VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Algoritmų teorija

Paskaitų konspektas

Dėstytojas: lekt. dr. Adomas Birštunas

Vilnius – 2015

TURINYS

Turinys

1. ALGORITMO SAMPRATA

1 Ap. (Algoritmas)**.** (Intuityvus apibrėžimas.) Veiksmų seka, kuri leidžia spręsti vieną ar kitą uždavinį.

Pagrindinės algoritmo savybės:

- žingsnių elementarumas;
- diskretumas (algoritmas suskirstytas į atskirus žingsnius);
- determinuotumas (atlikus žingsnį aišku, kokį kitą žingsnį reikia atlikti);
- masiškumas (skirtas ne vienam uždaviniui, bet jų klasei spręsti).

Algoritmo formalizavimo būdai:

1. sukurti idealizuotą matematinę mašiną (Turingo, RAM);
2. apibrėžti rekursyvių funkcijų klasę (Dalinai rekursyvios funkcijos, λ-skaičiavimas).

1 T. *(Church tezė) Algoritmiškai apskaičiuojamų funkcijų aibė sutampa su rekursyviųjų funkcijų klase. (Šio teiginio neįmanoma įrodyti.)*

2 T. *Turingo mašinomis apskaičiuojamų funkcijų aibė sutampa su rekursyviųjų funkcijų klase.*

2. HILBERTO TIPO TEIGINIŲ SKAIČIAVIMAS

2 Ap. (Hilberto tipo teiginių skaičiavimas)**.** Skaičiavimas, kuriame yra aksiomų schemos:

1. 1.1. $A \to (B \to A)$

   1.2. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$

2. 2.1. $(A \wedge B) \to A$

   2.2. $(A \wedge B) \to B$

   2.3. $(A \to B) \to ((A \to C) \to (A \to (B \wedge C)))$

3. 3.1. $A \to (A \vee B)$

   3.2. $B \to (A \vee B)$

   3.3. $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$

4. 4.1. $(A \to B) \to (\neg B \to \neg A)$

   4.2. $A \to \neg\neg A$

   4.3. $\neg\neg A \to A$

ir taisyklė *Modus Ponens (MP)*:

$$\begin{array}{r} \text{prielaidos}\{ \\ \text{išvados}\{ \end{array} \frac{A \qquad A \to B}{B} \text{ (MP), čia A, B, C – bet kokios teiginių logikos formulės.}$$

Sakysime, jog formulė F yra įvykdoma Hilberto tipo teiginių skaičiavime, jeigu galima parašyti tokią formulių seką, kurioje kiekviena formulė yra:

- arba aksioma,
- arba gauta pagal *MP* iš ankstesnių

ir kuri (seka) baigiasi formule F.

3 T. *Jei formulė yra įrodoma Hilberto tipo teiginių skaičiavime, tai ji yra tapačiai teisinga ir atvirkščiai (jei nėra įrodoma, tai nėra tapačiai teisinga).*

Sakysime, jog skaičiavimo aksioma yra *nepriklausoma*, jei ją išmetus iš skaičiavimo, ji nėra išvedama jame.

Visos Hilberto teiginių skaičiavimo aksiomos yra nepriklausomos.

2.1. Dedukcijos teorema

Sakysime, kad formulė F yra išvedama iš prielaidų $A_1, A_2, A_3, \ldots, A_n$, jei egzistuoja tokia formulių seka, kurioje kiekviena formulė yra:

- arba aksioma,
- arba prielaida,
- arba gauta pagal *MP* iš ankstesnių

ir kuri (seka) baigiasi formule F.

Žymėjimas: $A_1, A_2, \ldots, A_n \vdash F$

4 T. *(Dedukcijos teorema)* $\Gamma \vdash A \to B \iff \Gamma, A \vdash B$, *kur A,B – bet kokios formulės,* Γ *– baigtinė formulių aibė (gali būti tuščia).*
Proof ▷

Būtinumas

Jei $\Gamma \vdash A \to B$, tada egzistuoja formulių seka:

1.	F_1	
2.	F_2	
	$\ldots$	
i.	F_i	
	$\ldots$	
k.	$F_k = A \to B$	
k+1.	A	prielaida
k+2.	B	*MP* iš k ir (k+1).

kurioje:

$$F_i = \begin{cases} \text{aksioma} \\ \text{prielaida iš } \Gamma \\ \text{gauta pagal } MP \text{ iš ankstesnių} \end{cases}$$

$\Gamma, A \vdash B$.

