KAPITOLA 1

Niektoré pojmy z teórie množín a matematickej logiky

V tejto kapitole zavedieme niektoré pojmy a označenia z teórie množín a matematickej logiky, ktoré budeme v ďalšom potrebovať. Väčšinou ich preberieme v podstate z [5], ale aby sme predišli prípadným nedorozumeniam, zopakujme (prípadne pre naše účely prispôsobíme) ich definície. Súčasne zavedieme niektoré označenia. Budeme vychádzať z axiomatického systému teórie množín z [5], i keď by pre naše účely stačili aj niektoré slabšie systémy.

Označenie 1.1. Prázdnu množinu budeme označovať symbolom $\emptyset$. Znakom $\in$ budeme označovať vzťah „patriť do“. Znakmi $\subseteq$ a $\subset$ budeme označovať množinovú inklúziu, resp. ostrú množinovú inklúziu.

Označenie 1.2. Znakom $\{x_1, \ldots, x_n\}$ budeme označovať množinu pozostávajúcu z prvkov x_1, $x_2, \ldots, x_n$. Symbol $\{x \mid \varphi(x)\}$ bude znamenať množinu všetkých takých x, pre ktoré platí $\varphi(x)$. Symbol $\{x \in A \mid \varphi(x)\}$ bude znamenať množinu všetkých $x \in A$, pre ktoré platí $\varphi(x)$.

Poznámka 1.3. Označenie $\{x_1, \ldots, x_n\}$ budeme používať aj pre $n = 0$; vtedy bude znamenať $\{x_1, \ldots, x_n\}$ prázdnu množinu.

Označenie 1.4. Množinu všetkých prirodzených čísel, včítane 0, budeme označovať $\mathbb{N}$. Prirodzené čísla i základné operácie s nimi budeme označovať obvyklým spôsobom.

Poznámka 1.5. Pre určitosť by sme mohli predpokladať napríklad, že prirodzené číslo n je množina všetkých prirodzených čísel menších ako n.

Označenie 1.6. Symbolom $\langle a, b\rangle$ budeme označovať množinu $\{\{a\}, \{a, b\}\}$; symbolom $X \otimes Y$ budeme označovať množinu všetkých takých $\langle a, b\rangle$, o ktorých platí $a \in X,\ b \in Y$.

Poznámka 1.7. Názvy „usporiadaná dvojica“ a „kartézsky súčin“ si rezervujeme pre trochu odlišné pojmy. Odstraňujeme tak istú dvojznačnosť z [5].

Definícia 1.8. *Zobrazenie* množiny X do množiny Y je taká množina $Z \subseteq X \otimes Y$, že pre každé $a \in X$ existuje práve jedno také $b \in Y$, že $\langle a, b\rangle \in Z$. *Čiastočné zobrazenie* množiny X do množiny Y je taká množina $Z \subseteq X \otimes Y$, že pre každé $a \in X$ existuje najviac jedno také $b \in Y$, že $\langle a, b\rangle \in Z$.

Definícia 1.9. *Usporiadanou n-ticou* prvkov $x_1, \ldots, x_n$ nazveme také zobrazenie f množiny $\{0, 1, \ldots, n-1\}$ do množiny $\{x_1, \ldots, x_n\}$, že pre všetky $i = 0, 1, \ldots, n-1$ platí $f(i) = x_{i+1}$. Usporiadanú n-ticu prvkov $x_1, \ldots, x_n$ budeme označovať $(x_1, \ldots, x_n)$.

Poznámka 1.10. Usporiadané n-tice majú nasledujúcu vlastnosť: Ak $(x_1, \ldots, x_m) = (y_1, \ldots, y_n)$, tak $m = n$ a $x_1 = y_1, \ldots, x_m = y_n$. Pre $n = 0$ platí $(x_1, \ldots, x_n) = \emptyset$.

Definícia 1.11. *Kartézsky súčin* n množín $X_1, \ldots, X_n$ je množina $X_1 \times \cdots \times X_n$ všetkých takých usporiadaných n-tíc $(x_1, \ldots, x_n)$, že $x_1 \in X_1, \ldots, x_n \in X_n$.

Poznámka 1.12. Všeobecne povedané neplatí $X_1 \times X_2 = X_1 \otimes X_2$ ani $X_1 \times \cdots \times X_n = X_1$ pre $n = 1$. Pre $n = 0$ platí $X_1 \times \cdots \times X_n = \{\emptyset\}$. (Naše označenie kartézskeho súčinu je síce dosť nevhodné pre $n = 0$ a $n = 1$, ale pretože sa tieto prípady nebudú samostatne vyskytovať, netreba zavádzať osobitné označenie.)

Označenie 1.13. Symbolom X^n budeme označovať kartézsky súčin n množín $X, \ldots, X$ (n-krát).

Poznámka 1.14. Platí $X^0 = \{\emptyset\}$. Pozor, pokiaľ je množina X neprázdna, $X^1 \neq X$.

Definícia 1.15. Ľubovoľnú časť množiny X^n nazveme *n-árnou reláciou* na množine X.

Definícia 1.16. (a) *n-árnou čiastočnou funkciou* na množine X nazveme ľubovoľné čiastočné zobrazenie množiny X^n do množiny X.

(b) *n-árnou funkciou* na množine X nazveme ľubovoľné zobrazenie množiny X^n do množiny X.

Poznámka 1.27. Samotný pojem „výraz“ nebudeme definovať a budeme ho pokladať za intuitívne jasný. Keby sme však chceli budovať formalizovanú teóriu, museli by sme túto medzeru v našom výklade odstrániť. Aj pojmy z matematickej logiky by sme museli vybudovať presnejšie a úplnejšie, ako to tu urobíme.

Poznámka 1.28. V 1.26 sa nežiada, aby vo výraze $F(x_1, \ldots, x_n)$ všetky premenné skutočne vystupovali, ani to, aby v tomto výraze nevystupovali iné premenné. Uvidíme to v nasledujúcom príklade.

Príklad 1.29. Nech $X = \mathbb{N}$. Uvažujme výraz $F(x, y, z) = x + y + z$. Výraz $\lambda_{\mathbb{N}} xyz(x + y + z)$ znamená ternárnu funkciu na množine $\mathbb{N}$. Výraz $\lambda_{\mathbb{N}} xz(x + y + z)$ znamená binárnu funkciu dvoch premenných x, z. V tomto prípade je y parameter a dosadzovaním rôznych prirodzených čísel namiesto y dostávame rôzne funkcie; napríklad pre $y = 0$ znamená výraz $\lambda_{\mathbb{N}} xz(x + y + z)$ binárnu funkciu „súčet“, t.j. tú istú funkciu ako aj výraz $\lambda_{\mathbb{N}} xy(x + y)$. Výraz $\lambda_{\mathbb{N}} xyzt(x + y + z)$ znamená funkciu štyroch premenných; pritom štvrtá premenná sa niekedy nazýva fiktívnou premennou, lebo od jej hodnoty nezávisí hodnota funkcie. Výraz $\lambda_{\mathbb{N}}(3)$ znamená 0-árnu funkciu, t.j. konštantu, rovnajúcu sa trom.

Príklad 1.30. Funkcie $\lambda_{\mathbb{N}} xy(x^y)$, $\lambda_{\mathbb{N}} yx(x^y)$ sú rôzne; do grafu prvej z nich patrí napríklad trojica $(2, 3, 8)$, ktorá nepatrí do grafu druhej z nich. Naproti tomu platí $\lambda_{\mathbb{N}} xy(x^y) = \lambda_{\mathbb{N}} yx(y^x)$

Zavedieme teraz informatívne niektoré pojmy a označenia z matematickej logiky. Pôjde viac len o objasnenie daných pojmov ako o ich riadne definície.

Definícia 1.31. *Výrok* je výraz alebo veta, o ktorom má zmysel hovoriť, že je pravdivý alebo že je nepravdivý. *Predikát* (často sa tiež hovorí „výroková funkcia“) na množine X je výraz, ktorý sa stáva výrokom, ak za všetky voľné premenné v ňom dosadíme konkrétne prvky X. (Voľné premenné sú práve tie, za ktoré má zmysel dosadzovať.) Predikát, ktorý má práve n voľných premenných, nazveme n-árnym. Dva predikáty na množine X budeme nazývať *ekvivalentnými,* ak výroky, ktoré z nich vzniknú ľubovoľným rovnakým dosadením prvkov X za ich voľné premenné, sú vždy obidva pravdivé alebo obidva nepravdivé. Ekvivalentné predikáty nebudeme v ďalšom texte obvykle rozlišovať. V zátvorke sú uvedené voľné premenné predikátu, ktorými sú obvykle písmená z konca abecedy, prípadne s indexmi. Výsledok dosadenia prvkov $a_1, \ldots, a_n$ za premenné $x_1, \ldots, x_n$ v predikáte $P(x_1, \ldots, x_n)$ budeme označovať $P(a_1, \ldots, a_n)$. Obdobne budeme označovať výsledok dosadenia za časť premenných.

Nebudeme určovať presné pravidlá na to, aké symboly smú byť použité pre konkrétne predikáty; budeme používať podľa potreby všetky bežné matematické symboly i slovný opis. Uvedieme však niekoľko spôsobov tvorenia nových predikátov, resp. výrokov z daných predikátov.

Definícia 1.32. Znaky $\wedge$, $\vee$, $\Longrightarrow$, $\Longleftrightarrow$, $\neg$ v tomto poradí nazývame *konjukciou, disjunkciou, implikáciou, ekvivalenciou, negáciou* a súhrne *logickými spojkami*. Znaky $\forall$, $\exists$ nazývame *kvantifikátormi*, a to $\forall$ všeobecným (alebo veľkým) a $\exists$ existenčným (alebo malým) kvantifikátorom.

Niekedy nazývame podľa logickej spojky aj predikát, ktorý je pomocou nej vytvorený z dvoch predikátov, v prípade negácie z jedného predikátu. Význam logických spojok a kvantifikátorov určujú nasledujúce definície, ktoré pre jednoduchosť vyslovíme len pre výroky, resp. unárne predikáty.

Definícia 1.33. Pre ľubovoľné výroky A, B:

(a) Výrok $A \wedge B$ je pravdivý práve vtedy, keď sú pravdivé oba výroky A, B.
(b) Výrok $A \vee B$ je pravdivý práve vtedy, keď je pravdivý aspoň jeden z výrokov A, B.
(c) Výrok $A \Longrightarrow B$ je pravdivý práve vtedy, keď je výrok B pravdivý alebo výrok A nepravdivý.
(d) Výrok $A \Longleftrightarrow B$ je pravdivý práve vtedy, keď sú obidva výroky A, B pravdivé alebo obidva nepravdivé.
(e) Výrok $\neg A$ je pravdivý práve vtedy, keď je výrok A nepravdivý.

Poznámka 1.34. Ak sa v niektorom výroku vyskytuje viac logických spojok, odstraňujeme prípadné nejednoznačnosti pomocou zátvoriek. Podľa definície 1.33 zodpovedajú logické spojky $\wedge$, $\vee$,

KAPITOLA 2

Klony čiastočných funkcií na množine

V celej tejto kapitole $\mathbb{X}$ znamená pevne zvolenú neprázdnu množinu. Pri využívaní výsledkov tejto kapitoly v ďalšom texte budeme za $\mathbb{X}$ voliť obvykle množinu $\mathbb{N}$ všetkých prirodzených čísiel alebo množinu všetkých slov v nejakej abecede (pozri kapitolu 6). Pokiaľ nebude hroziť nedorozumenie, budeme niekedy vynechávať slová „na množine“ $\mathbb{X}$, index $\mathbb{X}$ pri λ-notácii a podobne.

Definícia 2.1. Pre ľubovoľné m, $n \in \mathbb{N}$, $0 < m \leq n$ budeme funkciu $\lambda x_1 \ldots x_n(x_m)$ nazývať *m-tou n-árnou projekciou na množine* $\mathbb{X}$ a označovať I^n_m. Všetky funkcie I^n_m budeme nazývať *projekciami na množine* $\mathbb{X}$.

Poznámka 2.2. Pre funkciu I^n_m teda platí $I^n_m(x_1, \ldots, x_n) = x_m$ pre všetky $x_1, \ldots, x_n \in \mathbb{X}$.

Definícia 2.3. Nech f je n-árna (čiastočná) funkcia ($n > 0$), g, $f_1, \ldots, f_n$ sú m-árne (čiastočné) funkcie. Budeme hovoriť, že (čiastočná) funkcia g *vzniká skladaním* (prípadne: *operáciou skladania (čiastočných) funkcií*) z (čiastočných) funkcií f, $f_1, \ldots, f_n$, a písať $g = \mathcal{S}^{n+1}(f, f_1, \ldots, f_n)$, ak pre všetky $x_1, \ldots, x_m \in \mathbb{X}$ platí

$$g(x_1, \ldots, x_m) = f\left(f_1(x_1, \ldots, x_m), \ldots, f_n(x_1, \ldots, x_m)\right)$$

Cvičenie 2.4. Nech $g = \mathcal{S}^{n+1}(f, f_1, \ldots, f_n)$. Dokážte, že ak sú f, $f_1, \ldots, f_n$ totálne funkcie, tak je aj g totálna funkcia! Vyplýva z totálnosti g totálnosť f ? Vyplýva z totálnosti g totálnosť $f_1, \ldots, f_n$?

Cvičenie 2.5. Dokážte, že množina všetkých projekcií na množine $\mathbb{X}$ je uzavretá vzhľadom na operáciu skladania čiastočných funkcií, t.j. že ak $g = \mathcal{S}^{n+1}(f, f_1, \ldots, f_n)$ a f, $f_1, \ldots, f_n$ sú projekcie, je aj g projekcia! Existujú vlastné podmnožiny všetkých projekcií na $\mathbb{X}$ uzavreté vzhľadom na operáciu skladania funkcií?

Definícia 2.6. Množinu (čiastočných) funkcií na množine $\mathbb{X}$ nazveme *klonom (čiastočných) funkcií na* $\mathbb{X}$, ak obsahuje všetky projekcie na $\mathbb{X}$ a je uzavretá vzhľadom na operáciu skladania (čiastočných) funkcií.

Príklad 2.7. Nech $\mathbb{X}$ je množina všetkých reálnych čísel. Všetky polynómy ľubovoľného počtu premenných s reálnymi koeficientmi (uvažované ako funkcie) tvoria klon funkcií na množine $\mathbb{X}$.

Dokážeme teraz niekoľko tvrdení o klonoch čiastočných funkcií na množine. Vyslovíme ich najprv bez použitia λ-notácie a potom ešte raz pomocou nej.

Veta 2.8. Nech $\mathcal{M}$ je klon čiastočných funkcií na množine $\mathbb{X}$, nech čiastočná funkcia $f(x_1, \ldots, x_m)$, $m \neq 0$, patrí do $\mathcal{M}$ a nech $k_1, \ldots, k_m$ je ľubovoľná konečná postupnosť prvkov množiny $\{1, \ldots, n\}$. Potom aj čiastočná funkcia g, $g(x_1, \ldots, x_n) = f(x_{k_1}, \ldots, x_{k_m})$ patrí do $\mathcal{M}$.

Dôkaz: Platí $g = \mathcal{S}^{m+1}\left(f, I^n_{k_1}, \ldots, I^n_{k_m}\right)$, všetky čiastočné funkcie na pravej strane rovnosti patria do $\mathcal{M}$, klon $\mathcal{M}$ je uzavretý vzhľadom na operáciu skladania čiastočných funkcií, a teda aj g patrí do $\mathcal{M}$. □

Poznámka 2.9. Veta 2.8 nám umožňuje napríklad „stotožňovať argumenty“ a ľubovoľne meniť ich poradie. Ak napríklad $\mathbb{X} = \mathbb{N}$ a $f(x, y) = x^y \in \mathcal{M}$, tak aj $g(x, y) = y^x \in \mathcal{M}$, $h(x) = x^x \in \mathcal{M}$. Umožňuje nám tiež „pridávať fiktívne premenné“. Ak znova $\mathbb{X} = \mathbb{N}$ a $f(x, y) = x^y \in \mathcal{M}$, tak aj $g_1(x, y, z) = x^y$ a $g_2(x, y, z) = z^y$ patria do klonu $\mathcal{M}$.

Cvičenie 2.10. Ako treba voliť postupnosť $k_1, \ldots, k_m$ (a čísla m, n) vo vete 2.8, aby sme dostali tvrdenia uvedené v poznámke 2.9 ?

Cvičenie 2.11. Ak klon $\mathcal{M}$ čiastočných funkcií na $\mathbb{X}$ obsahuje všetky unárne konštantné funkcie, tak obsahuje aj všetky n-árne konštantné funkcie pre každé $n \in \mathbb{N}$, $n \neq 0$. Dokážte! Možno vynechať podmienku $n \neq 0$?

Veta 2.19. Nech klon $\mathcal{M}$ čiastočných funkcií na množine $\mathbb{X}$ obsahuje trojicu číslovacích funkcií C, L, R. Potom pre každé $n \in \mathbb{N}$, $n \neq 0$ existuje v klone $\mathcal{M}$ n-árna funkcia C^n a také unárne funkcie $C_{n,i}$, $i = 1, \ldots, n$, že pre všetky $i = 1, \ldots, n$ a pre všetky $x, x_1, \ldots, x_n \in \mathbb{X}$ platí

$$C^n\left(C_{n,1}\left(x\right), \ldots, C_{n,n}\left(x\right)\right) = x \tag{2.19.1}$$

$$C_{n,i}\left(C^n(x_1, \ldots, x_n)\right) = x_i \tag{2.19.2}$$

Dôkaz: Pre $n = 1$ zrejme stačí zvoliť $C^1 = C_{1,1} = I_1^1$. Ak sme už našli funkcie C^n, $C_{n,1}, \ldots, C_{n,n}$, tak môžeme položiť napríklad

$$C^{n+1}(x_1, \ldots, x_{n+1}) = C\left(C^n(x_1, \ldots, x_n), x_{n+1}\right) \tag{2.19.3}$$

$$C_{n+1,i}(x) = C_{n,i}\left(L(x)\right) \quad \text{pre } i = 1, \ldots, n \tag{2.19.4}$$

$$C_{n+1,n+1}(x) = R(x) \tag{2.19.5}$$

Vzťahy (2.19.1) a (2.19.2) (s $n+1$ miesto n) sa ľahko bezprostredne overia z indukčného predpokladu. □

Cvičenie 2.20. Funkcie C^n, $C_{n,i}$ z vety 2.19 (nie z jej dôkazu) nie sú v netriviálnych prípadoch určené jednoznačne. Uveďte aspoň päť konštrukcií funkcií C^3, $C_{3,1}$, $C_{3,2}$, $C_{3,3}$, ktoré by dávali vo všeobecnosti rôzne štvorice funkcií zo systému $\mathcal{M}$ vyhovujúce podmienkam vety pre $n = 3$! Za akých predpokladov sú funkcie C^n, $C_{n,i}$ vo vete 2.19 určené jednoznačne?

Poznámka 2.21. Kvôli určitosti budeme z trojice číslovaných funkcií (ktoré „číslujú“ usporiadané dvojice prvkov množiny $\mathbb{X}$) vytvárať funkcie C^n, $C_{n,1}, \ldots, C_{n,n}$ (ktoré „číslujú“ usporiané n-tice prvkov množiny $\mathbb{X}$) podľa predpisu z dôkazu vety 2.19, pokiaľ výslovne neuvedieme iný predpis. Pre trojicu číslovacích funkcií budeme obvykle používať označenie C, L, R, aj keď môže ísť v rôznych klonoch o rôzne funkcie.

Poznámka 2.22. Ak klon $\mathcal{M}$ čiastočných funkcií na množine $\mathbb{X}$ obsahuje trojicu číslovacích funkcií C, L, R, tak pre každé $n \in \mathbb{N}$, $n \neq 0$ existuje jednojednoznačné priradenie medzi unárnymi a n-árnymi prvkami klonu, v ktorom n-árnej čiastočnej funkcii $F(x_1, \ldots, x_n)$ zodpovedá unárna čiastočná funkcia $F\left(C_{n,1}\left(x\right), \ldots, C_{n,n}\left(x\right)\right)$, a obrátene, unárnej čiastočnej funkcii $f(x)$ zodpovedá n-árna čiastočná funkcia $f\left(C^n(x_1, \ldots, x_n)\right)$. Toto priradenie nám umožňuje niekedy obmedziť sa na štúdium unárnych prvkov klonu $\mathcal{M}$.

Označenie 2.23. Pre ľubovoľnú množinu (špeciálne klon) čiastočných funkcií $\mathcal{M}$ na množine $\mathbb{X}$ a ľubovoľné $n \in \mathbb{N}$ budeme symbolom $\mathcal{M}^{(n)}$ označovať množinu všetkých n-árnych čiastočných funkcií množiny $\mathcal{M}$.

Nasledujúci pojem bude mať v teórii čiastočne rekurzívnych funkcií (štúdium jej základov je jedna z našich hlavných úloh) mimoriadnu dôležitosť.

Definícia 2.24. Nech $\mathcal{M}$ je množina n-árnych (čiastočných) funkcií na množine $\mathbb{X}$. Nech F je $(n+1)$-árna (čiastočná) funkcia na množine $\mathbb{X}$. Budeme hovoriť, že (čiastočná) funkcia F je *univerzálna (čiastočná) funkcia pre množinu* $\mathcal{M}$, ak

(a) pre každé $a \in \mathbb{X}$ n-árna (čiastočná) funkcia g, $g(x_1, \ldots, x_n) = F(a, x_1, \ldots, x_n)$ patrí do $\mathcal{M}$ a

(b) pre každú (čiastočnú) funkciu $g \in \mathcal{M}$ existuje také $a \in \mathbb{X}$, že pre všetky $x_1, \ldots, x_n \in \mathbb{X}$ platí $g(x_1, \ldots, x_n) = F(a, x_1, \ldots, x_n)$.

Poznámka 2.25. Podmienky (a), (b) z predchádzajúcej definície by sme mohli zapísať aj jedným vzťahom

$$\mathcal{M} = \{\lambda x_1 \ldots x_n F(x_0, x_1, \ldots, x_n) \mid x_0 \in \mathbb{X}\} \tag{2.25.1}$$

Prípad $n = 0$ sme síce v definícii nevylúčili, ale často bude mať odlišné vlastnosti od prípadu $n \neq 0$.

Cvičenie 2.26. Nech $\mathcal{M}$ je klon čiastočných funkcií na množine $\mathbb{X}$, ktorý obsahuje trojicu číslovacích funkcií C, L, R. Nech F je univerzálna čiastočná funkcia pre množinu $\mathcal{M}^{(1)}$ všetkých unárnych prvkov klonu $\mathcal{M}$. Zostrojte z funkcií C, L, R, F a projekcií na množine $\mathbb{X}$ len pomocou operácie skladania funkcií univerzálnu čiastočnú funkciu pre množinu $\mathcal{M}^{(2)}$ a množinu $\mathcal{M}^{(3)}$ všetkých binárnych, respektíve ternárnych prvkov klonu $\mathcal{M}$!

Cvičenie 2.36. Nájdite klon funkcií na nejakej nekonečnej množine $\mathbb{X}$, ktorý obsahuje trojicu číslovacích funkcií a ktorý pritom nie je normálny!

Cvičenie 2.37. Nájdite klon funkcií na nejakej nekonečnej množine $\mathbb{X}$, ktorý obsahuje unárnu funkciu g s vlastnosťou $g(x) \neq x$ pre všetky $x \in \mathbb{X}$ a ktorý pritom nie je normálny!

Definícia 2.38. Nech Y je ľubovoľná množina (čiastočných) funkcií na množine X. Najmenší klon (čiastočných) funkcií na množine X, ktorý obsahuje množinu Y, budeme nazývať klonom *generovaným množinou* Y. Ak $Y = \{f_1, \ldots, f_k\}$, budeme ho tiež nazývať *klonom generovaným (čiastočnými) funkciami* $f_1, \ldots, f_k$.

Ako obvykle, slovo „najmenší“ treba chápať v zmysle množinovej inklúzie.

Cvičenie 2.39. Uvažujme klon funkcií na množine $\mathbb{R}$ všetkých reálnych čísel generovaný sčítaním, t.j. funkciou $f_0 = \lambda_{\mathbb{R}} xy(x + y)$. Rozhodnite, ktoré z nasledujúcich funkcií tento klon obsahuje, a ktoré nie:

$$f_1 = \lambda_{\mathbb{R}} xyz(x + 2y + 3z) \qquad f_2 = \lambda_{\mathbb{R}} xyz(x + 1) \qquad f_3 = \lambda_{\mathbb{R}} xy(x - 2y)$$
$$f_4 = \lambda_{\mathbb{R}} xyz(x + 2z) \qquad f_5 = \lambda_{\mathbb{R}} x(x + 1) \qquad f_6 = \lambda_{\mathbb{R}} xy(x - y)$$

Cvičenie 2.40. Ako sa zmenia odpovede v predchádzajúcom cvičení, ak namiesto klonu budeme uvažovať najmenšiu množinu funkcií obsahujúcu súčet (t.j. f_0) a uzavretú vzhľadom na operáciu skladania funkcií? (Rozdiel: nemáme k dispozícii projekcie.)

Cvičenie 2.41. Riešte analógiu cvičenia 2.39, 2.40 pre funkciu rozdiel (t.j. f_6) namiesto súčtu. (Odpovedajte aj pre funkciu f_0.)

Cvičenie 2.42. Nech f je binárna operácia na nejakej množine X. Vyjadrite komutatívnosť a asociatívnosť f pomocou operácie S skladania funkcií, projekcií a samotného symbolu f.

Cvičenie 2.43. Ukážte, že žiadna z booleovských funkcií $\wedge$, $\vee$, $\Longrightarrow$, $\Longleftrightarrow$, $\neg$ negeneruje klon všetkých booleovských funkcií.

Cvičenie 2.44. Uveďte binárnu booleovskú funkciu, ktorá generuje klon všetkých booleovských funkcií.

Cvičenie 2.45. Nech $\mathcal{M}$ je klon generovaný množinou funkcií X. Ukážte, že ak X neobsahuje žiadne konštanty, tak ani $\mathcal{M}$ neobsahuje žiadne konštanty.

Cvičenie 2.46. Nazvime n-árnu funkciu na množine X *v podstate* k*-árnou*, ak existuje k-árna funkcia g na X a indexy $i_1, \ldots, i_k \in \{1, \ldots, n\}$ také, že pre všetky $x_1, \ldots, x_n \in X$ platí

$$f(x_1, \ldots, x_n) = g(x_{i_1}, \ldots, x_{i_k})$$

Tvoria všetky

- v podstate unárne
- v podstate binárne

funkcie na množine $\mathbb{N}$ klon funkcií?

Cvičenie 2.47. Určte mohutnosť množiny všetkých klonov na množine $\mathbb{N}$. (*Odpoveď:* 2^c, kde c je mohutnosť kontinua. *Návod:* Stačí uvážiť len klony generované množinami nula-jednotkových v podstate unárnych funkcií zachovávajúcich 0 a 1. Aj tých je „dosť“.)

Cvičenie 2.48. Nech klon $\mathcal{M}$ funkcií na $\mathbb{N}$ obsahuje všetky unárne funkcie a funkciu súčet. Dokážte, že potom $\mathcal{M}$ obsahuje všetky n-árne funkcie pre všetky $n > 0$.

Cvičenie 2.49. Nech (C, L, R) je nejaká trojica číslovacích funkcií na množine X a nech $\mathcal{M}$ je klon funkcií na X. Rozhodnite, či platí:

(a) Ak $L, R \in \mathcal{M}$, tak aj $C \in \mathcal{M}$.
(b) Ak $C \in \mathcal{M}$, tak aj $L, R \in \mathcal{M}$.
(c) Ak $C, L \in \mathcal{M}$, tak aj $R \in \mathcal{M}$.

Cvičenie 2.50. Rozhodnite, či platí nasledujúce tvrdenie: Ak klon funkcií na nejakej nekonečnej množine X obsahuje aspoň jednu trojicu číslovacích funkcií, tak obsahuje nekonečne mnoho trojíc číslovacích funkcií.

KAPITOLA 3

Primitívne rekurzívne, rekurzívne a čiastočne rekurzívne funkcie

V tejto kapitole sa budeme zaoberať výlučne čiastočnými funkciami na množine $\mathbb{N}$ všetkých prirodzených čísel; slová „na množine $\mathbb{N}$“ budeme vynechávať. Niekedy budeme namiesto „prirodzené číslo“ hovoriť len „číslo“.

Budeme vychádzať z niekoľkých funkcií, ktoré sú v intuitívnom zmysle zrejme algoritmicky vypočítateľné a z niekoľkých operácií na (čiastočných) funkciách, ktoré — ako čitateľ ľahko zistí — zachovávajú (čiastočnú) algoritmickú vypočítateľnosť (čiastočných) funkcií. Pritom (čiastočnú) funkciu pokladáme za *algoritmicky (čiastočne) vypočítateľnú*, ak vieme pre ľubovoľné dané hodnoty jej argumentov určovať príslušnú hodnotu (čiastočnej) funkcie (za predpokladu, že je táto hodnota definovaná). Všetky (čiastočné) funkcie, ktoré dostaneme z východiskových funkcií týmito operáciami, budú teda algoritmicky (čiastočne) vypočítateľné v intuitívnom zmysle. Špeciálne teda (ako uvidíme bezprostredne z príslušných definícií) všetky (čiastočne) rekurzívne funkcie sú algoritmicky (čiastočne) vypočítateľné. O možnosti a význame obráteného tvrdenia budeme hovoriť neskôr.

Teraz zavedieme primitívne rekurzívne funkcie a dokážeme primitívnu rekurzívnosť niektorých často uvažovaných funkcií.

Definícia 3.1. Nech h je n-árna čiastočná funkcia, g je $(n+2)$-árna čiastočná funkcia a f je $(n+1)$-árna čiastočná funkcia. Budeme hovoriť, že čiastočná funkcia f vzniká z čiastočných funkcií g, h *operáciou primitívnej rekurzie* a písať $f = \mathcal{R}(g,h)$, ak pre všetky $x_1, \ldots, x_n, y \in \mathbb{N}$ platí

$$f(0, x_1, \ldots, x_n) = h(x_1, \ldots, x_n) \tag{3.1.1}$$

$$f(y+1, x_1, \ldots, x_n) = g(y, f(y, x_1, \ldots, x_n), x_1, \ldots, x_n) \tag{3.1.2}$$

Nechávame čitateľovi na rozmyslenie, že ku každej $(n+2)$-árnej čiastočnej funkcii g a n-árnej čiastočnej funkcii h existuje práve jedna $(n+1)$-árna čiastočná funkcia f, ktorá spĺňa podmienky (3.1.1), (3.1.2) a že v prípade, ak g, h sú totálne funkcie, aj f je totálna funkcia.

