Automaty, jazyky, zložitosť

Ing. Michal Hospodár, PhD.

Burza bakalárskych prác

8. apríl 2026

Čo je automat

- Automat znamená samočinné zariadenie (z gréčtiny)
- Je to (formálny) stroj, ktorý vykonáva vopred definovanú funkciu
- Funkcia = prevod zo známych vstupov na výstupy závislé od nich
- Funkcia automatu:
  vstup = slovo (postupnosť písmen),
  výstup = rozhodnutie, či slovo patrí do jazyka
- Množina slov = jazyk

Formálne jazyky

- Formálny jazyk = jazyk, ktorý je možné formálne zapísať
- Konečný jazyk je možné zapísať vymenovaním slov, nekonečný nie
- Existuje konečný zápis nekonečného jazyka? Áno!
- Formálny zápis jazyka má nejaké pravidlá
= model (automat, gramatika)
- Veľkosť najmenšieho modelu popisujúceho jazyk
= **zložitosť jazyka**

Príklady

- Povedzme, že máme k dispozícii dva **symboly** {a,b}. Takú množinu nazývame **abeceda**.
- Z týchto symbolov vieme povytvárať ľubovoľné postupnosti (reťazce), napr. aaa, abba, bbbaabba, aabbbaa atď... Tie reťazce sa nazývajú **slová**.
- Keď vytvoríme nejakú množinu slov, tak hovoríme že máme **jazyk** nad abecedou {a,b}.

Príklady jazykov

1) množina všetkých slov

$$L_1 = \{a, b\}^*$$

2) najprv je párny počet symbolov *a*, za ktorými nasledujú už len symboly *b*

$$L_2 = \{w \in \{a, b\}^* | (\exists n, m \in N) w = a^n b^m, n \equiv 0 (\mathrm{mod}\ 2)\}$$

3) slová obsahujúce len písmeno *a* také, ktorých dĺžka je prvočíslo

$L_3 = \{w \in \{a\}^* | (\exists n \in P) w = a^n\}$, kde P je množina prvočísel

ZOVŠEOBECNENIE

- **ABECEDA** je ľubovoľná konečná množina.
- **SYMBOL** je prvok abecedy.
- **SLOVO** je ľubovoľná konečná postupnosť symbolov z abecedy. Pripúšťa sa aj prázdna postupnosť, vtedy sa slovo nazýva **prázdne**. Označuje sa zvyčajne ε.
- **JAZYK** je ľubovoľná množina slov z abecedy.

Príklady z bežnej praxe.

1) Nech ∑ je abeceda všetkých symbolov slovenskej abecedy. Teda ∑={A,Á,Ä,B,C,Č,D,Ď,E,É,...,Z,Ž}. Potom zrejme existuje jazyk, ktorý je zhodný so slovnou zásobou **slovenského jazyka**.
2) Nech ∑ z predchádzajúceho bodu je obohatená o symboly {‘ ‘ (medzera), ‘,‘ (čiarka), ‘.‘(bodka) }. Teraz už vieme nájsť aj taký jazyk, ktorý bude obsahovať všetky **slovenské texty**.
3) Nech ∑={0,1,2,3,4,5,6,7,8,9,+,=}. Zrejme existuje jazyk, kde slová sú správne určené **súčty prirodzených čísel**.
4) Nech ∑={0,1}. Potom zrejme existuje jazyk, kde sú slová reprezentujúce všetky **čísla v binárnom tvare deliteľné číslom 3**.

Ďalšie označenia ...

- Označme abecedu $\sum$.
- Označenie $\sum^*$ reprezentuje množinu všetkých možných slov, ktoré je možné vytvoriť zo symbolov abecedy $\sum$.
- Označenie $\sum^+$ reprezentuje množinu všetkých možných slov, ktoré je možné vytvoriť zo symbolov abecedy $\sum$ **okrem** ε.
- Označenie $\sum^k$ reprezentuje množinu všetkých slov dĺžky *k*.
- Zrejme platí nasledujúci vzťah:

$$\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i$$

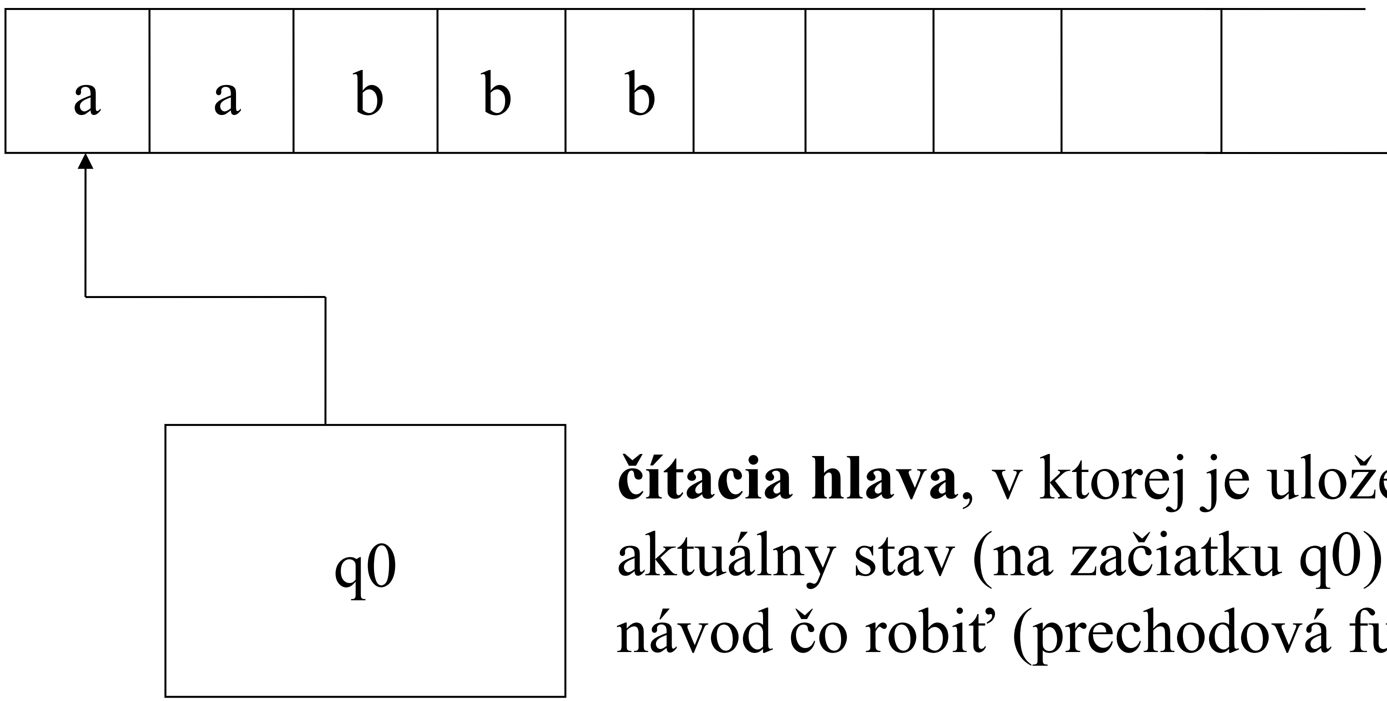
KONEČNÝ AUTOMAT

je to jednoduché zariadenie schopné čítať slovo z nejakej pásky, postupne symbol po symbole (po prečítaní symbolu sa posunie na ďalší symbol doprava), meniť svoj stav v závislosti od načítaného symbolu a aktuálneho stavu, v ktorom sa zariadenie nachádza

Predstava, čo má robiť automat …

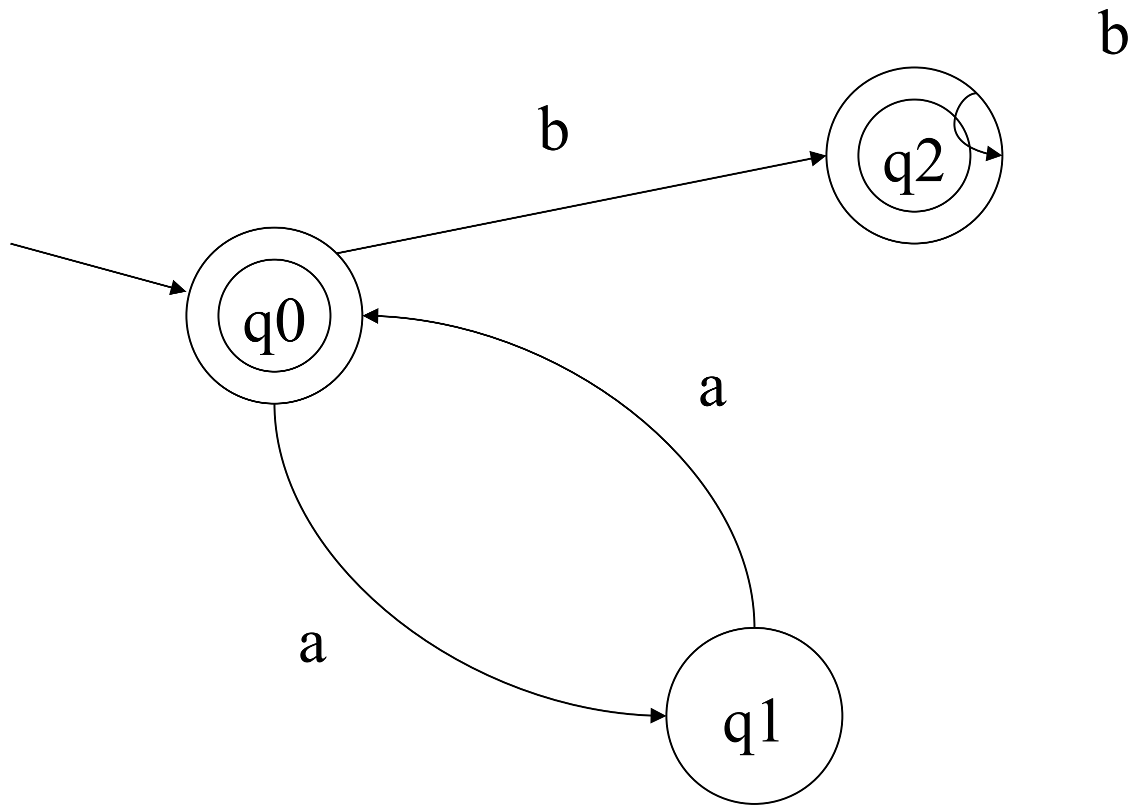
„Čierna skrinka“

páska, kde je napísané akési slovo
(políčka (**bunky**) idú do nekonečna)



čítacia hlava, v ktorej je uložený
aktuálny stav (na začiatku q0) a
návod čo robiť (prechodová funkcia)

schéma práce nejakého automatu



OTÁZKA: Ktorý jazyk popisuje ? Mali sme už taký jazyk uvedený?

