# Cvičenie 9 - Rozšírenia TS, vlastnosti RE jazykov

Ing. Viliam Hromada, PhD.

C-510
Ústav informatiky a matematiky
FEI STU

viliam.hromada@stuba.sk

2.12.2020

# TS

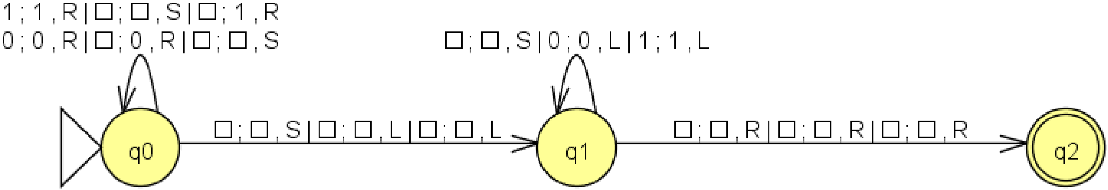
Zostrojte TS, ktoré akceptujú uvedené jazyky. Ak to uvážite za vhodné, môžete zostrojiť TS s viacerými páskami, nedeterministické Turingove stroje, prípadne nedeterministické TS s viacerými páskami:

- $L_1 = \{w \in \{0,1\}^* \mid \sharp_0(w) = \sharp_1(w)\}$
- $L_2 = \{wcw^R \mid w \in \{a,b\}^*\}$
- $L_3 = \{w_1 \sharp w_2 \sharp \ldots \sharp w_n \mid n \geq 0, w_i \in \{0,1\}^*, w_j = 010$ pre nejaké $j \in \{1, \ldots, n\}\}$
- $L_4$ je jazyk binárnych reťazcov, v ktorých sa opakuje nejaký podreťazec dĺžky 3, nie nevyhnutne za sebou, t.j. $L_4 = \{wxyxz \mid w, x, y, z \in \{0,1\}^*, |x| = 3\}$
- $L_5 = \{ww \mid w \in \{a,b\}^*\}$

# Jazyk $L_1$

Nájdite TS akceptujúci jazyk $L_1 = \{w \in \{0,1\}^* \mid \sharp_0(w) = \sharp_1(w)\}$.

- Uvedený jazyk tvoria reťazce, v ktorých je rovnaký počet núl a jednotiek.
- Vedeli by sme zostrojiť TS s jednou páskou, ktorý takýto jazyk akceptuje - my však využijeme fakt, že môžeme mať aj pásky tri.
- Z prvej pásky čítame vstup, na druhú pásku si zapíšeme každú prečítanú 0, na tretiu pásku zapíšeme každú prečítanú 1.
- Po prečítaní celého vstupu z prvej pásky následne skontrolujeme, či je na druhej a tretej páske rovnaký počet symbolov - ak áno, vstup akceptujeme.


1 ; 1 , R | □ ; □ , S | □ ; 1 , R
0 ; 0 , R | □ ; 0 , R | □ ; □ , S
□ ; □ , S | 0 ; 0 , L | 1 ; 1 , L
□ ; □ , S | □ ; □ , L | □ ; □ , L
□ ; □ , R | □ ; □ , R | □ ; □ , R
q0
q1
q2

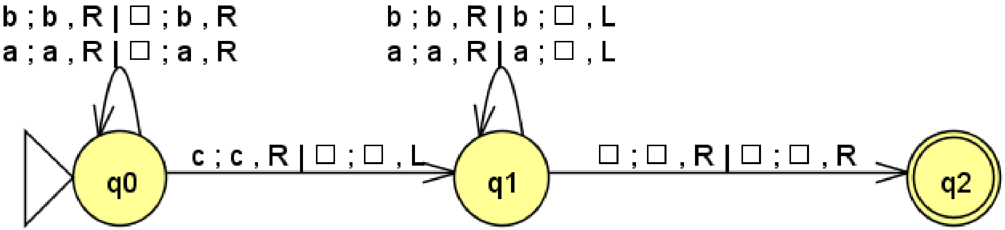
Vysvetlenie:

- Stav $q_0$: z pásky č. 1 prečítame symbol - ak je to 0, tak ju vložíme na pásku č. 2, ak je to 1, vložíme na pásku č. 3
- Keď narazíme na koniec slova na páske č. 1, presunieme sa do stavu $q_1$, zároveň sa na páske č. 2 a č. 3 posunieme vľavo - na koniec vložených slov.
- Stav $q_1$: na páske č.2 musí byť čítaný symbol 0 a súčasne na páske č. 3 musí byť čítaný symbol 1. V takom prípade oba zmažeme a posúvame sa ďalej vľavo.
- Ak sa v stave $q_1$ podarí súčasne pásky č. 2 a č. 3 zmazať, tak je možný prechod do stavu $q_2$, ktorý je akceptačný.
- Táto situácia nastane len v prípade, že na oboch páskach boli rovnako dlhé slová, čo muselo znamenať, že na prvej páske bol reťazec s rovnakým počtom 0 a 1.

# Jazyk $L_2$

Nájdite TS akceptujúci jazyk $L_2 = \{wcw^R \mid w \in \{a, b\}^*\}$

- Uvedený jazyk tvoria palindrómy z $\{a, b, c\}$, kde $c$ sa nachádza len uprostred.
- Vedeli by sme zostrojiť TS s jednou páskou, ktorý takýto jazyk akceptuje - my však využijeme fakt, že môžeme mať dve pásky
- Najprv z prvej pásky čítame vstup, súčasne si na druhú pásku postupne zapíšeme každý prečítaný symbol.
- Po prečítaní symbolu $c$ začneme porovnávať, či čítaný symbol na prvej páske je rovnaký, ako symbol na konci druhej pásky. Ak áno, tak symbol na druhej páske zmažeme a pokračujeme v tomto porovnávaní ďalej.
- Ak dočítame slovo na prvej páske a súčasne všetko z druhej pásky zmažeme, tak máme istotu, že na vstupe bolo slovo $wcw^R$, $w \in \{a, b\}^*$.


b ; b , R | □ ; b , R
a ; a , R | □ ; a , R
b ; b , R | b ; □ , L
a ; a , R | a ; □ , L
c ; c , R | □ ; □ , L
□ ; □ , R | □ ; □ , R
q0
q1
q2

Vysvetlenie:

