Automatele Pushdown (PDA)

Andrei Paun

Cuprins

- Motivare
- Definitie
- Exemple
- Moduri de acceptare
- Determinism si nedeterminism
- Relatia cu gramaticile independente de context
- Determinism si CFG
- Sumar

Motivare

- DFA/NFA nu sunt suficient de puternice

- $\{a^i b^i | \ i>0\}$

  gramatica: S -> aSb | ab
- $\{ww^r | w$ din $\{a+b\}^*\}$ palindrom de lungime para

  gramatica: S -> aSa | bSb | λ
- limbajul format din paranteze balansate

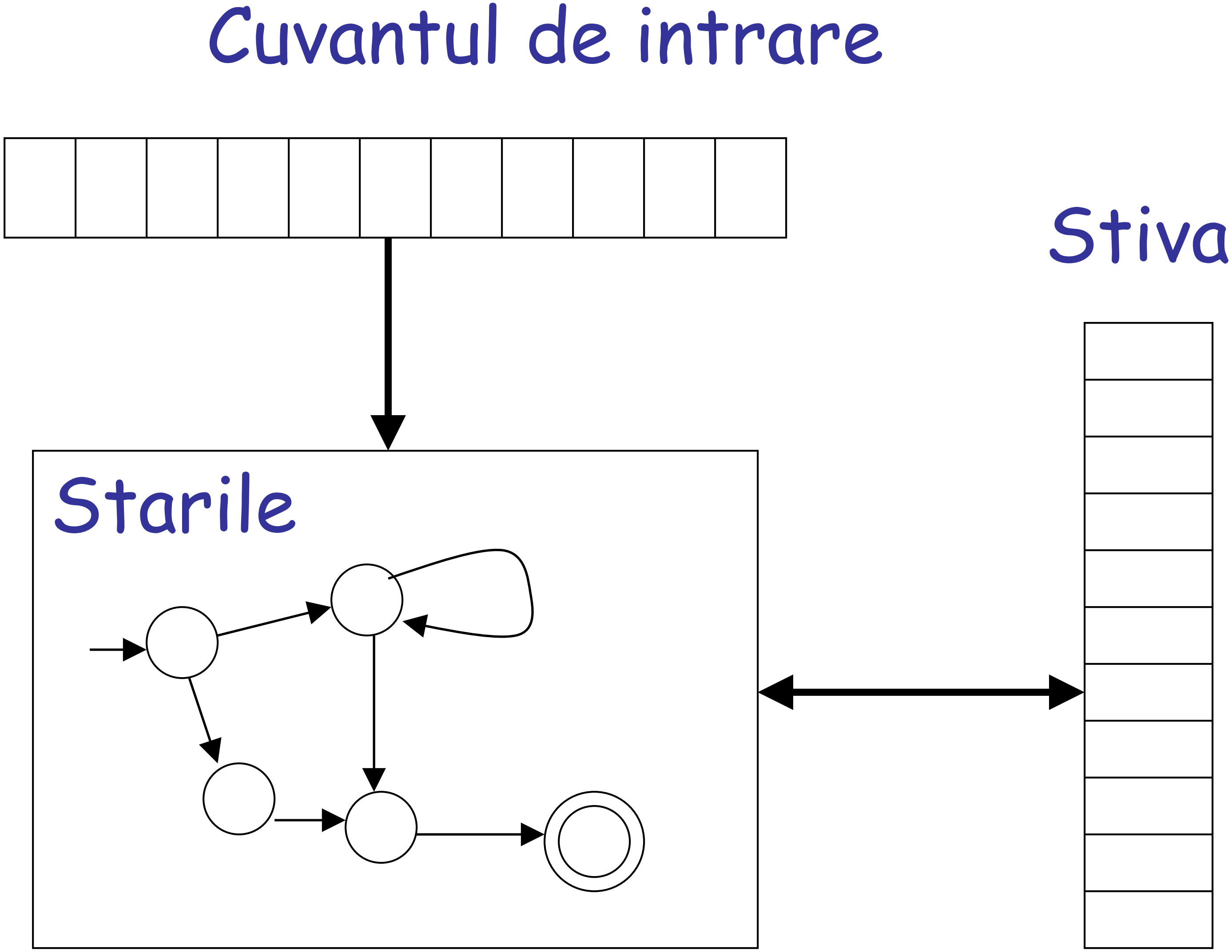
  gramatica: S -> (S) | SS | λ

Motivare

- Automate Pushdown: λ-NFA cu stiva

- pentru marit puterea NFA-urilor restrictionam “regulile”/ tranzitiile
- avem acces la o stiva (LIFO)

