Latvijas Universitāte
Fizikas un matemātikas fakultāte
Matemātiskās analīzes katedra

Jānis Buls

IEVADS ALGORITMU TEORIJĀ

Lekciju konspekts — 2007

Ievads

Sākumā algoritmu teorija attīstījās saistībā ar teorētiskās matemātikas iekšējām vajadzībām. Pirmsākums saistāms ar matemātisko loģiku, matemātikas pamatiem, algebru, ğeometriju un matemātisko analīzi. Saprotams, visas šīs matemātikas nozares šodien algoritmu teorijas atziņas izmanto daudz plašāk nekā pirmsākumos kaut vai tā vienkāršā iemesla dēļ, ka daudzas skaitliska rakstura problēmas mūsdienās tiek risinātas izmantojot datoru; tātad lietojot dažnedažādus algoritmus.

Algoritmu teorijas pamatrezultātus šodien praktiski izmanto visās matemātikas nozarēs, tai skaitā gan diferenciālvienādojumu teorijā, gan kriptogrāfijā, gan varbūtību teorijā un statistikā. Sakarā ar e-pārvaldes un e-komercijas ieviešanu aktualizējas datu aizsardzības problēmas, kas būtiski balstās uz algoritmu teorijas atziņām. Tā šobrīd ir viena no daudzajām algoritmu teorijas lietojumu sfērām.

Pagājušā gadsimta četrdesmitajos gados iezīmējās jauna algoritmu teorijas lietojumu sfēra. Tā saistāma ar datortehnoloğiju attīstību. Racionāla adresu mašīna samērā precīzi apraksta moderna datora darbības principus. Taču tas jau ir sešdesmito gadu sasniegums.

Mūsdienās algoritmu teorijas atziņas izmanto arī lingvistikā, ekonomikā, molekulārbioloğijā, kā arī smadzeņu darbības modelēšanā (gan izpratnes vairošanai, gan mākslīgā intelekta radīšanas nolūkos).

Kurss iepazīstina ar algoritma jēdziena eksplikāciju izmantojot jēdzienu par adresu mašīnām, Tjūringa mašīnām, Markova normāliem algoritmiem, primitīvi rekursīvām funkcijām un daļēji rekursīvām funkcijām. Algoritmu teorija nenodarbojas nedz ar reālu skaitļotāju izpēti, nedz arī aplūko konkrētas programmēšanas valodas. To interesē datoru teorētiskās iespējas un ierobežotība. Mūsdienās tā ir neatņemama teorētiskās un praktiskās datorzinātnes sastāvdaļa.

Kursa mērķis — iepazīstināt ar algoritmu teorijas pamatjēdzieniem un metodēm, kuras plaši lieto citās disciplīnās.

Apzīmējumi

$\neg$ — negācija,
$\vee$ — disjunkcija, $\wedge$ — konjunkcija,
$\Rightarrow$ — implikācija, $\Leftrightarrow$ — ekvivalence,
$\mathfrak{A} \sim a$ — izteikums $\mathfrak{A}$ ir aplams,
$\mathfrak{A} \sim p$ — izteikums $\mathfrak{A}$ ir patiess,
$\exists$ — eksistences kvantors, $\forall$ — universālkvantors,
$\exists! x\ P(x)$ — eksistē viens vienīgs tāds x, kam izpildās nosacījums $P(x)$,

$x \in X$ — elements x pieder kopai X jeb x ir kopas X elements,
$A \subseteq B$ — kopa A ir kopas B apakškopa,
$A \cup B$, $A \cap B$, $A \setminus B$ — kopu A un B apvienojums, šķēlums, starpība,
$\min K$ — kopas K minimālais elements,
$\max K$ — kopas K maksimālais elements,

$\leftharpoondown$, $\rightharpoondown$ — vienādības saskaņā ar definīciju,
$\overline{1,n} \leftharpoondown \{1, 2, \ldots, n\}$; $\overline{k,n} \leftharpoondown \{k, k+1, \ldots, n\}$, te $k \leq n$,
$\mathbb{Z}$ — veselo skaitļu kopa, $\mathbb{Z}_+ \leftharpoondown \{x \mid x \in \mathbb{Z} \wedge x > 0\}$,
$\mathbb{N} \leftharpoondown \mathbb{Z}_+ \cup \{0\}$, $\mathbb{N}_- \leftharpoondown \mathbb{Z} \setminus \mathbb{Z}_+$,
$\mathbb{P}$ — visu pirmskaitļu kopa,
$\mathbb{Q}$ — racionālo skaitļu kopa,
$\mathbb{R}$ — reālo skaitļu kopa, $\mathbb{C}$ — komplekso skaitļu kopa,
$|K|$ — kopas K apjoms,
$\aleph_0$ — kopas $\mathbb{N}$ apjoms, $\mathfrak{c}$ — reālo skaitļu kopas $\mathbb{R}$ apjoms,

$\langle x, y \rangle \leftharpoondown (x, y) \leftharpoondown \{\{x\}, \{x, y\}\}$,
$x_1 x_2 \ldots x_n \leftharpoondown \langle x_1, x_2, \ldots, x_n \rangle \leftharpoondown (x_1, x_2, \ldots, x_n) \leftharpoondown ((x_1, x_2, \ldots, x_{n-1}), x_n)$,
$A_1 \times A_2 \times \ldots \times A_n \leftharpoondown \{(x_1, x_2, \ldots, x_n) \mid \forall i \in \overline{1,n}\, (x_i \in A_i)\}$, A^n, $|u|$,
$f : x \mapsto y$, $f : X \mathrel{-\!\!\circ\!\!\!\rightarrow} Y$, $X \overset{f}{\mathrel{-\!\!\circ\!\!\!\rightarrow}} Y$,
$\mathrm{Dom}(f) \leftharpoondown \{x \mid \exists y \in Y\ (f : x \mapsto y)\}$, $\mathrm{Ran}(f) \leftharpoondown \{y \mid \exists x \in X\ (f : x \mapsto y)\}$,
$f : X \to Y$, $X \xrightarrow{f} Y$, $f : X \twoheadrightarrow Y$, $f : X \hookrightarrow Y$,
$\mathrm{pr}_i \varrho$, $\mathrm{pr}_i g$,
A^+, λ, A^*, u^n,
$\lfloor x \rfloor \leftharpoondown \max\{t \mid t \leq x\ \wedge\ t \in \mathbb{Z}\}$,

$$\sum_{i=k}^{m} a_i \leftharpoondown\!\!\!\!\!\!\!\!\overline{}\ a_k + a_{k+1} + \ldots + a_m,$$
$$\prod_{i=k}^{m} a_i \leftharpoondown\!\!\!\!\!\!\!\!\overline{}\ a_k a_{k+1} \ldots a_m,$$

$a \searrow b$ — skaitlis b ir skaitļa a daudzkārtnis,
$a \nmid b$ — skaitlis b nav skaitļa a daudzkārtnis,

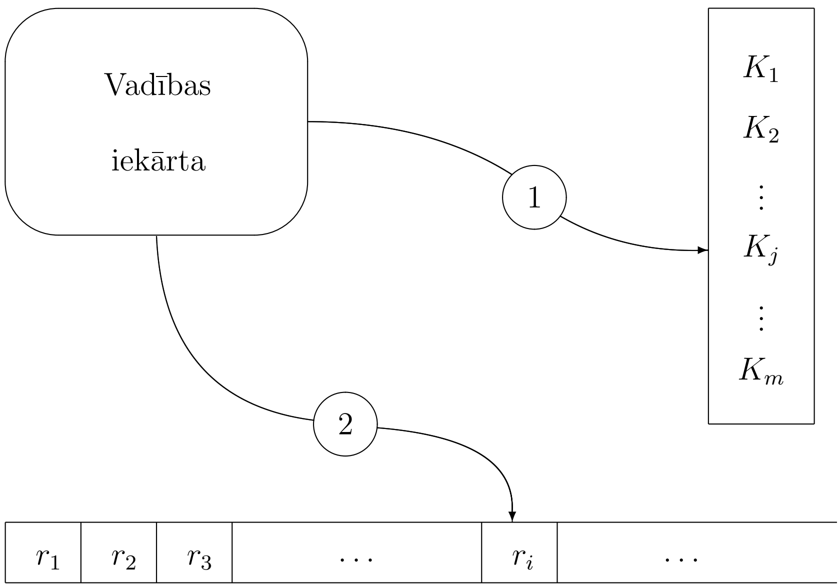
$\square$ — pierādījuma sākums,
$\blacksquare$ — pierādījuma beigas;
$\Rightarrow$ — implikācijas zīmi pierādījuma sākumā mēs izmantojam, lai norādītu, ka tagad sākas teorēmas nepieciešamā nosacījuma pierādījums,
$\Leftarrow$ — šo zīmi pierādījumos mēs izmantojam, lai norādītu, ka tagad sākas teorēmas pietiekamā nosacījuma pierādījums.

1. Racionālas adresu mašīnas (RAM)

Racionālas adresu mašīnas (RAM), piemēri. RAM izrēķināmas funkcijas, piemēri.

1.1. RAM fizikālais modelis

Mūsu mērķis — konstruēt idealizētu datoru. Mēs vēlamies konstruēt iekārtu, kas būtu pēc iespējas vienkāršāka, un tai pašā laikā spētu veikt visus aprēķinus, ko mūsdienās spēj veikt dators. Mēs aprakstīsim racionālas adresu mašīnas (turpmāk lietosim saīsinājumu: RAM) sastāvdaļas, līdzīgi tam kā mēs aprakstītu vieglās automašīnas uzbūvi pa sastāvdaļām, lai varētu paskaidrot, kas ir vieglā automašīna.



1. zīm. RAM principiālā darbības shēma.

Būtiska RAM sastāvdaļa ir bezgalīga atmiņas iekārta, kas sadalīta šūnās:

$$R_1, R_2, \ldots, R_i, \ldots$$

Katrā šūnā R_i var ierakstīt patvaļīgu naturālu skaitli $r_i \in \mathbb{N}$. Gadījumā, ja katrā šūnā R_i ir ierakstīts kāds naturāls skaitlis $r_i \in \mathbb{N}$, mēs teiksim, ka RAM *konfigurācija* ir (r_n). Tas nozīmē, ka fiksēta naturālo skaitļu virkne

$$r_1, r_2, r_3, \ldots, r_n, \ldots$$

Otra RAM būtiska sastāvdaļa ir iekārta, kurā var ievietot (ierakstīt) programmu. Programma — tā ir galīga komandu virkne

$$K_1, K_2, \ldots, K_\tau \,.$$

Visbeidzot trešā sastādaļa ir vadības iekārta, kas funkcionē saskaņā ar ievietoto programmu. Shematiski tas attēlots 1. zīmējumā. Te RAM vadības iekārtas 1. galviņa aplūko j–to komandu K_j un 2. galviņa aplūko i–to atmiņas šūnu R_i, kurā ierakstīts skaitlis r_i.

RAM programmu drīkst rakstīt izmantojot tikai četru veidu komandas:

$$Z(n),\ S(n),\ T(m,n),\ J(m,n,q)\,.$$

Šeit $(m, n, q) \in \mathbb{Z}_+^3$.

Tagad aprakstīsim RAM vadības iekārtas darbību. RAM darbojas diskrētos laika momentos

$$0, 1, 2, \ldots, t, \ldots$$

Pieņemsim, ka laika momentā t RAM konfigurācija ir (r_n) un 1. galviņa šai laika momentā t aplūko j–to komandu K_j (ilustrāciju skatīt 1. zīmējumā).

a) Ja $K_j = Z(i)$, tad 2. galviņa pievēršas šūnai R_i un skaitli r_i aizstāj ar skaitli 0. Mēs šai situācijā teiksim, ka RAM i–tajā šūnā ieraksta 0. Pēc šī darba veikšanas, ja $j < \tau$, 1. galviņa laika momentā $t + 1$ aplūkos $j + 1$–mo komandu K_{j+1}. Pretējā gadījumā, t.i., ja $j = \tau$, RAM turpmākajos laika momentos nekādas darbības neveiks. Mēs šai situācijā teiksim, ka RAM beidz darbu (apstājas).

b) Ja $K_j = S(i)$, tad 2. galviņa pievēršas šūnai R_i un skaitli r_i aizstāj ar skaitli $r_i + 1$. Mēs šai situācijā teiksim, ka RAM i–tās šūnas saturu palielina par skaitli 1. Pēc šī darba veikšanas, ja $j < \tau$, 1. galviņa laika momentā $t+1$ aplūkos $j + 1$–mo komandu K_{j+1}. Pretējā gadījumā RAM beidz darbu.

c) Ja $K_j = T(m, i)$, tad 2. galviņa pievēršas šūnai R_i un skaitli r_i aizstāj ar skaitli r_m. Mēs šai situācijā teiksim, ka RAM m–tās šūnas saturu pārsūta uz i–to šūnu. Pēc šī darba veikšanas, ja $j < \tau$, 1. galviņa laika momentā $t + 1$ aplūkos $j + 1$–mo komandu K_{j+1}. Pretējā gadījumā RAM beidz darbu.

d) Ja $K_j = J(m, n, q)$, tad RAM salīdzina šūnas R_m saturu ar šūnas R_n saturu.

- Ja $r_m = r_n$ un $q \leq \tau$, tad 1. galviņa laika momentā $t+1$ aplūkos q–to komandu K_q.
- Ja $r_m = r_n$ un $q > \tau$, tad RAM beidz darbu.
- Ja $r_m \neq r_n$ un $j < \tau$, tad 1. galviņa laika momentā $t+1$ aplūkos $j+1$–mo komandu K_{j+1}.
- Ja $r_m \neq r_n$ un $j = \tau$, tad RAM beidz darbu.

— Kam tad īsti domāta RAM?
Pirmajā tuvinājumā atbilde varētu būt šāda:
— Datu apstrādei.
Ja reiz tā, tad rodas jautājums:
— Kā lietojama RAM?
Lietotājs, teiksim Alise, vispirms uzraksta RAM programmu

$$K_1, K_2, \ldots, K_\tau.$$

Šo programmu ievieto mašīnā. Konkrētas RAM realizācijas gadījumā konstruktoram saprotams jāparedz, kā šī programma tiks ievadīta mašīnā, taču mūs neinteresē konkrēta RAM realizācija, tāpēc mēs uzskatīsim, ka Alises vienīgais pienākums ir korekti uzrakstīt RAM programmu, piemēram, uz papīra.

Nākošias solis: Alise ievada RAM atmiņas visās šūnās R_i sākuma datus, t.i., skaitļus r_i. Šo skaitļu virkni

$$r_1, r_2, r_3, \ldots, r_n, \ldots$$

turpmāk sauksim par *sākuma konfigurāciju.*

Visbeidzot konstruktori ir paredzējuši, ka nospiežot starta pogu RAM sāk darbu, t.i., sākas diskrētu laika momentu atskaite. Laika momentā $t = 0$ RAM 1. galviņa aplūko komandu K_1. Tālāko RAM darbību mēs jau esam aprakstījuši iepriekš (skatīt tekstu sākot ar teikumu "Pieņemsim, ka 1. galviņa laika momentā t aplūko j–to komandu K_j").

Ja kādā laika momentā RAM beidz darbu, un tās konfigurācija pēc darba beigšanas ir (ϱ_n), tad teiksim, ka (ϱ_n) ir *beigu konfigurācija* un RAM sākuma konfigurāciju (r_n) pārstrādā par beigu konfigurāciju (ϱ_n). Pretējā gadījumā, t.i., ja RAM strādās bezgalīgi ilgi, teiksim:

— RAM *diverģē* sākuma konfigurācijai (r_n).

Atzīmēsim, ka RAM ir idealizēta mašīna, un realitātē neko tādu nav iespējams uzkonstruēt.

1) Mēs paredzam, ka RAM ir bezgalīga atmiņa. Nekas tāds realitātē nav realizējams.

2) Mēs paredzam, ka Alise drīkst rakstīt patvaļīgi garas programmas, piemēram, komandu skaits τ drīkst būt arī skaitlis $10^{10^{10}}$. Nekas tāds realitātē nav iedomājams, jo pārredzamajā visumā atomu skaits ir mazāks par $10^{10^{10}}$.

3) Mēs uzskatam, ka mūsu rīcībā ir patvaļīgi laika resursi, proti, RAM drīkst strādāt bezgala ilgi.

— Kāpēc mēs aplūkojam šādu nereālistisku modeli?

Atbilde varētu būt šāda:

— *Matemātiskā idealizācija* nozīmē, ka kaut kāda nematemātiska jēdziena matematizācijas gaitā tam piešķir dažas īpašības, kuras šim jēdzienam tā pirmveidā nav piemitušas. Šīs jaunās īpašības matemātisku teoriju ļauj izveidot pārskatāmāku un nav pretrunā ar realitāti.

Tipisks piemērs ir ģeometrija. Taisne ir neierobežota, kaut arī neviens cilvēks realitātē neko tādu nav redzējis. Jēdziens par neierobežota garuma taisni ir ļāvis izveidot mums pazīstamo ģeometriju, turpretī rīkošanās tikai ar nogriežņiem būtu ievērojami sarežģījusi ģeometrijas loģisko struktūru.

Līdzīgā veidā algoritmu teorijā būtu neparocīgi, ja mēs atmiņas lielumu ierobežotu ar kādu iepriekš fiksētu skaitli $\varkappa$. Saprotams, katram reālam datoram šāds ierobežojums eksistē, taču datoru jaudas un atmiņa laika gaitā mainās. Mēs ar RAM modeli cenšamies aptvert būtisko pēc iespējas plašākos mērogos.

Turpmākajam fiksēsim šādu terminoloģiju.

- Komandu $Z(n)$ mēs sauksim par *nulles piešķiršanu.* Ja gribēsim precizēt, tad teiksim:

  — Nulles piešķiršana n–tajai šūnai.

- Komandu $S(n)$ mēs sauksim par *vieninieka pieskaitīšanu.* Ja gribēsim precizēt, tad teiksim:

  — Vieninieka pieskaitīšana n–tajai šūnai.

- Komandu $T(m, n)$ sauksim par *pāradresāciju.* Ja gribēsim precizēt, tad teiksim:

  — m–tās šūnas satura pārsūtīšana uz n–to šūnu.

- Komandu $J(m,n,q)$ sauksim par *nosacītās vadības maiņu* jeb nosacītās vadības maiņas komandu. Ja gribēsim precizēt, tad teiksim:

  — Nosacītās vadības maiņa ar m–to un n–to šūnu pārejot uz q–to komandu.

  Speciālā gadījumā , ja $m = n$, nosacītās vadības maiņu $J(m,m,q)$ sauksim par *beznosacītās vadības maiņu* jeb beznosacītās vadības maiņas komandu. Ja gribēsim precizēt, tad teiksim:

  — Beznosacītās vadības maiņa uz q–to komandu.

Ievērojam, katra komanda

$$Z(n),\ S(n),\ T(m,n)$$

izmaina tikai vienas šūnas saturu. Savukārt nosacītās (beznosacītās) vadības nemaina šūnu saturu.

Turpmāk RAM mēs identificēsim ar tās vienu būtisku sastāvdaļu, proti, programmu. Tā, piemēram, tai vietā, lai teiktu, ka mūsu rīcībā ir RAM, kuras programma ir

1. $Z(1)$
2. $S(1)$
3. $J(3,6,5)$
4. $T(4,1)$

mēs teiksim:

— Pieņemsim, ka dota RAM

1. $Z(1)$
2. $S(1)$
3. $J(3,6,5)$
4. $T(4,1)$

Šādi mēs reprezentējam RAM ar programmu K_1, K_2, K_3, K_4, kur

$$K_1 = Z(1),\ K_2 = S(1),\ K_3 = J(3,6,5),\ K_4 = T(4,1).$$

1.1. Piemēri.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
a	2	3	1	6	23	0	0	...
b	2	0	1	6	23	0	0	...
c	2	4	1	6	23	0	0	...
d	2	3	1	3	23	0	0	...

(i) RAM

$$1.\ Z(2)$$

sākuma konfigurāciju

$$r_1 = 2,\ r_2 = 3,\ r_3 = 1,\ r_4 = 6,\ r_7 = 23 \quad \text{un} \quad \forall i > 7\ (r_i = 0) \tag{1}$$

pārstrādā par beigu konfigurāciju

$$r_1 = 2,\ r_2 = 0,\ r_3 = 1,\ r_4 = 6,\ r_7 = 23 \quad \text{un} \quad \forall i > 7\ (r_i = 0)\,.$$

Tabulā sākuma konfigurācija uzrādīta rindiņā ar pazīmi a, beigu konfigurācija — ar pazīmi b. Tātad izmaiņas ir notikušas tikai 2. šunā. Te skaitlis 3 aizstāts ar skaitli 0.

(ii) RAM

$$1.\ S(2)$$

sākuma konfigurāciju (1) pārstrādā par beigu konfigurāciju, kas tabulā uzrādīta rindiņā ar pazīmi c. Izmaiņas notikušas tikai 2. šūnā. Te skaitlim 3 pieskaitīts vieninieks.

(iii) RAM

$$1.\ T(2,4)$$

sākuma konfigurāciju (1) pārstrādā par beigu konfigurāciju, kas tabulā uzrādīta rindiņā ar pazīmi d. Izmaiņas notikušas tikai 4. šūnā. Te skaitlis 6 aizstāts ar skaitli, kas bija 2. šūnā, proti, skaitlis 6 aizstāts ar skaitli 3.

(iv) RAM

$$1.\ J(1,1,1)$$

diverğē sākuma konfigurācijai (1). Īstenībā šī RAM diverğē jebkurai sākuma konfigurācijai kaut arī izmaiņas nenotiek nevienā šūnā.

(v) RAM

1. $J(1,2,6)$
2. $S(2)$
3. $S(3)$
4. $J(1,2,6)$
5. $J(1,1,2)$
6. $T(3,1)$

sākuma konfigurāciju

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
5	3	1	6	23	0	0	...

pārstrādā par beigu konfigurāciju

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
3	5	3	6	23	0	0	...

Nākošā tabula demonstrē, kā šāda pārstrāde notiek.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	...	
1. $J(1,2,6)$	5	3	1	6	23	0	0	...	Sākuma konfigurācija
2. $S(2)$	5	3	1	6	23	0	0	...	
3. $S(3)$	5	4	1	6	23	0	0	...	
4. $J(1,2,6)$	5	4	2	6	23	0	0	...	
5. $J(1,1,2)$	5	4	2	6	23	0	0	...	
2. $S(2)$	5	4	2	6	23	0	0	...	
3. $S(3)$	5	5	2	6	23	0	0	...	
4. $J(1,2,6)$	5	5	3	6	23	0	0	...	
6. $T(3,1)$	5	5	3	6	23	0	0	...	
	3	5	3	6	23	0	0	...	Beigu konfigurācija

RAM darbu sāk ar sākuma konfigurāciju un pirmo komandu (1. galviņa aplūko pirmo komandu). Konkrētajā piemērā RAM darbība ir šāda.

- Laika moments $t = 0$.
  Laika momentā $t = 0$ RAM konfigurācija ir

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
5	3	1	6	23	0	0	...

  un 1. galviņa aplūko komandu $J(1, 2, 6)$.

- Laika moments $t = 1$.
  Komanda $J(1, 2, 6)$ nemaina šūnu saturu, tāpēc laika momentā $t = 1$ RAM konfigurācija ir tāda pati kā iepriekšējā laika momentā

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
5	3	1	6	23	0	0	...

  Tā kā laika momentā $t = 0$ 1. galviņa aplūko komandu $J(1, 2, 6)$ un tā ir 1. komanda, tad laika momentā $t = 1$ 1. galviņa aplūko 2. komandu, jo $r_1 = 5 \neq 3 = r_2$. Konkrētajā gadījumā 2. komanda ir $S(2)$.

- Laika moments $t = 2$.
  Komanda $S(2)$ pieskaita 2. šūnas saturam skaitli 1, tāpēc laika momentā $t = 2$ RAM konfigurācija ir

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
5	4	1	6	23	0	0	...

  Tā kā laika momentā $t = 1$ 1. galviņa aplūko komandu $S(2)$ un tā ir 2. komanda, tad laika momentā $t = 2$ 1. galviņa aplūko 3. komandu. Konkrētajā gadījumā 3. komanda ir $S(3)$.

- Laika moments $t = 3$.
  Komanda $S(3)$ pieskaita 3. šūnas saturam skaitli 1, tāpēc laika momentā $t = 3$ RAM konfigurācija ir

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
5	4	2	6	23	0	0	...

Tā kā laika momentā $t = 2$ 1. galviņa aplūko komandu $S(3)$ un tā ir 3. komanda, tad laika momentā $t = 3$ 1. galviņa aplūko 4. komandu. Konkrētajā gadījumā 4. komanda ir $J(1, 2, 6)$.

- Laika moments $t = 4$.
  Komanda $J(1, 2, 6)$ nemaina šūnu saturu, tāpēc laika momentā $t = 4$ RAM konfigurācija ir tāda pati kā iepriekšējā laika momentā

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
5	4	2	6	23	0	0	...

Tā kā laika momentā $t = 3$ 1. galviņa aplūko komandu $J(1, 2, 6)$ un tā ir 4. komanda, tad laika momentā $t = 4$ 1. galviņa aplūko 5. komandu, jo $r_1 = 5 \neq 4 = r_2$. Konkrētajā gadījumā 5. komanda ir $J(1, 1, 2)$.

- Laika moments $t = 5$.
  Komanda $J(1, 1, 2)$ nemaina šūnu saturu, tāpēc laika momentā $t = 5$ RAM konfigurācija ir tāda pati kā iepriekšējā laika momentā

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
5	4	2	6	23	0	0	...

Tā kā laika momentā $t = 4$ 1. galviņa aplūko komandu $J(1, 1, 2)$ un tā ir 5. komanda, tad laika momentā $t = 5$ 1. galviņa aplūko 2. komandu, jo $r_1 = 5 = r_1$. Konkrētajā gadījumā 2. komanda ir $S(2)$.

- Laika moments $t = 6$.
  Komanda $S(2)$ pieskaita 2. šūnas saturam skaitli 1, tāpēc laika momentā $t = 6$ RAM konfigurācija ir

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
5	5	2	6	23	0	0	...

Tā kā laika momentā $t = 5$ 1. galviņa aplūko komandu $S(2)$ un tā ir 2. komanda, tad laika momentā $t = 6$ 1. galviņa aplūko 3. komandu. Konkrētajā gadījumā 3. komanda ir $S(3)$.

- Laika moments $t = 7$.
  Komanda $S(3)$ pieskaita 3. šūnas saturam skaitli 1, tāpēc laika momentā $t = 7$ RAM konfigurācija ir

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
5	5	3	6	23	0	0	...

  Tā kā laika momentā $t = 6$ 1. galviņa aplūko komandu $S(3)$ un tā ir 3. komanda, tad laika momentā $t = 7$ 1. galviņa aplūko 4. komandu. Konkrētajā gadījumā 4. komanda ir $J(1, 2, 6)$.

- Laika moments $t = 8$.
  Komanda $J(1, 2, 6)$ nemaina šūnu saturu, tāpēc laika momentā $t = 8$ RAM konfigurācija ir tāda pati kā iepriekšējā laika momentā

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
5	5	3	6	23	0	0	...

  Tā kā laika momentā $t = 7$ 1. galviņa aplūko komandu $J(1, 2, 6)$ un $r_1 = 5 = r_2$, tad laika momentā $t = 8$ 1. galviņa aplūko 6. komandu. Konkrētajā gadījumā 6. komanda ir $T(3, 1)$.

- Laika moments $t = 9$.
  Komanda $T(3, 1)$ 1. šūnas saturam piešķir 3. šūnas saturu, tāpēc laika momentā $t = 9$ RAM konfigurācija ir

R_1	R_2	R_3	R_4	R_5	R_6	R_7	...
3	5	3	6	23	0	0	...

  Tā kā laika momentā $t = 8$ 1. galviņa aplūko komandu $T(3, 1)$ un tā ir 6. komanda, tad laika momentā $t = 9$ 1. galviņa aplūko 7. komandu. Konkrētajā gadījumā 7. komandas nav, tāpēc RAM beidz darbu.

1.2. Algebriskas sistēmas

Attīstītas teorijas pamatiezīme ir "spēles noteikumu" fiksēšana. Tāpēc rodas uzdevums veidot teoriju ar vislielāko rūpību un loģisko precizitāti.

Katrā definīcijā jaunais jēdziens tiek konstruēts ar citu jēdzienu palīdzību. Tā rezultātā katra definīcija saistās ar citām, kuras definē tos jēdzienus, kas apskatāmajā definīcijā tiek uzskatīti par zināmiem. Piemēram, par taisnes nogriezni sauc taisnes daļu, kas atrodas starp diviem punktiem. Bet kā definēt jēdzienus "taisne" un "starp"?

Tie jēdzieni, kurus izmanto kaut kādā definīcijā, arī paši ir jādefinē ar kādu agrāku jēdzienu palīdzību, savukārt agrākie jēdzieni arī jādefinē, utt. Kurā vietā šai definīciju ķēdei būs gals (precīzāk — sākums)? Tāda vispār nav. Kāda ir izeja no aprakstītā šķietami bezcerīgā stāvokļa? Matemātiķi šo "Gordija mezglu" nav atraisījuši, bet vienkārši pārcirtuši. Proti, kādā vietā definīciju ķēdē dažus jēdzienus izvēlas *bez definīcijas.* Tos sauc par *sākuma jēdzieniem* jeb *pamatjēdzieniem.* Pārējos (definētos) jēdzienus sauc par *atvasinātiem jēdzieniem.*

Matemātikas sistematizācija deviņpadsmitā gadsimta beigu posmā ļāva secināt, ka viens no perspektīvākajiem pamatjēdzieniem matemātikā ir kopas jēdziens. To var izvēlēties par vienīgo pamatjēdzienu visā matemātikā.

Kopu

$$\{\{x\},\{x,y\}\}$$

sauc par elementu $x \in X$ un $y \in Y$ *sakārtotu pāri* un lieto apzīmējumu (x,y) vai $\langle x,y\rangle$. Pāri $((x_1,x_2,\dots,x_{n-1}),x_n)$, kur $\forall i \in \overline{1,n}\,(x_i \in A_i)$ sauc par *n-dimensionālu kortežu* pār kopām $A_1,A_2,\dots,A_n$. Turpmāk n-dimensionāla korteža apzīmēšanai lietosim pierakstu $(x_1,x_2,\dots,x_n)$ vai arī $\langle x_1,x_2,\dots,x_n\rangle$. Par kopu $A_1,A_2,\dots,A_n$ *Dekarta reizinājumu* sauc visu n-dimensionālo kortežu kopu pār kopām $A_1,A_2,\dots,A_n$, t.i.,

$$A_1 \times A_2 \times \ldots \times A_n \leftharpoondown \{(x_1,x_2,\dots,x_n)\,|\,\forall i \in \overline{1,n}\,(x_i \in A_i)\}.$$

Ja $A = A_1 = A_2 = \ldots = A_n$, tad lieto arī pierakstu $A^n \leftharpoondown A_1 \times A_2 \times \ldots \times A_n$. Kopas $A_1 \times A_2 \times \ldots \times A_n$ apakškopu ϱ mēdz saukt arī par *n-vietīgu attieksmi (attiecību, predikātu)*, kas definēta kopā $A_1 \times A_2 \times \ldots \times A_n$. Šai situācijā kopu A_i, $i \in \overline{1,n}$, sauc par attieksmes ϱ *i-to projekciju* un lieto apzīmējumu $A_i = \mathrm{pr}_i\varrho$.

Trijnieku $f = \langle X,Y,F\rangle$, kur $F \subseteq X \times Y$ sauc par *attēlojumu* jeb *funkciju*, ja visiem kopas F elementiem $(x,y),(x,z)$ ir spēkā vienādība $y = z$. Kopu

X sauc par attēlojuma f *starta* jeb *izejas* kopu, Y — par *finiša* jeb *ieejas* kopu, F sauc par *grafiku.* Ja $(x, y) \in F$, tad lieto pierakstu $f(x) = y$ jeb $f : x \mapsto y$. Vispārīgs pieraksts $f : X \mathrel{-\!\circ\!\!\rightarrow} Y$ (lieto arī pierakstu $X \overset{f}{\mathrel{-\!\circ\!\!\rightarrow}} Y$) norāda, ka f ir attēlojums ar starta kopu X un finiša kopu Y.

Kopu

$$\mathrm{Dom}(f) \rightleftharpoons \{x \mid \exists y \in Y\ (f : x \mapsto y)\}$$

sauc par attēlojuma $f : X \mathrel{-\!\circ\!\!\rightarrow} Y$ *definīcijas apgabalu.* Savukārt kopu

$$\mathrm{Ran}(f) \rightleftharpoons \{y \mid \exists x \in X\ (f : x \mapsto y)\}$$

sauc par attēlojuma f *vērtību apgabalu.* Attēlojumu $f : X \mathrel{-\!\circ\!\!\rightarrow} Y$ sauc par *visur definētu attēlojumu,* ja $\mathrm{Dom}(f) = X$. Šai gadījumā mēdz lietot vienu no apzīmējumiem

$$f : X \to Y \quad \text{vai} \quad X \xrightarrow{f} Y.$$

Pretējā gadījumā attēlojumu $f : X \mathrel{-\!\circ\!\!\rightarrow} Y$ sauc par *daļēji definētu,* proti, ja

$$\exists x \in X\ x \notin \mathrm{Dom}(f).$$

Visur definētu attēlojumu $g : X_1 \times X_2 \times \ldots \times X_n \to X_{n+1}$ sauc arī par *n-vietīgu algebrisku operāciju.* Šai situācijā kopu X_i, $i \in \overline{1, n+1}$, sauc par operācijas g *i-to projekciju* un lieto apzīmējumu $X_i = \mathrm{pr}_i g$.