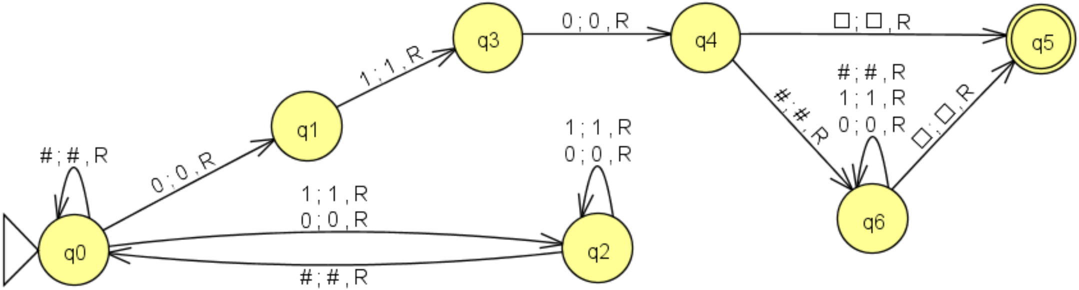
- Stav $q_0$: z pásky č. 1 prečítame symbol a zároveň ho zapíšeme na ďalšiu voľnú pozíciu na 2. páske
- Keď narazíme na symbol $c$ na prvej páske, tak sa TS prepne do stavu $q_1$. Súčasne sa na 2. páske posunie hlava vľavo na posledný neprázdny symbol druhej pásky.
- Stav $q_1$: z pásky č. 1 čítame symbol, súčasne musí byť taký istý symbol ako posledný neprázdny symbol na 2. páske. Ak áno, tak ho z 2. pásky zmažeme a posunieme sa vľavo na aktuálny posledný neprázdny symbol 2. pásky.
- Ak sa v stave $q_1$ podarí súčasne dočítať slovo na páske č. 1 do konca a všetko z pásky č. 2 zmazať, tak vstup musel byť v tvare $wcw^R$, $w \in \{a, b\}^*$.
- V takom prípade čítame na 1. a 2. páske prázdne symboly - vtedy prejdeme do akceptačného stavu $q_2$.

# Jazyk $L_3$

Nájdite TS akceptujúci jazyk
$L_3 = \{w_1\#w_2\#...\#w_n \mid n \geq 0, w_i \in \{0,1\}^*, w_j = 010 \text{ pre nejaké } j \in \{1,...,n\}\}$

- Uvedený jazyk tvoria reťazce zložené z binárnych podreťazcov $w_i$ oddelených symbolom $\#$, pričom minimálne jeden z týchto binárnych reťazcov je 010.
- Také reťazce sú napríklad:
  - 010
  - $\#010\#$
  - $1\#000\#010\#$
  - $\#1\#01\#000\#010$

- Úlohou TS bude hľadať medzi symbolmi $\#$ postupnosť núl a jednotiek 010.
- V prípade, že ju TS nájde, tak môže vstup akceptovať.
- Keďže dopredu nevieme, či za symbolom $\#$ nasleduje práve postupnosť 010, alebo nejaká iná binárna postupnosť, tak sa vždy nedeterministicky rozhodujeme medzi obyčajným čítaním postupnosti a medzi jej porovnávaním s 010. V prípade, že sa nájde 010$\#$, tak prejdeme do akceptačného stavu.


#;#,R
0;0,R
1;1,R
0;0,R
□;□,R
q0
q1
q3
q4
q5
1;1,R
0;0,R
#;#,R
1;1,R
0;0,R
q2
#;#,R
#;#,R
1;1,R
0;0,R
□;□,R
q6

Vysvetlenie:

- Stav $q_0$: V stave $q_0$ sa nedeterministicky rozhodneme, či práve čítaná nula začína hľadaný reťazec 010, v takom prípade prejdeme do $q_1$. Ak čítame nulu, ktorý je len súčasťou iného reťazca, tak prejdeme do $q_2$. Ak čítame $\#$, tak to znamená, že $w_i$ bol prázdny a zostávame v $q_0$.
- Stav $q_2$: v stave $q_2$ dočítame aktuálny reťazec $w_i$ do konca a keď narazíme na $\#$, tak sa vrátime do $q_0$.
- Zo stavu $q_1$ prejdeme cez $q_3$ do $q_4$ len v prípade, ak prečítame práve 010. Ak bol 010 zároveň posledný binárny podreťazec vstupu, tak je na páske čítaný prázdny symbol a prejdeme do akceptačného stavu $q_5$.
- Ak sa za reťazcom 010 nachádza $\#$ a za ním ešte ďalšie binárne reťazce oddelené $\#$, tak prejdeme do stavu $q_6$, v ktorom dočítame zvyšok vstupu po prvý prázdny symbol za ním a jeho čítaním prejdeme do akceptačného stavu $q_5$.

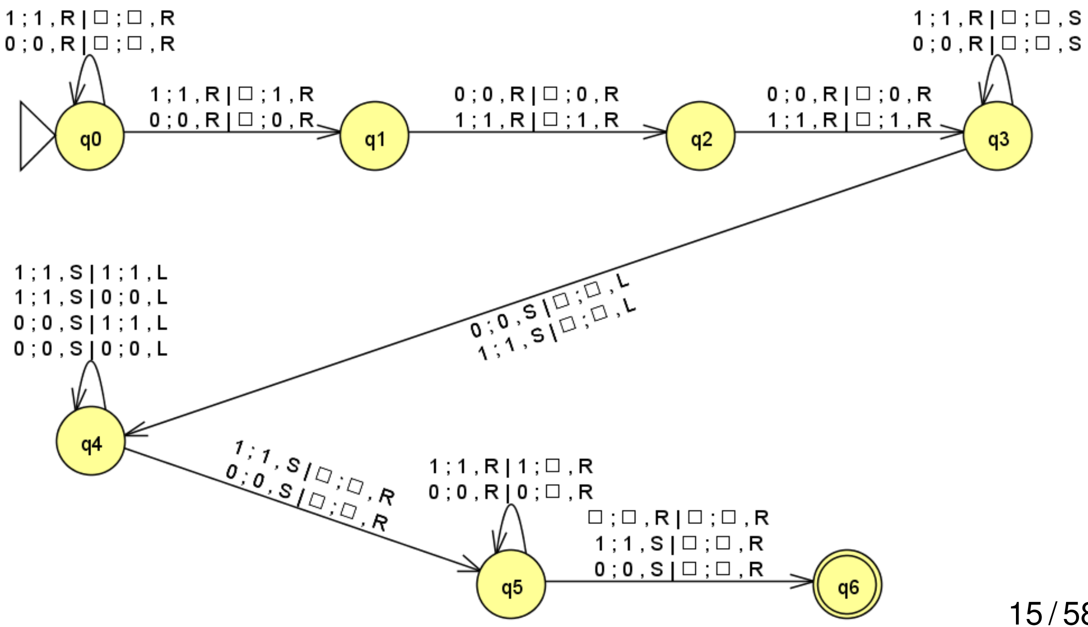
# Jazyk $L_4$

Nájdite TS akceptujúci jazyk $L_4$, ktorý tvoria reťazce, v ktorých sa opakuje nejaký podreťazec dĺžky 3, nie nevyhnutne za sebou, t.j.
$L_4 = \{wxyxz \mid w, x, y, z \in \{0, 1\}^*, |x| = 3\}$.