Automate Pushdown -- PDA

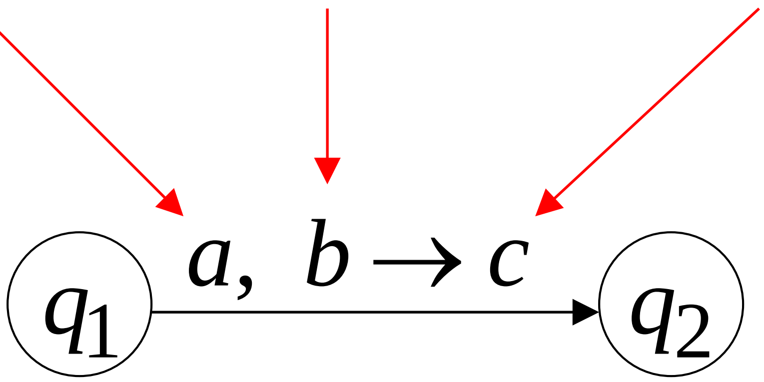


Reprezentare grafica

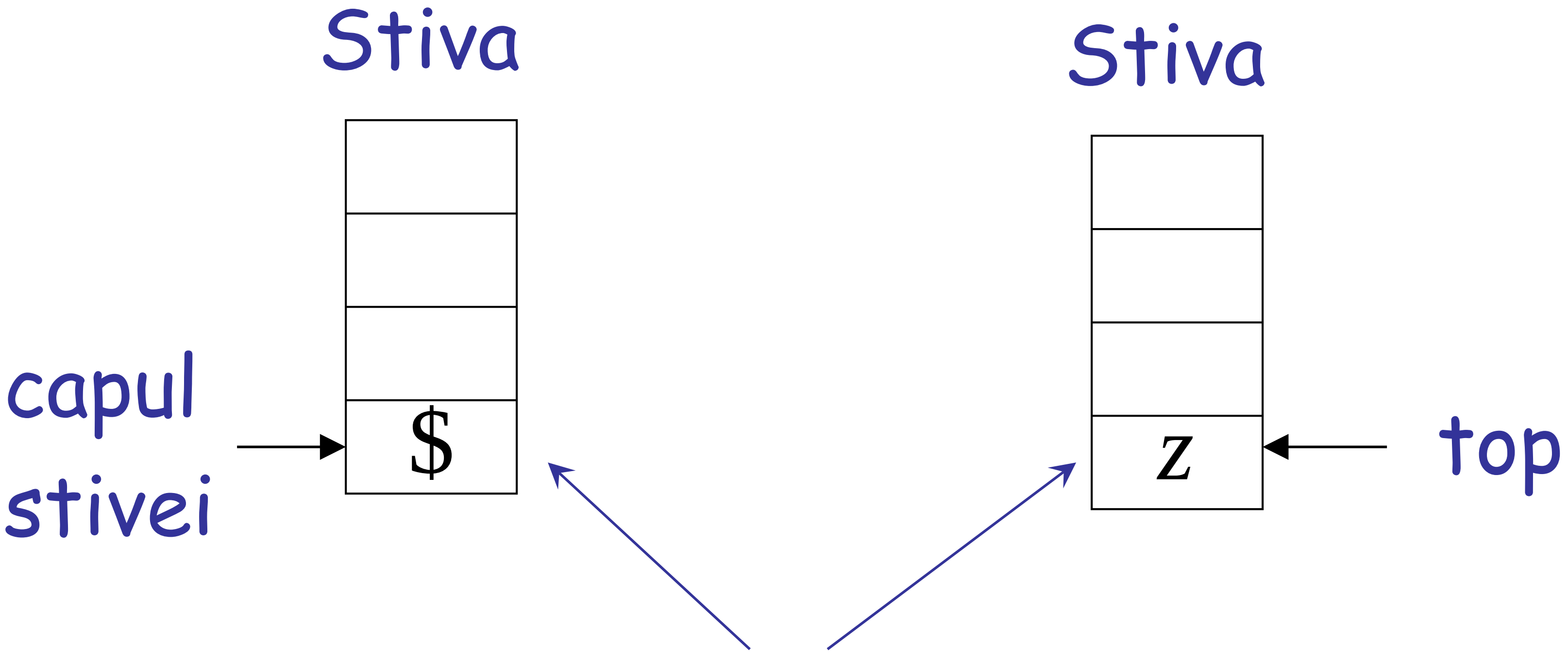
Simbol de intrare

simbol pop

cuvant push

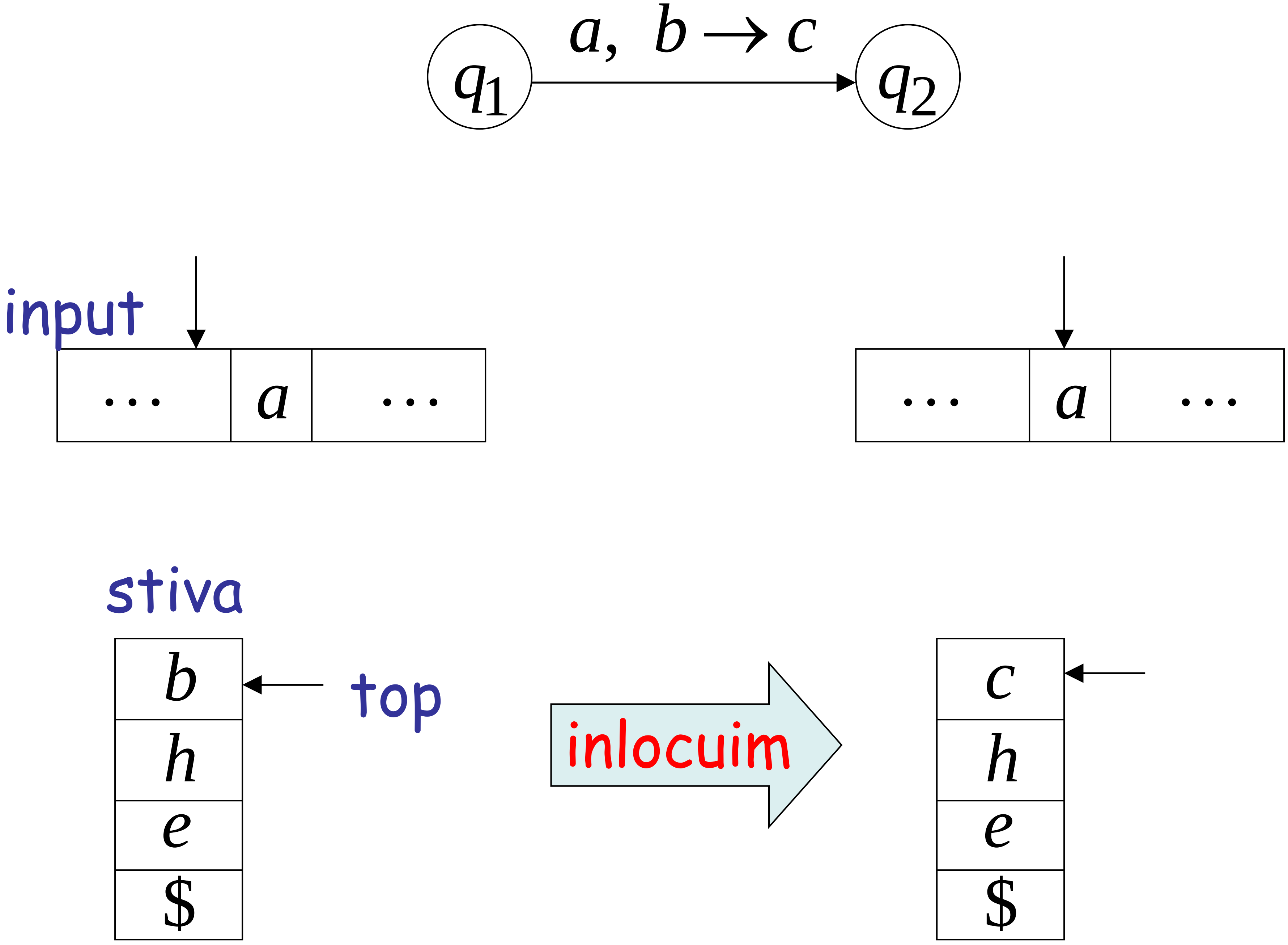


Simbolul initial pe stiva

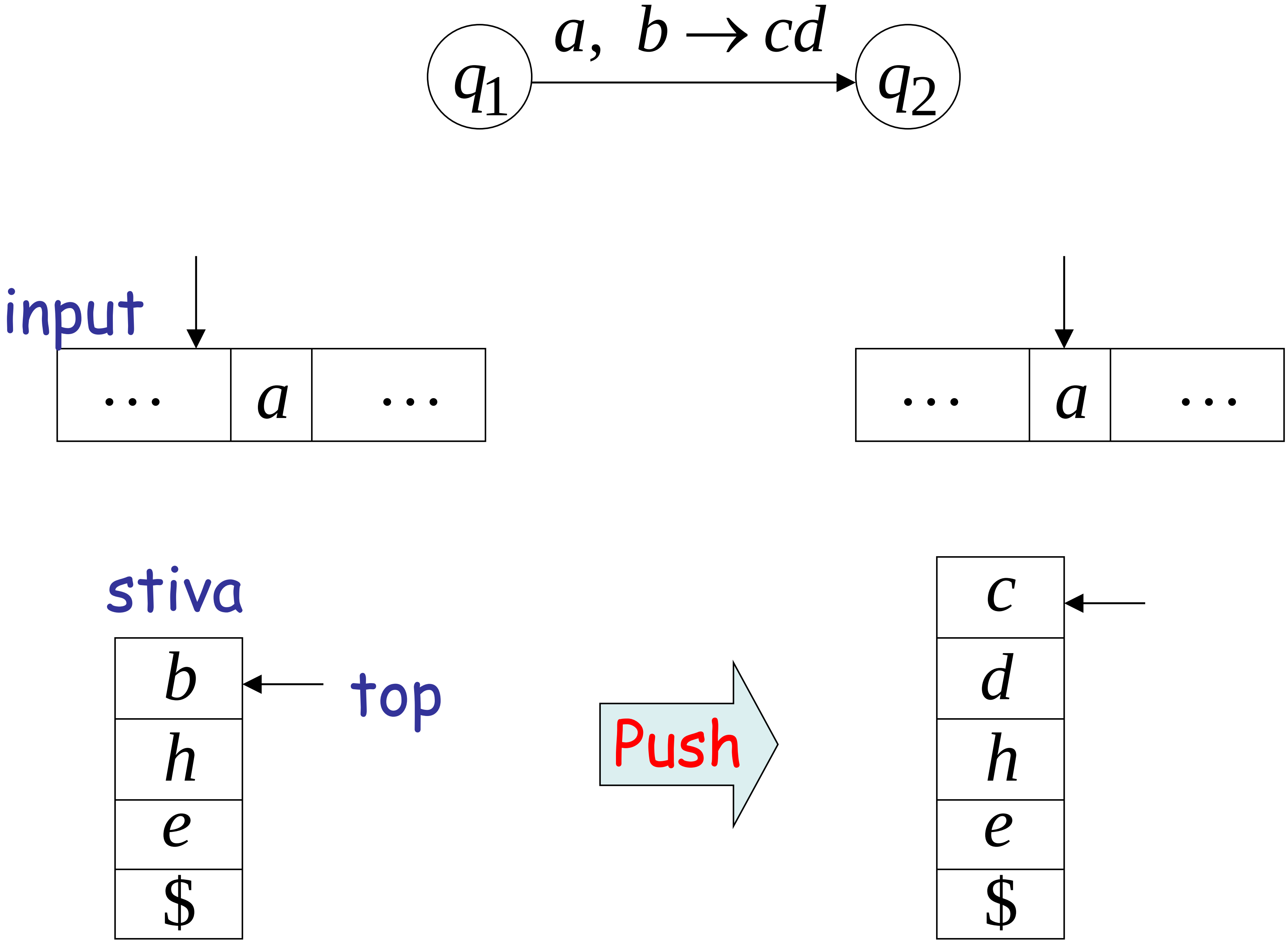


Sibolul special (ultimul din stiva)

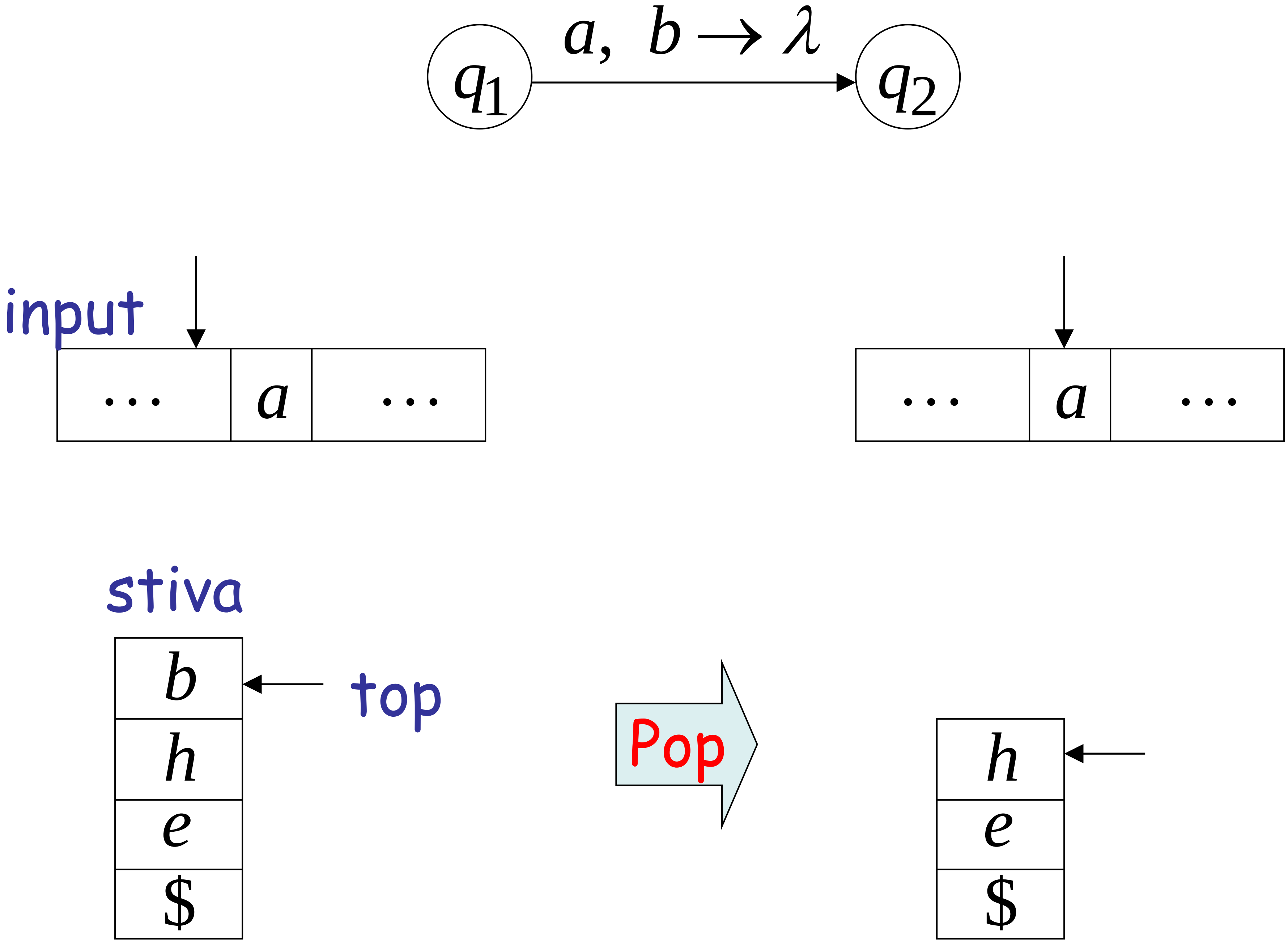
Apare la momentul 0



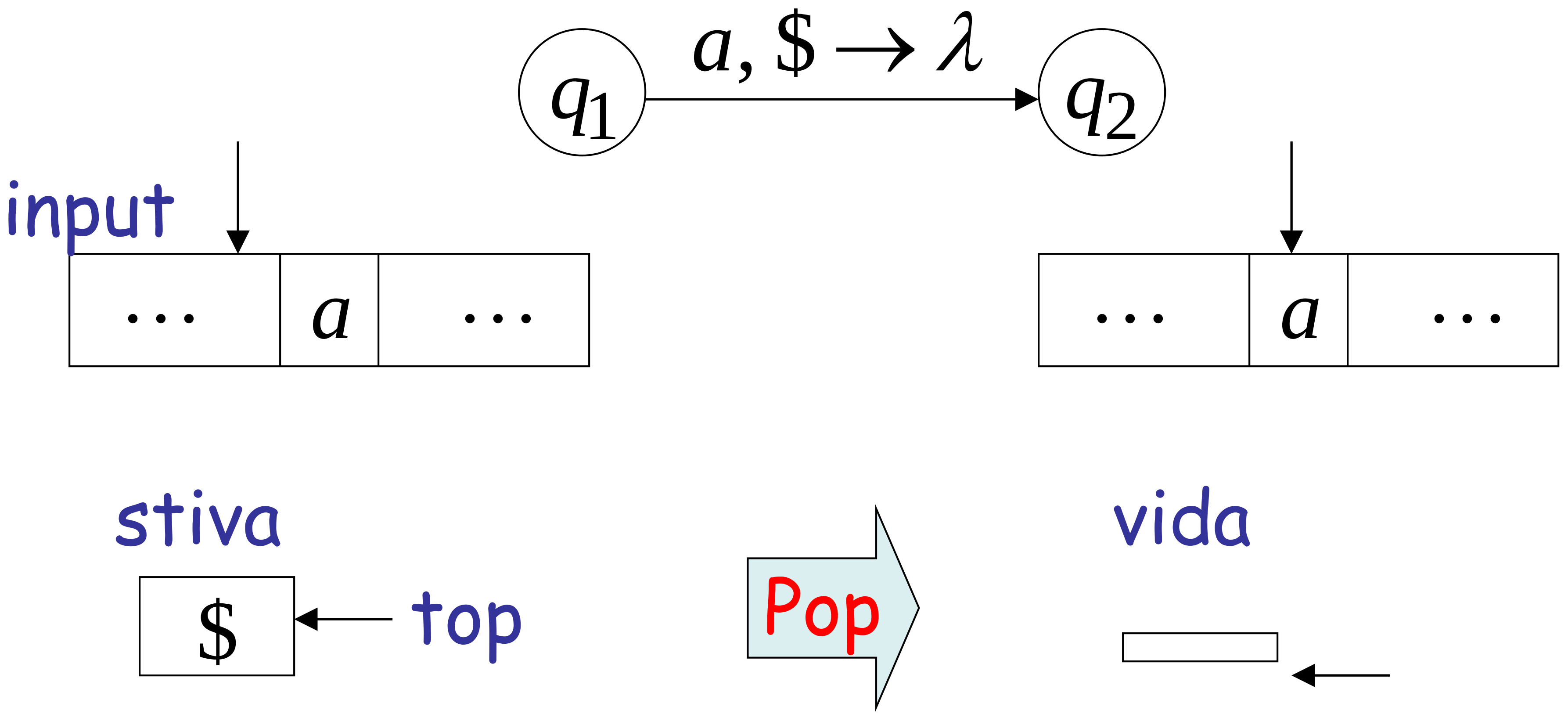
$a, \; b \rightarrow c$
q_1
q_2
input
$\cdots$ a $\cdots$
$\cdots$ a $\cdots$
stiva
b
h
e
$\$$
top
inlocuim
c
h
e
$\$$



$a, \ b \to cd$
q_1
q_2
input
$\cdots$ a $\cdots$
$\cdots$ a $\cdots$
stiva
b
h
e
$\$$
top
Push
c
d
h
e
$\$$


q_1
$a, \; b \rightarrow \lambda$
q_2
input
$\cdots$ a $\cdots$
$\cdots$ a $\cdots$
stiva
b
h
e
$\$$
top
Pop
h
e
$\$$

