Přehled přednášky

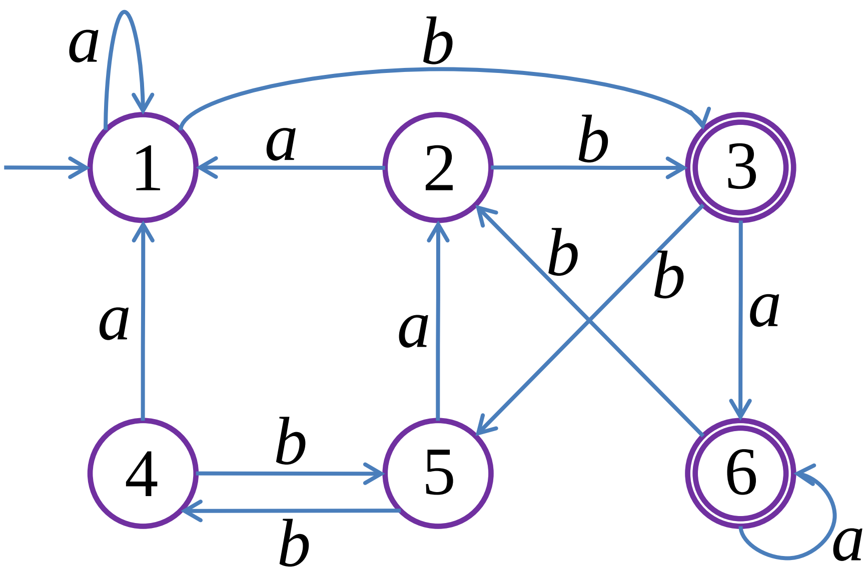
- Redukce stavů konečného automatu
- Nedeterministický konečný automat
- Zobecněný nedeterministický konečný automat
- Regulární výrazy, ekvivalence s konečnými automaty

Normovaný (redukovaný) KA

Převod konečného automatu na ekvivalentní automat
s minimálním počtem stavů.

1. Odstraníme nedosažitelné stavy
2. Sloučíme ekvivalentní stavy

Redukce konečného automatu



	a	b	
1	1	3	
2	1	3	
4	1	5	*I*
5	2	4	
3	6	5	*II*
6	6	2	

Redukce konečného automatu

	a	b	
1	I	II	I
2	I	II	
4	I	I	II
5	I	I	
3	II	I	III
6	II	I	

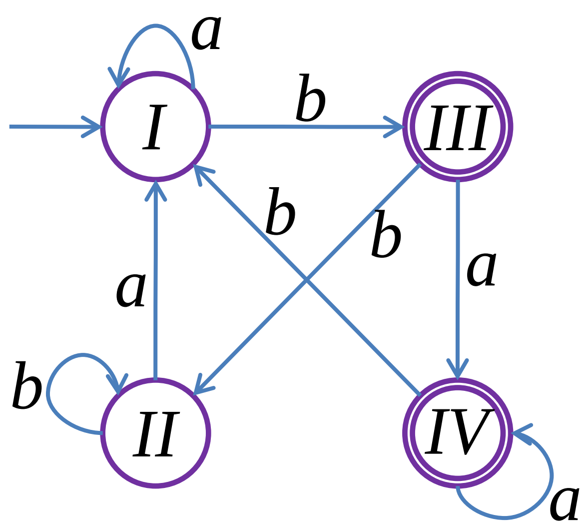
	a	b	
1	I	III	I
2	I	III	
4	I	II	II
5	I	II	
3	III	II	III
6	III	I	IV

