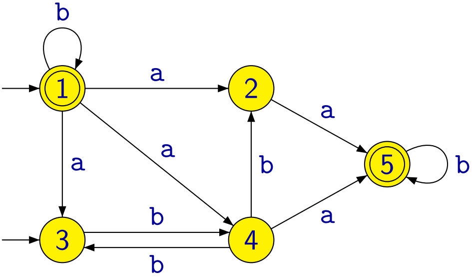
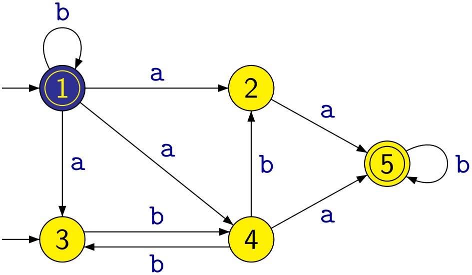
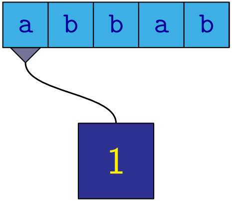
Nedeterministický konečný automat



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

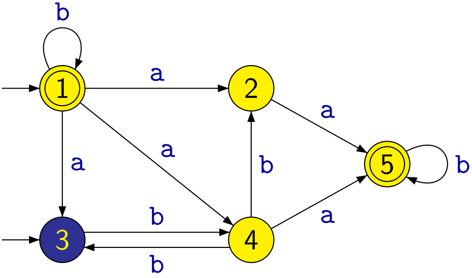
Nedeterministický konečný automat

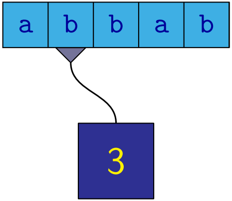




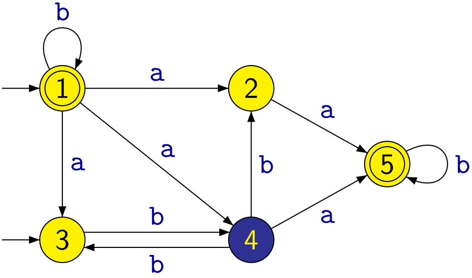
1

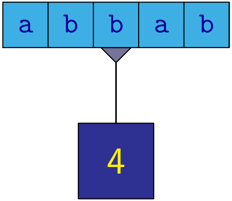
Nedeterministický konečný automat





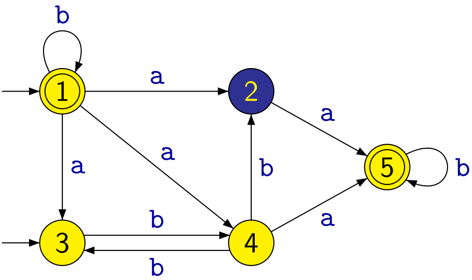
$$1 \xrightarrow{a} 3$$

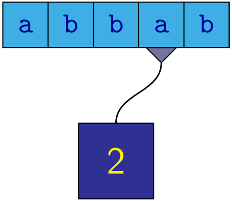




$$1 \xrightarrow{\texttt{a}} 3 \xrightarrow{\texttt{b}} 4$$

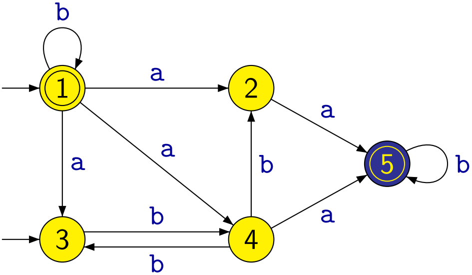
Nedeterministický konečný automat

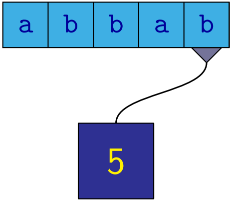




$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{b} 2$$

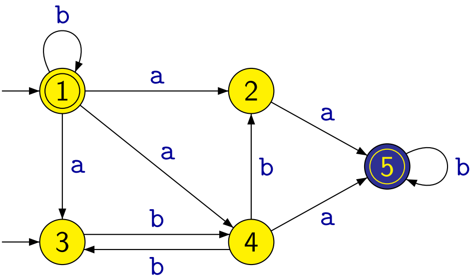
Nedeterministický konečný automat

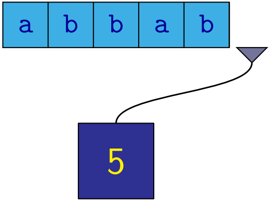




$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{b} 2 \xrightarrow{a} 5$$

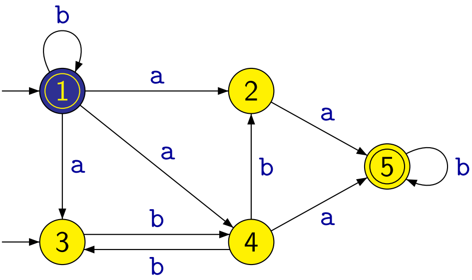
Nedeterministický konečný automat

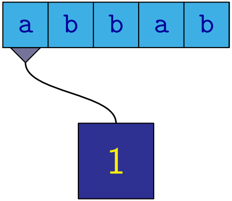




$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{b} 2 \xrightarrow{a} 5 \xrightarrow{b} 5$$

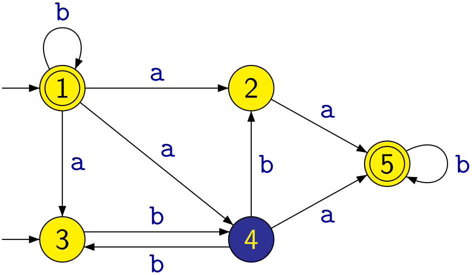
Nedeterministický konečný automat

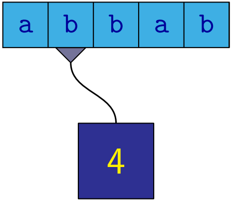




1

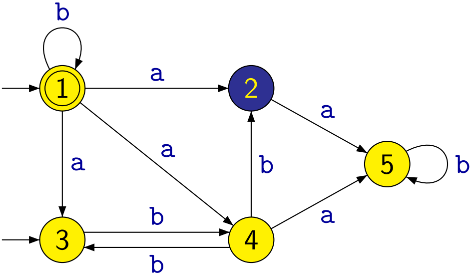
Nedeterministický konečný automat

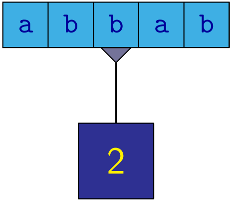




$$1 \xrightarrow{a} 4$$

Nedeterministický konečný automat

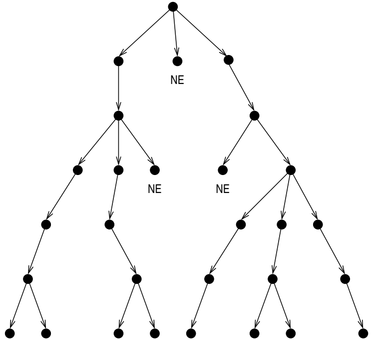




$$1 \xrightarrow{\mathtt{a}} 4 \xrightarrow{\mathtt{b}} 2$$

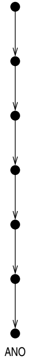
Nedeterministický konečný automat

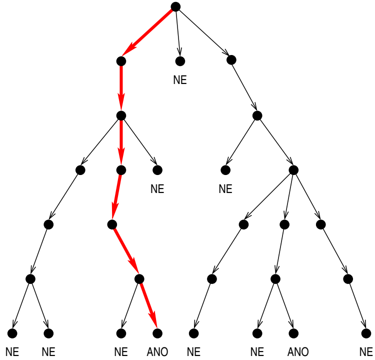
Nedeterministický konečný automat přijímá dané slovo, jestliže **existuje** alespoň jeden jeho výpočet, který vede k přijetí tohoto slova.



Nedeterministický konečný automat

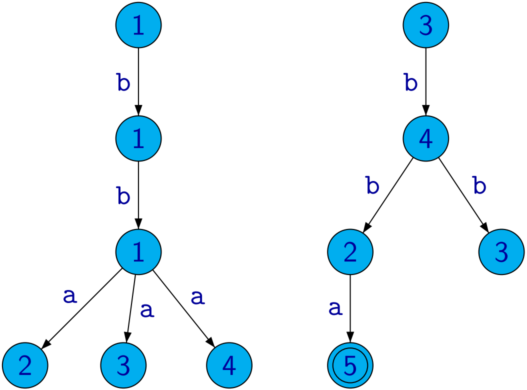
Nedeterministický konečný automat přijímá dané slovo, jestliže **existuje** alespoň jeden jeho výpočet, který vede k přijetí tohoto slova.





