

Vyčíslitelnost

Doc. RNDr. Antonín Kučera, CSc.

1. července 2021

Obsah

1	Zkratky	2
2	Uvod	2
2.1	Historická vsuvka	2
2.2	Terminologie	3
3	Rekurzivně spočetné množiny a predikáty	7
3.1	10 Hilbertův problém	8
3.2	Selektory	9
3.3	Imunní množiny	10
4	Věty o rekurzi	10
5	Produktivní množiny	14
6	Dvojice množin	19
7	Gödelovy věty	22
7.1	Kalibrace síly teorie	24
8	Relativní vyčíslitelnost	24
8.1	Formalizace relativního vypočtu	25
8.2	Struktura T-stupňu	28
8.3	Relativizace dřívějších výsledků	28
8.4	Operace skoku	29
8.5	Stejnoměrnost	31
9	Limitní vyčíslitelnost	31
9.1	Limitní vyčíslitelnost pro ORF	31
9.2	Obecná limitní vyčíslitelnost	33
10	Aritmetická hierarchie	35
10.1	Numerace	37
11	Pokročilejší vyčíslitelnost	41
11.1	R.S. množiny	41
11.2	Forcing	41
11.3	Algoritmická náhodnost	44
11.4	Kolmogorovská složitost	44
11.5	Martingale	44

1 Zkratky

1. ČRF - částečně rekurzivní funkce.
2. ORF - obecně rekurzivní funkce.
3. PRF - primitivně rekurzivní funkce.
4. r.s. - rekurzivně spočetná.
5. ZAS - základní aritmetika síla.
6. PA - Peano Aritmetika.
7. ORP - obecně rekurzivní predikát.
8. PNF - prenexní normální tvar.

2 Uvod

2.1 Historická vsuvka

Hilbertův 10. problém, úplnost aritmetiky 1900. Gödel dokázal že nejde. V prvním větě použil *Primitivně rekurzivní funkce*.

Definice 2.1 (Primitivně rekurzivní funkce). Primitivně rekurzivní funkce - podmnožina efektivně vyčíslitelných funkcí, jsou všude definované.

Tyto ale nestačí pro hlavní problém dokazatelnosti.

Při dalším vývoje se vyvinul kalkulus tzv obecně rekurzivních funkcí ORF a částečně rekurzivních funkcí ČRF.

Značení 2.2 (Definiční obor). $dom(\varphi)$ - definiční obor.

Značení 2.3 (Obor hodnot). $range(\varphi)$ - obor hodnot.

Definice 2.4 (Rekurzivní funkce). Funkce $f : \Sigma^* \rightarrow \Sigma^*$ je *částečně rekurzivní*, pokud je turingovsky vyčíslitelná (tedy existuje Turingův stroj M takový, že $\forall x \in dom(f) \Leftrightarrow M(x) \downarrow \wedge f(x)$ odpovídá obsahu pásky $M(x)$ na konci výpočtu).

Funkce $f : \Sigma^* \rightarrow \Sigma^*$ je *obecně rekurzivní*, je-li částečně rekurzivní a $dom(f) = \Sigma^*$ (všude definovaná).

Poznámka 2.5 (Church-Turing teze). Historický vývoj:

- A. Church vyvinul λ -konverze (λ -calculus) a dokázal, že neexistuje algoritmus tzv "rozhodovací". Lambda konverze jsou poměrně obtížné.
- Turing, nezávisle na Churchovi v roce 36 vyvinul Turing Machines a dokázal nevyčíslitelnost Halting problému.

Pak ostatní prohlásili Church-Turing teze, že všechno co je efektivně vyčíslitelné je Turingovský nebo λ -konverzi vyčíslitelné.

Definice 2.6 (λ -calculus). Necht C je množina konstant, necht V je (spočetná) množina proměnných. Množina tzv λ terms Λ je nejmenší množina tž:

- $C \subseteq \Lambda$.
- $V \subseteq \Lambda$
- necht $t_1, t_2 \in \Lambda$ termy, pak aplikace $t_1 t_2$ jako v Haskellu je taky term
- $t \in \Lambda, x \in V \Rightarrow \lambda x.t \in \Lambda$. V Haskellu:
 $(\backslash x \rightarrow t)$
 Což je funkce s parametrem x a vrací t .

Jako závěr, formální, efektivně dokazovací systém nemůže úplně popsat pravdu. Je mnohem složitější.

Poznámka 2.7 (System PRF(odbočka)). Funkcionální systém, postavený na axiomech:

- Základní funkce: 0, +1, id (resp vydělení i-té složky)
- 2 Odvozovací pravidla:
 - 1) substituce
 - 2) operátor primitivní rekurze. Jednoduše řečeno, výpočet v bode $(y + 1)$ uděláme rekurzivně z bodu "y".

Pak se vezme *tranzitivní uzávěr* - všechno co jde odvodit ze základních funkcí pomocí odvozovacích pravidel. Na rozdíl od ČRF nemáme **while**, čímž dostaneme jenom podmnožinu ČRF.

Substitute:

$$S(f, g_1, \dots, g_n) = f(g_1(y_1, \dots, y_n), \dots, g_n(y_1, \dots, y_n))$$

Primitivní rekurze:

Poznámka 2.8 (Kleeneho system ČRF). Pak přidáním operátoru μ a **while** k předchozímu systému dostaneme ČRF (znovu *tranzitivní uzávěr*). Je komplikovaný, je lepší používat nějaké dokazování.

Q: je možné, že existuje mnohem komplikovanější systém než všechny které máme v Church-Turingové téze, který by byl silnější z pohledu vyčíslitelnosti?

Q: Teorie vyčíslitelnosti zkoumá \mathbb{N} jako analýza začínala zkoumáním \mathbb{R} ?

2.2 Terminologie

Přibližně do roku 1990 převládala terminologie ORF, ČRF zavedená Kleene. Pak byla snaha změnit na **computable functions** - efektivně vyčíslitelné.

Definice 2.9 (Rekurzivní množina). Množina je rekurzivní, neboli rozhodnutelná (decidable, computable) - efektivně rozhodnutelná. Jednoduše řečeno, máme program, který na každém vstupu se vždy zastaví a rozhodne ANO nebo NE (jestli slovo patří do ní).

Definice 2.10 (Rekurzivní spočetná množina). Množina je rekurzivní spočetná (částečně rozhodnutelná), nebo computably enumerable. Formálně je definičním oborem nějakého programu (tzn částečně rekurzivní funkce, TS etc.).

Poznámka 2.11. Na rozdíl od kurzu ZSV, kde jsme definovali funkce $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$, budeme zkoumat funkce aritmetické.

$$f : \mathbb{N} \rightarrow \mathbb{N}, f : \mathbb{N}^k \rightarrow \mathbb{N}$$

Nemusí být všude definované.

Přístupy jsou ekvivalentní, protože můžeme očíslovat slova.

Poznámka 2.12. Formálně nemáme klasické k-tice jako vektory, ale kodujeme všechno do přirozených čísel. Je známá jako Cantorová metoda parování.

Značení 2.13 (Konvergence). Program **konverguje** - znamená že se zastaví za konečný počet kroků.

Připomenutí 2.14. Rekurzivita, rekurzivní spočetnost se zachovává na \cup, \cap . Rekurzivita je taky zachovaná při \neg (doplňek). Rekurzivní spočetnost nikoliv.

Věta 2.15 (Postova). L je rozhodnutelný $\iff L \wedge \bar{L}$ jsou c.r.

Důkaz. \Rightarrow . Z TS pro L sestavíme pro doplněk znegováním všech odpovědi.

\Leftarrow . Necht $L(M) = L \wedge L(B) = \bar{L}$, pak sestavíme TS pro rozhodnutí L .

1. Pust B, M paralelně
2. if(Acc(M, x))
3. accept
4. if(Acc(B, x))
5. reject

Pokud se aspoň 1 zacykli - reject. Paralelní spuštění lze implementovat pomoci 2 pasek, případně je slepit do 1. \square

Definice 2.16 (Gödelové číslo). Index programu. Necht φ je ČRF, P_e je program který ji vyčísluje. Pak index funkce φ je e .

Poznámka 2.17. Každá ČRF má nekonečně mnoho programu, takže i nekonečně mnoho indexu. Očíslování programu generuje očíslování funkci.

V jistém smyslu, nezáleží na konkrétním očíslování pokud je efektivní (nemáme čas toto dokazovat). 2 různé indexace jsou *efektivně ekvivalentní*.

Q: pokud zafixujeme "jazyk programování" má program jednoznačné očíslování? Q: jaký z jazyků programování je nejbližší k ČRF? Asi λ -calculus.

Připomenutí 2.18 (Univerzální TS). Dostane program M a data x , simuluje výpočet $M(x)$.

Pro nás to bude *univerzální ČRF*.

Definice 2.19 (Univerzální ČRF).

$$\Psi_n(e, x_1, x_2, \dots, x_n)$$

kde e je index programu, x_i jsou data.

Občas se značí

$$\varphi_e^n(x_1, \dots, x_n) \simeq e(x_1, \dots, x_n)$$

Značení 2.20 (Univerzální ČRF).

$$\varphi_e^n(x_1, \dots, x_n) \simeq U(\mu_y(T_n(x_1, \dots, x_n, y)))$$

- Kde T_n je primitivně rekurzivní predikát, který říká "za n kroků".
- U je primitivně rekurzivní funkce 1 proměnné. Který "vydělí" výsledek z mezivýsledků (jelikož máme všechno zakódované jako přirozená čísla).
- μ_y říká "nejmenší y ".

Věta 2.21 (s-m-n (BD)).

$$\varphi_e^{m+n}(x_1, \dots, x_n, y_1, \dots, y_m) \simeq \varphi_{s_n^m(e, x_1, \dots, x_n)}^n(y_1, \dots, y_m)$$

$V \Psi$ notace

$$\Psi_{n+m}(e, \bar{x}, \bar{y}) \simeq \Psi_m(s_n^m(e, \bar{x}, \bar{y}))$$

kde \bar{x}, \bar{y} jsou vektory pro kratší zápis.

Funkce $s_n^m : e, x_1, \dots, x_n$ vyrobí nový program. Ten čeká na vstup y_1, \dots, y_m , k tomu přidá zahardkodované data x_1, \dots, x_n a spustí na to e . Je to pouze syntaktická manipulace dat.

Definice 2.22 (e-ta rekurzivně spočetná množina).

$$W_e = \text{dom}(\varphi_e) = \{x : \varphi_e(x) \downarrow\} = \{\Psi_1(e, x) \downarrow\}$$

Poznámka 2.23. Rekurzivní spočetné funkce se definují jako obor hodnot ČRF.

M je rekurzivní množina \iff je oborem hodnot **rostoucí** ČRF.

M je rekurzivně spočetná množina \iff je oborem hodnot **prosté** ČRF.

Rozdíl v definici souvisí s Halting problémem.

Definice 2.24 (1,m převoditelnost). $A \leq_1 B \iff \exists \text{ ORF } f \text{ prostá:}$

$$x \in A \iff f(x) \in B$$

$A \leq_m B \iff \exists \text{ ORF } f \text{ (ne nutně prostá):}$

$$x \in A \iff f(x) \in B$$

Definice 2.25 (1-úplnost). Množina M je 1-úplná jestliže je rekurzivně spočetná každá rekurzivní spočetná je 1-převoditelná na M .

Definice 2.26 (K, DIAG).

$$K = \{x : x \in W_x\} = \{x : \varphi_x(x) \downarrow\} = \{x : \Psi_1(x, x) \downarrow\}$$

Taky

$$K_0 = \{\langle x, y \rangle : \varphi_x(y) \downarrow\} = \{\langle x, y \rangle : y \in W_x\}$$

Značení z ZSV

$$DIAG = \{\langle M \rangle : M \in L(M)\} = \{\langle M \rangle : M(\langle M \rangle)\}$$

Lemma 2.27 (DIAG, K). $DIAG$ je rekurzivně spočetný (částečně rozhodnutelný) ale není rekurzivní (rozhodnutelný).

Důkaz pomoci Cantorové diagonální metody.

Důkaz.

$$\overline{K} = \{x : x \notin W_x\}$$

W_x ale jsou všechny rekurzivně spočetné. Z toho

$$\forall W_x : \overline{K} \neq W_x$$

Takže \overline{K} není částečně rekurzivní. Dle Postové věty 2.15 K není rozhodnutelná. □

Věta 2.28 (K 1-úplná). *K (také K_0) je 1-úplná.*

Důkaz. Zavedeme ČRF

$$\alpha(x, y, w) \downarrow \iff \varphi_x(y) \downarrow$$

kde w je fiktivní proměnná. Je to ekvivalentní

$$\Psi_1(e, x, y, w)$$

použijeme 2.21

$$\Psi_1(e, x, y, w) \simeq \Psi_1(s_2^1(e, x, y), w) \simeq \varphi_{s_2^1(e, x, y)}(w)$$

dosadíme za $w = s_2^1(e, x, y)$.

Pomocí w se dostáváme na diagonálu. Pak

$$x \in W_y \iff s_2^1(e, x, y) \in K$$

□

Věta 2.29 (Rozšíření Ψ_n). *Ψ_n nemá obecně rekurzivní rozšíření. Jinými slovy neexistuje ORF h rozšíření Ψ takové že*

$$\forall x \in \text{dom}(\Psi_n) : \Psi_n(x) = h(x)$$

a h je definovaná pro vstupy mimo $\text{dom}(\Psi_n)$.

Dokonce, pokud α částečně rekurzivně rozšiřuje Ψ_n , tak najdeme vstup na kterém diverguje

$$\exists x_1 : \alpha(x_1, x_1) \uparrow$$

Důkaz. Použijeme Cantorovou diagonální metodu. Definujme pomocnou ČRF:

$$\beta(x) \simeq 1 \div \alpha(x, x)$$

Kde \div je dodefinovaná operace odečítání pro přirozená čísla. Např $1 \div 100 = 0$. Jelikož je ČRF \Rightarrow má index e_β , neboli

$$\beta(e_\beta) \simeq \Psi_1(e_\beta, e_\beta) \simeq 1 \div \alpha(e_\beta, e_\beta)$$

Nechť sporem $\alpha(e_\beta, e_\beta) \downarrow$, pak

$$\Psi_n(e_\beta, e_\beta) \downarrow$$

Protože α je rozšíření

$$\Psi_n(e_\beta, e_\beta) = \alpha(e_\beta, e_\beta)$$

což je spor protože

$$1 \div \alpha(e_\beta, e_\beta) = \Psi_n(e_\beta, e_\beta) = \alpha(e_\beta, e_\beta)$$

□

3 Rekurzivně spočetné množiny a predikáty

Poznámka 3.1. R.s. množiny a predikáty je jedno a totéž protože obor pravdivostí predikátu je množina a nalezení do množiny je predikát.

Lemma 3.2 ($\exists Q$ r.s). *Pokud Q je rekurzivní $\Rightarrow \exists y Q(\dots)$ je rekurzivně spočetný.*

Důkaz. Uvažme charakteristickou funkci C_Q predikátu Q . Je všude definovaná, čili je ORF.

