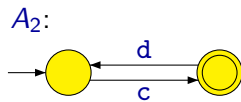
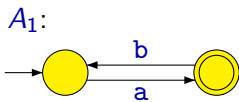


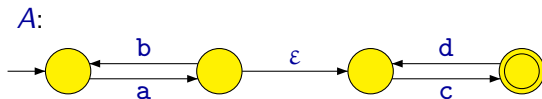
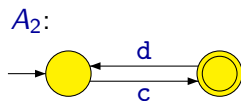
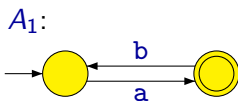
Zřetězení jazyků

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



Zřetězení jazyků

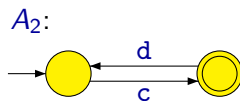
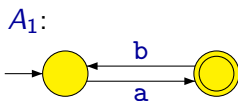
$\Sigma = \{a, b, c, d\}$



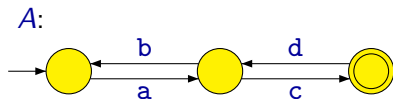
$$L(A) = L(A_1) \cdot L(A_2)$$

Zřetězení jazyků

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

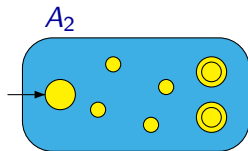
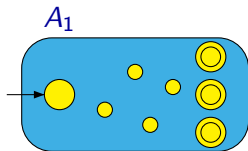


Chybná konstrukce:

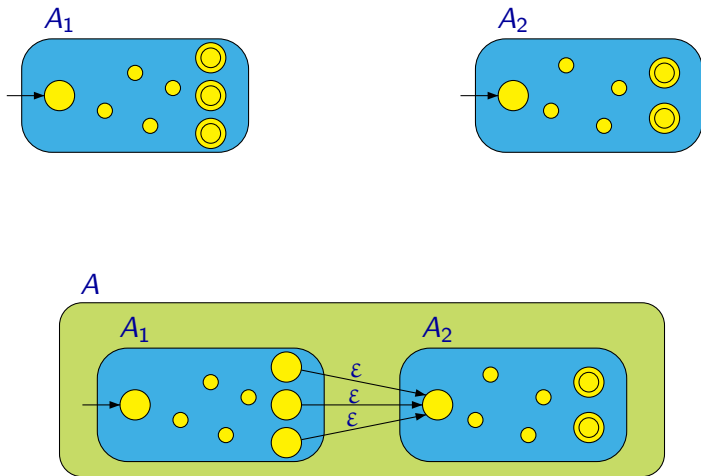


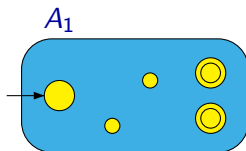
$acdbac \in L(A)$, ale $acdbac \notin L(A_1) \cdot L(A_2)$

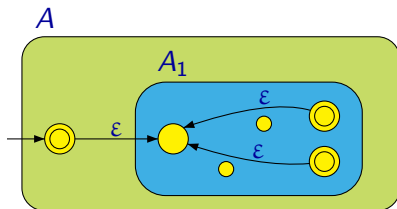
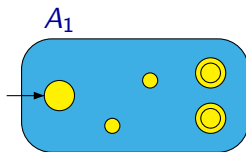
Zřetězení jazyků



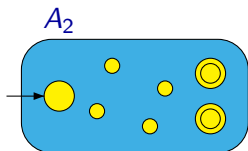
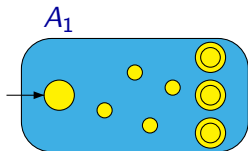
Zřetězení jazyků





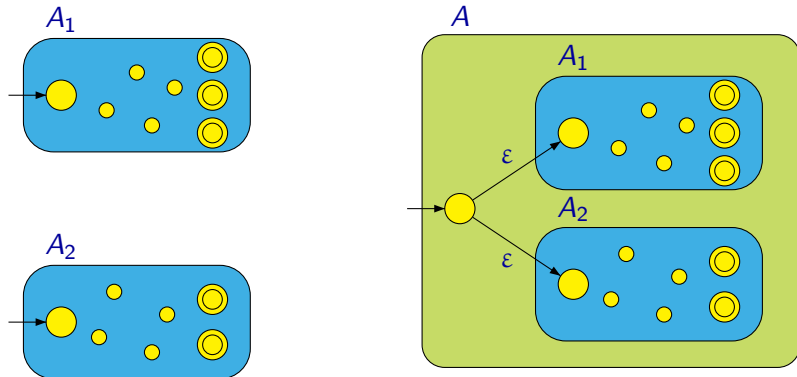


Alternativní konstrukce pro sjednocení jazyků:



Sjednocení jazyků

Alternativní konstrukce pro sjednocení jazyků:



Množina (všech) regulárních jazyků je uzavřená vůči operacím:

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

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 $[\alpha]$.

