Turingovy stroje

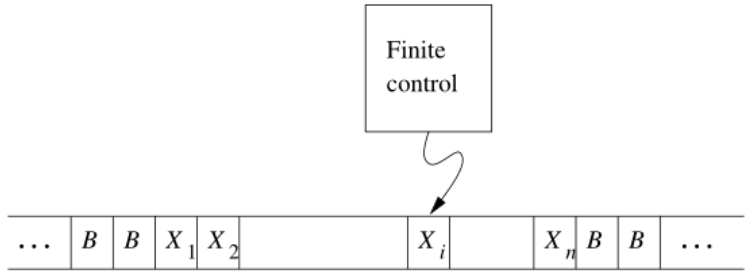
Marta Vomlelová

MS 303

Turingovy stroje – historie a motivace

- 1931–1936 pokusy o formalizaci pojmu algoritmu

  Gödel, Kleene, Church, Turing
- **Turingův stroj**
  - zachycení práce matematika
    - ⋆ nekonečná tabule
      lze z ní číst a lze na ni psát
    - ⋆ mozek (řídící jednotka)



  - Formalizace TM:
    - ⋆ místo tabule **nekonečná páska**
    - ⋆ místo křídy **čtecí a zapisovací hlava, kterou lze posunovat v obou směrech**
    - ⋆ místo mozku konečná řídící jednotka (jako u PDA)
  - další formalizace:
    - ⋆ částečně rekurzivní funkce, booleovské funkce, RAM, λ–kalkul,
- Snažíme se definovat problémy nerozhodnutelné jakýmkoli počítačem.

Definition 9.1

Turingův stroj (TM) je sedmice $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ se složkami:

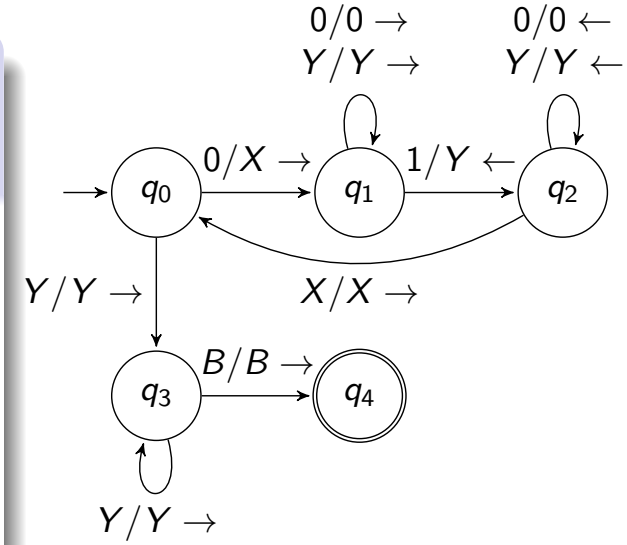
- Q konečná množina **stavů**
- Σ konečná neprázdná množina **vstupních symbolů**
- Γ konečná množina všech **symbolů pro pásku**. Vždy $\Gamma \supseteq \Sigma$, $Q \cap \Gamma = \emptyset$.
- δ (částečná) **přechodová funkce**. $(Q - F) \times \Gamma \to Q \times \Gamma \times \{L, R\}$, v $\delta(q, x) = (p, Y, D)$:
  - $q \in (Q - F)$**!** aktuální stav
  - $X \in \Gamma$ aktuální symbol na pásce
  - p nový stav, $p \in Q$.
  - $Y \in \Gamma$ symbol pro zapsání do aktuální buňky, přepíše aktuální obsah.
  - $D \in \{L, R\}$ je **směr** pohybu hlavy (doleva, doprava).
- $q_0 \in Q$ **počáteční stav**.
- $B \in \Gamma - \Sigma$. Blank. Symbol pro prázdné buňky, na začátku všude kromě konečného počtu buněk se vstupem.
- $F \subseteq Q$ množina **koncových** neboli **přijímajících** stavů.

Pozn: někdy se nerozlišuje Γ a Σ a neuvádí se prázdný symbol B, tj. pětice.

Přechodový diagram pro Turingův stroj

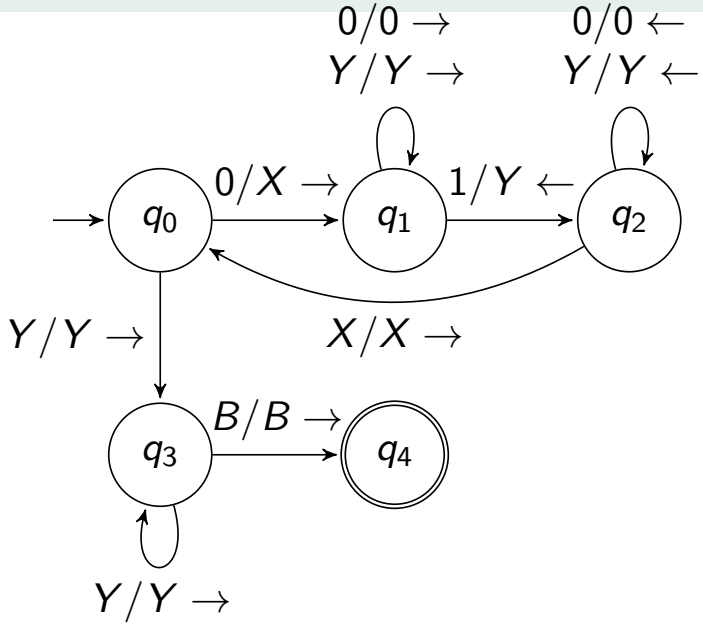
Definition 9.2 (Přechodový diagram pro TM)

Přechodový diagram pro TM sestává z uzlů odpovídajícím stavům TM. Hrany $q \to p$ jsou označeny seznamem všech dvojic X/YD, kde $\delta(q, X) = (p, Y, D)$, $D \in \{\leftarrow, \rightarrow\}$.
Pokud neuvedeme jinak, B značí blank – prázdný symbol.



State	0	1	X	Y	B
q_0	(q_1, X, R)	–	–	(q_3, Y, R)	–
q_1	$(q_1, 0, R)$	(q_2, Y, L)	–	(q_1, Y, R)	–
q_2	$(q_2, 0, L)$	–	(q_0, X, R)	(q_2, Y, L)	–
q_3	–	–	–	(q_3, Y, R)	(q_4, B, R)
q_4	–	–	–	–	–

TM pro $\{0^n1^n; n \geq 1\}$



Slovo 0011

$$q_0 0011 \vdash X q_1 011 \vdash X0 q_1 11 \vdash X q_2 0 Y 1 \vdash q_2 X0 Y 1 \vdash X q_0 0 Y 1 \vdash XX q_1 Y 1 \vdash$$

$$\vdash XXY q_1 1 \vdash XX q_2 YY \vdash X q_2 XYY \vdash XX q_0 YY \vdash XXY q_3 Y \vdash XXYY q_3 B \vdash XXYYB q_4 B$$

Slovo 0010

$$q_0 0010 \vdash X q_1 010 \vdash X0 q_1 10 \vdash X q_2 0 Y 0 \vdash q_2 X0 Y 0 \vdash X q_0 0 Y 0 \vdash XX q_1 Y 0 \vdash$$

$\vdash XXY q_1 0 \vdash XXY0 q_1 B$ a skončí neúspěchem, protože nemá instrukci.

Definition 9.3 (Konfigurace Turingova stroje)

Konfigurace Turingova stroje (Instantaneous Description ID) je řetězec $X_1X_2\ldots X_{i-1}qX_iX_{i+1}\ldots X_n$ kde

- q je stav Turingova stroje
- čtecí hlava je vlevo od i–tého symbolu
- $X_1\ldots X_n$ je část pásky mezi nejlevějším a nejpravějším symbolem různým od prázdného (B). S výjimkou v případě, že je hlava na kraji – pak na tom kraji vkládáme jeden B navíc.

Definition 9.4 (Krok Turingova stroje)

Kroky Turingova stroje M značíme $\vdash_M, \vdash_M^*, \vdash^*$ jako u zásobníkových automatů.

Pro $\delta(q, X_i) = (p, Y, R)$

- $X_1X_2\ldots X_{i-1}qX_iX_{i+1}\ldots X_n \vdash_M X_1X_2\ldots X_{i-1}\mathbf{Y}pX_{i+1}\ldots X_n$.

Pro $\delta(q, X_i) = (p, Y, L)$

- $X_1X_2\ldots X_{i-1}qX_iX_{i+1}\ldots X_n \vdash_M X_1X_2\ldots X_{i-2}pX_{i-1}\mathbf{Y}X_{i+1}\ldots X_n$

Odvození v konečném počtu kroků $\vdash_M^*$ definuji rekurzivně pro $\mathcal{I}, \mathcal{J}, \mathcal{K}$ konfigurace

Základ: $\mathcal{I} \vdash_M^* \mathcal{I}$ Rekurze: Pokud $\mathcal{I} \vdash_M \mathcal{J}$ a $\mathcal{J} \vdash_M^* \mathcal{K}$, tak i $\mathcal{I} \vdash_M^* \mathcal{K}$.

