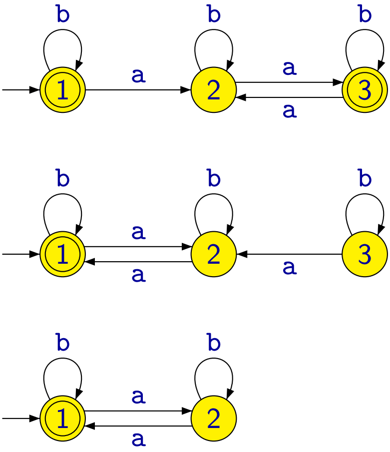
Ekvivalence automatů



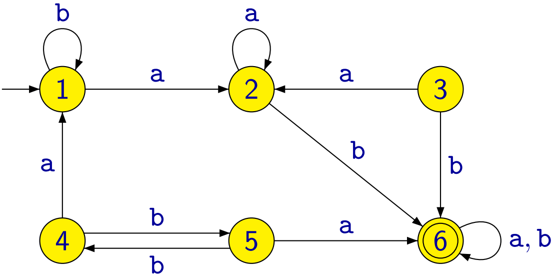
- Všechny 3 automaty přijímají jazyk všech slov se sudým počtem *a*.
- Nejvýhodnější je pro nás poslední z nich – má nejméně stavů.

Ekvivalence automatů

Definice

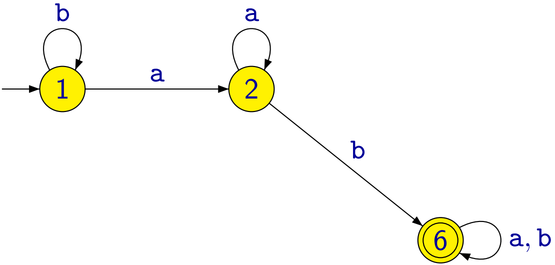
O konečných automatech A_1, A_2 řekneme, že jsou **ekvivalentní**, jestliže $L(A_1) = L(A_2)$.

Nedosažitelné stavy automatu



- Automat přijímá jazyk $L = \{w \in \{a, b\}^* \mid w \text{ obsahuje podslovo } ab\}$
- Pro žádnou posloupnost vstupních symbolů se automat nedostane do stavů 3, 4 nebo 5.

Nedosažitelné stavy automatu



- Automat přijímá jazyk $L = \{w \in \{a, b\}^* \mid w \text{ obsahuje podslovo } ab\}$
- Pro žádnou posloupnost vstupních symbolů se automat nedostane do stavů 3, 4 nebo 5.
- Pokud tyto stavy odstraníme, pořád automat přijímá stejný jazyk L.

Nedosažitelné stavy automatu

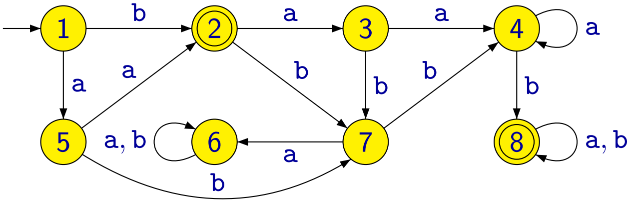
Definice

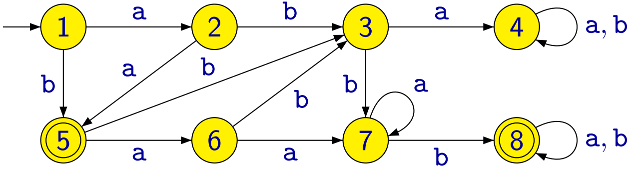
Stav q konečného automatu $A = (Q, \Sigma, \delta, q_0, F)$ je **dosažitelný** pokud existuje nějaké slovo w takové, že $q_0 \xrightarrow{w} q$.

V opačném případě stav nazýváme **nedosažitelný**.

- Do nedosažitelných stavů nevede v grafu automatu žádná orientovaná cesta z počátečního stavu.
- Nedosažitelné stavy můžeme z automatu odstranit (spolu se všemi přechody vedoucími do nich a z nich). Jazyk přijímaný automatem se nezmění.

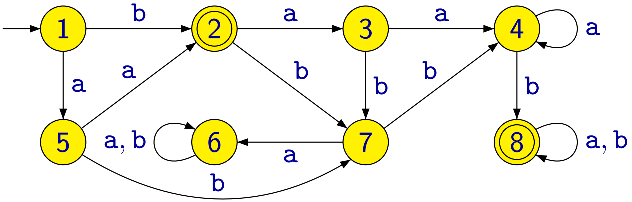
Normovaný tvar automatu

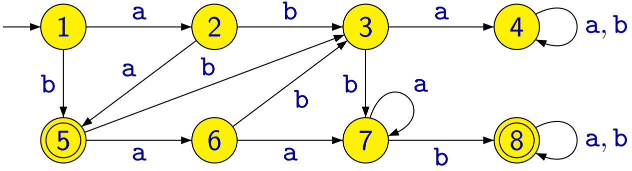




$\mathcal{A}_1$	a	b
$\rightarrow$ 1	5	2
$\leftarrow$ 2	3	7
3	4	7
4	4	8
5	2	7
6	6	6
7	6	4
$\leftarrow$ 8	8	8

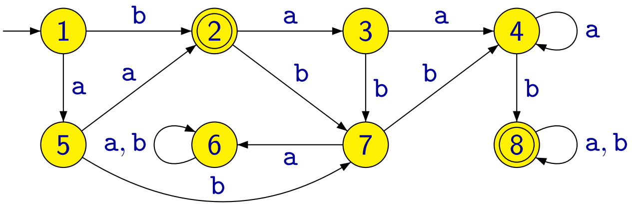
Normovaný tvar automatu

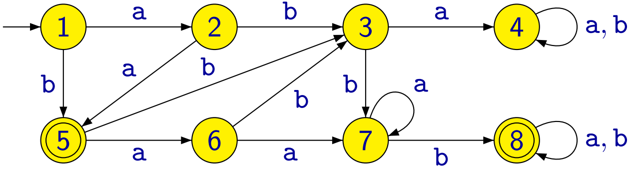




$\mathcal{A}_2$	a	b
$\rightarrow 1$	2	5
2	5	3
3	4	7
4	4	4
$\leftarrow 5$	6	3
6	7	3
7	7	8
$\leftarrow 8$	8	8

Normovaný tvar automatu

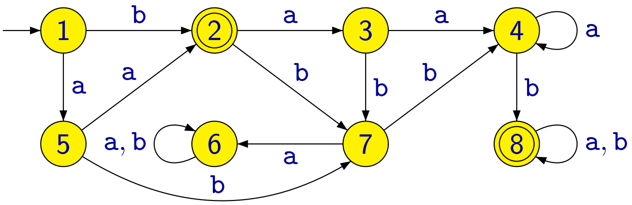




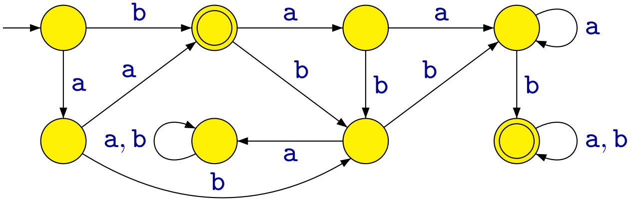
$\mathcal{A}_2$	a	b
$\rightarrow 1$	2	5
2	5	3
3	4	7
4	4	4
$\leftarrow 5$	6	3
6	7	3
7	7	8
$\leftarrow 8$	8	8

- Pokud se automaty liší jen pojmenováním stavů, pak jsou zcela jistě ekvivalentní.
- Automaty na obrázku můžeme čísly 1 až 8 pojmenovat 8! způsoby.
- Chtěli bychom jednoznačně vybrat jedno z možných pojmenování.