Attēlojumu $f : X \mathrel{-\!\circ\!\!\rightarrow} Y$ sauc par *sirjekciju* un lieto apzīmējumu

$$f : X \twoheadrightarrow Y,$$

ja $\mathrm{Ran}(f) = Y$. Attēlojumu f sauc par *injekciju* un lieto apzīmējumu

$$f : X \hookrightarrow Y,$$

ja dažādiem elementiem x_1, x_2 atbilst dažādi y_1, y_2, t.i.,

$$\forall (x_1, x_2) \in X^2\ [x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)].$$

Ja algebriska operācija $h : X \to Y$ ir gan sirjekcija, gan injekcija, tad to sauc par *bijekciju.*

Trijnieku $\langle K, O, A\rangle$ sauc par *n-šķiru algebrisku sistēmu*, ja

(i) $K = \{K_1, K_2, \ldots, K_n\}$, kur K_i, $i \in \overline{1, n}$, ir dažādas netukšas kopas,

(ii) O ir algebrisku operāciju $\circ_i : X_1 \times X_2 \times \ldots \times X_{k(i)} \to Y$ kopa, kur $\forall j \in \overline{1, k(i)}\, (X_j \in K)$, kā arī $Y \in K$,

(iii) A ir dažādu attieksmju $\varrho_i \subseteq X_1 \times X_2 \times \ldots \times X_{m(i)}$ kopa, kur $\forall j \in \overline{1, m(i)}\, (X_j \in K)$,

(iv) $\forall i \in \overline{1, n}\, [\, \exists \circ \in O\, \exists j\, (\, K_i = \mathrm{pr}_j \circ) \vee \exists \varrho \in A\, \exists j\, (K_i = \mathrm{pr}_j \varrho)]$.

Ja kopas O un A ir galīgas, un nerodas pārpratumi, piemēram,

$$O = \{\circ_1, \circ_2, \ldots, \circ_k\}, \qquad A = \{\varrho_1, \varrho_2, \ldots, \varrho_m\},$$

tad $\langle K, O, A\rangle$ vietā lieto pierakstu

$$\langle K_1, K_2, \ldots, K_n; \circ_1, \circ_2, \ldots, \circ_k; \varrho_1, \varrho_2, \ldots, \varrho_m\rangle.$$

Ja $O = \emptyset$, tad algebrisko sistēmu sauc par *modeli*, ja turpretī $A = \emptyset$, tad — par *algebru*. Šai situācijā $\langle K, O, A\rangle$ vietā attiecīgi lieto pierakstu $\langle K, A\rangle$ vai $\langle K, O\rangle$, vai arī attiecīgi

$$\langle K_1, K_2, \ldots, K_n; \varrho_1, \varrho_2, \ldots, \varrho_m\rangle \quad \text{vai} \quad \langle K_1, K_2, \ldots, K_n; \circ_1, \circ_2, \ldots, \circ_k\rangle.$$

1.3. RAM definīcija

1.2. Definīcija. *Visur definētu funkciju*

$$r : \mathbb{Z}_+ \to \mathbb{N}$$

sauc par konfigurāciju.

Konfigurācijas apzīmēšanai parasti parasti lietosim apzīmējumu

$$r_1, r_2, \ldots, r_k, \ldots$$

vai arī lietosim pierakstu (r_k). Visu konfigurāciju kopas apzīmēšanai lietosim pierakstu K°.

Kā redzams no matemātiskās analīzes viedokļa konfigurācija (r_k) nav nekas cits kā virkne, kuras elementi ir naturāli skaitļi. Konfigurācijas i–tā elemeta vērtību r_i turpmāk mēs sauksim arī par i–tās šūnas saturu. Lietojot šo terminoloģiju mēs ņemam vērā iepriekš aprakstīto RAM fizikālo interpretāciju.

1.3. Definīcija. *Ja $n \in \mathbb{Z}_+$, tad visur definētu attēlojumu*

$$Z_i(n) : K^\circ \to \mathbb{N} \times K^\circ : r_k \mapsto (i+1, (r'_k)),$$

kur

$$r'_k \leftharpoondown \begin{cases} r_k, & \text{ja} \quad k \neq n; \\ 0, & \text{ja} \quad k = n \end{cases}$$

sauc par nulles piešķiršanu n–tajai šūnai.

1.4. Piemērs. *Ja $(r_k) = (2^k)$, tad*

$$Z_5(3)[(2^k)] = (6; 2,\ 4,\ 0,\ 16,\ 32, \ldots, 2^k, \ldots)$$

1.5. Definīcija. *Ja $n \in \mathbb{Z}_+$, tad visur definētu attēlojumu*

$$S_i(n) : K^\circ \to \mathbb{N} \times K^\circ : r_k \mapsto (i+1, (r'_k)),$$

kur

$$r'_k \leftharpoondown \begin{cases} r_k, & \text{ja} \quad k \neq n; \\ r_n + 1, & \text{ja} \quad k = n \end{cases}$$

sauc par vieninieka pieskaitīšanu n–tajai šūnai.

1.6. Piemērs. *Ja $(r_k) = (2^k)$, tad*

$$S_{17}(3)[(2^k)] = (18; 2,\ 4,\ 9,\ 16,\ 32, \ldots, 2^k, \ldots)$$

1.7. Definīcija. *Ja $(m, n) \in \mathbb{Z}_+^2$, tad visur definētu attēlojumu*

$$T_i(m, n) : K^\circ \to \mathbb{N} \times K^\circ : r_k \mapsto (i+1, (r'_k)),$$

kur

$$r'_k \leftharpoondown \begin{cases} r_k, & \text{ja} \quad k \neq n; \\ r_m, & \text{ja} \quad k = n \end{cases}$$

sauc par m–tās šūnas satura pārsūtīšanu uz n–to šūnu.

1.8. Piemērs. *Ja $(r_k) = (2^k)$, tad*

$$T_{84}(3, 1)[(2^k)] = (85; 8,\ 4,\ 8,\ 16,\ 32, \ldots, 2^k, \ldots)$$

1.9. Definīcija. *Ja* $(m,n,q) \in \mathbb{Z}_+^3$*, tad visur definētu attēlojumu*

$$J_i(m,n,q) : K^\circ \to \mathbb{N} \times K^\circ : r_k \mapsto (j,(r_k)),$$

kur

$$j \leftharpoondown \begin{cases} i+1, & \text{ja} \quad r_m \neq r_n; \\ q, & \text{ja} \quad r_m = r_n \end{cases}$$

sauc par nosacītās vadības maiņu ar m*–to un* n*–to šūnu pārejot uz* q*–to komandu.*

1.10. Piemērs. *Ja* $(r_k) = (2^k)$*, tad*

$$J_7(1,3,1)[(2^k)] = (8,(2^k))\,.$$

1.11. Definīcija. *Ja* $(i,m,n,q) \in \mathbb{Z}_+^4$*, tad attēlojumus*

$$Z_i(n),\ S_i(n),\ T_i(m,n),\ J_i(m,n,q)$$

sauc par komandām.

Turpmākajam fiksēsim šādus apzīmējumus:

$$\begin{array}{ll}
Z^\circ \leftharpoondown \{Z_i(n) \mid (i,n) \in \mathbb{Z}_+^2\}, & Z_i^\circ \leftharpoondown \{Z_i(n) \mid n \in \mathbb{Z}_+\}; \\
S^\circ \leftharpoondown \{S_i(n) \mid (i,n) \in \mathbb{Z}_+^2\}, & S_i^\circ \leftharpoondown \{S_i(n) \mid n \in \mathbb{Z}_+\}; \\
T^\circ \leftharpoondown \{T_i(m,n) \mid (i,m,n) \in \mathbb{Z}_+^3\}, & T_i^\circ \leftharpoondown \{T_i(m,n) \mid (m,n) \in \mathbb{Z}_+^2\}; \\
J^\circ \leftharpoondown \{J_i(m,n,q) \mid (i,m,n,q) \in \mathbb{Z}_+^4\}, & J_i^\circ \leftharpoondown \{J_i(m,n,q) \mid (m,n,q) \in \mathbb{Z}_+^3\}
\end{array}$$

1.12. Definīcija. *Modeli*

$$\langle L, \preccurlyeq \rangle$$

sauc par lineāri sakārtotu kopu, ja

- $\preccurlyeq$ *ir kopā* L *definēta divvietīga attiecība;*
- $x \preccurlyeq x$ *— refleksivitāte;*
- $x \preccurlyeq y\ \wedge y \preccurlyeq z \Rightarrow x \preccurlyeq z$ *— transitivitāte;*
- $x \preccurlyeq y\ \wedge y \preccurlyeq x \Rightarrow x = y$ *— antisimetriskums;*
- $x \preccurlyeq y\ \vee y \preccurlyeq x$ *— salīdzināmība.*

Kā tas ierasts algebrā, ja nerodas pārpratumi, tad tai vietā, lai teiktu, ka modelis $\langle L, \preccurlyeq \rangle$ ir lineāri sakārtota kopa, saka:

— Kopa L ir lineāri sakārtota.

Ja kopa L ir galīga, tad lineārā sakārtojuma definīcija reducējas uz 1. elementa definīciju, 2. elementa definīciju, utt., līdz nodefinēts n-tais elements. Te n ir kopas L apjoms $|L|$.

1.13. Definīcija. *Galīgu lineāri sakārtotu kopu $K_1 \preccurlyeq K_2 \preccurlyeq \ldots \preccurlyeq K_\tau$ sauc par programmu, ja $\forall i \; K_i \in Z_i^\circ \cup S_i^\circ \cup T_i^\circ \cup J_i^\circ$.*

Turpmāk programmas i-to elementu sauksim par programmas i–to komandu un lietosim pierakstu K_i. Visu iespējamo programmu kopas apzīmēšanai lietosim pieraksru P°.

1.14. Definīcija. *Attēlojumu*

$$\mathrm{nxt} : \mathbb{N} \times K^\circ \times P^\circ \to \mathbb{N}$$

sauc par nākošo komandu jeb nākošo izpildāmo komandu, ja tās definīcija ir šāda:

$$\mathrm{nxt} : (i, (r_k), P) \;\mapsto \begin{cases} i+1, & \text{ja} \quad \exists n \; K_i = Z_i(n); \\ i+1, & \text{ja} \quad \exists n \; K_i = S_i(n); \\ i+1, & \text{ja} \quad \exists mn \; K_i = T_i(m,n); \\ i+1, & \text{ja} \quad \exists mnq \; K_i = J_i(m,n,q) \quad \text{un} \quad r_m \neq r_n; \\ q, & \text{ja} \quad \exists mnq \; K_i = J_i(m,n,q) \quad \text{un} \quad r_m = r_n; \\ 0, & \text{pārējos gadījumos.} \end{cases}$$

1.15. Definīcija. *Attēlojumu*

$$\mathrm{conf} : \mathbb{N} \times K^\circ \times P^\circ \to K^\circ$$

sauc par nākošo konfigurāciju, ja tās definīcija ir šāda:

$$\mathrm{conf} : (i, (r_k), P) \;\mapsto \begin{cases} K_i[(r_k)], & \text{ja} \quad \text{programma } P \text{ satur } i\text{-to komandu;} \\ (r_k), & \text{pārējos gadījumos.} \end{cases}$$

1.16. Definīcija. *Algebru*

$$\langle \mathbb{N}, K^\circ, P^\circ; P, \mathrm{nxt}, \mathrm{conf} \rangle$$

sauc par racionālu adresu mašīnu (RAM), ja

- $\mathbb{N}$ — *naturālo skaitļu kopa;*
- K° — *konfigurāciju kopa;*
- P° — *programmu kopa;*
- $P \in P^\circ$*;*
- nxt — *nākošā komanda;*
- conf — *nākošā konfigurācija.*

Kā redzams RAM ir ļoti specifiska algebra, proti, ne katra algebra

$$\langle A, B, C; c, \circ_1, \circ_2 \rangle$$

ir RAM, pat ja

- A, B, C — kopas;
- $c \in C$;
- $A \times B \times C \stackrel{\circ_1}{\to} A$ un $A \times B \times C \stackrel{\circ_2}{\to} B$.

Terminoloģija. Latviskojums RAM saistīts ar šādiem apsvērumiem.

- Termins "racionāla adresu mašīna" asociējas ar tādiem vārdiem, kā: saprātīgs, lietderīgs, ekonomisks — tātad taupīgs līdzekļu izvēlē.
- RAM spēj darboties ar racionāliem skaitļiem.
- Visbeidzot, angliskā termina "random-access machine" saīsinājums ir RAM.

1.17. Definīcija. *Virkni*

$$c_1, n_1, c_2, n_2, \ldots, c_t, n_t, \ldots$$

sauc par rēķināšanu ar programmu P *pie sākuma konfigurācijas* (r_k)*, ja*

- $c_1 = \mathrm{conf}(1, (r_k), P)$*,* $n_1 = \mathrm{nxt}(1, (r_k), P)$*;*
- $c_{t+1} = \mathrm{conf}(n_t, c_t, P)$*,* $n_{t+1} = \mathrm{nxt}(n_t, c_t, P)$.

Šai situācijā lieto apzīmējumu $P(r_k)$ un (r_k) sauc par sākuma konfigurāciju. Tātad apzīmējums $P(r_k)$ ir nekas cits kā virknes

$$c_1, n_1, c_2, n_2, \ldots, c_t, n_t, \ldots$$

īsāks pieraksts. Ja $\forall i > n\ r_i = 0$, tad lieto apzīmējumu

$$P(r_1, r_2, \ldots, r_n) \leftharpoondown P(r_k).$$

No definīcijas uzreiz izriet, ka apakšvirkne (c_t) ir konfigurāciju virkne, savukārt apašvirkne n_t ir ierobežota naturālo skaitļu virkne.

1.18. Sekas. *Ja eksistē tāds t, ka $n_t = 0$, tad*

$$\forall i \geq t\ (n_i = 0\ \wedge\ c_i = c_t)\,.$$

□ Tā kā komandas ar numuru 0 nav, tad

$$n_i = \mathrm{nxt}(0, c_{i-1}, P) = 0 \qquad \text{un} \qquad c_i = \mathrm{conf}(0, c_{i-1}, P) = c_{i-1}\,.$$

Tagad induktīvi secināms, ka $c_{i-1} = c_t$. ■

1.19. Definīcija. *Konfigurāciju c_t sauc par beigu konfigurāciju, ja $n_t = 0$.*

Šai situācijā lieto apzīmējumu $P(r_k)\downarrow$ un saka, ka rēķināšana $P(r_k)$ *konverģē*, t.i., rēķināšana ar programmu P pie sākuma konfigurācijas (r_k) konverģē. Ja $\forall i > n\ r_i = 0$, tad lieto apzīmējumu

$$P(r_1, r_2, \ldots, r_n)\downarrow\ \leftharpoondown P(r_k)\downarrow.$$

Ja turpretī $\forall i\ n_i \neq 0$, tad lieto apzīmējumu $P(r_k)\uparrow$ un saka, ka rēķināšana $P(r_k)$ *diverģē*, t.i., rēķināšana ar programmu P pie sākuma konfigurācijas (r_k) diverģē. Ja $\forall i > n\ r_i = 0$, tad lieto apzīmējumu

$$P(r_1, r_2, \ldots, r_n)\uparrow\ \leftharpoondown P(r_k)\uparrow.$$

Pieņemsim, ka $c_t = (c_{tk})$ ir beigu konfigurācija

$$c_{t1}, c_{t2}, \ldots, c_{ti}, \ldots$$

Saka, ka rēķināšana rēķināšana $P(r_k)$ konverģē uz b, ja $b = c_{t1}$, t.i., $P(r_k) \downarrow$ un $b = c_{t1}$. Šai situācijā lieto apzīmējumu $P(r_k) \downarrow b$. Ja $\forall i > n\ r_i = 0$, tad lieto apzīmējumu

$$P(r_1, r_2, \ldots, r_n) \downarrow b \leftharpoondown P(r_k) \downarrow b.$$

Vienošanās. Lai nepārslogotu apzīmējumus, mēs vienkāršosim programmas pierakstu. Tā, piemēram, programmu

$$S_1(4) \preccurlyeq T_2(7,3) \preccurlyeq J_3(2,3,1) \preccurlyeq Z_4(8)$$

mēs pierakstīsim šādi:

1. $S(4)$
2. $T(7,3)$
3. $J(2,3,1)$
4. $Z(8)$

1.4. RAM izrēķināmas funkcijas

1.20. Definīcija. *Saka, ka programma P RAM-rēķina funkciju*

$$f : \mathbb{N}^n \multimap\!\!\rightarrow \mathbb{N},$$

ja

- $P(x_1, x_2, \ldots, x_n) \downarrow \Leftrightarrow (x_1, x_2, \ldots, x_n) \in \mathrm{Dom}(f)$*;*
- $P(x_1, x_2, \ldots, x_n) \downarrow y \ \Leftrightarrow \ f(x_1, x_2, \ldots, x_n) = y$*.*

1.21. Definīcija. *Funkciju $f : \mathbb{N}^n \multimap\!\!\rightarrow \mathbb{N}$ sauc par RAM-izrēķināmu, ja eksistē tāda programma P, kas RAM-rēķina šo funkciju.*

Vienošanās. Parasti RAM-izrēķināmas funkcijas sauc vienkārši par *izrēķināmām funkcijām.*

$$f_P^{(n)}(x_1, x_2, \ldots, x_n) \leftharpoondown \begin{cases} y, & \text{ja } \ P(x_1, x_2, \ldots, x_n) \downarrow y, \\ \text{nav definēta, ja } P(x_1, x_2, \ldots, x_n) \uparrow . \end{cases}$$

Līdz ar to matemātikā precizēts jēdziens "eksistē algoritms, kas rēķina funkciju $f : \mathbb{N}^n \multimap\!\!\rightarrow \mathbb{N}$ ", proti, matemātikā uzskata, ka izrēķināmām un tikai izrēķināmām funkcijām eksistē algoritmi, kas tās rēķina.

1.22. Apgalvojums. *Ja P — programma bez nosacītās vadības maiņas, tad*

$$\exists\mu\,\forall x\,[f_P^1(x) = \mu \ \vee \ f_P^1(x) = x + \mu]\,.$$

□ Pierādījums induktīvs pa programmas P garumu. Mēs pierādīsim: ja (ϱ_i) ir beigu konfigurācija, tad $\varrho_i = m_i \ \vee \varrho_i = x + m_i$, piedevām $m_i \le |P|$.

Indukcijas bāze. Ja programmas P garums $|P| = 1$, tad tā ir viena no prgrammām

$$Z(n),\ S(n),\ T(m,n).$$

Ņemam vērā, ka tiek rēķināta vienargumenta funkcija, tāpēc sākuma konfigurācija ir

$$x, 0, 0, , \ldots, 0, \ldots$$

proti, sākuma konfigurācijas (r_i) elementi ir šādi:

$$r_1 = x \quad \text{un} \quad \forall i > 1 \ \ r_i = 0\,.$$

Visos šajos gadījumos beigu konfigurācijas (ϱ_i) elementi ir šādi:

$$\varrho_i \in \{0, 1, x, x+1\}.$$

Indukcijas solis. Mēs pieņemam: ja programmas P garums $|P| = \tau$, tad beigu konfigurācijas (ϱ_i) elementi ir šādi:

$$\varrho_i \in \{m_i, x + m_i\} \quad \text{un} \quad m_i \le |P|\,.$$

Pieņemsim, ka programmas P' garums $|P'| = \tau + 1$, tad

$$P' = K_1, K_2, \ldots, K_\tau, K_{\tau+1}\,.$$

Te

$$\forall i\ K_i \in Z^\circ \cup S^\circ \cup T^\circ\,,$$

t.i., visas komandas ir izskatā $Z(n), S(n)$ vai $T(m,n)$.

Saskaņā ar indukcijas pieņēmumu, ja rēķināšana notiek ar programmu $P = K_1, K_2, \ldots, K_\tau$, tad beigu konfigurācijas (ϱ_i) elementi ir šādi:

$$\varrho_i \in \{m_i, x + m_i\} \quad \text{un} \quad m_i \le |P|\,.$$

Tā kā $K_{\tau+1}$ ir izskatā $Z(n), S(n)$ vai $T(m,n)$, tad programmas P' beigu konfigurācijas (ϱ'_i) elementi ir iegūti no konfigurācijas (ϱ_i), un tāpēc tie ir šādi:

$$\varrho'_i \in \{0, m_i, m_i + 1, x + m_i, x + m_i + 1\} \cup \{m_j | j \in \mathbb{Z}_+\} \cup \{x + m_j | j \in \mathbb{Z}_+\}.$$

Tas demonstrē, ka

$$\varrho_i' \in \{m_i', x + m_i'\} \quad \text{un} \quad m_i' \leq |P'| \,. \quad \blacksquare$$

1.23. Apgalvojums. *Ja programma P nesatur vieninieka pieskaitīšanu un $x \in \mathrm{Dom}(f_P^1)$, tad*

$$f_P^1(x) \in \{0, x\}$$

$\square$ Ņemam vērā, ka tiek rēķināta vienargumenta funkcija, tāpēc sākuma konfigurācija ir

$$x, 0, 0, , \ldots, 0, \ldots$$

proti, sākuma konfigurācijas (r_i) elementi ir šādi:

$$r_1 = x \quad \text{un} \quad \forall i > 1 \;\; r_i = 0 \,.$$

Saskaņā ar doto programma $P = K_1, K_2, \ldots, K_\tau$ nesatur nevienu komandu izskatā $S(n)$, tāpēc K_i ir viena no komandām

$$Z(n), \; T(m, n), \; J(m, n, q) \,.$$

Nosacītās vadības maiņa $J(m, n, q)$ nemaina konfigurāciju; $Z(n)$ n–tajai šūnai piešķir vērtību 0; savukārt $T(m, n)$ m–tās šūnas saturu pārsūta uz n–to šūnu. Tas nozīmē, ka šo komandu darbības rezultātā katras šūnas saturs ir vai nu nulle, vai arī vienāds ar kādas citas šūnas saturu. Tā kā sākuma konfigurācijā (r_i) katra elementa r_i vērtība ir 0 vai x, tad arī beigu konfigurācijas (ϱ_i) katra vērtība ϱ_i būs 0 vai x. $\blacksquare$

1.24. Definīcija. *Programmas P un P' sauc par ekvivalentām programmām, ja katrai sākuma konfigurācijai (r_k) izpildās nosacījums:*

$$P(r_k) \downarrow \; \Leftrightarrow \; P'(r_k) \downarrow \,,$$

turklāt vēl abām rēķināšanām sakrīt beigu konfigurācijas.

1.25. Apgalvojums. *Katrai programmai P eksitē tai ekvivalenta programma P', kas nesatur pāradresācijas komandas.*

□ Vispirms ievērojam, ka programmas

1.	$T(m,n)$	1.	$Z(n)$
		2.	$J(m,n,5)$
		3.	$S(n)$
		4.	$J(1,1,2)$

ir ekvivalentas.

Pieņemsim, ka $P = K_1, K_2, \ldots, K_\tau$ un $K_\nu = T(m,n)$. Rakstam jaunu programmu

$$P' = K'_1, K'_2, \ldots, K'_\nu, K'_{\nu+1}, K'_{\nu+2}, K'_{\nu+3}, \ldots, K'_{\tau+3}\,.$$

Te

- $K'_\nu \leftharpoondown Z(n)$,
  $K'_{\nu+1} \leftharpoondown J(m,n,\nu+4)$,
  $K'_{\nu+2} \leftharpoondown S(n)$,
  $K'_{\nu+3} \leftharpoondown J(1,1,\nu)$;
- $\forall j \in \overline{1,\nu-1}\ K'_j \leftharpoondown K_j$, ja K_j ir kāda no komandām $Z(n), S(n)$, $T(m,n)$ vai $J(m,n,q)$ ar $q \leq \nu$;
- $\forall j \in \overline{\nu+1,\tau}\ K'_{j+3} \leftharpoondown K_j$, ja K_j ir kāda no komandām $Z(n), S(n)$, $T(m,n)$ vai $J(m,n,q)$ ar $q \leq \nu$;
- $\forall j \in \overline{1,\nu-1}\ K'_j \leftharpoondown J(m,n,q+3)$, ja $K_j = J(m,n,q)$ ar $q > \nu$;
- $\forall j \in \overline{\nu+1,\tau}\ K'_{j+3} \leftharpoondown J(m,n,q+3)$, ja $K_j = J(m,n,q)$ ar $q > \nu$.

Ņemot vērā iepriekš konstatēto programmas P un P' ir ekvivalentas.

Ja programma P' satur vēl kādu komandu izskatā $T(m,n)$, tad rakstam jaunu programmu P'' pēc tā paša parauga, kā programmai P rakstījām programmu P'. Tā galīgu skaitu reižu rakstot jaunas programmas galarezultātā iegūsim programmu, kas būs ekvivalenta sākotnējai programmai P un nesaturēs pāradresācijas komandas. ■

1.26. Vingrinājums. *Katrai RAM izrēķināmai funkcijai*

$$f : \mathbb{N}^n \multimap \mathbb{N}$$

eksistē programma P bez nulles piešķiršanas, kas RAM-rēķina funkciju f.

1.27. Piemēri. *Sekojošās funkcijas ir RAM-izrēķināmas.*

(i) Lai parādītu, ka $f(x) = 5$ ir RAM-izrēķināma, mums jāuzraksta RAM-programmu, kas rēķina šo funkciju.

1. $Z(1)$
2. $S(1)$
3. $S(1)$
4. $S(1)$
5. $S(1)$
6. $S(1)$

(ii) $f(x) = x + 3.$

1. $S(1)$
2. $S(1)$
3. $S(1)$

(iii)

$$f(x) = \begin{cases} 0, & \text{ja} \quad x = 0, \\ 1, & \text{ja} \quad x \neq 0. \end{cases}$$

1. $J(1, 2, 4)$
2. $Z(1)$
3. $S(1)$

(iv) $f(x, y) = x + y.$

1. $J(2, 3, 5)$
2. $S(1)$
3. $S(3)$
4. $J(1, 1, 1)$

Nākošā tabula demonstrē, kā notiek rēķināšana, ja $x = 12$ un $y = 3$.

	R_1	R_2	R_3	R_4	R_5	R_6	...	
1. $J(2,3,5)$	12	3	0	0	0	0	...	Sākuma konfigurācija
2. $S(1)$	12	3	0	0	0	0	...	
3. $S(3)$	13	3	0	0	0	0	...	
4. $J(1,1,1)$	13	3	1	0	0	0	...	
1. $J(2,3,5)$	13	3	1	0	0	0	...	
2. $S(1)$	13	3	1	0	0	0	...	
3. $S(3)$	14	3	1	0	0	0	...	
4. $J(1,1,1)$	14	3	2	0	0	0	...	
1. $J(2,3,5)$	14	3	2	0	0	0	...	
2. $S(1)$	14	3	2	0	0	0	...	
3. $S(3)$	15	3	2	0	0	0	...	
4. $J(1,1,1)$	15	3	3	0	0	0	...	
1. $J(2,3,5)$	15	3	3	0	0	0	...	
	15	3	3	0	0	0	...	Beigu konfigurācija

Shematiski vispārīgā gadījumā tas izskatās šādi.

R_1	R_2	R_3	R_4	R_5	R_6	...	
x	y	0	0	0	0	...	Sākuma konfigurācija
$x+k$	y	k	0	0	0	...	
$x+y$	y	y	0	0	0	...	Beigu konfigurācija

(v)

$$x \dot{-} 1 \leftharpoondown \begin{cases} 0, & \text{ja} \quad x = 0, \\ x-1, & \text{ja} \quad x > 0. \end{cases}$$

1. $J(1,2,8)$
2. $S(2)$
3. $J(1,2,7)$
4. $S(2)$
5. $S(3)$
6. $J(1,1,3)$
7. $T(3,1)$

Nākošā tabula demonstrē, kā notiek rēķināšana, ja $x = 3$.

	R_1	R_2	R_3	R_4	R_5	R_6	...	
1. $J(1,2,8)$	3	0	0	0	0	0	...	Sākuma konfigurācija
2. $S(2)$	3	0	0	0	0	0	...	
3. $J(1,2,7)$	3	1	0	0	0	0	...	
4. $S(2)$	3	1	0	0	0	0	...	
5. $S(3)$	3	2	0	0	0	0	...	
6. $J(1,1,3)$	3	2	1	0	0	0	...	
3. $J(1,2,7)$	3	2	1	0	0	0	...	
4. $S(2)$	3	2	1	0	0	0	...	
5. $S(3)$	3	3	1	0	0	0	...	
6. $J(1,1,3)$	3	3	2	0	0	0	...	
3. $J(1,2,7)$	3	3	2	0	0	0	...	
7. $T(3,1)$	3	3	2	0	0	0	...	
	2	3	2	0	0	0	...	Beigu konfigurācija

Shematiski vispārīgā gadījumā tas izskatās šādi.

R_1	R_2	R_3	R_4	R_5	R_6	...	
x	0	0	0	0	0	...	Sākuma konfigurācija
x	k	$k-1$	0	0	0	...	
$x \dot{-} 1$	x	$x \dot{-} 1$	0	0	0	...	Beigu konfigurācija

(vi)

$$f(x) = \begin{cases} \dfrac{x}{2}, & \text{ja } \ x \text{ ir pārskaitlis,} \\ \text{nav definēta pretējā gadījumā.} \end{cases}$$

1. $J(1,2,6)$
2. $S(2)$
3. $S(2)$
4. $S(3)$
5. $J(1,1,1)$
6. $T(3,1)$

Nākošā tabula demonstrē, kā notiek rēķināšana, ja $x = 4$.

	R_1	R_2	R_3	R_4	R_5	R_6	...	
1. $J(1,2,6)$	4	0	0	0	0	0	...	Sākuma konfigurācija
2. $S(2)$	4	0	0	0	0	0	...	
3. $S(2)$	4	1	0	0	0	0	...	
4. $S(3)$	4	2	0	0	0	0	...	
5. $J(1,1,1)$	4	2	1	0	0	0	...	
1. $J(1,2,6)$	4	2	1	0	0	0	...	
2. $S(2)$	4	2	1	0	0	0	...	
3. $S(2)$	4	3	1	0	0	0	...	
4. $S(3)$	4	4	1	0	0	0	...	
5. $J(1,1,1)$	4	4	2	0	0	0	...	
1. $J(1,2,6)$	4	4	2	0	0	0	...	
6. $T(3,1)$	4	4	2	0	0	0	...	
	2	4	2	0	0	0	...	Beigu konfigurācija

Shematiski, ja x ir pārskaitlis, tas izskatās šādi.

R_1	R_2	R_3	R_4	R_5	R_6	...	
x	0	0	0	0	0	...	Sākuma konfigurācija
x	$2k$	k	0	0	0	...	
$\frac{x}{2}$	x	$\frac{x}{2}$	0	0	0	...	Beigu konfigurācija

(vii) $f(x) = \left\lfloor \dfrac{x}{2} \right\rfloor$.

1. $J(1,2,7)$
2. $S(2)$
3. $J(1,2,7)$
4. $S(2)$
5. $S(3)$
6. $J(1,1,1)$
7. $T(3,1)$

Nākošā tabula demonstrē, kā notiek rēķināšana, ja $x = 3$.

	R_1	R_2	R_3	R_4	R_5	R_6	...	
1. $J(1,2,7)$	3	0	0	0	0	0	...	Sākuma konfigurācija
2. $S(2)$	3	0	0	0	0	0	...	
3. $J(1,2,7)$	3	1	0	0	0	0	...	
4. $S(2)$	3	1	0	0	0	0	...	
5. $S(3)$	3	2	0	0	0	0	...	
6. $J(1,1,1)$	3	2	1	0	0	0	...	
1. $J(1,2,7)$	3	2	1	0	0	0	...	
2. $S(2)$	3	2	1	0	0	0	...	
3. $J(1,2,7)$	3	3	1	0	0	0	...	
7. $T(3,1)$	3	3	1	0	0	0	...	
	1	3	1	0	0	0	...	Beigu konfigurācija

Shematiski vispārīgā gadījumā tas izskatās šādi.

R_1	R_2	R_3	R_4	R_5	R_6	...	
x	0	0	0	0	0	...	Sākuma konfigurācija
x	$2k-1$	$k-1$	0	0	0	...	
x	$2k$	k	0	0	0	...	

Visu laiku notiek pārbaude $2k-1=x$ vai $2k=x$. Atbilstoši 3. šūnā ir vērtība

$$\left\lfloor \frac{2k-1}{2} \right\rfloor = k-1 \qquad \text{vai} \qquad \left\lfloor \frac{2k}{2} \right\rfloor = k$$

līdz beigās 3. šūnā ir izrēķināta vērtība $\lfloor \frac{x}{2} \rfloor$.

(viii)

$$f(x,y) = \begin{cases} 0, & \text{ja} \quad x \leq y, \\ \text{nav definēta pretējā gadījumā.} \end{cases}$$

1. $J(1,2,4)$
2. $S(1)$
3. $J(1,1,1)$
4. $Z(1)$

Nākošā tabula demonstrē, kā notiek rēķināšana, ja $x=12$ un $y=14$.

	R_1	R_2	R_3	R_4	R_5	R_6	...	
1. $J(1,2,4)$	12	14	0	0	0	0	...	Sākuma konfigurācija
2. $S(1)$	12	14	0	0	0	0	...	
3. $J(1,1,1)$	13	14	0	0	0	0	...	
1. $J(1,2,4)$	13	14	0	0	0	0	...	
2. $S(1)$	13	14	0	0	0	0	...	
3. $J(1,1,1)$	14	14	0	0	0	0	...	
1. $J(1,2,4)$	14	14	0	0	0	0	...	
4. $Z(1)$	14	14	0	0	0	0	...	
	0	14	0	0	0	0	...	Beigu konfigurācija

Shematiski vispārīgā gadījumā tas izskatās šādi.

R_1	R_2	R_3	R_4	R_5	R_6	...	
x	y	0	0	0	0	...	Sākuma konfigurācija
$x+k$	y	0	0	0	0	...	

Visu laiku notiek pārbaude vai $x+k=y$? Ja $x \leq y$, tad rēķināšana kādreiz beigsies, pretējā gadījumā rēķināšana diverģē.

