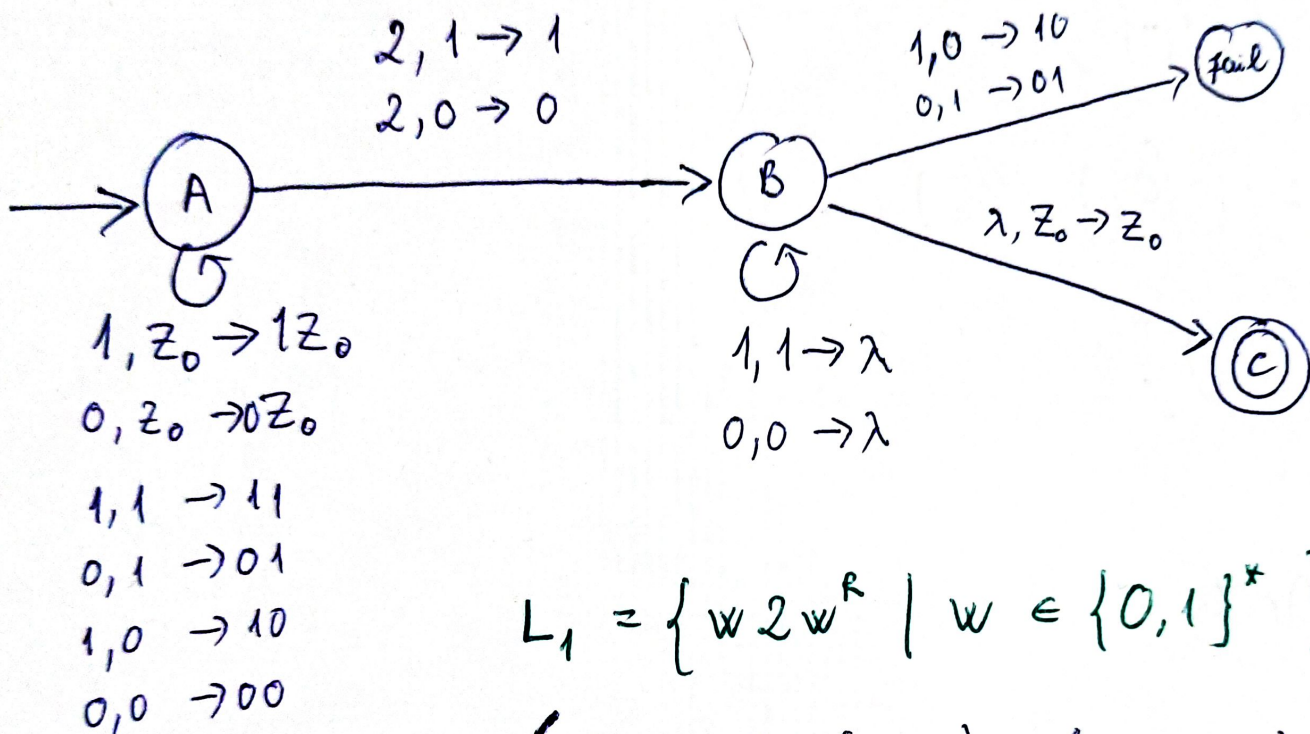


Uzel fail je většinou zbytečný, použije se pouze u deterministického PDA když tvoříte automat přijímající doplněk jazyka. Ale neškodí, přechodová funkce je v pořádku.

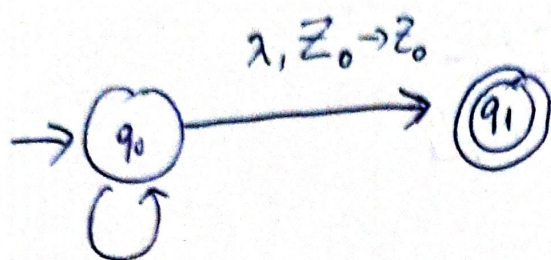


$$L_1 = \{ w 2 w^R \mid w \in \{0,1\}^* \}$$

$$P = (\{A, B, C\}; \{0, 1\}; \{z_0, 0, 1\}; \delta; \{A\}; \{z_0\}; \{C\})$$

Počáteční stav a zásobníkový symbol vždy jen jeden, tj. bez množinových závorek. Jinak správně.

