1 Základy

Množina je souhrn objektů, které jsou přesně určené a rozlišitelné. Tyto objekty nazýváme **prvky** množiny.

Potenční množina (ang. Power set) 2^X je množina všech podmnožin množiny X.

Relace je podmnožinou kartézského součinu (je to tedy množina n-tic), *binární relace* $\oplus$ se pak většinou namísto $\oplus(a,b)$ značí $a \oplus b$.

Reflexifní vlastnost relace: $\forall x \in A : x \oplus x$. Irreflexivní pak analogicky.

Symetrická vlastnost relace: $\forall x, y \in A : x \oplus y \Rightarrow y \oplus x$.

Antisymetrická vlastnost relace: $\forall x, y \in A : x \oplus y \wedge y \oplus x \Rightarrow x = y$.

Tranzitivní vlastnost relace: $\forall x, y, z \in A : \ x \oplus y \wedge y \oplus z \Rightarrow x \oplus z$.
Relace ekvivalence je reflexivní, tranzitivní a symetrická.
Relace uspořádání je reflexivní, tranzitivní a antisymetrická.

X uzávěr relace je nejmenší nadmnožina relace taková, aby splňovala vlastnost X.

Abeceda je konečná množina, jejíž prvky nazýváme symboly, značíme Σ.

Σ^* je volný monoid nad abecedou Σ generovaný operací konkatenace, tedy množina všech konečných posloupností w tvaru $w = a_1 \dots a_n \in \Sigma$ pro $i = 1, \dots, n$, také zvaná **iterace množiny** Σ. Σ^+ je tzv. *pozitivní iterace množiny* Σ.

Řetězec nad abecedou Σ je prvek množiny Σ^*. Délka řetězce je $|w| = n$. Řetězec s nulovou délkou značíme ε, tedy **prázdný řetězec**.

Konkatenací rozumíme binární operaci $\cdot$ nad dvěma rětězci: $w_1 \cdot w_2 = a_1 \dots a_n b_1 \dots b_m$, kde $w_1 = a_1 \dots a_n$ a $w_2 = b_1 \dots b_m$. Tato operace je asociativní, ε je jednotkovým prvkem vzhledem ke konkatenaci.

Dále zavádíme pojmy **(vlastní) prefix, (vlastní) sufix, (vlastní) podřetězec, reverze řetězce**.

Formální jazyk je jakákoli podmnožina $L \subseteq \Sigma^*$.

Konkatenací jazyků rozumíme binární operaci $\cdot$: $L_1 \cdot L_2 = \{xy \mid x \in L_1 \wedge y \in L_2\}$.

Iteraci jazyka (a pozitivní iteraci jazyka) definujeme takto:

$$\begin{aligned}
L^0 &= \{\varepsilon\} \\
L^n &= L \cdot L^{n-1} \mid n \geq 1 \\
L^* &= \textstyle\bigcup_{n \geq 0} L^n \\
L^+ &= \textstyle\bigcup_{n \geq 1} L^n
\end{aligned}$$

Gramatika G je čtveřice $G = (N, T, P, S)$, kde:

- N je konečná množina *nonterminálních symbolů* (nonterminálů)
- T je konečná množina *terminálních symbolů* (terminálů)
- P je konečná množina *přepisovacích pravidel*, definovaná jako podmnožina kartézského součinu $(N \cup T)^* N (N \cup T)^* \times (N \cup T)^*$.
- $S \in N$ je počáteční (výchozí, startovací) neterminál

Prvek $(\alpha, \beta) \in P$ je *přepisovací pravidlo*, zapisujeme ve tvaru $\alpha \rightarrow \beta$, kde α je levá strana a β je pravá strana pravidla.

Pro $\sigma, \mu \in (N \cup T)^*$ platí binární relace $\Rightarrow$, zvaná přímá derivace, pokud můžeme σ, μ zapsat ve tvaru $\sigma = \gamma\alpha\delta$, $\mu = \gamma\beta\delta$ a $\alpha \rightarrow \beta \in P$. Pak píšeme $\sigma \Rightarrow \mu$.