Pak následující je ČRF:

$$\mu_y Q(\dots) \simeq \mu_y (C_Q(\dots) = 1)$$

□

Věta 3.3 (Univerzální Kleeneho r.s. predikát). *Každý rekurzivně spočetný predikát je tvaru:*

$$\exists y Q(\dots)$$

Pak r.s. množiny jsou definiční obory ČRF.

Dokonce máme univerzální rekurzivně spočetný predikát

$$\exists y T_n(e, x_1, \dots, x_n, y)$$

Důsledek 3.4 (Index r.s. predikátů). *Lze definovat index rekurzivně spočetných predikátů.*

Poznámka 3.5. s-m-n věta 2.21 platí i pro predikáty T_n .

Věta 3.6 (Uzavřenost RS). *Rekurzivní spočetnost je uzavřená na \cup, \cap . Dokonce efektivně z indexu.*

Máme ORF(dokonce PRF)

$$W_{\alpha(a,b)} = W_a \cap W_b$$

Důkaz. Formálně pro \cap :

$$\exists s_1 T_1(a, x, s_1) \wedge \exists s_2 T_1(b, x, s_2) \iff \exists w (T_1(a, x, (w)_{2,1}) \wedge T_1(b, x, (w)_{2,2}))$$

kde w koduje dvojici s_1, s_2 .

$$(w)_{2,1}$$

říká, že w je n -tice velikosti 2, vezmi 1. složku.

$$\exists s_1 T_1(a, z, s_1)$$

je reprezentace množiny W_a pomocí univerzálního predikátu.

Dohromady máme rekurzivně spočetný predikát. Takže

$$\exists z T_3(e, a, b, x, z)$$

Což je program, který pouští oba dva programy a čeká až se jeden z nich zastaví. Použijeme s-m-n 2.21 pro predikáty

$$\exists z T_3(e, a, b, x, z) \iff \exists z T_1(s_2(e, a, b), x, z)$$

Pak definujeme

$$\alpha(a, b) := s_2(e, a, b)$$

Analogicky pro \cup .

□

Otázka: pokud všude použijeme omezené kvantifikátory s $y =$ velikost částic ve vesmíru, dostaneme upravenou logiku pro počítače?

Definice 3.7 (Omezený existenční kvantifikátor).

$$\exists_{y < z} Q$$

Jmenuje se konečná disjunkce.

Definice 3.8 (Omezený všeobecný kvantifikátor).

$$\forall_{y < z} Q$$

Jmenuje se konečná konjunkce.

Věta 3.9 (Omezená kvantifikace (BD)). *Rekurzivní spočetnost je uzavřená na omezené kvantifikace. Dokonce efektivně na indexech.*

Důkaz. Pro existenční spustíme z programů paralelně a čekáme až jeden přijme.

Pro všeobecný spustíme paralelně a čekáme jestli všechny přijmou. \square

Věta 3.10 (Neomezená kvantifikace). *Rekurzivní spočetnost je uzavřená na existenční kvantifikace.*

Důkaz. Analogicky jako důkaz pro \cap , nahradíme dva existenční kvantifikátory jediným s dvojicí.

$$\exists y \exists s : T_n(\dots) \simeq \exists k = \langle y, s \rangle \dots$$

\square

Poznámka 3.11. Pro všeobecnou již neplatí (ani pro částečně rozhodnutelné). Protipříkladem je \overline{K} která není č.r. Již lze zapsat pomocí všeobecného kvantifikátory

$$x \in \overline{K} \iff \forall s \neg T_1(x, x, s)$$

Kde T_1 je částečně rozhodnutelný predikát, negace taky.

3.1 10 Hilbertův problém

V moderní terminologii 10. Hilbertův problém zní: "zda existuje algoritmus, který by pro libovolný celočíselný polynom rozhodnul jestli existuje řešení v celých číslech.

Libovolný celočíselný polynom je ekvivalentní 2 polynomům v \mathbb{N} , řešení pak taky hledáme v \mathbb{N} . Nejprve dáme záporné koeficienty na pravou stranu, pak aplikujeme Lagrangeovou větu o 4 \square .

Věta 3.12 (RDPM (BD)). *Predikát Q je rekurzivně spočetný \iff je tzv diofantický:*

$$\exists x_1, \dots, x_k \in \mathbb{N} : (p_1(x_1, \dots, x_k, y_1, \dots, y_n) = p_2(x_1, \dots, x_k, y_1, \dots, y_n))$$

Důsledek 3.13 (10. Hilbertův problém). *10. Hilbertův problém má negativní odpověď. Protože máme množiny které nejsou rozhodnutelné, třeba DIAG.*

Dodatek 3.14. *Jako byprodukt dostáváme ekvivalenci:*

$$\exists (PRP) \iff \exists (p_1(\dots) = p_2(\dots))$$

Bez existenčního kvantifikátoru vůbec není pravda. V aritmetice lze vytvořit i superexponenciálu e^{n^n} atd, polynomy jsou ale omezené. Taky polynomy lze elementárně vyjádřit pomocí Robinsonové aritmetiky.

3.2 Selektory

Obecné, Selektor je definovaný pro "hezke relace", např $Q(x, y)$. Pak selektor vybírá y pro $\forall x$.

Věta 3.15 (O selektoru). *Nechť $Q(x, y)$ je rekurzivně spočetný (resp $Q(x_1, \dots, x_n, y)$), pak $\exists \varphi \in \text{ČRF}$:*

$$\begin{aligned}\varphi(x) \downarrow &\iff \exists y : Q(x, y) \\ \varphi(x) \downarrow &\Rightarrow Q(x, \varphi(x))\end{aligned}$$

Jinými slovy vybere y pokud existuje.

Důkaz. Pozor, nemůžeme vzít nejmenší, musíme vzít první protože lepší už třeba nebude. Q je r.s. \Rightarrow má index e , napíšeme pomoci univerzálního predikátu

$$\exists s : T_2(e, x, y, s)$$

Predikát zapíšeme jako množinu

$$\text{dom}(\varphi_e(x, y))$$

Pak pro dané x probereme všechny $\langle y, s \rangle$ a hledáme nejmenší dvojici tž platí

$$T_2(e, x, y, s)$$

Neboli hledáme první $\langle y, s \rangle$ tž za s kroků $\varphi_e(x, y) \downarrow$.

Formálně:

$$\varphi(x) \simeq (\mu_{\langle y, s \rangle} T_2(e, x, y, s))_{2,1}$$

indexy vydělí y . □

Definice 3.16 (Graf ČRF).

$$\text{graph}(\varphi) = \{ \langle x, y \rangle \mid \varphi(x) = y \}$$

Důsledek 3.17 (Graf ČRF). φ je ČRF $\iff \text{graph}(\varphi)$ je r.s.

Důkaz. " \Rightarrow " $\langle x, y \rangle \in \text{graph}(\varphi) \Rightarrow \exists s$ (za s kroků $\varphi(x) = y$).

Což je r.s. predikát.

" \Leftarrow " Aplikuj selektor. Volba v totalitním režimu, buď jeden kandidát nebo nic. □

Poznámka 3.18. Při zobecněních vyčíslitelnost do vyšších hierarchii, definujme vyčíslitelnost tak, že graf je rozumný.

Věta 3.19 (Postova podruhe).

$$Q(x, y) = (x \in M \wedge y = 1) \vee (x \in \overline{M} \wedge y = 0)$$

Což je rekurzivně spočetný predikát. Selektor φ pro Q je ORF. φ je charakteristická funkce množiny M .

3.3 Imunní množiny

Definice 3.20 (Imunní množina). A je *imunní* pokud je nekonečná a neobsahuje nekonečnou rekurzivní spočetnou podmnožinu.

$$\forall W_x \subseteq A : |W_x| < \infty$$

Je nekonečná, ale nemůžeme to efektivně zkontrolovat. Protože veškeré algoritmický zkontrolovatelné podmnožiny jsou konečné.

Definice 3.21 (Simple). A je *simple* pokud je rekurzivní spočetná a \bar{A} je imunní.

Poznámka 3.22. Postův problém: co je mezi Rekurzivní množiny a nerekurzivní množinou K ?

Definoval Simple, hypersimple, hyper-hyper ... atd až do Maximalní. Ale tato klasifikace neuspěla.

Věta 3.23 (Existence Simple). *Existuje Simple množina.*

Důkaz. Uděláme predikát

$$Q(x, y) \iff y \in W_x \wedge y > 2x$$

je rekurzivně spočetný protože nalezení je r.s. a druhá podmínka taky. Necht $\varphi \in \text{ČRF}$ je selektor pro Q . Pak

$$A = \text{range}(\varphi)$$

Podrobněji:

$$W_x \subseteq \bar{A} \Rightarrow W_x \subseteq \{0, \dots, 2x\}$$

Neboli \bar{A} neobsahuje nekonečnou r.s. množinu.

\bar{A} nekonečná?

Do $\{0, \dots, 2x\}$ mohou přispět nejvýše W_0, \dots, W_{x-1} množin. Neboli nejvýše x čísel. Pak ale v \bar{A} zůstane nejméně $(x+1)$ čísel, neboli \bar{A} je nekonečná.

Dohromady \bar{A} je imunní.

□

4 Věty o rekurzi

Taky se jmenují věty o pevném bodě. Používá se self-refrenční trik

Věta 4.1 (O rekurzi 1). *Pokud f je ČRF (pro jednoduchost 1 proměnné) \Rightarrow (efektivně z indexů)*

$$\exists a \forall x : \varphi_a(x) \simeq \varphi_{f(a)}(x)$$

Jinými slovy: pokud $f(a) \Downarrow \Rightarrow \varphi_a$ a $\varphi_{f(a)}$ jsou stejné funkce. Programy nejsou stejné, ale vyčíslují stejnou funkci.

Pokud ale $f(a) \Uparrow \Rightarrow \forall x : \varphi_a(x) \Uparrow$.

Důkaz.

$$\varphi_{f(s_1(z,z))}(x) \simeq \Psi_2(e, z, x)$$

Protože levá strana je efektivně vyčíslitelná. e je program který počítá levou stranu.

Pak dle s-m-n věty 2.21

$$\varphi_{f(s_1(z,z))}(x) \simeq \Psi_2(e, z, x) \simeq \varphi_{s_1(e,z)}(x)$$

polož $z = e$, dostaneme

$$\varphi_{f(s_1(e,e))}(x) \simeq \varphi_{s_1(e,e)}(x)$$

Neboli

$$a = s_1(e, e)$$

Který program počítá déle?

Program $e = f(s_1(z, z))$:

1. spočítej $s_1(z, z)$.
2. spočítej $f(s_1(z, z))$.
který ale nemusí konvergovat
3. $if(f(s_1(z, z)) \downarrow)$
spust e na vstup x .

Program $a = s_1(z, z)$:

1. dostane x na vstupu, kvůli s-m-n přidá e ke vstupu
2. spustí program e na vstup $\langle e, x \rangle$.
3. spočítej $a = s_1(z, z)$.
Tady spočítal svůj vlastní index.
4. spočítej $f(s_1(z, z))$ tzn $f(a)$.
který ale nemusí konvergovat
5. $if(f(s_1(z, z)) \downarrow)$ then
spust $f(a)$ na vstup x .

Takže a počítá déle. □

Věta 4.2 (O rekurzi 2). *Pokud f je ČRF $(n+1)$ proměnných \Rightarrow ORF (dokonce PRF)*

$$\varphi_{h(y_1, \dots, y_n)}(x) \simeq \varphi_{f(h(y_1, \dots, y_n, y_1, \dots, y_n))}(x)$$

Pokud smažeme y -ny, tak dostaneme právě Větu o rekurzi 1 4.1.

Pevné body efektivně na parametrech.

Důkaz. Analogicky jako důkaz Věty o rekurzi 1 4.1. Jenom aplikujeme s-m-n 2.21 na větší počet parametrů.

$$\varphi_{f(s_{n+1}(z, z, y_1, \dots, y_n), y_1, \dots, y_n)}(x) \simeq \Psi_{n+2}(e, z, y_1, \dots, y_n, x) \simeq \varphi_{s_{n+1}(e, z, y_1, \dots, y_n)}(x)$$

Pak

$$h(y_1, \dots, y_n) = s_{n+1}(e, e, y_1, \dots, y_n)$$

□

Věta 4.3 (O rekurzi ∞). Pokud f je ČRF $\Rightarrow \exists$ prostá ORF g (dokonce PRF)

$$\varphi_{g(j)}(x) \simeq \varphi_{f(g(j))}(x)$$

Pak pevných bodů je nekonečno

$$g(0), g(1), \dots$$

Důkaz.

$$\varphi_{f(s_2(z,z,j))}(x) \simeq \Psi_2(e, z, j, x) \simeq \varphi_{s_2(e,z,j)}(x)$$

Zvolme

$$g(j) = s_2(e, e, j)$$

□

Věta 4.4 (O rekurzi 3). Pokud $h(x, z_1, \dots, z_n)$ je ČRF, pak existuje $a \in \mathbb{N}$ t.ž. a je indexem funkce

$$h(a, z_1, \dots, z_n)$$

Důkaz.

$$h(x, z_1, \dots, z_n) \simeq \Psi_{n+1}(e, x, z_1, \dots, z_n) \simeq \varphi_{s_1(e,x)}(z_1, \dots, z_n)$$

Pak aplikujeme Větu o rekurzi 4.1 na funkci $s_1(e, x)$.

□

Poznámka 4.5. Věty o rekurzi platí nejen pro ČRF ale taky pro jejich definiční obory. Takže

$$f \in \text{ČRF} \Rightarrow \exists a : W_a = W_{f(a)}$$

Věta 4.6 (Program vlastní kod). Existuje n_0 :

$$\varphi_{n_0}(n_0) = n_0$$

Program který vypíše svůj vlastní kód.

Důkaz. Pořídíme si pomocnou ORF

$$\alpha(n, w) = n$$

použijeme s-m-n 2.21

$$\alpha(n, w) \simeq \varphi_{f(n)}(w)$$

Kde

$$\alpha(n, w) \simeq \Psi(e, n, w) \simeq \varphi_{s_1(e,n)}(w)$$

TODO Pak $f(x) = s_1(x, x)$??? Stačí použít Větu o rekurzi 4.1 k f

$$\varphi_{f(n_0)}(w) \simeq \varphi_{n_0}(w) = n_0$$

□

Věta 4.7 (Rekurze v indexech (BD)). Existuje ORF f :

$$W_{f(y)} = \{f(0), \dots, f(y-1)\}$$

Důkaz. Hint: hledáme index f kterých je spočetně mnoho. Musíme použít Větu o rekurzi na úrovně indexů.

□

Věta 4.8 (Rice podruhe). *Pokud F je netriviální třída ČRF (nebo r.s množin). Není prázdná, nebo má všechny. Pak indexová množina*

$$A_F = \{x \mid \varphi_x \in F\}$$

Je nerekurzivní.

Důkaz. Z netriviality F

$$\exists a \in A \wedge b \in \bar{A}$$

Nechť sporem A je rekurzivní.