FORMÁLNY POPIS AUTOMATU

AUTOMAT ***A*** je definovaný ako pätica:

$$\left(Q, \Sigma, \delta, q_0, F\right)$$

- **Q** je **množina stavov**
- $\sum$ je **vstupná abeceda**
- δ je **prechodová funkcia**
- $\mathbf{q_0}$ je **počiatočný stav**
- ***F*** je **množina koncových stavov**

PRECHODOVÁ FUNKCIA:

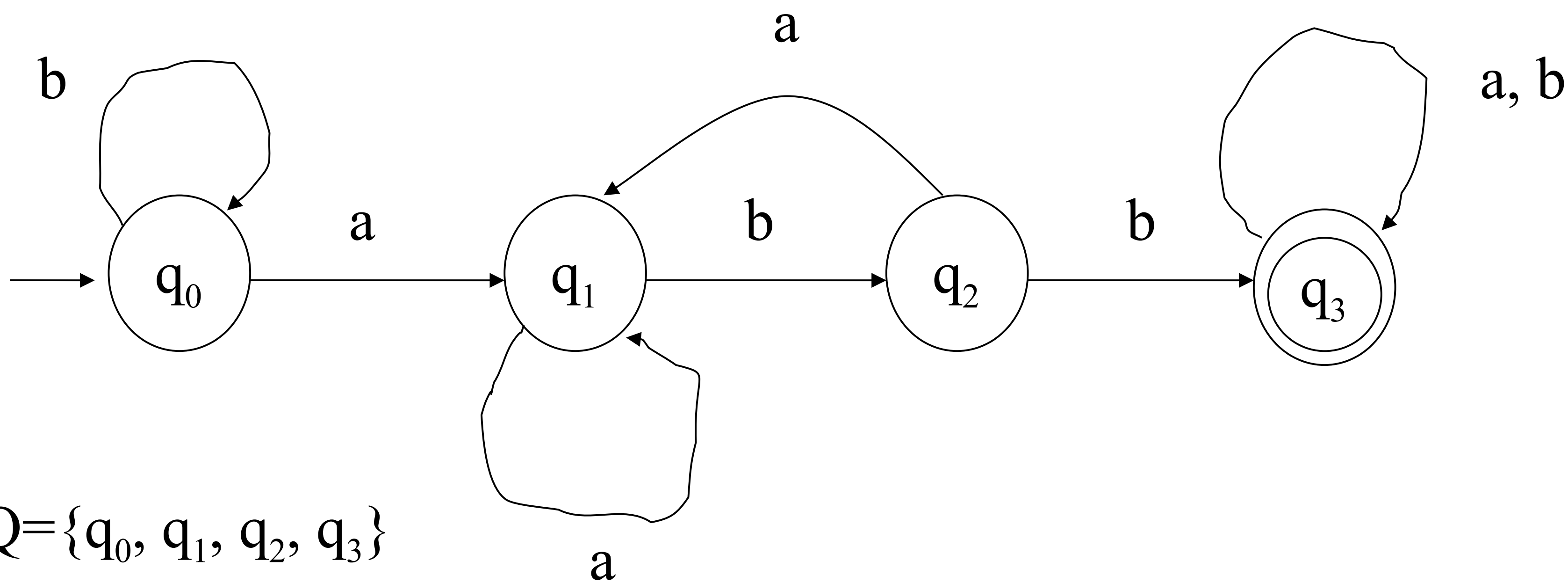
$$\delta : Q \times \Sigma \to Q$$

Môže byť zadaná:

- predpisom
- tabuľkou
- grafom

Zoberme si nasledujúci jazyk a skonštruujme pre neho automat:

$$L_6 = \{ w \in \{a,b\}^* \mid (\exists u, v \in \{a,b\}^*)\ w = uabbv \}$$



- $Q=\{q_0, q_1, q_2, q_3\}$
- $\sum=\{a, b\}$

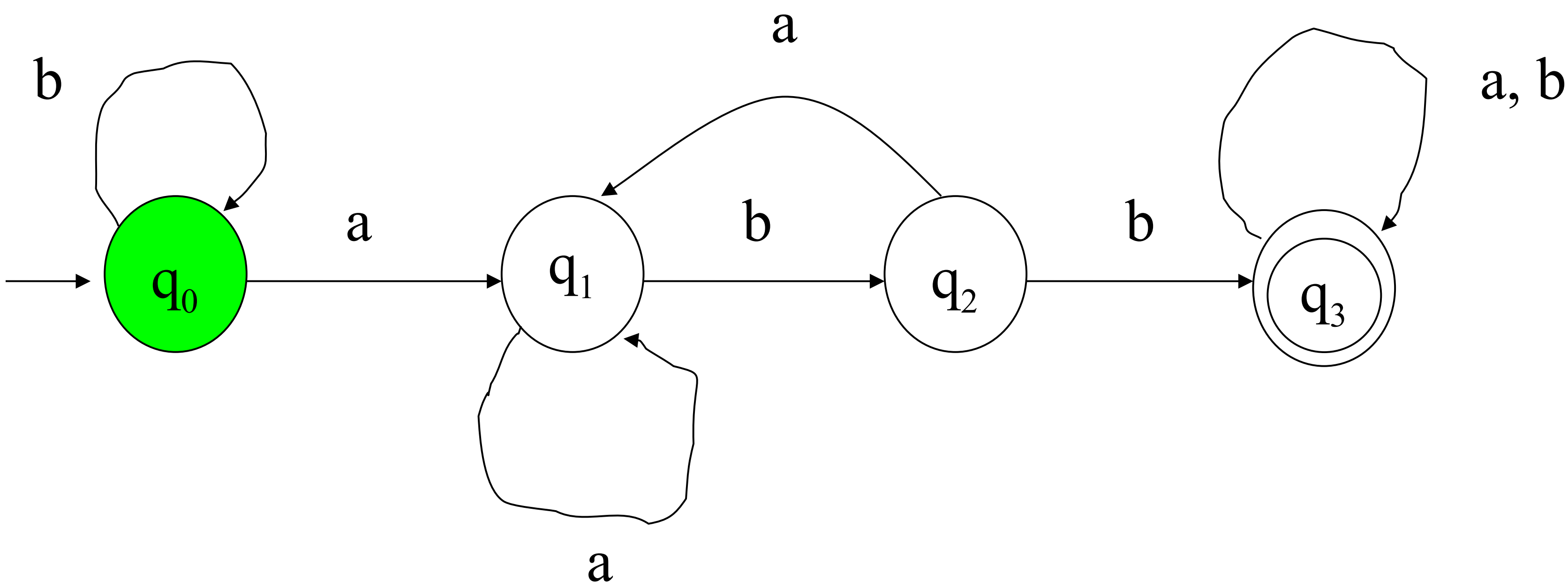
δ je zadaná grafom

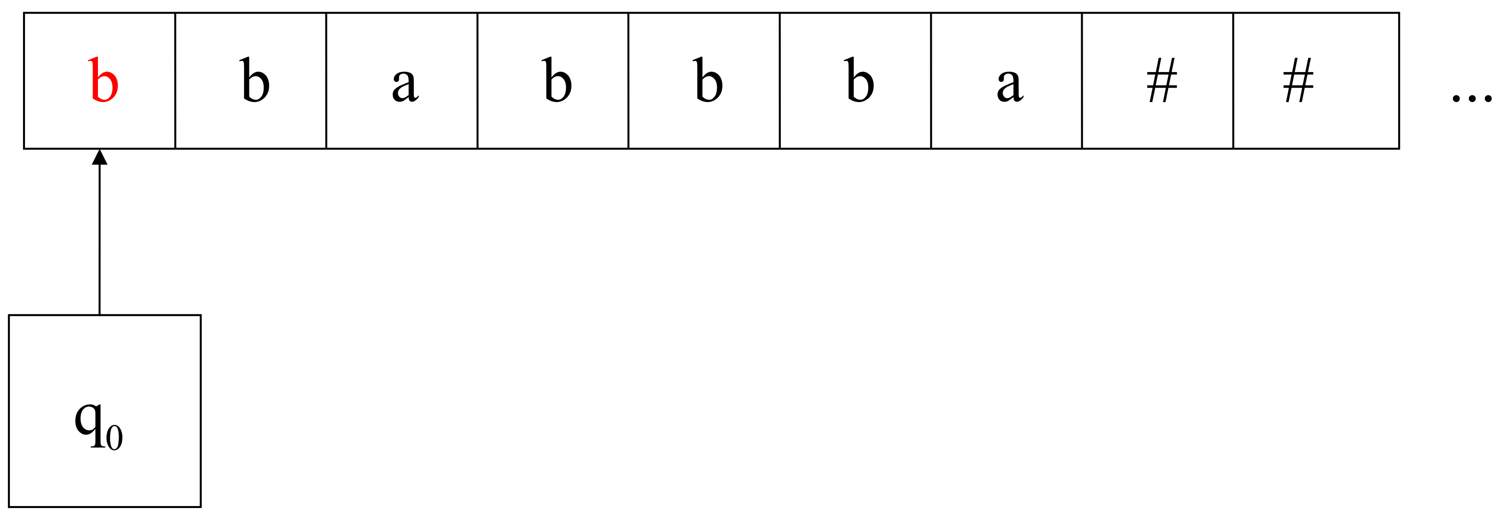
- q_0 počiatočný stav
- $F= \{q_3\}$

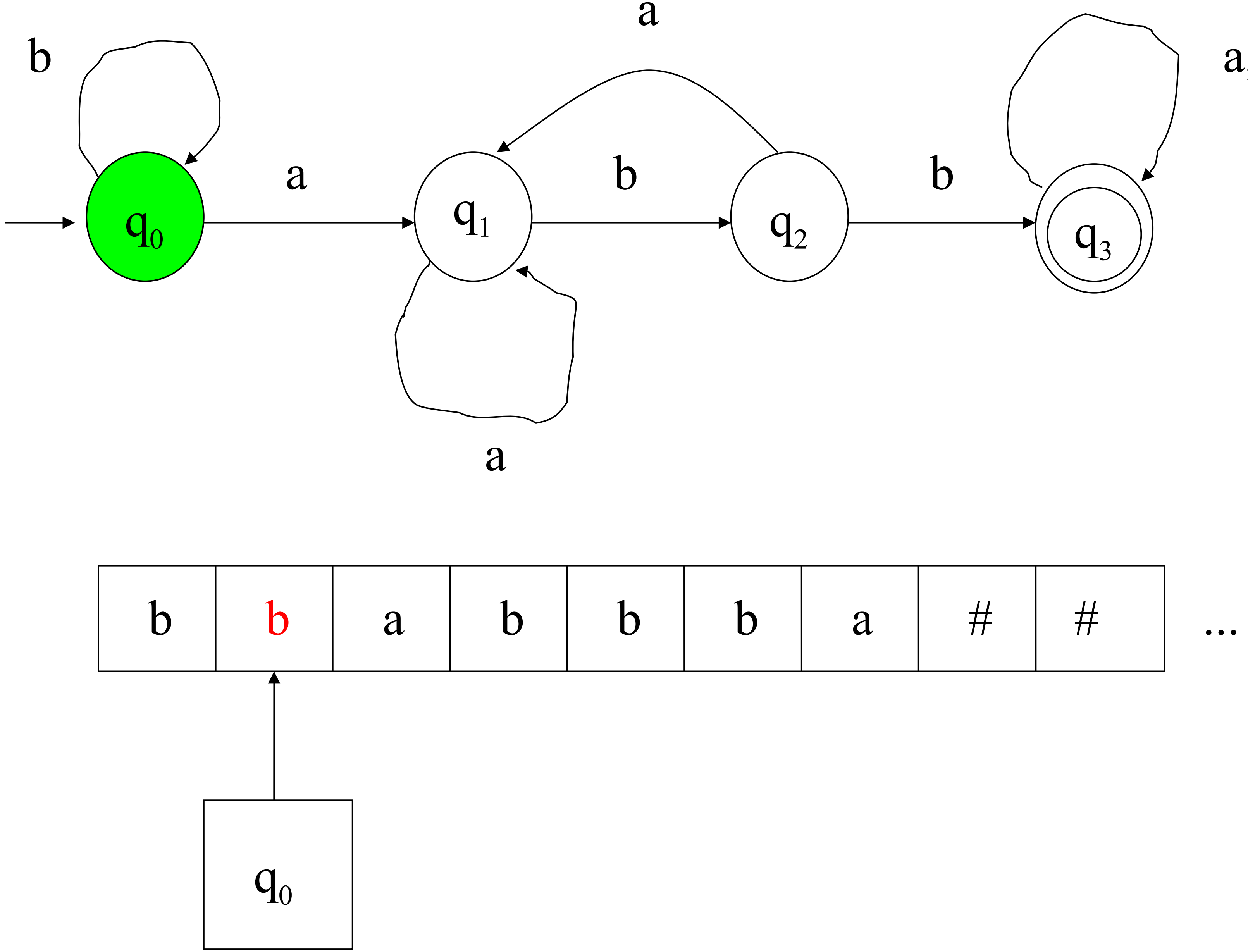
Nasledujúcich niekoľko sekvencií názorne **ilustruje prácu** konečnostavového automatu.

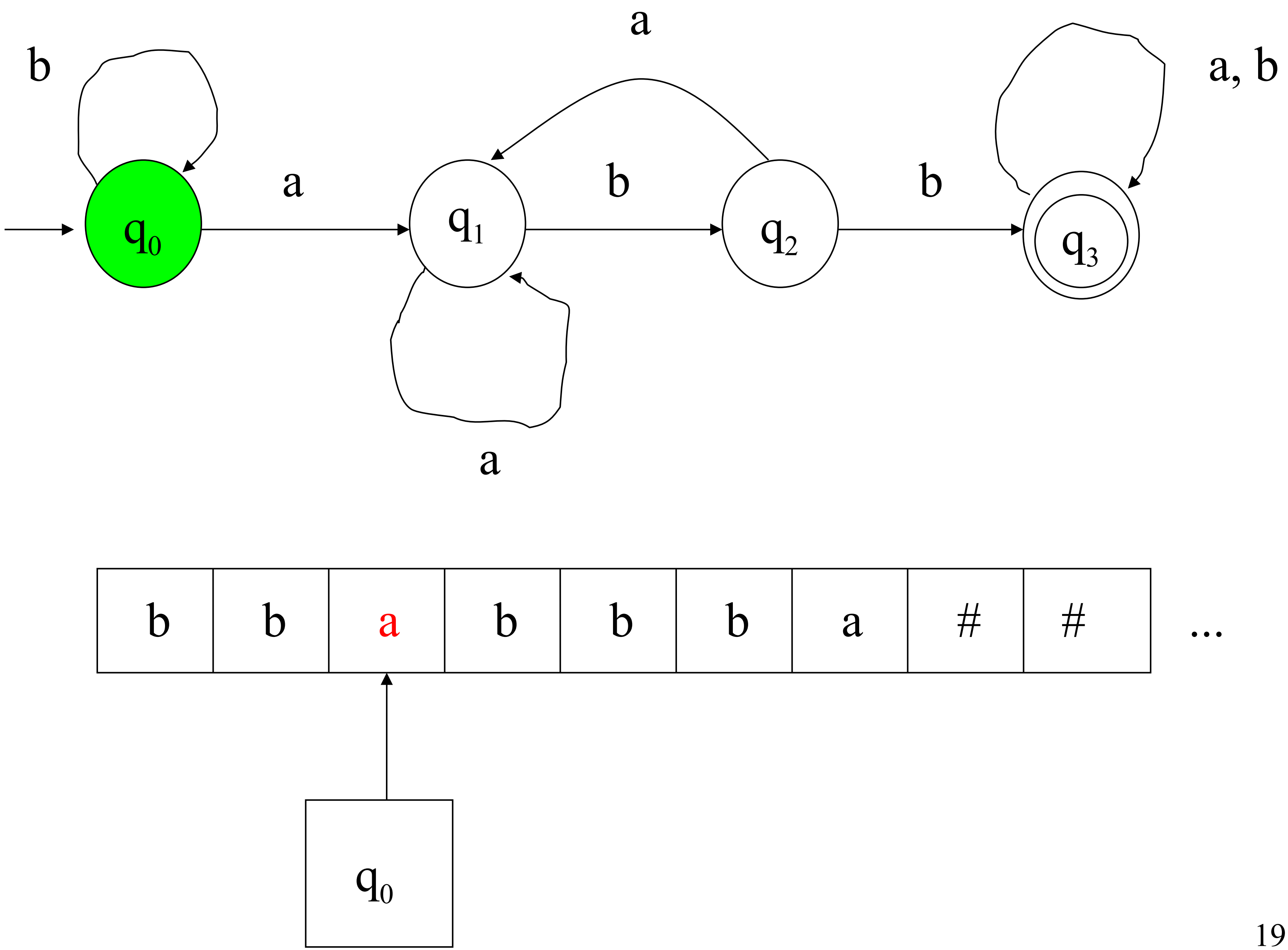
V hornej časti snímky je nakreselná **prechodová funkcia** (algoritmus) v podobe grafu. Vrcholy grafu reprezentujú stavy automatu v ktorých sa môže nachádzať. Stav zafarbený na zeleno je aktuálny stav čítacej hlavy v kotrom sa práve nachádza.

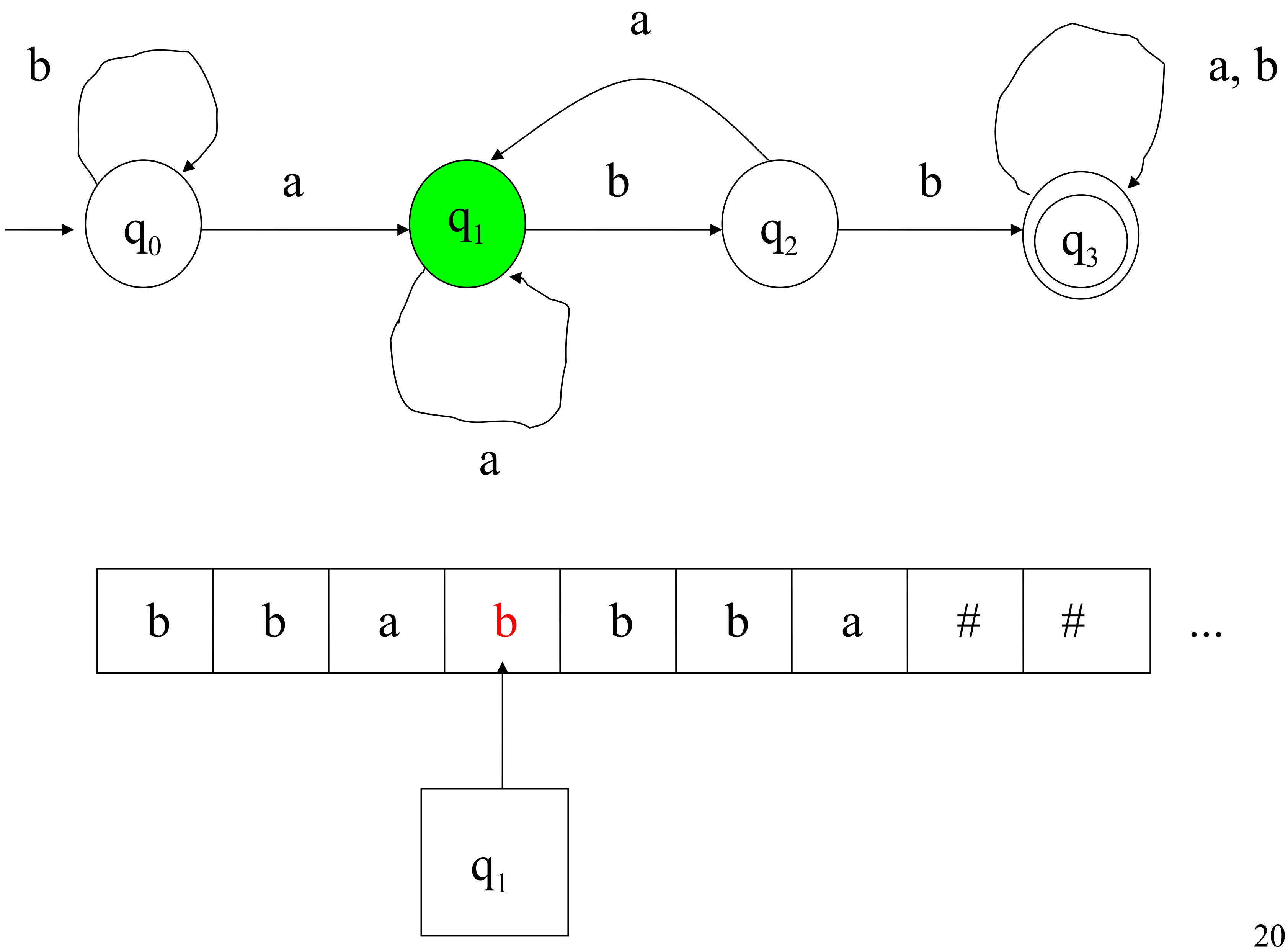
V dolnej časti snímky je nakreslená **páska** na ktorej je uložené nejaké vstupné slovo (v našom prípade je to *bbabbba*), tak že každý symbol slova je uložený zvlášť na políčku (bunke) pásky. Symbol zafarbený na červeno je aktuálny symbol, ktorý práve čítacia hlava číta.