Také reťazce sú napríklad (t.j. vždy sa v nich dá nájsť nejaká trojica bitov, ktorá sa zopakuje, označená $x$)

- 110110, lebo $w = y = z = \varepsilon, x = 110$
- 0101010, lebo $w = z = \varepsilon, y = 1, x = 010$
- 011111110, lebo $w = 01, y = \varepsilon, z = 0, x = 111$
- 011111110, lebo $w = 0, y = \varepsilon, z = 10, x = 111$
- 1100001101, lebo $w = \varepsilon, y = 000, z = 1, x = 110$

- Tento jazyk vieme akceptovať napríklad nedeterministickým Turingovym strojom s 2 páskami.
- Chceme akceptovať také reťazce, ktoré sú v tvare $wxyxz$, kde $|x| = 3$. To znamená, že počas čítania vstupu sa nedeterministicky rozhodneme, či práve čítaná časť vstupu predstavuje $w$,$x$,$y$,$x$,$z$.
- Ak čítame prvý výskyt $x$, tak si ho zároveň skopírujeme na druhú pásku. Následne, pri "potenciálnom" druhom čítaní $x$ zo vstupu, ho budeme porovnávať s druhou páskou, aby sme sa uistili, že sa zhodujú. Ak áno, tak vstup akceptujeme.


1 ; 1 , R | □ ; □ , R
0 ; 0 , R | □ ; □ , R
1 ; 1 , R | □ ; 1 , R
0 ; 0 , R | □ ; 0 , R
0 ; 0 , R | □ ; 0 , R
1 ; 1 , R | □ ; 1 , R
0 ; 0 , R | □ ; 0 , R
1 ; 1 , R | □ ; 1 , R
1 ; 1 , R | □ ; □ , S
0 ; 0 , R | □ ; □ , S
q0
q1
q2
q3
0 ; 0 , S | □ ; □ , L
1 ; 1 , S | □ ; □ , L
1 ; 1 , S | 1 ; 1 , L
1 ; 1 , S | 0 ; 0 , L
0 ; 0 , S | 1 ; 1 , L
0 ; 0 , S | 0 ; 0 , L
q4
1 ; 1 , S | □ ; □ , R
0 ; 0 , S | □ ; □ , R
1 ; 1 , R | 1 ; □ , R
0 ; 0 , R | 0 ; □ , R
q5
□ ; □ , R | □ ; □ , R
1 ; 1 , S | □ ; □ , R
0 ; 0 , S | □ ; □ , R
q6

Vysvetlenie:

- Stav $q_0$: V stave $q_0$ čítame časť vstupu $w$. Keď sa nedeterministicky rozhodneme, že začína reťazec $x$, tak prečítame jeho "prvý symbol", uložíme ho na 2. pásku a prejdeme do $q_1$.
- Ďalej z $q_1$ prečítame druhý symbol $x$, uložíme ho na 2. pásku a prejdeme do $q_2$.
- Ďalej z $q_2$ prečítame tretí (posledný) symbol $x$, uložíme ho na 2. pásku a prejdeme do $q_3$.
- V stave $q_3$ máme teda na druhej páske 3 symboly reprezentujúce časť $x$.
- V stave $q_3$ ďalej v slučke čítame časť $y$ vstupu. Keď sa znovu nedeterministicky rozhodneme, že nasleduje opakovanie $x$, tak sa najprv presúvame do $q_4$, pričom tento prechod ignoruje prvú pásku (prechody sú na hocijaký symbol, pričom hlava ostáva stáť), akurát na druhej páske sa posúvame doľava.
- Rovnako v $q_4$ ignorujeme prvú pásku, avšak na druhej páske sa presúvame na jej začiatok.

Vysvetlenie(pokr.):

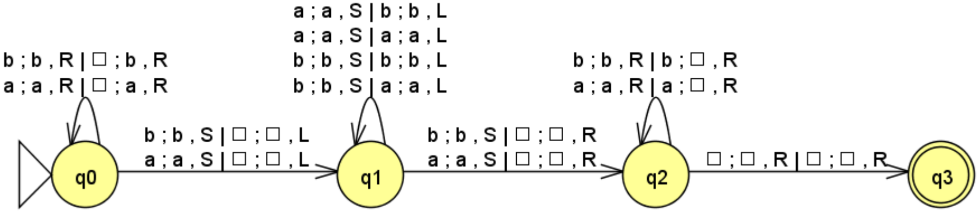
- Keď v stave $q_4$ na druhej páske nájdeme prázdnu bunku, vieme, že posun vpravo druhej pásky nás presunie na začiatok slova $x$ na druhej páske - vykonáme a prejdeme do stavu $q_5$.
- V stave $q_5$ porovnávame teraz 3 symboly na prvej páske s 3 symbolmi na druhej páske - ak sa nezhodujú, TS sa zastaví a neakceptuje vstup.
- Ak sa zhodujú, dôjde k tomu,že sa 2. páska vymaže a hlava sa postupne presúva na prázdnu bunku na 2. páske. V takom prípade sa úspešne podarilo rozpoznať $x$ v slove na prvej páske, t.j. v slove sa naozaj nejaký reťazec dĺžky 3 zopakoval.
- Vtedy prejdeme do stavu $q_6$ a akceptujeme vstup.

# Jazyk $L_5$

Nájdite TS akceptujúci jazyk $L_5 = \{ww \mid w \in \{a, b\}^*\}$ .

