

Prostorová složitost, Shrnutí

Marta Vomlelová

MS 303

Opakování *NP*

Definition (Verifikátor, *NP*, $NTIME(t(n))$)

- **Verifikátor** jazyka L je algoritmus (Turingův stroj) V , kde:

$$L = \{w \mid V \text{ pro nějaký řetězec } c \text{ přijímá } \langle w, c \rangle\}.$$

- **Třída jazyků rozhodnutelných v polynomiálním čase** (značíme *NP*) je tvořena jazyky s polynomiálním verifikátorem.
- Mějme funkci $t : \mathbb{N} \rightarrow \mathbb{R}^+$. Definujeme třídu

$$NTIME(t(n)) = \{L \mid L \text{ jazyk rozhodnutelný nedeterminist. TM v čase } O(t(n))\}.$$

Theorem

$$NP = \bigcup_k NTIME(n^k).$$

co-NP, Tautologičnost

Definition 13.1 (co-NP)

Jazyk $L \subseteq \Sigma^*$ patří do třídy **co-NP** právě když jeho doplněk $\Sigma^* - L$ patří do NP.

- P je částí NP i co-NP.
- Domníváme se, že NP-úplné problémy nejsou v co-NP.
 - ▶ pokud $P = NP$, tak jsou.

Definition 13.2 (tautologičnost)

Problém, zda je daná DNF formule výrokové logiky tautologie, nazýváme **tautologičnost TAUT**.

Theorem 13.1

Problém tautologičnosti TAUT je co-NP.

- Důkaz z pozorování, že doplněk TAUT (do množiny korektních formulí VL) je snadno převoditelný na SAT a SAT je v NP.
- Doplněk SAT je otázka, jestli negace dané formule je tautologie.
- Doplněk SAT je co-NP.
- SAT rozhoduje všechny formule, tedy i jejich negace.

$NP \cap co-NP$ (nezkouší se)

Example 13.1 (Dělitel menší než)

Mějme dvě přirozená čísla $m, n \in \mathbb{N}_+$. Existuje d dělitel m který je $1 < d \leq n$?

Lemma

Problém existence dělitele m menšího než n je NP i $co-NP$.

NP Dělitel je certifikátem, vydělíme a zkontrolujeme nulový zbytek v polynomiálním čase, tj. problém je NP .

$co-NP$ Za certifikát vezmeme seznam prvočíselných dělitelů (faktorů) m větších než n . Vynásobením ověříme rovnost m , AKS (Agrawal–Kayal–Saxena) testem ověříme prvočíselnost dělitelů v polynomiálním čase.

Prostorová složitost

- Podobně jako časovou složitost měříme prostor potřebný k výpočtu.

Definition 13.3 (prostorová složitost)

- Pro deterministický Turingův stroj M , který zastaví na každém vstupu, je **prostorová složitost** M funkce $f : \mathbb{N} \rightarrow \mathbb{N}$, kde $f(n)$ je maximální počet buněk pásky, které M přečte při jakémkoli vstupu délky n .
 - Pro nedeterministický Turingův stroj M , který všechny větve výpočtu zastaví na každém vstupu, je **prostorová složitost** M je funkce $f : \mathbb{N} \rightarrow \mathbb{N}$, kde $f(n)$ je maximální počet buněk pásky, které M přečte při jakémkoli vstupu délky n na libovolné větvi výpočtu.
 - V takovém případě říkáme, že jazyk $L(M)$ je **rozhodnutelný strojem M v prostoru $f(n)$** .
-
- Pro logaritmickou složitost musíme modifikovat výpočetní model:
 - vstupní páska je pouze na čtení
 - pracovní páska na psaní i čtení, na ní měříme prostorovou složitost.

Definition 13.4 (třídy prostorové složitosti)

Mějme funkci $f : \mathbb{N} \rightarrow \mathbb{R}^+$. Definujeme **třídy prostorové složitosti** $SPACE(f(n))$ a $NSPACE(f(n))$:

- $SPACE(f(n)) = \{L \mid L \text{ je jazyk rozhodnutelný v prostoru } O(f(n)) \text{ deterministickým TM}\},$
- $NSPACE(f(n)) = \{L \mid L \text{ je jazyk rozhodnutelný v prostoru } O(f(n)) \text{ nedeterministickým TM}\}.$

Přijímá NFA vše?