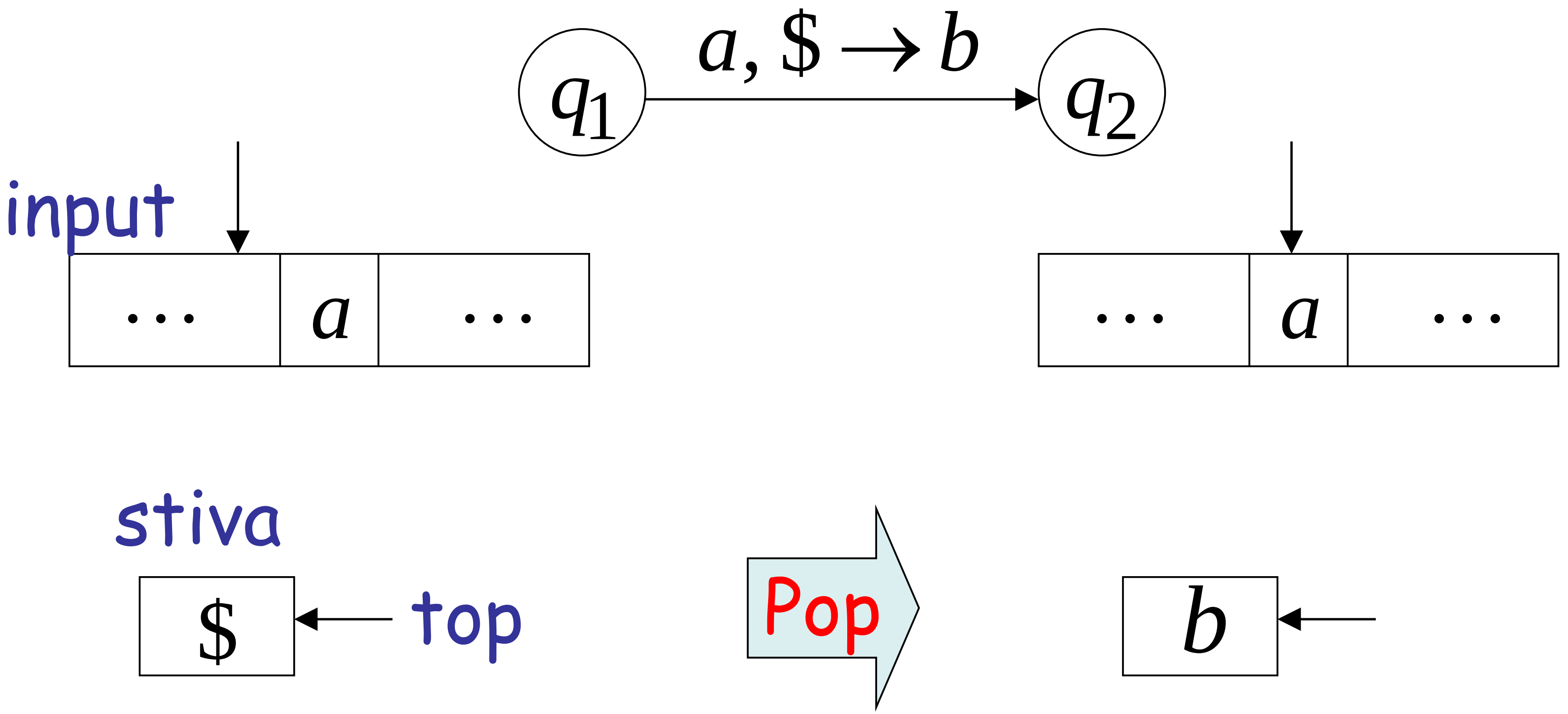
Stiva vida



Automatul SE OPRESTE

nu mai avem tranzitii posibile dupa q_2

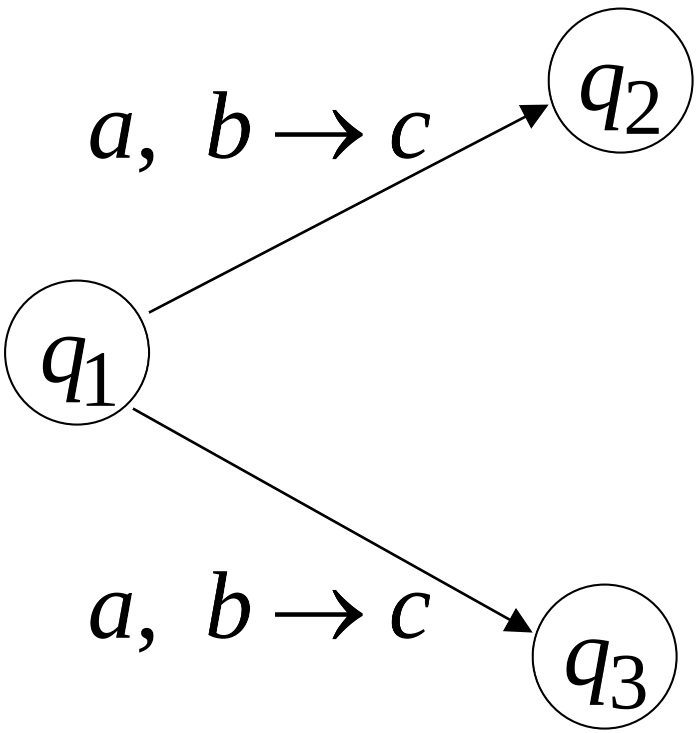
O tranzitie posibila

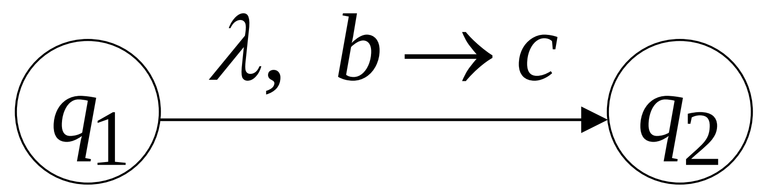


Nedeterminism

PDA-urile sunt nedeterministe

Tranzitiile nedeterministe sunt posibile



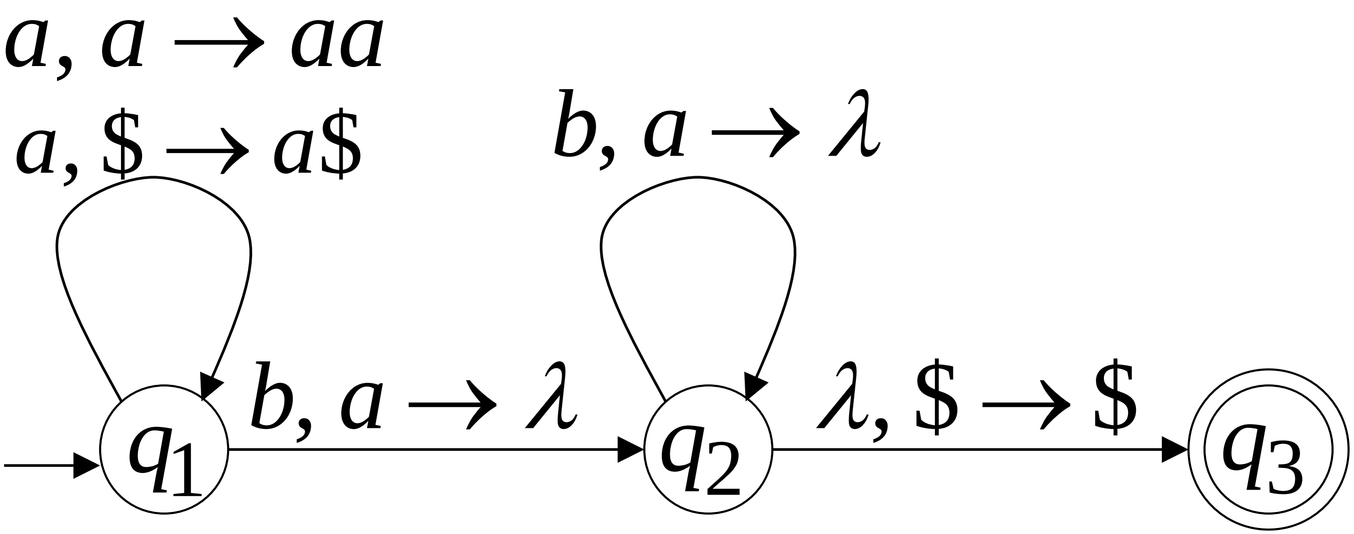


$\lambda -$ transition

Exemplu de PDA

PDA M

$$L(M) = \{a^n b^n \mid n \geq 1\}$$

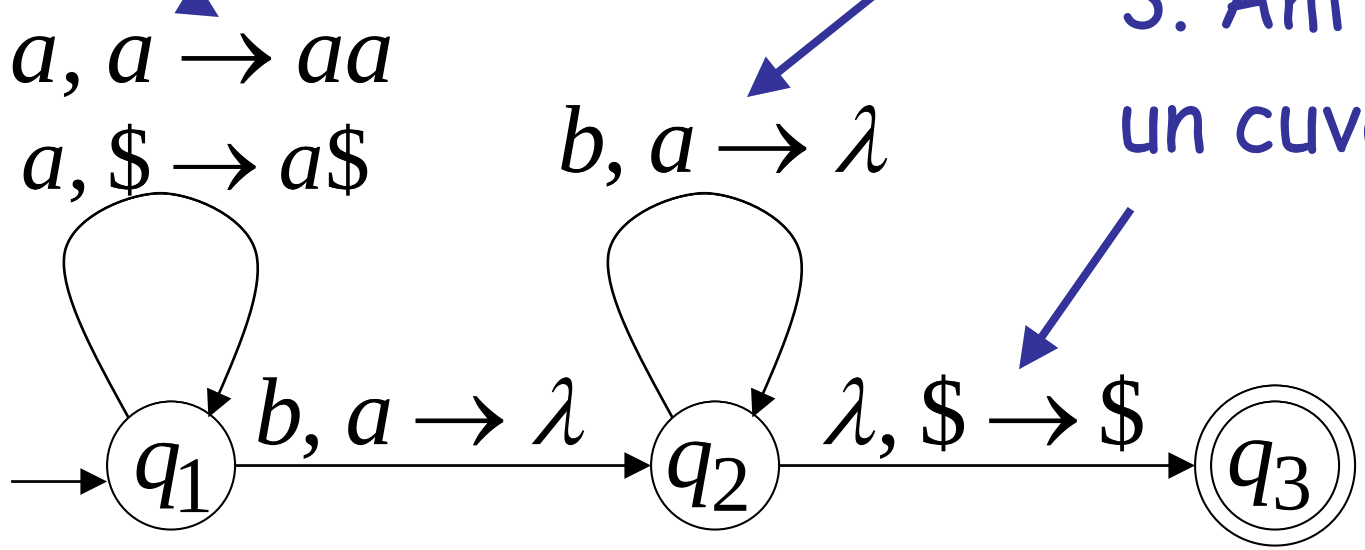


Ideea de baza:

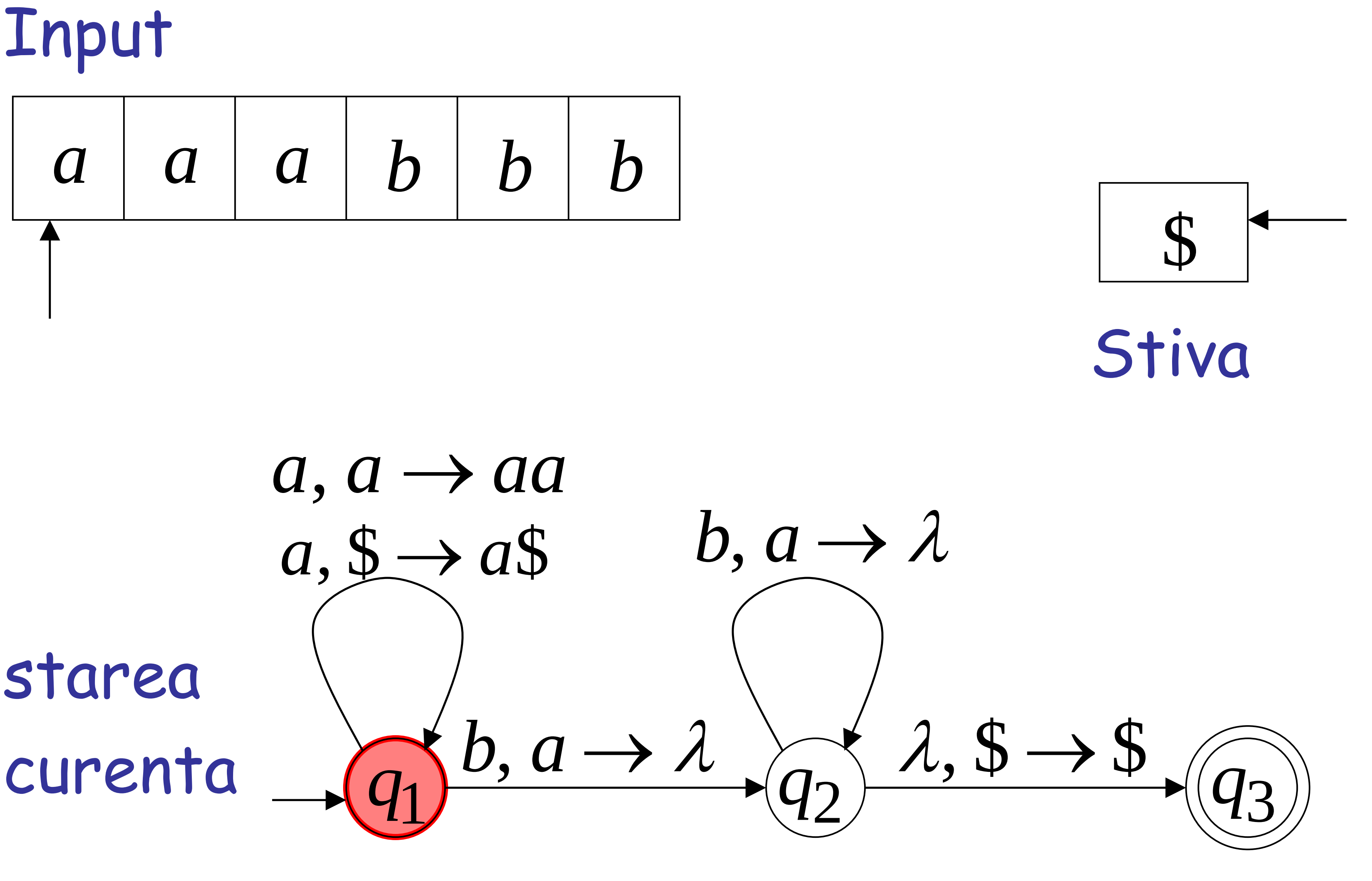
1. Push a-urii pe stiva

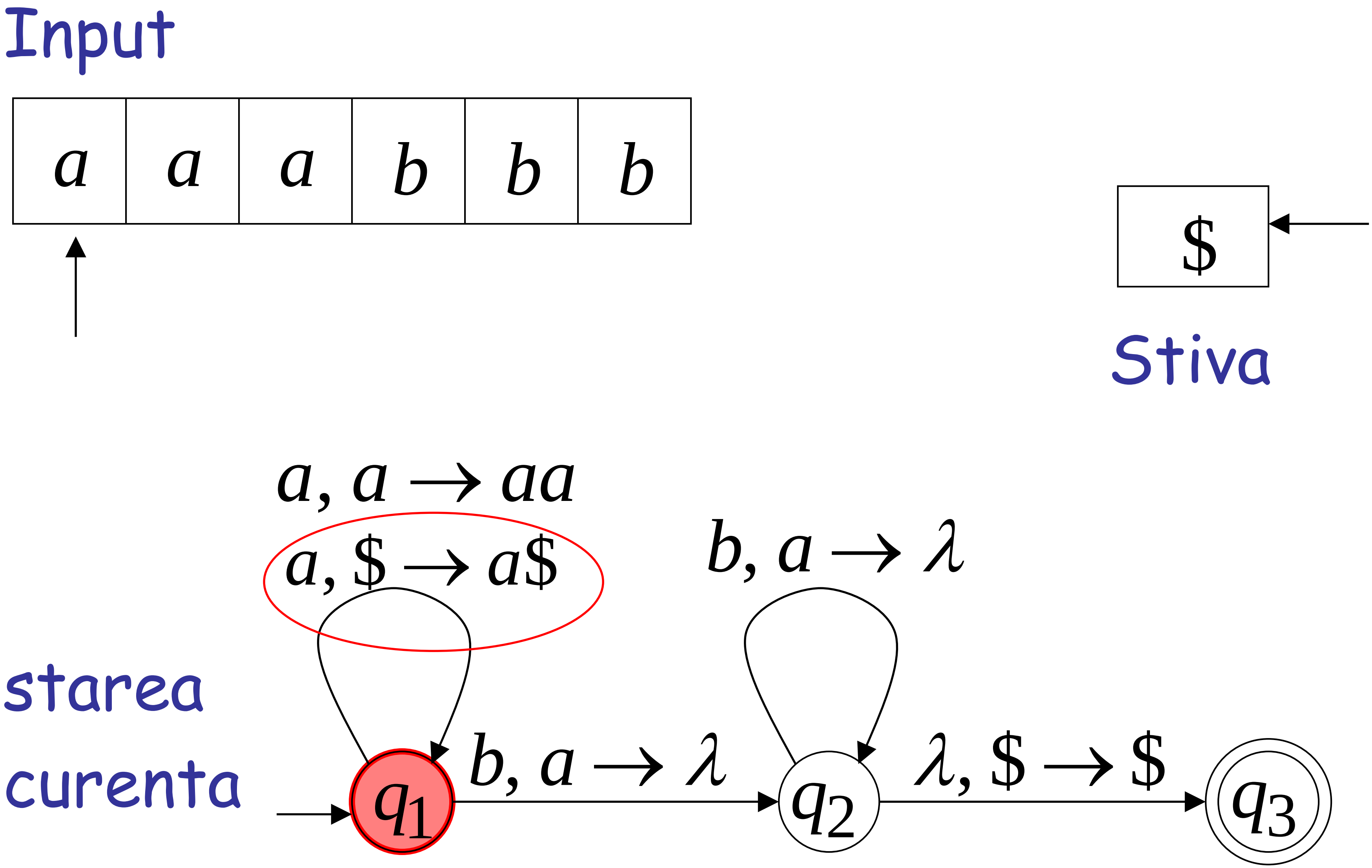
2. pentru fiecare b de la intrare consumam un a de pe stiva