A TM for $\{0^n1^n; n \geq 1\}$

Definition 9.5 (TM přijímá jazyk, rekurzivně spočetný jazyk)

Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ **přijímá jazyk** $L(M) = \{w \in \Sigma^* : q_0 w \overset{*}{\underset{M}{\vdash}} \alpha p \beta, p \in F, \alpha, \beta \in \Gamma^*\}$, tj. množinu slov, po jejichž přečtení se dostane do koncového stavu. Pásku (u nás) nemusí uklízet.

Jazyk nazveme **rekurzivně spočetným**, pokud je přijímán nějakým Turingovým strojem T (tj. $L = L(T)$).

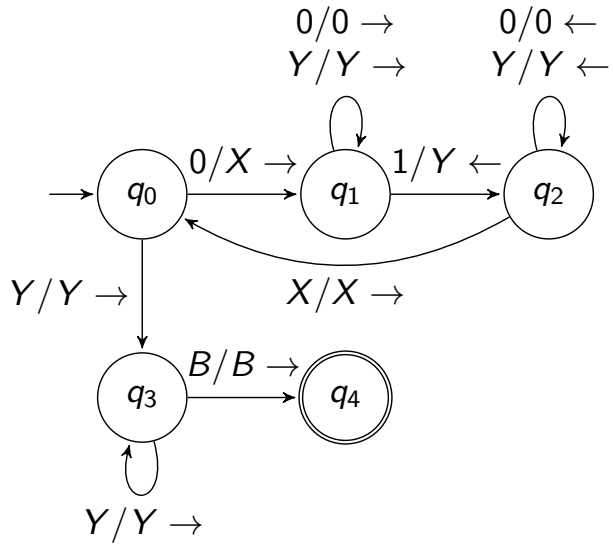
Example 9.1 (TM pro jazyk $\{0^n1^n; n \geq 1\}$)

Turingův stroj $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, X, Y, B\}, \delta, q_0, B, \{q_4\})$ s δ v tabulce přijímá jazyk $\{0^n1^n; n \geq 1\}$.

Stav	0	1	X	Y	B
q_0	(q_1, X, R)	–	–	(q_3, Y, R)	–
q_1	$(q_1, 0, R)$	(q_2, Y, L)	–	(q_1, Y, R)	–
q_2	$(q_2, 0, L)$	–	(q_0, X, R)	(q_2, Y, L)	–
q_3	–	–	–	(q_3, Y, R)	(q_4, B, R)
q_4	–	–	–	–	–

A TM for $\{0^n1^n; n \geq 1\}$

- Na pásce vždy výraz typu $\mathbf{X}^*\mathbf{0}^*\mathbf{Y}^*\mathbf{1}^*$
  - postupně přepisujeme 0 na X a odpovídající 1 na Y
  - q_0 přepíše 0 na X a předá řízení q_1
  - q_1 najde první 1, přepíše na Y a předá řízení q_2
  - q_2 se vrátí k X, nechá ho být a předá řízení q_0
  - ...
  - pokud q_0 vidí Y, předá řízení q_3
  - q_3 dojde zkontrolovat, jestli na konci nezbyly 1
  - pokud q_3 našlo B, předá řízení q_4
  - q_4 skončí úspěchem (je přijímající)
  - ...
  - pokud q_3 narazilo na 1, tak skončí neúspěchem
    - ★ nemá instrukci
    - ★ není přijímající.



Ještě příklad, rekurzivně spočetné jazyky

Example 9.2

Jazyk $L = \{a^{2^n} | n \geq 0\}$
přijímá Turingův stroj $M = (\{q_0, q_1, q_F\}, \{a\}, \{a, B\}, \delta, q_0, B, \{q_F\})$ s δ v tabulce:

δ	komentář
$\delta(q_0, B) = (q_F, B, R)$	prázdné slovo, konec výpočtu
$\delta(q_0, a) = (q_1, a, R)$	zvětší čítač ($2k + 1$ symbolů)
$\delta(q_1, a) = (q_0, a, R)$	nuluje čítač ($2k$ symbolů).

- Regulární jazyky:
  - simulujeme konečný automat, pohyb hlavy vždy vpravo,
  - vidím-li B, tj. konec vstupu. Pokud je stav DFA přijímající, přejdu do nového přijímajícího stavu q_F.
  - (Z q_F nesmí být instrukce, z přijímajících stavů DFA potřebuji instrukce kopírovat.)
- Bezkontextové jazyky: nejsnáze s pomocnou páskou simulující zásobník, bude za chvíli.

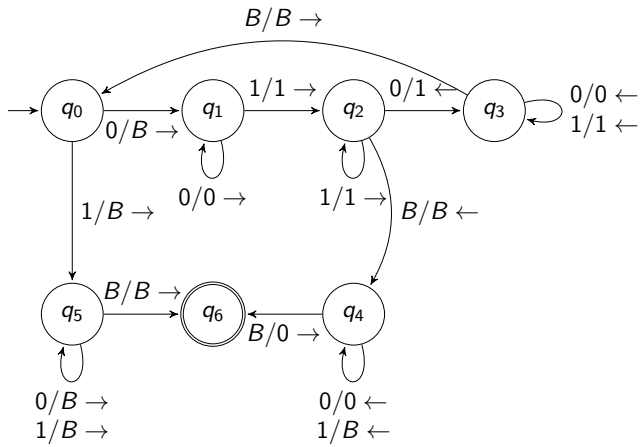
TM s výstupem

Example 9.3 (TM s výstupem)

Turingův stoj počítající **monus** $m \dot{-} n = max(m - n, 0)$.

- $M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B, \{q_6\})$
- Počáteční páska $0^m 1 0^n$.
- M zastaví s páskou $0^{m \dot{-} n}$ obklopenou prázdnem B.

- Najdi nejlevější 0, přepiš na B.
- Jdi doprava a najdi 1; pokračuj, najdi 0 a přepiš na 1.
- Vrať se doleva.
- Pokud nenajdeš 0 (ukliď):
  - vpravo: přepiš všechny 1 na B.
  - vlevo: $m < n$: přepiš všechny 1 a 0 na B, nech pásku prázdnou.



Rekurzivní jazyky

Definition 9.6 (TM zastaví)

TM **zastaví** pokud vstoupí do stavu q, s čteným symbolem X, a není instrukce pro tuto konfiguraci, t.j., $\delta(q, X)$ není definováno.

- Předpokládáme, že v přijímajícím stavu $q \in F$ TM zastaví,
- dokud nezastaví, nevíme, jestli přijme nebo nepřijme slovo.

Paralela:

- Místo 'zastavit a nepřijmout' můžeme spustit nekonečnou smyčku výpočtu.
- Pak otázka **náležení slova do jazyka** odpovídá otázce **zastavení Turingova stroje**

Definition 9.7 (Rekurzivní jazyky)

Říkáme, že TM M **rozhoduje jazyk** L, pokud $L = L(M)$ a pro každé $w \in \Sigma^*$ stroj nad w zastaví.
Jazyky rozhodnutelné TM nazýváme **rekurzivní jazyky**.

Paměť v řídící jednotce, Pomocná stopa

Example 9.4 (Příklad paměti ve stavu TM)

Pro jazyk $L(M) = (01^* + 10^*)$ si pamatujeme první znak, stav je dvojice (obecně n–tice). $M = (\{q_0, q_1\} \times \{0, 1, B\}, \{0, 1\}, \{0, 1, B\}, \delta, [q_0, B], B, \{[q_1, B]\})$

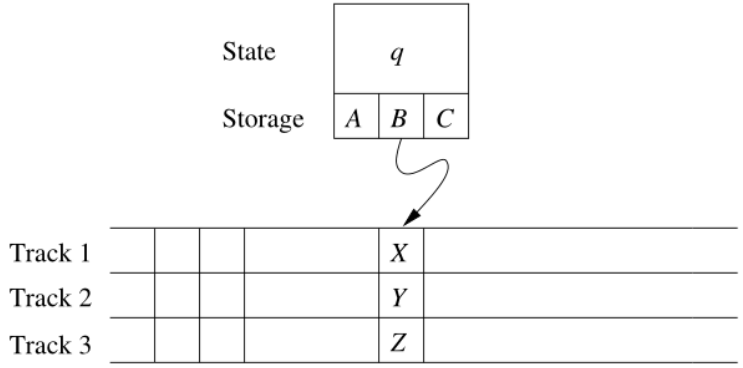
δ	0	1	B
$\rightarrow [q_0, B]$	$([q_1, 0], 0, R)$	$([q_1, 1], 1, R)$	
$[q_1, 0]$		$([q_1, 0], 1, R)$	$([q_1, B], B, R)$
$[q_1, 1]$	$([q_1, 1], 0, R)$		$([q_1, B], B, R)$
$*[q_1, B]$			

Example 9.5 (Příprava pomocné stopy)

Budeme potřebovat dvě stopy na pásce. Přepíšeme vstup na dvojice, nahoře budeme mít volno na poznámky. V definici δ používám zástupný znak pro vstupní písmeno $a \in \{0, 1\}$.