Nedeterministický konečný automat

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



Příklad: Les reprezentující všechny možné výpočty nad slovem bba.

Nedeterministický konečný automat

Formálně je **nedeterministický konečný automat** (**NKA**) definován jako pětice

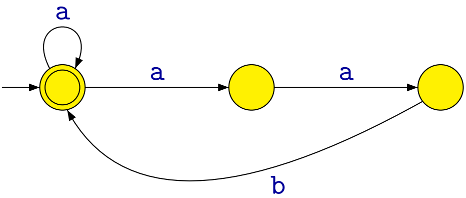
$$(Q, \Sigma, \delta, I, F)$$

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**

Příklady nedeterministických konečných automatů

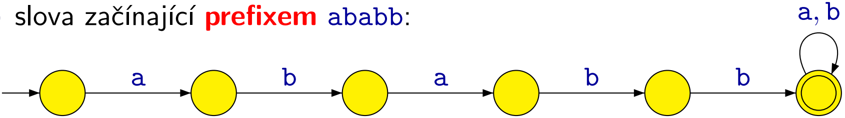
Příklad: Automat rozpoznávající jazyk nad abecedou $\{\mathtt{a}, \mathtt{b}\}$ tvořený slovy, kde každému výskytu symbolu $\mathtt{b}$ bezprostředně předchází dva symboly $\mathtt{a}$.



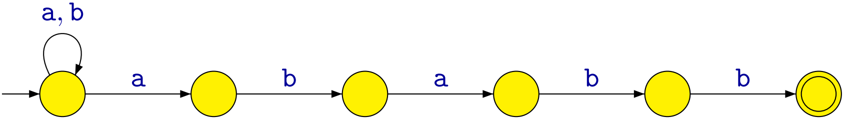
Příklady nedeterministických konečných automatů

Příklad: Automaty rozpoznávající jazyky nad abecedou $\{\mathtt{a}, \mathtt{b}\}$:

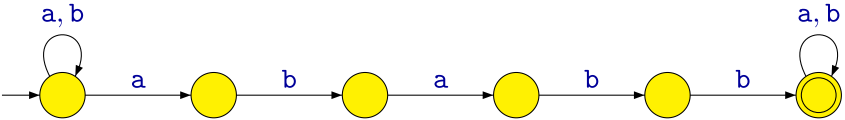
- slova začínající **prefixem** `ababb`:



- slova končící **sufixem** `ababb`:

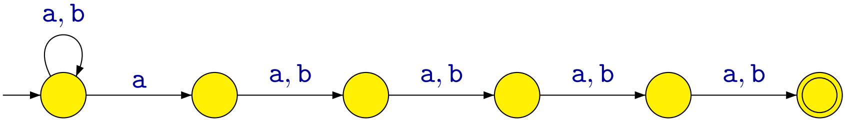


- slova obsahující **podslovo** `ababb`:

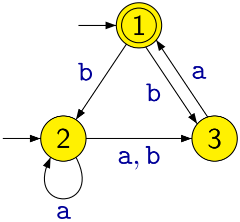


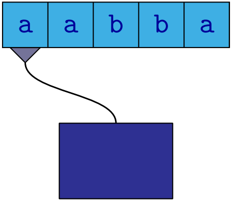
Příklady nedeterministických konečných automatů

Příklad: Automat rozpoznávající jazyk nad abecedou $\{a, b\}$ tvořený slovy, kde pátý symbol od konce je a.

