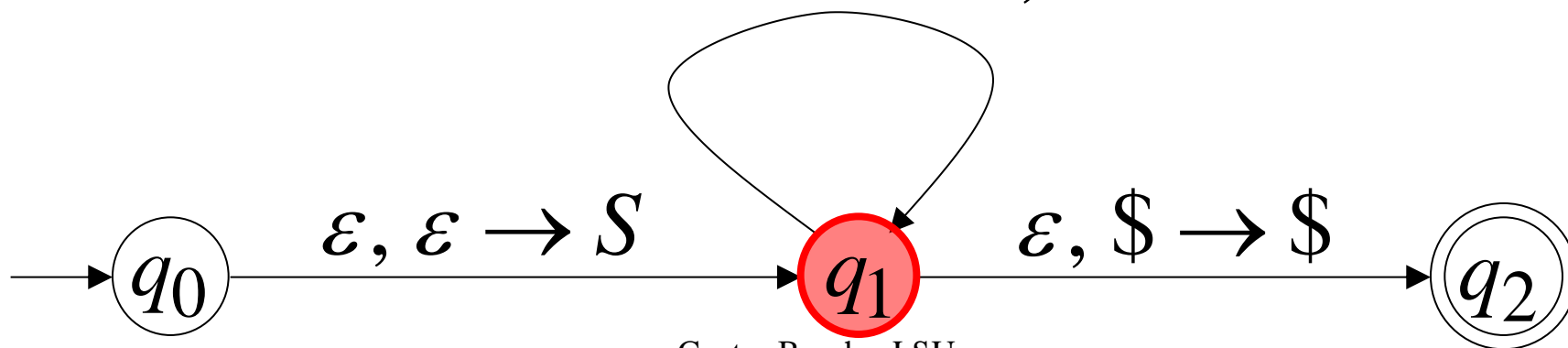
$\varepsilon, S \rightarrow b$

$\varepsilon, T \rightarrow Ta$ $a, a \rightarrow \varepsilon$

$\varepsilon, T \rightarrow \varepsilon$ $b, b \rightarrow \varepsilon$

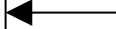
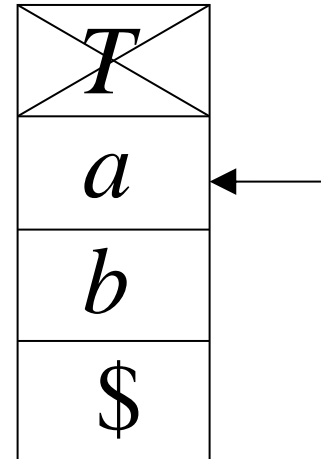
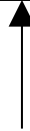
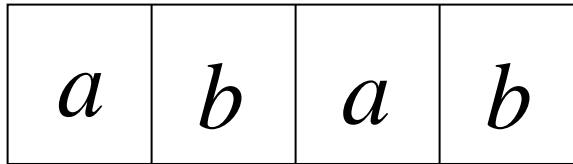


Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb \Rightarrow abTab \Rightarrow abab$

Input



Time 7

$\varepsilon, S \rightarrow aSTb$

$\varepsilon, S \rightarrow b$

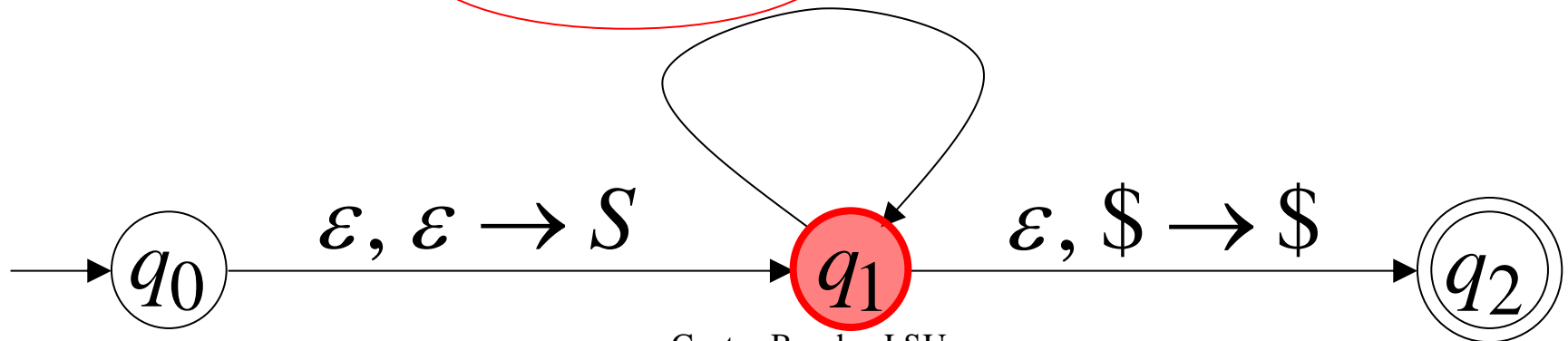
$\varepsilon, T \rightarrow Ta$

$a, a \rightarrow \varepsilon$

$\varepsilon, T \rightarrow \varepsilon$

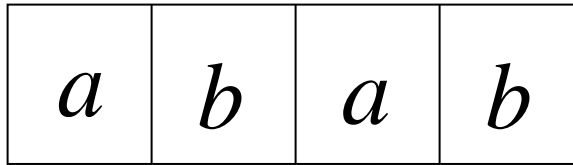
$b, b \rightarrow \varepsilon$

Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb \Rightarrow abTab \Rightarrow abab$

Input



Time 8

$\varepsilon, S \rightarrow aSTb$

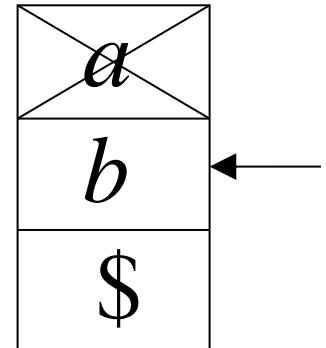
$\varepsilon, S \rightarrow b$

$\varepsilon, T \rightarrow Ta$

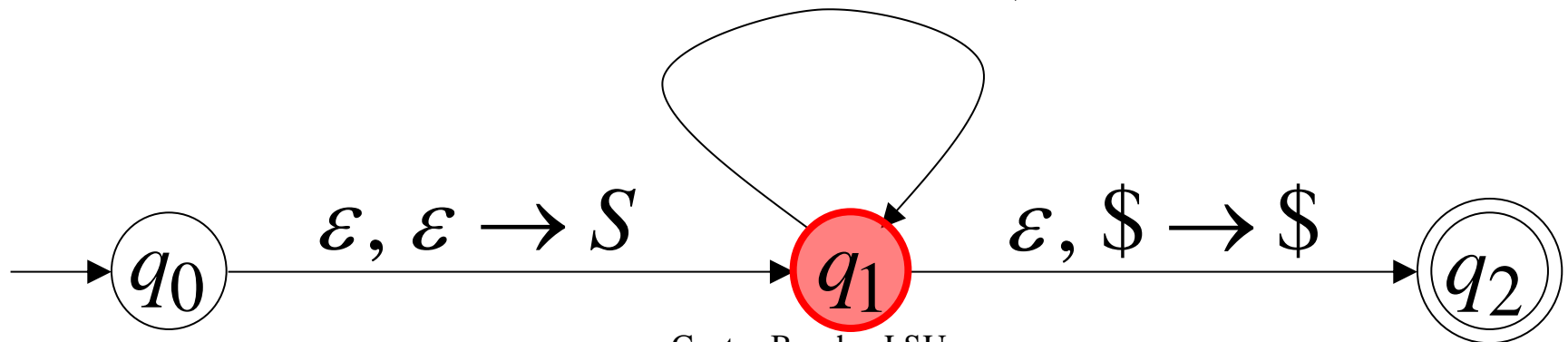
$a, a \rightarrow \varepsilon$

$\varepsilon, T \rightarrow \varepsilon$

$b, b \rightarrow \varepsilon$

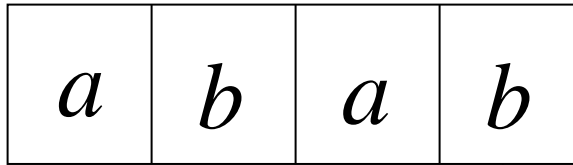


Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb \Rightarrow abTab \Rightarrow abab$