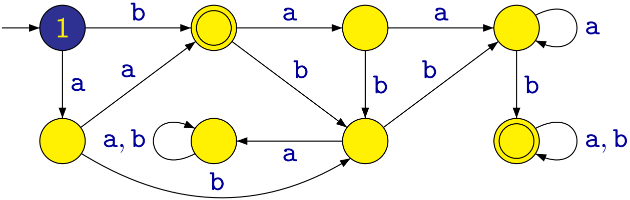
Normovaný tvar automatu



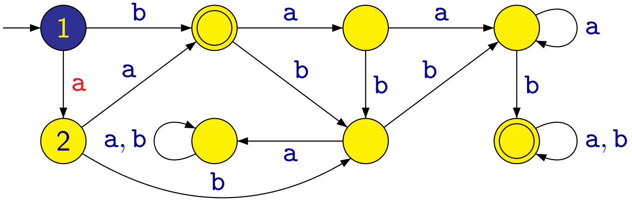
Normovaný tvar automatu



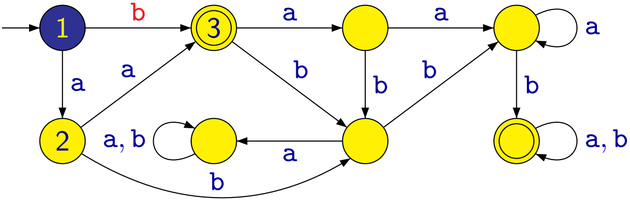
Normovaný tvar automatu



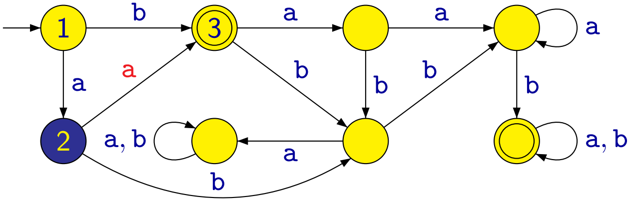
Normovaný tvar automatu



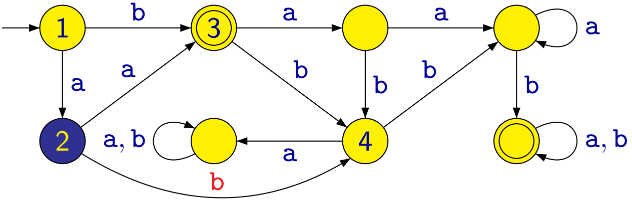
Normovaný tvar automatu



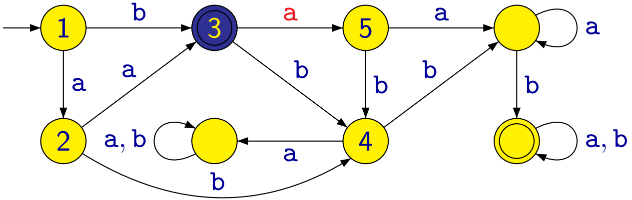
Normovaný tvar automatu



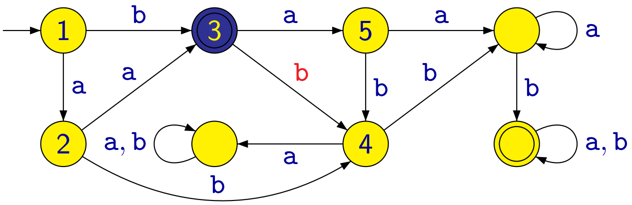
Normovaný tvar automatu



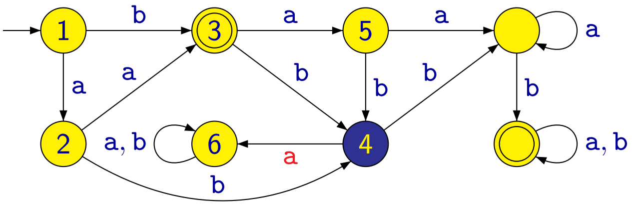
Normovaný tvar automatu



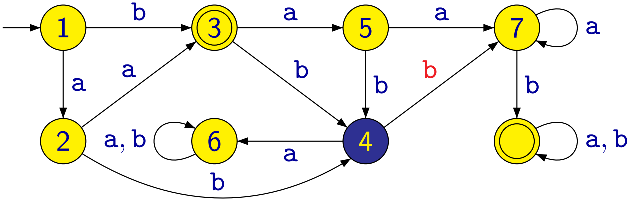
Normovaný tvar automatu



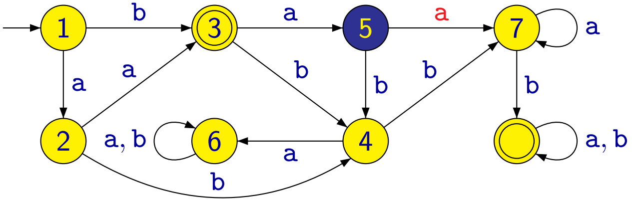
Normovaný tvar automatu



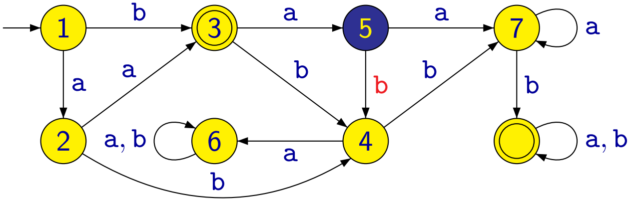
Normovaný tvar automatu



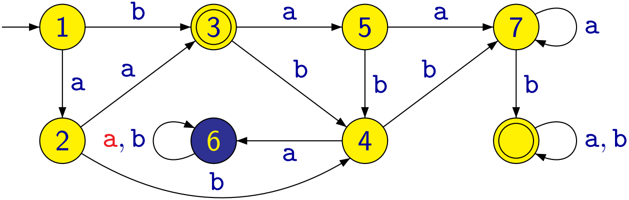
Normovaný tvar automatu



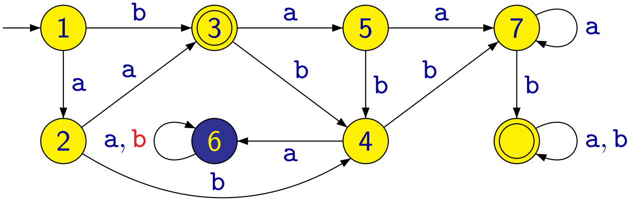
Normovaný tvar automatu



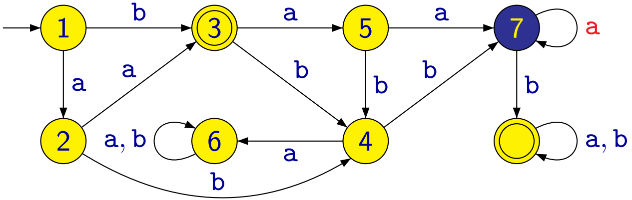
Normovaný tvar automatu



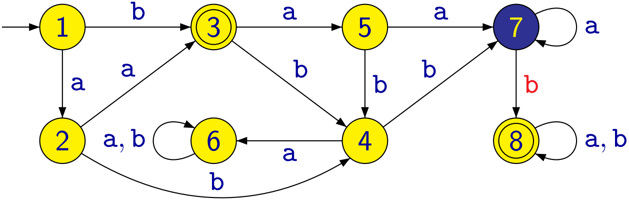
Normovaný tvar automatu



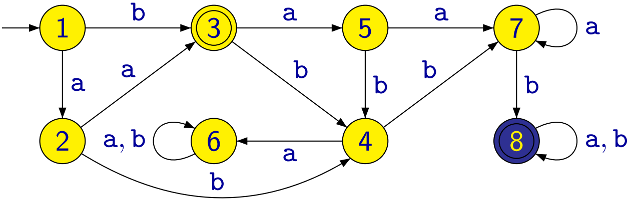
Normovaný tvar automatu



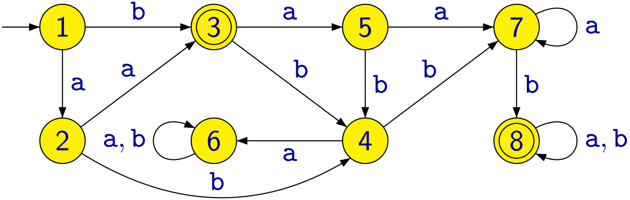
Normovaný tvar automatu



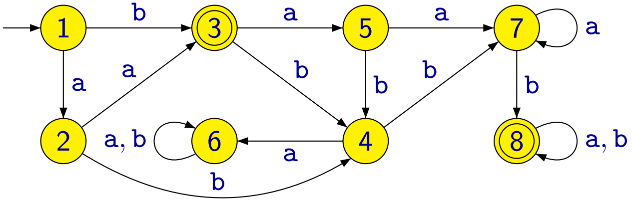
Normovaný tvar automatu

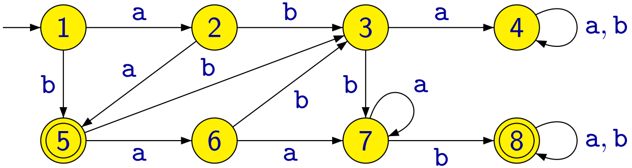


Normovaný tvar automatu

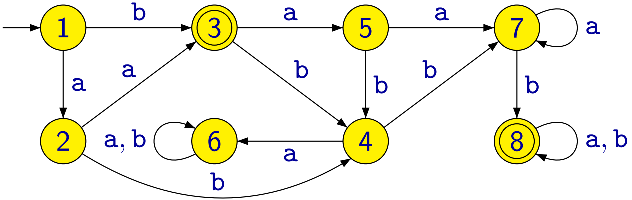


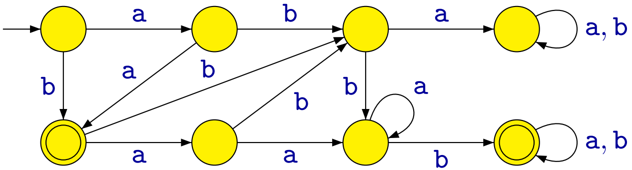
Normovaný tvar automatu



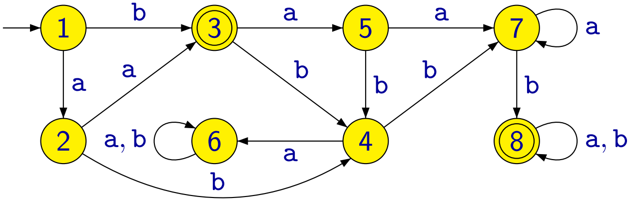


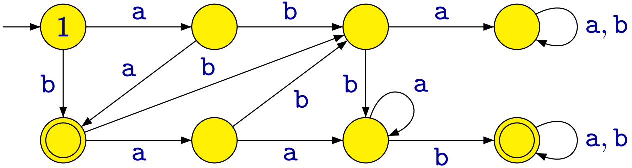
Normovaný tvar automatu



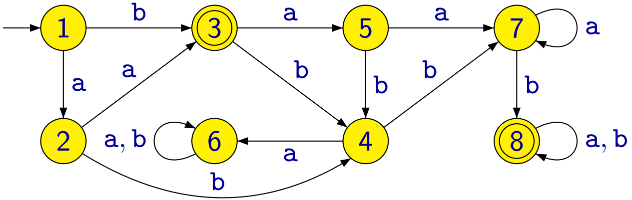


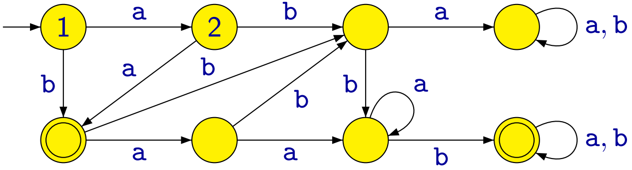
Normovaný tvar automatu



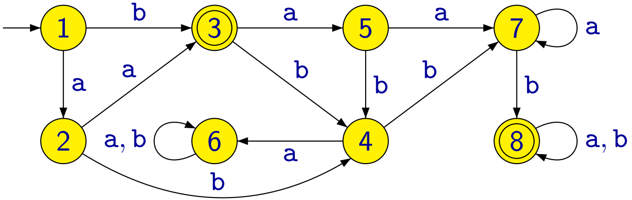


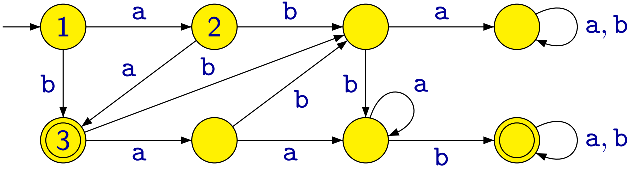
Normovaný tvar automatu



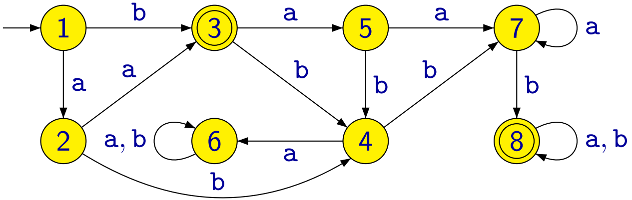


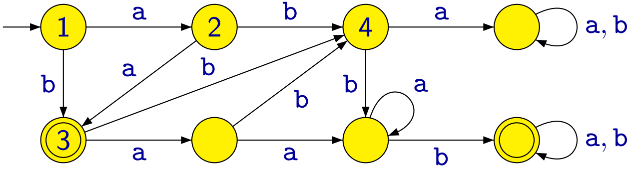
Normovaný tvar automatu



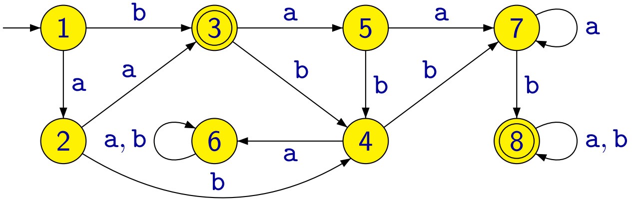


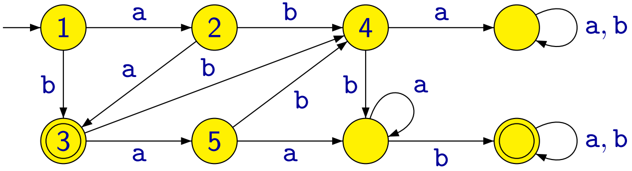
Normovaný tvar automatu



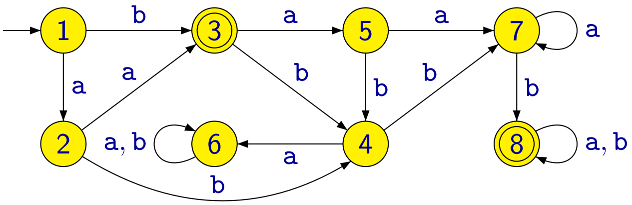


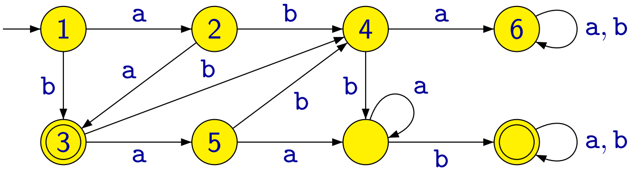
Normovaný tvar automatu



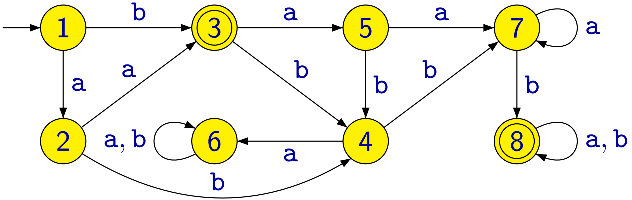


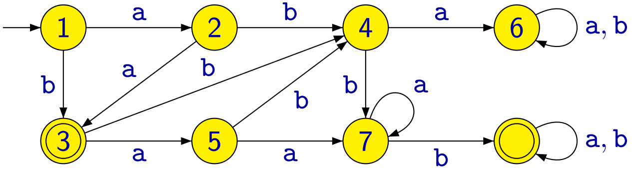
Normovaný tvar automatu



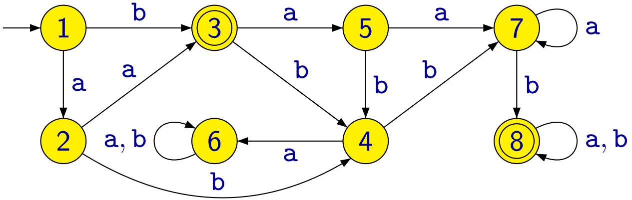


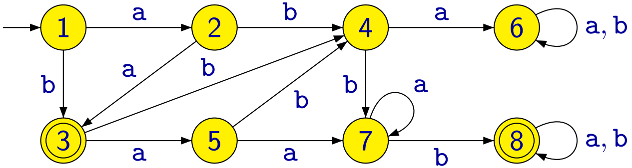
Normovaný tvar automatu

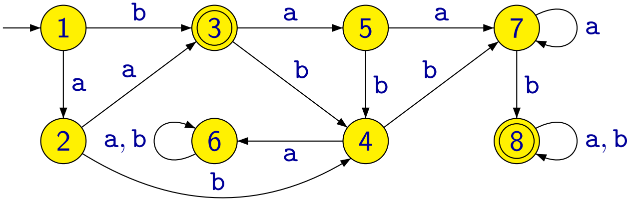


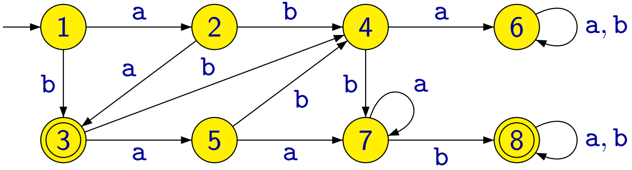


Normovaný tvar automatu



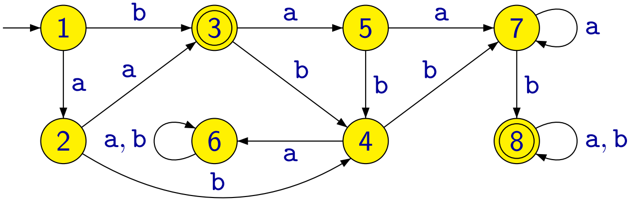


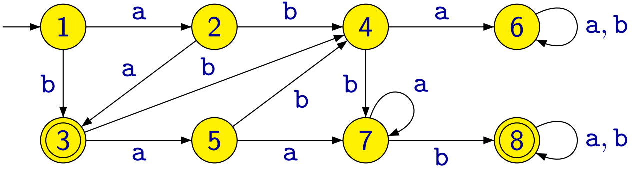




	a	b
$\rightarrow$ 1	2	3
2	3	4
$\leftarrow$ 3	5	4
4	6	7
5	7	4
6	6	6
7	7	8
$\leftarrow$ 8	8	8

Normovaný tvar automatu





	a	*b*
$\to$ 1	2	3
2	3	4
$\leftarrow$ 3	5	4
4	6	7
5	7	4
6	6	6
7	7	8
$\leftarrow$ 8	8	8

- Nyní už je přechodová funkce stejná pro oba automaty, stejné jsou i počáteční a přijímající stavy.
- Automaty jsou tedy nejen ekvivalentní, ale dokonce stejné.
- Říkáme, že automaty jsou v normovaném tvaru.

Normovaný tvar automatu

Mějme deterministický konečný automat $A = (Q, \Sigma, \delta, q_0, F)$ bez nedosažitelných stavů.

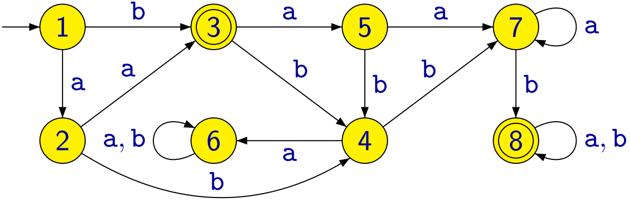
Předpokládejme, že máme určité uspořádání prvků abecedy Σ, kterým je určeno uspořádání $<_L$ na Σ^*.

Označme pro každý stav $q \in Q$ symbolem u_q nejmenší slovo podle uspořádání $<_L$ takové, že $q_0 \xrightarrow{u_q} q$.

DKA A je v **normovaném tvaru**, pokud:

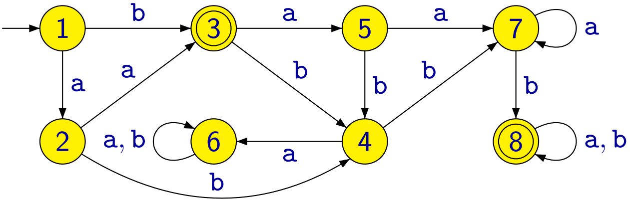
- $Q = \{1, 2, 3, \ldots, n\}$ pro nějaké $n \geq 1$.
- 1 je počáteční stav.
- Pro každou dvojici stavů $i, j \in \{1, 2, \ldots, n\}$ takovou, že $i < j$, platí $u_i <_L u_j$.

Normovaný tvar automatu



- Do stavu 4 se dostaneme slovy ab, bb, aab, bab, tedy $u_4 = ab$.
- Do stavu 5 se dostaneme slovy ba, aaa, tedy $u_5 = ba$.
- $ab <_L ba$, proto jsou stavy $4, 5$ označeny v tomto pořadí.

Normovaný tvar automatu



- Do stavu 4 se dostaneme slovy ab, bb, aab, bab, tedy $u_4 = ab$.
- Do stavu 5 se dostaneme slovy ba, aaa, tedy $u_5 = ba$.
- $ab <_L ba$, proto jsou stavy $4, 5$ označeny v tomto pořadí.
- Pro ostatní stavy jsou nejmenší slova následující:

$$\begin{array}{ll} u_1 = \varepsilon & u_5 = ba \\ u_2 = a & u_6 = aba \\ u_3 = b & u_7 = abb \\ u_4 = ab & u_8 = abbb \end{array}$$

Automaty a operace na jazycích

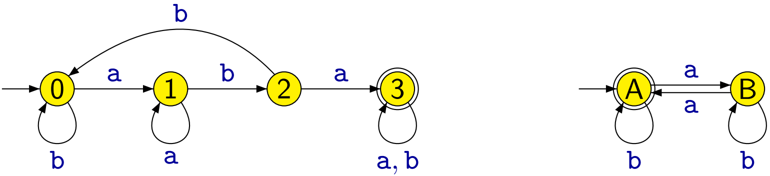
Pro jednoduché jazyky můžeme být schopni navrhnout automat přímo. Pro složitější jazyky už to může být obtížnější.

