

Univerzita Karlova v Praze, Filozofická fakulta
Katedra logiky

ANNA HORSKÁ

FRIEDBERG-MUCHNIKOVA VETA

Ročníková práce

Vedúci práce: Vítězslav Švejdar

2007

Prehlasujem, že som ročníkovú prácu vypracovala samostatne a že som uviedla všetky použité pramene a literatúru.

V Prahe, 7. mája 2007

Anna Horská

Obsah

1	Úvod	3
2	Turingova prevoditeľnosť	4
3	Friedberg-Muchnikova veta	7
3.1	Dôkaz	7
3.2	Prioritná metóda	13
4	Prioritná metóda a Σ_3-kompletnosť	14

1 Úvod

V roku 1944 si poľský matematik a logik Emil Leon Post položil otázku, či existujú rekurzívne spočetné množiny, ktoré nie sú ani obecné rekurzívne ani T-kompletné. Táto otázka sa stala známa ako Postov problém. Ostala nevyriešená až do roku 1956, keď Richard M. Friedberg a Albert Abramovič Muchnik nezávisle na sebe dokázali, že takéto množiny existujú. Načrtnime teraz cestu, ktorá doviedla Posta k tejto otázke.

V roku 1920 dokončil Post svoju dizertačnú prácu, v ktorej analyzoval systém z Russellovho a Whiteheadovho diela *Principia Mathematica*. Podarilo sa mu dokázať úplnosť, konzistentnosť a rozhodnuteľnosť výrokového kalkulu uvedeného v spomínanom diele. Práve na poslednú zmieňovanú vlastnosť formálnych systémov sa sústredil v nasledujúcom roku. Pokúsil sa otázku rozhodnuteľnosti zovšeobecniť pre ľubovoľný formálny systém a na daný účel definoval *kanonický systém*, abstrakciu formálneho systému, viz. [Odi89]. Následne ukázal, že systém z *Principia Mathematica* sa dá preložiť do *kanonického systému*. Začal sa teda zaoberať otázkou rozhodnuteľnosti *kanonických systémov* a dúfal, že nezávislosť od detailov konkrétnych systémov mu prácu uľahčí. Čoskoro však narazil na problém, ktorý sa zdal byť vo všeobecnosti veľmi komplikovaný. Preto zmenil svoj pôvodný plán a začal sa zaujímať o nerozhodnuteľnosť. Definoval univerzálny *kanonický systém*, ktorý bol vlastne verziou množiny K_0 a ukázal, že jeho diagonálna množina K je rekurzívne spočetná ale nerekurzívna. Aby z tohto mohol vyvodiť ďalšie výsledky o nerozhodnuteľnosti, potreboval Post istú verziu Churchovej tézy, ktorú formuloval takto: každá efektívne generovateľná množina je generovateľná v *kanonickom systéme*. V modernej terminológii je to ekvivalentné výroku, že efektívne generovateľné množiny sú rekurzívne spočetné. Celá táto Postova práca, ktorú dokončil v roku 1921, ostala neuvyverejnená, pretože Post nebol presvedčený o svojej verzii Churchovej tézy a požadoval pre ňu verifikáciu vo forme psychologickéj analýzy počítačného procesu. Krok k takejto analýze urobil v roku 1936, keď vymyslel verziu Turingovho stroja nezávisle od Turinga. *Kanonické systémy* boli publikované až v roku 1943 v článku [Pos43].

Vzájomné prevoditeľnosti medzi rôznymi *kanonickými systémami* priviedli Posta k vytvoreniu konceptu prevoditeľnosti medzi množinami. Dovtedy známe dôkazy nerozhodnuteľnosti formálnych systémov (Post, Church. . .) využívali prevoditeľnosť množiny K na daný systém. (Keby bol daný systém rozhodnuteľný, vedeli by sme rozhodnúť i o náležaní do K a to je spor, lebo K nie je rekurzívna len rekurzívne spočetná.) Post si teda položil otázku, či sa dá K previesť na každý nerozhodnuteľný systém (rekurzívne spočetnú nerekurzívnu množinu), alebo existujú rekurzívne spočetné množiny, na ktoré K nemôžeme previesť. Ekvivalentná otázka je, či existujú rekurzívne spočetné nerekurzívne množiny, ktoré nie sú kompletné. (Např. pre T-prevoditeľnosť: Ak nájdeme A , ktorá nie je T-kompletná, tak existuje B rekurzívne spočetná, že $B \not\leq_T A$. Množina K je T-kompletná, preto $B \leq_T K$. Keby $K \leq_T A$, tak z tranzitivity relácie \leq_T dostávame $B \leq_T A$ a to je spor. Z toho vyplýva, že K nie je T-prevoditeľná na A . Naopak, ak K nie je T-prevoditeľná na A , tak A samozrejme nie je T-kompletná.)

V roku 1944 napísal Post článok [Pos44]. Predstavil v ňom prevoditeľnosti silnejšie ako T-prevoditeľnosť (ktorú v roku 1936 navrhol Alan M. Turing), konkrétne *many-one* (m), *one-one* (1), *bounded truth-table* (btt), *truth-table* (tt) prevoditeľnosti. Ukázal existenciu prostých množín a dokázal, že nie sú btt-kompletné (a preto ani m- a 1-kompletné). Ďalej definoval hyperproste množiny, o ktorých opäť dokázal, že existujú a nemôžu byť tt-kompletné. (Žiadna kreatívna množina nie je btt- resp. tt- prevoditeľná na prostú resp. hyperprostú množinu.) Následne sa zaoberal tou istou otázkou pre T-prevoditeľnosť a vyvinul koncept špeciálneho druhu hyperprostých množín tzv. hyperhyperprostých množín. Otázkou ich existencie a T-*ne*kompletnosti nechal však otvorenú. Túto vyriešil C. E. M. Yates, ktorý ukázal hyperhyperprostú množinu, ktorá T-kompletná je.

Článok [Pos44] je pozoruhodný vďaka Postovmu problému, teda problému existencie rekurzívne spočetných, nerekurzívnych, T-*ne*kompletných množín a vďaka tzv. Postovmu programu. Postovým programom bolo pokúsiť sa skonštruovať rekurzívne spočetnú nerekurzívnu a nekompletnú množinu pomocou definovania vlastností množiny, dokázaním, že množina s danými vlastnosťami existuje a ukázaním, že tieto vlastnosti zaručia jej nerekurzivnosť a nekompletnosť. Ako sme videli, do istej miery bol úspešný, no pre T-prevoditeľnosť zlyhal.

Postov problém bol významný v dvoch ohľadoch:

- kladie si otázku rôznorodosti rekurzívne spočetných množín,
- ak by K bola prevoditeľná na každý nerozhodnuteľný systém, stala by sa prevoditeľnosť z K univerzálnou metódou dôkazu nerozhodnuteľnosti formálneho systému.

Predmetom tejto práce je dokázať Friedberg-Muchnikov teorém, ktorý je riešením Postovho problému. Súčasne bude čitateľovi predstavená prioritná metóda, ktorú dôkaz využíva. Najprv sa ale stručne oboznámme s T-prevoditeľnosťou, ktorá je pokladaná za najzákladnejšiu prevoditeľnosť a pre ktorú Post svoj problém formuloval.

2 Turingova prevoditeľnosť

Povieme, že množina A je *rekurzívna v množine* B práve vtedy, keď existuje program s orákulom B , ktorý rozhoduje o náležaní do A . To znamená, že program dostane vstup x , na ktorý začne výpočet. V ľubovoľnom kroku výpočtu sa môže orákula B spýtať otázku typu „patrí y množine B ?“. Orákulum mu správne odpovie. Výstupom je áno ak $x \in A$ alebo nie ak $x \notin A$.

Povieme, že množina A je *rekurzívne spočetná v množine* B práve vtedy, keď existuje program s orákulom B , ktorý prijíma množinu A . To znamená, že program dostane vstup x ,

na ktorý začne výpočet. V ľubovoľnom kroku výpočtu sa môže orákula B spýtať otázku typu „patrí y množine B ?“. Orákulum mu správne odpovie. Program sa dopočíta vtedy a len vtedy, keď $x \in A$. Inak sa zacyklí.

Definícia A je turingovsky prevoditeľná na B , ak A je rekurzívna v B . Značíme $A \leq_T B$.