3. Am gasit un cuvant



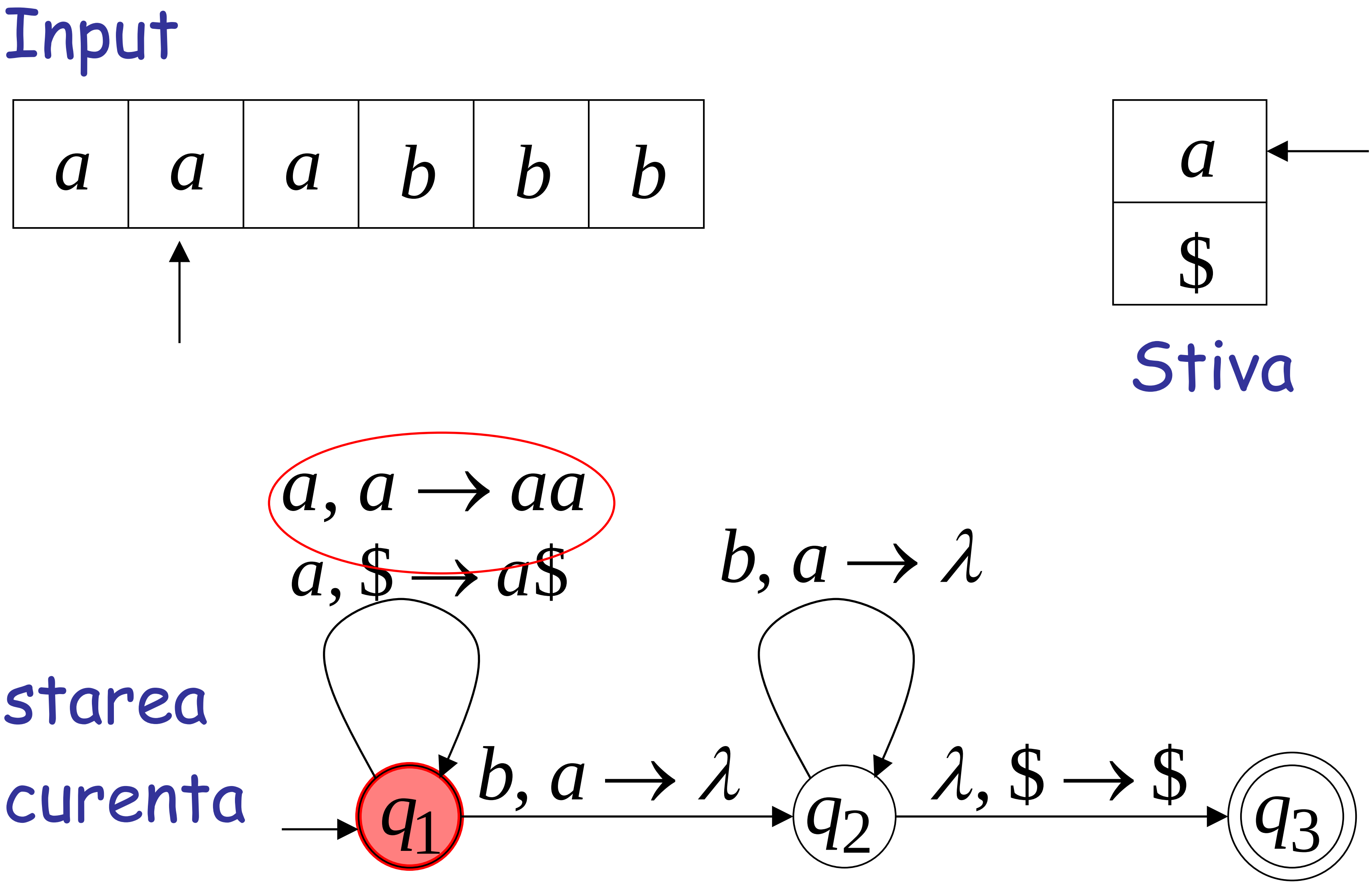
Exemplu de executie: momentul 0



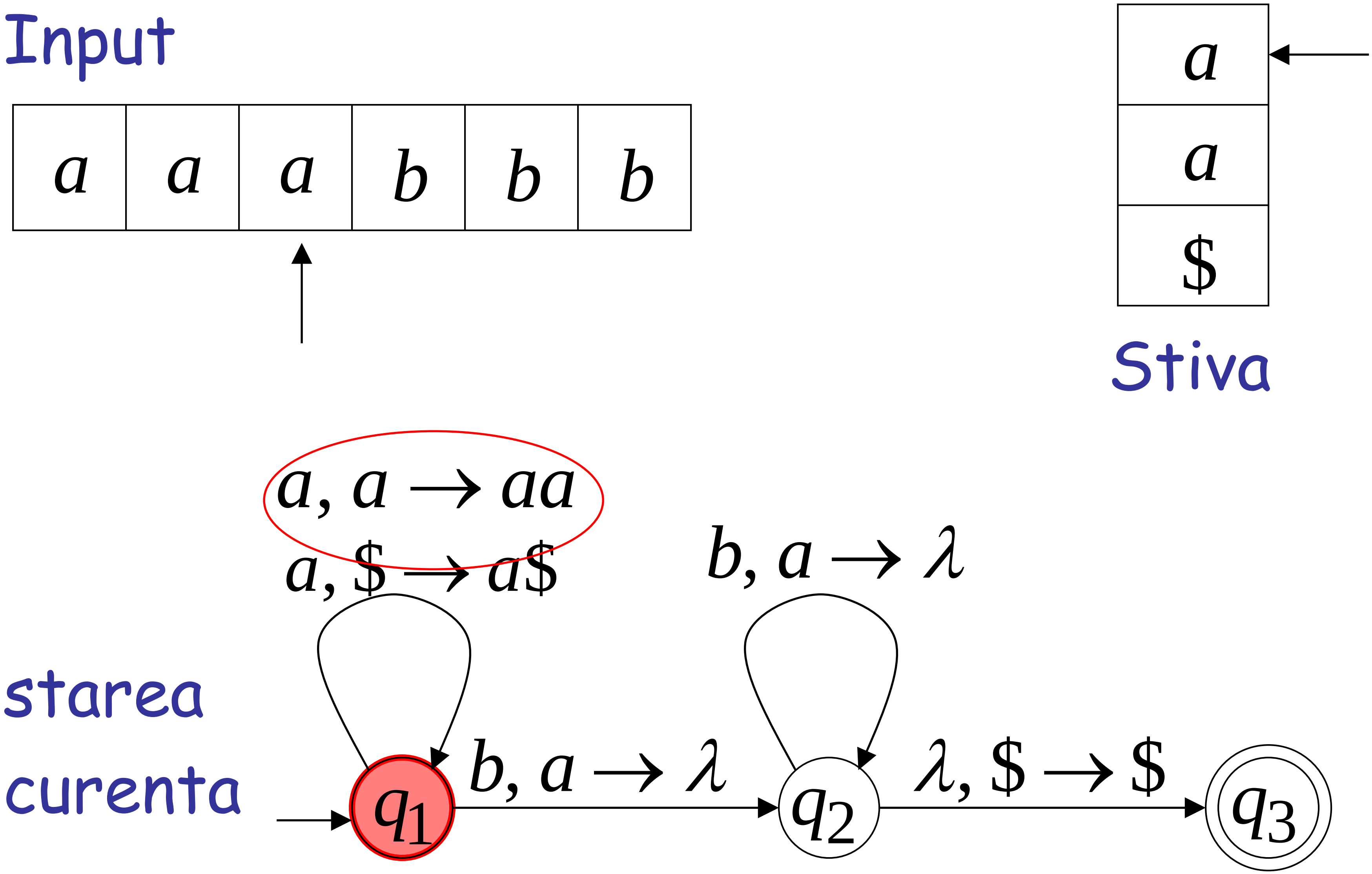


Input
a a a b b b
$
Stiva
a, a → aa
a, $ → a$
b, a → λ
starea
curenta
q1
b, a → λ
q2
λ, $ → $
q3

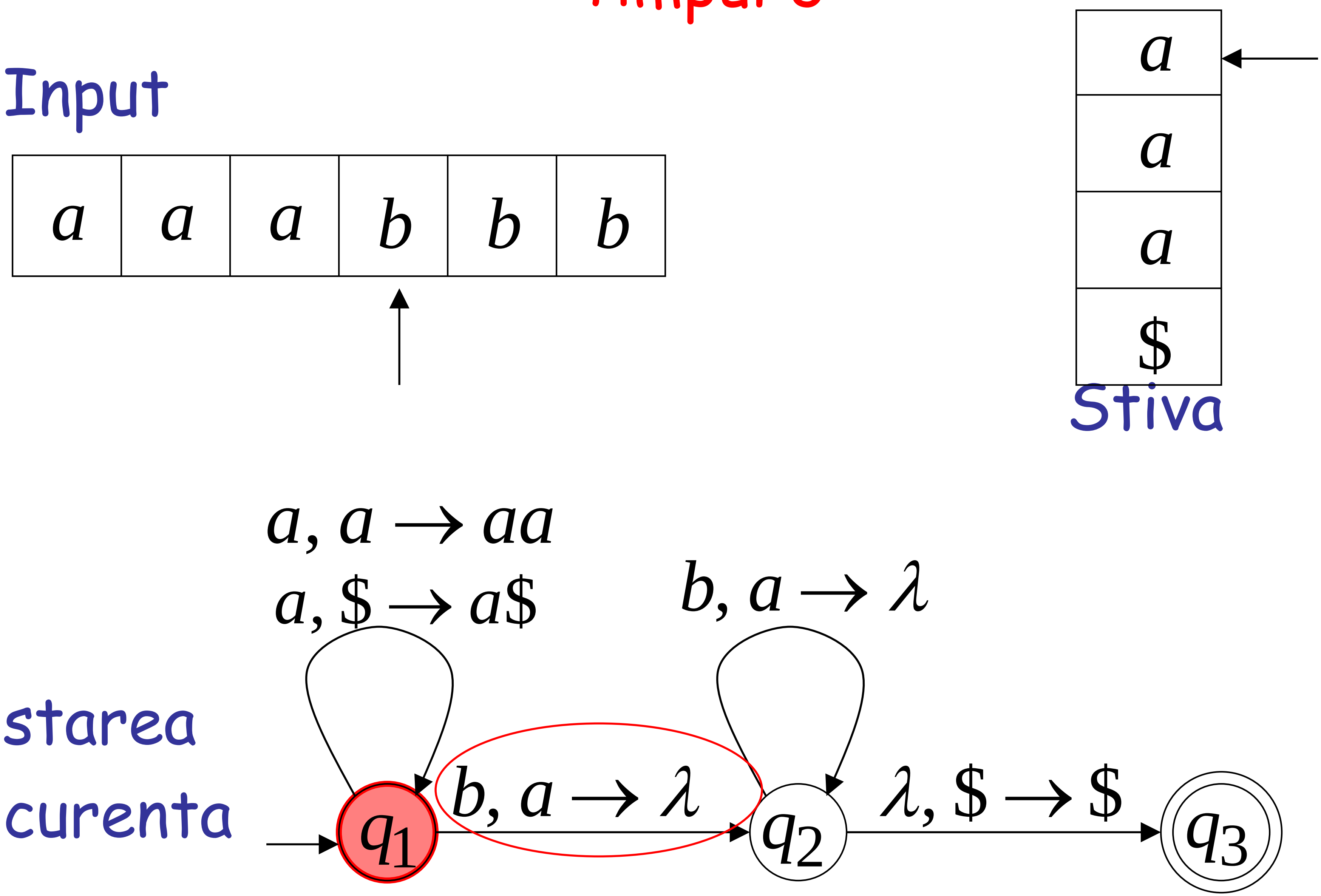
Timpul 1



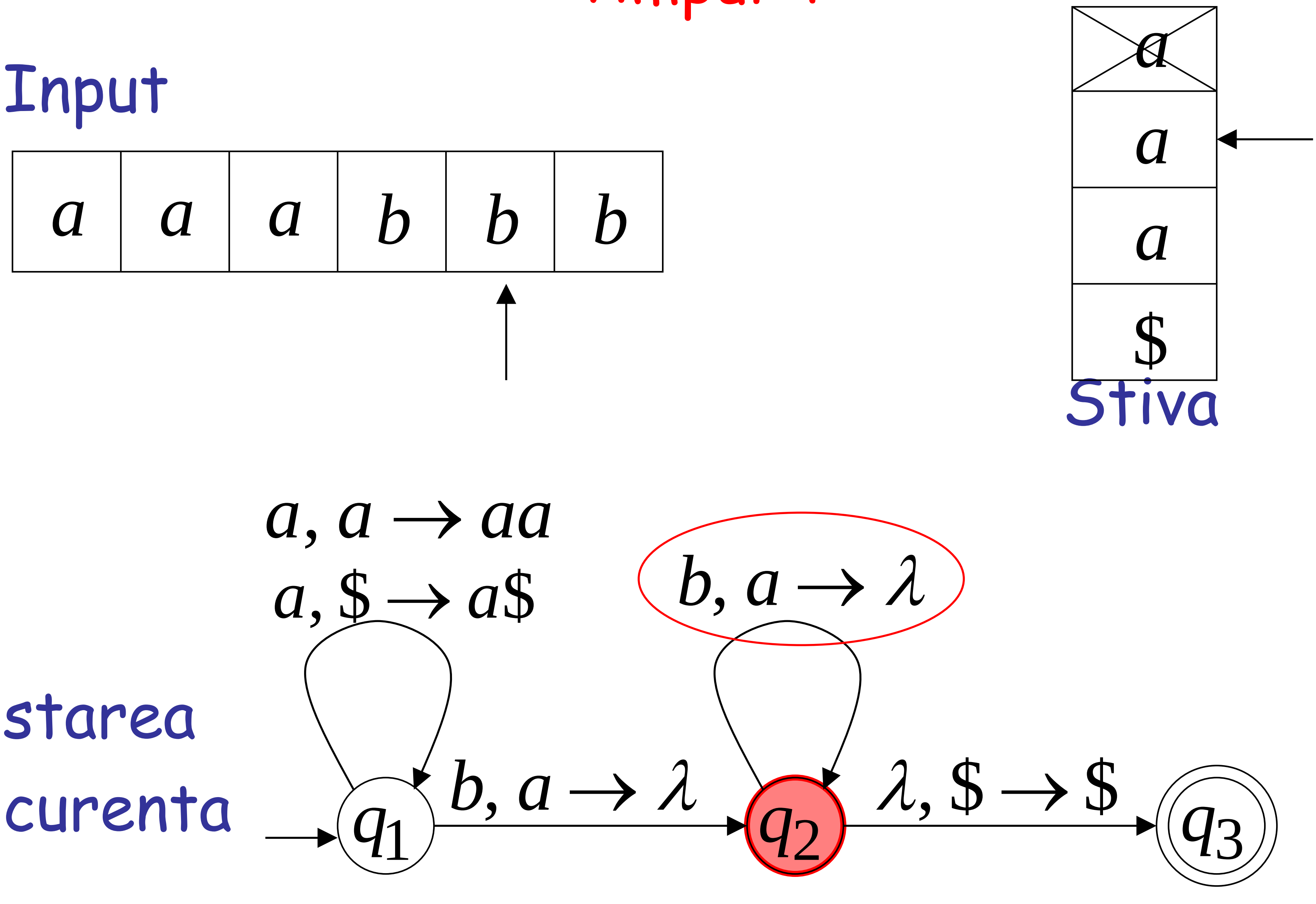
Timpul 2


Input
a a a b b b
a
a
$
Stiva
a, a → aa
a, $ → a$
b, a → λ
b, a → λ
λ, $ → $
starea
curenta
q1
q2
q3

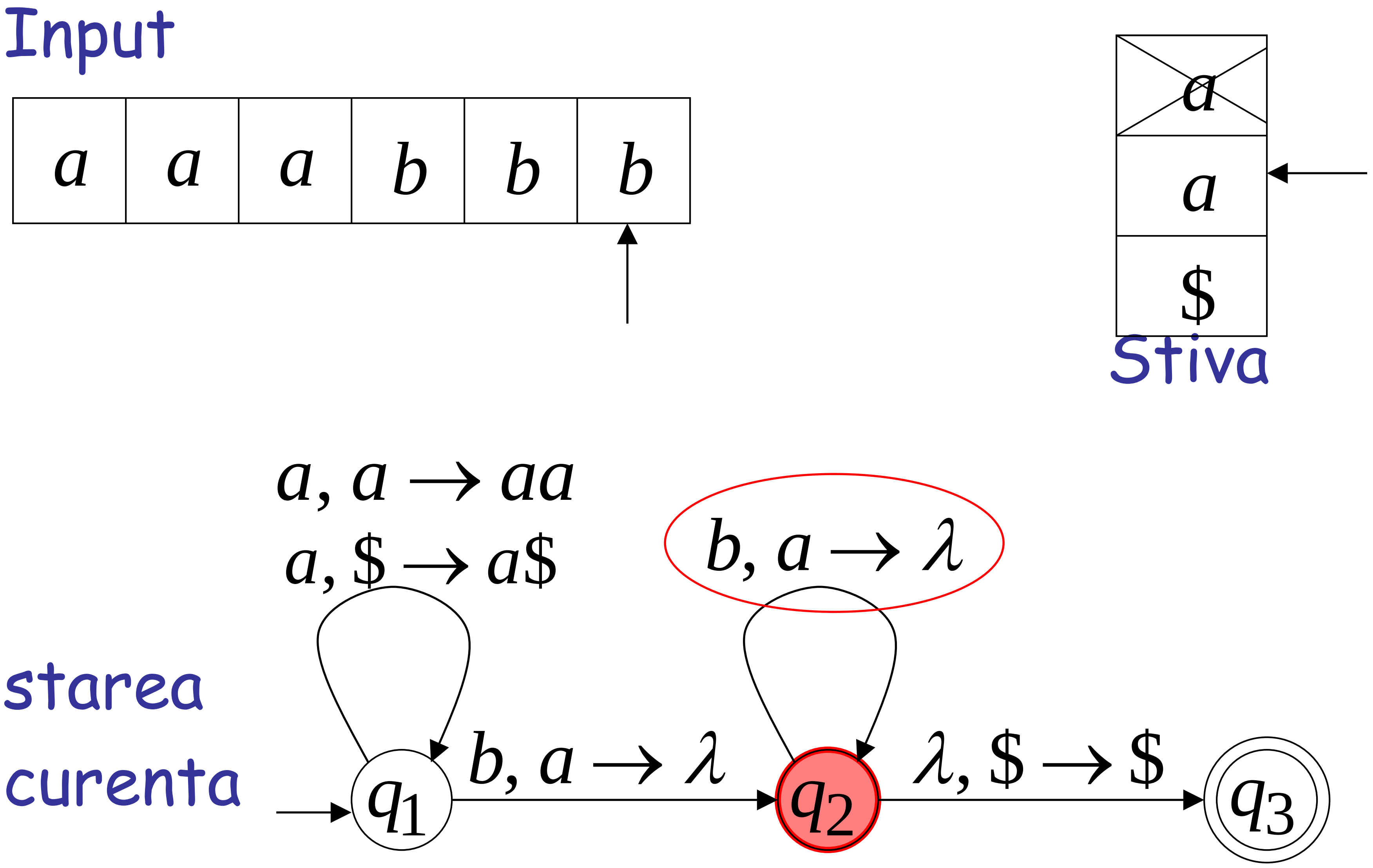
Timpul 3


Input
a a a b b b
a
a
a
$
Stiva
$a, a \to aa$
$a, \$ \to a\$$
$b, a \to \lambda$
$b, a \to \lambda$
$\lambda, \$ \to \$$
starea
curenta
q_1
q_2
q_3