Uděláme funkci h t.ž

$$\forall x \in A : h(x) = b$$

$$\forall y \in \bar{A} : h(y) = a$$

Protože A je rekurzivní, tak h je ORF \Rightarrow existuje pevný bod třeba n_0

$$n_0 \in A \Rightarrow h(n_0) \in \bar{A}$$

Z Věty o rekurzi ale víme

$$\varphi_{n_0} = \varphi_{h(n_0)}$$

takže $n_0, h(n_0)$ musí být ve stejné množině.

Z toho A není rekurzivní. □

Věta 4.9 (O rekurzi (BD)). *Nechť $f \in \text{ČRF}$ pak //TODO*

Důkaz.

$$\varphi_{\varphi_u(u)}(z) \simeq \Psi_2(a, u, z) \stackrel{\text{s-m-n}}{\simeq} \varphi_{s_1(a, u)}(z) \simeq \varphi_{d(u)}(z) \simeq \varphi_{\varphi_e(u)}(z)$$

Všimneme si, že

$$\varphi_u(x)$$

je *matice* funkci. Implikace nahoře ukazuje, že její diagonála $\varphi_u(u)$ se rovná řádku φ_e . Ukážeme, že f permutuje řádky.

$$\varphi_{f \circ \varphi_u(x)}(z) \simeq \varphi_{\beta(u, x)} \stackrel{\text{s-m-n}}{\simeq} \varphi_{\varphi_{H(u)}(x)}(z)$$

Kde

$$H(u) = s_1(b, u)$$

Z toho u -tý řádek se zobrazí na $H(u)$ -tý. Speciálně

$$e \rightarrow H(e)$$

Z toho $\varphi_e(H(e))$ je pevný bod. Protože:

$$\varphi_{f \circ \varphi_e(H(e))}(z) \simeq \varphi_{\varphi_{H(e)}(H(e))}(z) \stackrel{\text{diagonála}}{\simeq} \varphi_{\varphi_e(H(e))}(z)$$

□

5 Produktivní množiny

Definice 5.1 (Produktivní množina). B je *produktivní* pokud

$$\exists \varphi \in \check{C}RF : W_x \subseteq B \Rightarrow (\varphi(x) \downarrow) \wedge \varphi(x) \in B \setminus W_x$$

Jinými slovy: non-rekurzivní spočetnost. Pokud máme uvnitř množinu W_x tak se nemůže rovnat B . Taky máme stroječek který najde $\varphi(x)$ který leží mimo danou W_x .

Definice 5.2 (Kreativní množina). Množina A je *kreativní* pokud A je rekurzivně spočetná a \overline{A} je produktivní.

Příklad 5.3 (K). \overline{K} je produktivní funkce je *id*, K je kreativní.
Protože

$$W_x \subseteq \overline{K} \Rightarrow x \in (\overline{K} - W_x)$$

Věta 5.4 (Modifikace K). *Modifikace předchozího příkladu:*
Nechť máme $f \in ORF$ prostá, pak uvažme množinu:

$$A = \{f(x) \mid f(x) \in W_x\}$$

A je kreativní, \overline{A} produktivní s f .

Důkaz. Nechť $W_x \subseteq \overline{A}$, kdyby $f(x) \in W_x$ tak

$$f(x) \in \overline{A}$$

ale dle definice A

$$f(x) \in A$$

Tedy

$$f(x) \notin W_x$$

a jelikož je **prostá** tak

$$f(x) \in \overline{A} - W_x$$

□

Věta 5.5 (Produktivní funkce ORF). *Každá produktivní množina má ORF produktivní funkce.*

Důkaz. Jednoduše dodefinovat ČRF na ORF nejde.
Chceme najít ORF h :

$$W_{h(y)} = \begin{cases} W_y & \text{pro } \varphi(h(y)) \downarrow \\ \emptyset & \text{pro } \varphi(h(y)) \uparrow \end{cases}, \text{ kde } \varphi \in \check{C}RF \text{ prod.}$$

Formálně:

$$\varphi \circ h \in ORF$$

Kdyby $\varphi(h(y)) \uparrow$ tak

$$\Rightarrow W_{h(y)} = \emptyset \subseteq B \Rightarrow \varphi(h(y)) \downarrow \text{ spor}$$

Dal

$$\forall y : W_{h(y)} = W_y$$

Taky

$$W_y \subseteq B \Rightarrow W_{h(y)} \subseteq B$$

Z toho

$$\varphi(h(y)) \in B - W_{h(y)} = B - W_y$$

Hledaná funkce je $\varphi \circ h$.

Získáme funkci h pomocí věty o rekurzi 4.1. Vezmeme pomocnou $f \in ORF$:

$$W_{f(x,y)} = \begin{cases} W_y & \text{pro } \varphi(x) \downarrow \\ \emptyset & \text{pro } \varphi(x) \uparrow \end{cases}$$

f pomocí s-m-n 2.21

$$f \simeq \alpha(x, y, w) \downarrow \iff w \in W_y \wedge \varphi(x) \downarrow$$

Taky

$$\alpha(x, y, w) \simeq \varphi_{f(x,y)}(w)$$

kde

$$f(x, y) = s_2(a, x, y)$$

Pak dle věty of rekurze:

$$W_{h(y)} = W_{f(h(y),y)} = \begin{cases} W_y & \text{pro } \varphi_{h(y)} \downarrow \\ \emptyset & \text{pro } \varphi_{h(y)} \uparrow \end{cases}$$

□

Věta 5.6 (Produktivní funkce ORF prostá(BD)). *Každá produktivní množina má dokonce prostou ORF produktivní funkce. Dokonce rekurzivní permutace.*

Věta 5.7 (Nekonečná množina). *Každá produktivní množina obsahuje nekonečnou r.s. podmnožinu.*

Důkaz. Máme B a $f \in ORF$ produktivní.

Vezmeme takové z_0 :

$$W_{z_0} = \emptyset$$

Množinu vytváříme iterativně, vždy na jeden z bodů co máme aplikujeme f a vezmeme sjednocení.

Formálně:

$$W_{g(x)} = W_x \cup \{f(x)\}$$

rekurze

$$h(0) = z_0 \tag{1}$$

$$h(y+1) = f(h(y)) \tag{2}$$

Pak

$$W_{h(y)} = \{f(z_0), \dots, f(h(y)-1)\}$$

což je y bodů z B .

□

Poznámka 5.8. Imunní a produktivní množiny jsou disjunktní pojmy.

Dodatek 5.9. Jak dlouho lze pokračovat v konstrukci množiny popsané ve Větě o nekonečné množině 5.7?

Odpověď: pokud to bude efektivní proces neboli aby množiny byly r.s.

Můžeme iterovat $\omega, 2\omega \dots$ podél tzv rekursivních ordinálů (viz ordinální číslo v teorii množin).

Lemma 5.10 (Produktivita a \leq_m). *A produktivní a $A \leq_m B \Rightarrow B$ je produktivní. Neboli produktivita se zachovává směrem vzhůru při \leq_m .*

Důkaz. Máme ORF funkce g z převoditelnosti. Pak nechť $W_x \subseteq B$, najdeme její preimage v A

$$P = g^{-1}(W_x) \subseteq A$$

Z toho že A je kreativní, pomocí kreativní funkce f najdeme bod $f(y) \notin P$. Zobražíme pomocí $g(f(y))$, tím dostaneme bod $\in B - W_x$.

Formálně:

$$W_{h(x)} = g^{-1}(W_x) = \{y \mid g(y) \in W_x\}$$

Pak

$$W_x \subseteq B \Rightarrow W_{h(x)} \subseteq A$$

Poslední krok

$$g \circ f \circ g^{-1}(x) \in B - W_x$$

□

Věta 5.11 (Ekvivalence Kreativní). *Nechť M množina. Následující tvrzení jsou ekvivalentní:*

(a) M je kreativní $\iff \overline{M}$ produktivní.

(b) M je 1-úplná $\iff \overline{K} \leq_1 \overline{M}$

(c) M je m -úplná $\iff \overline{K} \leq_m \overline{M}$

Každý z pojmu zahrnuje rekursivní spočetnost.

Ekvivalence mezi totálně různými pojmy. 1-úplnost jako u NP znamená, že je to nejtěžší ze všech takových množin.

Důkaz. (b) \Rightarrow (c) z vlastnosti 1 a m převoditelnosti.

(c) \Rightarrow (a)

Z vlastnosti převoditelnosti

$$K \leq_m M \iff \overline{K} \leq_m \overline{M}$$

pak použijeme lemma lemma 5.10. Víme že \overline{K} je produktivní, takže i \overline{M} . Pak dle definice, M je kreativní.

(a) \Rightarrow (b) (\overline{M} produktivní $\Rightarrow \overline{K} \leq_1 \overline{M}$)

Cíl

$$W_{h(y)} = \begin{cases} \{f \circ h(y)\} & \text{pro } y \in K \\ \emptyset & \text{pro } y \notin K \end{cases}$$

kde f je ORF prostá, produktivní pro \overline{M} .

Konstrukce funkce h

$$W_{g(x,y)} = \begin{cases} \{f(x)\} & \text{pro } y \in K \\ \emptyset & \text{pro } y \notin K \end{cases}$$

g dostaneme pomoci s-m-n věty 2.21:

$$\alpha(x, y, w) \simeq \varphi_{g(x, y)}(w) \downarrow \iff y \in K \wedge w = f(x)$$

Strojil skripta chyba, rovnice č. 57.

Pak použijeme větu o rekurzi

$$W_{h(y)} = W_{g(h(y), y)}$$

Z toho platí

$$\begin{aligned} y \notin K &\Rightarrow W_{h(y)} = \emptyset \subseteq \overline{M} \Rightarrow f \circ h(y) \in \overline{M} \\ y \in K &\Rightarrow W_{h(y)} = \{f \circ h(y)\} \end{aligned}$$

kdyby $f \circ h(y) \in \overline{M}$ tak

$$W_{h(y)} \subseteq \overline{M} \Rightarrow f \circ h(y) \in \overline{M} - W_{h(y)}$$

což je spor.

Neboli

$$f \circ h(y) \in M \Rightarrow \overline{K} \leq_1 \overline{M}$$

□

Důsledek 5.12 (\overline{K} produktivní). \overline{K} je nejjednodušší produktivní množinou při \leq_1 nebo \leq_m . Protože všechny produktivní množiny jsou

$$\{B \mid \overline{K} \leq_m B\}$$

Definice 5.13 (Úplně produktivní). B je úplně produktivní když existuje ORF f tž:

$$f(x) \in B - W_x \vee f(x) \in W_x - B$$

Příklad 5.14. \overline{K} je úplně produktivní dle definice K.

$$x \in \overline{K} - W_x \vee x \in W_x - \overline{K}$$

neboli funkce je id .

Věta 5.15 (Úplna produktivita ekvivalence). B je úplně produktivní $\iff B$ je produktivní.

Důkaz. \Rightarrow triviálně z definice.

\Leftarrow lze dokázat 2ma způsoby. První je inspekci minulého důkazu. Uděláme

1. $g^{-1}(W_x)$
2. $f(\dots)$
3. $g \circ f \circ g^{-1}$.

Jen se musí ověřit o 1 disjunkci víc.

Druhý pomoci věty o rekurzi:

$$W_{h(y)} = \begin{cases} \{f \circ h(x)\} & \text{pro } f \circ h(x) \in W_y \\ \emptyset & \text{pro } f \circ h(x) \notin W_y \end{cases}$$

f je ORF produktivní funkce, h dostaneme pomocí věty o rekurzi a s-m-n věty. Pak

$$f \circ h(x) \notin W_y \Rightarrow W_{h(y)} = \emptyset \Rightarrow f \circ h(x) \in B \Rightarrow f \circ h(x) \in B - W_y$$

$$f \circ h(x) \in W_y \Rightarrow W_{h(y)} = \{f \circ h(x)\}$$

Kdyby $f \circ h(x) \in B$ tak

$$\Rightarrow W_{h(y)} \in B \Rightarrow f \circ h(x) \in B - W_{h(y)}$$

Z toho

$$f \circ h(x) \in W_y - B$$

□

Definice 5.16 (Totální množina).

$$Tot = \{x \mid \varphi_x \text{ totální}\} = \{x \mid \exists y \varphi_x(y) \downarrow\}$$

Lemma 5.17 (Totalni je produktivni). *Totální množina je produktivní.*

Důkaz. Pomoci m -převodu na \bar{K} .

$$\varphi_{h(x)}(y) \downarrow \iff x \notin K_j$$

Kde $x \notin K_j$ znamená, že $x \notin K$ za j kroků.

$$x \notin K \iff h(x) \in Tot$$

Pokud x není v K , tak tam nebude za žádný počet kroků. Pak i $h(x)$ je všude definovaná. Jinak

$$x \in K \Rightarrow |dom(\varphi_{h(x)})| < \infty$$

Definiční obor je konečný a rovna se nějakému $\{0, \dots, j_0\}$, což je počet kroků za který x vstoupí do K .

Problém ale je, že dostáváme nový program, ale ne zaručeně novou funkci.

Dokážeme silnější tvrzení a konkrétně vytvoříme novou funkci. Máme

$$W_y \in Tot$$

uděláme novou $F \in ORF$ která roste rychleji než $\varphi_a : \forall a \in W_y$. Jinými slovy

$$\forall a \in W_y \exists z_0 \forall z \geq z_0 : F(z) \geq \varphi_a(z)$$

F majorizuje $\varphi_a : \forall a \in W_y$.

BUNO: W_y je nekonečná, jinak přidáme nekonečně indexů prázdného programu. Kvůli enumeratoru, můžeme W_y efektivně generovat, neboli vypisovat

$$a_0, a_1, \dots$$

Pak

$$F(x) = \max_{j \in \{1, \dots, x\}} (\varphi_{a_j}(x)) + 1$$

□

Důsledek 5.18 (Omezení logiky). *Z věty plyne omezení logiky.*

Vezmeme třeba Peano aritmetiku (PA). Můžeme efektivně generovat sentence které PA dokazuje, tudíž lze efektivně generovat ty $a : \varphi_a$ totální.

Pak můžeme dle předchozí věty můžeme najít F která roste rychleji, než cokoliv co PA dokazuje.

Libovolná efektivně zadaná teorie má jen r.s. množinu dokazatelných sentencí. Pokud z nich vybereme ty, co dokazují o nějakém programu že je všude definovaný, tak vyrobíme sentenci na kterou daná teorie nestačí.

6 Dvojice množin

Poznámka 6.1. Motivace z logiky: pokud máme rozumnou teorii, tak určuje sentence které dokazuje a sentence které vyvrací. Když teorie je bezesporná, tak množiny jsou disjunktí.

Definice 6.2 (Rekurzivní neoddělitelnost). Disjunktí dvojice množin A, B jsou rekurzivně neoddělitelné když neexistuje M t.ž.

$$A \subseteq M \wedge M \cap B = \emptyset (B \subseteq \overline{M})$$

Definice 6.3 (Efektivní neoddělitelnost). Disjunktí dvojice množin A, B jsou efektivně neoddělitelné když existuje $f \in \text{CRF}$ t.ž.

$$\forall x, y : A \subseteq W_x \wedge B \subseteq W_y \wedge W_x \cap W_y = \emptyset \Rightarrow f(x, y) \downarrow \notin W_x \cup W_y$$

Efektivně můžeme najít bod který leží mimo obaly A, B .