Relace $\Rightarrow^+$ se nazývá *relace derivace* a je tranzitivním uzávěrem relace $\Rightarrow$. $\Rightarrow^*$ je tranzitivní a reflexivní uzávěr relace $\Rightarrow$.

Větnou formu definujeme jako $\alpha \in (N \cup T)^* \mid S \Rightarrow^* \alpha$. Větná forma obsahující pouze terminály se nazývá *věta*.

Jazyk generovaný gramatikou G je definován jako $L(G) = \{w \mid S \Rightarrow^* w \wedge w \in \Sigma^*\}$.

2 Chomského klasifikace gramatik a formálních jazyků

Klasifikace je definována podle tvaru přepisovacích pravidel příslušných gramatik:

- Typ 0 – obecné (neomezené gramatiky) – rekurzivně vyčíslitelné jazyky – Turingovy stroje

$$\alpha \rightarrow \beta \mid \alpha \in (N \cup T)^* N (N \cup T)^*, \beta \in (N \cup T)^*$$

- Typ 1 – kontextové gramatiky – kontextové jazyky – lineárně omezené automaty

$$\alpha A \beta \rightarrow \alpha\gamma\beta \mid A \in N, \alpha, \beta \in (N \cup T)^*, \gamma \in (N \cup T)^+$$

$$\text{nebo } S \rightarrow \varepsilon$$

- Typ 2 – bezkontextové gramatiky – bezkontextové jazyky – zásobníkové automaty

$$A \rightarrow \alpha \mid A \in N, \alpha \in (N \cup T)^*$$

- Typ 3 – pravě/levě lineární gramatiky – regulární jazyky – konečné automaty

$$A \rightarrow xB, A \rightarrow x \mid A, B \in N, x \in T^+$$

3 Konečné automaty, varianty, minimalizace

Nedeterministický konečný automat je pětice $M = (Q, \Sigma, \delta, q_0, F)$, kde:

- Q je konečná množina stavů
- Σ je vstupní abeceda
- δ je přechodová funkce (množina pravidel) tvaru $\delta : Q \times \Sigma \rightarrow 2^Q$
- $q_0 \in Q$ je počáteční stav automatu
- $F \subseteq Q$ je množina koncových stavů

Je-li $\delta : Q \times \Sigma \rightarrow Q \cup \{nedef.\}$, (tedy $|\delta(q, a)| = 1$) jedná se o **deterministický konečný automat**. Automat lze také reprezentovat graficky nebo tabulkou.

Konfigurace C konečného automatu je dvojice $C = (q, w) \mid (q, w) \in Q \times \Sigma^*$. Počáteční konfigurace je tedy (q_0, w) a koncová (q_f, ε), kde $q_f \in F$.

Přechod automatu je binární relace $\vdash$ v množině konfigurací, tedy $\vdash \subseteq C \times C$ a je definován jako:

$$(q, aw) \vdash (q', w) \Leftrightarrow q' \in \delta(q, a) \mid q, q' \in Q, a \in \Sigma, w \in \Sigma^*$$

Jazyk přijímaný automatem M je $L(M) = \{w \mid (q_0, w) \vdash^* (q_f, \varepsilon) \wedge q_f \in F\}$.

NKA a DKA jsou ekvivalentní, NKA lze převést na DKA.

Každá pravá lineární gramatika s pravidly $A \to aB|a$ je převoditelná na gramatiku $G = (N, T, P, S)$ s pravidly $A \to aB|\varepsilon$. Taková gramatika je jednoduše převoditelná na NKA $M = (Q, \Sigma, \delta, q_0, F)$: $Q = N$, $\Sigma = T$, $B \in \delta(A, a)$ pokud $A \to aB \in P$, $q_0 = S$ a $F = \{A | A \to \varepsilon \in P\}$.

Každý DKA (a tedy i NKA) je převoditelný na gramatiku typu 3 takto: Pokud $\delta(q, a) = r$, pak $q \to ar \in P$. Pokud $p \in F$, pak $p \to \varepsilon \in P$.

Úplný KA má definován přechod pro všechny symboly abecedy u každého stavu.