(ix)

$$f(x,y) = \begin{cases} 0, & \text{ja} \quad x \leq y, \\ 1, & \text{ja} \quad x > y. \end{cases}$$

1. $T(1,3)$
2. $T(1,5)$
3. $Z(1)$
4. $T(2,4)$
5. $J(2,3,11)$
6. $S(3)$
7. $S(4)$
8. $J(4,5,10)$
9. $J(1,1,5)$
10. $S(1)$

Nākošā tabula demonstrē, kā notiek rēķināšana, ja $x=14$ un $y=12$.

	R_1	R_2	R_3	R_4	R_5	R_6	...	
1. $T(1,3)$	14	12	0	0	0	0	...	Sākuma konfigurācija
2. $T(1,5)$	14	12	14	0	0	0	...	
3. $Z(1)$	14	12	14	0	14	0	...	
4. $T(2,4)$	0	12	14	0	14	0	...	
5. $J(2,3,11)$	0	12	14	12	14	0	...	
6. $S(3)$	0	12	14	12	14	0	...	
7. $S(4)$	0	12	15	12	14	0	...	
8. $J(4,5,10)$	0	12	15	13	14	0	...	
9. $J(1,1,5)$	0	12	15	13	14	0	...	
5. $J(2,3,11)$	0	12	15	13	14	0	...	
6. $S(3)$	0	12	15	13	14	0	...	
7. $S(4)$	0	12	16	13	14	0	...	
8. $J(4,5,10)$	0	12	16	14	14	0	...	
10. $S(1)$	0	12	16	14	14	0	...	
	1	12	16	14	14	0	...	Beigu konfigurācija

Shematiski vispārīgā gadījumā tas izskatās šādi.

R_1	R_2	R_3	R_4	R_5	R_6	...	
x	y	0	0	0	0	...	Sākuma konfigurācija
0	y	x	y	x	0	...	
0	y	$x+k$	$y+k$	x	0	...	

Visu laiku notiek pārbaude $x + k = y$ vai $y + k = x$. Tā rezultātā tiek konstatēts $x \leq y$ vai $x > y$.

1.5. Veseli un racionāli skaitļi

RAM var interpretēt arī kā ierīci, kas rēķina vesela argumenta funkcijas

$$f : \mathbb{Z}^n \multimap\!\!\!\longrightarrow \mathbb{Z}\,.$$

Veselo skaitļu kopu var ieviest balstoties uz naturālo skaitļu kopu $\mathbb{N}$. Shēma ir sekojoša: vispirms kopā $\mathbb{N}^2$ definējam ekvivalences tipa predikātu, proti,

$$(a,b) \sim (x,y) \underset{\text{def}}{\Leftrightarrow} a - b = x - y \vee b - a = y - x.$$

Tā kā kopā $\mathbb{N}$ atņemšana ne vienmēr ir definēta, tad mums jābūt nedaudz uzmanīgiem. Šī iemesla dēļ mēs lietojam nosacījumu

$$a - b = x - y \vee b - a = y - x,$$

nevis nosacījumu $a - b = x - y$. Atzīmēsim, ja $a \geq b$, tad $(a, b) \sim (a - b, 0)$; ja turpretī $b > a$, tad $(a, b) \sim (0, b - a)$.

Ekvivalences tipa predikāts $\sim$ kopā $\mathbb{N}^2$ definē klases

$$\begin{aligned} [-a] &\leftharpoondown \{(x, y) | (x, y) \sim (a, 0)\}, \\ [a] &\leftharpoondown \{(x, y) | (x, y) \sim (0, a)\}. \end{aligned}$$

Tā rezultātā iegūstam faktorkopu

$$\mathbb{N}^2/\!\sim \;\leftharpoondown \{[-a] | a \in \mathbb{N}\} \cup \{[a] | a \in \mathbb{N}\}.$$

Vēl tikai šai faktorkopā jānodefinē saskaitīšana, reizināšana un attiecība mazāks par, un tad jau gredzenu $\mathbb{N}^2/\sim$ ar precizitāti līdz izomorfismam var uzlūkot par veselo skaitļu gredzenu $\mathbb{Z}$.

Šo visu ņemot vērā, mēs katru veselu skaitli RAM interpretēsim kā naturālo skaitļu pāri. Proti, mēs teiksim, ka programma P RAM-rēķina funkciju

$$f : \mathbb{Z}^n \multimap \mathbb{Z}\,,$$

ja

- $P(a_1, a_2, \ldots, a_{2n}) \downarrow \Leftrightarrow\ (x_1, x_2, \ldots, x_n) \in \mathrm{Dom}(f)$;
  te

  $$a_{2i-1} = \begin{cases} -x_i, & \text{ja } x_i < 0, \\ 0, & \text{ja } x_i \geq 0; \end{cases} \qquad a_{2i} = \begin{cases} 0, & \text{ja } x_i < 0, \\ x_i, & \text{ja } x_i \geq 0; \end{cases}$$

- ja $P(a_1, a_2, \ldots, a_{2n}) \downarrow$, (ϱ_i) ir beigu konfigurācija un

  $$f(x_1, x_2, \ldots, x_n) = y\,,$$

  tad

  $$\varrho_1 = \begin{cases} -y & \text{ja } y < 0, \\ 0, & \text{ja } y \geq 0; \end{cases} \qquad \varrho_2 = \begin{cases} 0, & \text{ja } y < 0, \\ y, & \text{ja } y \geq 0; \end{cases}$$

1.28. Piemērs. *Funkcija* $f : \mathbb{Z} \to \mathbb{Z} : x \mapsto x+1$ *ir RAM-izrēķināma.*

1. $J(1,3,9)$
2. $S(3)$
3. $J(1,3,7)$
4. $S(3)$
5. $S(4)$
6. $J(1,1,3)$
7. $T(4,1)$
8. $J(1,1,10)$
9. $S(2)$

Mēs ņēmām vērā, ka katrs vesels skaitlis x reprezentēts kā pāris (a,b). Tā, piemēram,

$$-7 = (7,0), \quad \text{tāpēc} \quad -7+1 = -6 = (6,0)\,.$$

Šo operāciju nodrošina komandas no 2.–8. Tā faktiski ir vieninieka atņemšana (salīdzinājumam skatīt piemēru 1.27(v)).

Savukārt

$$7 = (0,7), \quad \text{tāpēc} \quad 7+1 = 8 = (0,8)\,.$$

Šo operāciju nodrošina viena pati 9. komanda.

Visbeidzot RAM var interpretēt arī kā ierīci, kas rēķina racionāla argumenta funkcijas

$$f : \mathbb{Q}^n \multimap\!\!\!\longrightarrow \mathbb{Q}\,.$$

Racionālo skaitļu kopu var ieviest balstoties uz veselo skaitļu kopu $\mathbb{Z}$. Shēma ir sekojoša: vispirms kopā $\mathbb{Z} \times \mathbb{Z}_+$ definējam ekvivalences tipa predikātu, proti,

$$(a,b) \sim (x,y) \underset{\text{def}}{\Leftrightarrow} ay = bx.$$

Ekvivalences tipa predikāts $\sim$ kopā $\mathbb{Z} \times \mathbb{Z}_+$ definē klases

$$\left[\frac{a}{b}\right] \leftharpoondown \{\,(xy) \mid (x,y) \sim (a,b)\}$$

Tā rezultātā iegūstam faktorkopu

$$\mathbb{Z} \times \mathbb{Z}_+/\!\sim \;\leftharpoondown\; \left\{ \left[\frac{a}{b}\right] \mid a \in \mathbb{Z} \;\wedge\; b \in \mathbb{Z}_+ \right\}.$$

Vēl tikai šai faktorkopā jānodefinē saskaitīšana, rezināšana un attiecība mazāks par, un tad jau lauku $\mathbb{Z} \times \mathbb{Z}_+/\!\sim$ ar precizitāti līdz izomorfismam var uzlūkot par racionālo skaitļu lauku $\mathbb{Q}$.

Šo visu ņemot vērā, mēs katru racionālu skaitli RAM interpretēsim kā naturālo skaitļu trijnieku. Proti, mēs teiksim, ka programma P RAM-rēķina funkciju

$$f : \mathbb{Q}^n \multimap \mathbb{Q}\,,$$

ja

- $P(a_1, a_2, \ldots, a_{3n}) \downarrow \Leftrightarrow\; (x_1, x_2, \ldots, x_n) \in \mathrm{Dom}(f)$;
  te

  $$a_{3i-2} \;=\; \begin{cases} -p_i, & \text{ja} \;\; p_i < 0, \\ 0, & \text{ja} \;\; p_i \geq 0; \end{cases} \qquad a_{3i-1} \;=\; \begin{cases} 0, & \text{ja} \;\; p_i < 0, \\ p_i, & \text{ja} \;\; p_i \geq 0; \end{cases}$$

  $$a_{3i} = q; \qquad x_i = \frac{p_i}{q_i}, \qquad (p_i, q_i) \in \mathbb{Z} \times \mathbb{Z}_+\,.$$

- ja $P(a_1, a_2, \ldots, a_{3n}) \downarrow$, (ϱ_i) ir beigu konfigurācija un

  $$f(x_1, x_2, \ldots, x_n) = y\,,$$

  tad

  $$(\varrho_1, \varrho_2, \varrho_3) \;=\; \begin{cases} (-p, 0, q), & \text{ja} \;\; p < 0, \\ (0, p, q), & \text{ja} \;\; p \geq 0; \end{cases}$$

  te

  $$\frac{p}{q} = y \qquad \text{un} \qquad (p, q) \in \mathbb{Z} \times \mathbb{Z}_+\,.$$

1.29. Piemērs. *Funkcija $f : \mathbb{Q} \to \mathbb{Q} : x \mapsto |x|$ ir RAM-izrēķināma.*

1. $J(1, 4, 4)$
2. $T(1, 2)$
3. $Z(1)$

Mēs ņēmām vērā, ka katrs racionāls skaitlis x reprezentēts kā trijnieks (a, b, c). Tā, piemēram,

$$-\frac{7}{3} = (7, 0, 3), \quad \text{tāpēc} \quad \left|-\frac{7}{3}\right| = \frac{7}{3} = (0, 7, 3)\,.$$

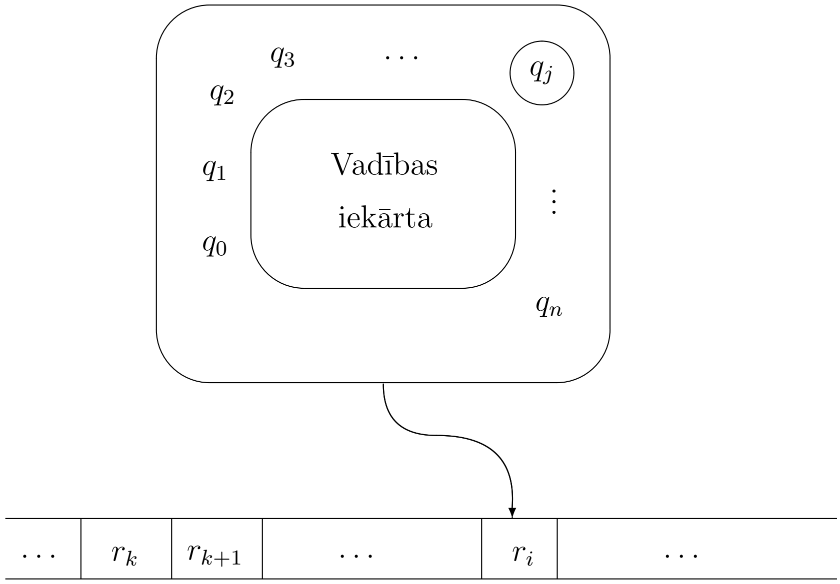
RAM nav paredzēta darbībām ar reāliem skaitļiem, tā vienkāršā iemesla dēļ, ka reālu skaitli vispārīgā gadījumā var reprezentēt tikai kā bezgalīgu naturālo skaitļu virkni. Tas nebūt nenozīmē, ka vispār ne pie kādiem nosacījumiem RAM nav iespējams interpretēt kā ierīci, kas rēķina reāla argumenta funkcijas

$$f : \mathbb{R}^n \multimap \mathbb{R}\,.$$

2. Tjūringa mašīnas

Tjūringa mašīnas, piemēri. Pēc Tjūringa izrēķināmas funkcijas, piemēri.

2.1. Tjūringa mašīnas modelis



2. zīm. Tjūringa mašīnas principiālā darbības shēma.

Līdzīgi kā RAM arī Tjūringa mašīnas būtiska sastāvdaļa ir bezgalīga atmiņas iekārta, kas sadalīta šūnās:

$$\ldots, R_{-1}, R_0, R_1, \ldots, R_i, \ldots$$

Šo atmiņas iekārtu sauc par *lentu.* Tā ir abpusēji bezgalīga. Katrā šūnā R_i ierakstīts patvaļīgs kādas fiksētas galīgas kopas

$$A = \{t, a_1, a_2, \ldots, a_n\}$$

elements. Kopu A sauc par Tjūringa mašīnas *ārējo alfabētu*, kopas A elementus — par *burtiem*.

Otra Tjūringa mašīnas būtiska sastāvdaļa ir tās *iekšējais alfabēts* jeb *stāvokļu kopa* Q. Tāpat kā ārējais alfabēts arī iekšējais alfabēts ir galīgs, proti, kopa Q ir galīga. Tjūringa mašīna vienmēr atrodas kādā no kopas Q stāvokļiem. Citiem vārdiem sakot katrā laika momentā ir aktivizēts kāds no stāvokļiem $q_j \in Q$.

Visbeidzot trešā sastādaļa ir vadības iekārta, kas funkcionē saskaņā ar ievietoto programmu. Programma — tā ir galīga komandu kopa

$$\{K_1, K_2, \ldots, K_s\}\,.$$

Katra komanda K_ν ir simbolu virkne izskatā

$$q_j r_i \mapsto a{\bigstar}q$$

Te

- q_j un q ir stāvokļu kopas Q elementi;
- r_i un a ir alfabēta A burti;
- $\bigstar$ ir viens no kopas $\{\Lsh, \top, \Rsh\}$ elementiem.

Shematiski tas viss attēlots 2. zīmējumā. Attēlā Tjūringa mašīnai aktivizēts iekšējais stāvoklis q_j (tā atrodas stāvoklī q_j). Tai pašā laikā vadības iekārtas galviņa aplūko i–to šūnu, kurā ierakstīts alfabēta A burts r_i.

Tagad aprakstīsim Tjūringa mašīnas vadības iekārtas darbību. Līdzīgi kā RAM arī Tjūringa mašīna darbojas diskrētos laika momentos

$$0, 1, 2, \ldots, \tau, \ldots$$

Pieņemsim, ka laika momentā τ Tjūringa mašīnas galviņa aplūko šūnu ar ierakstu r_i un tā atrodas stāvoklī q_j (ilustrāciju skatīt 2. zīmējumā). Tas nozīmē, ka laika momentā τ Tjūringa mašīna izpilda komandu

$$q_j r_i \mapsto a{\bigstar}q\,,$$

proti,

- Tjūringa mašīna šai laika momentā τ aizstāj ierakstu r_i ar ierakstu a un aktivizē stāvokli q;

- izlemj, kuru šūnu galviņa aplūkos nākošajā laika momentā $\tau + 1$:

a) ja $\bigstar = \Lsh$, tad nākošajā laika momentā galviņa aplūkos $i-1$–mo šūnu, t.i., galviņa pārvietojas vienu šūnu pa kreisi;

b) ja $\bigstar = \top$, tad nākošajā laika momentā galviņa aplūkos to pašu i–to šūnu, t.i., galviņa nepārvietojas;

c) ja $\bigstar = \Rsh$, tad nākošajā laika momentā galviņa aplūkos $i+1$–mo šūnu, t.i., galviņa pārvietojas vienu šūnu pa labi.

Mēs tomēr ne katru galīgu komandu kopu atzīsim par Tjūringa mašīnas programmu (RAM gadījumā mēs par programmu atzinām patvaļīgu galīgu komandu virkni). Pieņemsim, ka

$$Q = \{q_0, q_1, \ldots, q_m\} \qquad \text{un} \qquad Q_0 \rightleftharpoons Q \setminus \{q_0\},$$

tad

- $q_j \neq q_0$, proti, stāvokli q_0 uzskatīsim par beigu stāvokli, t.i., ja Tjūringa mašīna aktivizē stāvokli q_0, tad tas nozīmē, ka tā darbu ir beigusi (apstājusies);

- $q_0 \neq q_1$. Stāvokli q_1 mēs izmantosim kā Tjūringa mašīnas sākuma stāvokli, proti, Tjūringa mašīna vienmēr darbu uzsāk atrodoties stāvoklī q_1. Tātad $|Q| \geq 2$.

- Mēs prasam, lai katram pārim $(q_j, r_i) \in Q_0 \times A$ atbilstu tieši viena programmas komanda
$$q_j r_i \mapsto a\bigstar q\,.$$
Tā rezultātā katra Tjūringa mašīnas programma P sastāv no $m(n+1)$ komandas, t.i., $|P| = m(n+1)$.

Tagad mēs varam atbildēt uz jautājumu:
— Kā lietojama Tjūringa mašīna $\mathfrak{T}$?

Lietotājs, teiksim Alise, vispirms uzraksta Tjūringa mašīnas programmu

$$\{K_1, K_2, \ldots, K_s\}.$$

Šo programmu ievieto mašīnā. Konkrētas Tjūringa mašīnas realizācijas gadījumā konstruktoram saprotams jāparedz, kā šī programma tiks ievadīta mašīnā, taču mūs neinteresē konkrēta Tjūringa mašīnas realizācija, tāpēc mēs uzskatīsim, ka Alises vienīgais pienākums ir korekti uzrakstīt Tjūringa mašīnas programmu, piemēram, uz papīra.

Nākošias solis: Alise ievada Tjūringa mašīnas atmiņas visās šūnās R_i sākuma datus, t.i., burtus $r_i \in A$. Taču te ir viens būtisks ierobežojums, proti, tikai galīgā skaitā šūnu drīkst ierakstīt burtus, kas atšķiras no burta t. Vēl vairāk, mēs prasam, lai tiktu ievērots šāds nosacījums:

$$\text{ja} \qquad r_\varkappa \neq t \neq r_{\varkappa+i}, \qquad \text{tad} \qquad \forall j \in \overline{0,i}\ (r_{\varkappa+j} \neq t).$$

Visbeidzot konstruktori ir paredzējuši, ka nospiežot starta pogu Tjūringa mašīna aktivizē stāvokli q_1, t.i., Tjūringa mašīna sāk darbu; sākas diskrētu laika momentu

$$0, 1, 2, \ldots, \tau, \ldots$$

atskaite. Laika momentā $\tau = 0$ Tjūringa mašīnas galviņa aplūko šūnu, kurā ierakstīts burts

$$r_\varkappa \neq t, \qquad \text{piedevām,} \qquad \forall j < \varkappa\ (r_j = t).$$

Šai situācijā mēs teiksim, ka galviņa atrodas uz ieraksta sākuma. Jāatzīmē viens izņēmums, proti, ja visās šūnās ierakstīts burts t, tad Tjūringa mašīnas galviņa aplūko šūnu R_0. Tālāko Tjūringa mašīnas darbību mēs jau esam aprakstījuši iepriekš (skatīt tekstu sākot ar teikumu "Pieņemsim, ka laika momentā τ Tjūringa mašīnas galviņa aplūko šūnu ar ierakstu r_i un tā atrodas stāvoklī q_j).

Ja kādā laika momentā Tjūringa mašīna beidz darbu un tās galviņa atrodas uz ieraksta sākuma, tad vārdu (t.i, burtu virkni)

$$\varrho_k, \varrho_{k+1}, \ldots, \varrho_{k+\iota}$$

uzskatīsim par Tjūringa mašīnas darba rezultātu. Te

$$\varrho_k \neq t \neq \varrho_{k+\iota}, \qquad \forall j < k\ (\varrho_j = t) \qquad \text{un} \qquad \forall j > k + \iota\ (\varrho_j = t).$$

Atkal jāatzīmē viens izņēmums, proti, ja visās šūnās ierakstīts burts t, tad tukšo vārdu uzskatīsim par Tjūringa mašīnas darba rezultātu.

Skaidrojums. Parasti vārda definīciju vēl papildina ar norunu, ka drīkst lietot arī tā saukto *tukšo vārdu*, kas nesatur nevienu burtu. Šis vārds ir, varētu sacīt, vienkārši tukša vieta. Vienosimies tukšo vietu apzīmēt ar grieķu burtu "lambda": λ.

Ja

$$\forall j < \varkappa\ (r_j = t), \qquad \forall j \in \overline{0,i}\ (r_{\varkappa+j} \neq t) \qquad \text{un} \qquad \forall j > \varkappa + i\ (r_j = t),$$

tad teiksim, ka Tjūringa mašīna $\mathfrak{T}$ vārdu

$$r_\varkappa, r_{\varkappa+1}, \ldots, r_{\varkappa+i}$$

pārstrādā par vārdu

$$\varrho_k, \varrho_{k+1}, \ldots, \varrho_{k+\iota}$$

Ja $\forall j\ (r_j = t)$, tad teiksim, ka Tjūringa mašīna $\mathfrak{T}$ tukšo vārdu pārstrādā par vārdu

$$\varrho_k, \varrho_{k+1}, \ldots, \varrho_{k+\iota}$$

Ja $\forall j\ (\varrho_j = t)$, tad teiksim, ka Tjūringa mašīna $\mathfrak{T}$ vārdu

$$r_\varkappa, r_{\varkappa+1}, \ldots, r_{\varkappa+i}$$

pārstrādā par tukšo vārdu λ. Ja gan $\forall j\ (r_j = t)$, gan $\forall j\ (\varrho_j = t)$, tad teiksim, ka Tjūringa mašīna $\mathfrak{T}$ tukšo vārdu λ pārstrādā par par tukšo vārdu λ. Visos šais gadījumos teiksim, ka rēķināšana ar Tjūringa mašīnu $\mathfrak{T}$ *konverğē*. Pārejos gadījumos teiksim, ka rēķināšana ar Tjūringa mašīnu $\mathfrak{T}$ *diverğē*.

Līdzīgi kā gadījumā ar RAM mēs Tjūringa mašīnu turpmāk identificēsim ar tās vienu būtisku sastāvdaļu, proti, programmu. Tā, piemēram, tai vietā, lai teiktu, ka mūsu rīcībā ir Tjūringa mašīna, kuras programma ir

$$\begin{array}{rcl} q_1\, t & \mapsto & t \top q_0 \\ q_1\, 0 & \mapsto & 0 \top q_0 \\ q_1\, 1 & \mapsto & 1 \top q_0 \end{array}$$

mēs teiksim:

— Pieņemsim, ka dota Tjūringa mašīna

$$\begin{array}{rcl} q_1\, t & \mapsto & t \top q_0 \\ q_1\, 0 & \mapsto & 0 \top q_0 \\ q_1\, 1 & \mapsto & 1 \top q_0 \end{array}$$

Šādi mēs reprezentējam Tjūringa mašīnu ar programmu $\{K_1, K_2, K_3\}$, kur

$$K_1 = q_1\, t \mapsto t \top q_0, \quad K_2 = q_1\, 0 \mapsto 0 \top q_0, \quad K_3 = q_1\, 1 \mapsto 1 \top q_0\,.$$

Dažkārt ērtības labad to visu mēs noformēsim kā tabulu:

	t	0	1
q_1	$t \top q_0$	$0 \top q_0$	$1 \top q_0$

2.2. Vārdi

Mēs jau iepriekšējā paragrāfā redzējām, ka Tjūringa mašīna pārveido vārdus par vārdiem. Tagad mēs noprecizēsim, ko īsti mēs saprotam ar terminu "vārds".

Pieņemsim, ka A — galīga kopa, ko turpmāk sauksim par *alfabētu.* Parasti alfabēta elementi ir *konstruktīvas* dabas objekti. Grāmatu lapās iespiestie burti ir tipisks konstruktīvu objektu piemērs. No kopu teorijas viedokļa var uzskatīt, ka alfabēts un kopa ir sinonīmi (vismaz klasiskās matemātikas ietvaros). Alfabēta elementus, t.i., elementāros simbolus, sauc arī par *burtiem.*

Katru kopas

$$A^+ \leftharpoondown \bigcup_{n=1}^{\infty} A^n$$

elementu $u \in A^+$ sauc par alfabēta A *netukšu vārdu.* Pieņemsim, ka

$$u = (u_1, u_2, \ldots, u_k), \qquad v = (v_1, v_2, \ldots, v_m)$$

ir alfabēta A netukši vārdi, tad

$$u\#v \leftharpoondown (u_1, u_1, \ldots, u_k, v_1, v_2, \ldots, v_m).$$

Šo kopā A^+ definēto operāciju sauc par *konkatenāciju.*

Parasti vārda definīciju vēl papildina ar norunu, ka drīkst lietot arī tā saukto *tukšo vārdu*, kas nesatur nevienu burtu. Šis vārds ir, varētu sacīt, vienkārši tukša vieta. Vienosimies tukšo vietu apzīmēt ar grieķu burtu "lambda": λ.

Saskaņā ar norunu

$$\lambda\#\lambda \leftharpoondown \lambda, \qquad \forall u \in A^+ \;\; \lambda\#u \leftharpoondown u \rightharpoondown u\#\lambda\,.$$

Kopu

$$A^* \leftharpoondown A^+ \cup \{\lambda\}$$

sauc par *alfabēta A vārdu kopu.* Kopas A^* elementus sauc par *vārdiem.* Kā tas tradicionāli pieņemts, ja nerodas pārpratumi, tad konkatenācijas operāciju izlaiž un lieto pierakstu

$$uv \leftharpoondown u\#v,$$

bez tam

$$u_1u_2\ldots u_k \leftharpoondown (u_1, u_2, \ldots, u_k).$$

Ja $a = u_1 = u_2 = \ldots = u_n$, tad lieto arī pierakstu $a^n \leftharpoondown u_1u_2\ldots u_n$. Savukārt $a^0 \leftharpoondown \lambda$.

Pieņemsim, ka $u \in A^n$, tad skaitli n sauc par vārda u garumu, ko turpmāk apzīmēsim ar $|u|$. Saskaņā ar definīciju pieņemsim, ka $|\lambda| \leftharpoondown 0$.

Trijnieku (u, v, u') sauc par vārda v *ieeju* vārdā w, ja $w = uvu'$. Ja $a \in A$ un $w \in A^*$, tad ar $|w|_a$ apzīmēsim burta a dažādo ieeju skaitu vārdā v.

2.1. Piemērs. Trijnieks $(p, asaka, \lambda)$ ir vārda *asaka* ieeja vārdā *pasaka.* Atzīmēsim, ka $|pasaka|_a = 3$, t.i., burts a vārdā *pasaka* ieiet 3 reizes.

2.2. Definīcija. *Vārdu $v \in A^*$ sauc par vārda $w \in A^*$ dalītāju jeb apašvārdu, ja eksistē tādi $u \in A^*$ un $u' \in A^*$, ka $w = uvu'$. Šai situācijā vārdu u sauc par priedēkli, bet u' — par piedēkli. Ja $\lambda \neq u \neq w$, tad u sauc par īstu priedēkli; līdzīgi, ja $\lambda \neq u' \neq w$, tad u' sauc par īstu piedēkli.*

Pieņemsim, ka $A = \{t, a_1, a_2, \ldots, a_n\}$, tad

$$A_t \leftharpoondown A \setminus \{t\} = \{a_1, a_2, \ldots, a_n\}\,.$$

Mēs lietosim pierakstu

$$\mathfrak{T}(u) = w\,,$$

ja Tjūringa mašīna $\mathfrak{T}$ vārdu $u \in A_t^*$ pārstrādā par $w \in A^*$. Tātad tā vietā, lai teiktu (skatīt iepriekšējo paragrāfu), ka Tjūringa mašīna vārdu

$$r_\varkappa, r_{\varkappa+1}, \ldots, r_{\varkappa+i}$$

pārstrādā par vārdu

$$\varrho_k, \varrho_{k+1}, \ldots, \varrho_{k+\iota}\,,$$

mēs lietosim pierakstu

$$\mathfrak{T}(r_\varkappa r_{\varkappa+1} \ldots r_{\varkappa+i}) = \varrho_k \varrho_{k+1} \ldots \varrho_{k+\iota}$$

un teiksim, ka Tjūringa mašīna $\mathfrak{T}$ vārdu

$$r_\varkappa r_{\varkappa+1} \ldots r_{\varkappa+i}$$

pārstrādā par vārdu

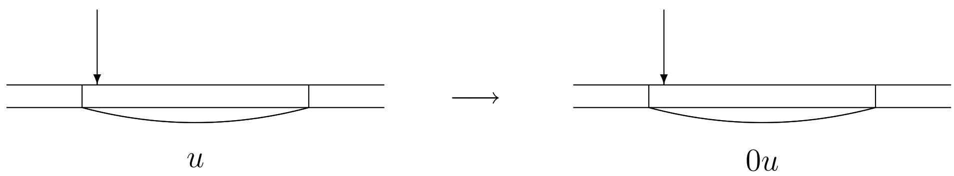
$$\varrho_k \varrho_{k+1} \ldots \varrho_{k+\iota}\,.$$

2.3. Piemēri. *Dotajos piemēros uzskatīsim, ka* $A = \{t, 0, 1\}$.

(i) Tjūringa mašīna $\mathfrak{T}_1$

$$\begin{aligned} q_1\,t &\mapsto 0 \top q_0 \\ q_1\,0 &\mapsto 0 \urcorner q_1 \\ q_1\,1 &\mapsto 1 \urcorner q_1 \end{aligned}$$

vārdam $u \in A_t^*$ priekšā pieraksta 0, proti, $\mathfrak{T}_1(u) = 0u$. Shematiski tas attēlots 3. zīmējumā.

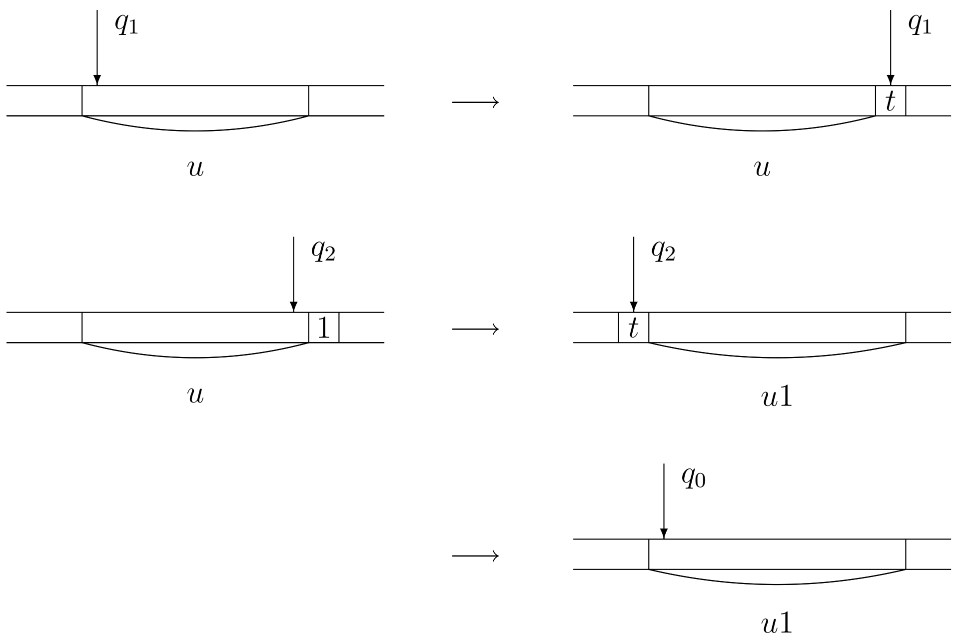


3. zīm. $\mathfrak{T}_1(u) = 0u$.

(ii) Tjūringa mašīna $\mathfrak{T}_2$

$$\begin{array}{lcl} q_1\,t & \mapsto & 1 \Lsh q_2 \\ q_1\,0 & \mapsto & 0 \Rsh q_1 \\ q_1\,1 & \mapsto & 1 \Rsh q_1 \\ q_2\,t & \mapsto & t \Rsh q_0 \\ q_2\,0 & \mapsto & 0 \Lsh q_2 \\ q_2\,1 & \mapsto & 1 \Lsh q_2 \end{array}$$

aiz vārda $u \in A_t^*$ pieraksta 1, proti, $\mathfrak{T}_2(u) = u1$. Tjūringa mašīnas galviņa vispirms dodas uz vārda beigām (stāvoklis q_1), ieraksta 1 (stāvoklis q_1) un aktivizē stāvokli q_2, tad dodas uz vārda sākumu (stāvoklis q_2) un apstājas uz vārda sākuma (stāvoklis q_0). Līdz ar to pēc darba beigšanas galviņa atrodas uz ieraksta sākuma. Shematiski tas attēlots 4. zīmējumā.



4. zīm. $\mathfrak{T}_2(u) = u1$.