Poznámka 6.4. Efektivní neoddělitelnost \Rightarrow rekurzivní neoddělitelnost.

Věta 6.5 (Efektivní neoddělitelné (BD)). *Existují rekurzivně neoddělitelné které nejsou efektivně neoddělitelné.*

Důkaz. Podobná konstrukce jako u Simple. □

Poznámka 6.6. Efektivní neoddělitelnost vždy lze definovat tak, aby $f \in \text{ORF}$ (neboli všude definovaná).

Věta 6.7 (Existence efektivní neoddělitelné). *Existují disjunktí r.s E, F které jsou efektivně neoddělitelné.*

Důkaz. Znovu diagonální metoda.

Vezmeme

$$E = \{x \mid \varphi_x(x) \simeq 0\}$$

$$F = \{x \mid \varphi_x(x) \simeq 1\}$$

Na konkrétních hodnotách nezáleží, šlo by vzít $i, j \in \mathbb{N} : i \neq j$.

Hned je vidět, že E, F jsou disjunktí a r.s.

Podle s-m-n 2.21 věty existuje PRF taková, že:

$$\varphi_{\alpha(x,y)}(w) = \begin{cases} 1 & w \text{ padne dříve do } W_x \text{ než do } W_y \\ 0 & w \text{ padne dříve do } W_y \text{ než do } W_x \\ \uparrow & w \notin W_x \cup W_x \end{cases}$$

Formálně (dříve než ...)

$$\exists j(T_1(x, w, j) \wedge \forall i \leq j : \neg T_1(y, w, i))$$

Pokud oba dva programy skončí za stejný počet kroků, tak vezmeme libovolný. Necht' $W_x \supseteq E$ je rekurzivní obal E , nápodobně $W_y \supseteq F$.

Uvažme

$$\varphi_{\alpha(x,y)}(\alpha(x,y))$$

Kdyby $\alpha(x,y)$ padlo do W_x , potom zřejmě padne dříve do W_x než do W_y . Pak

$$\varphi_{\alpha(x,y)}(\alpha(x,y)) = 1$$

a tedy by muselo $\alpha(x,y)$ padnout do F . To nelze.

Symetricky $\alpha(x,y)$ nemůže padnout do W_y neboli

$$\alpha(x,y) \notin W_x \cup W_y$$

□

Poznámka 6.8. Paradox lháře: tento výrok je lživý. Používá self-referenci.

Ten ale operuje s pojmem pravdy, který není matematický. Můžeme ale podobný trik provést s např. pojmem dokazatelnosti (viz Godelová 1. věta o neúplnosti 7.9).

Definice 6.9 (1-převoditelnost dvojic). Disjunktní dvojice

$$(C, D) \leq_1 (A, B)$$

právě když existuje prostá $f \in \text{ČRF}$:

$$\begin{aligned} x \in C &\iff f(x) \in A \\ x \in D &\iff f(x) \in B \\ x \notin C \cup D &\iff f(x) \notin A \cup B \end{aligned}$$

Definice 6.10 (1-úplnost dvojic). Disjunktní dvojice r.s. množin (A, B) je 1-úplná právě když libovolná disjunktní dvojice r.s. množin (C, D) platí:

$$(C, D) \leq_1 (A, B)$$

Věta 6.11 (Dvojná věta o rekurzi). Pro libovolné $f, g \in \text{ORF}$:

$$\exists m, n : \varphi_m = \varphi_{f(m,n)}, \varphi_n = \varphi_{g(m,n)}$$

Obecněji: pro libovolné $f, g \in \text{ORF}$, obě $(k+2)$ proměnných, existují $w_1, w_2 \in \text{PRF}$:

$$\varphi_{w_1(y_1, \dots, y_k)} = \varphi_{f(w_1(y_1, \dots, y_k), w_2(y_1, \dots, y_k), y_1, \dots, y_k)}$$

$$\varphi_{w_2(y_1, \dots, y_k)} = \varphi_{g(w_1(y_1, \dots, y_k), w_2(y_1, \dots, y_k), y_1, \dots, y_k)}$$

Důkaz. Z Věty o rekurzi 4.1 existuje $h \in \text{ORF}$:

$$\varphi_{h(y)} = \varphi_{f(h(y), y)}$$

Vezmeme

$$\varphi_{g(h(y), y)}, \exists n : \varphi_n = \varphi_{g(h(n), n)}$$

Položme $m = h(n)$.

□

Věta 6.12 (Efektivní neoddělitelné, 1-úplnost). *Disjunktní r.s dvojice množin jsou efektivně neoddělitelné \iff jsou 1-úplné.*

Důkaz. "1-úplnost \Rightarrow efektivní neoddělitelnost".

Nechť (C, D) efektivně neoddělitelné s funkcí f a $(C, D) \leq_1 (A, B)$ s funkcí g , neboli (C, D) je 1-úplná.

Vezmeme vzory r.s. obalů $A \subseteq W_x, B \subseteq W_y$:

$$W_{\alpha(x)} = g^{-1}(W_x), W_{\alpha(y)} = g^{-1}(W_y)$$

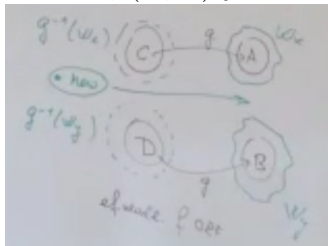
Jelikož, (C, D) efektivně neoddělitelné $\Rightarrow W_{\alpha(x)} \cap W_{\alpha(y)} = \emptyset$. Taky funkce f dá bod

$$new = f(\alpha(x), \alpha(y)) : new \notin W_{\alpha(x)} \cup W_{\alpha(y)}$$

Zobrazíme new pomocí g zpátky. Pak

$$g(new) = f \circ g(\alpha(x), \alpha(y)) \notin W_x \cup W_y$$

Z čehož, (A, B) je efektivně neoddělitelná.



"1-úplnost \Leftarrow efektivní neoddělitelnost".

Nechť (A, B) efektivně neoddělitelné s funkcí $f \in \check{R}F$. Nechť (C, D) libovolné disjunktní množiny.

Sestavíme množiny

$$W_{w_1(x)} = \begin{cases} A \cup \{f(w_1(x), w_2(x))\} & \text{pro } x \in D \\ A & \text{pro } x \notin D \end{cases}$$

$$W_{w_2(x)} = \begin{cases} B \cup \{f(w_1(x), w_2(x))\} & \text{pro } x \in C \\ B & \text{pro } x \notin C \end{cases}$$

Které dostaneme pomocí dvojné věty o rekurzi 6.11

$$W_{\alpha(y_1, y_2, x)} = \begin{cases} A \cup \{f(y_1(x), y_2(x))\} & \text{pro } x \in D \\ A & \text{pro } x \notin D \end{cases}$$

$$W_{\beta(y_1, y_2, x)} = \begin{cases} B \cup \{f(y_1(x), y_2(x))\} & \text{pro } x \in C \\ B & \text{pro } x \notin C \end{cases}$$

Zkontrolujeme 2 případy:

$$x \notin C \cup D \Rightarrow W_{w_1(x)} = A, W_{w_2(x)} = B \Rightarrow f(w_1(x), w_2(x)) \notin A \cup B$$

jinak např $x \in C \Rightarrow x \notin D$ protože jsou disjunktní dle předpokladu. Pak

$$W_{w_1(x)} = A, W_{w_2(x)} = B \cup \{f(w_1(x), w_2(x))\}$$

Pokud $f(w_1(x), w_2(x)) \notin A$ tak množiny $W_{w_1(x)}, W_{w_2(x)}$ jsou obaly A, B . Pak z efektivní neoddělitelnosti f by měla vracet nový bod, ležící mimo:

$$f(w_1(x), w_2(x)) \notin W_{w_1(x)} \cup W_{w_2(x)}$$

což je spor s konstrukcí $W_{w_2(x)}$, protože $f(w_1(x), w_2(x)) \in W_{w_2(x)}$.

Symetricky pro D .

Dohromady $f(w_1(x), w_2(x))$ 1-převádí (C, D) k (A, B) . □

7 Gödelovy věty

Definice 7.1 (Rozumná teorie). Rozumná teorie musí být:

- bezesporná
- axiomatizovatelná
- Základní aritmetické sily (adekvátní)

Definice 7.2 (Axiomatizovatelná teorie). Axiomatizovatelná teorie je právě když množina dokazatelných formulí je r.s.

Způsoby řešení paradoxu v teorii množin:

- intuicionisty/konstruktivisty finitisty - odmítají nekonečno. Jde vybudovat spoustu věcí, ale vznikající teorie je kostrbatá a nepříjemná.
Již Bolzano upozorňoval, že nekonečno je nevlastní pojem, překračující lidskou existenci. Viz [1]
- Hilbertův formalismus - vybudovat teorie z logiky 1. řadu a aby každé tvrzení šlo jednoznačně dokázat nebo vyvrátit + aby teorie byla bezesporná.

Poznámka 7.3. Tzv. Presburgerová Aritmetika bez násobení je rekurzivní a konzistentní.

Poznámka 7.4. Teorie vyčíslitelnosti dává ekvivalentní pohled na Gödelovy věty. Které původně byly vyjádřené přes jazyk aritmetiky.

Definice 7.5 (Reprezentovatelnost). $f \in \check{R}F$ je reprezentovatelná v teorii T pokud existuje formule F :

$$f(x) = y \Rightarrow \vdash_T F(\bar{x}, \bar{y})$$

$$\vdash_T F(x, y) \wedge F(x, z) \Rightarrow y = x \text{ (funkční vlastnost, aby nebyla jen relace)}$$

Pokud platí obě podmínky a T je bezesporná, tak

$$\{(x, y), \vdash_T F(\bar{x}, \bar{y})\} \text{ graf nějaké funkce } \supseteq f$$

Pozorování 7.6. $\check{R}F$ jsou reprezentovatelné v libovolné teorii ZAS.

Víme, že pro $\varphi \in \check{R}F$ graf

$$\{(x, y) : \varphi(x) \simeq y\}$$

je r.s.

Z důsledku RDPM věty ?? 3.14 máme

$$\text{r.s.} \approx \exists (p_1(\dots) = p_2(\dots))$$

Rovnost dvou polynomu je jednoduše formalizovatelné v teorii ZAS.

Věta 7.7 (Vztah \mathbb{N} a T). Pak pokud máme Σ_1 formule v jazyce ZAS, které jsou pravdivé v \mathbb{N} jsou v T (ZAS) dokazatelné.

$$\mathbb{N} \models \exists(\dots) \Rightarrow \vdash_T \exists(\dots)$$

Důkaz. Vezmeme např následující formule:

$$\exists x(x + \bar{7} = \bar{17})$$

Od teorie T chceme, aby formuli ověřila.

Pokud v \mathbb{N} je pravdivá nějaká Σ_1 formule, tak existuje tzv. Σ_1 svědek. Což je jedno nebo několik přirozených čísel které splňují formuli po dosazení. Teorie zkontroluje rovnost termu. \square

Lemma 7.8 (Disjunktní množiny formule(BD)). *Nechť T je bezesporná, ZAS. Pak A, B jsou disjunktní r.s. tak existuje Σ_1 -formule G :*

$$\begin{aligned} x \in A &\Rightarrow \vdash_T G(\bar{x}) \\ x \in B &\Rightarrow \vdash_T \neg G(\bar{x}) \end{aligned}$$

Důkaz. Sestavíme formuli:

$$\begin{cases} \varphi(x) = 0 & \text{pro } x \in A \\ \varphi(x) = 1 & \text{pro } x \in B \end{cases}$$

Pak

$$\begin{aligned} x \in A &\Rightarrow \vdash_T F(\bar{x}, \bar{0}) \\ x \in B &\Rightarrow \vdash_T F(\bar{x}, \bar{1}) \end{aligned}$$

Z bezespornosti

$$x \in B \Rightarrow \vdash_T F(\bar{x}, \bar{1}) \Rightarrow \vdash_T \neg F(\bar{x}, \bar{0})$$

Pak hledaná formule je

$$G(x) = F(x, \bar{0})$$

\square

Věta 7.9 (Gödelové věty). *Jestliže teorie T 1. řádu má základní aritmetickou sílu a je bezesporná, pak:*

1. množina dokazatelných v T formulí není rekurzivní
2. pokud je T navíc axiomatizovatelná, tak existuje uzavřená formule (sentence) F taková, že:

$$T \not\vdash F \wedge T \not\vdash \neg F$$

3. (2. Věta) axiomatizovatelnost + Indukce Σ_1 (stačí i trochu méně) tak v T nelze dokázat její bezespornost (consistency). Formálně:

$$T \not\vdash \text{Con}_T$$

kde Con_T je formule vyjadřující konsistence, např

$$\neg \exists \text{proof } (\bar{0} = \bar{1})$$

Důkaz. 1. Nechť T ZAS, bezesporná. Nechť (A, B) -neoddělitelné.

$$A_1 = \{x : \vdash_T G(\bar{x})\}$$

$$B_1 = \{x : \vdash_T \neg G(\bar{x})\}$$

$A \subseteq A_1$ je obal, podobně $B \subseteq B_1$.

Jelikož A, B jsou efektivně neoddělitelné \Rightarrow jsou rekurzivně neoddělitelné. Z toho A_1, B_1 nejsou rekurzivní. // TODO proč?

Q: protože jinak kdyby byly rekurzivní, tak by nevyplnili celý prostor?

2. Když přidáme axiomatizovatelnost, tak A_1, B_1 jsou r.s.

Pak z efektivní neoddělitelnosti efektivně najdeme takové $k \notin A_1 \cup B_1$ že:

$$\not\vdash_T G(\bar{k}) \wedge \vdash_T \neg G(\bar{k})$$

3. BD, formalizace, hodně logiky.

Jinými slovy, máme následující:

\exists můj důkaz IF existuje kratší důkaz mé negace

A symetricky pro gace. Nedochází k žádnému paradoxu, oproti paradoxu lháře. \square

7.1 Kalibrace síly teorie

Poznámka 7.10. Konečná verze Ramsey věty je v PA nedokazatelná.

Poznámka 7.11. PA má sílu

$$\varepsilon_0 = w^{w^{\dots}}$$

kde exponent je dlouhý w .

Což je největší ordinál, který ještě dává dobré uspořádání.

Zkoumá tuto oblast *proof theory*.

8 Relativní vyčíslitelnost

Zobecnění 1-převoditelnosti na relativný výpočet (s Orákulem).

Definice 8.1 (tt-Převoditelnost). Tzv tt (truth table) převoditelnost znamená, že existuje $f \in ORF$ která vrátí:

$$x \rightarrow \begin{cases} n_x & \\ \alpha_x & \text{n-ární booleovskou funkci} \\ y_1, \dots, y_n & \text{body} \end{cases}$$

Pro niž platí:

$$C_A(x) = \alpha_x(C_B(y_1), \dots, C_B(y_n))$$

Kde C_i je charakteristická funkce.