Input



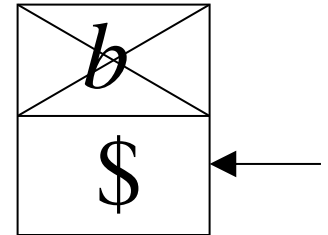
Time 9

$\varepsilon, S \rightarrow aSTb$

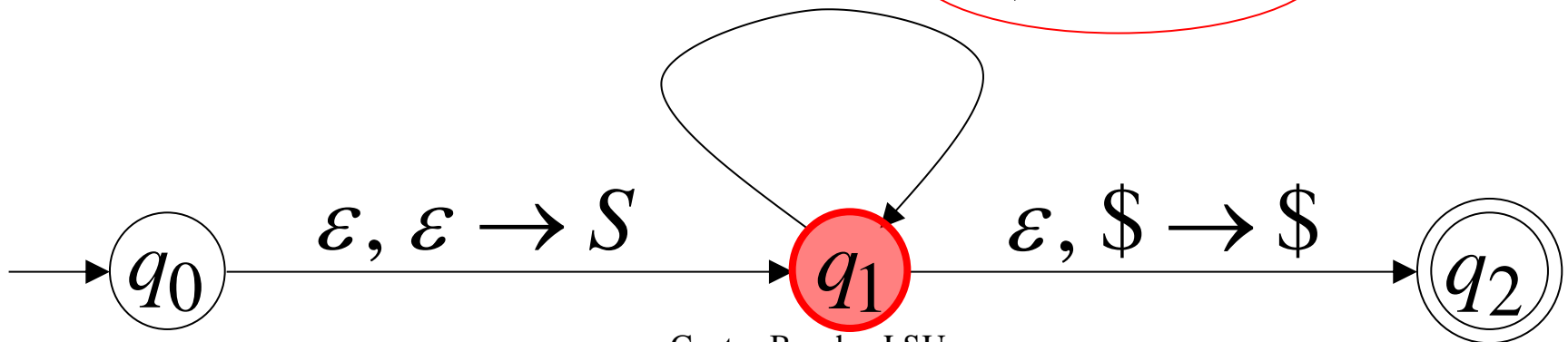
$\varepsilon, S \rightarrow b$

$\varepsilon, T \rightarrow Ta \quad a, a \rightarrow \varepsilon$

$\varepsilon, T \rightarrow \varepsilon \quad b, b \rightarrow \varepsilon$

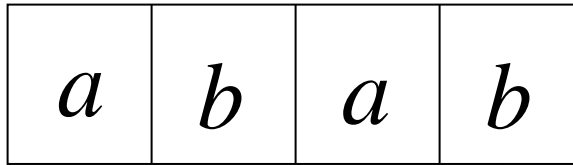


Stack



Derivation: $S \Rightarrow aSTb \Rightarrow abTb \Rightarrow abTab \Rightarrow abab$

Input



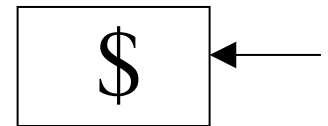
Time 10

$\varepsilon, S \rightarrow aSTb$

$\varepsilon, S \rightarrow b$

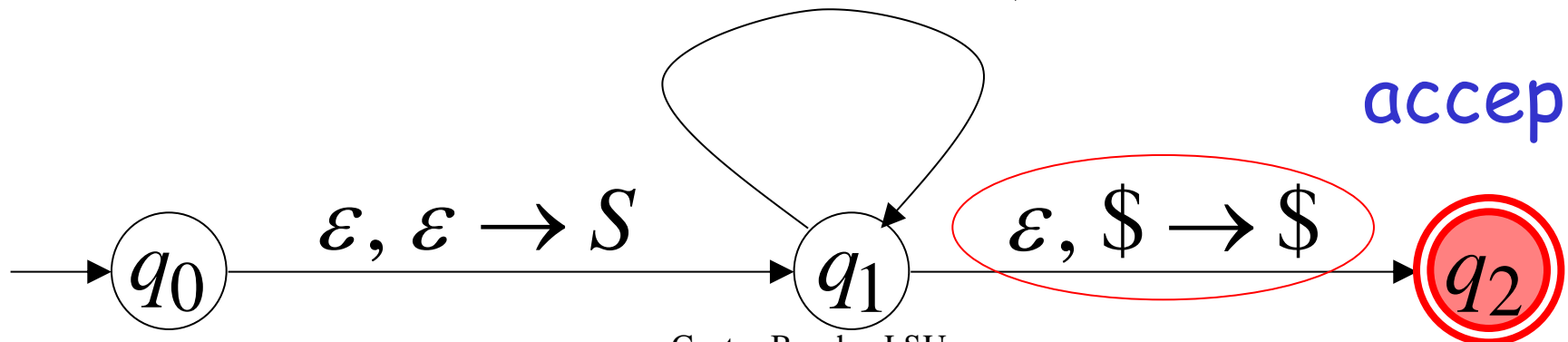
$\varepsilon, T \rightarrow Ta \quad a, a \rightarrow \varepsilon$

$\varepsilon, T \rightarrow \varepsilon \quad b, b \rightarrow \varepsilon$



Stack

accept



Grammar

Leftmost Derivation

S

$\Rightarrow aSTb$

$\Rightarrow abTb$

$\Rightarrow abTab$

$\Rightarrow abab$

PDA Computation

$(q_0, abab, \$)$

$\succ (q_1, abab, S\$)$

$\succ (q_1, bab, STb\$)$

$\succ (q_1, bab, bTb\$)$

$\succ (q_1, ab, Tb\$)$

$\succ (q_1, ab, Tab\$)$

$\succ (q_1, ab, ab\$)$

$\succ (q_1, b, b\$)$

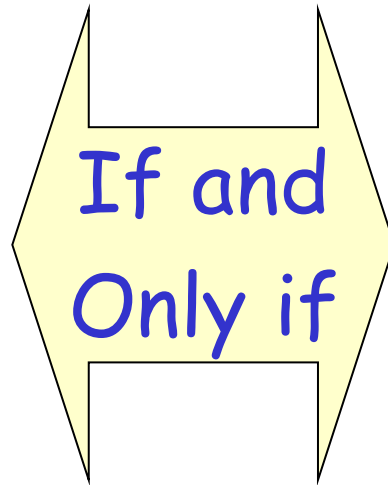
$\succ (q_1, \varepsilon, \$)$

$\succ (q_2, \varepsilon, \$)$

In general, it can be shown that:

Grammar G
generates
string w

$$S \xRightarrow{*} w$$



PDA M
accepts w

$$(q_0, w, \$) \xRightarrow{*} (q_2, \varepsilon, \$)$$

Therefore $L(G) = L(M)$

Proof - step 2

Convert

PDAs

to

Context-Free Grammars

Take an arbitrary PDA M

We will convert M
to a context-free grammar G such that:

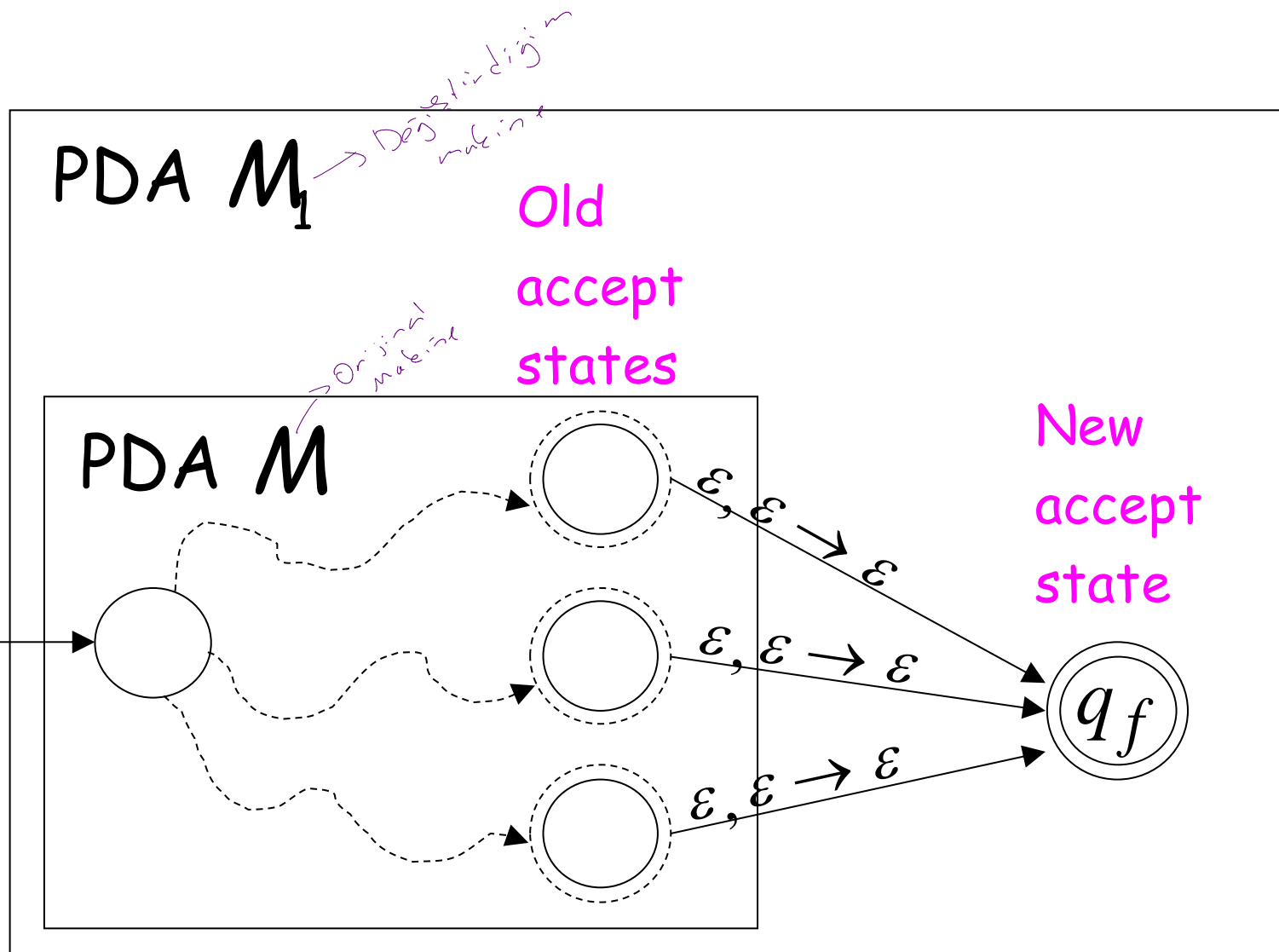
$$L(M) = L(G)$$

First modify PDA M so that:

Code
on the
left

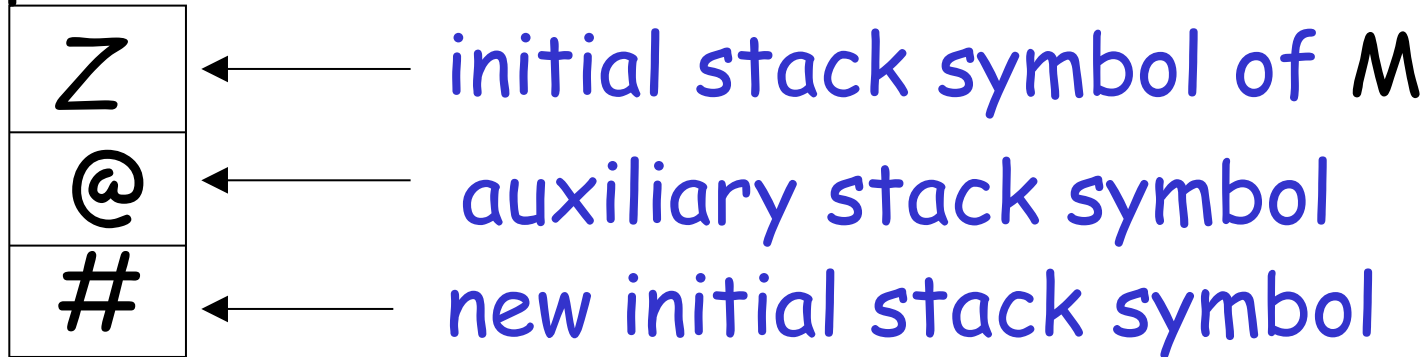
1. The PDA has a single accept state
2. Use new initial stack symbol $\#$
3. On acceptance the stack contains only stack symbol $\#$ (this symbol is not used in any transition)
4. Each transition either pushes a symbol or pops a symbol but not both together