- $\delta([q_0, B], a) = ([q_0, B], [B, a], R)$
- $\delta([q_0, B], B) = ([q_{-1}, B], [B, B], L)$ jsem na konci, otáčím
- $\delta([q_{-1}, B], [B, a]) = ([q_{-1}, B], [B, a], L)$ jdi vlevo
- $\delta([q_{-1}, B], B) = ([q_1, B], B, R)$ dvě stopy připraveny pro další práci.

Více stop na pásce



- $L_{wcw} = \{wcw | w \in \{0,1\}^+\}$,
- $M = (\{q_1, \ldots, q_9\} \times \{0,1,B\}, \{[B,0],[B,1],[B,c]\}, \{B,*\} \times \{0,1,B,c\}, \delta, [q_1,B], [B,B], \{[q_9,B]\})$
- δ je definováno $(a, b \in \{0,1\})$:
  - $\delta([q_1,B],[B,a]) = ([q_2,a],[*,a],R)$ načti symbol a
  - $\delta([q_2,a],[B,b]) = ([q_2,a],[B,b],R)$ jdi vpravo, hledej střed c,
  - $\delta([q_2,a],[B,c]) = ([q_3,a],[B,c],R)$ pokračuj vpravo ve stavu q_3,
  - $\delta([q_3,a],[*,b]) = ([q_3,a],[*,b],R)$ pokračuj vpravo,
  - $\delta([q_3,a],[B,a]) = ([q_4,B],[*,a],L)$ zkontroluj shodu, vymaž paměť a jdi vlevo,
  - $\delta([q_4,B],[*,a]) = ([q_4,B],[*,a],L)$ jdi vlevo,
  - $\delta([q_4,B],[B,c]) = ([q_5,B],[B,c],L)$ c pokračuj za střed ve stavu q_5,
- rozeskok podle toho, jestli je ještě co kontrolovat
  - $\delta([q_5,B],[B,a]) = ([q_6,B],[B,a],L)$ ještě budeme kontolovat,
  - $\delta([q_6,B],[B,a]) = ([q_6,B],[B,a],L)$ jdi vlevo,
  - $\delta([q_6,B],[*,a]) = ([q_1,B],[*,a],R)$ znovu začni,
  - $\delta([q_5,B],[*,a]) = ([q_7,B],[*,a],R)$ už vše vlevo od c porovnáno, jdi vpravo,
  - $\delta([q_7,B],[B,c]) = ([q_8,B],[B,c],R)$ pokračuj vpravo,
  - $\delta([q_8,B],[*,a]) = ([q_8,B],[*,a],R)$ pokračuj vpravo,
  - $\delta([q_8,B],[B,B]) = ([q_9,B],[B,B],R)$ přijmi.

TM rozšíření: Vícepáskový TM

Definition 9.8 (Vícepáskový Turingův stroj)

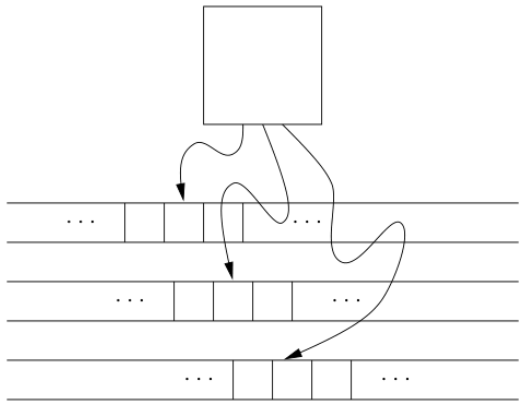
Počáteční pozice

- vstup na první pásce, ostatní zcela prázdné
- první hlava vlevo od vstupu, ostatní libovolně
- hlava v počátečním stavu

Jeden krok vícepáskového TM

- hlava přejde do nového stavu
- na každé pásce napíšeme nový symbol
- každá hlava se nezávisle posune vlevo, zůstane, vpravo.

Vícepáskový TM



Theorem 9.1 (Vícepáskový TM)

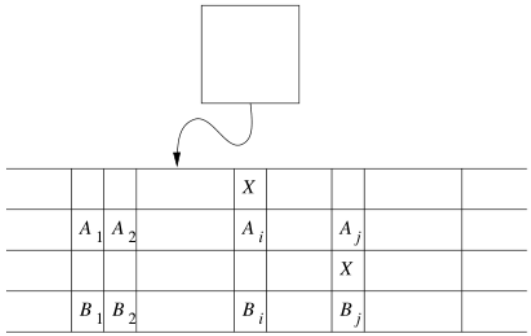
Každý jazyk přijímaný vícepáskovým TM je přijímaný i nějakým (jednopáskovým) Turingovým strojem TM.

Proof: vícepáskový TM

- konstruujeme Turingův stroj M
- pásku si představíme, že má $2k$ stop
  - liché stopy: pozice k–té hlavy
  - sudé stopy: znak na k–té pásce
- pro simulaci jednoho kroku navštívíme všechny hlavy
- ve stavu si pamatujeme
  - počet hlav vlevo
  - $\forall k$ symbol pod k–tou hlavou
- pak už umíme provést jeden krok (znovu běhat)

□

Simulace 2–páskového TM na jedné pásce



- Simulaci výpočtu k–páskového stroje o n krocích lze provést v čase $O(n^2)$ (simulace jednoho kroku z prvních n trvá $4n + 2k$, hlavy nejvýš $2n$ daleko, přečíst, zapsat, posunout značky).

Rozšíření: Nedeterministické Turingovy stroje

Definition 9.9 (Nedeterministický TM)

Nedeterministickým Turingovým strojem nazýváme sedmici $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, kde $Q, \Sigma, \Gamma, q_0, B, F$ jsou jako u TM a $\delta : (Q - F) \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$.
Slovo $w \in \Sigma^*$ **je přijímáno nedeterministickým TM** M, pokud existuje nějaký výpočet $q_0 w \vdash^* \alpha p \beta$, $p \in F$.

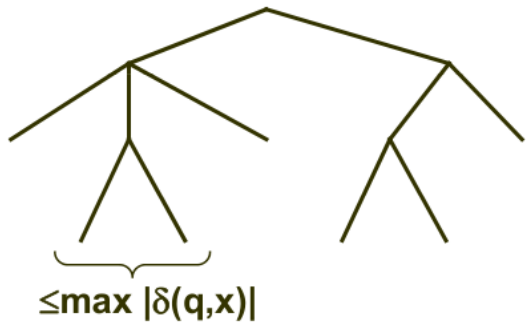
Theorem 9.2 (Nedeterministický TM)

Pro každý M_N nedeterministický TM existuje deterministický TM M_D takový, že $L(M_N) = L(M_D)$.

Velmi stručně (příprava)

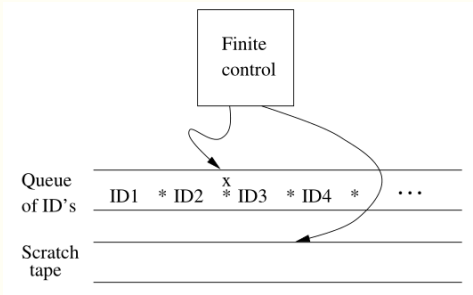
prohledáváme do šířky možné výpočty M_N

- odvozeno v k krocích
- maximálně m^k konfigurací
  - kde $m = max|\delta(q, x)|$ je max. počet voleb M_N



Proof: idea důkazu

- páska nekonečná – nelze použít podmnožinovou konstrukci
- prohledáváme do šířky všechny výpočty M_N
- modelujeme TM se dvěma páskami
  - první páska: posloupnost konfigurací
    - ★ aktuální označena (křížkem na obrázku)
    - ★ vlevo už prozkoumané, můžeme zapomenout
    - ★ vpravo aktuální a pak další čekající
  - druhá páska: pomocný výpočet



- zpracování jedné konfigurace obnáší
  - přečti stav a symbol aktuální konfigurace ID
  - je–li stav přijímající $\in F$, přijmi a skonči
  - napiš konfiguraci ID na pomocnou pásku
  - pro každý možný krok δ (uložený v hlavě M_D)
    - ★ proveď krok a napiš novou ID na konec první pásky
  - vrať se k označené ID, značku vymaž a posuň o 1 doprava
  - opakuj

□

Jednosměrná páska

X_0	X_1	X_2	...
*	X_{-1}	X_{-2}	...

Lemma (Jednosměrná páska)

Pro každý Turingův stroj M_2 existuje Turing. stroj M_1, který přijímá stejný jazyk a

- M_1 nikdy nepíše blank B,
- M_1 nikdy nejde vlevo od počáteční pozice.