$$I = \{1,2\}$$

$$II = \{4,5\}$$

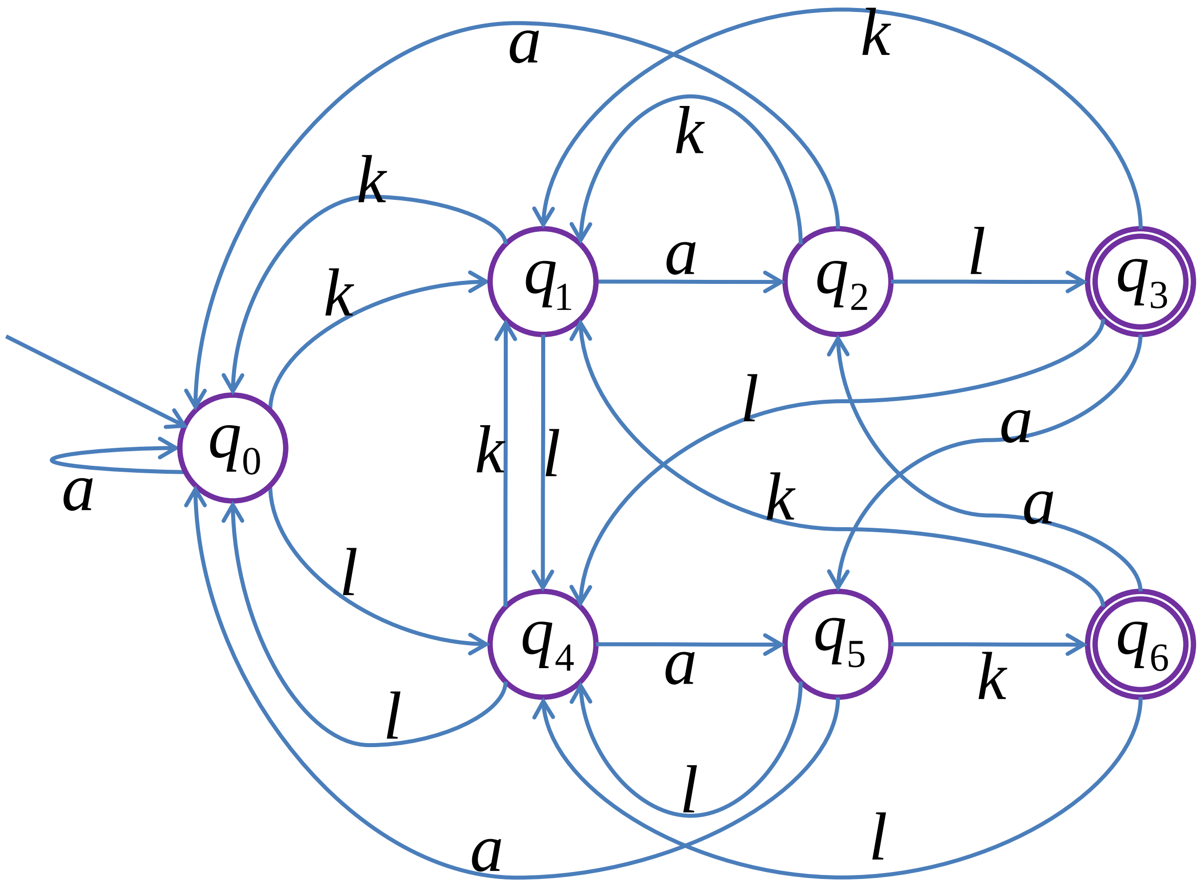
$$III = \{3\}$$

$$IV = \{6\}$$



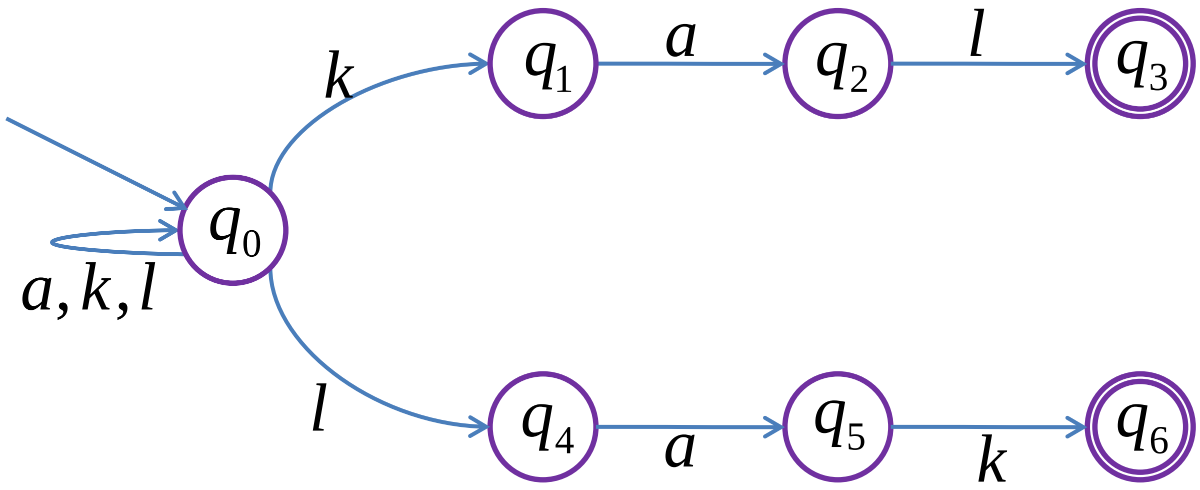
Konečný automat – příklad z minula

Komplikované, nepřehledné.



Nedeterministický konečný automat

Povolme pro každou konfiguraci automatu více možností, do kterých stavů přejít.



Nedeterministický konečný automat

Nedeterministický konečný automat (NKA) je pětice

$A = (Q, \Sigma, \delta, I, F)$, kde

- Q je konečná množina stavů
- Σ je abeceda
- $\delta : Q \times \Sigma \to P(Q)$ je přechodová funkce
- $I \subseteq Q$ je množina počátečních (iniciálních) stavů
- $F \subseteq Q$ je množina přijímacích (koncových) stavů

Rozpoznávaný jazyk

- Výpočet je opět posloupnost konfigurací $K_0, \ldots, K_n$, kde K_{i+1} bezprostředně následuje po K_i .