Timpul 4

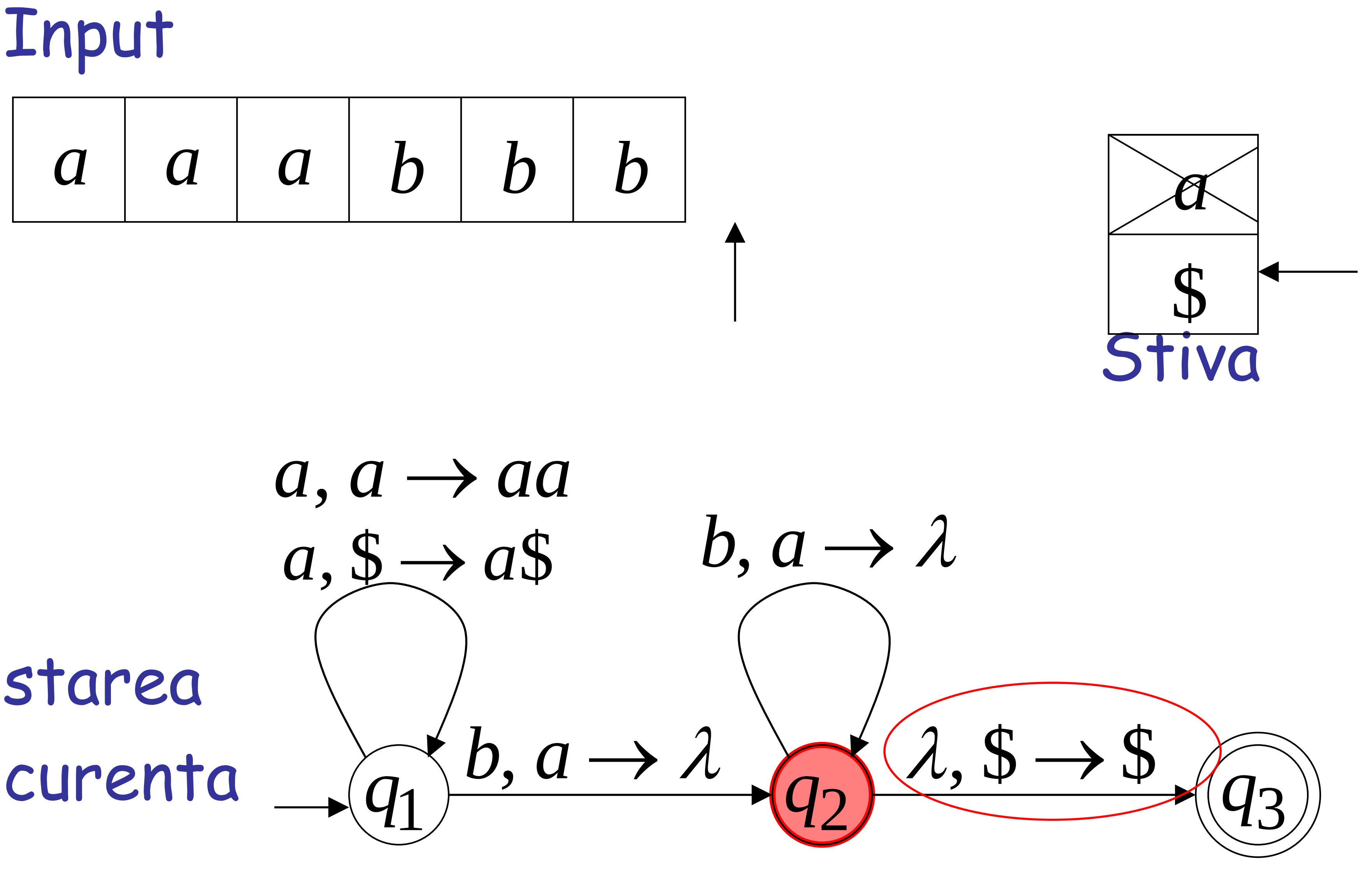

Input
a a a b b b
Stiva
a
a
a
$
starea curenta
$a, a \to aa$
$a, \$ \to a\$$
$b, a \to \lambda$
$b, a \to \lambda$
$\lambda, \$ \to \$$
q_1
q_2
q_3

Timpul 5



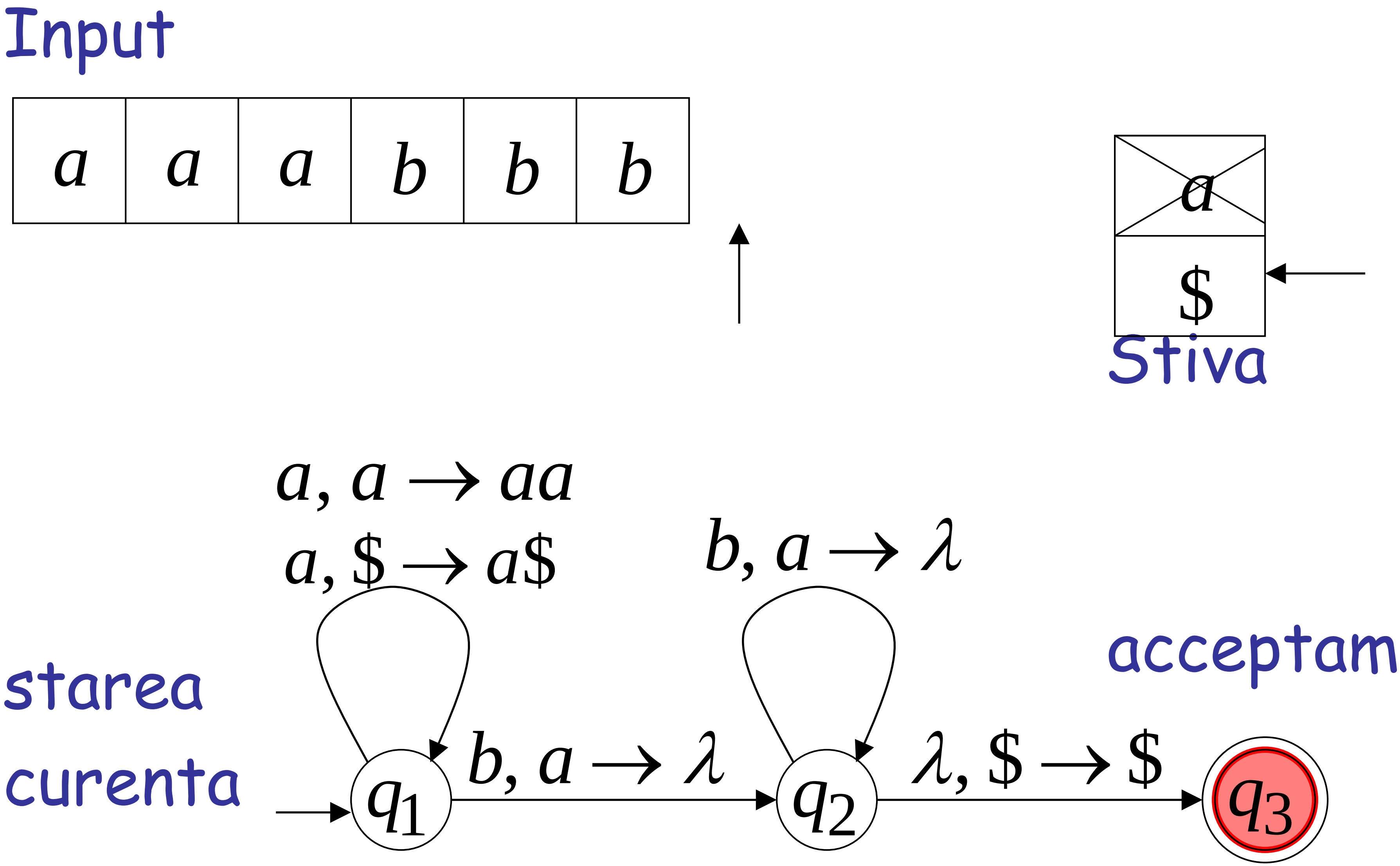
Input
a a a b b b
$\cancel{a}$
a
$\$$
Stiva
starea
curenta
$a, a \rightarrow aa$
$a, \$ \rightarrow a\$$
$b, a \rightarrow \lambda$
$b, a \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
q_1
q_2
q_3

Timpul 6



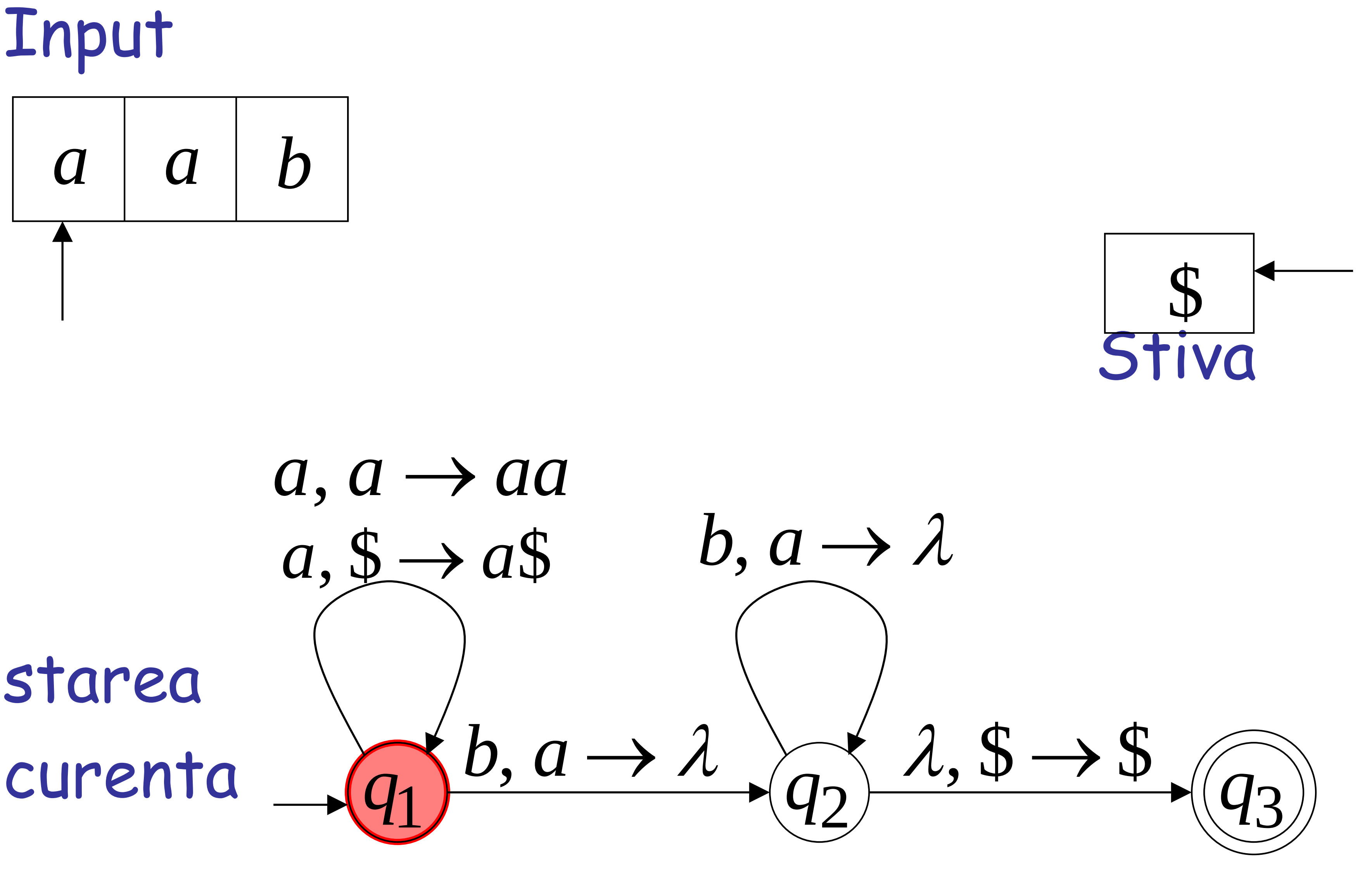
Input
a a a b b b
$\cancel{a}$
$\$$
Stiva
$a, a \rightarrow aa$
$a, \$ \rightarrow a\$$
$b, a \rightarrow \lambda$
$b, a \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
starea
curenta
q_1
q_2
q_3