Definícia B je T -kompletná práve vtedy, keď platí:

- $B \in RS$,
- $\forall A(A \in RS \rightarrow A \leq_T B)$.

Definícia $y \in W_z^B \Leftrightarrow$ program s indexom z a orákulum B sa dopočíta na vstup y . W_z^B je vlastne „definičným oborom“ programu s orákulum, ktorý má index z , ak je k nemu pripojené orákulum množiny B .

Vieme, že pre program s orákulum s ľubovoľným indexom z existuje rekurzívne spočítaná množina $W_{\rho(z)}$, viz. [Rog67], ktorej prvky, zjednodušene povedané, sú dvojice $[x, s]$. Prvok x je kandidát na náležanie do $W_z^{\text{orákulum}}$ a prvok s kóduje prvky, na ktoré má naše zvolené orákulum v priebehu výpočtu na vstup x odpovedať nie a na ktoré má odpovedať áno. Dáta s nám tak poskytnú informáciu o tom, či x pre zvolené orákulum, napr. orákulum B , patrí do W_z^B alebo nie. Ak B je rekurzívne (napr. konečné), tak aj získanie tejto informácie je rekurzívne. Z toho vyplýva, že množina W_z^B pre rekurzívne B je rekurzívne spočítaná a sme schopní generovať jej prvky. (Budeme generovať $W_{\rho(z)}$. Pre každý prvok $W_{\rho(z)}$ skontrolujeme, či s súhlasí s našim orákulum, teda či prvky, na ktoré sa má odpovedať áno patria do orákula a prvky, na ktoré sa má odpovedať nie nepatria do orákula. Ak súhlasí, zaradíme x do W_z^B , ak nie, pokračujeme v generovaní $W_{\rho(z)}$.) Pod pojmom n krokov vo výpočte množiny W_z^B budeme rozumieť n strojových krokov v generovaní množiny $W_{\rho(z)}$.

Definícia A je rekurzívne spočítaná v B ak $\exists z(A = W_z^B)$.

Neformálne je to zapísané v poslednom odseku na strane 4.

Lemma 1 A je rekurzívna v $B \Leftrightarrow A$ i \bar{A} sú rekurzívne spočítané v B .

Dôkaz (\Rightarrow) Ak A je rekurzívna v B , tak existuje program, ktorý s pomocou orákula B rozhoduje o náležaní do A . Tento môžeme ľahko modifikovať. Keď dá výstup áno, nič nemeníme, keď dá výstup nie, necháme ho zacyklíť sa. Z toho vyplýva, že A je rekurzívne spočítaná v B . Keď máme program, ktorý rozhoduje o náležaní do A , máme aj program, ktorý rozhoduje o náležaní do \bar{A} . Teda \bar{A} je tiež rekurzívna v B , čo opäť znamená, že \bar{A} je aj rekurzívne spočítaná v B .

(\Leftarrow) Nech A a \bar{A} sú rekurzívne spočítateľné v B . Máme teda program P_A , ktorý prijíma množinu A a máme program $P_{\bar{A}}$, ktorý prijíma množinu \bar{A} . Chceme z nich vytvoriť program P , ktorý rozhoduje o náležaní do A . Vezmime si ľubovoľný vstup x . Na vstupe x necháme súčasne počítať P_A i $P_{\bar{A}}$. Jeden z nich sa v konečnom čase dopočíta, pretože $x \in A$ alebo $x \in \bar{A}$. Keď P_A prijme x , tak P dá na vstupe x odpoveď áno. Keď $P_{\bar{A}}$ prijme x , tak P dá na vstupe x odpoveď nie. Takto skonštruovaný P je totálny a rozhoduje o náležaní do A . \square

Po krátkych úvodných úvahách nasledujú úvahy bezpodmienečne nutné k pochopeniu dôkazu:

Lemma 2 Nech A, B sú rekurzívne spočítateľné množiny. Potom platí:

$$A \not\leq_T B \Leftrightarrow \exists f \forall x (f(x) \in A \Leftrightarrow f(x) \in W_x^B)$$

Dôkaz Keď A je rekurzívne spočítateľná, tak existuje program, ktorý sa doráta vtedy a len vtedy, keď jeho vstup x patrí A . Môžeme povedať, že je to program s orákulom B (hoci sa orákula v priebehu výpočtu nič nespýta). Teda, keď A je rekurzívne spočítateľná, tak je aj rekurzívne spočítateľná v B . Potom

$$A \not\leq_T B \Leftrightarrow A \text{ nie je rekurzívna v } B \Leftrightarrow \text{def. T-prevoditeľnosti}$$

$$\Leftrightarrow A \text{ alebo } \bar{A} \text{ nie je rekurzívne spočítateľná v } B.$$

Lemma 1

Z predpokladu vieme, že A je rekurzívne spočítateľná v B , teda dostávame, že \bar{A} nie je rekurzívne spočítateľná v B .

$$\begin{aligned} \bar{A} \text{ nie je rekurzívne spočítateľná v } B &\Leftrightarrow \forall z (\bar{A} \neq W_z^B) \Leftrightarrow \\ &\Leftrightarrow \forall z \exists y ((y \in \bar{A} \ \& \ y \notin W_z^B) \vee (y \notin \bar{A} \ \& \ y \in W_z^B)) \Leftrightarrow \\ &\Leftrightarrow \forall z \exists y ((y \notin A \ \& \ y \notin W_z^B) \vee (y \in A \ \& \ y \in W_z^B)) \Leftrightarrow \\ &\Leftrightarrow \forall z \exists y (y \in A \Leftrightarrow y \in W_z^B) \end{aligned}$$

Pre každé z môže existovať aj viac y splňujúcich danú vlastnosť. Vyberme práve jedno y pre každé z a označme ho $f(z)$. Definovali sme tak funkciu f (nemusí byť rekurzívna) s vlastnosťou $\forall z (f(z) \in A \Leftrightarrow f(z) \in W_z^B)$.

Naopak, keď máme funkciu f (nie nutne rekurzívnu) splňujúcu danú vlastnosť, tak dôkaz, že $A \not\leq_T B$ je veľmi podobný tomuto. Vieme, že pre každé z existuje y (y je práve funkčná hodnota $f(z)$) s vlastnosťou $(y \in A \Leftrightarrow y \in W_z^B)$. Vracaním sa späť po ekvivalenciách prideme až k tvrdeniu, že \bar{A} nie je rekurzívne spočítateľná v B . Podľa Lemmy 1 potom A nie je rekurzívna v B , čo podľa definície T-prevoditeľnosti znamená, že $A \not\leq_T B$. \square

Veta (Friedberg, Muchnik)

Existujú rekurzívne spočítateľné množiny A, B nezrovnateľné vzhľadom k relácii \leq_T , teda $A \not\leq_T B$ a $B \not\leq_T A$.

3 Friedberg-Muchnikova veta

3.1 Dôkaz

Aby sme dokázali Friedberg-Muchnikovu vetu, potrebujeme zadať inštrukcie na skonštruovanie množín A, B (tak, aby ich mohol simulovať napr. Turingov stroj). A, B musia byť rekurzívne spočítateľné (ale nie rekurzívne) a nezrovnateľné vzhľadom k \leq_T . To znamená, že podľa Lemmy 2 stačí ukázať existenciu funkcií f, g takých že

$$\forall x(f(x) \in A \Leftrightarrow f(x) \in W_x^B) \quad \& \quad \forall x(g(x) \in B \Leftrightarrow g(x) \in W_x^A).$$

Konštrukcia množín A, B používa dva identické nekonečné zoznamy všetkých prirodzených čísel v rastúcom poradí. Budeme ich nazývať A -zoznam a B -zoznam. Predstavíme si, že zoznamy sú vertikálne, teda, keď chceme nájsť väčšie číslo, musíme sa posunúť nadol. V každom okamihu výpočtu platí, že predchádzajúci výpočet použil len konečnú časť každého zoznamu. V priebehu výpočtu označíme niektoré prirodzené čísla na oboch zoznamoch znamienkom plus (+) a niektoré znamienkom mínus (-). Znamienko plus v A -zozname označuje čísla, ktoré sú v danom štádiu prijaté do množiny A . Podobne v B -zozname. Znamienko mínus dočasne označuje fakt, že takto ohodnotené prirodzené číslo nepatrí do množiny A resp. B (podľa toho, v ktorom zozname sa dané číslo nachádza). V priebehu výpočtu je povolené, že znamienko mínus bude vymazané a nahradené znamienkom plus. Inštrukcie algoritmu ale zaistia, že znamienko plus nikdy vymazané nebude. Výpočet bude postupovať v krokoch. Je možné, ale nie isté, že v danom kroku pridáme nejaký prvok do množiny A resp. B .