- Výpočet je pro slovo w přijímající, pokud $K_0 = (q_0, w)$ a $K_n = (q, \varepsilon)$, kde $q \in F$.

- NKA A přijímá vstupní slovo w, pokud pro w existuje alespoň jeden přijímající výpočet.

$$L(A) = \{w \mid A \text{ přijímá } w\}$$

Simulace NKA

- Mějme NKA, který má *n* stavů.

- Simulaci tohoto NKA můžeme provést na počítači s použitím *n* bitů (*i*-tý bit reprezentuje dosažitelnost *i*-tého stavu v daném okamžiku).

Převod na deterministický automat

Nechť $A = (Q, \Sigma, \delta, I, F)$ je nedeterministický konečný automat.
Sestrojíme (deterministický) automat $A_d = (Q_d, \Sigma, \delta_d, q_0, F_d)$.

$Q_d = P(Q)$

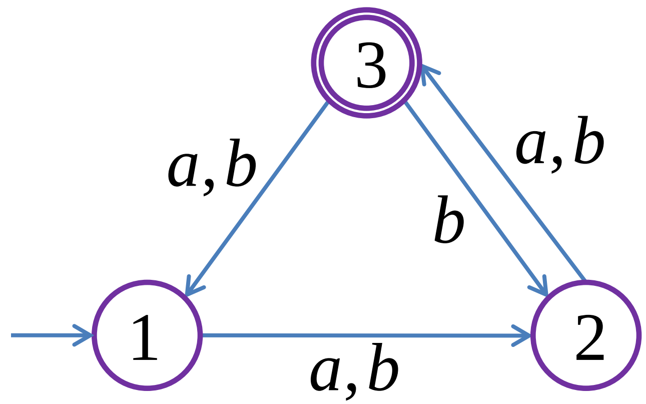
$q_0 = I$

$F_d \subseteq P(Q)$ .. všechny podmnožiny, které obsahují alespoň jeden přijímací stav