Často můžeme být schopni složitější jazyk popsat jako výsledek nějaké operace (sjednocení, průnik, doplněk, zřetězení, iterace, zrcadlový obraz apod.) aplikované na nějaké jednodušší jazyky.

V takém případě může být schůdnější následující „modulární“ postup:

- Nejprve sestrojit automaty pro ony jednodušší jazyky.
- Z vytvořených automatů určitou algoritmickou konstrukcí vytvořit automat rozpoznávající jazyk, který je výsledkem příslušné operace aplikované na jazyky rozpoznávané původními automaty.

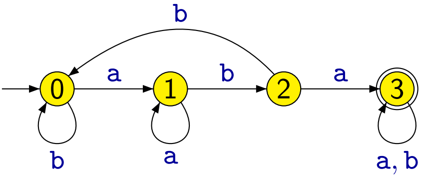
Máme následující dva automaty:

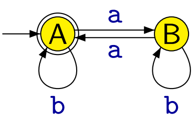


Příjmou oba slovo `ababb`?

Automat pro průnik jazyků

Máme následující dva automaty:



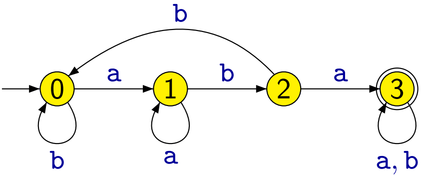


Příjmou oba slovo ababb?

Můžeme postupně nechat přečíst slovo oběma automatům. Odpověď bude Ano, pokud oba odpoví Ano.

Automat pro průnik jazyků

Máme následující dva automaty:



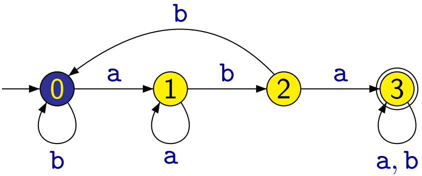
Příjmou oba slovo ababb?

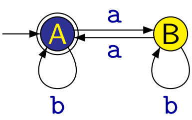
Můžeme postupně nechat přečíst slovo oběma automatům. Odpověď bude Ano, pokud oba odpoví Ano.

Místo toho můžeme nechat oba automaty číst dané slovo současně.

Automat pro průnik jazyků

Máme následující dva automaty:





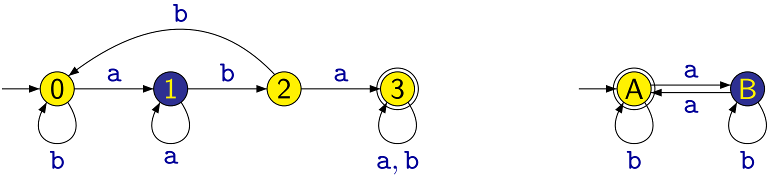
Příjmou oba slovo ababb?

Můžeme postupně nechat přečíst slovo oběma automatům. Odpověď bude ANO, pokud oba odpoví ANO.

Místo toho můžeme nechat oba automaty číst dané slovo současně.

Automat pro průnik jazyků

Máme následující dva automaty:



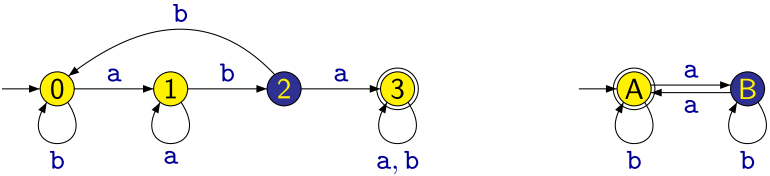
Příjmou oba slovo ababb?

Můžeme postupně nechat přečíst slovo oběma automatům. Odpověď bude Ano, pokud oba odpoví Ano.

Místo toho můžeme nechat oba automaty číst dané slovo současně.

Automat pro průnik jazyků

Máme následující dva automaty:

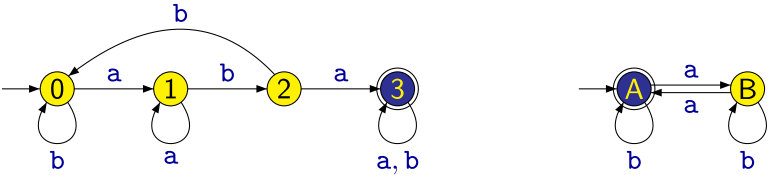


Příjmou oba slovo ababb?

Můžeme postupně nechat přečíst slovo oběma automatům. Odpověď bude Ano, pokud oba odpoví Ano.

Místo toho můžeme nechat oba automaty číst dané slovo současně.

Máme následující dva automaty:



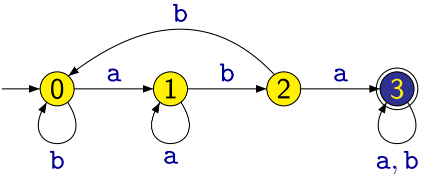
Příjmou oba slovo ababb?

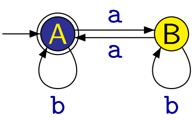
Můžeme postupně nechat přečíst slovo oběma automatům. Odpověď bude Ano, pokud oba odpoví Ano.

Místo toho můžeme nechat oba automaty číst dané slovo současně.

Automat pro průnik jazyků

Máme následující dva automaty:





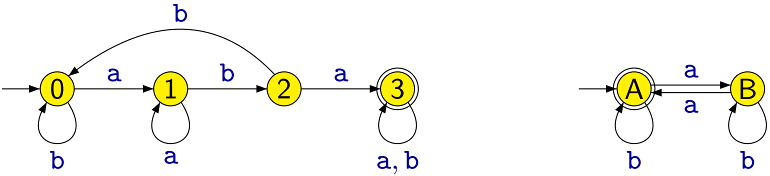
Příjmou oba slovo ababb?

Můžeme postupně nechat přečíst slovo oběma automatům. Odpověď bude ANO, pokud oba odpoví ANO.

Místo toho můžeme nechat oba automaty číst dané slovo současně.

Automat pro průnik jazyků

Máme následující dva automaty:



Příjmou oba slovo `ababb`?

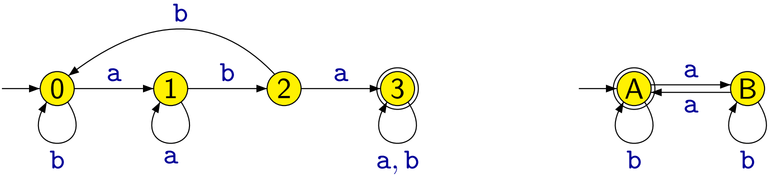
Můžeme postupně nechat přečíst slovo oběma automatům. Odpověď bude Ano, pokud oba odpoví Ano.

Místo toho můžeme nechat oba automaty číst dané slovo současně.

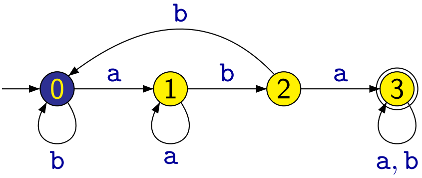
Při tomto čtení si stačí v každém kroku pamatovat dvojici stavů, ve kterých se oba automaty nacházejí.

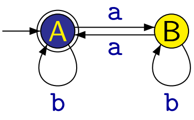
Toto může být realizováno konečným automatem, který simuluje činnost obou automatů současně! (Dvojic stavů je konečně mnoho.)