Definícia $A_0 = B_0 = \emptyset$
 $A_n = \{x, x \text{ má znamienko plus v } A\text{-zozname na konci kroku } n\}$
 $B_n = \{x, x \text{ má znamienko plus v } B\text{-zozname na konci kroku } n\}$
 $n = 1, 2, 3 \dots$

Keďže znamienka plus mazané nebudú, vidíme, že platí: $A_0 \subseteq A_1 \subseteq A_2 \subseteq \dots$
 $B_0 \subseteq B_1 \subseteq B_2 \subseteq \dots$

Definícia $A = \bigcup_0^\infty A_n, \quad B = \bigcup_0^\infty B_n.$

Okrem A -zoznamu a B -zoznamu máme ešte dve nekonečné sady premiestniteľných značiek, jedna sada pre jeden zoznam. Značky zo sady pre A -zoznam zapisujeme $\boxed{0}_A, \boxed{1}_A, \boxed{2}_A \dots$. Značky zo sady pre B -zoznam zapisujeme $\boxed{0}_B, \boxed{1}_B, \boxed{2}_B \dots$. Tieto značky môžeme zoradiť nasledovne podľa ich zmenšujúcej sa priority: $\boxed{0}_A, \boxed{0}_B, \boxed{1}_A, \boxed{1}_B, \boxed{2}_A, \boxed{2}_B \dots$. Sú premiestniteľné v tomto zmysle: i -ta značka v prioritnom usporiadaní sa prvýkrát objaví v i -tom kroku algoritmu (značky ako aj kroky číslujeme od jedna). Je priradená niektorému prirodzenému číslu v zozname, s ktorým korešponduje. Je možné, že existuje budúci

krok, v ktorom jej spojenie s týmto prirodzeným číslom skončí a bude priradená väčšiemu prirodzenému číslu, ktoré je nižšie v zozname. Konkrétnejšie povedané, keď ľubovoľné číslo s ľubovoľnou značkou v ľubovoľnom zozname dostane znamienko plus, všetky značky s nižšou prioritou v druhom zozname budú presunuté nadol, ale tak, aby ich poradie podľa priority ostalo zachované. V každom kroku algoritmu bude mať teda značka s vyššou prioritou vyššiu pozíciu v zozname ako značka s nižšou prioritou. Posun jednej značky v priebehu výpočtu môže nastať viackrát. Z opisu algoritmu budú zrejmé tieto vzťahy medzi značkami a znamienkami:

- Znamienko plus môže byť zapísané len k číslu, pri ktorom je v danom kroku značka.
- Ak je k číslu zapísané znamienko plus, tak značka, ktorá zápis tohto plus spôsobila je posledná, ktorá sa pri čísle môže vyskytovať (v zmysle, že keď táto bude posunutá, tak už žiadna značka sa k číslu nedostane).
- Znamienko plus nebude nikdy prepísané ani zmazané.
- Mínus môže byť udelený len číslam bez znamienka plus, pričom nie je zakázané, aby pri nich bola nejaká značka.
- K číslu so znamienkom mínus nemôže prísť značka, ale mohla sa tam už nachádzať, keď sme mínus zapisovali.
- Nie je vylúčené, že znamienko mínus pri čísle bude prepísané na plus. Toto môže spôsobiť jedine značka, ktorá pri danom čísle bola v čase, keď mu bolo udelené mínus.

Ukážeme, až konštrukciu presne popíšeme, že každá značka je posunutá len konečnekrát. To znamená, že každá má pozíciu v zozname, ktorej je definitívne priradená. (Pozícia je prirodzené číslo v zozname, pri ktorom sa značka práve nachádza.)

Označme konečnú pozíciu značky \boxed{j}_A v A -zozname $f(j)$ a konečnú pozíciu značky \boxed{j}_B v B -zozname nech je $g(j)$. Konštrukcia bude prevedená tak, aby pre každé j bolo možné dokázať, že platí:

$$\begin{aligned} f(j) \in A &\Leftrightarrow f(j) \text{ má plus} \Leftrightarrow f(j) \in W_j^B, \\ g(j) \in B &\Leftrightarrow g(j) \text{ má plus} \Leftrightarrow g(j) \in W_j^A. \end{aligned}$$

Keď sa nám to podarí, dostaneme, že existujú funkcie f, g , ktoré podľa Lemmy 2 zaručia

$$A \not\leq_T B \text{ a } B \not\leq_T A.$$

Definícia Povieme, že prirodzené číslo v zozname je *voľné*, keď jemu a ľubovoľnému väčšiemu prirodzenému číslu nie je v danom čase priradené žiadne znamienko alebo značka.

Povieme, že prirodzené číslo v zozname je *neobsadené*, keď mu v danom čase nie je priradené znamienko plus.

Opíšme teraz inštrukcie na výpočet množín A , B :

Krok 1: Priraď číslu 0 v A -zozname značku $\boxed{0}_A$.

Krok 2: Priraď číslu 0 v B -zozname značku $\boxed{0}_B$.

...

Krok $2n + 1$ ($n > 0$):

- Priraď prvému *voľnému* číslu v A -zozname značku \boxed{n}_A . Nech $a_0^{(n)}, \dots, a_n^{(n)}$ sú aktuálne pozície značiek $\boxed{0}_A, \dots, \boxed{n}_A$ v A -zozname.
- Urob n krokov¹ vo výpočte množín $W_0^{B_{2n}}, \dots, W_n^{B_{2n}}$. Nech $a_i^{(n)}$ je najmenšie prirodzené číslo z množiny $\{a_0^{(n)}, \dots, a_n^{(n)}\}$ také, že $a_i^{(n)}$ je *neobsadené* a súčasne sa pri robení n krokov zistilo, že patrí do $W_i^{B_{2n}}$. Ak také $a_i^{(n)}$ nenájdeš, choď rovno na krok $2n + 2$ a ostatné inštrukcie z tohto kroku vynechaj.
- Označ $a_i^{(n)}$ v A -zozname znamienkom plus (+). Označ znamienkom mínus (−) všetky tie prirodzené čísla v B -zozname, ktoré patria do \overline{B}_{2n} a na ktoré sme sa orákula B_{2n} pýtali pri ukazovaní, že $a_i^{(n)} \in W_i^{B_{2n}}$. Tieto sú nutne *neobsadené*, pretože inak by nepatrili do komplementu B_{2n} a orákulum by na ne neodpovedalo nie.
- Ak $i < n$, prejdí k B -zoznamu a presuň všetky značky \boxed{j}_B , ktoré majú menšiu prioritu ako \boxed{i}_A nadol k *voľným* prirodzeným číslam v B -zozname. Tj. presuň všetky \boxed{j}_B také, že $i \leq j < n$. Pozor, \boxed{i}_B sa presúva, lebo v rade podľa zmenšujúcej sa priority stojí za \boxed{i}_A , \boxed{n}_B sa nepresúva, pretože sa v B -zozname ešte nenachádza. Do výpočtu bude zahrnuté až v nasledujúcom kroku.
Nová pozícia značiek je taká, že značka s vyššou prioritou je umiestnená vyššie v zozname ako značka s nižšou prioritou.

Krok $2n + 2$ ($n > 0$):

- Priraď prvému *voľnému* číslu v B -zozname značku \boxed{n}_B . Nech $b_0^{(n)}, \dots, b_n^{(n)}$ sú aktuálne pozície značiek $\boxed{0}_B, \dots, \boxed{n}_B$ v B -zozname.
- Urob n krokov vo výpočte množín $W_0^{A_{2n+1}}, \dots, W_n^{A_{2n+1}}$. Nech $b_i^{(n)}$ je najmenšie prirodzené číslo z množiny $\{b_0^{(n)}, \dots, b_n^{(n)}\}$ také, že $b_i^{(n)}$ je *neobsadené* a súčasne sa pri robení n krokov zistilo, že patrí do $W_i^{A_{2n+1}}$. Ak také $b_i^{(n)}$ nenájdeš, choď rovno na krok $2n + 3$ a ostatné inštrukcie z tohto kroku vynechaj.