Pakankamumas

Jei $\Gamma, A \vdash B$, tai yra tokia formulių seka:

1.	F_1
2.	F_2
	$\ldots$
i.	F_i
	$\ldots$
k.	$F_k = B$

kurioje:

$$F_i = \begin{cases} \text{aksioma} \\ \text{prielaida iš } \Gamma \\ \text{prielaida} A \\ \text{gauta pagal } \textit{MP} \text{ iš ankstesnių} \end{cases}$$

Sukonstruokime tokią formulių seką (kuri nebūtinai yra išvedimas):

$$\begin{array}{ll} 1. & A \to F_1 \\ 2. & A \to F_2 \\ & \dots \\ \text{i.} & A \to F_i \\ & \dots \\ \text{k.} & A \to F_k = A \to B \end{array}$$

Dabar kiekvieną $A \to F_i$ keičiam tokiu būdu:

1. Jei F_i – aksioma, tada keičiam į:

$$\begin{array}{lll} \text{i.1} & F_i \to (A \to F_i) & \text{1.1 aksioma} \\ \text{i.2} & F_1 & \text{aksioma} \\ \text{i.3} & A \to F_1 & \textit{MP} \text{ iš (i.1) ir (i.2)} \end{array}$$

2. Jei F_i – prielaida iš Γ, tada keičiam į:

$$\begin{array}{lll} \text{i.1} & F_i \to (A \to F_i) & \text{1.1 aksioma} \\ \text{i.2} & F_1 & \text{prielaida iš } \Gamma \\ \text{i.3} & A \to F_1 & \textit{MP} \text{ iš (i.1) ir (i.2)} \end{array}$$

3. Jei F_i – prielaida A, tada $A \to F_i = A \to A$. Keičiam $A \to A$ išvedimu.
4. Tegu $A \to F_u$ yra visos išvestos, kur $u < i$. F_i buvo gauta iš kažkokių F_r ir F_l (kur $r, l < i$) pritaikius *MP*. Taigi mes jau turime išvestas $A \to F_r$ ir $A \to F_l$. Kadangi $F_l = F_r \to F_i$ (arba $F_r = F_l \to F_i$), tai:

$$\begin{array}{lll} \text{i.1} & (A \to \overbrace{(F_r \to F_i)}^{F_l}) \to ((A \to F_r) \to (A \to F_i)) & \text{1.2 aksioma} \\ \text{i.2} & (A \to F_r) \to (A \to F_i) & \text{pagal } \textit{MP} \\ \text{i.3} & A \to F_i & \text{pagal } \textit{MP} \end{array}$$

$\lhd$

Sakysime, jog skaičiavimas yra *pilnas* formulių aibės A atžvilgiu, jei formulė $\in A$ tada ir tik tada, jei ji yra įrodoma skaičiavimu.

Hilberto tipo teiginių skaičiavimas yra pilnas tapačiai teisingų formulių aibės atžvilgiu.

3. SEKVENCINIS SKAIČIAVIMAS

3 Ap. (Sekvencija). Tokia išraiška $\underbrace{F_1, F_2, \ldots, F_n}_{anticedentas} \vdash \underbrace{G_1, G_2, \ldots, G_m}_{sukcedentas}$, jei $n + m > 0$.

Pastaba. Jei atitinkama sekvencija yra išvedama, tada teiginiai po brūkšnio yra teisingi:

- $\vdash F$ – formulė yra tapačiai teisinga.
- $A_1, A_2, \ldots, A_n \vdash F$ – iš prielaidų seka išvada F.
- $A_1, A_2, \ldots, A_n \vdash B_1, B_2, \ldots, B_m$ – iš prielaidų seka bent viena iš išvadų $B_1, \ldots, B_m$.
- $F \vdash$ – formulė yra tapačiai klaidinga.

4 Ap. (Sekvencinis skaičiavimas G). Skaičiavimas su aksioma $\Gamma', A, \Gamma'' \vdash \Delta', A, \Delta''$ ir taisyklėmis ($\Gamma, \Gamma', \Gamma'', \Delta, \Delta', \Delta''$ – formulių sekos, A, B – formulės):

$(\vdash \neg)$

$$\frac{\Gamma', A, \Gamma'' \vdash \Delta', \Delta''}{\Gamma', \Gamma'' \vdash \Delta', \neg A, \Delta''}$$

$(\neg \vdash)$

$$\frac{\Gamma', \Gamma'' \vdash \Delta', A, \Delta''}{\Gamma', \neg A, \Gamma'' \vdash \Delta', \Delta''}$$

