

Univerzální TM, Diagonální jazyk

Marta Vomlelová

MS 303

Definice 8.1: 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.

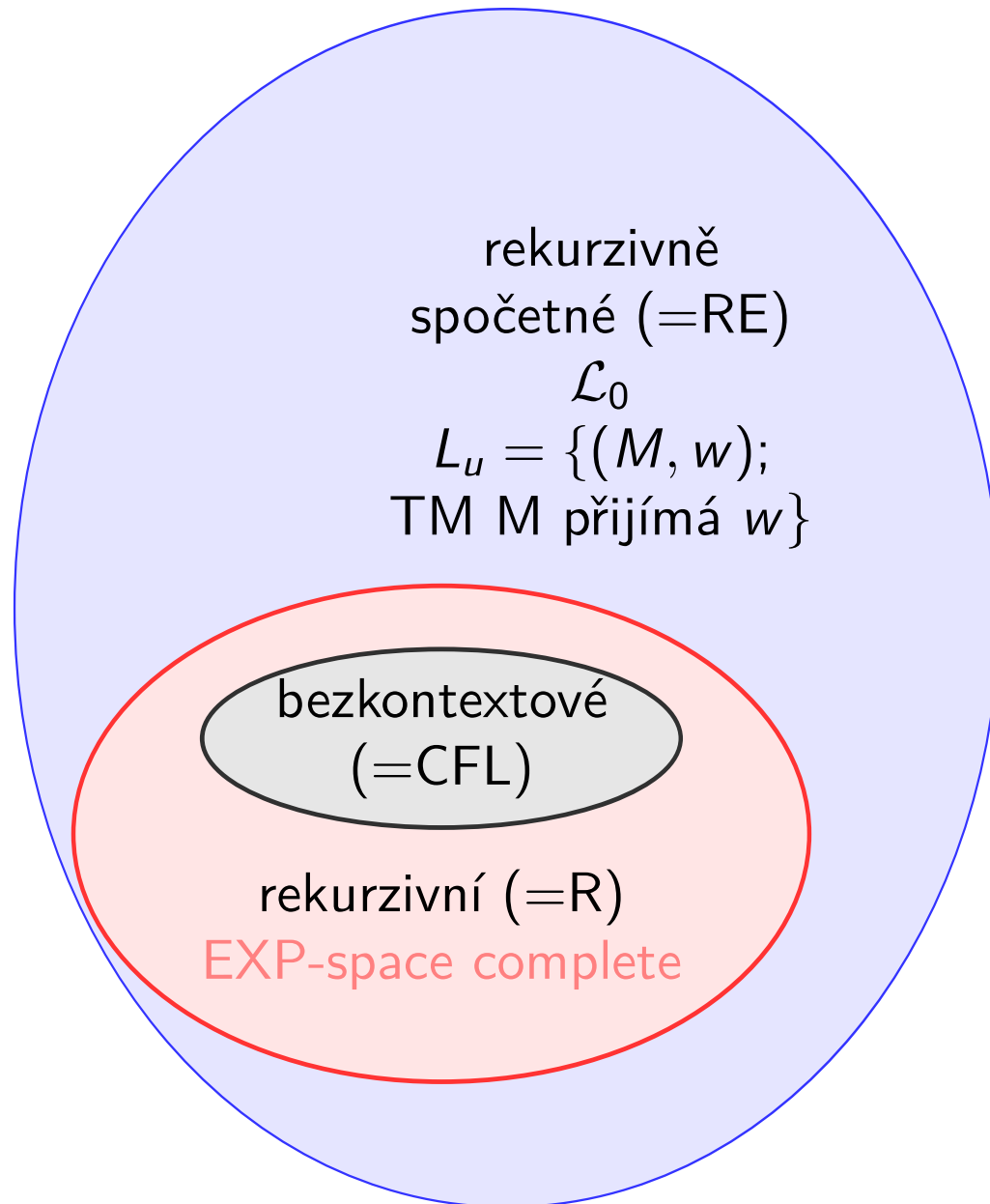
- Z definice, v přijímajícím stavu $q \in F$ TM zastaví,
- dokud nezastaví, nevíme, jestli přijme nebo nepřijme slovo.

Definice 8.2: 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)



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

Jazyk který není rekurzivně spočetný

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

- M je binárně kódovaný Turingův stroj s abecedou $\{0, 1\}$,
- $w = M \in \{0, 1\}^*$ 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 ;
 $L_d = \{w; \text{TM reprezentovaný jako } w \text{ takový, že } \mathbf{nepřijímá } w\}.$
- 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.

- Pro kódování TM $M = (Q, \{0, 1\}, \Gamma, \delta, q_1, B, \{q_2\})$ očíslováme 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 očíslováme 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: $C_1 11 C_2 11 \dots C_{n-1} 11 C_n$.