Príklad 3.2. (a) Nech $f = \lambda xy(x.y)$, $g = \lambda xyz(y+z)$, $h = \lambda x(0)$. Potom platí

$$f(0, y) = 0 = h(y)$$
$$f(x+1, y) = x.y + y = g(x, f(x,y), y)$$

a teda funkcia f vzniká z funkcií g, h operáciou primitívnej rekurzie. Môžeme teda písať $f = \mathcal{R}(g,h)$ alebo aj $\lambda xy(x.y) = \mathcal{R}(\lambda xyz(y+z), \lambda x(0))$.

(b) Nech $g = \emptyset$, $h = \lambda x(2^x)$. V tomto prípade je $f = \mathcal{R}(g,h)$ taká binárna čiastočná funkcia, že $f(x,y) = 2^y$ pre $x = 0$ a $f(x,y)\uparrow$ pre $x \neq 0$.

Cvičenie 3.3. Nájdite také čiastočné funkcie f, g, h, že $f = \mathcal{R}(g,h)$, pričom f, h sú totálne a g nie je totálna. Možno dosiahnuť, aby $f = \mathcal{R}(g,h)$, f, g boli totálne a h nebola totálna?

Cifru 0 budeme používať nielen na označenie čísla 0, ale aj pre nulárnu funkciu s hodnotou 0. (Obdobne budeme postupovať aj pri iných číslach.) Budeme používať označenie I_m^n $(m, n \in \mathbb{N}, 0 < m \leq n)$ pre projekcie na množine $\mathbb{N}$ a označenie s pre funkciu $\lambda x(x+1)$, t.j. pre funkciu *nasledovník*.

Definícia 3.4. Budeme hovoriť, že funkcia f je *primitívne rekurzívna* , ak vzniká z funkcií 0, s a I_m^n konečným počtom operácií skladania funkcií a primitívnej rekurzie.

Poznámka 3.5. Keby sme chceli našu teóriu formalizovať, mohli by vzniknúť ťažkosti so zvratom „vzniká konečným počtom operácií“ použitým v predchádzajúcej definícii. Preto by mohlo byť vhodné upraviť definíciu 3.4 napríklad takto:

a teda funkcia $x + y$ vzniká z primitívne rekurzívnych funkcií I_1^1 a $\mathcal{S}^2(s, I_2^3)$ operáciou primitívnej rekurzie, teda je sama primitívne rekurzívna. Z už dokázaného a zo vzťahov

$$\begin{aligned} 0.y &= o(y) \\ (x+1).y &= I_2^3(x, x.y, y) + I_3^3(x, x.y, y) \end{aligned} \tag{3.10.2}$$

vyplýva primitívna rekurzívnosť funkcie $x.y$ a z nej a zo vzťahov

$$\begin{aligned} y^0 &= K_1^1(y) \\ y^{x+1} &= I_2^3(x, y^x, y) \cdot I_3^3(x, y^x, y) \end{aligned} \tag{3.10.3}$$

(kde sme použili označenie K_1^1 pre unárnu funkciu identicky rovnajúcu sa jednej) vyplýva primitívna rekurzívnosť funkcie y^x. □

Poznámka 3.11. Funkcia $\lambda xy(x^y)$ je tiež primitívne rekurzívna, lebo platí

$$\lambda xy(x^y) = \mathcal{S}^3(\lambda xy(y^x), I_2^2, I_1^2) \tag{3.11.1}$$

Veta 2.15 nám umožňuje všetky podobné zmeny poradia premenných, pridávanie fiktívnych premenných a pod. Preto odteraz nebudeme prísne dbať na poradie a počty premenných pri zápise primitívnej rekurzie, ani pri zápise skladania funkcií. Vždy sme ho totiž mohli upraviť presne na tvar požadovaný definíciou, pravda, niekedy za cenu zvýšenia počtu zápisov.

Veta 3.12. Funkcie

$$\mathrm{sg}(x) = \begin{cases} 0, & \text{ak } x = 0 \\ 1, & \text{ak } x \neq 0 \end{cases} \tag{3.12.1}$$

$$\overline{\mathrm{sg}}(x) = 1 - \mathrm{sg}(x) \tag{3.12.2}$$

sú primitívne rekurzívne.

Dôkaz: Tieto funkcie vznikajú primitívnou rekurziou z konštantných funkcií, napríklad

$$\begin{aligned} \mathrm{sg}(0) &= 0 \\ \mathrm{sg}(x+1) &= 1 \end{aligned} \tag{3.12.3}$$

a preto sú primitívne rekurzívne. □

Lema 3.13. Funkcia

$$x \dot{-} 1 = \begin{cases} 0, & \text{ak } x = 0 \\ x - 1 & \text{ak } x \neq 0 \end{cases} \tag{3.13.1}$$

je primitívne rekurzívna.

Dôkaz: Funkcie 0, I_1^2 sú primitívne rekurzívne a platí

$$\begin{aligned} 0 \dot{-} 1 &= 0 \\ (x+1) \dot{-} 1 &= I_1^2(x, x \dot{-} 1) \end{aligned} \tag{3.13.2}$$

Tým je lema dokázaná. □

Veta 3.14. Funkcie

$$x \dot{-} y = \begin{cases} 0, & \text{ak } x < y \\ x - y, & \text{ak } x \geq y \end{cases} \tag{3.14.1}$$

$$|x - y| = (x \dot{-} y) + (y \dot{-} x) \tag{3.14.2}$$

sú primitívne rekurzívne.

Dôkaz: Pre prvú funkciu stačí uvážiť vzťahy

$$\begin{aligned} x \dot{-} 0 &= x \\ x \dot{-} (y+1) &= (x \dot{-} y) \dot{-} 1 \end{aligned} \tag{3.14.3}$$

Primitívna rekurzívnosť funkcie $|x - y|$ vyplýva teraz už bezprostredne z jej definície. □

Často definujeme funkcie tzv. rozborom prípadov — obor definície rozdelíme na niekoľko častí a na každej z nich vyjadríme zvláštnym predpisom hodnoty funkcie. Tento postup za istých podmienok zachováva primitívnu rekurzívnosť, ako vyplýva z nasledujúcej vety.

Veta 3.20. Nech $f_1, \ldots, f_{s+1}, h_1, \ldots, h_s$ sú n-árne primitívne rekurzívne funkcie. Nech pre žiadne $x_1, \ldots, x_n \in \mathbb{N}$ sa žiadne dve z funkcií $h_1, \ldots, h_s$ súčasne nerovnajú nule. Potom funkcia f definovaná predpisom

$$f(x_1, \ldots, x_n) = \begin{cases} f_1(x_1, \ldots, x_n), & \text{ak } h_1(x_1, \ldots, x_n) = 0, \\ \quad \vdots & \\ f_s(x_1, \ldots, x_n), & \text{ak } h_s(x_1, \ldots, x_n) = 0, \\ f_{s+1}(x_1, \ldots, x_n), & \text{v ostatných prípadoch} \end{cases}$$

je primitívne rekurzívna.

Dôkaz: Funkciu f možno vyjadriť pomocou operácie skladania funkcií a primitívne rekurzívnych funkcií takto:

$$f = f_1 . \overline{\text{sg}}(h_1) + f_2 . \overline{\text{sg}}(h_2) + \cdots + f_s . \overline{\text{sg}}(h_s) + f_{s+1} . \text{sg}(h_1 . \ldots . h_s)$$

Preto je funkcia f primitívne rekurzívna. □

Cvičenie 3.21. Dokážte, že funkcie $\lfloor \frac{x}{2} \rfloor$, $x!$ sú primitívne rekurzívne.

Často je výhodné použiť pri definícii (čiastočných) funkcií tzv. operáciu minimalizácie. Táto operácia vo všeobecnom prípade nezachováva primitívnu rekurzívnosť, ale v prípade tzv. ohraničenej minimalizácie áno. Budeme hneď definovať operáciu minimalizácie vo všeobecnom prípade a až potom uvedieme nutné ohraničenia pre náš prípad.

Definícia 3.22. Nech f je n-árna čiastočná funkcia, g je $(n+1)$-árna čiastočná funkcia. Budeme hovoriť, že čiastočná funkcia f vzniká z čiastočnej funkcie g *minimalizáciou* a písať $f = \mathcal{M}(g)$ alebo

$$f(x_1, \ldots, x_n) = \mu_y(g(y, x_1, \ldots, x_n) = 0), \tag{3.22.1}$$

ak pre všetky $x_1, \ldots, x_n \in \mathbb{N}$ platí $f(x_1, \ldots, x_n) = y$ práve vtedy, keď pre všetky $z < y$ je $g(z, x_1, \ldots, x_n)$ definované a kladné a súčasne $g(y, x_1, \ldots, x_n) = 0$.

Definícia 3.23. Ak $f = \mathcal{M}(g)$ a f, g sú totálne funkcie, budeme hovoriť, že funkcia f vzniká z funkcie g *regulárnou minimalizáciou*.

Príklad 3.24. Nech v definícii 3.22 $n = 1$ a $g = I_2^2$. Vtedy je $f(x)$ unárna čiastočná funkcia a platí $f(x) = 0$ pre $x = 0$, $f(x) \uparrow$ pre $x \neq 0$. Vidíme teda, že minimalizáciou môže z totálnej funkcie vzniknúť čiastočná funkcia, ktorá nie je totálna.

Poznámka 3.25. Z príkladu 3.24 vidieť, že operácia minimalizácie nezachováva primitívnu rekurzívnosť. Dá sa tiež ukázať, že ani operácia regulárnej minimalizácie nezachováva primitívnu rekurzívnosť, to je však už podstatne komplikovanejšie.

Veta 3.26. Nech $g(y, x_1, \ldots, x_n)$, $h(x_1, \ldots, x_n)$ sú primitívne rekurzívne funkcie a nech pre každé $x_1, \ldots, x_n \in \mathbb{N}$ existuje také $z \leq h(x_1, \ldots, x_n)$, že $g(z, x_1, \ldots, x_n) = 0$. Potom aj funkcia $f(x_1, \ldots, x_n) = \mu_y(g(y, x_1, \ldots, x_n) = 0)$ je primitívne rekurzívna.

Dôkaz: Nech

$$f_1(z, x_1, \ldots, x_n) = \prod_{i=0}^{z} g(i, x_1, \ldots, x_n) \tag{3.26.1}$$

V postupnosti $f_1(0, x_1, \ldots, x_n)$, $f_1(1, x_1, \ldots, x_n)$, $f_1(2, x_1, \ldots, x_n)$, ... je presne

$$\mu_y(g(y, x_1, \ldots, x_n) = 0)$$

nenulových členov, a všetky členy počnúc $f_1(h(x_1, \ldots, x_n), x_1, \ldots, x_n)$ sú nulové. Preto platí

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{h(x_1, \ldots, x_n)} \text{sg}(f_1(i, x_1, \ldots, x_n)) \quad \square$$

Pre funkciu $\pi(x)$ zrejme platí

$$\pi(x) = \sum_{i=0}^{x} \overline{\mathrm{sg}}(\chi_p(i)) \tag{3.32.4}$$

Z uvedených vyjadrení vyplýva, že funkcie $\chi_p(x)$, $\pi(x)$ sú primitívne rekurzívne. □

Veta 3.33. Funkcia

$$\mathrm{p}(x) = x\text{-té prvočíslo}, \tag{3.33.1}$$

t.j. $\mathrm{p}(0) = 2$, $\mathrm{p}(1) = 3$, $\mathrm{p}(2) = 5$, ... je primitívne rekurzívna.

Dôkaz: Platí

$$\mathrm{p}(x) = \mu_y(|\pi(y) - (x+1)| = 0) \tag{3.33.2}$$

teda funkcia $\mathrm{p}(x)$ vzniká z funkcie $\lambda yx|\pi(y) - (x+1)|$ minimalizáciou. Matematickou indukciou sa dá dokázať, že platí

$$\mathrm{p}(x) \leq 2^{2^x} \tag{3.33.3}$$

Funkcia 2^{2^x} je primitívne rekurzívna, a preto podľa vety 3.26 aj funkcia $\mathrm{p}(x)$ je primitívne rekurzívna. □

Cvičenie 3.34. Dokážte odhad (3.33.3) alebo nejaký iný horný odhad funkcie $\mathrm{p}(x)$ primitívne rekurzívnou funkciou.

Veta 3.35. Funkcia

$$\mathrm{ex}(x,y) = \begin{cases} \text{exponent prvočísla } \mathrm{p}(x) \text{ v rozklade čísla } y \text{ na prvočinitele}, & \text{ak } y \neq 0 \\ 0, & \text{ak } y = 0 \end{cases} \tag{3.35.1}$$

je primitívne rekurzívna.

Dôkaz: Platí $\mathrm{ex}(x,y) \leq y$ a $\mathrm{ex}(x,y) = \mu_u(\mathrm{sg}(y).\,\mathrm{div}(y,(\mathrm{p}(x))^{u+1}) = 0)$. Teraz stačí použiť vetu 3.26. □

Poznámka 3.36. Namiesto $\mathrm{p}(x)$, $\mathrm{ex}(x,y)$ budeme niekedy písať p_x, $\mathrm{ex}_x(y)$.

Veta 3.37. Funkcie

$$\lfloor\sqrt{x}\rfloor = \mu_z(\overline{\mathrm{sg}}((z+1)^2 \dot{-} x) = 0) \tag{3.37.1}$$

$$\mathrm{q}(x) = x \dot{-} \lfloor\sqrt{x}\rfloor^2 \tag{3.37.2}$$

sú primitívne rekurzívne.

Dôkaz: Veta bezprostredne vyplýva zo vzťahov v nej uvedených a z nerovnosti $\lfloor\sqrt{x}\rfloor \leq x$. □

Poznámka 3.38. Funkcia $\mathrm{q}(x)$ sa nazýva *kvadratický zvyšok*.

Cvičenie 3.39. Dokážte primitívnu rekurzívnosť nasledujúcich funkcií (všetky označenia chápte ako celok v zmysle poznámky 3.15):

- $\lfloor\log_x y\rfloor$ (s nejakým vhodným dodefinovaním pre prípady $y = 0$, $x = 0, 1$)
- $\lfloor x.\sqrt{6}\rfloor$; $\lfloor x.\sqrt{x}\rfloor$; $\lfloor\sqrt{x} - \sqrt[3]{x}\rfloor$
- $\lfloor 4 \cdot \arctan(x)\rfloor$
- $\pi_2(x)$ = počet prvočíselných dvojčiat, ktorých aritmetický priemer nepresahuje x. (Prvočíselné dvojčatá sú dvojice tvaru $(p, p+2)$, kde p, $p+2$ sú prvočísla.)

Príklad 3.40. Na ukážku vyriešime z cvičenia 3.39 úlohu s funkciou $f(x) = \lfloor\sqrt{x} - \sqrt[3]{x}\rfloor$. Najprv urobíme prípravné matematické úvahy o reálnom čísle $u = \sqrt{x} - \sqrt[3]{x}$. Pre každé $x \in \mathbb{N}$ je číslo u celé algebraické, teda je koreňom nejakého polynómu (alebo aj: algebraickej rovnice) s celočíslenými koeficientmi. Túto rovnicu dostaneme zo vzťahu $\sqrt[3]{x} = \sqrt{x} - u$ umocnením na tretiu, prevedením členov bez $\sqrt{x}$ na ľavú stranu a novým umocnením na druhú. Po úprave dostaneme

$$u^6 - 3xu^4 + 2xu^3 + 3x^2u^2 + 6x^2u + x^2 - x^3 = 0$$

Zrejme platí:

Veta 3.43. Funkcie c, l, r tvoria trojicu číslovacích funkcií na množine $\mathbb{N}$, t.j. pre všetky $x, y \in \mathbb{N}$ platí

$$\mathrm{c}(\mathrm{l}(x), \mathrm{r}(x)) = x \tag{3.43.1}$$
$$\mathrm{l}(\mathrm{c}(x, y)) = x \tag{3.43.2}$$
$$\mathrm{r}(\mathrm{c}(x, y)) = y \tag{3.43.3}$$

Vyjadríme teraz funkcie c, l, r takým spôsobom, z ktorého bude zrejmá ich primitívna rekurzívnosť.

Hodnota $\mathrm{c}(x, y)$ sa rovná počtu usporiadaných dvojíc, ktoré sú menšie ako dvojica (x, y). Medzi týmito dvojicami je jedna dvojica so súčtom 0, dve dvojice so súčtom 1, ... až $x + y$ dvojíc so súčtom $x + y - 1$, a ďalej x dvojíc so súčtom $x + y$, ale prvou zložkou menšou ako x. Preto platí

$$\mathrm{c}(x, y) = 1 + 2 + \cdots + (x + y) + x = \frac{(x + y)(x + y + 1)}{2} + x$$

Nech teraz $z = \mathrm{c}(x, y)$. Postupne dostávame

$$\begin{aligned} 2z &= (x + y)(x + y + 1) + 2x \\ 8z + 1 &= (2x + 2y)(2x + 2y + 2) + 8x + 1 \\ 8z + 1 &= (2x + 2y + 1)^2 + 8x \\ 8z + 1 &= (2x + 2y + 3)^2 - 8x - 8 \end{aligned}$$

Odtiaľ dostávame nerovnosti

$$\begin{aligned} (2x + 2y + 1)^2 &\leq 8z + 1 < (2x + 2y + 3)^2 \\ 2x + 2y + 1 &\leq \lfloor \sqrt{8z + 1} \rfloor < 2x + 2y + 3 \\ x + y + 1 &\leq \left\lfloor \frac{\lfloor \sqrt{8z + 1} \rfloor + 1}{2} \right\rfloor < x + y + 2 \end{aligned}$$

Preto platí

$$x + y + 1 = \left\lfloor \frac{\lfloor \sqrt{8z + 1} \rfloor + 1}{2} \right\rfloor$$

Pretože $2x = 2z - (x + y)(x + y + 1)$, platí aj

$$\mathrm{l}(z) = x = z - \frac{(x + y)(x + y + 1)}{2} = z \dot{-} \left\lfloor \frac{\left\lfloor \frac{\lfloor \sqrt{8z+1} \rfloor}{2} \right\rfloor \cdot \left\lfloor \frac{\lfloor \sqrt{8z+1} \rfloor + 1}{2} \right\rfloor}{2} \right\rfloor$$

$$\mathrm{r}(z) = y = \left\lfloor \frac{\lfloor \sqrt{8z + 1} \rfloor + 1}{2} \right\rfloor \dot{-} (\mathrm{l}(z) + 1)$$

Dokázané výsledky sformulujeme ako vetu.

Veta 3.44. Pre funkcie c, l, r platí

$$\mathrm{c}(x, y) = \frac{(x + y)(x + y + 1)}{2} + x \tag{3.44.1}$$

$$\mathrm{l}(x) = x \dot{-} \left\lfloor \frac{\left\lfloor \frac{\lfloor \sqrt{8x+1} \rfloor}{2} \right\rfloor \cdot \left\lfloor \frac{\lfloor \sqrt{8x+1} \rfloor + 1}{2} \right\rfloor}{2} \right\rfloor \tag{3.44.2}$$

$$\mathrm{r}(x) = \left\lfloor \frac{\lfloor \sqrt{8x + 1} \rfloor + 1}{2} \right\rfloor \dot{-} (\mathrm{l}(x) + 1) \tag{3.44.3}$$

a teda tieto funkcie sú primitívne rekurzívne.

Dôkaz: Funkcie

$$\mathrm{npr}(x) = \mu_z \left(\sum_{i=z}^{x} \mathrm{ex}_i(x) = 0 \right) \tag{3.58.1}$$

$$\mathrm{pos}(x,y) = \mathrm{sg}(y).\left(\prod_{i=0}^{y} \mathrm{p}_{i+x}^{\mathrm{ex}_i(y)} \right) \tag{3.58.2}$$

sú zrejme primitívne rekurzívne. Pritom pre $x \neq 0$ je $\mathrm{npr}(x)$ také najmenšie číslo z, že prvočísla $\mathrm{p}_z, \mathrm{p}_{z+1}, \ldots$ nevystupujú v rozklade čísla x na prvočinitele. Číslo $\mathrm{pos}(x,y)$ je v prípade $y \neq 0$ to číslo, ktoré vznikne z čísla y „posunutím" exponentov v rozklade čísla y o x miest doprava. Preto pre $x \neq 0$, $y \neq 0$ platí

$$x * y = x.\,\mathrm{pos}(\mathrm{npr}(x), y) \tag{3.58.3}$$

Tento vzťah však platí aj pre $x = 0$ alebo $y = 0$, lebo vtedy sa obe strany tejto rovnosti rovnajú 0. Z vyjadrenia (3.58.3) vidieť, že aj funkcia $x * y$ je primitívne rekurzívna. □

Funkcie c^n nám umožňujú číslovať usporiadané n-tice prirodzených čísel pre pevne zvolené n, avšak neumožňujú nám číslovať n-tice prirodzených čísel pre všetky n súčasne. Na tieto účely môžeme využívať vlastnosti rozkladu prirodzeného čísla na prvočinitele a použiť princíp kódovania, ktorý bol spomenutý vyššie (ešte by bolo treba doriešiť prípad postupností končiacich sa nulou). Na niektoré účely však môžeme použiť aj Gödelovu funkciu Γ, ktorá je v istom zmysle jednoduchšia ako číslovanie pomocou exponentov.

Definícia 3.59. Pre každé $x, y \in \mathbb{N}$

$$\Gamma(x,y) = \mathrm{l}(x) \text{ MOD } (1 + (y+1) \cdot \mathrm{r}(x)) \tag{3.59.1}$$

Veta 3.60. Funkcia Γ je primitívne rekurzívna.

Dôkaz: Vyplýva bezprostredne z definície. □

Základnú vlastnosť Gödelovej funkcie Γ vyjadruje nasledujúca veta.

Veta 3.61. Pre každé prirodzené číslo n a každú konečnú postupnosť $(a_0, a_1, \ldots, a_n)$ prirodzených čísel existuje také číslo x, že platí

$$\Gamma(x,i) = a_i \qquad \text{pre } i = 0, 1, \ldots, n \tag{3.61.1}$$

Dôkaz: Číslo x budeme hľadať v tvare $x = \mathrm{c}(u,b)$, kde $b = (1 + n + a_0 + a_1 + \cdots + a_n)!$ Nech pre $y = 0, 1, \ldots, n$ platí $m_y = 1 + (y+1)b$. Teraz stačí nájsť také číslo u, aby platilo $u \text{ MOD } m_y = a_y$ pre $y = 0, 1, \ldots, n$. Pretože zrejme $m_y > a_y$, stačí dosiahnuť, aby každé $u - a_y$ bolo násobkom čísla m_y. Čísla $m_0, m_1, \ldots, m_n$ sú po dvoch nesúdeliteľné. Skutočne, ak $i < j$ a nejaké prvočíslo p delí m_i aj m_j, tak p delí aj $(j-i)b$, a teda aj b. To je ale spor s voľbou čísel m_y. Označme teraz $M_i = m_0 m_1 \cdots m_{i-1} m_{i+1} \cdots m_n$ $(i = 0, 1, \ldots, n)$. Čísla m_i, M_i sú nesúdeliteľné, a preto existujú také čísla z_i, w_i, že pre všetky $i = 0, 1, \ldots, n$ platí $M_i z_i - m_i w_i = 1$. Teraz stačí, keď $u = a_0 M_0 z_0 + {} + a_1 M_1 z_1 + \cdots + a_n M_n z_n$. Pre každé $i = 0, 1, \ldots, n$ platí $u \text{ MOD } m_i = a_i M_i z_i \text{ MOD } m_i = a_i(1 + {} + m_i w_i) \text{ MOD } m_i = a_i$, čo sme potrebovali dosiahnuť. □

Poznámka 3.62. Vo vete 3.61 sa netvrdí, že x je jednoznačne určené postupnosťou $(a_0, \ldots, a_n)$. Naopak, pre každú takúto postupnosť zrejme existuje nekonečne mnoho x s požadovanou vlastnosťou. Analýzou dôkazu vety 3.61 by sme zistili, že pre každé prirodzené číslo n existuje taká $(n+1)$-árna primitívne rekurzívna funkcia h, že najmenšie x s vlastnosťou (3.61.1) neprevyšuje $h(a_0, a_1, \ldots, a_n)$ a podľa vety 3.26 by sme teda vhodnou voľbou h mohli dosiahnuť aj $x = {} = h(a_0, a_1, \ldots, a_n)$.

Pomocou doteraz uvedených vlastností primitívne rekurzívnych funkcií čitateľ už ľahko dokáže primitívnu rekurzívnosť mnohých ďalších funkcií, s ktorými sa stretneme. My uvedieme na túto tému ešte niekoľko príkladov a cvičení.

Z neho vyplýva, že funkcia f je primitívne rekurzívna.

Cvičenie 3.65. Dokážte, že nasledujúce funkcie sú primitívne rekurzívne:

- $f_1(x) = \mathrm{c}^{x+1}(0, 1, \ldots, x)$
- $f_2(x) =$ počet dokonalých čísel nepresahujúcich x (číslo sa nazýva dokonalé, ak súčet jeho kladných deliteľov sa rovná jeho dvojnásobku)
- $f_3(x) = x$-tá cifra dekadického rozvoja čísla $\log 7$
- $f_4(x) =$ počet núl v dekadickom zápise čísla x^x
- $f_5(x) =$ počet sedmičiek na prvých x miestach dekadického rozvoja čísla $\sqrt{1+\sqrt{2}}$
- $f_6(x, y) =$ najväčší spoločný deliteľ čísel x, y (pre $x = y = 0$ kladieme $f_7(x, y) = 0$)
- $f_7(x, y) =$ najväčší možný determinant štvorcovej matice typu $(x+1) \times (x+1)$ s prvkami množiny $\{0, 1, \ldots, y\}$
- $f_8(x, y) =$ také najmenšie číslo n, že číslo x sa dá písať ako súčet n y-tych mocnín
- $f_9(x) =$ počet spôsobov, ktorými sa číslo x dá písať ako súčet tretích mocnín prirodzených čísel (pričom nehľadíme na poradie sčítancov)
- $f_{10}(y, z) =$ počet regulárnych matíc typu $(y+1) \times (y+1)$ s prvkami z množiny $\{0, 1, \ldots, z\}$

Príklad 3.66. Ukážeme, že funkcia

$$h(x) = x\text{-tá číslica (za desatinnou čiarkou) čísla e}$$

je primitívne rekurzívna. (Platí $\mathrm{e} = 2.718\ldots$; nech teda 2 je nultá číslica, 7 prvá číslica, atď.)

Návod: Stačí ukázať, že funkcia $\lfloor \mathrm{e}\, x \rfloor$ je primitívne rekurzívna; potom možno postupovať podľa príkladu 3.64. Ďalej, pretože pre $x > 0$ zrejme platí

$$\lfloor \mathrm{e}\, x \rfloor = \lfloor \mathrm{e}\, x! \rfloor \ \mathrm{DIV}\ (x-1)!$$

stačí namiesto funkcie $\lfloor \mathrm{e}\, x \rfloor$ uvažovať funkciu $\lfloor \mathrm{e}\, x! \rfloor$. Budeme vychádzať zo známeho vyjadrenia e v tvare súčtu prevrátených hodnôt faktoriálov. Pre $x > 0$ platí

$$\lfloor \mathrm{e}\, x! \rfloor = \left\lfloor x\,! \sum_{k=0}^{\infty} \frac{1}{k!} \right\rfloor = \left\lfloor \sum_{k=0}^{x} \frac{x!}{k!} + \sum_{k=x+1}^{\infty} \frac{x!}{k!} \right\rfloor = \sum_{k=0}^{x} \left\lfloor \frac{x!}{k!} \right\rfloor + \left\lfloor \sum_{k=x+1}^{\infty} \frac{x!}{k!} \right\rfloor = \sum_{k=0}^{x} \left\lfloor \frac{x!}{k!} \right\rfloor$$

Z nájdeného vyjadrenia už vidno, že funkcia $\lfloor \mathrm{e}\, x! \rfloor$ je primitívne rekurzívna. (Ešte je potrebná malá úprava kvôli $x = 0$.) V poslednom kroku sme využili odhad

$$0 < \sum_{k=x+1}^{\infty} \frac{x!}{k!} < \sum_{k=x+1}^{\infty} \frac{1}{(x+1)^k} < 1$$

Preto je celá časť odhadovaného výrazu rovná nule.

Našou úlohou je skúmanie algoritmicky vypočítateľných (čiastočných) funkcií. Všetky primitívne rekurzívne funkce sú algoritmicky vypočítateľné, ale existujú algoritmicky vypočítateľné funkcie, ktoré nie sú primitívne rekurzívne. Uvedieme príklad takejto funkcie.

Príklad 3.67. Zostrojíme algoritmicky vypočítateľnú funkciu univerzálnu pre množinu všetkých unárnych primitívne rekurzívnych funkcií. Podľa vety 3.50 táto funkcia nebude primitívne rekurzívna.

Každá primitívne rekurzívna funkcia vzniká z funkcií 0, s, I_m^n operáciami skladania funkcií $\mathcal{S}^2$, $\mathcal{S}^3$, $\mathcal{S}^4$, ... a operáciou primitívnej rekurzie $\mathcal{R}$. Preto každej primitívne rekurzívnej funkcii zodpovedá aspoň jeden správne zostrojený výraz z týchto symbolov, čiarok a zátvoriek. Napríklad funkcii o zodpovedá výraz $\mathcal{R}(0, I_2^2)$, lebo funkcia o vzniká primitívnou rekurziou z funkcií 0, I_2^2. Funkcii o^3, t.j. ternárnej funkcii identicky rovnajúcej sa nule, zodpovedá výraz $\mathcal{S}^2(\mathcal{R}(0, I_2^2), I_1^3)$, lebo platí $o^3 = \mathcal{S}^2(o, I_1^3)$. Všetky správne zostrojené výrazy zoradíme do postupnosti

$$V_0, V_1, V_2, V_3, V_4, \ldots \tag{3.67.1}$$

tak, aby výrazy boli zoradené podľa rastúcej dĺžky, pričom pod dĺžkou výrazu rozumieme súčet počtov všetkých $\mathcal{S}$, $\mathcal{R}$, I, 0, s vo výraze a všetkých indexov vo výraze. Výrazy tej istej dĺžky zoradíme ľubovoľným pevne zvoleným efektívnym spôsobom. Funkciu zodpovedajúcu výrazu V_x

Definícia 3.76. Čiastočnú funkciu f budeme nazývať *čiastočne rekurzívnou funkciou*, ak f vzniká z funkcií 0, s, I_m^n $(1 \leq m \leq n)$ konečným počtom operácií skladania funkcií, primitívnej rekurzie a minimalizácie.

Poznámka 3.77. Prípadné ťažkosti s formalizáciou zvratu „vzniká konečným počtom operácií" v definíciách 3.76, 3.77 by sme mohli odstrániť podľa poznámky 3.5.