Neboli

$$x \in A \iff \alpha_x(\dots) = 1$$

Značení:

$$A \leq_{tt} B$$

Poznámka 8.2. TT převoditelnost musí napřed říct, na které body se bude ptát. Což je omezení.

8.1 Formalizace relativního výpočtu

Existuje několik možností formalizace:

Definice 8.3 (Formalizace relativního výpočtu). 1. TS s orákulem. Přidáme další pasku, kde TS bude umisťovat slova pro dotazy k orákulu. Pak množina B (asi jazyk orákula) je dalším vstupem programu.

2. ČRF. Přidáme charakteristické funkce C_B , kde B je proměnná.

3. Programovací jazyk. Přidáme funkci B , v console se objeví dotaz, jestli slovo patří nebo nepatří do jazyka.

Poznámka 8.4. Relativní výpočet je jedním z druhů paralelizace. Pro konkrétní vstup x vzniká tzv výpočtový strom.

Důležité je, že množina konečných větví je r.s. Každou z větví lze charakterizovat pomocí

$$\langle x, v, y, n \rangle$$

Kde x je vstup, v je výstup, y je index konečné množiny kladně zodpovězených orákulem dotazů, n je index množiny negativních dotazů. Přitom

$$D_y \subseteq B, D_n \subseteq \overline{B}$$

Předpokládáme, že jazyk orákula je korektní, neboli

$$D_y \cap D_n = \emptyset$$

Tento přístup formalizace není nejvýhodnější, protože body na které se ptáme tvoří souvislý počátek přirozených čísel.

Úmluva 8.5 (Strings). Nadále pracujeme s konečné binární řetízky (string), které značíme buď $\{0, 1\}^*$, nebo $2^{<\omega}$.

Operace:

- konkaténace: $\sigma * \tau$.
- délka: $|\sigma|$
- indexování: $\sigma(0) * \sigma(1) * \dots * \sigma(|\sigma| - 1)$.
- počátek (ostrý): $\alpha \prec \beta$ ($\alpha \prec \beta$).
- počátek množiny: $\alpha \prec B$ což znamená $\alpha \prec C_B$.

Definice 8.6 (Částečně rekurzivní funkcionál). Částečně rekurzivní funkcionál je r.s. množina Φ trojic taková, že pokud platí:

$$\begin{aligned} \langle \sigma, x, y \rangle &\in \Phi \\ \langle \sigma^*, x, y^* \rangle &\in \Phi \\ \sigma &\preceq \sigma^* \end{aligned}$$

Tak $y = y^*$.

Funkcionál je funkce vyššího řádu, vrací funkce.

Q: co přesně znamená y^* .

Poznámka 8.7. Přístupy jsou ekvivalentní, protože v případě Částečně rekurzivního funkcionálu číslo na které se orákula neptáme označíme nulou v řetízku.

Příklad 8.8. Program má na vstupu x , na výstup vypíše y s použitím $\alpha \preceq B$.

Definice 8.9 (ČRF-nál zobrazení). Částečně rekurzivní funkcionál určuje částečné zobrazení:

$$\begin{aligned}\Phi(\sigma)(x) \simeq y &\iff \langle \sigma, x, y \rangle \in \Phi \\ \Phi(\tau)(x) \simeq y &\iff \text{pro nějaké } \sigma \preceq \tau : \Phi(\sigma)(x) \simeq y \\ \Phi(B)(x) \simeq y &\iff \text{pro nějaké } \sigma \preceq B : \Phi(\sigma)(x) \simeq y\end{aligned}$$

Q: proč funkcionál místo zobrazení s 2ma parametry?

Poznámka 8.10. Máme funkcionální term, který aplikujeme na 0,1 (charakteristickou funkci. Tím dostaneme funkční term. Aplikace funkčního termu na číselný term může ale nemusí davat číselnou hodnotu.

Vlastnosti 8.11 (Částečně rekurzivní funkcionál). 1. $\Phi(B)$ je korektně definováno.

2. $\Phi(B)$ je intuitivně efektivně vyčíslitelné pomocí B . Postup: efektivně generuj trojice $\langle \sigma, x, y \rangle$. Pak $\sigma \prec B$? Pokud ano, stop. Jinak pokračuj dal.

3. Výpočetní strom $\rightarrow \Phi$ vystihuje pojem efektivní vyčíslitelnosti vzhledem k B .

Definice 8.12 (T-Převoditelnost).

$$A \leq_T B$$

Pokud existuje nějaký ČRFunkcionál Φ :

$$\Phi(B) = A, \forall x (A(x) = \Phi(B)(x))$$

Taky se říká: A je B -rekurzivní, A je reflexivní vzhledem k B .

Definice 8.13 (T-Převoditelnost pro funkce). φ je B -ČRF pokud

$$\varphi(x) \simeq \Phi(B)(x)$$

Lemma 8.14 (Regularizační funkce). Existuje ORF (dokonce PRF) ρ regularizační funkce. Splňující:

1. $W_{\rho(x)} \subseteq W_x$.
2. $W_{\rho(x)}$ je ČRFunkcionál.
3. W_x je ČRFunkcionál $\Rightarrow W_{\rho(x)} = W_x$.

Důkaz. Není formální důkaz.

Budeme efektivně generovat W_x .

1. foreach($tmp = \langle \sigma, x, y \rangle \in W_x$)
2. if($W_{\rho(x),s} \cup \{tmp\}$ je regulární)
3. $W_{\rho(x)} = W_{\rho(x)} \cup \{tmp\}$ //add

□

Definice 8.15 (Numerace funkcionálu). $W_{\rho(x)}$ z lemmatu je e -tý ČRF-nál, značíme Φ_e .

$$\Phi_e(B)(x) \simeq y \iff \langle \sigma, x, y \rangle \in W_{\rho(x)} : \sigma \prec B$$

Taky za s kroků: $\Phi_{e,s}(B)(x)$.

Pozorování 8.16 (ČRF-nál r.s.). $\Phi_e(\sigma)(X) \downarrow$ je r.s.

$\Phi_{e,s}(\sigma)(X) \downarrow$ je rekurzivní.

$\Phi_e(B)(X) \downarrow$ je r.s. v B .

$\Phi_{e,s}(B)(X) \downarrow$ je rekurzivní v B .

Věta 8.17 (s-m-n pro Relativní). // *todo zneni*

Důkaz. Nemůžeme rovnou použít standardní s-m-n větu, protože ne každá W_x splňuje funkční vlastnost. Proto potřebujeme Regularizační funkce lemma 8.14.

Pak $\Phi_e(B)(x)$ je univerzální B -ČRF. Pak platí

$$\forall B, \forall x_1, \dots, x_m, y_1, \dots, y_n : \Phi_e(B)(x_1, \dots, x_m, y_1, \dots, y_n) \simeq \Phi_{\overline{s_m}(e, x_1, \dots, x_m)}(B)(y_1, \dots, y_n)$$

Kde $\overline{s_m}$ jsou ORF (dokonce PRF).

Formálně, uděláme ČRF takovou, že

$$\alpha(e, x, w) \downarrow \iff w = \langle \sigma, y, t \rangle : \langle \sigma, \langle x, y \rangle, t \rangle \in W_{\rho(x)}$$

S tím, že

$$\alpha(e, x, w) \simeq \varphi_{s_2(a, e, x)}(w)$$

Položme

$$\overline{s_2}(e, x) = s_2(a, e, x)$$

Rozbor: s pomoci σ orákulu a vstupu y vypočti t jestliže

$$\Phi_e(\sigma)(\langle x, y \rangle) = t$$

Což znamená (pro jednoduchost pro 2 proměnné):

$$\Phi_e(B)(\langle x, y \rangle) \simeq \Phi_{\overline{s_2}(e, x)}(B)(y)$$

□

Vlastnosti 8.18 (Turingovská převoditelnost). 1. \leq_T je reflexivní, tranzitivní.

2. A rekurzivní $\Rightarrow \forall B : A \leq_T B$. Pokud umíme počítat A , tak to můžeme udělat s libovolným orákulem bez dotazů.

3. B rekurzivní $\wedge A \leq_T B \Rightarrow A$ je rekurzivní. Pro dotazy k orákulu B použijeme TS který rozhoduje B . Jako vnoření vypočtu v složitosti.

Definice 8.19 (Turingovská ekvivalence).

$$A =_T B \iff A \leq_T B \wedge B \leq_T A$$

Definice 8.20 (T Stupně převoditelnosti).

$$\deg_T(A) = \{B : B \leq_T A\}$$

Poznámka 8.21. $\{\varphi_e\}_x$ a $\{\Phi_e(\emptyset)(x)\}$ jsou různá vyjádření právě všech ČRF. Jsou rekurzivně izomorfní: máme efektivní překladač mezi těmito systémy.

Definice 8.22 (B-rekurzivní spočetnost). A je B -r.s. právě když

$$A = \text{dom}(\Phi_e(B))$$

Značení 8.23 (e-ta B-r.s. množina).

$$W_e^B = \text{dom}(\Phi_e(B))$$

Podobně za s kroků:

$$W_{e,s}^B = \text{dom}(\Phi_e(B))$$

Definice 8.24 (T-úplnost). A je T -úplná právě když je r.s. a platí

$$\forall B \in r.s. : B \leq_T A$$

Taky

$$A <_T B \iff A \leq_T B \wedge B \not\leq_T A$$

8.2 Struktura T-stupňů

Definice 8.25 (T stupně struktura). T -stupně tvoří horní polosvaz (upper semilattice), označují se $\mathcal{D}(\leq)$.

Nechť a, b třídy ekvivalence v $\mathcal{D}(\leq)$. Pak

$$a \leq b \iff \exists A \in a, \exists B \in b : A \leq_T B$$

Definice 8.26 (Join).

$$A \text{ join } B = A \oplus B = \{2x : x \in A, 2x+1 : x \in B\}$$

Vlastnosti 8.27 (Join). • $A \leq_T A \oplus B$.

- $B \leq_T A \oplus B$.
- $B \leq_T C \wedge A \leq_T C \Rightarrow A \oplus B \leq_T C$.

8.3 Relativizace dřívějších výsledků

Pozorování 8.28. • Postová věta: A je B -rekurzivní $\iff A, \bar{A}$ jsou B -r.s.

- r.s. které mají enumerator jsou efektivně generovatelné. Podobně, B -r.s která má enumerator je efektivně generovatelná relativně k B .
- r.s. množiny jsou právě ty, které lze vyjádřit pomocí \exists (rekurzivní podmínka). Podobně: B -r.s. množiny jsou právě ty, které lze vyjádřit pomocí $\exists(B\text{-rekurzivní podmínka})$.

Q: co je B -rekurzivní podmínka? Zahrnuje taky $y \in B$?

8.4 Operace skoku

Definice 8.29 (Jump). Skok neboli relativizovaný Halting problém.

$$A' = \{x : \Phi_x(A)(x) \downarrow\} = \{x : x \in W_x^A\}$$

Věta 8.30 (Vlastnosti skoku). 1. A' je r.s.

2. A' není A -rekurzivní & $\overline{A'}$ není A -r.s.

3. B je A -r.s. $\iff B \leq_1 A'$.

4. B je A -r.s. & $A \leq_1 C \Rightarrow B$ je C -r.s.

5. $A \leq_T B \iff A' \leq_1 B'$

6. $A \equiv_T B \iff A' \equiv_1 B'$

Kde \equiv je znak rekurzivní izomorfie.

Důkaz. 1. z definice, A' je definičním oborem programu $\Phi_x(A)(x)$.

2. Cantorová diagonální metoda. Formálně:

$$\overline{A'} = \{x : x \notin W_x^A\} \Rightarrow \forall x : \overline{A'} \neq W_x^A$$

Pak z relativní Postové věty ?? 8.28: A' není A -rekurzivní.

3. " \Leftarrow ". Necht $B \leq_1 A'$. Pak

$$\exists f \in ORF : x \in B \iff f(x) \in A'$$

Z toho můžeme spočítat $f(x)$ a pokud $f(x) \in A$ tak $x \in B$. Což je program pro rozhodnutí B .

" \Rightarrow " Pomoci fiktivní proměnné. Sestavíme

$$\alpha(x, y, w) \downarrow \iff y \in W_x$$

Dle s-m-n věty 8.17

$$\alpha(x, y, w) \simeq \varphi_{h(x, y)}(w)$$

Dosadíme za fiktivní proměnnou $w = h(x, y)$. Pak

$$h(x, y) \in K \iff y \in W_x$$

Zvolme $x = x_0$, pak $h(x_0, y)$ 1-převádí W_{x_0} na K .

V relativním případě máme W_x^A a A' místo K .

4. Když B je A -r.s. a $A \leq_T C$. Víme:

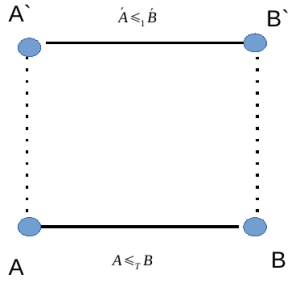
$$B = \text{dom}(\Phi_e(A))$$

v průběhu výpočtu se objeví dotazy $z \in A$? Vnoříme pro každý dotaz proceduru, která rozhoduje A pomocí C . Pak

$$B = \text{dom}(\Phi_i(C))$$

Pozor, důkaz není formální.

5. Máme následující diagram:



" \Rightarrow ". Necht $A \leq_T B$. Víme z 1), že A' je A -r.s. Podle 4) $A \leq_T B \Rightarrow A'$ je B -r.s. Tedy dle 3) protože B' je "nejtěžší" mezi B -r.s (je 1-úplná pro B -r.s):

$$A' \leq_1 B'$$

" \Leftarrow ". Necht $A' \leq_1 B'$. Triviálně: A, \bar{A} jsou A -rekurzivní proto dle relativní Postové věty ?? 8.28: A, \bar{A} jsou A -r.s.

Pak dle 3) A' je 1-úplná pro všechny A -r.s.

$$A, \bar{A} \leq_1 A'$$

Z předpokladu, relace je tranzitivní

$$A, \bar{A} \leq_1 B'$$

Proto A, \bar{A} jsou B -r.s. Konečně dle relativní Postové věty ?? 8.28 A je rekurzivní v B :

$$A \leq_T B$$

6. Přímý důsledek 5), aplikujeme relaci na obou stranách. □

Poznámka 8.31.

$$\deg_T(A) = \{B : A \equiv_T B\}$$

Se po skoku zobrazí na třídu 1-ekvivalence A' .

Definice 8.32 (Jump na T-stupních). \underline{a}' skok T-stupně \underline{a} je třída

$$\{B : B \geq_T A', A \in a\}$$

Na volbě A nezáleží protože relace \equiv_T je ekvivalence.

