Vyčíslitelnost I

Doc. RNDr. Antonín Kučera, CSc.

Obsah

přepsal Petr Hošek v ak. roce 2008/2009

1 Výpočetní model

Definice (Turingův stroj)**.** *Turingův stroj* je pětice $T = (A, Q, \delta, \overline{q}, \overline{F})$ kde

A je vnější abeceda stroje obsahující prázdný znak λ

Q je vnitřní abeceda stroje, tedy aktivní a pasivní stavy

δ je přechodová funkce stroje, tedy funkce $\delta : A \times Q \to Q \times \{\leftarrow, \circ, \rightarrow\} \times A$

$\overline{q}$ je počáteční stav stroje

$\overline{F}$ jsou pasivní (koncové) stavy stroje.

Poznámka 1.1. Turingův stroj provádí tři operace v jednom taktu: čtení, zápis a změnu stavu.

Definice (Konfigurace Turingova stroje)**.** Konfigurace Turingova stroje je obsah nejmenší souvislé části pásky, která obsahuje všechny neprázdné pole, čtené pole a stav stroje.

1.0.1 Operace Turingova stroje

skládání $T_1(S)T_2(S)$ kde T_1 a T_2 jsou Turingovy stroje. Nejprve dojde k provedení programu T_1, poté programu T_2. Formálně dojde k nahrazení koncových stavů T_1 počátečním stavem T_2, sloučení δ_1 a δ_2 přičemž Q_1 a Q_2 musí být disjunktní.

podmíněná rovnost $T_1(S) \simeq T_2(S)$ kde T_1 a T_2 jsou Turingovy stroje. Pokud jeden výraz má smysl (má hodnotu, je definován) pak má i druhý a platí rovnost.

konvergence $T(S)\downarrow$ kde T je Turingův stroj. Platí pokud T končí.

divergence $T(S)\uparrow$ kde T je Turingův stroj. Platí pokud T nekončí.

1.0.2 Základní modifikace Turingova stroje

(1) Turingův stroj kde $\delta : A \times Q \to Q \times \{L, R\} \times A$ (vždy pohyb) je stejně silný.

(2) Turingův stroj kde $\delta : A \times Q \to Q \times \{L\} \times A$ nebo $\delta : A \times Q \to Q \times \{R\} \times A$ (pohyb pouze jedním směrem) je stejně silný jako konečný automat.

(3) Turingův stroj s jednostrannou páskou je stejně silný (převedení na vícestopou pásku jako kartézký součin $A = (A \cup \{\lambda\}) \times (A \cup \{\lambda\})$, stroj si pamatuje stopu).

(4) Turingův stroj s omezenou činností je stejně silný (standardně tři operace v jednom taktu, stačí dvě i jedna).

(5) Turingův stroj s omezenou abecedou je stejně silný (stačí λ, | a použijeme binární kódování).

(6) Turingův stroj s omezenými stavy je stejně silný (stačí dva stavy, jeden však již nestačí).

(7) Turingův stroj je silnější než zásobníkový automat, ale lze ho simulovat pomocí dvou zásobníkových automatů.

Poznámka 1.2. Často používanou modifikací je Turingův stroj s okraji/omezovači které označují "pracovní část" pásky. Pokud při běhu stroje narazím na okraj nebo mažu znaky sousedící s okrajem, musím posunout okraje.

Definice (Univerzální Turingův stroj)**.** Máme dánu abecedu A, univerzální Turingův stroj $\mathcal{U}(\text{code}(T), \text{code}(S)) \simeq T(S)$, kde T je Turingův stroj v abecedě A.

$A = \{\lambda, |\}$, $Q = \{r_0, r_1, r_2, \ldots, r_m\}$ kde r_0 je koncový stav, r_1 počáteční stav. Kódujeme na pásku stroje $\mathcal{U}$ pásku stroje T se čteným symbolem, stav stroje a lineární kód stroje. Její obsah je tedy

$$Y \text{ blok } 1\ Y \text{ blok } 2\ \Delta \text{ blok } 3\ \times O_1\ \times O_2\ \ldots\ \times O_m\ Y$$

kde blok 1 obsahuje kód S, tedy nejmenší souvislou část pásky T, blok 2 obsahuje kód stavu T a blok 3 obsahuje čtené pole označené jako M.

Pro každý aktivní stav r_i stroje T máme dvě instrukce $r_i\lambda$ a $r_i|$, potom O_i je tvaru

$$\underbrace{\text{písmeno, směr } \square \text{ nový stav } r_i\lambda}_{r_i\lambda}\underbrace{\text{písmeno, směr } \square \text{ nový stav } r_i|}_{r_i|}.$$

Při běhu stroje kontrolujeme zda blok 2 obsahuje jedinou | která je ekvivalentní r_0. Pokud ano, pak "vyčistíme" výstup a upravíme blok 1 na výsledek. V opačném případě blok 2 obsahuje $i+1$ |. Musíme tedy najít instrukci O_i a to tak že po bitech umazávám | a označuji $\mod 2\ \square$. Podle obsahu bloku 3 najdu hledanou instrukci která je ve tvaru

$$\text{písmeno, směr } \square \text{ nový stav.}$$

Nový stav přenesu do bloku 2. V případě že směr je $\circ$ dojde k přepisu bloku 3, jinak dojde k zápisu symbolu na místo pole M.

Pozorování 1.3. *Otázka zda* $\mathcal{U}(\text{code}(T), \text{code}(S))\downarrow$ *kde* T *je Turingův stroj s daty* S *je algoritmicky nerozhodnutelný problém (neexistuje Turingův stroj který by vždy konvergoval a rozhodl, tedy* $T(S)\downarrow$*). Tento problém se nazývá* halting problém.

Důkaz. Pro spor předpokládejme že ano. Sestrojíme nový Turingův stroj Alg takový, že $Alg(K)\downarrow \iff \mathcal{U}(K,K)\uparrow$. Nechť Alg má kód Q, pak $Alg(Q)\downarrow \iff \mathcal{U}(Q,Q)\uparrow \iff Alg(Q)\uparrow$ což je spor. ■

Poznámka 1.4. Důkaz předchozího tvrzení je založen na *Cantorově diagonální metodě.*

Poznámka 1.5 (Cantorova diagonální metoda)**.** Nechť $\{A_n\}_n$, $B = \{n | n \notin A_n\}$. Pak platí $B \neq A_n$.

2 Primitivně a částečně rekurzivní funkce

Pracujeme s částečnými funkcemi $\mathbb{N}^k \to \mathbb{N}$ (aritmetické funkce). V jistém smyslu jde o funkcionální pohled, který dovoluje jak pohled na programy, tak abstrakci od programů (jenom funkce). Máme tedy jednu vyčíslitelnou funkci, a mnoho programů které ji vyčíslují.

Induktivní výstavba, máme základ (axiomy) a odvozovací pravidla (obdoba logiky 1. řádu). Pojem důkazu je definován jako odvoditelnost z axiomů.

2.0.3 Základní funkce

Jsou definovány všude (nebo-li jsou totální) a jsou efektivně vyčíslitelné.

Nulová funkce

$$o(x) \simeq 0.$$

Následník

$$s(x) \simeq x + 1.$$

Projekce

$$I_n^j(x_1, \ldots, x_n) \simeq x_j,\ 1 \leq j \leq n.$$

2.0.4 Odvozovací pravidla (operátory)

Operátor substituce $S_n^m(f, g_1, \ldots, g_m) = h$, kde

$$h(x_1, \ldots, x_n) \quad \simeq \quad f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)).$$

Operátor primitivní rekurze $R_n(f, g) = h$ kde

$$\begin{aligned} h(0, x_2, \ldots, x_n) &\simeq f(x_2, \ldots, x_n) \\ h(y+1, x_2, \ldots, x_n) &\simeq g(y, h(y, x_2, \ldots, x_n), x_2, \ldots, x_n). \end{aligned}$$

Operátor minimalizace $M_n(f) = h$ kde

$$h(x_1, \ldots, x_n) \simeq \mu_y(f(x_1, \ldots, x_n, y) \simeq y)$$

$$h(x_1, \ldots, x_n) \simeq y \iff (\forall z)_{z<y}(f(x_1, \ldots, x_n, z)\downarrow \neq 0) \ \& \ (f(x_1, \ldots, x_n, y)\downarrow = 0).$$

Poznámka 2.1. S_n^m, R_n jsou operátory které z funkcí "vytvářejí" nové funkce přičemž zachovávají totálnost (všude definovanost) a algoričmitnost (efektivní vyčíslitelnost). M_n zachovává algoritmickou vyčíslitelnost, nemusí zachovávat totálnost.

Definice (Primitivně rekurzivní funkce)**.** Třída *primitivně rekurzivních funkcí* (PRF) je nejmenší třídou funkcí, obsahující základní funkce a je uzavřená na operátory S_n^m, R_n.

Definice (Částečně rekurzivní funkce)**.** Třída *částečně rekurzivních funkcí* (ČRF) je nejmenší třídou funkcí, obsahující základní funkce a je uzavřená na operátory S_n^m, R_n a M_n.

Definice (Obecně rekurzivní funkce)**.** Třída *obecně rekurzivních funkcí* (ORF) což jsou ty ČRF, které jsou totální (tzn. všude definované).

Tvrzení 2.2. *Všechny PRF jsou totální.*

Důkaz. Indukcí triviálně z definice. ■

Tvrzení 2.3. *Všechny ČRF jsou efektivně vyčíslitelné.*

Důkaz. Zvolme programovací jazyk, fixujme programy pro vyčíslení základních funkcí, operátory zachovávají efektivní vyčíslitelnost (tedy z programů dělají programy), ■

Tvrzení 2.4. *Platí PRF $\subseteq$ ORF $\subseteq$ ČRF a navíc PRF $\neq$ ORF $\neq$ ČRF.*

Důkaz. $g(x, y) \simeq y + 1$ je jistě PRF, $\mu_y(g(x, y) \simeq 0)$ není nikde definovaná. ■

Příklad. Součet $x + y$ je PRF, Důkaz (Idea) je $0 + y = y$, $(x + 1) + y = (x + y) + 1$. Potřebuji funkci I_1^1, s, $S_3^1(s, I_3^2)$, $R_2(I_1^1, S_3^1(s, I_3^2))$ $((x + 1) + y = g(x, x + y, y))$.