Poznámka 3.78. Bolo by dôslednejšie hovoriť čiastočne rekurzívna čiastočná funkcia f namiesto čiastočne rekurzívna funkcia f, ale takémuto opakovaniu slov v jednom termíne sa budeme vyhýbať.

Veta 3.79. (a) Každá primitívne rekurzívna funkcia je rekurzívna.

(b) Každá rekurzívna funkcia je čiastočne rekurzívna.

Dôkaz: Vyplýva bezprostredne z uvedených definícií. □

Veta 3.80. Množina všetkých (čiastočne) rekurzívnych funkcií je normálny klon (čiastočných) funkcií na množine $\mathbb{N}$.

Definícia 3.81. Klon všetkých rekurzívnych funkcií budeme označovať $\mathcal{R}$. Klon všetkých čiastočne rekurzívnych funkcií budeme označovať $\mathcal{P}$.

Ako uvidíme neskôr, klon všetkých rekurzívnych funkcií možno dostať zo systému všetkých čiastočne rekurzívnych funkcií podľa vety 2.16.

Cvičenie 3.82. Dokážte, že funkcia $G(x, y)$ z príkladu 3.67 je rekurzívna. (Toto cvičenie je dosť náročné, a preto ho môže čitateľ odložiť a riešiť až po preštudovaní nasledujúcej kapitoly.)

Príklad 3.83. Čiastočná funkcia $\lambda xy(x - y)$, teda rozdiel, je čiastočne rekurzívna, lebo vzniká z primitívne rekurzívnej funkcie $\lambda zxy\big(|(z + y) - x|\big)$ minimalizáciou:

$$x - y = \mu_z\big(|(z + y) - x| = 0\big) \tag{3.83.1}$$

Poznámka 3.84. V predchádzajúcom príklade pre každé x, y existuje najviac jedno také z, že $|(z+y)-x| = 0$. Toto z, pokiaľ existuje, je samozrejme i najmenším z s požadovanou vlastnosťou, a preto ho môžeme nájsť minimalizáciou. Týmto spôsobom budeme minimalizáciu používať často.

Podľa cvičenia 3.82 a príkladu 3.83 nedá sa obrátiť ani jedna časť vety 3.79.

Cvičenie 3.85. Dokážte, že nasledujúce čiastočné funkcie sú čiastočne rekurzívne (pritom bežné označenia pre reálne funkcie reálnej premennej chápte ako označenia pre zúženia týchto funkcií na množine $\mathbb{N}$ a každý vzorec chápte (v zmysle poznámky 3.15) ako celok):

- $f_1(x) = \log_{36} x$
- $f_2(x) = \mathrm{r}(\mathrm{r}(x)) - \mathrm{l}(x)$
- $f_3(x) = \sqrt{x}$
- $f_4(x, y) = \lfloor \log_x y \rfloor$ (bez dodefinovania)
- $f_5(x, y) = 2^x - 3y$
- $f_6(x, y) = \uparrow$

Cvičenie 3.86. Dokážte, že nasledujúce funkcie sú rekurzívne:

- $f_1(x) = x$-tá cifra dekadického rozvoja čísla π
- $f_2(x) = \lfloor \mathrm{tg}\, x \rfloor$
- $f_3(x, y) = \log_{x+2}(y + 1)$
- $f_4(x) = \lfloor \sqrt{x} + \sqrt[3]{x} \rfloor$
- $f_5(x) = \lfloor \sin(x) + \cos(x) + x \rfloor$

Cvičenie 3.87. Dokážte, že lema 3.17, veta 3.18 a veta 3.20 budú platiť aj vtedy, keď z nich všade vynecháme slovo „primitívne".

Poznámka 3.88. Vety z predchádzajúceho cvičenia, a aj niektoré ďalšie vety tejto kapitoly možno ľahko preformulovať tak, aby platili pre ľubovoľný klon funkcií na množine $\mathbb{N}$, ktorý obsahuje funkcie 0, s a je uzavretý vzhľadom na operáciu primitívnej rekurzie. Ani dôkazy týchto viet by sa v podstate nezmenili. Ale pri úprave týchto viet pre klony čiastočných funkcií musíme byť opatrní; dokonca aj vtedy, keď nejaká veta bude platiť aj pre klony čiastočných funkcií (s vyššie spomenutými vlastnosťami), môže byť potrebné podstatne zmeniť jej dôkaz.

(e) $x = y$
(f) $x \neq y$
(g) $ex_x(y) = ex_y(z)$
(h) existuje obdĺžnik s dĺžkami strán x, y s dĺžkou uhlopriečky z
(i) x je deliteľné aspoň troma rôznymi prvočíslami

Vyslovíme teraz niekoľko viet o tvorení nových (primitívne) rekurzívnych množín a obdobné vety pre predikáty. Vo vetách o množinách nebudeme zvlášť uvádzať prípad množín prirodzených čísel (ktorý zodpovedá bodu (b) definície 4.1), ale necháme čitateľa, aby si sám odvodil z prípadu „1-tíc" príslušné tvrdenie.

Veta 4.7. Nech $n \in \mathbb{N}$ je ľubovoľné a nech $M_1 \subseteq \mathbb{N}^n$, $M_2 \subseteq \mathbb{N}^n$ sú (primitívne) rekurzívne množiny. Potom sú aj množiny $M_1 \cup M_2$, $M_1 \cap M_2$, $M_1 \setminus M_2$, $\mathbb{N}^n \setminus M_1$ (primitívne) rekurzívne.

Dôkaz: Ak sú $\chi_1(x_1, \ldots x_n)$, resp. $\chi_2(x_1, \ldots x_n)$ charakteristické funkcie množín M_1, resp. M_2, tak $1 \dot{-} \chi_1(x_1, \ldots x_n)$, $\chi_1(x_1, \ldots x_n) \cdot \chi_2(x_1, \ldots x_n)$ sú charakteristické funkcie množín $\mathbb{N}^n \setminus M_1$, $M_1 \cup M_2$ a sú zrejme (primitívne) rekurzívne. Pretože rozdiel a prienik množín možno vyjadriť pomocou operácií zjednotenia a komplementu, sú aj množiny $M_1 \cap M_2$, $M_1 \setminus M_2$ (primitívne) rekurzívne. □

Veta 4.8. Nech $n \in \mathbb{N}$, $m \in \mathbb{N}$ sú ľubovoľné a nech $M_1 \subseteq \mathbb{N}^n$, $M_2 \subseteq \mathbb{N}^m$ sú (primitívne) rekurzívne množiny. Potom aj množina všetkých takých $(m+n)$-tíc $(a_1, \ldots, a_n, a_{n+1}, \ldots, a_{n+m})$, že $(a_1, \ldots, a_n) \in M_1$, $(a_{n+1}, \ldots, a_{n+m}) \in M_2$, je primitívne rekurzívna.

Dôkaz: Ak sú $\chi_1(x_1, \ldots x_n)$, $\chi_2(y_1, \ldots y_m)$ charakteristické funkcie množín M_1, M_2, je

$$\chi_3(x_1, \ldots x_n, y_1 \ldots, y_m) = \operatorname{sg}(\chi_1(x_1, \ldots x_n) + \chi_2(y_1, \ldots y_m))$$

charakteristická funkcia množiny M_3. Ak sú funkcie χ_1, χ_2 (primitívne) rekurzívne, je aj funkcia χ_3 (primitívne) rekurzívna, čo bolo treba dokázať. □

Cvičenie 4.9. Veta 4.8 sa často stručne, i keď nie celkom presne, vyslovuje takto: Ak sú množiny M_1, M_2 (primitívne) rekurzívne, je aj množina $M_1 \times M_2$ (primitívne) rekurzívna. V čom spočíva nepresnosť tejto formulácie ?

Veta 4.10. Ak sú $P(x_1, \ldots, x_n)$, $Q(y_1, \ldots, y_m)$ (primitívne) rekurzívne predikáty, sú aj

$$\begin{aligned} &P(x_1, \ldots, x_n) \wedge Q(y_1, \ldots, y_m) \\ &P(x_1, \ldots, x_n) \vee Q(y_1, \ldots, y_m) \\ &P(x_1, \ldots, x_n) \implies Q(y_1, \ldots, y_m) \\ &P(x_1, \ldots, x_n) \iff Q(y_1, \ldots, y_m) \\ \neg &P(x_1, \ldots, x_n) \end{aligned}$$

(primitívne) rekurzívne.

Poznámka 4.11. Vo vete 4.10 sa nežiada, aby $\{x_1, \ldots, x_n\}$, $\{y_1, \ldots, y_m\}$ boli disjunktné množiny premenných. Preto ak je aj $P(x_1, \ldots, x_n)$ n-árny predikát a $Q(y_1, \ldots, y_m)$ m-árny predikát, nemusí byť napríklad $P(x_1, \ldots, x_n) \implies Q(y_1, \ldots, y_m)$ $(n+m)$-árny predikát.

Dôkaz vety 4.10: Vetu zrejme stačí dokázať pre logické spojky $\neg$, $\vee$, lebo ostatné logické spojky už možno pomocou nich vyjadriť. Ak sú $\chi_P(x_1, \ldots, x_n)$, $\chi_Q(y_1, \ldots, y_m)$ charakteristické funkcie predikátov $P(x_1, \ldots, x_n)$, $Q(y_1, \ldots, y_m)$, sú $\chi_P(x_1, \ldots, x_n) \cdot \chi_Q(y_1, \ldots, y_m)$, $1 \dot{-} \chi_P(x_1, \ldots, x_n)$ charakteristické funkcie predikátov $P(x_1, \ldots, x_n) \vee Q(y_1, \ldots, y_m)$, $\neg P(x_1, \ldots, x_n)$. Z predpokladov vety vyplýva, že tieto charakteristické funkcie, a teda aj príslušné predikáty, sú (primitívne) rekurzívne. □

Ak je $P(x, y)$ (primitívne) rekurzívny predikát, tak, všeobecne povedané, predikáty

$$(\exists x) P(x, y), \qquad (\forall x) P(x, y)$$

nie sú (primitívne) rekurzívne. Platí však nasledujúca veta:

Veta 4.20. Nech $P(y, x_1, \ldots, x_n)$ je primitívne rekurzívny predikát, $f(x_1, \ldots, x_n)$ je primitívne rekurzívna funkcia a nech pre každé $x_1, \ldots, x_n \in \mathbb{N}$ existuje také $y \leq f(x_1, \ldots, x_n)$, že platí $P(y, x_1, \ldots, x_n)$. Potom funkcia g,

$$g(x_1, \ldots, x_n) = \min_y P(y, x_1, \ldots, x_n) \tag{4.20.1}$$

je primitívne rekurzívna.

Dôkaz: Označme $\chi_P(y, x_1, \ldots, x_n)$ charakteristickú funkciu predikátu $P(y, x_1, \ldots, x_n)$. Potom platí

$$g(x_1, \ldots, x_n) = \min_y (\chi_P(y, x_1, \ldots, x_n) = 0)$$

Teraz stačí použiť vetu 3.26. □

Veta 4.20 platí, aj keď z nej všade vynecháme slová „primitívny", „primitívne". V takom prípade platí však i silnejšie tvrdenie:

Veta 4.21. Nech $P(y, x_1, \ldots, x_n)$ je rekurzívny predikát a nech pre všetky $x_1, \ldots, x_n \in \mathbb{N}$ existuje také $y \in \mathbb{N}$, že platí $P(y, x_1, \ldots, x_n)$. Potom funkcia (4.20.1) je rekurzívna.

Dôkaz tejto vety prenechávame čitateľovi. Prenechávame mu i dôkaz vety:

Veta 4.22. Ak $P(y, x_1, \ldots, x_n)$ je rekurzívny predikát, tak čiastočná funkcia (4.20.1) je čiastočne rekurzívna.

Budeme sa teraz zaoberať vzťahom medzi vlastnosťami funkcie a vlastnosťami jej grafu.

Cvičenie 4.23. (a) Vyjadrite čiastočnú funkciu f pomocou operácií skladania funkcií a minimalizácie, charakteristickej funkcie χ grafu čiastočnej funkcie f a primitívne rekurzívnych funkcií.

(b) Vyjadrite charakteristickú funkciu χ grafu funkcie f pomocou operácie skladania funkcií, funkcie f a primitívne rekurzívnych funkcií.

Na základe výsledkov tohto cvičenia možno vysloviť nasledujúce dve vety:

Veta 4.24. Graf primitívne rekurzívnej funkcie je primitívne rekurzívna množina.

Veta 4.25. Nech f je (totálna) funkcia. Potom funkcia f je rekurzívna práve vtedy, keď graf funkcie f je rekurzívna množina.

Poznámka 4.26. Ako čitateľ hneď uvidí z nasledujúcich cvičení, veta 4.24 sa nedá obrátiť a vo vete 4.25 nie je možné vynechať predpoklad, že funkcia f je totálna. Pri riešení týchto cvičení bude už potrebný výsledok cvičenia 3.82.

Cvičenie 4.27. Nájdite čiastočnú funkciu f, ktorá nie je totálna a ktorej graf je primitívne rekurzívny!

Cvičenie 4.28. Dokážte, že existuje taká primitívne rekurzívna funkcia $f(x, y)$, že

$$g(y) = \mu_x (f(x, y) = 0)$$

je rekurzívna, ale nie je primitívne rekurzívna funkcia!

Cvičenie 4.29. Dokážte, že existuje funkcia $g(x)$, ktorá nie je primitívne rekurzívna a ktorej graf je primitívne rekurzívny.

Cvičenie 4.30. Dokážte, že existuje binárna rekurzívna funkcia univerzálna pre množinu všetkých unárnych primitívne rekurzívnych funkcií, ktoré nadobúdajú hodnoty len z množiny $\{0, 1\}$!

Nasledujúca veta, ktorá vyplýva z uvedených cvičení, ukazuje, že rozlišovanie rekurzívnosti a primitívnej rekurzívnosti nebolo zbytočné, ale že ide (i v prípade množín a predikátov) o naozaj rozličné pojmy.

Veta 4.31. (a) Existuje rekurzívna funkcia, ktorá nie je primitívne rekurzívna.

(b) Existuje rekurzívna množina, ktorá nie je primitívne rekurzívna.

(c) Existuje rekurzívny predikát, ktorý nie je primitívne rekurzívny.

Zavedieme teraz pojem rekurzívnej spočítateľnosti, ktorý je v istom zmysle analógiou pojmu čiastočnej rekurzívnosti. O prípade $n = 0$ v nasledujúcich definíciách platí to isté, čo sme uviedli v poznámke 4.3 k definíciám 4.1 a 4.2.

Veta 4.41. Ak $M \subseteq \mathbb{N}^{n+1}$ je rekurzívne spočítateľná množina, tak aj množina

$$M_1 = \{(x_1, \ldots, x_n) \in \mathbb{N}^n \mid (\exists z)\,((x_1, \ldots, x_n, z) \in M)\}$$

je rekurzívne spočítateľná.

Dôkaz: Nech f je $(n+2)$-árna rekurzívna funkcia priradená množine M. Potom rekurzívna funkcia $g(y, x_1, \ldots, x_n) = f(l(y), x_1, \ldots, x_n, r(y))$ je priradená množine M_1, a teda množina M_1 je rekurzívne spočítateľná. □

Veta 4.42. Nech $M \subseteq \mathbb{N}^n$ je rekurzívne spočítateľná množina a nech π je permutácia množiny $\{1, 2, \ldots, n\}$. Potom aj $M_1 = \{(x_1, \ldots, x_n) \in \mathbb{N}^n \mid (x_{\pi(1)}, \ldots, x_{\pi(n)}) \in M\}$ je rekurzívne spočítateľná.

Dôkaz: Nech f je $(n+1)$-árna rekurzívna funkcia priradená množine M. Potom $(n+1)$-árna funkcia $f_1(y, x_1, \ldots, x_n) = f(y, x_{\pi(1)}, \ldots, x_{\pi(n)})$ je rekurzívna a je priradená množine M_1. Preto je aj M_1 rekurzívne spočítateľná. □

Veta 4.43. Neprázdna množina $M \subseteq \mathbb{N}$ je rekurzívne spočítateľná práve vtedy, keď M je množinou hodnôt nejakej (unárnej) rekurzívnej funkcie.

Dôkaz: Nech M je množinou hodnôt rekurzívnej funkcie f; môžeme zariadiť, že f je unárna. Potom binárna funkcia $g(x, y) = |y - f(x)|$ je rekurzívna funkcia priradená množine M. Obrátene, nech M je rekurzívne spočítateľná a g je k nej priradená rekurzívna funkcia. Ďalej nech $y_0 \in M$. Potom funkcia

$$h(x, y) = \begin{cases} y, & \text{ak } g(x, y) = 0 \\ y_0, & \text{inak} \end{cases}$$

je rekurzívna a jej množina hodnôt je M; ak chceme nájsť unárnu funkciu, položíme

$$h_1(x) = h(l(x), r(x)) \quad \square$$

Veľmi dôležitým výsledkom o rekurzívne spočítateľných množinách je nasledujúca Postova veta.

Veta 4.44. Ak $M \subseteq N^n$ a obe množiny M, $\mathbb{N}^n \setminus M$ sú rekurzívne spočítateľné, tak množina M je rekurzívna.

Dôkaz: Nech f, g sú $(n+1)$-árne rekurzívne funkcie priradené množinám M, $\mathbb{N}^n \setminus M$. Definujme $h(x_1, \ldots, x_n) = \mu_y(f(y, x_1, \ldots, x_n) \cdot g(y, x_1, \ldots, x_n) = 0)$. Použitá minimalizácia je regulárna, a preto funkcia h je rekurzívna. Charakteristickú funkciu χ_M množiny M možno teraz vyjadriť vzorcom

$$\chi_M(x_1, \ldots, x_n) = \operatorname{sg}(f(h(x_1, \ldots, x_n), x_1, \ldots, x_n)) \tag{4.44.1}$$

teda χ_M je rekurzívna funkcia a M je rekurzívna množina. □

Cvičenie 4.45. Ak sú množiny $M_1, M_2 \subseteq \mathbb{N}$ rekurzívne spočítateľné a množiny $M_1 \cap M_2$, $M_1 \cup M_2$ sú rekurzívne, tak sú aj množiny M_1 a M_2 rekurzívne. Dokážte!

Veta 4.46. Čiastočná charakteristická funkcia rekurzívne spočítateľnej množiny je čiastočne rekurzívna.

Dôkaz: Nech $M \subseteq \mathbb{N}^n$ má priradenú rekurzívnu $(n+1)$-árnu funkciu f. Potom pre čiastočnú charakteristickú funkciu $\varkappa_M$ množiny M platí

$$\varkappa_M(x_1, \ldots, x_n) = 0.\mu_y(f(y, x_1, \ldots, x_n) = 0) \tag{4.46.1}$$

teda $\varkappa_M$ je čiastočne rekurzívna. □

Poznámka 4.47. Obrátenie vety 4.46 dokážeme v kapitole 6.

Cvičenie 4.48. Ak sú množiny $M_1, M_2 \subseteq \mathbb{N}$ rekurzívne spočítateľné, sú aj množiny $\{x + y \mid x \in \in M_1, y \in M_2\}$, $\{x.y \mid x \in M_1, y \in M_2\}$ rekurzívne spočítateľné. Dokážte!

Veta 4.49. Ak graf čiastočnej funkcie f je rekurzívne spočítateľná množina, tak f je čiastočne rekurzívna funkcia.

Veta 4.56. Ak $P(x_1, \ldots, x_m)$ je rekurzívne spočítateľný predikát a

$$f_1(y_{1,1}, \ldots, y_{1,n_1}), \ldots, f_m(y_{m,1}, \ldots, y_{m,n_m})$$

sú rekurzívne funkcie, tak predikát

$$P\big(f_1(y_{1,1}, \ldots, y_{1,n_1}), \ldots, f_m(y_{m,1}, \ldots, y_{m,n_m})\big) \tag{4.56.1}$$

je rekurzívne spočítateľný.

Dôkaz: je obdobný ako dôkaz vety 4.14. Ani v tejto vete nie je potrebné predpokladať, že premenné $y_{1,1}, \ldots, y_{1,n_1}, \ldots, y_{m,1}, \ldots, y_{m,n_m}$ sú po dvoch rôzne. □

Poznámka 4.57. Keby sme vo vete 4.56 chceli nahradiť predpoklad, že $f_1, \ldots, f_m$ sú rekurzívne funkcie, predopkladom, že sú iba čiastočne rekurzívne, museli by sme predikát (4.56.1) pokladať za pravdivý len vtedy, keď sú všetky hodnoty $a_i = f_i(y_{i,1}, \ldots, y_{i,n_i})$, $i = 1, \ldots, m$ definované; tým by nám však vznikli rôzne ťažkosti, napríklad v súvislosti s naším chápaním rovnosti.

Veta 4.58. Ak sú predikáty $P(x_1, \ldots, x_n)$, $\neg P(x_1, \ldots, x_n)$ rekurzívne spočítateľné, je predikát $P(x_1, \ldots, x_n)$ rekurzívny.

Cvičenie 4.59. Ak sú predikáty $P(x_1, \ldots, x_n) \wedge Q(x_1, \ldots, x_n)$, $P(x_1, \ldots, x_n) \vee Q(x_1, \ldots, x_n)$ rekurzívne a predikáty $P(x_1, \ldots, x_n) \implies Q(x_1, \ldots, x_n)$, $Q(x_1, \ldots, x_n) \implies P(x_1, \ldots, x_n)$ rekurzívne spočítateľné, sú predikáty $P(x_1, \ldots, x_n)$, $Q(x_1, \ldots, x_n)$ rekurzívne. Dokážte!

alebo

$$(q_i; a_0, a_1, \ldots, a_n), \tag{5.5.2}$$

kde n zvolíme tak, aby $a_{n+1} = a_{n+2} = \cdots = 0$.

Poznámka 5.6. Ak je (5.5.1) stavom registrového stroja Z, tak q_i znamená (v predchádzajúcom neformálnom opise) momentálny vnútorný stav stroja Z a čísla a_0, a_1, a_2, ... znamenajú v tomto poradí obsahy registrov R_0, R_1, R_2, ...; ak niektoré $a_n = 0$, môže to znamenať, že obsah registra R_n je nula, alebo že stroj Z nemá register R_n. Označenia $(q_i; a_0, a_1, \ldots, a_n)$, $(q_i; a_0, a_1, \ldots, a_n, 0)$ zodpovedajú tomu istému stavu.

Nasledujúca definícia opisuje jeden krok výpočtu registrového stroja.

Definícia 5.7. Nech Z je registrový stroj. Budeme písať

$$(q_i; a_0, a_1, a_2, \ldots) \xrightarrow{Z} (q_j; b_0, b_1, b_2, \ldots), \tag{5.7.1}$$

ak existuje také $n \in \mathbb{N}$, že pre všetky $x \in \mathbb{N}$, $x \neq n$ platí $a_x = b_x$ a existuje také $y \in \mathbb{N}$, že je splnená jedna z nasledujúcich podmienok:

$$(q_i R_n P q_j) \in Z \text{ a } b_n = a_n + 1 \tag{5.7.2}$$

$$(q_i R_n M q_j) \in Z \text{ a } b_n = a_n \dot{-} 1 \tag{5.7.3}$$

$$(q_i R_n q_j q_y) \in Z \text{ a } b_n = a_n \neq 0 \tag{5.7.4}$$

$$(q_i R_n q_y q_j) \in Z \text{ a } b_n = a_n = 0 \tag{5.7.5}$$

Dohoda 5.8. Ak nebude hroziť nedorozumenie, budeme z označenia (5.7.1) vynechávať „Z“.

Príklad 5.9. Nech $Z = \{(q_1 R_1 q_2 q_3), (q_2 R_2 M q_1), (q_3 R_0 P q_1)\}$. Podľa definície 5.3 je Z registrový stroj. Podľa definície 5.7 platí napríklad:

$$(q_1; 1, 0, 1, 0, 2) \xrightarrow{Z} (q_3; 1, 0, 1, 0, 2)$$

$$(q_1; 0, 1, 0, 0, 0) \xrightarrow{Z} (q_2; 0, 1) \qquad \text{(pozri dohodu 5.5)}$$

$$(q_2; 0, 0, 1) \xrightarrow{Z} (q_1; 0, 0, 0)$$

$$(q_3; 4, 0, 3) \xrightarrow{Z} (q_1; 5, 0, 3)$$

$$(q_1; 5, 0, 3) \xrightarrow{Z} (q_3; 5, 0, 3)$$

$$(q_2; 5, 0, 3) \xrightarrow{Z} (q_1; 5, 0, 2)$$

Cvičenie 5.10. Nájdite všetky také stavy X, aby pre stroj Z z predchádzajúceho príkladu platilo:

(a) $(q_1; 1, 0, 2) \xrightarrow{Z} X$

(b) $(q_2; 1, 0, 2) \xrightarrow{Z} X$

(c) $(q_3; 1, 0, 2) \xrightarrow{Z} X$

(d) $X \xrightarrow{Z} (q_1; 1, 0, 2)$

(e) $X \xrightarrow{Z} (q_2; 1, 0, 2)$

(f) $X \xrightarrow{Z} (q_3; 1, 0, 2)$

Poznámka 5.11. V neformálnom opise zrejme zodpovedá inštrukcia $(q_i R_j q_m q_n)$ podmienenému skoku podľa podmienky, či sa obsah registra R_j rovná nule alebo nie. Inštrukcia $(q_i R_j P q_k)$ zodpovedá pripočítaniu jednotky k obsahu registra R_j a inštrukcia $(q_i R_j M q_k)$ zodpovedá odčítaniu jednotky od registra R_j, pokiaľ je tento obsah rôzny od nuly.

teda špeciálne

$$(q_1; 0, x) \overset{Z}{\Longmapsto} (q_0; x, 0)$$

Teda pre každé x platí $\Phi_Z^1(x) = x$, a teda $\Phi_Z^1 = I_1^1$. Pre tento istý stroj Z zrejme platí aj $\Phi_Z^n = I_1^n$ pre každé $n \in \mathbb{N}$, $n \neq 0$ a $\Phi_Z^0 = 0$. Obdobné úvahy nebudeme odteraz rozpisovať tak podrobne.

Cvičenie 5.19. Nájdite pre každé $m \in \mathbb{N}$, $m \neq 0$ taký registrový stroj Z, že pre všetky $n \in \mathbb{N}$, $n \geq m$ platí $\Phi_Z^n = I_m^n$!

Príklad 5.20. Stroj Z, ktorý počíta funkciu s, t.j. stroj Z s vlastnosťou $\Phi_Z^1 = s$, je napríklad $Z = \{(q_1 R_1 q_2 q_4), (q_2 R_1 M q_3), (q_3 R_0 P q_1), (q_4 R_0 P q_0)\}$.

Cvičenie 5.21. Nájdite stroj Z, ktorý počíta funkciu s a ktorý obsahuje len tri inštrukcie!

Príklad 5.22. Uvažujme registrový stroj $Z = \{(q_1 R_1 q_2 q_3), (q_2 R_0 P q_0)\}$. Pre $x \neq 0$ zrejme platí $\Phi_Z^1(x) = 1$. Skúmajme $\Phi_Z^1(0)$. Výpočet stroja Z zo stavu $(q_1; 0, 0)$ sa končí stavom $(q_3; 0, 0)$, a preto je $\Phi_Z^1(0)$ nedefinované. Čiastočnú funkciu Φ_Z^1 možno vyjadriť napríklad vzorcom $\Phi_Z^1(x) = sg(x) - \overline{sg}(x)$. Iný stroj, ktorý počíta tú istú čiastočnú funkciu, je $\{(q_1 R_1 q_2 q_1), (q_2 R_0 P q_0)\}$. Výpočet tohto stroja zo stavu $(q_1; 0, 0)$ je nekonečný.

Veta 5.23. Ku každému registrovému stroju Z existuje taký registrový stroj Y, že pre každé $n \in \mathbb{N}$ platí $\Phi_Y^n = \Phi_Z^n$ a pre všetky $x_1, \ldots, x_n \in \mathbb{N}$ je $\Phi_Y^n(x_1, \ldots, x_n)$ definované práve vtedy, keď je výpočet stroja Y zo stavu $(q_1; 0, x_1, \ldots, x_n)$ konečný.

Cvičenie 5.24. Vyslovte predpis, ktorý bude každému registrovému stroju Z jednoznačne a efektívne priraďovať stroj Y s vlastnosťami uvedenými v predchádzajúcej vete! Predpokladajme pritom, že stroj Z je daný tak, že sú vypísané všetky jeho inštrukcie a rovnakým spôsobom tiež určite stroj Y!

Cvičenie 5.25. Pokúste sa nájsť predpis s vlastnosťami z predchádzajúceho cvičenia, ktorý by mal navyše vlastnosť, že stroju Z priraďoval stroj $Y \neq Z$ len vtedy, keď stroj $Y = Z$ nevyhovuje podmienkam vety 5.23!

Ďalej v tejto kapitole budeme musieť zostrojovať rôzne registrové stroje buď priamo, alebo z iných, už zostrojených registrových strojov. Vtedy bude často výhodné znázorňovať registrové stroje graficky pomocou blokových schém spôsobom, ktorý teraz opíšeme.

Definícia 5.26. Nech je daný registrový stroj Z. Jeho *blokovú schému* dostaneme tak, že:

(1) Každej inštrukcii stroja Z tvaru $(q_i R_j q_m q_n)$ priradíme šesťuholník, do ktorého vpíšeme „$R_j = 0$“.

(2) Každej inštrukcii stroja Z tvaru $(q_i R_j M q_k)$, resp. $(q_i R_j P q_k)$ priradíme obdĺžnik, do ktorého vpíšeme „$R_j\ M$“, resp. „$R_j\ P$“.

(3) Ak má inštrukcia, ku ktorej patrí obdĺžnik A, na štvrtom mieste rovnaký vnútorný stav ako má inštrukcia patriaca k obdĺžniku alebo šesťuholníku B na prvom mieste, spojíme A, B šípkou od A k B.

(4) Ak má inštrukcia, ku ktorej patrí šesťuholník A, na treťom, resp. štvrtom mieste ten vnútorný stav, ktorým sa začína inštrukcia patriaca k obdĺžniku alebo šesťuholníku B, spojíme A, B šípkou od A k B; jej začiatok pri A označíme písmenom „n“, resp. písmenom „a“ (od slov „nie“, „áno“).

(5) Ak obsahuje stroj Z inštrukciu začínajúcu sa vnútorným stavom q_1, doplníme blokovú schému dvojitým oválom, do ktorého vpíšeme slovo „Štart“. (Takýto ovál budeme nazývať začiatočným oválom.) Potom vedieme šípku od tohto oválu k šesťuholníku alebo obdĺžniku patriacemu k tejto inštrukcii.

