Predchádzajúce

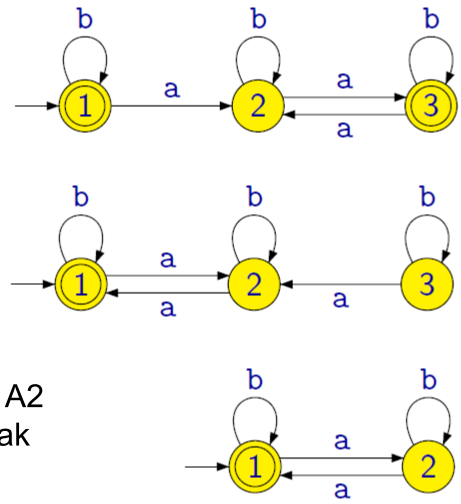
- **Abeceda Σ** je končená neprázdna množina symbolov (písmen)
  - **Σ = {a, b, c}**.
- Deterministický konečný automat
  - **$(K, \Sigma, \delta, q_0, F)$**
- Konfigurácia DKA
  - **$(1, ababb) \vdash (2, babb) \vdash (5, abb) \vdash (4, bb) \vdash (3, b) \vdash (4, \varepsilon)$**
- Regulárny jazyk
  - **$L(A) = \{w \in \Sigma^* \mid \exists q \in F: q_0 \xrightarrow{w} q\}$**
- Nedeterministický KA
  - **$(K, \Sigma, \delta, q_0, F)$**, $\delta : K \times (\Sigma \times \{\varepsilon\}) \rightarrow 2^K$ je prechodová funkcia
  - Výraz $\delta(q, a) = \{p_1, p_2, \ldots, p_n\}$ potom chápeme tak, že automat sa v stave q pri čítaní symbolu **a** **môže rozhodnúť**, do akého stavu sa dostane po posune čítacej hlavy.
- Ekvivalencia DKA a NKA

Teoretické základy informatiky I.

Ekvivalencia KA / nedosiahnuteľné stavy KA / normovaný tvar KA

Ekvivalencia automatov

- Všetky 3 automaty akceptujú jazyk všetkých slov s párnym počtom **a**
- Najvýhodnejšie je pre nás posledný z nich
  - má najmenej stavov.

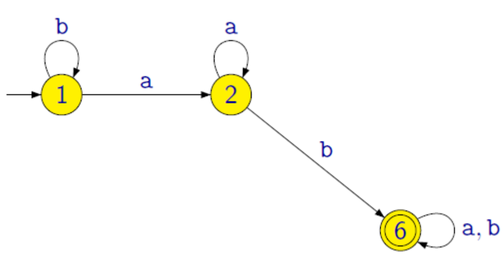


Definícia

- O konečných automatoch A1, A2 povieme, že sú ekvivalentné, ak
  L (A1) = L (A2).

Nedosiahnuteľné stavy automatu

- Automat rozpoznáva jazyk
  L = {w ∈ {a, b} * | w obsahuje podslovo ab}
- Pre žiadnu postupnosť vstupných symbolov sa automat nedostane do stavov **3**, **4** alebo **5**.



- Ak tieto stavy odstránime, stále automat rozpoznáva rovnaký jazyk L.

Nedosiahnuteľné stavy automatu

Definícia

- Stav q konečného automatu A = **(K, Σ, δ, q0, F)** je **dosiahnuteľný** ak existuje nejaké slovo w také, že $q_0 \xrightarrow{w} q$.
- V opačnom prípade stav nazývame **nedosiahnuteľný**.

- Do nedosiahnuteľných stavov nevedie v grafe automatu žiadna orientovaná cesta z počiatočného stavu.
- Nedosiahnuteľné stavy môžeme z automatu **odstrániť** (spolu so všetkými prechodmi vedúcimi do nich a z nich). Jazyk rozpoznávaný automatom **sa nezmení**.
- Ku každému DKA A možno skonštruovať DKA A', v ktorom každý stav je dosiahnuteľný a L(A') = L(A).