Definice. Odvození funkce f ("analogie důkazu") je konečná posloupnost funkcí z nichž každá je buď základní, nebo vzniká z předchozích povoleným operátorem spolu s analýzou, kterou základní, z kterých předchozích a jakým operátorem. Poslední funkcí posloupnosti je funkce f.

Poznámka 2.5. Analogicky lze definovat pojem primitivně rekurzivního termu resp. částečně rekurzivního termu, kde o, s, I_n^j jsou základní termy a operátory S_n^m, R_n, M_n které korektním způsobem vytvářejí další termy.

Poznámka 2.6. Lze formálně ve funkcionální logice kde $R_2(I_1^1, S_3^1(s, I_3^2))$ je funkční term a $x+2$ je číselný term. Přechod mezi těmito termy je možný pomocí

(1) λ-abstrakce, $\lambda_{xy}(x + y)$ je funkce 2 proměnných x, y

(2) aplikace Ap což je vyhodnocení funkčního termu na číselném, $Ap(\lambda_{xy}(x + y), 2, 3) = 5$.

Příklad. Násobení $\lambda_{xy}(x \cdot y)$, Důkaz (Idea) je $0 \cdot y = y$, $(x+1) \cdot y = x \cdot y + y$. Potřebuji funkci $R_2(o, S_3^2(+, I_3^2, I_3^3))$ $((x+1) \cdot y = g(x, x \cdot y, y))$.

Poznámka 2.7. Pro vyčíslení PRF potřebuji programovací jazyk podobný Pascalu který má $\leftarrow$, $+1$, aritmetické konstanty a konstrukce **if then else**, **for**. Pro vyčíslení ČRF navíc **while**. Nesmí mít **goto**.

Definice. Množina M je *rekurzivní* jestliže C_M je ORF (kde C je charakteristická funkce).

Pozorování 2.8. *Množina M je* rekurzivní *jestliže existuje program, který na libovolném vstupu konverguje ($\downarrow$) a vrací hodnotu* pravda *(1) nebo* nepravda *(0) (množina M je algoritmicky rozhodnutelná).*

Definice. Množina M je *rekurzivně spočetná* jestliže funkce $\alpha(x)$ taková, že $\alpha(x)\downarrow \iff x \in M$, je ČRF.

Poznámka 2.9. $\alpha(x)\downarrow \iff x \in M$ a pokud $\alpha(x)\downarrow$, pak $\alpha(x) = 1$, tedy M jednoznačně určuje α.

Pozorování 2.10. *Množina M je* rekurzivně spočetná *jestliže je definičním oborem nějakého programu. Tedy α je ČRF pokud má program který $Pr(x)\downarrow \iff x \in M$.*

Poznámka 2.11. α je ČRF právě tehdy, když α je odvoditelná, což je právě tehdy, když existuje program, který přesně vyčísluje α.

Pozorování 2.12. *Duální k množinám jsou predikáty (vlastnosti, podmínky).*

Definice. Predikát P je *obecně rekurzivní* jesliže jeho C_P je ORF.

Definice. Predikát P je *rekurzivně spočetný* jestliže P je definičním oborem nějaké ČRF (tzn. definičním oborem nějakého programu).

Definice. Predikát P je *primitivně rekurzivní* jestliže C_P je PRF (tzn. "naprogramovatelná" pouze pomocí R_n).

Poznámka 2.13. PRF, ČRF mají výhody i nevýhody. Mezi nevýhody patří, že vše (konečné objekty, konfigurace Turingových stroju, apod.) je potřeba kódovat pomocí *přirozených čísel*.

Poznámka 2.14. Budeme kódovat primitivně rekurzivními prostředky.

Pozorování 2.15. *Všechny "rozumné" všude definované funkce jsou PRF.*

Příklad (Kódování objektů)**.**

- dvojice čísel se kóduje jako $\langle x, y\rangle = \frac{(x+y)(x+y+1)}{2} + x$, přičemž $l(\langle x, y\rangle) = x$, $r(\langle x, y\rangle) = y$
- n-tice čísel se kóduje jako $\langle x_1, \ldots, x_{n+1}\rangle_{n+1} = \langle x_1, \langle x_2, \ldots, x_n\rangle_n\rangle$, přičemž $(\langle x_1, \ldots, x_n\rangle_n)_{n,i} = x_i$
- slova v konečné abecedě, $p(i) = i$-té prvočíslo, abeceda $A = \{a_1, \ldots, a_k\}$, slovo $w = a_{i_0} \ldots a_{i_l}$ se kóduje jako $p(0)^{i_0} \cdot \ldots \cdot \ldots \cdot p(l)^{i_l}$
- konečná množina přirozených čísel, $\emptyset$ má kód 0, $\{x_1, \ldots, x_k\}$ má kód $z = \sum_{i=1}^{k} 2^{x_i}$

Není důležité jak kódujeme, ale je potřeba kódovat efektivně což lze primitivně rekurzivními prostředky.

Příklad (Příklady PRF)**.**

(1) $x \dot{-} y = \max\{0, x - y\}$

(2) $\mathrm{sg}\,(x) = 0$ pro $x = 0$ a $\mathrm{sg}\,(x) = 1$ pro $x \geq 1$

(3) $\overline{\text{sg}}(x) = 1 - \text{sg}(x)$

Příklad (Příklady PRP)**.**

(1) $x = y$

(2) $x \leq y$

(3) $x < y$

Lemma 2.16. *Spojky výrokového počtu ($\wedge, \vee, \neg$) zachovávají jak primitivní, tak obecnou rekurzivitu. Tedy průnik, sjednocení a doplněk rekurzivních množin je rekurzivní.*

Důkaz. Buď C_P charakteristická funkce P a C_Q charakteristická funkce Q. Pak platí

$$\begin{aligned} P \wedge Q &\iff C_P \cdot C_Q \\ P \vee Q &\iff \text{sg}(C_P + C_Q) \\ \neg P &\iff \overline{\text{sg}}(C_P) \end{aligned}$$

■

Lemma 2.17. *Omezené kvantifikátory zachovávají primitivní a obecnou rekurzivitu.*

Důkaz. $(\forall x)_{x \leq y}, (\forall x)_{x<y}$ je omezená konjunkce (posloupnosti $0, \ldots, y$); $(\exists x)_{x \leq y}, (\exists x)_{x<y}$ je omezená disjunkce (posloupnosti $0, \ldots, y$). ■

Věta 2.18. *Pokud P je ORP, pak P i $\neg P$ jsou rekurzivně spočetné.*

Důkaz. Víme že logická spojka $\neg$ zachovává rekurzivitu, stačí tedy dokázat pro P. Rekurzivita implikuje rekurzivní spočetnost. Je vidět, že $P(x)$ je ORP a použijeme algoritmus který je definován jako "zastav v případě že $P(x)$ je pravdivý". ■

Pozorování 2.19. *Pokud M je rekurzivní množina, pak M a $\overline{M}$ jsou rekurzivně spočetné.*

2.1 Ekvivalence Turingových strojů a částečně rekurzivních funkcí

Poznámka 2.20 (Turingovská vyčíslitelnost)**.** Fixujeme kódování přirozených čísel jako k-tic na pásku. *Turingovská vyčíslitelnost* funkce $\varphi : \mathbb{N}^k \to \mathbb{N}$ znamená, že existuje Turingův stroj T, který při zvoleném kódování vyčísluje φ.

Věta 2.21. *Každá ČRF je Turingovsky vyčíslitelná.*

Důkaz. Třída ČRF byla definována indukcí, dokážeme tedy indukcí. Pro základní funkce věta jistě platí, stejně tak pro základní operátory. Vše ostatní je vytvořeno pomocí induktivní výstavby. ■

Poznámka 2.22. Konečné objekty kódovány přirozenými čísly, tedy primitivně rekurzivními prostředky.

Věta 2.23. *Turingův stroj lze reprezentovat pomocí ČRF.*

Důkaz. Lze to co dělá Turingův stroj reprezentovat pomocí ČRF? Turingovy stroje nejsou induktivně definované. Je nutné kódování konfigurací Turingova stroje pomocí přirozených čísel.

Dostaneme číslo c které kóduje konfiguraci S, to rozkóduji, provedu jeden krok Turingova stroje (lokální záležitost) a novou konfiguraci S_1 zakóduji jako c_1.

Je číslo a kódem slova, které je vlevo od výskytu nějakého $q \in Q$? Takové a najdi, b vpravo, $aqsb$ program Turingova stroje. ■

Věta 2.24. *Jeden krok Turingova stroje je primitivně rekurzivní záležitost. Obecněji pro procesor, jeden takt práce je primitivně rekurzivní záležitost.*

Důkaz. Ke každému Turingovu stroji T lze sestroji PRF $Step_T$, takovou že $Step_T(\text{code}(S)) = \text{code}(T(S))$. Jde o aritmetizaci práce Turingova stroje. ■

Důsledek 2.25. *Ke každému Turingovu stroji T lze sestrojit PRF $Comp_T$ takovou že*

$$Comp_T(\text{code}(S), t) = \text{code}(\textit{práce } T(S) \textit{ za } t \textit{ kroků}).$$

Důkaz. Provedeme t iterací $Step_T$. ■

Poznámka 2.26. Otázka je jak dlouho by měl Turingův stroj pracovat (jaké zvolit t). Řešení je "pracuj dokud neskončíš"! Tedy iterace kroků Turingova stroje dokud není $q \in F$ (nebo-li μ_t(za t kroků v koncovém stavu)).

Věta 2.27. *Ke každému Turingovu stroji T lze sestrojit PRF $Comp_T$ a ČRF g_T takovou že pro libovolný stav S stroje T*

$$g_T(\text{code}(S)) \sim \text{code}(T(S)).$$

Navíc je

$$g_T(x) \sim Result(Comp_T(x, \mu_t(\textit{za } t \textit{ kroků skončí } T \textit{ nad konfigurací s kódem } x)).$$

Poznámka 2.28. Někdy je vhodné použít modifikaci

$$\mu_t(\text{za } t_0 \text{ kroků skončí a } t_1 \text{ je kód mezivýsledků za } t_0 \text{ kroků})$$

kde t je kód dvojice $\langle t_0, t_1 \rangle$. Nyní je *Result* vydělení druhé složky,

$$g(x) \sim Result(\mu_t(\text{za } t_0 \text{ kroků skončí a } t_1 \text{ je kód mezivýsledků za } t_0 \text{ kroků})),$$

tedy *Result* je jednoduchá PRF (primitivně rekurzivní záležitost, iterace po krocích).