b
a
a
b
a, b
b
q_0
q_1
q_2
q_3
a

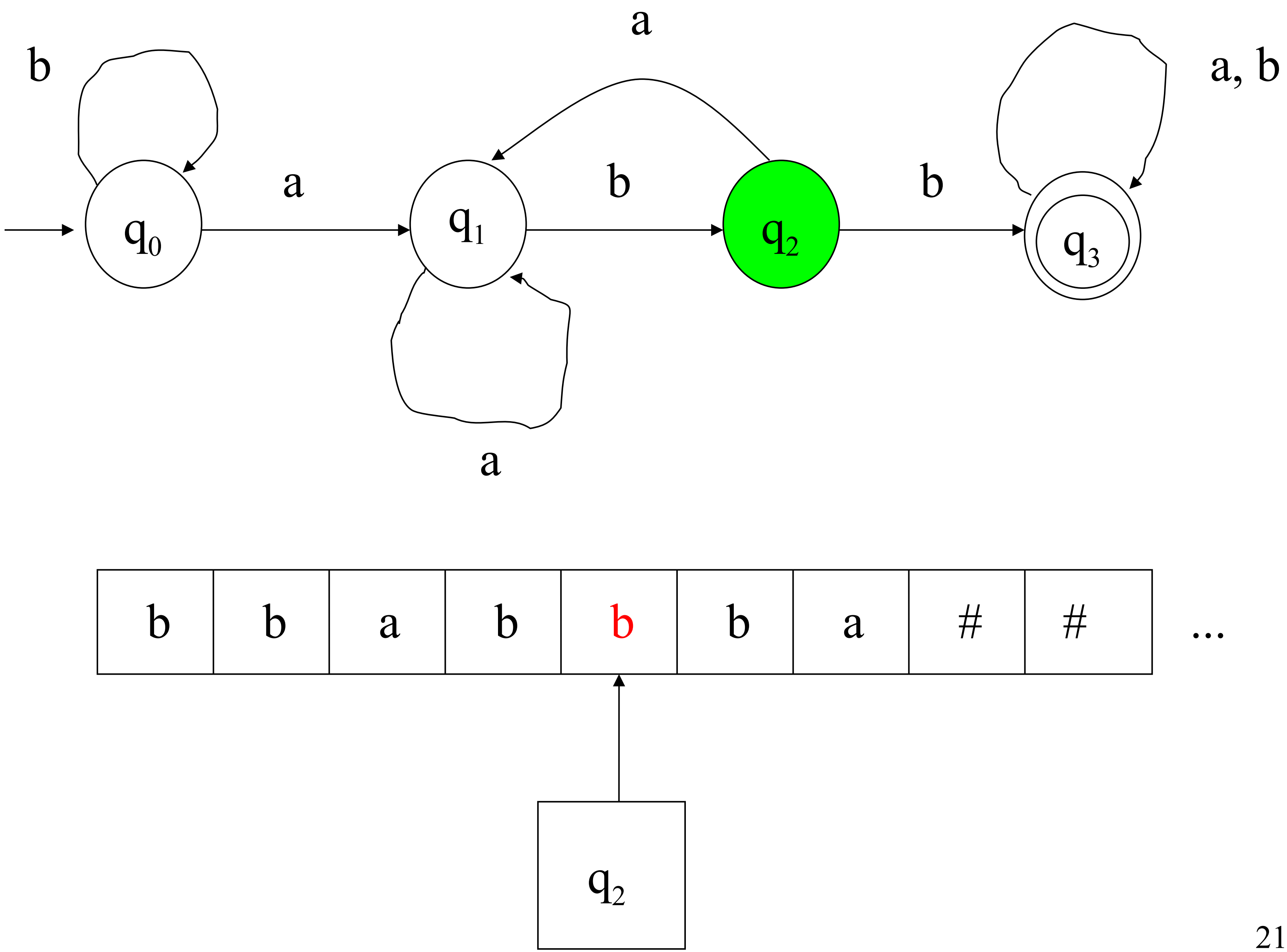

b b a b b b a # # ...
q_0

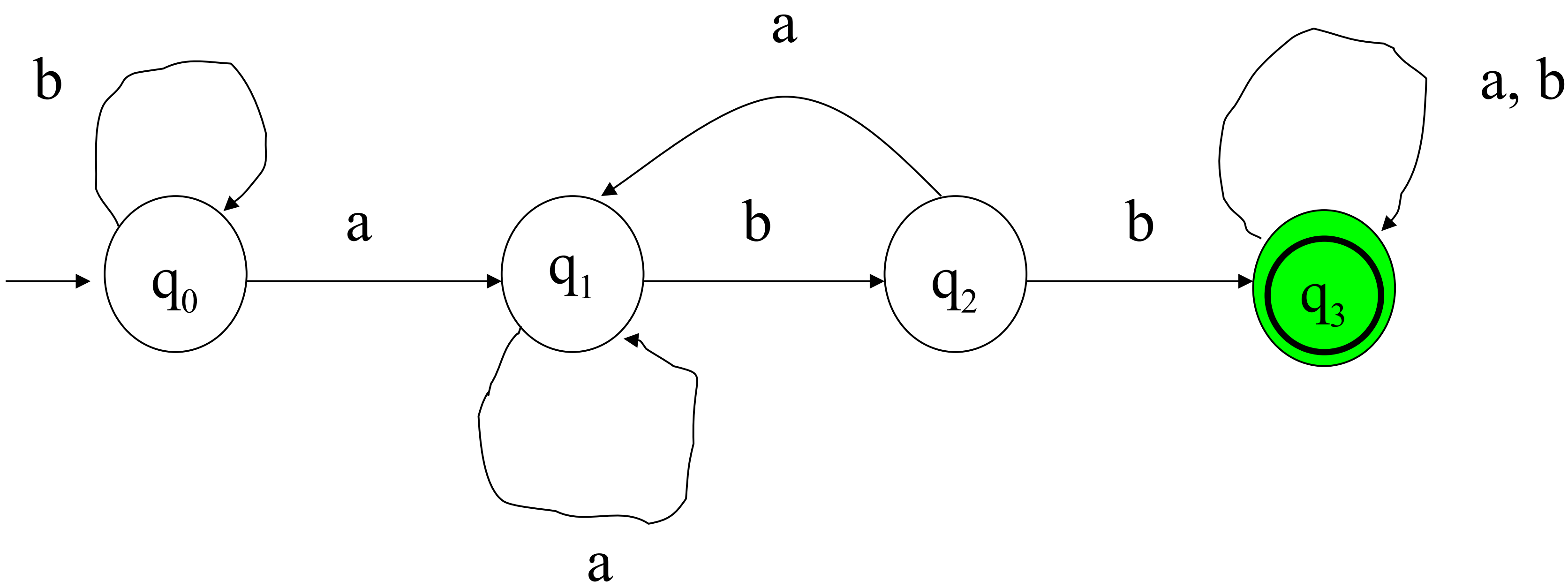

b
a
a
a, b
q_0
q_1
q_2
q_3
a
b
b
a
b b a b b b a # # ...
q_0

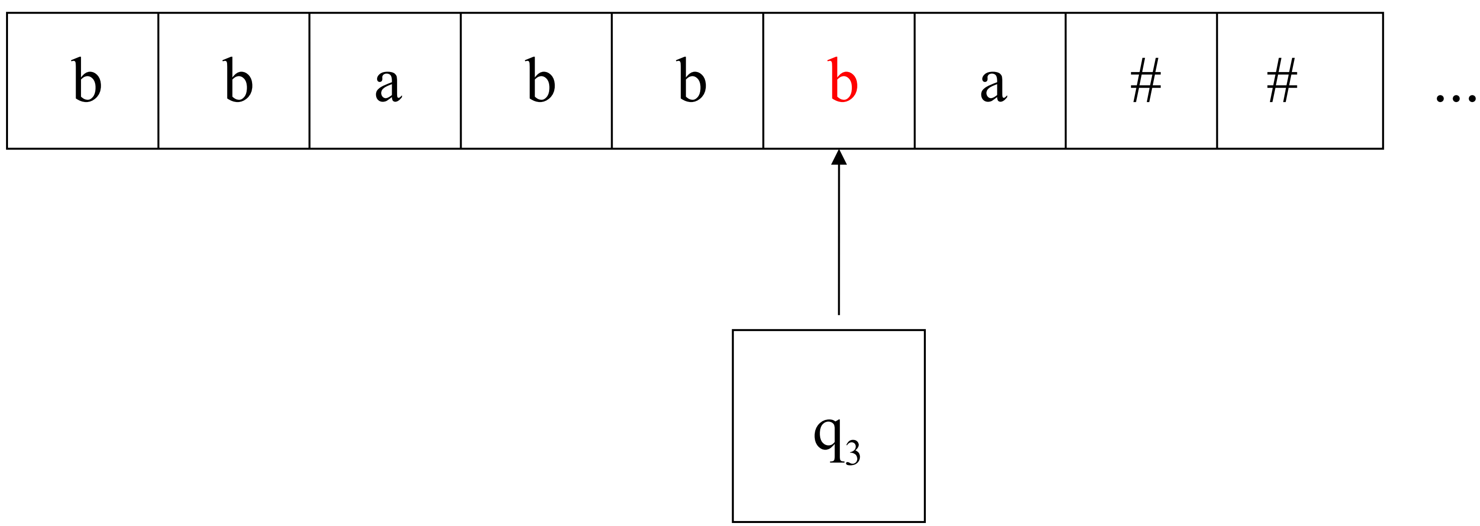

b
a
a
b
b
a, b
q0
q1
q2
q3
a
b b a b b b a # # ...
q0

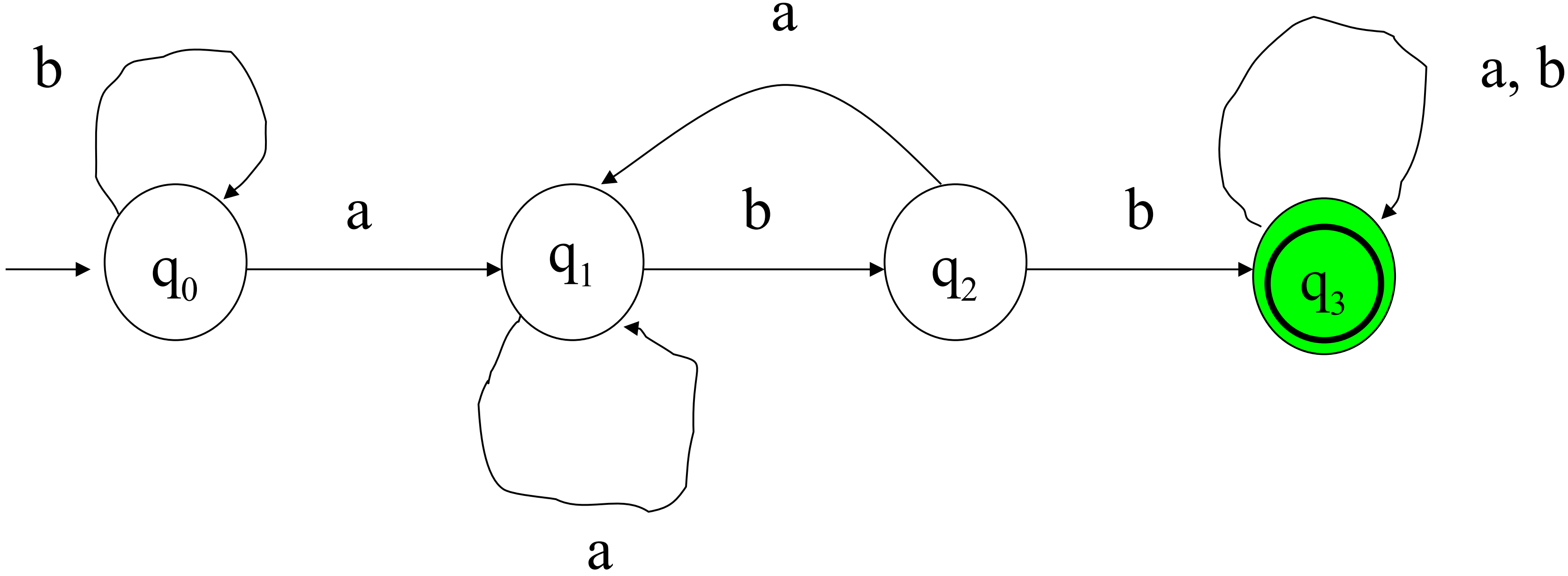


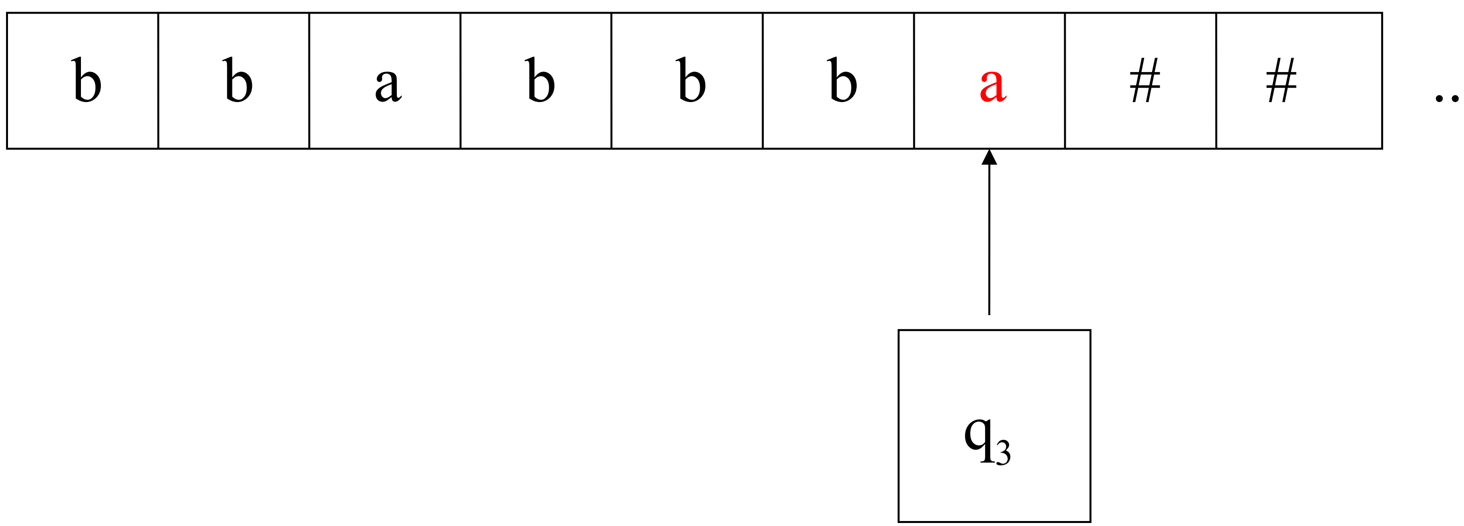
b
a
a
b
b
a, b
a
q0
q1
q2
q3
b b a b b b a # # ...
q1

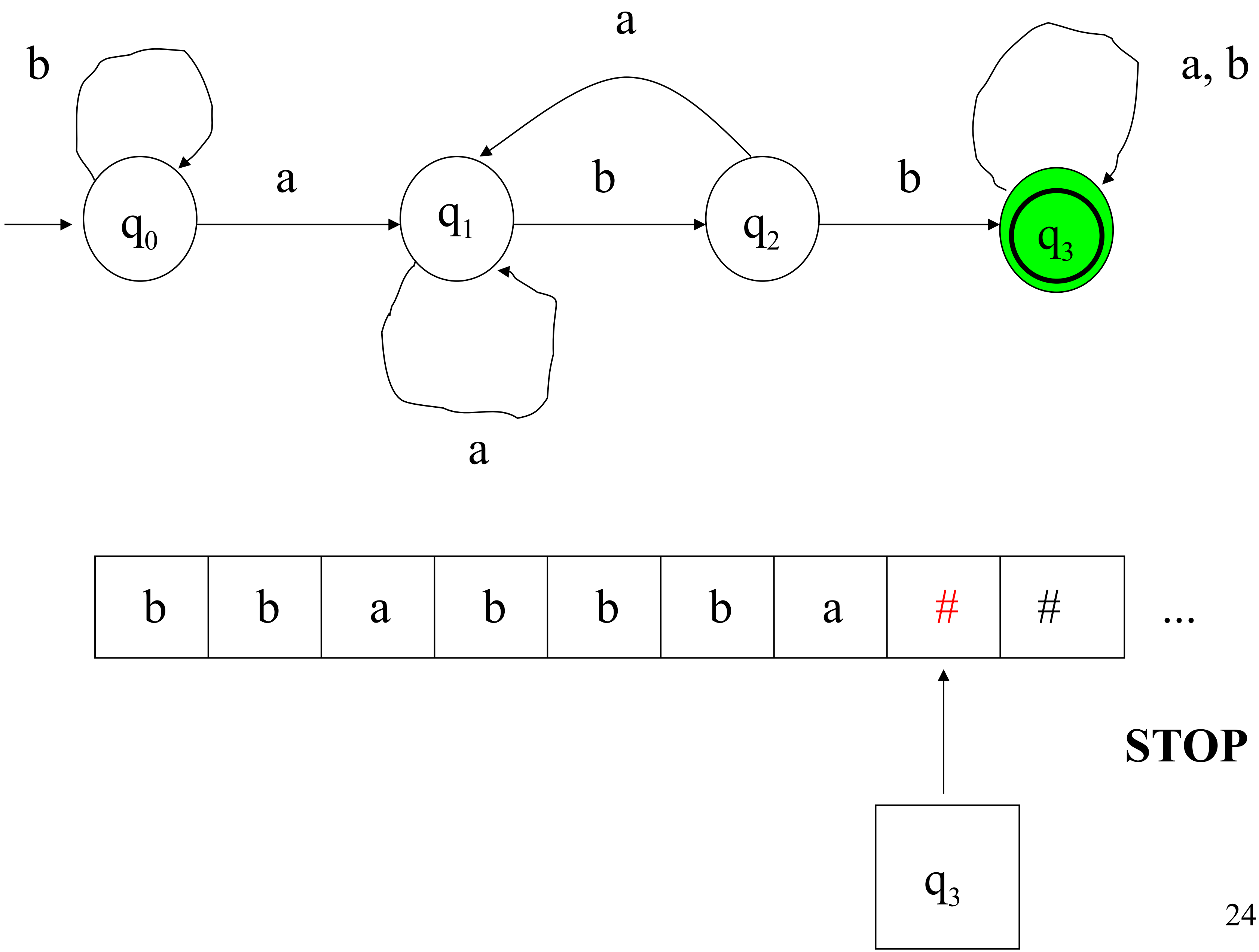

b
a
a
b
b
a, b
q0
q1
q2
q3
a
b
b
a
b
b
b
a
#
#
...
q2


b
a
a
b
b
a, b
a
q_0
q_1
q_2
q_3


b b a b b b a # # …