Tento jazyk tvoria reťazce, ktoré sú tvorené 2 rovnakými podreťazcami $w$. Pre tento jazyk sme na minulom cvičení našli TS s jednou páskou. Rovnako by sme ho však vedeli akceptovať aj nedeterministickým TS s 2 páskami podľa nasledovnej logiky:

- Čítame vstup a postupne si prečítané vstupné symboly ukladáme na 2. pásku.
- Vo vhodnom momente sa nedeterministicky rozhodneme, že sme prečítali prvú polovicu vstupu, presunieme hlavu 2. pásky na začiatok slova na nej napísanej a začneme kontrolovať, či sa zhodujú obsah 2. pásky a ešte neprečítané symboly prvej pásky.
- Ak áno, tak potom je na vstupe naozaj slovo v tvare $ww$.


b ; b , R | □ ; b , R
a ; a , R | □ ; a , R
a ; a , S | b ; b , L
a ; a , S | a ; a , L
b ; b , S | b ; b , L
b ; b , S | a ; a , L
b ; b , R | b ; □ , R
a ; a , R | a ; □ , R
b ; b , S | □ ; □ , L
a ; a , S | □ ; □ , L
b ; b , S | □ ; □ , R
a ; a , S | □ ; □ , R
□ ; □ , R | □ ; □ , R
q0
q1
q2
q3

Vysvetlenie:

- Stav $q_0$: V stave $q_0$ čítame vstup a ukladáme ho na druhú pásku. Vo vhodnom momente sa TS nedeterministicky rozhodne, že práve prečítal polovicu vstupu a prejde do stavu $q_1$. Zároveň posúva hlavu druhej pásky na posledný neprázdny symbol.
- V stave $q_1$ ignorujeme obsah prvej pásky - pričom jej hlava stojí na mieste - a posúvame hlavu druhej pásky na jej začiatok, resp. na prvú prázdnu bunku pred slovom na druhej páske. Keď ju nájdeme, posunieme hlavu vpravo, t.j. na prvý symbol druhej pásky a prejdeme do stavu $q_2$.
- V stave $q_2$ porovnávame obsah druhej pásky so slovom, ktoré sme ešte neprečítali na prvej páske. Ak sú rovnaké, tak sa dostaneme do situácie, že obe hlavy budú čítať prázdne bunky na oboch páskach. V takom prípade prejdeme do akceptačného stavu $q_3$.

Ďalšie úlohy možno nájsť v knihe [1] v sekcii 8.4.5.

# Kódovanie TS do binárneho reťazca

Uveďte binárny reťazec $w_i$ a Turingov stroj $M_i$ s uvedeným poradovým číslom:

- $i = 10$
- $i = 2708$
- $i = 89444642$
- $i = 85077332$

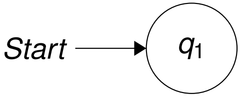
# $i = 10$

Uveďte binárny reťazec $w_{10}$ a Turingov stroj $M_{10}$. Binárny rozvoj čísla 10:

$$(10)_{10} = (1010)_2$$

Z binárneho rozvoja priamo vidieť, že $w_{10} = 010$

Keďže reťazec 010 nezodpovedá žiadnemu dekódovateľnému reťazcu na TS, tak $M_{10}$:



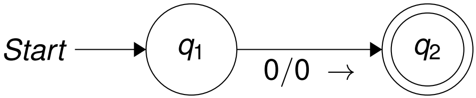
# $i = 2708$

Uveďte binárny reťazec $w_{2708}$ a Turingov stroj $M_{2708}$. Binárny rozvoj čísla 2708:

$$(2708)_{10} = (101010010100)_2$$

Z binárneho rozvoja priamo vidieť, že $w_{2708} = 01010010100$

Reťazec 01010010100 je dekódovateľný na prechod: $\delta(q_1, X_1, q_2, X_1, R)$. Ak budeme uvažovať z prednášky $X_1 = 0$, $X_2 = 1$, $X_3 = B$, $q_1$ je počiatočný, $q_2$ akceptačný, tak TS:




T.j. $L(M_{2708}) = \{0\}$

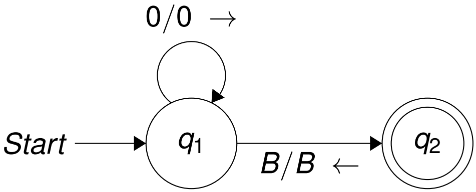
# $i = 89444642$

Uveďte binárny reťazec $w_{89444642}$ a Turingov stroj $M_{89444642}$. Binárny rozvoj čísla 89444642:

$$(89444642)_{10} = (101010101001101000100100010)_2$$

Z binárneho rozvoja priamo vidieť, že
$w_{89444642} = 01010101001101000100100010$

Reťazec 01010101001101000100100010 je dekódovateľný na 2 prechody:
$\delta(q_1, X_1, q_1, X_1, R)$, $\delta(q_1, X_3, q_2, X_3, L)$




T.j. $L(M_{89444642}) = 0^*$

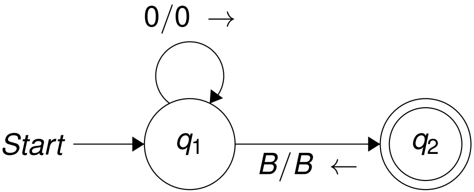
# $i = 85077332$

Uveďte binárny reťazec $w_{85077332}$ a Turingov stroj $M_{85077332}$. Binárny rozvoj čísla 89444642:

$$(85077332)_{10} = (101000100100010110101010100)_2$$

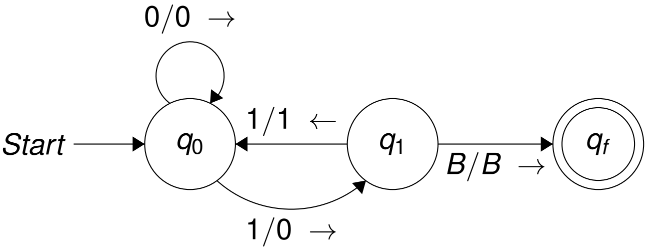
Z binárneho rozvoja priamo vidieť, že
$w_{85077332} = 01000100100010110101010100$

Reťazec 01000100100010110101010100 je dekódovateľný na 2 prechody:
$\delta(q_1, B, q_2, B, L)$, $\delta(q_1, X_1, q_1, X_1, R)$

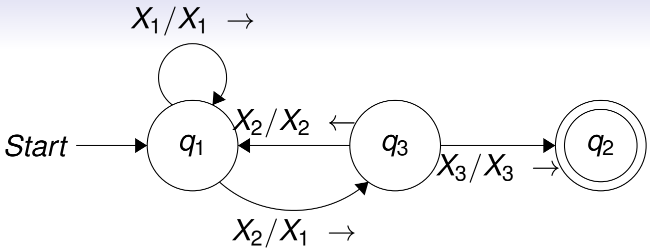



T.j. $L(M_{89444642}) = L(M_{85077332}) = 0^*$

# Zakódujte TS na obrázku




Najprv si prečíslujme stavy, aby bol počiatočný stav $q_1$ a akceptačný stav $q_2$. Páskové symboly budú očíslované: $X_1 = 0, X_2 = 1, X_3 = B$. Smery pohybu na páske budú označené $L = D_1, R = D_2$