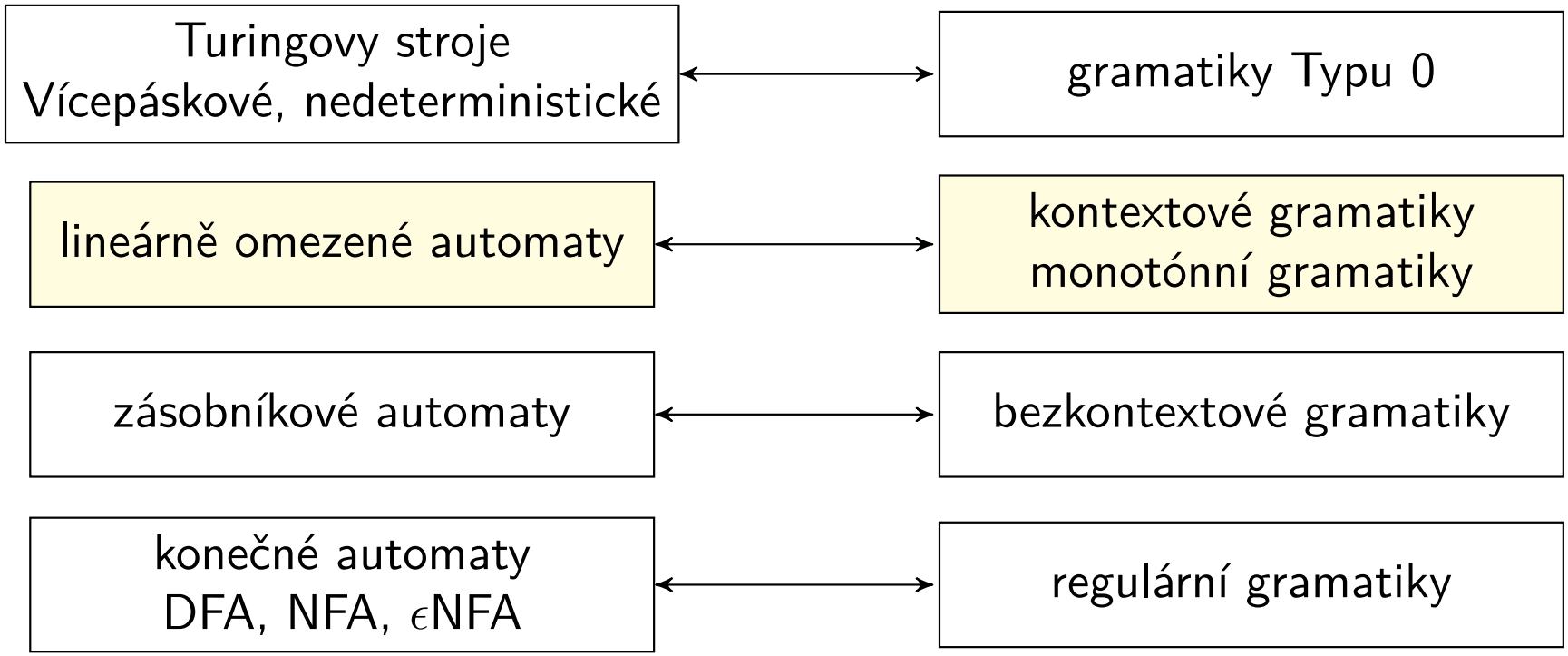
Proof.

- Místo B blank zavedeme nový páskový symbol, B_1.
  - Místo každého psaní B píšeme B_1 a všechny instrukce pro čtení B zkopírujeme též pro čtení B_1.
  - Takto modifikovaný TM nikdy nepíše B (píše B_1).
- Pro jednosměrnou pásku
  - Nejdřív přepíšeme vstup, aby byl v horní stopě dvoustopé pásky.
  - Pod nejlevější symbol dáme nový znak $*$, abychom věděli, že jsme na levém okraji, a máme přepnout z horní stopy do dolní. Ve stavu ('hlavě') si pamatujeme, jestli čteme horní stopu ('normálně') nebo spodní stopu (kde L znamená doprava a R doleva). Pokud vidíme $*$, odpovídající instrukce přepíšeme na změnu stopy (zhora vlevo přepnu na dolů, pokud jsem dole přešla na "*", mám rovnou číst horní symbol a směrovat dle horní pásky).

Shrnutí

- **Turingův stroj**: nekonečná oboustranná páska, může číst, psát, pohybovat hlavou.
- **Přijímání TM**: TM přijímá pokud vstoupí do koncového stavu.
  TM s výstupem: Hlava na prvním písmenu odpovědi, kromě odpovědi B.
- **Rekurzivně spočetné jazyky (RE)**: jazyky přijímané nějakým Turingovým strojem.
- **Konfigurace TM**: Všechny symboly pásky mezi nejlevějším a nejpravějším ne–B. Stav a pozice hlavy hned vlevo od právě čteného symbolu.
- modelovací triky
  - **Paměť v řídící jednotce**
  - **Více stop**
- Rozšíření TM bez rozšíření třídy přijímaných jazyků:
  - **Vícepáskové TM** Samostatný pohyb hlav na páskách (lze simulovat na přidaných stopách).
  - **Nedeterministický TM**: Má instrukce na výběr, na přijetí stačí jeden přijímající výpočet.
- Budou: **Lineárně omezené automaty LBA**
  - Vstupní slovo mezi levou a pravou značkou, hlava nesmí za tyto značky ani je přepsat.
  - LBA rozpoznávají právě kontextové jazyky.

Kontextové jazyky



- **gramatiky typu 1** (kontextové jazyky $\mathcal{L}_1$)
  - pouze pravidla ve tvaru $\alpha A \beta \to \alpha \omega \beta$
    $$A \in V, \alpha, \beta \in (V \cup T)^*, \omega \in (V \cup T)^+$$
  - jedinou výjimkou je pravidlo $S \to \epsilon$, potom se ale S nevyskytuje na pravé straně žádného pravidla

Chomského hierarchie

- **gramatiky typu 0** (rekurzivně spočetné jazyky $\mathcal{L}_0$)

  pravidla v obecné formě $\alpha \to \beta$, $\alpha, \beta \in (V \cup T)^*, \alpha$ obsahuje neterminál

gramatiky typu 1 (kontextové jazyky $\mathcal{L}_1$)

- pouze pravidla ve tvaru $\alpha A \beta \to \alpha \omega \beta$

  $A \in V, \alpha, \beta \in (V \cup T)^*, \omega \in (V \cup T)^+$
  - jedinou výjimkou je pravidlo $S \to \epsilon$, potom se ale S nevyskytuje na pravé straně žádného pravidla

- **gramatiky typu 2** (bezkontextové jazyky $\mathcal{L}_2$)

  pouze pravidla ve tvaru $A \to \omega, A \in V, \omega \in (V \cup T)^*$
- **gramatiky typu 3** (regulární/pravé lineární jazyky $\mathcal{L}_3$)

  pouze pravidla ve tvaru $A \to \omega B, A \to \omega, A, B \in V, \omega \in T^*$

Monotónní a kontextové gramatiky

Example 10.1 (kontextový jazyk)

$L = \{a^n b^n c^n | n \geq 1\}$ je kontextový jazyk, není bezkontextový.

$S \to aSBC | abC$

$CB \to BC$ není kontextové, nutno rozepsat!

Monotónní gramatika:

$bB \to bb$

$bC \to bc$

$cC \to cc$

- Je snažší napsat monotónní gramatiku

  a upravit ji na kontextovou.
- Vygeneruji stejný počet a, B, C.
- Přeuspořádám - proto potřebuji B, C neterminály.
- Dovolím přepsat jen v abecedním pořadí
  - Začínám s malými a s jedním b za nimi.
  - B se může přepsat, když je na řadě - vlevo od něj b.

  $\Rightarrow$ Především nedovolím přepsat B, pokud je před ním c.

Separované gramatiky

Definition 10.1 (Separovaná gramatika)

Gramatika je **separovaná**, pokud obsahuje pouze pravidla tvaru $\alpha \to \beta$, kde:

- buď $\alpha, \beta \in V^+$ (neprázdné posloupnosti neterminálů)
- nebo $\alpha \in V$ a $\beta \in T \cup \{\epsilon\}$.

Lemma

Ke každé gramatice G lze sestrojit ekvivalentní separovanou gramatiku G'.

Proof:

- Nechť $G = (V, T, P, S)$
- pro každý terminál $a \in T$ zavedeme nový neterminál A'.
- v pravidlech z P
  - nahradíme terminály odpovídajícími neterminály
  - přidáme pravidla $A' \to a$
- Výsledná gramatika je separovaná a zřejmě $L(G) = L(G')$. □

Od monotonie ke kontextovosti

Definition 10.2 (monotónní (nevypouštějící) gramatika)

Gramatika je **monotónní (nevypouštějící)**, jestliže pro každé pravidlo $(\alpha \to \beta) \in P$ platí $|\alpha| \leq |\beta|$. Monotónní gramatiky slovo v průběhu generování nezkracují.

Lemma

Ke každé monotónní gramatice lze nalézt ekvivalentní gramatiku kontextovou.

Proof:

- nejprve převedeme gramatiku na separovanou
  - tím se monotonie neporuší (a pravidla $A' \to a$ jsou kontextová)
- zbývající pravidla $A_1 \dots A_m \to B_1 \dots B_n$, $m \leq n$ převedeme na pravidla s novými neterminály C

$$
\begin{array}{llll}
A_1 A_2 \dots A_m & \to C_1 A_2 \dots A_m & C_1 C_2 \dots C_m & \to B_1 C_2 \dots C_m \\
C_1 A_2 \dots A_m & \to C_1 C_2 \dots A_m & B_1 C_2 \dots C_m & \to B_1 B_2 \dots C_m \\
\vdots & \vdots & \vdots & \vdots \\
C_1 \dots C_{m-1} A_m & \to C_1 \dots C_{m-1} C_m & B_1 \dots B_{m-1} C_m & \to B_1 \dots B_{m-1} B_m \dots B_n
\end{array}
$$

Příklad kontextového jazyka

Example 10.2

Jazyk $L = \{a^i b^j c^k | 1 \leq i \leq j \leq k\}$ je kontextový jazyk, není bezkontextový.