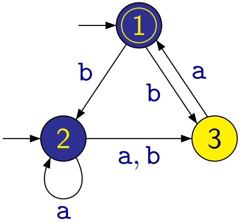


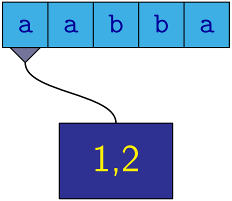
Převod NKA na DKA



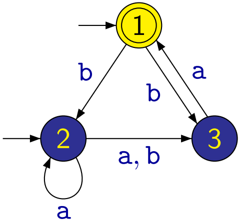


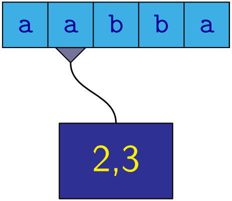
Převod NKA na DKA



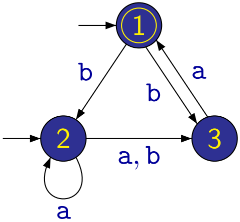


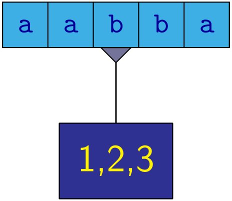
Převod NKA na DKA



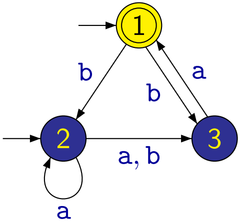


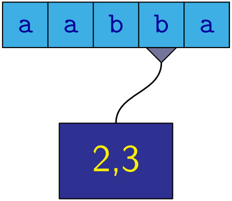
Převod NKA na DKA



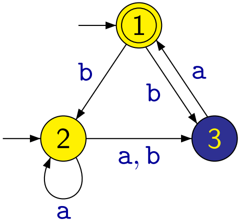


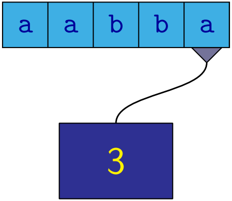
Převod NKA na DKA



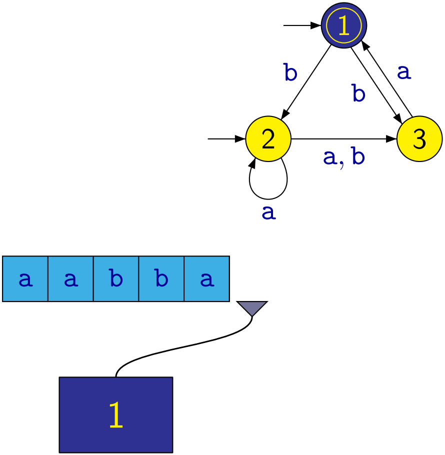


Převod NKA na DKA

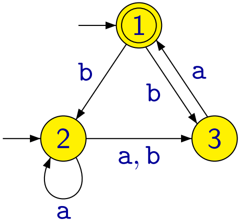




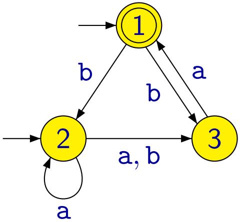
Převod NKA na DKA



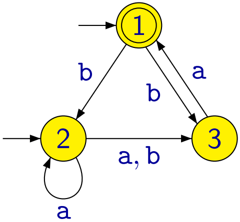
Převod NKA na DKA

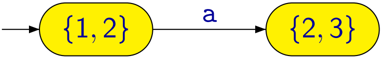


Převod NKA na DKA

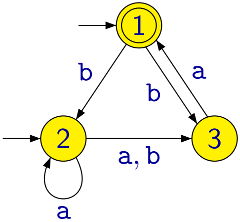


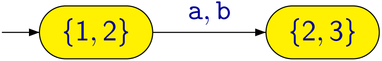
Převod NKA na DKA



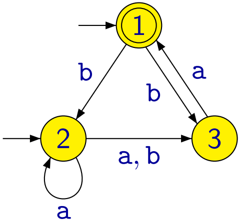


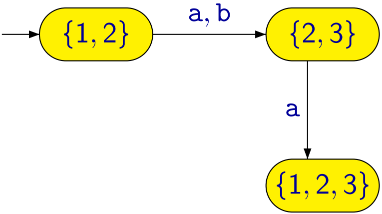
Převod NKA na DKA



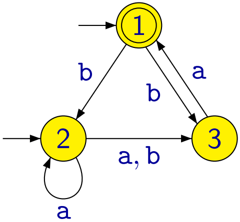


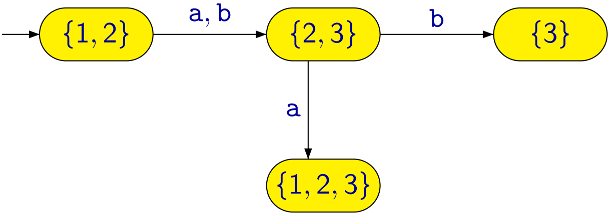
Převod NKA na DKA



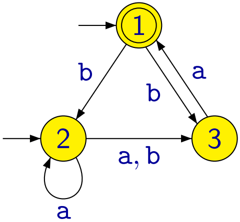


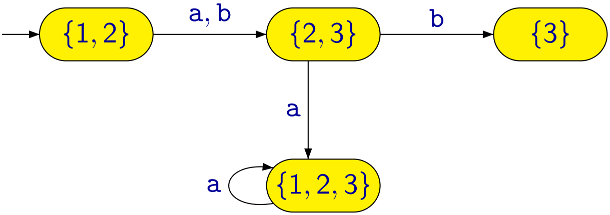
Převod NKA na DKA



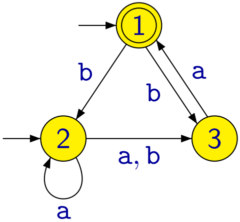


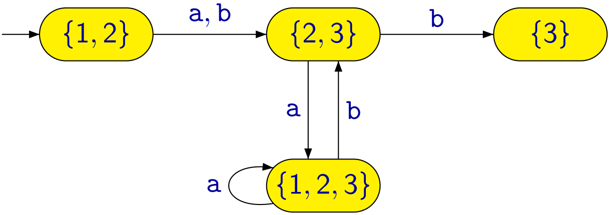
Převod NKA na DKA



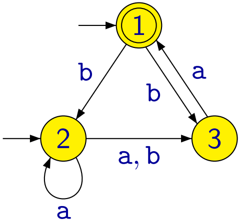


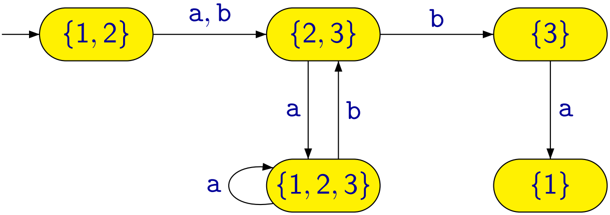
Převod NKA na DKA



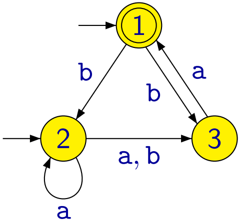


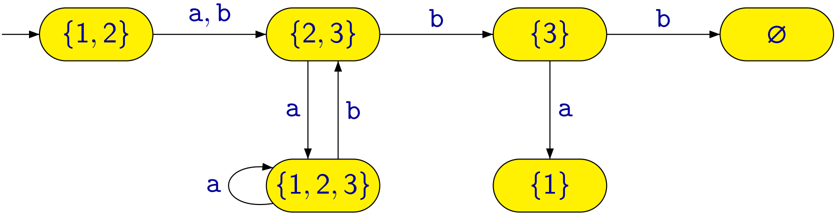
Převod NKA na DKA



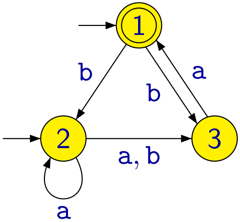


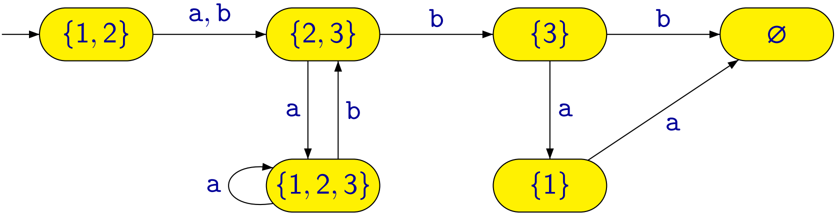
Převod NKA na DKA



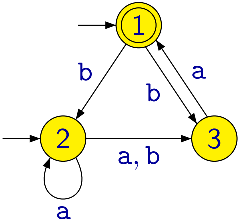


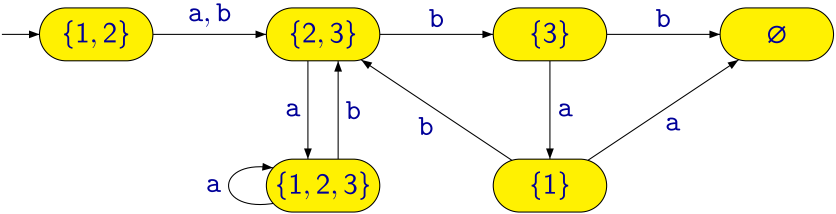
Převod NKA na DKA



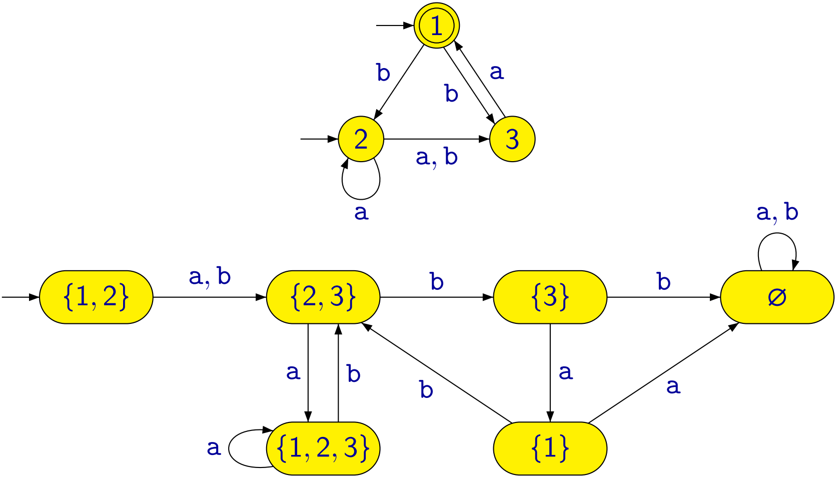


Převod NKA na DKA

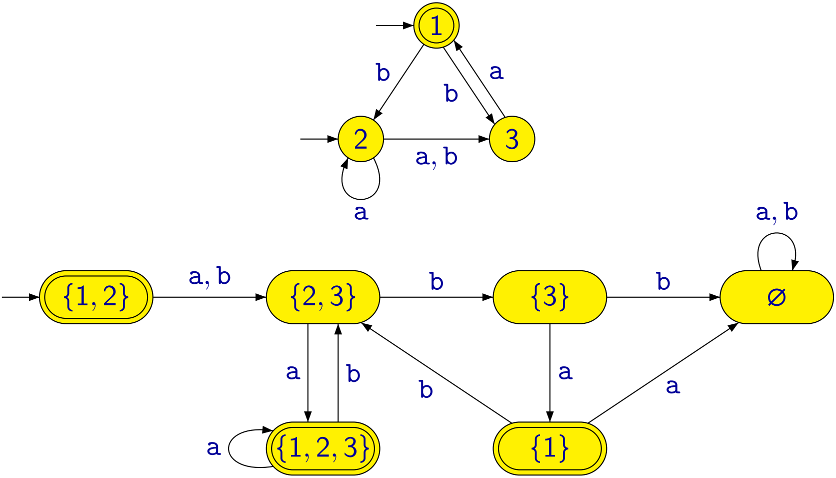




Převod NKA na DKA



Převod NKA na DKA



Převod NKA na DKA

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

Převod NKA na DKA

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

	a	b

Převod NKA na DKA

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

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

Převod NKA na DKA

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

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

Převod NKA na DKA

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

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

Převod NKA na DKA

	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 NKA na DKA

	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 NKA na DKA

	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 NKA na DKA

	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 NKA na DKA

	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 NKA na DKA

	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 NKA na DKA

	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 NKA na DKA

	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 NKA na DKA

	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 NKA na DKA

	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 NKA na DKA

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

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

Převod NKA na DKA

	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 NKA na DKA

	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 NKA na DKA

	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 NKA na DKA

	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

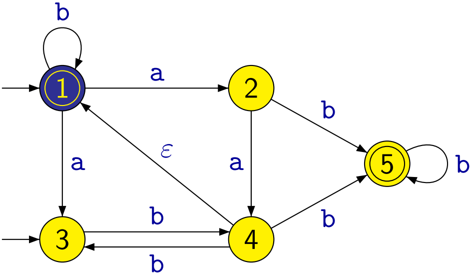
Převod NKA na DKA

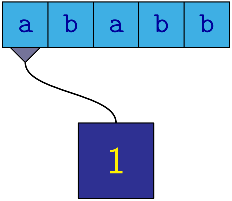
Poznámka: Při převodu nedeterministického automatu, který má n stavů, může mít výsledný deterministický automat až 2^n stavů.