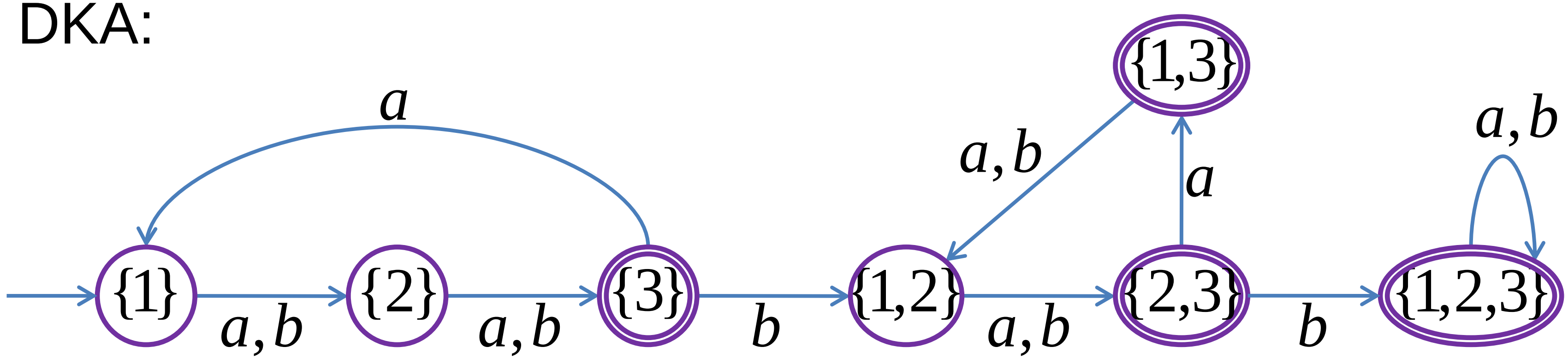
$\delta_d : P(Q) \times \Sigma \to P(Q)$

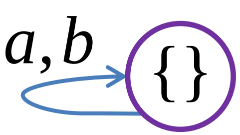
Příklad převodu NKA na DKA

NKA:



DKA:





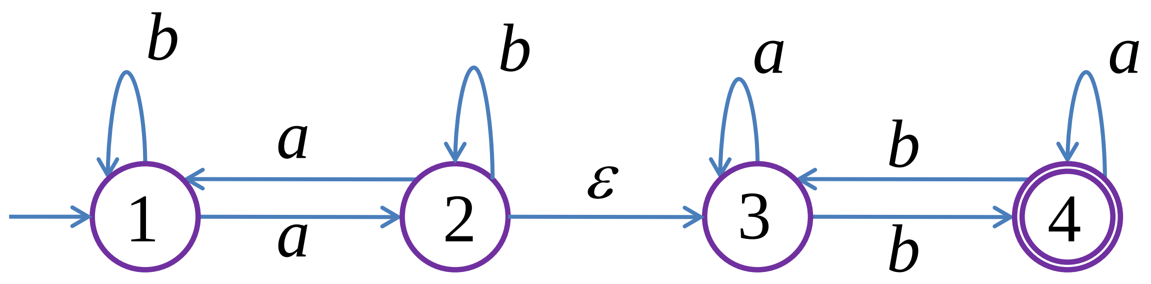
nedosažitelný stav (můžeme jej vynechat)

Zobecněný NKA

- modifikovaný tvar přechodové funkce:

$$\delta : Q \times \left(\Sigma \cup \{\varepsilon\}\right) \to P(Q)$$

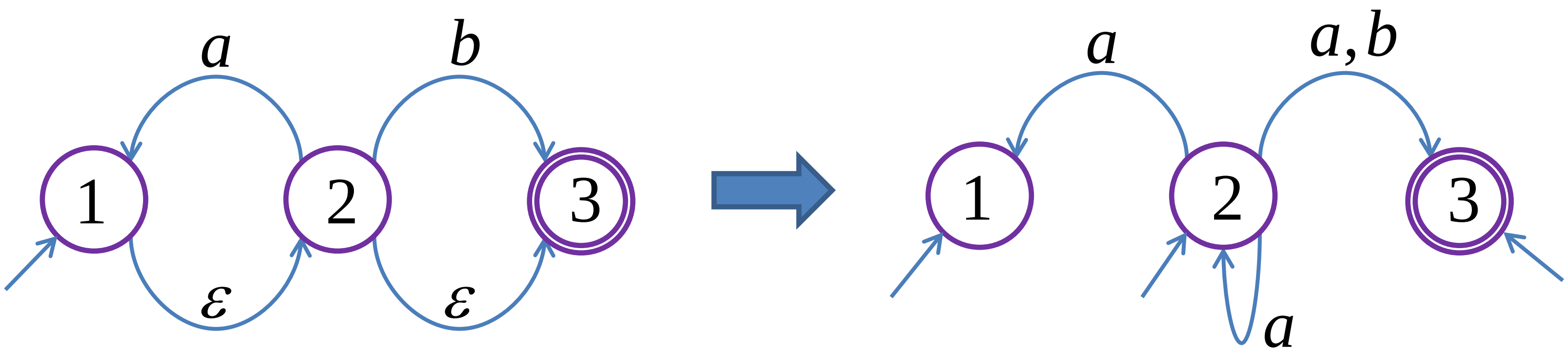
Zobecněný nedeterministický konečný automat (ZNKA) může provést změnu stavu, aniž by přečetl znak vstupu, jedná se o tzv. ε-přechody.