Nedosiahnuteľné stavy automatu

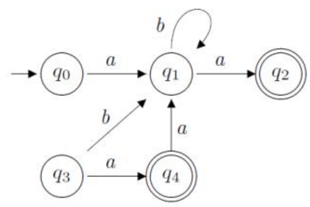
Postup:

- vytvoríme množinu S_0, do nej dáme počiatočný stav automatu, $S_0 = \{q0\}$,
- vytvoríme množinu S_1 dáme do nej prvky množiny S_0 a všetky stavy, **do ktorých vedie prechod** zo stavov množiny S_0
  - na začiatku pridáme všetky stavy, do ktorých sa dá dostať priamo z q0,
- postupne vytvárame množiny S_i tak, že do S_i zaradíme najskôr **obsah množiny** S_{i-1} a potom **pridáme všetky stavy**, do ktorých **vedie prechod** z niektorého stavu z množiny S_{i-1},
- končíme, keď už sa do množiny nič nedá pridať, teda $S_i = S_{i-1}$,

$$S_0 = \{q0\}$$
$$S_i = S_{i-1} \cup \{q \mid \delta(p, a) \ni q,\ p \in S_{i-1},\ a \in \Sigma\}$$

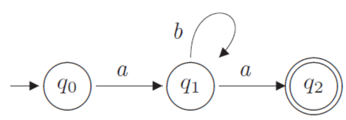
Nedosiahnuteľné stavy automatu - príklad

	a	b
$\rightarrow q_0$	q_1	-
q_1	q_2	q_1
$\leftarrow q_2$	-	-
q_3	q_4	q_1
$\leftarrow q_4$	q_1	-



- S0 = {q0}, hľadáme prvky S_{i-1} na označených riadkoch, pridáme obsah buniek riadkov
- S1 = {q0, q1} (z q0 sa dá dostať do q1)
- S2 = {q0, q1, q2} (z q0 a q1 sa dá dostať tiež do q2)
- S3 = {q0, q1, q2} = S2 . . . nová množina stavov

	a	b
$\rightarrow q_0$	q_1	-
q_1	q_2	q_1
$\leftarrow q_2$	-	-



Nadbytočné stavy automatu

- sú také stavy, z ktorých sa žiadnym spôsobom nedá dostať do žiadneho **akceptačného stavu**
  - aj takéto stavy je možné z KA odstrániť.

Definícia

- Nech A = **(K, Σ, δ, q0, F)** je konečný automat.
  Stav q patri K je **nadbytočný**, ak z tohto stavu neexistuje žiadna zovšeobecnená prechodová funkcia **δ*(q, v') = q_F**, kde **v'** je dosiaľ neprečítaná časť vstupného slova a **q_F** patri do F.

Nadbytočné stavy automatu

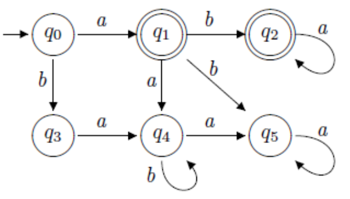
Postup:

- odstránime nedosiahnuteľné stavy
- vytvoríme množinu $\mathbf{E_0}$, do ktorej zaradíme všetky koncové stavy automatu, t.j. $\mathbf{E_0 = F}$,
- vytvoríme množinu $\mathbf{E_1}$ tak, že do nej dáme prvky množiny $\mathbf{E_0}$ a ďalej všetky stavy, z ktorých **vedie prechod** do stavov množiny $\mathbf{E_0}$,
- postupne vytvárame množiny $\mathbf{E_i}$ tak, že do $\mathbf{E_i}$ zaradíme najskôr obsah množiny $\mathbf{E_{i-1}}$ a potom **pridáme všetky stavy**, z ktorých **vedie prechod** do niektorého stavu z množiny $\mathbf{E_{i-1}}$,
- končíme, keď už sa do množiny nič nedá pridať, teda $\mathbf{E_i} = \mathbf{E_{i-1}}$,
- Výsledkom je nová množina stavov.

$$\mathbf{E_0 = F}$$
$$\mathbf{E_i = E_{i-1} \cup \{q \mid \delta(q, a) \ni p,\ p \in E_{i-1},\ a \in \Sigma\}}$$

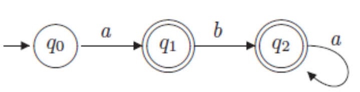
Nadbytočné stavy automatu - príklad

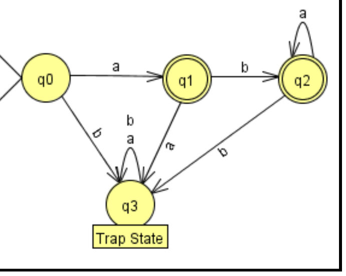
	a	b
$\rightarrow q_0$	q_1	q_3
$\leftarrow q_1$	q_4	q_2, q_5
$\leftarrow q_2$	q_2	-
q_3	q_4	-
q_4	q_5	q_4
q_5	q_5	-