Example 13.2 (Přijímá NFA vše?)

Mějme nedeterministický konečný automat NFA . Přijímá všechny řetězce?

$$ALL_{NFA} = \{ \langle A \rangle \mid A \text{ je NFA a } L(A) = \Sigma^* \}.$$

- Označme q počet stavů A . Pokud A nepřijímá nějaký řetězec, musí nepřijímat i nějaký délky nanejvýš 2^q
 - ▶ Po přečtení písmene si pamatujeme, v jakých stavech se A může nacházet.
 - ▶ pokud se nachází ve stejné podmnožině stavů, v jaké už byl, můžeme 'smyčku' mezi těmito stavy vynechat.
 - ▶ počet možných podmnožin stavů A je 2^q .
 - ▶ Potřebujeme prostor pro pamatování *ano/ne* každého stavu, jestli je dosažitelný aktuálním slovem, a čítač cyklu do 2^q - stačí lineární prostor ($q \leq n$, stavy jsou na vstupu).
 - ★ protože čísla zapisujeme v binárním tvaru
 - ▶ nedeterministicky ověříme každý řetězec do 2^q délky; pokud při nějakém není *ano* pro žádný z přijímajících stavů, máme svědka $L(A) \neq \Sigma^*$.

Směřujeme k: ALL_{NFA} je $coNSPACE(n)$, $SPACE(n^2)$, tedy i $PSPACE$.

Theorem 13.2 (Savitch-ova věta)

Pro libovolnou funkci $f : \mathbb{N} \rightarrow \mathbb{R}_+$, pro kterou $f(n) \geq n$ platí:

$$NSPACE(f(n)) \subseteq SPACE(f^2(n)).$$

- Kdybychom simulovali NTM přímo:
 - ▶ musíme si pamatovat aktuální větev výpočtu
 - ▶ větev s $f(n)$ prostorem může běžet $2^{O(f(n))}$ kroků, v každém z nich může mít nedeterministickou volbu.
 - ▶ potřebovali bychom $2^{O(f(n))}$ prostor si volby pamatovat.
 - ▶ Máme dovoleno jen $O(f^2(n)) < 2^{O(f(n))}$ (asymptoticky).

Důkaz Savitch-ovy věty přes $CANYIELD(c_1, c_2, t)$.

V kvadratickém čase vyřešíme **problém dosažitelnosti (yieldability)**

- Pro dvě dané konfigurace NTM c_1, c_2 a číslo $t \in \mathbb{N}$,
- Je v NTM z konfigurace c_1 dosažitelná konfigurace c_2 v maximálně t krocích a maximálně používající $f(n)$ prostor?
- Pak c_1 počáteční konfigurace, c_2 přijímající, t maximální počet kroků NTM.

dosažitelnost konfigurací NTM

```
1: procedure CANYIELD(  $\langle NTM, c_1, c_2, t \rangle$  )
2:   if  $t = 1$  then
3:     return  $c_1 = c_2$  nebo dosažitelné  $NTM$  v jednom kroku?
4:   end if
5:   if  $t > 1$  then
6:     for každou konfiguraci  $c_m$  stroje  $NTM$  na prostoru  $f(n)$  do
7:        $prvni \leftarrow CANYEALD(c_1, c_m, \frac{t}{2})$ 
8:        $druha \leftarrow CANYEALD(c_m, c_2, \frac{t}{2})$ 
9:       if ( $prvni \ \& \ druha$ ) =  $TRUE$  then
10:        return Accept
11:      end if
12:    end for
13:    if pokud dosud nepřijato then
14:      return Reject
15:    end if
16:  end if
17: end procedure
```

TM M simulující NTM N v kvadratickém prostoru

- Mějme nedeterministický NTM N .
- Modifikujeme ho, aby před přijetím smazal pásku a posunul se na nejlevější políčko (při jednostranné pásce).
- Všechny přijímající stavy sloučíme do jednoho, pak stejně víc nedělá.
- Tím je přijímající konfigurace c_{accept} jednoznačná.
- Najdeme d maximální počet štěpení konfigurace v jednom kroku, tj. horní odhad pro počet konfigurací $2^{d \cdot f(n)}$ (kde $n = |w|$).
- Tím je $2^{d \cdot f(n)}$ i horní odhad času běhu libovolné větve N .