3 Ackermannova funkce

Pozorování 3.1. *Iterací operace součtu získáváme operaci násobení, iterací operace násobení získáváme operaci mocnění, atd. iterace předchozí aritmetické operace $f(i, x, y)$ kde i je počet iterací a x, y jsou operandy.*

Definice (Ackermannova funkce)**.** *Ackermannova funkce* je definována jako

$$\begin{aligned} A(0, x) &= \begin{cases} 1 & x = 0 \\ 2 & x = 1 \\ x + 2 & x > 1 \end{cases}, \\ A(y + 1, 0) &= 1, \\ A(y + 1, x + 1) &= A(y, A(y + 1, x)). \end{aligned}$$

Tedy jde o induktivní schéma, které určuje jedinou jednoznačně definovanou totální funkci.

Pozorování 3.2. *Funkce A je ORF.*

Důkaz. Snadno vidět. ■

Poznámka 3.3. A je v podstatě matice funkcí $A(y, x) = f_y(x)$, tedy

$$\begin{aligned} f_0(x) &= x + 2 && x \geq 2 \\ f_1(x) &= 2 \cdot x && x \geq 1 \\ f_2(x) &= 2^x && \text{všechna } x \\ f_3(x) &= 2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}} && x \geq 1, \end{aligned}$$

obecně potom $f_i(x) = A(i, x)$.

Poznámka 3.4. Naším cílem je dokázat že A není PRF. Tedy na vyčíslení funkce A nestačí konečný počet **for** cyklů (ale je potřeba **while** cyklus).

Definice. Mějme primitivně rekurzivní termy, pak hloubka termu *depth* je definována jako

(1) $depth(s) = depth(o) = depth(I_n^j) = 0$

(2) $depth(S_n^m(P, Q_j)) = \max(depth(P), depth(Q_j))$

(3) $depth(R_n(P, Q)) = \max(depth(P), depth(Q) + 1)$

Poznámka 3.5. Hloubka primitivně rekurzivního programu je ekvivalentní počtu do sebe vnořených **for** cyklů.

Definice. R_i buď třída funkcí, pro které existuje primitivně rekurzivní program hloubky $\leq i$ a platí, že třída PRF odpovídá $\bigcup_i R_i$.

Pozorování 3.6. *Buď $(\forall i)_{i\geq 1} f_i \in R_i - R_{i-1}$, potom f_{i+1} majorizuje všechny funkce z R_i, tedy*

$$(\forall g(x)) \in R_i : (\exists x_0)((\forall x) x \geq x_0 \ f_{i+1}(x) > g(x)).$$

Lemma 3.7. *$f_i(x) = A(i, x)$ roste v obou proměnných vyjma konečně mnoha hodnot.*

Tvrzení 3.8. *Funkce A není PRF.*

Důkaz. Nechť je, potom $A(x, x)$ má primitivně rekurzivní program, tedy platí $A(x, x) \in R_{i_0}$ pro nějaké i_0 a $A(x, x) < f_{i_0+1}(x) < f_x(x) = A(x, x)$ což je spor. ■

Pozorování 3.9. *Funkce A^{-1} roste nesmírně pomalu.*

4 Univerzální a částečně rekurzivní funkce

Příklad. Mějme spočetnou třídu $\mathcal{F}$ (1 proměnné). Buď $\mathcal{U}$ univerzální funkce pro třídu $\mathcal{F}$. Platí

(1) $(\forall i)(\lambda_x \mathcal{U}(i, x) \in \mathcal{F})$

(2) $(\forall g) g \in \mathcal{F}(\exists i)(g = \lambda_x \mathcal{U}(i, x))$.

$\mathcal{U}$ poskytuje numeraci (indexaci) funkcí z $\mathcal{F}$, $\mathcal{U}(i, x)$ je i-tá funkce. Pro ČRF "hezké". Otázka je jak "složitá" a "ošklivá" je $\mathcal{U}$?

Věta 4.1. *Třída PRF (jedné proměnné) nemá univerzální funkci která je PRF.*

Důkaz. Pomocí Cantorovy diagonální metody. Pro spor předpokládejme že ano. Buď touto $\mathcal{U}(i, x)$ která je PRF. Vezměme $1 \dot{-} \mathcal{U}(x, x) \simeq \mathcal{U}(i_0, x)$. Pro $x = i_0$ platí $1 \dot{-} \mathcal{U}(i_0, i_0) \simeq \mathcal{U}(i_0, i_0)$, obě strany jsou tedy definovány což znamená spor. ■

Pozorování 4.2. *Existuje univerzální funkce pro PRF, která je ORF.*

Důkaz (Idea). Efektivně očísluj primitivně rekurzivní termy (primitivně rekurzivní programy) P_x. Potom $\mathcal{U}(i, x) = Comp(P_i(x))$. ■

Definice. Buď $\mathcal{U}$ ČRF a platí $\mathcal{U}(e, x) \simeq \{e\}(x)$. Vždy lze očíslovat programy P_e a platí $\mathcal{U}(e, x) \simeq Comp(P_e(x))$. $\mathcal{U}(p, \mathcal{U}(q, x)) \simeq \mathcal{U}(c(p, q), x)$, efektivně $c(p, q)$ je programová kompozice. Jde o gödelovskou numeraci. $c(p, q)$ bývá často PRF, ale stačí ORF.

Ekvivalentní je gödelovská numerace. Buď V ČRF 2 proměnných, $V(m, x) \simeq \mathcal{U}(\beta(m), x)$ kde β je PRF. Fixací m vzniká $\lambda_x V(m, x)$ která musí mít index, zde efektivně $\beta(m)$.

Vždy platí, že pokud přirozeně očíslujeme programy P_e, pak $\mathcal{U}(e, x) \simeq P_e(x)$ je gödelovská numerace. Platí, že e-tá funkce je vyčíslována e-tým programem. V gödelovské numeraci má jedna funkce nekonečně mnoho programů.

Věta 4.3 (Kleeneho věta o normální formě)**.** *Existuje PRF U a pro každé $k \geq 1$ existují*

- *PRF s_k ($k+1$ proměnných)*
- *ČRF ψ_k ($k+1$ proměnných)*
- *PRP T_k ($k+2$ proměnných)*

takové, že

(1) $\psi_k(e, x_1, \ldots, x_k)$ *je univerzální pro třídu ČRF k proměnných, tedy je gödelovskou numerací*

(2) $\psi_k(e, x_1, \ldots, x_k) \simeq U(\mu_z T_k(e, x_1, \ldots, x_k, z))$ *kde* $\mu_{\langle z_0, z_1 \rangle}$ *za z_0 kroků skončí s mezivýsledkem z_1, U vydělí 2. složku*

(3) s_k *roste ve všech proměnných*

(4) $\psi_{m+n}(e, y_1, \ldots, y_m, x_1, \ldots, x_n) \simeq \psi_n(s_m(e, y_1, \ldots, y_m), x_1, \ldots, x_n)$ *(tzv. $s-m-n$ věta)*

(5) $T_{m+n}(e, y_1, \ldots, y_m, x_1, \ldots, x_n, z) \iff T_n(s_m(e, y_1, \ldots, y_m), x_1, \ldots, x_n, z)$.

Důkaz. Syntaktická manipulace s $e, y_1, \ldots, y_m$. Funkce s_m může byt definována jako "čekej na $x_1, \ldots, x_n$, vezmi e a pusť jej na $y_1, \ldots, y_m, x_1, \ldots, x_n$". Tedy s_m závisí pouze na m (historicky $s_{m,n}$, ale díky šikovnému kódování nezávisí na n). ■

Poznámka 4.4. Jak dokázat větu 4.3 pomocí univerzálního Turingova stroje? Máme univerzální TS a díky ekvivalenci TS a ČRF převedeme na univerzální ČRF.

Mějme univerzální TS, vlastní program bez dat

$$Y \ \text{code}(y_1, \ldots, y_m) \ \Lambda \ M \ Y \ldots Y$$

kde M znamená čekání na data x, tedy

$$Y \ \text{code}(y_1, \ldots, y_m) \ \Delta \ \text{code}(x_1, \ldots, x_n) \ Y \ldots Y.$$

Poznámka 4.5. $\psi_k(e, x_1, \ldots, x_k) \simeq U(\mu_y T_k(e, x_1, \ldots, x_k, y))$, gödelovská numerace

$$\psi_{m+n}(e, y_1, \ldots, y_m, x_1, \ldots, x_n) \simeq \psi_n(s_m(e, y_1, \ldots, y_m), x_1, \ldots, x_n),$$

s_m je syntaktická manipulace a $e, y_1, \ldots, y_m$ nám dává nový program $s_m(e, y_1, \ldots, y_m)$.

Věta 4.6.

(1) Predikát $\psi_k(e, x_1, \ldots, x_k)\downarrow$ *je rekurzivně spočetný, ale není rekurzivní,*

(2) negace, tzn. $\psi_k(e, x_1, \ldots, x_k)\uparrow$ *není rekurzivně spočetná,*

(3) neexistuje ORF, která by byla rozšířením ψ_k.

Důkaz. Buď $\psi_1(e, x)\downarrow$ rekurzivně spočetný. Kdyby byl rekurzivní, pak jeho negace by byla také rekurzivní a tím spíše rekurzivně spočetná. Stačí tedy dokázat že negace, tzn. $\psi_1(e, x)\uparrow$ není rekurzivně spočetná což lze snadno Cantorovou diagonální metodou.

Stačí dokázat, že $\psi_1(x, x)\uparrow$ není rekurzivně spočetná. Nechť ano, pak existuje x_0 takové, že $\psi_1(x, x)\uparrow \iff \psi_1(x_0, x)\downarrow$, dosadíme $x = x_0$ a dostáváme spor.