(6) Ak má stroj Z vnútorný stav q_0 na treťom alebo štvrtom mieste niektorej svojej inštrukcie a nemá ho na prvom mieste žiadnej svojej inštrukcie, doplníme blokovú schému dvojitým oválom, do ktorého vpíšeme slovo „Stop“. Potom vedieme šípky od všetkých obdĺžnikov a šesťuholníkov patriacich k inštrukciám, na ktorých treťom alebo štvrtom mieste je vnútorný stav q_0, k tomuto oválu. (Tento ovál budeme nazývať koncovým oválom.) Začiatky šípok od šesťuholníka označíme písmenami „n“ resp. „a“ podľa toho, či sa vnútorný stav q_0 nachádza na treťom alebo na štvrtom mieste príslušnej inštrukcie.

Dohoda 5.30. (a) V blokovej schéme registrového stroja nemusia byť vyznačené vnútorné stavy.

(b) Ak z niektorého obdĺžnika nevychádza šípka alebo z niektorého šesťuholníka nevychádzajú šípky, možno z neho namiesto chýbajúcej šípky viesť čiaru ukončenú malým čiernym krúžkom. (Tým sa zreteľne vyznačí, že príslušné šípky neboli v schéme „zabudnuté".)

(c) Bloková schéma môže obsahovať viac koncových oválov (aby sme sa vyhli dlhým čiaram vedúcim k jedinému takémuto oválu).

(d) Ak vedie šípka z obdĺžnika A do obdĺžnika B, možno túto šípku vynechať a nakresliť obdĺžnik B pod obdĺžnik A tak, aby sa jednou stranou dotýkali.

(e) Taká časť blokovej schémy, ktorá sa skladá z nejakej množiny X obdĺžnikov a šesťuholníkov a všetkých šípok medzi nimi, že v nej existuje jeden obdĺžnik alebo šesťuholník A, do ktorého smerujú všetky šípky začínajúce sa v obdĺžnikoch alebo šesťuholníkoch nepatriacich do X, môže byť nahradená dvojitým obdĺžnikom, vo vnútri ktorého vyznačíme činnosť tejto časti. Všetky šípky smerujúce do A, ktoré sa nezačínajú v prvkoch množiny X, nahradíme šípkami smerujúcimi do dvojitého obdĺžnika (ich začiatky nezmeníme). Ku každému obdĺžniku alebo šesťuholníku nepatriacemu do X, ku ktorému smerovala aspoň jedna šípka od niektorého prvku množiny X, bude smerovať (jedna) šípka z dvojitého obdĺžnika. Ak z dvojitého obdĺžnika smeruje viac šípok, vyznačíme vhodným spôsobom, kedy treba po ktorej z nich postupovať. Ak vychádzala z niektorého prvku množiny X šípka smerujúca ku koncovému oválu, bude takáto šípka vychádzať i z dvojitého obdĺžnika; v prípade potreby bude vyznačená aj podmienka, kedy sa má podľa tejto šípky postupovať.

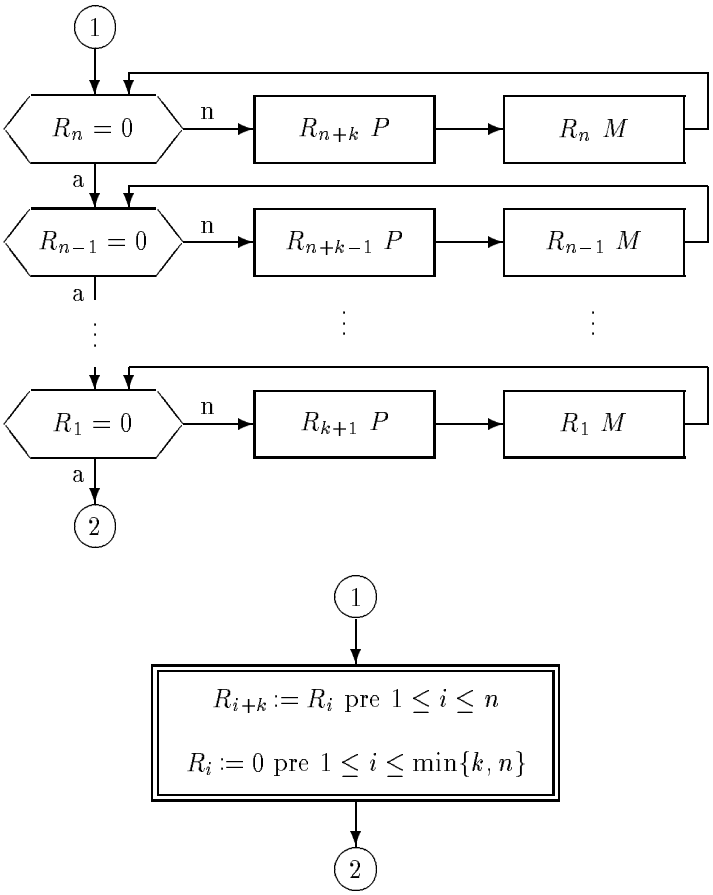
(f) Ak časť blokovej schémy, ktorú nahradzujeme dvojitým obdĺžnikom podľa bodu (e), možno dostať tak, že z nejakého registrového stroja vynecháme začiatočný a koncový ovál a šípky z neho vychádzajúce, resp. doňho vchádzajúce, vpisujeme do dvojitého obdĺžnika miesto opisu činnosti len označenie tohto stroja. (V tomto prípade vychádza z dvojitého obdĺžnika jediná šípka.)

Príklad 5.31. Bod (e) dohody 5.30 neurčuje presne, ako treba opisovať činnosť časti schémy, ktorú nahradzujeme dvojitým obdĺžnikom. Uvedieme aspoň tri ukážky úseku schémy a jeho nahradenia. Používame pritom logické spojky, obvyklé matematické symboly a znak „:=" z PASCALu. Pre obsah registra používame to isté označenie ako pre samotný register. Krúžkami s rovnakými číslami sú vždy označené vzájomne si zodpovedajúce začiatky a konce šípok v úplnom úseku schémy a v jeho skrátenom označení. V prvej ukážke (obr. 5.2) ide o zisťovanie rovnosti obsahov registrov R_1, R_2. Tieto obsahy sa majú zachovať pre ďalší výpočet. To sa dá urobiť len s pomocou ďalšieho registra (zvolili sme register R_5), ktorého obsah sa však nezachová; my sme tento register pre určitosť vynulovali. V druhej ukážke (obr. 5.3) ide o presun obsahov registrov $R_1, \ldots, R_n$, pričom predpokladáme, že na začiatku sú obsahy všetkých registrov $R_{n+1}, \ldots, R_{n+k}$ nulové. Všimnime si, že v prípade $k < n$ je potrebné presun začať od registra R_n, a nie od registra R_k. V tretej ukážke (obr. 5.4) ide o zložitejšiu logickú podmienku. Obsahy registrov R_1, R_2, R_3, sa majú zachovať pre ďalší výpočet. Vychádzajúce šípky sú vyznačené tým istým spôsobom, ako vyznačujeme šípky vychádzajúce zo šesťuholníkov. Poznamenajme ešte, že do dvojitých obdĺžnikov nebudeme vždy vyznačovať „vedľajšie efekty" spôsobované príslušnou činnosťou schémy. Takým „vedľajším efektom" môže byť napríklad vynulovanie registra R_5 v prvej ukážke alebo vynulovanie registrov $R_1, \ldots, R_{min(k,n)}$ v druhej ukážke.

Cvičenie 5.32. Úseky blokových schém z obrázkov 5.2 a 5.4 sa dajú podstatne zjednodušiť, ak nežiadame, aby sa zachovali pôvodné obsahy registrov. Nakreslite tieto zjednodušené úseky!

Cvičenie 5.33. Nakreslite schémy registrových strojov, ktoré počítajú nasledovné funkcie:

- $f_1(x, y) = 3x + 2y$
- $f_2(x, y) = x.y$
- $f_3(x, y) = x \text{ DIV } y$
- $f_4(x, y) = \mathrm{c}(x, y)$
- $f_5(x, y) = \lfloor\sqrt{x}\rfloor + \lfloor\sqrt{y}\rfloor$
- $f_6(x, y) = x^y$



OBR. 5.3.

vyhýbať dvojnásobnému použitiu slov „čiastočne“, „čiastočné“ v jednom termíne a budeme hovoriť „čiastočne vypočítateľná funkcia“ namiesto „čiastočne vypočítateľná čiastočná funkcia“.

Budeme teraz skúmať uzavretosť množiny všetkých (čiastočne) vypočítateľných funkcií vzhľadom na niektoré operácie. Najprv dokážeme, prípadne necháme čitateľa dokázať niekoľko pomocných tvrdení.

Veta 5.38. Ku každým dvom registrovým strojom Z_1, Z_2 existuje taký registrový stroj Y, že pre každé dve postupnosti $(a_0, a_1, a_2, \dots)$, $(b_0, b_1, b_2, \dots)$ prirodzených čísel platí

$$(q_1; a_0, a_1, \dots) \overset{Y}{\longmapsto} (q_0; b_0, b_1, \dots)$$

práve vtedy, keď existuje taká postupnosť $(c_0, c_1, c_2, \dots)$, že platí

$$(q_1; a_0, a_1, \dots) \overset{Z_1}{\longmapsto} (q_0; c_0, c_1, \dots) \quad \text{a} \quad (q_1; c_0, c_1, \dots) \overset{Z_2}{\longmapsto} (q_0; b_0, b_1, \dots)$$

Dôkaz: Nech n je najvyšší index vnútorného stavu, ktorý sa vyskytuje v inštrukciách stroja Z_1. Ak $n = 0$, položme $Y = \emptyset$. Inak vytvorme stroj Z_1', ktorý vznikne zo stroja Z_1 tak, že vo všetkých jeho inštrukciách zameníme vnútorný stav q_0 vnútorným stavom q_{n+1}. Zo stroja Z_2 vytvoríme stroj Z_2' tak, že všetky nenulové indexy vnútorných stavov v jeho inštrukciách zvýšime o číslo n. Potom položíme $Y = Z_1' \cup Z_2'$. Dokážeme teraz, že stroj Y má vo vete požadovanú vlastnosť. Pre $n = 0$ platí $Y = \emptyset$, a teda nikdy neplatí $(q_1; a_0, a_1, \dots) \overset{Y}{\longmapsto} (q_0; b_0, b_1, \dots)$. Stroj Z_1 neobsahuje inštrukciu začínajúcu sa q_1, a teda nikdy neplatí ani $(q_1; a_0, a_1, \dots) \overset{Z_1}{\longmapsto} (q_0; c_0, c_1, \dots)$. Nech teraz $n \neq 0$ a nech $A_0 = (q_1; a_0, a_1, \dots)$, $A_2, \dots, A_s = (q_0; b_0, b_1, \dots)$ je výpočet stroja Y. Označme

Cvičenie 5.42. Dokážte, že pre každé k, $n \in \mathbb{N}$, $k > n > 0$ existuje taký registrový stroj Z, že pre všetky $x_1, \ldots, x_n \in \mathbb{N}$ platí

$$(q_1; 0, x_1, \ldots, x_n) \overset{Z}{\longmapsto} (q_0; 0, x_1, \ldots, x_n, 0, \ldots, 0, x_1, \ldots, x_n), \tag{5.42.1}$$

kde predpokladáme, že druhá skupina $x_1, \ldots, x_n$ sa nachádza na miestach patriacich k registrom $R_{k+1}, \ldots, R_{k+n}$.

Veta 5.43. Pre každý registrový stroj Z a každé $n \in \mathbb{N}$ existuje taký registrový stroj Y, že $\Phi^n_Z = \Phi^n_Y$ a pre všetky $x_1, \ldots, x_n$, $y \in \mathbb{N}$ platí $\Phi^n_Y(x_1, \ldots, x_n) = y$ práve vtedy, keď

$$(q_1; 0, x_1, \ldots, x_n) \overset{Y}{\longmapsto} (q_0; y, x_1, \ldots, x_n)$$

Dôkaz: Konštrukciu stroja Y z daného stroja Z a čísla n prenechávame čitateľovi, podávame mu však návrh, ako môže stroj Y pracovať. Stroj Y môže napríklad najprv uložiť argumenty $x_1, \ldots, x_n$ aj do nejakých nových registrov (podľa cvičenia 5.42), potom počítať ako stroj Z, ďalej vynulovať všetky už nepotrebné registre a nakoniec obnoviť obsahy registrov $R_1, \ldots, R_n$. □

Príklad 5.44. K registrovým strojom Z_1, Z_2 zostrojíme taký stroj Y, že $\Phi^1_Y = \mathcal{S}^2(\Phi^1_{Z_2}, \Phi^1_{Z_1})$, t.j. stroj, ktorý počíta unárnu čiastočnú funkciu zloženú z funkcií $\Phi^1_{Z_2}, \Phi^1_{Z_1}$. Najprv zostrojíme podľa vety 5.43 taký stroj Z_1', že $\Phi^1_{Z_2} = \Phi^1_{Z_1'}$ a že pre všetky $x, y \in \mathbb{N}$ platí $\Phi^1_{Z_1'}(x) = y$ práve vtedy, keď $(q_1; 0, x) \overset{Z_1'}{\longmapsto} (q_0; y, x)$. Nech n je najvyšší index vnútorného stavu v inštrukciách stroja Z_1'. Nahraďme všade v stroji Z_1' vnútorný stav q_0 vnútorným stavom q_{n+1} a pridajme k nemu inštrukcie $(q_{n+1}R_1q_{n+2}q_{n+3})$, $(q_{n+2}R_1Mq_{n+1})$, $(q_{n+3}R_0q_{n+4}q_0)$, $(q_{n+4}R_0Mq_{n+5})$, $(q_{n+5}R_1Pq_{n+3})$. Stroj Z_1'', ktorý takto vznikne zo stroja Z_1' má vlastnosť: pre všetky $x, y \in \mathbb{N}$ $\Phi^1_{Z_1}(x) = y$ práve vtedy, keď $(q_1; 0, x) \overset{Z_1''}{\longmapsto} (q_0; 0, y)$. Teraz stačí zo strojov Z_1'', Z_2 zostrojiť stroj Y podľa vety 5.38.

Cvičenie 5.45. Nakreslite blokovú schému stroja Y z predchádzajúcej vety (použite pritom dvojité obdĺžniky so Z_1, Z_2)!

Cvičenie 5.46. Pre každé m, $n \in \mathbb{N}$, $n \neq 0$ a ľubovoľné registrové stroje Z, $Z_1, \ldots, Z_n$ existuje taký registrový stroj Y, že $\Phi^m_Y = \mathcal{S}^{n+1}(\Phi^n_Z, \Phi^m_{Z_1}, \ldots, \Phi^m_{Z_n})$. Dokážte!

Dôsledkom tohto cvičenia je:

Veta 5.47. (Čiastočná) funkcia, ktorá vzniká z (čiastočne) vypočítateľných funkcií operáciou skladania (čiastočných) funkcií, je (čiastočne) vypočítateľná.

Z tejto vety a cvičenia 5.19 vyplýva:

Veta 5.48. Množina všetkých (čiastočne) vypočítateľných funkcií je klon (čiastočných) funkcií na množine $\mathbb{N}$.

Príklad 5.49. Nech Z je registrový stroj a nech Φ^1_Z je totálna funkcia. Nech funkcia f vzniká z funkcie Φ^1_Z iteráciou, t.j. $f(0) = 0$ a pre všetky $x \in \mathbb{N}$ $f(x+1) = \Phi^1_Z(f(x))$. Nájdime taký registrový stroj Y, že $\Phi^1_Y = f$. Najprv zostrojíme zo stroja Z stroj Z', ktorý nepoužíva register R_1 a pre ktorý platí: pre všetky $x \in \mathbb{N}$ $\Phi^2_{Z'}(0, x) = \Phi^1_Z(x)$ a $(q_1; 0, 0, x) \overset{Z'}{\longmapsto} (q_0; \Phi^1_Z(x), 0, 0)$. Teraz vytvoríme stroj Z'' tak, že všetky indexy vnútorných stavov stroja Z' zväčšíme o 6. Potom môžeme povedať, že $Y = Z'' \cup \{(q_1R_1q_2q_0), (q_2R_1Mq_3), (q_3R_0q_4q_7), (q_4R_0Mq_5), (q_5R_2Pq_3), (q_6R_1q_1q_1)\}$.

Cvičenie 5.50. (a) Opíšte podrobne zostrojenie stroja Z' zo stroja Z!
(b) Nakreslite blokovú schému stroja Y!
(c) Dokážte, že stroj Y z príkladu 5.49 má požadovanú vlastnosť!

Veta 5.51. Pre každé $n \in \mathbb{N}$ a každé dva registrové stroje Z_1, Z_2 existuje taký registrový stroj Z, že čiastočná funkcia Φ^{n+1}_Y vzniká primitívnou rekurziou z čiastočných funkcií $\Phi^{n+2}_{Z_1}$, $\Phi^n_{Z_2}$, t.j. $\Phi^{n+1}_Y = \mathcal{R}(\Phi^{n+2}_{Z_1}, \Phi^n_{Z_2})$.

Veta 5.55. Ku každému prirodzenému číslu n a každému registrovému stroju Z existuje taký registrový stroj Y, že Φ_Y^n je čiastočná funkcia, ktorá vzniká z čiastočnej funkcie Φ_Z^{n+1} operáciou minimalizácie, t.j. $\Phi_Y^n = \mathcal{M}(\Phi_Z^{n+1})$.

Dôkaz: Pripomeňme najprv, že

$$\Phi_Y^n(x_1, \ldots, x_n) = \mu_y\big(\Phi_Z^{n+1}(y, x_1, \ldots, x_n) = 0\big)$$

Bloková schéma stroja Y je na obrázku 5.5(b). Z' je stroj, ktorý vznikne zo Z podľa vety 5.43. Dokončenie dôkazu, ktoré je rovnaké ako pri vete 5.51, prenechávame čitateľovi. □

Dôsledkom viet 5.48 , 5.51 , 5.55 a príkladu 5.20 je:

Veta 5.56. Každá (čiastočne) rekurzívna funkcia je (čiastočne) vypočítateľná na registrovom stroji.

Obrátenie tejto vety dokážeme v nasledujúcej kapitole. Teraz ukážeme ešte jednu charakterizáciu rekurzívne spočítateľných množín pomocou registrových strojov.

Definícia 5.57. Budeme hovoriť, že registrový stroj Z *generuje* množinu $A \subseteq \mathbb{N}^n$, ak pre každé $(x_1, \ldots, x_n) \in \mathbb{N}^n$ platí: $(x_1, \ldots, x_n) \in A$ práve vtedy, keď existujú $a, b_1, \ldots, b_k \in \mathbb{N}$ také, že

$$(q_1; 0) \overset{Z}{\Longrightarrow} (q_0; a, x_1, \ldots, x_n, b_1, \ldots, b_k)$$

Rovnako, ako to bolo pri počítaní funkcií, číslo n nie je určené strojom Z, ale musí byť určené osobitne. Ten istý stroj generuje napríklad nejakú množinu usporiadaných dvojíc A_2 i nejakú množinu usporiadaných trojíc A_3. Samozrejme, medzi A_2 a A_3 je istý vzťah.

Neformálne povedané, množinu $A \subseteq \mathbb{N}^n$, ktorú generuje stroj Z, dostaneme tak, že necháme stroj Z počítať s prázdnymi registrami zo začiatočného vnútorného stavu q_1. Zakaždým, keď stroj dosiahne vnútorný stav q_0 zostavíme usporiadanú n-ticu z obsahov registrov $R_1, \ldots, R_n$ a zaradíme ju do množiny A (ak tam ešte nebola zaradená).

Príklad 5.58. Registrový stroj s inštrukciami $\{(q_1R_0Mq_0)$, $(q_0R_1Pq_2)$, $(q_2R_1Pq_3)$, $(q_3R_2Pq_4)$, $(q_4R_2Pq_5)$, $(q_5R_2Pq_0)\}$ generuje množinu všetkých párnych čísel (pre $n = 1$) a súčasne množinu $\{(x, y) \in \mathbb{N}^2 \mid 3x = 2y\}$.

Cvičenie 5.59. Dokážte, že ak nejaký registrový stroj generuje nekonečnú množinu usporiadaných dvojíc, tak generuje aj nekonečnú množinu usporiadaných trojíc. Platí to aj obrátene?

Cvičenie 5.60. Nájdite registrový stroj, ktorý generuje

(a) množinu $\mathbb{N}^2$
(b) množinu $\{(x, y, x + y) \mid x, y \in \mathbb{N}\}$
(c) množinu $\{(x, y, xy) \mid x, y \in \mathbb{N}\}$
(d) množinu $\mathbb{N}^3$

Cvičenie 5.61. Dokážte, že každá rekurzívna množina je generovaná nejakým registrovým strojom.

Veta 5.62. Pre každú rekurzívne spočítateľnú množinu $A \subseteq \mathbb{N}^n$ existuje registrový stroj Z, ktorý ju generuje.

Dôkaz: Nech f je nejaká rekurzívna funkcia priradená množine A a nech g je funkcia definovaná vzťahom $g(x_1, \ldots, x_n, y) = f(y, x_1, \ldots, x_n)$ (len sme vhodne zmenili poradie argumentov). Stroj Z bude pracovať tak, že v registroch $R_1, \ldots, R_n, R_{n+1}$ bude postupne vytvárať všetky usporiadané $(n+1)$-tice prirodzených čísel. Vždy po vytvorení $(n+1)$-tice vypočíta príslušnú hodnotu funkcie g (ako obvykle, argumenty číta v $R_1, \ldots, R_{n+1}$ a výsledok ukladá do R_0). Ak je táto hodnota nulová, prejde stroj cez stav q_0, ak nie, vynuluje R_0 bez prejdenia cez q_0; potom pokračuje ďalej, s obnovou obsahov niektorých registrov, ak je potrebná. Rozpísanú etapu môžu robiť napríklad inštrukcie $(q_6R_0q_7q_0)$, $(q_7R_0Mq_8)$, $(q_8R_0q_7q_9)$ $(q_0R_0q_9q_9)$; vnútorné stavy q_0, q_7, q_8 sa už nebudú vyskytovať v nijakých ďalších inštrukciách stroja Z. □

Veta 5.62 sa dá obrátiť, to ale dokážeme až v kapitole 6.

(b) *Číslom konečnej postupnosti stavov* (špeciálne, číslom konečného výpočtu) $X_0, X_1, \ldots, X_n$ nazveme číslo

$$2^{\mathrm{num}(X_0)} \cdot 3^{\mathrm{num}(X_1)} \cdots \mathrm{p}_n^{\mathrm{num}(X_n)}$$

kde $\mathrm{num}\,(X_i)$ označuje číslo stavu X_i.

Cvičenie 6.10. Nájdite číslo výpočtu stroja z cvičenia 6.4(b) zo stavu $(q_1; 0, 1, 1)$. (Mocniny neroznásobujte; odhadnite počet dekadických cifier výsledku.)

Definícia 6.11. (a) Symbolom $\mathrm{Mprech}\,(z, x, y)$ označíme predikát „stav číslo y vzniká zo stavu číslo x jediným krokom výpočtu stroja číslo z".

(b) Pre každé $n \in \mathbb{N}$ symbolom $\mathrm{Mvýp}^{n+2}\,(y, z, x_1, \ldots, x_n)$ označíme predikát „y je číslom výpočtu stroja číslo z zo stavu $(q_1; 0, x_1, \ldots, x_n)$, pričom tento výpočet končí vnútorným stavom q_0".

(c) Symbolom $\mathrm{mvýp}^{n+2}$ označíme charakteristickú funkciu predikátu $\mathrm{Mvýp}^{n+2}$.

(d) Symbolom $\mathrm{mobs}\,(y)$ označíme obsah registra R_0 v poslednom člene konečnej postupnosti stavov, ktorá má číslo y.

Poznámka 6.12. V bode (d) nie je funkcia mobs úplne definovaná, pretože nie každé číslo y je číslom konečnej postupnosti stavov. V takýchto prípadoch budeme postupovať obdobne ako sme to už urobili v kapitole 3, napríklad pri funkcii $\log_x y$: funkciu pre tieto y ľubovoľne dodefinujeme. Nebudeme to však robiť dopredu, ale požijeme niektorý taký vzorec, ktorý platí v prípadoch pre nás dôležitých, pre všetky prípady. To sa týka aj prípadov (a), (b), (c); tu pre $x = 0$, $y = 0$ alebo $z = 0$ bude predikát nepravdivý.

Veta 6.13. (a) Predikát Mprech a predikáty $\mathrm{Mvýp}^{n+2}$ $(n \in \mathbb{N})$ sú primitívne rekurzívne.

(b) Funkcia mobs a funkcie $\mathrm{mvýp}^{n+2}$ $(n \in \mathbb{N})$ sú primitívne rekurzívne.

Dôkaz: Primitívna rekurzívnosť predikátu Mprech vyplýva z vyjadrenia

$$\begin{aligned}
&\mathrm{Mprech}(z, x, y) \iff \\
&\quad\iff (\exists i, j, k, m \le x + y + z) \\
&\qquad \big(\mathrm{ex}(0, x) = i \ \wedge\ \mathrm{ex}(0, y) = k \ \wedge\ x > 0 \ \wedge\ y > 0 \ \wedge \\
&\qquad \wedge\ (\forall t \le x + y)(t \ne j \implies \mathrm{ex}(t + 1, x) = \mathrm{ex}(t + 1, y)) \ \wedge \\
&\qquad \wedge\ \big((\mathrm{ex}(i, z) = 2\,\mathrm{c}^3(j, k, m) + 2 \ \wedge \\
&\qquad\qquad \wedge\ \mathrm{ex}(j + 1, x) = \mathrm{ex}(j + 1, y) \ \wedge\ \mathrm{ex}(j + 1, x) > 0) \ \vee \\
&\qquad\quad \vee\ (\mathrm{ex}(i, z) = 2\,\mathrm{c}^3(j, m, k) + 2 \ \wedge \\
&\qquad\qquad \wedge\ \mathrm{ex}(j + 1, x) = 0 \ \wedge\ \mathrm{ex}(j + 1, y) = 0) \ \vee \\
&\qquad\quad \vee\ (\mathrm{ex}(i, z) = 4\,\mathrm{c}(j, k) + 1 \ \wedge\ \mathrm{ex}(j + 1, x) \dot{-} 1 = \mathrm{ex}(j + 1, y)) \ \vee \\
&\qquad\quad \vee\ (\mathrm{ex}(i, z) = 4\,\mathrm{c}(j, k) + 3 \ \wedge\ \mathrm{ex}(j + 1, x) + 1 = \mathrm{ex}(j + 1, y))))
\end{aligned} \tag{6.13.1}$$

Na vysvetlenie uveďme, že posledných šesť riadkov po rade zodpovedá tomu, že vykonávaná inštrukcia je $(q_i R_j q_k q_m)$, $(q_i R_j q_m q_k)$, $(q_i R_j M q_k)$, resp. $(q_i R_j P q_k)$; prechádza sa z vnútorného stavu q_i do vnútorného stavu q_k.

Pre každé $n \in \mathbb{N}$ teraz primitívna rekurzívnosť predikátu $\mathrm{Mvýp}^{n+2}$ vyplýva z vyjadrenia

$$\begin{aligned}
&\mathrm{Mvýp}^{n+2}(y, z, x_1, \ldots, x_n) \iff \\
&\quad\iff \big(\mathrm{ex}(0, y) = 2^1 \cdot 3^0 \cdot 5^{x_1} \cdot 7^{x_2} \cdots \mathrm{p}_{n+1}^{x_n} \ \wedge \\
&\qquad \wedge\ (\forall t \le \mathrm{npr}(y) \dot{-} 2)\,\mathrm{Mprech}\,\big(z, \mathrm{ex}(t, y), \mathrm{ex}(t + 1, y)\big) \ \wedge \\
&\qquad \wedge\ \mathrm{ex}(0, \mathrm{ex}(\mathrm{npr}(y) \dot{-} 1, y)) = 0 \ \wedge\ \mathrm{ex}(0, z) = 0\big)
\end{aligned} \tag{6.13.2}$$

Na vysvetlenie uveďme, že prvý riadok pravej strany vyjadruje, že výpočet začína predpísaným stavom; n je pevné a tri bodky teda možno nahradiť konkrétnym rozpisom, napr. pre $n = 0$ vyjde $\mathrm{ex}(0, y) = 2$. Druhý riadok vyjadruje, že y je číslom počiatočného úseku nejakého výpočtu. Tretí riadok vyjadruje, že v poslednom člene tohoto úseku je vnútorný stav q_0 a že stroj s číslom z neobsahuje inštrukciu začínajúcu q_0.

Cvičenie 6.22. Dokážte vetu 6.20. (Návod: Dokazujte sporom s vetou 6.19, v poslednom prípade s vetou 4.58.)

Veta 6.23. Množina $M \subseteq \mathbb{N}^n$ je rekurzívne spočítateľná práve vtedy, keď jej čiastočná charakteristická funkcia je čiastočne rekurzívna.

Dôkaz: Nech čiastočná charakteristická funkcia $\varkappa_M$ množiny M je čiastočne rekurzívna. Potom existuje $a \in \mathbb{N}$ také, že platí

$$\varkappa_M(x_1, \ldots, x_n) = \mathrm{mobs}\left(\mu_y(\mathrm{mvp}_a^{n+1}(y, x_1, \ldots, x_n) = 0)\right) \tag{6.23.1}$$

Ľahko sa zistí, že mvp_a^{n+1} je primitívne rekurzívna funkcia priradená množine M. (Dokázali sme teda viac, než sa tvrdí vo vete, pozri 4.36(b).) Obrátené tvrdenie obsahuje veta 4.46. □

Cvičenie 6.24. Množina $M \subseteq \mathbb{N}^n$ je rekurzívne spočítateľná práve vtedy, keď existuje čiastočne rekurzívna funkcia priradená množine M. Dokážte!

Cvičenie 6.25. Predikát je rekurzívne spočítateľný práve vtedy, keď jeho čiastočná charakteristická funkcia je čiastočne rekurzívna. Dokážte!

Veta 6.26. Nech f je n-árna čiastočná funkcia a $M \subseteq \mathbb{N}^{n+1}$ je jej graf. Potom f je čiastočne rekurzívna funkcia práve vtedy, keď M je rekurzívne spočítateľná množina.

Dôkaz: Ak f je čiastočne rekurzívna funkcia, tak aj čiastočná charakteristická funkcia $\varkappa_M$ množiny M je čiastočne rekurzívna, pretože pre všetky $x_1, \ldots, x_n, y$ platí

$$\varkappa_M(x_1, \ldots, x_n, y) = 0 - |f(x_1, \ldots, x_n) - y|$$

Teda M je rekurzívne spočítateľná podľa vety 6.23. Obrátené tvrdenie bolo vyslovené v 4.49. □

Cvičenie 6.27. Rozhodnite, ktoré z nasledujúcich tvrdení sú pravdivé:

(a) Graf rekurzívnej funkcie je rekurzívna množina.
(b) Ak je graf čiastočnej funkcie f rekurzívny, je f rekurzívna funkcia.
(c) Čiastočná funkcia f je čiastočne rekurzívna práve vtedy, keď jej obor definície a obor hodnôt sú rekurzívne spočítateľné množiny.
(d) Ak je graf funkcie f rekurzívny, je f rekurzívna funkcia.