4 Regulární množiny, výrazy, jazyky a vlastnosti

Regulární množiny nad abecedou Σ definujeme rekurzivně:

1. $\emptyset$ je regulární množina nad Σ
2. $\{\varepsilon\}$ je regulární množina nad Σ
3. $\{a\}$ je regulární množina nad Σ pro všechny $a \in \Sigma$
4. Jsou-li P a Q regulární množiny nad Σ, pak také
   - $P \cup Q$
   - $P.Q$
   - P^*

   jsou regulární množiny nad Σ.

Regulární výrazy nad abecedou Σ definujeme rekurzivně:

1. $\emptyset$ je regulární výraz označující regulární množinu $\emptyset$
2. ε je regulární výraz označující regulární množinu $\{\varepsilon\}$
3. a je regulární výraz označující regulární množinu $\{a\}$ pro všechny $a \in \Sigma$
4. Jsou-li p a q regulární výrazy označující regulární množiny P a Q, pak
   - $(p + q)$ je regulární výraz označující regulární množinu $P \cup Q$
   - $p.q$ je regulární výraz označující regulární množinu $P.Q$
   - p^* je regulární výraz označující regulární množinu P^*

Rovnice nad regulárními výrazy jsou rovnice obsahující koeficienty a neznámé reprezentující regulární výrazy. Například řešením, tedy *nejmenším pevným bodem*, rovnice $X = pX + q$ je $X = p^*q$.

Soustavy rovnic nad regulárními výrazy jsou základním prostředkem pro *převod KA na RV*.

Rozšířený konečný automat je dalším základním prostředkem pro *převod RV na KA*. Pro RKA platí $\delta \in Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$. Je tedy rozšířen o ε-přechody. Převod RV na RKA již je jednoduchý (např. grafickou cestou).

Vlastnosti regulárních jazyků dělíme na *strukturální*, *uzávěrové* a *rozhodnutelné problémy*

Strukturální vlastnosti RJ: Konečnost (každý konečný jazyk je regulární), Pumping lemma

Uzávěrové vlastnosti RJ: Třída RJ je uzavřena vůči sjednocení, konkatenaci a iteraci (z definice RM a RV).

Rozhodnutelné problémy RJ: Neprázdnost, náležitost, ekvivalence (přes KA)

Pumping lemma: Pro každý nekonečný RJ L existuje celočíselná konstanta p taková, že pro všechny řetězce z jazyka L délky větší nebo rovné p platí:

$$w = xyz \wedge 0 < |y| \leq p \Rightarrow xy^iz \in L \mid i \geq 0$$

Této vlastnosti RJ je možno využít při důkazu sporem, že daný jazyk není regulární.

5 Transformace a normální formy bezkontextových gramatik

Bezkontextová gramatika je gramatika s pravidly tvaru $P: \; N \times (N \cup T)^*$.

Víceznačná věta je věta, kterou lze vygenerovat alespoň dvěma různými derivačními stromy (postupy).

Jednoznačná CFG je CFG která neobsahuje víceznačné věty.

Ekvivalentní gramatiky G_1 a G_2 splňují podmínku $L(G_1) = L(G_2)$, ale mohou se lišit například v nedostupných a zbytečných symbolech.

Nedostupný symbol X je takový, že neexistuje derivace $S \Rightarrow^* \alpha X \beta$.

Zbytečný symbol X je takový, že neexistuje derivace $S \Rightarrow^* \alpha X \beta \Rightarrow^* xyz$.

Vlastní gramatika je gramatika bez zbytečných symbolů, zbytečných pravidel, ε-pravidel a cyklů.

Chomského normální forma je forma gramatiky s pravidly tvaru $A \to BC \mid a$ a

Greibachové normální forma je forma gramatiky s pravidly tvaru $A \to a\alpha$, kde $\alpha \in N^*$, pokud $\varepsilon \in L(G)$, tak také $S \to \varepsilon$ je platné pravidlo.