Příklad 8.1 (TM a jeho kódování)

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

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	0001000100010010

- Kód celého TM:

01001000101001100010101001001100010010010100110001000100010010.

Diagonální jazyk

Uspořádání řetězců (technický detail)

- Řetězce $\{0, 1\}^*$ bereme uspořádané podle délky, stejně dlouhé uspořádáme lexikograficky.
- První je ϵ , druhý 0, třetí 1, čtvrtý 00 atd.
- i -tý řetězec označujeme w_i .

Definice 8.3: Diagonální jazyk

Diagonální jazyk L_d je

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

Věta 8.1: L_d není RE

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

Na diagonále pro $w_d = \text{code}(M_d)$ spor

i - kód TM

j - vstup w

1/0 - přijímá/nepřijímá

		$j \rightarrow$				
		1	2	3	4	...
$i \downarrow$	1	0	1	1	0	...
	2	1	1	0	0	...
	3	0	0	1	1	...
	4	0	1	0	1	...
	...	...	...	...	...	...
	...	...	...	...	...	...
	...	...	...	...	...	...

Diagonal

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

Důkaz: L_d není RE.

- 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 $\text{code}(M_d) = w_d$.
- 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. $\square$

Definice 8.4: Univerzální jazyk

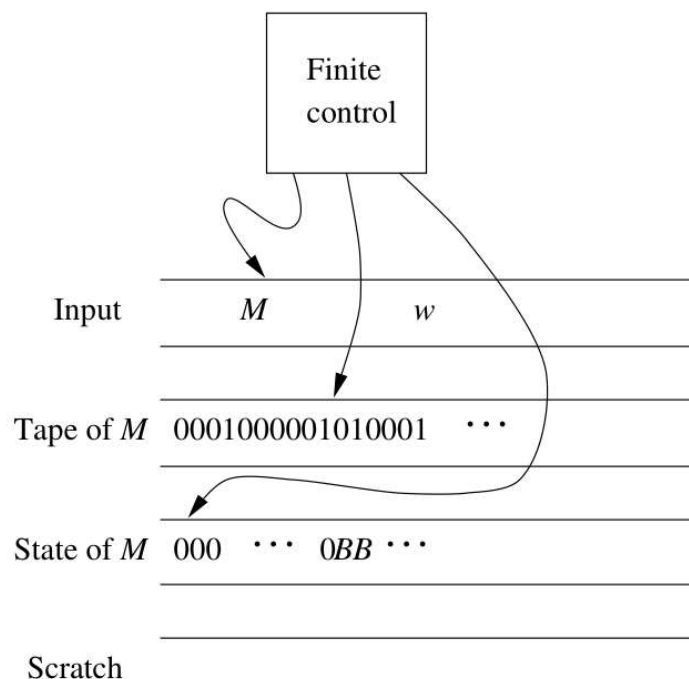
Definujeme **univerzální jazyk** L_u jakožto množinu binárních řetězců které kódují pár (M, w) , kde M je TM a $w \in L(M)$, tj. M přijímá slovo w .

$$L_u = \{(M, w) \mid M \text{ je TM a } w \in L(M)\}$$

TM přijímající L_u se nazývá **Univerzální Turingův stroj**.

Věta 8.2: 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. □

Tedy máme TM přijímající $L_u = \{(M, w) \mid w \in L(M)\}$.

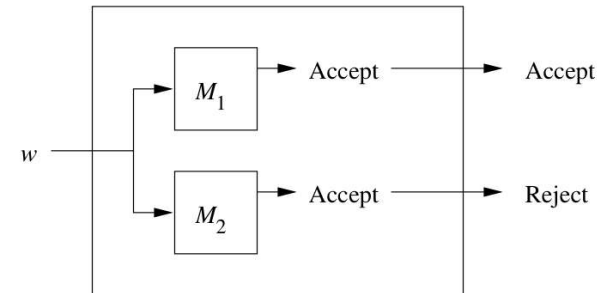
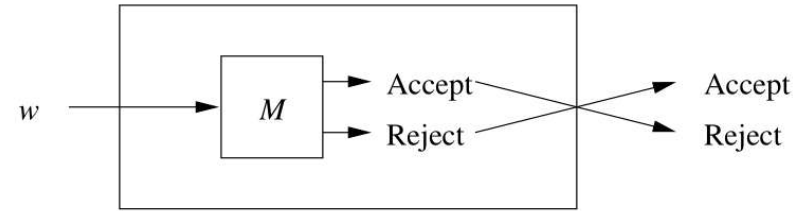
Postova věta: $(L \in RE \ \& \ \bar{L} \in RE) \Leftrightarrow L$ je rekurzivní

Lemma

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

Věta 8.3: Postova věta

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



Důkaz: Postova věta

$\Rightarrow$ Vezmeme TM M pro L , který vždy zastaví a vydá odpověď (existuje, protože L je rekurzivní). Pro test náležení do $\bar{L}$ předložíme vstup M , počkáme na odpověď a odpovíme opak.