Prechodová funkcia a jej zakódovanie do binárnych reťazcov:

- $\delta(q_1, X_1) = (q_1, X_1, D_2) = 0101010100$
- $\delta(q_1, X_2) = (q_3, X_1, D_2) = 0100100010100$
- $\delta(q_3, X_2) = (q_1, X_2, D_1) = 0001001010010$
- $\delta(q_3, X_3) = (q_2, X_3, D_2) = 0001000100100100$

Teda výsledný binárny reťazec:
010101010011010010001010011000100101001011000100010010001000100

Keď na začiatok pridáme jednotku, dostaneme poradové číslo:
76832588814450116

# Kódovanie TS a jeho vstupu do binárneho reťazca

Pre uvedený binárny reťazec zistite, či kóduje Turingov stroj so vstupom. Ak áno, pokúste sa určiť, či uvedený TS akceptuje daný reťazec.

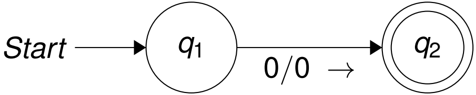
- 010
- 01010010100
- 010100101001110
- 01000100100010110101010100111000
- 01000100100010110101010100111100

Reťazec: 010

- Keďže kódovanie TS $M$ a vstupu $w$ by malo byť také, že reťazec vyzerá ako $M111w$, t.j. binárny reťazec predstavujúci TS $M$ by mal mať za sebou 3 jednotky, za ktorými ide vstup $w$, vidíme, že reťazec 010 nekóduje TS a vstup
- Maximálne "kóduje" preddefinovaný Turingov stroj s 1 stavom a bez prechodov - pretože samotný reťazec 010 nepredstavuje platný kód TS.

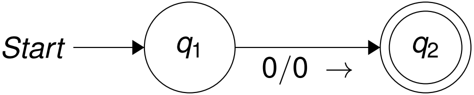
Reťazec: 01010010100

- Reťazec neobsahuje 111, to znamená, že znovu neobsahuje vstup.
- Obsahuje platný kód Turingovho stroja, ktorý vyzerá:



Reťazec: 010100101001110

- Reťazec obsahuje 111, to znamená, že časť reťazca pred 111, t.j. 01010010100 je kód Turingovho stroja.
- Časť reťazca za 111, t.j. 0 znamená, že uvažovaný vstup TS $w = 0$
- Turingov stroj:



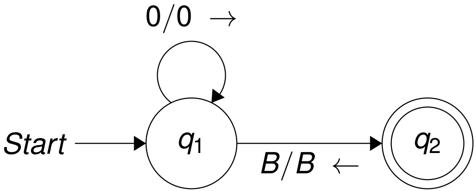

A výpočet pre $w = 0$:

$$q_1 0 \vdash 0 q_2 B$$

$w = 0$ daný TS akceptuje.

Reťazec: 0100010010001011010101010100111000

- Reťazec obsahuje 111, to znamená, že časť reťazca pred 111, t.j. 01000100100010110101010100 je kód Turingovho stroja.
- Časť reťazca za 111, t.j. 000 znamená, že uvažovaný vstup TS $w = 000$
- Turingov stroj:



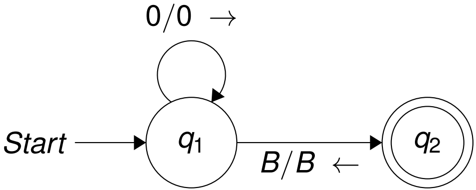

A výpočet pre $w = 000$:

$$q_1 000 \vdash 0 q_1 00 \vdash 00 q_1 0 \vdash 000 q_1 B \vdash 000 B q_2 B$$

$w = 000$ daný TS akceptuje.

Reťazec: 01000100100010110101010100111100

- Reťazec obsahuje 111, to znamená, že časť reťazca pred 111, t.j. 01000100100010110101010100 je kód Turingovho stroja.
- Časť reťazca za 111, t.j. 100 znamená, že uvažovaný vstup TS $w = 100$
- Turingov stroj:




A výpočet pre $w = 100$:

$$q_1 000 \vdash \text{ (zásek)}$$

$w = 100$ daný TS neakceptuje.

## Zopakovanie

- **Rekurzívne vyčísliteľný** jazyk je taký jazyk $L$, ku ktorému existuje TS $M$, ktorý ho akceptuje, $L = L(M)$. Ak $w \in L$, potom ho $M$ vie akceptovať. Ak $w \notin L$, potom $M$ **nemusí vedieť zastaviť počas jeho spracovania.**
- **Rekurzívny** jazyk je taký jazyk $L$, ku ktorému existuje TS $M$, ktorý ho akceptuje, $L = L(M)$ a navyše, $M$ zastaví **vždy** pri spracovaní každého $w$, bez ohľadu na to, či $w \in L$ alebo $w \notin L$.
- Je zrejmé, že Rekurzívne jazyky sú **podmnožinou** rekurzívne vyčísliteľných.

# Zopakovanie

- Jazyk $L_d = \{w_i \in \{0,1\}^* \mid w_i \notin L(M_i)\}$, t.j. jazyk tvorený binárnymi reprezentáciami tých Turingovych strojov, ktoré neakceptujú svoju binárnu reprezentáciu ako vstup, **nie je rekurzívne vyčísliteľný**, t.j. **neexistuje TS, ktorý by ho akceptoval**.
- Jazyk $L_u = \{w \in \{0,1\}^* \mid w$ predstavuje (M,x), kde $w \in L(M)\}$, t.j. jazyk tvorený binárnymi reprezentáciami Turingovho stroja $M$ a jeho vstupu $x$, ktorý akceptuje, t.j. $x \in L(M)$. Tento jazyk **je rekurzívne vyčísliteľný**, ale **nie je rekurzívny**, t.j. existuje TS, ktorý ho akceptuje, ale nemusí vždy zastaviť.
- Ak je jazyk $L$ rekurzívny, aj jeho doplnok $\overline{L}$ je rekurzívny.
- Ak sú aj jazyk $L$, aj jeho doplnok $\overline{L}$ rekurzívne vyčísliteľné, potom aj $L$, aj $\overline{L}$ sú rekurzívne.