$$L_3 = \{ w \mid w \in \{0,1\}^* \text{ \& } |w|_0 = |w|_1 \}$$



$$1, z_0 \rightarrow 1z_0$$

$$0, z_0 \rightarrow 0z_0$$

$$1, 1 \rightarrow 11$$

$$0, 0 \rightarrow 00$$

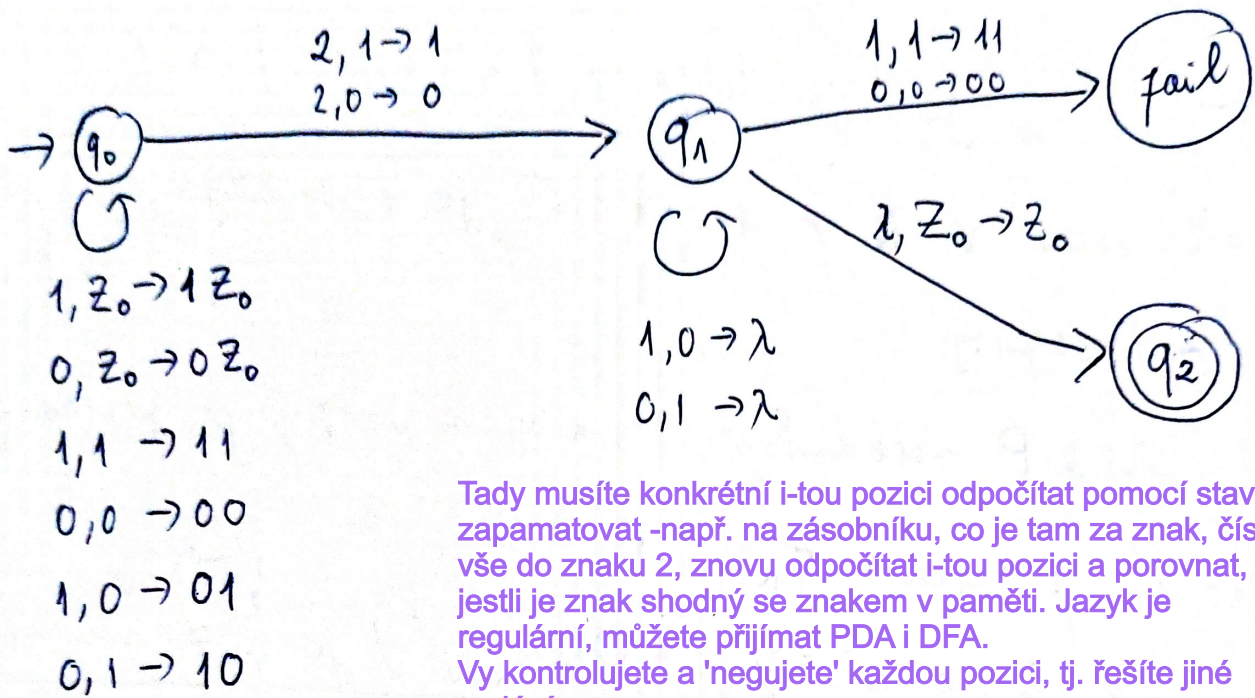
$$1, 0 \rightarrow \lambda$$

$$0, 1 \rightarrow \lambda$$

$$P = \left(\{q_0, q_1\}; \{0, 1\}; \{z_0, 0, 1\}; \right. \\ \left. \delta; q_0; z_0; \{q_1\} \right)$$