q_3


b
a
a
a, b
b
b
q_0
q_1
q_2
q_3
a


b b a b b b a # # …
q_3



b
a
a
b
b
a, b
q0
q1
q2
q3
a
b
b
a
b
b
b
a
#
#
...
q3
STOP

Rozšírená prechodová funkcia δ^* je definovaná nasledovne:

Základný krok:

$$\left(\forall q \in Q, \forall a \in \Sigma\right) \delta^*\left(q, \varepsilon\right) = q$$

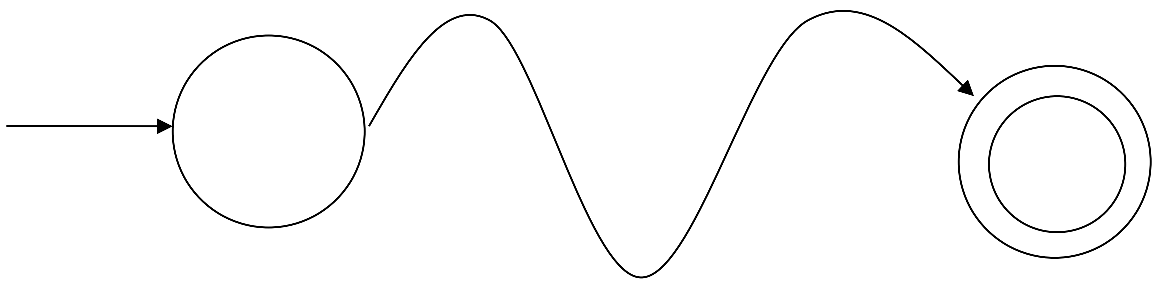
Indukčný krok:

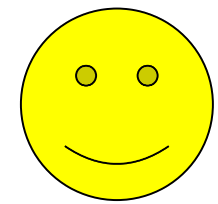
$$\left(\forall q \in Q, \forall a \in \Sigma, \forall w \in \Sigma^n\right) \delta^*\left(q, wa\right) = \delta\left(\delta^*\left(q, w\right), a\right)$$

Jazyk rozpoznávaný automatom:

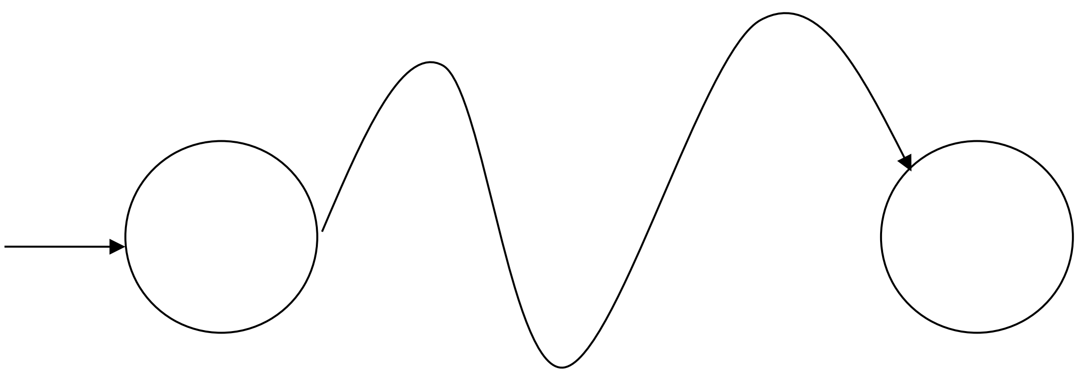
$$L(\mathrm{A}) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \in F\}$$

$$w \in L(\mathrm{A})$$





Koncový stav



Nekoncový stav

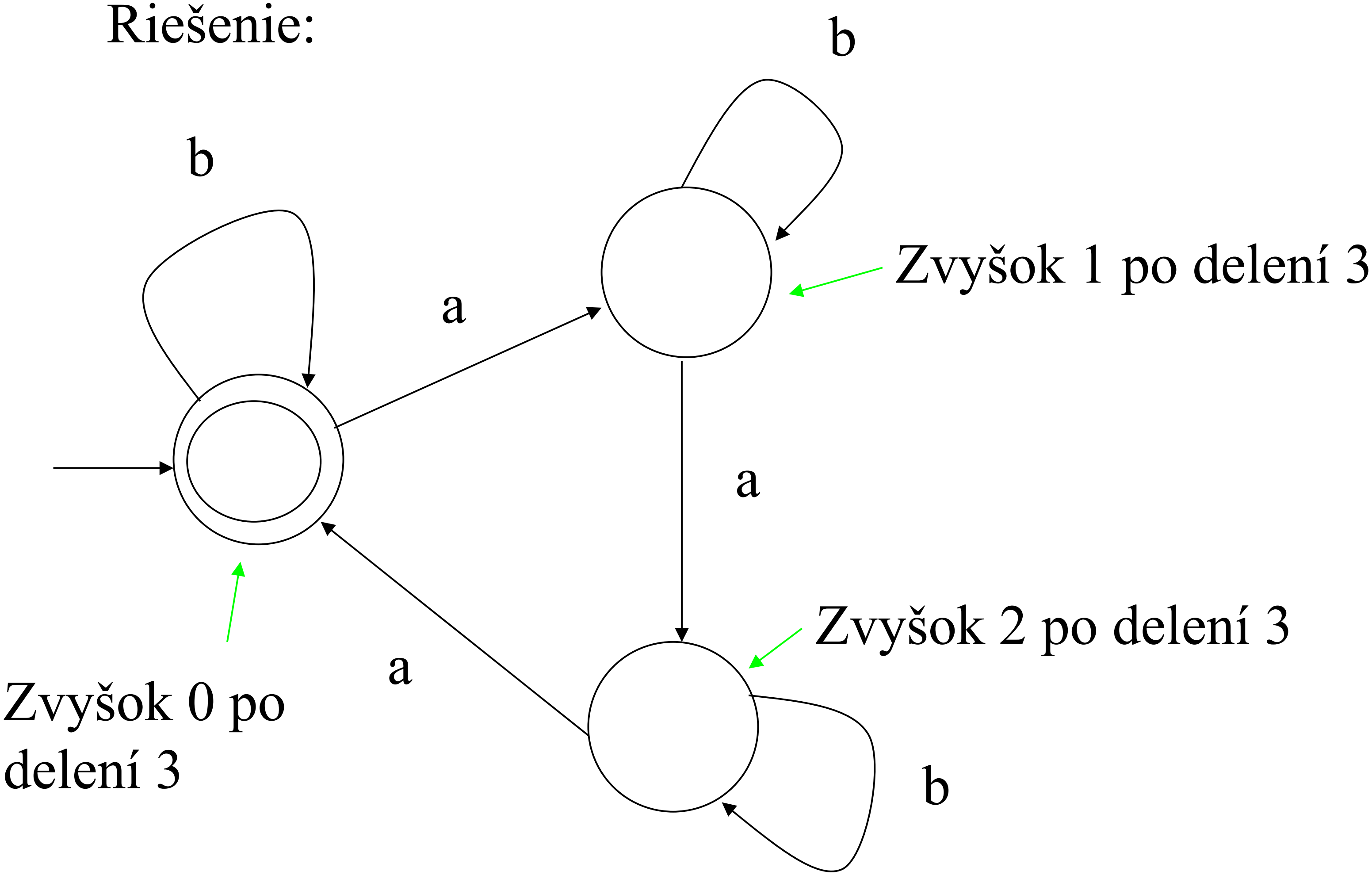
$$w \notin L(\mathrm{A})$$

Niekoľko úloh ...

$$L_4 = \{w \in \{a,b\}^* | \#_a(w) \equiv 0 (\text{mod } 3)\}$$

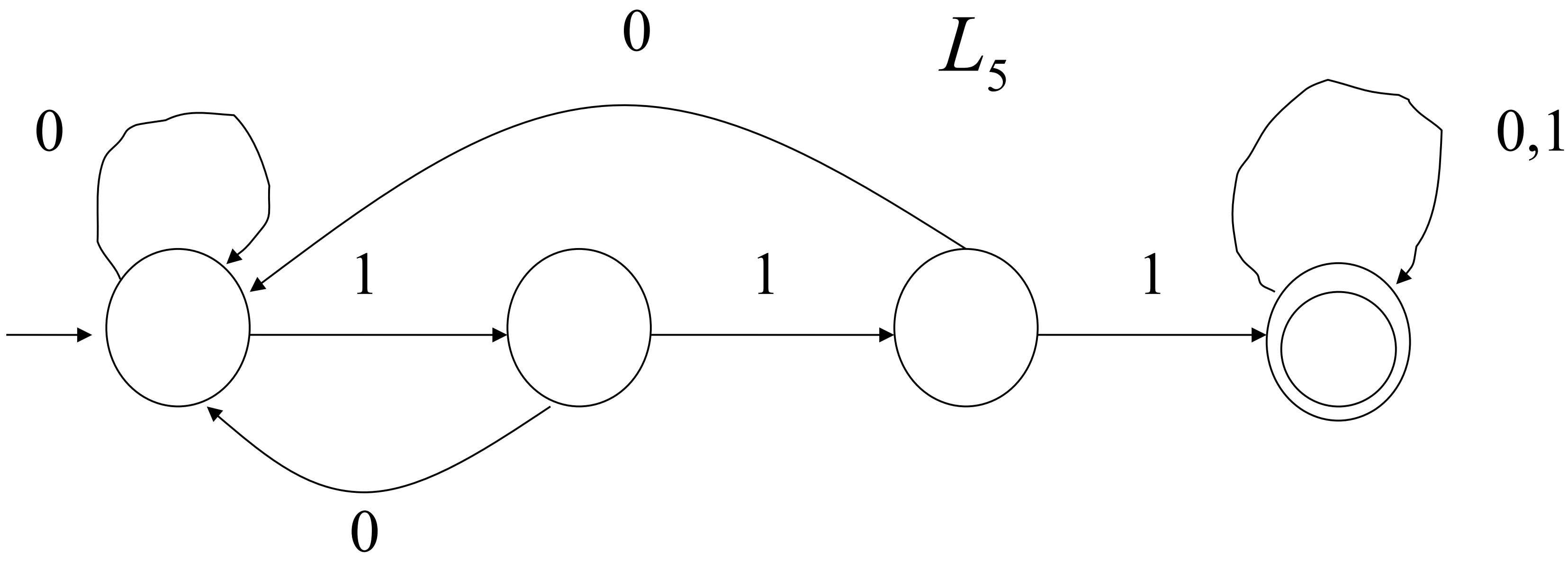
Teda chceme automat na zistenie či v slove sa nachádza toľko symbolov ***a***, ktorých počet je možné deliť číslom 3.

Riešenie:

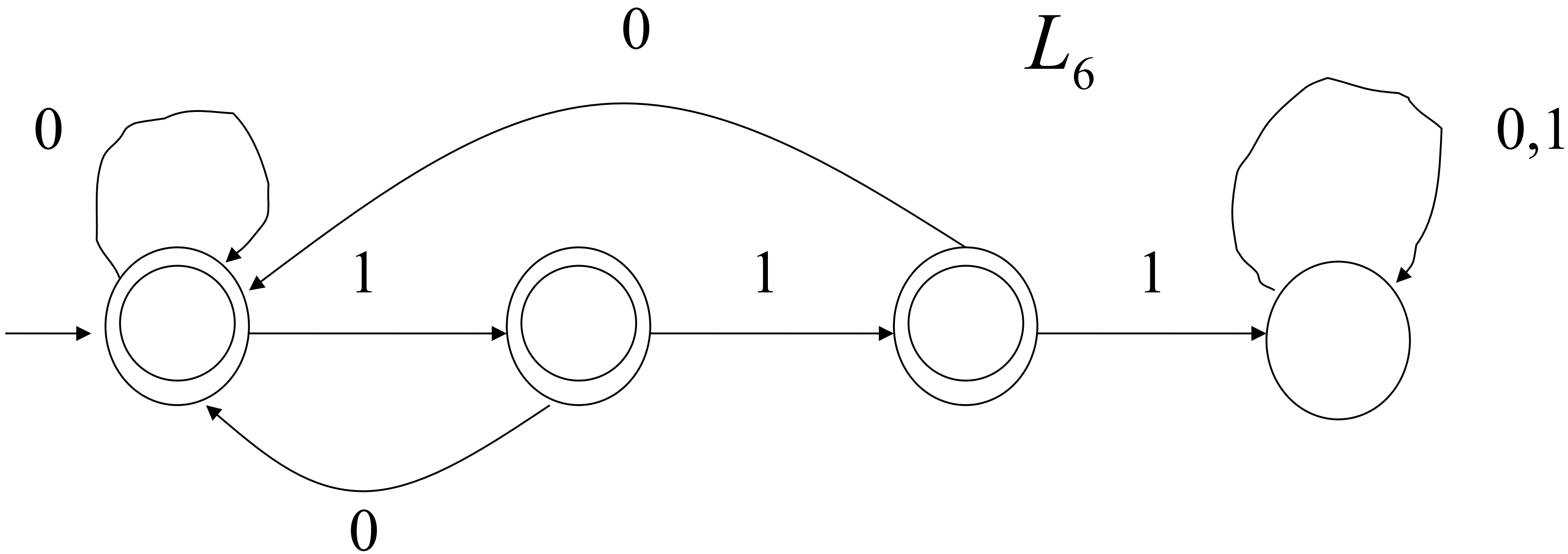

b
b
a
Zvyšok 1 po delení 3
a
Zvyšok 2 po delení 3
a
Zvyšok 0 po delení 3
b

$$L_5 = \{w \in \{0,1\}^* | (\exists u, v \in \{0,1\}^*) | w = u111v\}$$
$$L_6 = \{0,1\}^* - L_5$$

Teda jazyk L_6 pozostáva postupností 0,1, kde sa však nenachádzajú súvislé podpostupnosti troch 111.



