

62. Klasifikace formálních jazyků (Chomského hierarchie), vlastnosti formálních jazyků a jejich rozhodnutelnost.

Odkaz na video: [SZZ: Chomského hierarchie a vlastnosti jazyků](#)

Video pokrývá témata: *Chomského hierarchie a vlastnosti jazyků. Gramatika. Třídy gramatik. Chomského hierarchie. Ukázka gramatik. Operace nad jazyky. Klasické problémy jazyků. Konečné jazyky. Regulární jazyky. Deterministické bezkontextové jazyky. Bezkontextové jazyky. Deterministické kontextové jazyky. Kontextové jazyky. Rekursivní jazyky. Rekursivně vyčíslitelné jazyky. Zjemnění Chomského hierarchie.*

Gramatiky

Gramatika je čtveřice $G = (N, \Sigma, P, S)$, kde

- N je konečná, neprázdná množina neterminálů (abeceda neterminálních symbolů)
- Σ je konečná, neprázdná množina terminálů (abeceda terminálních symbolů), kde $N \cap \Sigma = \emptyset$
- $P \subseteq (N \cup \Sigma)^* N (N \cup \Sigma)^* \times (N \cup \Sigma)^*$ je konečná množina pravidel
- $S \in N$ je startující neterminál

Ať $G = (N, \Sigma, P, S)$ je gramatika. Řetězec $a \in (N \cup \Sigma)^*$ nazveme **větnou formou**, řetězec $b \in \Sigma^*$ nazveme **větou**. Definujeme **relaci derivačního kroku** $\Rightarrow$ mezi větnými formami jako

$$(\beta\alpha\gamma \Rightarrow \beta\delta\gamma) \Leftrightarrow \beta, \gamma, \delta \in (N \cup \Sigma)^* \wedge \alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^* \wedge (\alpha, \delta) \in P,$$

tedy část větné formy α obsahující alespoň jeden neterminální symbol je možné v rámci derivačního kroku nahradit za větnou formu δ , pokud to umožňuje nějaké pravidlo gramatiky.

Ať $G = (N, \Sigma, P, S)$ je gramatika, pak **jazyk generovaný gramatikou** je množina vět

$$L(G) = \{w \in \Sigma^* \mid S \Rightarrow^+ w\}.$$

Lze definovat různé třídy gramatik, jež se budou vzájemně lišit povoleným tvarem pravidel. Ať $G = (N, \Sigma, P, S)$ je gramatika. Dle typu pravidel rozlišujeme následující gramatiky.

Obecná (neomezená) gramatika (gramatika typu 0)

- $P \subseteq (N \cup \Sigma)^* N (N \cup \Sigma)^* \times (N \cup \Sigma)^*$, P je konečná množina

- neklademe žádné omezení na nejobecnější tvar pravidel, podstatné pouze je, aby na levé straně každého pravidla byl alespoň jeden neterminál

Kontextová gramatika (gramatika typu 1)

- pravidla mají některý z následujících tvarů:
 - $(\beta A \gamma, \beta \alpha \gamma)$, kde $\beta, \gamma \in (N \cup \Sigma)^* \wedge \alpha \in (N \cup \Sigma)^+ \wedge A \in N$
 - takové pravidlo vyžaduje určitý kontext kolem přepisovaného neterminálu, bez něž není možné pravidlo použít
 - tento kontext zůstane zachován i po přepsání neterminálu A
 - neterminál A zde není možné přepsat na ε
 - (S, ε) , pokud se S nevyskytuje na pravé straně žádného pravidla
 - toto pravidlo zajišťuje, aby gramatika mohla generovat řetězec ε
- alternativně lze kontextovou gramatiku zavést tak, že obsahuje pravidla s následujícími vlastnostmi:
 - (α, β) , kde $\alpha \in (N \cup \Sigma)^* N (N \cup \Sigma)^* \wedge \beta \in (N \cup \Sigma)^+ \wedge |\alpha| \leq |\beta|$
 - tato pravidla mají stejnou podobu jako u neomezených gramatik, na levé straně musí být alespoň jeden neterminál, na pravé cokoliv
 - nevyžaduje se zde přenášení kontextu jako v případě předchozí definice
 - pravidla ale musí být **nezkracující**, tedy pravá strana pravidla se musí skládat alespoň z tolika symbolů jako levá strana, což u neomezených gramatik nepožadujeme
 - opět si všimněme, že čistě pomocí tohoto typu pravidel nelze vygenerovat řetězec ε o délce 0, neboť už starující neterminál má délku 1
 - (S, ε) , pokud se S nevyskytuje na pravé straně žádného pravidla
 - speciální pravidlo, které nám dovoluje vygenerovat ε
- jde o omezenou variantu obecných gramatik, tedy každá kontextová gramatika splňuje definici obecné gramatiky, ale existují obecné gramatiky, které nejsou kontextové

Bezkontextová gramatika (gramatika typu 2)

- $P \subseteq N \times (N \cup \Sigma)^*$, P je konečná množina
- pravidla mají vždy na levé straně právě jeden neterminál a žádný terminál, pravá strana je libovolná
- jde o omezenou variantu kontextových gramatik, tedy každá bezkontextová gramatika *(ovšem dodatečně upravená tím způsobem, že neobsahuje epsilon pravidla s výjimkou pravidla $S \rightarrow \varepsilon$, přičemž S není na pravé straně žádného pravidla)* splňuje definici kontextové gramatiky, ale existují kontextové gramatiky, které nejsou bezkontextové

Regulární gramatika (gramatika typu 3)

- pravidla mají některý z následujících tvarů (pravá regulární gramatika):
 - (A, aB) , kde $A, B \in N \wedge a \in \Sigma$
 - (A, a) , kde $A \in N \wedge a \in \Sigma$

- (A, ϵ) , kde $A \in N$
- alternativně lze regulární gramatiku zavést následovně (levá regulární gramatika):
 - (A, Ba) , kde $A, B \in N \wedge a \in \Sigma$
 - (A, a) , kde $A \in N \wedge a \in \Sigma$
 - (A, ϵ) , kde $A \in N$
- další alternativa (pravá lineární gramatika):
 - (A, wB) , kde $A, B \in N \wedge w \in \Sigma^*$
 - (A, w) , kde $A \in N \wedge w \in \Sigma^*$
- další alternativa (levá lineární gramatika):
 - (A, Bw) , kde $A, B \in N \wedge w \in \Sigma^*$
 - (A, w) , kde $A \in N \wedge w \in \Sigma^*$
- jde o omezenou variantu bezkontextových gramatik, tedy každá regulární gramatika splňuje definici bezkontextové gramatiky, ale existují bezkontextové gramatiky, které nejsou regulární

Chomského hierarchie jazyků

Ať Σ je abeceda. Množina 2^{Σ^*} je nespočetná třída všech jazyků nad abecedou Σ . **Chomského hierarchie jazyků** je hierarchie následujících významných podtříd třídy 2^{Σ^*} definovaných pomocí gramatik:

- $\mathcal{L}_3$ - třída **regulárních jazyků**
 - třída jazyků generovaných **regulárními gramatikami** (gramatikami typu 3)
- $\mathcal{L}_2$ - třída **bezkontextových jazyků**
 - třída jazyků generovaných **bezkontextovými gramatikami** (gramatikami typu 2)
- $\mathcal{L}_1$ - třída **kontextových jazyků**
 - třída jazyků generovaných **kontextovými gramatikami** (gramatikami typu 1)
- $\mathcal{L}_0$ - třída **rekurzivně vyčíslitelných jazyků**
 - třída jazyků generovaných **obecnými gramatikami** (gramatikami typu 0)

Přičemž platí

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0 \subset 2^{\Sigma^*},$$

tedy

- každý regulární jazyk je bezkontextový
- každý bezkontextový jazyk je kontextový
- každý kontextový jazyk je rekurzivně vyčíslitelný
- existuje bezkontextový jazyk, který není regulární
 - neregularitu lze dokázat pomocí Pumping lemmatu pro regulární jazyky, případně pomocí Myhill-Nerodovy věty
 - bezkontextovost lze ukázat konstrukcí bezkontextové gramatiky
 - $L = \{a^n b^n \mid n \geq 0\}$
- existuje kontextový jazyk, který není bezkontextový
 - nebezkontextovost lze dokázat pomocí Pumping lemmatu pro bezkontextové jazyky

- kontextovost lze ukázat konstrukcí kontextové gramatiky
 - $L = \{a^n b^n c^n \mid n \geq 0\}$
- existuje rekurzivně vyčíslitelný jazyk, který není kontextový
 - nekontextovost lze dokázat pomocí diagonalizace, případně pomocí redukce (redukci typicky ukážeme silnější vlastnost, tedy nerekurzivitu)
 - rekurzivní vyčíslitelnost lze ukázat konstrukcí neomezené gramatiky
 - HP: $L = \{ \langle M \rangle \# \langle w \rangle \mid \langle M \rangle \text{ je kód Turingova stroje, } \langle w \rangle \text{ je kód řetězce a } M \text{ zastaví na } w \}$
- existuje jazyk, který není rekurzivně vyčíslitelný
 - lze dokázat pomocí diagonalizace, případně pomocí redukce
 - co-HP: $L = \{ \langle M \rangle \# \langle w \rangle \mid \langle M \rangle \text{ je kód Turingova stroje, } \langle w \rangle \text{ je kód řetězce a } M \text{ cyklí na } w \}$

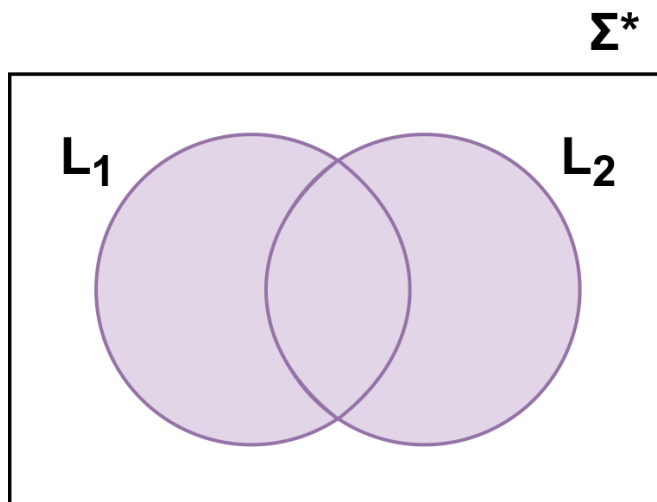
Operace nad formálními jazyky

Ať Σ je abeceda, $\mathbb{L}$ je třída jazyků, kde $\mathbb{L} \subseteq 2^{\Sigma^*}$, a ať $\circ : (2^{\Sigma^*})^n \rightarrow 2^{\Sigma^*}$ je n -ární operace na množině 2^{Σ^*} . Třída $\mathbb{L}$ je **uzavřena na operaci** $\circ$, pokud $\forall L_1, L_2, \dots, L_n \in \mathbb{L} : \circ(L_1, L_2, \dots, L_n) \in \mathbb{L}$.

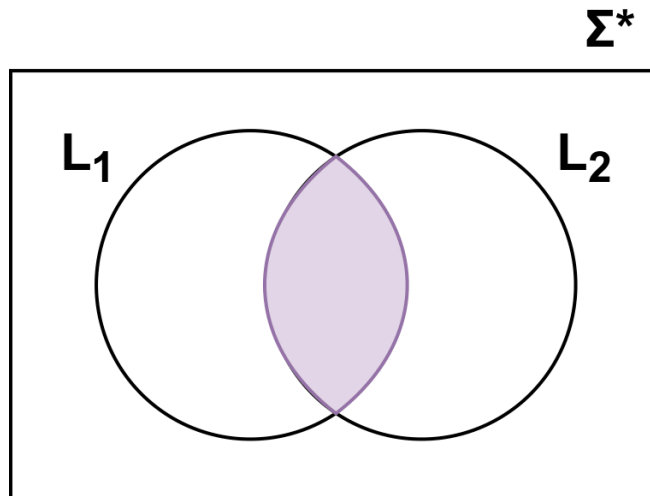
Neformálně řečeno, pokud operace přijme na vstup pouze jazyky ze třídy $\mathbb{L}$, výsledkem bude opět jazyk ze třídy $\mathbb{L}$.

V kontextu formálních jazyků se typicky bavíme o operacích:

- binární operace sjednocení $\cup$**
 - $L_1 \cup L_2 \triangleq \{w \in \Sigma^* \mid w \in L_1 \vee w \in L_2\}$
 - množina řetězců univerza, které patří do jazyka L_1 nebo do jazyka L_2
 - $\{a, b, ab\} \cup \{bbb\} = \{a, b, ab, bbb\}$

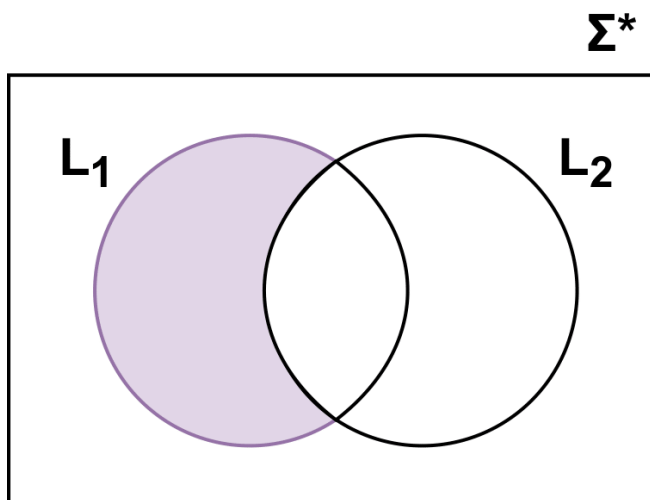


- binární operace průniku $\cap$**
 - $L_1 \cap L_2 \triangleq \{w \in \Sigma^* \mid w \in L_1 \wedge w \in L_2\}$
 - množina řetězců univerza, které patří do jazyka L_1 i do jazyka L_2
 - $\{a, b, ab\} \cap \{bbb, b\} = \{b\}$