Převod ZNKA na NKA

Nechť $A=(Q,\Sigma,\delta,I,F)$ je ZNKA, sestrojíme ekvivalentní NKA $A'=(Q,\Sigma,\delta',I',F)$.

- $I' := I \cup$ {stavy dosažitelné z I přes ε-přechody}
- $\delta'(q,a) := \delta(q,a) \cup$ {stavy dosažitelné z $\delta(q,a)$ přes ε-přechody}



Regulární operace s jazyky

sjednocení:

$$L_1 \cup L_2 = \{w \mid w \in L_1 \vee w \in L_2\}$$

zřetězení:

$$L_1 \cdot L_2 = \{uv \mid u \in L_1, v \in L_2\}$$

iterace:

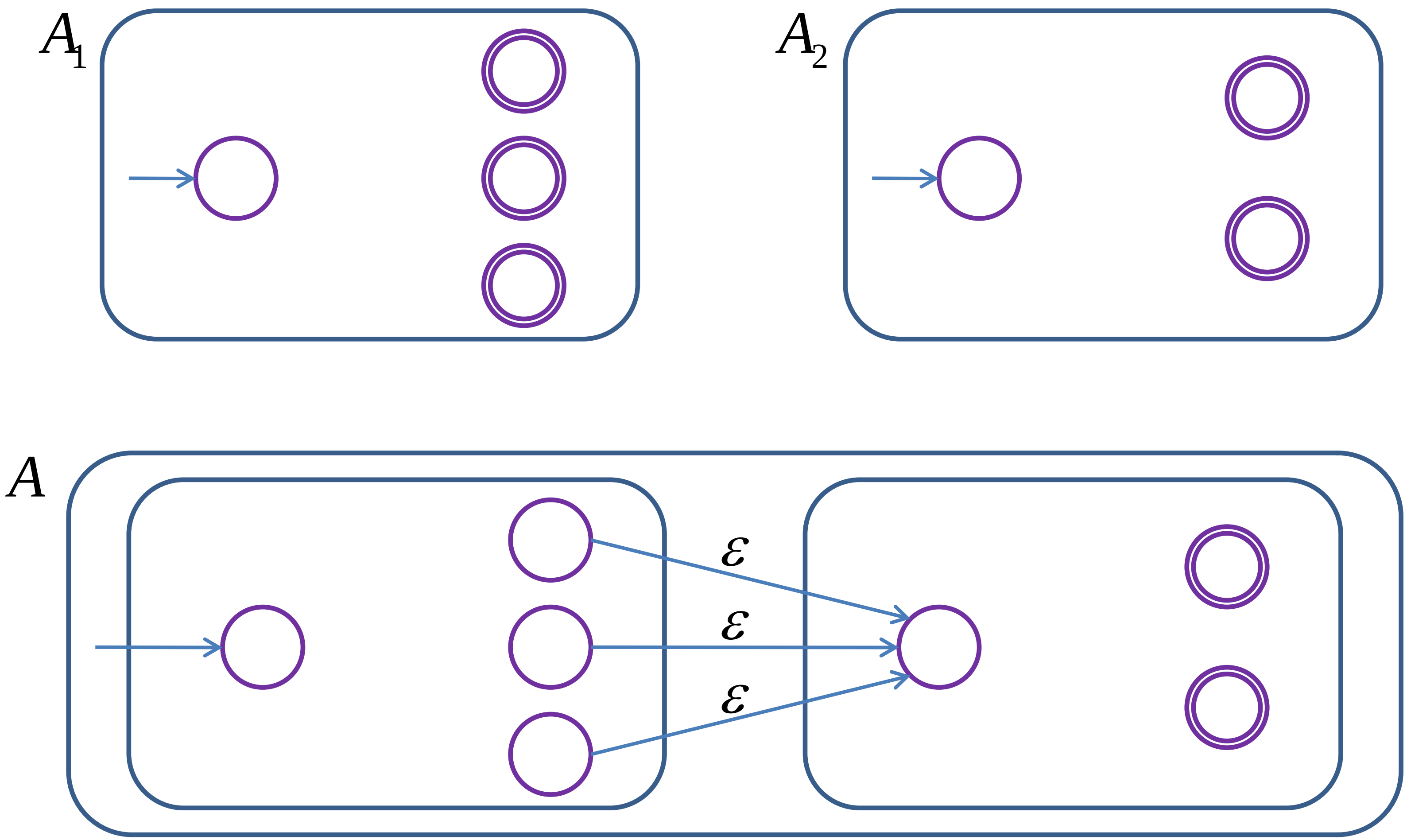
$$L^* = \bigcup_{n=0}^{\infty} L^n \qquad L^0 = \{\varepsilon\} \qquad L^{n+1} = L \cdot L^n$$