¹Pozor, n krokov vo výpočte množiny W_x^B je n strojových krokov pri generovaní množiny $W_{\rho(x)}$.

- Označ $b_i^{(n)}$ v B -zozname znamienkom plus (+). Označ znamienkom mínus (−) všetky tie prirodzené čísla v A -zozname, ktoré patria do \bar{A}_{2n+1} a na ktoré sme sa orákula A_{2n+1} pýtali pri ukazovaní, že $b_i^{(n)} \in W_i^{A_{2n+1}}$. Tieto sú nutne *neobsadené*, pretože inak by nepatrili do komplementu A_{2n+1} a orákulum by na ne neodpovedalo nie.
- Ak $i < n$, prejdí k A -zoznamu a presuň všetky značky \boxed{j}_A , ktoré majú menšiu prioritu ako \boxed{i}_B nadol k *voľným* prirodzeným číslam v A -zozname. Tj. presuň všetky \boxed{j}_A také, že $i < j \leq n$. Pozor, \boxed{i}_A sa už nepresúva, lebo v prioritnom poradí stojí pred \boxed{i}_B .

Nová pozícia značiek je taká, že značka s vyššou prioritou je umiestnená vyššie v zozname ako značka s nižšou prioritou.

Môžeme tiež vidieť, že množiny A, B takto skonštruované sú rekurzívne spočetné: práve sme opísali procedúru, ktorá ich generuje. Nie sú ale rekurzívne, pretože rekurzívna množina je T-prevoditeľná na ľubovoľnú inú množinu. (Existuje program, ktorý rozhoduje o náležaní do rekurzívnej množiny a môžeme si predstaviť, že je k nemu pripojené ľubovoľné orákulum, ktorého sa pri výpočte aj tak nič pýtať nebudeme.) Lenže keď dôkaz dokončíme, budeme mať $A \not\leq_T B$ a $B \not\leq_T A$.

Ukážme teraz, že každá premiestniteľná značka je posunutá len konečnekrát, takže existuje prirodzené číslo, ktoré je jej finálna pozícia. Nech $p(x)$ je počet rôznych pozícií, ktoré zaujala značka x odvtedy, čo sa prvý raz objavila vo výpočte. Zvážme, kedy môže byť značka \boxed{i}_A posunutá. Uvedomme si, že ak je k -krát posunutá, tak má $k+1$ rôznych pozícií (k posunutí a jedna počiatočná). Značka \boxed{i}_A je posunutá vždy, keď $\boxed{i-1}_A$ je posunutá. (Ak $\boxed{i-1}_A$ je posunutá, tak aj všetky, ktoré majú menšiu prioritu ako ona.) Keďže $\boxed{i-1}_A$ je vo výpočte predstavená skôr ako \boxed{i}_A , môže sa stať, že bude posunutá, kým \boxed{i}_A vo výpočte ešte nie je. (Ak sme $\boxed{i-1}_A$ predstavili v kroku $2l+1$, kde $l = i-1$, tak hneď v $2l+2$ môže byť posunutá, pričom značku \boxed{i}_A ideme predstavovať až v nasledujúcom kroku.) Počet posunutí značky $\boxed{i-1}_A$ je teda väčší alebo rovný ako počet posunutí značky \boxed{i}_A spôsobených značkou $\boxed{i-1}_A$. Ak $\boxed{i-1}_A$ nie je posunutá, \boxed{i}_A môže ešte byť, ak $\boxed{i-1}_B$ dostane v danom kroku plus. $\boxed{i-1}_B$ môže dostať znamienko plus najviac raz na každej svojej pozícií. Takže \boxed{i}_A sa posunie najviac raz za každú pozíciu značky $\boxed{i-1}_B$. Dostávame teda $p(\boxed{i}_A) \leq p(\boxed{i-1}_A) + p(\boxed{i-1}_B)$. Podobne značka \boxed{i}_B je posunutá vždy, keď $\boxed{i-1}_B$ je posunutá a vždy, keď číslu pri značke \boxed{i}_A je pridelený plus. V tomto prípade dostávame $p(\boxed{i}_B) \leq p(\boxed{i-1}_B) + p(\boxed{i}_A)$. Nech $z(i)$ je i -ta značka v prioritnom usporiadaní. Potom môžeme povedať, že $p(z(i)) \leq p(z(i-1)) + p(z(i-2))$.

Definícia Fibonacciho postupnosť je postupnosť čísel 1, 1, 2, 3, 5, 8, 13, 21 ..

definovaná nasledovne: $F(0) = 1, F(1) = 1,$

$$F(n) = F(n-1) + F(n-2) \text{ pre } n > 1.$$

Lemma 3 $p(z(i)) \leq F(i)$ pre $\forall i \geq 1$ (Pozor, značky číslujeme od jedna!)

Dôkaz Indukciou podľa i :

Značka $\boxed{0}_A$ sa neposúva nikdy. Teda má len jednu stálu pozíciu a platí

$$p(\boxed{0}_A) \leq F(1) = 1.$$

Značka $\boxed{0}_B$ sa posunie najviac raz, keď $\boxed{0}_A$ dostane znamienko plus. Preto môže zaujať najviac dve rôzne pozície a dostávame

$$p(\boxed{0}_B) \leq F(2) = 2.$$

Nech pre prvú až i -tu značku daný vzťah už platí. Chceme ukázať, že platí aj pre $i + 1$. značku, teda pre $z(i + 1)$:

$$p(z(i + 1)) \leq p(z(i)) + p(z(i - 1)) \leq F(i) + F(i - 1) = F(i + 1) \quad \square$$

Dostávame, že každá značka má konečnú pozíciu. Nech $f(x)$ je konečná pozícia značky \boxed{x}_A a $g(x)$ nech je konečná pozícia značky \boxed{x}_B .

Lemma 4 $f(x)$ dostane znamienko plus(v A -zozname) $\Leftrightarrow f(x) \in W_x^B$

$g(x)$ dostane znamienko plus(v B -zozname) $\Leftrightarrow g(x) \in W_x^A$

Dôkaz (\Rightarrow) $f(x)$ je konečná pozícia značky \boxed{x}_A . Teda, $f(x)$ je isté prirodzené číslo v A -zozname. Predpokladáme, že mu je priradené znamienko plus. $f(x)$ muselo toto znamienko dostať v kroku $2n + 1$ pre nejaké n . To znamená, že v tomto kroku sa vyskytlo v množine aktuálnych pozícií značiek $\boxed{0}_A, \dots, \boxed{n}_A$. Súčasne bolo najmenšie z týchto pozícií také, že pri ňom nebolo žiadne znamienko plus a objavilo sa v n krokoch výpočtu množiny $W_x^{B_{2n-1}}$. Ak všetky znamienka mínus napísané v tomto kroku² nebudú v žiadnom z ďalších krokov prepísané na plus, $f(x)$ sa vyskytne aj v n krokoch výpočtu množiny W_x^B , a preto bude platiť $f(x) \in W_x^B$. Na to, aby znamienko mínus pri čísle bolo zmenené na plus, musí byť toto číslo v niektorom ďalšom kroku pozíciou istej pohyblivej značky. Vieme, že v kroku, keď $f(x)$ dostalo znamienko plus, bolo označené značkou \boxed{x}_A a všetky značky menšej priority ako \boxed{x}_A , nachádzajúce sa v B -zozname ($\boxed{x}_B, \dots, \boxed{n-1}_B$), boli posunuté nadol k *voľným* prirodzeným číslam (číslam bez znamienok a značiek). Tým pádom sa dostali za všetky prirodzené čísla, ktorým sme v tom kroku prideliť mínus. Z toho vyplýva, že značky s menšou prioritou ako \boxed{x}_A nespôsobia zmenu žiadneho z týchto mínus na plus. Nech teda existuje značka \boxed{w}_B s vyššou prioritou ako \boxed{x}_A , ktorá niektoré z týchto mínus zmení na plus. To by spôsobilo, že všetky značky v A -zozname, ktoré majú menšiu prioritu ako \boxed{w}_B (teda aj \boxed{x}_A) by sa posunuli nadol. To je spor s tým, že \boxed{x}_A sa pri čísle $f(x)$ nachádza na svojej konečnej pozícii.