(iii) Tjūringa mašīna $\mathfrak{T}_3$ kopē vārdu, proti, $\mathfrak{T}_3(u) = utu$. $\mathfrak{T}_3$ vispirms konstatē, kurš burts jākopē (stāvoklis q_1):

- ja galviņa aplūko burtu t, tad kopēšanas darbi ir beigušies, un atliek galviņu novietot uz ieraksta sākumu (stāvoklis q_{10});
- ja galviņa aplūko burtu 0, tad ar stāvokļu q_2, q_4 palīdzību galviņa nonāk līdz ieraksta beigām, ieraksta 0 (stāvoklis q_4), un atgriežas (stāvokļi q_6, q_8), lai nolasītu nākošo burtu;
- ja galviņa aplūko burtu 1, tad ar stāvokļu q_3, q_5 palīdzību galviņa nonāk līdz ieraksta beigām, ieraksta 1 (stāvoklis q_5), un atgriežas (stāvokļi q_7, q_9), lai nolasītu nākošo burtu.

	t	0	1
q_1	$t \Lsh q_{10}$	$t \Rsh q_2$	$t \Rsh q_3$
q_2	$t \Rsh q_4$	$0 \Rsh q_2$	$1 \Rsh q_2$
q_3	$t \Rsh q_5$	$0 \Rsh q_3$	$1 \Rsh q_3$
q_4	$0 \Lsh q_6$	$0 \Rsh q_4$	$1 \Rsh q_4$
q_5	$1 \Lsh q_7$	$0 \Rsh q_5$	$1 \Rsh q_5$
q_6	$t \Lsh q_8$	$0 \Lsh q_6$	$1 \Lsh q_6$
q_7	$t \Lsh q_9$	$0 \Lsh q_7$	$1 \Lsh q_7$
q_8	$0 \Rsh q_1$	$0 \Lsh q_8$	$1 \Lsh q_8$
q_9	$1 \Rsh q_1$	$0 \Lsh q_9$	$1 \Lsh q_9$
q_{10}	$t \Rsh q_0$	$0 \Lsh q_{10}$	$1 \Lsh q_{10}$

(iv) Tjūringa mašīna $\mathfrak{T}_4$ kopē vārdu pilnīgi blakus, proti, $\mathfrak{T}_4(u) = u^2$. $\mathfrak{T}_4$ vispirms strādā kā Tjūringa mašīna $\mathfrak{T}_3$ (stāvokļi q_1 līdz q_{10}), tad pirmo vārdu nobīda par vienu vietu pa labi, proti,

- ieraksta t un atceras, kādu burtu ir nodzēsusi (stāvoklis q_{11});
- ja nodzēsta 0, tad pāriet stāvoklī q_{12};
- ja nodzēsts 1, tad pāriet stāvoklī q_{13};
- stāvoklis q_{12} nodrošina 0 ierakstīšanu un atcerēšanos, kāds burts ir nodzēsts;
- stāvoklis q_{13} nodrošina 1 ierakstīšanu un atcerēšanos, kāds burts ir nodzēsts;

- ja parādās burts t, tad darbs ir padarīts; ar stāvokļa q_{14} palīdzību galviņa atgriežas uz vārda sākumu.

	t	0	1
q_1	$t \Lsh q_{10}$	$t \Rsh q_2$	$t \Rsh q_3$
q_2	$t \Rsh q_4$	$0 \Rsh q_2$	$1 \Rsh q_2$
q_3	$t \Rsh q_5$	$0 \Rsh q_3$	$1 \Rsh q_3$
q_4	$0 \Lsh q_6$	$0 \Rsh q_4$	$1 \Rsh q_4$
q_5	$1 \Lsh q_7$	$0 \Rsh q_5$	$1 \Rsh q_5$
q_6	$t \Lsh q_8$	$0 \Lsh q_6$	$1 \Lsh q_6$
q_7	$t \Lsh q_9$	$0 \Lsh q_7$	$1 \Lsh q_7$
q_8	$0 \Rsh q_1$	$0 \Lsh q_8$	$1 \Lsh q_8$
q_9	$1 \Rsh q_1$	$0 \Lsh q_9$	$1 \Lsh q_9$
q_{10}	$t \Rsh q_{11}$	$0 \Lsh q_{10}$	$1 \Lsh q_{10}$
q_{11}	$t \top q_0$	$t \Rsh q_{12}$	$t \Rsh q_{13}$
q_{12}	$0 \Lsh q_{14}$	$0 \Rsh q_{12}$	$0 \Rsh q_{13}$
q_{13}	$1 \Lsh q_{14}$	$1 \Rsh q_{12}$	$1 \Rsh q_{13}$
q_{14}	$t \Rsh q_0$	$0 \Lsh q_{14}$	$1 \Lsh q_{14}$

(v) Par vārda w *apgriezto vārdu* w^{-1} sauc vārdu, kurā burti pierakstīti apgrieztā secībā. Apgrieztā vārda formālā definīcija ir šāda:

$$w^{-1} \rightleftharpoons \begin{cases} w, & \text{ja } w \in A, \\ u^{-1}a, & \text{ja } w = au, \text{ kur } a \in A \text{ un } u \in A^+. \end{cases}$$

Tjūringa mašīna $\mathfrak{T}_5$ vārdam w priekšā konstruē apgriezto vārdu, proti, $\mathfrak{T}_5(w) = w^{-1}tw$.

	t	0	1
q_1	$t \Lsh q_{10}$	$t \Lsh q_2$	$t \Lsh q_3$
q_2	$t \Lsh q_4$	$0 \Lsh q_2$	$1 \Lsh q_2$
q_3	$t \Lsh q_5$	$0 \Lsh q_3$	$1 \Lsh q_3$
q_4	$0 \Rsh q_6$	$0 \Lsh q_4$	$1 \Lsh q_4$
q_5	$1 \Rsh q_7$	$0 \Lsh q_5$	$1 \Lsh q_5$
q_6	$t \Rsh q_8$	$0 \Rsh q_6$	$1 \Rsh q_6$
q_7	$t \Rsh q_9$	$0 \Rsh q_7$	$1 \Rsh q_7$
q_8	$0 \Rsh q_1$	$0 \Rsh q_8$	$1 \Rsh q_8$
q_9	$1 \Rsh q_1$	$0 \Rsh q_9$	$1 \Rsh q_9$
q_{10}	$t \Lsh q_{11}$	$0 \Lsh q_{10}$	$1 \Lsh q_{10}$
q_{11}	$t \Rsh q_0$	$0 \Lsh q_{11}$	$1 \Lsh q_{11}$

Tjūringa mašīnas $\mathfrak{T}_5$ darbība līdzīga $\mathfrak{T}_3$ darbībai. $\mathfrak{T}_5$ vispirms konstatē, kurš burts jākopē (stāvoklis q_1):

- ja galviņa aplūko burtu t, tad kopēšanas darbi ir beigušies, un atliek galviņu novietot uz ieraksta sākumu (stāvokļi q_{10}, q_{11});
- ja galviņa aplūko burtu 0, tad ar stāvokļu q_2, q_4 palīdzību galviņa nonāk līdz ieraksta sākumam, ieraksta 0 (stāvoklis q_4), un atgriežas (stāvokļi q_6, q_8), lai nolasītu nākošo burtu;
- ja galviņa aplūko burtu 1, tad ar stāvokļu q_3, q_5 palīdzību galviņa nonāk līdz ieraksta sākumam, ieraksta 1 (stāvoklis q_5), un atgriežas (stāvokļi q_7, q_9), lai nolasītu nākošo burtu.

(vi) Tjūringa mašīna $\mathfrak{T}_6$ konstruē vārda w apgriezto vārdu w^{-1}, proti, $\mathfrak{T}_5(w) = w^{-1}$.

	t	0	1
q_1	$t \Lsh q_{10}$	$t \Lsh q_2$	$t \Lsh q_3$
q_2	$t \Lsh q_4$	$0 \Lsh q_2$	$1 \Lsh q_2$
q_3	$t \Lsh q_5$	$0 \Lsh q_3$	$1 \Lsh q_3$
q_4	$0 \Rsh q_6$	$0 \Lsh q_4$	$1 \Lsh q_4$
q_5	$1 \Rsh q_7$	$0 \Lsh q_5$	$1 \Lsh q_5$
q_6	$t \Rsh q_8$	$0 \Rsh q_6$	$1 \Rsh q_6$
q_7	$t \Rsh q_9$	$0 \Rsh q_7$	$1 \Rsh q_7$
q_8	$0 \Rsh q_1$	$0 \Rsh q_8$	$1 \Rsh q_8$
q_9	$1 \Rsh q_1$	$0 \Rsh q_9$	$1 \Rsh q_9$
q_{10}	$t \Lsh q_{11}$	$t \Lsh q_{10}$	$t \Lsh q_{10}$
q_{11}	$t \Rsh q_0$	$0 \Lsh q_{11}$	$1 \Lsh q_{11}$

Tjūringa mašīnas $\mathfrak{T}_6$ darbība līdzīga $\mathfrak{T}_5$ darbībai. Vienīgā atšķirība ir komandas ar stāvokli q_{10}. Tjūringa mašīna $\mathfrak{T}_6$ nodzēš sākotnējo vārdu w.

(vii) Tjūringa mašīna $\mathfrak{T}_7$ vārdam w tieši priekšā konstruē apgriezto vārdu, proti, $\mathfrak{T}_7(w) = w^{-1}w$. $\mathfrak{T}_7$ vispirms strādā kā Tjūringa mašīna $\mathfrak{T}_5$ (stāvokļi q_1 līdz q_9), tad vārdu w nobīda par vienu vietu pa kreisi, proti,

- ieraksta t un atceras, kādu burtu ir nodzēsusi (stāvoklis q_{10});
- ja nodzēsta 0, tad pāriet stāvoklī q_{12};

- ja nodzēsts 1, tad pāriet stāvoklī q_{11};
- stāvoklis q_{12} nodrošina 0 ierakstīšanu un atcerēšanos, kāds burts ir nodzēsts;
- stāvoklis q_{11} nodrošina 1 ierakstīšanu un atcerēšanos, kāds burts ir nodzēsts;
- ja parādās burts t, tad darbs ir padarīts; ar stāvokļa q_{13} palīdzību galviņa dodas uz vārda sākumu.

	t	0	1
q_1	$t \Lsh q_{10}$	$t \Lsh q_2$	$t \Lsh q_3$
q_2	$t \Lsh q_4$	$0 \Lsh q_2$	$1 \Lsh q_2$
q_3	$t \Lsh q_5$	$0 \Lsh q_3$	$1 \Lsh q_3$
q_4	$0 \Rsh q_6$	$0 \Lsh q_4$	$1 \Lsh q_4$
q_5	$1 \Rsh q_7$	$0 \Lsh q_5$	$1 \Lsh q_5$
q_6	$t \Rsh q_8$	$0 \Rsh q_6$	$1 \Rsh q_6$
q_7	$t \Rsh q_9$	$0 \Rsh q_7$	$1 \Rsh q_7$
q_8	$0 \Rsh q_1$	$0 \Rsh q_8$	$1 \Rsh q_8$
q_9	$1 \Rsh q_1$	$0 \Rsh q_9$	$1 \Rsh q_9$
q_{10}	$t \top q_0$	$t \Lsh q_{12}$	$t \Lsh q_{11}$
q_{11}	$1 \Lsh q_{13}$	$1 \Lsh q_{12}$	$1 \Lsh q_{11}$
q_{12}	$0 \Lsh q_{13}$	$0 \Lsh q_{12}$	$0 \Lsh q_{11}$
q_{13}	$t \Rsh q_0$	$0 \Lsh q_{13}$	$1 \Lsh q_{13}$

2.4. Vingrinājumi. *Uzrakstīt Tjūringa mašīnu programmas!*

(i) $\mathfrak{T}_8(w) = wtw^{-1}$;

(ii) $\mathfrak{T}_9(w) = ww^{-1}$.

2.3. Tjūringa mašīnas definīcija

2.5. Definīcija. *Algebru*

$$\mathfrak{T} = \langle Q_0, Q, A, S, q_0, q_1, t, T\rangle$$

sauc par Tjūringa mašīnu, ja:

(i) $Q_0 = Q \setminus \{q_0\}$;

(ii) Q, A — *galīgas kopas;*

(iii) $Q \cap A = \emptyset$;

(iv) $q_0, q_1 \in Q \wedge q_0 \neq q_1$;

(v) $t \in A$;

(vi) $S = \{\Lsh, \top, \Rsh\}$;

(vii) $T : Q_0 \times A \to A \times S \times Q$.

Intuitīvā nozīmē tas viss mums jau ir pazīstams.

2.6. Definīcija. *Vārdu* $w \in (Q \cup A)^+$ *sauc par mašīnas vārdu, ja*

$$w = uqav,$$

kur $u, v \in A^*,\ q \in Q,\ a \in A$.

$$M(T) \leftharpoondown \{w \mid w \text{ ir mašīnas vārds}\}.$$

Mēs lietojam apzīmējumu $M(T)$, lai uzsvērtu, ka atšķirīgām Tjūringa mašīnām var būt dažādi mašīnu vārdi. Viss atkarīgs no alfabētiem Q un A.

Tagad definēsim attēlojumu $M(T) \xrightarrow{\vdash} M(T)$. Tai vietā, lai rakstītu $\vdash(w) = w'$, turpmāk mēs lietosim pierakstu $w \vdash w'$, un teiksim, ka T tieši pārstrādā vārdu w par w'. Pieņemsim, ka

$$w = uqav, \quad \text{kur} \quad u, v \in A^*,\ q \in Q,\ a \in A.$$

(i) Ja $\quad T : qa \mapsto b \Lsh q' \quad$ un $\quad u = \lambda, \quad$ tad $\quad w \vdash q'tbv$.

(ii) Ja $\quad T : qa \mapsto b \Lsh q' \quad$ un $\quad u = u'c, \quad$ kur $c \in A, \quad$ tad $\quad w \vdash u'q'cbv$.

(iii) Ja $T : qa \mapsto b \top q'$, tad $w \vdash uq'bv$.

(iv) Ja $T : qa \mapsto b \Rsh q'$ un $v \neq \lambda$, tad $w \vdash ubq'v$.

(v) Ja $T : qa \mapsto b \Rsh q'$ un $v = \lambda$, tad $w \vdash ubq't$.

Saka, ka T mašīnas vārdu w pārstrādā par mašīnas vārdu w', ja eksistē tāda mašīnas vārdu virkne

$$w_1, w_2, \ldots, w_n, \qquad \text{ka}$$

(i) $w = w_1$;

(ii) $\forall i \in \overline{2, n} \quad w_{i-1} \vdash w_i$;

(iii) $w_n = w'$.

Šai gadījumā lietosim pierakstu $w \Vdash w'$.

Pieņemsim, ka $A_t \leftharpoondown A \setminus \{t\}$. Definēsim divus attēlojumus

$$\varphi : A_t^* \to M(T) \qquad \text{un} \qquad \psi : M(T) \multimap A^*.$$

(i) $\varphi : \lambda \mapsto q_1 t$;
$\varphi : u \mapsto q_1 u$, ja $u \in A_t^+$.

(ii) $\psi : t^k q_0 t^m \mapsto \lambda$;
$\psi : t^k q_0 a t^m \mapsto a$, ja $a \in A_t^+$;
$\psi : t^k q_0 a u b t^m \mapsto aub$, ja $a, b \in A_t^*$ un $u \in A^*$;
pārejos gadījumos attēlojums ψ nav definēts.

2.7. Definīcija. *Saka, ka Tjūringa mašīna $\mathfrak{T}$ vārdu $w \in A_t^*$ pārstrādā par vārdu $w' \in A^*$, ja eksistē tāda vārdu virkne*

$$w, w_0, w_1, w', \qquad \text{ka}$$

(i) $\varphi(w) = w_0$;

(ii) $w_0 \Vdash w_1$;

(iii) $\psi(w_1) = w'$.

Šai gadījumā lietosim pierakstu $\mathfrak{T}(w) = w'$. Saka, ka rēķināšana vārdam w konverğē, un lieto apzīmējumu $\mathfrak{T}(w)\downarrow$, ja eksistē tāds vārds w', ka $\mathfrak{T}(w) = w'$. Pretējā gadījumā saka, ka rēķināšana vārdam w diverğē, un lieto pierakstu $\mathfrak{T}(w)\uparrow$.

Ja $a_n a_{n-1} \dots a_1 a_0$ ir skaitļa $x \in \mathbb{N}$ pieraksts 2-nieku sistēmā, tad

$$x = 2^n a_n + 2^{n-1} a_{n-1} + \dots + 2a_1 + a_0,$$

kur $a_n = 1$ un visi $a_i \in \{0, 1\}$. Šai situācijā turpmāk lietosim apzīmējumu

$$(x)_2 \leftrightharpoons a_n a_{n-1} \dots a_1 a_0.$$

Tā, piemēram, $(2)_2 = 10$, $(5)_2 = 101$, $(9)_2 = 1001$ un $(14)_2 = 1110$.

2.8. Definīcija. *Saka, ka Tjūringa mašīna* $\mathfrak{T} = \langle Q_0, Q, A, S, q_0, q_1, t, T\rangle$ *rēķina funkciju* $f : \mathbb{N} \multimap \mathbb{N}$, *ja:*

(i) $\{0, 1\} \subset A$;

(ii) $x \in \mathrm{Dom}(f) \Leftrightarrow \mathfrak{T}((x)_2)\downarrow$;

(iii) $f(x) = y \Leftrightarrow \mathfrak{T}((x)_2) = (y)_2$.

2.9. Definīcija. *Funkciju* $f : \mathbb{N} \multimap \mathbb{N}$ *sauc par izrēķināmu pēc Tjūringa, ja eksistē tāda Tjūringa mašīna* $\mathfrak{T}$*, kas to rēķina.*

2.10. Piemēri. *Sekojošās funkcijas ir izrēķināmas pēc Tjūringa.*

(i) $y = 0$.

$$\begin{array}{lcl} q_1\, t & \mapsto & 0 \top q_0 \\ q_1\, 0 & \mapsto & t \Rsh q_1 \\ q_1\, 1 & \mapsto & t \Rsh q_1 \end{array}$$

Tjūringa mašīna nodzēš ierakstu $(x)_2$ un ieraksta ciparu 0.

(ii) $y = 2x$.

$$\begin{array}{lcl} q_1\, t & \mapsto & 0 \Lsh q_2 \\ q_1\, 0 & \mapsto & 0 \Rsh q_1 \\ q_1\, 1 & \mapsto & 1 \Rsh q_1 \\ q_2\, t & \mapsto & t \Rsh q_0 \\ q_2\, 0 & \mapsto & 0 \Lsh q_2 \\ q_2\, 1 & \mapsto & 1 \Lsh q_2 \end{array}$$

Mēs ņēmām vērā 2–ku sistēmas specifiku, proti, ja

$$(x)_2 = a_n \dots a_1 a_0\,,$$

tad

$$(2x)_2 = a_n \dots a_1 a_0 0\,.$$

Tā, piemēram,

$$\begin{aligned} (3)_2 &= 11\,, \\ (2 \cdot 3)_2 = (6)_2 &= 110\,. \end{aligned}$$

Līdz ar to $\mathfrak{T}((x)_2) = (x)_2 0$, t.i., Tjūringa mašīna $\mathfrak{T}$ vārdam $(x)_2$ pieraksta 0 galā (stāvoklis q_1) un atgriež galviņu uz ieraksta sākumu (stāvoklis q_2).

(iii) $y = x + 1$.

$$\begin{array}{rcl} q_1\,t & \mapsto & t \ \Lsh\ q_2 \\ q_1\,0 & \mapsto & 0 \ \Rsh\ q_1 \\ q_1\,1 & \mapsto & 1 \ \Rsh\ q_1 \\ q_2\,t & \mapsto & 1 \top q_0 \\ q_2\,0 & \mapsto & 1 \ \Lsh\ q_3 \\ q_2\,1 & \mapsto & 0 \ \Lsh\ q_2 \\ q_3\,t & \mapsto & t \ \Rsh\ q_0 \\ q_3\,0 & \mapsto & 0 \ \Lsh\ q_3 \\ q_3\,1 & \mapsto & 1 \ \Lsh\ q_3 \end{array}$$

Vispirms galviņa dodas uz vārda $(x)_2$ beigām (stāvoklis q_1), tad veic pieskaitīšanu (stāvoklis q_2). Piemēram,

$$\begin{array}{rll} (4)_2 = 100. & \text{No šejienes} & 100 + 1 = 101 \text{ (komanda : } \ q_2\,0 \mapsto 1\ \Lsh\ q_3) \\ (19)_2 = 10011. & \text{No šejienes} & 10011 + 1 = 10100 \\ & \text{(komandas : } & q_2\,1 \mapsto 0\ \Lsh\ q_2 \quad \text{un} \quad q_2\,0 \mapsto 1\ \Lsh\ q_3) \\ (7)_3 = 111. & \text{No šejienes} & 111 + 1 = 1000 \\ & \text{(komandas : } & q_2\,1 \mapsto 0\ \Lsh\ q_2 \quad \text{un} \quad q_2\,t \mapsto 1 \top q_0) \end{array}$$

Visbeidzot galviņa dodas uz ieraksta sākumu (stāvoklis q_3).

2.11. Vingrinājumi. *Pierādīt, ka sekojošās funkcijas ir izrēķināmas pēc Tjūringa!*

$$
\begin{array}{rl}
\text{(i)} & y = 1; \\
\text{(ii)} & y = 4x; \\
\text{(iii)} & y = x + 2; \\
\text{(iv)} & y = x \dot{-} 1\,.
\end{array}
$$

Pieņemsim, ka $\bar{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{N}^n$, tad

$$(\bar{x})_2 \leftharpoondown (x_1)_2 * (x_2)_2 * \ldots * (x_n)_2.$$

2.12. Definīcija. *Saka, ka Tjūringa mašīna* $\mathfrak{T} = \langle Q_0, Q, A, S, q_0, q_1, t, T\rangle$ *rēķina funkciju* $f : \mathbb{N}^n -\!\!\circ\!\!\!\longrightarrow \mathbb{N}$, *ja:*

(i) $\{*, 0, 1\} \subset A$;

(ii) $\bar{x} \in \mathrm{Dom}(f) \Leftrightarrow \mathfrak{T}((\bar{x})_2) \downarrow$;

(iii) $f(\bar{x}) = y \Leftrightarrow \mathfrak{T}((\bar{x})_2) = (y)_2$.

2.13. Definīcija. *Funkciju* $f : \mathbb{N}^n -\!\!\circ\!\!\!\longrightarrow \mathbb{N}$ *sauc par izrēķināmu pēc Tjūringa, ja eksistē tāda Tjūringa mašīna* $\mathfrak{T}$, *kas to rēķina.*

2.14. Piemēri. *Sekojošās funkcijas ir izrēķināmas pēc Tjūringa.*

(i) $u_3^7(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = x_3$.

	t	$*$	0	1
q_1	$t \top q_0$	$t \Rsh q_2$	$t \Rsh q_1$	$t \Rsh q_1$
q_2	$t \top q_0$	$t \Rsh q_3$	$t \Rsh q_2$	$t \Rsh q_2$
q_3	$t \top q_0$	$t \Rsh q_4$	$0 \Rsh q_3$	$1 \Rsh q_3$
q_4	$t \Lsh q_5$	$t \Rsh q_4$	$t \Rsh q_4$	$t \Rsh q_4$
q_5	$t \Lsh q_5$	$* \Lsh q_6$	$0 \Lsh q_6$	$1 \Lsh q_6$
q_6	$t \Rsh q_0$	$* \Lsh q_6$	$0 \Lsh q_6$	$1 \Lsh q_6$

Ņemam vērā, ka sākuma ieraksts ir

$$(x_1)_2 * (x_2)_2 * (x_3)_2 * (x_4)_2 * (x_5)_2 * (x_6)_2 * (x_7)_2\,,$$

tāpēc jānodzēš vārds

$$(x_1)_2 * (x_2)_2 * \quad — \quad \text{stāvokļi} \quad q_1, q_2,$$

tad jāsaglabā vārds

$$(x_3)_2 \quad — \quad \text{stāvokis} \quad q_3,$$

un jānodzēš vārds

$$*(x_4)_2 * (x_5)_2 * (x_6)_2 * (x_7)_2 \quad — \quad \text{stāvokļi} \quad q_3, q_4.$$

Visbeidzot galviņa atgriežas uz ieraksta sākumu (stāvokļi — q_5, q_6).

(ii) $z = x + y$.

	t	$*$	0	1
q_1		$* \Rsh q_2$	$0 \Rsh q_1$	$1 \Rsh q_1$
q_2			$* \Rsh q_4$	$* \Rsh q_3$
q_3	$1 \top q_5$		$1 \Rsh q_4$	$1 \Rsh q_3$
q_4	$0 \top q_5$		$0 \Rsh q_4$	$0 \Rsh q_3$
q_5		$t \Lsh q_{15}$	$t \Lsh q_6$	$t \Lsh q_7$
q_6		$* \Lsh q_8$	$0 \Lsh q_6$	$1 \Lsh q_6$
q_7		$* \Lsh q_9$	$0 \Lsh q_7$	$1 \Lsh q_7$
q_8		$* \Lsh q_{10}$	$0 \Lsh q_8$	$1 \Lsh q_8$
q_9		$* \Lsh q_{11}$	$0 \Lsh q_9$	$1 \Lsh q_9$
q_{10}	$* \Rsh q_{12}$		$* \Rsh q_{12}$	$* \Rsh q_{14}$
q_{11}	$1 \Rsh q_{13}$		$1 \Rsh q_{13}$	$0 \Lsh q_{11}$
q_{12}		$0 \Rsh q_{16}$		
q_{13}		$* \Lsh q_{10}$	$0 \Rsh q_{13}$	$1 \Rsh q_{13}$
q_{14}		$1 \Rsh q_{16}$		
q_{15}			$t \Lsh q_{18}$	$t \Lsh q_{17}$
q_{16}	$t \Lsh q_5$	$* \Rsh q_{16}$	$0 \Rsh q_{16}$	$1 \Rsh q_{16}$
q_{17}		$1 \Lsh q_{19}$	$1 \Lsh q_{18}$	$1 \Lsh q_{17}$
q_{18}		$0 \Lsh q_{19}$	$0 \Lsh q_{18}$	$0 \Lsh q_{17}$
q_{19}	$t \Rsh q_0$		$0 \Lsh q_{19}$	$1 \Lsh q_{19}$

Lielākas uzskatāmības labad mēs tabulā neierakstījām $t \ \top \ q_0$, jo šais vietās varēja rakstīt arī ko citu, piemēram, $* \ \Rsh \ q_7$. Šie ieraksti neietekmē saskaitīšanas $x+y$ rēķināšanu. Kā mašīnas vārds kortežs $(1,3)$ izskatās šādi:

$$q_1 1 * 11\,.$$

Tagad nodemonstrēsim, kā notiek rēķināšana (mēs praktiski visur atmetīsim burtus t vārda sākumā un beigās). Papildus mēs atzīmēsim svarīgākos rēķināšanas etapus.

$$\begin{array}{llllll}
\text{(i)} & q_1 1 * 11 & \vdash 1 q_1 * 11 & \vdash 1 * q_2 11 & \vdash 1 * * q_3 1 & \vdash 1 * * 1 q_3 t \\
 & & \vdash 1 * * 1 q_5 1 & & & \\
\text{(ii)} & \vdash 1 * * q_7 1 & \vdash 1 * q_7 * 1 & \vdash 1 q_9 * * 1 & \vdash q_{11} 1 * * 1 & \vdash q_{11} t 0 * * 1 \\
 & & \vdash 1 q_{13} 0 * * 1 & \vdash 10 q_{13} * * 1 & \vdash 1 q_{10} 0 * * 1 & \vdash 1 * q_{12} * * 1 \\
 & & \vdash 1 * 0 q_{16} * 1 & \vdash 1 * 0 * q_{16} 1 & \vdash 1 * 0 * 1 q_{16} t & \vdash 1 * 0 * q_5 1 \\
\text{(iii)} & \vdash 1 * 0 q_7 * & \vdash 1 * q_9 0 * & \vdash 1 q_9 * 0 * & \vdash q_{11} 1 * 0 * & \vdash q_{11} t 0 * 0 * \\
 & & \vdash 1 q_{13} 0 * 0 * & \vdash 10 q_{13} * 0 * & \vdash 1 q_{10} 0 * 0 * & \vdash 1 * q_{12} * 0 * \\
 & & \vdash 1 * 0 q_{16} 0 * & \vdash 1 * 00 q_{16} & \vdash 1 * 00 * q_{16} t & \vdash 1 * 00 q_5 * \\
\text{(iv)} & \vdash 1 * 0 q_{15} 0 & \vdash 1 * q_{18} 0 & \vdash 1 q_{18} * 0 & \vdash q_{19} 100 & \vdash q_{19} t 100 \\
 & & \vdash q_0 100 & & &
\end{array}$$

(i) Tjūringa mašīna ieraksta vēl vienu zvaigznīti, un galviņa pārvietojas uz vārda beigām (stāvoklis q_5):

$$1 * 11 \ \mapsto \ 1 * * 11\,.$$

(ii) Tjūringa mašīna pieskaita 1, atzīmē nākošo poziciju un galviņa atgriežas uz vārda beigām (stāvoklis q_5):

$$1 * * 11 \ \mapsto \ 1 * 0 * 1\,.$$

(iii) Tjūringa mašīna pieskaita 2, atzīmē nākošo poziciju un galviņa atgriežas uz vārda beigām (stāvoklis q_5):

$$1 * 0 * 1 \ \mapsto \ 1 * 00 * \,.$$

(iv) Tjūringa mašīna nodzēš visas zvaigznītes un beidz darbu (stāvoklis q_0):

$$1 * 00 * \ \mapsto \ 100\,.$$

2.15. Vingrinājums. *Uzrakstīt Tjūringa mašīnas programmu, kas rēķina funkciju*

$$s(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = x_3 + x_5\,.$$

3. Markova normālie algoritmi.

Markova normālie algoritmi, piemēri. Normāli izrēķināmas funkcijas, piemēri.

Markova normālie algoritmi priekšplānā izvirza ideju par vārdu mehāniskiem pārveidojumiem, tos tiešā veidā nesaistot ar kādu idealizētu mašīnu.

3.1. Substitūcijas

Mūs interesēs tā sauktās substitūcijas

$$u \to v\,;$$

te u un v ir kāda fiksēta alfabēta A vārdi, proti, $uv \in A^*$. Pielietot substitūciju vārdam $w \in A^*$, nozīmē tā pirmo ieeju u aizstāt ar v. Tā, piemēram, pielietot substitūciju

$$01 \to 110$$

vārdam 0111 nozīmē šo vārdu aizstāt ar vārdu 11011. Nedaudz sarežģītāka ir situācija ar vārdu

$$110110001100110\,.$$

Šai vārdā ir trīs dažādas vārda 01 ieejas, proti,

$$(11, 01, 10001100110),\ (1101100, 01, 100110) \quad \text{un} \quad (11011000110, 01, 10)\,.$$

Kā jau iepriekš teicām, pielietot substitūciju, nozīmē tā pirmo ieeju u aizstāt ar v. Tā rezultātā, pielietot substitūciju

$$01 \quad \to \quad 110$$

vārdam

$$110110001100110$$

nozīmē šo vārdu aizstāt ar vārdu

$$1111010001100110\,,$$

nevis

$$1101100110100110 \quad \text{vai} \quad 1101100011011010\,.$$

Tagad noprecizēsim, ko mēs saprotam ar terminu *vārda u pirmā ieeja vārdā w*.

3.1. Definīcija. *Vārda v ieeju (u_1, v, u_1') vārdā w sauc par vārda v pirmo ieeju vārdā w, ja katrai vārda v ieejai (u, v, u') vārdā w ir spēkā nevienādība $|u_1| \le |u|$.*

Markova normālais algoritms paredz substitūcijas pielietošanu atkārtoti, tik ilgi, kamēr tas iespējams. Tā rezultātā pielietojot substitūciju

$$1001 \;\to\; 110$$

vārdam

$$1001010010$$

iegūstam šādu vārdu virkni

$$1001010010 \vdash 110010010 \vdash 11100010\,.$$

Sarežģītāka situācija ir ja dotas vairākas substitūcijas, tad nepieciešams fiksēt lineāru sakārtojumu. Piemēram, dotas substitūcijas

$$1001 \;\to\; 110 \qquad (2)$$

$$01 \;\to\; 10 \qquad (3)$$

Šai situācijā pielietojam visu laiku substitūciju (2), bet substitūciju (3) pielietojam tikai tad, ja nevar pielietot substitūciju (2). Piemēram, vārdam 010011 pielietojot šīs substitūcijas iegūstam šādu vārdu virkni

$$010011 \;\vdash\; 01101 \;\vdash\; 10101 \;\vdash\; 11001 \;\vdash\; 1110$$

Ievērojam, ka katrā solī cenšamies pielietot substitūciju (2), un tikai tad, ja to nevar pielietot, tad mēģinam pielietot substitūciju (3). Tā vārdam 010011 var pielietot substitūciju (2), tāpēc iegūstam pārveidojumu

$$010011 \vdash 01101$$

taču vārdam 01101 nevar pielietot substitūciju (2), tāpēc lietojam substitūciju (3)

$$01101 \;\vdash\; 10101$$

Tas pats attiecas uz vārdu 10101, proti, vārdam 0101 nevar pielietot substitūciju (2), tāpēc lietojam substitūciju (3)

$$10101 \vdash 11001$$

taču vārdam 11001 substitūciju (2) var pielietot, tāpēc arī to pielietojam (kaut arī iepriekšējā solī lietojām substitūciju (3))

$$11001 \vdash 1110$$

Ja pielietota substitūcija

$$u \rightarrow .v$$

tad vairs nevienu substitūciju nedrīkst lietot. Šāda veida substitūcijas sauc par *noslēdzošām* substitūcijām. Tās paredzētas, lai varētu pārtraukt substitūciju pielietošanu. Piemēram, dotas substitūcijas

$$\begin{array}{rcl} 1001 & \rightarrow & 110 \\ 11 & \rightarrow & . \end{array}$$

Vārdam 1010011 pielietojot šīs substitūcijas iegūstam šādu vārdu virkni

$$1010011 \vdash 101101 \vdash 1001$$

Kaut arī potenciālā nozīmē vārdam 1001 būtu iespējams pielietot substitūciju $1001 \rightarrow 110$, tomēr tas netiek darīts, jo iepriekšējā solī pielietota noslēdzošā substitūcija $11 \rightarrow .$

3.2. Markova normālā algoritma definīcija

3.2. Definīcija. *Kortežu*

$$S = \langle \langle u_1, v_1, \delta_1 \rangle, \langle u_2, v_2, \delta_2 \rangle, \ldots, \langle u_s, v_s, \delta_s \rangle \rangle$$

sauc par shēmu S alfabētā A, ja

$$\forall i \in \overline{1, s} \ \ (u_i v_i \in A^* \ \wedge \ \delta_i \in \overline{0, 1} \,) \, .$$

Kortežu $\langle u_i, v_i, \delta_i \rangle$ sauc par substitūciju. Pāri

$$\mathfrak{A} = \langle \, A, S \, \rangle$$

sauc par Markova normālo algoritmu alfabētā A.