Cvičenie 6.28. Dokážte:

(a) Existuje rekurzívne spočítateľná množina $M \subseteq \mathbb{N}^n$, ktorej komplement $\mathbb{N}^n \setminus M$ nie je rekurzívne spočítateľný.
(b) Existuje rekurzívne spočítateľný predikát, ktorého negácia nie je rekurzívne spočítateľný predikát.

Cvičenie 6.29. Dokážte, že existujú také rekurzívne spočíteľné predikáty $P(x)$, $Q(x)$, že predikáty $P(x) \iff Q(x)$, $P(x) \implies Q(x)$ nie sú rekurzívne spočítateľné. (Návod: pomocou $\implies$, $\iff$ a vhodných predikátov vyjadrite negáciu.)

Veta 6.30. Predikát $P(x_1, \ldots, x_n)$ je rekurzívne spočítateľný práve vtedy, keď existuje taký primitívne rekurzívny predikát $Q(y, x_1, \ldots, x_n)$, že pre všetky $x_1, \ldots, x_n$ platí

$$P(x_1, \ldots, x_n) \iff (\exists y) Q(y, x_1, \ldots, x_n) \tag{6.30.1}$$

Cvičenie 6.31. (a) Dokážte vetu 6.30.

(b) Ukážte, že veta 6.30 nebude platiť v žiadnom smere, ak v (6.30.1) nahradíme kvantifikátor $(\exists y)$ kvantifikátorom $(\forall y)$.

Cvičenie 6.32. Nájdite taký predikát $Q(x, y)$, ktorý nie je rekurzívne spočítateľný, že predikáty $(\exists y)Q(x, y)$, $(\forall y)Q(x, y)$ budú primitívne rekurzívne.

Cvičenie 6.33. Dokážte, že existuje taký primitívne rekurzívny predikát $Q(y, x_1, \ldots, x_n)$, že predikát $(\exists y)Q(y, x_1, \ldots, x_n)$ je rekurzívny, ale nie primitívne rekurzívny.

Cvičenie 6.34. Ku každému rekurzívnemu predikátu $P(x_1, \ldots, x_n)$ existuje taký primitívne rekurzívny predikát $Q(y, x_1, \ldots, x_n)$, že pre všetky $x_1, \ldots, x_n \in \mathbb{N}$ platí

$$P(x_1, \ldots, x_n) \iff (\forall y) Q(y, x_1, \ldots, x_n) \tag{6.34.1}$$

Dokážte!

Lema 6.44. Ak A je kreatívna množina, B rekurzívne spočítateľná množina a $A \leq_m B$, tak B je kreatívna množina.

Veta 6.45. Množina

$$K_0 = \{y \mid y \in \mathrm{Dom}(\mathrm{muniv}^1)\} \tag{6.45.1}$$

je kreatívna.

Dôkaz: Ku každému M-stroju Z a ku každému $a \in \mathbb{N}$ zostrojíme stroj $S(Z,a)$ takto: Najprv zvýšime indexy všetkých vnútorných stavov stroja Z okrem stavu q_0 o číslo a; tým dostaneme M-stroj Z'. Potom pridáme k stroju Z' a inštrukcií $(q_1R_1Pq_2)$, $(q_2R_1Pq_3)$, ..., $(q_aR_1Pq_{a+1})$. Zrejme pre každý stroj Z a každé a platí

$$\Phi^1_Z(a) = \Phi^0_{S(Z,a)} \tag{6.45.2}$$

Označme teraz g tú binárnu funkciu, ktorá každému číslu $y \in \mathbb{N} \setminus \{0\}$ a každému $a \in \mathbb{N}$ priraďuje číslo stroja $S(Z,a)$, kde Z je stroj s číslom y; nech ďalej $g(0,a) = 0$. Funkcia g je rekurzívna. Dokážeme teraz, že $K \leq_m K_0$. Nech $z \in K$, $z = \mathrm{c}(y,x)$ a nech Z je stroj s číslom y. Potom $z \in K$ práve vtedy, keď $\Phi^1_Z(x)$ je definovaná, t.j. keď $\Phi^0_{S(Z,x)}$ je definovaná, t.j. keď $g(y,x) \in K_0$. Teda platí

$$z \in K \iff g(\mathrm{l}(z), \mathrm{r}(z)) \in K_0 \tag{6.45.3}$$

a funkcia $f(z) = g(\mathrm{l}(z), \mathrm{r}(z))$ je rekurzívna. Množina K_0 je zrejme rekurzívne spočítateľná, a teda podľa lemy 6.44 kreatívna. □

Cvičenie 6.46. Dokážte, že funkcia g z dôkazu vety 6.45 je primitívne rekurzívna. (Vhodne ju vyjadrite.)

Poznámka 6.47. Z vety 6.45 by sme ľahko dokázali vetu 6.20 pre prípad $n = 0$. Teda predpoklad $n \neq 0$ vo vete 6.20 možno vynechať; pre pôvodný dôkaz podľa návodu však bol potrebný.

Cvičenie 6.48. Dokážte, že nasledujúce množiny sú kreatívne:

(a) množina všetkých čísel M-strojov Z takých, že hodnota $\Phi^1_Z(7)$ je definovaná
(b) množina $\{x \mid \mathrm{muniv}^2(x,7) = 5\}$
(c) množina $\{\mathrm{c}^3(z,y,x) \mid z = \mathrm{muniv}^2(y,x)\}$
(d) množiny $K_n = \{\mathrm{c}(y,x_1,\ldots,x_n) \mid (y,x_1,\ldots,x_n) \in \mathrm{Dom}(\mathrm{muniv}^{n+1})\}$ pre každé $n \in \mathbb{N}$ (prípady $n = 0$, 1 sme už prebrali)
(e) množiny $\{\mathrm{c}^{n+2}(y,x_1,\ldots,x_n,z) \mid z = \mathrm{muniv}^{n+1}(y,x_1,\ldots,x_n)\}$ pre každé $n \in \mathbb{N}$

Cvičenie 6.49. Nech f je bijektívna rekurzívna funkcia zobrazujúca $\mathbb{N}$ na $\mathbb{N}$. Potom množina A je kreatívna práve vtedy, keď množina $\{f(x) \mid x \in A\}$ je kreatívna. Dokážte!

Zatiaľ poznáme dva druhy rekurzívne spočítateľných množín: kreatívne a rekurzívne. Aby sme ukázali, že existujú aj iné rekurzívne spočítateľné množiny, zavedieme ďalšie pojmy. (Kreatívne množiny tvoria jednu triedu ekvivalenicie $\equiv_m$, rekurzívne množiny tri triedy. Existuje nekonečne mnoho ďalších tried ekvivalenicie $\equiv_m$ tvorených rekurzívne spočítateľnými množinami; my však dokážeme iba to, že existuje aspoň jedna ďalšia takáto trieda.)

Definícia 6.50. Množiny M_0, M_1 nazveme *rekurzívne oddeliteľnými*, ak existuje rekurzívna množina A taká, že $M_0 \subseteq A$ a $M_1 \subseteq \mathbb{N} \setminus A$. Množiny M_0, M_1 nazveme *rekurzívne neoddeliteľnými*, ak nie sú rekurzívne oddeliteľné.

Veta 6.51. Existujú navzájom disjunktné rekurzívne spočítateľné množiny M_0, M_1, ktoré sú rekurzívne neoddeliteľné.

Dôkaz: Nech $f(x) = \mathrm{sg}\left(\mathrm{muniv}^2(\mathrm{l}(x), \mathrm{r}(x))\right)$, $M_0 = \{x \mid f(x) = 0\}$, $M_1 = \{x \mid f(x) = 1\}$. Množiny M_0, M_1 sú zrejme disjunktné a rekurzívne spočítateľné. Keby boli rekurzívne oddeliteľné, mala by funkcia f rekurzívne zúplnenie, a to by bol spor s bodom (a) vety 6.20. □

Cvičenie 6.52. Ak je jedna z množín M_0, M_1 rekurzívna, tak M_0, M_1 sú rekurzívne oddeliteľné práve vtedy, keď sú disjunktné. Dokážte!

Definícia 6.60. Budeme hovoriť, že hromadný problém „Zistiť pre ľubovoľné dané $x \in \mathbb{N}$, či platí $x \in A$ alebo platí $x \notin A$“ je *rekurzívne riešiteľným*, ak je množina A rekurzívna. Tento problém nazývame *rekurzívne neriešiteľným*, ak množina A nie je rekurzívna.

Pojmy rekurzívnej riešiteľnosti a neriešiteľnosti aplikujeme aj na hromadné problémy iného tvaru než je problém uvedený v 6.60, ak je jasné, ako ho možno na problém predpísaného tvaru previesť. Napríklad pre $A \subseteq \mathbb{N}^2$ (a potom samozrejme tiež $x \in \mathbb{N}^2$), možno použiť už pre nás bežným spôsobom číslovacie funkcie. Ak A bude množina M-strojov, možno použiť ich číslovanie. (V kapitole 10 uvedieme dôvody, prečo sa možno domnievať, že konkrétna voľba očíslovania nie je podstatná.) Budeme sa teraz zaujímať o nasledujúce tri hromadné problémy. Odteraz až do konca kapitoly nech Z_k znamená M-stroj s číslom k, X_m znamená M-stav s číslom m; pre $k = 0$ nech $Z_0 = \emptyset$, a ak m nie je číslom stavu, nech $X_m = (q_1; 0)$.

Definícia 6.61. *Všeobecným problémom zastavenia pre registrové stroje* nazveme hromadný problém „pre ľubovoľný daný M-stroj Y a ľubovoľný daný M-stav X zistiť, či je výpočet stroja Y zo stavu X konečný alebo nie“.

Definícia 6.62. *Problémom zastavenia pre M-stroj Z_k* nazveme hromadný problém „pre ľubovoľný daný M-stav X zistiť, či je výpočet stroja Z_k zo stavu X konečný alebo nie.“

Definícia 6.63. *Problémom zastavenia pre M-stav X_m* nazveme hromadný problém „pre ľubovoľný daný M-stroj Y zistiť, či je výpočet stroja Y zo stavu X_m konečný alebo nie.“

Týmto trom hromadným problémom (ale v 6.62, 6.63 máme pre každé k, resp. m iný hromadný problém!) zodpovedajú pri zavedenom číslovaní M-strojov a M-stavov nasledujúce číselné množiny

$$M_1 = \{\mathrm{c}(u, v) \mid \text{výpočet stroja } Z_u \text{ zo stavu } X_v \text{ je konečný}\},$$
$$M_2(k) = \{v \mid \text{výpočet stroja } Z_k \text{ zo stavu } X_v \text{ je konečný}\},$$
$$M_3(m) = \{u \mid \text{výpočet stroja } Z_u \text{ zo stavu } X_m \text{ je konečný}\}.$$

Dá sa dokázať, že množina M_1 nie je rekurzívna, množina $M_3(m)$ nie je rekurzívna pre žiadne $m \in \mathbb{N}$ a existuje také k, že množina $M_2(k)$ nie je rekurzívna. Preto platí nasledujúca veta.

Veta 6.64. (a) Všeobecný problém zastavenia pre M-stroje je rekurzívne neriešiteľný.
(b) Pre každý M-stav X je problém zastavenia pre tento M-stav rekurzívne neriešiteľný.
(c) Existuje taký M-stroj Z, ktorého problém zastavenia je rekurzívne neriešiteľný.

Cvičenie 6.65. Dokážte, že množiny M_1, $M_2(k)$, $M_3(m)$ sú rekurzívne spočítateľné.

Cvičenie 6.66. (a) Dokážte, že množina M_1 nie je rekurzívna. (Návod: dokážte $K \leq_m M_1$, kde K je množina z vety 6.43.)
(b) Dokážte, že množina $M_3(m)$ nie je rekurzívna pre žiadne $m \in \mathbb{N}$.
(c) Dokážte, že existuje také k, že množina $M_2(k)$ nie je rekurzívna. (Návod: uvážte stroj, ktorý počíta muniv^2.)

Cvičenie 6.67. Nech

$$M_2'(k) = \{x \in M_1 \mid \mathrm{l}(x) = k\}, \qquad M_3'(m) = \{x \in M_1 \mid \mathrm{r}(x) = m\}$$

Bez využívania výsledku cvičenia 6.66 dokážte, že pre každé $k \in \mathbb{N}$ množina $M_2'(k)$ je rekurzívna práve vtedy, keď $M_2(k)$ je rekurzívna a pre každé $m \in \mathbb{N}$ množina M_3' je rekurzívna práve vtedy, keď $M_3(m)$ je rekurzívna.

Na záver zosilníme vetu 5.62 o generovaní rekurzívne spočítateľných množín registrovými strojmi.

Veta 6.68. Množina $A \subseteq \mathbb{N}^n$ je rekurzívne spočítateľná práve vtedy, keď existuje registrový stroj Z, ktorý ju generuje.

Dôkaz: Priama implikácia je veta 5.62; dokazujme len obrátenú implikáciu. Nech Z je registrový stroj. Množina X čísel všetkých stavov, ktoré sa vyskytnú vo výpočte stroja Z zo stavu $(q_1; 0)$ je rekurzívne spočítateľná. (Na dôkaz možno nadefinovať rekurzívne spočítateľný predikát „y je počiatočný úsek výpočtu stroja číslo z zo stavu $(q_1; 0, x)$“, ktorý sa od predikátu Mvýp^3 líši tým, že neuvažuje ukončenie výpočtu.) Pomocou nej ľahko vyjadríme množinu A tak, že jej rekurzívna spočítateľnosť bude zrejmá: Treba zobrať nepárne prvky z X (tie obsahujú vnútorný stav q_0)

KAPITOLA 7

Abecedy, slová a hromadné úlohy

V matematike i v jej aplikáciách často treba riešiť veľký počet úloh rovnakého typu, na riešenie ktorých je výhodné mať vypracovaný jeden predpis, ktorý potom len na jednotlivé prípady aplikujeme. Napríklad pre riešenie kvadratických rovníc máme vzorec a každú kvadratickú rovnicu riešime dosadzovaním do tohto vzorca. (Ďalšie príklady i s podrobnejším rozborom uvedieme neskôr.) Pritom to, ktoré úlohy pokladáme za úlohy rovnakého typu, závisí od mnohých okolností, často aj od toho, či pre tie úlohy nájdeme spoločný algoritmus na ich riešenie. Najprv sa však budeme zaoberať otázkou formulácie týchto úloh, a nie otázkou ich riešenia. Pokiaľ sa nejaký typ úloh vyskytuje v praktických (napríklad v technických) aplikáciach, vždy je možné, aby údaje pre jednu takúto úlohu odovzdal jeden človek druhému v konečnom čase, ústne alebo písomne. Odovzdanie v písomnej forme si môžeme predstaviť v tvare konečnej (prípadne veľmi dlhej) postupnosti znakov z nejakej konečnej množiny, napríklad z množiny znakov klávesnice písacieho stroja, ktorú často nazývame abecedou. To isté sa týka aj odovzdávania výsledkov riešenia.

Rozoberieme stručne niekoľko príkladov, z ktorých niektoré by azda mohli u čitateľa vzbudiť pochybnosti. Odovzdávanie prirodzených, celých alebo racionálnych čísel, ako aj konečných postupností týchto čísel v tvare konečnej postupnosti znakov je zrejme možné. Pokiaľ sa vyskytujú reálne čísla ako údaje meracích prístrojov, môžeme ich vždy nahradiť vhodnými racionálnymi aproximáciami (a prakticky vždy to aj robíme). Pokiaľ sa vykytnú iracionálne reálne čísla, ku ktorým sme prišli nejakou teoretickou úvahou (napríklad e, π, $\log 2$) a nie je možné nahradiť ich vhodnou racionálnou aproximáciou, možno uviesť vzorcom alebo slovami ich definíciu. Graf funkcie zostrojený nejakým prístrojom alebo narysovaný možno nahradiť dostatočne podrobnou tabuľkou hodnôt funkcie. Iné obrázky a diagramy možno rozdeliť na dostatočne malé „body“, ktoré už ľudské oko nerozlišuje. Každému takémuto „bodu“ priradíme prirodzené číslo podľa jeho farebného odtieňa (rozlíšiteľných odtieňov je zasa konečne mnoho), a potom napíšeme postupnosť týchto prirodzených čísel. Upozorňujeme, že nám teraz ide iba o principiálnu možnosť zapísania vstupných, resp. výstupných údajov v tvare konečnej postupnosti znakov, a nie o nájdenie prakticky výhodných postupov.

Poznamenajme ešte, že pri niektorých množinách úloh, ktoré sa uvažujú (ale nie každá úloha z nich sa zvlášť rieši) v čistej matematike, sa môže stať, že nie je možné určiť vstupné údaje jednej úlohy konečnou postupnosťou znakov. Pokiaľ však takéto množiny úloh vznikajú abstrakciou z nejakej množiny praktických úloh, znamená to, že zvolený stupeň abstrakcie je nevhodný pre naše terajšie úvahy (i keď mohol byť veľmi vhodný pre úvahy iného druhu). Čitateľa, ktorý pozná matematickú logiku, môžeme okrem toho upozorniť, že aj veľmi abstraktným úlohám z čistej matematiky zodpovedajú po formalizácii teórií úlohy v matematike, ktoré už spĺňajú našu podmienku o možnosti odovzdávania vstupných i výstupných údajov v tvare postupnosti znakov.

Z doterajších úvah teda vidíme, že budú pre nás dôležité konečné postupnosti znakov z konečných množín. Pri našich úvahách nebude dôležitý druh použitých znakov, a preto budeme tieto znaky vyberať z nejakej pevne zvolenej postupnosti znakov. Táto postupnosť musí byť nekonečná, pretože počet znakov, ktoré budeme v jednotlivých príkladoch potrebovať, nie je dopredu zhora ohraničený. V príkladoch však nebudeme voliť vždy znaky z tejto postupnosti znakov, ale prípadne aj iné znaky, a pri teoretických úvahách budeme ich stotožňovať s niektorými znakmi z našej postupnosti. Niektoré také stotožnenia budú stanovené stále, niektoré budeme robiť od prípadu k prípadu, ale o väčšine z nich len budeme predpokladať, že sú urobené. V našej teórii nebudeme zavádzať nový základný pojem „znak“; znakmi budú prvky istej pevne zvolenej množiny.

V nasledujúcich definíciách zavádzame terminológiu a niektoré základné operácie.

Definícia 7.11. Pre ľubovoľné $n \in \mathbb{N}$, ľubovoľné slovo w a ľubovoľnú množinu slov A kladieme

$$w^0 = \varepsilon \qquad w^{n+1} = w^n w \tag{7.11.1}$$

$$A^0 = \{\varepsilon\} \qquad A^{n+1} = A^n A \tag{7.11.2}$$

Cvičenie 7.12. Rozhodnite, či pre každú množinu slov A a pre každé prirodzené číslo n platí

$$A^n = \{w^n \mid w \in A\}$$

Definícia 7.13. Pre ľubovoľnú množinu slov A definujeme

$$A^* = \bigcup_{n=0}^{\infty} A^n \qquad A^+ = \bigcup_{n=1}^{\infty} A^n \tag{7.13.1}$$

Ďalšie často používané operácie na množinách slov sú množinové operácie prienik $\cap$ a zjednotenie $\cup$ a aj ďalšie množinové operácie. Ak napríklad uvažujeme len množiny slov v abecede A, je takouto operáciu aj operácia (relatívneho) komplementu $A^* - X$. (Tieto operácie však nemusíme teraz definovať, lebo patria k pojmom prebratým z [5].) Zavedieme ešte operáciu zrkadlového obrazu.

Definícia 7.14. (a) *Zrkadlovým obrazom slova* $w = a_{i_1}a_{i_2}\ldots a_{i_{n-1}}a_{i_n}$ nazveme slovo

$$z = a_{i_n}a_{i_{n-1}}\ldots a_{i_2}a_{i_1}$$

Zrkadlový obraz slova w budeme označovať w^R.

(b) *Zrkadlovým obrazom množiny slov* A nazveme množinu všetkých zrkadlových obrazov slov množiny A. Zrkadlový obraz množiny A budeme značiť A^R.

Príklad 7.15. Nech $A = \{OO, OI, IO, II\}$, $B = \{O, OI\}$. Potom A^* je množina všetkých slov párnej dĺžky v abecede $\{O, I\}$. A^+ je množina všetkých neprázdnych slov párnej dĺžky v tej istej abecede. B^* je množina všetkých slov v abecede $\{O, I\}$, v ktorých sa nevyskytujú dve I za sebou ani I na začiatku; B^+ je množina všetkých neprázdnych slov množiny B^*. Pre množinu A platí $A^R = A$.

Cvičenie 7.16. Pre ktoré množiny slov A platí rovnosť $A^* = A^+$?

Cvičenie 7.17. Rozhodnite, či platia pre všetky množiny slov A, B nasledujúce vzťahy! V tých prípadoch, keď neplatí rovnosť, zistite, či platí aspoň množinová inklúzia (niektorým smerom); tam, kde sa vo vzťahoch vyskytuje prirodzené číslo n, urobte rozbor pre jednotlivé hodnoty n!

$$(A \cup B)^n = A^n \cup B^n \qquad (A \cap B)^n = A^n \cap B^n \qquad (A \setminus B)^n = A^n \setminus B^n$$

$$(A \cup B)^+ = A^+ \cup B^+ \qquad (A \cap B)^+ = A^+ \cap B^+ \qquad (A \setminus B)^+ = A^+ \setminus B^+$$

$$(A \cup B)^* = A^* \cup B^* \qquad (A \cap B)^* = A^* \cap B^* \qquad (A \setminus B)^* = A^* \setminus B^*$$

$$(A \cup B)^R = A^R \cup B^R \qquad (A \cap B)^R = A^R \cap B^R \qquad (A \setminus B)^R = A^R \setminus B^R$$

$$(A^n)^n = A \qquad (A^*)^* = A \qquad (A^+)^+ = A \qquad (A^R)^R = A$$

$$(A^n)^n = A^n \qquad (A^*)^* = A^* \qquad (A^+)^+ = A^+ \qquad (A^R)^R = A^R$$

$$(A^n)^* = (A^*)^n \qquad (A^n)^+ = (A^+)^n \qquad (A^n)^R = (A^R)^n$$

$$(A^*)^+ = (A^+)^* \qquad (A^*)^R = (A^R)^* \qquad (A^+)^R = (A^R)^+$$

$$(A^n)^* = A^* \qquad (A^+)^* = A^* \qquad (A^R)^* = A^* \qquad (A^*)^n = A^*$$

Príklad 7.18. Operáciu mocniny slov možno podľa dohody 7.7 využiť aj na skrátený zápis niektorých slov. Napríklad slovo $OIIIOIIIIIOOOII$ môžeme skrátene zapísať ako $OI^3OI^5O^3I^2$. (Tu aj inde vzťahujeme exponent len na jednopísmenové slová; pokiaľ by sa mal vzťahovať na väčšie slovo, vyznačili by sme to zátvorkami.)

D (pozri príklad 7.21). Zobrazenie $\varkappa$ v tejto hromadnej úlohe je definované predpisom

$$\varkappa(w) = \begin{cases} O, & \text{ak } w \text{ je zápis párneho čísla,} \\ I, & \text{ak } w \text{ je zápis nepárneho čísla} \end{cases}$$

Prakticky dôležitým problémom je napríklad zisťovanie správnosti programov pre samočinné počítače. Jednou z podmienok správnosti programu býva aj podmienka, aby sa výpočet podľa tohto programu skončil, a to alebo pre dané, alebo pre ľubovoľné vstupné údaje. Sformulujeme tu obdobné problémy pre registrové stroje bez toho, aby sme podrobnejšie rozoberali otázku, nakoľko verná je analógia medzi nimi a skutočnými samočinnými počítačmi.

Vráťme sa teraz k problémom zastavenia registrových strojov, ktoré sme už uvažovali v predchádzajúcej kapitole. Tam sme však o nich mohli hovoriť iba dosť neformálne, a pre presnejšie formulácie sme ich museli nahradiť problémami týkajúcimi sa číselných množín. Vzťah medzi M-strojom a jeho číslom (resp. medzi M-stavom a jeho číslom) je však veľmi nenázorný, a teraz už máme možnosť zakódovať M-stroje a M-stavy omnoho názornejšie.

Definícia 7.26. (a) *Kódmi M-inštrukcií* $(q_iR_jMq_k)$, $(q_iR_jPq_k)$, $(q_iR_jq_mq_n)$ budú po rade slová

$$QI^iRI^jMQI^kO, \qquad QI^iRI^jPQI^kO, \qquad QI^iRI^jQI^mQI^nO$$

(b) *Kódom M-stroja* bude zreťazenie kódov jeho inštrukcií usporiadaných podľa indexov ich prvých zložiek.

(c) *Kódom M-stavu* $X = (q_n; b_0, b_1, \ldots, b_k)$, kde $b_k \neq 0$, ak $k \neq 0$, bude slovo

$$OI^nOI^{b_0}OI^{b_1}O \ldots OI^{b_k}O$$

Všimnime si, že na kódovanie čísel sme nepoužili ich dekadické (prípadne binárne), ale tzv. unárne zápisy. Podmienka $b_k \neq 0$, ak $k \neq 0$ v bode (c) je potrebná len kvôli jednoznačnosti; často ju vôbec nepotrebujeme alebo ju môžeme výhodnejšie zabezpečiť voľbou konštantného k.

Ak teraz označíme $A = \{Q, R, M, P, O, I\}$, O odpoveď „áno“, I odpoveď „nie“, môžeme problémy 6.61 až 6.63 preformulovať takto:

Definícia 7.27. *Problémom zastavenia* pre M-stroj Z nazveme hromadnú úlohu

$$(A, \{O, I\}, U_1, \varkappa_1) \tag{7.27.1}$$

kde U_1 je množina všetkých kódov M-stavov a

$$\varkappa_1(w) = \begin{cases} O, & \text{ak je výpočet } M\text{-stroja } Z \text{ zo stavu s kódom } w \text{ konečný,} \\ I, & \text{v opačnom prípade} \end{cases}$$

pre všetky $w \in U_1$.

Definícia 7.28. *Problémom zastavenia* pre M-stav X nazveme hromadnú úlohu

$$(A, \{O, I\}, U_2, \varkappa_2) \tag{7.28.1}$$

kde U_2 je množina všetkých kódov M-strojov a

$$\varkappa_2(w) = \begin{cases} O, & \text{ak je výpočet } M\text{-stroja s kódom } w \text{ zo stavu } X \text{ konečný,} \\ I, & \text{v opačnom prípade} \end{cases}$$

pre všetky $w \in U_2$.

Pre nasledujúcu definíciu si uvedomíme, že slovo $w \in U_2U_1$ sa dá práve jedným spôsobom písať v tvare $w = w_2w_1$, $w_2 \in U_2$, $w_1 \in U_1$. (Pretože w obsahuje podslovo OO, ktoré nie je podslovom žiadneho slova z U_2 ani z U_1.)

Definícia 7.29. *Všeobecným problémom zastavenia* pre M-stroje nazveme hromadnú úlohu

$$(A, \{O, I\}, U_3, \varkappa_3) \tag{7.29.1}$$

kde $U_3 = U_2U_1$ a

$$\varkappa(w_2w_1) = \begin{cases} O, & \text{ak výpočet } M\text{-stroja s kódom } w_2 \text{ pre } M\text{-stav s kódom } w_1 \text{ je konečný,} \\ I, & \text{v opačnom prípade} \end{cases}$$

KAPITOLA 8

Turingove stroje

V tejto kapitole zavedieme Turingove stroje. Je takmer isté (a na príkladoch sa o tom presvedčíme), že Turingove stroje stačia na riešenie všetkých algoritmicky riešiteľných úloh, pokiaľ sú tieto úlohy vhodne zakódované.

Opíšeme najprv neformálne Turingove stroje a ich činnosť. Obdobne ako registrové stroje, ani Turingove stroje nie sú technické zariadenia, ale len matematické abstrakcie, bolo by však možné realizovať ich technicky až na nekonečnosť ich pásky. Toto ohraničenie však nie je podstatné, lebo ak bude mať Turingov stroj veľmi dlhú konečnú pásku, tak bude počas dlhého času pracovať tak isto, ako keby mal nekonečnú pásku. Okrem toho si môžeme tiež predstavovať, že páska je síce konečná, že ju však vždy v prípade potreby možno predĺžiť. Rovnako ako registrové stroje, aj Turingove stroje pracujú v diskrétnom čase.

Turingov stroj sa skladá z troch častí: riadiacej jednotky, čítacej a zapisovacej hlavy a pásky. Riadiaca jednotka sa môže nachádzať v jednom z konečne mnohých vnútorných stavov, prijíma signály od čítacej a zapisovacej hlavy a dáva signály pre jej činnosť. Páska je lineárna a v oboch smeroch nekonečná. Je rozdelená na políčka, do každého z nich sa zapisuje jedno písmeno z nejakej abecedy. O políčkach pásky, do ktorých sa ešte nezapisovalo, predpokladáme, že je v nich zapísaný symbol B (z anglického „blank“); vo veľkej väčšine prípadov bude teda symbol B napísaný vo všetkých políčkach pásky až na ich konečný počet a iba výnimočne budeme uvažovať, že bol zaplnený nekonečný počet políčok pásky. Hlava číta zakaždým jedno políčko pásky, odošle signál o jeho obsahu riadiacej jednotke a po prijatí signálu od riadiacej jednotky zmení predpísaným spôsobom obsah čítaného políčka a vykoná predpísaný pohyb o jedno políčko doľava alebo doprava, alebo zostane na mieste. Riadiaca jednotka v každom kroku činnosti Turingovho stroja najprv prijme signál od čítacej a zapisovacej hlavy, potom tejto hlave odošle signál pre jej činnosť a nakoniec sama prejde do nového vnútorného stavu. Ak pre niektorý vnútorný stav a pre niektorý obsah čítaného políčka nie je určená ďalšia činnosť, výpočet sa ukončí.

Pristúpime teraz k formálnej definícii Turingových strojov, ich stavov a výpočtov.

Definícia 8.1. Usporiadané 5-tice tvarov

$$(q_i a_j a_m L q_n)\,, \qquad (q_i a_j a_m N q_n)\,, \qquad (q_i a_j a_m P q_n) \tag{8.1.1}$$

budeme nazývať *turingovskými inštrukciami*, prípadne len *inštrukciami* alebo *T-inštrukciami*. (Budeme ich písať bez čiarok.)

Poznámka 8.2. Odteraz pod inštrukciami budeme rozumieť vždy T-inštrukcie; inštrukcie registrových strojov budeme nazývať M-inštrukciami. Aj pri ostatných pojmoch z kapitoly 5, ktoré budú zavedené v novom význame v tejto kapitole, budeme dôsledne používať písmeno M (alebo úplný názov).