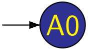
Automat pro průnik jazyků



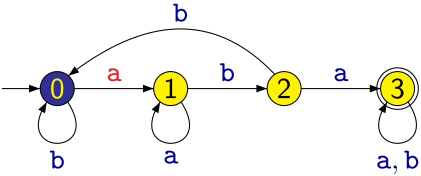
Automat pro průnik jazyků

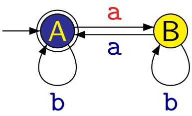


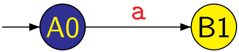




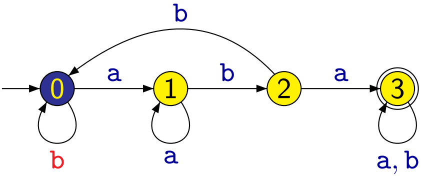
Automat pro průnik jazyků

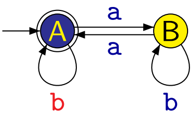


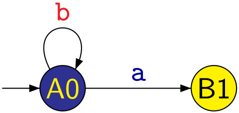




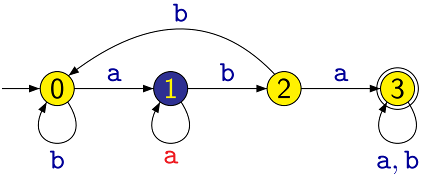
Automat pro průnik jazyků

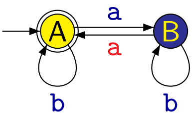


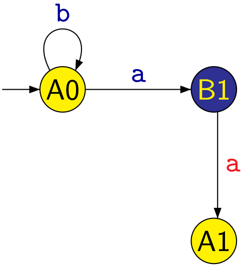


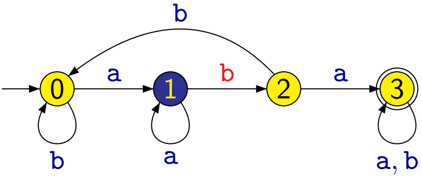


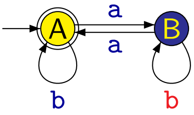
Automat pro průnik jazyků

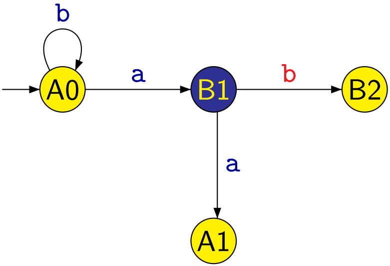




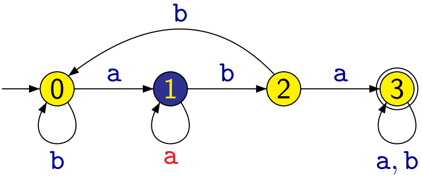


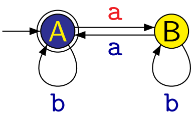

b
0
a
1
b
2
a
3
b
a
a, b

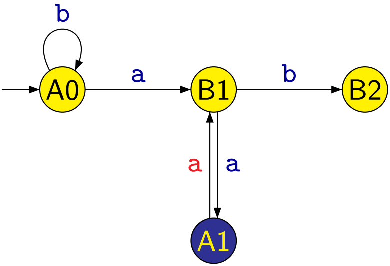

A
a
B
a
b
b


b
A0
a
B1
b
B2
a
A1

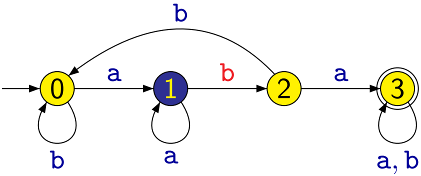
Automat pro průnik jazyků

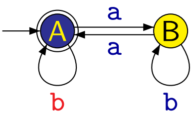


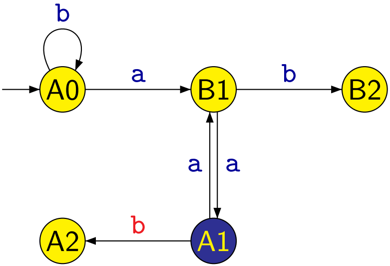




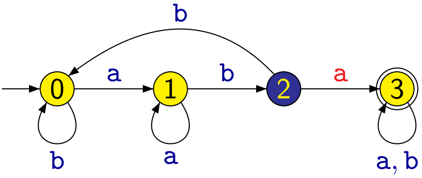
Automat pro průnik jazyků

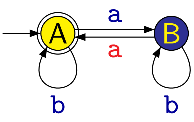


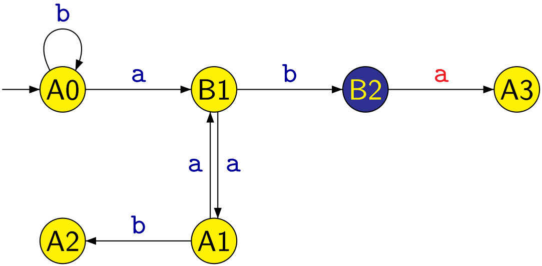




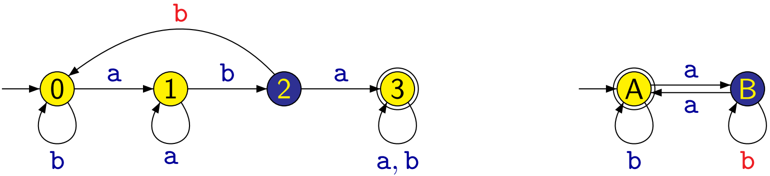
Automat pro průnik jazyků

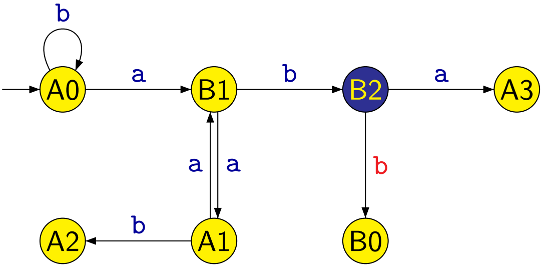




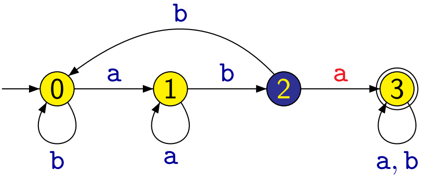


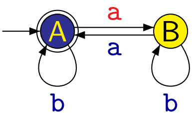
Automat pro průnik jazyků

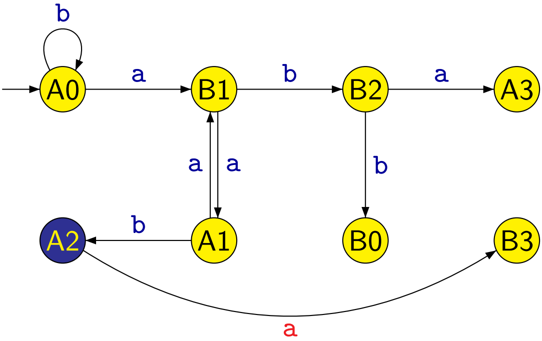




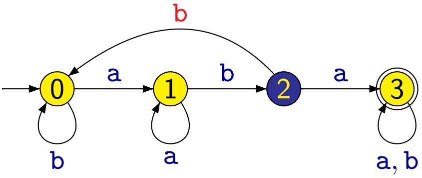
Automat pro průnik jazyků

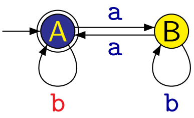


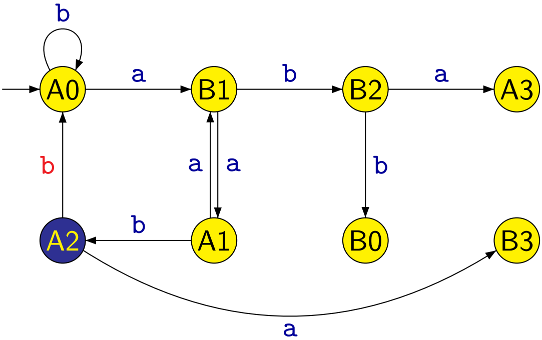




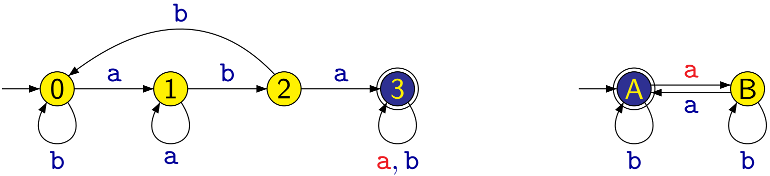
Automat pro průnik jazyků

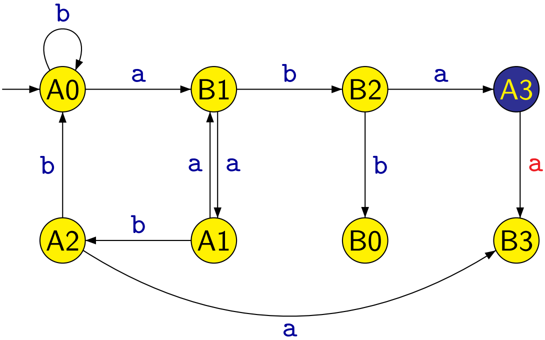




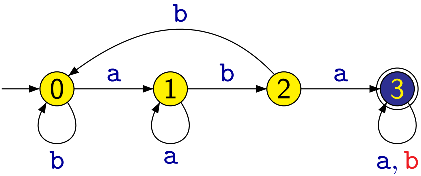


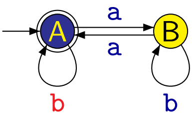
Automat pro průnik jazyků

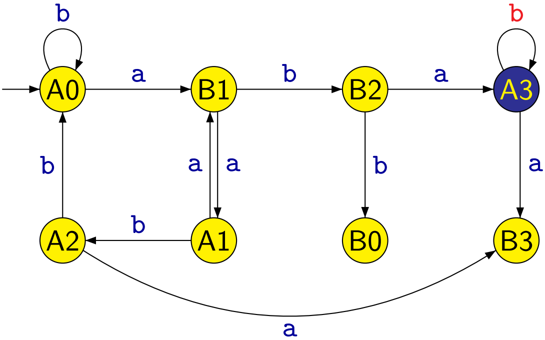




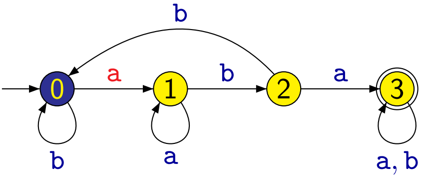
Automat pro průnik jazyků

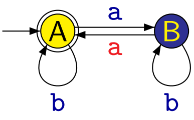


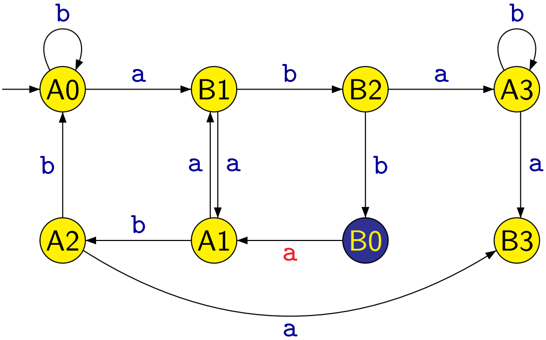




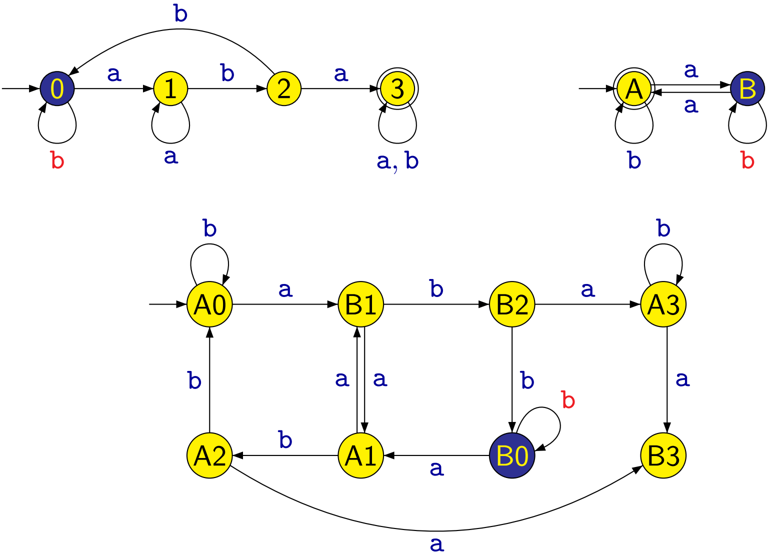
Automat pro průnik jazyků



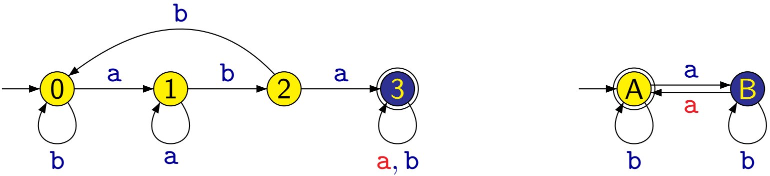


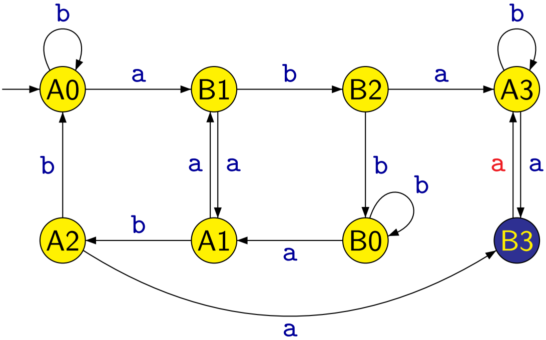


Automat pro průnik jazyků

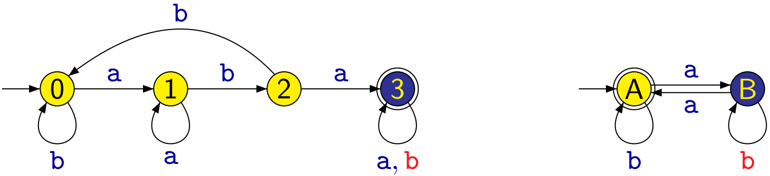


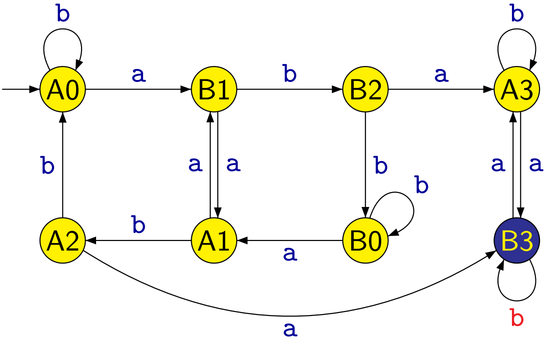
Automat pro průnik jazyků



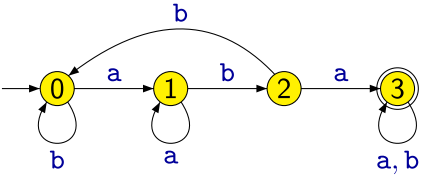


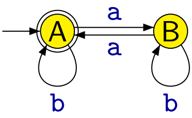
Automat pro průnik jazyků

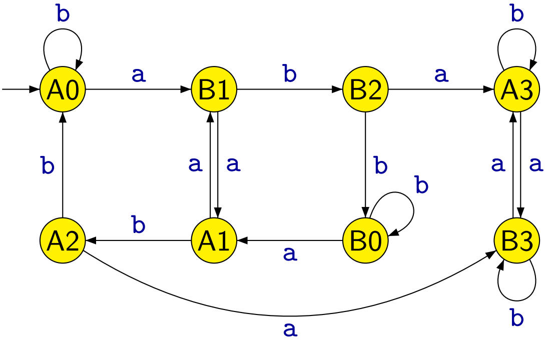




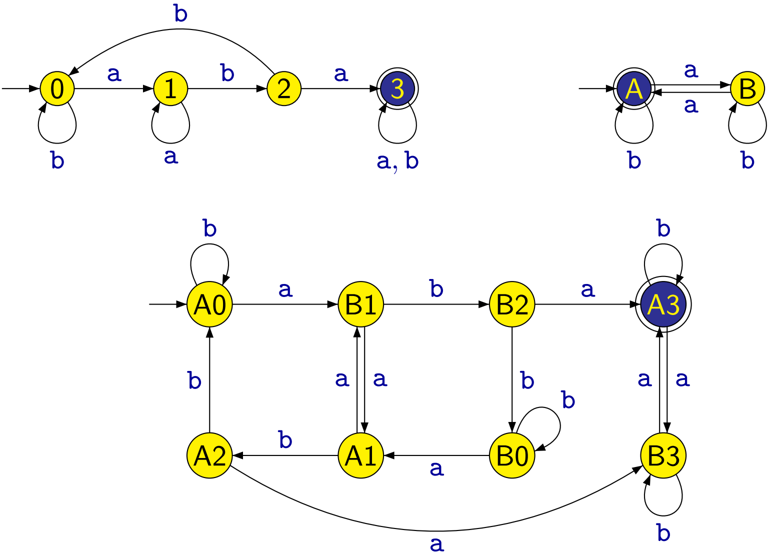
Automat pro průnik jazyků



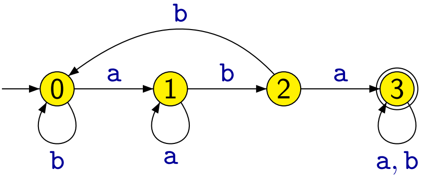


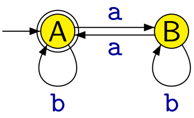


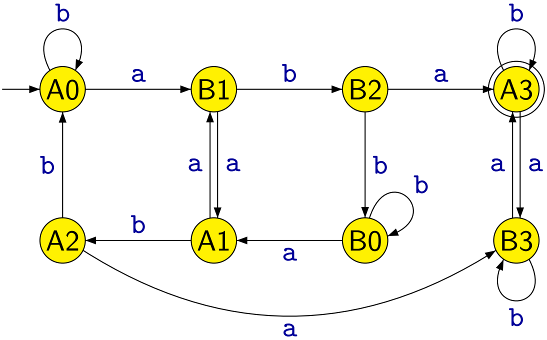
Automat pro průnik jazyků



Automat pro průnik jazyků







Automat pro průnik jazyků

Formálně můžeme popsat tuto konstrukci následovně:

Předpokládáme, že máme dva deterministické konečné automaty $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ a $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$.

K nim setrojíme DKA $A = (Q, \Sigma, \delta, q_0, F)$ kde:

- $Q = Q_1 \times Q_2$
- $\delta(\,(q_1, q_2), a\,) = (\,\delta_1(q_1, a), \delta_2(q_2, a)\,)$ pro všechna $q_1 \in Q_1$, $q_2 \in Q_2$, $a \in \Sigma$
- $q_0 = (q_{01}, q_{02})$
- $F = F_1 \times F_2$

Není težké ověřit, že pro libovolné slovo $w \in \Sigma^*$ platí, že $w \in L(A)$ právě tehdy, když $w \in L(A_1)$ a $w \in L(A_2)$, tj.

$$L(A) = L(A_1) \cap L(A_2)$$

Průnik regulárních jazyků

Věta

Jestliže jazyky $L_1, L_2 \subseteq \Sigma^*$ jsou regulární, pak také jazyk $L_1 \cap L_2$ je regulární.

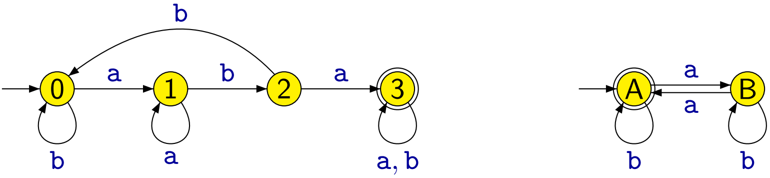
Důkaz: Předpokládejme, že A_1 a A_2 jsou deterministické konečné automaty takové, že

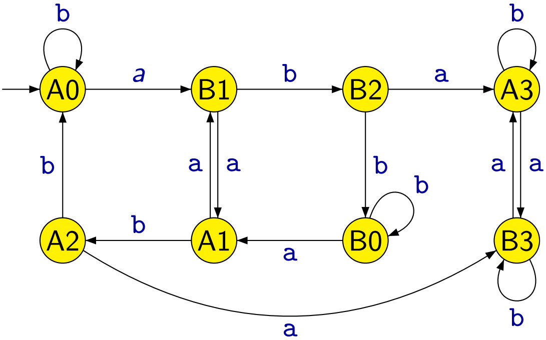
$$L_1 = L(A_1) \qquad\qquad L_2 = L(A_2)$$

Popsanou konstrukcí k nim můžeme sestrojit deterministický konečný automat A takový, že

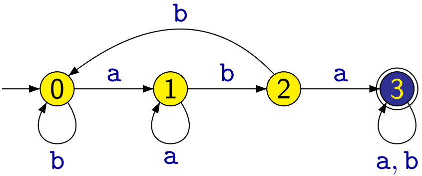
$$L(A) = L(A_1) \cap L(A_2) = L_1 \cap L_2$$