$\psi_1(e, x)\downarrow$ v proměnné e řadí do posloupnosti všechny rekurzivně spočetné predikáty. Existuje e_0 kde $\psi_1(e_0, x)\uparrow$, tedy ψ_1 triviálně není totální.

Nechť $g(e, x)$ je rozšířením $\psi_1(e, x)$ a je ORF. Buď $f(x) \simeq 1 \dot{-} g(x, x)$, f je ORF a má index, pak $f(x) \simeq \psi_1(e_0, x)$, a tedy $f(e_0) \simeq \psi_1(e_0, e_0)$ kde obě strany konvergují a $\psi_1(e_0, e_0) \simeq g(e_0, e_0)$. Z toho $1 \dot{-} g(e_0, e_0) \simeq g(e_0, e_0)$ přičemž obě strany konvergují což je spor. ■

Důsledek 4.7. Halting problém $\psi_1(e, x)\downarrow$ *(e-tý program zastaví na vstupu x) není algoritmicky rozhodnutelný (není rekurzivní).*

Tvrzení 4.8. *Buď ČRF β rozšířením ψ_1, pak lze efektivně (z programu pro β, tedy indexu β) nalézt e_1 takové, že $\beta(e_1, e_1)\uparrow$ (naše efektivní vyhrávající strategie).*

Důkaz. Nechť β je ČRF. $\alpha(x) \simeq 1 \dot{-} \beta(x, x)$ a tedy α je ČRF a má index (program) e_1. $\alpha(x) \simeq \psi_1(e_1, x)$ a tedy $\alpha(e_1) \simeq \psi_1(e_1, e_1)$. Kdyby $\alpha(e_1)\downarrow$, pak i $\psi_1(e_1, e_1)\downarrow$. Z toho vyplývá že $\beta(e_1, e_1)\downarrow$ a $\beta(e_1, e_1) \simeq \psi_1(e_1, e_1)$. Platí tedy $1 \dot{-} \beta(e_1, e_1) \simeq \beta(e_1, e_1)$ což je spor. Tedy e_1 lze nalézt efektivně z indexu β. ■

Poznámka 4.9. $\psi_1(e, x)$ je univerzální ČRF, ona sama je ČRF druhé proměnné. Buď $\mathcal{U}$ univerzální program, potom $\mathcal{U}(e, x)$ vyčísluje e-tý program na x.

Poznámka 4.10. Platí $\psi_1(e, x) \simeq \varphi_e(x)$, někdy také $\psi_1(x, e) \simeq \{e\}(x)$.

Definice. Rekurzivně spočetné množiny jsou definiční obory ČRF. Tedy e-tá rekurzivně spočetná množina je definiční obor e-té ČRF (tzn. e-tého programu). W_e je e-tá rekurzivně spočetná množina, tedy

$$W_e = \{x | \psi_1(e, x)\downarrow\} = \{x | \varphi_e(x)\downarrow\}.$$

Definice. Množina K buď definovaná jako

$$K = \{x | x \in W_x\} = \{x | \psi_1(x, x)\downarrow\} = \{x | \varphi_x(x)\downarrow\}.$$

Věta 4.11.

(1) K je rekuzivně spočetná, není rekurzivní,

(2) $\overline{K}$ není rekurzivně spočetná.

Důkaz. K je rekurzivně spočetná z definice. Kdyby K byla rekurzivní, pak i $\overline{K}$ by byla rekurzivní a tedy rekurzivně spočetná. Stačí tedy dokázat, že $\overline{K} = \{x | x \notin W_x\}$ není rekurzivně spočetná.

Kdyby $\overline{K}$ byla rekurzivně spočetná, pak $(\exists x_0)$ takové, že $\overline{K} = W_{x_0}$. Platí $x \in \overline{K} \iff x \in W_{x_0}$ a tedy $x_0 \in \overline{K} \iff x_0 \in W_{x_0}$, odtud $x_0 \in K$ což je spor. ■

4.1 1-převeditelnost, m-převeditelnost

Definice. $A \leq_1 B$ (1-převeditelnost) pokud existuje prostá ORF f, taková, že $x \in A \iff f(x) \in B$. $A \leq_m B$ pokud existuje ORF, taková že $x \in A \iff f(x) \in B$.

Pozorování 4.12. *Platí $f(A) \subseteq B$ a $f(\overline{A}) \subseteq \overline{B}$.*

Definice. B je 1-úplná pokud B je rekurzivně spočetná a existuje libovolná rekurzivně spočetná A taková, že $A \leq_1 B$.

Věta 4.13. *K je 1-úplná.*

Důkaz. Víme, že K je rekurzivně spočetná. Buď $y \in W_x$, sestrojím pomocnou ČRF $\alpha(x, y, w)\downarrow \iff y \in W_x$ $(\alpha(x, y, w) \simeq \psi_1(x, y))$. α má index a, tedy $\alpha(x, y, w) \simeq \psi_3(a, x, y, w) \simeq \psi_1(s_2(a, x, y), w) \simeq \varphi_{s_2(a,x,y)}(w)$. Nechť $w = s_2(a, x, y)$, pak $y \in W_x \iff \alpha(x, y, w)\downarrow \iff \varphi_{s_2(a,x,y)}(s_2(a, x, y))\downarrow \iff s_2(a, x, y) \in K$.

Pro pevné x funkce $\lambda_y s_2(a, x, y)$ 1-převádí W_x na K. ■

Tvrzení 4.14. *Buď $K_0 = \{\langle x, y\rangle | y \in W_x\}$, je vidět, že K_0 je 1-úplná, $K \leq_1 K_0$.*

Definice. $A \equiv B$ pokud existuje rekurzivní permutace f taková, že $f(A) = B$.

Lemma 4.15. *$\equiv$ je ekvivalence (reflexivní, tranzitivní a symetrická).*

Věta 4.16 (Myhill)**.** *Pokud $A \leq_1 B$ a $B \leq_1 A$, pak $A \equiv B$.*

Důkaz (Idea). Mějme dvojici funkcí $f : A \to B$ a $g : B \to A$ pomocí kterých párujeme prvky z množin A a B. Mějme k dvojic, zobrazuji $(k+1)$-ní prvek, na druhé straně musím najít volný protějšek. ■

Věta 4.17. *Buď $A \leq_1 B$, pak*

(1) pokud B je rekurzivní, pak A je rekurzivní

(2) pokud B je rekurzivně spočetná, pak A je rekurzivně spočetná.

Analogicky pro $\leq_m$.

Důkaz. Platí $x \in A \iff f(x) \in B$. Jestliže $B = dom(\beta)$, pak $A = dom(\beta \circ f)$. ■

Věta 4.18. *Relace $\leq_1$ a $\leq_m$ jsou reflexivní a tranzitivní.*

Definice. $A \equiv_1 B$ pokud $A \leq_1 B$ a $B \leq_1 A$.

5 Rekurzivně spočetné množiny

Lemma 5.1. *Jestliže $Q(x_1, \ldots, x_n, y)$ je ORF, pak $(\exists y)Q(x_1, \ldots, x_n, y)$ je RSP.*

Důkaz. Intuitivně jasné, $\mu_y Q$ je ČRF jejíž definiční obor odpovídá $(\exists y)Q$, ta je ekvivalentní $\mu_y(1 \dot{-} C_Q(x_1, \ldots, x_n, y) \simeq 0)$ což je ČRF. ■

Tvrzení 5.2. *$(\exists y)T_k(e, x_1, \ldots, x_k, y)$ je univerzálním RSP pro třídu všech RSP k proměnných.*

Důkaz. $(\forall e)(\exists y)T_k(e, x_1, \ldots, x_k, y)$ je rekurzivně spočetný predikát k proměnných. $P(x_1, \ldots, x_n)$ je RSP k proměnných, tedy

$$(\exists e)P(x_1, \ldots, x_k) \iff (\exists y)T_k(e, x_1, \ldots, x_k, y).$$

Predikát

$$\psi_k(e, x_1, \ldots, x_k) \simeq U(\mu_y T_k(e, x_1, \ldots, x_k, y))$$

je definován definičním oborem. $(\exists y)T_k(e, x_1, \ldots, x_k, y)$ je univerzální pro všechny RSP. ■

Pozorování 5.3. *Tato numerace RSP je gödelovská (hlavní, atd.), tzn. platí pro ni věta 4.3.*

Tvrzení 5.4 (Základní vlastnosti)**.**

(1) Průnik a sjednocení dvou rekurzivně spočetných množin je rekurzivně spočetná, nebo-li $\wedge, \vee$ zachovává rekurzivní spočetnost predikátů. Dokonce existují PRF α, β takové, že $W_{\alpha(x,y)} = W_x \cap W_y$ a $W_{\beta(x,y)} = W_x \cup W_y$, α, β je aritmetizace těchto operací.

(2) Nechť Q je RSP, potom $(\forall y)_{y \leq t} Q$ je RSP, tedy omezená obecná kvantifikace zachovává rekurzivní spočetnost.

(3) Nechť Q je RSP, potom $(\exists y)Q$ je RSP, tedy existenční kvantifikace zachovává rekurzivní spočetnost.

Důkaz.

(1) Intuitivně jasné, mějme 2 programy Pr_1, Pr_2, pro průnik musí oba konvergovat, pro sjednocení aspoň jeden. Pro formální důkaz je potřeba věta 4.3

Tedy $(\exists y(T_1(a, x, y) \vee (\exists y)T_1(b, x, y)$ je univerzální vyjádření disjunkce a-té a b-té rekurzivně spočetné, což je ekvivalentní $(\exists y)(T_1(a, x, y) \vee T_1(b, x, y))$ což je rekurzivně spočetné a ekvivalentní (pro nějaké e) $(\exists y)T_3(e, a, b, x, y) \iff (\exists y)T_1(s_2(e, a, b), x, y)$, tedy $\beta(a, b) = s_2(e, a, b)$. Platí tedy

$$(\exists y)T_1(a, x, y) \wedge (\exists y)T_1(b, x, y) \iff (\exists z)(T_1(a, x, (z)_{2,1}) \wedge T_1(b, x, (z)_{2,2})$$

kde $z = \langle y_0, y_1 \rangle$ a $(z)_{2,1}, (z)_{2,2}$ znamená vydělení první, resp. druhé složky. To je ekvivalentní $(\exists z)(T_3(e_1, a, b, x, z)) \iff (\exists z)T_1(s_2(e_1, a, b), x, z)$, tedy $\alpha(a, b) = s_2(e_1, a, b)$.