Mēs parasti lietosim īsāku terminu un Markova normālos algoritmus sauksim par *normāliem algoritmiem*, vai vēl īšāk — par *algoritmiem*, ja no konteksta tāpat būs skaidrs kādā nozīmētas šis termins lietots. Parasti korteža $\langle u, v, 0\rangle$ apzīmēšanai lieto pierakstu $u \ \rightarrow \ v$, bet korteža $\langle u, v, 1\rangle$ apzīmēšanai — pierakstu $u \ \rightarrow \ .v$.

3.3. Definīcija. *Saka, ka vārds w nepakļaujas algoritmam $\mathfrak{A}$, ja neviens no vārdiem*

$$u_1, u_2, \ldots, u_s$$

nav vārda w dalītājs.

Saka, ka algoritms $\mathfrak{A}$ vārdu w *tieši pārstrādā* par vārdu w', ja:

- $\exists \sigma \in \overline{1, s} \quad (w = uu_\sigma v \ \wedge \ w' = uv_\sigma v)$;
- σ ir mazākais indekss i, kuram vārds u_i ir vārda w dalītājs;
- (u, u_σ, v) ir vārda u_σ pirmā ieeja vārdā w.

Ja $\delta_\sigma = 1$, tad mēdz lietot arī terminu *algoritms $\mathfrak{A}$ vārdu w noslēdzoši pārstrādā par vārdu w'*. Šai situācijā lieto apzīmējumu

$$\mathfrak{A} : w \vdash .w'$$

Ja no konteksta tāpat ir skaidrs, kurš algoritms domāts, tad pieraksta $\mathfrak{A} : w \vdash .w'$ vietā mēdz lietot īsāku pierakstu $w \vdash .w'$. Ja $\delta_\sigma = 0$, tad lieto apzīmējumu

$$\mathfrak{A} : w \vdash w'$$

vai arī apzīmējumu $w \vdash w'$, ja no konteksta tāpat ir skaidrs, kurš algoritms domāts.

Saka, ka algoritms $\mathfrak{A}$ vārdu w *pārstrādā* par vārdu w', ja eksistē tāda vārdu virkne

$$w_0, w_1, \ldots, w_\tau \ ,$$

ka

- $w_0 = w \ \wedge \ w_\tau = w'$;
- ja nu gadījumā $\tau = 0$, tad w nepakļaujas algoritmam $\mathfrak{A}$;

- $\forall i \in \overline{0, \tau - 2} \quad \mathfrak{A} : w_i \vdash w_{i+1}$;
- $\mathfrak{A} : w_{\tau-1} \vdash .w_\tau$, vai arī
  $\mathfrak{A} : w_{\tau-1} \vdash w_\tau$ un w_τ nepakļaujas algoritmam $\mathfrak{A}$.

Šai situācijā lieto apzīmējumu $\mathfrak{A}(w) = w'$. Ja $\mathfrak{A} : w_{\tau-1} \vdash .w_\tau$, tad lieto apzīmējumu

$$\mathfrak{A} : w \models .w' \qquad \text{vai} \qquad w \models .w'$$

ja turpretī $\mathfrak{A} : w_{\tau-1} \vdash w_\tau$, vai arī $\tau = 0$, tad lieto apzīmējumu

$$\mathfrak{A} : w \models w' \qquad \text{vai} \qquad w \models w'$$

Katru vārdu w_i mēs sauksim par *starprezultātu* un lietosim kādu no apzīmējumiem

$$\mathfrak{A} : w_i \Vdash w_{i+j} \qquad \text{vai} \qquad w_i \Vdash w_{i+j}$$

te $\quad 0 \leq i \leq i + j \leq \tau$.

Saka, ka algoritms $\mathfrak{A}$ vārdam w *diverģē* un lieto apzīmējumu $\mathfrak{A}(w) \uparrow$, ja eksistē tāda bezgalīga vārdu virkne

$$w_0, w_1, \ldots, w_n, \ldots$$

ka

- $w_0 = w \qquad$ un
- $\forall i \quad \mathfrak{A} : w_i \vdash w_{i+1}$.

Pretējā gadījumā saka, ka algoritms vārdam w *konverģē*, un lieto apzīmējumu $\mathfrak{A}(w) \downarrow$.

Parasti Markova normālo algoritmu $\langle A, S \rangle$ identificē ar shēmu S. Arī mēs tā darīsim, ja neradīsies pārpratumi.

3.4. Piemēri. *Visos piemēros vienādības $\mathfrak{A}_i(u) = w$ attiecas tikai uz vārdiem $u \in \overline{0,1}^*$.*

(i) $\quad \mathfrak{A}_1(u) = u0$.

$$\begin{array}{rcl} \bigstar 0 & \to & 0\bigstar \\ \bigstar 1 & \to & 1\bigstar \\ \bigstar & \to & .0 \\ & \to & \bigstar \end{array}$$

Algoritms $\mathfrak{A}_1$ vispirms vārda sākumā ieraksta $\bigstar$ (substitūcija $\;\to \bigstar$), tad pārvieto $\bigstar$ uz vārda beigām (substitūcijas $\bigstar 0 \to 0\bigstar$ un $\bigstar 1 \to 1\bigstar$); visbeidzot vārda beigās algoritms ieraksta 0 (substitūcija $\bigstar \to .0$) un beidz darbu. Ja $u = 011$, tad pārstrādāšanas procss izskatās šādi:

$$011 \vdash \bigstar 011 \vdash 0\bigstar 11 \vdash 01\bigstar 1 \vdash 011\bigstar \vdash .0111$$

(ii) $\mathfrak{A}_2(u) = uw$; te w — patvaļīgs fiksēts alfabēta $\overline{0,1}$ vārds.

$$\begin{array}{rcl} \bigstar 0 & \to & 0\bigstar \\ \bigstar 1 & \to & 1\bigstar \\ \bigstar & \to & .w \\ & \to & \bigstar \end{array}$$

(iii) $\mathfrak{A}_3(u) = w$; te w — patvaļīgs fiksēts alfabēta $\overline{0,1}$ vārds.

$$\begin{array}{rcl} 0 & \to & \\ 1 & \to & \\ & \to & .w \end{array}$$

(iv) $\forall n \in \mathbb{N} \quad \mathfrak{A}_4(u01^n) = \mathfrak{A}_4(1^n) = 1^n$.

$$\begin{array}{rcl} 00 & \to & 0 \\ 10 & \to & 0 \\ 0 & \to & \end{array}$$

Tā, piemēram,

$$\mathfrak{A}_4 : 1100111 \models 111$$

(v) $\forall n \in \mathbb{N} \quad \mathfrak{A}_5(1^n0u) = \mathfrak{A}_5(1^n) = 1^n$.

$$\begin{array}{rcl} 00 & \to & 0 \\ 01 & \to & 0 \\ 0 & \to & \end{array}$$

Tā, piemēram,

$$\mathfrak{A}_5 : 1100111 \models 11,$$
$$\mathfrak{A}_5 : 01100111 \models \lambda$$

(vi)

$$\mathfrak{A}_6(u) = \begin{cases} 00, & \text{ja} \quad u \neq w; \\ 11, & \text{ja} \quad u = w. \end{cases}$$

$$\begin{array}{rcl} 0\bigstar & \to & \bigstar 0 \\ 1\bigstar & \to & \bigstar 1 \\ \bigstar 0 & \to & \bigstar \\ \bigstar 1 & \to & \bigstar \\ \bigstar & \to & .00 \\ w0 & \to & \bigstar \\ w1 & \to & \bigstar \\ 0w & \to & \bigstar \\ 1w & \to & \bigstar \\ w & \to & .11 \\ & \to & \bigstar \end{array}$$

Algoritms $\mathfrak{A}_6$ vispirms pārliecinās, vai w nav vārda u dalītājs, piedevām, ja $w \neq u$, tad ar substitūciju $w0 \to \bigstar$, $w1 \to \bigstar$, $0w \to \bigstar$, $1w \to \bigstar$ palīdzību tas tiks konstatēts (vārds u tiks pārveidots par vārdu, kas satur $\bigstar$). Tā rezultātā

$$\mathfrak{A}_6 : u \Vdash u'\bigstar u'' \Vdash \bigstar u'u'' \Vdash \bigstar \vdash .00$$

- Ja turpretī $u = w$, tad tiks pielietota substitūcija $\quad w \to .11$
- Ja w nav vārda u dalītājs, tad

$$u \vdash \bigstar u \models .00$$

Pieņemsim, ka $w = 011$, tad

$$\mathfrak{A}_6 : 101100 \vdash 1\bigstar 0 \vdash \bigstar 10 \vdash \bigstar 0 \vdash \bigstar \vdash .00$$

(vii) $\mathfrak{A}_7(u) = u^2$.

$$
\begin{array}{rcl}
00\clubsuit & \to & 0\clubsuit 0 \\
01\clubsuit & \to & 1\clubsuit 0 \\
10\clubsuit & \to & 0\clubsuit 1 \\
11\clubsuit & \to & 1\clubsuit 1 \\
\diamondsuit 0 & \to & 0\clubsuit 0\diamondsuit \\
\diamondsuit 1 & \to & 1\clubsuit 1\diamondsuit \\
\clubsuit & \to & \bigstar \\
\bigstar & \to & \\
\diamondsuit & \to & . \\
 & \to & \diamondsuit
\end{array}
$$

Vispirms apskatīsim konkrētu piemēru.

$$
\begin{array}{lll}
\mathfrak{A}_7 : & 101 & \vdash \diamondsuit 101 \vdash 1\clubsuit 1\diamondsuit 01 \vdash 1\clubsuit 10\clubsuit 0\diamondsuit 1 \vdash 1\clubsuit 0\clubsuit 10\diamondsuit 1 \\
 & & \vdash 1\clubsuit 0\clubsuit 101\clubsuit 1\diamondsuit \vdash 1\clubsuit 0\clubsuit 11\clubsuit 01\diamondsuit \vdash 1\clubsuit 0\clubsuit 1\clubsuit 101\diamondsuit \\
 & & \Vdash 1\bigstar 0\bigstar 1\bigstar 101\diamondsuit \Vdash 101101\diamondsuit \vdash .101101 = (101)^2
\end{array}
$$

Pieņemsim, ka $u = a_1 a_2 \dots a_n$, kur visi $a_i \in \overline{0,1}$, tad

$$u^{\clubsuit} \leftharpoondown a_1\clubsuit a_2\clubsuit \dots \clubsuit a_n\clubsuit$$

Tagad analizēsim vispārīgo gadījumu.

3.5. Lemma. *Ja* $a \in \overline{0,1}$, $pv \in \overline{0,1}^*$, *tad*

$$\mathfrak{A}_7 : p^{\clubsuit} va\clubsuit w \Vdash p^{\clubsuit} a^{\clubsuit} vw$$

jebkuram $w \in \{0, 1, \diamondsuit\}^*$.

□ (i) Pieņemsim, ka $v = b \in \overline{0,1}$, tad, ņemot vērā pirmās četras substitūcijas, secināms

$$p^{\clubsuit} va\clubsuit w = p^{\clubsuit} ba\clubsuit w \vdash p^{\clubsuit} a\clubsuit bw = p^{\clubsuit} a^{\clubsuit} vw.$$

(ii) Tālākie spriedumi induktīvi. Pieņemsim, ka $v = ub$ un

$$p^{\clubsuit} ua\clubsuit w' \Vdash p^{\clubsuit} a^{\clubsuit} uw'$$

jebkuram $w' \in \{0, 1, \diamondsuit\}^*$, tad

$$\mathfrak{A}_7 : \quad p^{\clubsuit}va\clubsuit w \quad = \quad p^{\clubsuit}uba\clubsuit w \vdash p^{\clubsuit}ua\clubsuit bw \Vdash p^{\clubsuit}a^{\clubsuit}ubw = p^{\clubsuit}a^{\clubsuit}vw. \ \blacksquare$$

3.6. Lemma. *Ja* $vw \in \overline{0,1}^*$, *tad*

$$\mathfrak{A}_7 : \diamondsuit vw \Vdash v^{\clubsuit}v\diamondsuit w.$$

$\square$ (i) Pieņemsim, ka $v = b \in \overline{0,1}$, tad

$$\diamondsuit vw = \diamondsuit bw \vdash b\clubsuit b\diamondsuit w = b^{\clubsuit}b\diamondsuit w.$$

(ii) Tālākie spriedumi induktīvi. Pieņemsim, ka $v = ua$ un

$$\diamondsuit uw' \Vdash u^{\clubsuit}u\diamondsuit w'$$

jebkuram $w' \in \overline{0,1}^*$, tad

$$\diamondsuit vw = \diamondsuit uaw \Vdash u^{\clubsuit}u\diamondsuit aw \vdash u^{\clubsuit}ua\clubsuit a\diamondsuit w \underset{\text{L } 3.5}{\Vdash} u^{\clubsuit}a^{\clubsuit}ua\diamondsuit w = v^{\clubsuit}v\diamondsuit w. \ \blacksquare$$

Tagad mēs varam pamatot, ka $\mathfrak{A}_7(u) = u^2$. Pieņemsim, ka $u = a_1a_2\ldots a_n$, kur visi $a_i \in \overline{0,1}$, tad

$$u \vdash \diamondsuit u \underset{\text{L } 3.6}{\Vdash} u^{\clubsuit}u\diamondsuit \Vdash a_1\bigstar a_2\bigstar\ldots\bigstar a_n\bigstar u\diamondsuit \Vdash a_1a_2\ldots a_nu\diamondsuit \vdash uu = u^2.$$

(viii) $\qquad \mathfrak{A}_8(u) = u^{-1}.$

$$\begin{array}{rcl}
\clubsuit\clubsuit & \to & \bigstar\clubsuit \\
\bigstar\clubsuit & \to & \bigstar \\
\bigstar 0 & \to & 0\bigstar \\
\bigstar 1 & \to & 1\bigstar \\
\bigstar & \to & . \\
\clubsuit 00 & \to & 0\clubsuit 0 \\
\clubsuit 01 & \to & 1\clubsuit 0 \\
\clubsuit 10 & \to & 0\clubsuit 1 \\
\clubsuit 11 & \to & 1\clubsuit 1 \\
 & \to & \clubsuit
\end{array}$$

Vispirms apskatīsim konkrētu piemēru.

$$
\begin{aligned}
\mathfrak{A}_8 : \quad 001 \ &\vdash \clubsuit 001 \vdash 0\clubsuit 01 \vdash 01\clubsuit 0 \\
&\vdash \clubsuit 01\clubsuit 0 \vdash 1\clubsuit 0\clubsuit 0 \\
&\vdash \clubsuit 1\clubsuit 0\clubsuit 0 \vdash \clubsuit\clubsuit 1\clubsuit 0\clubsuit 0 \\
&\vdash \bigstar\clubsuit 1\clubsuit 0\clubsuit 0 \vdash \bigstar 1\clubsuit 0\clubsuit 0 \vdash 1\bigstar\clubsuit 0\clubsuit 0 \\
&\vdash 1\bigstar 0\clubsuit 0 \vdash 10\bigstar\clubsuit 0 \vdash 10\bigstar 0 \vdash 100\bigstar \vdash .100
\end{aligned}
$$

Pieņemsim, ka

$$u = a_1 a_2 \dots a_n,$$

tad pateicoties pēdējām piecām substitūcijām

$$
\begin{aligned}
u \ \vdash \ & \clubsuit a_1 a_2 a_3 \dots a_n \vdash a_2 \clubsuit a_1 a_3 \dots a_n \vdash a_2 a_3 \clubsuit a_1 \dots a_n \Vdash a_2 a_3 \dots a_n \clubsuit a_1 \\
\vdash \ & \clubsuit a_2 a_3 \dots a_n \clubsuit a_1 \vdash a_3 \clubsuit a_2 \dots a_n \clubsuit a_1 \Vdash a_3 \dots a_n \clubsuit a_2 \clubsuit a_1 \\
\vdash \ & \clubsuit a_3 \dots a_n \clubsuit a_2 \clubsuit a_1 \Vdash \clubsuit a_n \clubsuit a_{n-1} \clubsuit \dots \clubsuit a_3 \clubsuit a_2 \clubsuit a_1 \\
\vdash \ & \clubsuit\clubsuit a_n \clubsuit a_{n-1} \clubsuit \dots \clubsuit a_3 \clubsuit a_2 \clubsuit a_1
\end{aligned}
$$

Tagad pienākusi kārta pārējām substitūcijām

$$
\begin{aligned}
\vdash \ & \bigstar\clubsuit a_n \clubsuit a_{n-1} \clubsuit \dots \clubsuit a_3 \clubsuit a_2 \clubsuit a_1 \vdash \bigstar a_n \clubsuit a_{n-1} \clubsuit \dots \clubsuit a_3 \clubsuit a_2 \clubsuit a_1 \\
\vdash \ & a_n \bigstar\clubsuit a_{n-1} \clubsuit \dots \clubsuit a_3 \clubsuit a_2 \clubsuit a_1 \Vdash a_n a_{n-1} \dots a_3 a_2 \bigstar\clubsuit a_1 \\
\vdash \ & a_n a_{n-1} \dots a_3 a_2 \bigstar a_1 \vdash a_n a_{n-1} \dots a_3 a_2 a_1 \bigstar \\
\vdash \ & .a_n a_{n-1} \dots a_3 a_2 a_1 = u^{-1}
\end{aligned}
$$

3.3. Normāli izrēķināmas funkcijas

3.7. Definīcija. *Funkciju $f : \mathbb{N}^n \multimap\!\!\rightarrow \mathbb{N}$ sauc par normāli izrēķināmu, ja eksistē tāds normālais algoritms $\mathfrak{A} = \langle A, S \rangle$, ka*

- $\{0, 1, *\} \subseteq A$,
-

$$
\mathfrak{A}((\bar{x})_2) \quad = \quad \begin{cases} (y)_2, & \text{ja} \quad f(\bar{x}) = y; \\ \mathfrak{A}((\bar{x})_2) \uparrow, & \text{ja} \quad \bar{x} \notin \mathrm{Dom}(f). \end{cases}
$$

Šai gadījumā saka, ka normākais algoritms $\mathfrak{A}$ rēķina funkciju $f : \mathbb{N}^n \multimap\!\!\rightarrow \mathbb{N}$. Ievērojiet, ja vārds u nav skaitļu korteža $\bar{x}$ pieraksts divnieku sistēmā, tad $\mathfrak{A}(u)$ izturēšanās drīkst būt patvaļīga.

3.8. Piemēri. *Sekojošās funkcijas ir normāli izrēķināmas.*

(i) $o(x) = 0$.

$$\begin{array}{rcl} 1 & \to & 0 \\ 00 & \to & 0 \end{array}$$

(ii) $s(x) = x + 1$.

$$\begin{array}{rcl} \bigstar 0 & \to & 0\bigstar \\ \bigstar 1 & \to & 1\bigstar \\ 0\bigstar & \to & .1 \\ 1\bigstar & \to & \Diamond 0 \\ 0\Diamond & \to & .1 \\ 1\Diamond & \to & \Diamond 0 \\ \Diamond & \to & .1 \\ & \to & \bigstar \end{array}$$

Vispirms atrodam vārda beigas (substitūcijas, kas satur $\bigstar$), tad pieskaitam 2–ku sistēmā skaitli 1. Piemēram,

$$11 \ \vdash \ \bigstar 11 \vdash 1\bigstar 1 \vdash 11\bigstar \vdash 1\Diamond 0 \vdash \Diamond 00 \vdash .100$$

(iii) $u^n_m(\bar{x}) = x_m$.

Ja $m = 1$

$$\begin{array}{rcl} *0 & \to & * \\ *1 & \to & * \\ * & \to & \end{array}$$

Ja $m > 1$

$$\begin{array}{rcl} \Diamond 0 & \to & \Diamond \\ \Diamond 1 & \to & \Diamond \\ \Diamond * & \to & \Diamond \\ \Diamond & \to & . \\ \bigstar 0 & \to & 0\bigstar \\ \bigstar 1 & \to & 1\bigstar \\ \bigstar * & \to & \Diamond \\ \bigstar & \to & . \\ 0\clubsuit & \to & \clubsuit \\ 1\clubsuit & \to & \clubsuit \\ \clubsuit^{m-1} & \to & \bigstar \\ * & \to & \clubsuit \end{array}$$

Pieņemsim, ka $\bar{x} = (3, 0, 1)$, tad $(\bar{x})_2 = 11 * 0 * 1$. Ja $m = 1$, tad

$$(\bar{x})_2 = 11 * 0 * 1 \vdash 11 * *1 \vdash 11 * * \vdash 11* \vdash 11 = (3)_2$$

Ja $m = 2$, tad

$$\begin{array}{rcl} (\bar{x})_2 & = & 11 * 0 * 1 \vdash 11\clubsuit 0 * 1 \vdash 1\clubsuit 0 * 1 \vdash \clubsuit 0 * 1 \\ & \vdash & \bigstar 0 * 1 \vdash 0\bigstar * 1 \vdash 0\diamondsuit 1 \vdash 0\diamondsuit \vdash .0 \end{array}$$

Ja $m = 3$, tad

$$\begin{array}{rcl} (\bar{x})_2 & = & 11 * 0 * 1 \vdash 11\clubsuit 0 * 1 \vdash 1\clubsuit 0 * 1 \vdash \clubsuit 0 * 1 \vdash \clubsuit 0\clubsuit 1 \vdash \clubsuit\clubsuit 1 \\ & \vdash & \bigstar 1 \vdash 1\bigstar \vdash .1 \end{array}$$

Saprotams, katram m atbilst savs normālais algoritms.

(iv) $z = x + y$.

$$\left.\begin{array}{rcl} 0\clubsuit\heartsuit & \to & \clubsuit 0 \\ 1\clubsuit\heartsuit & \to & \clubsuit 1 \\ \clubsuit\heartsuit & \to & \clubsuit 0 \end{array}\right\} \text{Jāpieskaita } 0 \cdot 2^k, \text{ tāpēc jāpārbīda tikai } \clubsuit.$$

$$\left.\begin{array}{rcl} 0\clubsuit\spadesuit & \to & \clubsuit 1 \\ 1\clubsuit\spadesuit & \to & +\clubsuit 0 \\ 0+ & \to & 1 \\ 1+ & \to & +0 \\ + & \to & 1 \end{array}\right\} \text{Jāpieskaita } 1 \cdot 2^k, \text{ tāpēc jāpārbīda } \clubsuit \text{ un jāveic pati pieskaitīšana.}$$

$$\left.\begin{array}{rcl} 0\heartsuit & \to & \heartsuit 0 \\ 1\heartsuit & \to & \heartsuit 1 \\ *\heartsuit & \to & \heartsuit * \end{array}\right\} \text{Dodamies pieskaitīt } 0 \cdot 2^k.$$

$$\left.\begin{array}{rcl} 0\spadesuit & \to & \spadesuit 0 \\ 1\spadesuit & \to & \spadesuit 1 \\ *\spadesuit & \to & \spadesuit * \end{array}\right\} \text{Dodamies pieskaitīt } 1 \cdot 2^k.$$

$$\begin{array}{rcll} 0\diamondsuit & \to & \heartsuit\diamondsuit & \text{Konstatējam, ka jāpieskaita } 0 \cdot 2^k. \\ 1\diamondsuit & \to & \spadesuit\diamondsuit & \text{Konstatējam, ka jāpieskaita } 1 \cdot 2^k. \end{array}$$

$$\left.\begin{array}{rcl} *\diamondsuit & \rightarrow & \\ \clubsuit & \rightarrow & . \end{array}\right\} \text{ Darba beigas.}$$

$$\left.\begin{array}{rcl} \bigstar 0 & \rightarrow & 0\bigstar \\ \bigstar 1 & \rightarrow & 1\bigstar \\ \bigstar & \rightarrow & \diamondsuit \\ * & \rightarrow & \clubsuit * \bigstar \end{array}\right\} \text{ Darba sākums.}$$

Iteratīvajā solī standartsituācija ir šāda

$$u\clubsuit u' * v0\diamondsuit \qquad \text{vai} \qquad u\clubsuit u' * v1\diamondsuit$$

Pirmajā gadījumā skaitlim u, kas reprezentēts 2–ku sistēmā jāpieskaita 0, otrajā — jāpieskaita 1.

Pieņemsim, ka

$$(y)_2 = b_n b_{n-1} \ldots b_1 b_0,$$

tad

$$x + y = x + b_0 + 2b_1 + \ldots + 2^{n-1}b_{n-1} + 2^n b_n.$$

Še tiešā veidā tiek realizēta šāda saskaitīšana, piemēram,

$$\begin{array}{rcl} (3,2)_2 = 11 * 10 & \vdash & 11\clubsuit * \bigstar 10 \Vdash 11\clubsuit * 10\diamondsuit \\ & \vdash & 11\clubsuit * 1\heartsuit\diamondsuit \Vdash 11\clubsuit\heartsuit * 1\diamondsuit \\ & \vdash & 1\clubsuit 1 * 1\diamondsuit \vdash 1\clubsuit 1 * \spadesuit\diamondsuit \Vdash 1\clubsuit\spadesuit 1 * \diamondsuit \vdash +\clubsuit 01 * \diamondsuit \\ & \vdash & 1\clubsuit 01 * \diamondsuit \vdash 1\clubsuit 01 \vdash .101 = (5)_2 \end{array}$$

3.9. Vingrinājums. *Uzrakstīt normālā algoritma, kas rēkina funkcija*

$$s(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = x_3 + x_5\,,$$

shēmu!

4. Daļēji rekursīvas funkcijas

Primitīvi rekursīvas funkcijas un predikāti, piemēri. Loģiskas operācijas ar primitīvi rekursīviem predikātiem. Ierobežota summa un reizinājums. Minimizācijas operators un ierobežotais minimizācijas operators, piemēri. Naturālo skaitļu kortežu kodēšana. Daļēji rekursīvas un vispārīgi rekursīvas funkcijas. Akkermana funkcija.

4.1. Primitīvi rekursīvas funkcijas

4.1. Definīcija. *Funkcijas*

$$\begin{aligned} o : \mathbb{N} \to \mathbb{N} \quad &: \quad x \mapsto 0; \\ s : \mathbb{N} \to \mathbb{N} \quad &: \quad x \mapsto x + 1; \\ u_m^n : \mathbb{N}^n \to \mathbb{N} \quad &: \quad (x_1, x_2, \ldots, x_n) \mapsto x_m, \end{aligned}$$

visiem $m \in \overline{1, n}$, *sauc par bāzes funkcijām.*

4.2. Vingrinājums. *Parādīt, ka bāzes funkcijas ir RAM izrēķināmas!*

4.3. Definīcija. *Funkciju* $h : X^n \multimap\!\!\to X$ *sauc par funkciju*

$$\begin{aligned} f \quad &: \quad X^m \multimap\!\!\to X, \\ g_1 \quad &: \quad X^n \multimap\!\!\to X, \\ g_2 \quad &: \quad X^n \multimap\!\!\to X, \\ &\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ g_m \quad &: \quad X^n \multimap\!\!\to X \end{aligned}$$

kompozīciju, ja

$$h(\bar{x}) \leftharpoondown f(g_1(\bar{x}), g_2(\bar{x}), \ldots, g_m(\bar{x})).$$

Īsāka pieraksta labad šai situācijā lietosim apzīmējumu $h = f(g_1, g_2, \ldots, g_m)$.

Vispārīgā gadījumā kopas var definēt izmantojot tā saukto vispārināto indukcijas metodi (principu).

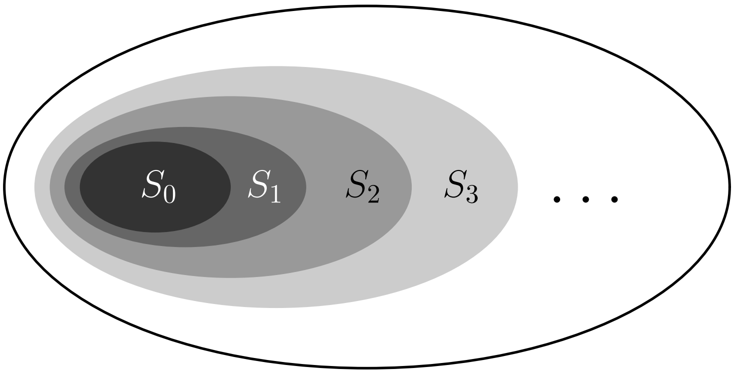
1. solis — nofiksējam sākotnējo kopu S_0;

2. solis — nofiksējam *izveduma likumus* $\mathfrak{F}$.

Kopu S_0' definējam kā kopu, kas iegūta no kopas S_0 elementiem izmantojot izveduma likumus $\mathfrak{F}$. Tagad kopu S_1 definējam kā apvienojumu $S_1 \leftharpoondown S_0 \cup S_0'$. Nākošajā solī definējam kopu S_1' kā kopu, kas iegūta no kopas S_1 elementiem izmantojot izveduma likumus $\mathfrak{F}$, un definējam S_2 kā apvienojumu $S_2 \leftharpoondown S_1 \cup S_1'$. Tā rezultātā iegūstam kopu virkni

$$S_0 \subseteq S_1 \subseteq S_2 \subseteq \ldots \subseteq S_n \subseteq \ldots$$

Visbeidzot definējam $S \leftharpoondown \bigcup\limits_{k=0}^{\infty} S_k$. Dotās shēmas ilustrāciju skatīt 5. zīmējumā.



5.zīm. Kopas S definīcija ar vispārināto indukciju.

Tagad šo pašu paskaidrosim ar konkrētu piemēru.

4.4. Definīcija. *Kopu $\mathfrak{K}(S)$ sauc par kopas*

$$S \subseteq \{f : X^n \to X \mid n \in \mathbb{N}\}$$

klonu, ja

(i) $\mathfrak{K}(S)$ *satur attēlojumus*

$$\omega_m^n : X^n \to X : (x_1, x_2, \ldots, x_n) \mapsto x_m$$

visiem $n \in \mathbb{Z}_+$ un $m \in \overline{1,n}$;

(ii) *ja* $f, g_1, g_2 \ldots, g_m \in \mathfrak{K}(S)$ *un* $h = f(g_1, g_2 \ldots, g_m)$*, tad* $h \in \mathfrak{K}(S)$.

Saskaņā ar vispārināto indukcijas principu klons $\mathfrak{K}(S)$ ir definēts korekti.

4.5. Definīcija. *Saka, ka funkcija* $h(\bar{x}, y)$ *iegūta no funkcijām* $f(\bar{x})$, $g(\bar{x}, y, z)$ *ar primitīvi rekursīvās shēmas palīdzību, ja* h *definēta induktīvi ar nosacījumiem*

$$\begin{cases} h(\bar{x}, 0) & \leftharpoondown & f(\bar{x}); \\ h(\bar{x}, y+1) & \leftharpoondown & g(\bar{x}, y, h(\bar{x}, y)). \end{cases}$$

Speciālā gadījumā, ja $h_1(y)$ *definēta induktīvi ar nosacījumiem*

$$\begin{cases} h_1(0) & \leftharpoondown & c \in \mathbb{N}; \\ h_1(y+1) & \leftharpoondown & g(y, h(y)), \end{cases}$$

tad saka, ka h_1 *iegūta no konstantes* c *un funkcijas* $g_2(y, z)$ *ar primitīvi rekursīvās shēmas palīdzību.*

Abos gadījumos saka, ka funkcija h iegūta no primitīvi rekursīvām funkcijām ar primitīvi rekursīvās shēmas palīdzību.

4.6. Definīcija. *Funkciju* h *sauc par primitīvi rekursīvu funkciju, ja tā apmierina kaut vienu no sekojošiem nosacījumiem:*

- h *ir bāzes funkcija;*
- h *ir primitīvi rekursīvu funkciju kompozīcija;*
- h *iegūta no primitīvi rekursīvām funkcijām ar primitīvi rekursīvās shēmas palīdzību.*

Saskaņā ar vispārināto indukcijas principu primitīvi rekursīvo funkciju klase definēta korekti. Šai gadījumā

$$S_0 \leftharpoondown \{o, s\} \cup \{u_m^n |\, n \in \mathbb{Z}_+ \ \text{ un } \ m \in \overline{1, n}\}.$$

Savukārt izveduma likumi $\mathfrak{F}$ ir šādi.

- Ja

$$\begin{array}{rcl} f & : & \mathbb{N}^m \to \mathbb{N}, \\ g_1 & : & \mathbb{N}^n \to \mathbb{N}, \\ g_2 & : & \mathbb{N}^n \to \mathbb{N}, \\ . & . & . \quad . \quad . \\ g_m & : & \mathbb{N}^n \to \mathbb{N} \end{array}$$

  ir kopas $\mathfrak{X}$ funkcijas, tad

$$h(\bar{x}) \leftharpoondown f(g_1(\bar{x}), g_2(\bar{x}), \ldots, g_m(\bar{x}))$$

  ir kopas $\mathfrak{X}$ funkcija.

- Ja

$$\begin{array}{rcl} f & : & \mathbb{N}^n \to \mathbb{N}, \\ g & : & \mathbb{N}^{n+2} \to \mathbb{N} \end{array}$$

  ir kopas $\mathfrak{X}$ funkcijas, tad $h : \mathbb{N}^{n+1} \to \mathbb{N}$, kas definēta induktīvi ar nosacījumiem

$$\begin{cases} h(\bar{x}, 0) & \leftharpoondown & f(\bar{x}); \\ h(\bar{x}, y+1) & \leftharpoondown & g(\bar{x}, y, h(\bar{x}, y)) \end{cases}$$

  ir kopas $\mathfrak{X}$ funkcija.