<u>Příklad</u>: $L = \{0,11\}$ $\qquad L^* = \{\varepsilon, 0, 00, 11, 000, 011, 110, \ldots\}$

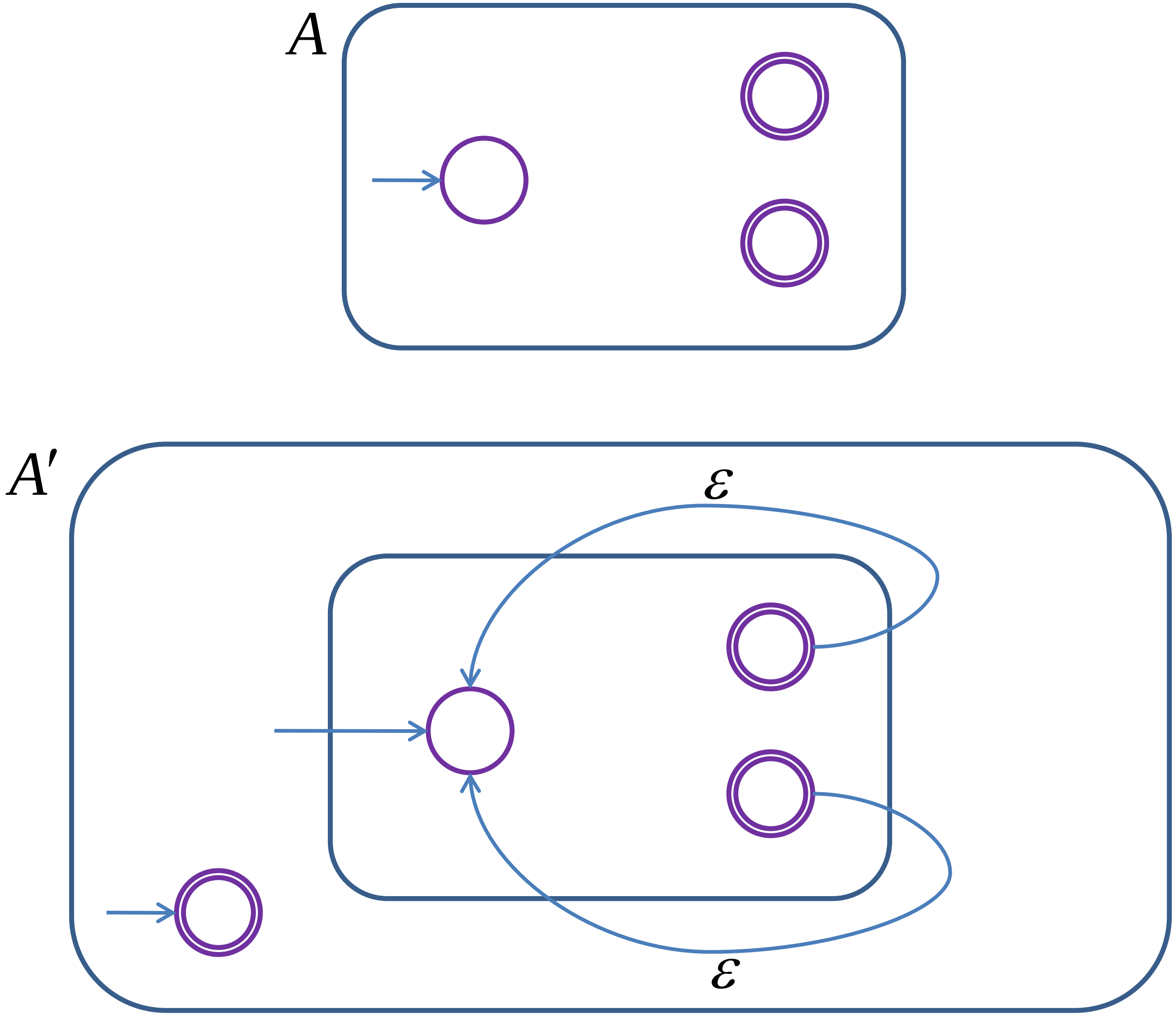
výskyt symbolu 1 je vždy „zdvojený“

Realizace zřetězení pomocí ZNKA

- Regulární operace zachovávají regulárnost jazyků.



Realizace iterace pomocí ZNKA



Regulární výraz

Regulární výraz nad abecedou Σ je každé slovo nad $\Sigma \cup \{\emptyset, \varepsilon, +, \cdot, *, (,)\}$ definované induktivně takto:

- $\emptyset, \varepsilon, a$ jsou regulární výrazy ($a \in \Sigma$)
- jestliže jsou α, β regulární výrazy, pak jsou také regulárními výrazy:

$(\alpha + \beta)$ .. sjednocení

$(\alpha \cdot \beta)$ .. zřetězení

(α^{*}) .. iterace

$[\alpha]$ .. označení pro jazyk reprezentovaný reg. výrazem α

Příklady, zjednodušení zápisu

Regulární výraz podle definice:

$$\left(\left(\left(0+\left(1\cdot 1\right)\right)^*\right)\cdot\left(\left(\left(\left(0\cdot 0\right)\cdot 0\right)+\left(1\cdot 1\right)\right)^*\right)\right)$$

Zjednodušený zápis (po vynechání závorek a operátoru zřetězení):

$$\left(0+11\right)^*\left(000+11\right)^*$$

Další příklad:

$$aa^*\left(\varepsilon+b\right)\left(cc\right)^*$$