¹Tvrdíme, že $f(x)$ získalo plus práve vtedy, keď pri ňom bola značka \boxed{x}_A . Nemohlo získať plus už skôr s inou značkou? Nie, lebo to by sa \boxed{x}_A už k nemu nikdy nedostala, keďže značky sa priradujú len *voľným* prirodzeným číslam. Nemohlo získať plus až neskôr s inou značkou? Nie, pretože $f(x)$ je konečnou pozíciou značky \boxed{x}_A .

²Znamienka mínus napísané v tomto kroku označujú tie prvky, ktoré patria komplementu B_{2n} a na ktoré sme sa v priebehu dôkazu, že $f(x) \in W_x^{B_{2n}}$ pýtali.

(\Leftrightarrow) Predpokladajme, že $f(x) \in W_x^B$ („definičný obor“ programu s indexom x a orákulom B). Pri generovaní prvkov množiny W_x^B sa $f(x)$ objaví po istom konečnom počte krokov m_1 . Keď $f(x)$ patrí do definičného oboru daného programu, tak sa daný program na vstup $f(x)$ dopočíta k výsledku, a preto existuje len konečný počet prvkov $y \in B$, na ktoré sme sa v priebehu výpočtu orákula B pýtali (a na ktoré bola odpoveď áno). Existuje teda aj konečná množina B_{2m_2} , ktorá už všetky tieto y obsahuje (a neobsahuje prvky, na ktoré sa odpovedá nie). Aj pri generovaní prvkov množiny $W_x^{B_{2m_2}}$ sa $f(x)$ objaví za m_1 krokov. (Bude to platiť i keď orákulum B_{2m_2} nahradíme ľubovoľnou jej nadmnožinou z našej postupnosti $B_0, B_1, B_2 \dots$ ktorej zjednotenie tvorí B .) Vezmime $m = \max\{m_1, m_2\}$. Pre každé n také, že $n \geq m$ platí, že $f(x)$ sa objaví za maximálne n krokov vo výpočte $W_x^{B_{2n-1}}$. Ukážme teraz, že existuje krok výpočtu, v ktorom $f(x)$ dostane znamienko plus. Nech $2l+1$ je krok taký, že $l > m$ a všetky značky $\boxed{i}_A, i \leq x$, dosiahli svoju konečnú pozíciu. Aby $f(x)$ mohlo dostať plus, musí spĺňať, že je *neobsadené* (nemá žiadne plus) a vyskytne sa v čiastočnom výpočte $W_x^{\text{orákulum}}$ v každom kroku po kroku $2l+1$ ($l > m$). Nech prvé kritérium spĺňa, pretože kedy $f(x)$ už malo plus, dôkaz je hotový. To, že druhé kritérium spĺňa, sme pred chvíľou ukázali. Hoci $f(x)$ spĺňa obe kritériá, môže sa stať, že plus nedostane, keď existuje značka s väčšou prioritou (vyššie v zozname), ktorá obe kritériá tiež spĺňa. Takýchto značiek môže byť maximálne x . Keďže každá z nich už dosiahla svoju konečnú pozíciu, táto situácia sa môže zopakovať najviac x -krát. Potom $f(x)$ dostane plus, a to najneskôr v kroku $2(l+x+1)+1$. \square

Dostali sme

$$\forall x (f(x) \text{ dostane znamienko plus} \Leftrightarrow f(x) \in W_x^B),$$

čo znamená

$$\forall x (f(x) \in A \Leftrightarrow f(x) \in W_x^B).$$

Teda $A \not\leq_T B$ podľa Lemmy 2. Podobne by sme dostali aj

$$\forall x (g(x) \in B \Leftrightarrow g(x) \in W_x^A).$$

Teda $B \not\leq_T A$ opäť podľa Lemmy 2.

¹Nech $m_1 < m_2$. Keď sa $f(x)$ vo výpočte $W_x^{B_{2m_2}}$ objaví za m_1 krokov, tak aj za ľubovoľne viac. Preto môžeme m_1 zväčšovať až dosiahneme m_2 . Potom môžeme zväčšovať oba spolu, lebo každý následník množiny B_{2m_2} ako orákulum už zaručí, že $f(x)$ sa objaví do m_1 krokov.

Nech $m_2 < m_1$. Podobne ako v predchádzajúcom prípade budeme zväčšovať m_2 a tým vytvárať následníkov množiny B_{2m_2} , ktorí použítí ako orákulum zaistia, že $f(x)$ sa objaví do m_1 krokov. Keď sa objaví do m_1 krokov, tak to môžeme ohraničiť ľubovoľným väčším číslom.

3.2 Prioritná metóda

Metóda použitá v dôkaze Friedberg-Muchnikovho teorému sa nazýva prioritná metóda a usporiadanie značiek $\boxed{0}_A, \boxed{0}_B, \boxed{1}_A, \boxed{1}_B, \boxed{2}_A, \boxed{2}_B \dots$ je prioritné usporiadanie. Dôkaz zakladajúci sa na tejto metóde nazývame prioritný argument. Vo svojej najjednoduchšej podobe (without injuries) ju v roku 1944 predstavil Post pri konštrukcii hyperprostej množiny. Jeho konštrukcia spočívala v tom, že si vytvoril zoznam požiadaviek. Požiadavky sa navzájom nemohli „rušiť“ (preto názov without injuries) a prioritné usporiadanie slúžilo len na to, aby sa rozhodlo, ktorú požiadavku splniť v kroku, v ktorom existuje viac ako jedna splniteľná požiadavka.

Friedberg s Muchnikom použili pri riešení Postovho problému konečnú prioritnú metódu (the finite injury priority method). Tu je povolené, aby sa požiadavky navzájom „rušili“. Prebieha to tak, že keď nejaká požiadavka určí, že istý prvok nemá náležať do konštruovanej množiny, môže sa nájsť požiadavka s väčšou prioritou, ktorá náležanie vyžaduje a prvok je do množiny prijatý. V dôkaze Friedberg-Muchnikovho teorému sú požiadavky reprezentované značkami. Všimnime si dve kľúčové vlastnosti tejto metódy na príklade dôkazu Friedberg-Muchnikovho teorému:

- každá značka je posunutá len konečnekrát,
- značky v každom kroku sú posúvané tak, aby neovplyvnili znamienka mínus získané v tomto kroku.

Práve tieto dve skutočnosti nám umožňujú definovať funkcie f, g s požadovanou vlastnosťou. Ak nejaké číslo c v A -zozname dostane v istom kroku $2i + 1$ znamienko plus, musí v danom kroku spĺňať tieto kritériá: je pri ňom nejaká značka \boxed{z}_A , je najmenšie *neobsadené* také, že sa vyskytlo v i krokoch čiastočného výpočtu množiny $W_z^{B_{2i}}$. To, že dostalo plus znamená, že bolo prijaté medzi prvky množiny A . Ak by sme definovali $f(z) = c$, chceli by sme, aby c patrilo aj do W_z^B .

Množina B_{2i} je množina tých čísel v B -zozname, ktoré majú na konci kroku $2i$ priradené znamienko plus a súčasne je podmnožinou výsledného B . Môžeme povedať, že B_{2i} bude „rásť“ až vznikne B . Keď c patrí $W_z^{B_{2i}}$, tak program s indexom z a orákulom B_{2i} sa na vstup c doráta k výsledku a počas výpočtu sa môžu vyskytnúť otázky typu „patrí y do B_{2i} ?“. Na niektoré sa odpovie áno, na niektoré nie. Podľa algoritmu opísaného v dôkaze Friedberg-Muchnikovho teorému dostanú všetky čísla, na ktoré sa odpovie nie znamienko mínus. Výpočet programu s indexom z na vstup c by prebiehal vždy rovnako, keby orákulá (hoci aj rôzne) dávali identické odpovede. Keď už vieme, že c patrí do $W_z^{B_{2i}}$ a naša požiadavka na *nenáležanie* istých čísel do B by ostala splnená, tak by sme obdržali „správne“ odpovede (myslí sa rovnaké ako od B_{2i}) aj od orákula B . Preto by sa program s indexom z na vstup c doráтал aj v prípade, že by k nemu bolo pripojené orákulum B (k tomu istému výsledku ako s orákulom B_{2i}). Dostali by sme $c \in W_z^B$.