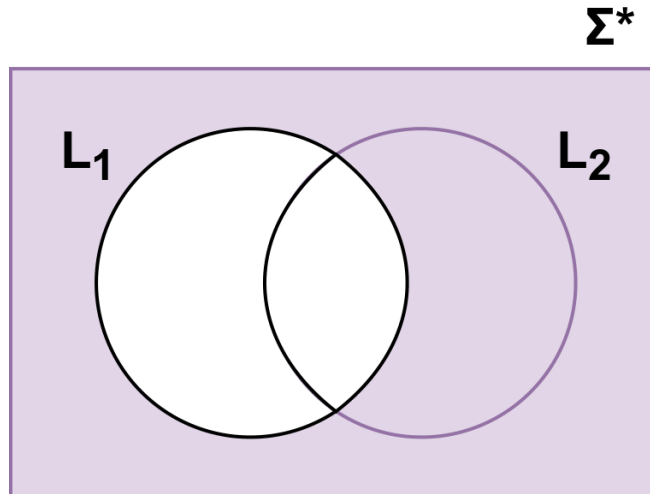
- **binární operace rozdílu $\setminus$**

- $L_1 \setminus L_2 \triangleq \{x \in \Sigma^* \mid x \in L_1 \wedge x \notin L_2\}$
- množina řetězců univerza, které patří do jazyka L_1 , ale nepatří do jazyka L_2
- $\{a, b, ab\} \setminus \{bbb, b\} = \{a, ab\}$



- **unární operace doplňku (komplementu) $\bar{}$, co-**

- $\bar{L}_1 = \text{co-}L_1 \triangleq \{x \in \Sigma^* \mid x \notin L_1\}$
- množina řetězců univerza, které nepatří do jazyka L_1



- **binární operace konkatence**
 - $L_1 \cdot L_2 \triangleq \{x_1 \cdot x_2 \in \Sigma^* \mid x_1 \in L_1 \wedge x_2 \in L_2\}$
 - množina řetězců univerza, které lze rozdělit na dvě části tak, že první z nich patří do jazyka L_1 a druhá z nich do jazyka L_2
 - $\{a, b, ab\} \cdot \{bbb, b\} = \{abbb, bbbb, abbbb, ab, bb, abb\}$
- **unární operace reverzace** ^R
 - $L^R \triangleq \{x^R \mid x \in L\}$
 - množina řetězců univerza, které jsou reverzací řetězců jazyka L
 - $\{abbb, bbbb, abbbb, ab, bb, abb\}^R = \{bbba, bbbb, bbbba, ba, bb, bba\}$
- **iterace** ^{*}
 - $L^* \triangleq \bigcup_{k=0}^{\infty} L^k$
 - množina řetězců univerza, které jsou konkatencí konečně mnoha řetězců z jazyka L
 - $\{a\}^* = \{a^n \mid n \geq 0\}$
- **pozitivní iterace** ⁺
 - $L^+ \triangleq \bigcup_{k=1}^{\infty} L^k$
 - $L^+ = L^*L$
 - $\{a\}^+ = \{a^n \mid n \geq 1\}$
- **nekonečné sjednocení**
 - *sjednocení nekonečně mnoha jazyků*
- **nekonečný průnik**
 - *průnik nekonečně mnoha jazyků*

Klasické problémy formálních jazyků

Ať Σ je abeceda. Dále ať M je nějaký výpočetní model reprezentující jazyk $L(M) \in 2^{\Sigma^*}$ (konečný automat, zásobníkový automat, Turingův stroj, gramatika,...). Definujeme:

- **problém náležitosti**
 - Je dán výpočetní model M a řetězec $w \in \Sigma^*$. Platí, že $w \in L(M)$?
 - Je daný řetězec přijímán/generován daným výpočetním modelem?

- **problém prázdnoty**
 - Je dán výpočetní model M . Platí, že $L(M) = \emptyset$?
 - Přijímá/generuje daný výpočetní model prázdný jazyk?
- **problém neprázdnosti**
 - Je dán výpočetní model M . Platí, že $\exists w \in \Sigma^* : w \in L(M)$?
 - Přijímá/generuje daný výpočetní model neprázdný jazyk?
- **problém univerzality**
 - Je dán výpočetní model M . Platí, že $L(M) = \Sigma^*$?
 - Přijímá/generuje daný výpočetní model univerzální jazyk?
- **problém konečnosti**
 - Je dán výpočetní model M . Platí, že $\exists n \in \mathbb{N} : |L(M)| = n$?
 - Přijímá/generuje daný výpočetní model konečný jazyk?
- **problém inkluze**
 - Jsou dány dva výpočetní modely stejného typu M_1, M_2 . Platí, že $L(M_1) \subseteq L(M_2)$?
 - Jsou jazyky daných výpočetních modelů v relaci inkluze?
- **problém ekvivalence**
 - Jsou dány dva výpočetní modely stejného typu M_1, M_2 . Platí, že $L(M_1) = L(M_2)$?
 - Jsou jazyky daných výpočetních modelů v relaci shodné?
- *problém regularity*
 - Je dán výpočetní model M . Platí, že $L(M) \in \mathcal{L}_3$?
 - Přijímá/generuje daný výpočetní model regulární jazyk?

