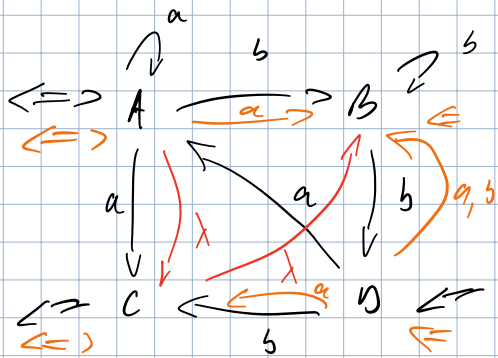


$$\delta: Q \times \Sigma \cup \{\lambda\} \rightarrow P(Q)$$

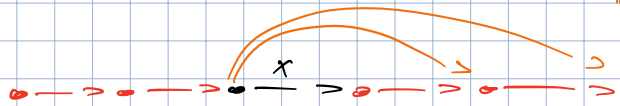


lambda přechody

"po zmizení lambda"

přidám ještě cesty

přes lambda přechod



$$\lambda\text{-uzavřít}(x) = \{y \mid \exists \text{ cesta z } x \text{ do } y \text{ přes } \lambda\}$$

$$\lambda\text{-uz}(A) = \{A, C, B\}$$

$$\lambda\text{-uz}(B) = \{B\}$$

$$\lambda\text{-uz}(C) = \{C, B\}$$

$$\lambda\text{-uz}(D) = \{D\}$$

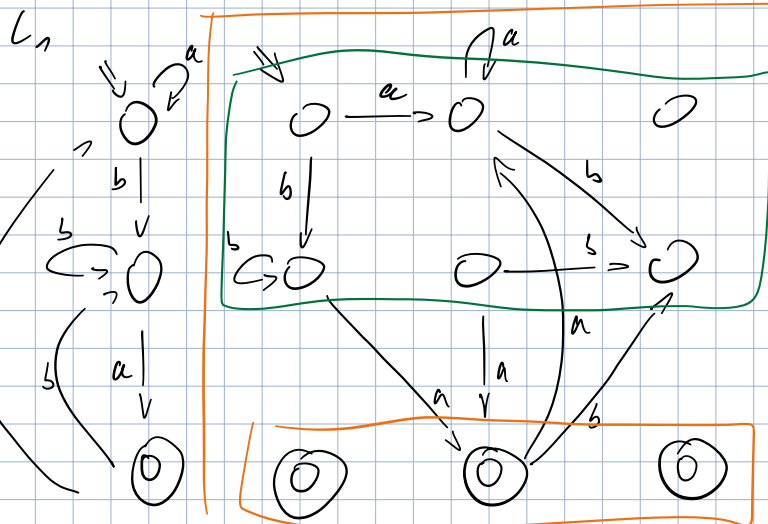
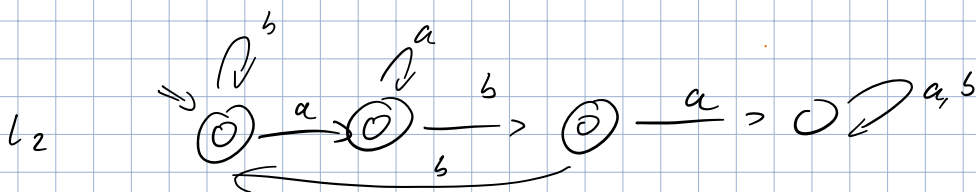
$$\text{přes } \delta'(p, x) = \bigcup_{q \in \delta(p, x)} \lambda\text{-uz}(q)$$

$$F' = F$$

$$S' = \bigcup_{q \in S} \lambda\text{-uz}(q) \rightarrow \text{sjednocení lambda uzavření výchozích stavů}$$

$$L_1 = \{w \in \{a, b\}^* \mid w \text{ končí na } ba\}$$

$$L_2 = \{w \in \{a, b\}^* \mid w \text{ neobsahuje } aba\}$$



Sjednocený automat

- není dokončený,

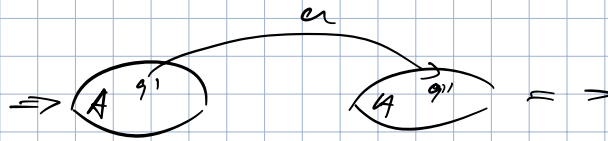
ale myšlenka je jasná

$$L_1 \Delta L_2 \text{ sym. rozdíl}$$

$$L_1 \cap L_2$$

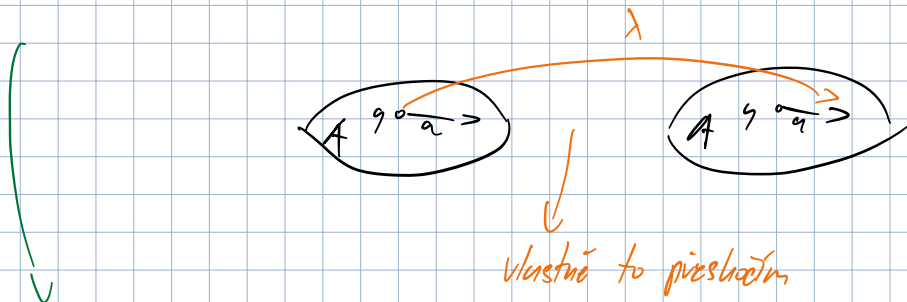
δ_{L_1, L_2} jde po složkách, tedy: $\delta_{L_1, L_2}((q_1, a_2), x) = (\delta_1(q_1, x), \delta_2(q_2, x))$