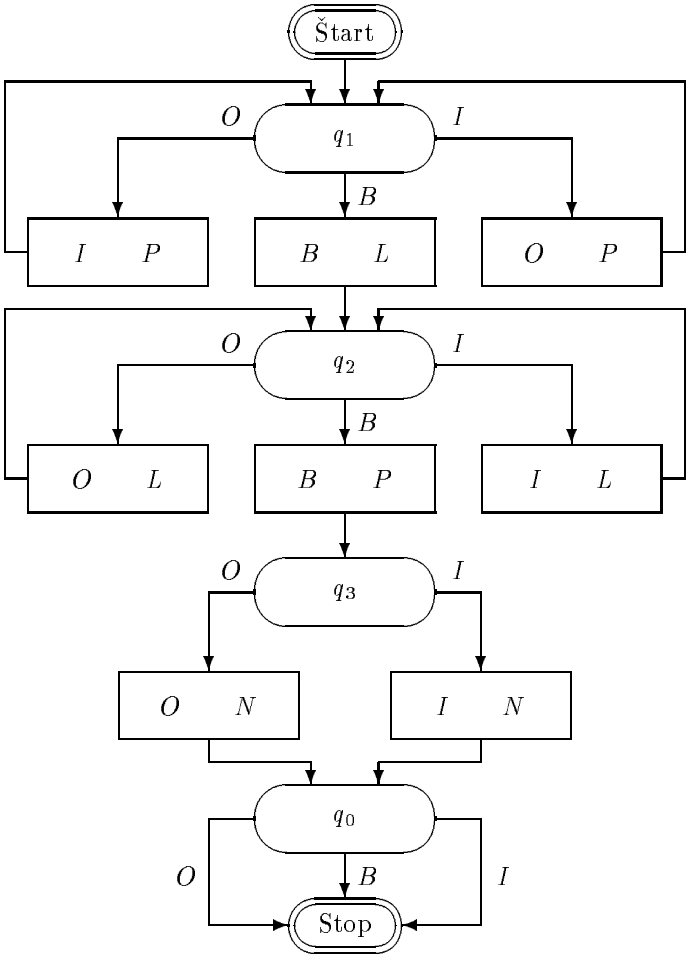
Poznámka 8.3. Vnútorné stavy q_i sme zaviedli v definícii 5.1(a). Symboly a_i sme zaviedli v definícii 7.1(a). P, L, N sú pevne zvolené symboly; zhoda symbolu N s označením množiny všetkých prirodzených čísel je iba náhodná.

Definícia 8.4. Konečnú množinu T-inštrukcií, ktorá neobsahuje dve rôzne inštrukcie s rovnakými prvými dvoma prvkami, budeme nazývať *Turingovým strojom*, prípadne len *T-strojom* alebo *strojom*.

Definícia 8.5. Takú konečnú postupnosť

$$a_{i_1} a_{i_2} \ldots a_{i_k} q_j a_{i_{k+1}} \ldots a_{i_n} \tag{8.5.1}$$

že $i_1 \neq 0, i_n \neq 0$, budeme nazývať *stavom Turingovho stroja*, prípadne len *stavom* alebo *T-stavom*.



OBR. 8.1.

zo stavu q_1OIIO. Platí teda $q_1OIIO \overset{T}{\longmapsto} q_0IOOI$. Okrem toho platí napríklad $Iq_1IIO \overset{T}{\Longrightarrow} q_2BIOOI$, $IOq_1IO \overset{T}{\Longrightarrow} IOq_2OI$.

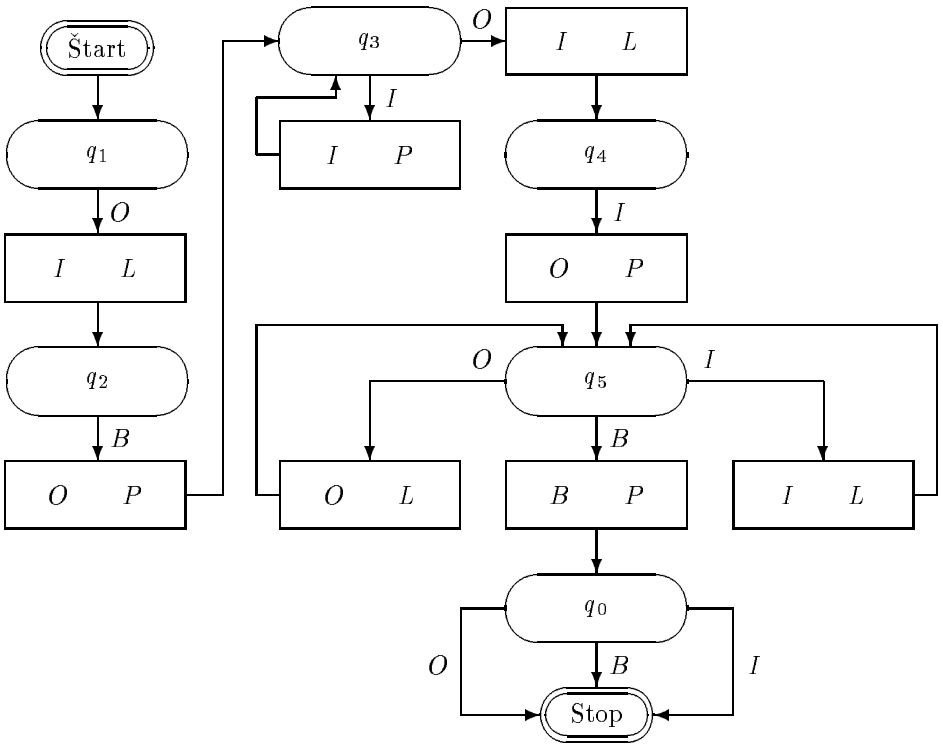
Cvičenie 8.16. Opíšte stav, ktorým sa končí výpočet stroja T z príkladu 8.12 zo stavu q_1w, kde w je ľubovolné slovo v abecede $\{O, I\}$!

Cvičenie 8.17. Riešte pre Turingove stroje cvičenia 5.14, 5.15 a 5.16!

Budeme sa teraz zaoberať možnosťou riešenia hromadných úloh na Turingových strojoch. Obmedzíme sa pritom na hromadné úlohy, ktorých vstupná ani výstupná abeceda neobsahuje symbol B. Toto obmedzenie samozrejme nie je podstatné, lebo by sme ho mohli obísť rôznymi spôsobmi, napríklad tak, že by sme v danej hromadnej úlohe so symbolom B všade nahradili symbol B nejakým prvkom množiny $\overline{A}$, ktorý sa nevyskytuje vo vstupnej ani vo výstupnej abecede tejto hromadnej úlohy. Zjednoduší nám však niektoré ďalšie úvahy.

Definícia 8.18. *Abecedou* Turingovho stroja T budeme nazývať množinu všetkých prvkov rôznych od B na druhých a tretích miestach jeho inštrukcií, pokiaľ je táto množina neprázdna. Ak je táto množina prázdna, budeme pod *abecedou* stroja T rozumieť množinu $\{O\}$ (t.j. $\{a_1\}$).

Poznámka 8.19. Ako uvidíme ďalej, druhá časť definície 8.18 je potrebná iba pre úplnosť a v netriviálnych prípadoch sa neprejaví.



OBR. 8.3.

Definícia 8.24. (a) Nech T je Turingov stroj a n prirodzené číslo. Znakom Φ_T^n budeme označovať takú n-árnu čiastočnú funkciu f na množine $\mathbb{N}$, že pre všetky y, x_1, ..., $x_n \in \mathbb{N}$ platí $f(x_1, \ldots, x_n) = y$ práve vtedy, keď $\mathrm{Rez}_T(OI^{x_1}OI^{x_2}O \ldots OI^{x_n}O) = OI^yO$. Čiastočnú funkciu Φ_T^n budeme nazývať *n-árnou čiastočnou funkciou*, ktorú *počíta stroj* T.

(b) Budeme hovoriť, že n-árna (čiastočná) funkcia f je *(čiastočne) vypočítateľná na Turingovom stroji* (alebo len *(čiastočne) T-vypočítateľná*), ak existuje taký Turingov stroj T, že $f = \Phi_T^n$.

Príklad 8.25. Nech $T = \{(q_1OIL q_2), (q_2BIL q_3), (q_3BIL q_4), (q_4BON q_0)\}$. Pre ľubovoľné prirodzené číslo k platí $q_1OI^kO \xrightarrow{T} q_2BI^{k+1}O \xrightarrow{T} q_3BI^{k+2}O \to q_4BI^{k+3}O \xrightarrow{T} q_0OI^{k+3}O$, a teda $\mathrm{Rez}_T(OI^kO) = OI^{k+3}O$. Preto $\Phi_T^1(k) = k + 3$ pre všetky $k \in \mathbb{N}$.

Príklad 8.26. Nech

$$T = \{(q_1OBP q_2), (q_2IIP q_2), (q_2OIN q_3), (q_3IIL q_3), (q_3BBP q_4), (q_4ION q_0)\}$$

Platí napríklad $q_1OIIOIIIO \xrightarrow{T} q_2IIOIIIO \xrightarrow{T} Iq_2IOIIIO \xrightarrow{T} IIq_2OIIIO \xrightarrow{T} IIq_3IIIIO \xrightarrow{T} Iq_3IIIIIO \xrightarrow{T} q_3IIIIIIO \xrightarrow{T} q_3BIIIIIIO \xrightarrow{T} q_4IIIIIIO \xrightarrow{T} q_0OIIIIIO$, a teda

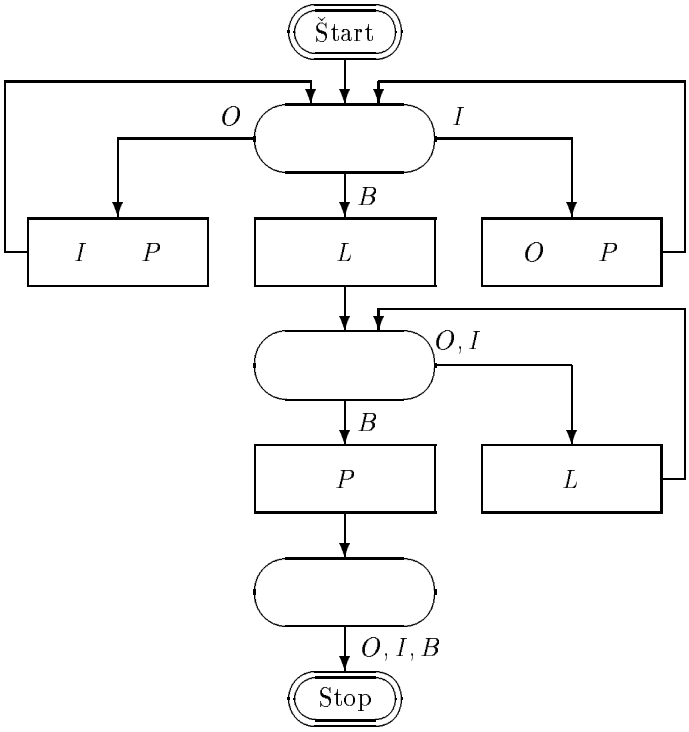
$$q_1OIIOIIIO \overset{T}{\Longmapsto} q_0OIIIIIO$$

t.j. $\mathrm{Rez}_T(OI^2OI^3O) = OI^5O$. Preto platí $\Phi_T^2(2,3) = 5$; ľahko sa tiež zistí, že pre všetky x, y platí $\Phi_T^2(x,y) = x + y$. Teda podľa definície 8.24 je funkcia $\lambda xy(x + y)$ T-vypočítateľná.

Príklad 8.27. Nech

$$T = \{(q_1OIP q_1), (q_1IOP q_1), (q_1BBL q_2), (q_2OOL q_2), (q_2IIL q_2), (q_2BBP q_0)\}$$

Pre tento stroj T platí napríklad $\mathrm{Rez}_T(OOIIIO) = IIOOOI$. Všeobecne, stroj T prepracuje každé vstupné slovo v abecede $\{O, I\}$, ktoré dostane na pásku, na jeho „negatív“, v ktorom budú



OBR. 8.5.

Definícia 8.30. *Blokovú schému Turingovho stroja T* dostaneme tak, že

(a) Každej inštrukcii $(q_i a_j a_m X q_n)$ (kde X je jeden zo symbolov P, L, N) priradíme obdĺžnik, do ktorého vpíšeme „a_m X“.

(b) Každému vnútornému stavu, ktorý sa nachádza na prvom alebo na poslednom mieste aspoň jednej inštrukcie stroja T, priradíme ovál, do ktorého vpíšeme tento vnútorný stav.

(c) Ak ovál A patrí k vnútornému stavu na prvom mieste inštrukcie patriacej k obdĺžniku B, vedieme šípku od A k B. K začiatku šípky pripíšeme písmeno z druhého miesta príslušnej inštrukcie.

(d) Ak ovál A patrí k vnútornému stavu na poslednom mieste inštrukcie patriacej k obdĺžniku B, vedieme šípku od B k A.

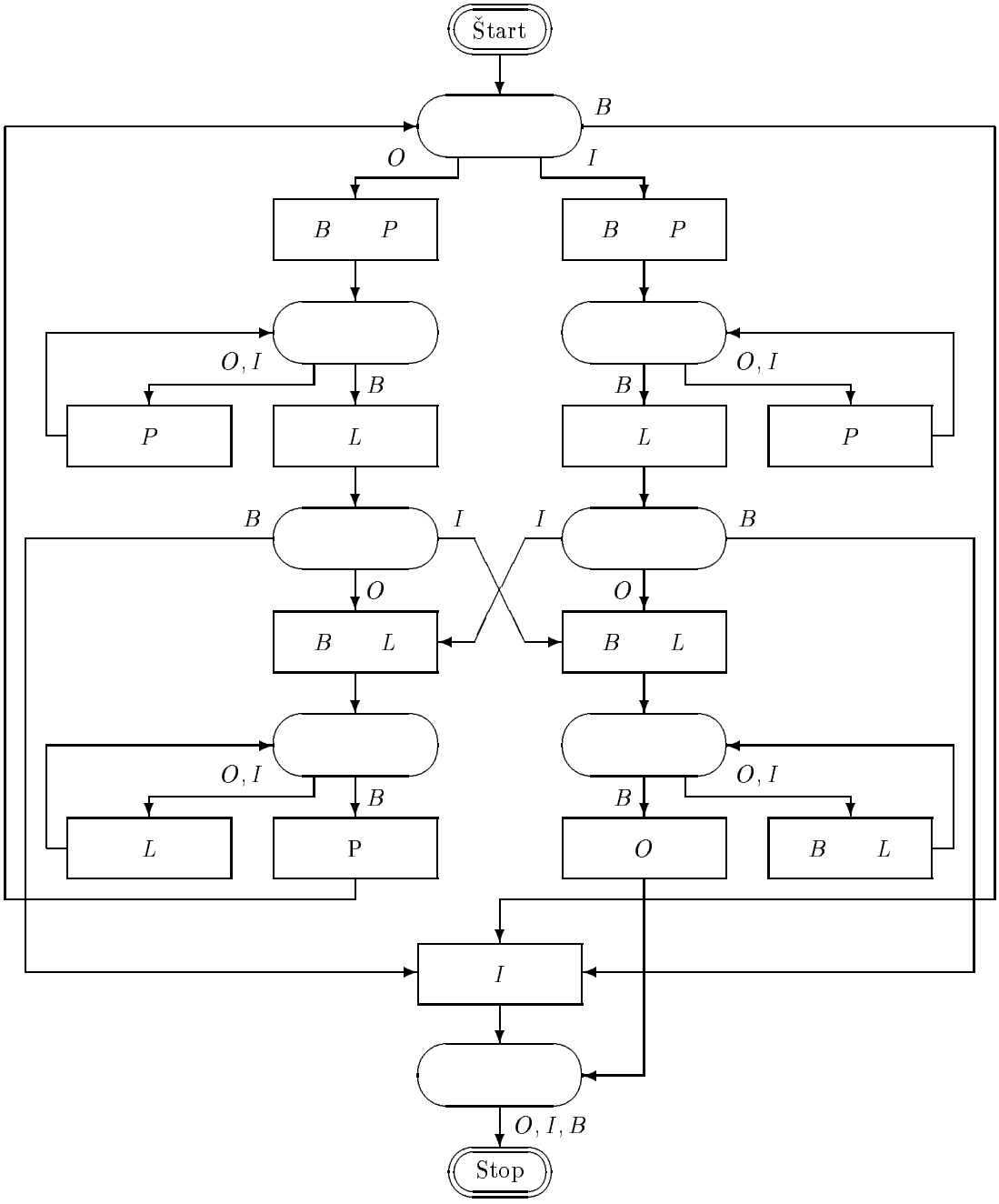
(e) Ak stroj T obsahuje inštrukciu začínajúcu sa vnútorným stavom q_1, doplníme blokovú schému začiatkovým oválom, od ktorého vedieme šípku ku krúžku patriacemu k vnútornému stavu q_1.

(f) Nech Y je množina všetkých takých x, že $x = B$ (t.j. a_0) alebo x je z abecedy stroja T a stroj T neobsahuje inštrukciu začínajúcu sa $q_0 x$. Ak stroj T obsahuje inštrukciu končiacu sa vnútorným stavom q_0 a $Y \neq \emptyset$, doplníme blokovú schému koncovým oválom, ku ktorému vedieme toľko šípok od krúžku patriaceho ku q_0, koľko prvkov má množina Y. Začiatky šípok označíme prvkami množiny Y.

Príklad 8.31. Blokové schémy T-strojov z príkladov 8.12, 8.27 a 8.28 sú na obrázkoch 8.1, 8.2 a 8.3.

Turingov stroj je blokovou schémou jednoznačne určený, avšak (nakoľko sme neurčili celkom detailne spôsob kreslenia schémy) jeden T-stroj môže mať niekoľko rôznych blokových schém (situácia je obdobná ako pri blokových schémach M-strojov).

Niekedy budú pre nás výhodné zjednodušené blokové schémy T-strojov, i keď týmito už nebude príslušný stroj určovaný jednoznačne. Pre zjednodušenie blokových schém T-strojov zavádzame:



OBR. 8.7.

stroj vymaže aj posledný symbol slova a vráti sa k začiatku slova na páske (ktoré je už o dva symboly kratšie). Potom opakuje tú istú činnosť, a to až dovtedy, kým nezmaže celé slovo w. V tomto prípade stroj T napíše na pásku symbol I a ukončí výpočet. Ak sa v niektorom kroku pri porovnávaní prvého a posledného symbolu vyskytne nezhoda, stroj preruší ďalšie porovnávanie, vymaže všetky symboly na páske, napíše symbol O a skončí výpočet. Na obrázku 8.7 je bloková schéma stroja T zjednodušená podľa niektorých bodov dohody 8.32. Nakreslenie nezjednodušenej blokovej schémy aj napísanie inštrukcií stroja T prenechávame čitateľovi. Len na ukážku uvádzame,

- $\Phi^1_T(x) = 2x$ a $\Phi^2_T(x,y) = x + y$
- $\Phi^1_T(x) = \lfloor \log_2 x \rfloor$

Cvičenie 8.42. Tabuľka Turingovho stroja určuje isté čiastočné zobrazenie množiny $\overline{Q} \times \overline{A}$ do množiny $\overline{A} \times \{P, L, N\} \times \overline{Q}$. Modifikujte definíciu Turingovho stroja tak, aby Turingove stroje boli práve tieto čiastočné zobrazenia! Potom vhodne modifikujte aj definíciu relácie „$\xrightarrow{T}$“!

Hlavne pre potreby dôkazu nasledujúcej vety zavedieme:

Označenie 8.43. Pre ľubovoľnú konečnú postupnosť $(k_0, k_1, \ldots, k_n)$ prirodzených čísel označíme znakom $\mathrm{Slv}(k_0, k_1, \ldots, k_n)$ slovo $OI^{k_0}OI^{k_1}\ldots OI^{k_n}0$.

Príklad 8.44. Platí $\mathrm{Slv}(2,3,5) = OIIOIIIOIIIIIO$, a teda miesto $q_3 OIIOIIIOIIIIIO$ môžeme písať $q_3\,\mathrm{Slv}(2,3,5)$.

Budeme sa teraz zaoberať otázkou vzťahu (čiastočnej) vypočítateľnosti na Turingových strojoch a (čiastočnej) vypočítateľnosti na registrových strojoch. Našim najbližším cieľom bude dôkaz vety:

Veta 8.45. Pre každý registrový stroj Z a každé $n \in \mathbb{N}$ existuje taký Turingov stroj T, že $\Phi^n_Z = \Phi^n_T$.

Dôkaz: Nech Q je množina všetkých vnútorných stavov v M-inštrukciách stroja Z a nech q_u, q_v $(u < v)$ sú vnútorné stavy s najnižšími indexmi, ktoré nepatria do množiny $Q \cup \{q_0, q_1\}$. Označme Z' stroj, ktorý vznikne zo Z nahradením q_1, q_0 symbolmi q_u, q_v. Každej M-inštrukcii X stroja Z' priraďme Turingov stroj T_X tak, aby boli splnené nasledujúce podmienky:

(1) Ak $X = (q_i S_j P q_k)$ alebo $X = (q_i S_j M q_k)$, tak existuje jediná inštrukcia stroja T_X, ktorá sa začína q_i, a jediná inštrukcia T-stroja T_X, ktorá sa končí q_k. Žiadne ďalšie T-inštrukcie stroja T_X neobsahujú vnútorné stavy z množiny $Q \cup \{q_0, q_1, q_u, q_v\}$.

(2) Ak $X = (q_i S_j q_r q_s)$, tak existuje jediná T-inštrukcia stroja T_X začínajúca sa q_i, jediná jeho T-inštrukcia končiaca sa q_r a jediná jeho T-inštrukcia končiaca sa q_s. Okrem týchto výnimiek neobsahujú T-inštrukcie stroja T_X vnútorné stavy z množiny $Q \cup \{q_0, q_1, q_u, q_v\}$.

(3) Žiaden vnútorný stav, ktorý nepatrí do množiny $Q \cup \{q_0, q_1, q_u, q_v\}$, sa nenachádza v inštrukciách dvoch strojov T_X, T_Y pre $X \neq Y$.

(4) Ak $X = (q_i S_j P q_k)$, resp. $X = (q_i S_j M q_k)$, tak pre každé $m \geq j$ a všetky $k_0, \ldots, k_m \in \mathbb{N}$ platí

$$q_i\,\mathrm{Slv}(k_0, \ldots, k_{j-1}, k_j, k_{j+1}, \ldots, k_m) \overset{T_X}{\Longmapsto} \overset{T_X}{\Longmapsto} q_k\,\mathrm{Slv}(k_0, \ldots, k_{j-1}, k_j + 1, k_{j+1}, \ldots, k_m) \tag{8.45.1}$$

resp.

$$q_i\,\mathrm{Slv}(k_0, \ldots, k_{j-1}, k_j, k_{j+1}, \ldots, k_m) \overset{T_X}{\Longmapsto} \overset{T_X}{\Longmapsto} q_k\,\mathrm{Slv}(k_0, \ldots, k_{j-1}, k_j - 1, k_{j+1}, \ldots, k_m) \tag{8.45.2}$$

(5) Ak $X = (q_i S_j q_r q_s)$, tak pre každé $m \geq j$ a všetky $k_0, \ldots, k_m \in \mathbb{N}$ platia vzťahy

$$q_i\,\mathrm{Slv}(k_0, \ldots, k_{j-1}, 0, k_{j+1}, \ldots, k_m) \overset{T_X}{\Longmapsto} \overset{T_X}{\Longmapsto} q_s\,\mathrm{Slv}(k_0, \ldots, k_{j-1}, 0, k_{j+1}, \ldots, k_m) \tag{8.45.3}$$

$$q_i\,\mathrm{Slv}(k_0, \ldots, k_{j-1}, k_j + 1, k_{j+1}, \ldots, k_m) \overset{T_X}{\Longmapsto} \overset{T_X}{\Longmapsto} q_r\,\mathrm{Slv}(k_0, \ldots, k_{j-1}, k_j + 1, k_{j+1}, \ldots, k_m) \tag{8.45.4}$$

Teraz vytvorme Turingov stroj T' ako množinové zjednotenie všetkých T-strojov T_X:

$$T' = \bigcup \{T_X \mid X \in Z'\} \tag{8.45.5}$$

Cvičenie 8.48. (a) Pre stroj T zo vzťahu (8.45.9) dokážte, že ak pre nejaké $v \in (OI^*)^{n_0}$, $w \in \{O,I\}^*$ platí $q_1 v \overset{T}{\Longmapsto} q_0 w$, tak existuje práve jedno také slovo $y \in \{O,I\}^*$ a práve jedno také slovo $z \in \{O,I\}^*$, že $q_1 v \overset{T_1}{\Longmapsto} q_u y$, $q_z y \overset{T}{\Longmapsto} q_v z$ a $q_v z \overset{T_2}{\Longmapsto} q_0 w$.

(b) Dokážte, že pre ľubovoľné $y, x_1, \ldots, x_n \in \mathbb{N}$ z rovnosti $\Phi_T^n(x_1, \ldots, x_n) = y$ vyplýva rovnosť $\Phi_Z^n(x_1, \ldots, x_n) = y$. ($Z$, T sú stále stroje z vety 8.45.)

(c) Dokážte, že pre ľubovoľné $y, x_1, \ldots, x_n \in \mathbb{N}$ z rovnosti $\Phi_Z^n(x_1, \ldots, x_n) = y$ vyplýva rovnosť $\Phi_T^n(x_1, \ldots, x_n) = y$.

Poznámka 8.49. Dôkaz vety 8.45 nám nielen dokazuje, že ku každému M-stroju Z a číslu $n \in \mathbb{N}$ existuje taký T-stroj, že $\Phi_Z^n = \Phi_T^n$, ale nám zároveň ukazuje, ako k danému Z a n tento stroj T zostrojiť. (Skutočný algoritmus na nájdenie stroja T však dostaneme až po doplnení niektorých detailov, ktoré sme väčšinov nechali do cvičení.)

Cvičenie 8.50. Nakreslite blokovú schému T-stroja počítajúceho funkciu I_1^1, ktorý dostaneme podľa dôkazu vety 8.45 z nejakého M-stroja Z počítajúceho I_1^1!

Cvičenie 8.51. Dokážte alebo vyvráťte nasledujúce tvrdenie: Ku každým rôznym dvom registrovým strojom Z_1, Z_2 a každým dvom rôznym číslam $m, n \in \mathbb{N}$ existuje taký Turingov stroj T, že $\Phi_{Z_1}^m = \Phi_T^m$ a $\Phi_{Z_2}^n = \Phi_T^n$!

Cvičenie 8.52. Dokážte alebo vyvráťte nasledujúce tvrdenie: Ku každým dvom rôznym T-strojom T_1, T_2 a každým dvom rôznym číslam $m, n \in \mathbb{N}$ existuje taký registrový stroj Z, že $\Phi_{T_1}^m = \Phi_Z^m$, $\Phi_{T_2}^n = \Phi_Z^n$.

Z vety 8.45 bezprostredne vyplýva

Veta 8.53. Každá (čiastočne) M-vypočítateľná funkcia je (čiastočne) T-vypočítateľná.

Veta 8.54. Každá (čiastočne) rekurzívna funkcia je (čiastočne) vypočítateľná na Turingovom stroji.

Poznámka 8.55. Vety 8.53 a 8.54 sa dajú obrátiť, ako ukážeme v kapitole 9. Aj výsledky nasledujúcich cvičení bude možné dostať z výsledkov kapitoly 9, prosíme však čitateľa, aby ich vyriešil bez používania týchto výsledkov.

Cvičenie 8.56. Množina všetkých (čiastočne) T-vypočítateľných funkcií tvorí normálny klon (čiastočných) funkcií na množine $\mathbb{N}$. Dokážte!

Cvičenie 8.57. Množina všetkých (čiastočne) T-vypočítateľných funkcií je uzavretá vzhľadom na operácie primitívnej rekurzie a minimalizácie. Dokážte!

(e) $\mathrm{tvst}^n(x_1,\ldots,x_n)$ bude pre ľubovoľné $n\in\mathbb{N}$ a všetky $x_1,\ldots,x_n$ znamenať číslo stavu

$$q_1OI^{x_1}OI^{x_2}O\ldots OI^{x_n}O$$

(f) $\mathrm{tvýp}_0$ bude charakteristickou funkciou predikátu $\mathrm{Tvýp}_0$.

(g) $\mathrm{tobs}(x)$ bude znamenať počet znakov I v poslednom člene konečnej postupnosti T-stavov, ktorá má číslo x; ak taká postupnosť neexistuje, potom $\mathrm{tobs}(x)=0$.

Dokážeme teraz primitívnu rekurzívnosť práve definovaných funkcií a predikátov. Na zjednodušenie ďalších zápisov zavádzame túto dohodu:

Dohoda 9.11. Zápis $\big(\exists x_1,x_2,\ldots,x_n\le f(y_1,y_2,\ldots,y_m)\big)$ budeme chápať ako skratku za zápis $\big(\exists x_1\le f(y_1,y_2,\ldots,y_m)\big)\big(\exists x_2\le f(y_1,y_2,\ldots,y_m)\big)\ldots\big(\exists x_n\le f(y_1,y_2,\ldots,y_m)\big)$. Táto dohoda platí pre ľubovoľnú funkciu f a ľubovoľné premenné $x_1,x_2,\ldots,x_n,\ y_1,y_2,\ldots,y_m$. Obdobným spôsobom budeme chápať aj zápis $\big(\forall x_1,x_2,\ldots,x_n\le f(y_1,y_2,\ldots,y_m)\big)$.

Cvičenie 9.12. (a) Dokážte, že pre každé $x\in\mathbb{N}$ platí:

$$\mathrm{Tstv}(x)=(\exists w,i,j\le x)\left(w\ \mathrm{MOD}\ \mathrm{p}_i\ne 0\ \wedge\ x=w\cdot \mathrm{p}_i^{2j+1}\right) \tag{9.12.1}$$

(b) Dokážte, že Tstv je primitívne rekurzívny predikát!

Cvičenie 9.13. Označme $\mathrm{Prech}_N(x,y,z)$ predikát znamenajúci „existujú T-stavy X, Y s číslami x, y, T-stroj Z s číslom z a taká T-inštrukcia $U\in Z$, ktorá má na štvtom mieste N, že $X\xrightarrow{\{U\}}Y$“ a obdobne definujeme $\mathrm{Prech}_L(x,y,z)$ a $\mathrm{Prech}_P(x,y,z)$.