6 Zásobníkové automaty, varianty

Zásobníkový automat je sedmice $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, kde

- Q je konečná množina stavů automatu
- Σ je vstupní abeceda
- Γ je zásobníková abeceda
- δ je přechodová funkce tvaru $\delta: \; Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$
- $q_0 \in Q$ je počáteční stav automatu

- $Z_0 \in \Gamma$ je počáteční symbol na zásobníku
- $F \subseteq Q$ je množina koncových stavů

Konfigurace C zásobníkového automatu M je $C: Q \times \Sigma^* \times \Gamma^*$

Přechod zásobníkového automatu $\vdash$ je definován jako:

$$(q, aw, Z\gamma) \vdash (q', w, \beta\gamma) \Leftrightarrow (q', \beta) \in \delta(q, a, Z)$$

Pokud $a = \varepsilon$, pak se jedná o ε-přechod.

Jazyk přijímaný ZA je $L(M) = \{w \mid (q_0, w, Z_0) \vdash^* (q_f, \varepsilon, \gamma) \wedge q_f \in F\}$.

Typy přijetí jazyka: Vyprázdněním zásobníku, přejitím do koncového stavu, obojím

Rozšířený zásobníkový automat je ZA s přechodovou funkcí: $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^* \rightarrow 2^{Q \times \Gamma^*}$. RZA a ZA jsou ekvivalentně silné.

Deterministický zásobníkový automat je ZA, pro který platí, že $\forall a \in \Sigma: |\delta(q, a, z)| \leq 1 \wedge \delta(q, \varepsilon, z) = \emptyset$ nebo $\forall a \in \Sigma: \delta(q, a, z) = \emptyset \wedge |\delta(q, \varepsilon, z)| \leq 1$.

DZA mají strikně menší vyjadřovací sílu než ZA. ($L = \{ww^R\}$)

7 Vlastnosti bezkontextových jazyků

Pumping lemma: Pro každý BJ L existuje celočíselná konstanta p taková, že pro všechny řetězce z jazyka L délky větší nebo rovné p platí:

$$z = uvwxy \wedge vx \neq \varepsilon \wedge |vwx| \leq p \wedge uv^i wx^i y \in L \mid i \geq 0$$

Této vlastnosti BJ je možno využít při důkazu sporem, že daný jazyk není bezkontextový.

Uzávěrové vlastnosti: Uzavřeny vůči substituci a morfismu, inverznímu morfismu, sjednocení, konkatenaci, iteraci, průniku s RJ. BJ jsou *neuzavřeny* vůči průniku a doplňku

Rozhodnutelné problémy: Neprázdnost, náležitost, konečnost

Nerozhodnutelné problémy: Ekvivalence jazyků dvou gramatik, inkluze jazyků dvou gramatik

8 Turingovy stroje

Turingův stroj je šestice $M = (Q, \Sigma, \Gamma, \delta, q_0, q_F)$, kde

- Q je konečná množina stavů TS
- Σ je vstupní abeceda TS ($\Delta \notin \Sigma$)
- Γ je pásková abeceda TS ($\Sigma \subset \Gamma \wedge \Delta \in \Gamma$)
- δ je přechodová funkce tvaru $\delta: (Q - \{q_F\}) \times \Gamma \rightarrow Q \times (\Gamma \cup \{L, R\})$, kde $L, R \notin \Gamma$
- $q_0 \in Q$ je počáteční stav TS
- $q_F \in Q$ je koncový stav TS

Symbol Δ je tzv. prázdný symbol (blank). Označuje nepoužité místo pásky, ale může být opět vložen.

Páska je nekonečný řetězec symbolů. V základní definici TS má páska pevný počátek a pokračuje směrem doprava.

Konfigurace pásky je kombinace obsahu pásky a pozice hlavy TS: $H: \{\gamma\Delta^\omega \mid \gamma \in \Gamma^*\} \times \mathbb{N}$

Konfigurace TS je pak dána stavem řízení a konfigurací pásky, tedy $C: Q \times H$

γ_n označuje n-tý symbol na pásce a $s_a^n(\gamma)$ je záměna znaku γ_n na pásce za symbol a

Krok výpočtu TS definujeme jako binární relaci $\vdash$, takovou, že platí:

- $q_1, q_2 \in Q, \gamma \in \Gamma^\omega, n \in \mathbb{N}, b \in \Gamma$
- $(q_1, \gamma, n) \vdash (q_2, \gamma, n+1)$ pro $\delta(q_1, \gamma_n) = (q_2, R)$ je operace posunu doprava
- $(q_1, \gamma, n) \vdash (q_2, \gamma, n-1)$ pro $\delta(q_1, \gamma_n) = (q_2, L) \wedge n > 0$ je operace posunu doleva
- $(q_1, \gamma, n) \vdash (q_2, s_a^n(\gamma), n)$ pro $\delta(q_1, \gamma_n) = (q_2, a)$ je operace zápisu symbolu a

Výpočet TS je posloupnost konfigurací $C_0, C_1, \ldots$, kde $\forall i \geq 0 : C_i \vdash C_{i+1}$ a která je buď *nekonečná* (TS nezastaví a nepřijme) nebo *konečná* s koncovou konfigurací $C_F : (q, \gamma, n)$. Pokud $q = q_F$, pak je zastavní *normální* (a TS přijme), nebo je zastavení *abnormální* (pak nepřijme). Abnormální zastavení je vyvoláno vyjetím mimo levý okraj pásky nebo nedefinovanou přechodovou funkcí pro aktuální konfiguraci.

Přijímání zvláštní konfigurací pásky – alternativně je možné definovat ukončení výpočtu jako přechod od konfigurace $\Delta w \Delta^\omega$ na $\Delta Y \Delta \ldots$, kde $Y \in \Gamma - \Sigma$.

Jazyk přijímaný TS M je definován jako $L(M) = \{w \mid w \text{ je přijat TS } M\}$. *Řetězec je přijat TS*, jestliže při aktivaci z počáteční konfigurace zastaví přechodem do koncového stavu, tedy platí, že $(q_0, \Delta w \Delta^\omega, 0) \vdash^* (q_F, \gamma, n)$ pro nějaké $\gamma \in \Gamma^* \wedge n \in \mathbb{N}$

Vícepáskový TS je TS s k páskami, kde každá má svou abecedu $\Gamma_1 \ldots \Gamma_k$ a s k odpovídajících hlavami. Každý vícepáskový TS je převoditelný na jednopáskový (symbol pásky je složený z pásek a pozic hlav vícepáskového).

Nedeterministický TS je TS jehož přechodová funkce má tvar $\delta : (Q - \{q_F\}) \times \Gamma \rightarrow 2^{Q \times (\Gamma \cup \{L, R\})}$

Jazyk NTS je pak množina řetězců pro které NTS *může zastavit* přechodem do q_F

Pro každý NTS existuje ekvivalentní TS.

9 Lineárně omezené automaty

Lineárně omezený automat je nedeterministický TS, který nikdy neopustí tu část pásky, na které je zapsán jeho vstup (formálněji: zavedeme páskový symbol, který nelze překročit ani přepsat).

Deterministický LOA je deterministický TS, který nikdy neopustí tu část pásky, na které je zapsán jeho vstup.

Není známo, zda DLOA je či není slabší než NLOA

Třída kontextových jazyků odpovídá třídě jazyků přijímaných LOA

Třída kontextových jazyků je uzavřena vůči průniku, sjednocení, konkatenaci, iteraci a doplňku. Lze rozhodnout náležitost, nelze rozhodnout inkluzi a prázdnost.

Každý kontextový jazyk je rekurzivní (viz. dále), ale nemusí platit naopak.

10 Nerozhodnutelnost

Úplný turingův stroj pro každý vstup zastaví.

Rekurzivně vyčíslitelný jazyk je přijímán nějakým TS.

Rekurzivní jazyk je přijímán nějakým *úplným* TS (rozhoduje tento jazyk).

Rekurzivní i rek. vyčíslitelné jazyky jsou uzavřeny vůči sjednocení, průniku, konkatenaci a iteraci. Rekurzivní pak také vůči doplňku.

Rozhodovací problém P je chápán jako funkce f_P s oborem hodnot $\{true, false\}$, specifikována jazykem L_P nad Σ

Problém je rozhodnutelný, pokud L_P je rekurzivní jazyk (protože TS *vždy* zastaví/rozhodne)

Problém je částečně rozhodnutelný, pokud L_P je rekurzivně vyčíslitelný (protože TS *může* zastavit/rozhodnout)

Problém je nerozhodnutelný, pokud není rozhodnutelný, ale může být zároveň částečně rozhodnutelný

Jazyky přijímané TS jsou jazyky typu 0.