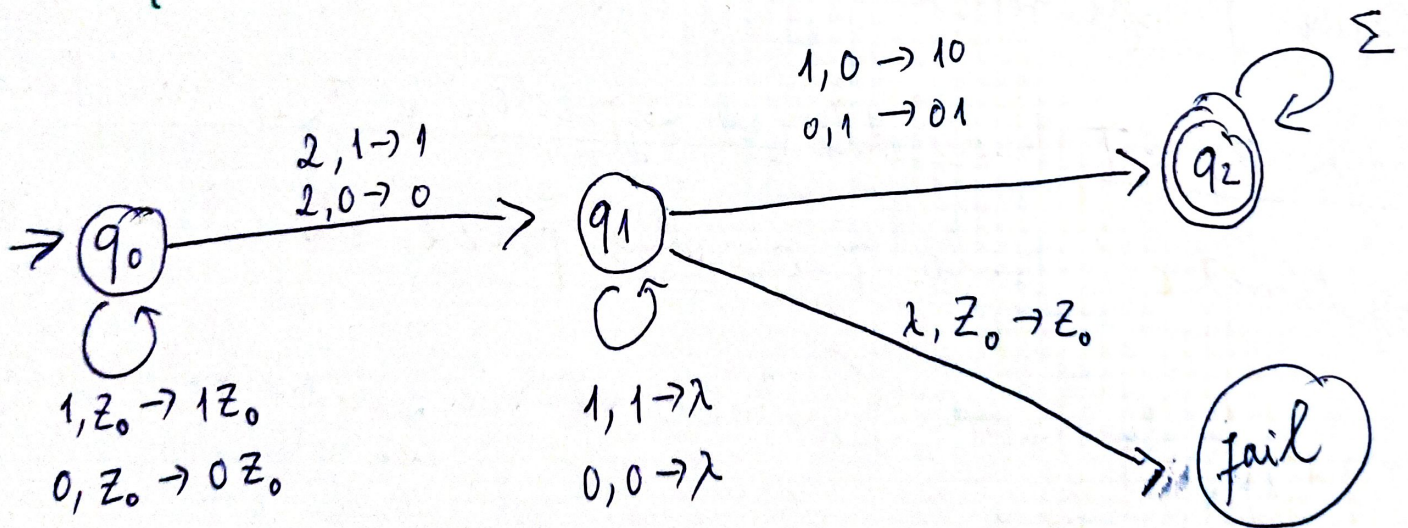
Naprosto správně.

$$L_i = \{ u \sqcup v \mid u, v \in \{0,1\}^* \& u[i] \neq v[i] \}$$



Tady musíte konkrétní i-tou pozici odpočítat pomocí stavů, zapamatovat -např. na zásobníku, co je tam za znak, číst vše do znaku 2, znovu odpočítat i-tou pozici a porovnat, jestli je znak shodný se znakem v paměti. Jazyk je regulární, můžete přijímat PDA i DFA. Vy kontrolujete a 'negujete' každou pozici, tj. řešíte jiné zadání.