(2) Intuitivně jasné, jde o konečnou konjunkci $0, \ldots, t$. Formálně

$$(\forall y)_{y \leq t}(\exists z) T_k(a, x_1, \ldots, x_{k-1}, y, z) \iff (\exists w)(\forall y)_{y \leq t} T_k(a, x_1, \ldots, x_{k-1}, y, (w)_{t+1,y})$$

kde $w = \langle z_0, \ldots, z_t \rangle$. Jde tedy o RSP a platí (pro nějaké e_2)

$$(\exists z) T_{k+1}(e_2, a, x_1, \ldots, x_{k-1}, t, z) \iff (\exists z) T_k(s_1(e_2, a), x_1, \ldots, x_{k-1}, t, z).$$

(3) Formálně

$$(\exists y)(\exists z) T_k(a, x_1, \ldots, x_k, y, z) \iff (\exists w) T_k(a, x_1, \ldots, x_k, (w)_{2,1}, (w)_{2,2})$$

kde $w = \langle y, z \rangle$ a stejně jako v předchozím případě platí (pro nějaké e_3)

$$(\exists z) T_{k+1}(e_3, a, x_1, \ldots, x_k, t, z) \iff (\exists z) T_k(s_1(e_3, a), x_1, \ldots, x_k, t, z).$$

■

Tvrzení 5.5.

(1) Negace nezachovává (obecně) rekurzivní spočetnost.

(2) Obecná kvantifikace nezachovává rekurzivní spočetnost.

Důkaz.

(1) Buď K rekurzivně spočetná, ale $\overline{K}$ není.

(2) $(\forall y)\neg T_1(x, x, y)$ není rekurzivně spočetný ($(\forall y)\neg T_1(x, x, y) \iff \neg(\exists y) T_1(x, x, y)$).

■

Lemma 5.6 (O selektoru)**.** *Buď $Q(x_1, \ldots, x_k, y)$ RSP, potom existuje ČRF φ a platí*

$$\begin{aligned} \varphi(x_1, \ldots, x_k)\downarrow &\iff (\exists y) Q(x_1, \ldots, x_k, y) \\ \varphi(x_1, \ldots, x_k)\downarrow &\implies Q(x_1, \ldots, x_k, \varphi(x_1, \ldots, x_k)) \end{aligned}$$

Důkaz. Bez újmy na obecnosti zvolme $k = 1$. Hledáme nejmenší dvojici $\langle y, s \rangle$ takovou, že za s-kroků na x, y konverguje. Formálně vyjádřeno, nechť Q je rekurzivně spočetný, lze jej tedy vyjádřit ve tvaru $(\exists s) T_2(a, x, y, s)$, budeme hledat $\varphi(x) \simeq (\mu_w\, T_2(a, x, (w)_{2,1}, (w)_{2,2}))_{2,1}$ kde $w = \langle y, s \rangle$. ■

Definice (Graf funkce)**.** Graf funkce $f(x) \simeq y$ je $\{\langle x, y \rangle | f(x) \simeq y\}$.

Tvrzení 5.7. *Funkce je ČRF právě tehdy, když má rekurzivně spočetný graf.*

Důkaz.

$\Longleftarrow$ Intiutivně svobodná volba jediného možného.

$\Longrightarrow$ $\{\langle x, y \rangle | (\exists s)$(za s kroků končí výpočet s výsledkem y)$\}$ $(\{\langle x, y \rangle | (\exists s) T_1(a, x, s) \& U(s) = y\})$.

■

Věta 5.8 (Postova věta)**.** *M je rekurzivní právě tehdy když $M, \overline{M}$ jsou rekurzivně spočetné.*

Důkaz.

$\Longleftarrow$ Triviálně platí.

$\Longrightarrow$ Intuitivně snadné, mějme dva programy Pr_1, Pr_2, spustíme oba paralelně a čekáme který se zastaví. Formálně $(x \in M \ \&\ y = 1) \vee (x \in \overline{M} \ \&\ y = 0)$ a použijeme selektor C_M.

■

5.1 Generování rekurzivně spočetných množin

Definice. Částečná funkce φ se nazývá *úseková* pokud $dom(\varphi)$ je počátečním úsekem $\mathbb{N}$ (lze také celé $\mathbb{N}$, tzn. že může být všude definovaná).

Věta 5.9. *Každá rekurzivně spočetná množina je oborem hodnot nějaké ČRF.*

Důkaz. Zvolíme predikát $P(x,y) \equiv x \in W_a \ \&\ y = x$, použijeme selektor a dostáváme ČRF φ, tedy $dom(\varphi) = range(\varphi) = W_a$. ■

Věta 5.10. *Každý obor hodnot ČRF je rekurzivně spočetná množina.*

Důkaz. Mějme ČRF α, chceme najít ČRF β takovou, že $dom(\beta) = range(\alpha)$ a $\beta(y)\downarrow$ právě tehdy když se na y něco zobrazí. Nechť je tedy dáno y, hledáme x tak aby $\alpha(x) = y$ (první které najdeme). Potom $\alpha(x) \simeq U(\mu_z T_1(e,x,z))$, $\beta(y) = \mu_w(\alpha(x)\downarrow$ za s kroků$\&\alpha(x) = y)$ kde $w = \langle x,s\rangle$ a $(\mu_z T_1(a,(z)_{2,1},(z)_{2,2})\ \&\ U(z) = y)_{2,1}$. ■

Lemma 5.11. *Rekurzivní množiny jsou právě obory hodnot rostoucích úsekových ČRF.*

Důkaz.

$\Longrightarrow$ Buď M je rekurzivní, potom $\alpha(0) \simeq \mu_y(y \in M)$, $\alpha(n+1) \simeq \mu_y(y > \alpha(n) \& y \in M)$.

$\Longleftarrow$ Buď α rostoucí úseková ČRF. Pokud je $dom(\alpha) = \{0,\ldots,k\}$ konečný, pak $M = \{\alpha(0),\ldots,\alpha(k)\}$ je rekurzivní. Pokud je naopak $dom(\alpha)$ nekonečný, pak M je rekurzivní neboť $x \in M \iff (\exists y)_{y \le x}(\alpha(y) = x)$ pro $x \in \{\alpha(0),\ldots,\alpha(x)\}$. Toto rozštěpení je ale neefektivní.

■

Věta 5.12. *Nechť množina M je nekonečná, je rekurzivní právě tehdy, když je oborem hodnot rostoucí ORF.*

Důkaz. Implikuje lemma 5.11. ■

Lemma 5.13. *Rekurzivně spočetné množiny jsou právě obory hodnot prostých úsekových ČRF.*

Důkaz.

$\Longleftarrow$ Implikuje věta 5.10.

$\Longrightarrow$ Rekurzivně spočetná množina M, $M = dom(\alpha)$ a α je ČRF. Buď

$$B = \{\langle x,s\rangle | \alpha(x)\downarrow \text{ za přesně } s \text{ kroků}\} = \{\langle x,s\rangle | T_1(a,x,s)\ \&\ (\forall j)_{j<s} \neg T_1(a,x,j)\},$$

pak B je rekurzivní. Generujeme B rostoucí úsekovou ČRF β ($range(\beta) = B$). Zvolme $f = (\beta)_{2,1}$ která vydělí první složku (tzn. x). Pak f je naše hledaná funkce protože je prostá, úseková a ČRF.

■

Věta 5.14. *Nechť množina M je nekonečná, je rekurzivně spočetná právě tehdy, když je oborem hodnot prosté ORF.*

Důkaz. Implikuje lemma 5.13. ■

Důsledek 5.15. *Každá nekonečná rekurzivně spočetná množina obsahuje nekonečnou rekurzivní podmnožinu.*

Důkaz. Vybereme rostoucí podposloupnost. Mějme ORF f která je prostá a generuje M. Definujme $g(0) \simeq f(0)$, $g(n+1) \simeq f(\mu_j(f(j) > g(n))$, potom $range(g)$ je hledaná podmnožina. ■

Definice. Množina A je imunní pokud je A nekonečná a neobsahuje žádnou nekonečnou rekursivně spočetnou podmnožinu.

Definice. Množina B je prostá pokud je B rekruzivně spočetná a $\overline{B}$ je imunní.

Tvrzení 5.16. *Existuje imunní množina.*

Poznámka 5.17. Rekurzivní a rekurzivně spočetné množiny je možné použít i v jiných oblastech než jen v $\mathbb{N}$, např. slova (jazyky), logika (množiny formulí), atd. Máme dva způsoby jak toho dosáhnout:

(1) obecnější programy (Turingovy stroje, atd.), pak rekurzivita je ekvivalentní algoritmické rozhodnutelnosti a rekurzivní spočetnost ekvivalentní definiční oborům/oborům hodnot (nějakému programu)

(2) efektivně očíslujeme konečné objekty (slova, formule), pak množina má nějakou vlastnost právě tehdy když množina kódů má nějakou vlastnost a ČRF jsou ekvivalentní Turingovým strojům, atd.

Poznámka 5.18.

(1) Jazyky typu 0 jsou právě rekurzivně spočetné.

(2) Jazyk 1. řádu, logika 1. rádu. Jestliže množina axiomů je rekurzivní, pak množina dokazatelných formulí je rekurzivně spočetná.

Poznámka 5.19 (10. Hilbertův problém)**.** Problém algoritmického rozhodnování zda pro polynom s celočíselnými koeficienty existuje řešení v celých číslech, což je ekvivalentní algoritmickému rozhodování o existenci řešení pro rovnost dvou polynomů s přirozenými koeficienty v přirozených číslech.

Věta 5.20 (Robinson, Davis, Putuam, Matijesevič)**.** *P je rekurzivně spočetný predikát právě tehdy, když existují polynomy p_1, p_2 s přirozenými koeficienty, nebo-li*

$$P(x_1, \dots, x_k) \iff (\exists y_1, \dots y_n)(p_1(x_1, \dots, x_k, y_1, \dots, y_n) = p_2(x_1, \dots, x_k, y_1, \dots, y_n))$$

(tzv. diofantické predikáty).