Savitch_Simulace

```
1: procedure SAVITCH_SIMULACE(  $\langle w \rangle$  )  
2:   return CANYIELD( $c_{start}, c_{accept}, 2^{d \cdot f(n)}$ )  
3: end procedure
```

Proof.

Složitost simulace

- CANYIELD potřebuje ukládat konfigurace a t , tj. $O(f(n))$ prostoru.
- Počet volání CANYIELD je logaritmický vzhledem k $t = 2^{d \cdot f(n)}$
 - ▶ Hloubka rekurze je $O(\log(2^{d \cdot f(n)}))$ tedy $O(f(n))$.
- Celkem $O(f(n)) \cdot O(f(n)) = O(f^2(n))$ prostoru.
- Zamlčeli jsme, že M potřebuje znát $f(n)$
 - ▶ buď zkouší postupně $f(n) = 1, 2, 3, \dots$
 - ▶ nebo přidáme do předpokladu že N rozhoduje jazyk v prostoru $f(n)$.



Definition 13.5 (PSPACE)

PSPACE je třída jazyků rozhodnutelých v polynomiálním prostoru deterministickým Turingovým strojem, tj.

$$PSPACE = \bigcup_{k \in \mathbb{N}} SPACE(n^k).$$

- **NPSPACE** ani nedefinujeme,
 - ▶ protože $NPSPACE = PSPACE$, protože máme Savich-ovu větu a polynom na druhou je také polynom.
- SAT is $SPACE(n)$, tj. $PSPACE$.
- ALL_{NFA} je
 - ▶ $\overline{ALL_{NFA}}$ je $NSPACE(n)$
 - ▶ dle Savitchovy věty i $SPACE(n^2)$
 - ▶ deterministické prostorové třídy jsou uzavřené na doplněk (stroj vždy zastaví - my jen obrátíme odpověď).
 - ▶ Proto ALL_{NFA} je $SPACE(n^2)$, tedy $PSPACE$.

Theorem 13.3

$$P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME = \bigcup_k TIME(2^{n^k}).$$

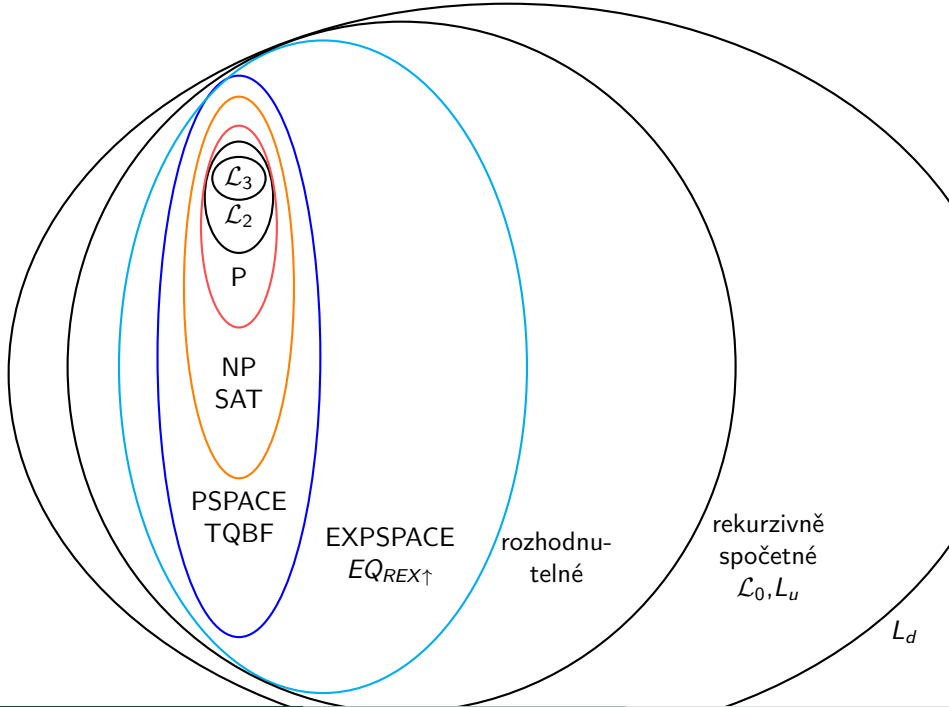
$P \subseteq PSPACE$ Stroj v čase $t(n)$ navštíví maximálně $t(n)$ políček, proto pro $t(n) \geq n$ stačí prostor $t(n)$.