(a) Dokážte, že platí (pre všetky $x,y,z\in\mathbb{N}$):

$$\begin{aligned}\mathrm{Prech}_N(x,y,z)\iff\big(&\mathrm{Tstv}(x)\ \wedge\ \mathrm{Tstv}(y)\ \wedge\ z\ \ne 0\ \wedge\\ &\wedge\ (\exists w,i,j,k,m,n\le x+y)\\ &\quad(x=w\cdot p_i^{2j+1}\cdot p_{i+1}^{2k}\ \wedge\ y=w\cdot p_i^{2m+1}\cdot p_{i+1}^{2n}\ \wedge\\ &\quad\wedge\ \mathrm{ex}(i,w)=0\ \wedge\ \mathrm{ex}(i+1,w)=0\ \wedge\\ &\quad\wedge\ \mathrm{ex}(\mathrm{c}(j,k),z)=3\,\mathrm{c}(n,m)+2)\big)\end{aligned} \tag{9.13.1}$$

$$\begin{aligned}\mathrm{Prech}_P(x,y,z)\iff\big(&\mathrm{Tstv}(x)\ \wedge\ \mathrm{Tstv}(y)\ \wedge\ z\ \ne 0\ \wedge\\ &\wedge\ (\exists w,i,j,k,m,n\le x+y)\\ &\quad(x=w\cdot p_i^{2j+1}\cdot p_{i+1}^{2k}\ \wedge\\ &\quad\wedge\ (y=w\cdot p_i^{2n}\cdot p_{i+1}^{2m+1}\ \vee\ \mathrm{pos}(1,y)=w\cdot p_i^{2n}\cdot p_{i+1}^{2m+1})\ \wedge\\ &\quad\wedge\ \mathrm{ex}(i,w)=0\ \wedge\ \mathrm{ex}(i+1,w)=0\ \wedge\\ &\quad\wedge\ \mathrm{ex}(\mathrm{c}(j,k),z)=3\,\mathrm{c}(n,m)+3)\big)\end{aligned} \tag{9.13.2}$$

$$\begin{aligned}\mathrm{Prech}_L(x,y,z)\iff\big(&\mathrm{Tstv}(x)\ \wedge\ \mathrm{Tstv}(y)\ \wedge\ z\ \ne 0\ \wedge\\ &\wedge\ (\exists w,i,j,k,m,n\le x+y)\\ &\quad((x=w\cdot p_i^{u}\cdot p_{i+1}^{2j+1}\cdot p_{i+2}^{2k}\ \vee\ \mathrm{pos}(1,x)=w\cdot p_i^{u}\cdot p_{i+1}^{2j+1}\cdot p_{i+z}^{2k})\ \wedge\\ &\quad\wedge\ y=w\cdot p_i^{2m+1}\cdot p_{i+1}^{u}\cdot p_{i+2}^{2n}\,\mathrm{ex}(i,w)=0\ \wedge\\ &\quad\wedge\ \mathrm{ex}(i+1,w)=0\ \wedge\ \mathrm{ex}(i+2,w)=0\ \wedge\\ &\quad\wedge\ \mathrm{ex}(\mathrm{c}(j,k),z)=3\,\mathrm{c}(n,m)+1)\big)\end{aligned} \tag{9.13.3}$$

$$\mathrm{Prech}(x,y,z)\iff\big(\mathrm{Prech}_N(x,y,z)\ \vee\ \mathrm{Prech}_P(x,y,z)\ \vee\ \mathrm{Prech}_L(x,y,z)\big) \tag{9.13.4}$$

(b) Dokážte, že predikáty Prech, Prech_L, Prech_P, Prech_N sú primitívne rekurzívne!

Rozšírime teraz pojmy primitívnej rekurzívnosti, rekurzívnosti a rekurzívnej spočítateľnosti z množín prirodzených čísel na niektoré množiny slov. V ďalších úvahách sa obmedzíme zásadne na abecedy a slová neobsahujúce symbol B, t.j. a_0. S obdobným ohraničením sme už definovali pojem T-riešiteľnosti hromadných úloh v kapitole 8, a tam sme tiež poukázali na jeho nepodstatnosť. Bolo však výhodné vzhľadom na zvláštnu úlohu symbolu B pri našej definícii Turingových strojov a ich výpočtov. V kapitole 7 sme takéto obmedzenie nepotrebovali, avšak pokiaľ preberáme pojmy z kapitoly 7 do tejto kapitoly, budeme ich používať s týmto ohraničením. (Teda napríklad pod regulárnou množinou odteraz rozumieme takú množinu, ktorej žiadne slovo neobsahuje symbol B.)

Nebudeme sa zaoberať ľubovoľnými množinami slov, ale iba množinami istého druhu, ktoré nazveme S-množinami.

Definícia 9.22. Množinu slov nazveme *S-množinou*, ak existuje taká konečná podmnožina A množiny $\overline{A} \setminus \{B\}$, že $X \subseteq A^*$.

Jedným z hlavných dôvodov, pre ktorý sa obmedzíme na S-množiny, je, že v praktických príkladoch sa takéto množiny slov vyskytujú najčastejšie. Niektoré ďalšie dôvody uvedieme neskôr.

Cvičenie 9.23. (a) Dokážte, že každá regulárna množina je S-množina.

(b) Dokážte, že množinové operácie zjednotenie, prienik a rozdiel dvoch S-množín sú S-množiny.

(c) Dokážte, že ak je X S-množina, sú aj X^*, X^+, X^R S-množiny.

Zavedieme teraz číslovanie slov a pomocou neho priradíme každej S-množine X istú podmnožinu $\mathrm{Num}(X)$ množiny $\mathbb{N}$.

Definícia 9.24. (a) *Číslom slova* $w = a_{i_0}a_{i_1}\ldots a_{i_n}$ nazveme číslo

$$\mathrm{num}(w) = 2^{2i_0} \cdot 3^{2i_1} \cdots \mathrm{p}_n{}^{2i_n} \tag{9.24.1}$$

(b) *Číselnou množinou priradenou S-množine X* budeme nazývať množinu

$$\mathrm{Num}(X) = \{\mathrm{num}(w) \mid w \in X\} \tag{9.24.2}$$

všetkých čísel prvkov množiny X.

Poznámka 9.25. V (9.24.1) sme zvolili exponenty $2i_0$, $2i_1$, ..., $2i_n$ namiesto i_0, i_1, ..., i_n iba preto, aby sa spôsob číslovania slov čo najviac zhodoval so spôsobom číslovania T-stavov. Niekedy budeme aj číslo T-stavu X označovať $\mathrm{num}(X)$.

Teraz už môžme definovať:

Definícia 9.26. (a) S-množinu X budeme nazývať *(primitívne) rekurzívnou*, ak je množina $\mathrm{Num}(X)$ (primitívne) rekurzívna.

(b) S-množinu X budeme nazývať *rekurzívne spočítateľnou*, ak je množina $\mathrm{Num}(X)$ rekurzívne spočítateľná.

Dôkaz nasledujúcich troch viet ponechávame čitateľovi.

Veta 9.27. (a) Množinové zjednotenie, prienik a rozdiel (primitívne) rekurzívnych S-množín sú (primitívne) rekurzívne S-množiny.

(b) Množinové zjednotenie a prienik rekurzívne spočítateľných S-množín sú rekurzívne spočítateľné S-množiny.

Veta 9.28. (a) Existuje rekurzívna S-množina, ktorá nie je primitívne rekurzívna.

(b) Existuje rekurzívne spočítateľná S-množina, ktorá nie je rekurzívna.

Veta 9.29. Nech X je rekurzívna S-množina, $Y \subseteq X$ a Y, $X \setminus Y$ sú rekurzívne spočítateľné S-množiny. Potom Y je rekurzívna S-množina.

Poznámka 9.30. Pokiaľ sa v predchádzajúcich vetách hovorí o množinovom rozdiele miesto o komplemente S-množiny, je to spôsobené tým, že nemôžeme vo všeobecnosti definovať komplement S-množiny.

Definícia 9.42. (a) S-zobrazenie $\varkappa$ budeme nazývať *(primitívne) rekurzívnym*, ak čiastočná funkcia $\mathrm{Num}(\varkappa)$ má (primitívne) rekurzívne zúplnenie a ak je obor definície $\mathrm{Num}(\varkappa)$ (primitívne) rekurzívna množina.

(b) S-zobrazenie $\varkappa$ budeme nazývať *čiastočne rekurzívnym*, ak je čiastočná funkcia $\mathrm{Num}(\varkappa)$ čiastočne rekurzívna.

Poznámka 9.43. Z obdobných dôvodov ako pri pojmoch (primitívnej) rekurzívnosti a rekurzívnej spočítateľnosti S-množín je možné aj pojmy (primitívnej) rekurzívnosti a čiastočnej rekurzívnosti S-zobrazení rozšíriť na ľubovoľné čiastočné zobrazenie $\varkappa$: $X^* \to Y^*$, kde X, Y sú ľubovoľné konečné neprázdne množiny.

Cvičenie 9.44. Dokážte, že následujúce S-zobrazenia sú primitívne rekurzívne:

(a) $\varkappa_1(w) = w^R$ pre všetky $w \in \{O, I\}^*$

(b) $\varkappa_2(w) = OwOwO$ pre všetky $w \in \{O, I\}^+$

(c) $\varkappa_3(w) = \begin{cases} O, & \text{ak } w^R \neq w \text{ a } w \in \{O, I\}^+ \\ I, & \text{ak } w^R \neq w \end{cases}$

Cvičenie 9.45. (a) Nech f je n-árna totálna funkcia a $\varkappa$ je také zobrazenie množiny $(OI^*)^n\, O$ do množiny OI^*O, že pre všetky $x_1, x_2, \ldots, x_n, y \in \mathbb{N}$ platí $f(x_1, \ldots, x_n) = y$ práve vtedy, keď

$$\varkappa(OI^{x_1}OI^{x_2}O \ldots OI^{x_n}O) = OI^yO$$

Dokážte, že funkcia f je (primitívne) rekurzívna práve vtedy, keď je S-zobrazenie $\varkappa$ (primitívne) rekurzívne!

(b) Vyslovte a dokážte obdobné tvrdenie pre čiastočnú funkciu f a čiastočnú rekurzívnosť!

Cvičenie 9.46. Nech $\varkappa$ je S-zobrazenie a M je množina všetkých slov tvaru $x\varkappa(x)$, pre ktoré je $\varkappa(x)$ definované.

(a) Dokážte, že M je S-množina!

(b) Zistite, či platia následujúce tvrdenia:

(1) Ak je $\varkappa$ (primitívne) rekurzívne S-zobrazenie, je M (primitívne) rekurzívna S-množina.

(2) Ak je $\varkappa$ čiastočne rekurzívne zobrazenie, je M čiastočne rekurzívna S-množina.

(3) Ak je M rekurzívna S-množina, je $\varkappa$ rekurzívne S-zobrazenie.

(c) Platí niektoré z tvrdení obrátených k tvrdeniam z bodu (b)?

Cvičenie 9.47. Nech $\varkappa_1$ je zobrazenie zložené z S-zobrazení $\varkappa_2$, $\varkappa_3$, t.j. pre každé slovo x platí $\varkappa_1(x) = \varkappa_2(\varkappa_3(x))$. Rozhodnite, či platia nasledujúce tvrdenia:

(a) Ak sú $\varkappa_2$, $\varkappa_3$ (primitívne) rekurzívne, je aj $\varkappa_1$ (primitívne) rekurzívne.

(b) Ak sú $\varkappa_2$, $\varkappa_3$ čiastočne rekurzívne, je aj $\varkappa_1$ čiastočne rekurzívne.

(c) Ak sú $\varkappa_1$, $\varkappa_3$ (primitívne) rekurzívne, je aj $\varkappa_2$ (primitívne) rekurzívne.

(d) Ak sú $\varkappa_1$, $\varkappa_2$ čiastočne rekurzívne, je aj $\varkappa_3$ čiastočne rekurzívne.

Nájdeme teraz nutnú a postačujúcu podmienku riešiteľnosti hromadnej úlohy na Turingovom stroji. (Obmedzíme sa pritom len na také hromadné úlohy, ktorých vstupná ani výstupná abeceda neobsahuje symbol B; nepodstatnosť tohto obmedzenia sme už zdôvodnili.)

Veta 9.53. Hromadná úloha $(A_1, A_2, Z, \varkappa)$ je T-strojove riešiteľná práve vtedy, keď existuje také čiastočne rekurzívne zobrazenie $\varkappa_1$, že pre všetky $x \in Z$ platí $\varkappa_1(x) = \varkappa(x)$.

Vo väčšine hromadných úloh $\mathcal{H} = (A_1, A_2, Z, \varkappa)$, s ktorými sa stretneme, je množina Z rekurzívna, a obvykle aj primitívne rekurzívna. Častý je aj prípad, že $\varkappa$ je zobrazenie množiny Z do A_2 a nie je iba čiastočné zobrazenie. Vtedy možno podmienku z vety 9.53 zjednodušiť podľa nasledujúcich viet.

Veta 9.54. Nech $\mathcal{H} = (A_1, A_2, Z, \varkappa)$ je hromadná úloha a Z je rekurzívne spočítateľná S-množina. Potom $\mathcal{H}$ je T-strojove riešiteľná práve vtedy, keď $\varkappa$ je čiastočne rekurzívne S-zobrazenie.

Veta 9.55. Nech $\mathcal{H} = (A_1, A_2, Z, \varkappa)$ je hromadná úloha a obor definície zobrazenia $\varkappa$ je rekurzívna S-množina. Potom hromadná úloha $\mathcal{H}$ je T-strojove riešiteľná práve vtedy, keď $\varkappa$ je rekurzívne S-zobrazenie.

Z vety 9.55 (ale samozrejme aj z vety 9.54) vyplýva:

Veta 9.56. (a) Všeobecný problém zastavenia (7.29.1) pre registrové stroje je T-strojove neriešiteľný.

(b) Pre každý M-stav X je problém zastavenia (7.28.1) pre tento M-stav T-strojove neriešiteľný.

(c) Existuje taký M-stroj Z, ktorého problém zastavenia (7.27.1) je T-strojove neriešiteľný.

Aby sme mohli vysloviť analogickú vetu aj pre T-stroje, definujme:

Definícia 9.57. (a) *Kódom T-inštrukcie* $(q_i a_j a_k X q_n)$, $X \in \{P, L, N\}$, bude slovo

$$QI^iAI^jAI^kXQI^nO$$

(b) *Kódom T-stroja* bude zreťazenie kódov všetkých jeho inštrukcií usporiadaných lexikograficky podľa indexov ich prvých dvoch členov.

(c) *Kódom T-stavu* $a_{i_0} a_{i_1} \dots a_{i_k} q_j a_{i_{k+1}} \dots a_{i_n}$ bude slovo

$$AOI^{i_0}AI^{i_1}A \dots AI^{i_k}QI^jAI^{i_{k+1}}A \dots AI^{i_n}O$$

Cvičenie 9.58. Definujte všeobecný problém zastavenia pre T-stroje; problém zastavenia pre T-stav X a problém zastavenia pre T-stroj T úplne analogicky k definiciám 7.27 až 7.29.

Veta 9.59. (a) Všeobecný problém zastavenia pre T-stroje je T-strojove neriešiteľný.

(b) Pre každý T-stav X je problém zastavenia pre tento T-stav T-strojove neriešiteľný.

(c) Existuje T-stroj, ktorého problém zastavenia je T-strojove nerišiteľný.

Cvičenie 9.60. Dokážte vetu 9.59.

Poznámka 9.61. Veta 9.59 sa často dokazuje bez prechodu k číselným funkciám tak, že sa zostrojí tzv. univerzálny Turingov stroj a preň sa urobí úvaha podobná dôkazu vety 2.30. Univerzálny Turingov stroj U dostáva ako vstup kód ľubovoľného T-stroja T a T-stavu X (najjednoduchšie: zreťazenie týchto kódov) a ako výstup dáva kód toho stavu, v ktorom končí výpočet stroja T zo stavu X, pokiaľ je tento výpočet konečný; ak je tento výpočet nekonečný, stroj U sa nezastaví.

Poznámka 10.7. Churchova téza 10.6 je v skutočnosti len špeciálny prípad tzv. Turingovej tézy. Podrobnejší rozbor Turingovej tézy a jej špeciálnych prípadov môže čitateľ nájsť v [6]. Nám však v ďalšom bude stačiť 10.6.

Churchovu tézu nie je možné matematicky dokázať; nie je totiž presným matematickým tvrdením, nakoľko sa v nej vyskytuje pojem algoritmickej (čiastočnej) vypočítateľnosti, ktorý nie je ani presne definovaný, ani nemáme axiómy opisujúce jeho vzťah k iným pojmom. Pri pokusoch o takúto definíciu alebo takéto axiómy už obvykle ohraničíme triedu všetkých algoritmov a opíšeme iba algoritmy istého typu. Takto teda dostaneme len (možno nový) špeciálny pojem algoritmickej (čiastočnej) vypočítateľnosti, o ktorom sa obvykle bez principiálnych ťažkostí dá dokázať, že je ekvivalentný s doterajšími pojmami. To môže byť ďalší dôvod pre uznanie Churchovej tézy, nie však jej matematický dôkaz.

Napriek tomu, že Churchovu tézu nie je možné matematicky dokázať, a teda ani používať v dôkazoch matematických viet, môže nám byť táto hypotéza pri dôkazoch viet z teórie algoritmov užitočná. Ak sa nám totiž podarí zdôvodniť platnosť nejakej matematickej vety použitím Churchovej tézy, je obvykle už len vecou rutiny prepracovať toto zdôvodnenie na presný matematický dôkaz. Napríklad ak pomocou Churchovej tézy dokážeme existenciu nejakého algoritmu, tak už obvykle bez principiálnych ťažkostí dokážeme existenciu Turingovho stroja, ktorý simuluje prácu tohto algoritmu. Pritom používanie Churchovej tézy nám často umožňuje vyhnúť sa nepríjemným technickým detailom, ktoré zneprehľadňujú presné dôkazy.

Budeme teraz charakterizovať niektoré pojmy týkajúce sa čiastočných funkcií na množine $\mathbb{N}$ a podmnožín množín $\mathbb{N}$ a $\mathbb{N}^n$ pomocou Churchovej tézy; správnosť týchto charakterizácií necháme na rozmyslenie čitateľovi. Znova upozorňujeme, že nejde o presné matematické vety. Ide však o praxou (v širšom zmysle) overené tvrdenia.

Tvrdenie 10.8. (a) n-árna funkcia f na $\mathbb{N}$ je rekurzívna práve vtedy, keď existuje algoritmus, ktorý každej n-tici $(x_1, \ldots, x_n) \in \mathbb{N}^n$ priradí hodnotu $f(x_1, \ldots, x_n)$.

(b) n-árna čiastočná funkcia f je čiastočne rekurzívna práve vtedy, keď existuje algoritmus, ktorý každej n-tici $(x_1, \ldots, x_n) \in \mathbb{N}^n$ priradí hodnotu $f(x_1, \ldots, x_n)$, pokiaľ je $f(x_1, \ldots, x_n)$ definované, a ktorý sa nekončí, ak $f(x_1, \ldots, x_n)$ nie je definované.

(c) Množina $M \subseteq \mathbb{N}^n$ je rekurzívna práve vtedy, keď existuje algoritmus, ktorý pre každé $(x_1, \ldots, x_n) \in \mathbb{N}^n$ dá odpoveď „áno“, ak $(x_1, \ldots, x_n) \in M$ a dá odpoveď „nie“, ak $(x_1, \ldots, x_n) \notin M$.

(d) Množina $M \subseteq \mathbb{N}^n$ je rekurzívne spočítateľná práve vtedy, keď existuje algoritmus, ktorý pre každé $(x_1, \ldots, x_n) \in \mathbb{N}^n$ dá odpoveď „áno“, ak $(x_1, \ldots, x_n) \in M$ a nedá žiadnu odpoveď, ak $(x_1, \ldots, x_n) \notin M$.

Cvičenie 10.9. Charakterizujte obdobným spôsobom rekurzívne a rekurzívne spočítateľné predikáty na množine $\mathbb{N}$ a rekurzívne a rekurzívne spočítateľné podmnožiny množiny $\mathbb{N}$.

Často je však výhodnejšia iná charakterizácia rekurzívne spočítateľných množín, ako je uvedená v 10.8. Použijeme pri nej trochu modifikované druhy algoritmov, ktoré budeme niekedy na odlíšenie nazývať generujúcimi algoritmami. Generujúci algoritmus bude obvykle pracovať bez zastavenia, a pritom pri niektorých krokoch vydávať výsledok; môže teda vydať celú postupnosť výsledkov. O množine všetkých vydaných výsledkov budeme hovoriť, že je generovaná týmto algoritmom, prípadne že ju tento algoritmus generuje. Potom platí

Tvrdenie 10.10. Množina $M \subseteq \mathbb{N}^n$ je rekurzívne spočítateľná práve vtedy, keď existuje generujúci algoritmus, ktorý generuje množinu M.

Zdôvodníme teraz ekvivalentnosť charakterizácií rekurzívnej spočítateľnosti z 10.8(d) a 10.10. Nech A je algoritmus z 10.8(d). Generujúci algoitmus B pre tú istú množinu M môže pracovať takto:

KAPITOLA 11

Rekurzívne reálne čísla

V tejto kapitole zavedieme rekurzívne reálne čísla. Sú to také reálne čísla, ktoré možno v istom (nižšie upresnenom) zmysle zadať efektívnym spôsobom. Rekurzívne reálne čísla tvoria spočítateľnú podmnožinu množiny všetkých reálnych čísel, je ich teda v zmysle mohutností „málo“. Napriek tomu len veľmi zriedka potrebujeme iné reálne konštanty než rekurzívne reálne čísla.

Dohodnime sa, že (ľubovoľné) reálne čísla budeme označovať gréckymi písmenami. Upozornime na jeden podstatný rozdiel medzi reálnymi a racionálnymi (resp. celými) číslami: Každé racionálne číslo možno zakódovať konečným počtom prirodzených čísel (keď chceme, aj jediným), prípadne slovom v nejakej abecede. Preto racionálne čísla možno bezprostredne použiť ako vstupy a výstupy pri výpočtoch. Možno najprv celé vstupy prečítať, potom urobiť výpočet, nakoniec celé výsledky zapísať a výpočet skončiť. (Takto by sme vedeli pracovať aj s niektorými iracionálnymi číslami, napríklad algebraickými.)

Naproti tomu (všetky) reálne čísla nemožno takto zadať pomocou prirodzených čisel, slov alebo iných konečných objektov. Prakticky zadávame a počítame len ich racionálne priblíženia, teoreticky ich uvažujeme prostredníctvom nejakých limitných procesov (napr. rezov alebo cauchyovských postupností). To sa bude týkať aj vstupu a výstupu. Ak sa nechceme celkom vzdať našich doterajších požiadaviek efektívnosti (s abstrakciou potenciálnej uskutočniteľnosti), musíme začať počítať, a aj písať výsledok, už počas čítania vstupu. Teoreticky výpočet neskončí nikdy, ale na výstupe postupne vzniká stále presnejšia aproximácia presného výsledku, pričom chybu vieme efektívne odhadnúť.

Uveďme ešte (neformálne) trochu iné hľadisko. Predstavme si, že pre niektoré reálne čísla máme k dispozícii nejaké zariadenia („čierne skrinky“ alebo orákulá, trochu podobné zariadeniam spomínaným v 6. kapitole), ktoré nám tieto čísla postupne určujú pomocou nejakých aproximácií. Spôsob tohto určovania presne nešpecifikujeme; môže ísť napríklad o postupne generovaný dekadický alebo binárny rozvoj zadávaného čísla. Iná možnosť je, že dostávame intervaly (s racionálnymi koncami), v ktorých dané číslo leží a ktorých dĺžku si zakaždým určíme. (Tento spôsob pokladáme za primeranejší.) Našou úlohou je vyrobiť obdobné zariadenia pre súčet, rozdiel, súčin a podiel daných čísel, prípadne pre ich logaritmy, odmocniny, hodnoty goniometrických funkcií a podobne. Môžeme ich vytvoriť z obdobných častí, z akých sme vytvárali registrové alebo Turingove stroje a z „čiernych skriniek“ pre argumenty. (Potrebné komponenty a „čierne skrinky“ máme k dispozícii v neobmedzenom množstve exemplárov, ale v každom konkrétnom zariadení ich smieme použiť len konečný počet.) Takto vytvorené zariadenia možno opakovane použiť obdobným spôsobom a dostať zariadenia, ktoré určujú ďalšie reálne čísla. Možno reprezentovať tým viac čísel, čím viac „čiernych skriniek“ máme na začiatku k dispozícii. Rekurzívne reálne čísla sú potom tie reálne čísla, ktoré možno reprezentovať vždy, aj keď nemáme k dispozícii žiadne „čierne skrinky“. (Vtedy naše zostrojené zariadenia sú obdobné ako registrové alebo Turingove stroje; len ich používame inak než v predchádzajúcich kapitolách.)

Ešte zdôvodníme, prečo je zadávanie intervalov, v ktorých číslo leží, primeranejší spôsob než zadávanie (napríklad) dekadického rozvoja. Samozrejme, je možné ľahko prechádzať od dekadického rozvoja k intervalom. Obrátený spôsob je však podstatne ťažší. Pritom aj keď máme „čierne skrinky“, ktoré dávajú rozvoje, vieme vo všeobecnosti skonštruovať len zariadenia, ktoré dávajú intervaly (Situácia sa môže zmeniť, ak budeme mať nejaké dodatočné informácie o zadávaných číslach.)

Príklad 11.1. Nech dve čierne skrinky zadávajú čísla $\frac{2}{7}$ a $\frac{5}{7}$. Máme z nich zostaviť zariadenie, ktoré určuje súčet zadávaných čísel. Ak nevieme, aké čísla sú naozaj zadávané, ale vidíme len (síce ľubovoľne dlhé, ale konečné) časti rozvojov, tak nevieme ani začať písať dekadický rozvoj súčtu.

Cvičenie 11.6. Budeme hovoriť (len v tomto cvičení) že funkcia $f\colon \mathbb{N} \to \mathbb{Z}$ faktoriálovo reprezentuje reálne číslo α, ak pre všetky $x \in \mathbb{N}$ platí

$$|\alpha \cdot x! - f(x)| < 1$$

(a) Dokážte, že všetky reálne čísla sú faktoriálovo reprezentovateľné.
(b) Zistite, ktoré reálne čísla sú faktoriálovo reprezentovateľné práve jednou funkciou.
(c) Zistite, ktoré reálne čísla sú faktoriálovo reprezentovateľné konečne mnohými funkciami.

Cvičenie 11.7. Nech $b > 1$ je celé číslo. Budeme hovoriť (len v tomto cvičení) že funkcia $f\colon \mathbb{N} \to \mathbb{Z}$ mocninovo reprezentuje (pri základe b) reálne číslo α, ak pre všetky $x \in \mathbb{N}$ platí

$$\left|\alpha \cdot x^b - f(x)\right| < 1$$

Riešte pre mocninovú reprezentovateľnosť otázky z predchádzajúceho cvičenia.

Cvičenie 11.8. Budeme hovoriť (len v tomto cvičení) že funkcia $f\colon \mathbb{N} \to \mathbb{Z}$ prvočíselne reprezentuje reálne číslo α, ak pre všetky $x \in \mathbb{N}$ platí

$$|\alpha \cdot \mathrm{p}_x - f(x)| < 1$$

pritom p_x znamená x-té prvočíslo. Riešte pre prvočíselnú reprezentovateľnosť obdoby otázok z cvičenia 11.6.

Definícia 11.9. Reálne číslo α nazveme *(primitívne) rekurzívnym reálnym číslom*, ak existuje (primitívne) rekurzívna funkcia $f\colon \mathbb{N} \to \mathbb{Z}$, ktorá ho reprezentuje.

Poznámka 11.10. Každé primitívne rekurzívne reálne číslo je zrejme aj rekurzívne. Uvedená definícia rekurzívnych reálnych čísel je všeobecne prijatá, resp. ekvivalentná s inými (navzájom ekvivalentnými) v literatúre uvádzanými definíciami. Ale pozor, uvedená definícia primitívne rekurzívnych reálnych čísel nie je všeobecne prijatá. Preto nasledujúce vety formulujeme pre oba pojmy aj vtedy, keď výsledky pre rekurzívnosť sú bezprostrednými dôsledkami výsledkov pre primitívnu rekurzívnosť.

Cvičenie 11.11. Dokážte, že reálne číslo α je (primitívne) rekurzívne práve vtedy, keď $|\alpha|$ je (primitívne) rekurzívne reálne číslo.

Cvičenie 11.12. Nech c, d sú racionálne čísla. Nazvime (len v tomto cvičení) reálne číslo α (primitívne) (c,d)-rekurzívnym, ak existuje (primitívne) rekurzívna funkcia $f\colon \mathbb{N} \to \mathbb{Z}$ taká, že pre všetky $x \in \mathbb{N}$ platí

$$f(x) - c < \alpha x < f(x) + d$$

Dokážte, že ak $d - c > 1$, tak každé reálne číslo je (primitívne) rekurzívne práve vtedy, keď je (primitívne) (c,d)-rekurzívne. Platí to aj pre $d - c = 1$?

Veta 11.13. Ak α, β sú (primitívne) rekurzívne reálne čísla, tak aj čísla $\alpha + \beta$, $\alpha\beta$, $\alpha - \beta$, sú (primitívne) rekurzívne. Ak ďalej $\beta \neq 0$, tak aj $\frac{\alpha}{\beta}$ je (primitívne) rekurzívne reálne číslo.

Dôkaz: Úvahy sa budú podobať dôkazom viet o limite súčtu, rozdielu, súčinu a podielu postupností. Rozdiely sú v tom, že nielen dokazujeme existenciu, ale zostrojujeme, a že aproximácie musia byť špeciálneho tvaru.

Nech rekurzívne funkcie f, g reprezentujú čísla α, β. Uvážme najprv súčet. Potrebujeme nájsť funkciu h, ktorá reprezentuje číslo $\alpha + \beta$ a vyjadriť ju tak, aby bola zrejmá jej (primitívna) rekurzívnosť. Definujme $h(x) = \left\lfloor \frac{f(3x)+g(3x)+1}{3} \right\rfloor$ (teda: $h(x)$ je celé číslo najbližšie k $\frac{f(3x)+g(3x)}{3}$; vzdialenosť neprevyšuje $\frac{1}{3}$). Funkcia h je (primitívne) rekurzívna a platí

$$\begin{aligned}
|(\alpha+\beta)\cdot x - h(x)| &\leq \frac{1}{3} + \left|(\alpha+\beta)\cdot x - \frac{f(3x)+g(3x)}{3}\right| = \\
&= \frac{1}{3} + \left|\frac{\alpha\cdot 3x - f(3x)}{3} + \frac{\beta\cdot 3x - g(3x)}{3}\right| \leq \\
&\leq \frac{1}{3} + \frac{|\alpha\cdot 3x - f(3x)|}{3} + \frac{|\beta\cdot 3x - g(3x)|}{3} < \frac{1}{3}+\frac{1}{3}+\frac{1}{3} = 1
\end{aligned}$$

Poznámka 11.15. (1) Implikácie (b) $\Longrightarrow$ (c), (c) $\Longrightarrow$ (a) platia aj pre primitívnu rekurzívnosť.