Důkaz. Negativní řešení 5.19. ■

Důsledek 5.21. *Problém rovnosti dvou polynomů ($p_1 = p_2$) je rekurzivní, problém existence ($\exists p_1 = p_2$) je rekurzivně spočetný. Na Σ_1 úrovni, tzn. po existenční kvantifikaci stejné, ale jádra (vnitřek) se liší. Tedy rekurzivita není ekvivalentní ($p_1 = p_2$) ale rekurzivní spočetnost je ekvivalentní ($\exists p_1 = p_2$).*

Výhodou je vyjádřitelnost v jazyce elementární aritmetiky.

Důsledek 5.22. *Platí*

$$P \iff (\exists y_1, \dots, y_n)(p_1 = p_2) \iff \models_{\mathbb{N}} (\exists \mathcal{F})$$

kde atomická formule $\mathcal{F}$ reprezentuje $p_1 = p_2$.

6 Věty o rekurzi

Věta 6.1. *Pro každou ČRF f existuje a takové, že*

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

tzn. pokud $f(a)\downarrow$, pak $a, f(a)$ jsou ekvivalentní programy a pokud $f(a)\uparrow$, pak a se nikdy nezastaví.

Důkaz. Vezměme $\varphi_{f(s_1(z,z))}(x) \simeq \psi_2(e,z,x) \simeq \varphi_{s_1(e,z)}(x)$, dosadíme $z = e$ a položíme $a = s_1(e,e)$. Dostáváme tedy $\varphi_{f(s_1(e,e))}(x) \simeq \varphi_{s_1(e,e)}(x)$. ■

Důkaz. $\varphi_{\varphi_u(u)}(x) \simeq \psi_2(b,v,x) \simeq \varphi_{d(u)}(x) \simeq \varphi_{\varphi_e(u)}(x)$ kde $d(u) = s_1(b,u)$ a $\varphi_e = d$. $\varphi_u(z)$ je ekvivalentní $\alpha_{u,z}$ což je matice funkcí, f permutuje řádky v této matici, nebo-li $\varphi_{f\circ\varphi_u(z)}(x) \simeq \varphi_{\beta(u,z)}(x) \simeq \varphi_{\varphi_{H(u)}(z)}(x)$. Pak $\varphi_{f\circ\varphi_e(H(e))}(x) \simeq \varphi_{\varphi_{H(e)}(H(e))}(x) \simeq \varphi_{\varphi_e(H(e))}(x)$. ■

Poznámka 6.2. Program a počítá déle.

Program e na vstupu dostane z, x a spočte $s_1(z,z)$ na vstup x, a pokud f konverguje, spustí program $f(s_1(z,z))$ na vstup x.

Program $a = s_1(e,e)$ na vstupu dostane x, program $s_1(e,e)$ vezme program e, přidá e k x a spustí, tzn. že e spustí na e, x. Výsledkem je $s_1(e,e) = a$, tedy program spočte vlastní kód, a pokud f konverguje, spustí program $f(s_1(e,e)) = f(a)$ na vstup x.

Věta 6.3. *Pro každou ČRF h $m+1$ proměnných existuje PRF g m proměnných taková, že*

$$\varphi_{h(g(y_1,\ldots,y_m),y_1,\ldots,y_m)}(x) \simeq \varphi_{g(y_1,\ldots,y_m)}(x).$$

Důkaz. Vezměme

$$\varphi_{h(s_{m+1}(z,z,y_1,\ldots,y_m),y_1,\ldots,y_m)}(x) \simeq \psi_{m+2}(e^*,z,y_1,\ldots,y_m,x) \simeq \varphi_{s_{m+1}(e^*,z,y_1,\ldots,y_m)}(x)$$

, a položme $g(y_1,\ldots,y_m) = s_{m+1}(e^*,e^*,y_1,\ldots,y_m)$. ■

Věta 6.4. *Pro každou ČRF f existuje prostá PRF α taková, že*

$$\varphi_{f(\alpha(j))}(x) \simeq \varphi_{\alpha(j)}(x).$$

Důkaz. Vezměme $\varphi_{f(s_2(z,z,j))}(x) \simeq \psi_2(e^\#,z,j,x) \simeq \varphi_{s_2(e^\#,z,j)}(x)$ a položme $\alpha(j) = s_2(e^\#,e^\#,j)$. ■

Věta 6.5. *Pro každou ČRF h $m+1$ proměnných existuje a která je indexem (gödelovským číslem) funkce $\lambda_{x_1,\ldots,x_m}h(a,x_1,\ldots,x_m)$, tedy $h(a,x_1,\ldots,x_m) \simeq \varphi_a(x_1,\ldots,x_m)$.*

Důkaz. $h(y,x_1,\ldots,x_m) \simeq \psi_{m+1}(b,y,x_1,\ldots,x_m) \simeq \varphi_{s_1(b,y)}(x_1,\ldots,x_m)$, následně aplikujeme větu 6.1 na $s_1(b,y)$ v proměnné y a dostáváme hledané a. ■

Příklad. Pomocná ČRF $\alpha(n,x) \simeq n$. Platí $\alpha(n,x) \simeq \psi_2(e,n,x) \simeq \varphi_{s_1(e,n)}(x) \simeq \varphi_{f(n)}(x)$ a podle věty 6.1 existuje n_0 takové, že $\varphi_{n_0} = \varphi_{f(n_0)}$, tzn. $\varphi_{n_0} \simeq n_0$.

Věta 6.6 (Rice)**.** *Nechť $\mathcal{A}$ je třída (skupina) ČRF jedné proměnné která je netriviální, potom $A_\mathcal{A} = \{x|\varphi_x \in \mathcal{A}\}$ není rekurzivní. Taková množina se nazývá* indexová množina*. Speciální případem je třída jediné ČRF φ_{x_0} kde $A = \{x|\varphi_x = \varphi_{x_0}\}$.*

Důkaz. Pokud $\mathcal{A}$ je triviální, pak $A_\mathcal{A}$ je rekurzivní. Jinak musí existovat aspoň jedna $x_0 \in A$ a $y_0 \notin A$. Kdyby byla A rekurzivní, pak by existovala funkce h taková, že $h(x) = x_0$ pokud $x \notin A$ nebo $h(x) = y_0$ pokud $x \in A$. Pak h je ORF, tzn. že existuje a takové, že $\varphi_a = \varphi_{h(a)}$, a tedy $a, h(a)$ jsou ekvivalentní programy což je spor. ■

Poznámka 6.7. Někdy může být množina $A_\mathcal{F}$ rekurzivně spočetná.

Pozorování 6.8. *Množina A je indexová pro ČRF, pokud pro $x \in A$ a y ekvivalentní x platí $y \in A$. Označení* indexová množina *tedy znamená, že je "uzavřená na ekvivalenci pojmenovaných objektů".*

Tvrzení 6.9. *Existuje PRF f taková, že*

$$(\forall y)_{y\geq 1}W_{f(y)} = \{f(0),\ldots,f(y-1)\}.$$

Důkaz. Chceme ORF f takovou, že $(\forall y)_{y\geq 1} W_{f(y)} = \{f(0), \ldots, f(y-1)\}$. Hledáme ORF f, tzn. že hledáme program, tedy číslo. Použijeme pomocnou ČRF $\alpha(v,y,x)\downarrow \iff (\exists j)_{j<y}(\varphi_v(j) \simeq x)$, tzn. že $x \in \{\varphi_v(0), \ldots, \varphi_v(y-1)\}$ pokud $\alpha(v,y,x)\downarrow$. Platí $\alpha(v,y,x) \simeq \psi_3(e,v,y,x) \simeq \varphi_{s_2(e,v,y)}(x)$ a $h(v,y) = s_2(e,v,y)$ kde e je číslo α. Víme že $x \in W_{h(v,y)} \iff x \in \{\varphi_v(0), \ldots, \varphi_v(y-1)\}$, a platí $h(v,y) \simeq \psi_2(b,v,y) \simeq \varphi_{g(v)}(y)$ a $g(v) = s_1(b,v)$. Podle věty 6.1 existuje hledané v_0 takové, že $\varphi_{v_0} = \varphi_{g(v_0)}$. ■

7 Produktivní a kreativní množiny

Definice. Množina A je *produktivní*, pokud existuje ČRF f (nazývaná *produktivní funkce*) taková, že

$$(\forall x) W_x \subseteq A \Longrightarrow f(x)\downarrow \ \& \ f(x) \in A \setminus W_x.$$

Poznámka 7.1. Pokud A není rekurzivně spočetná množina, pak (vždy) existuje nějaké $z \in A \setminus W_x$. Efektivně najít ho lze u produktivní množin.

Definice. Množina B je *kreativní*, pokud B je rekurzivně spočetná a $\overline{B}$ je produktivní.

Tvrzení 7.2. *Množina $\overline{K}$ je produktivní s identickou funkcí jakožto produktivní funkcí, K je kreativní.*

Důkaz. Buď $W_x \subseteq \overline{K}$, kdyby $x \in W_x$, pak buď $x \in K$ nebo $x \notin \overline{K}$ což je spor. Tedy $x \notin W_x$ odkud vyplývá $x \in \overline{K} \setminus W_x$. ■

Tvrzení 7.3. *Množina $B = \{f(x) | f(x) \in W_x\}$ kde f je prostá ORF, pak B je kreativní a $\overline{B}$ produktivní s funkcí f.*

Důkaz. Buď $W_x \subseteq \overline{B}$, kdyby $f(x) \in W_x$, pak by $f(x) \in B$ nebo $f(x) \notin \overline{B}$ což je spor. Tedy $f(x) \notin W_x$, a protože f je prostá, platí $f(x) \in \overline{B} \setminus W_x$. ■

Věta 7.4. *Každá produktivní množina má produktivní funkci která je ORF (tzn. totální).*

Důkaz. Buď A produktivní a α ČRF. Platí pokud $W_x \subseteq A$, pak $\alpha(x)\downarrow \ \& \ \alpha(x) \in A \setminus W_x$. Hledáme ORF h pro kterou by platilo, že pokud $W_x \subseteq A$, pak $h(x) \in A \setminus W_x$. Protože $\alpha(y)\downarrow$ nemusí být algoritmicky rozhodnutelná, vytvoříme ORF g pro kterou platí

$$W_{g(y)} = \begin{cases} W_y & \alpha(g(y))\downarrow \\ \emptyset & \alpha(g(y))\uparrow \end{cases},$$

potom $\alpha \circ g$ je hledaná ORF, produktivní pro A.