Timpul 7

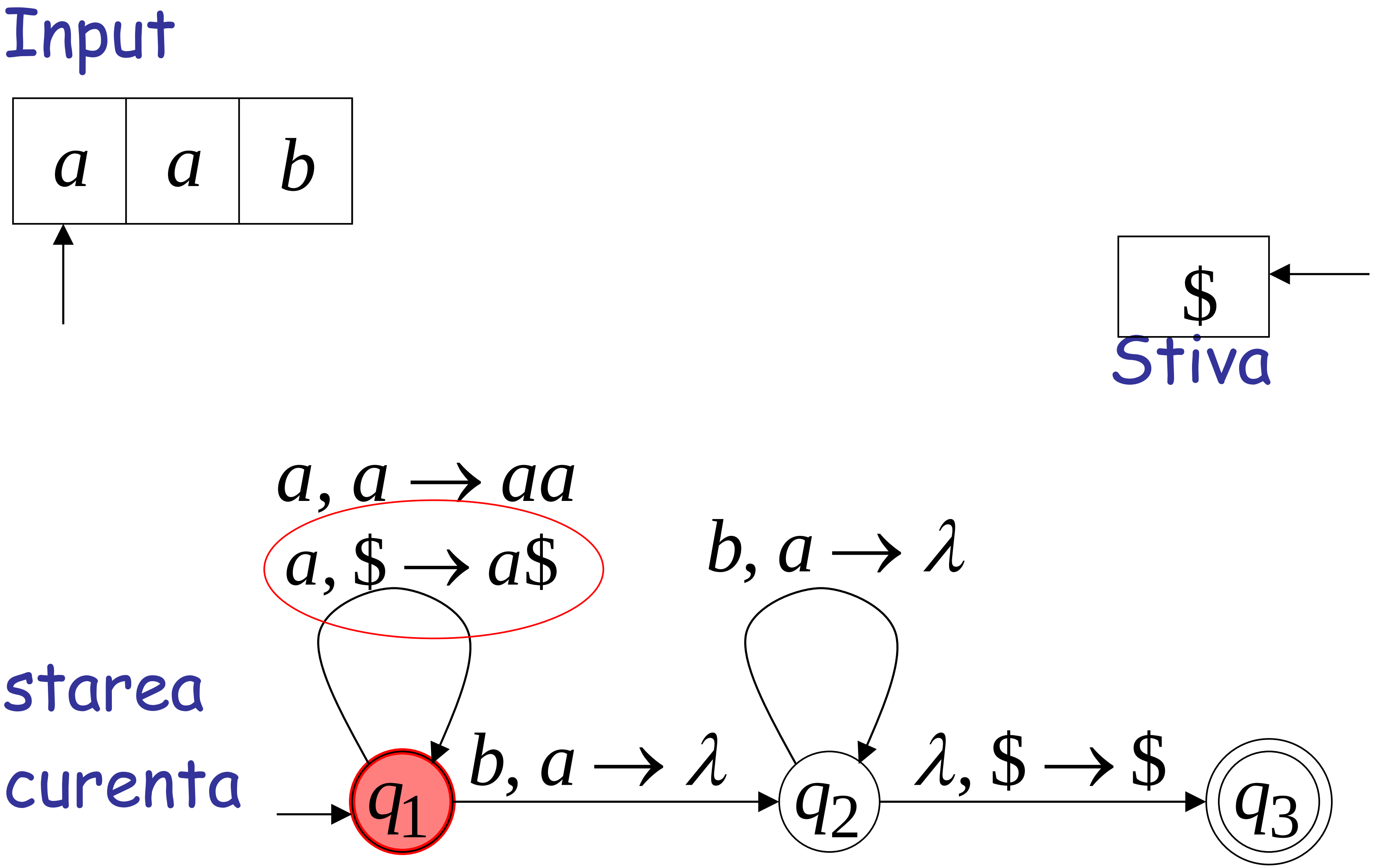


Input
a a a b b b
a
$\$$
Stiva
$a, a \rightarrow aa$
$a, \$ \rightarrow a\$$
$b, a \rightarrow \lambda$
$b, a \rightarrow \lambda$
$\lambda, \$ \rightarrow \$$
starea
curenta
acceptam
q_1
q_2
q_3

Exemplu de Rejectare: Timpul 0

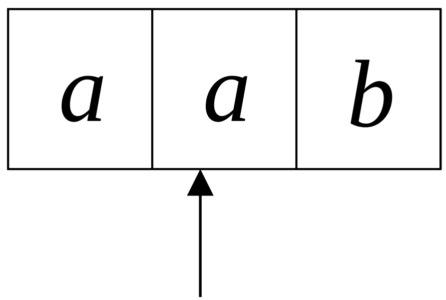


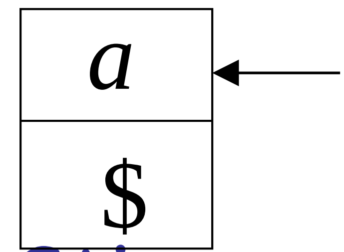
Timpul 0


Input
a
a
b
$
Stiva
$a, a \rightarrow aa$
$a, \$ \rightarrow a\$$
$b, a \rightarrow \lambda$
starea
curenta
q_1
$b, a \rightarrow \lambda$
q_2
$\lambda, \$ \rightarrow \$$
q_3