Problém pre nás nastane, keď sa vyskytne značka s vyššou prioritou, ktorá je pri čísle so znamienkom mínus a v niektorom z budúcich krokov toto mínus zmení na plus. Tým sa naša požiadavka stala nesplnenou (a nespĺniteľnou). Značka \boxed{z}_A , ktorá ju zaviedla sa posunie nadol. To je pre nás výhoda, lebo môžeme definovať nové $f(z)$ (nová pozícia značky \boxed{z}_A).

Dôležité je, že takéto „poškodenie“ požiadavky nastane len konečnekrát. Vieme totiž, že každá značka \boxed{z}_A resp. \boxed{z}_B má svoju konečnú pozíciu k_A resp. k_B . Keď konečná pozícia k_A (k_B) má dostať plus, je zrejmé, že ho musí dostať práve vtedy, keď je pri nej značka \boxed{z}_A (\boxed{z}_B). Mínusy udelené v kroku, keď konečná pozícia dostane plus už nezmiznú: značky s nižšou prioritou sú presunuté až za tieto mínusy, značky s vyššou prioritou ich tiež neprepíšu, lebo by to spôsobilo, že \boxed{z}_A (\boxed{z}_B) by sa posunula zo svojej finálnej pozície. Preto definujeme funkcie f , g tak, aby ich obory hodnôt boli konečné pozície značiek. (Oborom hodnôt f konečné pozície v A -zozname, oborom hodnôt g konečné pozície v B -zozname.)

Existujú aj nekonečné modifikácie prioritnej metódy (infinite injury priority methods), v ktorých sa môžu požiadavky rušiť nekonečnekrát a ktoré sú veľmi rôznorodé: napr. tzv. pinball machine model, priority tree model, $0'''$ prioritná metóda .. Nekonečnú prioritnú metódu ako takú prvý raz predstavil Joseph R. Shoenfield vo svojom článku [Sho61], v ktorom dokázal, že existuje axiomatizovateľná, konzistentná teória stupňa menej ako $0'$, v ktorej je reprezentovateľná každá rekurzívna funkcia.

Je prirodzené sa pýtať, či Postov problém môže byť vyriešený aj bez prioritnej metódy. Český matematik Antonín Kučera takéto riešenie našiel a uverejnil ho v článku [Kuč86].

4 Prioritná metóda a Σ_3 -kompletnosť

V ďalšom texte budeme pracovať s m -prevoditeľnosťou.

Definícia Množina A je m -prevoditeľná na B , ak existuje obecné rekurzívna funkcia f pre ktorú platí: $\forall x(x \in A \Leftrightarrow f(x) \in B)$. Píšeme $A \leq_m B$.

Nasledujúce tvrdenie je zjednodušenou formou Lemmy 1:

Lemma 5 Množina A je rekurzívne spočetná a súčasne je komplementom rekurzívne spočetnej množiny \Leftrightarrow množina A je rekurzívna.

Dôkaz (\Rightarrow) Nech A je rekurzívne spočetná a nech existuje B rekurzívne spočetná taká, že $A = \overline{B}$. Potom aj $\overline{A} = B$. Keďže sa rovnajú, tak aj \overline{A} je rekurzívne spočetná. Keď obe, A aj \overline{A} sú rekurzívne spočetné, tak podľa Postovej vety je A rekurzívna.

(\Leftarrow) Ak A je rekurzívna, tak i jej komplement \overline{A} je rekurzívny. Potom sú obe aj rekurzívne spočetné.

Definícia Povieme, že množina A je Σ_3 -kompletná práve vtedy, keď platí:

- $A \in \Sigma_3$,
- $\forall B(B \in \Sigma_3 \Rightarrow B \leq_m A)$.

Chceme ukázať, že množina $\{x|W_x \text{ je rekurzívna}\}$ je Σ_3 -kompletná. Najprv pomocou Tarského-Kuratowského algoritmu ukážeme, že v aritmetickej hierarchii naozaj patrí do Σ_3 . Podobne ukážeme aj o množine $\{x|\exists y(y \in W_x \& W_y \text{ je nekonečná})\}$, ktorú budeme v dôkaze využívať, že je tiež Σ_3 .

Lemma 6 $\{x|\exists y(y \in W_x \& W_y \text{ je nekonečná})\} \in \Sigma_3$.

Dôkaz $z \in \{x|\exists y(y \in W_x \& W_y \text{ je nekonečná})\} \Leftrightarrow$
 $\Leftrightarrow \exists y(y \in W_z \& W_y \text{ je nekonečná}) \Leftrightarrow$
 $\Leftrightarrow \exists y(\exists w_1 T(z, y, w_1) \& \forall u \exists v(u < v \& v \in W_y)) \Leftrightarrow$
 $\Leftrightarrow \exists y(\exists w_1 T(z, y, w_1) \& \forall u \exists v(u < v \& \exists w_2 T(y, v, w_2))) \Leftrightarrow$
 $\Leftrightarrow \exists y(\exists w_1 T(z, y, w_1) \& \forall u \exists v \exists w_2(u < v \& T(y, v, w_2))) \Leftrightarrow$
 $\Leftrightarrow \exists y \exists w_1 (T(z, y, w_1) \& \forall u \exists v \exists w_2(u < v \& T(y, v, w_2))) \Leftrightarrow$
 $\Leftrightarrow \exists y \exists w_1 \forall u \exists v \exists w_2 (T(z, y, w_1) \& (u < v \& T(y, v, w_2)))$

Teraz máme podmienku v prenexnom normálnom tvare. Budeme sa zaujímať len o kvantifikátory. Vieme, že množiny Σ_n sú uzavreté na existenčnú kvantifikáciu a množiny Π_n sú uzavreté na všeobecnú kvantifikáciu. V zátvorke máme primitívne rekurzívnu podmienku. Kvantifikátory $\exists v \exists w_2$ z nej spravia rekurzívne spočetnú, kvantifikátor $\forall u$ spraví Π_2 podmienku a napokon po zohľadnení $\exists y \exists w_1$ dostávame Σ_3 podmienku. \square

Lemma 7 $\{x|W_x \text{ je rekurzívna}\} \in \Sigma_3$.

Dôkaz $z \in \{x|W_x \text{ je rekurzívna}\} \Leftrightarrow$
 $\Leftrightarrow W_z \text{ je rekurzívna} \Leftrightarrow$
 $\Leftrightarrow \exists y(W_z = \overline{W}_y) \Leftrightarrow$ Lemma 5
 $\Leftrightarrow \exists y \forall x(x \in W_z \Leftrightarrow x \notin W_y) \Leftrightarrow$
 $\Leftrightarrow \exists y \forall x(\exists w_1 T(z, x, w_1) \Leftrightarrow \forall w_2 \neg T(y, x, w_2)) \Leftrightarrow$
 $\Leftrightarrow \exists y \forall x[(\exists w_1 T(z, x, w_1) \Rightarrow \forall w_2 \neg T(y, x, w_2)) \& (\forall v_1 \neg T(y, x, v_1) \Rightarrow \exists v_2 T(z, x, v_2))] \Leftrightarrow$
 $\Leftrightarrow \exists y \forall x[\forall w_1 (T(z, x, w_1) \Rightarrow \forall w_2 \neg T(y, x, w_2)) \& \exists v_1 (\neg T(y, x, v_1) \Rightarrow \exists v_2 T(z, x, v_2))] \Leftrightarrow$
 $\Leftrightarrow \exists y \forall x[\forall w_1 \forall w_2 (T(z, x, w_1) \Rightarrow \neg T(y, x, w_2)) \& \exists v_1 \exists v_2 (\neg T(y, x, v_1) \Rightarrow T(z, x, v_2))] \Leftrightarrow$
 $\Leftrightarrow \exists y \forall x \forall w_1 \forall w_2 [\exists v_1 \exists v_2 [(T(z, x, w_1) \Rightarrow \neg T(y, x, w_2)) \& (\neg T(y, x, v_1) \Rightarrow T(z, x, v_2))]]$

Teraz máme podmienku v prenexnom normálnom tvare. Podobnou úvahou ako pri Lemme 6 by sme dospeli k záveru, že je Σ_3 . \square

Lemma 8 Množina $\{x|\exists y(y \in W_x \& W_y \text{ je nekonečná})\}$ je Σ_3 -kompletná.