Funkce $\alpha \circ g$ je ORF ale také ČRF. Kdyby $\alpha(g(y))\uparrow$, pak by $W_{g(y)} = \emptyset$ odkud $\alpha(g(y))\downarrow$ což je spor. Vždy tedy $W_{g(y)} = W_y$. Triviálně pokud $W_y \subseteq A$, pak $\alpha(g(y)) \in A \setminus W_{g(y)} = A \setminus W_y$.

Tedy $h = \alpha \circ g$. Funkci g vytvoříme pomocí věty 6.1. Buď tedy ORF f pro kterou

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

a ČRF β pro kterou platí $\beta(x,y,w)\downarrow \iff \alpha(x)\downarrow \ \& \ w \in W_y$ kde $\beta(x,y,w) \simeq \varphi_{s_2(e,x,y)}(w)$ a $f(x,y) = s_2(e,x,y)$. Podle věty 6.1 platí $W_{g(y)} = W_{f(g(y),y)}$. ■

Věta 7.5. *Každá produktivní množina obsahuje nekonečnou rekurzivně spočetnou podmnožinu.*

Důkaz. Mějme ORF f produktivní pro množinu A.

Neformálně pokud $W_{z_0} = \emptyset \subseteq A$, pak $f(z_0) \in A \setminus W_{z_0} = A$; $W_{z_1} = \{f(z_0)\} \subseteq A$, pak $f(z_1) \in A \setminus W_{z_1}$, atd.

Formálně buď $W_{g(x)} = W_x \cup \{f(x)\}$. Nechť $h(0) = z_0$, $h(y+1) = g(h(y))$, pak $W_{h(y)}$ má právě y bodů $\{f(h(0)), f(h(1)), \ldots, f(h(y-1))\}$, tedy $f \circ h$ prostě generuje nekonečnou rekurzivně spočetnou množinu. ■

Věta 7.6. *Každá produktivní množina má produktivní funkci která je rekurzivní permutací.*

Důkaz. Víme že má ORF funkci. Stačí dokázat že je prostá a na. ■

Věta 7.7. *Následující tvrzení jsou ekvivalentní*

(1) množina M je kreativní,

(2) množina M je 1-úplná,

(3) množina M je m-úplná.

Lemma 7.8. *Pokud je množina B produktivní, $B \leq_m A$, pak A je produktivní.*

Důkaz. Mějme ORF f produktivní pro B. Buď g ORF taková, že $x \in B \iff g(x) \in A$. Zvolme ORF h takovou, že $W_{h(x)} = \{y | g(y) \in W_x\} = g^{-1}(W_x)$ použitím věty 4.3. Platí, že pokud $W_x \subseteq A$, pak $W_h(x) \subseteq B$. Odtud $f(h(x)) \in B \setminus W_{h(x)}$ a $g(f(h(x))) \in A \setminus W_x$. ■

Věta 7.9. *Následující tvrzení jsou ekvivalentní*

(1) množina A je produktivní,

(2) $\overline{K} \leq_1 A$,

(3) $\overline{K} \leq_m A$.

Důkaz.

(2) $\implies$ (3) Triviálně platí.

(3) $\implies$ (1) Implikuje lemma 7.8.

(1) $\implies$ (2) Existuje prostá ORF f pro A. Hledáme ORF g (prostou) tak, aby

$$W_{g(y)} = \begin{cases} \{f(g(y))\} & y \in K \\ \emptyset & y \notin K \end{cases}.$$

Funkce $f \circ g$ převádí (redukuje) $\overline{K}$ do A, tedy $\overline{K} \leq_1 A$ a $\overline{K} \leq_m A$. Pokud $y \notin K$, pak $W_{g(y)} = \emptyset \subseteq A$ a tedy $f(g(y)) \in A$. Pokud by naopak $y \in K$, pak $W_{g(y)} = \{f(g(y))\}$ a pokud $f(g(y)) \in A$, pak by $W_{g(y)} \subseteq A$ odkud $f(g(y)) \in A \setminus W_{g(y)}$ což je spor. Tedy nutně $f(g(y)) \in \overline{A}$. Buď

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

pro ORF f dle věty 4.3 a dle věty 6.1 máme ORF g takovou, že $W_{g(y)} = W_{f(g(y),y)}$.

■

Pozorování 7.10. *Věta 7.9 implikuje větu 7.7.*

Důkaz. Množina M je kreativní pokud M je rekuzivně spočetná a $\overline{M}$ produktivní. Množina M je 1-úplná pokud M je rekurzivně spočetná a $K \leq_1 M \iff \overline{K} \leq_1 \overline{M}$. ■

Tvrzení 7.11. *Pro $\leq_1, \leq_m$ je $\overline{K}$ nejjednodušší produktivní množina.*

Definice. A je *úplně produktivní*, pokud pro ORF f platí

$$(f(x) \in A \setminus W_x) \vee (f(x) \in W_x \setminus A).$$

Pozorování 7.12. *Pokud A je úplně produktivní, pak A je produktivní.*

Tvrzení 7.13. *$\overline{K}$ je úplně produktivní s identitou.*

Věta 7.14. *A je úplně produktivní právě tehdy, když A je produktivní*

Důkaz.

$\Longrightarrow$ Triviálně platí.

$\Longleftarrow$ Neformálně, A je produktivní pokud $\overline{K} \leq_1 A$ a $\overline{K}$ je úplně produktivní a použijeme variaci lemma 7.8, tedy pokud B úplně produktivní a $B \leq_m A$, pak A úplně produktivní.

Formálně buď

$$W_{g(x)} = \begin{cases} \{f(g(x))\} & f(g(x)) \in W_x \\ \emptyset & f(g(x)) \notin W_x \end{cases} .$$

Pokud $f(g(x)) \notin W_x$, pak $W_{g(x)} = \emptyset$ odkud $f(g(x)) \in A$ a tedy $f(g(x)) \in A \setminus W_x$. Pokud naopak $f(g(x)) \in W_x$, pak $W_{g(x)} = \{f(g(x))\}$ a pokud $f(g(x)) \in A$, pak $W_{g(x)} \subseteq A$ odkud $f(g(x)) \in A \setminus W_{g(x)}$ což je spor, tedy nutně $f(g(x)) \in \overline{A}$ odkud $f(g(x)) \in W_x \setminus A$. Platí tedy, že $f \circ g$ je úplně produktivní funkce pro A.

■

8 Dvojice množin

Definice. Disjunktní množiny A, B se nazývají *rekurzivně neoddělitelné* jestliže je nelze rekurzivně oddělit, tedy neexistuje rekurzivní množina M taková, že $A \subseteq M$ a $B \subseteq \overline{M}$.

Definice. Disjunktní rekurzivně spočetné množiny A, B se nazývají *efektivně neoddělitelné*, jestliže existuje ORF f pro kterou, pokud $A \subseteq W_x$, $B \subseteq W_y$ a $W_x \cap W_y = \emptyset$, platí $f(x,y)\downarrow$ & $f(x,y) \notin W_x \cup W_y$.

Poznámka 8.1. Efektivně neoddělitelné je efektivně nerekurzivně oddělitelné, nebo-li efektivní neoddělitelnost implikuje rekurzivní neoddělitelnost.

Důkaz. Kdyby existovala rekurzivní množina M, pak položme $W_x = M$ a $W_y = \overline{M}$ a $f(x,y)$ musí ležet v $W_x \cup W_y$ neboť $W_x \cup W_y = \mathbb{N}$ což je spor. ■

Věta 8.2. *Existují rekurzivně spočetné množiny A, B které jsou rekurzivně neoddělitelné a nejsou efektivně neoddělitelné.*

Důkaz (Idea). Mějme A, B rekurzivně neoddělitelné, a nechť existují W_x, W_y pro které platí $A \subseteq W_x$, $B \subseteq W_y$ a $W_x \cap W_y = \emptyset$, odtud vyplývá $(\exists z)(z \notin W_x \cup W_y)$, otázka je jak takové z najít (efektivně). ■

Věta 8.3. *Existují disjunktní rekurzivně spočetné množiny A, B které jsou efektivně neoddělitelné.*

Důkaz. Buď $A = \{x|\varphi_x(x) \simeq 0\}$, $B = \{x|\varphi_x(x) \simeq 1\}$ (lze i $A = \{x|\varphi_x(x) \simeq 0\}$, $B = \{x|\varphi_x(x)\downarrow$ & $\varphi_x(x) \geq 1\}$). Tyto množiny jsou disjunktní a rekurzivně spočetné. Podle věty 4.3 existuje PRF α pro kterou

$$\varphi_{\alpha(x,y)}(w) \simeq \begin{cases} 1 & (\exists s)(T_1(x,w,s) \ \&\ (\forall j)_{j\leq s}\neg T_1(y,w,j)) \\ 0 & (\exists s)(T_1(x,w,s) \ \&\ (\forall j)_{j>s}\neg T_1(y,w,j)) \\ \uparrow & w \notin W_x \cup W_y \end{cases} .$$

Platí $\varphi_{\alpha(x,y)}(\alpha(x,y))\uparrow$. Kdyby totiž $\alpha(x,y) \in W_x$, pak by

$$(\exists s)(T_1(x,\alpha(x,y),s) \ \&\ (\forall j)_{j\leq s}\neg T_1(y,\alpha(x,y),j))$$

a tedy $\varphi_{\alpha(x,y)}(\alpha(x,y)) \simeq 1$ odkud $\alpha(x,y) \in B$ což je spor. Stejně tak pokud $\alpha(x,y) \in W_y$, pak by

$$(\exists s)(T_1(x,\alpha(x,y),s) \ \&\ (\forall j)_{j>s}\neg T_1(y,\alpha(x,y),j))$$

a tedy $\varphi_{\alpha(x,y)}(\alpha(x,y)) \simeq 0$ odkud $\alpha(x,y) \in A$ což je spor.