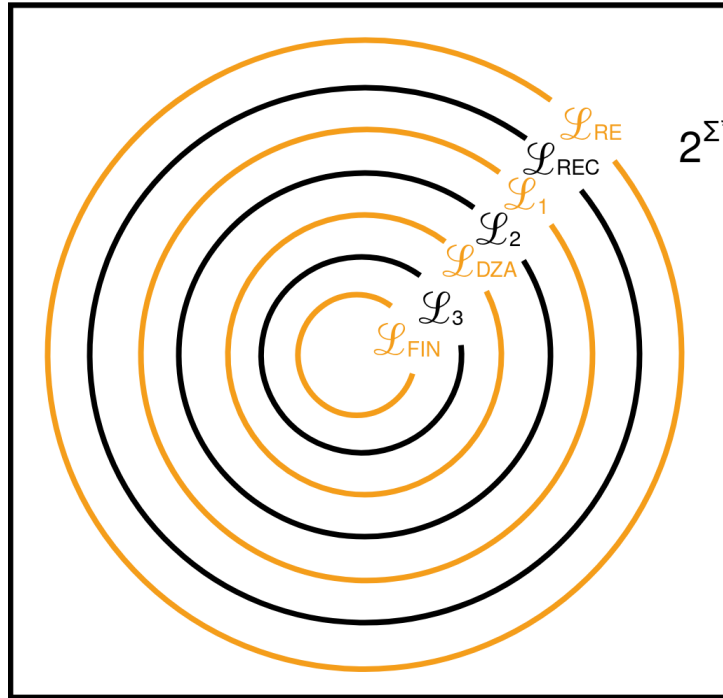
Ať Σ je abeceda, $\mathbb{L}$ je třída jazyků, kde $\mathbb{L} \subseteq 2^{\Sigma^*}$. Problém P (na nějž je odpověď ano, nebo ne) je pro třídu $\mathbb{L}$ **rozhodnutelný**, pokud existuje úplný Turingův stroj rozhodující jazyk problému P . Jinak je tento problém **nerozhodnutelný**. Pokud existuje Turingův stroj, který jazyk problému P přijímá, ale neexistuje úplný Turingův stroj, který tento jazyk rozhoduje, pak je tento problém **částečně rozhodnutelný**.

Jinak řečeno, problém P je pro třídu $\mathbb{L}$ rozhodnutelný právě tehdy, pokud existuje algoritmus, který na vstup přijímá libovolnou instanci daného problému a v konečném čase pro každou takovou instanci korektně skončí s korektním výsledkem (se správnou odpovědí pro daný problém) – tedy odpovídající TS nikdy necyklí.

Například problém prázdnoty je pro třídu regulárních jazyků rozhodnutelný, neboť jazyk $\{ \langle A \rangle \mid \langle A \rangle \text{ je kód konečného automatu } A \text{ s abecedou } \Sigma, \text{ kde } L(A) = \emptyset \}$ je rekurzivní, tedy existuje úplný Turingův stroj, který tento jazyk rozhoduje. Ovšem problém prázdnoty pro třídu kontextových jazyků rozhodnutelný není, neboť jazyk $\{ \langle M \rangle \mid \langle M \rangle \text{ je kód nedeterministického lineárně omezeného automatu s abecedou } \Sigma, \text{ kde } L(M) = \emptyset \}$ není rekurzivní, neexistuje žádný úplný Turingův stroj, který by tento problém rozhodoval.

Významné třídy jazyků a jejich vlastnosti

V následujícím textu budou představeny podrobněji jednotlivé třídy Chomského hierarchie. Navíc bude představeno několik dalších významných podtříd třídy 2^{Σ^*} , které však nejsou definovány pomocí gramatik.



Konečné jazyky

Třída **konečných jazyků** L_{FIN} je třída jazyků přijímána/generována:

- **konečnými automaty bez dosažitelných a současně ukončujících cyklů**
- **regulárními výrazy bez iterace**
- **regulárními gramatikami bez rekurze**

Každý konečný jazyk je regulární, ale existují regulární jazyky, které nejsou konečné, tedy

$$L_{FIN} \subset L_3$$

Příkladem regulárního jazyka, který není konečný, je $L = \{a^n \mid n \geq 0\}$.

Uzávěrové vlastnosti. Třída konečných jazyků **je uzavřena** na

- **sjednocení**
- **průnik**
- **rozdíl**
- **konkatenaci**
- **reverzaci**

Třída konečných jazyků **není uzavřena** na

- **doplňěk**

$$\circ \quad \text{CO-}\emptyset = \Sigma^* \notin L_{FIN}$$

- **iteraci**
 - $\{a\}^* = \{a^n \mid n \geq 0\} \notin \mathcal{L}_{\text{FIN}}$
- **pozitivní iteraci**
 - $\{a\}^+ = \{a^n \mid n \geq 1\} \notin \mathcal{L}_{\text{FIN}}$

Rozhodnutelné problémy. Pro třídu konečných jazyků jsou rozhodnutelné problémy

- **náležitosti**
 - pro deterministický konečný automat $A = (Q, \Sigma, \delta, s_0, F)$ bez dosažitelných a současně ukončujících cyklů a $w \in \Sigma^*$ lze v konečném čase otestovat, zda $\exists f \in F : (s_0, w) \vdash^* (f, \epsilon)$, tedy zda při čtení řetězce w skončíme v koncovém stavu
- **prázdnosti**
 - pro deterministický konečný automat $A = (Q, \Sigma, \delta, s_0, F)$ bez dosažitelných a současně ukončujících cyklů lze v konečném čase spočítat množinu dosažitelných stavů $R \subseteq Q$ a otestovat, zda $R \cap F = \emptyset$, tedy zda jsou všechny koncové stavy nedosažitelné
- **neprázdnosti**
 - pro deterministický konečný automat $A = (Q, \Sigma, \delta, s_0, F)$ bez dosažitelných a současně ukončujících cyklů lze v konečném čase spočítat množinu dosažitelných stavů $R \subseteq Q$ a otestovat, zda $R \cap F \neq \emptyset$, tedy zda je některý koncový stav dosažitelný
- **univerzality**
 - triviálně rozhodnutelné, žádný konečný jazyk nemůže být univerzální
- **konečnosti**
 - triviálně rozhodnutelné, každý konečný jazyk je konečný
- **inkluze**
 - pro automaty A_1, A_2 bez dosažitelných a současně ukončujících cyklů lze v konečném čase ověřit, zda platí $L(A_1) \cap \text{co-}L(A_2) = \emptyset$
- **ekvivalence**
 - pro automaty A_1, A_2 bez dosažitelných a současně ukončujících cyklů lze v konečném čase ověřit, zda platí $L(A_1) \subseteq L(A_2) \wedge L(A_2) \subseteq L(A_1)$

Regulární jazyky

Třída **regulárních jazyků** $\mathcal{L}_3$ je třída jazyků přijímána/generována:

- **regulárními gramatikami**
- **konečnými automaty**
- **regulárními výrazy**

Každý regulární jazyk je deterministický bezkontextový, ale existují deterministické bezkontextové jazyky, které nejsou regulární, tedy

$$\mathcal{L}_3 \subset \mathcal{L}_{\text{DZA}}$$

Příkladem deterministického bezkontextového jazyka, který není regulární, je $L = \{a^n b^n \mid n \geq 0\}$.