# Zhrnutie

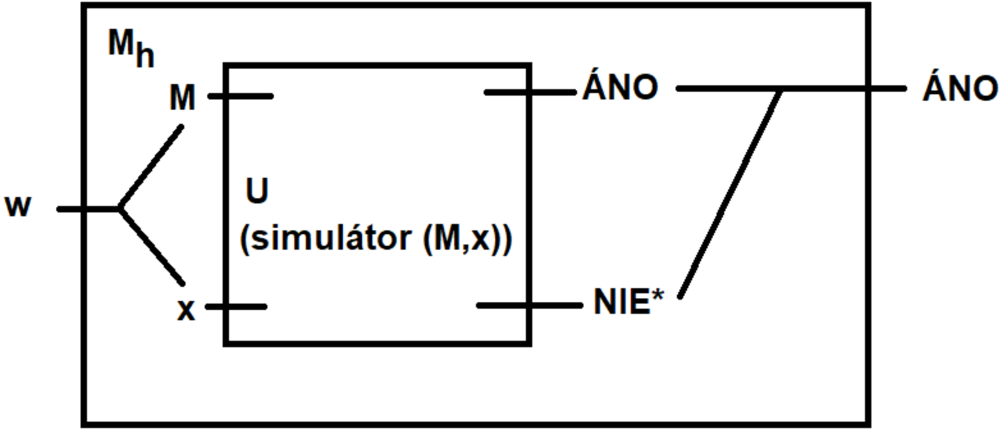
- Existuje tzv. univerzálny Turingov stroj $U$, ktorý dokáže simulovať činnosť ľubovoľného Turingovho stroja $M$ na základe jeho kódu, pre ľubovoľný vstup $x$. Tento univerzálny Turingov stroj akceptuje také kombinácie $(M, x)$, pre ktoré $M$ akceptuje $x$, t.j. $x \in M$.
- Čiže univerzálny Turingov stroj $U$ akceptuje práve jazyk $L_u$.
- Teda vždy vieme uvažovať, že máme k dispozícii nejaký TS, ktorý dokáže simulovať iné TS.
- Jeho nevýhodou je, že ak $M$ nikdy nezastaví pre vstup $x$, tak ani UTS $U$ nezastaví.
- Preto univerzálny TS **nie je rozhodovač**, t.j. existujú vstupy, pre ktoré nemusí zastaviť.
- Na prednáške sme ukázali, že taký TS, ktorý by vždy zastavil, pre jazyk $L_u$ **neexistuje**.

# Dôkaz č. 1

Dokážte, že jazyk
$L_h = \{w \in \{0,1\}^* \mid w$ predstavuje (M,x), kde $M$ zastaví pri spracovaní vstupu $x\}$
je rekurzívne vyčísliteľný, avšak nie rekurzívny, jazyk.

Dôkaz, že $L_h$ je rekurzívne vyčísliteľný, t.j. že existuje TS, ktorý ho akceptuje:

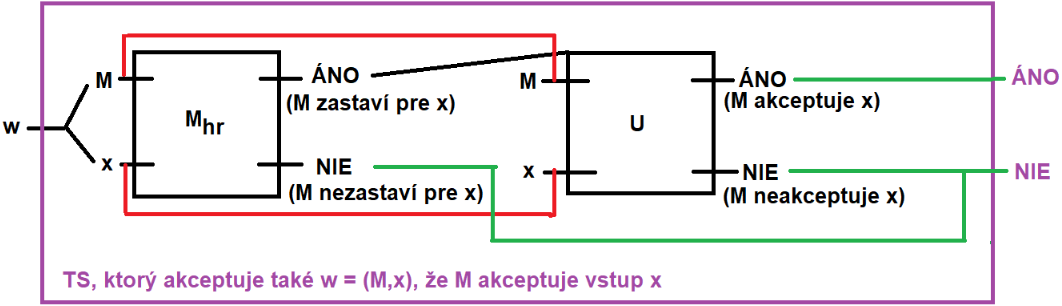
1. Nech $M_h$ je TS, ktorý vezme ako vstup $(M, x)$ a simuluje činnosť $M$ na slove $x$.
2. Ak sa $M$ zastaví a ocitne v akceptačnom stave, tak $M_h$ bude akceptovať vstup $(M, x)$.
3. Ak sa $M$ zastaví a ocitne v neakceptačnom stave, tak $M_h$ bude akceptovať vstup $(M, x)$.
4. Ak sa $M$ nezastaví, tak ani $M_h$ **nezastaví**.
5. Teda $M_h$ akceptuje **práve jazyk** $L_h$.
6. Zároveň však $M_h$ nemusí zastaviť - čiže sme zistili, že jazyk $L_h$ je rekurzívne vyčísliteľný.




(NIE* znamená, že simulátor *U* zistí, že *M* zastaví pre vstup *x* v neakceptačnom stave - táto situácia **nemusí nastať** pre všetky neakceptované *x* strojom *M*)

Dôkaz sporom, že $L_h$ nie je rekurzívny, t.j. že neexistuje TS, ktorý by ho akceptoval a zároveň by **vždy skončil pre všetky vstupy**.

1. Nech $L_h$ je rekurzívny. Potom musí existovať nejaký TS $M_{hr}$ taký, že akceptuje $L_h$ a skončí pre všetky možné vstupy. To znamená, že $M_{hr}$ dokáže pre ľubovoľný TS $M$ a jeho vstup $x$ povedať, či $M$ zastaví pri spracúvani $x$, alebo nie.
2. Pre ľubovoľný pár $(M, x)$ použijeme $M_{hr}$. Ak povie, že:
   - $M$ zastaví pre vstup $x$, tak zavoláme univerzálny TS $U$ a simulujeme $M$ na $x$. Keďže máme istotu, že $U$ zastaví, tak:
     - ak $U$ povie, že $M$ akceptuje $x$, tak potom celkovo povieme, že $M$ akceptuje $x$.
     - ak $U$ povie, že $M$ neakceptuje $x$, tak celkovo povieme, že $M$ neakceptuje $x$
   - $M$ nezastaví pre vstup $x$, tak celkovo povieme, že $M$ neakceptuje $x$.
3. Práve sme našli TS, ktorý **vždy skončí** a povie, či $x \in L(M)$. Taký však **nemôže existovať**, lebo by akceptoval $L_u$, čo by znamenalo, že $L_u$ je rekurzívny jazyk - čo vieme, že nie je pravda. Preto predpoklad bol zlý a $L_h$ **nie je rekurzívny jazyk**.




Uvedený TS nemôže existovať, lebo by vedel vždy zastaviť a povedať, či $M$ akceptuje $x$, t.j. by dokazoval, že $L_u$ je rekurzívny jazyk - my však vieme, že to nie je pravda! Daný TS neexistuje, pretože nemôže existovať jeho komponent $M_{hr}$.