Jazyky mimo třídu 0, jsou jazyky, které nelze přijímat ani TS. Pro každou abecedu takový jazyk existuje. Množina všech TS je totiž spočetná, ale množina 2^{Σ^*} není.

11 Univerzální TS

Kódování TS je proces zakódování všech stavů, symbolů pásky a přechodové funkce. Z konvence kódujeme do posloupností nul oddělených jedničkami. Stavy i symboly uspořádáme a kódujeme postupně (ε, 0, 00,...,0^ω). Přechody kódujeme jako čtveřice (p, x, q, y), kde $\delta(p, x) = (q, y)$.

Univerzální TS je koncept stroje, který umožňuje na vstupu specifikovat jak TS, tak jeho data, a tento TS odsimulovat (odděleno speciálním znakem). Může být implementován jako 3-páskový TS (vstup/výstup, simulace pásky daného TS, stav daného TS).

12 Problém zastavení TS

Halting problem je problém, zda daný TS při daném vstupu zastaví, *není rozhodnutelný*, je *částečně rozhodnutelný*. Nerozhodnutelnost lze dokázat diagonalizací, částečnou rozhodnutelnost upraveným TS, který necyklí ale skončí ve speciálním stavu.

Diagonalizace při HP – řádky indexujeme TS $M_\varepsilon, M_0, M_1, M_{00}, M_{01}, \ldots$, tedy všemi možnými TS nad $\Sigma = \{0, 1\}$ a sloupce všemi možnými řetězci nad Σ. Tabulku vyplníme hodnotami $\{zastavi, cykli\}$, pro tuto tabulku existuje úplný TS, kterému se zadají indexy (TS a vstup) a ten pak přijme nebo odmítne, pokud daný TS skončí nebo cyklí. Pak sestavíme TS N, který sestaví vstup pro úplný TS tvaru $M_x\#x$, ale invertuje jeho výsledek. Takto získané výsledky jsou komplementem diagonály předchozí tabulky. Takový TS se ale liší se všemi uvedenými v tabulce, tabulka pak neobsahuje všechny TS, což je spor, takže tabulka všechny neobsahuje a tedy množina není spočetná, problém není rozhodnutelný.

13 Redukce

Redukce spočívá v technice, kdy známý jazyk A (který je/není rekurzivní popř. rekurzivně vyčíslitelný) převedeme (redukujeme) pomocí úplného TS na jazyk B, který zkoumáme. Takto dokážeme, že jazyk B rovněž je/není rekurzivní popř. rekurzivně vyčíslitelný.

Problém náležitosti (MP) lze redukovat na HP a tím dokázat nerozhodnutelnost. Libovolný TS lze upravit na TS, který přijme, pokud původní TS zastaví (stačí dodat chybějící přechody a ochránit levý okraj). Pak jde o problém zastavení a ten není rekurzivní.

14 Postův korespondenční problém

Postův systém je neprázdný seznam dvojic neprázdných řetězců $\alpha, \beta \in \Sigma^+$. $S = (\alpha_1, \beta 1), \ldots, (\alpha_k, \beta_k)$.

Řešením postova systému je neprázdná posloupnost přirozených čísel $i_1, \ldots, i_n$, která označuje pořadí použití jednotlivých dvojic tak, aby platilo $\alpha_{i_1} \ldots \alpha_{i_n} = \beta_{i_1} \ldots \beta_{i_n}$.

PCP je nerozhodnutelný a také všechny problémy z něj redukované.