Syntaxe regulárních výrazů v Pythonu

<table>
<tr><td>.</td><td>jakýkoliv znak kromě \n</td><td>a..d</td></tr>
<tr><td>[]</td><td>znak ze skupiny</td><td>[a-zA-Z01]</td></tr>
<tr><td>[^]</td><td>znak, který není ve skupině</td><td>[^abcde]</td></tr>
<tr><td>*</td><td>nic nebo opakování libovolněkrát</td><td>ab*</td></tr>
<tr><td>+</td><td>opakování alespoň 1x</td><td>ab+</td></tr>
<tr><td>?</td><td>nic nebo výskyt 1x</td><td>ab?</td></tr>
<tr><td>{m,n}</td><td>opakování m až n krát</td><td>a{3,5}</td></tr>
<tr><td>()</td><td>uzávorkování výrazu</td><td>([^a][0-1])+</td></tr>
<tr><td>|</td><td>nebo</td><td>(a+)|(b*)</td></tr>
</table>

Ekvivalence reg. výrazů a jazyků

Věta: Ke každému regulárnímu výrazu α lze sestrojit konečný automat přijímající jazyk $[\alpha]$.

- Konečný automat sestrojíme induktivně podle definice regulárního výrazu.
- Pro jazyky $\emptyset, \{\varepsilon\}, \{a\}$ můžeme sestrojit jednoduché KA.
- KA umíme sestrojit i pro sjednocení, zřetězení a iteraci.

Věta: Ke každému nedeterministickému konečnému automatu A lze sestrojit regulární výraz α takový, že $[\alpha] = L(A)$.

- Důkaz je zdlouhavý, vynecháme jej.