# Dôkaz č. 2

Nech $L_1, L_2, ..., L_k$ sú jazyky nad abecedou $\Sigma$ také, že:

- Pre všetky $i \neq j, L_i \cap L_j = \emptyset$, t.j. žiaden reťazec nepatrí súčasne do 2 rôznych jazykov $L_i, L_j$
- $L_1 \cup L_2 \cup ... \cup L_k = \Sigma^*$, t.j. ľubovoľný reťazec nad $\Sigma$ patrí do nejakého jazyka $L_i$
- Každý jazyk $L_1, L_2, ..., L_k$ je rekurzívne vyčísliteľný.

Dokážte, že každý jazyk je zároveň aj rekurzívny.

# Dôkaz č. 2

Treba si uvedomiť, čo sa od nás požaduje a čo máme zadané.

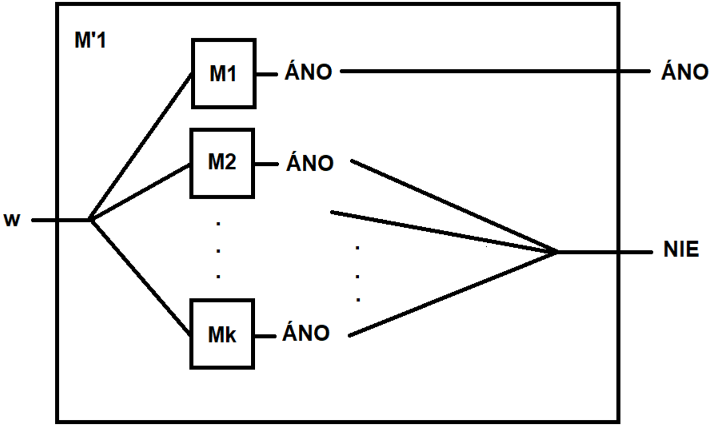
**Máme dané:**

- $k$ jazykov $L_1, L_2, ..., L_k$, ktoré tvoria **rozklad** množiny $\Sigma^*$, t.j. ľubovoľný reťazec $w \in \Sigma^*$ vždy patrí **práve do jedného** jazyka $L_i$.
- Navyše každý $L_i$ je RE, t.j. existuje Turingov stroj $M_i$, ktorý akceptuje $L_i$. Ak $w_i \in L_i$, potom ho $M_i$ akceptuje. Ak $w_i \notin L_i$, potom $M_i$ môže bežať nekonečne dlho.

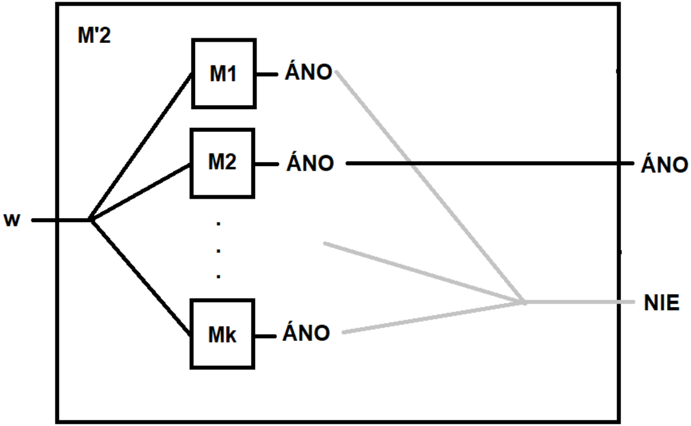
**Potrebujeme dokázať:**

- Jazyky $L_i$ sú **rekurzívne** - t.j. existujú také Turingove stroje $M_i'$, ktoré nielen, že ich akceptujú, ale navyše **vždy zastavia**, t.j. aj v prípade, že $w \notin L_i$.
- Cieľom je teda dokázať, **ako zostrojím** Turingove stroje $M_i'$, ktoré **vždy zastavia**, z Turingových strojov $M_i$, ktoré **nie vždy zastavia**.

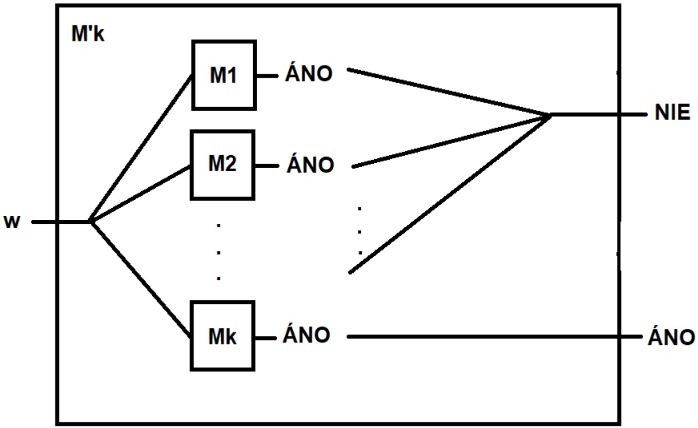
# TS $M_1'$



# TS $M_2'$



# TS $M'_k$



# TS $M_1'$

Z obrázkov je zrejmé, ako z Turingových strojov $M_1, M_2, ..., M_k$, ktoré vedia zastaviť len v prípade, že na vstupe majú reťazec, ktorý akceptujú, $w \in L(M_i)$ zostrojíme Turingov stroj, ktorý vie zastaviť vždy:

1. Pre jazyk $L_1$ zostrojíme TS taký, ktorý pošle svoj vstup $w$ na vstup všetkým Turingovym strojom $M_1, M_2, ..., M_k$. Jeden z týchto Turingovych strojov tento reťazec určite akceptuje, pretože zjednotenie ich jazykov predstavuje množinu všetkých reťazcov.
2. Ak $w$ akceptuje $M_1$, tak potom aj $M'1$ bude $w$ akceptovať a zastaví sa.
3. Ak $w$ akceptuje $M_2, M_3, ..., M_k$, tak potom $M'1$ akceptovať $w$ nebude a zastaví sa.
4. Jazyk $M'1$ je teda rovnaký, ako jazyk $M_1$ a navyše **vždy** skončí, pretože určite pre každý reťazec $w$ skončí **niektorý z Turingovych strojov** $M_1, M_2, ..., M_k$.

