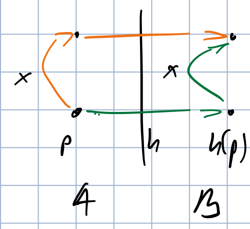
Homomorfismus automatů A, B nad stejnou abecedou:

$$A = (Q_A, \Sigma, \delta_A, q_0^A, F^A)$$

$$B = (Q_B, \Sigma, \delta_B, q_0^B, F^B)$$

$h: Q_A \rightarrow Q_B$



1) $h(q_A) = q_B$

2) $h(\delta_A(p,x)) = \delta_B(h(p), x)$

3) $p \in F_A \iff h(p) \in F_B$

Pokud existuje homomorfismus, pak jsou ekvivalentní.

Automatová kongruence $\equiv$

$$A = (Q, \Sigma, \delta, q_0, F)$$

$\forall p \equiv q$: 1) $p \in F \iff q \in F$

2) $\delta(p, x) = \delta(q, x) \quad \forall x \in \Sigma$

Faktorizace:

Vytvoření automatu na jeho faktor-množině.

-pak je to vytvořeno pomocí reprezentantů

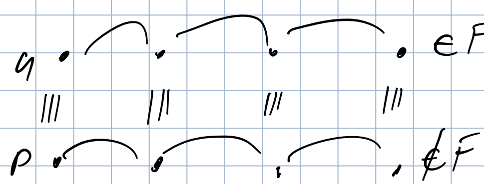
186.373

184.840

363.065

393.590

$p \equiv q \iff p \sim q$



-> tohle nenastane

$L(A) = L(A/\equiv)$

<= jde o homomorfismus, tam platí rovnost jazyků

Redukce aut. A:

1) $A \setminus \sim$

2) odstranění nedosažitelných stavů --> tohle je dobré dělat první

Procvičit hledání rozkladu ekvivalentních tříd !

Mějme A, B. Pokud $L(A) = L(B)$

$$\begin{array}{ccc} L(A) & = & L(B) \\ \downarrow & & \downarrow \\ A' & \cong & B' \end{array}$$

redukty $A' \cong B'$ --> existuje izomorfismus

Nedeterministické konečné automaty

$A = (Q, \Sigma, Q_0, \delta, F)$

-> množina

-> finální množina stavů

-> nová přechodová funkce

$$\delta : Q \times \Sigma \longrightarrow \mathcal{P}(Q)$$

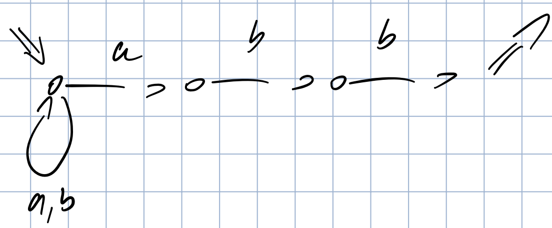
$$\delta^* : Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$$

$$\delta^*(p, \lambda) = \{p\} \quad \forall p \in \Sigma$$

$$\delta^*(p, xw) = \bigcup_{q \in \delta(p,x)} \delta^*(q, w) \quad \forall p \in \Sigma, \forall w \in \Sigma^*$$

$$L(A) = \{ w \in \Sigma^* \mid \exists q_0 \in Q_0 : \delta^*(q_0, w) \cap F \neq \emptyset \}$$

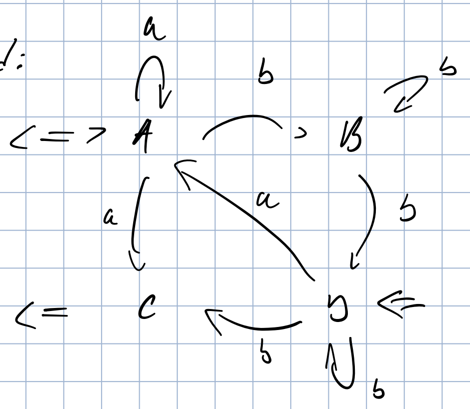
$$L = \{ w \in \{a, b\}^* \mid w \text{ končí na } abb \}$$

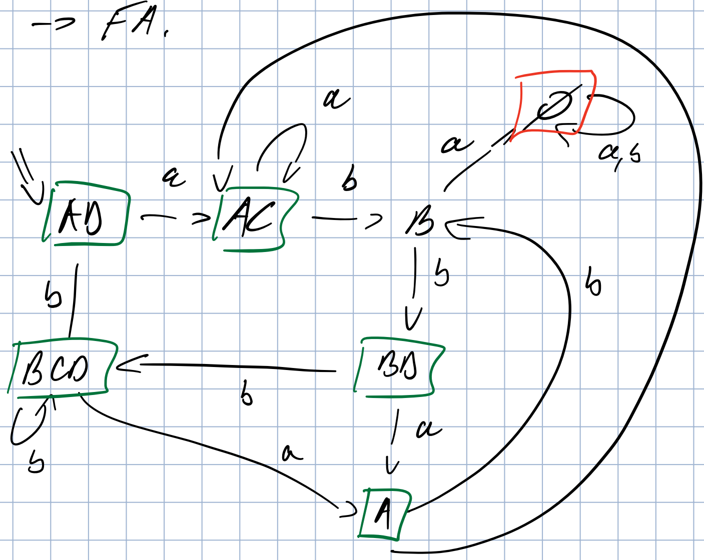


U konečných automatů jsou nedeterministické stejně silné co deterministické.

↳ Protože existuje převod NFA -> FA.