Uzávěrové vlastnosti. Třída regulárních jazyků **je uzavřena** na

- sjednocení
- průnik
- rozdíl
- konkatenaci
- reverzaci
- doplněk
- iteraci
- pozitivní iteraci

Rozhodnutelné problémy. Pro třídu regulárních jazyků jsou rozhodnutelné problémy

- **náležitosti**
 - pro $A = (Q, \Sigma, \delta, s_0, F)$, $w \in \Sigma^*$ lze v konečném čase otestovat, zda $\exists f \in F : (s_0, w) \vdash^* (f, \epsilon)$, tedy zda při čtení řetězce w skončíme v koncovém stavu
- **prázdnosti**
 - pro $A = (Q, \Sigma, \delta, s_0, F)$ lze v konečném čase spočítat množinu dosažitelných stavů $R \subseteq Q$ a otestovat, zda $R \cap F = \emptyset$, tedy zda jsou všechny koncové stavy nedosažitelné
- **neprázdnosti**
 - pro $A = (Q, \Sigma, \delta, s_0, F)$ lze v konečném čase spočítat množinu dosažitelných stavů $R \subseteq Q$ a otestovat, zda $R \cap F \neq \emptyset$, tedy zda je některý koncový stav dosažitelný
- **univerzality**
 - pro $A = (Q, \Sigma, \delta, s_0, F)$ lze v konečném čase sestavit komplementární automat $\text{co-}A$ a ověřit, zda $L(\text{co-}A) = \emptyset$
- **konečnosti**
 - pro $A = (Q, \Sigma, \delta, s_0, F)$ lze v konečném čase odstranit všechny neukončující a nedosažitelné stavy a následně otestovat, zda je takto upravený automat acyklickým grafem
- **inkluze**
 - pro automaty A_1, A_2 lze v konečném čase ověřit, zda platí $L(A_1) \cap \text{co-}L(A_2) = \emptyset$
- **ekvivalence**
 - pro automaty A_1, A_2 lze v konečném čase ověřit, zda platí $L(A_1) \subseteq L(A_2) \wedge L(A_2) \subseteq L(A_1)$

Deterministické bezkontextové jazyky

Třída **deterministických bezkontextových jazyků** $\mathcal{L}_{\text{DZA}}$ je třída jazyků přijímána/generována:

- **deterministickými zásobníkovými automaty**

Každý deterministický bezkontextový jazyk je bezkontextový, ale existují bezkontextové jazyky, které nejsou deterministické bezkontextové, tedy

$$\mathcal{L}_{\text{DZA}} \subset \mathcal{L}_2$$

Příkladem bezkontextového jazyka, který není deterministický bezkontextový, je $L = \{ww^R \mid w \in \{a, b\}^*\}$.

Uzávěrové vlastnosti. Třída deterministických bezkontextových jazyků **je uzavřena** na

- **doplňěk**
- **průnik s regulárním jazykem**
 - průnik každého deterministického bezkontextového jazyka s libovolným regulárním jazykem je deterministický bezkontextový

Třída deterministických bezkontextových jazyků **není uzavřena** na

- **sjednocení**
 - $\{a^{2^n}b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\} \notin \mathcal{L}_{DZA}$
 - po přečtení symbolů a není zřejmé, zda máme očekávat stejný, nebo poloviční počet symbolů b , nelze to deterministicky určit
- **průnik**
 - $\{a^n b^n c^m \mid m, n \geq 0\} \cap \{a^m b^n c^n \mid m, n \geq 0\} = \{a^n b^n c^n \mid n \geq 0\} \notin \mathcal{L}_{DZA}$
- **konkatenaci**
 - $L_1 = \{\#\} \{a^{2^n}b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\} \in \mathcal{L}_{DZA}$
 - $L_2 = \{\#\}^* \in \mathcal{L}_{DZA}$
 - $L_2 \cdot L_1 = \{\#\}^* (\{\#\} \{a^{2^n}b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}) = \{\#\}^* \{a^{2^n}b^n \mid n \geq 0\} \cup \{\#\}^* \{a^n b^n \mid n \geq 0\} \notin \mathcal{L}_{DZA}$
 - zde jsme si vypůjčili jazyk $\{a^{2^n}b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$ z příkladu neuzavřenosti $\mathcal{L}_{DZA}$ na sjednocení. Tento jazyk není deterministický bezkontextový, ale pokud před řetězcem z první části jazyka umístíme řetězec $\#$, tzn. $\{\#\} \{a^{2^n}b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\}$, tento jazyk již deterministický bezkontextový je, neboť přítomnost symbolu $\#$ nám dává informaci o tom, kolik symbolů b máme očekávat po dočtení symbolů a
 - Pokud před takový jazyk přikonkatenujeme jazyk $\{\#\}^*$, který je deterministický bezkontextový, získáme $\{\#\}^* \{a^{2^n}b^n \mid n \geq 0\} \cup \{\#\}^* \{a^n b^n \mid n \geq 0\}$, tzn. informace o symbolu $\#$ na začátku řetězce nám již nestačí k určení, kolik symbolů b máme očekávat po přečtení symbolů a , nelze to deterministicky stanovit
 - deterministické bezkontextové jazyky tedy nejsou uzavřeny na konkatenaci
- **iteraci**
 - $L_1 = \{\#\} \{a^{2^n}b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\} \cup \{\#\} \in \mathcal{L}_{DZA}$
 - $L_2 = \{\#\} \{a^n b^m \mid m, n \geq 0\} \in \mathcal{L}_3$
 - $L_1 \cap L_2 \in \mathcal{L}_{DZA}$ (z uzavřenosti na průnik s regulárními jazyky)
 - $(L_1)^* \cap L_2 = \{\#\} \{a^{2^n}b^n \mid n \geq 0\} \cup \{\#\} \{a^n b^n \mid n \geq 0\} \notin \mathcal{L}_{DZA}$
 - v tomto průniku mohou být obsaženy pouze řetězce, které začínají na $\#$, pokračují nějakou posloupností symbolů a a dále posloupností symbolů b
 - průnik tedy z $(L_1)^*$ odstraní všechny řetězce, které nezačínají na $\#$ (tzn. část $\{a^n b^n \mid n \geq 0\}$ jazyka L_1 zmizí), ale oproti jazyku L_1 přidá řetězce typu $\{\#\} \{a^n b^n \mid n \geq 0\}$ (neboť iterace obsahuje i jazyk $L_1 \cdot L_1$)
 - výsledkem průniku je tedy $(L_1)^* \cap L_2 = \{\#\} \{a^{2^n}b^n \mid n \geq 0\} \cup \{\#\} \{a^n b^n \mid n \geq 0\}$, ovšem tento jazyk nemůže být deterministický bezkontextový, neboť přítomnost

symbolu $\#$ nám opět neříká nic o tom, kolik máme očekávat symbolů b po přečtení symbolů a