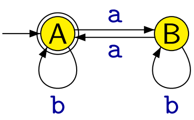
Automat pro sjednocení jazyků

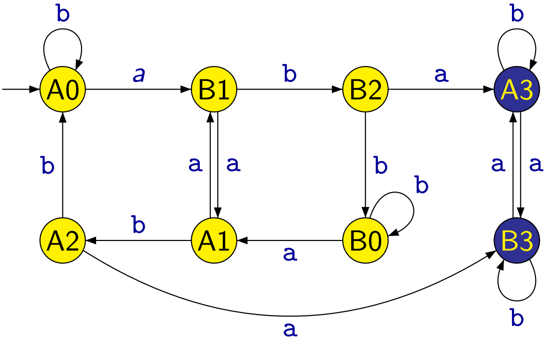




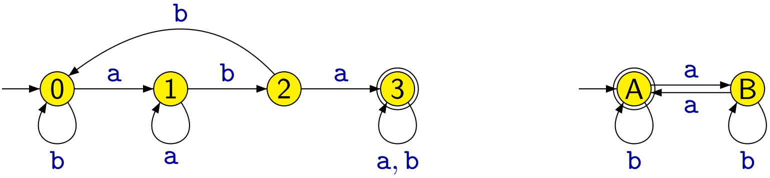
Automat pro sjednocení jazyků

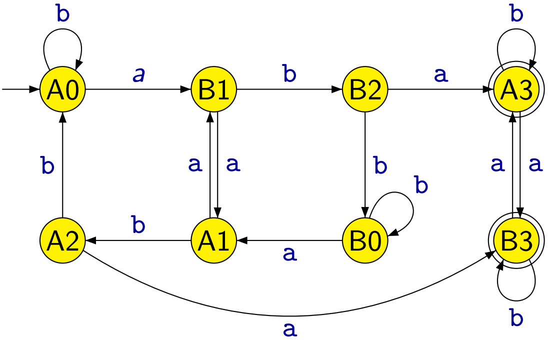




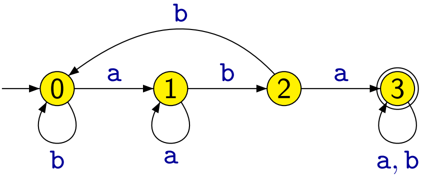


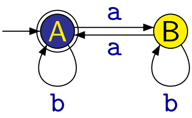
Automat pro sjednocení jazyků

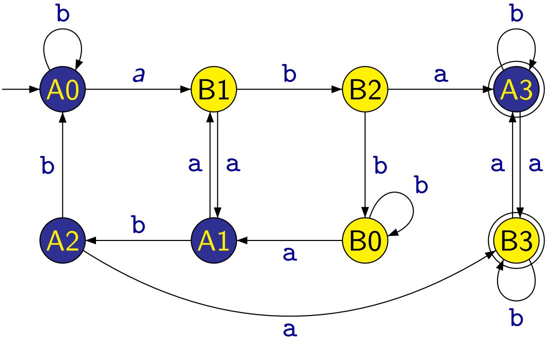




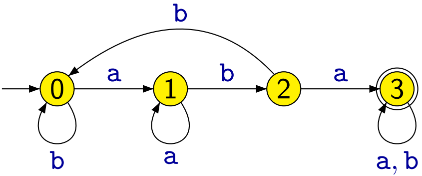
Automat pro sjednocení jazyků

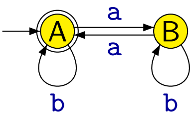


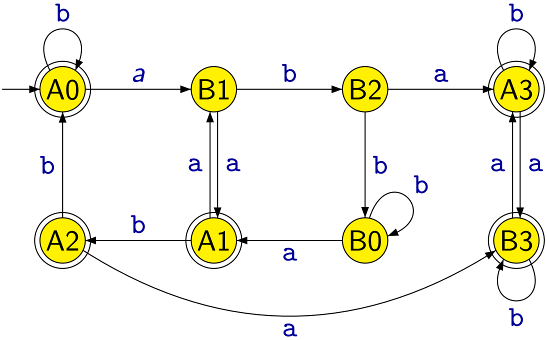




Automat pro sjednocení jazyků







Sjednocení regulárních jazyků

Konstukce automatu A, který přijímá **sjednocení** jazyků přijímaných automaty A_1 a A_2, tj. jazyk

$$L(A_1) \cup L(A_1)$$

je téměř stejná jako v případě automatu přijímajícího $L(A_1) \cap L(A_2)$.

Jediný rozdíl je v definici množiny přijímajících stavů:

- $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$

Sjednocení regulárních jazyků

Konstukce automatu A, který přijímá **sjednocení** jazyků přijímaných automaty A_1 a A_2, tj. jazyk

$$L(A_1) \cup L(A_1)$$

je téměř stejná jako v případě automatu přijímajícího $L(A_1) \cap L(A_2)$.

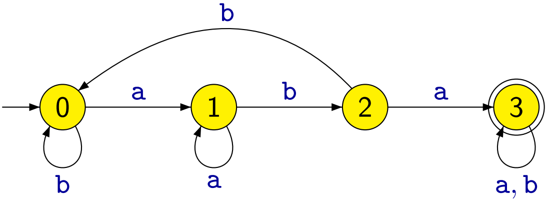
Jediný rozdíl je v definici množiny přijímajících stavů:

- $F = (F_1 \times Q_2) \cup (Q_1 \times F_2)$

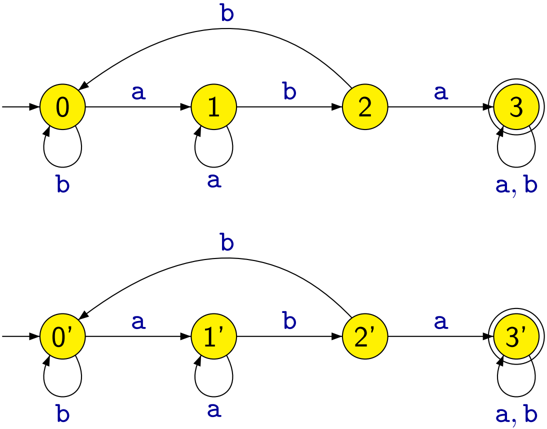
Věta

Jestliže jazyky $L_1, L_2 \subseteq \Sigma^*$ jsou regulární, pak také jazyk $L_1 \cup L_2$ je regulární.

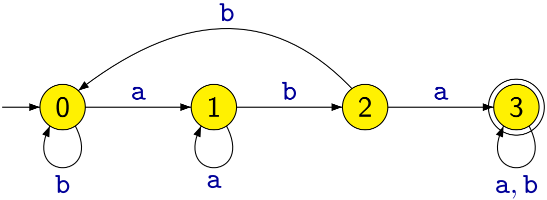
Automat pro doplněk jazyka

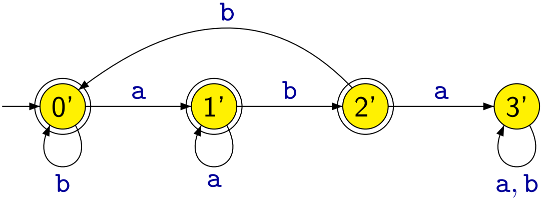


Automat pro doplněk jazyka



Automat pro doplněk jazyka





Doplněk regulárního jazyka

K DKA $A = (Q, \Sigma, \delta, q_0, F)$ sestrojíme DKA $A' = (Q, \Sigma, \delta, q_0, Q - F)$.

Je očividné, že pro každé slovo $w \in \Sigma^*$ platí, že $w \in L(A')$ právě tehdy, když $w \notin L(A)$, tj.

$$L(A') = \overline{L(A)}$$

Doplněk regulárního jazyka

K DKA $A = (Q, \Sigma, \delta, q_0, F)$ sestrojíme DKA $A' = (Q, \Sigma, \delta, q_0, Q - F)$.

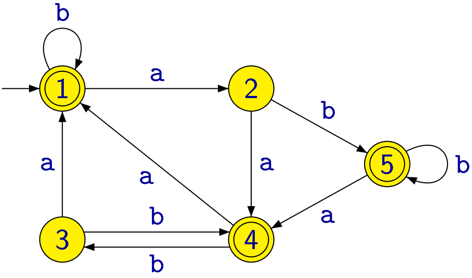
Je očividné, že pro každé slovo $w \in \Sigma^*$ platí, že $w \in L(A')$ právě tehdy, když $w \notin L(A)$, tj.

$$L(A') = \overline{L(A)}$$

Věta

Jestliže jazyk L je regulární, pak také jeho doplněk $\overline{L}$ je regulární.

Jazyk rozpoznávaný automatem

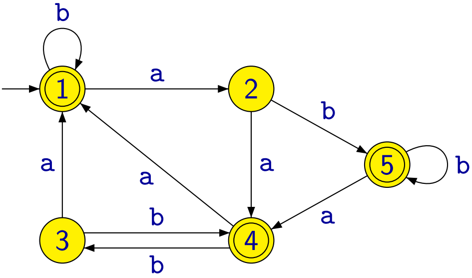


Uvažujme libovolnou **cestu** (na které se mohou opakovat vrcholy i hrany) v grafu takovou, že:

- Začíná v počátečním stavu.
- Končí v některém z přijímajících stavů.

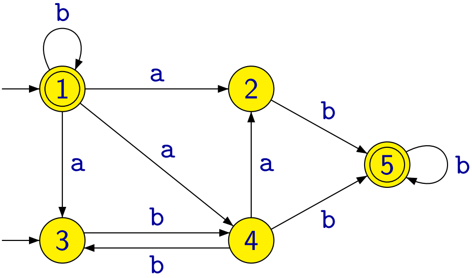
Symboly, jimiž jsou ohodnoceny hrany (tj. přechody) na této cestě, tvoří **slovo**, které je přijímáno daným automatem.

Jazyk rozpoznávaný automatem



Jazyk rozpoznávaný automatem je množina všech slov, pro které v grafu existuje takováto cesta.

Nedeterministický konečný automat



Je očividné, že pokud jazyk definujeme tímto způsobem, nemusíme se omezovat na grafy, kde:

- Z každého stavu vede právě jedna hrana označená daným symbolem abecedy.
- Máme právě jeden počáteční stav.

Nedeterministický konečný automat

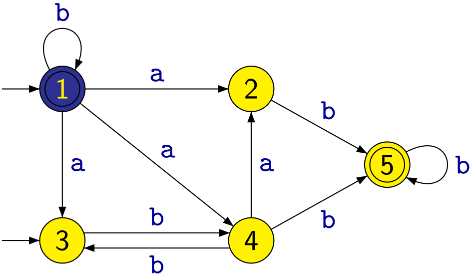
Takto obecněji definovaný automat se nazývá **nedeterministický konečný automat**.

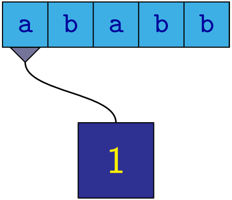
Rozdíly oproti deterministickým konečným automatům:

- Z jednoho stavu může vézt libovolný (i nulový) počet přechodů označených stejným symbolem.
- V automatu může být víc než jeden počáteční stav.

Pokud se na automat díváme jako na zařízení čtoucí slovo, vidíme, že jednomu slovu může odpovídat více než jeden výpočet (nebo naopak žádný).

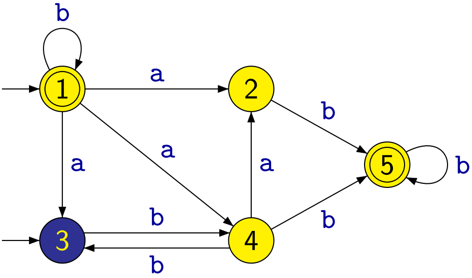
Nedeterministický konečný automat

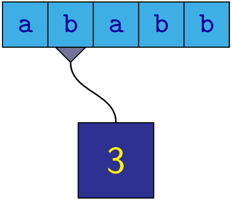




$(1, \mathtt{ababb})$

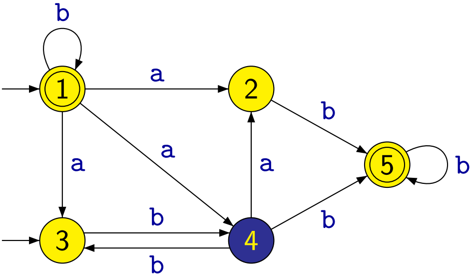
Nedeterministický konečný automat

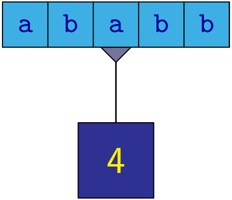




$$(1, \mathtt{ababb}) \vdash (3, \mathtt{babb})$$

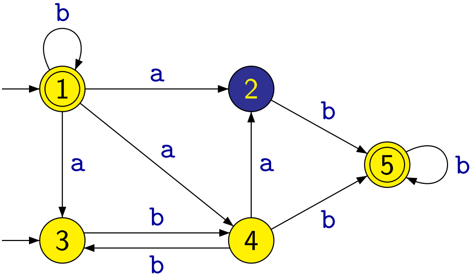
Nedeterministický konečný automat

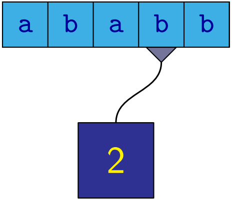




$$
\begin{aligned}
&(1, \mathtt{ababb}) \\
&\vdash (3, \mathtt{babb}) \\
&\vdash (4, \mathtt{abb})
\end{aligned}
$$

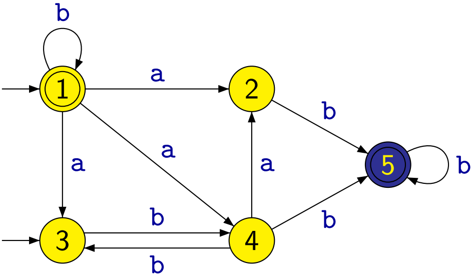
Nedeterministický konečný automat

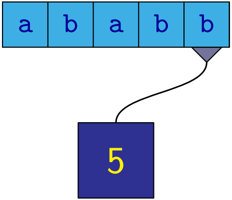




$$
\begin{aligned}
&(1, \mathtt{ababb}) \\
&\vdash (3, \mathtt{babb}) \\
&\vdash (4, \mathtt{abb}) \\
&\vdash (2, \mathtt{bb})
\end{aligned}
$$

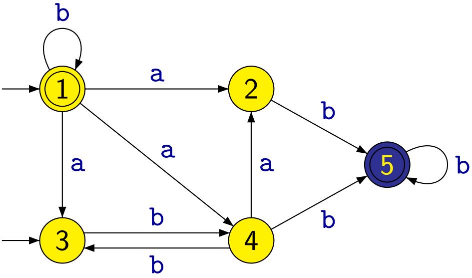
Nedeterministický konečný automat

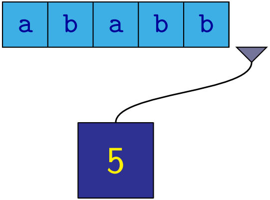




$(1, \mathtt{ababb})$
$\vdash (3, \mathtt{babb})$
$\vdash (4, \mathtt{abb})$
$\vdash (2, \mathtt{bb})$
$\vdash (5, \mathtt{b})$

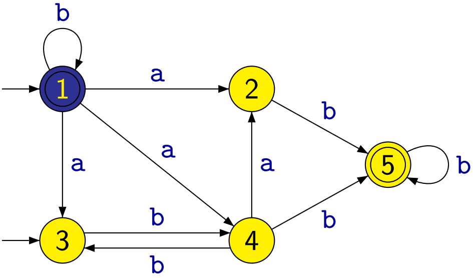
Nedeterministický konečný automat

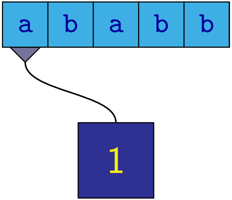




$$(1, \mathtt{ababb}) \vdash (3, \mathtt{babb}) \vdash (4, \mathtt{abb}) \vdash (2, \mathtt{bb}) \vdash (5, \mathtt{b}) \vdash (5, \varepsilon)$$