$$L_5 = \{ u \sqcup v^R \mid u, v \in \{0,1\}^* \& u \neq v \}$$



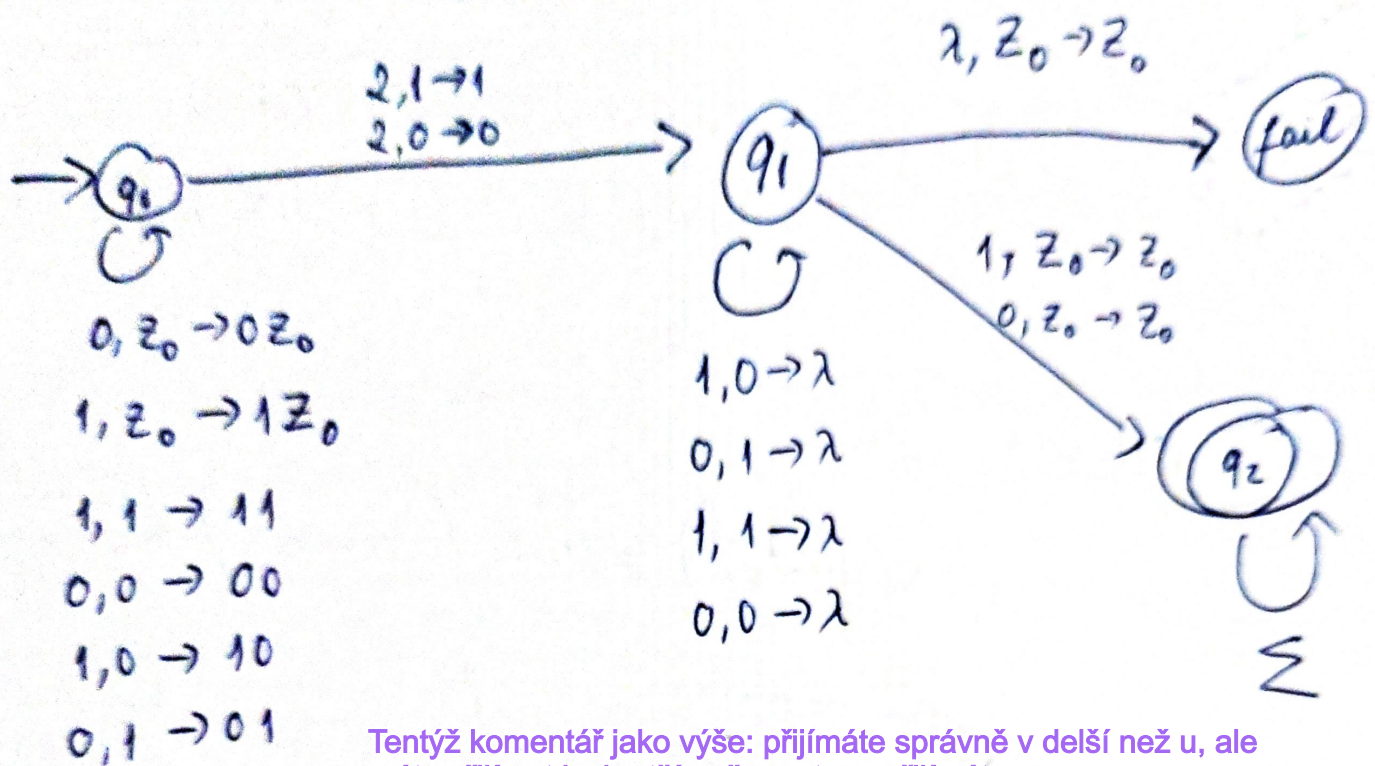
Tady když je $u \sqcup v^R$ nebo $u \sqcup v$ tak je to jedno ne?
Podle mě je stačí aby $u \neq v$

K předchozí straně:

Váš automat přijímá správně slova s odlišným znakem, ale nepřijmete, pokud v je kratší než u . To byste si musel označit poslední znak u , jeho přečtením při celkové shodě přejít do nepřijímacího stavu, dalším znakem do přijímacího.