- jelikož deterministické bezkontextové jazyky jsou uzavřeny na průnik s regulárními jazyky a jazyk L_2 je regulární, pak by $(L_1)^* \cap L_2$ byl deterministický bezkontextový právě tehdy, pokud by i $(L_1)^*$ byl deterministický bezkontextový
- jazyk $(L_1)^* \cap L_2$ ale není deterministický bezkontextový, takže ani $(L_1)^*$ nemůže být deterministický bezkontextový
- jazyk L_1 je deterministický bezkontextový a jazyk $(L_1)^*$ není deterministický bezkontextový, takže deterministické bezkontextové jazyky nejsou uzavřeny na iteraci
- **pozitivní iteraci**
 - lze použít obdobný příklad jako v předchozím bodě
- **reverzaci**
 - $L = \{\#\} \{a^{2n}b^n \mid n \geq 0\} \cup \{a^n b^n \mid n \geq 0\} \in \mathcal{L}_{DZA}$
 - $L^R = \{b^n a^{2n} \mid n \geq 0\} \{ \#\} \cup \{b^n a^n \mid n \geq 0\} \notin \mathcal{L}_{DZA}$

Rozhodnutelné problémy. Pro třídu deterministických bezkontextových jazyků jsou rozhodnutelné problémy

- **náležitosti**
 - pro deterministický zásobníkový automat $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ a řetězec $w \in \Sigma^*$ lze v konečném čase otestovat, zda $\exists f \in F : (s_0, w, Z_0) \vdash^* (f, \varepsilon, \gamma)$, tedy zda při čtení řetězce w skončíme v koncovém stavu s libovolným obsahem zásobníku
- **prázdnosti**
 - pro deterministický zásobníkový automat $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ lze v konečném čase provést jeho transformaci na ekvivalentní bezkontextovou gramatiku $G = (N, \Sigma, P, S)$, zkonstruovat množinu $N_t = \{B \in N \mid \exists w \in \Sigma^* : B \Rightarrow^+ w\}$ neterminálů generujících terminální řetězce a ověřit, zda $S \notin N_t$, tedy zda startující symbol generuje nějakou větu
- **neprázdnosti**
 - pro deterministický zásobníkový automat $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ lze v konečném čase provést jeho transformaci na ekvivalentní bezkontextovou gramatiku $G = (N, \Sigma, P, S)$, zkonstruovat množinu $N_t = \{B \in N \mid \exists w \in \Sigma^* : B \Rightarrow^+ w\}$ neterminálů generujících terminální řetězce a ověřit, zda $S \in N_t$, tedy zda startující symbol generuje nějakou větu
- **univerzality**
 - jelikož deterministické bezkontextové jazyky jsou uzavřeny na doplněk, lze deterministický zásobníkový automat $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ komplementovat a následně komplement otestovat na prázdnotu
- **konečnosti**
 - pro deterministický zásobníkový automat $A = (Q, \Sigma, \Gamma, \delta, s_0, Z_0, F)$ lze v konečném čase provést jeho transformaci na ekvivalentní bezkontextovou gramatiku $G = (N, \Sigma, P, S)$, z níž lze odstranit ε -pravidla a jednoduchá pravidla
 - dále je třeba zkonstruovat množinu $N_t = \{B \in N \mid \exists w \in \Sigma^* : B \Rightarrow^+ w\}$ neterminálů generujících terminální řetězce

- dále sestrojít relaci $\hookrightarrow \subseteq N \times N$, kde $\forall B, C \in N : B \hookrightarrow C \Leftrightarrow \exists \alpha, \beta \in (N \cup \Sigma)^* : (B, \alpha C \beta) \in P$, tedy dva neterminály jsou v relaci $\hookrightarrow$, pokud lze první přepsat v jednom kroku na větnou formu obsahující druhý neterminál
- otestujeme, zda $\exists B \in N_t : S \hookrightarrow^* B \wedge B \hookrightarrow^+ B$, tedy zda je možné z neterminálu S zderivovat neterminál, který se jednak v kladném počtu kroků přepíše sám za sebe, jednak generuje nějakou větu
- pokud ne, gramatika generuje konečný jazyk
- **ekvivalence**
 - příliš komplikované, odolávalo důkazu desítky let

Nerozhodnutelné problémy. Pro třídu deterministických bezkontextových jazyků jsou nerozhodnutelné problémy

- **inkluze**

Bezkontextové jazyky

Třída **bezkontextových jazyků** $\mathcal{L}_2$ je třída jazyků přijímána/generována:

- **bezkontextovými gramatikami**
- **zásobníkovými automaty**

Každý bezkontextový jazyk je kontextový, ale existují kontextové jazyky, které nejsou bezkontextové, tedy

$$\mathcal{L}_2 \subset \mathcal{L}_1$$

Příkladem kontextového jazyka, který není bezkontextový, je $L = \{ww \mid w \in \{a, b\}^*\}$.

Uzávěrové vlastnosti. Třída bezkontextových jazyků **je uzavřena** na

- **sjednocení**
- **konkatenaci**
- **iteraci**
- **pozitivní iteraci**
- **reverzaci**

Třída bezkontextových jazyků **není uzavřena** na

- **doplňěk**
 - $L = \{a^n b^n c^n \mid n \geq 0\} \in \mathcal{L}_2$
 - $\text{co-}L = \{a^n b^n c^n \mid n \geq 0\} \notin \mathcal{L}_2$
- **průnik**
 - $\{a^n b^n c^m \mid m, n \geq 0\} \cap \{a^m b^n c^n \mid m, n \geq 0\} = \{a^n b^n c^n \mid n \geq 0\} \notin \mathcal{L}_2$
- **rozdíl**
 - $\{a^n b^n c^m \mid m, n \geq 0\} \setminus \{w \in \{a, b, c\}^* \mid \#_b(w) \neq \#_c(w)\} = \{a^n b^n c^n \mid n \geq 0\} \notin \mathcal{L}_2$

Rozhodnutelné problémy. Pro třídu bezkontextových jazyků jsou rozhodnutelné problémy

- **náležitosti**

- pro bezkontextovou gramatiku $G = (N, \Sigma, P, S)$ převedenou do Chomského normální formy a pro řetězec $w \in \Sigma^*$, lze využít algoritmus *Cocke-Younger-Kasami*
- **prázdnosti**
 - pro bezkontextovou gramatiku $G = (N, \Sigma, P, S)$ lze vytvořit množinu $N_t = \{B \in N \mid \exists w \in \Sigma^* : B \Rightarrow^+ w\}$ neterminálů generujících terminální řetězce lze ověřit, zda $S \in N_t$, tedy zda startující symbol generuje nějakou větu
- **neprázdnosti**
 - pro bezkontextovou gramatiku $G = (N, \Sigma, P, S)$ lze vytvořit množinu $N_t = \{B \in N \mid \exists w \in \Sigma^* : B \Rightarrow^+ w\}$ neterminálů generujících terminální řetězce lze ověřit, zda $S \in N_t$, tedy zda startující symbol generuje nějakou větu
- **konečnosti**
 - pro bezkontextovou gramatiku $G = (N, \Sigma, P, S)$, z níž jsou odstraněna ϵ -pravidla a jednoduchá pravidla, je třeba zkonstruovat množinu $N_t = \{B \in N \mid \exists w \in \Sigma^* : B \Rightarrow^+ w\}$ neterminálů generujících terminální řetězce
 - dále sestrojit relaci $\hookrightarrow \subseteq N \times N$, kde $\forall B, C \in N : B \hookrightarrow C \Leftrightarrow \exists \alpha, \beta \in (N \cup \Sigma)^* : (B, \alpha C \beta) \in P$, tedy dva neterminály jsou v relaci $\hookrightarrow$, pokud lze první přepsat v jednom kroku na větnou formu obsahující druhý neterminál
 - otestuje se, zda $\exists B \in N_t : S \hookrightarrow^* B \wedge B \hookrightarrow^+ B$, tedy zda je možné z neterminálu S zderivovat neterminál, který se jednak v kladném počtu kroků přepíše sám za sebe, jednak generuje nějakou větu
 - pokud ne, gramatika generuje konečný jazyk