15 Primitivně rekurzivní funkce

Vyčíslitelné funkce jsou spočítatelné, značíme $f: \mathbb{N}^m \to \mathbb{N}^n$

Totální funkce – můžeme vybrat jakoukoli hodnotu z def. oboru

Parciální funkce – def. obor je více omezen

Počáteční funkce je soubor předem definovaných funkcí jako základní kameny vyšších funkcí:

- Nulová funkce – $\xi: \emptyset \to 0$, $\xi() = 0$
- Funkce následníka – $\sigma: \mathbb{N} \to \mathbb{N}$, $\sigma(x) = x + 1$
- Projekce – $\pi_k^n: \mathbb{N}^n \to \mathbb{N}$, $\pi_2^3(7, 6, 4) = 6$

Způsob tvorby nových funkcí:

- **Kombinace** – pro $f: \mathbb{N}^k \to \mathbb{N}^m$ a $g: \mathbb{N}^k \to \mathbb{N}^n$ je $f \times g: \mathbb{N}^k \to \mathbb{N}^{m+n}$, $f \times g(\bar{x}) = (f(\bar{x}), g(\bar{x}))$
- **Kompozice** – pro $f: \mathbb{N}^k \to \mathbb{N}^m$ a $g: \mathbb{N}^m \to \mathbb{N}^n$ je $g \circ f: \mathbb{N}^k \to \mathbb{N}^n$, $g \circ f(\bar{x}) = g(f(\bar{x}))$
- **Primitivní rekurze** – pomocí fcí $g: \mathbb{N}^k \to \mathbb{N}^m$ a $h: \mathbb{N}^{k+m+1} \to \mathbb{N}^m$ sestrojíme fci $f: \mathbb{N}^{k+1} \to \mathbb{N}^m$
  $f(\bar{x}, 0) = g(\bar{x})$
  $f(\bar{x}, y + 1) = h(\bar{x}, y, f(\bar{x}, y))$

Primitivní rekurzivní funkce jsou všechny funkce vytvořené z počátečních funkcí pomocí kombinace, kompozice a primitivní rekurze.

Každá primitivní rekurzivní funkce je *totální funkcí.*

Konstantní funkce přiřadí libovolné n-tici konstantní hodnotu $\kappa_m^0 = \sigma \circ \sigma \circ \ldots \circ \xi()$, kde σ je uplatněna m-krát.

Funkce mimo primitivně rekurzivní třídu mohou být stále vyčíslitelné, Jsou to všechny strikně parciální funkce, ale i některé totální. (μ-rekurzivní funkce).

Parciálně rekurzivní funkce jsou funkce tvořeny z počátečních funkcí pomocí kombinace, kompozice, primitivní rekurze a *minimalizace*.

Minimalizace je technika vytvoření funkce $f: \mathbb{N}^n \to \mathbb{N}$ z funkce $g: \mathbb{N}^{n+1} \to \mathbb{N}$ předpisem $f(\bar{x}) = \mu y[g(\bar{x}, y) = 0]$. Hledáme *nejmenší* y takové, aby platilo $g(\bar{x}, y) = 0$ a $g(\bar{x}, z)$ je definováno pro $\forall z < y$.

Turingovsky vyčíslitelná funkce je parciální funkce, kterou může počítat nějaký TS. Každá parciálně rekurzivní funkce je turingovsky vyčíslitelná.

16 Časová a paměťová složitost

Church-Turingova teze říká, že každý algoritmus je implementovatelný jistým TS a můžeme tedy složitost algoritmu chápat jako složitost TS (čas a prostor).

Rozlišujeme analýzu složitosti *časové a prostorové* nebo *nejlepšího, nejhoršího a průměrného případu*.

Časová složitost je měřena jako počet kroků (přechodů) TS provedených při výpočtu.

Prostorová složitost je měřena jako počet okének pásky potřebných k výpočtu.

Je-li časová složitost výpočtu n, pak prostorová složitost nemůže být větší než $n + 1$.

Složitost je možné analyzovat i mimo prostředí TS, je důležité najít *cenové kritérium*.

17 Asymptotická ohraničení složitostí

Při popisu složitosti se vylučuje vliv aditivních a multiplikativních konstant, proto se používají *asymptotické odhady*.

Asymptotické horní omezení $O(f(n)) = \{g(n) \mid \exists c, n_0 : \forall n \geq n_0 \Rightarrow 0 \leq g(n) \leq c.f(n)\}$

Asymptotické dolní omezení $\Omega(f(n)) = \{g(n) \mid \exists c, n_0 : \forall n \geq n_0 \Rightarrow 0 \leq c.f(n) \leq g(n)\}$

Asymptotické oboustranné omezení $\Theta(f(n)) = \{g(n) \mid \exists c_1, c_2, n_0 : \forall n \geq n_0 \Rightarrow c_1 f(n) \leq g(n) \leq c_2.f(n)\}$

18 Třída P a NP problémů

Také problémy (nejen algoritmy) lze kategorizovat do tříd složitosti, snažíme se je zařadit do nejnižší možné třídy.