Poznámka 8.33. Skok lze iterovat:

$$(A)^0 = A, (A)^{(n+1)} = (A^{(n)})'$$

Taky všechny konečné:

$$A^{(\omega)} = \{\langle x, y \rangle : x \in A^{(y)}\}$$

Analogicky na třídách

$$\underline{a}^0 = \underline{a}, (\underline{a})^{(n+1)} = (\underline{a}^{(n)})'$$

Pozorování 8.34 ($\deg_T(\emptyset)$).

$$\underline{0} = \deg_T(\emptyset)$$

Což jsou právě všechny rekurzivní množiny.

Lemma 8.35 (K a \emptyset'). K a \emptyset' jsou různá vyjádření Halting problému. Jsou rekurzivně izomorfní: máme efektivní překladač mezi těmito systémy.

Důkaz. Protože \emptyset' je \emptyset -r.s. $\Rightarrow \emptyset'$ -r.s. $\Rightarrow \emptyset' \leq_1 K$.

Opačně K je r.s. (absolutně) $\Rightarrow \emptyset$ -r.s. $\Rightarrow K \leq_1 \emptyset$. □

8.5 Stejnoměrnost

Tvrzení o skoku platí stejnoměrně (jako v analýze).

Věta 8.36 (Stejnoměrnost skoku).

$$\exists z_0 \forall A (W_{z_0}^A = A')$$

Existuje pevný program, který funguje pro všechny množiny.

Důkaz.

$$W_{z_0} = \{\langle \sigma, x, y \rangle : \langle \sigma, x, y \rangle \in W_{\rho(x)}\}$$

Protože chceme $\Phi_x(A)(x)$. Pak W_{z_0} je regulární, protože prvky bereme z regulární množiny $W_{\rho(x)}$. Dal

$$x \in A' \iff \Phi_x(A)(x) \downarrow \iff \exists \sigma, \exists y (\langle \sigma, x, y \rangle \in W_{\rho(x)} \wedge \sigma \prec A) \iff x \in W_{z_0}^A$$

Podobně:

$$\exists f \in ORF \forall A, B, \forall z : A = \Phi_z(B) \Rightarrow A' \leq_1 B' \text{ pomoci } \varphi_{f(z)}$$

Kde $\varphi_{f(z)}$ je ORF, prostá. □

9 Limitní vyčíslitelnost

9.1 Limitní vyčíslitelnost pro ORF

Definice 9.1 (Limitní vyčíslitelnost). A je *limitně vyčíslitelná* právě když

$$\exists f \in ORF, \forall x : A(x) = \lim_s f(x, s)$$

Taky je známá jako *efektivní aproximace*.

Poznámka 9.2. V diskretním prostoru, limita znamená že hodnota se stabilizuje (od určitého okamžiku je vždy 0 nebo 1).

Věta 9.3 (Limitní vyčíslitelnost).

$$A \leq_T \emptyset' \iff A \text{ je limitně vyčíslitelná}$$

Věta je jednodušší verzi limitní vyčíslitelnosti, protože pro A máme všude definovanou charakteristickou funkci C_A .

Důkaz. " \Leftarrow "Necht

$$\exists f \in ORF, \forall x : A(x) = \lim_s f(x, s)$$

Hledáme místo, od kterého se hodnota funkce nemění:

$$\mu_s(\forall j \geq s : (f(x, j) = f(x, s)))$$

$\forall j \geq s : f(x, j) = f(x, s)$ je \emptyset' -rekurzivní, protože negace je

$$\exists j \geq s : (f(x, j) \neq f(x, s))$$

Vnitřek je rekurzivní + existenční kvantifikátor dává r.s. (while cyklus). R.S. jsou $\geq_1 \emptyset'$, což implikuje taky $\geq_T \emptyset'$. Pokud vrátíme negaci, tak je formule je r.s. (rekurzivita je uzavřená na negaci).

Pak máme

$$\mu_s(\emptyset' \text{--rekurzivní podmínka})$$

Neboli je to \emptyset' -ČRF. Je to procedura, na kterou Halting problém umí odpovědět. Nejmenší s hledáme ve while cyklu.

Jelikož $A(x) = C_A(x)$ je všude definovaná, tak existuje příslušná limita:

$$\exists s \forall j \geq s : (f(x, j) = f(x, s))$$

Neboli je to dokonce \emptyset' -ORF, z čehož

$$A \leq_T \emptyset'$$

" \Rightarrow "Předpokládáme $A \leq_T \emptyset'$, ekvivalentně:

$$\Phi_z(\emptyset')(x) = A(x)$$

Potřebujeme konvergentní posloupnost, která aproximuje A .

$$f(x, s) = \begin{cases} \Phi_{z,s}(\emptyset'_s)(x) & \text{pro } \Phi_{z,s}(\emptyset'_s)(x) \downarrow \\ s+1 & \text{pro jinak} \end{cases}$$

Jelikož \emptyset' je r.s., lze jí generovat po krocích. Takže \emptyset'_s je \emptyset' za s kroků. Zřejmé:

$$\emptyset' = \bigcup_s \emptyset'_s$$

Pak

$$A'_s = W_{z_0,s}^A$$

Z definice, $f \in ORF$.

Ověříme existence limity. Pro dané x :

$$\Phi_z(\emptyset')(x) = A(x) \downarrow$$

Neboli musí existovat konečný začátek, pomocí kterého počítáme:

$$\exists \sigma \prec \emptyset' \Phi_z(\sigma)(x) = A(x)$$

Taky počet kroků je konečný:

$$\exists s_1 : \Phi_{z,s_1}(\sigma)(x) = A(x)$$

σ je konečný začátek \emptyset'

$$\sigma \prec \emptyset' \Rightarrow \forall j < |\sigma| : \sigma(j) = 1 \iff j \in \emptyset'$$

Najdeme maximum s_2 takový, že

$$\sigma \prec \emptyset'_{s_2} \wedge \forall j \geq s_2 (\sigma \prec \emptyset'_j)$$

Tedy $\sigma \prec \emptyset'$.

Vezmeme $s := \max(s_1, s_2)$, pak platí:

$$f(x, s) = A(x)$$

s_1 nám stačí na generování $A(x)$, s_2 zaručí, že σ je korektní aproximace \emptyset' . □

Definice 9.4 (Modulus limity (P)).

$$m(x) = \mu_s(\forall j \leq x, \forall t \leq s : f(j, t) = f(j, s))$$

Definice 9.5 (Weak Modulus(P)). Nebo taky *first true moment*. Okamžik kdy aproximativní posloupnost se rovná $A(x)$. Ještě nezaručuje, že nadále bude stabilní. Až v moment úplného modulu.

Věta 9.6 (Limitní vyčíslitelnost (P)). *Nechť g je všude definovaná funkce, A je r.s.*

$$g \leq_T A \iff g = \lim_s(x, s)$$

Kde modulus limity $\leq_T A$.

Podmínka se jmenuje definite stable.

Důkaz. " \Leftarrow " Pokud $m(x) \leq_T A \Rightarrow g \geq_T A$. Plyne z vlastnosti modulu.

" \Rightarrow " Inspekce důkazu předchozí věty 9.3.

V průběhu výpočtu se ptáme, jestli σ je již korektní začátek A . Pokud ne, pokračujeme dál. □

9.2 Obecná limitní vyčíslitelnost

Věta 9.7 (Limitní vyčíslitelnost (P)). 1. Pokud $f \in ORF \Rightarrow \lim_s f(x, s)$ je \emptyset' -ČRF.

2. Každá \emptyset' -ČRF je limitně vyčíslitelná:

$$\exists h \in ORF : \Phi_z(\emptyset')(x) \simeq \lim_x h(z, x, s)$$

Aproximace je **uniformní**.

Důkaz. 1) pokud $g(x) \simeq \lim_s f(x, s)$, tak jako v důkazu 9.3:

$$\mu_s(\forall j \geq s : f(x, j) = f(x, s))$$

Hledáme s od kterého funkce se stabilizuje, a existuje limita. Pak

$$g(x) \simeq f(x, \mu_s(\dots))$$

Jako i minule je to \emptyset' -ČRF. Problém ale je, že s nemusí existovat, pak ale $g(x) \uparrow$.

2) Je potřeba dokazovat opatrněji.

$$h_0(z, x, s) = \begin{cases} \langle \sigma, x, y \rangle & \text{pro } \sigma \prec \emptyset'_s \wedge \Phi_{z,s}(\sigma)(x) = y \\ s+1 & \text{pro jinak} \end{cases}$$

V předchozím důkazu bylo možné dávat přímo hodnotu výstupu a čekat pokud se stabilizuje. Nyní ale čekáme na stabilizaci celého aproximačního výpočtu.

V průběhu výpočtu si ukládáme nejen výstup y ale i jak jsme se k tomu dostali. Pak vydělíme y pokud bude vyhovující:

$$h(z, x, s) = \begin{cases} (h_0(z, x, s))_{3,3} & \text{pro } h_0(z, x, s) = h_0(z, x, s-1) \\ s+1 & \text{pro jinak} \end{cases}$$

Q: proč je dostatečné zkontrolovat 2 body $s, s-1$?

Podobně jako v předchozím důkazu najdeme s_1, s_2 :

$$\Phi_z(\emptyset')(x) \downarrow \Rightarrow \exists s_2 \forall j \geq s_2 : \sigma \prec \emptyset'_j$$

Pak pro $s = \max(s_1, s_2)$ se stabilizuje celý výpočet.

$$h_0(z, x, s) = \langle \sigma, x, y \rangle \Rightarrow \lim_s h(z, x, s) = y$$

2) Pokud existuje limita, tak z definice h musí existovat i $\lim_s h_0(z, x, s)$. Pak

$$\lim_s h_0(z, x, s_0) = \langle \sigma, x, y \rangle$$

Pak pro s_0 platí: $\forall j \geq s_0 : \sigma \prec \emptyset'_j \Rightarrow \sigma \prec \emptyset'$. Dohromady:

$$\Phi_z(\emptyset')(x) \downarrow \wedge \Phi_z(\emptyset')(x) = y \wedge \sigma \prec \emptyset'$$

Q: $(s+1)$ je fiktivní hodnota, která rozhodí limitu? □

Věta 9.8 (Limitní vyčíslitelnost relativní (P)). 1. Pokud $f \in A-ORF \Rightarrow \lim_s f(x, s)$ je A' -ČRF.

2. Každá A' -ČRF je limitně vyčíslitelná

$$\exists l, \forall s \forall z \forall x : \Phi_z(A')(x) \simeq \Phi_l(A)(z, x, s)$$

Na levé straně je A -ČRF, na pravé je limita A -ORF. Aproximace je **uniformní**.

Důkaz. Relativizace předchozího důkazu. □

Důsledek 9.9 (Jump and lim). Jeden skok definition 8.29 odpovídá 1 limitnímu přechodu.

Věta 9.10 (Limitní vyčíslitelnost relativní nejobecnější (P)). Pokud $h \in ORF$:

$$\Phi_z(\emptyset^{(n+1)})(x) \simeq \lim_{s_0} \dots \lim_{s_n} h(z, x, s_0, \dots, s_n)$$

Poznámka 9.11 (R.S. limitní hierarchie). • Pokud A je rekurzivní, tak $f(x, 0) = f(x, s) = A(x)$. Není potřeba aproximovat.

- A je r.s. tak $f(x, s) = 1 \iff x \in A_s$. BUNO $f(x, 0) = 0$. Pak A je tzv 1-r.s., protože v každém sloupci dochází k nejvýš 1 změně.
- 2-r.s. (2 změny) bude rozdílem dvou r.s. množin.
- analogicky n -r.s. je booleovským rozdílem n množin.
- ω -r.s. nejvýš ω změn v každém sloupci. Ekvivalentně: $\exists h \in ORF : \# \text{ změn} \leq h(x)$. Obecně limita existuje, ale nejde udělat žádný odhad počtu změn ve sloupci.

10 Aritmetická hierarchie

Q: k čemu je dobrá aritmetická hierarchie? Kde se používá/aplikuje? Totiž všechno nad Halting problémem není efektivně vyčíslitelné.

Poznámka 10.1. Aritmetická hierarchie je v jistém smyslu efektivní verzi Borelovské hierarchie. Vezmeme konečně mnoho intervalu, budeme střídat \cup, \cap .
 \cup odpovídá \exists a $\cap - \forall$.

Poznámka 10.2. Podobná konstrukce jako polynomiální hierarchie v teorii složitosti.

Definice 10.3 (Σ_n, Π_n). Σ_n resp Π_n prefix je skupina (aritmetických kvantifikátoru). Σ_n začíná \exists , Π_n naopak \forall .

Každá ze skupin je homogenní - několik kvantifikátoru stejného typu, např $\exists\exists\exists$ nebo $\forall\forall$.

Definice 10.4 (Redukovaný prefix). Každá ze skupin obsahuje pouze jeden kvantifikátor.

Příklad 10.5. $\exists\exists\forall\exists\exists$ je Σ_3 .

$\exists\forall\exists$ je redukovaný Σ_3 .

Poznámka 10.6. Aritmetická hierarchie se označuje Σ_n^0, Π_n^0 protože kvantifikace je přes \mathbb{N} (aritmetická). Dolní index označuje počet střídavých kvantifikátoru.

Σ_n^1, Π_n^1 by byla kvantifikace navíc přes funkce $f : \mathbb{N} \rightarrow \mathbb{N}$.

Definice 10.7 (Predikát $\in \Sigma_n$). Predikát (resp množina) je ve třídě $\Sigma_n(\Pi_n)$ jestliže je vyjádřitelný ve tvaru $\Sigma_n(\Pi_n)$ prefix na rekurzivní základ (ORP).

Podobně pro relativní:

$\Sigma_n^{0,A}$, predikáty jsou A -ORP.

Pozorování 10.8. $\Sigma_0^0 = \Pi_0^0$ jsou právě rekurzivní predikáty.

Q: rovnost plyne z toho, že můžeme prohodit kvantifikace? Na vyšších úrovních neplatí protože...??

Definice 10.9 (Aritmetický predikát). Predikát je *aritmetický* právě když ho lze vyjádřit pomocí logiky 1. řádu, kde atomické části jsou rekurzivní.

Pozorování 10.10. Predikát je aritmetický právě když patří do Σ_n nebo Π_n .

Důkaz. " \Leftarrow " zřejmé.

" \Rightarrow " Úpravou výrazu do prenexního normálního tvaru. Z logiky, libovolnou formuli lze do tohoto tvaru převést. \square

Poznámka 10.11. Pokračováním do rekurzivních ordinálů lze studovat hyperaritmetickou hierarchii. Dal analytická hierarchie.

V rámci kurzu končíme u konečných, neboli ω .

Příklad 10.12. Množina Tot definition 5.16 je v Π_2^0 .

Důkaz.

$$x \in Tot \iff \forall y \exists s : \varphi_{x,s}(y) \downarrow$$

Máme 2 střídavé kvantifikátory, $\varphi_{x,s}(y) \downarrow$ je rekurzivní. \square

Věta 10.13 (Omezené kvantifikátory). *Omezené kvantifikátory nezvyšují složitost (lze prohodit doprava).*

Důkaz. Rozebereme 2 případy dle typu omezeného kvantifikátoru na začátku formule $(\forall_{x \leq t}, \exists_{x \leq t})$.