- Ja

$$\begin{array}{rcl} c & \in & \mathbb{N}, \\ g & : & \mathbb{N}^2 \to \mathbb{N} \end{array}$$

  ir kopas $\mathfrak{X}$ funkcijas, tad $h : \mathbb{N} \to \mathbb{N}$, kas definēta induktīvi ar nosacījumiem

$$\begin{cases} h(0) & \leftharpoondown & c, \\ h(y+1) & \leftharpoondown & g(y, h(y)) \end{cases}$$

  ir kopas $\mathfrak{X}$ funkcija.

4.7. Piemēri. *Sekojošās funkcijas ir primitīvi rekursīvas.*

(i) $\forall k \in \mathbb{N} \quad s_k(x) = k$.

Pierādījums induktīvs. Ja $k = 0$, tad $s_0(x) = o(x)$. Tā ir bāzes funkcija, tātad — primitīvi rekursīva.

Induktīvā pāreja: pieņemam, ka funkcija $s_k(x)$ ir primitīvi rekursīva. Tas ir induktīvais pieņēmums. Mums tagad jāparāda: no šejienes izriet, ka funkcija $s_{k+1}(x)$ ir primitīvi rekursīva. Pats pierādījums ir šāds:

$$s_{k+1}(x) \quad = \quad k + 1 = s(k) = s(s_k(x)).$$

Tātad $s_{k+1}(x)$ ir primitīvi rekursīvu funkciju kompozīcija, tāpēc (Definīcija 4.6) $s_{k+1}(x)$ ir primitīvi rekursīva funkcija.

(ii) $\forall n \in \mathbb{Z}_+ \quad o^n(x_1, x_2, \dots, x_n) = 0$.

$$o^n(x_1, x_2, \dots, x_n) = 0 = o(x_1) = o(u_1^n(x_1, x_2, \dots, x_n))$$

Tātad $o^n(x_1, x_2, \dots, x_n)$ ir primitīvi rekursīvu funkciju kompozīcija, tāpēc (Definīcija 4.6) $o^n(x_1, x_2, \dots, x_n)$ ir primitīvi rekursīva funkcija.

(iii) $\forall n \in \mathbb{Z}_+ \forall k \in \mathbb{N} \quad s_k^n(x_1, x_2, \dots, x_n) = k$.

Pierādījums induktīvs. Ja $k = 0$, tad

$$s_0^n(x_1, x_2, \dots, x_n) = 0 = o^n(x_1, x_2, \dots, x_n).$$

Mēs tikko punktā (ii) parādījām, ka $o^n(x_1, x_2, \dots, x_n)$ ir primitīvi rekursīva, tātad arī $s_0^n(x_1, x_2, \dots, x_n)$ ir primitīvi rekursīva.

Induktīvā pāreja: pieņemam, ka funkcija $s_k^n(x_1, x_2, \dots, x_n)$ ir primitīvi rekursīva. Tas ir induktīvais pieņēmums. Mums tagad jāparāda: no šejienes izriet, ka funkcija $s_{k+1}^n(x_1, x_2, \dots, x_n)$ ir primitīvi rekursīva. Pats pierādījums ir šāds:

$$s_{k+1}^n(x_1, x_2, \dots, x_n) \quad = \quad k + 1 = s(k) = s(s_k^n(x_1, x_2, \dots, x_n)).$$

Tātad $s_{k+1}^n(x_1, x_2, \dots, x_n)$ ir primitīvi rekursīvu funkciju kompozīcija, tāpēc (Definīcija 4.6) $s_{k+1}^n(x_1, x_2, \dots, x_n)$ ir primitīvi rekursīva funkcija.

4.8. Teorēma. *Funkcija $f : \mathbb{N}^n \to \mathbb{N}$ ir primitīvi rekursīva tad un tikai tad, ja eksistē tāda virkne*

$$f_1, f_2, \dots, f_m, \tag{4}$$

ka $f = f_m$ un katram $k \leq m$ izpildās kaut viens no nosacījumiem:

- *$f_k \in \mathbb{N}$;*
- *f_k ir bāzes funkcija;*
- *f_k ir funkciju $f_{i_1}, f_{i_2}, \ldots, f_{i_s}$ kompozīcija, kur $\forall \sigma \in \overline{1, s}$ $(i_\sigma < k)$;*
- *f_k iegūta no funkcijām f_i, f_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$;*
- *f_k iegūta no kādas konstantes f_i un funkcijas f_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$.*

$\square$ $\Rightarrow$ Mēs demonstrēsim pierādījumu, kas būtiski balstās uz vispārināto indukcijas principu. Pieņemsim, ka f ir primitīvi rekursīva funkcija, tad saskaņā ar definīciju 4.6 izpildās kaut viens no sekojošiem nosacījumiem:

(i) f ir bāzes funkcija;

(ii) f ir primitīvi rekursīvu funkciju kompozīcija;

(iii) f iegūta no primitīvi rekursīvām funkcijām ar primitīvi rekursīvās shēmas palīdzību.

- Ja f ir bāzes funkcija, tad funkciju virkne f_1, kur $f = f_1$, ir teorēmā minētā virkne (4), tikai te $m = 1$.
- Ja f ir primitīvi rekursīvu funkciju kompozīcija, tad eksistē tādas primitīvi rekursīvas funkcijas $h, g_1, g_2, \ldots, g_n$, ka $f = h(g_1, g_2, \ldots, g_n)$. Tagad ņemam vērā definīciju 4.6. Saskaņā ar vispārināto indukcijas principu (indukcijas pieņēmums), eksistē tādas virknes

$$\begin{gathered} h_1,\, h_2,\, \ldots\,,\, h_\varkappa, \\ g_{11}, g_{12}, \ldots, g_{1i_1}, \\ g_{21}, g_{22}, \ldots, g_{2i_2}, \\ \ldots\ldots\ldots\ldots\ldots \\ g_{n1}, g_{n2}, \ldots, g_{ni_n}, \end{gathered}$$

ka visas šīs virknes apmierina tādus pašus nosacījumus, kā virkne (4), proti, $h_\varkappa = h, g_{1i_1} = g_1, g_{2i_2} = g_2, \ldots, g_{ni_n} = g_n$; turklāt vēl katram $k \leq \varkappa$ izpildās kaut viens no nosacījumiem:

- $h_k \in \mathbb{N}$;
- h_k ir bāzes funkcija;
- h_k ir funkciju $h_{i_1}, h_{i_2}, \ldots, h_{i_s}$ kompozīcija, kur $\forall\sigma \in \overline{1,s}\ (i_\sigma < k)$;
- h_k iegūta no funkcijām h_i, h_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$;
- h_k iegūta no kādas konstantes h_i un funkcijas h_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$.

Tas pats attiecas arī uz virknēm $g_{\nu 1}, g_{\nu 2}, \ldots, g_{\nu i_\nu}$, proti, katram $k \leq i_\nu$ izpildās kaut viens no nosacījumiem:

- $g_{\nu k} \in \mathbb{N}$;
- $g_{\nu k}$ ir bāzes funkcija;
- $g_{\nu k}$ ir funkciju $g_{\nu j_1}, g_{\nu j_2}, \ldots, g_{\nu j_s}$ kompozīcija, kur $\forall\sigma \in \overline{1,s}\ (j_\sigma < k)$;
- $g_{\nu k}$ iegūta no funkcijām $g_{\nu i}$, $g_{\nu j}$ ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$;
- $g_{\nu k}$ iegūta no kādas konstantes $g_{\nu i}$ un funkcijas $g_{\nu j}$ ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$.

No šejienes, virkne

$$h_1, h_2, \ldots, h_\varkappa, g_{11}, g_{12}, \ldots, g_{1i_1}, \ \ldots\ , g_{n1}, g_{n2}, \ldots, g_{ni_n}, f$$

ir teorēmā minētā virkne (4).

- Ja f iegūta no primitīvi rekursīvām funkcijām ar primitīvi rekursīvās shēmas palīdzību, tad eksistē tādas primitīvi rekursīvas funkcijas g, h, ka
$$\begin{cases} f(\bar{x}, 0) & \leftharpoondown & g(\bar{x}); \\ f(\bar{x}, y+1) & \leftharpoondown & h(\bar{x}, y, h(\bar{x}, y)). \end{cases}$$
Tagad ņemam vērā definīciju 4.6. Saskaņā ar vispārināto indukcijas principu (indukcijas pieņēmums), eksistē tādas virknes
$$\begin{gathered} g_1, g_2, \ldots, g_\varkappa, \\ h_1, h_2, \ldots, h_\nu, \end{gathered}$$

ka visas šīs virknes apmierina tādus pašus nosacījumus, kā virkne (4), proti, $g_\varkappa = g$ un $h_\nu = h$; turklāt vēl katram $k \leq \varkappa$ izpildās kaut viens no nosacījumiem:

- $g_k \in \mathbb{N}$;
- g_k ir bāzes funkcija;
- g_k ir funkciju $g_{i_1}, g_{i_2}, \ldots, g_{i_s}$ kompozīcija, kur $\forall \sigma \in \overline{1, s}\ (i_\sigma < k)$;
- g_k iegūta no funkcijām g_i, g_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$;
- g_k iegūta no kādas konstantes g_i un funkcijas g_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$.

Tas pats attiecas arī uz virkni $h_1, h_2, \ldots, h_\nu$, proti, katram $k \leq \nu$ izpildās kaut viens no nosacījumiem:

- $h_k \in \mathbb{N}$;
- h_k ir bāzes funkcija;
- h_k ir funkciju $h_{j_1}, h_{j_2}, \ldots, h_{j_s}$ kompozīcija, kur $\forall \sigma \in \overline{1, s}\ (j_\sigma < k)$;
- h_k iegūta no funkcijām h_i, h_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$;
- h_k iegūta no kādas konstantes h_i un funkcijas h_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$.

No šejienes, virkne

$$g_1, g_2, \ldots, g_\varkappa, h_1, h_2, \ldots, h_\nu, f$$

ir teorēmā minētā virkne (4).

Ja f iegūta no primitīvi rekursīvām funkcijām ar primitīvi rekursīvās shēmas palīdzību, tad iespējama vēl otra situācija, proti, eksistē tāda konstante $c \in \mathbb{N}$ un primitīvi rekursīva funkcija g, ka

$$\begin{cases} f(0) & \leftharpoondown\ c, \\ f(y+1) & \leftharpoondown\ g(y, h(y)). \end{cases}$$

Tagad atkal ņemam vērā definīciju 4.6. Saskaņā ar vispārināto indukcijas principu (indukcijas pieņēmums), eksistē tāda virkne

$$g_1, g_2, \ldots, g_\varkappa,$$

ka katram $k \leq \varkappa$ izpildās kaut viens no nosacījumiem:

- $g_k \in \mathbb{N}$;
- g_k ir bāzes funkcija;
- g_k ir funkciju $g_{i_1}, g_{i_2}, \ldots, g_{i_s}$ kompozīcija, kur $\forall \sigma \in \overline{1,s}\ (i_\sigma < k)$;
- g_k iegūta no funkcijām g_i, g_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$;
- g_k iegūta no kādas konstantes g_i un funkcijas g_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < k$ un $j < k$.

No šejienes, virkne

$$g_1, g_2, \ldots, g_\varkappa, c, f$$

ir teorēmā minētā virkne (4).

Pamatojoties uz vispārināto indukcijas metodi, secināms, ka teorēmas nepieciešamais nosacījums ir pierādīts.

$\Leftarrow$ Pieņemsim, ka $f : \mathbb{N}^n \to \mathbb{N}$ ir n–argumentu funkcija, kas ir virknes (4) pēdējā funkcija, proti, $f = f_m$. Mēs demonstrēsim induktīvu pierādījumu pēc m. Ja virknes (4) garums $m = 1$, tad iespējamas tikai divas situācijas.

- $f_m \in \mathbb{N}$. Tas nozīmē, ka eksistē tāda konstante $c \in \mathbb{N}$, ka
$$f(\bar{x}) = c = s_c^n(\bar{x}).$$
Saskaņā ar Piemēru 4.7 (iii) funkcija $s_c^n(\bar{x})$ ir primitīvi rekursīva.
- f_m ir bāzes funkcija. Saskaņā ar Definīciju 4.6 bāzes funkcija ir primitīvi rekursīva.

Induktīvā pāreja: pieņemam, ja virknes (4) garums $m \leq l$, tad f_m ir primitīvi rekursīva. Parādīsim, ja virknes (4) garums $m = l + 1$ un $f = f_m$, tad f ir primitīvi rekursīva. Saskaņā ar virknes (4) konstrukciju f_{m+1} apmierina vienu no sekojosīem nosacījumiem.

- $f_{m+1} \in \mathbb{N}$. Tas nozīmē, ka eksistē tāda konstante $c \in \mathbb{N}$, ka
$$f(\bar{x}) = c = s_c^n(\bar{x}).$$
Saskaņā ar Piemēru 4.7 (iii) funkcija $s_c^n(\bar{x})$ ir primitīvi rekursīva.

- f_{m+1} ir bāzes funkcija. Saskaņā ar Definīciju 4.6 bāzes funkcija ir primitīvi rekursīva.

- f_{m+1} ir funkciju $f_{i_1}, f_{i_2}, \ldots, f_{i_s}$ kompozīcija, kur

$$\forall \sigma \in \overline{1, s}\ (i_\sigma < m + 1).$$

  Saskaņā ar indukcijas pieņēmumu visas funkcijas f_{i_σ} ir primitīvi rekursīvas, tāpēc balstoties uz Definīciju 4.6 secināms: f_{m+1} arī ir primitīvi rekursīva, jo f_{m+1} ir primitīvi rekursīvu funkciju kompozīcija.

- f_{m+1} iegūta no funkcijām f_i, f_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < m + 1$ un $j < m + 1$. Saskaņā ar indukcijas pieņēmumu funkcijas f_i un f_j ir primitīvi rekursīvas, tāpēc balstoties uz Definīciju 4.6 secināms: f_{m+1} arī ir primitīvi rekursīva, jo f_{m+1} ir iegūta no primitīvi rekursīvām funkcijām ar primitīvi rekursīvās shēmas palīdzību.

- f_{m+1} iegūta no kādas konstantes f_i un funkcijas f_j ar primitīvi rekursīvās shēmas palīdzību, kur $i < m + 1$ un $j < m + 1$. Saskaņā ar indukcijas pieņēmumu funkcija f_j ir primitīvi rekursīva, tāpēc balstoties uz Definīciju 4.6 secināms: f_{m+1} arī ir primitīvi rekursīva, jo f_{m+1} ir iegūta no primitīvi rekursīvām funkcijām ar primitīvi rekursīvās shēmas palīdzību.

Pamatojoties uz indukcijas metodi, secināms, ka teorēmas pietiekamais nosacījums ir pierādīts. ■

4.9. Teorēma. *Ja $f(y_1, y_2, \ldots, y_m)$ ir primitīvi rekursīva funkcija, tad*

$$h(x_1, x_2, \ldots, x_n) \leftharpoondown f(x_{i_1}, x_{i_2}, \ldots, x_{i_m}),$$

kur $\forall s \in \overline{1, m}\ (i_s \in \overline{1, n})$, ir primitīvi rekursīva funkcija.

$$\square \qquad h(\bar{x}) = f(u^n_{i_1}(\bar{x}), u^n_{i_2}(\bar{x}), \ldots, u^n_{i_m}(\bar{x})) \qquad \blacksquare$$

4.10. Apgalvojums. *Ja $f(x, y)$ ir primitīvi rekursīva funkcija, tad*

(i) $h_1(x_1, x_2) \leftharpoondown f(x_2, x_1)$;

(ii) $h_2(x) \leftharpoondown f(x, x)$;

(iii) $h_3(x_1, x_2, x_3) \leftharpoondown f(x_2, x_3)$;

(iv) $h_4(x) \leftharpoondown f(x, m)$, *kur* $m \in \mathbb{N}$,

ir primitīvi rekursīvas funkcijas.

□ Pirmie trīs pierādījumi balstās uz Teorēmu 4.9.

(i) Izvēlamies $i_1 = 2$ un $i_2 = 1$, tad $h(x_1, x_2) \leftharpoondown f(x_{i_1}, x_{i_2})$ ir primitīvi rekursīva funkcija un

$$h(x_1, x_2) = f(x_{i_1}, x_{i_2}) = f(x_2, x_1) = h_1(x_1, x_2).$$

(ii) Izvēlamies $i_1 = 1$ un $i_2 = 1$, tad $h(x_1) \leftharpoondown f(x_{i_1}, x_{i_2})$ ir primitīvi rekursīva funkcija un

$$h(x_1) = f(x_{i_1}, x_{i_2}) = f(x_1, x_1) = h_2(x_1).$$

(iii) Izvēlamies $i_1 = 2$ un $i_2 = 3$, tad $h(x_1, x_2, x_3) \leftharpoondown f(x_{i_1}, x_{i_2})$ ir primitīvi rekursīva funkcija un

$$h(x_1, x_2, x_3) = f(x_{i_1}, x_{i_2}) = f(x_2, x_3) = h_3(x_1, x_2, x_3).$$

(iv) Funkcija $f(x_1, m)$ ir primitīvi rekursīvu funkciju $f(y_1, y_2)$ un $s_m(x_1)$ (Piemērs 4.7 (i)), $u_1^1(x_1)$ kompozīcija.

$$h_4(x_1) = f(x_1, m) = f(u_1^1(x_1), s_m(x_1)). \quad \blacksquare$$

4.11. Piemēri. *Biežāk sastopamo primitīvi rekursīvo funkciju piemēri.*

(i) $h_1(x, y) \leftharpoondown x + y$.

$$\begin{aligned} h_1(x, 0) &= x; \\ h_1(x, y+1) &= (x+y)+1. \end{aligned}$$

Balstoties uz Teorēmu 4.9 varam apgalvot, ka funkcija $g(x, y, z) \leftharpoondown s(z)$ ir primitīvi rekursīva. Te

x ir mainīgā x_1 lomā;
y — mainīgā x_2 lomā;
z — mainīgā x_3 lomā.

Tā rezultātā izvēlamies $i_1 = 3$ un saskaņā ar Teorēmu 4.9 varam apgalvot, ka funkcija $g(x_1, x_2, x_3) \leftharpoondown s(x_3)$ ir primitīvi rekursīva. No šejienes: funkcija $h_1(x, y)$ iegūta no primitīvi rekursīvām funkcijām $u_1^1(x)$, $g(x, y, z)$ ar primitīvi rekursīvās shēmas palīdzību:

$$\begin{aligned} h_1(x, 0) &= u_1^1(x); \\ h_1(x, y + 1) &= g(x, y, h_1(x, y)), \end{aligned}$$

tādēļ (Definīcija 4.6) $h_1(x, y)$ ir primitīvi rekursīva. Turpmākajos piemēros mēs vairs tik detalizēti nepaskaidrosim, kā jābalstās uz Teorēmu 4.9.

(ii) $h_2(x, y) \leftharpoondown xy$.

$$\begin{aligned} h_2(x, 0) &= 0; \\ h_2(x, y + 1) &= x(y + 1) = xy + y. \end{aligned}$$

Šoreiz funkcija $h_2(x, y)$ iegūta no primitīvi rekursīvām funkcijām $o(x)$, $g(x, y, z) \leftharpoondown h_1(y, z)$ ar primitīvi rekursīvās shēmas palīdzību.

(iii)

$$h_3(x, y) \leftharpoondown \begin{cases} x^y, & \text{ja} \quad x \neq 0 \vee y \neq 0; \\ 1, & \text{ja} \quad x = y = 0. \end{cases}$$

$$\begin{aligned} h_3(x, 0) &= 1; \\ h_3(x, y + 1) &= x^{y+1} = x^y x. \end{aligned}$$

Funkcija $h_3(x, y)$ iegūta no primitīvi rekursīvām funkcijām

$$s_1(x), \quad g(x, y, z) \leftharpoondown h_2(x, z)$$

ar primitīvi rekursīvās shēmas palīdzību.

(iv) $h_4(x) \leftharpoondown x \dot{-} 1$.

$$\begin{aligned} h_4(0) &= 0; \\ h_4(y + 1) &= y. \end{aligned}$$

Funkcija $h_4(y)$ iegūta no konstantes 0 un primitīvi rekursīvās funkcijas $g(y, z) \leftharpoondown u_1^2(y, z)$ ar primitīvi rekursīvās shēmas palīdzību.

(v)

$$h_5(x, y) \leftharpoondown x \dot{-} y \leftharpoondown \begin{cases} 0, & \text{ja} \quad x < y; \\ x - y, & \text{ja} \quad x \geq y. \end{cases}$$

$$\begin{aligned} h_5(x,0) &= x; \\ h_5(x,y+1) &= x \dot{-} (y+1) = (x \dot{-} y) \dot{-} 1. \end{aligned}$$

Funkcija $h_5(x,y)$ iegūta no primitīvi rekursīvām funkcijām

$$u_1^1(x), \quad g(x,y,z) \leftharpoondown h_4(z)$$

ar primitīvi rekursīvās shēmas palīdzību.

(vi)

$$h_6(x) \leftharpoondown \mathrm{sg}(x) \leftharpoondown \begin{cases} 0, & \text{ja} \quad x = 0; \\ 1, & \text{ja} \quad x \neq 0. \end{cases}$$

$$\begin{aligned} h_6(0) &= 0; \\ h_6(y+1) &= 1. \end{aligned}$$

Funkcija $h_6(y)$ iegūta no konstantes 0 un primitīvi rekursīvās funkcijas $g(y,z) \leftharpoondown s_1(y)$ ar primitīvi rekursīvās shēmas palīdzību.

(vii)

$$h_7(x) \leftharpoondown \overline{\mathrm{sg}}(x) \leftharpoondown \begin{cases} 1, & \text{ja} \quad x = 0; \\ 0, & \text{ja} \quad x \neq 0. \end{cases}$$

$$\overline{\mathrm{sg}}(x) = 1 \dot{-} \mathrm{sg}(x),$$

tātad $\overline{\mathrm{sg}}(x)$ ir primitīvi rekursīvo funkciju $y_1 \dot{-} y_2$ un $s_1(x), \mathrm{sg}(x)$ kompozīcija.

(viii) $h_8(x,y) \leftharpoondown |x-y|$.

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

tātad $|x-y|$ ir primitīvi rekursīvo funkciju $y_1 + y_2$ un $h_5(x,y), h_5(y,x)$ kompozīcija.

(ix) $h_9(x) \leftharpoondown x!$.

$$\begin{aligned} h_9(0) &= 1; \\ h_9(y+1) &= (y+1)! = (y+1)y! \end{aligned}$$

Funkcija $h_9(y)$ iegūta no konstantes 1 un primitīvi rekursīvās funkcijas $g(y,z) \leftharpoondown s(y)z$ (Vingrinājums 4.12 (i)) ar primitīvi rekursīvās shēmas palīdzību.

(x) $h_{10}(x,y) \rightleftharpoons \min(x,y)$.

$$\min(x,y) = x \dot{-} (x \dot{-} y),$$

tātad $\min(x,y)$ ir primitīvi rekursīvo funkciju $y_1 \dot{-} y_2$ un $u_1^2(x,y)$, $x \dot{-} y$ kompozīcija.

(xi) $h_{11}(x,y) \rightleftharpoons \max(x,y)$.

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

tātad $\max(x,y)$ ir primitīvi rekursīvo funkciju $y_1 + y_2$ un $u_1^2(x,y)$, $y \dot{-} x$ kompozīcija.

(xii) Pieņemsim, ka $x \neq 0$, tad katru y var izteikt kā summu

$$y = qx + r, \quad \text{kur} \quad 0 \leq r < x.$$

$$h_{12}(x,y) \rightleftharpoons \begin{cases} y, & \text{ja} \quad x = 0; \\ r, & \text{ja} \quad x \neq 0. \end{cases}$$

Ievērojam, ka $y + 1 = qx + (r + 1)$. No šejienes

$$\begin{aligned} h_{12}(x, y+1) &= \begin{cases} h_{12}(x,y) + 1, & \text{ja} \quad h_{12}(x,y) + 1 \neq x; \\ 0, & \text{ja} \quad h_{12}(x,y) + 1 = x. \end{cases} \\ &= (h_{12}(x,y) + 1)\,\mathrm{sg}|h_{12}(x,y) + 1 - x|. \end{aligned}$$

Tā rezultātā

$$\begin{aligned} h_{12}(x,0) &= 0; \\ h_{12}(x,y+1) &= (h_{12}(x,y) + 1)\,\mathrm{sg}|h_{12}(x,y) + 1 - x|. \end{aligned}$$

Līdz ar to funkcija $h_{12}(x,y)$ iegūta no primitīvi rekursīvām funkcijām

$$o(x), \quad \text{un} \quad g(x,y,z) \rightleftharpoons (z+1)\,\mathrm{sg}|z + 1 - x| \quad \text{(Vingrinājums 4.12 (ii))}$$

ar primitīvi rekursīvās shēmas palīdzību.

(xiii) Pieņemsim, ka $x \neq 0$, tad katru y var izteikt kā summu

$$y = qx + r, \quad \text{kur} \quad 0 \leq r < x.$$

$$h_{13}(x,y) \rightleftharpoons \begin{cases} 0, & \text{ja} \quad x = 0; \\ q, & \text{ja} \quad x \neq 0. \end{cases}$$

Ievērojam, ka $y+1 = qx + (r+1)$. No šejienes

$$\begin{aligned} h_{13}(x,y+1) &= \begin{cases} h_{13}(x,y), & \text{ja} \quad h_{12}(x,y+1) \neq 0; \\ h_{13}(x,y)+1, & \text{ja} \quad h_{12}(x,y+1) = 0. \end{cases} \\ &= h_{13}(x,y) + \overline{\mathrm{sg}}(h_{12}(x,y+1)). \end{aligned}$$

Atzīmēsim, ka $0 = h_{13}(0,0) = h_{13}(0,y) = h_{13}(0,y+1)$. Tā kā

$$h_{12}(0,y) \quad = \quad y \begin{cases} = 0, & \text{ja} \quad y = 0; \\ \neq 0, & \text{ja} \quad y \neq 0, \end{cases}$$

tad

$$h_{13}(0,y+1) = h_{13}(0,y) = h_{13}(0,y) + \overline{\mathrm{sg}}(h_{12}(x,y+1)),$$

jo $h_{12}(0,y+1) = y+1 \neq 0$. Tā rezultātā

$$\begin{aligned} h_{13}(x,0) &= 0; \\ h_{13}(x,y+1) &= h_{13}(x,y) + \overline{\mathrm{sg}}(h_{12}(x,y+1)). \end{aligned}$$

Līdz ar to funkcija $h_{13}(x,y)$ iegūta no primitīvi rekursīvām funkcijām

$$o(x) \quad \text{un} \quad g(x,y,z) \rightleftharpoons z + \overline{\mathrm{sg}}(h_{12}(x,y+1)) \ \text{(Vingrinājums 4.12 (iii))}$$

ar primitīvi rekursīvās shēmas palīdzību.

(xiv)

$$\mathrm{div}(x,y) \rightleftharpoons \begin{cases} 1, & \text{ja} \quad h_{12}(x,y) = 0; \\ 0, & \text{ja} \quad h_{12}(x,y) \neq 0. \end{cases}$$

$$\mathrm{div}(x,y) = \overline{\mathrm{sg}}(h_{12}(x,y)).$$

Līdz ar to funkcija $\mathrm{div}(x,y)$ ir primitīvi rekursīvu funkciju $\overline{\mathrm{sg}}(y_1)$ un $h_{12}(x,y)$ kompozīcija.

Atzīmēsim, ja $x \neq 0$, tad

$$\begin{aligned} \mathrm{div}(x,y) &= \begin{cases} 1, & \text{ja} \quad x \diagdown y; \\ 0, & \text{ja} \quad x \nmid y. \end{cases} \\ &= \begin{cases} 1, & \text{ja} \quad \text{skaitlis } y \text{ ir skaitļa } x \text{ daudzkārtnis}; \\ 0, & \text{ja} \quad \text{skaitlis } y \text{ nav skaitļa } x \text{ daudzkārtnis}. \end{cases} \end{aligned}$$

4.12. Vingrinājumi. *Pierādīt, ka dotās funkcijas ir primitīvi rekursīvas!*

$$
\begin{array}{rll}
\text{(i)} & g_1(x_1,x_2,x_3) & \leftharpoondown (x_2+1)^{x_3},\\
\text{(ii)} & g_2(x_1,x_2,x_3) & \leftharpoondown (x_3+1)\,\mathrm{sg}|x_3+1-x_1|,\\
\text{(iii)} & g_3(x_1,x_2,x_3) & \leftharpoondown x_3+\overline{\mathrm{sg}}(h_{12}(x_1,x_2+1)).
\end{array}
$$

(iv) Pieņemsim, ka $f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})$ ir primitīvi rekursīvas funkcijas. Pierādīt, ka

$$s_k^+(\bar{x}) \leftharpoondown f_1(\bar{x}) + f_2(\bar{x}) + \ldots + f_k(\bar{x})$$

ir primitīvi rekursīva funkcija!

(v) Pieņemsim, ka $f(\bar{x})$ un $g(\bar{x})$ ir primitīvi rekursīvas funkcijas. Pierādīt, ka $\overline{\mathrm{sg}}|f(\bar{x}) - g(\bar{x})|$ ir primitīvi rekursīva funkcija!

4.2. Primitīvi rekursīvi predikāti

4.13. Definīcija. *Kopas $\mathbb{N}^n$ apakškopu M sauc par naturālo skaitļu kopā $\mathbb{N}$ definētu n–vietīgu predikātu (attiecību).*

Ja $\bar{x} \in M$, tad mēdz teikt, ka kortežam $\bar{x}$ predikāts M ir *patiess.* Šai situācijā parasti lieto pierakstu $M(\bar{x}) \sim p$. Pretējā gadījumā, ja $\bar{x} \notin M$, saka, ka kortežam $\bar{x}$ predikāts M ir *aplams* un lieto pierakstu $M(\bar{x}) \sim a$.

4.14. Definīcija. *Funkciju*

$$\chi(\bar{x}) \;=\; \begin{cases} 1, & \text{ja} \quad M(\bar{x}) \sim p; \\ 0, & \text{ja} \quad M(\bar{x}) \sim a. \end{cases}$$

sauc par predikāta $M(\bar{x})$ raksturīgo (harakteristisko) funkciju.

Predikātu $M(\bar{x})$ sauc par *primitīvi rekursīvu* predikātu, ja tā raksturīgā funkcija $\chi(\bar{x})$ ir primitīvi rekursīva.

4.15. Lemma. *Ja $M(\bar{x})$ ir primitīvi rekursīvs predikāts, tad negācija $\neg M(\bar{x})$ ir primitīvi rekursīvs predikāts.*

□ Pieņemsim, ka $\chi(\bar{x})$ ir predikāta $M(\bar{x})$ raksturīgā funkcija, tad predikāta $\neg M(\bar{x})$ raksturīgā funkcija $\chi_\neg(\bar{x}) = 1 \dot{-} \chi(\bar{x})$. ■

4.16. Lemma. *Ja $M(\bar{x})$ un $Q(\bar{x})$ ir primitīvi rekursīvi predikāti, tad*

$$M(\bar{x}) \wedge Q(\bar{x})$$

ir primitīvi rekursīvs predikāts.

□ Pieņemsim, ka $\chi_1(\bar{x})$ un $\chi_2(\bar{x})$ ir attiecīgi predikātu $M(\bar{x})$ un $Q(\bar{x})$ raksturīgās funkcijas, tad konjunkcijas $M(\bar{x}) \wedge Q(\bar{x})$ raksturīgā funkcija

$$\chi_\wedge(\bar{x}) = \chi_1(\bar{x})\chi_2(\bar{x}). \quad \blacksquare$$

4.17. Vingrinājumi. *Ja $M(\bar{x})$ un $Q(\bar{x})$ ir primitīvi rekursīvi predikāti, tad*

$$\begin{array}{cl}
\text{(i)} & M(\bar{x}) \vee Q(\bar{x}), \\
\text{(ii)} & M(\bar{x}) \Rightarrow Q(\bar{x}), \\
\text{(iii)} & M(\bar{x}) \Leftrightarrow Q(\bar{x})
\end{array}$$

ir primitīvi rekursīvi predikāti.

4.18. Teorēma. *Ja $f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})$ ir primitīvi rekursīvas funkcijas,*

$$M_1(\bar{x}), M_2(\bar{x}), \ldots, M_k(\bar{x})$$

ir primitīvi rekursīvi predikāti, turklāt vēl katram $\bar{x} \in \mathbb{N}^n$ tieši viens no šiem predikātiem ir patiess, tad

$$f(\bar{x}) = \begin{cases} f_1(\bar{x}), & \text{ja} \quad M_1(\bar{x}) \sim p; \\ f_2(\bar{x}), & \text{ja} \quad M_2(\bar{x}) \sim p; \\ \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ f_k(\bar{x}), & \text{ja} \quad M_k(\bar{x}) \sim p. \end{cases}$$

ir primitīvi rekursīva funkcija.