Nerozhodnutelné problémy. Pro třídu bezkontextových jazyků jsou nerozhodnutelné problémy

- inkluze
- univerzality
- ekvivalence

Deterministické kontextové jazyky

Třída **deterministických kontextových jazyků** $\mathcal{L}_{DLOA}$ je třída jazyků přijímána/generována:

- **deterministickými lineárně omezenými automaty**

K 29. červnu 2023 není dle dostupných informací známo, zda je tato třída vlastní podtřídou třídy kontextových jazyků, nebo je jí rovna. $\mathcal{L}_{DLOA} = DSPACE[n]$, zatímco $\mathcal{L}_{LOA} = \mathcal{L}_1 = NSPACE[n]$.
Není známo, zda $DSPACE[n] = NSPACE[n]$.

Kontextové jazyky

Třída **kontextových jazyků** $\mathcal{L}_1$ je třída jazyků přijímána/generována:

- **kontextovými gramatikami**
- **nedeterministickými lineárně omezenými automaty**

Každý kontextový jazyk je rekurzivní, ale existují rekurzivní jazyky, které nejsou kontextové, tedy

$$\mathcal{L}_1 \subset \mathcal{L}_{\text{REC}}$$

Mějme kódování kontextových gramatik nad abecedou $\{0, 1\}$. Pro $w \in \{0, 1\}^*$, kde w je kódem kontextové gramatiky, je G_w kontextovou gramatikou s kódem w . Pokud $w \in \{0, 1\}^*$ není korektním kódem kontextové gramatiky, pak $G_w = (\{S\}, \{a\}, \emptyset, S)$.

Příkladem rekurzivního jazyka, který není kontextový, je $L = \{w \in \{0, 1\}^* \mid w \notin L(G_w)\}$, což lze ukázat pomocí diagonalizace a pomocí konstrukce úplného Turingova stroje akceptujícího L .

Uzávěrové vlastnosti. Třída kontextových jazyků **je uzavřena** na

- sjednocení
- průnik
- rozdíl
- doplněk
- konkatenaci
- iteraci
- pozitivní iteraci
- reverzaci

Rozhodnutelné problémy. Pro třídu kontextových jazyků jsou rozhodnutelné problémy

- **náležitosti**
 - lineárně omezený automat (LOA) $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f)$ je úplným Turingovým strojem, jenž je možné determinizovat (výsledný úplný deterministický Turingův stroj M' nemusí být deterministickým lineárně omezeným automatem). Stačí zahájit deterministický běh stroje M' na vstupu $w \in \Sigma^*$, který vždycky skončí v konečném počtu kroků.

Částečně rozhodnutelné problémy. Pro třídu kontextových jazyků jsou částečně rozhodnutelné problémy

- **neprázdnoti**
 - lineárně omezený automat (LOA) $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f)$ je úplným Turingovým strojem, jenž je možné determinizovat (výsledný úplný deterministický Turingův stroj M' nemusí být deterministickým lineárně omezeným automatem). Stačí lexikograficky generovat veškerá slova $w \in \Sigma^*$ a zahajovat běh stroje M' na těchto slovech. Pokud je jazyk stroje M neprázdný, některé slovo bude dříve či později akceptováno. Pokud je jazyk stroje M prázdný, bude generování a testování slov cyklit.

Nerozhodnutelné problémy. Pro třídu kontextových jazyků jsou nerozhodnutelné problémy

- prázdnoti
- univerzality
- konečnosti
- inkluze
- ekvivalence

Rekurzivní jazyky

Třída **rekurzivních jazyků** $\mathcal{L}_{\text{REC}}$ je třída jazyků přijímána/generována:

- **úplnými Turingovými stroji**

Každý rekurzivní jazyk je rekurzivně vyčíslitelný, ale existují rekurzivně vyčíslitelné jazyky, které nejsou rekurzivní, tedy

$$\mathcal{L}_{\text{REC}} \subset \mathcal{L}_{\text{RE}}$$

Příkladem rekurzivně vyčíslitelného jazyka, který není rekurzivní, je $L = \{ \langle M \rangle \# \langle w \rangle \mid \langle M \rangle \text{ je kód Turingova stroje } M, \langle w \rangle \text{ je kód řetězce } w \text{ a } M \text{ zastaví na } w \}$.

Uzávěrové vlastnosti. Třída rekurzivních jazyků **je uzavřena** na

- sjednocení
- průnik
- rozdíl
- doplněk
- konkatenaci
- iteraci
- pozitivní iteraci
- reverzaci

Rozhodnutelné problémy. Pro třídu rekurzivních jazyků jsou rozhodnutelné problémy

- **náležitosti**
 - úplný Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f)$ lze determinizovat a vytvořit tak stroj M' . Stačí zahájit deterministický běh stroje M' na vstupu $w \in \Sigma^*$, který vždycky skončí v konečném počtu kroků.

Částečně rozhodnutelné problémy. Pro třídu rekurzivních jazyků jsou částečně rozhodnutelné problémy

- **neprázdnoti**
 - úplný Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f)$ lze determinizovat a vytvořit tak stroj M' . Stačí lexikograficky generovat veškerá slova $w \in \Sigma^*$ a zahajovat běh stroje M' na těchto slovech. Pokud je jazyk stroje M neprázdný, některé slovo bude dříve či později akceptováno. Pokud je jazyk stroje M prázdný, bude generování a testování slov cyklit.

Nerozhodnutelné problémy. Pro třídu kontextových jazyků jsou nerozhodnutelné problémy

- prázdnoti
- univerzality
- konečnosti
- inkluze
- ekvivalence

Rekurzivně vyčíslitelné jazyky

Třída **rekurzivně vyčíslitelných jazyků** $\mathcal{L}_{RE} = \mathcal{L}_0$ je třída jazyků přijímána/generována:

- **obecnými gramatikami**
- **(obecnými) Turingovými stroji**

Existují jazyky, které nejsou rekurzivně vyčíslitelné, tedy

$$\mathcal{L}_{RE} \subset 2^{\Sigma^*}$$

Příkladem jazyka, který není rekurzivně vyčíslitelný, je $L = \{ \langle M \rangle \# \langle w \rangle \mid \langle M \rangle \text{ je kód Turingova stroje } M, \langle w \rangle \text{ je kód řetězce } w \text{ a } M \text{ cyklí na } w \}$.