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 $[\beta]$ a $[\gamma]$.
 - Z nich sestrojíme automat rozpoznávající jazyk $[\alpha]$.

Převod regulárního výrazu na konečný automat

Automaty pro elementární výrazy:



\emptyset



ϵ



a

Převod regulárního výrazu na konečný automat

Automaty pro elementární výrazy:



\emptyset

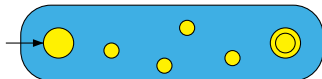
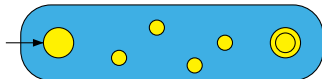


ϵ



a

Konstrukce pro sjednocení:



Převod regulárního výrazu na konečný automat

Automaty pro elementární výrazy:



\emptyset

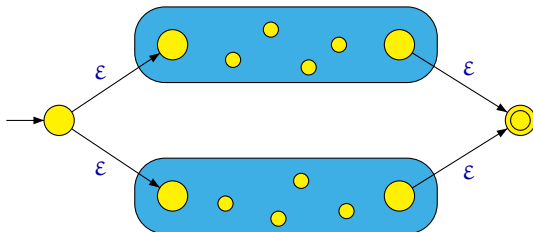


ϵ



a

Konstrukce pro sjednocení:



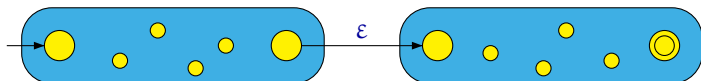
Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:



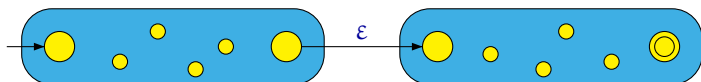
Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:

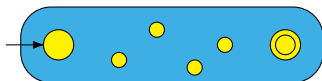


Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:

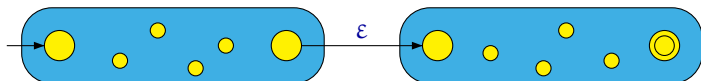


Konstrukce pro iteraci:

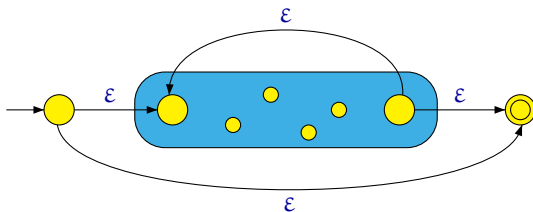


Převod regulárního výrazu na konečný automat

Konstrukce pro zřetězení:



Konstrukce pro iteraci:

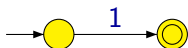


Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:

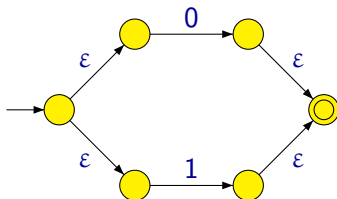
Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:



Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:

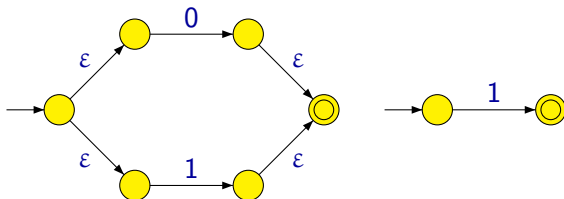


Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:



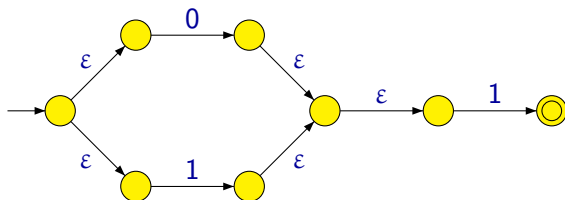
Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:

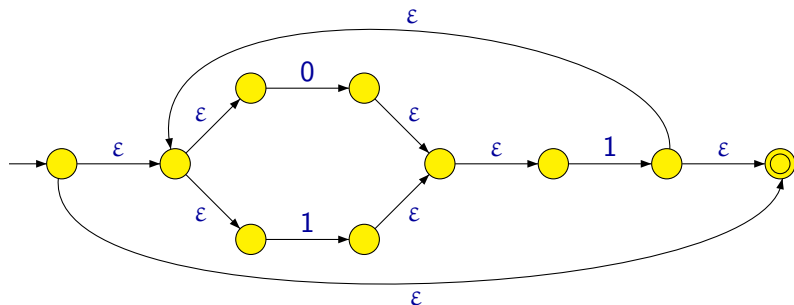


Převod regulárního výrazu na konečný automat

Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:



Příklad: Konstrukce automatu pro výraz $((0 + 1) \cdot 1)^*$:



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.