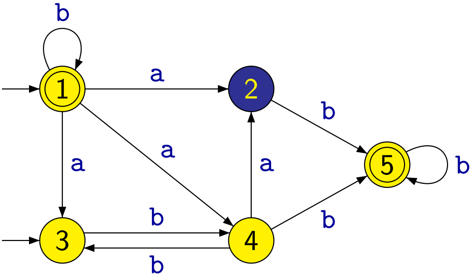
Nedeterministický konečný automat

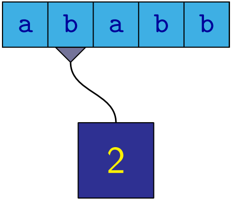




$(1, \mathtt{ababb})$

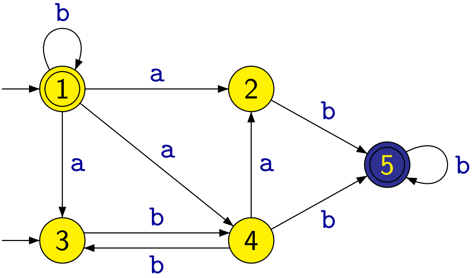
Nedeterministický konečný automat

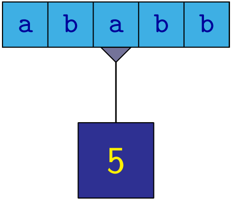




$(1, \mathtt{ababb})$
$\vdash (2, \mathtt{babb})$

Nedeterministický konečný automat

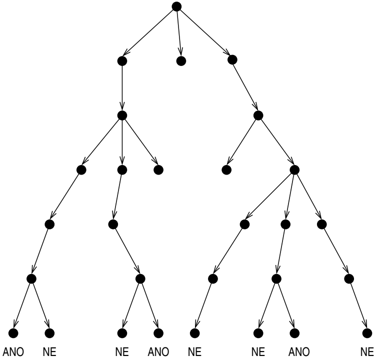




$$(1, \mathtt{ababb}) \vdash (2, \mathtt{babb}) \vdash (5, \mathtt{abb})$$

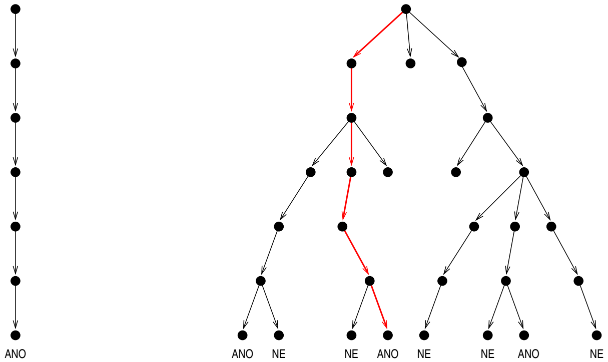
Nedeterministický konečný automat

Nedeterministický konečný automat přijímá dané slovo, jestliže **existuje** alespoň jeden jeho výpočet, který vede k přijetí tohoto slova.



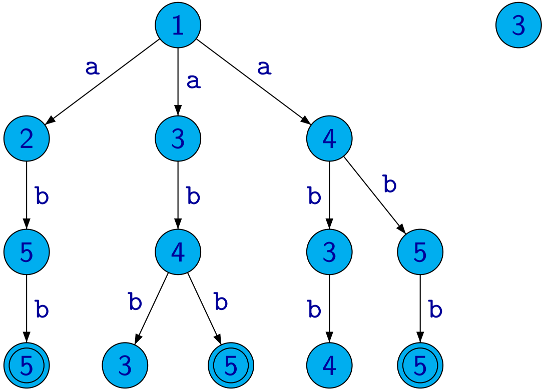
Nedeterministický konečný automat

Nedeterministický konečný automat přijímá dané slovo, jestliže **existuje** alespoň jeden jeho výpočet, který vede k přijetí tohoto slova.



Nedeterministický konečný automat

	a	b
$\leftrightarrow 1$	$2, 3, 4$	1
2	$-$	5
$\rightarrow 3$	$-$	4
4	2	$3, 5$
$\leftarrow 5$	$-$	5



Příklad: Les reprezentující všechny možné výpočty nad slovem abb.

Nedeterministický konečný automat

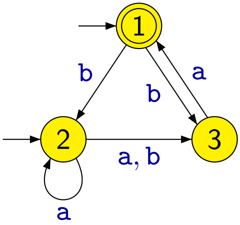
Formálně je **nedeterministický konečný automat** definován jako pětice

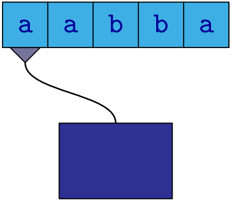
$$(Q, \Sigma, \delta, I, F)$$

kde:

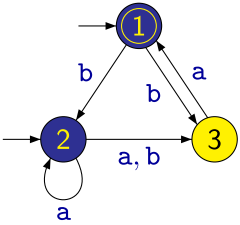
- Q je konečná množina **stavů**
- Σ je konečná **abeceda**
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ je **přechodová funkce**
- $I \subseteq Q$ je množina **počátečních stavů**
- $F \subseteq Q$ je množina **přijímajících stavů**

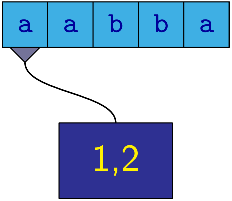
Převod nedeterministického automatu na deterministický



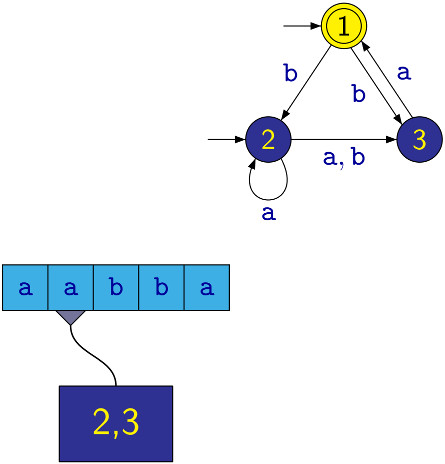


Převod nedeterministického automatu na deterministický

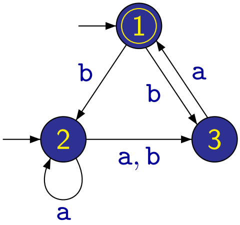


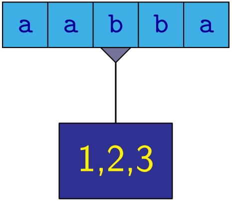


Převod nedeterministického automatu na deterministický

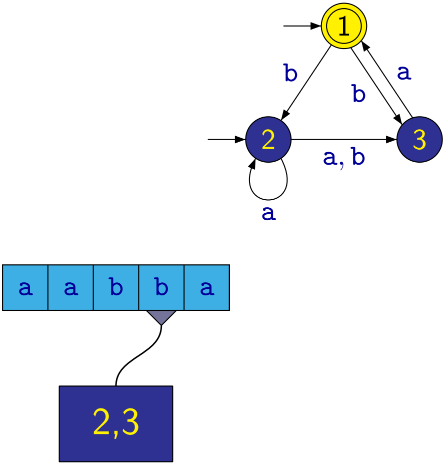


Převod nedeterministického automatu na deterministický

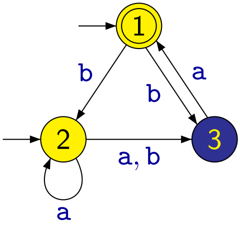


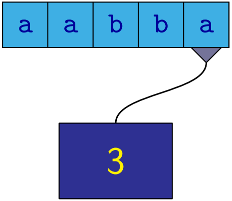


Převod nedeterministického automatu na deterministický

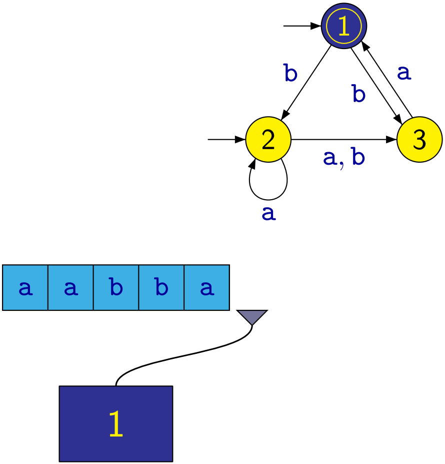


Převod nedeterministického automatu na deterministický

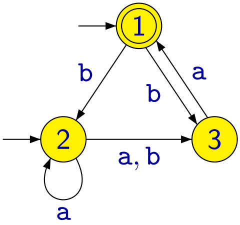




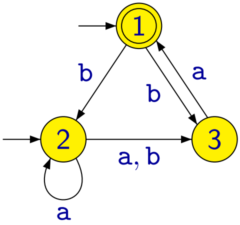
Převod nedeterministického automatu na deterministický



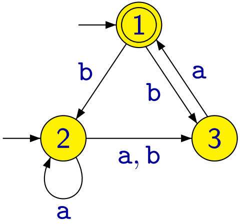
Převod nedeterministického automatu na deterministický

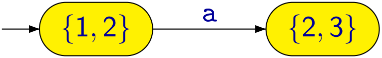


Převod nedeterministického automatu na deterministický

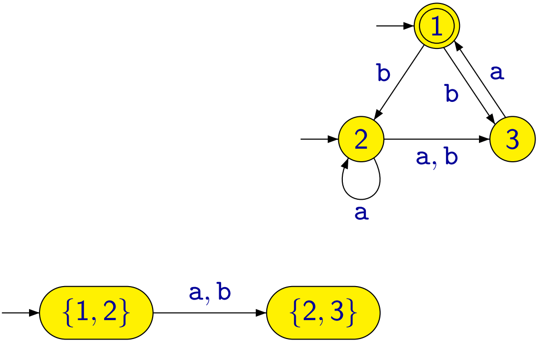


Převod nedeterministického automatu na deterministický

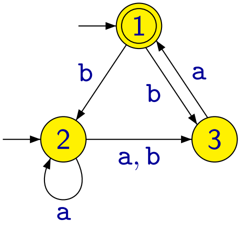


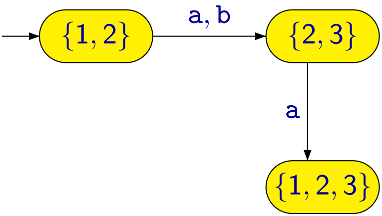


Převod nedeterministického automatu na deterministický

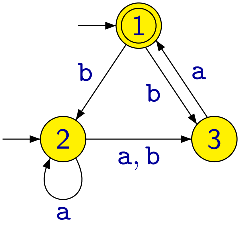


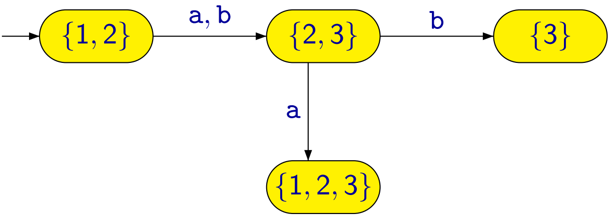
Převod nedeterministického automatu na deterministický



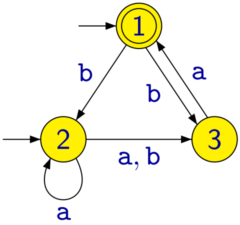


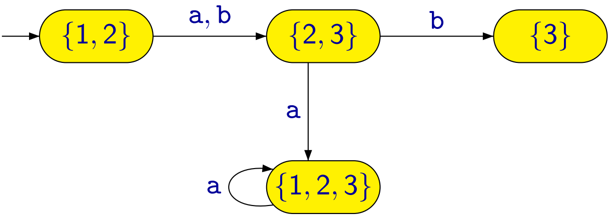
Převod nedeterministického automatu na deterministický



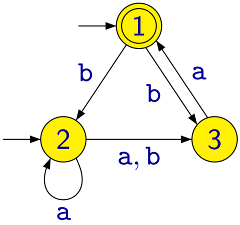


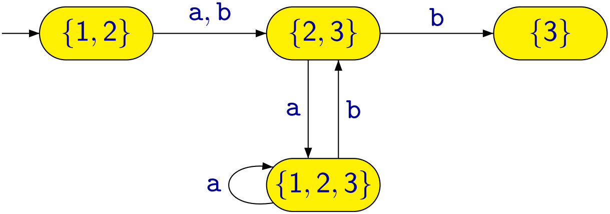
Převod nedeterministického automatu na deterministický



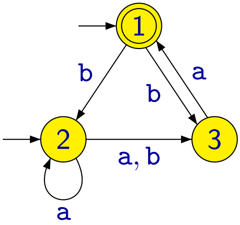


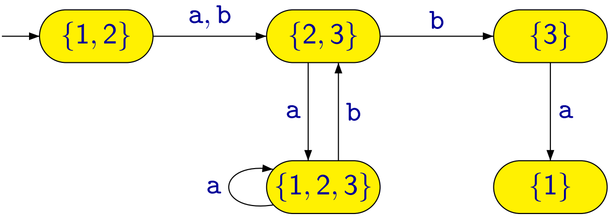
Převod nedeterministického automatu na deterministický