$\Leftarrow$ ▶ Máme TM $L = L(M_1)$ a $\bar{L} = L(M_2)$.

- ▶ pro dané slovo w naráz simulujeme M_1 i M_2 (dvě pásy, 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

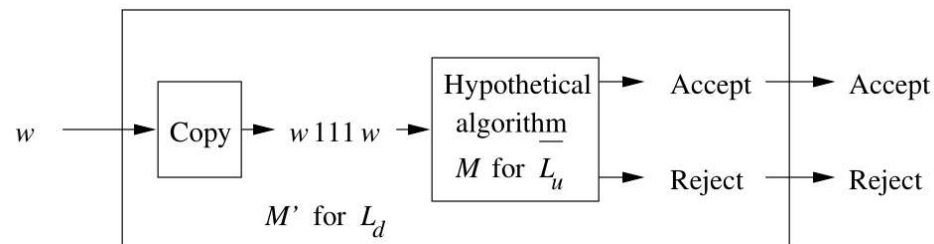
Věta 8.4: Nerozhodnutelnost univerzálního jazyka

L_u je rekurzivně spočetný, ale není rekurzivní.

Důkaz: Nerozhodnutelnost L_u

- 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. $\square$

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



- Řetězec w přepiš na $w111w$ (2-páskový, převed' 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 .

Věta 8.5: Rekurzivně spočetné jsou $\mathcal{L}_0$

Každý rekurzivně spočetný jazyk je typu 0.

Důkaz: 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 $wB^n\underline{W}^RQ_0B^m$, 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) $S \rightarrow DQ_0E$
 $D \rightarrow xDX|E$ generuje slovo a jeho revizní kopii pro výpočet
 $E \rightarrow BE|\epsilon$ generuje volný prostor pro výpočet
- 2) $\underline{X}P \rightarrow Q\underline{X}'$ pro $\delta(p, x) = (q, x', R)$
 $\underline{X}P\underline{Y} \rightarrow \underline{X}'YQ$ pro $\delta(p, x) = (q, x', L)$
- 3) $P \rightarrow C$ pro $p \in F$
 $C\underline{A} \rightarrow C, \underline{A}C \rightarrow C$ mazání pásky
 $C \rightarrow \epsilon$ konec výpočtu



Příklad 8.2 (Gramatika z TM)

Mějme 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

Pro tento TM zkonstruujeme gramatiku

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

Inicializace

Přechodová funkce

$$P_1 = \left\{ \begin{array}{lcl} S & \rightarrow & DQ_0E \\ D & \rightarrow & aD\underline{a}|E \\ E & \rightarrow & BE|\epsilon \end{array} \right\} \quad P_2 = \left\{ \begin{array}{lcl} \underline{a}Q_0 & \rightarrow & Q_1\underline{a} \\ \underline{a}Q_1 & \rightarrow & Q_0\underline{a} \\ BQ_0\underline{a} & \rightarrow & B\underline{a}Q_F \end{array} \right\} \quad P_3 = \left\{ \begin{array}{lcl} Q_F & \rightarrow & C \\ C\underline{a} & \rightarrow & C \\ \underline{a}C & \rightarrow & C \\ BC & \rightarrow & C \\ C & \rightarrow & \epsilon \end{array} \right\}.$$

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

Potřebujeme ověřit, zda $L(T) = L(G)$?

- $w \in L(T)$
 - ▶ 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.

Příklad 8.3 (Gramatika pro TM)

Mějme 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

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

$$P = \left\{ \begin{array}{l} S \rightarrow DQ_0 \\ D \rightarrow aD\underline{a}|B \\ BQ_0 \rightarrow C \\ \underline{a}Q_0 \rightarrow Q_1\underline{a} \\ \underline{a}Q_1 \rightarrow Q_0\underline{a} \\ C\underline{a} \rightarrow C \\ C \rightarrow \epsilon \end{array} \right\}.$$

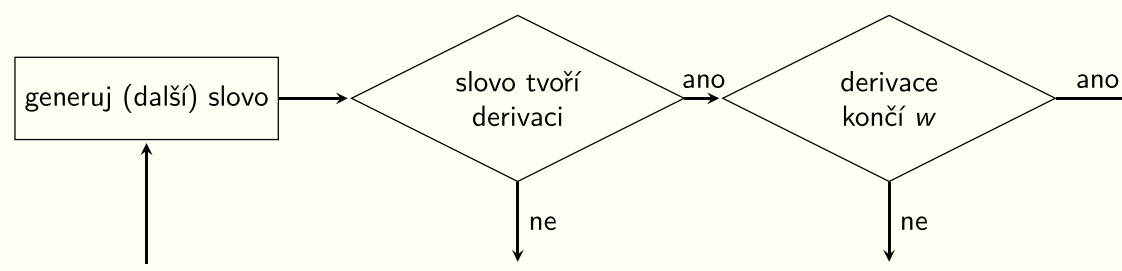
Věta 8.6: $\mathcal{L}_0$ jsou rekurzivně spočetné.