- predpokladáme, že nedosiahnuteľné stavy sú už odstránené
- E0 = {q1, q2}, hľadáme prvky $\mathbf{E_{i-1}}$ v bunkách riadkov, pridáme označenie riadkov
- E1 = {q1, q2, q0} (z q0 sa dá dostať do q1)
- E2 = {q1, q2, q0} = E1 . . . nová množina stavov

	a	b
$\rightarrow q_0$	q_1	q_3
$\leftarrow q_1$		q_2
$\leftarrow q_2$	q_2	-

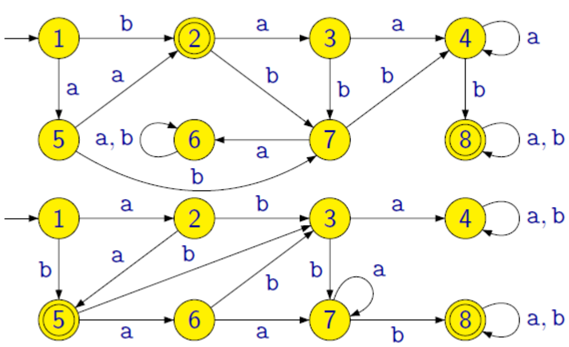




Normovaný tvar automatu

- Možno ľahko nahliadnuť, že pre jazyk L(A) akceptujúci automat A môžeme navrhnúť (**nekonečne**) veľa ďalších automatov, ktoré ho tiež akceptujú.
- Stačí pridávať nové stavy, do ktorých sa automat pre žiadne slovo počas výpočtu nedostane.
- Je ale zrejmé, že výhodnejšie budú pre nás tie automaty, ktoré sú čo **najmenšie** a teda také zbytočné, **nedosiahnuteľné** stavy **neobsahujú**.

Normovaný tvar automatu

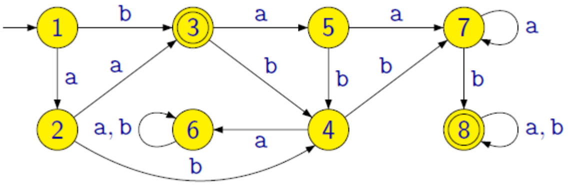


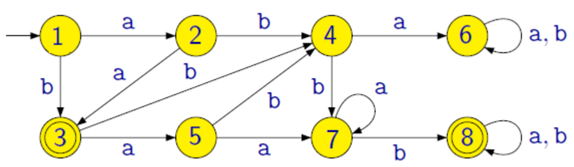
$\mathcal{A}_1$	a	b
$\rightarrow 1$	5	2
$\leftarrow 2$	3	7
3	4	7
4	4	8
5	2	7
6	6	6
7	6	4
$\leftarrow 8$	8	8

$\mathcal{A}_2$	a	b
$\rightarrow 1$	2	5
2	5	3
3	4	7
4	4	4
$\leftarrow 5$	6	3
6	7	3
7	7	8
$\leftarrow 8$	8	8

- Ak sa automaty líšia len pomenovaním stavov, potom sú celkom iste ekvivalentné.
- Automaty na obrázku môžeme číslami 1 až 8 pomenovať 8! spôsobmi.
- Chceli by sme **jednoznačne vybrať jedno** z možných pomenovaní.

Normovaný tvar automatu





	a	*b*
→ 1	2	3
2	3	4
← 3	5	4
4	6	7
5	7	4
6	6	6
7	7	8
← 8	8	8

- Teraz už je prechodová funkcia rovnaká pre obidva automaty, rovnaké sú aj počiatočné a akceptujúce stavy.
- Automaty sú teda nielen ekvivalentné, ale dokonca rovnaké.
- Hovoríme, že automaty sú v **normovanom** tvare.

Normovaný tvar automatu

Definícia

- Majme deterministický konečný automat A = **($K, \Sigma, \delta, q_0, F$)** bez nedosiahnuteľných stavov.
- Predpokladajme, že máme určité usporiadanie prvkov abecedy **Σ**, ktorým je určené usporiadanie **$<_L$** na **Σ^***.
- Označme pre každý stav $q \in K$ symbolom **u_q** najmenšie slovo podľa usporiadania **$<_L$** také, že **$q_0 \stackrel{u_q}{\to} q$**.
- DKA A je v **normovanom tvare**, ak:
  - **$K = \{1, 2, 3,. . . , N\}$** pre nejaké **$n \geq 1$**.
  - **1** je počiatočný stav.
  - Pre každú dvojicu stavov **$i, j \in \{1, 2,. . . , n\}$** takú, že **$i<j$**, platí **$u_i <_L u_j$**.