□ Pieņemsim, ka $\chi_i(\bar{x})$ ir predikāta $M_i(\bar{x})$ raksturīgā funkcija, tad (Vingrinājums 4.12 (iv))

$$f(\bar{x}) = f_1(\bar{x})\chi_1(\bar{x}) + f_2(\bar{x})\chi_2(\bar{x}) + \ldots + f_k(\bar{x})\chi_k(\bar{x})$$

ir primitīvi rekursīva funkcija. $\blacksquare$

4.19. Apgalvojums. *Ja $f(\bar{x})$ un $g(\bar{x})$ ir primitīvi rekursīvas funkcijas, tad $f(\bar{x}) = g(\bar{x})$ ir primitīvi rekursīvs predikāts.*

□ Funkcija $\overline{\mathrm{sg}}|f(\bar{x}) - g(\bar{x})|$ ir predikāta $f(\bar{x}) = g(\bar{x})$ raksturīgā funkcija. Tā ir primitīvi rekursīva (Vingrinājums 4.12 (v)). ■

4.20. Vingrinājumi. *Pieņemsim, ka $f(\bar{x})$ un $g(\bar{x})$ ir primitīvi rekursīvas funkcijas. Pierādīt, ka dotie predikāti ir primitīvi rekursīvi.*

$$\begin{array}{rl} \text{(i)} & f(\bar{x}) \neq g(\bar{x}), \\ \text{(ii)} & f(\bar{x}) \leq g(\bar{x}), \\ \text{(iii)} & f(\bar{x}) < g(\bar{x}). \end{array}$$

4.3. Ierobežota summa un reizinājums

4.21. Definīcija. *Par funkcijas $f(\bar{x}, t)$ ierobežoto summu $\sum\limits_{t<y} f(\bar{x}, t)$ sauc funkciju*

$$\begin{cases} \sum\limits_{t<0} f(\bar{x}, t) & \leftharpoondown \quad 0, \\ \\ \sum\limits_{t<y+1} f(\bar{x}, t) & \leftharpoondown \quad \sum\limits_{t<y} f(\bar{x}, z) + f(\bar{x}, y). \end{cases}$$

4.22. Apgalvojums. *Ja $f(\bar{x}, t)$ ir primitīvi rekursīva funkcija, tad ierobežotā summa ir primitīvi rekursīva funkcija.*

□ Pieņemsim, ka $S(\bar{x}, y) \leftharpoondown \sum\limits_{t<y} f(\bar{x}, t)$, tad

$$\begin{array}{rcl} S(\bar{x}, 0) & = & 0; \\ S(\bar{x}, y+1) & = & S(\bar{x}, y) + f(\bar{x}, y). \end{array}$$

Funkcija $S(\bar{x}, y)$ iegūta no primitīvi rekursīvām funkcijām

$$o^n(\bar{x}), \quad g(\bar{x}, y, z) \leftharpoondown z + f(\bar{x}, y)$$

ar primitīvi rekursīvās shēmas palīdzību. ■

4.23. Sekas. *Ja $f(\bar{x},t)$ un $k(\bar{x},y)$ ir primitīvi rekursīvas funkcijas, tad ierobežotā summa $\sum\limits_{t<k(\bar{x},y)} f(\bar{x},t)$ ir primitīvi rekursīva funkcija.*

□ Ņemot vērā iepriekšējā pierādījuma apzīmējumus, secināms

$$S(\bar{x},k(\bar{x},y)) = \sum_{t<k(\bar{x},y)} f(\bar{x},t).$$

Tātad ierobežotā summa $\sum\limits_{t<k(\bar{x},y)} f(\bar{x},t)$ ir primitīvi rekursīvu funkciju $S(\bar{x},y)$ un $k(\bar{x},y)$ kompozīcija. ■

4.24. Sekas. *Ja $f(\bar{x},t)$ ir primitīvi rekursīva funkcija, tad $\sum\limits_{t\le y} f(\bar{x},t)$ ir primitīvi rekursīva funkcija.*

□

$$\sum_{t\le y} f(\bar{x},t) = \sum_{t<y+1} f(\bar{x},t). \qquad \blacksquare$$

4.25. Definīcija. *Par funkcijas $f(\bar{x},t)$ ierobežoto reizinājumu $\prod\limits_{t<y} f(\bar{x},t)$ sauc funkciju*

$$\begin{cases} \prod\limits_{t<0} f(\bar{x},t) & \leftharpoondown \quad 1, \\ \prod\limits_{t<y+1} f(\bar{x},t) & \leftharpoondown \quad \left(\prod\limits_{t<y} f(\bar{x},z)\right) f(\bar{x},y). \end{cases}$$

4.26. Vingrinājumi. *Pierādīt sekojošos faktus!*

(i) Ja $f(\bar{x},t)$ ir primitīvi rekursīva funkcija, tad ierobežotais reizinājums ir primitīvi rekursīva funkcija.

(ii) Ja $f(\bar{x},t)$ un $k(\bar{x},y)$ ir primitīvi rekursīvas funkcijas, tad ierobežotais reizinājums $\prod\limits_{t<k(\bar{x},y)} f(\bar{x},t)$ ir primitīvi rekursīva funkcija.

(iii) Ja $f(\bar{x},t)$ ir primitīvi rekursīva funkcija, tad $\prod\limits_{t\le y} f(\bar{x},t)$ ir primitīvi rekursīva funkcija.

4.4. Minimizācijas operators

Vispirms iepazīsimies ar šī operatora rekursīvu aprakstu. Mēs meklējam vienādojuma $f(\bar{x}, t) = 0$ atrisinājumu; mēs gribam atrast mazāko t, kas ir šī vienādojuma sakne. Tomēr mēs šo atrisinājumu meklēsim ļoti primitīvi.

0–tais solis:
a) ja $(\bar{x}, 0) \notin \mathrm{Dom}(f)$, tad atzīstam, ka meklēšana ir bijusi nesekmīga, un pārtraucam tālāko meklēšanu;
b) ja $f(\bar{x}, 0) = 0$, tad esam atraduši mazāko atrisinājumu, un darbu beidzam;
c) ja $f(\bar{x}, 0) \neq 0$, tad pārejam pie nākošā soļa;
. . . .
k**–tais solis**:
a) ja $(\bar{x}, k) \notin \mathrm{Dom}(f)$, tad atzīstam, ka meklēšana ir bijusi nesekmīga, un pārtraucam tālāko meklēšanu;
b) ja $f(\bar{x}, k) = 0$, tad esam atraduši mazāko atrisinājumu, un darbu beidzam;
c) ja $f(\bar{x}, k) \neq 0$, tad pārejam pie nākošā soļa;
. . . .

Secinājums. Šī procedūra definē kādu funkciju $g(\bar{x})$, ko tradicionāli apzīmē šādi $g(\bar{x}) \rightharpoondown \mu t(f(\bar{x}, t) = 0)$.

4.27. Definīcija. *Vispirms definējam kopu*

$$M(\bar{x}) \leftharpoondown \{t \mid f(\bar{x}, t) = 0\}.$$

Tagad varam nodefinēt pašu funkciju

$$g(\bar{x}) \leftharpoondown \begin{cases} \min M(\bar{x}), & \text{ja} \quad M(\bar{x}) \neq \emptyset \ \wedge \ \forall t < \min M(\bar{x}) \ [(\bar{x}, t) \in \mathrm{Dom}(f)]; \\ \text{nav definēta}, & \text{pretējā} \quad \text{gadījumā.} \end{cases}$$

Diemžēl, funkcija $g(\bar{x})$ var nebūt primitīvi rekursīva pat ja $f(\bar{x}, t)$ ir primitīvi rekursīva.

4.28. Piemērs.

$$\mu t(x + t = 0) = \begin{cases} 0, & \text{ja} \quad x = 0; \\ \text{nav definēta}, & \text{pretējā gadījumā.} \end{cases}$$

Tā kā primitīvi rekursīva funkcija ir visur definēta, tad $\mu t(x + t = 0)$ nav primitīvi rekursīva.

Pieņemsim, ka

$$\mathrm{Fun}(\mathbb{N}^n) \leftharpoondown \{f | f : \mathbb{N}^n \multimap \mathbb{N}\},$$

tad attēlojumu

$$\mu : \bigcup_{n=2}^{\infty} \mathrm{Fun}(\mathbb{N}^n) \to \bigcup_{n=1}^{\infty} \mathrm{Fun}(\mathbb{N}^n) : f(\bar{x}, y) \mapsto \mu t(f(\bar{x}, t) = 0)$$

sauc par *minimizācijas* jeb *neierobežoto μ operatoru.*

4.5. Ierobežotais μ–operators

Mēs meklēšanu varam ierobežot nofiksējot, cik soļus mēs darbināsim meklēšanas procedūru. Tātad nofiksējam kādu skaitli y un sākam meklēšanu.

0–tais solis:
a) ja $(\bar{x}, 0) \notin \mathrm{Dom}(f)$, tad atzīstam, ka meklēšana ir bijusi nesekmīga, un pārtraucam tālāko meklēšanu;
b) ja $f(\bar{x}, 0) = 0$, tad esam atraduši mazāko atrisinājumu, un darbu beidzam;
c) ja $f(\bar{x}, 0) \neq 0$, tad pārejam pie nākošā soļa;
. . . .

k–tais solis:
a) ja $(\bar{x}, k) \notin \mathrm{Dom}(f)$, tad atzīstam, ka meklēšana ir bijusi nesekmīga, un pārtraucam tālāko meklēšanu;
b) ja $f(\bar{x}, k) = 0$, tad esam atraduši mazāko atrisinājumu, un darbu beidzam;
c) ja $f(\bar{x}, k) \neq 0$, tad pārejam pie nākošā soļa;
. . . .

solis y: uzskatam, ka atrisinājums ir y, un beidzam darbu.

Secinājums. Šī procedūra definē kādu funkciju $g(\bar{x}, y)$, ko tradicionāli apzīmē šādi $g(\bar{x}, y) \rightleftharpoons \mu t < y\ (f(\bar{x}, t) = 0)$.

4.29. Definīcija. *Vispirms definējam kopu $N(\bar{x}, y) \leftharpoondown M(\bar{x}) \cup \{y\}$. Tagad varam nodefinēt pašu funkciju*

$$g(\bar{x}, y) \leftharpoondown \begin{cases} \min N(\bar{x}, y), & \text{ja} \quad \forall t < \min N(\bar{x}, y)\ [(\bar{x}, t) \in \mathrm{Dom}(f)]; \\ \text{nav definēta}, & \text{pretējā} \quad \text{gadījumā}. \end{cases}$$

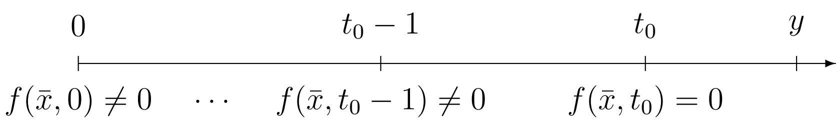
4.30. Sekas. *Ja* $f(\bar{x}, t)$ *ir visur definēta funkcija, tad arī*

$$\mu t < y \ (f(\bar{x}, t) = 0)$$

ir visur definēta funkcija.

4.31. Teorēma. *Ja* $f(\bar{x}, t)$ *ir primitīvi rekursīva funkcija, tad* $\mu t < y \ (f(\bar{x}, t) = 0)$ *ir primitīvi rekursīva funkcija.*

$\square$ $h(\bar{x}, t) \leftharpoondown \prod_{u \leq t} \mathrm{sg}(f(\bar{x}, u))$



Ievērojam, ja $t_0 = \mu t < y \ (f(\bar{x}, t) = 0)$, tad

$$h(\bar{x}, t) = \begin{cases} 1, & \text{ja} \quad t < t_0; \\ 0, & \text{ja} \quad t_0 \leq t < y. \end{cases}$$

No šejienes

$$\sum_{t<y} h(\bar{x}, t) = t_0 = \mu t < y \ (f(\bar{x}, t) = 0)\,.$$

Tā kā $h(\bar{x}, t)$ ir primitīvi rekursīva funkcija, tad $\sum_{t<y} h(\bar{x}, t)$ ir primitīvi rekursīva funkcija (Apgalvojums 4.22). Līdz ar to arī $\mu t < y \ (f(\bar{x}, t) = 0)$ ir primitīvi rekursīva funkcija, jo $\sum_{t<y} h(\bar{x}, t) = \mu t < y \ (f(\bar{x}, t) = 0)$. $\blacksquare$

4.32. Sekas. *Ja* $f(\bar{x}, t)$ *un* $k(\bar{x}, y)$ *ir primitīvi rekursīvas funkcijas, tad* $\mu t < k(\bar{x}, y) \ (f(\bar{x}, t) = 0)$ *ir primitīvi rekursīva funkcija.*

$\square$ Lielākas uzskatāmības labad pieņemsim, ka $g(\bar{x}, u) \leftharpoondown \mu t < u \ (f(\bar{x}, t) = 0)$, bet $h(\bar{x}, y) \leftharpoondown g(\bar{x}, k(\bar{x}, y))$, tad $h(\bar{x}, y) \rightharpoondown \mu t < k(\bar{x}, y) \ (f(\bar{x}, t) = 0)$. $\blacksquare$

4.33. Sekas. *Ja* $f(\bar{x}, t)$ *ir primitīvi rekursīva funkcija, tad* $\mu t \leq y \ (f(\bar{x}, t) = 0)$ *ir primitīvi rekursīva funkcija.*

$\Box$ $\mu t \leq y\ (f(\bar{x},t)=0) = \mu t < y+1\ (f(\bar{x},t)=0)$. $\blacksquare$

Pieņemsim, ka $k(\bar{x},y)$ ir naturālu argumentu funkcija, tad attēlojumu

$$\mu_k : \bigcup_{n=1}^{\infty} \mathrm{Fun}(\mathbb{N}^n) \to \bigcup_{n=1}^{\infty} \mathrm{Fun}(\mathbb{N}^n) : f(\bar{x},y) \mapsto \mu t < k(\bar{x},y)(f(\bar{x},t)=0)$$

sauc par *ierobežoto μ operatoru.* Te speciālā gadījumā pieļaujams , ka $f(\bar{x},t)$ ir viena argumenta t funkcija $f(t)$, tāpat arī pieļaujams, ka $k(\bar{x},y)$ ir viena argumenta y funkcija $k(y)$.

4.6. Citas rekursijas shēmas

4.34. Definīcija. *Pieņemsim, ka $K \subseteq \mathbb{N}^n$. Kopas K apakškopu P sauc par kopā K definētu n–vietīgu predikātu (attiecību).*

Ja $\bar{x} \in P$, tad mēdz teikt, ka kortežam $\bar{x}$ predikāts P ir *patiess.* Šai situācijā parasti lieto pierakstu $P(\bar{x}) \sim p$. Pretējā gadījumā, ja $\bar{x} \in K \setminus P$, saka, ka kortežam $\bar{x}$ predikāts P ir *aplams* un lieto pierakstu $P(\bar{x}) \sim a$. Ja $\bar{x} \in \mathbb{N}^n \setminus K$, tad saka, ka predikāts P *nav definēts.* Šī iemesla dēļ predikātu P sauc par kopā $\mathbb{N}^n$ *daļēji definētu predikātu*, un kopu K sauc par predikāta P *definīcijas apgabalu.* Līdzīgi kā funkciju gadījumā definīcijas apgabala apzīmēšanai lieto pierakstu $\mathrm{Dom}(P)$; tātad $\mathrm{Dom}(P) \rightleftharpoons K$.

4.35. Definīcija. *Funkciju*

$$\chi(\bar{x}) \;=\; \begin{cases} 1, & \text{ja} \quad P(\bar{x}) \sim p; \\ 0, & \text{ja} \quad P(\bar{x}) \sim a; \\ \text{nav definēta}, & \text{ja} \quad \bar{x} \notin \mathrm{Dom}(P). \end{cases}$$

sauc par predikāta $P(\bar{x})$ raksturīgo (harakteristisko) funkciju.

Pieņemsim, ka $P(\bar{x},t)$ ir daļēji definēts predikāts.

0–tais solis:
a) ja $(\bar{x},0) \notin \mathrm{Dom}(P)$, tad atzīstam, ka meklēšana ir bijusi nesekmīga, un pārtraucam tālāko meklēšanu;
b) ja $P(\bar{x},0) \sim p$, tad par atrisinājumu ņemam skaitli 0, un darbu beidzam;
c) ja $P(\bar{x},0) \sim a$, tad pārejam pie nākošā soļa;
. . . .

k–**tais solis**:
a) ja $(\bar{x}, k) \notin \mathrm{Dom}(P)$, tad atzīstam, ka meklēšana ir bijusi nesekmīga, un pārtraucam tālāko meklēšanu;
b) ja $P(\bar{x}, k) \sim p$, tad par atrisinājumu ņemam skaitli k, un darbu beidzam;
c) ja $P(\bar{x}, k) \sim a$, tad pārejam pie nākošā soļa;
. . . .

Secinājums. Šī procedūra definē kādu funkciju $g(\bar{x})$, ko tradicionāli apzīmē šādi $g(\bar{x}) \rightleftharpoons \mu t(P(\bar{x}, t))$.

Līdzīgi var definēt shēmu ar ierobežoto μ–operatoru. Nofiksējam kādu funkciju $k(\bar{x}, y)$ un sākam meklēšanu. Ja $(\bar{x}, y) \notin \mathrm{Dom}(k)$, tad atzīstam, ka meklēšana ir bijusi nesekmīga, un pārtraucam tālāko meklēšanu.
0–tais solis:
a) ja $(\bar{x}, 0) \notin \mathrm{Dom}(P)$, tad atzīstam, ka meklēšana ir bijusi nesekmīga, un pārtraucam tālāko meklēšanu;
b) ja $P(\bar{x}, 0) \sim p$, tad par atrisinājumu ņemam skaitli 0, un darbu beidzam;
c) ja $P(\bar{x}, 0) \sim a$, tad pārejam pie nākošā soļa;
. . . .
k–**tais solis**:
a) ja $(\bar{x}, k) \notin \mathrm{Dom}(P)$, tad atzīstam, ka meklēšana ir bijusi nesekmīga, un pārtraucam tālāko meklēšanu;
b) ja $P(\bar{x}, k) \sim p$, tad par atrisinājumu ņemam skaitli k, un darbu beidzam;
c) ja $P(\bar{x}, k) \sim a$, tad pārejam pie nākošā soļa;
. . . .
solis $k(\bar{x}, y)$: uzskatam, ka atrisinājums ir $k(\bar{x}, y)$, un beidzam darbu.

Secinājums. Šī procedūra definē kādu funkciju $g(\bar{x}, y)$, ko tradicionāli apzīmē šādi $g(\bar{x}, y) \rightleftharpoons \mu t < k(\bar{x}, y)\ (P(\bar{x}, t))$.

4.36. Apgalvojums. *Ja $P(\bar{x}, t)$ ir primitīvi rekursīvs predikāts, tad $\mu t < y\ (P(\bar{x}, t))$ ir primitīvi rekursīva funkcija.*

□ Ja reiz $P(\bar{x}, t)$ ir primitīvi rekursīvs predikāts, tad šī predikāta raksturīgā funkcija
$$\chi(\bar{x}, t) = \begin{cases} 1, & \text{ja} \quad P(\bar{x}, t) \sim p\,; \\ 0, & \text{ja} \quad P(\bar{x}, t) \sim a \end{cases}$$
ir primitīvi rekursīva. Tā kā $\mu t < y\ (\overline{\mathrm{sg}}(\chi(\bar{x}, t)) = 0)$ ir primitīvi rekursīva funkcija un $\mu t < y\ (P(\bar{x}, t)) = \mu t < y\ (\overline{\mathrm{sg}}(\chi(\bar{x}, t)) = 0)$, tad arī $\mu t < y\ (P(\bar{x}, t))$ ir primitīvi rekursīva funkcija. ■

4.37. Vingrinājums. *Ja* $P(\bar{x}, t)$ *ir primitīvi rekursīvs predikāts un* $k(\bar{x}, y)$ *ir primitīvi rekursīva funkcija, tad* $\mu t < k(\bar{x}, y)\ (P(\bar{x}, t))$ *ir primitīvi rekursīva funkcija.*

4.38. Apgalvojums. *Ja* $P(\bar{x}, t)$ *ir primitīvi rekursīvs predikāts, tad* $\forall t < y\ P(\bar{x}, t)$ *un* $\exists t < y\ P(\bar{x}, t)$ *ir primitīvi rekursīvi predikāti.*

□ Ja reiz $P(\bar{x}, t)$ ir primitīvi rekursīvs predikāts, tad šī predikāta raksturīgā funkcija

$$\chi(\bar{x}, t) = \begin{cases} 1, & \text{ja} \quad P(\bar{x}, k) \sim p\,; \\ 0, & \text{ja} \quad P(\bar{x}, k) \sim a \end{cases}$$

ir primitīvi rekursīva. No šejienes predikāta $\forall t < y\ P(\bar{x}, t)$ raksturīgā funkcija

$$\chi^{\forall}(\bar{x}, y) = \prod_{t<y} \chi(\bar{x}, t).$$

Savukārt $\chi^{\exists}(\bar{x}, y) = \mathrm{sg}(\sum_{t<y} \chi(\bar{x}, t))$ ir predikāta $\exists t < y\ P(\bar{x}, t)$ raksturīgā funkcija. ■

4.39. Apgalvojums. *Ja* $P(y_1, \ldots, y_m)$ *ir primitīvi rekursīvs predikāts un* $f_1(\bar{x}), \ldots, f_m(\bar{x})$ *ir primitīvi rekursīvas funkcijas, tad* $P(f_1(\bar{x}), \ldots, f_m(\bar{x}))$ *ir primitīvi rekursīvs predikāts.*

□ Pieņemsim, ka $\chi(y_1, \ldots, y_m)$ ir predikāta $P(y_1, \ldots, y_m)$ raksturīgā funkcija, tad $\chi(f_1(\bar{x}), \ldots, f_m(\bar{x}))$ ir predikāta $P(f_1(\bar{x}), \ldots, f_m(\bar{x}))$ raksturīgā funkcija. ■

4.40. Piemēri.
(i) $(t + 1)^2 > x$ ir primitīvi rekursīvs predikāts.
(ii) $\lfloor\sqrt{x}\rfloor$ ir primitīvi rekursīva funkcija.

$$\lfloor\sqrt{x}\rfloor = \mu t \leq x\ ((t + 1)^2 > x)$$

(iii) Funkcija $D(x) \leftharpoondown \sum_{t \leq x} \mathrm{div}(t, x)$ ir primitīvi rekursīva. Šīs funkcijas sašaurinājums $D|\mathbb{Z}_+$ ir vienāds ar skaitļa x dalītāju skaitu. Funkcija $\mathrm{div}(t, x)$ definēta Piemērā 4.11 (xiv).

(iv) Pieņemsim, ka $\mathbb{P}$ ir visu pirmskaitļu kopa. Predikāts $x \in \mathbb{P}$ ir primitīvi rekursīvs. Pieņemsim, ka $\mathbb{P}(x)$ ir predikāta $x \in \mathbb{P}$ raksturīgā funkcija, tad

$$\mathbb{P}(x) = \begin{cases} 1, & \text{ja} \quad x \in \mathbb{P}\,; \\ 0, & \text{ja} \quad x \notin \mathbb{P}\,. \end{cases}$$

$\mathbb{P}(x) = \overline{\text{sg}}|D(x) - 2|$

(v) Pieņemsim, ka $p(0) \leftharpoondown 0$, toties ja $x \neq 0$, tad $p(x)$ ir pirmskaitlis, kura numurs dabiskajā uzskaitījumā ir x. Tātad

$$\begin{array}{l} p(1) = 2, \quad p(2) = 3, \quad p(3) = 5, \quad p(4) = 7,\; p(5) = 11,\; p(6) = 13, \\ p(7) = 17,\; p(8) = 19,\; p(9) = 23,\; p(10) = 29,\; \ldots \end{array}$$

Vispirms definēsim funkciju

$$g(w) \leftharpoondown \mu t \leq (w! + 1)(t > w \wedge t \in \mathbb{P})\,.$$

Šī funkcija ir primitīvi rekursīva, jo funkcija $w! + 1$ ir primitīvi rekursīva, bet predikāti $t > w$ un $t \in \mathbb{P}$ ir primitīvi rekursīvi predikāti. Atzīmēsim, ka

$$g(0) = 2,\; g(1) = 2,\; g(2) = 3,\; g(3) = 5,\; g(4) = 5,\; g(5) = 7,\; \ldots$$

Pieņemsim, ka $w > 4$, tad skaitli $w! + 1$ nedala neviens no skaitļiem $2, 3, 4, \ldots, w$, tāpēc starp skaitļiem $w + 1, w + 2, \ldots, w! + 1$ ir vismaz viens pirmskaitlis. Līdz ar to funkcija $g(x)$ definē pirmo pirmskaitli, kas lielāks par w.

$$\begin{cases} p(0) & \leftharpoondown \quad 0\,; \\ p(x+1) & \leftharpoondown \quad g(p(x))\,. \end{cases}$$

4.7. Kantora numerācija

4.41. Definīcija. *Funkciju*

$$c(x, y) \leftharpoondown \frac{1}{2}\,(x^2 + 2xy + y^2 + 3x + y)$$

sauc par naturālo skaitļu kopas $\mathbb{N}$ *pāru* (x, y) *Kantora numerāciju.*

4.42. Lemma.

$$c(x, y) = T(x + y) + x, \quad \text{kur} \quad T(n) \leftharpoondown \frac{n(n+1)}{2}\,.$$

□

$$\begin{aligned} T(x+y)+x &= \frac{(x+y)(x+y+1)}{2}+x = \frac{(x+y)^2+(x+y)}{2}+x \\ &= \frac{1}{2}\,(x^2+2xy+y^2)+\frac{1}{2}\,(x+y)+\frac{1}{2}\,(2x) \\ &= \frac{1}{2}\,(x^2+2xy+y^2+3x+y) = c(x,y)\,. \quad \blacksquare \end{aligned}$$

4.43. Sekas. $\mathrm{Ran}(c) \subseteq \mathbb{N}$.

□ Tā kā viens no skaitļiem n, vai $n+1$ ir pārkaitlis, tad $T(n) = \frac{n(n+1)}{2} \in \mathbb{N}$ katram naturālam skaitlim n. No šejienes

$$c(x,y) = T(x+y)+x \in \mathbb{N}\,. \quad \blacksquare$$

4.44. Lemma. $\forall n \in \mathbb{N} \quad T(n)+n+1 = T(n+1)$.

□

$$\begin{aligned} T(n)+n+1 &= \frac{n(n+1)}{2}+n+1 = \frac{n(n+1)}{2}+\frac{2(n+1)}{2} \\ &= \frac{(n+2)(n+1)}{2} = T(n+1)\,. \quad \blacksquare \end{aligned}$$

4.45. Sekas. *Funkcija* $T:\mathbb{N}\to\mathbb{N}: n \mapsto \frac{n(n+1)}{2}$ *ir augoša.*

4.46. Sekas. $c:\mathbb{N}^2\to\mathbb{N}$ *ir injekcija.*

□ (i) Pieņemsim, ka $x+y < x'+y'$, tad

$$\begin{aligned} c(x,y) &= T(x+y)+x < T(x+y)+x+y+1 \\ &= T(x+y+1) \le T(x'+y') \le T(x'+y')+x' \\ &= c(x',y')\,. \end{aligned}$$

(ii) Pieņemsim, ka $x+y = x'+y'$ un $c(x,y)=c(x',y')$, tad

$$T(x+y)+x = c(x,y) = c(x',y') = T(x',y')+x'\,.$$

No šejienes $x = x'$, tāpēc $y = y'$. Tas nozīmē, ja $(x,y) \neq (x',y')$, tad $c(x,y) \neq c(x',y')$. $\blacksquare$

4.47. Lemma. *$c : \mathbb{N}^2 \to \mathbb{N}$ ir sirjekcija.*

□ Pieņemsim, ka $n \in \mathbb{N}$, tad eksistē (Sekas 4.45) tāds y, ka

$$T(y) \le n < T(y+1).$$

Tā kā (Lemma 4.42) $\quad c(x,y) = T(x+y) + x, \quad$ tad

$$\begin{array}{rcl} c(0,y) & = & T(y), \\ c(1,y-1) & = & T(y)+1, \\ \cdot\ \cdot\ \cdot & \cdot & \cdot\ \cdot\ \cdot \\ c(k,y-k) & = & T(y)+k, \\ \cdot\ \cdot\ \cdot & \cdot & \cdot\ \cdot\ \cdot \\ c(y,0) & = & T(y)+y = T(y+1)-1. \text{ (Lemma 4.44)} \end{array}$$

Ja reiz $T(y) \le n < T(y+1)$, tad eksistē tāds k, ka $c(k, y-k) = n$. ■

4.48. Teorēma. *Funkcija $c : \mathbb{N}^2 \to \mathbb{N}$ ir bijekcija.*

□ Sekas 4.46 un Lemma 4.47. ■

4.49. Lemma. *Ja $c(x,y) = n$, tad*

$$\begin{array}{rcl} x & = & n \dot{-} \frac{1}{2} \left\lfloor \frac{q(n)\dot{-}1}{2} \right\rfloor \left\lfloor \frac{q(n)+1}{2} \right\rfloor \rightleftharpoons c_1^2(n), \\ y & = & \left\lfloor \frac{q(n)\dot{-}1}{2} \right\rfloor \dot{-} c_1^2(n) \rightleftharpoons c_2^2(n), \end{array}$$

kur $q(n) \leftrightharpoons \lfloor \sqrt{8n+1} \rfloor$.

□ Pieņemsim, ka $n = c(x,y) = \frac{1}{2}\left(x^2 + 2xy + y^2 + 3x + y\right)$, tad

$$\begin{array}{rcl} 2n & = & x^2 + 2xy + y^2 + 3x + y, \\ 8n+1 & = & 4x^2 + 8xy + 4y^2 + 12x + 4y + 1. \end{array}$$

Ņemam vērā, ka

$$(2x + 2y + 1)^2 = 4x^2 + 8xy + 4y^2 + 4x + 4y + 1,$$

tāpēc $8n+1=(2x+2y+1)^2+8x$. Savukārt

$$(2x+2y+3)^2 = 4x^2+8xy+4y^2+12x+12y+9,$$

tāpēc $8n+1=(2x+2y+3)^2-8x-8$. No šejienes

$$\begin{array}{rcl}
(2x+2y+1)^2 \leq & 8n+1 & < (2x+2y+3)^2, \\
2x+2y+1 \leq & \sqrt{8n+1} & < 2x+2y+3, \\
2x+2y+1 \leq & \lfloor\sqrt{8n+1}\rfloor & < 2x+2y+3, \\
2x+2y+1 \leq & q(n) & < 2x+2y+3, \\
x+y+\frac{1}{2} \leq & \frac{q(n)}{2} & < x+y+1+\frac{1}{2}, \\
x+y \leq & \frac{q(n)-1}{2} & < x+y+1.
\end{array}$$

Ievērojam, ka $q(n)=\lfloor\sqrt{8n+1}\rfloor \geq 1$, tāpēc $q(n)-1=q(n)\dot{-}1$. Līdz ar to

$$\begin{array}{rcl}
x+y \leq & \frac{q(n)\dot{-}1}{2} & < x+y+1, \\
x+y \leq & \left\lfloor\frac{q(n)\dot{-}1}{2}\right\rfloor & < x+y+1.
\end{array}$$

Tā kā abi skaitļi gan $x+y$, gan $x+y+1$ ir veseli skaitļi un

$$\left\lfloor\frac{q(n)\dot{-}1}{2}\right\rfloor < x+y+1,$$

tad

$$x+y=\left\lfloor\frac{q(n)\dot{-}1}{2}\right\rfloor.$$

Atgriežamies nedauz atpakaļ:

$$\begin{array}{rcl}
x+y+\frac{1}{2} \leq & \frac{q(n)}{2} & < x+y+1+\frac{1}{2}, \\
x+y+1 \leq & \frac{q(n)+1}{2} & < x+y+2.
\end{array}$$

Tas ļauj secināt, ka

$$x+y+1=\left\lfloor\frac{q(n)+1}{2}\right\rfloor.$$

Līdz ar to

$$T(x+y) = \frac{1}{2}(x+y)(x+y+1) = \frac{1}{2}\left\lfloor \frac{q(n)\dot{-}1}{2} \right\rfloor \left\lfloor \frac{q(n)+1}{2} \right\rfloor .$$

Tagad atsaucamies uz Lemmu 4.42: $c(x,y) = T(x+y)+x$, tādēļ

$$x = n\dot{-}T(x+y) = n \dot{-} \frac{1}{2}\left\lfloor \frac{q(n)\dot{-}1}{2} \right\rfloor \left\lfloor \frac{q(n)+1}{2} \right\rfloor = c_1^2(n)\,.$$

Tā kā $x+y = \left\lfloor \frac{q(n)\dot{-}1}{2} \right\rfloor$, tad $y = \left\lfloor \frac{q(n)\dot{-}1}{2} \right\rfloor \dot{-} x = \left\lfloor \frac{q(n)\dot{-}1}{2} \right\rfloor \dot{-}\, c_1^2(n) = c_2^2(n)$. ■

4.50. Sekas. *Funkcijas*

$$c_1^2 : \mathbb{N} \to \mathbb{N}, \quad c_2^2 : \mathbb{N} \to \mathbb{N}$$

ir primitīvi rekursīvas.

4.51. Apgalvojums. *Attēlojums*

$$c^{-1} : \mathbb{N} \to \mathbb{N}^2 : n \mapsto (c_1^2(n), c_2^2(n))$$

ir attēlojuma $c : \mathbb{N}^2 \to \mathbb{N}$ *inversais attēlojums.*

□ Pieņemsim, ka $n \in \mathbb{N}$, tad eksistē tādi x, y, ka $c(x,y) = n$, jo $c : \mathbb{N}^2 \to \mathbb{N}$ ir sirjekcija. Saskaņā ar Lemmu 4.49 $x = c_1^2(n)$ un $y = c_2^2(n)$. Līdz ar to

$$c \circ c^{-1}(n) = c(c_1^2(n), c_2^2(n)) = c(x,y) = n.$$

Pieņemsim, ka $(a,b) \in \mathbb{N}^2$, tad saskaņā ar Lemmu 4.49

$$c^{-1} \circ c(a,b) = (c_1^2 \circ c(a,b), c_2^2 \circ c(a,b)) = (a,b). \qquad ■$$

Vispārīgā gadījumā naturālo skaitļu kopas $\mathbb{N}$ n–dimensionālo kortežu $(x_1, x_3, \ldots, x_n)$ Kantora numerāciju $c^n : \mathbb{N}^n \to \mathbb{N}$ definē induktīvi.

$$\begin{array}{rcl}
c^1 : \mathbb{N} \to \mathbb{N} & : & x_1 \mapsto x_1, \\
c^2 : \mathbb{N}^2 \to \mathbb{N} & : & (x_1, x_2) \mapsto c(x_1, x_2), \\
c^3 : \mathbb{N}^3 \to \mathbb{N} & : & (x_1, x_2, x_3) \mapsto c(c^2(x_1, x_2), x_3), \\
\ldots & & \ldots \\
c^{n+1} : \mathbb{N}^{n+1} \to \mathbb{N} & : & (x_1, x_2, \ldots, x_n, x_{n+1}) \mapsto c(c^n(x_1, x_2, \ldots, x_n), x_{n+1}).
\end{array}$$

4.52. Definīcija. *Funkciju*

$$c^n : \mathbb{N}^n \to \mathbb{N}$$

sauc par naturālo skaitļu kopas $\mathbb{N}$ *n–dimensionālo kortežu* $(x_1, x_3, \ldots, x_n)$ *Kantora numerāciju.*

4.53. Vingrinājumi.
(i) Pierādīt, ka $c^n : \mathbb{N}^n \to \mathbb{N}$ ir bijekcija.

(ii) Pieņemsim, ka $n > 2$ un

$$\begin{cases} c_i^n & \leftharpoondown & c_i^{n-1} \circ c_1^2, \quad \text{ja} \quad i \in \overline{1, n-1}, \\ c_n^n & \leftharpoondown & c_2^2. \end{cases}$$

Pierādīt, ka attēlojums

$$c^{-n} : \mathbb{N} \to \mathbb{N}^n : a \mapsto (c_1^n(a), c_2^n(a), \ldots, c_n^n(a))$$

ir attēlojuma $c^n : \mathbb{N}^n \to \mathbb{N}$ inversais attēlojums.