- Podobně $NP \subseteq NPSPACE$.

$PSPACE = NPSPACE$ ze Savichovy věty.

$PSPACE \subseteq EXPTIME$ Pro $f(n) \geq n$ dosáhne TM M pracující v prostoru $f(n)$ maximálně $f(n) \cdot 2^{O(f(n))}$ různých konfigurací, v konečném deterministickém výpočtu se žádná konfigurace neopakuje proto existuje TM M_2 simulující TM M v čase $f(n) \cdot 2^{O(f(n))}$, tedy $PSPACE \subseteq EXPTIME$.

- Jediná nerovnost, co víme, je $P \neq EXPTIME$.



PSPACE – complete

Toto už není třeba ke zkoušce.

Definition 13.6

- Problém je **PSPACE-těžký**, pokud je každý PSPACE problém na něj převoditelný v polynomiálním čase.
- Problém je **PSPACE-úplný**, pokud je PSPACE-těžký a zároveň PSPACE.
- Jde o polynomiální časovou převoditelnost, chceme 'jednoduchou'.

Example 13.3

- **TQBF true quantified boolean formulas** Mějme plně kvantifikovanou booleovskou formuli v prenexním tvaru. Rozhodnout její pravdivost je PSPACE-úplný problém.
- Booleovská znamená, že univerzum pro proměnné je dvohodnotové $\{TRUE, FALSE\}$.
$$TQBF = \{ \langle \phi \rangle \mid \phi \text{ je pravdivá plně kvantifikovaná booleovská formule} \}$$

$TQBF \in PSPACE$

```
1: procedure  $TQBF(\langle \phi \rangle)$ 
2:   if  $\phi$  neobsahuje kvantifikátory then
3:     ověř dosazenou formuli vhodně accept/reject
4:   end if
5:   if první kvantifikátor je existenční  $\phi = (\exists x)\psi$  then
6:     if  $\psi[x/0]$  je TRUE then
7:       return accept
8:     else if  $\psi[x/1]$  je TRUE then
9:       return accept
10:    else
11:      return reject
12:    end if
13:  end if
14:  if první kvantifikátor je univerzální  $\phi = (\forall x)\psi$  then
15:    zkus dosadit 0 i 1, pokud oba TRUE, přijmi
16:  end if
17:  return reject
```

TQBF $\in$ PSPACE-těžká

- Jazyk $L(M)$ rozhodovaný TM M v prostoru $f(n) = n^k$ redukuje na TQBF.
- Vytváříme formule $\phi_{c_1, c_2, t}$ dosažitelnosti konfigurace v čase t .
- Stačí $t \leq 2^{d \cdot f(n)} = 2^{dn^k}$ pro štěpící konstantu d . Pro jednoduchost předpokládáme t je mocnina dvojky.
- Formule pro jeden krok obdobná jako v důkazu Cook-Levin-ovy věty.
- Obecně by byla $\phi_{c_1, c_2, t} \leftarrow \exists m_1 [\phi_{c_1, m_1, \frac{t}{2}} \& \phi_{m_1, c_2, \frac{t}{2}}]$ moc dlouhá.
- Formule zdvojnásobí délku, tj. prostor potřebujeme $O(t) = O(2^{df(n)})$.
- Pomůžeme si univerzálním kvantifikátorem:

$$(\exists m_1) \forall (c_3, c_4) \in \{(c_1, m_1), (m_1, c_2)\} [\phi_{c_3, c_4, \frac{t}{2}}]$$

- Kde $(\exists m_1) \forall (c_3, c_4) \in \{(c_1, m_1), (m_1, c_2)\}$ je zkratka přes kvantifikaci proměnných reprezentující konfigurace a $c_3 = c_1 \& \dots \vee \dots \rightarrow \dots$
- Formule zůstává $O(f(n))$, počet vnoření je $O(\log h) = O(\log 2^{df(n)}) = O(f(n))$, potřebný prostor je tedy $O(f^2(n))$.