Tvrzení

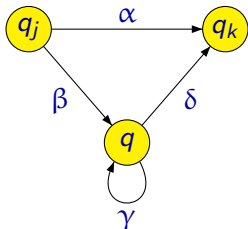
Každý regulární jazyk je možné popsat nějakým regulárním výrazem.

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

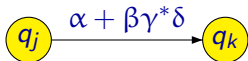
- A upravíme tak, aby měl právě jeden počáteční a právě jeden koncový stav.
- Budeme postupně odebírat jednotlivé stavy.
- Přejchody budou označeny regulárními výrazy.
- Zbude automat se dvěma stavy – počátečním a koncovým, a jedním přechodem ohodnoceným výsledným regulárním výrazem.

Převod konečného automatu na regulární výraz

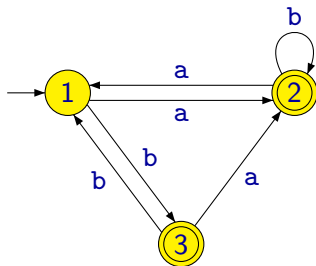
Hlavní myšlenka: Při odstraňování stavu q nahradit pro každou dvojici zbylých stavů q_j , q_k cestu z q_j do q_k vedoucí přes q .



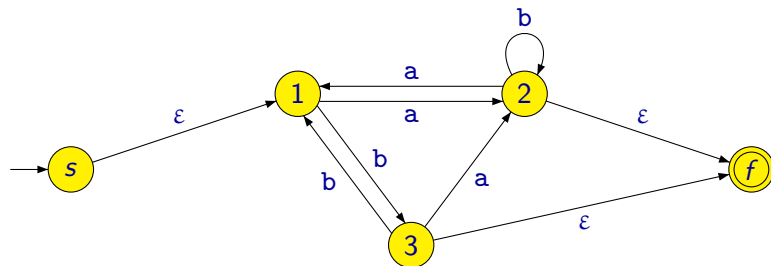
Po odstranění stavu q :



Příklad:

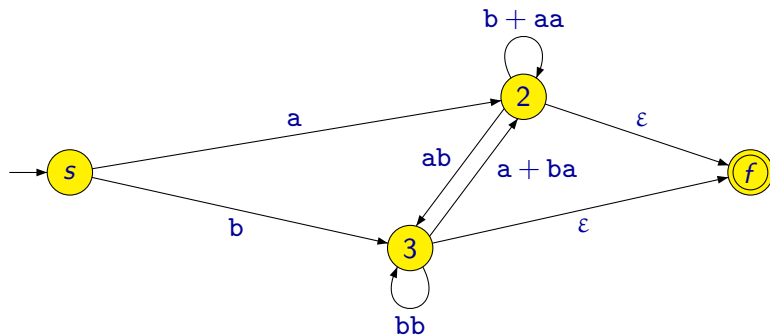


Příklad:



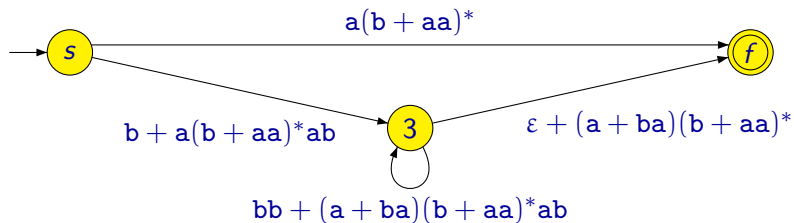
Převod konečného automatu na regulární výraz

Příklad:



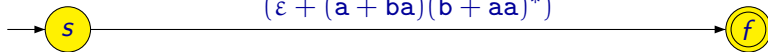
Převod konečného automatu na regulární výraz

Příklad:



Příklad:

$$\begin{aligned} & a(b + aa)^* + \\ & (b + a(b + aa)^* ab) \\ & (bb + (a + ba)(b + aa)^* ab)^* \\ & (\varepsilon + (a + ba)(b + aa)^*) \end{aligned}$$



Věta

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