Uzávěrové vlastnosti. Třída rekurzivně vyčíslitelných jazyků **je uzavřena** na

- **sjednocení**
- **průnik**
- **konkatenaci**
- **iteraci**
- **pozitivní iteraci**
- **reverzaci**

Třída rekurzivně vyčíslitelných jazyků **není uzavřena** na

- **doplňěk**
 - Ať L je rekurzivně vyčíslitelný nerekurzivní jazyk, pak jeho doplněk není rekurzivně vyčíslitelný
 - Předpokládejme, že rekurzivně vyčíslitelné jazyky jsou uzavřeny na doplněk. Ať L je rekurzivně vyčíslitelný nerekurzivní jazyk, pak by existoval Turingův stroj M , kde $L(M) = L$. Pokud by rekurzivně vyčíslitelné jazyky byly uzavřeny na doplněk, musel by existovat i Turingův stroj M' , kde $L(M') = \text{co-}L$. Potom by bylo možné sestavit Turingův stroj M'' , který by střídavě po jednom kroku simuloval stroje M , M' na svém vstupu w . Pokud by simulace M skončila přijetím, stroj M'' by akceptoval. Pokud by simulace stroje M' skončila přijetím, stroj M'' by odmítl. Jeden ze strojů M , M' by musel dříve či později akceptovat, tedy M'' je úplný Turingův stroj rozhodující L . To je ale spor, protože L je nerekurzivní. Tedy doplněk rekurzivně vyčíslitelného nerekurzivního jazyka není rekurzivně vyčíslitelný.
 - Z toho plyne, že rekurzivně vyčíslitelné jazyky nejsou uzavřeny na doplněk.
- **rozdíl**
 - $\Sigma^* \setminus HP \notin \mathcal{L}_{RE}$

Částečně rozhodnutelné problémy. Pro třídu rekurzivně vyčíslitelných jazyků jsou částečně rozhodnutelné problémy

- **náležitosti**
 - Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f)$ je možné determinizovat, výsledkem je deterministický Turingův stroj M' . Stačí zahájit deterministický běh stroje M' na

vstupu $w \in \Sigma^*$, který skončí akceptováním, pokud $w \in L(M)$. V případě, že $w \notin L(M)$, může běh cyklit.

- **neprázdnosti**
 - Turingův stroj $M = (Q, \Sigma, \Gamma, \delta, q_0, q_f)$ je možné determinizovat, výsledkem je deterministický Turingův stroj M' . Turingův stroj N řeší problém neprázdnosti nedeterministicky vygeneruje řetězec $w \in \Sigma^*$ a spustí simulaci M' na w . Pokud M' přijímá nějaký řetězec, může být nedeterministicky vygenerován. V případě, že $L(M) = \emptyset$, může běh cyklit.

Nerozhodnutelné problémy. Pro třídu rekurzivně vyčíslitelných jazyků jsou nerozhodnutelné problémy

- **prázdnosti**
- **univerzality**
- **konečnosti**
- **inkluze**
- **ekvivalence**

Jazyky mimo třídu rekurzivně vyčíslitelných jazyků

Třída jazyků, které nejsou rekurzivně vyčíslitelné, tzn. třída $2^{\Sigma^*} \setminus \mathcal{L}_{RE}$, obsahuje nespočetně mnoho jazyků. Není znám žádný konečně stavový model, který by ji přijímal.

Uzávěrové vlastnosti. Tato třída jistě není uzavřena na

- **sjednocení**
 - $\{ \langle M \rangle \mid M \text{ je úplný TS} \} \cup \text{co-}\{ \langle M \rangle \mid M \text{ je úplný TS} \} = \Sigma^* \notin 2^{\Sigma^*} \setminus \mathcal{L}_{RE}$
- **průnik**
 - $\{ \langle M \rangle \mid M \text{ je úplný TS} \} \cap \text{co-}\{ \langle M \rangle \mid M \text{ je úplný TS} \} = \emptyset \notin 2^{\Sigma^*} \setminus \mathcal{L}_{RE}$
- **doplňěk**
 - $\text{co-}(\text{co-HP}) \notin 2^{\Sigma^*} \setminus \mathcal{L}_{RE}$
- **rozdíl**
 - $\text{co-HP} \setminus \text{co-HP} = \emptyset \notin 2^{\Sigma^*} \setminus \mathcal{L}_{RE}$
- **konkatenaci**
 - $(\{ \langle M \rangle \mid M \text{ je úplný TS} \} \cup \{ \epsilon \}) \cdot (\text{co-}\{ \langle M \rangle \mid M \text{ je úplný TS} \} \cup \{ \epsilon \}) = \Sigma^* \notin 2^{\Sigma^*} \setminus \mathcal{L}_{RE}$
- **iteraci**
 - $(\text{co-HP} \cup \Sigma)^* = \Sigma^* \notin 2^{\Sigma^*} \setminus \mathcal{L}_{RE}$
- **pozitivní iteraci**
 - $(\text{co-HP} \cup \Sigma)^+ = \Sigma^+ \notin 2^{\Sigma^*} \setminus \mathcal{L}_{RE}$

Nerozhodnutelné problémy. Pro tuto třídu není rozhodnutelný žádný výše z uvedených problémů.

Shrnutí

Uzávěrové vlastnosti	$\mathcal{L}_{\text{FIN}}$	$\mathcal{L}_3$	$\mathcal{L}_{\text{DZA}}$	$\mathcal{L}_2$	$\mathcal{L}_1$	$\mathcal{L}_{\text{REC}}$	$\mathcal{L}_{\text{RE}}$
sjednocení							
průnik							
rozdíl							
doplňěk							
konkatenace							
iterace							
pozitivní iterace							
reverzace							
průnik s $\mathcal{L}_3$							
nekonečné sjednocení							
nekonečný průnik							

rozhodnutelné problémy	$\mathcal{L}_{\text{FIN}}$	$\mathcal{L}_3$	$\mathcal{L}_{\text{DZA}}$	$\mathcal{L}_2$	$\mathcal{L}_1$	$\mathcal{L}_{\text{REC}}$	$\mathcal{L}_{\text{RE}}$
náležitost							 částečně
prázdnost							
neprázdnost					 částečně	 částečně	 částečně
univerzalita							

	triviálně						
konečnost	 triviálně						
inkluze							
ekvivalence							
<i>regularita</i>	 triviálně	 triviálně					

Pokud v textu najdete chybu, nebudete něčemu rozumět nebo budete mít dojem, že by bylo vhodné něco doplnit, kontaktujte na discordu uživatele kocotom.