1. The PDA has a single accept state



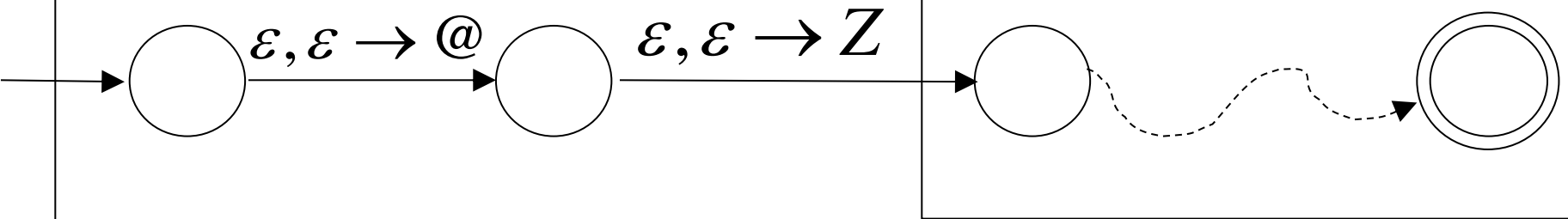
2. Use new initial stack symbol

Top of stack



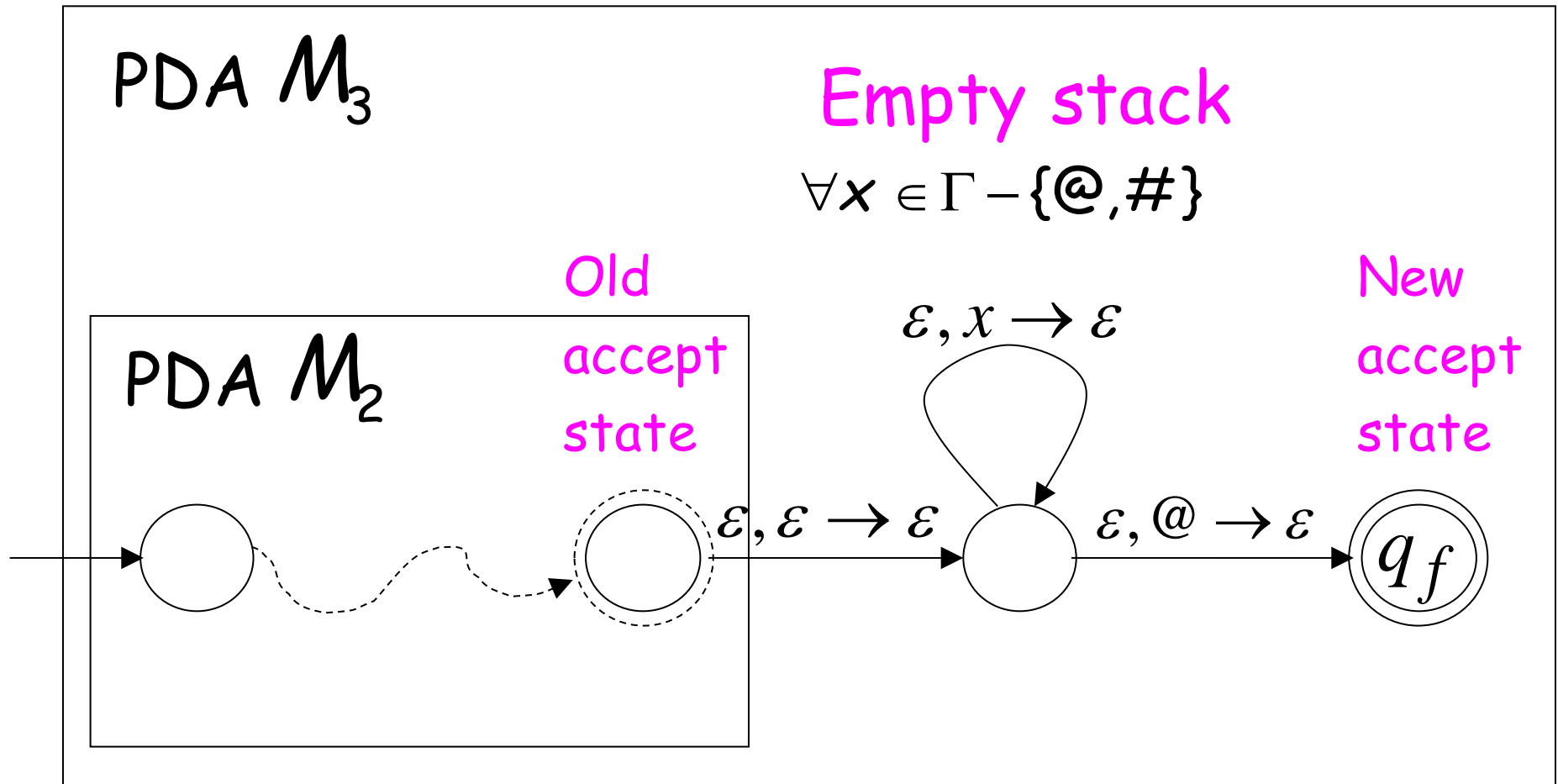
PDA M_2

PDA M_1

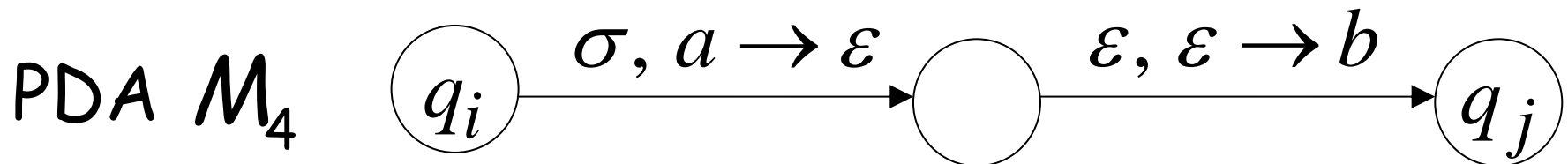
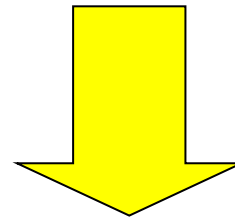
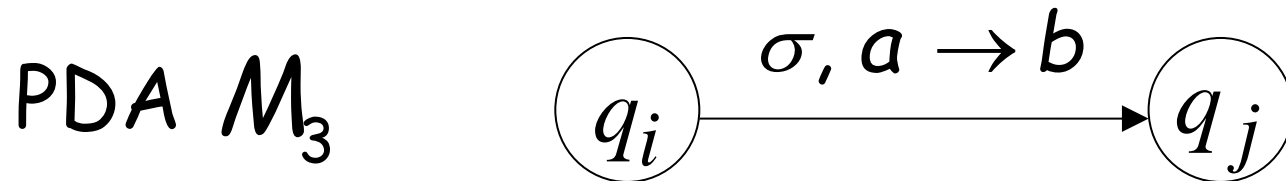


M_1 still thinks that Z is the initial stack

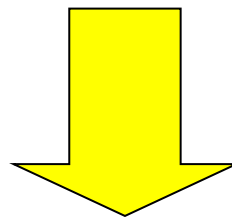
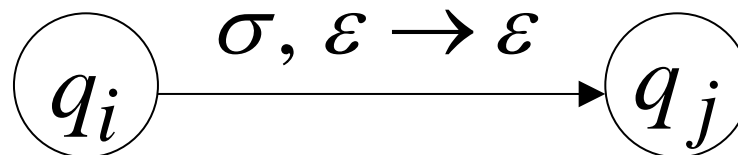
3. On acceptance the stack contains only stack symbol #
(this symbol is not used in any transition)



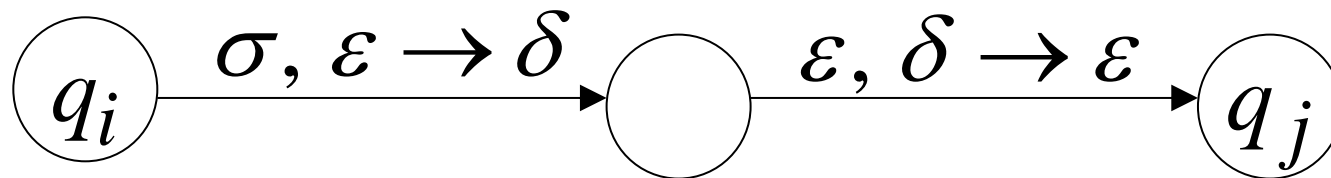
4. Each transition either pushes a symbol or pops a symbol but not both together