(2) Minimalizácia v predchádzajúcom dôkaze je regulárna len pre iracionálne α. Ani vtedy ju vo všeobecnosti nevieme ohraničiť, a preto veta nie je formulovaná pre primitívne rekurzívne reálne čísla.

(3) Napriek vete 11.14 je definícia rekurzívnych reálnych čísel v 11.9 technicky výhodnejšia než možné definície založené na podmienkach (b), (c) tejto vety. Prirodzenejšie sa prenáša na primitívnu rekurzívnosť a lepšie sa s ňou pracuje. Keby sme napríklad v dôkaze vety 11.13 (len pre rekurzívnosť!) pracovali s nejakou takto pozmenenou definíciou, museli by sme rozlišovať prípady racionálneho a iracionálneho výsledku operácií.

Pre neformálnu charakterizáciu rekurzívnosti sme používali Churchovu tézu, ktorá nám umožňovala ľahko a prehľadne, i keď nie formálne presne, robiť mnohé úvahy. Môžeme ju obdobne využívať i tu a rekurzívne reálne čísla môžeme pomocou nej charakterizovať napríklad takto:

Tvrdenie 11.16. Reálne číslo α je rekurzívne práve vtedy, keď ku každému kladnému racionálnemu číslu ε vieme zostrojiť interval s racionálnymi koncovými bodmi a dĺžkou (nanajvýš) ε, ktorý obsahuje α.

Bez dôkazu uvedieme nasledujúce tvrdenia, ktoré ukazujú, že skoro všetky konkrétne (!) reálne čísla vyskytujúce sa v bežných matematických úvahách sú (primitívne) rekurzívne. (Pod konkrétnymi číslami rozumieme *konštanty*, ktoré sa explicitne vyskytujú alebo môžu vyskytnúť.) V princípe dôkazy nie sú ťažké a čitateľ si ich môže spraviť sám. (Nevyskytnú sa v nich tie ťažkosti, ktoré bolo treba prekonať pri dôkazoch, že funkcie $\lfloor \alpha x \rfloor$ pre $\alpha = \mathrm{e}$ alebo niektoré $\alpha > 0$ algebraické sú primitívne rekurzívne. Tam sme totiž potrebovali hodnoty z jednej strany presne, čo definícia 11.3 nevyžaduje.)

Veta 11.17. (a) Všetky algebraické reálne čísla sú (primitívne) rekurzívne.

(b) Ak $f(x)$ je nenulový polynóm s (primitívne) rekurzívnymi reálnymi koeficientmi, tak všetky reálne korene polynomu $f(x)$ sú (primitívne) rekurzívne.

(c) Čísla $\mathrm{e} = 2.71828\ldots$ a $\pi = 3.14159\ldots$ sú (primitívne) rekurzívne.

(d) Ak α je (primitívne) rekurzívne reálne číslo, tak aj $\sin\alpha$, $\cos\alpha$, $\operatorname{arctg}\alpha$ sú (primitívne) rekurzívne reálne čísla. Ak okrem toho $\alpha > 0$, tak aj číslo $\ln\alpha$ je (primitívne) rekurzívne.

(e) Ak α, β sú kladné (primitívne) rekurzívne reálne čísla a $\beta \neq 1$, tak aj čísla $\log_\beta \alpha$ a α^β sú (primitívne) rekurzívne.

Cvičenie 11.18. Dokážte aspoň niektoré časti predchádzajúcej vety.

Pretože existuje len spočítateľne mnoho rekurzívnych reálnych čísel, zrejme existujú reálne čísla, ktoré nie sú (primitívne) rekurzívne. Dva zaujímavejšie príklady uvádzame nižšie.

Veta 11.19. Existuje rastúca zhora ohraničená primitívne rekurzívna postupnosť racionálnych čísel, ktorej limita nie je rekurzívne reálne číslo.

Dôkaz: Nech h je nejaká prostá unárna primitívne rekurzívna funkcia, ktorej obor hodnôt X je nerekurzívna množina. Položme

$$\alpha = \sum_{x=0}^{\infty} 8^{-h(x)} \tag{11.19.1}$$

Postupnosť čiastočných súčtov uvedeného nekonečného radu je zrejme rastúca a zhora ohraničená (napríklad číslom 2). Ostáva dokázať, že α nie je rekurzívne reálne číslo. Nech f je funkcia, ktorá reprezentuje α. Pomocou nej sa dá vyjadriť charakteristická funkcia množiny X takto:

$$\chi_X(y) = \big(\big(f\big(2^{3y+2}\big) + 5\big) \operatorname{MOD} 8\big) \operatorname{DIV} 4 \tag{11.19.2}$$

Skutočne, ak $y \in X$, tak $\lfloor \alpha \cdot 2^{3y+2} \rfloor \operatorname{MOD} 8 = 4$, a preto $f(2^{3y+2}) \operatorname{MOD} 8$ je 3, 4 alebo 5. Potom $\chi_X(y) = 0$. Obdobne zistíme, že pre $y \notin X$ platí $\chi_X(y) = 1$.

Z nájdeného vyjadrenia vidíme, že ak funkcia f je rekurzívna, tak aj χ_X je rekurzívna. To však neplatí, a preto číslo α nie je rekurzívne. □

Veta 11.20. Existuje rastúca zhora ohraničená primitívne rekurzívna postupnosť racionálnych čísel, ktorej limita je rekurzívne reálne číslo, ale nie primitívne rekurzívne reálne číslo.

Poznámka 11.27. Nech $\tau_i(x)$ je čas výpočtu stroja Z_i pre argument x. Netvrdili sme, že existuje primitívne rekurzívna funkcia F taká, že pre ľubovoľné stroje Z_1, Z_2 a všetky $x \in \mathbb{N}$ platí $\tau_3(x) \leq F(x, \tau_1(x), \tau_2(x))$. To ani nie je pravda, a to ani keď sa obmedzíme na stroje Z_1, Z_2, ktoré reprezentujú primitívne rekurzívne reálne čísla a počítajú v primitívne rekurzívnom čase. (Obdobne pre indexy 4, 5, 6 namiesto 3.)

Obdobné pozitívne výsledky by sme mohli dosiahnuť napríklad pre exponenciálne a goniometrické funkcie. Sformulujeme teraz niektoré negatívne výsledky.

Veta 11.28. Neexistuje algoritmus, ktorý by pre ľubovoľný registrový stroj Z zistil, či Z reprezentuje reálne číslo.

Dôkaz: Ako obvykle, použijeme redukciu k problému zastavenia. Ku každému registrovému stroju M priradíme registrový stroj Z tak, že M sa zastaví (pri výpočte $\Phi_M(0)$) práve vtedy, keď Z nereprezentuje žiadne reálne číslo. Ak sa M nezastaví, tak Z reprezentuje číslo 0.

Dosiahneme to tak, že pre vstup x stroj Z najprv simuluje x krokov stroja M (alebo menej, ak sa M skôr zastaví). Ak sa stroj M nezastavil, stroj Z obnoví obsah registra R_1 a ukončí výpočet; v R_0, R_2 zostanú nuly. Ak sa M zastavil, dá Z jedničku do registra R_0 a tiež ukončí výpočet. (Stroj Z vôbec nepoužíva register R_2; register R_0 použije najviac raz, a to uvedeným spôsobom.) Treba si uvedomiť, že stroj Z vždy počíta ohraničenú funkciu. Pritom jediná ohraničená funkcia, ktorá reprezentuje nejaké reálne číslo, je funkcia identicky rovná nule (ktorá reprezentuje číslo 0). □

Veta 11.29. Neexistuje algoritmus, ktorý by pre ľubovoľný registrový stroj Z zistil, či Z reprezentuje číslo 0, a to ani vtedy, keď dopredu vieme, že Z reprezentuje nejaké (rekurzívne) reálne číslo.

Dôkaz: Prispôsobíme predchádzajúcu konštrukciu, ale použijeme len stroje M, ktoré robia aspoň jeden krok.

Stroj Z bude reprezentovať číslo 0, ak sa M nezastaví, a bude reprezentovať číslo $\frac{1}{k}$, ak sa M zastaví po presne k krokoch. Pre vstup x bude stroj Z zasa simulovať (najviac) x krokov výpočtu stroja M. Ak sa stroj M zastaví, tak Z „bude poznať“ k, a bude môcť vypočítať celé číslo najbližšie k $\frac{x}{k}$. Inak môže Z dať výsledok 0. (Obsah registra R_1 treba obnovovať vždy.) □

Veta 11.30. Neexistuje algoritmus, ktorý by pre ľubovoľné dva registrové stroje zistil, či reprezentujú to isté číslo, a to ani vtedy, keď dopredu vieme, že každý z nich reprezentuje nejaké (rekurzívne) reálne číslo.

Dôkaz: Podľa predchádzajúcej vety to nevieme urobiť ani v špeciálnom prípade, keď vieme, že jeden zo strojov reprezentuje číslo 0. □

Sformulujeme predchádzajúce vety ešte raz, trochu menej presne, ale názornejšie. (Túto formuláciu môžeme prijať bez výhrad, ak prijmeme konvenciu, že rekurzívne reálne číslo zásadne určujeme príslušným reprezentujúcim registrovým strojom.) Pridáme niekoľko ďalších tvrdení, kotré by sa dali dokázať podobne, alebo sa dajú redukovať k predchádzajúcim.

Veta 11.31. Žiaden z nasledujúcich problémov nie je algoritmicky riešiteľný:

(a) Zistiť, či dané rekurzívne reálne číslo je rovné nule.
(b) Zistiť, či dané dve rekurzívne reálne čísla sú si rovné.
(c) Zistiť, či dané rekurzívne reálne číslo je kladné.
(d) Zistiť, či dané rekurzívne reálne číslo je nezáporné.
(e) Zistiť, či dané rekurzívne reálne číslo je záporné.

Na rozdiel od predchádzajúcej vety, platí:

Veta 11.32. Nasledujúce problémy sú algoritmicky riešiteľné:

(a) Zistiť, či dané nenulové rekurzívne reálne číslo je kladné.
(b) Zistiť, ktoré z dvoch navzájom rôznych rekurzívnych reálnych čísel je väčšie.

Dôkaz: Na riešenie treba jednoducho postupne počítať stále kratšie intervaly obsahujúce uvažované čísla. Skončiť možno vtedy, keď sú tieto intervaly disjunktné; potom je rozhodnutie ľahké. □

KAPITOLA 12

Malý univerzálny registrový stroj

Hlavným a v podstate jediným výsledkom tejto kapitoly je

Veta 12.1. Existuje registrový stroj s 34 inštrukciami, ktorý počíta univerzálnu čiastočnú funkciu pre množinu všetkých unárnych čiastočne rekurzívnych funkcií.

Jeden takýto stroj naozaj zostrojíme; je znázornený na obrázku 12.1. Podľa počtu jeho inštrukcií ho označujeme U_{34}. Tento stroj *nepočíta* čiastočnú funkciu muniv^2 z kapitoly 6, ale inú univerzálnu čiastočnú funkciu. Najprv stroj a jeho činnosť zhruba popíšeme a až potom pridáme niektoré detaily dôkazu.

Stroj U_{34} nebude *bezprostredne* simulovať všetky registrové stroje, ale iba tzv. špeciálne registrové stroje, ktoré len pre tento účel zavedieme. Očíslujeme ich zasa špeciálnym spôsobom vhodným na dekódovanie týchto čísel malým počtom inštrukcií. Súhrnne povedané, technické detaily prispôsobíme cieľu nájsť univerzálny registrový stroj s čo najmenším počtom inštrukcií. (Nebudeme však tomuto cieľu prispôsobovať samotnú definíciu univerzálneho registrového stroja, teda napríklad nezavedieme špeciálne vstupné a výstupné kódovanie. Tým by sa dal počet inštrukcií ešte ďalej znížiť, pričom by aj takýto stroj mal neriešiteľný problém zastavenia.) Hneď upozornime čitateľa, že nie je známy najmenší možný počet inštrukcií univerzálneho registrového stroja (ani univerzálneho Turingovho stroja). Známe horné a dolné odhady sa ešte od seba dosť líšia.

Stroj U_{34} je znázornený na obrázku 12.1. Pre svoj výpočet vždy dostáva v registri R_1 číslo simulovaného stroja (možno ho chápať ako číslo unárnej čiastočne rekurzívnej funkcie) a v registri R_2 vstup pre simulovaný stroj (argument funkcie). Výsledok dáva v registri R_0. Stroj U_{34} pracuje v cykle; v jednom behu vždy najprv určí číslo simulovanej inštrukcie; to robí čítač inštrukcií. Číslo inštrukcie sa uloží v registri R_5. Potom sa z čísla inštrukcie v dekóderi určí operačný kód (t.j. požadovaná činnosť) a číslo nového vnútorného stavu, ktoré sa uloží do R_4. Operačný kód (jeden z troch) je určený výstupnou šípkou. Potom simulujúci blok bezprostredne vykoná činnosť simulovaného stroja a ešte prípadne modifikuje číslo nasledujúcej inštrukcie; môže ho zmeniť len o 1. Táto modifikácia sa neprejaví zmenou obsahu registra R_4, ale len voľbou inej výstupnej šípky. Nakoniec (možno) nasleduje test ukončenia a ak sa výpočet simulovaného stroja neskončil, začína nový beh cyklu.

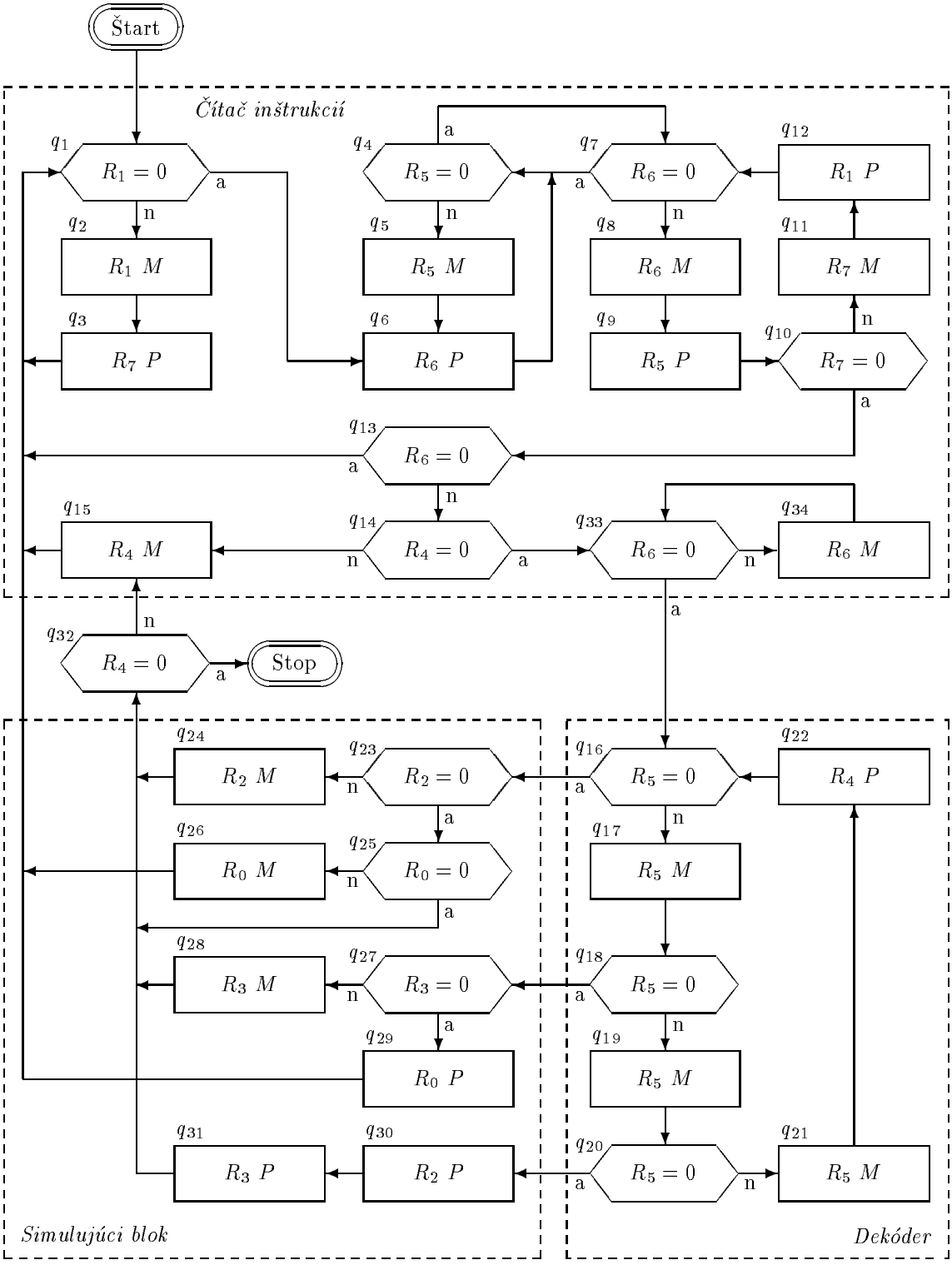
Špeciálne registrové stroje používajú tri registre: R_0, R_2 a R_3. Vstup čítajú z registra R_2, výsledok dávajú v registri R_0. Keby sme povolili vstupné a výstupné kódovanie mocninami dvojky (teda: namiesto x by bolo vstupom, resp. výstupom 2^x), stačili by na výpočty ľubovoľných (unárnych) čiastočne rekurzívnych funkcií dva registre. V podstate tak pracujú i špeciálne stroje, len na začiatku výpočtu zakódujú x na 2^x a na konci zasa dekódujú $2^{f(x)}$ na $f(x)$. Dôležitá zmena však je, že v blokových schémach špeciálnych registrových strojov nemožno kombinovať pripočítavacie, odčítavacie a rozhodovacie bloky ľubovoľne, ale len v skupinách zodpovedajúcich trom častiam simulačného bloku. (Treba zachovať čísla registrov, ale pochopiteľne nie čísla vnútorných stavov.) Každá takáto skupina inštrukcií má jeden vstup a najviac dva výstupy. Ak má dva výstupy, musia sa indexy zodpovedajúcich vnútorných stavov líšiť o jedničku (pričom výstupu na nulu zodpovedá vyšší). Dá sa však overiť, že tieto obmedzenia nevadia.

Pri číslovaní špeciálnych registrových strojov sa používa funkcia

$$F(x,y) = y\text{-tý nenulový člen postupnosti } \{(x+1) \text{ MOD } y\}_{y=0}^{\infty}$$

Pomocou čínskej zvyškovej vety sa dá ukázať, že pre každú konečnú postupnosť $(a_1, a_2, \ldots, a_k)$ kladných celých čísel existuje x také, že pre všetky $i = 1, \ldots, k$ platí $F(x,i) = a_i$.

Priraďme trom typom skupín inštrukcií spomínaným vyššie čísla 0, 1, 2 v poradí zhora nadol podľa simulačného bloku na obrázku 12.1. Číslo x simulovaného stroja je definované tak, že ak i



Obr. 12.1. Univerzálny registrový stroj U_{34}

0: $(q_1;\ 0, 99, 2, 0)$	17: $(q_{15};\ 2, 99, 2, 1)$	34: $(q_{12};\ 1, 99, 4, 0)$
1: $(q_2;\ 1, 99, 2, 0)$	18: $(q_{12};\ 2, 99, 1, 1)$	35: $(q_{13};\ 1, 99, 3, 0)$
2: $(q_3;\ 1, 99, 1, 0)$	19: $(q_{13};\ 2, 99, 0, 1)$	36: $(q_{15};\ 1, 99, 4, 1)$
3: $(q_2;\ 2, 99, 1, 0)$	20: $(q_{15};\ 2, 99, 1, 2)$	37: $(q_{12};\ 1, 99, 3, 1)$
4: $(q_3;\ 2, 99, 0, 0)$	21: $(q_{12};\ 2, 99, 0, 2)$	38: $(q_{13};\ 1, 99, 2, 1)$
5: $(q_2;\ 3, 99, 0, 0)$	22: $(q_{14};\ 1, 99, 0, 2)$	39: $(q_{15};\ 1, 99, 3, 2)$
6: $(q_4;\ 2, 99, 0, 0)$	23: $(q_{10};\ 0, 99, 0, 2)$	40: $(q_{12};\ 1, 99, 2, 2)$
7: $(q_5;\ 2, 99, 1, 1)$	24: $(q_{11};\ 0, 99, 0, 1)$	41: $(q_{13};\ 1, 99, 1, 2)$
8: $(q_{14};\ 2, 99, 0, 1)$	25: $(q_6;\ 0, 99, 1, 2)$	42: $(q_{15};\ 1, 99, 2, 3)$
9: $(q_{10};\ 1, 99, 0, 1)$	26: $(q_7;\ 0, 99, 1, 1)$	43: $(q_{12};\ 1, 99, 1, 3)$
10: $(q_{11};\ 1, 99, 0, 0)$	27: $(q_8;\ 0, 99, 2, 2)$	44: $(q_{13};\ 1, 99, 0, 3)$
11: $(q_6;\ 1, 99, 1, 1)$	28: $(q_{10};\ 0, 99, 2, 1)$	45: $(q_{15};\ 1, 99, 1, 4)$
12: $(q_7;\ 1, 99, 1, 0)$	29: $(q_{11};\ 0, 99, 2, 0)$	46: $(q_{12};\ 1, 99, 0, 4)$
13: $(q_8;\ 1, 99, 2, 1)$	30: $(q_6;\ 0, 99, 3, 1)$	47: $(q_{14};\ 0, 99, 0, 4)$
14: $(q_{10};\ 1, 99, 2, 0)$	31: $(q_7;\ 0, 99, 3, 0)$	48: $(q_9;\ 0, 99, 0, 4)$
15: $(q_{12};\ 2, 99, 2, 0)$	32: $(q_8;\ 0, 99, 4, 1)$	
16: $(q_{13};\ 2, 99, 1, 0)$	33: $(q_{10};\ 0, 99, 4, 0)$	

Obr. 12.3. Ukážka výpočtu bloku prvej etapy pre $x = 2$.

Definícia 12.7. Budeme hovoriť, že x je *F-číslom* $R3a$-stroja X, ak pre každú jeho inštrukciu A platí $F(x, i) = 3k + \mathrm{typ}(A)$, kde i je prvá zložka inštrukcie, k je jej tretia zložka a $\mathrm{typ}(A)$ je 0, 1 alebo 2, podľa toho, či je A prvého, druhého, alebo tretieho tvaru z definície 12.6.

Lema 12.8. Každý $R3a$-stroj má aspoň jedno F-číslo.

Dôkaz: Zostavíme konečnú postupnosť kladných čísel, kde i-ty člen bude zodpovedať inštrukcii začínajúcej q_i; chýbajúce členy doplníme ľubovoľne. Potom aplikujeme vetu 12.3. □

Poznámka 12.9. Jeden $R3a$-stroj má nekonečne mnoho F-čísel, a aj obrátene, jedno číslo môže zodpovedať viacerým $R3a$-strojom. (Všetky ale majú rovnakú komponentu obsahujúcu inštrukciu s q_1 na prvom mieste. Stroj U_{34} bude simulovať vlastne len túto komponentu.)

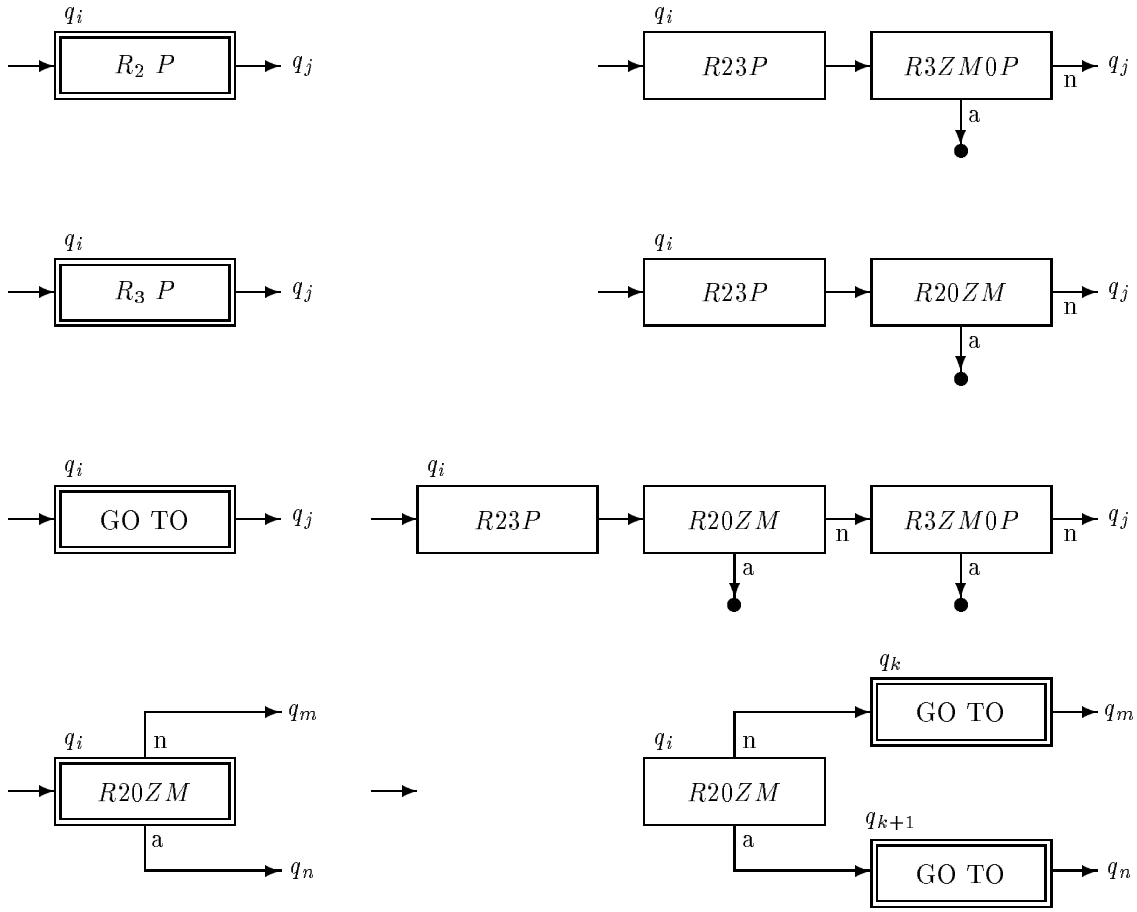
Veta 12.10. Pre každú unárnu čiastočne rekurzívnu funkciu existuje $R3a$-stroj, ktorý ju počíta.

Dôkaz: Výpočet $R3a$-stroja, ktorý počíta funkciu f, môže prebiehať v nasledujúcich troch etapách:

(1) Zakódovanie vstupu; z čísla x sa vypočíta 2^x.
(2) Výpočet $2^{f(x)}$ z hodnoty 2^x; nech táto časť neskončí, ak $f(x)$ nie je definované.
(3) Dekódovanie výstupu; z čísla $2^{f(x)}$ sa vypočíta $f(x)$.

Pre prvú a tretiu etapu, ktoré nezávisia od f, uvádzame blokové schémy príslušných častí $R3a$-stroja na obrázkoch 12.2 a 12.4; na obrázkoch 12.3 a 12.5 sú ukážky ich výpočtov. Pre sprehľadnenie schém sú šípky, ktoré sa nepoužijú v žiadnom výpočte, skrátené a nasmerované do znaku •; ľahko zistíme, kam pôvodne smerovali. V ukážkach výpočtov sa uvádza i register R_1 (s obsahom 99; pri simulácii na U_{34} číslo stroja), hoci ho $R3a$-stroje nepoužívajú.

V druhej etape sa používajú len dva registre, R_2 a R_3 a pracuje sa v podstate s jedinou celočíselnou premennou; jej počiatočná hodnota je 2^x. My si ju ale predstavíme ako $2^x \cdot 3^0 \cdot 5^0 \cdots \mathrm{p}_n^0$ a budeme chápať exponenty v rozklade ako obsahy registrov. Ich zmenám o jedničku zodpovedá násobenie alebo delenie vhodným číslom a ich testom na nulu testy na deliteľnosť (uvedenými prvočíslami). Všetko to zrejme možno urobiť pomocou základných inštrukcií registrových strojov. Možno to ale urobiť aj pomocou $R3a$-inštrukcií. Detaily nechávame čitateľovi. Zostavíme však z $R3a$-inštrukcií niektoré makroinštrukcie, ktoré mu pri tom môžu pomôcť. Sú znázorné na obrázku 12.6; vľavo je symbol pre makroinštrukciu, vpravo jej rozpis. Dve schémy na obrázku 12.7 ukazujú použitie makroinštrukcií pre výpočet $x \mapsto ax + b$ a delenie číslom d s rozlíšením zvyškov miestom výstupu. Na obrázku 12.6 vidíme pripočítanie jedničky do registra R_2, resp. R_3. (Bohužiaľ nevieme tak jednoducho urobiť odčítanie jedničky a test na nulu.) Ďalej je tam inštrukcia GoTo, ktorá



Obr. 12.6. Makroinštrukcie $R3a$-strojov.

Cvičenie 12.11. Nakreslite takú blokovú schému (bez pripísaných vnútorných stavov) z blokov v definícii 12.6, v ktorej aspoň niektoré prvky musia byť chápané ako makroinštrukcie. (Nech inak nie je možné vyhovujúcim spôsobom pripísať vnútorné stavy.)

Cvičenie 12.12. Nájdite nejaký $R3a$-stroj, ktorý počíta funkciu $f(x) = 2x+1$, napíšte podmienky pre jeho číslo a odhadnite toto číslo. (Netreba hľadať najmenšie riešenie, ale nezväčšujte číslo stroja zbytočne.)

Literatúra

[1] Brookshear, J. G.: *Formal languages, automata, and complexity*, Benjamin/Cummings, Redwood City, 1989.

[2] Cohen, D. E.: *Computability and logic*, Ellis Horwood, Chichester, 1987.

[3] Cutland, N. J.: *Computability*, Cambridge University Press, 1980.

[4] Davis, M.: *Computability and Unsolvability*, McGraw-Hill, 1958.

[5] Kuratowski, C., Mostowski, A.: *Teoria mnogości*, Warszawa, 1952.

[6] Maľcev, A.: *Algoritmy i rekursivnyje funkcii*, Moskva, 1965.

[7] Minsky, M. L.: *Computation: Finite and Infinite Machines*, Prentice Hall, 1967.

[8] Rogers, H.: *Theory of recursive functions and effective computability*, McGraw-Hill, New York, 1967.

[9] Weihrauch, K.: *Computability*, Springer-Verlag, Berlin, 1987.

Obsah