# TS $M_2'$

Analogicky zostrojíme TS $M_2'$:

1. Pre jazyk $L_2$ zostrojíme TS taký, ktorý pošle svoj vstup $w$ na vstup všetkým Turingovym strojom $M_1, M_2, ..., M_k$. Jeden z týchto Turingovych strojov tento reťazec určite akceptuje, pretože zjednotenie ich jazykov predstavuje množinu všetkých reťazcov.
2. Ak $w$ akceptuje $M_2$, tak potom aj $M'2$ bude $w$ akceptovať a zastaví sa.
3. Ak $w$ akceptuje $M_1, M_3, ..., M_k$, tak potom $M'2$ akceptovať $w$ nebude a zastaví sa.
4. Jazyk $M'2$ je teda rovnaký, ako jazyk $M_2$ a navyše **vždy** skončí, pretože určite pre každý reťazec $w$ skončí **niektorý z Turingovych strojov** $M_1, M_2, ..., M_k$.

Podobne pre ostatné jazyky, resp. Turingove stroje $M_i'$

Nielen teda, že vieme pre jazyky $L_1, L_2, ..., L_k$ zostrojiť **Turinove stroje**, ale vieme zostrojiť aj také Turingove stroje, ktoré **zastavia pre každý vstupný reťazec** - či už s akceptáciou, alebo s neakceptáciou.

Preto jazyky $L_1, L_2, ..., L_k$ sú nielen **rekurzívne vyčísliteľné**, ale aj **rekurzívne.**

# Dôkaz č. 3

Nech $L$ je rekurzívne vyčísliteľný jazyk a jeho doplnok $\overline{L}$ nie je rekurzívne vyčísliteľný jazyk. Uvažujte jazyk:

$$L' = \{0w \mid w \in L\} \cup \{1w \mid w \in \overline{L}\}$$

Vyberte správnu odpoveď:

- $L'$ je rekurzívny jazyk.
- $L'$ je rekurzívne vyčísliteľný jazyk, avšak nie je rekurzívny.
- $L'$ nie je rekurzívne vyčísliteľný jazyk.

Vo všeobecnosti, ak chceme dokázať, že jazyk $L$ je:

- *rekurzívny* - musíme nájsť TS $M$ taký, že $L(M) = L$ a zároveň $M$ vždy zastaví pre všetky možné vstupy
- *rekurzívne vyčísliteľný* musíme nájsť TS $M$ taký, že $L(M) = L$ (tu $M$ nemusí zastaviť pre tie vstupy, ktoré nie sú z jazyka $L$)

Vo všeobecnosti, ak chceme dokázať, že jazyk $L$ nie je:

- *rekurzívny* - musíme ukázať, že neexistuje TS $M$, ktorý $L(M) = L$ a vždy skončí - Štandardne sa to robí tak, že predpokladáme, že taký TS existuje a použijeme ho na konštrukciu iného TS, ktorý vždy zastaví a akceptuje jazyk, ktorý rekurzívny nie je - čiže k **niečomu, čo nemôže byť, t.j. k sporu.** Preto daný TS $M$ nemôže existovať a jazyk nie je rekurzívny.
- *rekurzívne vyčísliteľný* - musíme ukázať, že neexistuje TS $M$, ktorý $L(M) = L$. Štandardne sa to robí tak, že predpokladáme, že taký TS existuje a použijeme ho na konštrukciu iného TS, ktorý akceptuje jazyk, ktorý nie je RE - čiže k **niečomu, čo nemôže byť, t.j. k sporu.** Preto daný TS $M$ nemôže existovať a jazyk nie je rekurzívne vyčísliteľný.

Najprv dokážeme, že $L'$ nie je rekurzívny jazyk.
Dôkaz sporom:

- Nech $L'$ je rekurzívny jazyk.
- Potom existuje Turingov stroj $M'$ taký, že: $L(M') = L'$. Tento Turingov stroj navyše vždy zastaví a uvedie, či vstup patrí alebo nepatrí do $L'$.
- Ak by takýto Turingov stroj existoval, potom ho vieme použiť nasledovným spôsobom.

Najprv dokážeme, že $L'$ nie je rekurzívny jazyk.(pokr.)

- Vezmime reťazec $w$ a napíšme pred neho nulu, t.j. uvažujme reťazec $0w$.
- Vložme ho na vstup TS $M'$. Tento TS určite skončí a povie, či akceptuje $0w$ alebo neakceptuje $0w$.
- Ak ho akceptuje, to znamená, že $0w$ má tú vlastnosť, že $w$ patrí do jazyka $L$.
- Ak ho neakceptuje, to znamená, že $0w$ má tú vlastnosť, že $w$ nepatrí do jazyka $L$.
- Máme teda spôsob, ako o $w$ vieme **v konečnom čase rozhodnúť**, či patrí / nepatrí do $L$, teda $L$ je rekurzívny jazyk. Potom by však aj $\overline{L}$ musel byť rekurzívny, čo je v **spore so zadaním**!
- Preto taký $M'$ nemôže existovať a teda $L'$ **nie je rekurzívny.**

Teraz dokážeme, že $L'$ nie je ani rekurzívne vyčísliteľný jazyk.
Dôkaz sporom:

- Nech $L'$ je rekurzívne vyčísliteľný jazyk.
- Potom existuje Turingov stroj $M'$ taký, že: $L(M') = L'$. Tento Turingov stroj vie zastaviť pre reťazce z jazyka $L'$. Pre reťazce mimo tento jazyk môže pracovať donekonečna.
- Ak by takýto Turingov stroj existoval, potom ho vieme použiť nasledovným spôsobom.

Dôkaz, že $L'$ nie je rekurzívne vyčísliteľný jazyk (pokr.)

- Vezmime reťazec $w$ a napíšme pred neho jednotku, t.j. uvažujme reťazec $1w$.
- Vložme ho na vstup TS $M'$. Ak $1w$ patrí do $L'$, tak TS $M'$ vie skončiť. V takom prípade zároveň vieme, že $w \in \overline{L}$.
- Ak teda $M'$ akceptuje $1w$, tak vieme, že $w \in \overline{L}$.
- Ak $M'$ zastane, ale $1w$ neakceptuje, prípadne nezastane, v oboch prípadoch $1w$ neakceptuje a teda $w \notin \overline{L}$.
- Potom však dostávame spôsob, ktorý vie vrátiť odpoveď ÁNO v prípade, že $w \in \overline{L}$, t.j. pre jazyk $\overline{L}$ existuje TS, ktorý vie vrátiť odpoveď ÁNO, teda $\overline{L}$ je RE - , čo je v **spore so zadaním**!
- Preto $L'$ nemôže byť ani **rekurzívne vyčísliteľný.**

# Použitá literatúra

1 Hopcroft, Motwani, Ullman - Introduction to Automata Theory, Languages and Computations, 3rd Ed.