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

Poznámka 8.4. Nechť A, B je efektivně neoddělitelná dvojice (disjunktních rekurzivně spočetných) množin, pak $A, B, A \cup B$ jsou kreativní množiny.

Důkaz. Buď A, B efektivně neoddělitelné množiny a máme ORF f. Chceme najít ORF g takovou, že pokud $W_z \subseteq \overline{A}$, pak $g(z) \in \overline{A} \setminus W_z$. Tedy $W_x = A, W_{g(z)} = B \cup W_z$ a $g(z) = f(x, g(z))$ ■

Definice. Disjunktní dvojice $(A, B), (C, D)$ jsou $(A, B) \leq_1 (C, D)$ pokud existuje ORF h (prostá) taková, že pokud $x \in A$, pak $h(x) \in C$; pokud $x \in B$, pak $h(x) \in D$ a pokud $x \notin A \cup B$, pak $h(x) \notin C \cup D$.

Definice. Disjunktní rekurzivně spočetná dvojice (A, B) je 1-*úplná*, pokud existuje rekurzivně spočetná dvojice (C, D) pro kterou platí $(C, D) \leq_1 (A, B)$.

Lemma 8.5. *Pokud je dvojice (A, B) 1-úplná, pak je efektivně neoddělitelná.*

Důkaz. Buď (A, B) 1-úplná, (C, D) je efektivně neoddělitelná (víme, že existuje) a buď ORF h taková, že $W_{\alpha(x)} = h^{-1}(W_x)$, $W_{\beta(y)} = h^{-1}(W_y)$, nebo-li $W_{\alpha(x)} = \{z | h(z) \in W_x\}$, $W_{\beta(y)} = \{z | h(z) \in W_y\}$. Pokud $A \subseteq W_x$, $B \subseteq W_y$, pak $C \subseteq W_{\alpha(x)}$, $D \subseteq W_{\beta(y)}$ a pokud $W_x \cap W_y = \emptyset$, pak $W_{\alpha(x)} \cap W_{\beta(y)} = \emptyset$ odkud $f(\alpha(x), \beta(y)) \notin W_{\alpha(x)} \cup W_{\beta(y)}$ a $h(f(\alpha(x), \beta(y))) \notin W_x \cup W_y$. ■

Věta 8.6 (Dvojná forma věty o rekurzi)**.** *Buď ORF f, g, pak existují čísla m, n taková, že*

$$W_m = W_{f(m,n)}, W_n = W_{g(m,n)}.$$

Pokud mají f, g $k+2$ proměnných, pak existují PRF ω_1, ω_2 k proměnných takové, že

$$W_{\omega_1(z_1,\ldots,z_k)} = W_{f(\omega_1(z_1,\ldots,z_k),\omega_2(z_1,\ldots,z_k),z_1,\ldots,z_k)},$$

$$W_{\omega_2(z_1,\ldots,z_k)} = W_{g(\omega_1(z_1,\ldots,z_k),\omega_2(z_1,\ldots,z_k),z_1,\ldots,z_k)}.$$

Obecně platí

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

Důkaz. Dle věty 6.3 existuje PRF α taková, že $\varphi_{\alpha(y)} = \varphi_{f(\alpha(y),y)}$ a $m = \alpha(n)$. Vezměme $\varphi_{g(\alpha(y),y)}$ a dle věty 6.1 existuje n takové, že $\varphi_n = \varphi_{g(\alpha(n),n)}$ a $m = \alpha(n)$. Pro další varianty obdobně. ■

Lemma 8.7. *Pokud je dvojice (A, B) efektivně neoddělitelná, pak je 1-úplná.*

Důkaz. Buď (A, B) efektivně neoddělitelná a ORF f. Dokážeme, že existuje nějaká disjunktní rekurzivně spočetná dvojice (C, D) pro kterou $(C, D) \leq_1 (A, B)$. Díky větě 8.6 máme ORF ω_1, ω_2 pro které platí

$$W_{\omega_1(z)} = \begin{cases} A \cup \{f(\omega_1(z), \omega_2(z)\} & x \in D \\ A & x \notin D \end{cases}$$

$$W_{\omega_2(z)} = \begin{cases} B \cup \{f(\omega_1(z), \omega_2(z)\} & x \in C \\ B & x \notin C \end{cases}.$$

Pokud $z \notin (C \cup D)$, pak $W_{\omega_1(z)} = A$, $W_{\omega_2(z)} = B$ odkud $f(\omega_1(z), \omega_2(z)) \notin (A \cup B)$. Pokud naopak $z \in C$, pak $W_{\omega_1(z)} = A$, $W_{\omega_2(z)} = B \cup \{f(\omega_1(z), \omega_2(z)\}$ odkud $f(\omega_1(z), \omega_2(z)) \in A$ (kdyby $f(\omega_1(z), \omega_2(z)) \notin A$, pak by $f(\omega_1(z), \omega_2(z)) \notin A \cup B \cup \{f(\omega_1(z), \omega_2(z))\}$ což je spor) a obdobně pokud $z \in D$, pak $f(\omega_1(z), \omega_2(z)) \in B$. ■

Věta 8.8. *Efektivně neoddělitelné, disjunktní rekurzivně spočetné dvojice jsou právě 1-úplné.*

Důkaz. Vyplývá z lemma 8.5 a 8.7. ■

Tvrzení 8.9. *Efektivně neoddělitelné dvojice jsou všechny navzájem rekurzivně izomorfní.*

9 Gödelovy věty

Definice. Teorie je *axiomatizovatelná* právě tehdy, když množina dokazatelných formulí je rekurzivně spočetná.

Definice. Jazyk obsahující numerály $\overline{0}, \overline{1}$, funkční symboly $+, \times$ a konečně mnoho axiomů (komutativita, asociativita, distributivita) označujeme jako *základní aritmetickou sílu* (Robinsonova aritmetika).

Definice. Funkce φ je reprezentovatelná v teorii T pokud existuje formule $\mathcal{F}$ taková, že

(1) pokud $\varphi(x_1, \ldots, x_n) \simeq y$, pak $\vdash_T \mathcal{F}(\overline{x}_1, \ldots, \overline{x}_n, \overline{y})$

(2) $\vdash_T \mathcal{F}(\alpha_1, \ldots, \alpha_n, \beta)$ & $\vdash_T \mathcal{F}(\alpha_1, \ldots, \alpha_n, \gamma) \Longrightarrow \beta = \gamma$ (funkcionální vlastnost).

Věta 9.1. *Každá ČRF je reprezentovatelná v libovolné teorii ZAS, dokonce existuje (k libovolné ČRF) jediná formule $\mathcal{F}$, která ji reprezentuje ve všech teoriích ZAS.*

Věta 9.2. *RSP jsou právě ty, které jsou vyjádřitelné ve tvaru $(\exists p_1(\vec{x}) = p_2(\vec{x}))$. $\varphi(x_1, \ldots, x_n) \simeq y$ právě tehdy, když $\exists_1$ podmínka, tedy $(\exists \vec{z})(p_1(\vec{z}, \vec{x}, y) = p_2(\vec{z}, \vec{x}, y))$.*

Důsledek 9.3. *ČRF jsou reprezentovatelné tzv. Σ_1 formulí, tzn. formulí tvaru $(\exists \vec{z})(p_1(\vec{z}, \vec{x}, y) = p_2(\vec{z}, \vec{x}, y))$.*

Tvrzení 9.4. *Pravdivé Σ_1 formule v $\mathbb{N}$ jsou v ZAS dokazatelné.*

Důsledek 9.5. *Nechť A, B jsou disjunktní rekurzivně spočetné množiny a T teorie ZAS, pak existuje formule G a platí, pokud $x \in A$, pak $\vdash_T \mathcal{G}(\overline{x})$ a pokud $x \in B$, pak $\vdash_T \neg\mathcal{G}(\overline{x})$.*

Důkaz. Buď

$$\varphi(x) \simeq \begin{cases} 0 & x \in A \\ 1 & x \in B \\ \uparrow & x \notin A \cup B \end{cases}$$

a $\varphi(x)$ budeme reprezentovat jako $\mathcal{F}(x, y)$, $\mathcal{G}(x) \equiv \mathcal{F}(x, \overline{0})$. Pokud $k \in A$, pak $\vdash_T \mathcal{F}(\overline{k}, \overline{0})$; jinak pokud $k \in B$, pak $\vdash_T \mathcal{F}(\overline{k}, \overline{1})$ odkud $\vdash_T \neg\mathcal{F}(\overline{k}, \overline{0})$. ■

Věta 9.6 (Göedelovy věty)**.** *Pro libovolnou bezespornou teorii T ZAS platí:*

(1) množina dokazatelných formulí není rekurzivní

(2) je-li navíc T axiomatizovatelná, pak existují uzavřené formule $\mathcal{F}$ pro které $\nvdash_T \mathcal{F}, \nvdash_T \neg\mathcal{F}$

(3) bezespornost T nelze v T dokázat (za určitých předpokladů postačuje Σ_1 indukce).

Důkaz. Mějme překladač ČRF do aritmetické teorie a T buď teorie ZAS. Buď A, B efektivně neoddělitelné množiny. Pak $A_1 = \{x | \vdash_T \mathcal{G}(\overline{x})\}$ a $B_1 = \{x | \vdash_T \neg\mathcal{G}(\overline{x})\}$. $A \subseteq A_1, B \subseteq B_1$ a pokud T je bezesporná, pak $A_1 \cap B_1 = \emptyset$. Tedy A_1, B_1 nesmí být rekurzivní, protože jinak by A, B byly rekurzivně oddělitelné.

Pokud je navíc T axiomatizovatelná, pak A_1, B_1 jsou rekurzivně spočetné. Protože A, B jsou efektivně neoddělitelné, $(\exists k) \notin A_1 \cup B_1$, tzn. $\nvdash_T \mathcal{G}(\overline{k})$ a $\nvdash_T \neg\mathcal{G}(\overline{k})$. ■

Tvrzení 9.7. *V teorii T ZAS která je axiomatizovatelná tvoří dvojice (dokazatelné, vyvratitelné) efektivně neoddělitelnou dvojici.*