Převod:





– vytvořím nové stavy „jako BCD" když sloučím ty společné cesty do jedné. Jakmile jsou všechny cesty hotové, mám převod hotový.

přijímající stavy

fail stavy

Zároveň převod nemusí být redukovaný. Celkem je $2^{\#\text{stavů}}$ velký

DÚ: $\forall n \geq 2$:

$\exists L \in \mathcal{F}$ t.ž. $\exists$ NFA $A : L(A) = L$ s n stavy

$\forall$ DFA $B : L(B) = L$ musí mít alespoň 2^n stavů.

DFA: $\bar{A} = (Q, \Sigma, \delta, q_0, Q \setminus F)$

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

Nechť nedeterministický automat $A = (Q, \Sigma, \delta, q_0, \{q_0\})$
pro $Q = \{1 \_\_ n\}$ pro libovolné $n \in \mathbb{N} > 2$.

Mějme dvouznakovou abecedu $\Sigma = \{a, b\}$, přechodovou funkci δ
definovanou jako:

tedy „a" vede do dalšího stavu

$$\delta(i, a) = \{i+1\} \quad \text{pro} \quad 1 \leq i \leq n-1$$

$$\delta(n, a) = \{1\}$$ → vytváří smyčku

„b" mě vrací na začátek nebo na stejné místo.

$$\delta(i, b) = \{1, i\} \quad \text{pro} \quad 2 \leq i \leq n$$

Takový automat má zjevně jenom n stavů. Nyní ukážeme, že
ekv. deterministický má alespoň 2^n stavů. Totiž libovolná podmnožina
stavů Q je dosažitelným stavem deterministického automatu: → $B = (Q', \Sigma, \delta', q_0', F')$

1) Pro prázdnou množinu platí: $\delta'(1, b) = \emptyset$
2) Pro neprázdnou podmnožinu (např.) $\{k_1 \_ k_l\} \subseteq Q'$ mějme $k_i < k_j$ pro $i < j$
   a označme $d_i = k_{i+1} - k_i$ pro $1 \leq i < l$, tedy vzdálenost
   mezi „sousedními" stavy z množiny.

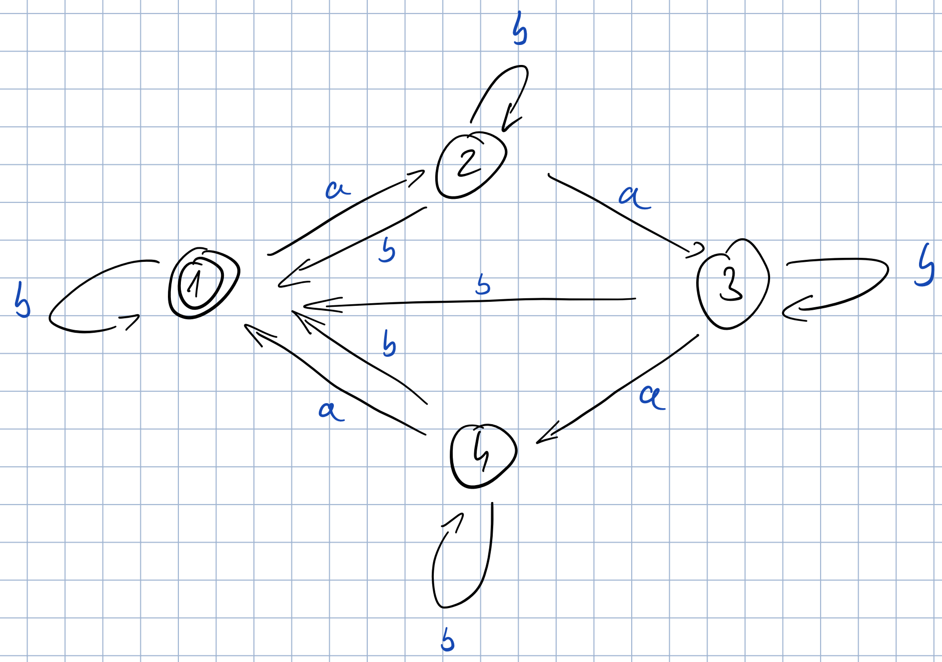
   Pak daný stav $\{k_1 - k_l\}$ je v B dosažitelný slovem

$$a^{d_{l-1}} b a^{d_{l-2}} b a^{d_{l-3}} \dots a^{d_1} b a^{k_1 - 1}.$$

Tedy každá podmnožina (kterých je 2^n) odpovídá jednomu stavu
deterministického automatu. Nyní stačí dokázat, že žádné dva
takové stavy nejsou ekvivalentní, tudíž jich existuje právě tolik co podmnožin.

Mějme dvě podmnožiny $P_1, P_2 \subseteq Q$. Pak určitě $P_1 \neq P_2$.
tedy určitě existuje $k \in \{1 \_ n\}$, které $\in P_1$ ale $\notin P_2$ nebo obráceně.
Pokud vybereme slovo a^{n-k+1}, tak tím tyto dva stavy rozlišíme,
jelikož stav $\delta^*(q_0, a^{n-k+1})$ nebude zastoupen v množině, co
neobsahovala k.

Pro lepší ilustraci přiložím obrázek nedeterministického automatu pro $n=4$.



Tedy pro zadané „n" jazyk obsahuje #a jako násobek n, pokud je zakončen znakem a nebo je zakončen znakem b a násobek #a již není potřeba. Zejména je potřeba si všimnout, že můžu libovolně přidávat znaky b, tudíž mi roste počet možností...

Daný jazyk je tedy popsán nedeter. automatem popsaným výše.