$FORMULAGAME = \{\langle \phi \rangle \mid \text{hráč } E \text{ má vyhrávající strategii pro } \phi\}.$

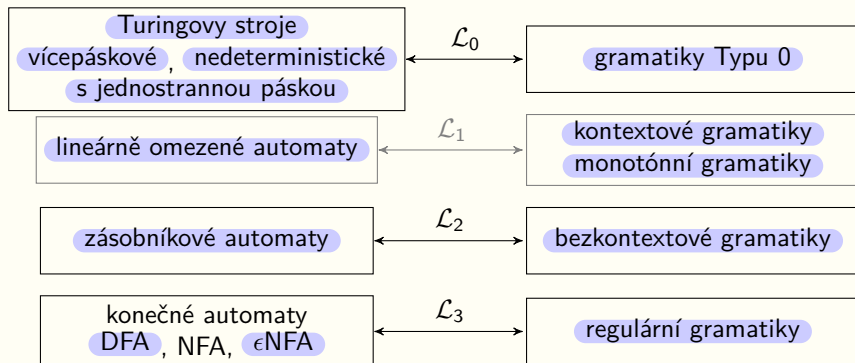
- $FORMULAGAME$ je $PSPACE$ -úplná. Je to převlečená TQBF.
 - ▶ Existenční kvantifikátor jsou moje tahy
 - ▶ univerzální jsou tahy oponenta.



Shrnutí - Automaty, gramatiky, jazyky

- L_d, L_u .
- $P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME = \bigcup_k TIME(2^{n^k})$.

Automaty a gramatiky – dva způsoby popisu



Definice

- Jednotlivé typy automatů a jazyky jimi přijímané, gramatik a jazyky jimi generované, jazyk generovaný gramatikou , a definice k tomu nutné.
- regulární výrazy , vztah k regulárním jazykům
- řetězcové operace , substitute , homomorfismus , inverzní homomorfismus
- bezkontextové gramatiky (CFG, CFL): derivační strom , jednoznačnost/víceznačnost gramatiky a CFL jazyka, Chomského normální tvar gramatiky
- zásobníkové automaty PDA, L(P), N(P), deterministické zásobníkové automaty, bezprefixové jazyky

2025 ne Dvousměrné konečné automaty , Dyckovy jazyky ,

- Turingův stroj , rekurzivní a rekurzivně spočetné jazyky, Diagonální jazyk $L_d = \{w; \text{TM s kódem } w \text{ nepřijímá vstup } w\}$, Univerzální jazyk (Univerzální Turingův stroj)
- SAT , 3SAT , PCP a nerozhodnutelné problémy pro CFG
- Třídy časové složitosti (TIME , NTIME ,) P , verifikátor , NP , co-NP , NP-úplnost jazyka , třídy prostorové složitosti PSPACE
- Redukce (převoditelnost) a polynomiální převoditelnost rozhodovacích problémů.

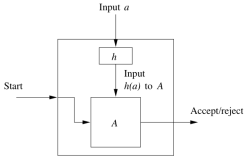
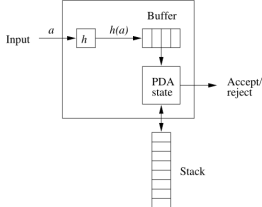
Věty

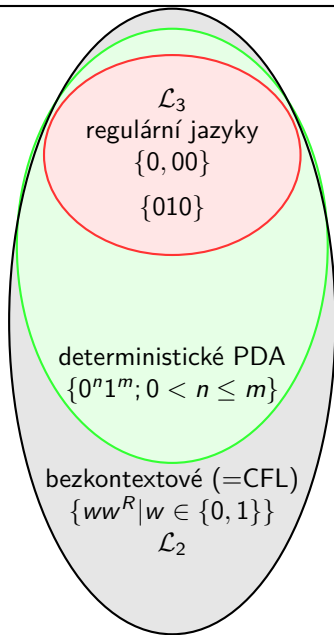
- Mihyll–Nerodova věta , Pumping lemma pro regulární jazyky , Pumping lemma pro bezkontextové jazyky , Kleeneho věta (algebraická definice regulárních jazyků),
- vztahy pojmů ve Figure Chomského hierarchie , i v rámci rámečku, i různých rámečků,
- nedeterminismus: nutný u zásobníkových automatů a lineárně omezených automatů, u konečných automatů a TM ne,
- Cook-Levin-ova věta (3SAT je NP-úplný), Postova věta
- uzávěrové vlastnosti – důkaz ANO, protipříklad NE k Tabulce níže, uzávěrové vlastnosti regulárních a CFL jazyků na řetězcové operace.