Proof: (na jedničku povinné)

$S \to aSBC \mid aBC$	generování symbolů a
$B \to BBC$	množení symbolů B
$C \to CC$	množení symbolů C
$CB \to BC$	uspořádání symbolů B a C
$aB \to ab$	začátek přepisu B na b
$bB \to bb$	pokračování přepisu B na b
$bC \to bc$	začátek přepisu C na c
$cC \to cc$	pokračování přepisu C na c

$CB \to BC$ není kontextové pravidlo, nahradíme ho

$CB \to XB$, $XB \to XY$, $XY \to BY$, $BY \to BC$

□

Dulezity je prepis, ktery chybi: $cB \to \times cb$.

Lineárně omezené automaty

- Ještě potřebujeme ekvivalent pro kontextové gramatiky
- kontextovou gramatiku dostaneme z libovolné monotónní gramatiky

Definition 10.3 (lineárně omezený automat (LBA))

Lineárně omezený automat LBA je nedeterministický TM, kde na pásce je označen levý a pravý konec $\underline{l}, \underline{r}$. Tyto symboly nelze při výpočtu přepsat a nesmí se jít nalevo od $\underline{l}$ a napravo od $\underline{r}$.
Slovo w **je přijímáno lineárně omezeným automatem**, pokud existuje přijímající výpočet $q_0\underline{l}w\underline{r} \vdash^* \underline{l}\alpha p\beta\underline{r}$, $p \in F$.

- Prostor výpočtu je definován vstupním slovem a automat při jeho přijímání nesmí překročit jeho délku
- u monotónních (kontextových) derivací to není problém – žádné slovo v derivaci není delší než vstupní slovo.

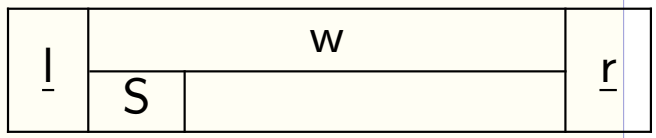
Od kontextových jazyků k LBA

Theorem 10.1

Každý kontextový jazyk lze přijímat pomocí LBA.

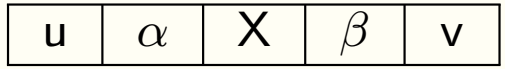
Proof: z kontextové gramatiky k LBA

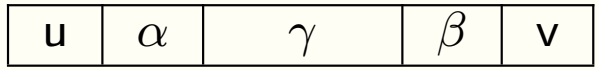
- prázdné slovo vyresíme zvlást
- derivaci gramatiky budeme simulovat pomocí LBA
- použijeme pásku se dvěma stopami
- slovo w dáme nahoru, na začátek dolní stopy S



Aplikace pravidla

$\alpha X \beta \to \alpha\gamma\beta$





- přepisujeme slovo ve druhé stopě podle pravidel G
  - nedeterministicky vybereme část k přepsání
  - provedeme přepsání dle pravidla (pravá část se odsune)
- pokud jsou ve druhé stopě samé terminály, porovnáme ji s první stopou
  - slovo přijmeme nebo zamítneme

□

Od LBA ke kontextovým jazykům

Theorem 10.2

LBA přijímají pouze kontextové jazyky.

Proof: z LBA ke kontextovým gramatikám

- potřebujeme převést LBA na monotónní gramatiku
  - tj. gramatika nesmí generovat nic navíc
- výpočet ukryjeme do 'dvoustopých' neterminálů
- generuj slovo ve tvaru $(a_0, [q_0, \underline{l}, a_0]), (a_1, a_1), \ldots, (a_n, [a_n, \underline{r}])$

w		
$q_0, \underline{l}, a_0$		$a_n, \underline{r}$

- simuluj práci LBA ve 'druhé' stopě (stejně jako u TM)
  - pro $\delta(p, x) = (q, x', R)$: $Px \to x'Q$
  - pro $\delta(p, x) = (q, x', L)$: $yPx \to Qyx'$
- pokud je stav koncový, smaž 'druhou' stop
- speciálně je třeba ošetřit přijímání prázdného slova
  - pokud LBA přijímá ϵ, přidáme speciální startovací pravidlo.

□

Mějme lineáně omezený automat pro jazyk $L = \{a^{2^n} | n \geq 1\}$, LBA $M = (\{q_0, q_1, q_2, q_F\}, \{a\}, \{a, \underline{l}, \underline{r}\}, \delta, q_0, B, \{q_F\})$ s δ v tabulce:

δ	komentář
$\delta(q_0, \underline{l}) = (q_0, \underline{l}, R)$	přeskočím prázdnou zarážku
$\delta(q_0, a) = (q_1, a, R)$	zvětší čítač ($2k+1$ symbolů)
$\delta(q_2, a) = (q_1, a, R)$	zvětší čítač ($2k+1$ symbolů)
$\delta(q_1, a) = (q_2, a, R)$	nuluje čítač ($2k$ symbolů)
$\delta(q_2, \underline{r}) = (q_F, \underline{r}, L)$	konec výpočtu, přijímám.

Example 10.3 (Gramatika z lineárně omezeného automatu)

Monotónní gramatika $G = (V, \{a\}, S, P_1 \cup P_2 \cup P_3)$

$$V = \left\{ \begin{array}{c} S, L, C, R, \begin{bmatrix} a \\ a \end{bmatrix}, \begin{bmatrix} a \\ q_0 \underline{l} a \end{bmatrix}, \begin{bmatrix} a \\ \underline{l} q_0 a \end{bmatrix}, \\ \begin{bmatrix} a \\ \underline{l} a \end{bmatrix}, \begin{bmatrix} a \\ q_2 a \end{bmatrix}, \begin{bmatrix} a \\ q_1 a \end{bmatrix}, \begin{bmatrix} a \\ a \underline{r} \end{bmatrix}, \begin{bmatrix} a \\ M_{\leftarrow} \end{bmatrix}, \\ \begin{bmatrix} a \\ q_1 a \underline{r} \end{bmatrix}, \begin{bmatrix} a \\ a q_2 \underline{r} \end{bmatrix}, \begin{bmatrix} a \\ \underline{l} q_0 a \underline{r} \end{bmatrix} \end{array} \right\} \quad P_1 = \left\{ \begin{array}{lcl} S & \to & LR | LCR \\ C & \to & CC | \begin{bmatrix} a \\ a \end{bmatrix} \\ L & \to & \begin{bmatrix} a \\ q_0 \underline{l} a \end{bmatrix} \\ R & \to & \begin{bmatrix} a \\ a \underline{r} \end{bmatrix} \end{array} \right\}$$

Pravidla pro přechodovou funkci

$$P_2 = \left\{ \begin{array}{ccc} \begin{bmatrix} a \\ q_0 \underline{l} a \end{bmatrix} & \to & \begin{bmatrix} a \\ \underline{l} q_0 a \end{bmatrix} \\ \begin{bmatrix} a \\ \underline{l} q_0 a \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} & \to & \begin{bmatrix} a \\ \underline{l} a \end{bmatrix} \begin{bmatrix} a \\ q_1 a \end{bmatrix} \\ \begin{bmatrix} a \\ q_2 a \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} & \to & \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} a \\ q_1 a \end{bmatrix} \\ \begin{bmatrix} a \\ q_1 a \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} & \to & \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} a \\ q_2 a \end{bmatrix} \\ \begin{bmatrix} a \\ q_2 a \end{bmatrix} \begin{bmatrix} a \\ a \underline{r} \end{bmatrix} & \to & \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} a \\ q_1 a \underline{r} \end{bmatrix} \\ \begin{bmatrix} a \\ q_1 a \underline{r} \end{bmatrix} & \to & \begin{bmatrix} a \\ a q_2 \underline{r} \end{bmatrix} \\ \begin{bmatrix} a \\ a q_2 \underline{r} \end{bmatrix} & \to & \begin{bmatrix} a \\ M_{\leftarrow} \end{bmatrix} \end{array} \right\}$$

Pravidla mazání

$$P_3 = \left\{ \begin{array}{ccc} \begin{bmatrix} a \\ a \end{bmatrix} \begin{bmatrix} a \\ M_{\leftarrow} \end{bmatrix} & \to & \begin{bmatrix} a \\ M_{\leftarrow} \end{bmatrix} a \\ \begin{bmatrix} a \\ \underline{l} a \end{bmatrix} \begin{bmatrix} a \\ M_{\leftarrow} \end{bmatrix} & \to & aa \end{array} \right\}$$

Obecněji

- Při více písmenech bychom měli varianty (téměř) všech neterminálů pro každé písmeno.
- V pravidlech P_1 jen totéž písmeno nahoře i dole.
- Pokud skončíme uvnitř slova, musíme mazat do obou stran až k zarážkám.