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.

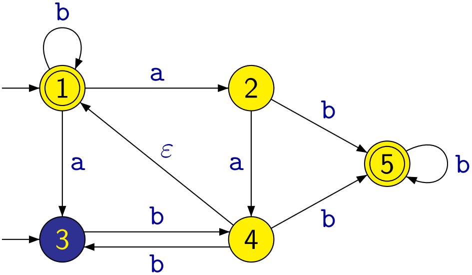
Zobecněný nedeterministický konečný automat

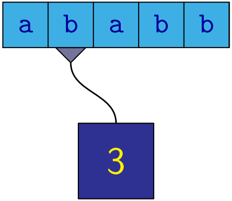




1

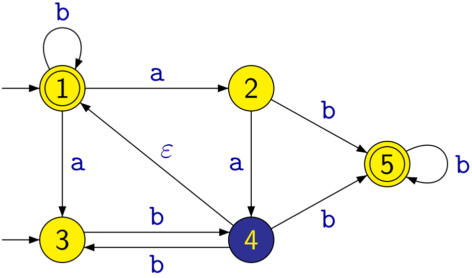
Zobecněný nedeterministický konečný automat

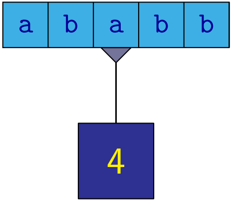




$$1 \xrightarrow{a} 3$$

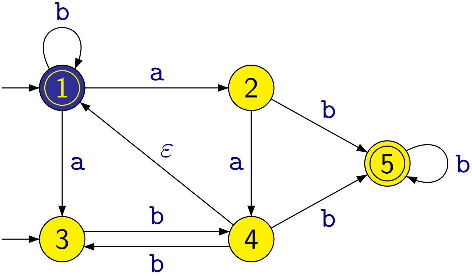
Zobecněný nedeterministický konečný automat

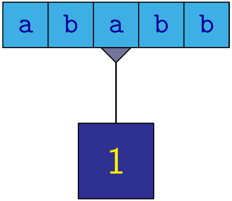




$$1 \xrightarrow{\texttt{a}} 3 \xrightarrow{\texttt{b}} 4$$

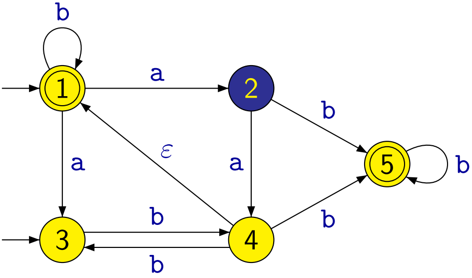
Zobecněný nedeterministický konečný automat

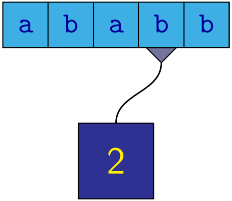




$$1 \xrightarrow{\mathtt{a}} 3 \xrightarrow{\mathtt{b}} 4 \xrightarrow{\varepsilon} 1$$

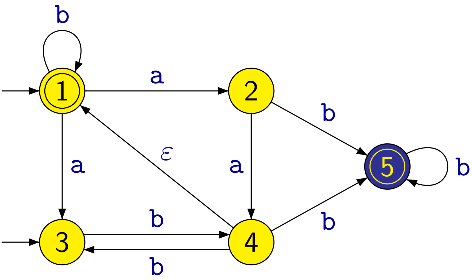
Zobecněný nedeterministický konečný automat

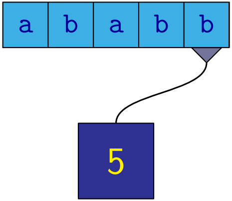




$$1 \xrightarrow{\texttt{a}} 3 \xrightarrow{\texttt{b}} 4 \xrightarrow{\varepsilon} 1 \xrightarrow{\texttt{a}} 2$$

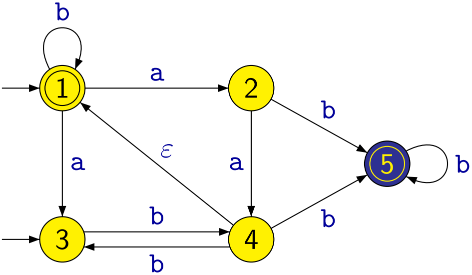
Zobecněný nedeterministický konečný automat

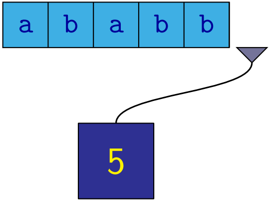




$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{\varepsilon} 1 \xrightarrow{a} 2 \xrightarrow{b} 5$$

Zobecněný nedeterministický konečný automat





$$1 \xrightarrow{a} 3 \xrightarrow{b} 4 \xrightarrow{\varepsilon} 1 \xrightarrow{a} 2 \xrightarrow{b} 5 \xrightarrow{b} 5$$

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** (**ZNKA**) definován jako pětice

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

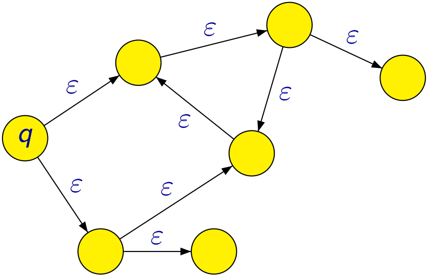
kde:

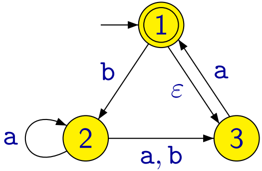
- Q je konečná množina **stavů**
- Σ je konečná **abeceda**
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ je **přechodová funkce**
- $I \subseteq Q$ je množina **počátečních stavů**
- $F \subseteq Q$ je množina **přijímajících stavů**

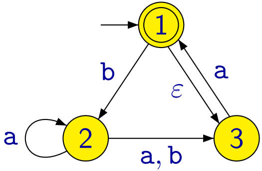
Poznámka: Na NKA můžeme nahlížet jako na speciální případ ZNKA, kde $\delta(q, \varepsilon) = \emptyset$ pro všechna $q \in Q$.

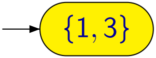
Převod na deterministický konečný automat

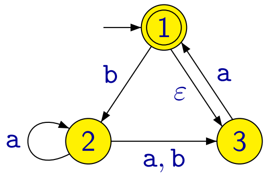
Zobecněný nedeterministický konečný automat je možné převést na deterministický podobnou konstrukcí jako nedeterministický konečný automat, s tím rozdílem, že do množin stavů musíme vždy přidat navíc i všechny stavy dosažitelné z již přidaných stavů nějakou sekvencí ε-přechodů.