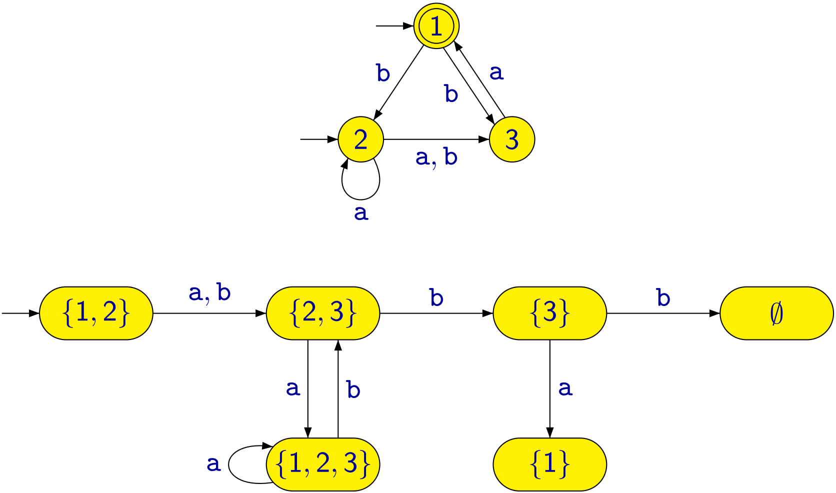


Převod nedeterministického automatu na deterministický

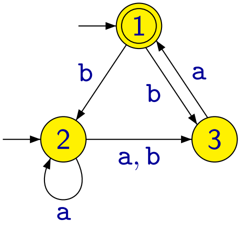


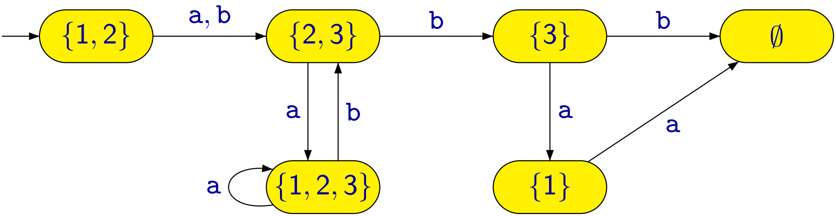


Převod nedeterministického automatu na deterministický

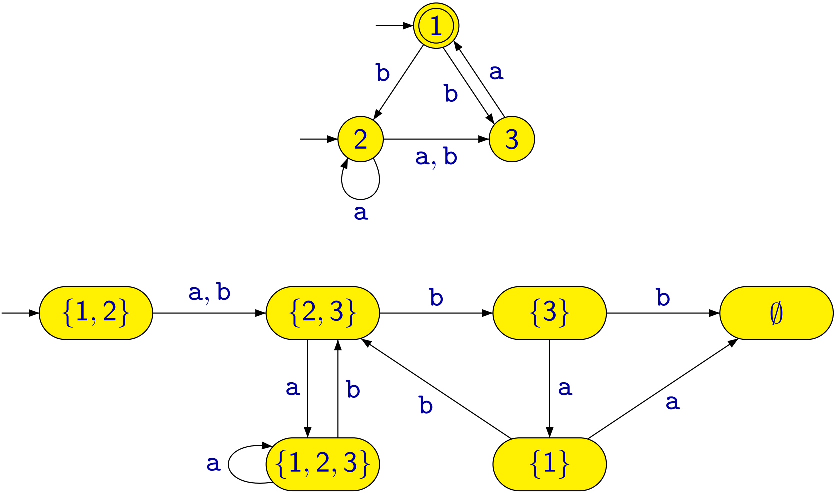


Převod nedeterministického automatu na deterministický

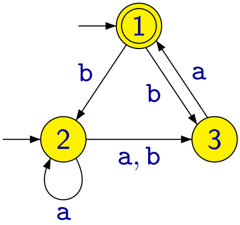


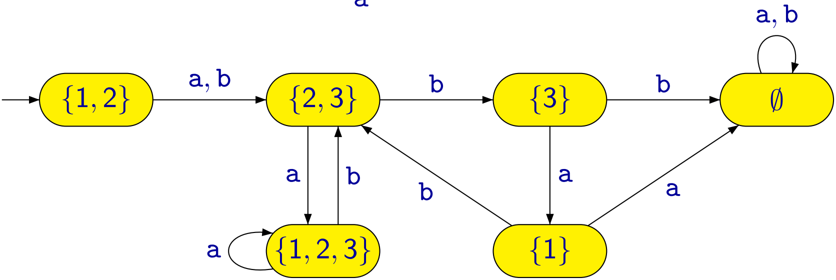


Převod nedeterministického automatu na deterministický

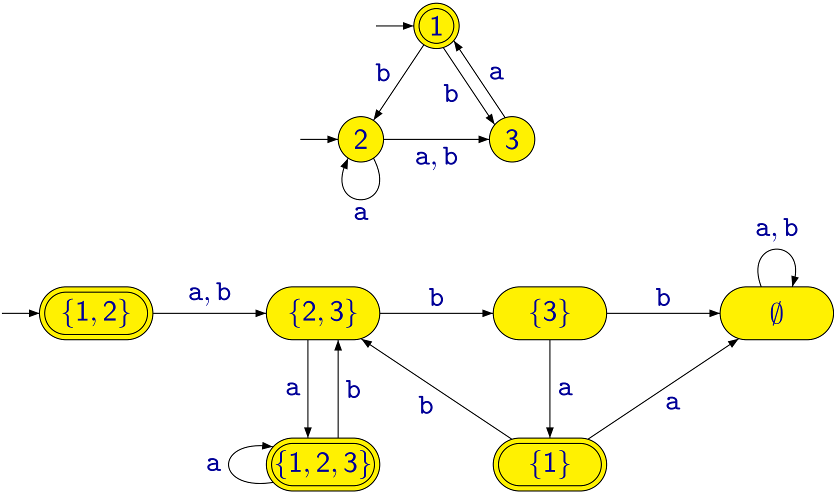


Převod nedeterministického automatu na deterministický





Převod nedeterministického automatu na deterministický



Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	–	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	–

	a	b

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	
$\{2,3\}$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2, 3$
$\rightarrow 2$	$2, 3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1, 2\}$	$\{2, 3\}$	$\{2, 3\}$
$\{2, 3\}$	$\{1, 2, 3\}$	

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2, 3$
$\rightarrow 2$	$2, 3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1, 2\}$	$\{2, 3\}$	$\{2, 3\}$
$\{2, 3\}$	$\{1, 2, 3\}$	
$\leftarrow \{1, 2, 3\}$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2, 3$
$\rightarrow 2$	$2, 3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1, 2\}$	$\{2, 3\}$	$\{2, 3\}$
$\{2, 3\}$	$\{1, 2, 3\}$	$\{3\}$
$\leftarrow \{1, 2, 3\}$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$		
$\{3\}$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2, 3$
$\rightarrow 2$	$2, 3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1, 2\}$	$\{2, 3\}$	$\{2, 3\}$
$\{2, 3\}$	$\{1, 2, 3\}$	$\{3\}$
$\leftarrow \{1, 2, 3\}$	$\{1, 2, 3\}$	
$\{3\}$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2, 3$
$\rightarrow 2$	$2, 3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1, 2\}$	$\{2, 3\}$	$\{2, 3\}$
$\{2, 3\}$	$\{1, 2, 3\}$	$\{3\}$
$\leftarrow \{1, 2, 3\}$	$\{1, 2, 3\}$	$\{2, 3\}$
$\{3\}$	$\{1\}$	
$\leftarrow \{1\}$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	–	$2, 3$
$\rightarrow 2$	$2, 3$	3
3	1	–

	a	b
$\leftrightarrow \{1, 2\}$	$\{2, 3\}$	$\{2, 3\}$
$\{2, 3\}$	$\{1, 2, 3\}$	$\{3\}$
$\leftarrow \{1, 2, 3\}$	$\{1, 2, 3\}$	$\{2, 3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$		
$\emptyset$		

	a	b
$\leftrightarrow 1$	$-$	$2, 3$
$\rightarrow 2$	$2, 3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1, 2\}$	$\{2, 3\}$	$\{2, 3\}$
$\{2, 3\}$	$\{1, 2, 3\}$	$\{3\}$
$\leftarrow \{1, 2, 3\}$	$\{1, 2, 3\}$	$\{2, 3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$	$\emptyset$	
$\emptyset$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$	$\emptyset$	$\{2,3\}$
$\emptyset$		

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$	$\emptyset$	$\{2,3\}$
$\emptyset$	$\emptyset$	$\emptyset$

Převod nedeterministického automatu na deterministický

	a	b
$\leftrightarrow 1$	$-$	$2,3$
$\rightarrow 2$	$2,3$	3
3	1	$-$

	a	b
$\leftrightarrow \{1,2\}$	$\{2,3\}$	$\{2,3\}$
$\{2,3\}$	$\{1,2,3\}$	$\{3\}$
$\leftarrow \{1,2,3\}$	$\{1,2,3\}$	$\{2,3\}$
$\{3\}$	$\{1\}$	$\emptyset$
$\leftarrow \{1\}$	$\emptyset$	$\{2,3\}$
$\emptyset$	$\emptyset$	$\emptyset$

	a	b
$\leftrightarrow 1$	2	2
2	3	4
$\leftarrow 3$	3	2
4	5	6
$\leftarrow 5$	6	2
6	6	6

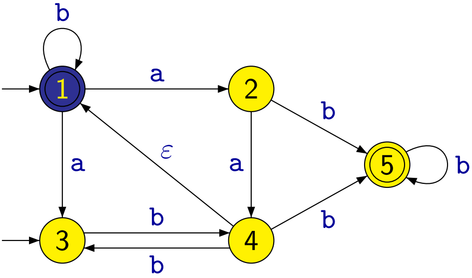
Převod nedeterministického automatu na deterministický

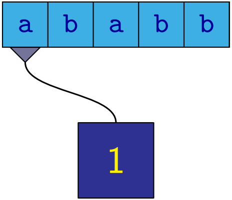
Poznámka: Při převodu nedeterministického automatu, který má n stavů, může mít výsledný deterministický automat až 2^n stavů.

Například při převodu automatu, který má 20 stavů, může vzniknout automat, který má $2^{20} = 1048576$ stavů.

Často má sice výsledný automat podstatně méně než 2^n stavů, nicméně tyto nejhorší případy občas nastávají.

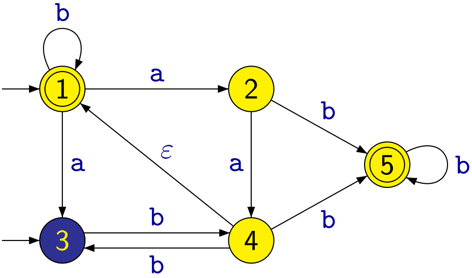
Zobecněný nedeterministický konečný automat

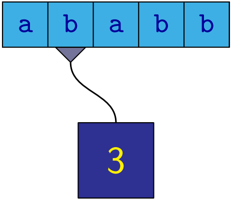




$(1, \mathtt{ababb})$

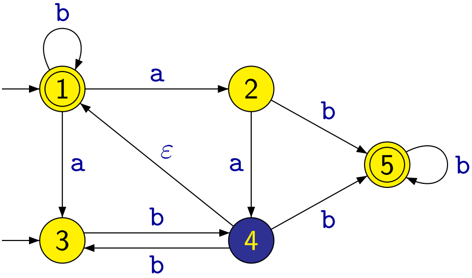
Zobecněný nedeterministický konečný automat

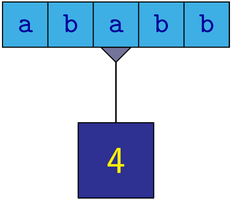




$$(1, \mathtt{ababb})$$
$$\vdash (3, \mathtt{babb})$$

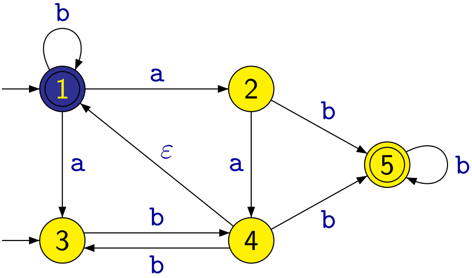
Zobecněný nedeterministický konečný automat

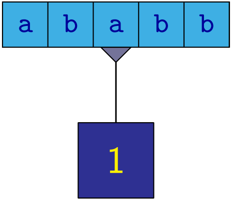




$(1, \texttt{ababb})$
$\vdash (3, \texttt{babb})$
$\vdash (4, \texttt{abb})$

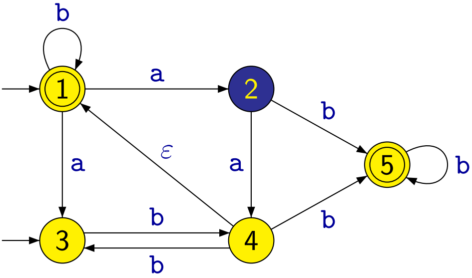
Zobecněný nedeterministický konečný automat

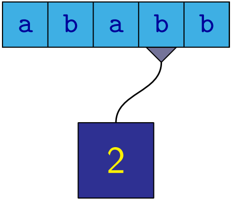




$$(1, \mathtt{ababb})$$
$$\vdash (3, \mathtt{babb})$$
$$\vdash (4, \mathtt{abb})$$
$$\vdash (1, \mathtt{abb})$$

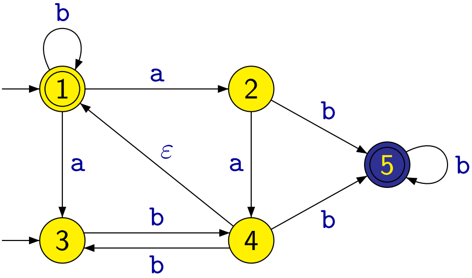
Zobecněný nedeterministický konečný automat

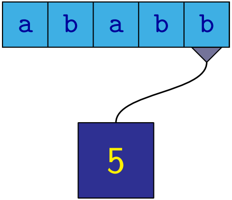




$$
\begin{aligned}
&(1, \mathtt{ababb})\\
&\vdash (3, \mathtt{babb})\\
&\vdash (4, \mathtt{abb})\\
&\vdash (1, \mathtt{abb})\\
&\vdash (2, \mathtt{bb})
\end{aligned}
$$

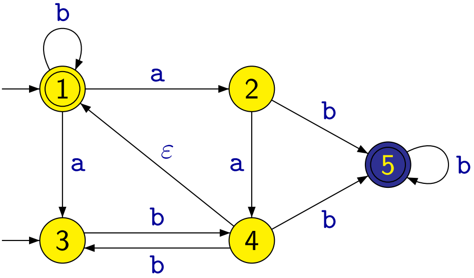
Zobecněný nedeterministický konečný automat

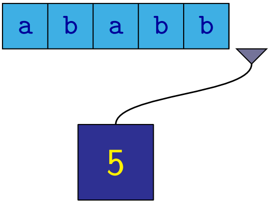




$$
\begin{aligned}
&(1, \mathtt{ababb}) \\
&\vdash (3, \mathtt{babb}) \\
&\vdash (4, \mathtt{abb}) \\
&\vdash (1, \mathtt{abb}) \\
&\vdash (2, \mathtt{bb}) \\
&\vdash (5, \mathtt{b})
\end{aligned}
$$

Zobecněný nedeterministický konečný automat





$$
\begin{aligned}
&(1, \mathtt{ababb}) \\
&\vdash (3, \mathtt{babb}) \\
&\vdash (4, \mathtt{abb}) \\
&\vdash (1, \mathtt{abb}) \\
&\vdash (2, \mathtt{bb}) \\
&\vdash (5, \mathtt{b}) \\
&\vdash (5, \varepsilon)
\end{aligned}
$$

Zobecněný nedeterministický konečný automat

Oproti nedeterministickému konečnému automatu má **zobecněný nedeterministický konečný automat** tzv. ε**-přechody**, tj. přechody označené symbolem ε.

Při provádění ε-přechodu se mění pouze stav řídící jednotky, ale hlava na pásce se neposouvá.

Poznámka: Výpočty zobecněného nedeterministického automatu mohou být libovolně dlouhé a dokonce i nekonečné (pokud graf obsahuje cyklus tvořený ε-přechody) bez ohledu na délku slova na pásce.

Zobecněný nedeterministický konečný automat

Formálně je **zobecněný nedeterministický konečný automat** definován jako pětice

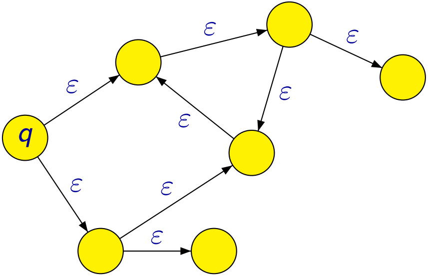
$$(Q, \Sigma, \delta, I, F)$$

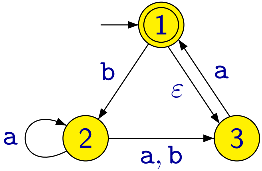
kde:

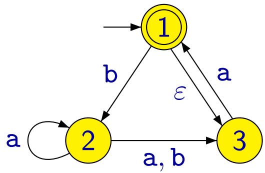
- Q je konečná množina **stavů**
- Σ je konečná **abeceda**
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ je **přechodová funkce**
- $I \subseteq Q$ je množina **počátečních stavů**
- $F \subseteq Q$ je množina **přijímajících stavů**

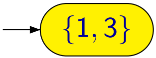
Převod na deterministický konečný automat

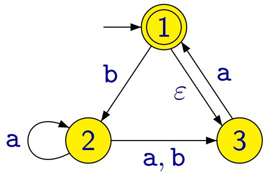
Zobecněný nedeterministický konečný automat je možné převést na deterministický podobnou konstrukcí jako nedeterministický konečný automat, s tím rozdílem, že do množin stavů musíme vždy přidat navíc i všechny stavy dosažitelné z již přidaných stavů nějakou sekvencí ε-přechodů.