BUNO máme formuli v PNF:

$$\forall_{x \leq t} \exists y(\dots)$$

vezmeme w jako kodování $(t+1)$ -tice, pak formuli lze ekvivalentně upravit na následující tvar

$$\exists w \forall_{x \leq t} (\dots (w)_{t+1, x} \dots)$$

Protože existence svědka pro všechny $x \leq t$ je stejný jako říct, že existuje skupina $(t+1)$ svědků.

Pak existenční kvantifikátor, BUNO máme formuli:

$$\exists_{x \leq t} \forall y(\dots)$$

Použijeme negaci a předchozí případ:

$$\forall_{x \leq t} \exists y \neg(\dots)$$

$$\exists w \forall_{x \leq t} \neg(\dots (w)_{t+1, x} \dots)$$

Teď odstraníme negaci:

$$\forall w \exists_{x \leq t} (\dots)$$

V libovolné formuli můžeme postupem popsaném nahoře posunout omezené kvantifikátory doprava. Pak dle věty o omezené kvantifikace 3.9 ORP a omezený kvantifikátor jsou dohromady ORP. Omezený kvantifikátor lze nahradit konečnou disjunkce/konjunkce. \square

Věta 10.14 (Redukovaný prefix). *Libovolnou formuli lze převést do redukovaného prefixu.*

Důkaz. Znovu 2 případy dle typu kvantifikátoru.

Podobně jako ve větě o neomezené kvantifikace 3.10 nahradíme n kvantifikátoru jediným kvantifikátorem n -tice.

$$\exists x \exists y \rightarrow \exists w ((w)_{2,1} \dots (w)_{2,2})$$

Analogicky pro \forall . \square

Příklad 10.15. Množina $\text{Rec} = \{x : W_x \text{ je rekurzivní} \} \in \Sigma_3^0$.

Důkaz. Dle Postové věty 2.15:

$$x \in \text{Rec} \iff \exists y (W_x \cup W_y = W \wedge W_x \cap W_y = \emptyset)$$

Neboli $W_y = \overline{W_x}$, dohromady vyplní celý prostor, ale průnik je prázdný.

Přepíšeme formuli:

$$\forall y (\forall z (z \in W_x \cup W_y) \wedge \forall z (z \notin W_x \cap W_y))$$

Po krocích:

$$\forall y (\forall z \exists s (z \in W_{x,s} \cup W_{y,s}) \wedge \forall z \forall s (z \notin W_{x,s} \cap W_{y,s}))$$

Pak šikovně vytáhneme jeden všeobecný kvantifikátor z levé části a 2 všeobecné z pravé části. V posledním kroku vytáhneme existenční kvantifikátor z levé části. Čímž dostaneme

$$\exists \forall \exists (\dots) \in \Sigma_3^0$$

\square

Věta 10.16 (Základní vlastnosti hierarchie). 1. $A \in \Sigma_n \iff \bar{A} \in \Pi_n$

2. $B \in \Sigma_n(\Pi_n) \Rightarrow \forall m > n : B \in \Sigma_m \cup \Pi_m$.

3. $A \leq_m B \wedge B \in \Sigma_n(\Pi_n) \Rightarrow A \in \Sigma_n(\Pi_n)$

Důkaz. 1. Plyne z De Morgan pravidla. Negace mění kvantifikátor na opačný.

2. Přidáme redundantní kvantifikátory přes fiktivní proměnné.

Pokud jdeme směrem $\Sigma_n \rightarrow \Sigma_{n+1}$, tak přidáme kvantifikátor na konec prefixe. Opačně $\Sigma_n \rightarrow \Pi_{n+1}$, přidáme kvantifikátor na začátek prefixe.

3. dle definice $\leq_m \exists f \in ORF$:

$$x \in A \iff f(x) \in B$$

$f(x)$ můžeme jednoduše kvantifikovat.

□

10.1 Numerace

Věta 10.17 (Neexistence Σ_0^0 univerzálního ORP). Třída $\Sigma_0^0 = \Pi_0^0$ nemá univerzální ORP (rekurzivní numerace).

Důkaz. Pomoci Cantorové diagonální metody. Nechť $R(e, x)$ je ORP.

Pak musí platit:

$$\neg R(e, e) = R(a_0, e)$$

položme $a_0 = e$ a dostáváme spor.

□

Poznámka 10.18. Univerzální ČRF, neboli univerzální r.s. predikát 3.3 je univerzální Σ_1^0 2 proměnných pro třídu Σ_1^0 1 proměnné.

Věta 10.19 (O numeraci, univerzálním predikátu). Pro $(n \geq 1)$ třída $\Sigma_n(\Pi_n)$ má univerzální $\Sigma_n(\Pi_n)$ predikát.

Tedy máme $\Sigma_n(\Pi_n)$ -indexu.

Důkaz. Pro Σ_n, Π_n analogicky.

Nechť máme Σ_n predikát. Nechť n liché. Pak je tvaru

$$\exists \forall \dots \exists Q(\dots)$$

Ořízneme poslední existenční kvantifikátor a predikát, dle věty o univerzálním r.s. predikátu 3.3:

$$\exists y_n Q(\dots, y_n) = \exists y_n T_n(e, \dots, y_n)$$

Tím dostaneme vyjádření přes univerzální predikát:

$$\exists y_1, \forall y_2, \dots \exists y_n T_n(e, y_1, y_2, \dots, y_n)$$

Pokud n je sudé, tak máme predikát:

$$\exists \forall \dots \forall Q(\dots)$$

Znovu použijeme negaci na

$$\forall Q(\dots) = \exists \neg Q(\dots)$$

Což je r.s., proto se rovná univerzálnímu predikátu:

$$\exists \neg Q(\dots) = \exists T_n(e, \dots)$$

zpět negace:

$$\exists T_n(e, \dots) = \forall y_n \neg T_n(e, \dots)$$

Dohromady:

$$\exists y_1, \dots, \forall y_n \neg T_n(e, y_1, \dots, y_n)$$

□

Poznámka 10.20. Ve třídě složitosti nemáme univerzální polynom, proto $P \neq NP$ problém.

Důsledek 10.21 (Predikat mimo Σ_n^0, Π_n^0). Pro $(n \geq 1) \Sigma_n^0 - \Pi_n^0 \neq \emptyset$.

Důkaz. Pro $n = 1$ máme $K \in \Sigma_1^0 - \Pi_1^0$.

Pro ostatní n stejný důkaz. Necht $U(e, x)$ je univerzální predikát pro Σ_n^0 . Kdyby $U(e, e) \in \Pi_n^0 \Rightarrow \neg U(e, e) \in \Sigma_n^0$. Z existence univerzálního $\neg U(e, e)$ má index i . Dosadíme index, dostaneme spor

$$U(i, i) = \neg U(i, i)$$

Tedy $U(i, i) \notin \Pi_n^0$.

□

Definice 10.22 (Δ_n).

$$\Delta_n = \Sigma_n \cap \Pi_n$$

Definice 10.23 (Σ_n^0 -úplnost). B je Σ_n^0 -úplná právě když $B \in \Sigma_n^0$ a

$$\forall A \in \Sigma_n^0 : A \leq_1 B$$

Věta 10.24 (O aritmetické hierarchii). 1. $\emptyset^{(n)}$ je Σ_n^0 -úplná pro $(n \geq 1)$.

2. A je r.s. v $\emptyset^{(n)} \iff A \in \Sigma_{n+1}^0$.

3. $A \leq_T \emptyset^{(n)} \iff A \in \Delta_{n+1}$.

Tato věta propojuje skok definition 8.29 s aritmetickou hierarchií a vyjádřitelností v PA.

Důkaz. Indukci, pro $n = 0$ platí, protože

$$A \text{ je r.s. } \iff A \in \Sigma_1^0$$

1. z vlastnosti operace skoku definition 8.29 je $\emptyset^{(n+1)}$ r.s. v $\emptyset^{(n)}$. Podle 2) $\emptyset^{(n+1)} \in \Sigma_{n+1}^0$. Pak $\emptyset^{(n+1)}$ je Σ_{n+1}^0 -úplná.

2. " \Leftarrow " Jelikož

$$A \in \Sigma_{n+1}^0$$

A lze vyjádřit jako

$$\exists \forall \dots Q(\dots)$$

Ořízneme od prvního kvantifikátoru

$$\forall \dots Q(\dots) \in \Pi_n^0$$

Použijeme trik s negací jako ve větě o numeraci 10.19. Čímž dostaneme predikát $P \in \Sigma_n^0$ který je dle i.p. $P \leq_1 \emptyset^{(n)}$. Tedy i $P \leq_T \emptyset^{(n)}$. Dáme zpět negace a dostaneme predikát tvaru

$$\exists(\emptyset^{(n)} \text{ rekurzivní relace})$$

Z toho A je r.s. v $\emptyset^{(n)}$.

" \Rightarrow ". Lze dokázat 2ma způsoby:

a) Jelikož A je r.s. v $\emptyset^{(n)}$. Tak

$$A = \text{dom}(\varphi)$$

Kde φ je $\emptyset^{(n)}$ -ČRF. Dále

$$x \in A \iff \varphi(x) \downarrow \Rightarrow \Phi(\emptyset^{(n)})(x) \downarrow$$

Kde $f = \Phi(\emptyset^{(n)})$ je $\emptyset^{(n)}$ -ČRF. Jelikož, $\exists s : \Phi(\sigma)(x) \simeq y$ je taky existenční, je to ekvivalentní

$$\exists \sigma \exists y (\Phi(\sigma)(x) \simeq y \wedge \sigma \prec \emptyset^{(n)})$$

Máme následující kvantifikátory:

$$\exists(\exists \wedge \sigma \prec \emptyset^{(n)})$$

Tvrdíme, že $(\sigma \prec \emptyset^{(n)})$ je $\Sigma_n^0 \wedge \Pi_n^0$. Protože pro $j \leq |\sigma|$:

$$\sigma(j) = 1 \Rightarrow j \in \emptyset^{(n)}$$

což dle i.p. je Σ_n^0 . Opačně:

$$\sigma(j) = 0 \Rightarrow j \notin \emptyset^{(n)}$$

což dle i.p. je Π_n^0 .

Dohromady:

$$\exists(\Sigma_n^0 \wedge \Pi_n^0)$$

Vytáhneme existenční kvantifikátor

$$\exists(\Pi_{n-1}^0 \wedge \Pi_n^0) = \Sigma_{n+1}^0$$

b) Jelikož A je r.s. v $\emptyset^{(n)}$. Tak

$$A = \text{dom}(\varphi)$$

Kde φ je $\emptyset^{(n)}$ -ČRF.

Dle věty o limitní vyčíslitelnosti 9.3

$$\varphi(x) \simeq \lim_s F(x, s)$$

kde F je $\emptyset^{(n-1)}$ -ORF. Pak

$$x \in A \iff \varphi(x) \downarrow = \exists \lim_s F(x, s)$$

Existenci limity lze zapsat:

$$\exists s_0 \forall t \geq s_0 : F(x, t) = F(x, s_0)$$

Protože jsme v diskretním prostoru, limita existuje když hodnota funkce se stabilizuje. Navíc $F(x, t) = F(x, s_0)$ je $\emptyset^{(n-1)}$ -ORF. Dle i.p. je Π_n^0 a Σ_n^0 . Vezmeme jen Π_n^0 a dostaneme predikát:

$$\exists \forall (\Pi_n^0) \in \Sigma_{n+1}^0$$

3. " \Leftarrow ". Podle 2) A je r.s. v $\emptyset^{(n)}$. Pak z vlastnosti hierarchie 10.16: $\overline{A} \in \Pi_n$. Takže \overline{A} je r.s. v $\emptyset^{(n)}$. Dohromady dle Postové Věty ?? 8.28 A je rekurzivní v $\emptyset^{(n)}$.
" \Rightarrow ". Pokud $A \leq_T \emptyset^{(n)}$ tak podle indukčního předpokladu A, \overline{A} je r.s. v $\emptyset^{(n)}$. Tedy $A \in \Delta_{n+1}$.

□

Poznámka 10.25. Předchozí věta souvisí s elementární aritmetikou, protože dle RDPM věty 3.12:

$$\Sigma_1^0 \models_{\mathbb{N}} \Sigma_1 - \text{formule}$$

Což je taky rovnost dvou polynomů.

Z 1) bodů předchozí věty:

Důsledek 10.26 (K, \overline{K}). 1. K, \emptyset' jsou 1-úplné neboli Σ_1^0 -úplné.

2. $\overline{K}, \overline{\emptyset'}$ jsou 1-úplné neboli Π_1^0 -úplné.

Věta 10.27 (Tot úplnost). Množina Tot definition 5.16 je Π_2^0 -úplná.

Důkaz. $Tot \in \Pi_2^0$, viz example 10.12.

Nechť $B \in \Pi_2^0$ libovolná. Pak

$$x \in B \iff \forall y \exists s : Q(x, y, s)$$

Kde Q je rekurzivní predikát. Uděláme program:

$$\varphi_{\alpha(x)}(y) \simeq \mu_s Q(x, y, s)$$

Pak

$$x \in B \Rightarrow \alpha(x) \in Tot$$

Protože program najde s pro všechna y . Opačně:

$$x \notin B \Rightarrow \alpha(x) \notin Tot$$

Protože $\exists y \forall s : \neg Q(x, y, s)$. Dokonce α lze udělat prostou.

Z toho

$$B \leq_1 Tot$$

□

Definice 10.28 (Fin, Inf).

$$Fin = \{x : |W_x| < \infty\}$$

$$Inf = \{x : |W_x| = \infty\}$$

Věta 10.29 (Fin úplnost). Množina Fin je Σ_2^0 -úplná.

Důkaz.

$$x \in Fin \iff \exists y \forall z, s (z \notin W_{x,s} \wedge z > y)$$

Pokud je konečná, tak od určitého místa do ní nepadne žádný prvek. y je buď maximální, nebo větší než \max .

Formule je Σ_2^0 .

Dokážeme přes komplement

$$\varphi_{\beta(x)}(y) \downarrow \iff \forall j \leq y (\varphi_x(j) \downarrow)$$

Pak

$$x \in Tot \iff \beta(x) \in Inf$$

Protože pokud není nekonečná, tak na nějakém vstupu nekonverguje a totiž není všude definována. Taky opačně:

$$x \notin Tot \iff \beta(x) \notin Inf$$

Alternativně:

$$\varphi_{\gamma(x)}(j) \downarrow \iff \exists j - \text{prvků} \in W_x$$

Pak platí

$$x \in Tot \iff \gamma(x) \in Inf$$

a i opačně. Dohromady:

$$Inf \equiv_1 Tot$$

□

Definice 10.30 (Rek).

$$Rek = \{x : W_x \text{ je rekurzivní} \}$$

Věta 10.31 (Rek úplnost (BD)). *Množina Rek je Σ_3^0 -úplná.*

11 Pokročilejší vyčíslitelnost

11.1 R.S. množiny

Definice 11.1 (Postův problém podruhe).

$$\exists A : \emptyset <_T A <_T \emptyset'$$

Byl vyřešen nezávislé Friedberg-Mučník pomocí tzv. prioritních metod.