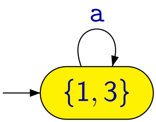


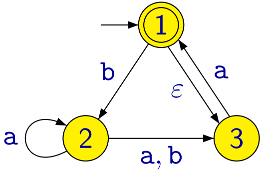

1
2
3
b
a
ε
a
a, b

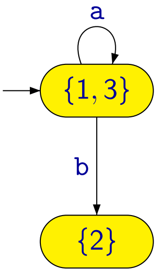

1
b
a
ε
a
2
3
a, b

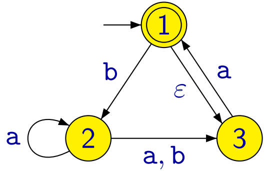

{1, 3}

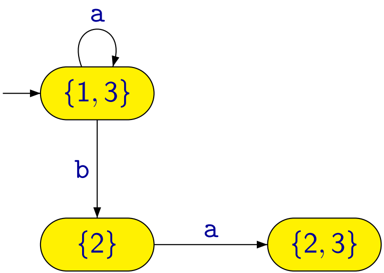

1
2
3
b
a
ε
a
a, b

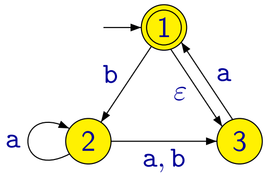

a
{1, 3}

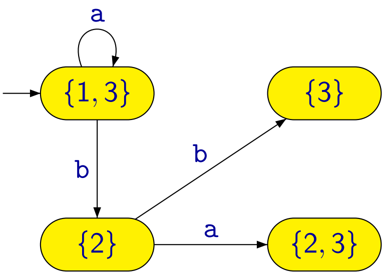

1
b
a
ε
a
2
3
a, b

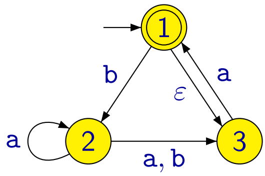

a
{1, 3}
b
{2}


1
b
a
ε
a
2
a, b
3

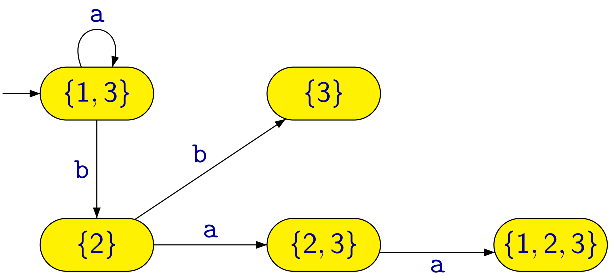

a
{1, 3}
b
{2}
a
{2, 3}


1
b
a
ε
a
2
3
a, b

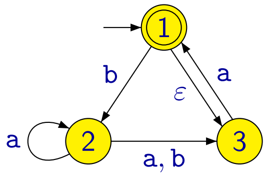

a
{1, 3}
{3}
b
b
{2}
a
{2, 3}

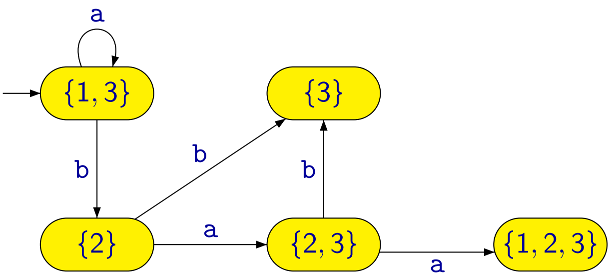


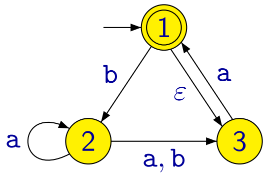
1
2
3
b
a
ε
a
a, b

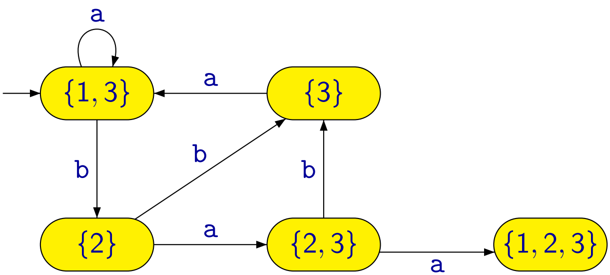


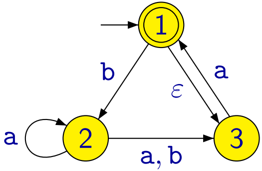
{1, 3}
{3}
{2}
{2, 3}
{1, 2, 3}
a
b
b
a
a

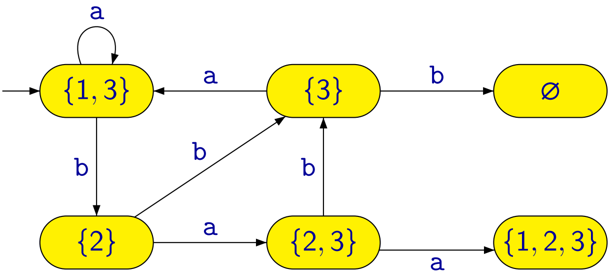

1
b
ε
a
a
2
3
a, b

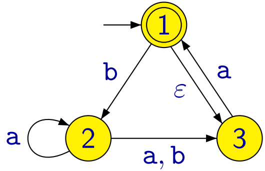

a
{1, 3}
{3}
b
b
b
{2}
a
{2, 3}
a
{1, 2, 3}

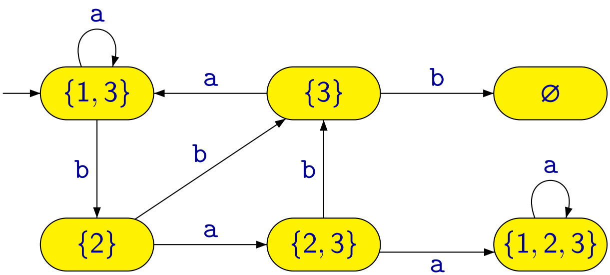

1
b
a
ε
a
2
3
a, b

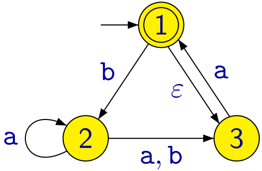

a
{1, 3}
a
{3}
b
b
b
{2}
a
{2, 3}
a
{1, 2, 3}

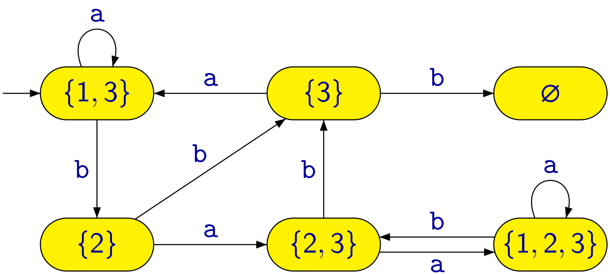

1
b
ε
a
a
2
3
a, b

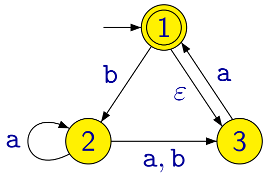

a
{1, 3}
a
{3}
b
∅
b
b
b
{2}
a
{2, 3}
a
{1, 2, 3}

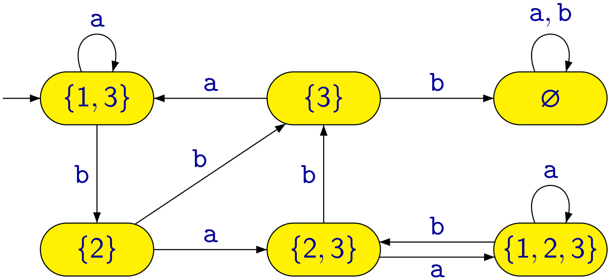

1
b
ε
a
a
2
3
a, b

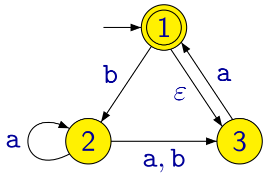

a
{1, 3}
a
{3}
b
∅
b
b
b
a
{2}
a
{2, 3}
a
{1, 2, 3}

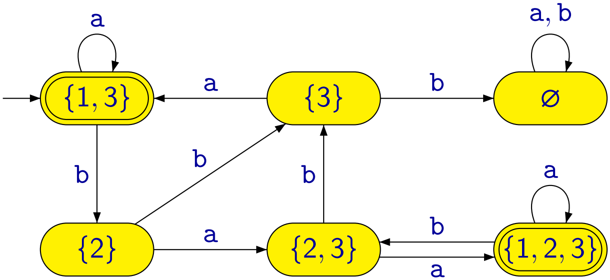

1
2
3
a
b
a
ε
a, b


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


1
b
ε
a
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
ε
a
a
2
a, b
3


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

Převod ZNKA na DKA

Předtím, než formálně popíšeme převod ZNKA na DKA, zaveďme si několik pomocných definic.

Předpokládejme nějaký daný ZNKA $\mathcal{A} = (Q, \Sigma, \delta, I, F)$.

Definujme funkci $\hat{\delta} : \mathcal{P}(Q) \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ tak, že pro $K \subseteq Q$ a $a \in \Sigma \cup \{\varepsilon\}$ je

$$\hat{\delta}(K, a) = \bigcup_{q \in K} \delta(q, a)$$

Převod ZNKA na DKA

Pro $K \subseteq Q$ označme $Cl_\varepsilon(K)$ množinu všech stavů dosažitelných ze stavů z množiny K nějakou libovolnou sekvencí ε-přechodů.

To znamená, že funkce $Cl_\varepsilon : \mathcal{P}(Q) \to \mathcal{P}(Q)$ je definována tak, že pro $K \subseteq Q$ je $Cl_\varepsilon(K)$ nejmenší (vzledem k inkluzi) množina splňující následující dvě podmínky:

- $K \subseteq Cl_\varepsilon(K)$
- Pro každé $q \in Cl_\varepsilon(K)$ platí, že $\delta(q, \varepsilon) \subseteq Cl_\varepsilon(K)$.

Poznámka: Všimněme si, že pro libovolné K je $Cl_\varepsilon(Cl_\varepsilon(K)) = Cl_\varepsilon(K)$.

Všimněme si také, že v případě NKA (kde $\delta(q, \varepsilon) = \emptyset$ pro každé $q \in Q$) je $Cl_\varepsilon(K) = K$.