Theorem 10.3 (Rekurzivně spočetné jsou $\mathcal{L}_0$)

Každý rekurzivně spočetný jazyk je typu 0.

Proof: Od Turingova stroje ke gramatice

pro Turingův stroj T najdeme gramatiku G, $L(T) = L(G)$

- $G = (\{S, C, D, E\} \cup \{\underline{X}\}_{x \in \Sigma \cup \Gamma} \cup \{Q_i\}_{q_i \in Q}, \Sigma, P, S)$, P je:
- gramatika nejdříve vygeneruje pásku stroje a kopii slova

  $wB^n\underline{W}^R Q_0 B^m$, kde B^i představují volný prostor pro výpočet
- potom simuluje výpočet (stavy jsou součástí slova)
- v koncovém stavu smažeme pásku, necháme pouze kopii slova

1)	$S \to DQ_0E$	
	$D \to xD\underline{X} \mid E$	generuje slovo a jeho revizní kopii pro výpočet
	$E \to BE \mid \epsilon$	generuje volný prostor pro výpočet
2)	$\underline{X}P \to Q\underline{X'}$	pro $\delta(p, x) = (q, x', R)$
	$\underline{X}P\underline{Y} \to \underline{X'}YQ$	pro $\delta(p, x) = (q, x', L)$
3)	$P \to C$	pro $p \in F$
	$C\underline{A} \to C, \underline{A}C \to C$	mazání pásky
	$C \to \epsilon$	konec výpočtu

Příklad

Turingův stroj pro jazyk $L = \{a^{2n} | n \geq 0\}$, TM
$M = (\{q_0, q_1, q_2, q_F\}, \{a\}, \{a\}, \delta, q_0, B, \{q_F\})$ s δ v tabulce:

δ	komentář
$\delta(q_0, a) = (q_1, a, R)$	první symbol
$\delta(q_1, a) = (q_0, a, R)$	nuluje čítač ($2k$ symbolů)
$\delta(q_0, B) = (q_F, B, L)$	přijme a zastaví

Example 10.4

Gramatika $G = (\{S, C, D, E, Q_0, Q_1, Q_F, \underline{a}\}, \{a\}, S, P_1 \cup P_2 \cup P_3)$,

Inicializace

$$P_1 = \left\{ \begin{array}{lcl} S & \to & DQ_0E \\ D & \to & aD\underline{a} | E \\ E & \to & BE | \epsilon \end{array} \right\}$$

Přechodová funkce

$$P_2 = \left\{ \begin{array}{lcl} \underline{a}Q_0 & \to & Q_1\underline{a} \\ \underline{a}Q_1 & \to & Q_0\underline{a} \\ BQ_0\underline{a} & \to & B\underline{a}Q_F \end{array} \right\}$$

Mazání

$$P_3 = \left\{ \begin{array}{lcl} Q_F & \to & C \\ C\underline{a} & \to & C \\ \underline{a}C & \to & C \\ BC & \to & C \\ C & \to & \epsilon \end{array} \right\}.$$

Konkrétně pro $aa \in L(M)$ vygeneruji $aaB\underline{aa}Q_0$, mezivýsledek $aaB\underline{a}Q_F\underline{a}$ a výsledek aa.

Od Turingova stroje ke gramatice

Ještě $L(T) = L(G)$?

- $w \in L(T)$
  - existuje konečný výpočet stroje T (konečný prostor)
  - gramatika vygeneruje dostatečně velký prostor pro výpočet
  - simuluje výpočet a smaže dvojníky.
- $w \in L(G)$
  - pravidla v derivaci nemusí být v pořadí, jakém chceme
  - derivaci můžeme přeuspořádat tak, že pořadí je 1),2),3).
  - podtržené symboly smazány, tj. vygenerován koncový stav.

Example 10.5

Turingův stroj $M = (\{q_0, q_1, q_F\}, \{a\}, \{a, B\}, \delta, q_0, B, \{q_F\})$

$\delta(q_0, B) = (q_F, B, R)$

$\delta(q_0, a) = (q_1, a, R)$

$\delta(q_1, a) = (q_0, a, R)$

Gramatika po zjednodušení

$G = (\{S, C, D, E, \underline{a}, Q_0, Q_1\}, \{a\}, P, S)$

$S \to DQ_0$

$D \to aD\underline{a} | B$

$BQ_0 \to C$

$\underline{a}Q_0 \to Q_1\underline{a}$

$\underline{a}Q_1 \to Q_0\underline{a}$

$C\underline{a} \to C$

$C \to \epsilon$

Od gramatik k Turingově stroji

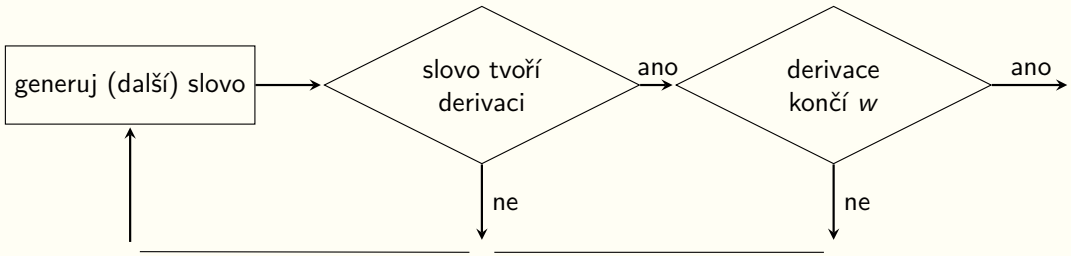
Theorem 10.4

Každý jazyk typu 0 je rekurzivně spočetný.

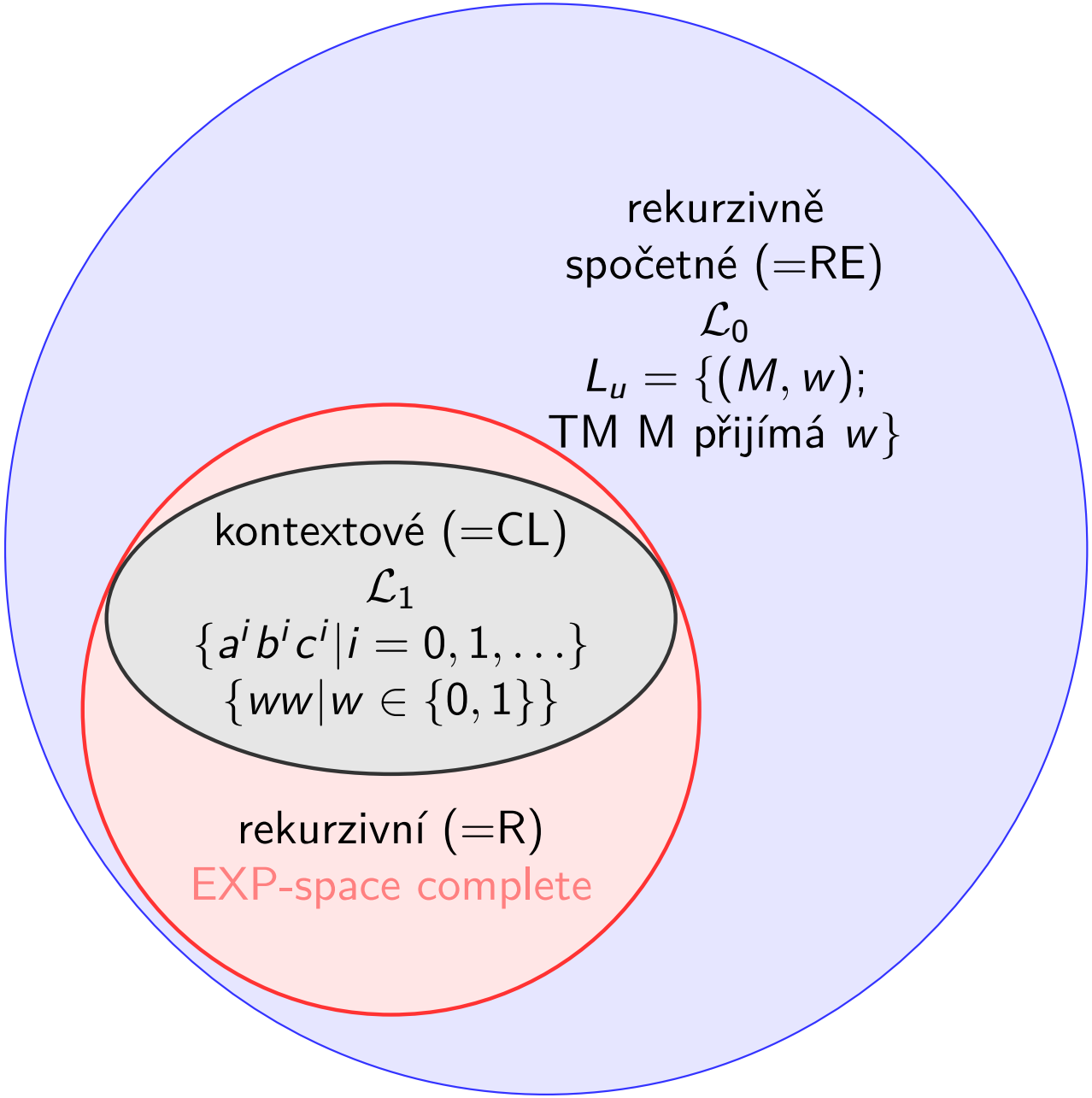
Proof:

idea: TM postupně generuje všechny derivace

- derivaci $S \Rightarrow \omega_1 \Rightarrow \ldots \Rightarrow \omega_n = w$ kódujeme jako slovo $\#S\#\omega_1\# \ldots \#w\#$
- umíme udělat TM, který přijímá slova $\#\alpha\#\beta\#$, kde $\alpha \Rightarrow \beta$
- umíme udělat TM, který přijímá slova $\#\omega_1\# \ldots \#\omega_k\#$, kde $\omega_1 \Rightarrow^* \omega_k$
- umíme udělat TM postupně generující všechna slova.