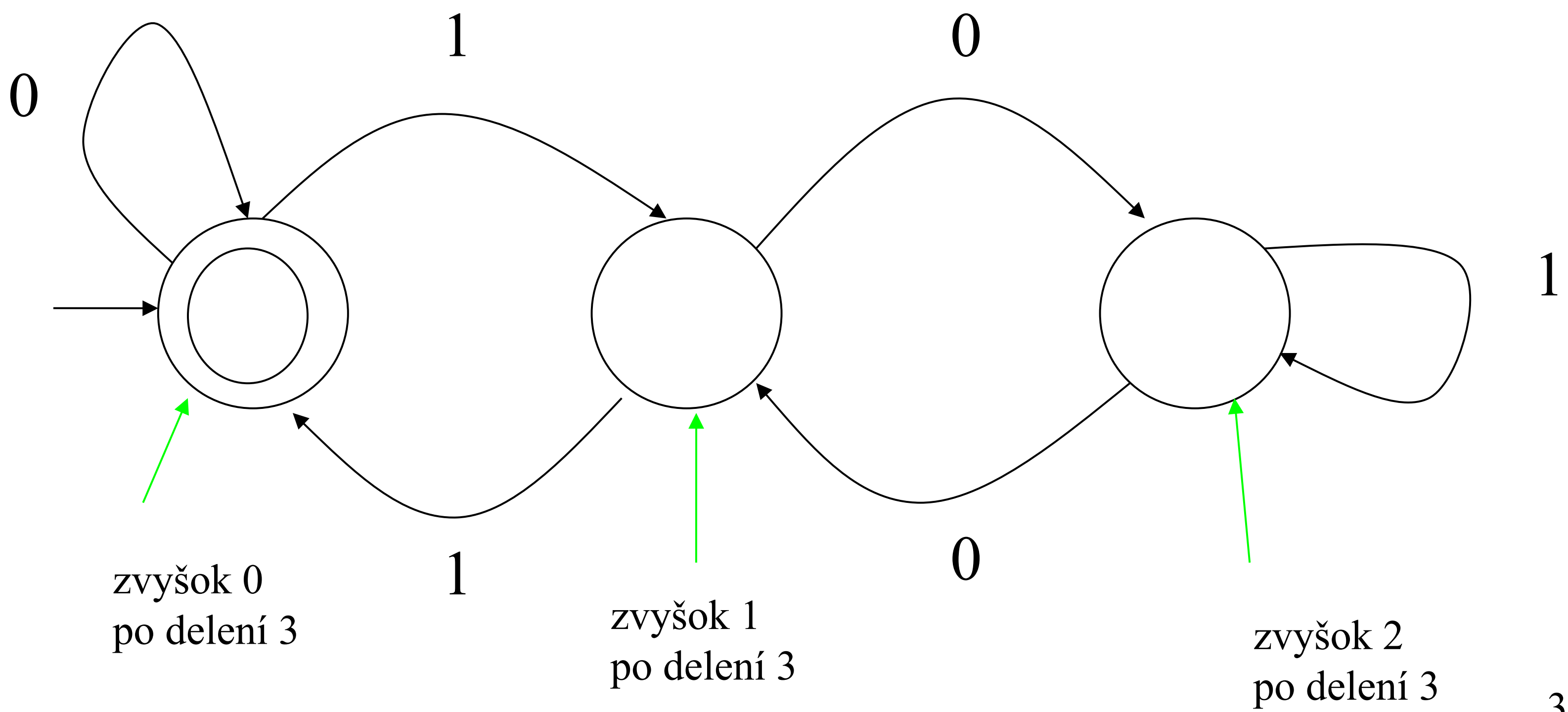
L_5
0
0
0,1
1
1
1
0



L_6
0
0
0,1
1
1
1
0

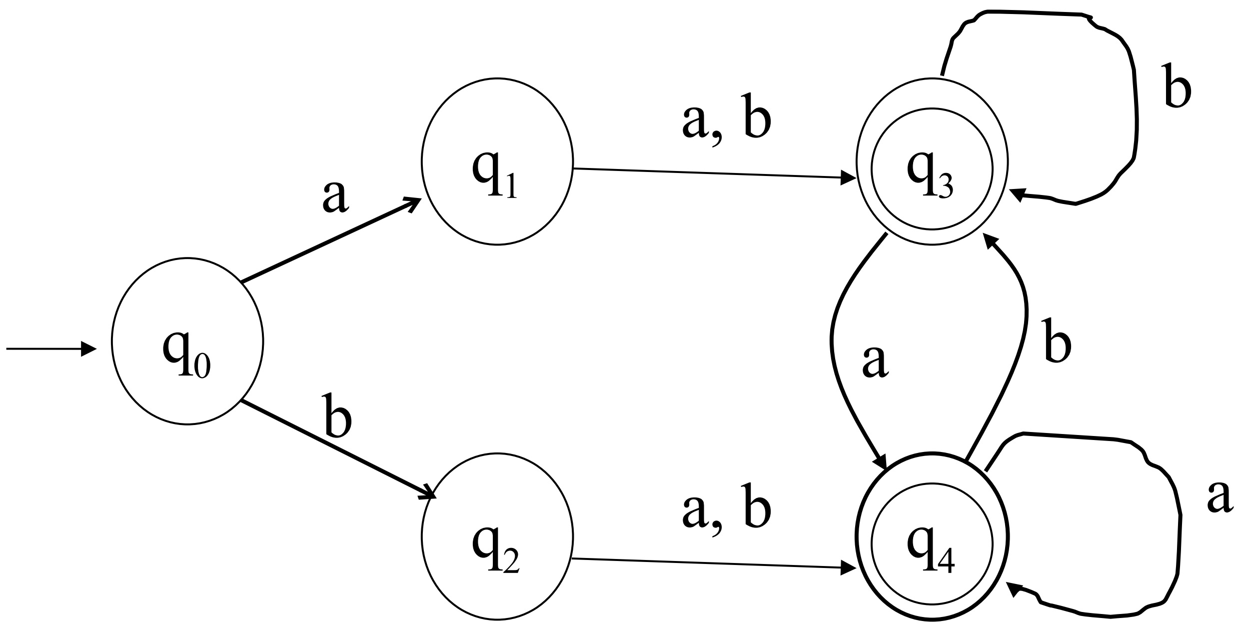
Nech *w* nejaká postupnosť symbolov 0, 1. Označme potom **b(w) prirodzené číslo,** reprezentované binárnym zápisom *w*

$$L_7 = \{w \in \{0,1\}^* | b(w) \equiv 0 (\mathrm{mod}\ \ 3)\}$$



Minimalita ...

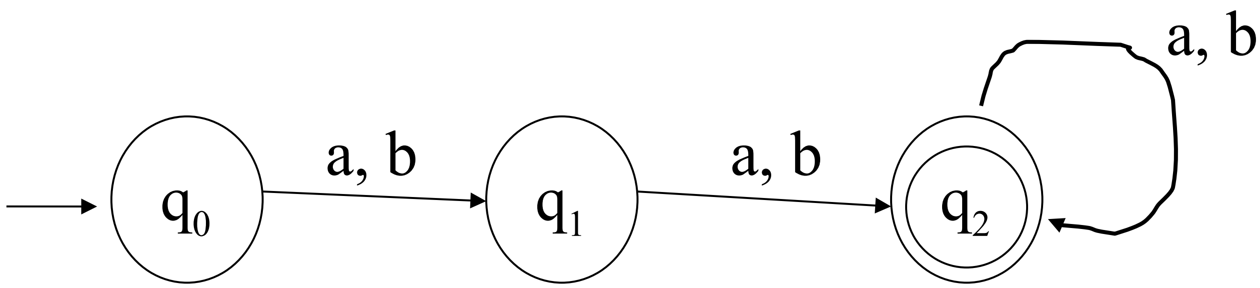
Aký jazyk rozpoznáva uvedený automat ?



Sú to všetky slová, ktoré majú dĺžku aspoň 2.