$$\text{ins}_n(L) = \{uqv \mid uv \in L\}$$

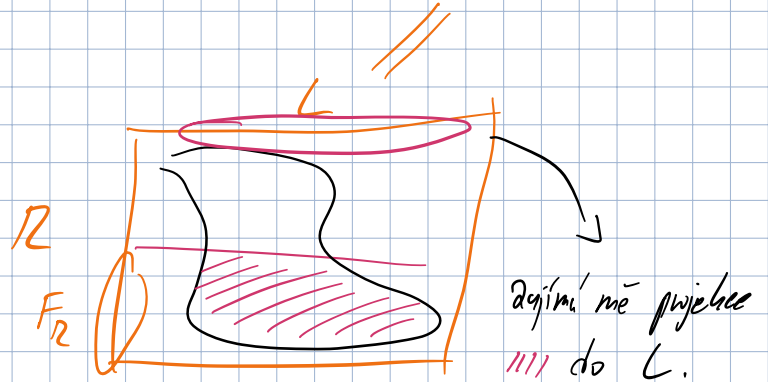


- to stejné: pro slovo z reg. jazyka máme jeho vlastního automatu

$$\text{del}_a(L) = \{uv \mid uav \in L\}$$



pro množinu slov z reg. jazyka by to šlo sestrojit automaty jazyků L a R .

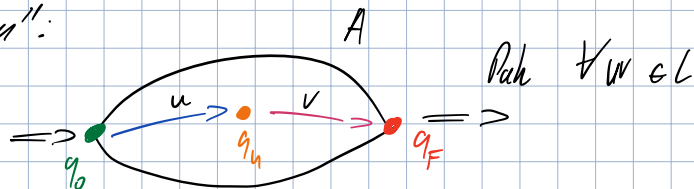


DŮ: $\forall$ reg. jazyk:

Ukážte, že shift je reg. jazyk.

$$\text{shift}(L) = \{uv \mid vuv \in L\}$$

Mějme vez. jazyk L a k němu automat A t.č. $L(A) = L$. Automat si budeme reprezentovat „šísčou“:



můžeme symbolický znázornit jako výše v mši „šísčou“.

Automat A' , který má přijímat libovolné shiftované slovo z původního jazyka L , musí reflektovat každý možný výchozí posun, kterým bude výraz jako první stav.

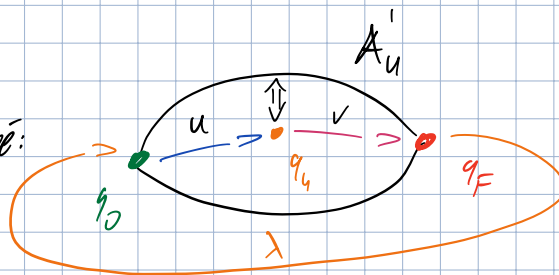
Pak jsem se „posunul“ už znovu na začátku.

Uvažme tedy libovolný stav q_u jako stav $\tilde{q}(q_0, u)$, tedy stav po přečtení lib. podřetězce u . Pak pokud má automat A' přijímat slovo vu , stane se stav q_u přijímajícím stavem. Stejně tak se ale stane výchozím stavem. Stačí zajistit,

že po přečtení části slova v , tedy ve stavu $\tilde{q}(q_u, v) = q_F$ můžeme pokračovat ve čtení, tedy zavedu λ -přechod mezi původním přijímajícím a výchozím.

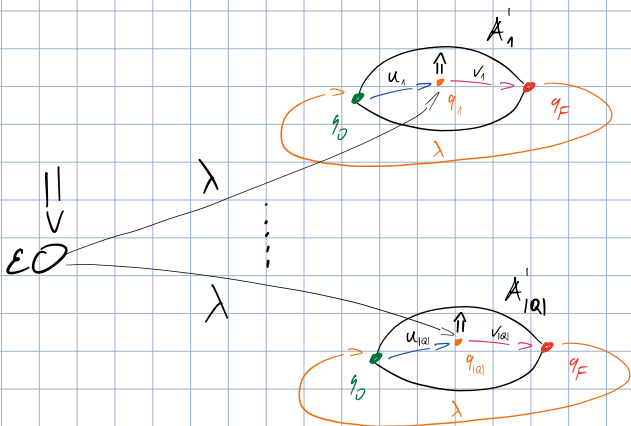
Pak budu schopen přičíst i první část (u) a pokud ji přečtu celou, skončím v přijímajícím stavu q_u .

Upravený automat pro lib. stav q_u vypadá následovně:



Nyní takový automat musím vytvořit pro každý stav, jelikož, jak zmíněno výše, možných shiftů může být vícero, takže je stavů. (pak by se řešilo „modul $|Q|$ “ opakovaně).

Proto všem stavům vytvořím nový globální výchozí stav a ke každému přijímajícímu stavu z A' připojím z nového výchozího stavu λ -přechod. Výsledný automat vypadá následovně:

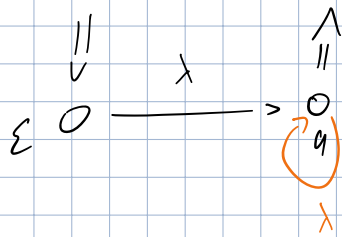


Tedy automat bude vyhodnocovat všechny možné posuny paralelně.

Jelikož stavů aut. A je konečně mnoho, bude i vytvořený A' konečný.

Z komentáře výše je zřejmé, že pokud je čtení slova shiftem původního, tak jeden podautomat A' ho určitě musí přijímat. Tedy i celý automat.

Speciální případ shiftování prázdného slova:



Záměr shift prázdného
slova i když je
zase prázdné slovo.