Dôkaz Nech $B = \{x|\exists y(y \in W_x \& W_y \text{ je nekonečná})\}$.

Vieme už, že B je Σ_3 . Stačí ukázať, že ľubovoľná Σ_3 množina, teda množine tvaru $\{z|\exists x_1\forall x_2\exists x_3R(z, x_1, x_2, x_3)\}$, kde R je rekurzívna podmienka, je m-prevoditeľná na B . Zjednodušíme to ešte trochu a kvantifikátor $\exists x_3$ s podmienkou R spojme do rekurzívne spočítanej podmienky Q . Chceme teda ukázať, že ľubovoľnú množinu tvaru $\{z|\exists x_1\forall x_2Q(z, x_1, x_2)\}$ prevedieme na B .

Nech z, x_1 sú dané. Definujme funkciu $\varphi_{g(z, x_1)}$ nasledovne:

$$! \varphi_{g(z, x_1)}(y) \Leftrightarrow \forall v \leq y Q(z, x_1, v)$$

Pýtame sa, či takáto rekurzívna funkcia g existuje. Predtým, než sa z, x_1 stali konštantami, sme mali funkciu ψ s troma premennými:

$$! \psi(z, x_1, y) \Leftrightarrow \forall v \leq y Q(z, x_1, v)$$

Funkcia ψ je čiastočne rekurzívna, lebo na druhej strane ekvivalencie je rekurzívne spočítaná podmienka, a preto podľa vety o parametroch rekurzívna g s danou vlastnosťou existuje.

Platí, že $W_{g(z, x_1)}$ je nekonečná práve vtedy, keď $\forall x_2Q(z, x_1, x_2)$. Definujme teraz množinu $W_{h(z)} = \{y|\exists x_1 g(z, x_1) = y\}$. Opäť pomocou vety o parametroch ukážeme, že takáto rekurzívna funkcia h existuje. Predtým, než sa z stalo konštantou sme mali množinu P takúto:

$$\begin{aligned} W_{h(z)} &= \{y|P(z, y)\}; \\ P &= \{[z, y]|\exists x_1 g(z, x_1) = y\}. \end{aligned}$$

P je rekurzívne spočítaná množina, teda hľadaná rekurzívna funkcia h existuje. Všimnime si, že $W_{h(z)}$ je množina indexov, ktoré sú oborom hodnôt funkcie g (pre pevné z). Platí:

$$\begin{aligned} z \in \{z|\exists x_1\forall x_2Q(z, x_1, x_2)\} &\Leftrightarrow \\ \Leftrightarrow \exists x_1\forall x_2Q(z, x_1, x_2) &\Leftrightarrow \\ \Leftrightarrow \exists x_1(W_{g(z, x_1)} \text{ je nekonečná}) &\Leftrightarrow \\ \Leftrightarrow \exists y(y \in W_{h(z)} \& W_y \text{ je nekonečná}) &\Leftrightarrow \\ \Leftrightarrow h(z) \in B. & \end{aligned}$$

To znamená, že sme ľubovoľnú Σ_3 množinu previedli na $\{x|\exists y(y \in W_x \& W_y \text{ je nekonečná})\}$ a teda táto je Σ_3 -kompletná. \square

Veta Množina $\{x | W_x \text{ je rekurzívna}\}$ je Σ_3 -kompletná.

Dôkaz Nech $C = \{x | W_x \text{ je rekurzívna}\}$.

Nech $B = \{x | \exists y (y \in W_x \ \& \ W_y \text{ je nekonečná})\}$. Aby sme dokázali, že množina C je Σ_3 -kompletná, potrebujeme ukázať, že platí:

- $C \in \Sigma_3$,
- $\forall D (D \in \Sigma_3 \Rightarrow D \leq_m C)$.

Podľa Lemmy 7 vieme, že C je v aritmetickej hierarchii skutočne Σ_3 . Lemma 8 nám zas hovorí, že B je Σ_3 -kompletná. Keďže relácia \leq_m je tranzitívna, stačí nám na demonštrovanie druhej vlastnosti ukázať, že $B \leq_m C$.

Na to opäť využijeme prioritnú metódu. Zvoľme pevné z . Vytvoríme si vertikálny zoznam všetkých prirodzených čísel a nazývajme ho hlavný zoznam (v tomto prípade nám stačí jediný). Číslam v hlavnom zozname budeme priraďovať značky $\boxed{0}$, $\boxed{1}$, $\boxed{2}$.., niekedy k číslu napíšeme znamienko plus, niekedy posunieme značku a niekedy oboje. (Znamienka mínus nepotrebujeme.) Čísla so znamienkom plus budú tvoriť množinu A . Ukážeme, že naša konštrukcia spĺňa tieto podmienky:

- $z \in B \Rightarrow A$ je nekonečná a \bar{A} je konečná (teda A je aj rekurzívna),
- $z \notin B \Rightarrow A$ nie je rekurzívna.

Ďalej ukážeme, že existuje funkcia h rekurzívna taká, že $A = W_{h(z)}$, z čoho dostaneme:

- $z \in B \Rightarrow W_{h(z)}$ je rekurzívna $\Rightarrow h(z) \in C$,
- $z \notin B \Rightarrow W_{h(z)}$ nie je rekurzívna $\Rightarrow h(z) \notin C$,

a teda $B \leq_m C$.

Povieme, že číslo v hlavnom zozname je *voľné*, ak pri ňom nie je značka a ani nikdy nebola. Číslo v hlavnom zozname je *neobsadené*, ak nemá znamienko plus. Opíšme teraz inštrukcie algoritmu. Algoritmus sa skladá z dvoch častí:

- pomocná procedúra,
- hlavná procedúra.

Pomocná procedúra:

Krok k:

- Urob k krokov vo výpočte W_z (z , ktoré sme si na začiatku pevne zvolili). Zober všetky $y \in W_z$, ktoré sa v rámci týchto k krokov objavili. Pre každé takéto y urob k krokov vo výpočte W_y .

Hlavná procedúra:

Začíname nultým krokom.

Krok 2n ($n \geq 0$):

- Priradiť značku \boxed{n} najmenšiemu *voľnému* číslu v zozname. Nech $x_0^{(n)}, \dots, x_n^{(n)}$ sú aktuálne pozície značiek $\boxed{0}, \dots, \boxed{n}$. Urob n krokov vo výpočte W_0, \dots, W_n .
- Pre každé i také, že $0 \leq i \leq n$ urob: ak $x_i^{(n)}$ je *neobsadené* (bez plus) a $x_i^{(n)}$ sa vyskytlo v čiastočnom výpočte W_i , ktorý sme pred chvíľou urobili, tak číslu $x_i^{(n)}$ priradiť znamienko plus.

Krok 2n + 1 ($n \geq 0$):

- Urob prvých n krokov pomocnej procedúry: krok 1, krok 2, .. krok n .
- Zisti, či existuje y také, že $y \leq n$, objaví sa vo W_z v ľubovoľnom z týchto n krokov a existuje také u , že $u \in W_y$ a prvý raz sa objaví v kroku n pomocnej procedúry (tj. v $n - 1$. kroku sa ešte nevygeneruje ako prvok W_y a v kroku n už áno).
- Ak sa nenájde také y , choď na krok $2n + 2$ a ostatné inštrukcie z tohto kroku ignoruj. Ak sa nájde, vezmi najmenšie z týchto y a pomenuj ho y_n . Pre každé i také, že $y_n \leq i \leq n$, daj plus k číslu $x_i^{(n)}$ (pozícia značky \boxed{i}) ak je *neobsadené* (ešte nemá plus) a posuň značky $\boxed{y_n}, \boxed{y_n + 1}, \dots, \boxed{n}$ nadol k $n - y_n + 1$ najmenším *voľným* číslam v hlavnom zozname.