Algoritmy

- Nedosažitelné a dosažitelné stavy konečného automatu (FA),
- Rozlišitelné a ekvivalentní stavy FA, Ekvivalence FA,
- Nalezení reduktu DFA,
- Podmnožinová konstrukce DFA z NFA
- Redukce bezkontextové gramatiky
- Převod CFG na gramatiku v Chomského normální formě , CYK (slovo v CFL).
- Převod CFG na zásobníkový automat , převod PDA na gramatiku.

Uzávěrové vlastnosti v kostce

jazyk	regulární (RL)	bezkontextové	deterministické CFL
sjednocení	$F_1 \times Q_2 \cup Q_1 \times F_2$	$S \rightarrow S_1 S_2$	$A \cap B = \overline{\overline{A} \cup \overline{B}}$
průnik	$F = F_1 \times F_2$	$L = \{0^n 1^n 2^n n \geq 1\}$ $= \{0^n 1^n 2^i n, i \geq 1\} \cap \{0^i 1^n 2^n n, i \geq 1\}$	
$\cap$ s RL	$F = F_1 \times F_2$	$F = F_1 \times F_2$	$F = F_1 \times F_2$
doplňěk	$F = Q_1 - F_1$	$A \cap B = \overline{\overline{A} \cup \overline{B}}$	$F = Q_1 - F_1, Z_0, \text{cykly}$
homo-	Kleene + elem.	$a \text{ nahrad' } S_a$	$h(0) = h(1) = 0 \text{ cca. } \cup$
morfismus	jazyky + uz.		
inverzní hom.			



$\mathcal{L}_3$
regulární jazyky
 $\{0, 00\}$
 $\{010\}$

deterministické PDA
 $\{0^n 1^m; 0 < n \leq m\}$

bezkontextové (=CFL)
 $\{ww^R | w \in \{0, 1\}^*\}$
 $\mathcal{L}_2$

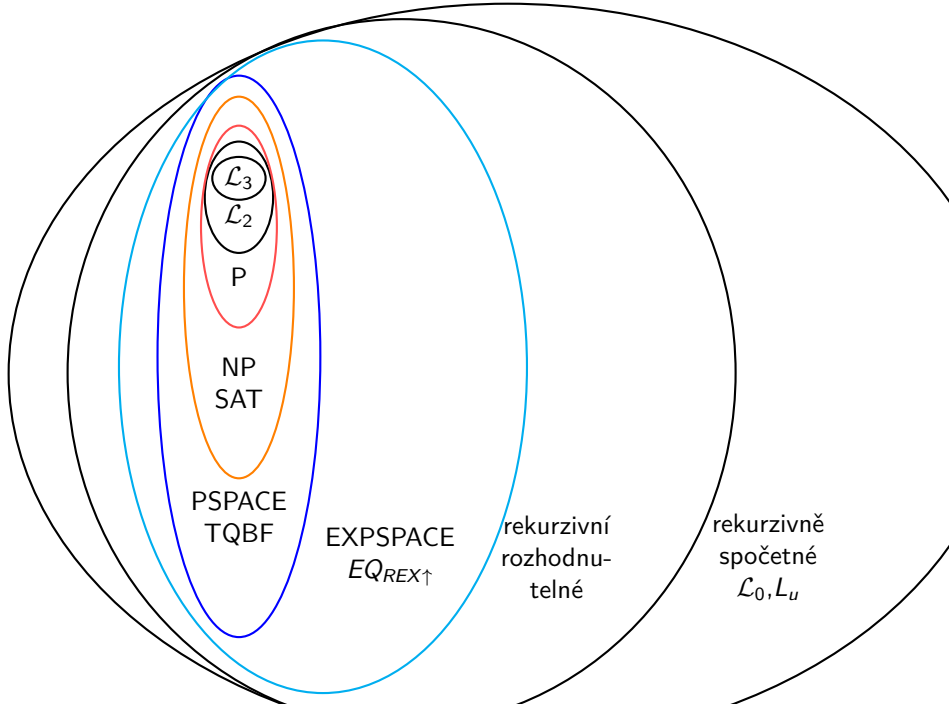
kontextové (=CL)

$\mathcal{L}_1$
 $\{a^i b^j c^i | i = 0, 1, \dots\}$
 $\{ww | w \in \{0, 1\}^*\}$

rekurzivně
spočetné

$\mathcal{L}_0$
 $L_u = \{(M, w);$
TM M přijímá $w\}$

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



Děkuji za pozornost.