Převod ZNKA na DKA

K danému ZNKA $\mathcal{A} = (Q, \Sigma, \delta, I, F)$ nyní můžeme sestrojit DKA $\mathcal{A}' = (Q', \Sigma, \delta', q_0', F')$, kde:

- $Q' = \mathcal{P}(Q)$ $\qquad$ ($K \in Q'$ tedy znamená, že $K \subseteq Q$)
- $\delta' : Q' \times \Sigma \to Q'$ je definová tak, že pro $K \in Q'$ a $a \in \Sigma$ je

$$\delta'(K, a) = Cl_\varepsilon(\hat{\delta}(Cl_\varepsilon(K), a))$$

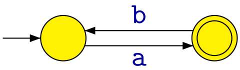
- $q_0' = Cl_\varepsilon(I)$
- $F' = \{K \in Q' \mid Cl_\varepsilon(K) \cap F \neq \emptyset\}$

Není težké ověřit, že $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.

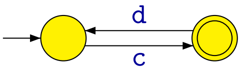
Zřetězení jazyků

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

$\mathcal{A}_1$:



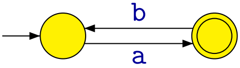
$\mathcal{A}_2$:



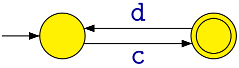
Zřetězení jazyků

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

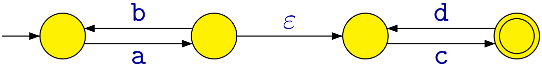
$\mathcal{A}_1$:



$\mathcal{A}_2$:



$\mathcal{A}$:

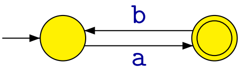


$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cdot \mathcal{L}(\mathcal{A}_2)$

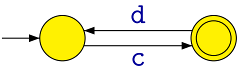
Zřetězení jazyků

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

$\mathcal{A}_1$:

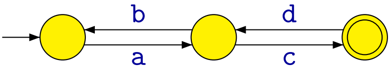


$\mathcal{A}_2$:



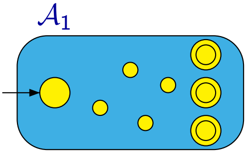
Chybná konstrukce:

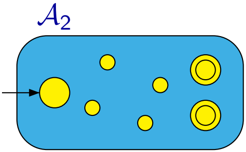
$\mathcal{A}$:



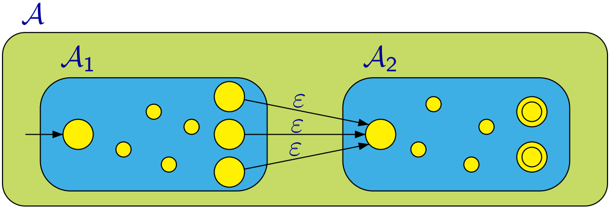
$\mathtt{acdbac} \in \mathcal{L}(\mathcal{A})$, ale $\mathtt{acdbac} \notin \mathcal{L}(\mathcal{A}_1) \cdot \mathcal{L}(\mathcal{A}_2)$

Zřetězení jazyků

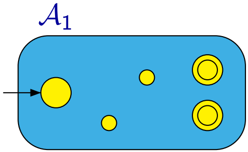




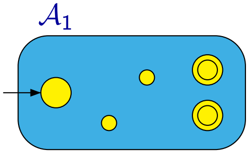
Zřetězení jazyků

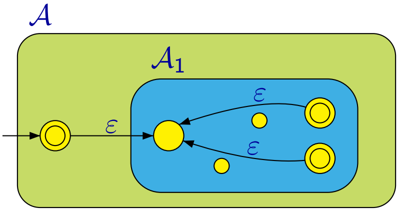


Iterace jazyka



Iterace jazyka

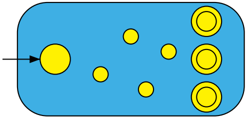




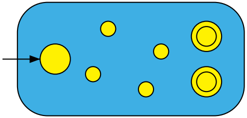
Sjednocení jazyků

Alternativní konstrukce pro sjednocení jazyků:

$\mathcal{A}_1$

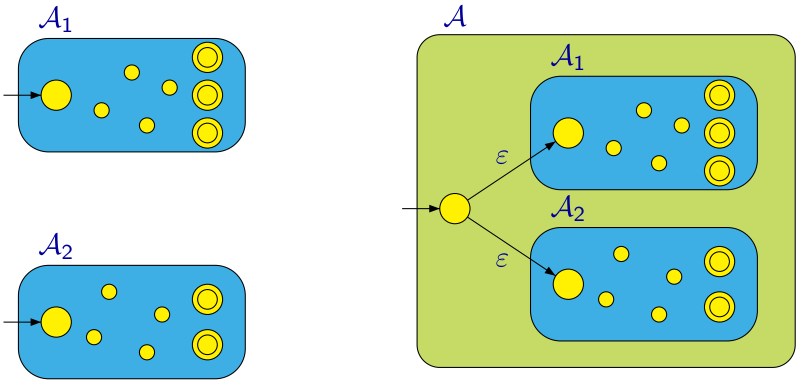


$\mathcal{A}_2$



Sjednocení jazyků

Alternativní konstrukce pro sjednocení jazyků:



Uzavřenost třídy regulárních jazyků

Množina (všech) regulárních jazyků je uzavřená vůči operacím:

- sjednocení
- průnik
- doplněk
- zřetězení
- iterace
- ...

Převod regulárního výrazu na konečný automat

Tvrzení

Každý jazyk, který je možné vyjádřit regulárním výrazem, je regulární (tj. rozpoznávaný nějakým konečným automatem).

Důkaz: Stačí ukázat, jak k danému regulárnímu výrazu α zkonstruovat konečný automat, který rozpoznává jazyk $\mathcal{L}(\alpha)$.

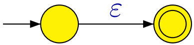
Konstrukce je rekurzivní a postupuje podle struktury výrazu α:

- Pokud je α elementární výraz (tj. $\emptyset$, ε nebo a):
  - Sestrojíme přímo odpovídající automat.
- Pokud je α tvaru $(\beta + \gamma)$, $(\beta \cdot \gamma)$ nebo (β^*):
  - Rekurzivně sestrojíme automaty rozpoznávající jazyky $\mathcal{L}(\beta)$ a $\mathcal{L}(\gamma)$.
  - Z nich sestrojíme automat rozpoznávající jazyk $\mathcal{L}(\alpha)$.

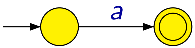
Převod regulárního výrazu na konečný automat

Automaty pro elementární výrazy:

$\emptyset$



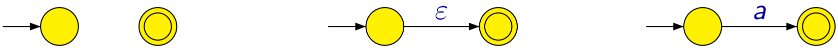
ε



a

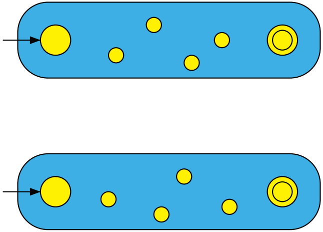
Převod regulárního výrazu na konečný automat

Automaty pro elementární výrazy:



$\emptyset$ ε a

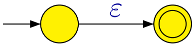
Konstrukce pro sjednocení:



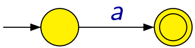
Převod regulárního výrazu na konečný automat

Automaty pro elementární výrazy:

$\emptyset$

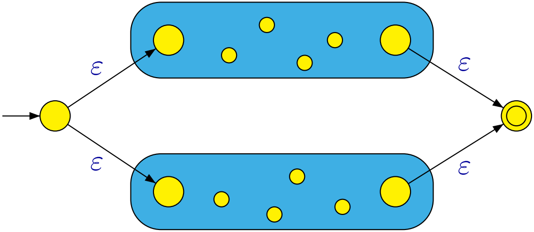


ε



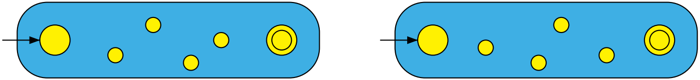
a

Konstrukce pro sjednocení:



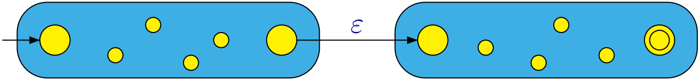
Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:



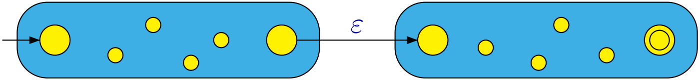
Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:

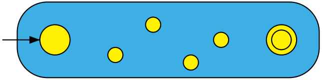


Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:

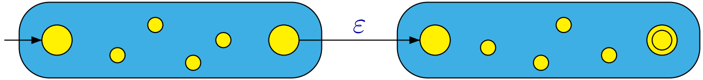


Konstrukce pro iteraci:

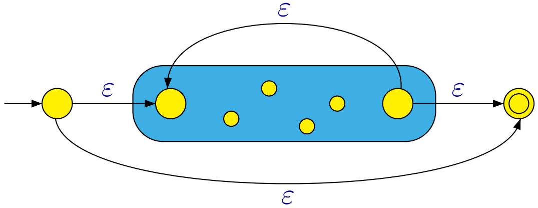


Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:



Konstrukce pro iteraci:

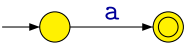


Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((\mathtt{a}+\mathtt{b})\cdot\mathtt{b})^*$:

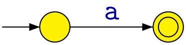
Převod regulárního výrazu na konečný automat

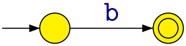
Příklad: Konstrukce automatu pro výraz $((\mathtt{a} + \mathtt{b}) \cdot \mathtt{b})^*$:



Převod regulárního výrazu na konečný automat

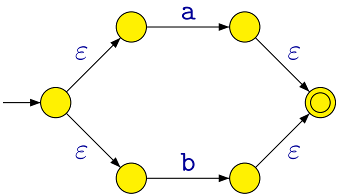
Příklad: Konstrukce automatu pro výraz $((\mathtt{a} + \mathtt{b}) \cdot \mathtt{b})^*$:





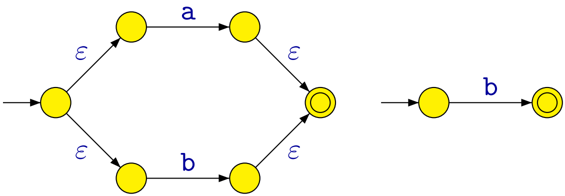
Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((\mathtt{a} + \mathtt{b}) \cdot \mathtt{b})^*$:



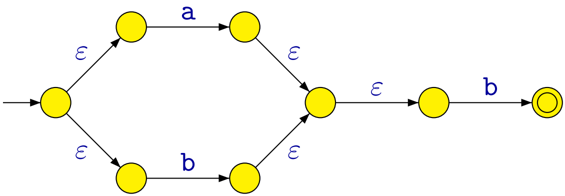
Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((\mathtt{a}+\mathtt{b})\cdot\mathtt{b})^*$:



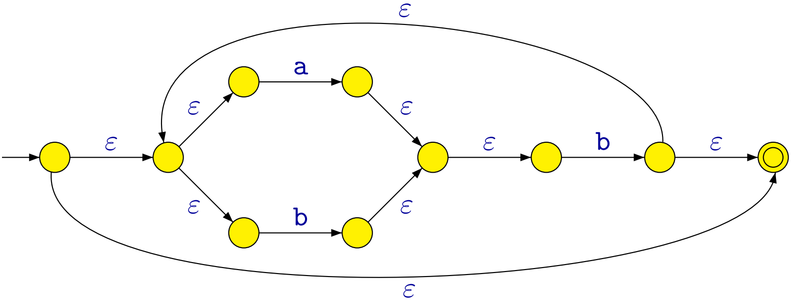
Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((\mathtt{a}+\mathtt{b})\cdot\mathtt{b})^*$:



Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((\mathtt{a}+\mathtt{b})\cdot\mathtt{b})^*$:



Převod regulárního výrazu na konečný automat

Pokud se výraz α skládá z n znaků (nepočítáme-li závorky), má výsledný automat:

- nejvýše $2n$ stavů,
- nejvýše $4n$ přechodů.

Poznámka: Převodem ze zobecněného nedeterministického automatu na deterministický však může počet stavů vzrůst exponenciálně, tj. výsledný automat pak může mít až $2^{2n} = 4^n$ stavů.

Převod konečného automatu na regulární výraz

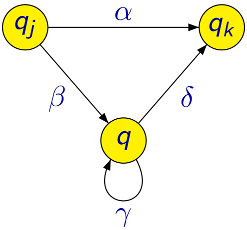
Tvrzení

Každý regulární jazyk je možné popsat nějakým regulárním výrazem.

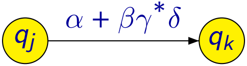
Důkaz: Stačí ukázat, jak pro libovolný konečný automat $\mathcal{A}$ zkonstruovat regulární výraz α takový, že $\mathcal{L}(\alpha) = \mathcal{L}(\mathcal{A})$.

- $\mathcal{A}$ upravíme tak, aby měl právě jeden počáteční a právě jeden přijímající stav.
- Budeme postupně odebírat jednotlivé stavy.
- Přechody budou označeny regulárními výrazy.
- Zbude automat se dvěma stavy – počátečním a koncovým, a jedním přechodem ohodnoceným výsledným regulárním výrazem.

Hlavní myšlenka: Při odstraňování stavu q nahradit pro každou dvojici zbylých stavů q_j, q_k cestu z q_j do q_k vedoucí přes q.

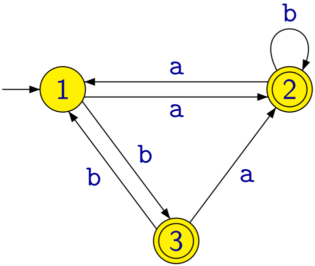


Po odstranění stavu q:



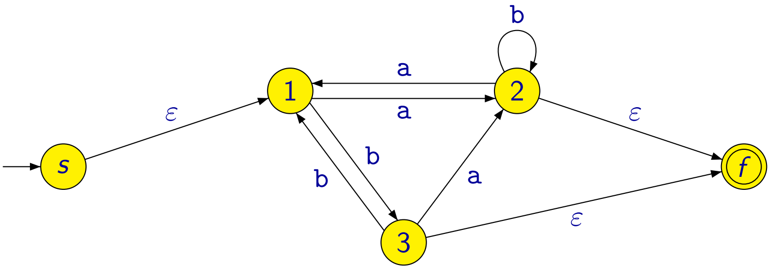
Převod konečného automatu na regulární výraz

Příklad:



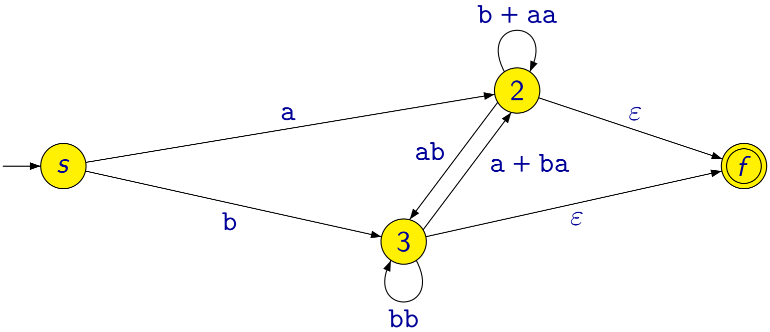
Převod konečného automatu na regulární výraz

Příklad:



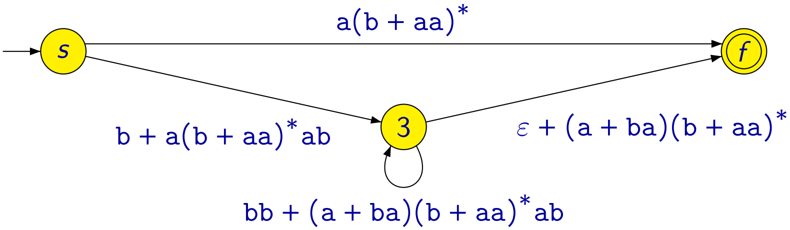
Převod konečného automatu na regulární výraz

Příklad:



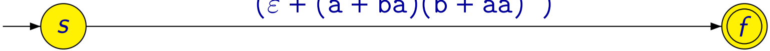
Převod konečného automatu na regulární výraz

Příklad:



Příklad:

$$\mathtt{a(b+aa)^*+}$$
$$\mathtt{(b+a(b+aa)^*ab)}$$
$$\mathtt{(bb+(a+ba)(b+aa)^*ab)^*}$$
$$\mathtt{(\varepsilon+(a+ba)(b+aa)^*)}$$



Ekvivalence konečných automatů a regulárních výrazů

Věta

Jazyk je regulární právě tehdy, když je ho možné popsat regulárním výrazem.

Neregulární jazyky

Ne všechny jazyky jsou regulární.

Existují jazyky, pro které neexistuje žádný konečný automat, který by je rozpoznával.