(iii) Pierādīt, ka attēlojumi

$$c^n, c_1^n, c_2^n, \ldots, c_n^n,$$

ir primitīvi rekursīvas funkcijas.

4.8. Daļēji un vispārīgi rekursīvas funkcijas

4.54. Definīcija. *Funkciju h sauc par daļēji rekursīvu funkciju, ja tā apmierina kaut vienu no sekojošiem nosacījumiem:*

- *h ir primitīvi rekursīva funkcija;*
- *h ir daļēji rekursīvu funkciju kompozīcija;*
- *h iegūta no daļēji rekursīvām funkcijām ar primitīvi rekursīvās shēmas palīdzību,*
- *h iegūta no daļēji rekursīvas funkcijas ar neierobežotā μ operatora palīdzību.*

4.55. Definīcija. *Predikātu $P(\bar{x}, t)$ sauc par daļēji rekursīvu predikātu, ja tā raksturīgā funkcija ir daļēji rekursīva.*

4.56. Apgalvojums. *Ja $P(\bar{x}, t)$ ir daļēji rekursīvs predikāts, tad $\mu t(P(\bar{x}, t))$ ir daļēji rekursīva funkcija.*

□ Ivērojam, ka

$$\mu t(P(\bar{x}, t)) = \mu t(\overline{\mathrm{sg}} \circ \chi(\bar{x}, t) = 0),$$

kur $\chi(\bar{x}, t)$ ir predikāta $P(\bar{x}, t)$ raksturīgā funkcija. Tā kā $P(\bar{x}, t)$ ir daļēji rekursīvs predikāts, tad (Definīcija 4.55) $\chi(\bar{x}, t)$ ir daļēji rekursīva funkcija. Līdz ar to $\mu t(P(\bar{x}, t))$ iegūta no daļēji rekursīvas funkcijas ar neierobežotā μ operatora palīdzību, tātad $\mu t(P(\bar{x}, t))$ ir daļēji rekursīva funkcija (Definīcija 4.54). ■

4.57. Definīcija. *Visur definētu daļēji rekursīvu funkciju sauc par vispārīgi rekursīvu funkciju.*

Izrādās, ka eksistē vispārīgi rekursīvas funkcijas, kas nav primitīvi rekursīvas. Tāda, piemēram, ir *Akkermana funkcija*, ko definē šādi:

$$\begin{array}{rcl} \psi(0, y) & \leftharpoondown & y + 1, \\ \psi(x + 1, 0) & \leftharpoondown & \psi(x, 1), \\ \psi(x + 1, y + 1) & \leftharpoondown & \psi(x, \psi(x + 1, y)). \end{array}$$

4.58. Lemma. *Akkermana funkcija ir definēta korekti.*

□ Vispirms kopā $\mathbb{N}^2$ definēsim lineāru sakārtojumu.

$$\begin{array}{ccccccccc} (0,0) & < & (0,1) & < & \ldots & < & (0,k) & < & \ldots \\ (1,0) & < & (1,1) & < & \ldots & < & (1,k) & < & \ldots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ (n,0) & < & (n,1) & < & \ldots & < & (n,k) & < & \ldots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{array}$$

Tagad $\mathbb{N}^2$ ir lineāri sakārtota kopa. Izmantojot indukciju parādīsim, ka Akkermana funkcija $\psi : \mathbb{N}^2 \to \mathbb{N}$ definēta korekti.

(i) $\psi(0, k) \leftharpoondown k + 1$.

(ii) Induktīvais pieņēmums: funkcija $\psi(x, y)$ definēta visiem pāriem (x, y), ja $(x, y) < (n + 1, 0)$. $\psi(n + 1, 0) \leftharpoondown \psi(n, 1)$.

(iii) Induktīvais pieņēmums: funkcija $\psi(x, y)$ definēta visiem pāriem (x, y), ja $(x, y) < (n + 1, k + 1)$. $\psi(n + 1, k + 1) \leftharpoondown \psi(n, \psi(n + 1, k))$. ■

5. RAM izrēķināmas funkcijas

Programma standartizskatā, programmu konkatenācija. Daļēji rekursīvo funkciju sakars ar RAM izrēķināmām funkcijām

Mūsu mērķis — parādīt, ka RAM izrēķināmo funkciju klase sakrīt ar daļēji rekursīvo funkciju klasi.

5.1. Programma standartizskatā

5.1. Definīcija. *Programmu $P = K_1, K_2, \ldots, K_\tau$ sauc par programmu standartizskatā, ja katrai šīs prgrammas nosacītās vadības komandai $J(m, n, q)$ skaitlis $q \leq \tau + 1$.*

5.2. Lemma. *Katrai programmai P eksistē programma P' standartizskatā, kas rēķina to pašu attēlojumu, proti,*

$$\forall (r_k) \in K^\circ \quad P(r_k) \downarrow b \Leftrightarrow P'(r_k) \downarrow b.$$

□ Pieņemsim, ka $P = K_1, K_2, \ldots, K_\tau$. Definējam jaunu programmu $P' \leftharpoondown K'_1, K'_2, \ldots, K'_\tau$, kur

$$K'_i \quad \leftharpoondown \quad \begin{cases} J(m, n, \tau + 1), & \text{ja} \quad K_i = J(m, n, q) \wedge q > \tau + 1; \\ K_i, & \text{pretējā} \quad \text{gadījumā.} \end{cases} \qquad \blacksquare$$

Pieņemsim, ka $P = K_1, K_2, \ldots, K_\tau$ un $P' = K'_1, K'_2, \ldots, K'_s$ ir divas programmas standartizskatā. Programmu

$$P'' = K_1, K_2, \ldots, K_\tau, K_{\tau+1}, K_{\tau+2}, \ldots, K_{\tau+s}$$

sauc par *programmu P un P' konkatnāciju*, ja

$$K_{\tau+i} \quad \leftharpoondown \quad \begin{cases} J(m, n, q + \tau), & \text{ja} \quad K'_i = J(m, n, q); \\ K'_i, & \text{pretējā} \quad \text{gadījumā.} \end{cases}$$

Šai gadījumā lietosim apzīmējumu $P'' = P \# P'$.

Pieņemsim, ka $P(r_k) \downarrow$ un (r'_k) ir beigu konfigurācija, tad lietosim apzīmējumu $P(r_k) \downarrow (r'_k)$.

No programmu konkatenācijas definīcijas uzreiz izriet, ka

$$\exists (r'_k) \; [P(r_k) \downarrow (r'_k) \wedge P'(r'_k) \downarrow (r''_k)] \Leftrightarrow P''(r_k) \downarrow (r''_k).$$

Tātad, ja mēs rēķinam ar programmu P pie sākuma konfigurācijas (r_k) un iegūstam beigu konfigurāciju (r'_k); pēc tam šo beigu konfigurāciju izmantojam kā sākuma konfigurāciju rēķināšanai ar programmu P' un iegūstam beigu konfigurāciju (r''_k), tad to pašu mēs iegūsim, ja veiksim rēķināšanu ar programmu konkatenāciju P'' pie sākuma konfigurācijas (r_k).

Turpmāk mēs uzskatīsim, ka visas programmas ir standartizskatā.

5.2. Programma $P[l_1, l_2, \dots, l_n \to l]$

Mēs vēlamies uzrakstīt programmu, kas saturiski dod to pašu rezultātu, ko programma P, tikai ar dažām modifikācijām. Pieņemsim, ka (r_k) ir sākuma konfigurācija un

$$r_{l_1} = x_1, r_{l_2} = x_2, \dots, r_{l_n} = x_n.$$

Mēs vēlamies, lai

- $P(x_1, x_2, \dots, x_n) \downarrow \;\Leftrightarrow\; P[l_1, l_2, \dots, l_n \to l](r_k) \downarrow$,
- $P(x_1, x_2, \dots, x_n) \downarrow b \wedge P[l_1, l_2, \dots, l_n \to l](r_k) \downarrow (r'_k) \;\Rightarrow\; r'_l = b$,
- $i \leq \varrho_n(P) \wedge i \neq l \Rightarrow r'_i = r_i$.

Tātad mēs vēlamies, lai programma $P[l_1, l_2, \dots, l_n \to l]$ ņemtu sākuma datus no šūnām ar numuriem $l_1, l_2, \dots, l_n$, beigu rezultātu ievietotu šūnā ar numuru l un darba gaitā neizbojātu šūnu saturu sākuma indeksiem $i \neq l$.

Definējam attēlojumu $\rho : Z^\circ \cup S^\circ \cup T^\circ \cup J^\circ \to \mathbb{Z}_+$ ar šādiem nosacījumiem:

$$\begin{aligned}
\rho : Z^\circ \to \mathbb{Z}_+ &\;:\; Z(u) \mapsto u, \\
\rho : S^\circ \to \mathbb{Z}_+ &\;:\; S(u) \mapsto u, \\
\rho : T^\circ \to \mathbb{Z}_+ &\;:\; T(u, v) \mapsto \max(u, v), \\
\rho : J^\circ \to \mathbb{Z}_+ &\;:\; J(u, v, q) \mapsto \max(u, v).
\end{aligned}$$

Pieņemsim, ka programma $P = K_1, K_2, \dots, K_\tau$, tad $\varrho(P) \leftharpoondown \max_{i \in \overline{1,\tau}} \rho(K_i)$. Saturīgi tas nozīmē, ka programma P rēķināšanā neizmanto šūnas, kuru numuri ir lielāki par $\varrho(P)$.

Pieņemsim, ka $\varrho_n(P) \leftharpoondown \max(n, \varrho(P))$. Saturīgi tas nozīmē, ka RAM rēķinot n argumentu funkciju ar programmu P neizmanto šūnas, kuru numuri ir lielāki par $\varrho_n(P)$, un šajās šūnās ierakstīts skaitlis 0.

Pieņemsim, ka $m > \max(\varrho_n(P), l_1, l_2, \ldots, l_n, l)$. Definējam programmu $P' \leftharpoondown K_1', K_2', \ldots, K_\tau'$, kur

$$K_i' \quad \leftharpoondown \quad \begin{cases} Z(u+m), & \text{ja} \quad K_i = Z(u); \\ S(u+m), & \text{ja} \quad K_i = S(u); \\ T(u+m, v+m), & \text{ja} \quad K_i = T(u,v); \\ J(u+m, v+m, q), & \text{ja} \quad K_i = J(u,v,q). \end{cases}$$

Pieņemsim, ka

$$\begin{aligned} P_0 \quad = \quad & T(l_1, m+1), T(l_2, m+2), \ldots, T(l_n, m+n), \\ & Z(m+n+1), Z(m+n+2), \ldots, Z(m+n+\varrho_n(P)), \end{aligned}$$

tad

$$P_m[l_1, l_2, \ldots, l_n \to l] \leftharpoondown P_0 \# P' \# T(m+1, l)\,.$$

Saskaņā ar konstrukciju, ja $m > \max(\varrho_n(P), l_1, l_2, \ldots, l_n, l)$, tad programmas $P[l_1, l_2, \ldots, l_n \to l]$ lomai der katra programma $P_m[l_1, l_2, \ldots, l_n \to l]$.

5.3. Lemma. *Ja $f(y_1, y_2, \ldots, y_m), g_1(\bar{x}), g_2(\bar{x}), \ldots, g_m(\bar{x})$ ir RAM izrēķināmas funkcijas, tad $h(\bar{x}) \leftharpoondown f(g_1(\bar{x}), g_2(\bar{x}), \ldots, g_m(\bar{x}))$ ir RAM izrēķināma funkcija.*

$\square$ Ja reiz $f(y_1, y_2, \ldots, y_m)$ ir RAM izrēķināma funkcija, tad eksistē programma P, kas RAM-rēķina šo funkciju. Tas pats attiecas uz funkcijām $g_i(\bar{x})$, proti, eksistē programmas G^i, kas RAM-rēķina funkcijas $g_i(\bar{x})$.

Izvēlamies $k > m + n$, tad programma

$$\begin{aligned} & G_k^1[1, 2, \ldots, n \to n+1] \\ & \# G_k^2[1, 2, \ldots, n \to n+2] \# \ldots \# G_k^m[1, 2, \ldots, n \to n+m] \\ & \# F_k[n+1, n+2, \ldots, n+m \to 1] \end{aligned}$$

RAM-rēķina funkciju $h(\bar{x})$. $\blacksquare$

5.4. Lemma. *Ja $f(\bar{x})$ un $g(\bar{x}, y, z)$ ir RAM izrēķināmas funkcijas, tad $h(\bar{x}, y)$, kas definēta ar primitīvi rekursīvo shēmu*

$$\begin{cases} h(\bar{x}, 0) & \leftharpoondown \quad f(\bar{x}); \\ h(\bar{x}, y+1) & \leftharpoondown \quad g(\bar{x}, y, h(\bar{x}, y)). \end{cases}$$

ir RAM izrēķināma funkcija.

□ Ja reiz $f(\bar{x})$ ir RAM izrēķināma funkcijas, tad eksistē programma P, kas RAM-rēķina šo funkciju. Tas pats attiecas uz funkciju $g(\bar{x}, y, z)$, proti, eksistē programma G, kas RAM-rēķina funkcijas $g(\bar{x}, y, z)$.

Izvēlamies $k > n + 3$, tad programma

$$\begin{array}{ll} & F_k[1, 2, \ldots, n \to n+2] \\ a. & J(n+1, n+3, b) \\ & S(n+3) \\ & G_k[1, 2, \ldots, n, n+1, n+2 \to n+2] \\ & J(1, 1, a) \\ b. & T(n+2, 1) \end{array}$$

RAM-rēķina funkciju $h(\bar{x})$. Te $a = |F_k[1, 2, \ldots, n \to n+2]| + 1$ un $b = a + |G_k[1, 2, \ldots, n, n+1, n+2 \to n+2]| + 3$. ■

5.5. Vingrinājums. *Ja $g(y, z)$ ir RAM izrēķināma funkcija un $h_1(y)$ definēta induktīvi ar nosacījumiem*

$$\begin{cases} h_1(0) & \leftharpoondown \quad c \in \mathbb{N}; \\ h_1(y+1) & \leftharpoondown \quad g(y, h(y)), \end{cases}$$

tad $h_1(y)$ ir RAM izrēķināma funkcija.

5.6. Lemma. *Ja $f(\bar{x}, y)$ ir RAM izrēķināma funkcija, tad*

$$h(\bar{x}) \leftharpoondown \mu y(f(\bar{x}, y) = 0)$$

ir RAM izrēķināma funkcija.

□ Ja reiz $f(\bar{x}, y)$ ir RAM izrēķināma funkcijas, tad eksistē programma P, kas RAM-rēķina šo funkciju.

Izvēlamies $k > n + 3$, tad programma

$$\begin{array}{ll} & F_k[1, 2, \ldots, n, n+1 \to n+2] \\ & J(n+2, n+3, b) \\ & S(n+1) \\ & J(1, 1, 1) \\ b. & T(n+1, 1) \end{array}$$

RAM-rēķina funkciju $h(\bar{x})$. Te $b = |F_k[1, 2, \ldots, n, n+1 \to n+2]| + 4$. ■

5.7. Apgalvojums. *Katra primitīvi rekursīva funkcija ir RAM izrēķināma.*

□ (i) Katra bāzes funkcija ir RAM izrēķināma (Vingrinājums 4.2).

(ii) RAM izrēķināmu funkciju kompozīcija ir RAM izrēķināma funkcija (Lemma 5.3).

(iii) Ja h iegūta ar primitīvi rekursīvās shemas palīdzību no RAM izrēķināmām funkcijām, tad h ir RAM izrēķināma (Lemma 5.4 un Vingrinājums 5.5).

Tagad balstoties uz primitīvi rekursīvas funkcijas definīciju (Definīcija 4.6) un vispārināto indukcijas metodi, secināms, ka katra primitīvi rekursīva funkcija ir RAM izrēķināma. ■

5.3. RAM izrēķināmas un daļēji rekursīvas funkcijas

5.8. Teorēma. *Katra daļēji rekursīva funkcija ir RAM izrēķināma.*

□ (i) Katra primitīvi rekursīva funkcija ir RAM izrēķināma (Apgalvojums 5.7).

(ii) RAM izrēķināmu funkciju kompozīcija ir RAM izrēķināma funkcija (Lemma 5.3).

(iii) Ja h iegūta ar primitīvi rekursīvās shemas palīdzību no RAM izrēķināmām funkcijām, tad h ir RAM izrēķināma (Lemma 5.4 un Vingrinājums 5.5).

(iv) Ja h iegūta no RAM izrēķināmas funkcijas ar neierobežotā μ operatora palīdzību, tad h ir RAM izrēķināma (Lemma 5.6).

Tagad balstoties uz daļēji rekursīvas funkcijas definīciju (Definīcija 4.54) un vispārināto indukcijas metodi, secināms, ka katra daļēji rekursīva funkcija ir RAM izrēķināma ■

Atlikušajā daļā mēs parādīsim, ka katra RAM izrēķināma funkcija ir daļēji rekursīva.

5.9. Piemēri. (i) Pieņemsim, ka

$$x = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_n^{\alpha_n},$$

kur $p_i \leftharpoondown p(i)$ ($p(i)$ definīciju skatīt Piemērā 4.40 (v)), tad

$$kan(x,y) \leftharpoondown \begin{cases} \alpha_y, & \text{ja} \quad x \neq 0 \neq y \wedge y \leq n, \\ 0, & \quad\quad \text{pretējā gadījumā.} \end{cases}$$

Tā, piemēram,

$$kan(12,y) = \begin{cases} 2, & \text{ja} \quad y = 1, \\ 1, & \text{ja} \quad y = 2, \\ 0, & \text{pārējos gadījumos.} \end{cases}$$

Rezultātā

$$12 = 2^2 \cdot 3 = p(1)^{kan(12,1)} p(2)^{kan(12,2)}.$$

No funkcijas $kan(x,y)$ definīcijas izriet, ka

$$kan(x,y) = \begin{cases} \mu t \leq x (p_y^{t+1} \nmid x), & \text{ja} \quad x \neq 0 \neq y, \\ 0, & \text{ja} \quad x = 0 \vee y = 0. \end{cases}$$

Ja $x \neq 0 \neq y$, tad predikāts $p_y^{t+1} \nmid x$ aizstājams ar nosacījumu (skatīt Piemēru 4.11 (xiv))

$$\text{div}(p_y^{t+1}, x) = 0\,.$$

Tas savukārt ekvivalents ar nosacījumu (skatīt Piemēru 4.11 (iii))

$$\text{div}(h_3(p_y, t+1), x) = 0$$

jeb

$$\text{div}(h_3(p(y), t+1), x) = 0.$$

Tā kā funkcija $\text{div}(h_3(p(y), t+1), x)$ ir primitīvi rekursīva, tad balstoties uz Sekām 4.33 un Teorēmu 4.18 secināms: funkcija $kan(x,y)$ ir primitīvi rekursīva.

(ii) Funkciju $f : \mathbb{N} \to \mathbb{N}$, kas definēta ar rekursiju

$$\begin{cases} f(0) &= 1, \\ f(1) &= 2, \\ f(x+2) &= f(x) + f(x+1), \end{cases}$$

sauc par *Fibonači virkni.*

Vispirms nodemonstrēsim, ka funkcija

$$g(x) \leftrightharpoons 2^{f(x)} 3^{f(x+1)}$$

ir primitīvi rekursīva. No funkcijas $g(x)$ definīcijas izriet, ka

$$\begin{aligned} f(x) &= kan(g(x), 1), \\ f(x+1) &= kan(g(x), 2); \end{aligned}$$

$$\begin{aligned} g(0) &= 2^{f(0)}3^{f(1)} = 2^1 \cdot 3^2 = 18, \\ g(x+1) &= 2^{f(x+1)}3^{f(x+2)} \\ &= 2^{f(x+1)}3^{f(x)+f(x+1)} \\ &= 2^{kan(g(x),2)}3^{kan(g(x),1)+kan(g(x),2)}. \end{aligned}$$

Tātad funkcija $g(x)$ iegūta no konstantes 18 un funkcijas

$$2^{kan(z,2)}3^{kan(z,1)+kan(z,2)}$$

ar primitīvi rekursīvās shēmas palīdzību.

5.10. Vingrinājumi. *Turpmāk pieņemsim, ka $m > 0$ un*

$$\begin{aligned} \hat{y} &\leftharpoondown (y_1, y_2, \ldots, y_m), \\ k\hat{a}n(z) &\leftharpoondown (kan(z,1), kan(z,2), \ldots, kan(z,m)). \end{aligned}$$

(i) Pierādīt, ka funkcija

$$m(t) \leftharpoondown \begin{cases} h_{12}(m,t), & \text{ja} \quad t \not\equiv 0 \mod m, \\ m, & \text{ja} \quad t \equiv 0 \mod m \end{cases}$$

ir primitīvi rekursīva. (Funkcijas h_{12} definīciju skatīt Piemērā 4.11 (xii).)

(ii) Pieņemsim, ka $\xi_i(\hat{y})$, $i \in \overline{1,m}$, ir primitīvi rekursīvas funkcijas. Pierādīt, ka

$$H(\hat{y}, t) \leftharpoondown \xi_i(\hat{y}), \text{ ja } t \equiv i \mod m$$

ir primitīvi rekursīva funkcija.

(iii) Parādīt, ka sekojošās funkcijas ir primitīvi rekursīvas! (Funkcijas h_{13} definīciju skatīt Piemērā 4.11 (xiii).)

$$\begin{aligned} p(z,t) &\leftharpoondown p(m(t))^{kan(z,m(t))}, \\ G_1(z,t) &\leftharpoondown h_{13}(p(z,t), z), \\ G_2(z,t) &\leftharpoondown p(m(t))^{H(\hat{y},t)}, \\ G(z,t) &\leftharpoondown G_1(z,t) \cdot G_2(z,t). \end{aligned}$$

5.11. Teorēma. *Ja funkcijas* $\zeta_i : \mathbb{N}^n \to \mathbb{N}$, $\xi_i : \mathbb{N}^m \to \mathbb{N}$, $i \in \overline{1,m}$, *ir primitīvi rekursīvas, tad funkcijas*

$$\begin{aligned} y_i(\bar{x},0) &\leftharpoondown \zeta_i(\bar{x}), \\ y_i(\bar{x},t+1) &\leftharpoondown \xi_i(y_1(\bar{x},t+1),\ldots,y_{i-1}(\bar{x},t+1),y_i(\bar{x},t),\ldots,y_m(\bar{x},t)) \end{aligned}$$

visiem $i \in \overline{1,m}$ *ir primitīvi rekursīvas.*

□ Tā kā $G(z,t)$ (skatīt Vingrinājumus 5.10) ir primitīvi rekursīva, tad funkcija $F(\bar{x},t)$, kas iegūta ar primitīvi rekursīvās shēmas palīdzību

$$\begin{aligned} F(\bar{x},0) &\leftharpoondown \prod_{j=1}^{m} p_j^{\zeta_j(\bar{x})}, \\ F(\bar{x},t+1) &\leftharpoondown G(F(\bar{x},t),t+1), \end{aligned}$$

ir primitīvi rekursīva.

Pieņemsim, ka

$$z = p_1^{y_1} p_2^{y_2} \ldots p_m^{y_m},$$

kur $\forall j\ p_j \leftharpoondown p(j)$, un $m(t) = i$ (skatīt Vingrinājumus 5.10), tad

$$\begin{aligned} p(z,t) &= p_i^{y_i}, \\ G_1(z,t) &= \prod_{j\neq i} p_i^{y_i}, \\ H(k\hat{a}n(z),t) &= \xi_i(\hat{y}), \\ G_2(z,t) &= p_i^{\xi_i(\hat{y})}, \\ G(z,t) &= p_i^{\xi_i(\hat{y})} \prod_{j\neq i} p_i^{y_i}. \end{aligned}$$

No šejienes

$$y_i(\bar{x},t) = kan(F(\bar{x},mt),i).$$

Tātad arī funkcijas $y_i(\bar{x},t)$ ir primitīvi rekursīvas. ■

5.12. Vingrinājums. *Ja funkcijas* $\zeta_i : \mathbb{N}^n \to \mathbb{N}$, $\xi_i : \mathbb{N}^m \to \mathbb{N}$, $i \in \overline{1,m}$, *ir primitīvi rekursīvas, tad funkcijas*

$$\begin{array}{rcl} y_i(\bar{x}, 0) & \leftharpoondown & \zeta_i(\bar{x}), \\ y_i(\bar{x}, t+1) & \leftharpoondown & \xi_i(y_1(\bar{x}, t), y_2(\bar{x}, t), \dots, y_m(\bar{x}, t)) \end{array}$$

visiem $i \in \overline{1,m}$ *ir primitīvi rekursīvas.*

5.13. Teorēma. *Katra RAM izrēķināma funkcija ir daļēji rekursīva.*

□ Pieņemsim, ka $f(\bar{x})$ ir n argumentu RAM izrēķināma funkcija. Tas nozīmē, ka eksistē tāda programma

$$P = K_1, K_2, \dots, K_\tau$$

ar kuru RAM-rēķina šo funkciju $f(\bar{x})$. Pieņemsim, ka $\varrho_n(P) = m$ (attēlojuma ϱ_n definīciju skatīt 104. lpp.). Mēs definējam funkcijas

$$\mathrm{conf}_i(\hat{y}, s) \leftharpoondown \begin{cases} 0, & \text{ja} \quad K_s = Z(i), \\ y_i + 1, & \text{ja} \quad K_s = S(i), \\ y_k, & \text{ja} \quad K_s = T(k, i), \\ y_i, & \text{pārējos gadījumos.} \end{cases}$$

Tā kā mēs zinām kāda izskatās s–tā komanda, tad

$$\mathrm{conf}_i(\hat{y}, s) = \begin{cases} f_{i1}(\hat{y}), & \text{ja} \quad s = 1, \\ f_{i2}(\hat{y}), & \text{ja} \quad s = 2, \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ f_{i\tau}(\hat{y}), & \text{ja} \quad s = \tau, \\ y_i, & \text{pārējos gadījumos.} \end{cases}$$

Katra funkcija f_{ij} ir primitīvi rekursīva, proti, tā ir kāda no funkcijām $o^m(\hat{y})$ (Piemērs 4.7 (ii)), $u_l^m(\hat{y})$, vai $s \circ u_l^m(\hat{y})$. Katrs predikāts $s = j$ ir primitīvi rekursīvs (Apgalvojums 4.19), tāpēc (Teorema 4.18) $\mathrm{conf}_i(\hat{y}, s)$ ir primitīvi rekursīva funkcija. Saturīgi funkcija $\mathrm{conf}_i(\hat{y}, s)$ modelē i–tās šūnas izmaiņu s–tās komandas iespaidā.

Tagad modelēsim nākošo komandu.

$$\mathrm{nx}(\hat{y}, s) \leftharpoondown \begin{cases} q, & \text{ja} \quad K_s = J(u, v, q) \text{ un } y_u = y_v, \\ s + 1, & \text{ja} \quad s \in \overline{1,\tau} \text{ un } K_s \text{ nav izskatā } J(u, v, q), \\ & \qquad \text{vai arī } y_u \neq y_v, \\ \tau + 1, & \text{pārējos gadījumos.} \end{cases}$$

Pieņemsim, ka programmā P nosacītās vadības maiņas komandas ir

$$\begin{array}{lcl} K_{i_1} & = & J(u_1, v_1, q_1), \\ K_{i_2} & = & J(u_2, v_2, q_2), \\ \cdot & \cdot & \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ K_{i_\varkappa} & = & J(u_\varkappa, v_\varkappa, q_\varkappa), \end{array}$$

tad

$$\mathrm{nx}(\hat{y}, s) \leftharpoondown \begin{cases} q_1, & \text{ja} \quad s = i_1 \wedge y_{u_1} = y_{v_1}, \\ s+1, & \text{ja} \quad s = i_1 \wedge y_{u_1} \neq y_{v_1}, \\ q_2, & \text{ja} \quad s = i_2 \wedge y_{u_2} = y_{v_2}, \\ s+1, & \text{ja} \quad s = i_2 \wedge y_{u_2} \neq y_{v_2}, \\ \cdot \quad \cdot \quad \cdot & \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ q_\varkappa, & \text{ja} \quad s = i_\varkappa \wedge y_{u_\varkappa} = y_{v_\varkappa}, \\ s+1, & \text{ja} \quad s = i_\varkappa \wedge y_{u_\varkappa} \neq y_{v_\varkappa}, \\ s+1, & \text{ja} \quad s \in \overline{1,\tau} \setminus \{i_1, i_2, \ldots, i_\varkappa\}, \\ \tau+1, & \text{pārējos gadījumos.} \end{cases}$$

Pieņemsim, ka

$$\overline{1,\tau} \setminus \{i_1, i_2, \ldots, i_\varkappa\} = \{j_1, j_2, \ldots, j_\sigma\},$$

tad

$$s \in \overline{1,\tau} \setminus \{i_1, i_2, \ldots, i_\varkappa\} \Leftrightarrow s = j_1 \vee s = j_2 \vee \ldots \vee s = j_\sigma.$$

Tagad atsaucoties uz Teoremu 4.18 secināms: $\mathrm{nx}(\hat{y}, s)$ ir primitīvi rekursīva funkcija.

Visbeidzot modelēsim rēķināsanu ar programmu P. Īsāka pieraksta labad ieviesīsim apzīmējumu

$$\widehat{\mathrm{Conf}}(\bar{x}, t) \leftharpoondown (\mathrm{Conf}_1(\bar{x}, t), \mathrm{Conf}_2(\bar{x}, t), \ldots, \mathrm{Conf}_m(\bar{x}, t)).$$

Definējam funkcijas, $i \in \overline{1,m}$,

$$\begin{cases} \mathrm{Conf}_i(\bar{x}, 0) & \leftharpoondown & \begin{cases} x_i & \text{ja} \quad i \leq n, \\ 0 & \text{ja} \quad i > n. \end{cases} \\ \\ \mathrm{Conf}_i(\bar{x}, t+1) & \leftharpoondown & \mathrm{conf}_i(\widehat{\mathrm{Conf}}(\bar{x}, t), \mathrm{Nx}(\bar{x}, t)). \\ \\ \mathrm{Nx}(\bar{x}, 0) & \leftharpoondown & 1, \\ \mathrm{Nx}(\bar{x}, t+1) & \leftharpoondown & \mathrm{nx}(\widehat{\mathrm{Conf}}(\bar{x}, t), \mathrm{Nx}(\bar{x}, t)) \end{cases}$$

Saskaņā ar Vingrinājumu 5.12 visas funkcijas $\mathrm{Conf}_i(\bar{x}, t)$ ir primitīvi rekursīvas, un arī funkcija $\mathrm{Nx}(\bar{x}, t)$ ir primitīvi rekursīva. Saturīgi kortežs $\widehat{\mathrm{Conf}}(\bar{x}, t)$ rāda, kāda izskatīsies konfigurācija pēc t darba soļiem, funkcija $\mathrm{Nx}(\bar{x}, t)$ rāda, kāda būs nākošā komanda pēc t darba soļiem.

Tagad varam nodemonstrēt, ka funkcija $f(\bar{x})$ ir daļēji rekursīva. Vispirms definējam funkciju

$$\mathrm{Stop}(\bar{x}, t) \leftharpoondown \mathrm{Nx}(\bar{x}, t)\ \mathrm{sg}|\tau + 1 - \mathrm{Nx}(\bar{x}, t)|.$$

Saturīgi ja $\mathrm{Stop}(\bar{x}, t_0) = 0$, tad rēķināšana ir beigusies un pirmajā šūnā atrodas funkcijas $f(\bar{x})$ vērtība.

Funkcija

$$T_0(\bar{x}) \leftharpoondown \mu t(\mathrm{Stop}(\bar{x}, t) = 0)$$

ir daļēji rekursīva, tāpēc arī funkcija

$$f(\bar{x}) = \mathrm{Conf}_1(\bar{x}, T_0(\bar{x}))$$

ir daļēji rekursīva. ■

Atzīmēsim, ka latviešu matemātiķi ir nopietni piedalījušies un arī mūsdienās piedalās algoritmu teorijas izpētē. Tā Vilnis Detlovs (ilggadējs Latvijas Universitātes docētājs) pierādīja teorēmu:

5.14. Teorēma. *Funkcija f ir normāli izrēķināma tad un tikai tad, ja tā ir daļēji rekursīva.*

Domāju, ka lasītājs, kas apguvis dotajā konspektā piedāvātos pierādījumus, spēs novērtēt šī rezultāta nozīmīgumu. Šis rezultāts 1953. gadā tika nopublicēts kā īss ziņojums; 1958. gadā tas tika publicēts izvērstā variantā. Tas ir viens no retajiem gadījumiem, kad respektabls žurnāls publicē visu disertāciju bez saīsinājumiem.