Timpul 1

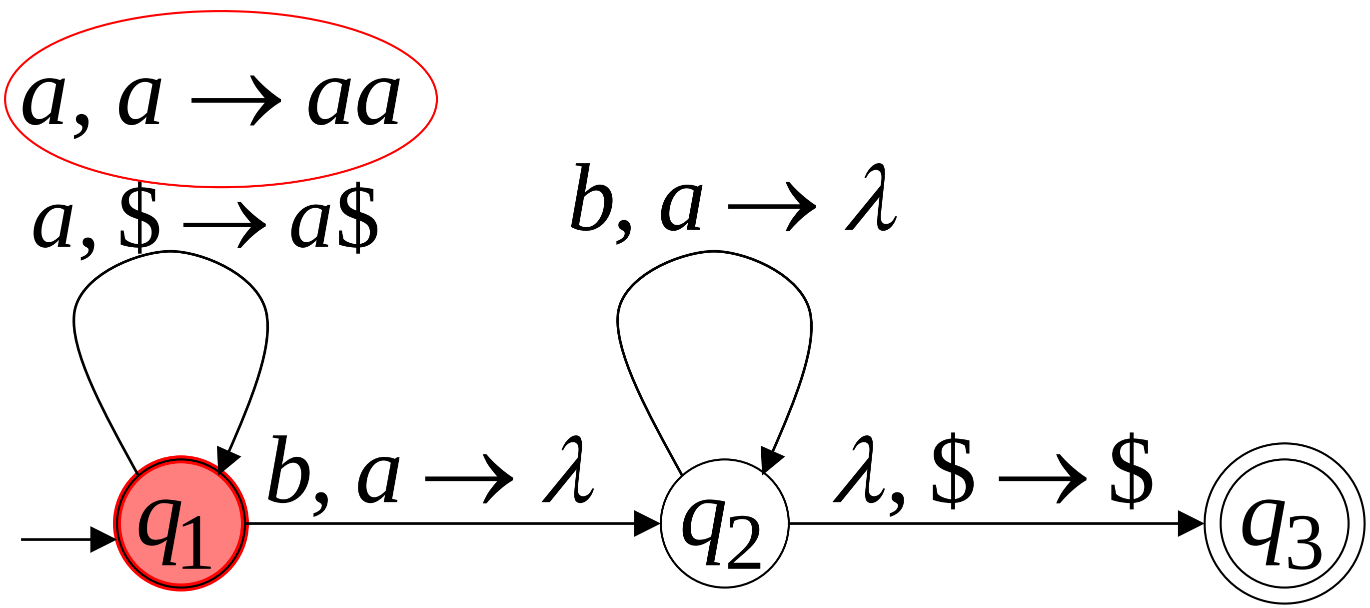
Input



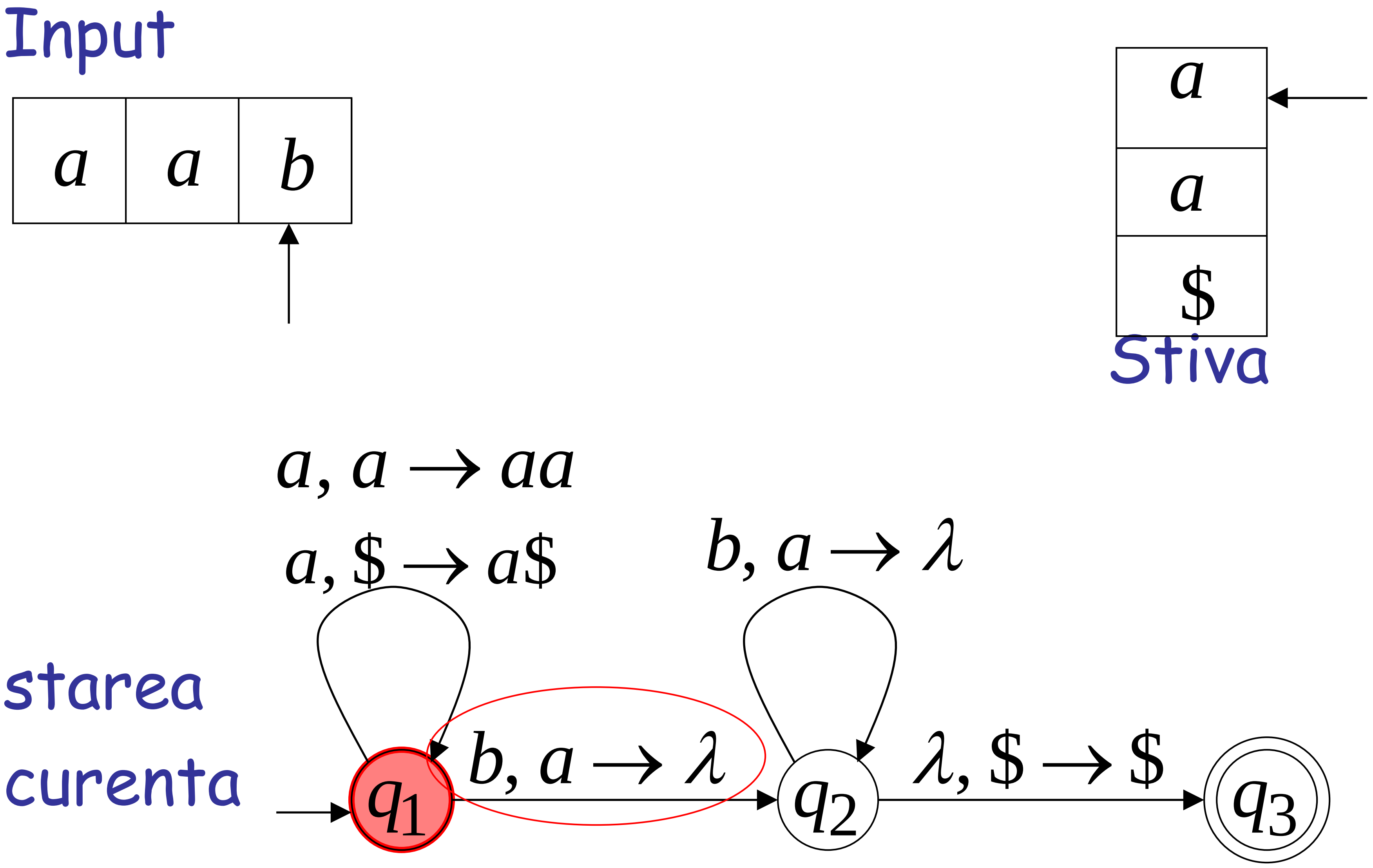


Stiva

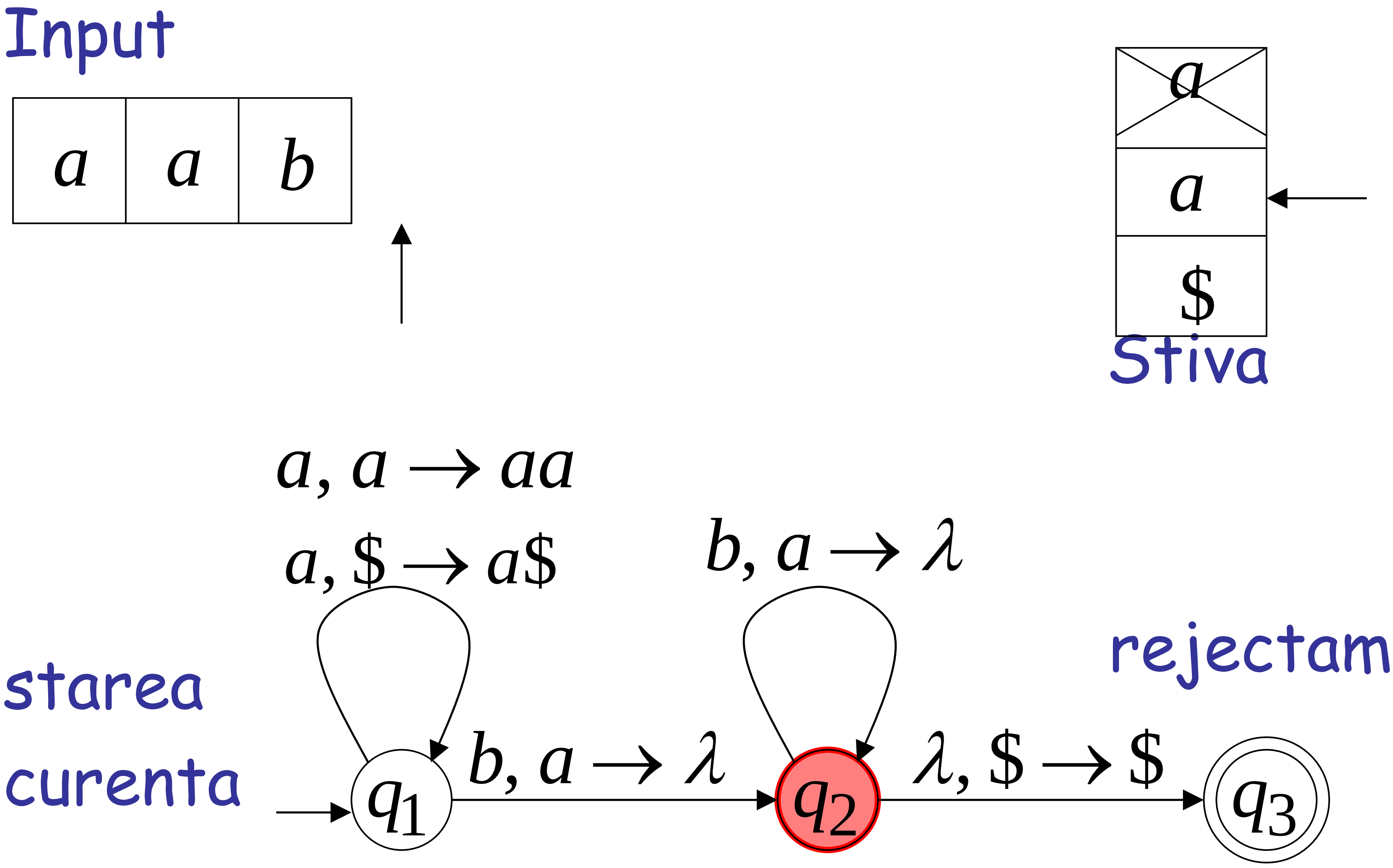
starea curenta



Timpul 2


Input
a a b
a
a
$\$$
Stiva
$a, a \to aa$
$a, \$ \to a\$$
$b, a \to \lambda$
starea curenta
q_1
$b, a \to \lambda$
q_2
$\lambda, \$ \to \$$
q_3

Timpul 3



Input
a
a
b
a
a
$
Stiva
a, a → aa
a, $ → a$
b, a → λ
b, a → λ
λ, $ → $
starea
curenta
q1
q2
q3
rejectam

Exemplu

- $\{ww^r | w \text{ din } \{a+b\}^*\}$
  - pornim intr-o stare s1 care va face o alegere nedeterminista daca am ajuns sau nu la mijlocul cuvantului. Daca nu am ajuns, se salveaza pe stiva literele citite de la intrare
  - din s1 mergem in s2 (in momentul alegerii nedeterministe)
  - in s2 se compara simbolurile de pe stiva cu simbolurile de intrare
  - daca ajungem la sfarsitul cuvantului de intrare si stiva este goala acceptam (mergem intr-o stare finala s3)

Definitie

Automate Pushdown

$$M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$$

Stari

alfabetul de intrare

alfabetul stivei

functia tranzitiilor

starea initiala

simbolul de pe stiva

starile finale

$$\delta(q, a, X) \ni (q', X') \; unde \; q, q' \in Q; a \in \Sigma \cup \{\lambda\}; X \in \Gamma; X' \in \Gamma^*$$

mergem din starea q in q’ citind de la intrare a (care poate sa fie λ) si citind de pe stiva simbolul X. In final ajungem in starea q’ si inlocuim X cu X’ pe stiva

Descriere instantanee

$$(q,\ u,\ s)$$

Starea curenta

cuvantul de intrare ramas

continutul curent al stivei

Modul de operare al PDA

- daca $(p,\alpha) \in \delta(q,a,X)$ definim
  $(q,aw,X\beta) \vdash (p,w,\alpha\beta), unde\ w \in \Sigma^*; \beta \in \Gamma^*$

Limbajele acceptate de PDA

- Acceptare prin stare finala: Se porneste din starea initiala, se accepta daca am ajuns intr-o stare finala (la fel ca in cazul automatelor finite)

$$M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$$

$$T(M) = \{w \mid (q_0, w, z) \vdash^* (q, \lambda, \alpha), q \in F\}$$

- Acceptare prin stiva vida: Se porneste din starea initiala, se accepta daca am ajuns la sfarsitul cuvantului intr-o configuratie cu stiva vida

$$M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$$

$$N(M) = \{w \mid (q_0, w, z) \vdash^* (q, \lambda, \lambda)\}$$

- se poate arata simplu echivalenta dintre cele doua moduri de acceptare

- Acceptare prin stare finala si stiva vida: Se porneste din starea initiala, se accepta daca am ajuns la sfarsitul cuvantului intr-o stare finala si cu stiva vida

$$M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$$

$$L(M) = \{w \mid (q_0, w, z) \vdash^* (q, \lambda, \lambda), q \in F\}$$

Determinism si nedeterminism

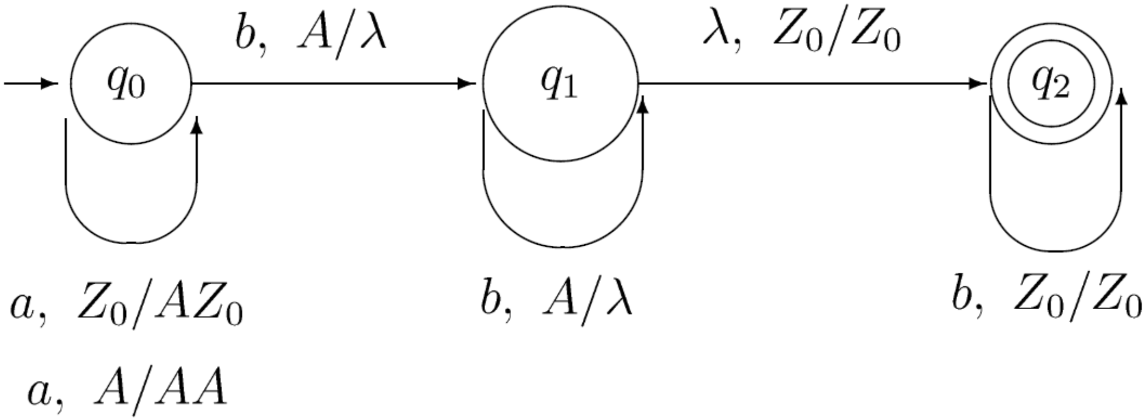
- Definitie: un PDA este deterministic daca
  - $\delta(q, a, X)$ contine cel mult un element pentru orice $q \in Q; a \in \Sigma \cup \{\lambda\}; X \in \Gamma$
  - daca $\delta(q, a, X)$ nu este vid pentru $a \in \Sigma$ atunci $\delta(q, \lambda, X)$ este vid

observatii pentru DPDA

- DPDA accepta λ-tranzitii
- fiecare tranzitie este determinata de catre starea curenta, simbolul de intrare curent si simbolul din capul stivei
- pentru o pereche stare-simbol de intrare putem sa avem o multitudine de tranzitii diferentiate de simbolul de pe stiva

Proprietati pentru DPDA

- daca avem T_{DPDA}, N_{DPDA}, L_{DPDA} fiind familiile de limbaje acceptate de DPDA prin “stare finala”, “stiva vida” si “stare finala si stiva vida”, respectiv
- avem $N_{DPDA} = L_{DPDA}$ inclus in T_{DPDA}



- exemplu: $L=\{a^m b^n | m \leq n,\ n>0\}$

- $L=T(A)$, dar L nu apartine lui N_{DPDA}

CFL deterministe

- Definitie: limbajele independente de context deterministe sunt toate limbajele acceptate de DPDA prin modul de acceptare “stare finala”

- DCFLs sunt incluse strict in CFLs
- exemple: $L=\{a^n b^n | 0 \leq n\} U \{a^n b^{2n} | 0 \leq n\}$
- $\{ww^r | w \text{ din } \{a+b\}^*\}$

Relatia cu limbajele independente de context

- PDA (nedeterministe) sunt echivalente cu CFG
- se demonstreaza ca pentru orice gramatica G se poate construi un PDA A care accepta limbajul generat de G
- si apoi se demonstreaza ca pentru orice PDA A se poate construi o gramatica G care genereaza toate cuvintele acceptate de automatul A

L(CFG) inclus in L(PDA)

- fie G o gramatica independenta de context
- construim automatul A cu 3 stari si tranzitiile
  - $\{(q_1, Sz)\} = \delta(q_0, \lambda, z)$
  - $(q_1, Y) \in \delta(q_1, \lambda, A)$ pentru orice tranzitie $A \to Y$
  - $\{(q_1, \lambda)\} = \delta(q_1, a, a)$
  - $\{(q_{\mathrm{accept}}, \lambda)\} = \delta(q_1, \lambda, z)$

DCFL proprietati de inchidere

- DCFL sunt inchise la
  - complementare
  - intersectie cu limbaje regulate
- DCFL nu sunt inchise la
  - reuniune
  - intersectie

Sumar

- PDA: un λ-NFA cu stiva
- Acceptare: stare finala, stiva vida, 1&2
- modurile de acceptare sunt echivalente pentru PDA
- PDA sunt echivalente cu CFG
- DPDA sunt strict incluse in PDA
- modurile de acceptare nu sunt echivalente pentru DPDA
- REG inclus in DPDA inclus in CFG