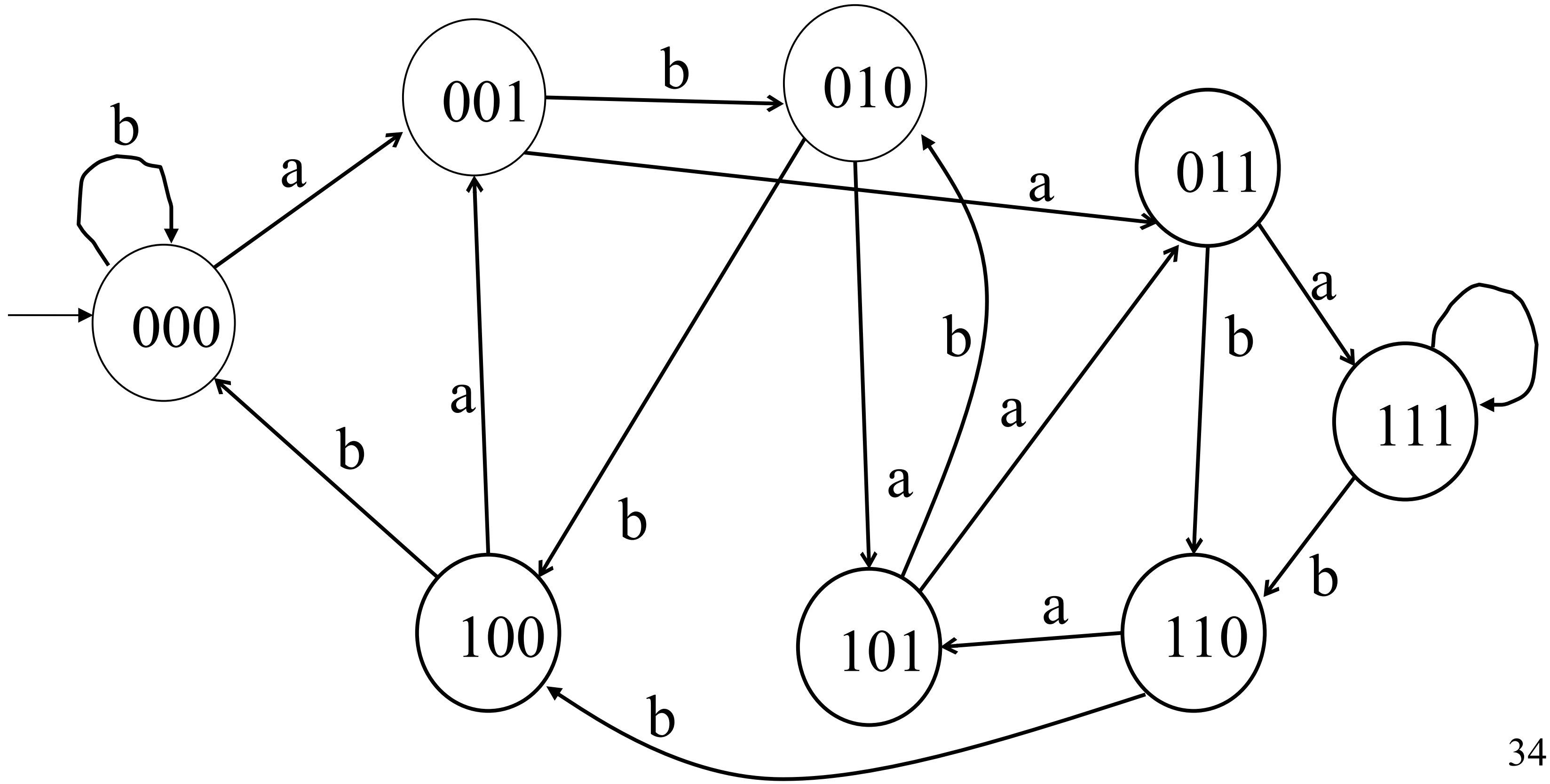
$$L = \{w \in \{a,b\}^*; |w| \geq 2\}$$

Automat je možné zmenšiť aj takto:

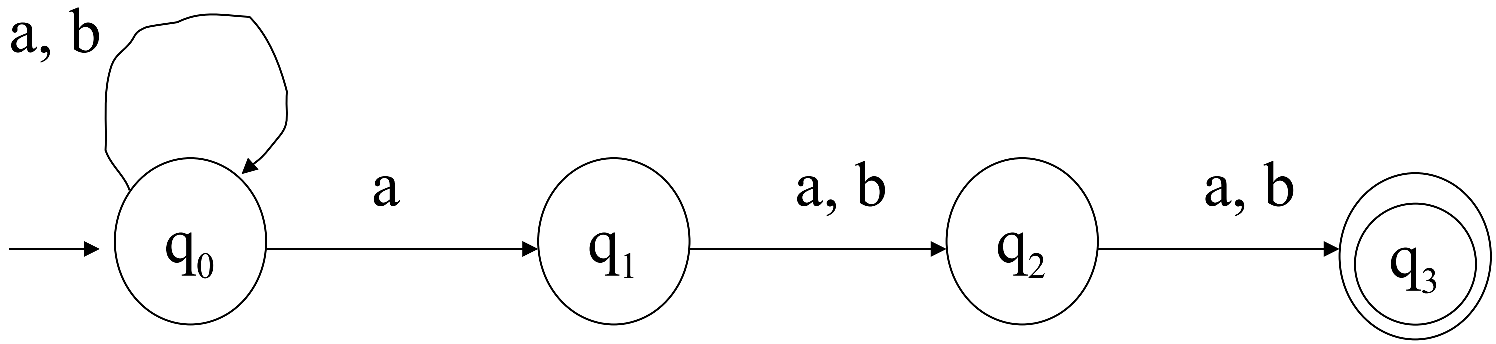


Nedeterminizmus

Uvažujme jazyk, ktorého slová majú na tretej pozícii od konca symbol **a**.

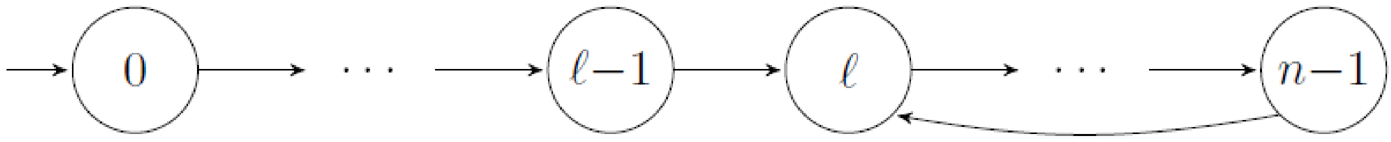


Nedeterministický automat ...



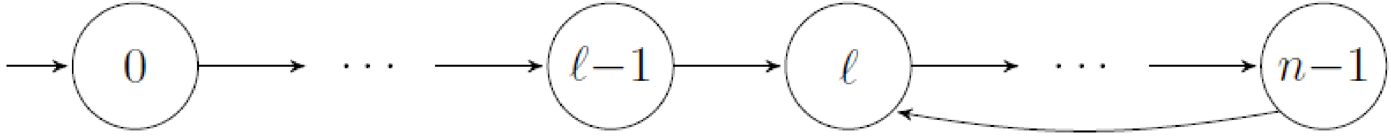
Unárne jazyky

- Sú to jazyky nad jednopísmenovou abecedou, t.j. $|\Sigma|=1$.
- Podľa dohody $\Sigma=\{a\}$. Namiesto $\{a\}^*$ píšeme a^*.
- Deterministický automat pre unárny jazyk má v každom stave práve jeden prechod.
- Každý stav má teda jednoznačne určeného nasledovníka.
- Nasledovníkom posledného stavu je jeden z prechádzajúcich stavov. Môže to byť aj on sám (vtedy je v ňom slučka).



Minimálne automaty

- Automat je minimálny, keď jeho jazyk nemôže byť akceptovaný menším automatom
- V minimálnom automate sú všetky stavy dosiahnuteľné a rozlíšiteľné
- Stavy mimo chvosta a cyklu nie sú dosiahnuteľné
- Ak sú stavy *t*-1 a *n*-1 oba (ne)koncové, tak nie sú rozlíšiteľné
- Ak postupnosť koncovosti stavov v slučke je mocninou kratšej postupnosti, tak slučka nie je minimálna a dá sa nahradiť menšou
- Minimálny unárny automat má minimálnu slučku a rôznu koncovosť stavov *t*-1 a *n*-1

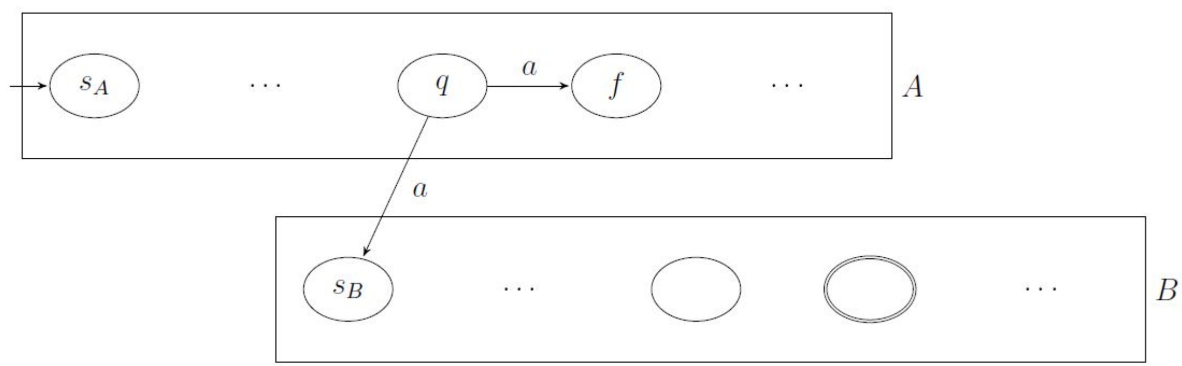


Operácie na jazykoch reprezentovaných automatmi

- Na formálnych jazykoch vykonávame rôzne operácie
- Operácia = funkcia, ktorá jazyku alebo dvojici jazykov priradí jazyk
- Príklady na operácie: $KL = \{uv \mid u \in K, v \in L\}$

$$L^k = \{u_1 u_2 \ldots u_k \mid u_1, u_2, \ldots u_k \in L\}$$

$$L^* = \bigcup_{i=0}^{\infty} L^i$$



Operácie na jazykoch reprezentovaných automatmi

- Na formálnych jazykoch vykonávame rôzne operácie
- Operácia = funkcia, ktorá jazyku alebo dvojici jazykov priradí jazyk
- Príklady na operácie:

$$KL = \{uv \mid u \in K, v \in L\}$$

$$L^k = \{u_1 u_2 \dots u_k \mid u_1, u_2, \dots u_k \in L\}$$

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

Stavová zložitosť jazykov a operácií

- Stavová zložitosť jazyka je veľkosť jeho minimálneho automatu.
- Nech K a L sú jazyky rozpoznané automatmi veľkosti m a n stavov.
- Nech + je nejaká operácia na jazykoch.
- Potom pre operáciu + existuje také číslo c, že pre každý jazyk $K+L$ existuje automat veľkosti c stavov, a niekedy je potrebných až c stavov.
- Funkciu, ktorá pre vstupy m a n dáva výstup c, voláme **stavová zložitosť operácie +**.
- Príklady na zložitosti operácií:
doplnok: n, prienik: mn, zreťazenie unárnych jazykov: mn.

Témy bakalárskej práce

- Zložitosť výsledku operácie závisí od počtu stavov pôvodných jazykov a od modelu automatov
- Model vstupného a výstupného jazyka môže byť odlišný
- Pri obmedzení na počet písmen abecedy môže byť zložitosť nižšia

1) Popisná zložitosť operácií
na rôznych modeloch konečných automatov

2) Popisná zložitosť operácií
na binárnych regulárnych jazykoch

Príklady (pre motiváciu)

- Galina Jirásková: State complexity of some operations on **binary** regular languages. Theor. Comput. Sci. 330(2): 287-298 (2005)
- Galina Jirásková, Juraj Šebej: Reversal of **binary** regular languages. Theor. Comput. Sci. 449: 85-92 (2012)
- Michal Hospodár, Jozef Jirásek, Galina Jirásková, Juraj Šebej: Operational complexity: **NFA-to-DFA** trade-off. Inf. Comput. 307: 105369 (2025)

ĎAKUJEM ZA POZORNOSŤ