Třídy složitosti NTS a DTS:

- $DTime[t(n)] = \{L \mid \exists k - tapeDTS\ M :\ L = L(M) \wedge T_M = O(t(n))\}$
- $NTime[t(n)] = \{L \mid \exists k - tapeNTS\ M :\ L = L(M) \wedge T_M = O(t(n))\}$
- $DSpace[t(n)] = \{L \mid \exists k - tapeDTS\ M :\ L = L(M) \wedge S_M = O(s(n))\}$
- $NSpace[t(n)] = \{L \mid \exists k - tapeNTS\ M :\ L = L(M) \wedge S_M = O(s(n))\}$

Časově zkonstruovatelná funkce – pokud existuje TS, který pro libovolný vstup w zastaví přesně po $t(|w|)$ krocích

Prostorově zkonstruovatelná funkce – pokud existuje TS, který pro libovolný vstup w zastaví s využitím přesně $s(|w|)$ okének pásky

Nejčastější třídy složitosti:

- **P:** $\bigcup_{k=0}^{\infty} DTime(n^k)$ **NP:** $\bigcup_{k=0}^{\infty} NTime(n^k)$
- **PSPACE:** $\bigcup_{k=0}^{\infty} DSpace(n^k)$ **NPSPACE:** $\bigcup_{k=0}^{\infty} NSpace(n^k)$
- **LOGSPACE:** $\bigcup_{k=0}^{\infty} DSpace(k.log(n))$ **NLOGSPACE:** $\bigcup_{k=0}^{\infty} NSpace(k.log(n))$
- **EXP:** $\bigcup_{k=0}^{\infty} DTime(2^{n^k})$ **NEXP:** $\bigcup_{k=0}^{\infty} NTime(2^{n^k})$
- **k-EXP:** $\bigcup_{l=0}^{\infty} DTime(2^{2^{\cdot^{\cdot^{2^{n^l}}}}})$ **k-NEXP:** $\bigcup_{l=0}^{\infty} NTime(2^{2^{\cdot^{\cdot^{2^{n^l}}}}})$ kde k je výška sloupce
- **ELEMENTARY:** $\bigcup_{k=0}^{\infty}$ k-EXP

Vrchol hierarchie složitosti – třída primitivně rekurzivních funkcí **PR**, přída rekurzivních funkcí **R** a třída rekurzivně vyčíslitelných funkcí **RE**

19 NP-úplnost

Jazyk je $\mathcal{R}$ redukovatelný pokud existuje třída funkcí $\mathcal{R}$ a v ní funkce f taková, že $w \in L_1 \Leftrightarrow f(w) \in L_2$, pak platí $L_1 \leq_{\mathcal{R}}^{m} L_2$

Jazyk L_0 je $\mathcal{C}$ těžký, pokud existuje třída funkcí $\mathcal{R}$ a třída jazyků $\mathcal{C}$ a $\forall L \in \mathcal{C}: \; L \leq_{\mathcal{R}}^{m} L_0$

Jazyk L_0 je $\mathcal{C}$ úplný, jestliže $L_0 \in \mathcal{C}$ a L_0 je $\mathcal{C}$ těžký.

Nejběžnější typy úplnosti: NP, PSPACE, EXP, P, NLOGSPACE, NEXP

20 SAT problém

Jedná se o NP-úplný problém.

Problém, zda je množina klausulí v konjuktivní normální formě splnitelná ($v \vee -x$, $x \vee -y \vee v, \dots$).

Klausule zakóduje jako posloupnost 0 a pouze na pořadovém místě klausule bude obsahovat p nebo n podle toho zda obsahuje literál kladný nebo záporný.

Pak se provede test, zada určité ohodnocení literálů je splnitelné pro dané klausule.

Lze zkonstruovat NTS, který problém řeší v polynomiálním čase.