Komentár: Teda automat A je v normovanom tvare, ak jeho stavy sú očíslované 1, 2,. . .
v abecednom poradí najmenších slov, ktorými tieto stavy možno dosiahnuť.

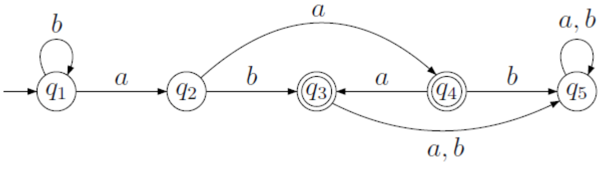
Normovaný tvar automatu

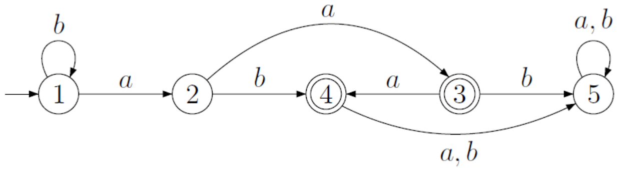
Postup prevodu DKA do normovaného tvaru (prečíslovaním stavov)

- Počiatočný stav označíme 1.
- Ďalej, napríklad v prípade abecedy {a, b}, zistíme stav q, do ktorého automat prejde zo stavu 1 symbolom a;
  - keď q nie je označený, označíme ho 2.
- Potom zistíme stav q, do ktorého automat prejde zo stavu 1 symbolom b;
  - keď q nie je dosiaľ označený, označíme ho najmenším doteraz nepoužitým číslom.
- Takto sme "vybavili" stav 1, pokračujeme "vybavením" 2 atď., kým nezískame všetky dosiahnuteľné stavy.
- Jedná sa vlastne o prechádzanie **grafu do šírky** pri zoradení hrán podľa abecedy.

Normovaný tvar automatu - príklad

Stavy DKA zoraďte ako majú byť číslované v normovanom tvare.



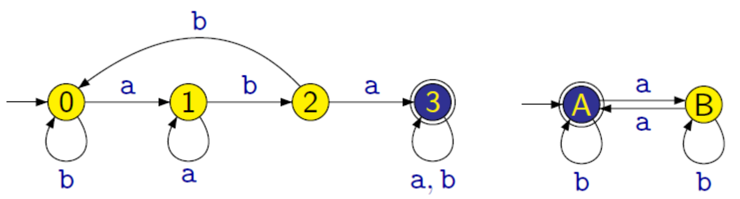


Automat pre prienik jazykov

- Pre jednoduché jazyky môžeme byť schopní navrhnúť automat priamo.
- Pre zložitejšie jazyky už to môže byť ťažšie.
- Často môžeme byť schopní zložitejší jazyk opísať ako **výsledok nejakej operácie** (zjednotenie, prienik, doplnok, zreťazenie, iterácie, zrkadlový obraz a pod.) aplikované na nejaké jednoduchšie jazyky.
- V takomto prípade môže byť reálnejší nasledujúci modulárny postup:
  - Najprv zostrojiť automaty pre dané jednoduchšie jazyky.
  - Z vytvorených automatov určitou algoritmickou konštrukciou vytvoriť automat rozpoznávajúci jazyk, ktorý je výsledkom príslušnej operácie aplikovanej na jazyky rozpoznávané pôvodnými automatmi.