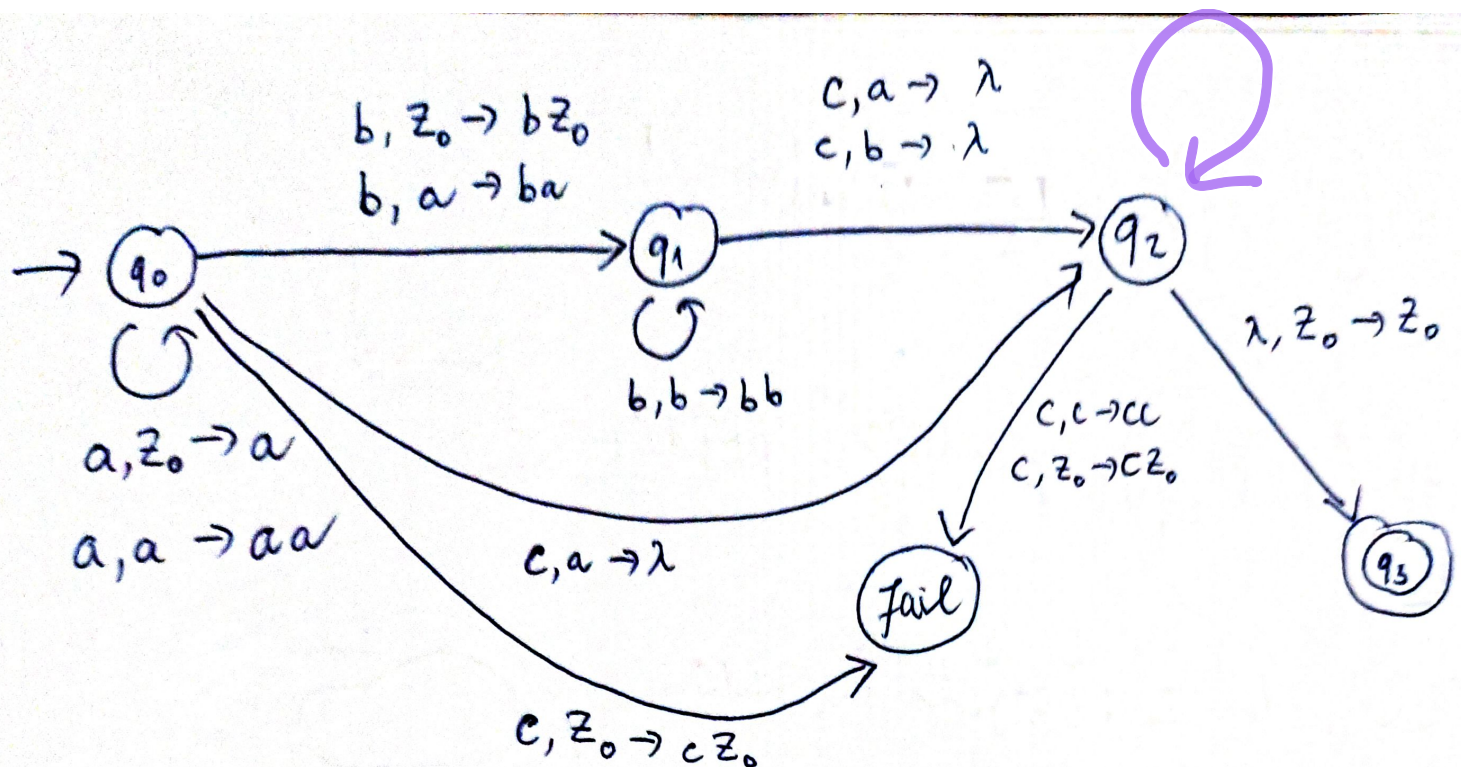
K červenému dotazu: odlišnost u a v bez reverze by byla výrazně složitější. Dělá se přes sjednocení jednotlivých variant, jak se u a v liší. Jednodušší je pro něj psát gramatiku než automat (alespoň pro mne).

$$L_4 = \{ u \neq v \mid u, v \in \{0,1\}^* \ \& \ |u| \neq |v| \}$$



Tentýž komentář jako výše: přijímáte správně v delší než u , ale máte přijímat i v kratší než u , a to nepřijímáte.

$c, b \rightarrow \lambda$
 $c, a \rightarrow \lambda$



$$L_G = \{ a^i b^j c^{i+j} \mid i, j \in \mathbb{N}_0 \}$$

Chybí Vám tam fialově domalovaná smyčka
 na čtení vstupních c , jinak správně.
 Uzel fail nevadí ale není třeba, viz komentář k
 1. příkladu.

$$L_7 = \{ a^i b^j c^{i+j} \mid i, j = 0, 1, 2, \dots \}$$

$$a^n b^n c^{n^2} \in L_7$$

Rozdělím slovo na $uxyzv$

Postupně budu rozebírat:

$$xz = \begin{bmatrix} a^k & a^l \\ a^k & b^l \\ b^k & b^l \\ b^k & c^l \\ c^k & c^l \end{bmatrix}$$

$$k+l \leq n$$

Pokud $k=1$ a $l=n$, tak iterováním dostanu $a^n b^{(n+1)} c^{(n^2+n)}$, o kterém máte dokázat, že nepatří do jazyka.

I u ostatních variant poznamenat, že bude vždy méně c (při iterování a a/nebo b) nebo více c (při iterování c).