PDA M_3



PDA M_4

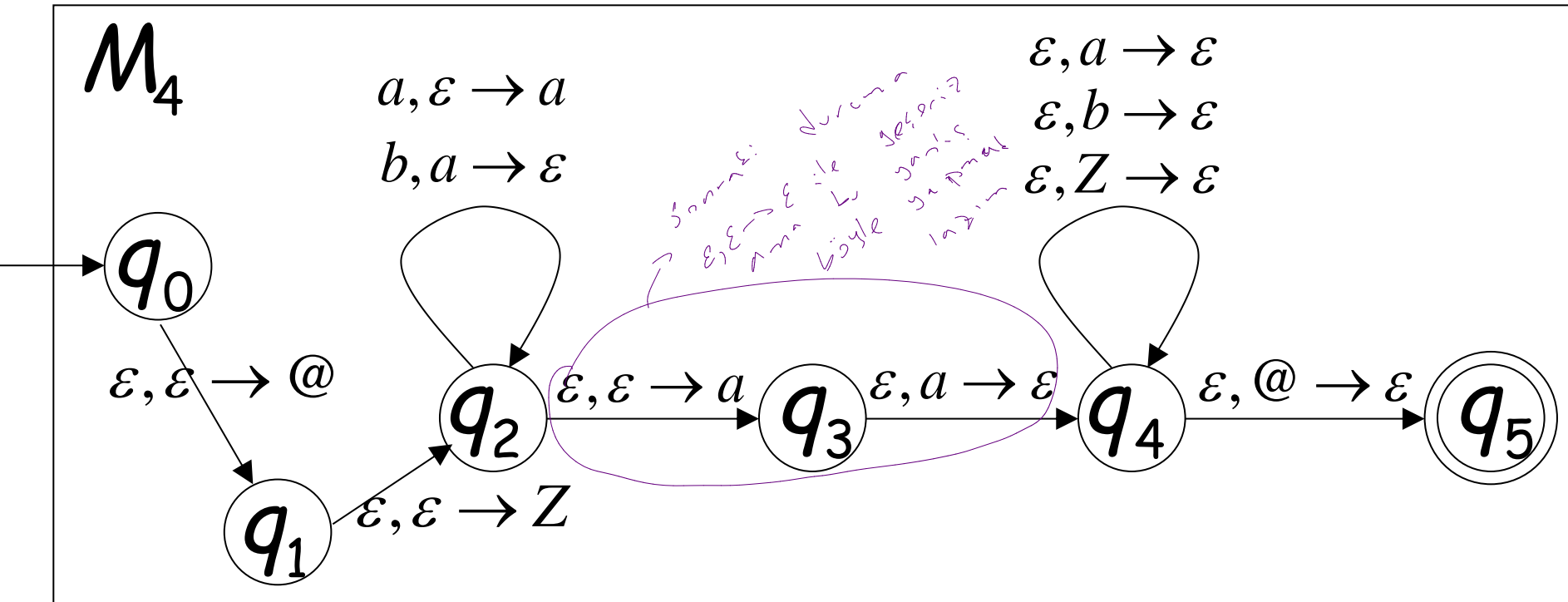
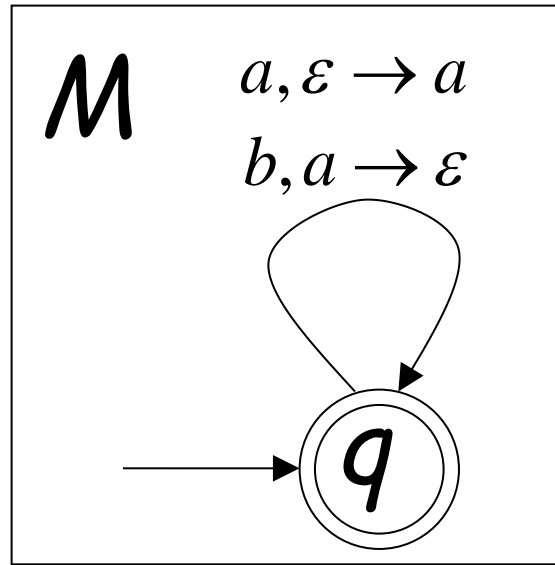


Where δ is a symbol of the stack alphabet

PDA M_4 is the final modified PDA

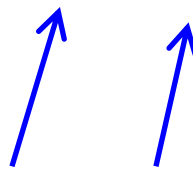
Note that the new initial stack symbol $\#$ is never used in any transition

Example:



Grammar Construction

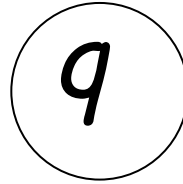
Variables: A_{q_i, q_j}



States of PDA

PDA

Kind 1: for each state



Grammar

$$A_{qq} \rightarrow \varepsilon$$

PDA

Kind 2: for every three states

p

q

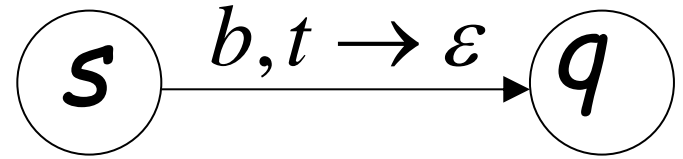
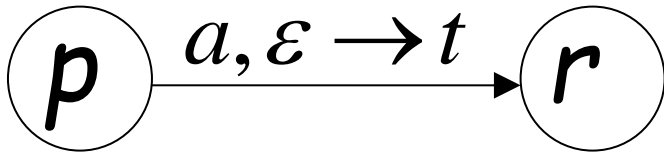
r

Grammar

$$A_{pq} \rightarrow A_{pr} A_{rq}$$

PDA

Kind 3: for every pair of such transitions

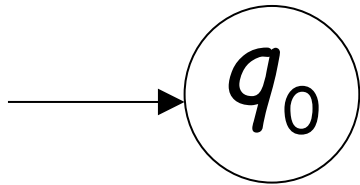


Grammar

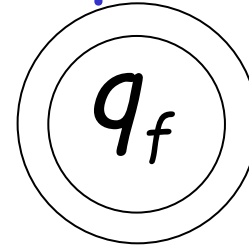
$$A_{pq} \rightarrow aA_{rs}b$$

PDA

Initial state



Accept state



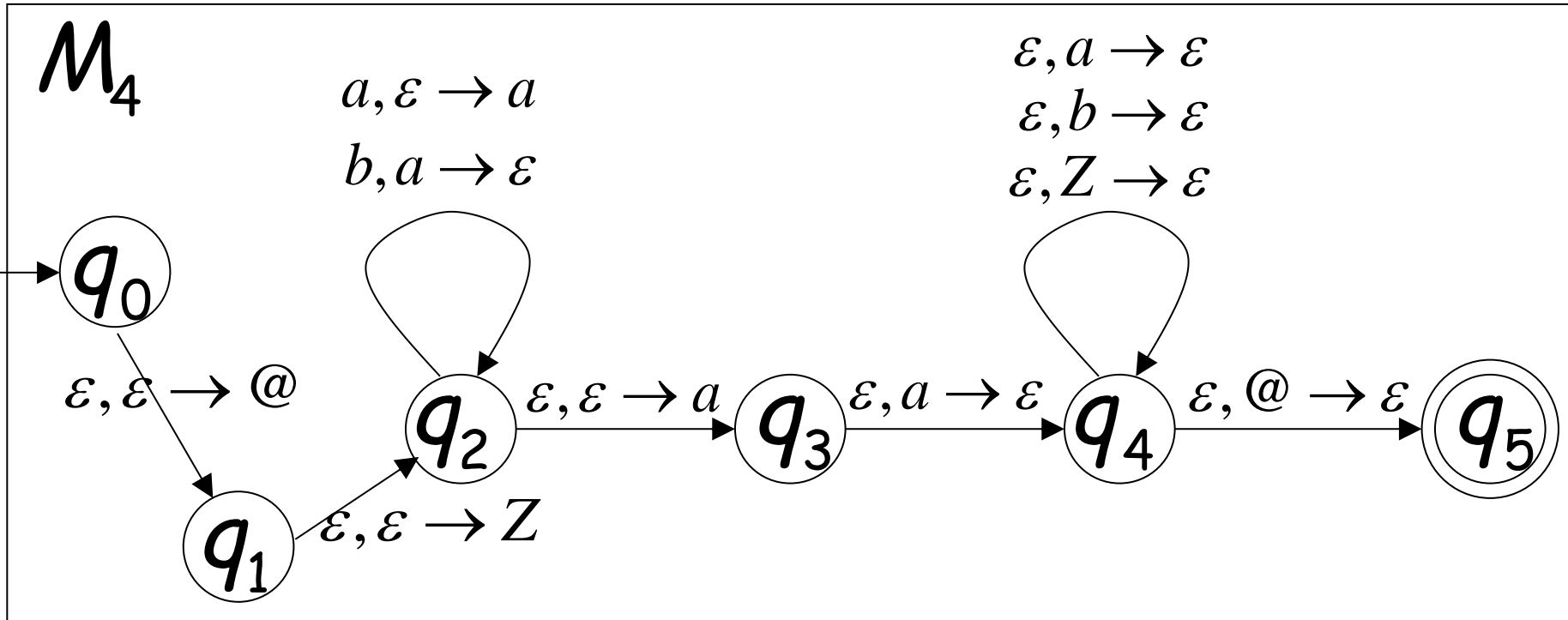
Grammar

Start variable

$A_{q_0 q_f}$

Example:

PDA



Grammar

Kind 1: from single states

$$A_{q_0q_0} \rightarrow \varepsilon$$

$$A_{q_1q_1} \rightarrow \varepsilon$$

$$A_{q_2q_2} \rightarrow \varepsilon$$

$$A_{q_3q_3} \rightarrow \varepsilon$$

$$A_{q_4q_4} \rightarrow \varepsilon$$

$$A_{q_5q_5} \rightarrow \varepsilon$$

Kind 2: from triplets of states

$$A_{q_0q_0} \rightarrow A_{q_0q_0} A_{q_0q_0} \mid A_{q_0q_1} A_{q_1q_0} \mid A_{q_0q_2} A_{q_2q_0} \mid A_{q_0q_3} A_{q_3q_0} \mid A_{q_0q_4} A_{q_4q_0} \mid A_{q_0q_5} A_{q_5q_0}$$

$$A_{q_0q_1} \rightarrow A_{q_0q_0} A_{q_0q_1} \mid A_{q_0q_1} A_{q_1q_1} \mid A_{q_0q_2} A_{q_2q_1} \mid A_{q_0q_3} A_{q_3q_1} \mid A_{q_0q_4} A_{q_4q_1} \mid A_{q_0q_5} A_{q_5q_1}$$

⋮

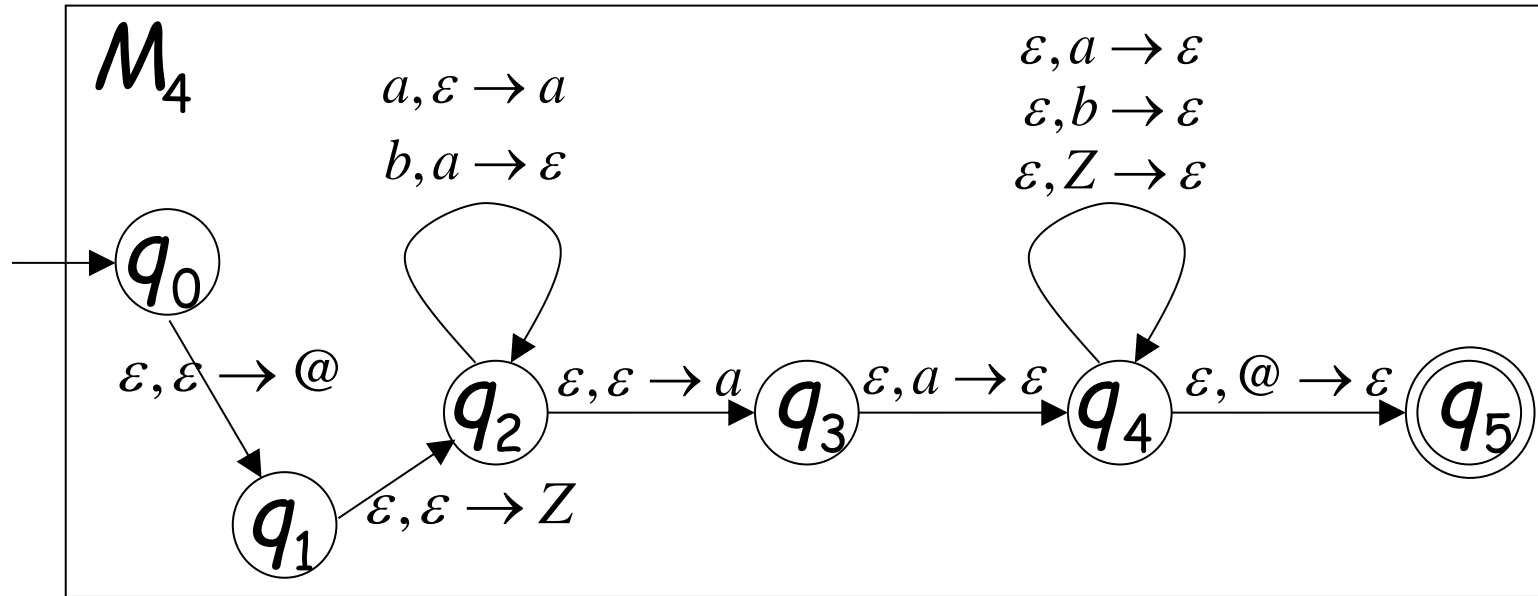
$$A_{q_0q_5} \rightarrow A_{q_0q_0} A_{q_0q_5} \mid A_{q_0q_1} A_{q_1q_5} \mid A_{q_0q_2} A_{q_2q_5} \mid A_{q_0q_3} A_{q_3q_5} \mid A_{q_0q_4} A_{q_4q_5} \mid A_{q_0q_5} A_{q_5q_5}$$

⋮

$$A_{q_5q_5} \rightarrow A_{q_5q_0} A_{q_0q_5} \mid A_{q_5q_1} A_{q_1q_5} \mid A_{q_5q_2} A_{q_2q_5} \mid A_{q_5q_3} A_{q_3q_5} \mid A_{q_5q_4} A_{q_4q_5} \mid A_{q_5q_5} A_{q_5q_5}$$

Start variable $A_{q_0q_5}$

Kind 3: from pairs of transitions



$$A_{q_0 q_5} \rightarrow A_{q_1 q_4}$$

$$A_{q_2 q_4} \rightarrow a A_{q_2 q_4}$$

$$A_{q_2 q_2} \rightarrow A_{q_2 q_2} b$$

$$A_{q_1 q_4} \rightarrow A_{q_2 q_4}$$

$$A_{q_2 q_2} \rightarrow a A_{q_2 q_2} b$$

$$A_{q_2 q_4} \rightarrow A_{q_3 q_3}$$

$$A_{q_2 q_4} \rightarrow a A_{q_2 q_3}$$

$$A_{q_2 q_4} \rightarrow A_{q_3 q_4}$$

Suppose that a PDA M is converted
to a context-free grammar G

G

Brace matching

We need to prove that $L(G) = L(M)$

or equivalently

$$L(G) \subseteq L(M)$$

$$L(G) \supseteq L(M)$$

$$L(G) \subseteq L(M)$$

We need to show that if G has derivation:

$$A_{q_0 q_f} \xRightarrow{*} w \quad (\text{string of terminals})$$

Then there is an accepting computation in M :

$$(q_0, w, \#) \xrightarrow{*} (q_f, \varepsilon, \#)$$

with input string w

We will actually show that if G has derivation:

$$A_{pq} \stackrel{*}{\Rightarrow} w$$

Then there is a computation in M :

$$(p, w, \varepsilon) \stackrel{*}{\succ} (q, \varepsilon, \varepsilon)$$

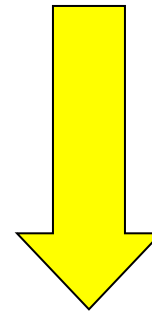
Therefore:

$$A_{q_0 q_f} \stackrel{*}{\Rightarrow} w$$



$$(q_0, w, \varepsilon) \stackrel{*}{\succ} (q_f, \varepsilon, \varepsilon)$$

Since there is no transition
with the # symbol



$$(q_0, w, \#) \stackrel{*}{\succ} (q_f, \varepsilon, \#)$$

Lemma:

If $A_{pq} \xRightarrow{*} w$ (string of terminals)

then there is a computation
from state p to state q on string w
which leaves the stack empty:

$$(p, w, \varepsilon) \xRightarrow{*} (q, \varepsilon, \varepsilon)$$

Proof Intuition:

$$A_{pq} \Rightarrow \dots \Rightarrow W$$

Type 2

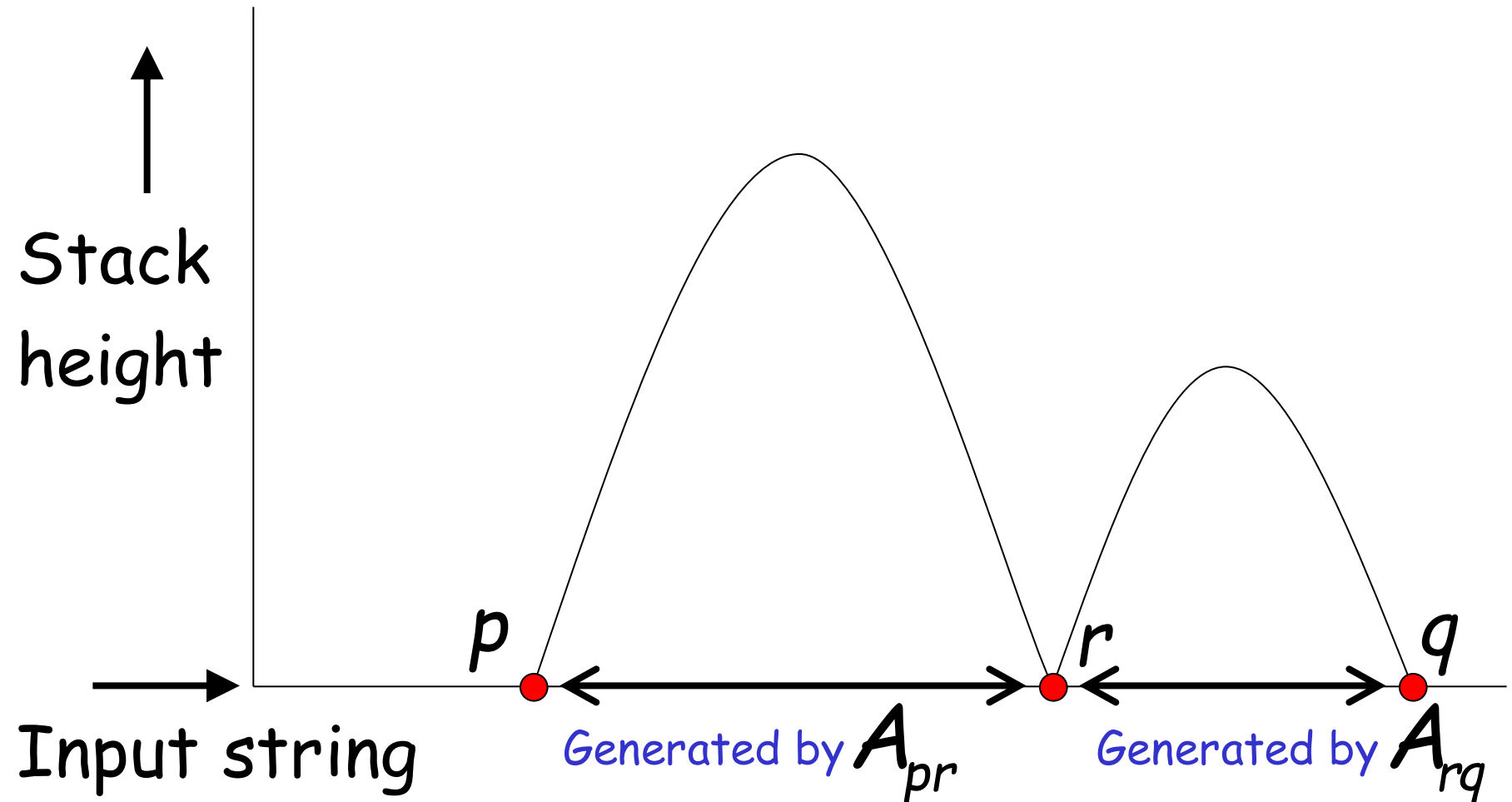
Case 1: $A_{pq} \Rightarrow A_{pr} A_{rq} \Rightarrow \dots \Rightarrow W$

Type 3

Case 2: $A_{pq} \Rightarrow a A_{rs} b \Rightarrow \dots \Rightarrow W$

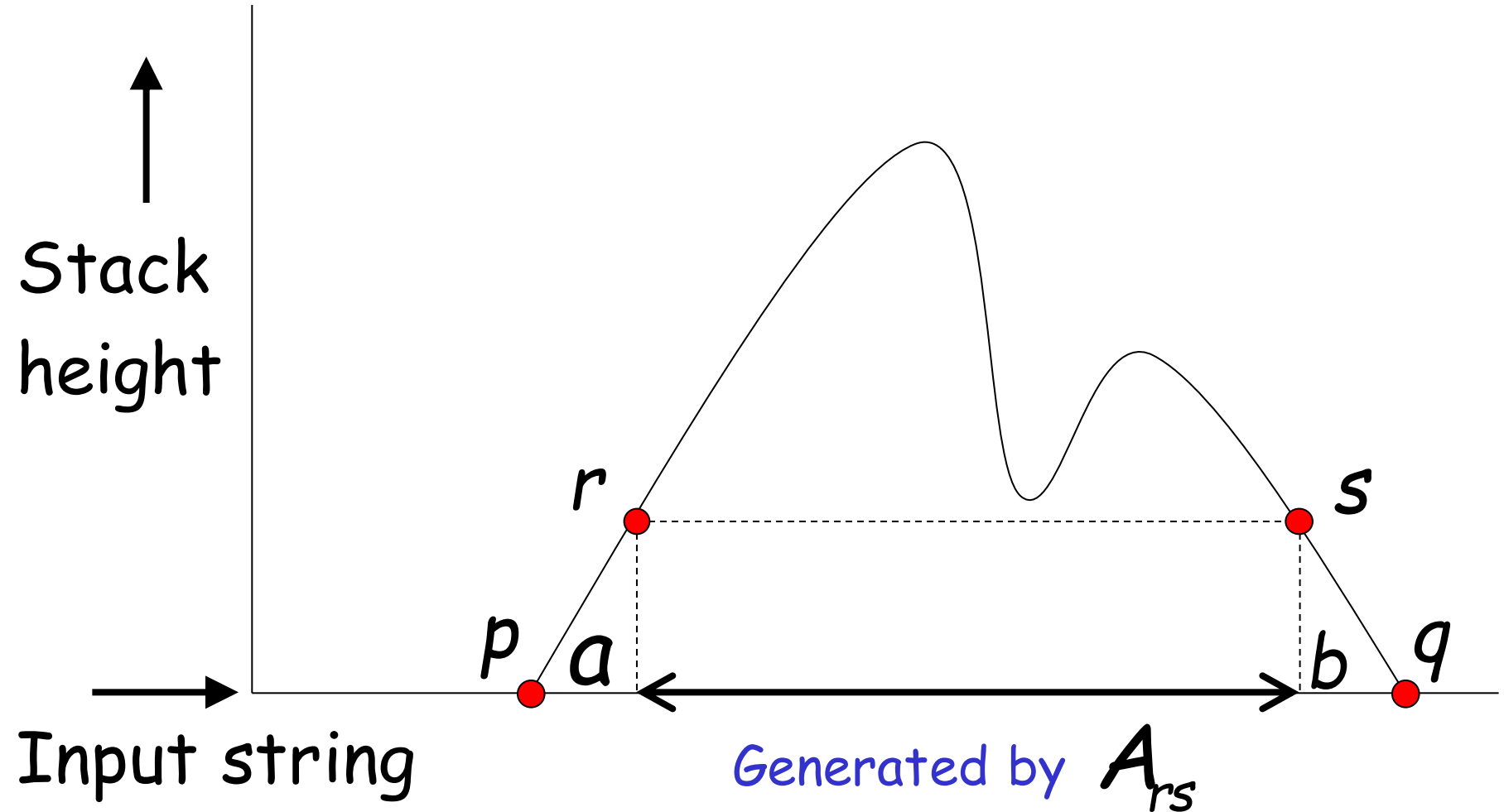
Type 2

Case 1: $A_{pq} \Rightarrow A_{pr} A_{rq} \Rightarrow \dots \Rightarrow w$



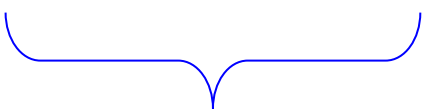
Type 3

Case 2: $A_{pq} \Rightarrow aA_{rs}b \Rightarrow \dots \Rightarrow w$



Formal Proof:

We formally prove this claim
by induction on the number
of steps in derivation:

$$A_{pq} \Rightarrow \cdots \Rightarrow W$$


number of steps

Induction Basis: $A_{pq} \Rightarrow w$

(one derivation step)

A Kind 1 production must have been used:

$$A_{pp} \rightarrow \varepsilon$$

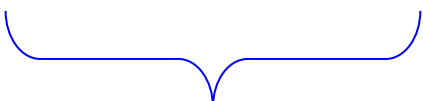
Therefore, $p = q$ and $w = \varepsilon$

This computation of PDA trivially exists:

$$(p, \varepsilon, \varepsilon) \stackrel{*}{\succ} (p, \varepsilon, \varepsilon)$$

Induction Hypothesis:

$$A_{pq} \Rightarrow \cdots \Rightarrow w$$

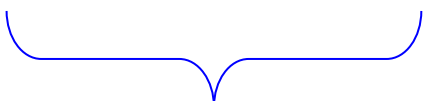

 k derivation steps

suppose it holds:

$$(p, w, \varepsilon) \stackrel{*}{\succ} (q, \varepsilon, \varepsilon)$$

Induction Step:

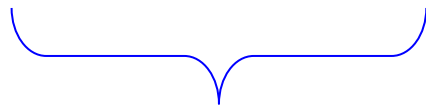
$$A_{pq} \Rightarrow \cdots \Rightarrow w$$


 $k + 1$ derivation steps

We have to show:

$$(p, w, \varepsilon) \stackrel{*}{\succ} (q, \varepsilon, \varepsilon)$$

$$A_{pq} \Rightarrow \dots \Rightarrow w$$



$k + 1$ derivation steps

Type 2

Case 1: $A_{pq} \Rightarrow A_{pr} A_{rq} \Rightarrow \dots \Rightarrow w$

Type 3

Case 2: $A_{pq} \Rightarrow a A_{rs} b \Rightarrow \dots \Rightarrow w$

Type 2

Case 1: $A_{pq} \Rightarrow A_{pr} A_{rq} \Rightarrow \dots \Rightarrow w$

$k + 1$ steps

We can write $w = yz$

$$A_{pr} \Rightarrow \dots \Rightarrow y$$

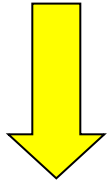
At most k steps

$$A_{rq} \Rightarrow \dots \Rightarrow z$$

At most k steps

$$A_{pr} \Rightarrow \dots \Rightarrow y$$

At most k steps

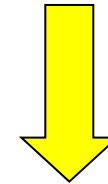


From induction
hypothesis, in PDA:

$$(p, y, \varepsilon) \stackrel{*}{\succ} (r, \varepsilon, \varepsilon)$$

$$A_{rq} \Rightarrow \dots \Rightarrow z$$

At most k steps

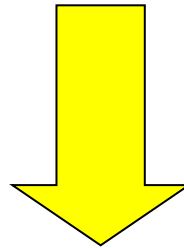


From induction
hypothesis, in PDA:

$$(r, z, \varepsilon) \stackrel{*}{\succ} (q, \varepsilon, \varepsilon)$$

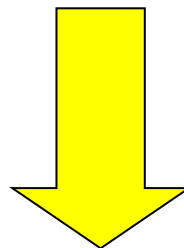
$$(p, y, \varepsilon) \succ^* (r, \varepsilon, \varepsilon)$$

$$(r, z, \varepsilon) \succ^* (q, \varepsilon, \varepsilon)$$



$$(p, yz, \varepsilon) \succ^* (r, z, \varepsilon) \succ^* (q, \varepsilon, \varepsilon)$$

since $w = yz$



$$(p, w, \varepsilon) \succ^* (q, \varepsilon, \varepsilon)$$

Type 3

Case 2: $A_{pq} \Rightarrow aA_{rs}b \Rightarrow \dots \Rightarrow w$

$k + 1$ steps

We can write $w = ayb$

$A_{rs} \Rightarrow \dots \Rightarrow y$

At most k steps

$$A_{rs} \Rightarrow \cdots \Rightarrow y$$

At most k steps

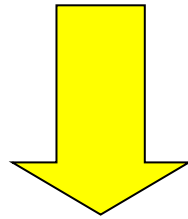


From induction hypothesis,
the PDA has computation:

$$(r, y, \varepsilon) \stackrel{*}{\succ} (s, \varepsilon, \varepsilon)$$

Type 3

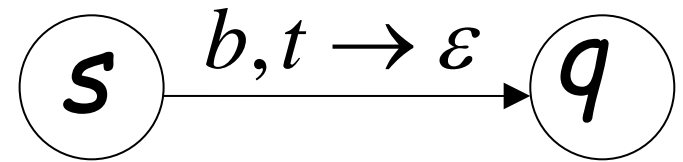
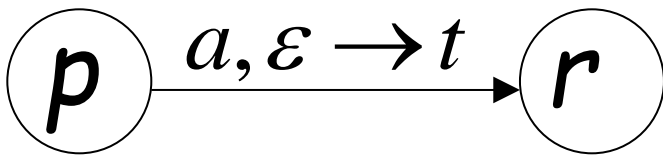
$$A_{pq} \Rightarrow aA_{rs}b \Rightarrow \dots \Rightarrow w$$

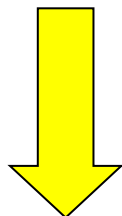
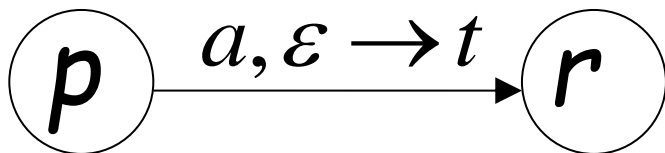


Grammar contains production

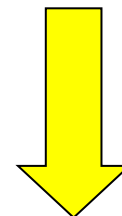
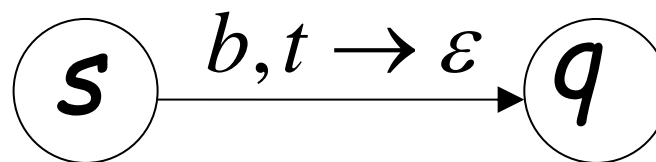
$$A_{pq} \rightarrow aA_{rs}b$$

And PDA Contains transitions





$$(p, ayb, \varepsilon) \succ (r, yb, t)$$



$$(s, b, t) \succ (q, \varepsilon, \varepsilon)$$

We know

$$(r, y, \varepsilon) \succ^* (s, \varepsilon, \varepsilon) \quad \Rightarrow \quad (r, yb, t) \succ^* (s, b, t)$$

We also know

$$(p, ayb, \varepsilon) \succ (r, yb, t)$$

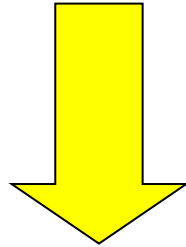
$$(s, b, t) \succ (q, \varepsilon, \varepsilon)$$

Therefore:

$$(p, ayb, \varepsilon) \succ (r, yb, t) \succ^* (s, b, t) \succ (q, \varepsilon, \varepsilon)$$

$$(p, ayb, \varepsilon) \succ (r, yb, t) \succ^* (s, b, t) \succ (q, \varepsilon, \varepsilon)$$

since $w = ayb$



$$(p, w, \varepsilon) \succ^* (q, \varepsilon, \varepsilon)$$

END OF PROOF

So far we have shown:

$$L(G) \subseteq L(M)$$

With a similar proof we can show

$$L(G) \supseteq L(M)$$

Therefore: $L(G) = L(M)$