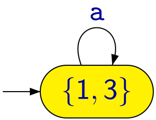


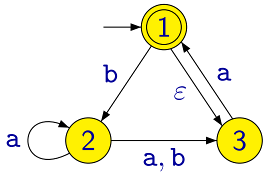

1
b
a
ε
a
2
3
a, b

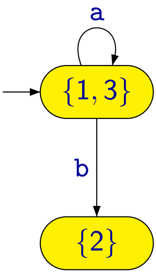

1
2
3
b
a
ε
a
a, b

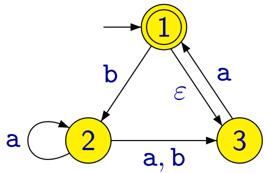

{1, 3}

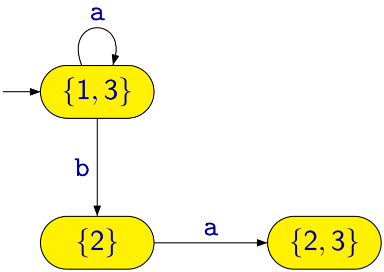

1
b
a
ε
a
2
3
a, b

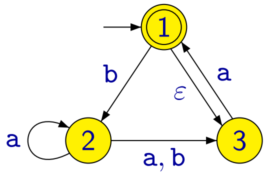

a
{1, 3}

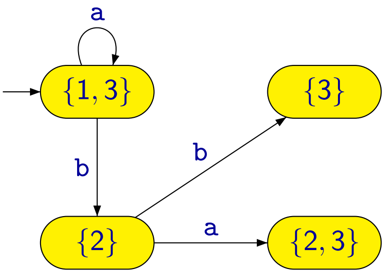

1
b
a
ε
2
3
a
a, b

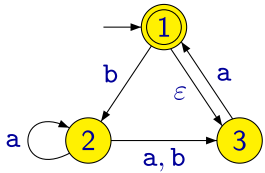

a
{1, 3}
b
{2}

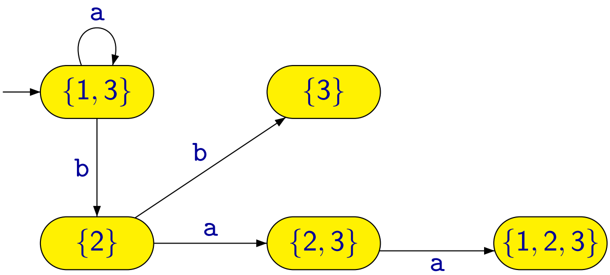

1
b
a
ε
a
2
a, b
3

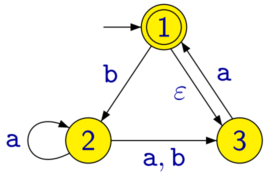

a
{1, 3}
b
{2}
a
{2, 3}

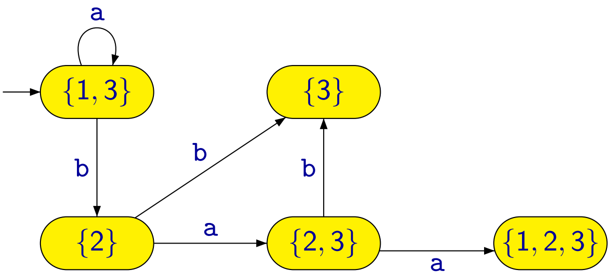

1
b
a
ε
a
2
3
a, b

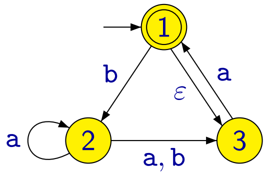

a
{1, 3}
{3}
b
b
{2}
a
{2, 3}

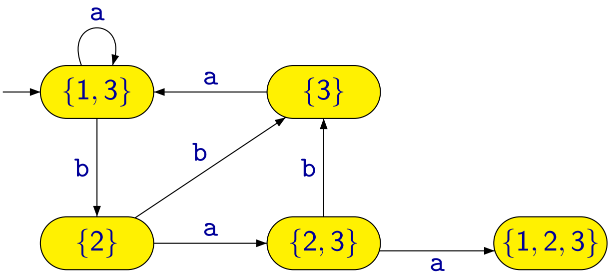

1
2
3
a
b
ε
a
a, b

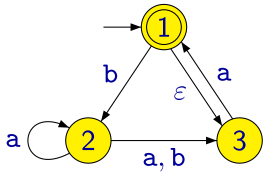

{1, 3}
{3}
{2}
{2, 3}
{1, 2, 3}
a
b
b
a
a

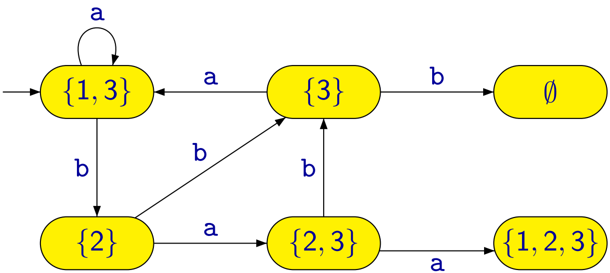

1
b
a
ε
a
2
a, b
3

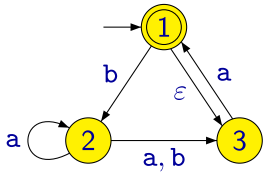

a
{1, 3}
{3}
b
b
b
{2}
a
{2, 3}
a
{1, 2, 3}

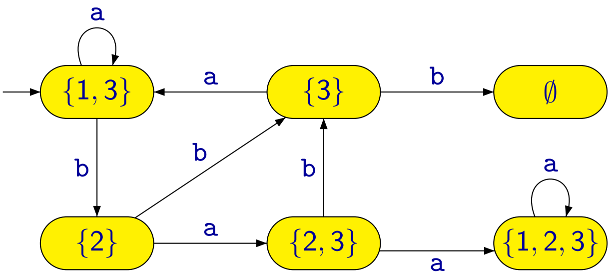

1
b
a
ε
a
2
a, b
3

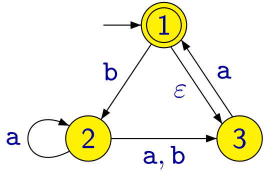

a
{1, 3}
a
{3}
b
b
b
{2}
a
{2, 3}
a
{1, 2, 3}

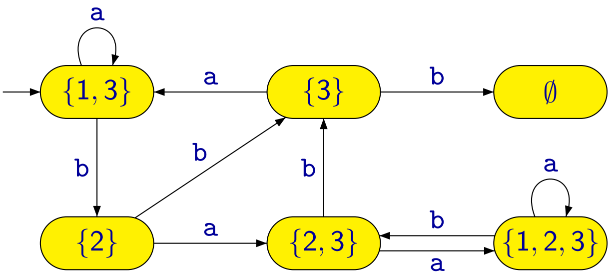

1
b
a
ε
a
2
3
a, b

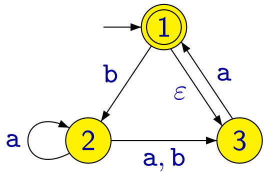

a
{1, 3}
a
{3}
b
∅
b
b
b
{2}
a
{2, 3}
a
{1, 2, 3}

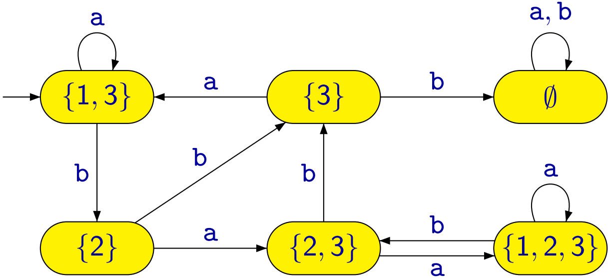

1
2
3
a
b
ε
a
a, b

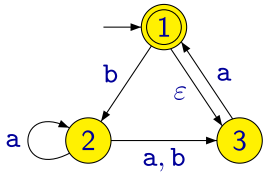

{1, 3}
{3}
∅
{2}
{2, 3}
{1, 2, 3}
a
a
b
b
b
b
a
a
a

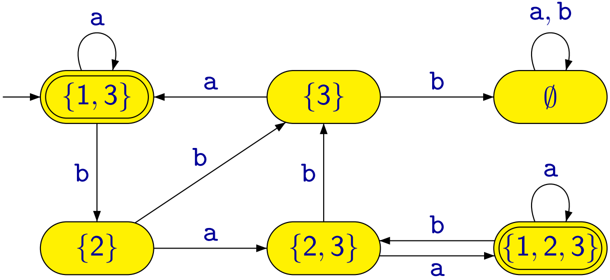

1
2
3
b
a
ε
a
a, b


{1, 3}
{3}
∅
{2}
{2, 3}
{1, 2, 3}
a
a
b
b
b
b
a
b
a
a


1
b
a
ε
a
2
3
a, b


a
a, b
{1, 3}
a
{3}
b
∅
b
b
b
a
{2}
a
{2, 3}
b
{1, 2, 3}
a

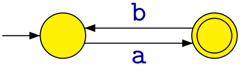

1
b
a
ε
a
2
a, b
3


a
a, b
{1, 3}
a
{3}
b
∅
b
b
b
a
{2}
a
{2, 3}
b
{1, 2, 3}
a

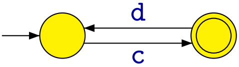
Zřetězení jazyků

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$

A_1:



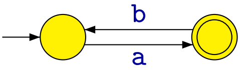
A_2:



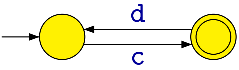
Zřetězení jazyků

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$

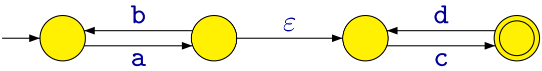
A_1:



A_2:



A:

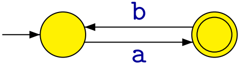


$L(A) = L(A_1) \cdot L(A_2)$

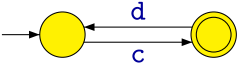
Zřetězení jazyků

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$

A_1:

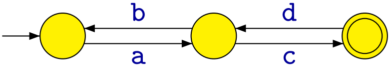


A_2:



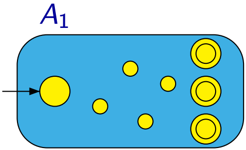
Chybná konstrukce:

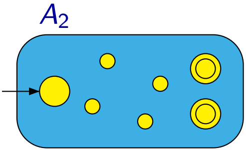
A:



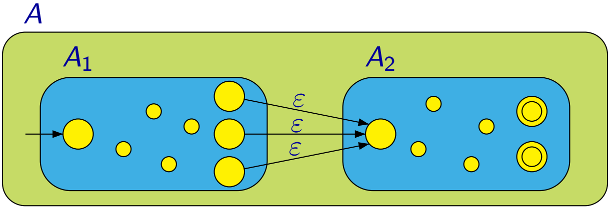
$\mathtt{acdbac} \in L(A)$, ale $\mathtt{acdbac} \notin L(A_1) \cdot L(A_2)$

Zřetězení jazyků

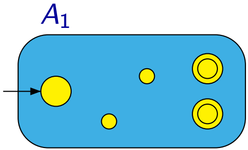




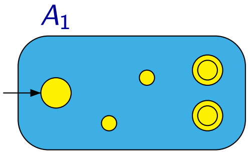
Zřetězení jazyků

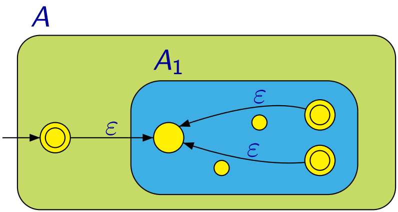


Iterace jazyka



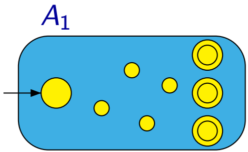
Iterace jazyka

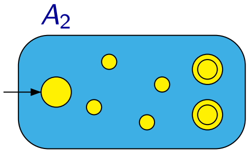




Sjednocení jazyků

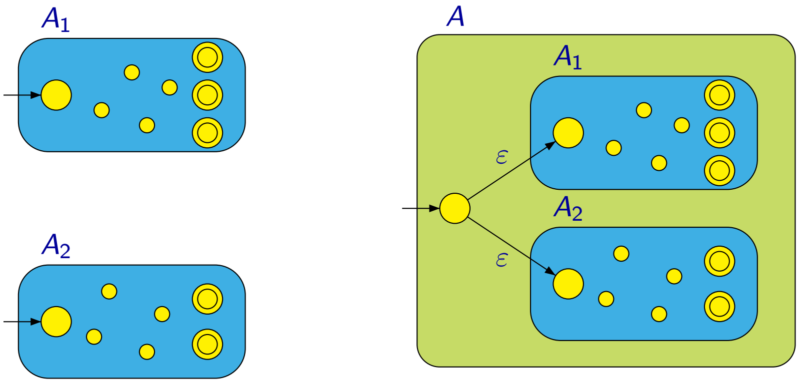
Alternativní konstrukce pro sjednocení jazyků:





Sjednocení jazyků

Alternativní konstrukce pro sjednocení jazyků:



Uzavřenost množiny vůči operacím

Předpokládejme, že máme dány množiny X a Y, kde $X \subseteq Y$, a dále nějakou operaci $f : Y \times Y \times \cdots \times Y \to Y$.

O množině X řekneme, že je **uzavřená** vůči operaci f, jestliže platí, že pokud $x_1, x_2, \cdots, x_k \in X$, pak $f(x_1, x_2, \cdots, x_k) \in X$.

Příklad: Uvažujme množinu přirozených čísel $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ a množinu reálných čísel $\mathbb{R}$.

- Množina $\mathbb{N}$ je uzavřená vůči operacím $+$ a $\times$, ale není uzavřená vůči operacím $-$ a $/$.

  Například $3 + 5 \in \mathbb{N}$, ale $3 - 5 \notin \mathbb{N}$ a $3/5 \notin \mathbb{N}$
- Množina $\mathbb{R}$ je uzavřená vůči operacím $+$, $-$, $\times$.
- Množina $\mathbb{R} - \{0\}$ je uzavřená vůči operaci $/$.

Uzavřenost třídy regulárních jazyků

Množina (všech) regulárních jazyků je uzavřená vůči operacím:

- sjednocení
- průniku
- doplňku
- zřetězení
- iteraci
- ...

Poznámka: Jazyk je regulární, pokud existuje konečný automat, který ho rozpoznává.

Existují i jazyky, které nejsou regulární.