1. finite injury \emptyset' -priority
2. infinite injury \emptyset''
3. \emptyset'' -priority

11.2 Forcing

Definice 11.2 (Cantor space). Hlavní myšlenka je použít tzv Cantorův prostor 2^ω . Což je prostor všech zobrazení

$$f : \mathbb{N} \rightarrow \{0, 1\}$$

Je úplný metrický prostor.

Poznámka 11.3. Okolí v Cantorovém prostoru je

$$o_\sigma = \{B : \sigma \prec B\}$$

Pak vzdálenost

$$\rho(\dots) \leq 2^{-\sigma}$$

Definice 11.4 (Finite extension method (Cohen)). Souvisí s Cohenovým forcingem v teorii množin kterým vyřešil hypotézu continua.

Věta 11.5 (Bairová věta o kategoriích). Máme úplný metrický prostor. Pak

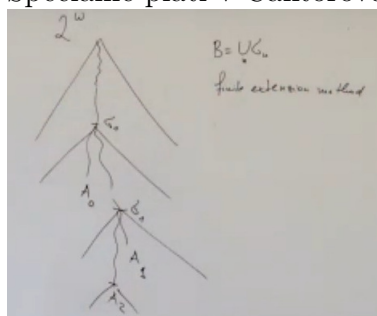
$$\bigcap_{\text{spočetná}} (\text{otevřené, husté}) \neq \emptyset$$

Jde o ekvivalentní formulace. Viz Baire category theorem.

Důkaz. Hint:

Nechť první bod je A_0 leží v otevřené husté množině. Jelikož je otevřená, existuje okolí A_0 které je uvnitř množiny. Z hustoty v tomto okolí je další bod, je taky s okolím atd. Takovým postupem vytvoříme Cauchy posloupnost, která kvůli úplnosti má limitu. Tato limita leží v průniku.

Speciálně platí v Cantorovém prostoru.



□

Definice 11.6 (1-generická). A je 1-generická když

$$\forall e \exists \sigma (\sigma \prec A \wedge (\Phi_e(\sigma)(e) \downarrow \vee (\forall \tau \geq \sigma : \Phi_e(\tau)(e) \uparrow)))$$

Buď s nějakým začátkem konverguje, nebo tzv. silně diverguje (i v okolí).

První podmínka je ekvivalentní

$$\forall \sigma \prec B : \Phi_e(B)(e) \downarrow$$

Druhá je ekvivalentní

$$\forall \sigma \prec B : \Phi_e(B)(e) \uparrow$$

Věta 11.7 (Existence 1-generické). Existuje 1-generická:

$$A \leq_T \emptyset'$$

Důkaz. Používá se finite extension method.

Pomocí \emptyset' vytvoříme \emptyset' -posloupnost $\{\sigma_n\}_n$:

$$\sigma_{n+1} \preceq \sigma_n$$

Pak

$$A = \bigcup \sigma_n$$

BUNO: $\sigma_0 = \emptyset$. Indukční krok: máme σ_n chceme další. Zkusíme:

$$\exists \sigma_e \preceq \tau : (\Phi_e(\tau)(e) \downarrow)$$

Pokud ano, vezmeme první takové a $\sigma_{e+1} = \tau$. Jinak $\sigma_{e+1} = \sigma_e$.

Otázka je 1-kvantifikátorová neboli $\leq_T \emptyset'$. Neboli \emptyset' umí rozhodnout.

□

Věta 11.8 (Kleene-Post). *Existují nerekurzivní $A, B \leq_T \emptyset'$ takové, že A, B jsou T -neporovnatelné.*

Důkaz. Pomoci \emptyset' vytvoříme monotonní \emptyset' -posloupnosti:

$$A = \bigcup \{\alpha_n\}_n, B = \bigcup \{\beta_n\}_n$$

BUNO: $\alpha_0 = \beta_0 = \emptyset$.

Indukční krok: 2 podkroky, protože potřebujeme zajistit

$$(A \not\leq_T B \simeq \Phi_e(B) \neq A) \wedge (B \not\leq_T A \simeq \Phi_e(A) \neq B)$$

Stačí jedná z podmínek, druhá symetricky.

Otázka

$$\exists \beta_e \preceq \tau : (\Phi_e(\tau)(x_0) \downarrow = 0)$$

kde $x_0 = |\alpha_e|$, $\alpha_e(x_0)$ není definované.

Pokud ANO, tak

$$\alpha_{e+1} = \alpha_e + 1, \beta_{e+1} = \tau$$

S tím, že τ první které padlo. Pak pro libovolnou $\beta_{e+1} \preceq B$ platí

$$\Phi_e(B)(x_0) = 0 \wedge A(x_0) = 1 (A \preceq \alpha_{e+1})$$

Jinak

$$\alpha_{e+1} = \alpha_e + 0, \beta_{e+1} = \beta_e$$

Pak

$$A(x_0) = 0 \wedge (B(x_0) \downarrow = 1 \vee B(x_0) \uparrow)$$

□

Věta 11.9 (R.S. a 1-generické). *Žádná rekurzivní není 1-generická.*

Důkaz. A je rekurzivní, najdeme e_0 takové, že

$$\Phi_{e_0}(A)(e_0) \uparrow \wedge \forall B \neq A : \Phi_{e_0}(B)(e_0) \downarrow$$

$$\mu_y(B(y) \neq A(y))$$

Pak máme divergenci v množině A ale nikoliv silnou (v okolí konverguje). □

Poznámka 11.10. Těch A , které nejsou 1-generické je málo. Přesněji v topologickém smyslu (2 kategorie) 1-generická jsou "všude".

Odbočka, pro funkcionál $\text{dom}(\Phi_e)$ je otevřená. Doplněk je dle definice uzavřená, vnitřek je největší otevřená, takže je ok. Zbývá hranice, a takových je málo.

Definice 11.11 (n-generická). Podobně jako 1-generická, ale místo Σ_1^0 vezmeme Σ_n^0 .

Poznámka 11.12. Problém prioritních metod.

Věta 11.13 (Low). *A je 1-generická a $A \leq_T \emptyset' \Rightarrow A$ je tzv. low.*

$$A' \equiv_T \emptyset'$$

Libovolný skok definition 8.29 je \emptyset' .

Důkaz.

□

11.3 Algoritmická náhodnost

Historicky existovali následující přístupy ve zkoumání algoritmické náhodnosti:

- frekvenční stabilita, tzv. v limitě v posloupnosti $\{a_n\}$ musí být stejně 0 a 1. Pak ještě musíme modifikovat: pro každou r.s. vybranou část.
- Kolmogorovská složitost (vice variant)
- Teorie míry (Martingale nebo Martin-Lof testy, jsou ekvivalentní).

11.4 Kolmogorovská složitost

Složitost napřed pro konečné řetízky $2^{<\omega}$ pak rozšíření na 2^ω neboli množiny.

Poznámka 11.14 (Kolmogorovské náhodné řetízky). Náhodné jsou nestlačitelné: nejde popsat kratším způsobem než délka.

"Popis" ale není přesně definovaný: polynomiální čas, ČRF, s orákulem atd. Čím máme silnější rozpoznávací mechanismus, tím více nenáhodných řetízku rozpoznáme.

Zvolíme přístup s ČRF (v literatuře spíš TM).

Definice 11.15 (Kolmogorov complexity).

$$C_f(x) = \min\{|y|, y \in \{0,1\}^* : f(y) \downarrow \wedge f(y) = x\}$$

Délka nejkratší

11.5 Martingale

List of Theorems

2.1	Definice (Primitivně rekurzivní funkce)	2
2.2	Značení (Definiční obor)	2
2.3	Značení (Obor hodnot)	2
2.4	Definice (Rekurzivní funkce)	2
2.6	Definice (λ -calculus)	3
2.9	Definice (Rekurzivní množina)	3
2.10	Definice (Rekurzivní spočetná množina)	3
2.13	Značení (Konvergence)	4
2.14	Připomenutí	4
2.16	Definice (Gödelové číslo)	4
2.18	Připomenutí (Univerzální TS)	4
2.19	Definice (Univerzální ČRF)	4
2.20	Značení (Univerzální ČRF)	5
2.22	Definice (e -ta rekurzivně spočetná množina)	5
2.24	Definice (1,m převoditelnost)	5
2.25	Definice (1-úplnost)	5
2.26	Definice (K, DIAG)	5
3.7	Definice (Omezený existenční kvantifikátor)	8
3.8	Definice (Omezený všeobecný kvantifikátor)	8
3.16	Definice (Graf ČRF)	9
3.20	Definice (Imunní množina)	10
3.21	Definice (Simple)	10
5.1	Definice (Produktivní množina)	14
5.2	Definice (Kreativní množina)	14
5.13	Definice (Úplně produktivní)	17
5.16	Definice (Totální množina)	18
6.2	Definice (Rekurzivní neoddělitelnost)	19
6.3	Definice (Efektivní neoddělitelnost)	19
6.9	Definice (1-převoditelnost dvojic)	20
6.10	Definice (1-úplnost dvojic)	20
7.1	Definice (Rozumná teorie)	22
7.2	Definice (Axiomatizovatelná teorie)	22
7.5	Definice (Reprezentovatelnost)	22
8.1	Definice (tt-Převoditelnost)	24
8.3	Definice (Formalizace relativního výpočtu)	25
8.6	Definice (Částečně rekurzivní funkcionál)	25
8.9	Definice (ČRF-nál zobrazení)	26
8.12	Definice (T-Převoditelnost)	26
8.13	Definice (T-Převoditelnost pro funkce)	26
8.15	Definice (Numerace funkcionálu)	27
8.19	Definice (Turingovská ekvivalence)	27
8.20	Definice (T Stupně převoditelnosti)	27
8.22	Definice (B-rekurzivní spočetnost)	28
8.23	Značení (e -ta B-r.s. množina)	28
8.24	Definice (T-úplnost)	28
8.25	Definice (T stupně struktura)	28
8.26	Definice (Join)	28
8.29	Definice (Jump)	29

8.32	Definice (Jump na T-stupních)	30
9.1	Definice (Limitní vyčíslitelnost)	31
9.4	Definice (Modulus limity (P))	33
9.5	Definice (Weak Modulus(P))	33
10.3	Definice (Σ_n, Π_n)	35
10.4	Definice (Redukovaný prefix)	35
10.7	Definice (Predikát $\in \Sigma_n$)	35
10.9	Definice (Aritmetický predikát)	35
10.22	Definice (Δ_n)	38
10.23	Definice (Σ_n^0 -úplnost)	38
10.28	Definice (Fin, Inf)	40
10.30	Definice (Rek)	41
11.1	Definice (Postův problém podruhe)	41
11.2	Definice (Cantor space)	41
11.4	Definice (Finite extension method (Cohen))	42
11.6	Definice (1-generická)	42
11.11	Definice (n-generická)	43
11.15	Definice (Kolmogorov complexity)	44

List of Theorems

2.15	Věta (Postova)	4
2.21	Věta (s-m-n (BD))	5
2.27	Lemma (DIAG, K)	5
2.28	Věta (K 1-úplná)	6
2.29	Věta (Rozšíření Ψ_n)	6
3.2	Lemma ($\exists Q$ r.s)	7
3.3	Věta (Univerzální Kleeneho r.s. predikát)	7
3.4	Důsledek (Index r.s. predikátů)	7
3.6	Věta (Uzavřenost RS)	7
3.9	Věta (Omezená kvantifikace (BD))	8
3.10	Věta (Neomezená kvantifikace)	8
3.12	Věta (RDPM (BD))	8
3.13	Důsledek (10. Hilbertův problém)	8
3.15	Věta (O selektoru)	9
3.17	Důsledek (Graf ČRF)	9
3.19	Věta (Postova podruhe)	9
3.23	Věta (Existence Simple)	10
4.1	Věta (O rekurzi 1)	10
4.2	Věta (O rekurzi 2)	11
4.3	Věta (O rekurzi ∞)	12
4.4	Věta (O rekurzi 3)	12
4.6	Věta (Program vlastní kod)	12
4.7	Věta (Rekurze v indexech (BD))	12
4.8	Věta (Rice podruhe)	13
4.9	Věta (O rekurzi (BD))	13
5.4	Věta (Modifikace K)	14
5.5	Věta (Produktivní funkce ORF)	14
5.6	Věta (Produktivní funkce ORF prostá(BD))	15
5.7	Věta (Nekonečná množina)	15
5.10	Lemma (Produktivita a \leq_m)	16
5.11	Věta (Ekvivalence Kreativní)	16
5.12	Důsledek (\bar{K} produktivní)	17
5.15	Věta (Úplna produktivita ekvivalence)	17
5.17	Lemma (Totalní je produktivní)	18
5.18	Důsledek (Omezení logiky)	19
6.5	Věta (Efektivní neoddělitelné (BD))	19
6.7	Věta (Existence efektivní neoddělitelné)	19
6.11	Věta (Dvojná věta o rekurzi)	20
6.12	Věta (Efektivní neoddělitelné, 1-úplnost)	21
7.7	Věta (Vztah \mathbb{N} a T)	22
7.8	Lemma (Disjunktní množiny formule(BD))	23
7.9	Věta (Gödelové věty)	23
8.11	Vlastnosti (Částečně rekurzivní funkcionál)	26
8.14	Lemma (Regularizační funkce)	26
8.17	Věta (s-m-n pro Relativní)	27
8.18	Vlastnosti (Turingovská převoditelnost)	27
8.27	Vlastnosti (Join)	28

8.30	Věta (Vlastnosti skoku)	29
8.35	Lemma (K a \emptyset')	31
8.36	Věta (Stejnoměrnost skoku)	31
9.3	Věta (Limitní vyčíslitelnost)	31
9.6	Věta (Limitní vyčíslitelnost (P))	33
9.7	Věta (Limitní vyčíslitelnost (P))	33
9.8	Věta (Limitní vyčíslitelnost relativní (P))	34
9.9	Důsledek (Jump and lim)	34
9.10	Věta (Limitní vyčíslitelnost relativní nejobecnější (P))	34
10.13	Věta (Omezené kvantifikátory)	36
10.14	Věta (Redukovaný prefix)	36
10.16	Věta (Základní vlastnosti hierarchie)	37
10.17	Věta (Neexistence Σ_0^0 univerzálního ORP)	37
10.19	Věta (O numeraci, univerzálním predikátu)	37
10.21	Důsledek (Predikat mimo Σ_n^0, Π_n^0)	38
10.24	Věta (O aritmetické hierarchii)	38
10.26	Důsledek (K, \overline{K})	40
10.27	Věta (Tot úplnost)	40
10.29	Věta (Fin úplnost)	40
10.31	Věta (Rek úplnost (BD))	41
11.5	Věta (Bairová věta o kategoriích)	42
11.7	Věta (Existence 1-generické)	42
11.8	Věta (Kleene-Post)	43
11.9	Věta (R.S. a 1-generické)	43
11.13	Věta (Low)	43

Reference

- [1] Bernard Bolzano. *Paradoxes of the Infinite (Routledge Revivals)*. Routledge, 2014.