Příklady neregulárních jazyků:

- $L_1 = \{a^n b^n \mid n \geq 0\}$
- $L_2 = \{ww \mid w \in \{a, b\}^*\}$
- $L_3 = \{ww^R \mid w \in \{a, b\}^*\}$

Poznámka: Existence neregulárních jazyků vyplývá již z faktu, že automatů pracujících nad nějakou abecedou Σ je jen spočetně mnoho, zatímco jazyků nad abecedou Σ je nespočetně mnoho.

Neregulární jazyky

Jak dokázat o nějakém jazyce L, že není regulární?

Jazyk není regulární, jestliže neexistuje (tj. není možné sestrojit) konečný automat, který by ho rozpoznával.

Jak ale dokázat, že něco neexistuje?

Neregulární jazyky

Jak dokázat o nějakém jazyce L, že není regulární?

Jazyk není regulární, jestliže neexistuje (tj. není možné sestrojit) konečný automat, který by ho rozpoznával.

Jak ale dokázat, že něco neexistuje?

Odpověď: Sporem.

Např. předpokládat, že existuje nějaký automat $\mathcal{A}$ rozpoznávající jazyk L, a ukázat, že tento předpoklad vede k logickému sporu.

Neregulární jazyky

Ukážeme, že jazyk $L = \{a^n b^n \mid n \geq 0\}$ není regulární.

Důkaz sporem.

Předpokládejme, že existuje DKA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ takový, že $\mathcal{L}(\mathcal{A}) = L$.

Neregulární jazyky

Ukážeme, že jazyk $L = \{a^n b^n \mid n \geq 0\}$ není regulární.

Důkaz sporem.

Předpokládejme, že existuje DKA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ takový, že $\mathcal{L}(\mathcal{A}) = L$.

Řekněme, že $|Q| = n$.

Ukážeme, že jazyk $L = \{a^n b^n \mid n \geq 0\}$ není regulární.

Důkaz sporem.

Předpokládejme, že existuje DKA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ takový, že $\mathcal{L}(\mathcal{A}) = L$.

Řekněme, že $|Q| = n$.

Vezměme si slovo $z = a^n b^n$.

Neregulární jazyky

Ukážeme, že jazyk $L = \{a^n b^n \mid n \geq 0\}$ není regulární.

Důkaz sporem.

Předpokládejme, že existuje DKA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ takový, že $\mathcal{L}(\mathcal{A}) = L$.

Řekněme, že $|Q| = n$.

Vezměme si slovo $z = a^n b^n$.

Protože $z \in L$, musí existovat přijímající výpočet automatu $\mathcal{A}$

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} \cdots \xrightarrow{a} q_{n-1} \xrightarrow{a} q_n \xrightarrow{b} q_{n+1} \xrightarrow{b} \cdots \xrightarrow{b} q_{2n-1} \xrightarrow{b} q_{2n}$$

kde q_0 je počáteční stav a $q_{2n} \in F$.

Neregulární jazyky

Vezměme si nyní prvních $n+1$ stavů ve výpočtu

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} \cdots \xrightarrow{a} q_{n-1} \xrightarrow{a} q_n \xrightarrow{b} q_{n+1} \xrightarrow{b} \cdots \xrightarrow{b} q_{2n-1} \xrightarrow{b} q_{2n}$$

tj. posloupnost stavů $q_0, q_1, \ldots, q_n$.

Je zřejmé, že všechny stavy v této posloupnosti nemohou být navzájem různé, protože $|Q| = n$ a tato posloupnost má $n+1$ prvků.

To znamená, že existuje nějaký stav $q \in Q$, který se v této posloupnosti vyskytuje (alespoň) dvakrát.

Neregulární jazyky

Vezměme si nyní prvních $n+1$ stavů ve výpočtu

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} \cdots \xrightarrow{a} q_{n-1} \xrightarrow{a} q_n \xrightarrow{b} q_{n+1} \xrightarrow{b} \cdots \xrightarrow{b} q_{2n-1} \xrightarrow{b} q_{2n}$$

tj. posloupnost stavů $q_0, q_1, \ldots, q_n$.

Je zřejmé, že všechny stavy v této posloupnosti nemohou být navzájem různé, protože $|Q| = n$ a tato posloupnost má $n+1$ prvků.

To znamená, že existuje nějaký stav $q \in Q$, který se v této posloupnosti vyskytuje (alespoň) dvakrát.

Jde o aplikaci tzv. **holubníkového principu (pigeonhole principle)**.

Holubníkový princip

Jestliže mám $n+1$ holubů rozmístěných do n klecí, pak jsou alespoň v jedné kleci minimálně dva holubi.

Neregulární jazyky

Vezměme si nyní prvních $n+1$ stavů ve výpočtu

$$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_2 \xrightarrow{a} \cdots \xrightarrow{a} q_{n-1} \xrightarrow{a} q_n \xrightarrow{b} q_{n+1} \xrightarrow{b} \cdots \xrightarrow{b} q_{2n-1} \xrightarrow{b} q_{2n}$$

tj. posloupnost stavů $q_0, q_1, \ldots, q_n$.

Je zřejmé, že všechny stavy v této posloupnosti nemohou být navzájem různé, protože $|Q| = n$ a tato posloupnost má $n+1$ prvků.

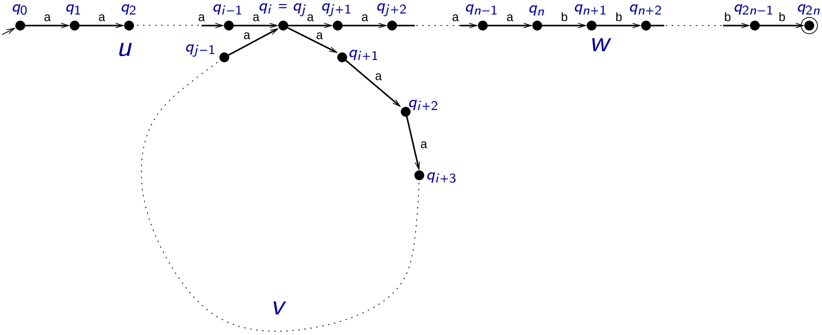
To znamená, že existuje nějaký stav $q \in Q$, který se v této posloupnosti vyskytuje (alespoň) dvakrát.

Tj. existují indexy i, j takové, že $0 \le i < j \le n$ a

$$q_i = q_j$$

což znamená, že automat $\mathcal{A}$ při čtení symbolů a ve slově $z = a^n b^n$ projde cyklem.

Neregulární jazyky



Slovo $z = a^n b^n$ můžeme rozdělit na tři části u, v, w takové, že $z = uvw$:

$$u = a^i \qquad v = a^{j-i} \qquad w = a^{n-j} b^n$$

Neregulární jazyky

Pro slova $u = a^i$, $v = a^{j-i}$ a $w = a^{n-j}b^n$ platí

$$q_0 \xrightarrow{u} q_i \qquad q_i \xrightarrow{v} q_j \qquad q_j \xrightarrow{w} q_{2n}$$

Označme r délku slova v, tj. $r = j - i$ (zjevně $r > 0$, protože $i < j$).

Protože $q_i = q_j$, tak automat přijme slovo $uw = a^{n-r}b^n$, které nepatří do jazyka L:

$$q_0 \xrightarrow{u} q_i \xrightarrow{w} q_{2n}$$

Rovněž slovo $uvvw = a^{n+r}b^n$, které také nepatří do L, bude přijato:

$$q_0 \xrightarrow{u} q_i \xrightarrow{v} q_i \xrightarrow{v} q_i \xrightarrow{w} q_{2n}$$

Neregulární jazyky

Podobně můžeme zdůvodnit, že každé slovo tvaru $uvvvv \cdots vvw$, tj. tvaru $uv^k w$ pro nějaké $k \geq 0$, bude automatem $\mathcal{A}$ přijato:

$$q_0 \xrightarrow{u} q_i \xrightarrow{v} q_i \xrightarrow{v} q_i \xrightarrow{v} \cdots \xrightarrow{v} q_i \xrightarrow{v} q_i \xrightarrow{w} q_{2n}$$

Slovo tvaru $uv^k w$ vypadá následovně: $a^{n-r+rk} b^n$.

Protože $r > 0$, tak následující rovnost platí jen pro $k = 1$:

$$n - r + rk = n$$

Pokud je tedy $k \neq 1$, tak slovo $uv^k w$ nepatří do jazyka L.

Automat $\mathcal{A}$ však každé takové slovo přijme, což je spor s předpokladem, že $\mathcal{L}(\mathcal{A}) = \{a^n b^n \mid n \geq 0\}$.