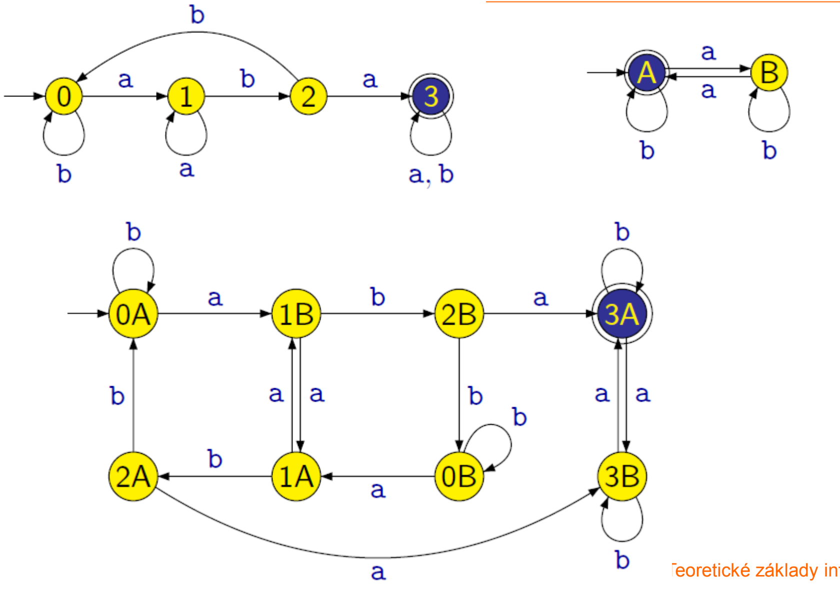
Automat pre prienik jazykov

- Máme nasledujúce dva automaty:



- Akceptujú obidva slovo **ababb**?
- Môžeme postupne nechať prečítať slovo obom automatom. Odpoveď bude **Áno**, ak obaja odpovedia **Áno**.
- Namiesto toho môžeme nechať oba automaty čítať dané slovo súčasne.
- Pri tomto čítaní si stačí v každom kroku pamätať dvojicu stavov, v ktorých sa oba automaty nachádzajú.
- Toto môže byť realizované konečným automatom, ktorý simuluje činnosť oboch automatov súčasne! (Dvojíc stavov je konečne veľa.)

Automat pre prienik jazykov



Automat pre prienik jazykov

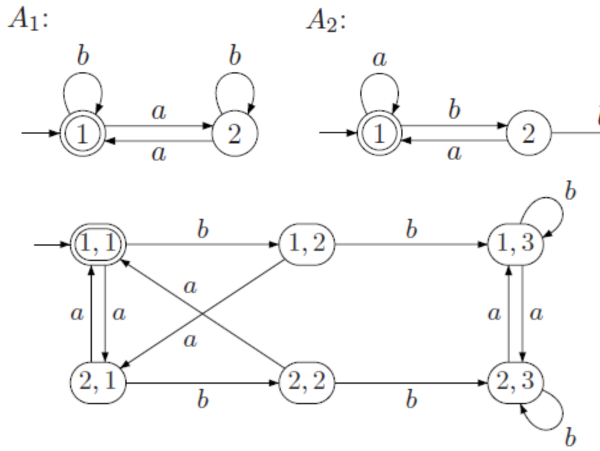
Formálne môžeme opísať túto konštrukciu nasledovne:

- Predpokladáme, že máme dva deterministické konečné automaty

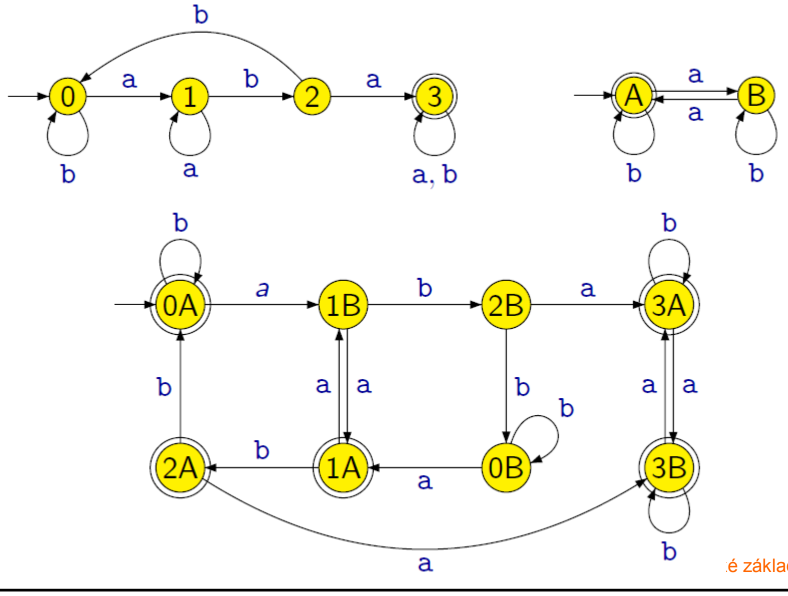
  A1 = **($K1$, Σ, $\delta1$, $q01$, $F1$)** a A2 = **($K2$, Σ, $\delta2$, $q02$, $F2$)**.
- K nim zostrojíme DKA A = **(K, Σ, δ, $q0$, F)**, kde:
  - $K = K1 \times K2$
  - $\delta((K1, K2), a) = (\delta1(q1, a), \delta2(q2, a))$ pre všetky $q1 \in K1$, $q2 \in K2$,
  - $a \in \Sigma$
  - $q0 = (q01, q02)$
  - $F = F1 \times F2$
- Nie je ťažké overiť, že pre ľubovoľné slovo $w \in \Sigma^*$ platí, že $w \in L(A)$ práve vtedy, keď $w \in L(A1)$ a $w \in L(A2)$, t.j. $L(A) = L(A1) \cap L(A2)$