Hierarchie jazyků (kontextové a výš)



$L_d = \{w;$ TM s kódem w nepřijímá vstup $w\}$

Rekurzivní jazyky

Definition 10.4 (TM zastaví)

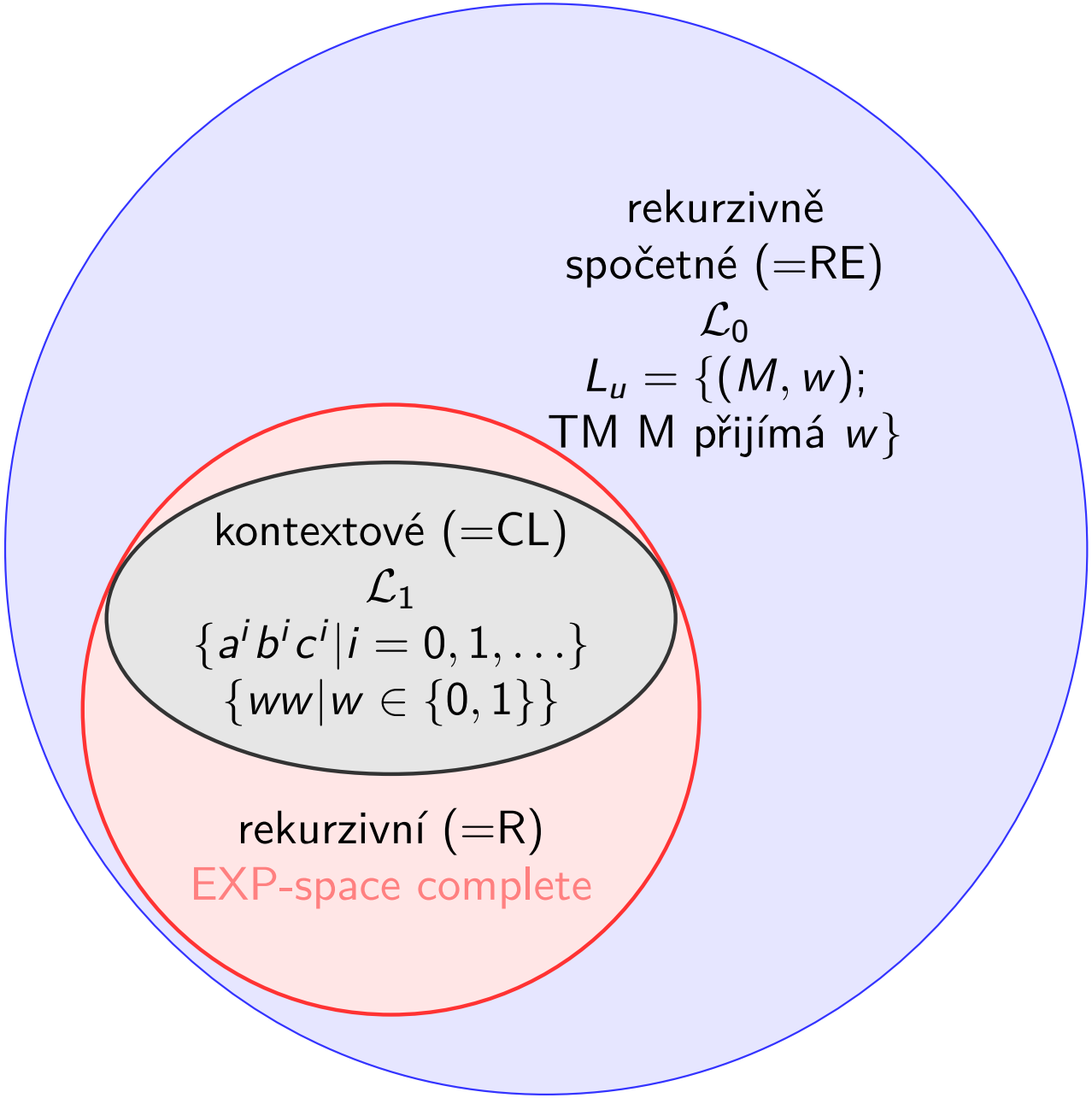
TM **zastaví** pokud vstoupí do stavu q, s čteným symbolem X, a není instrukce pro tuto konfiguraci, t.j., $\delta(q, X)$ není definováno.

- Z definice, v přijímajícím stavu $q \in F$ TM zastaví,
- dokud nezastaví, nevíme, jestli přijme nebo nepřijme slovo.

Definition 10.5 (Rekurzivní jazyky)

Říkáme, že TM M **rozhoduje jazyk** L, pokud $L = L(M)$ a pro každé $w \in \Sigma^*$ stroj nad w zastaví.
Jazyky rozhodnutelné TM nazýváme **rekurzivní jazyky**.

Hierarchie jazyků (kontextové a výš)



$L_d = \{w;$ TM s kódem w nepřijímá vstup $w\}$

Jazyk který není rekurzivně spočetný

Směřujeme k důkazu nerozhodnutelnosti jazyka dvojic (M, w) takových, že:

- M je binárně kódovaný Turingův stroj s abecedou $\{0, 1\}$,
- $w \in \{0, 1\}^*$ a
- M nepřijímá vstup w.

Postup:

- Kódování TM binárním kódem pro libovolný počet stavů TM.
- Kód TM vezmeme TM jako binární řetězec.
- Pokud kód nedává smysl, reprezentuje TM bez transakcí. Tedy každý kód reprezentuje nějaký TM.
- **Diagonální jazyk** L_d;
  $L_d = \{w; \text{TM reprezentovaný jako } w \text{ takový, že } \textbf{nepřijímá } w\}$.
- Neexistuje TM přijímající jazyk L_d. Spuštění takového stroje na vlastním kódu by vedlo k paradoxu.

Jazyk L_d není rekurzivně spočetný. Proto $\overline{L_d}$ není rekurzivní. Lze dokázat, že $\overline{L_d}$ je rekurzivně spočetný.

Kódování

- Pro kódování TM $M = (Q, \{0,1\}, \Gamma, \delta, q_1, B, \{q_2\})$ očíslujeme stavy, symboly a směry L, R.
- Předpokládejme:
  - Počáteční stav je vždy q_1.
  - Stav q_2 je vždy jediný koncový stav (nepotřebujeme víc, TM zastaví).
  - První symbol je vždy 0, druhý 1, třetí B, prázdný symbol. Ostatní symboly pásky očíslujeme libovolně.
  - Směr L je 1, směr R je 2.
- Jeden krok $\delta(q_i, X_j) = (q_k, X_l, D_m)$ kódujeme: $0^i 10^j 10^k 10^l 10^m$. Všechna $i, j, k, l, m \geq 1$ takže se dvě jedničky za sebou nevyskytují.
- Celý TM se skládá z kódů všech přechodů v nějakém pořadí oddělených dvojicemi jedniček 11: $C_1 11 C_2 11 \ldots C_{n-1} 11 C_n$.

Budeme potřebovat uspořádat řetězce do posloupnosti:

- Řetězce bereme uspořádané podle délky, stejně dlouhé uspořádáme lexikograficky.
- První je ϵ, druhý 0, třetí 1, čtvrtý 00 atd.
- i–tý řetězec označujeme w_i.

Příklad kódování TM

Turingův stroj

$M = (\{q_1, q_2, q_3\}, \{0, 1\}, \{0, 1, B\}, \delta, q_1, B, \{q_2\})$

δ	0	1	B
$\rightarrow q_1$		$(q_3, 0, R)$	
$*q_2$			
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	$(q_3, 1, L)$.

- Kód pro transakce:

C_1	C_2	C_3	C_4
0100100010100	0001010100100	00010010010100	000100010010010

- Kód celého TM:
  0100100010100110001010100100110001001001010011000100010010010.