Každý jazyk generovaný gramatikou typu 0 je rekurzivně spočetný.

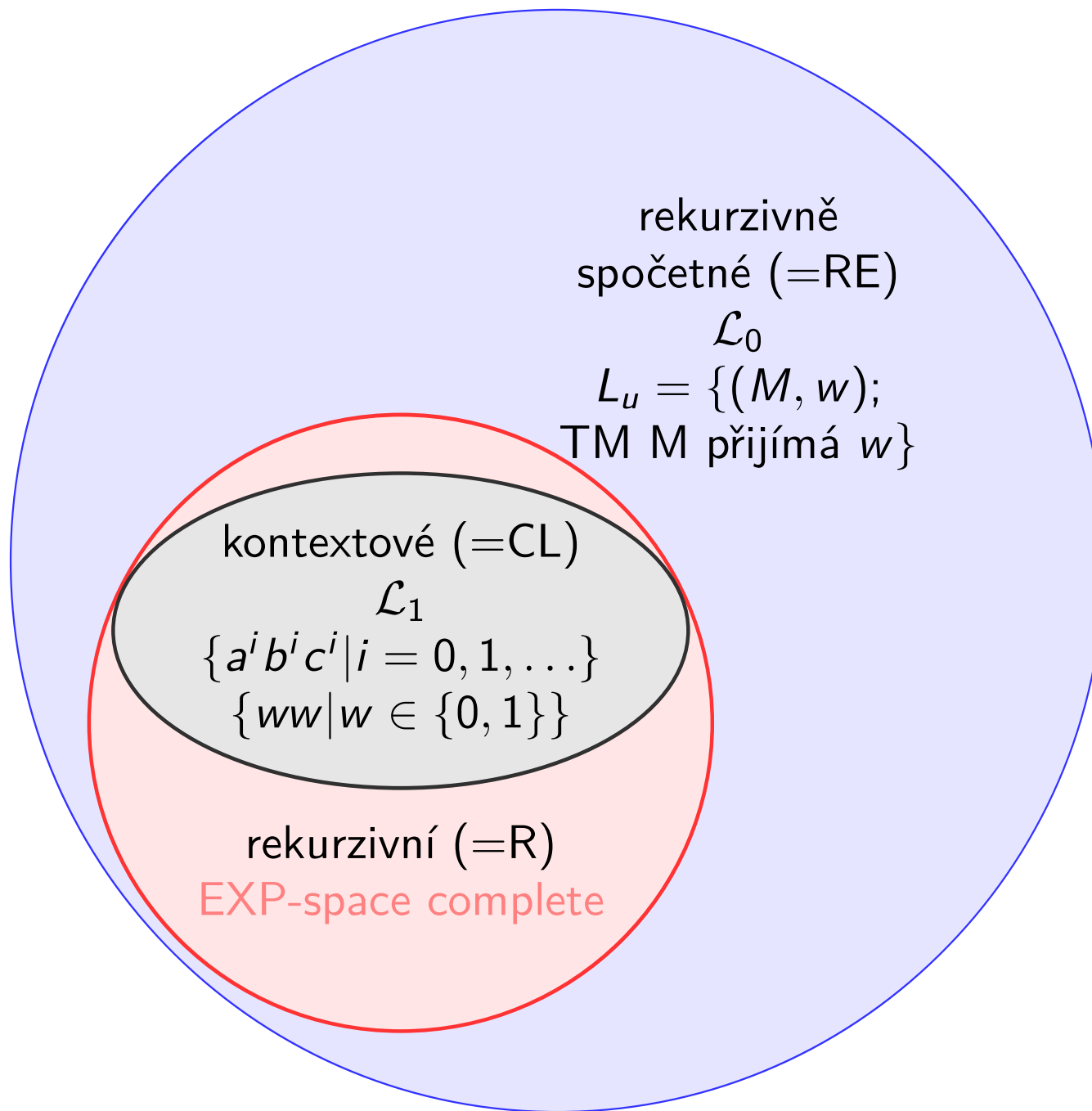
Důkaz: Každý jazyk generovaný obecnou gramatikou je rekurzivně spočetný.

idea: TM postupně generuje všechny derivace

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



Hierarchie jazyků (kontextové a výš)



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