Automat pre prienik jazykov

- L(A1) = {w ∈ {a, b} * | $\#_a$w mod 2 = 0}
- L(A2) = {w ∈ {a, b} * | vo w je každý výskyt symbolu *b* nasledovaný symbolom *a*}
- L = L(A1) ∩ L(A2)



Automat pre zjednotenie jazykov



Automat pre zjednotenie jazykov

- Konštrukcia automatu A, ktorý prijíma zjednotenie jazykov prijímaných automaty A1 a A2, t.j. jazyk

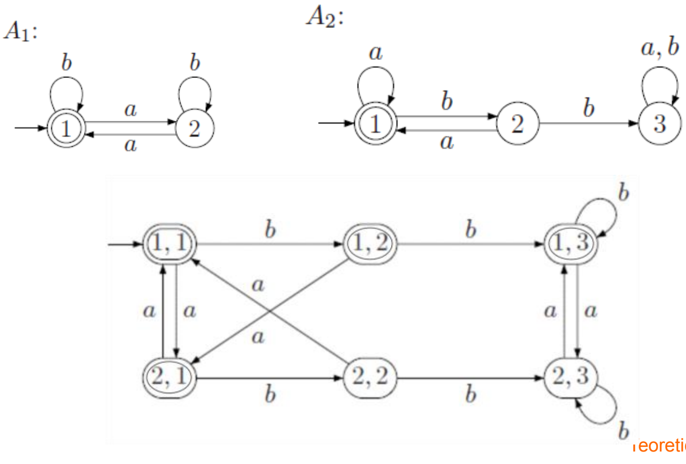
  L(A1) ∪ L(A2)
- je takmer rovnaká ako v prípade automatu prijímajúceho L(A1) ∩ L(A2).

- Jediný rozdiel je v definícii množiny prijímajúcich stavov:

  F = (F1 × K2) ∪ (K1 × F2)

Automat pre zjednotenie jazykov

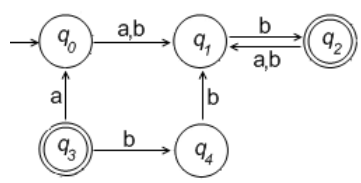
- L(A1) = {w ∈ {a, b} * | $\#_a$w mod 2 = 0}
- L(A2) = {w ∈ {a, b} * | vo w je každý výskyt symbolu *b* nasledovaný symbolom *a*}
- L = L(A1) ∪ L(A2)



NDÚ – úlohy

- Ku konečnému automatu A vytvorte ekvivalentný konečný automat A' bez **nedosiahnuteľných** stavov.

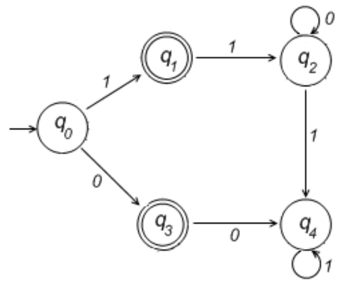
	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_2	-
$\leftarrow q_2$	q_1	q_1
$\leftarrow q_3$	q_0	q_4
q_4	-	q_1



NDÚ – úlohy

- Ku konečnému automatu A vytvorte ekvivalentný konečný automat A' bez **nadbytočných** stavov.

	0	1
$\rightarrow q_0$	q_3	q_1
$\rightarrow q_1$	-	q_2
q_2	q_2	q_4
$\leftarrow q_3$	q_4	-
q_4	-	q_4



NDÚ – úlohy

- Zadaný konečný automat A preveďte do normovaného tvaru.