Definition 10.6 (Diagonální jazyk)

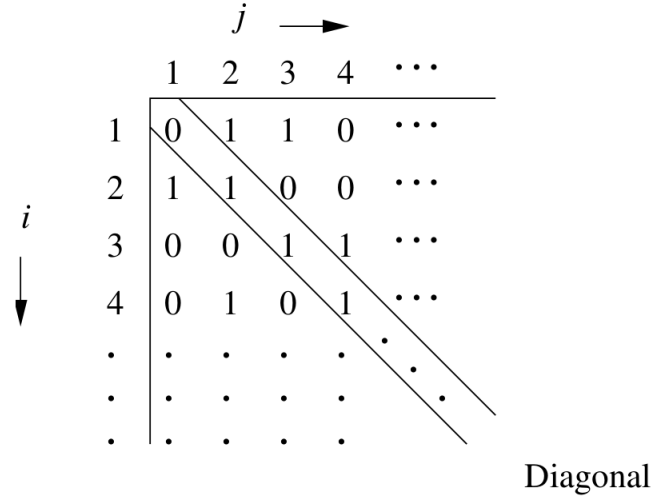
Diagonální jazyk L_d je definovaný
$L_d = \{w \in \{0,1\}^*; \text{TM reprezentovaný jako } w \text{ který nepřijímá slovo } w\}$.

$L_d = \{w; \text{na diagonále je } 0\}$.

Theorem 10.5

L_d není rekurzivně spočetný jazyk, tj. neexistuje TM přijímající L_d.

i - kód TM
j - vstup w
0/1 - prijima/neprijema



Proof.

- Předpokládejme L_d je RE, $L_d = L(M_d)$ pro nějaký TM M_d.
- Jeho jazyk je $\{0,1\}$, tedy je v seznamu na obrázku: 'Přijímá TM M_i vstupní slovo w_j?'
- Alespoň jeden řetězec ho kóduje, řekněme $code(M_d) = w_d$.
- Je $w_d \in L_d$
  - Pokud 'ano', na diagonále má být 0, tj. $w_d \notin L(M_d) = L_d$, spor.
  - Pokud 'ne', na diagonále má být 1, $w_d \in L(M_d) = L_d$, spor.

Proto takový M_d neexistuje. Tedy L_d není rekurzivně spočetný.

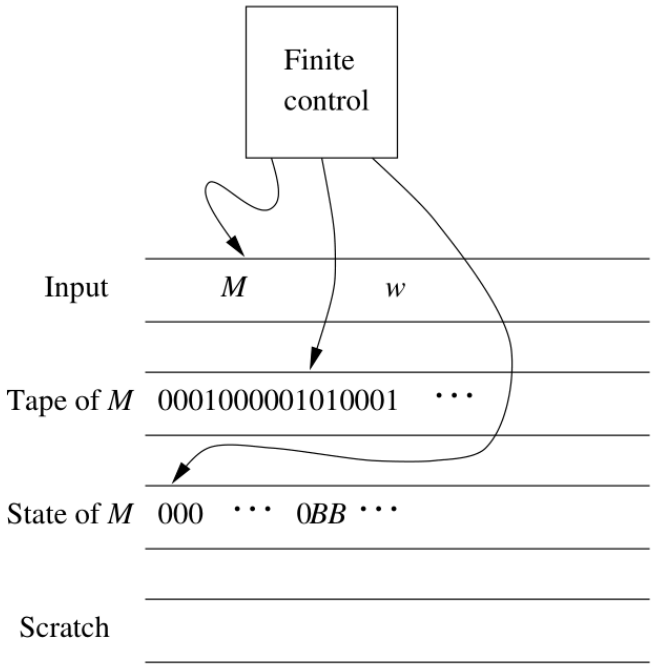
Univerzální Turingův stroj

Definition 10.7 (Univerzální jazyk)

Definujeme **univerzální jazyk** L_u jakožto množinu binárních řetězců které kódují pár (M, w), kde M je TM a $w \in L(M)$.
TM přijímající L_u se nazývá **Univerzální Turingův stroj**.

Theorem 10.6 (Existence Univerzálního Turingova stroje)

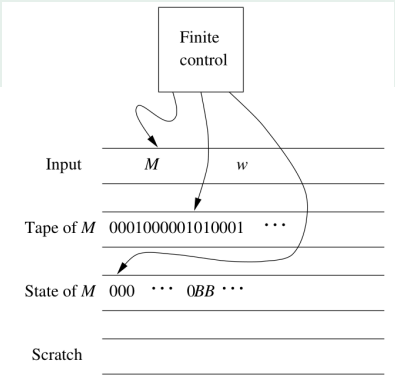
Existuje Turingův stroj U, pro který $L_u = L(U)$.



Popíšeme U jako vícepáskový Turingův stroj.

- Přechody M jsou napsány na první pásce spolu s řetězcem w.
- Na druhé pásce simulujeme výpočet M, používající formát jako kód M, tj. symboly 0^i oddělené jedničkou 1.
- Třetí páska obsahuje stav M reprezentovaný i nulami.

Operace univerzálního Turingova stroje



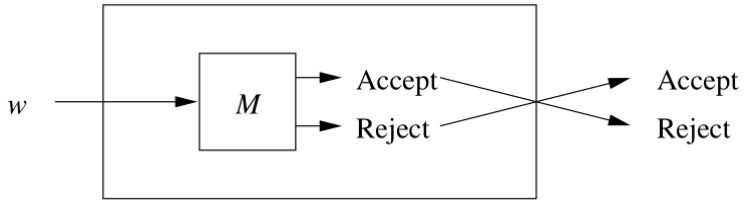
Operace U jsou následující:

- Otestuj, zda je kód M legitimní;
  pokud ne, U zastav bez přijetí.
- Inicializuj druhou pásku kódovaným slovem w: 10 pro 0 ve w, 100 pro 1; blank jsou nechané prázdné a nahrazeny 1000 pouze 'v případě potřeby'.
- Napiš 0, počáteční stav M, na třetí pásku. Posuň hlavu druhé pásky na první simulované políčko.
- Simuluj jednotlivé přechody M
  - Najdi na první pásce správnou transakci $0^i10^j10^k10^l10^m$, 0^i na pásce 3, 0^j na pásce 2.
  - Změň obsah pásky 3 na 0^k.
  - Nahraď 0^j na 2. pásce řetězcem 0^l. Použij čtvrtou 'scratch tape' pro správné mezery.
  - Posuň hlavu 2. pásky na pozici vedle 1 vlevo nebo vpravo, podle pohybu m.
- Pokud jsme nenašli instrukci pro M, zastavíme.
- Pokud M přejde do přijímajícího stavu, pak U také přijme. □

$L \& \overline{L} \in RE \Rightarrow L, \overline{L}$ je rekurzivní

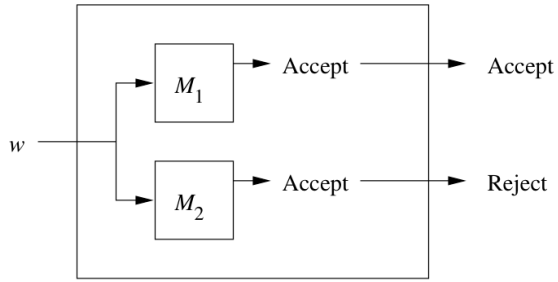
Lemma

Je–li L rekurzivní jazyk, je rekurzivní i $\bar{L}$.



Theorem 10.7 (Postova věta)

Jazyk L je rekurzivní, právě když L i $\overline{L}$ (doplněk) jsou rekurzivně spočetné.



Proof:

- Máme TM $L = L(M_1)$ a $\overline{L} = L(M_2)$.
- pro dané slovo w naráz simulujeme M_1 i M_2 (dvě pásky, stav se dvěma komponentami).
- Pokud jeden z M_i přijme, M zastaví a odpoví.
- Jazyky jsou komplementární, jeden z M_i vždy zastaví, L je rekurzivní. □

Nerozhodnutelnost univerzálního jazyka

Theorem 10.8 (Nerozhodnutelnost univerzálního jazyka)

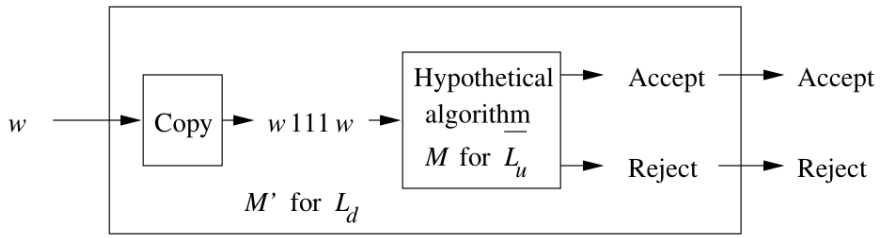
L_u je rekurzivně spočetný, ale není rekurzivní.

Proof.

- Máme TM přijímající L_u, tj. je RE.
- Předpokládejme, že je L_u rekurzivní.
- Pak $\overline{L_u}$ by byl také rekurzivní.
- Pro TM přijímající $\overline{L_u}$ můžeme zkonstruovat TM přijímající L_d (vpravo).
- Protože víme, že L_d není RE, $\overline{L_u}$ není RE a L_u není rekurzivní.

□

Modifikace TM pro $\overline{L_u}$ na TM pro L_d:



- Řetězec w přepiš na $w111w$ (2–páskový, převeď na 1–páskový).
- Simuluj M na novém vstupu. Přijmi iff M přijme.
- Stroj M' přijímá $\overline{L_u}(w, w)$, tj. případy kdy M_i nepřijímá w_i, tj. jazyk L_d.