Môžeme vidieť tieto zákonitosti v posúvaní značiek a priradovaní znamienok:

- Značky sa posúvajú jedine vtedy, keď čísla pri nich dostanú plus resp. plus už majú. Neexistuje teda číslo, pri ktorom by kedysi bola bývala značka, pohla by sa a ono by ostalo bez plus. Z toho vyplýva, že si nepotrebujeme špeciálne značiť čísla, pri ktorých bola značka (buď tam ešte je, alebo má dané číslo plus).

- Číslo dostane plus, len ak (okrem iného) spĺňa, že je pri ňom značka.
- Existujú čísla, ktoré majú aj plus aj značku. (Podľa párnych krokov hlavnej procedúry.)
- Znamienko plus nemôže byť zmazané.
- Prioritné poradie značiek $\boxed{0}$, $\boxed{1}$, $\boxed{2}$.. ostane vždy zachované.

Takto získame množinu A . Množina $A = \{x_1, x_2, \dots\}$ je rekurzívne spočetná, pretože je generovaná procedúrou. Procedúra používa pevne dané z ako parameter. Keby sme umožnili, aby sa z menilo, dostali by sme rekurzívne spočetné množiny $Q_1 = \{[z_1, x_1], [z_1, x_2] \dots\}$, $Q_2 = \{[z_2, x_1], [z_2, x_2] \dots\}$, $Q_3 = \{[z_3, x_1], [z_3, x_2] \dots\}$, .. a ich zjednotenie $Q = Q_1 \cup Q_2 \cup Q_3 \dots$ by bola tiež rekurzívne spočetná množina. (Vzali by sme ľubovoľnú dvojicu $[z_i, x_j]$ a pýtali by sme sa, či patrí do Q . Číslo z_i by sme si zvolili za „pevné“ a začali by sme generovať množinu Q_i podľa krokov algoritmu.) Nech $A = W_{h(z)} = \{x | Q(z, x)\}$. Keďže Q je rekurzívne spočetná, tak podľa vety o parametroch existuje rekurzívna funkcia h , ktorá na základane z vyráta index množiny A .

Ak $z \notin B$, tak podľa definície množiny B platí:

$$\forall y (y \notin W_z \vee W_y \text{ je konečná}).$$

Môžeme to zapísať pomocou implikácie:

$$\forall y (y \in W_z \Rightarrow W_y \text{ je konečná}).$$

To znamená, že ľubovoľné y , ktoré sa v pomocnej procedúre vygeneruje ako prvok W_z , je indexom konečnej množiny. Preto v nepárnych krokoch hlavnej procedúry pre dostatočne veľké n nebudú existovať y patriace W_z také, že po n krokoch výpočtu W_y by sa objavil nový prvok $u \in W_y$, o ktorom sme po $n - 1$ krokoch ešte nevedeli. Jednoducho povedané, keď W_y sú konečné, tak sa ich prvky raz „minú“. To ale spôsobí, že žiadna značka sa už nikdy nepohne. Preto má každá značka v prípade, že $z \notin B$ konečnú pozíciu. Nech x_n je konečná pozícia značky \boxed{n} . Potom platí:

$$x_n \in A \Leftrightarrow x_n \in W_n.$$

Implikácia (\Rightarrow) je zrejmá už z inštrukcií algoritmu (párny krok hlavnej procedúry). Ak x_n dostane plus, tak jedine vtedy, ak spĺňa aj podmienku, že patrí do W_n . V nepárnom kroku plus získať nemôže, pretože značka \boxed{n} by sa pohla zo svojej finálnej pozície.

Ukážme si opačnú implikáciu. Číslo x_n je aktuálnou pozíciou značky \boxed{n} a aj ňou ostane, lebo \boxed{n} je tu na finálnej pozícii. Nech x_n ešte nemá plus. Keď x_n patrí do W_n , tak po konečne mnoho krokoch, nech je to po i krokoch, ho vygenerujeme. Preto v kroku $2i$ spĺňa x_n všetky kritériá na získanie plus a aj ho dostane. Máme:

$$x_n \in A \Leftrightarrow x_n \in W_n,$$

to znamená

$$x_n \in \bar{A} \Leftrightarrow x_n \notin W_n.$$

Pre každé n existuje x_n , že

$$((x_n \in \bar{A} \& x_n \notin W_n) \vee (x_n \notin \bar{A} \& x_n \in W_n)).$$

Preto platí:

$$\forall n(\bar{A} \neq W_n),$$

čo znamená, že \bar{A} nie je rekurzívne spočetná množina. Z toho podľa Postovej vety dostávame, že A nie je rekurzívna. Celkovo teda máme:

$$z \notin B \Rightarrow A = W_{h(z)} \text{ nie je rekurzívna} \Rightarrow h(z) \notin C.$$

Ak z patrí B , tak podľa definície množiny B platí:

$$\exists y(y \in W_z \& W_y \text{ je nekonečná}).$$

Veźmime také y patriace W_z , že W_y je nekonečná. Keďže $y \in W_z$, tak existuje krok k pomocnej procedúry, v ktorom toto y vygenerujeme. Vo všetkých nepárnych krokoch $2n+1$, $n \geq k$ hlavnej procedúry sa nám toto y objaví a so zvyšujúcim sa počtom krokov sa vždy nájde nový prvok patriaci W_y . Zaujímajú nás takéto $y \leq n$ pre daný krok $2n+1$, ale n sa zvyšuje, takže po istom počte krokov to bude platiť. Vďaka nekonečnosti W_y teda máme zaručené, že pre ľubovoľný krok existuje krok budúci (nemusí byť bezprostredne budúci), v ktorom vygenerujeme taký prvok u patriaci do W_y , o ktorom náležaní do W_y sme zatiaľ nevedeli. Vygenerovanie takéhoto u spôsobí, že všetky značky s menšou prioritou ako značka označená číslom y (vrátane) dajú svojim pozíciám znamienko plus a posunú sa nadol k väčším číslam, ktoré ešte plus nemajú. Takéto u sa vždy nájde, preto značky $\boxed{y}, \dots, \boxed{n}$ vždy dajú svojim pozíciám plus a vždy sa posunú. Z toho dostávame, že nekonečne veľa čísel nadol od počiatočnej pozície značky \boxed{y} dostane plus. Preto je komplement A konečný. Dostali sme teda:

$$z \in B \Rightarrow \bar{A} \text{ je konečná} \Rightarrow A = W_{h(z)} \text{ je rekurzívna} \Rightarrow h(z) \in C. \quad \square$$

Zároveň bola dokázaná ešte jedna dôležitá veta: tiež množina $D = \{x \mid \overline{W}_x \text{ je konečná}\}$ je Σ_3 -kompletná:

$$\begin{aligned} z \notin B \Rightarrow A = W_{h(z)} \text{ nie je rekurzívna} &\Rightarrow \overline{W}_{h(z)} \text{ je nekonečná} \Rightarrow h(z) \notin D, \\ z \in B \Rightarrow \overline{W}_{h(z)} \text{ je konečná} &\Rightarrow h(z) \in D. \end{aligned}$$

Literatúra

- [Kuč86] Antonín Kučera. An alternative, priority-free, solution to Post's problem. In J. Gruska, B. Rován, and J. Wiedermann, editors, *Mathematical Foundations of Computer Science 1986, Bratislava, Czechoslovakia, August 25–29, 1986*, volume 233 of *Lecture Notes in Computer Science*, pages 493–500. Springer, 1986.
- [Odi89] Piergiorgio Odifreddi. *Classical Recursion Theory*. North-Holland, Amsterdam, 1989.
- [Pos43] Emil Leon Post. Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*, 65:197–215, 1943.
- [Pos44] Emil Leon Post. Recursively enumerable sets of positive integers and their decision problems. *Bulletin of the American Mathematical Society*, 50:284–316, 1944.
- [Rog67] Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, New York, 1967.
- [Sho61] Joseph R. Shoenfield. Undecidable and creative theories. *Fundamenta Mathematicae*, 49:171–179, 1961.

Internetové zdroje:

http://www.cis.upenn.edu/~tanmoy/post_ens.pdf
<http://cs.nyu.edu/pipermail/fom/1999-January/002529.html>
http://www.cs.umb.edu/~fejer/articles/History_of_Degrees.pdf
www.math.ucla.edu/~asl/bsl/0703/0703-007.ps
<http://cs.nyu.edu/pipermail/fom/1999-September/003365.html>