$(\vdash \vee)$

$$\frac{\Gamma', \Gamma'' \vdash \Delta', A, B, \Delta''}{\Gamma', \Gamma'' \vdash \Delta', A \vee B, \Delta''}$$

$(\vee \vdash)$

$$\frac{\Gamma', A, \Gamma'' \vdash \Delta', \Delta'' \qquad \Gamma', B, \Gamma'' \vdash \Delta', \Delta''}{\Gamma', A \vee B, \Gamma'' \vdash \Delta', \Delta''}$$

$(\vdash \wedge)$

$$\frac{\Gamma', \Gamma'' \vdash \Delta', A, \Delta'' \qquad \Gamma', \Gamma'' \vdash \Delta', B, \Delta''}{\Gamma', \Gamma'' \vdash \Delta', A \wedge B, \Delta''}$$

$(\wedge \vdash)$

$$\frac{\Gamma', A, B, \Gamma'' \vdash \Delta', \Delta''}{\Gamma', A \wedge B, \Gamma'' \vdash \Delta', \Delta''}$$

$(\vdash \rightarrow)$

$$\frac{\Gamma', A, \Gamma'' \vdash \Delta', B, \Delta''}{\Gamma', \Gamma'' \vdash \Delta', A \rightarrow B, \Delta''}$$

$(\rightarrow \vdash)$

$$\frac{\Gamma', \Gamma'' \vdash \Delta', A, \Delta'' \qquad \Gamma', B, \Gamma'' \vdash \Delta', \Delta''}{\Gamma', A \rightarrow B, \Gamma'' \vdash \Delta', \Delta''}$$

Sakysime, kad sekvencija $F_1, \ldots, F_n \vdash G_1, \ldots, G_m$ yra išvedame skaičiavime G, jei galime sukonstruoti tokį medį (grafą), kurio visose viršūnėse yra sekvencijos, šaknyje yra pradinė sekvencija $F_1, \ldots, F_n \vdash G_1, \ldots, G_m$ ir kiekvienai viršūnei teisinga: jei viršūnės N, kurioje yra sekvencija S, vaike(-uose) N_1 (ir N_2) yra sekvencija(-os) S_1 (ir S_2), tai sekvencija S yra gauta iš sekvencijos(-jų) S_1 (ir S_2) pagal kažkurią taisyklę, ir kurio (medžio) visuose lapuose yra aksiomos.

Sakysime, kad taisyklė yra *apverčiama*, jei teisinga: jei taisyklės išvada išvedama, tai išvedamos ir visos taisyklės prielaidos.

Visos sekvencinio skaičiavimo G taisyklės yra apverčiamos.

Jei išvedant gautas lapas, kuriame nėra nei vienos formulės su kokia nors operacija ir jis nėra aksioma, tai reiškia, kad:

1. sekvencija yra neišvedama, arba

2. išvedime buvo panaudota neapverčiama taisyklė.

5 T. *Formulė F yra tapačiai teisinga tada ir tik tada, kai sekvenciniame skaičiavime G išvedama sekvencija* $\vdash F$.

4. DISJUNKTŲ DEDUKCINĖ SISTEMA. REZOLIUCIJŲ METODAS

5 Ap. (Disjunktas). Literų disjunkcija. Litera yra kintamasis arba kintamasis su neigimu.

6 Ap. (Atkirtos taisyklė (AT)).

$$\frac{C' \vee p \vee C'' \qquad D' \vee \neg p \vee D''}{C' \vee C'' \vee D' \vee D''}$$, kur p – litera, o C', C'', D', D'' – disjunktai.

7 Ap. ($S \vdash C$). Sakysime, kad disjunktas C yra išvedamas iš disjunktų aibės S, jei galima parašyti tokią disjunktų seką, kur kiekvienas disjunktas:

- arba iš aibės S,
- arba gautas pagal atkirtos taisyklę iš jau parašytų

ir kuri (seka) baigiasi disjunktu C.

8 Ap. (Prieštaringa formulių aibė). Sakysime, kad formulių aibė S yra prieštaringa, jei su *bet kokia* interpretacija ν *bent viena* formulė iš aibės S yra klaidinga.

Pastaba. Jei S nėra prieštaringa, tai S – įvykdoma.

6 T. *Jei iš disjunktų aibės S išvedamas disjunktas C ir disjunktas C nėra įvykdomas (tapačiai klaidingas), tai S – prieštaringa.*

Proof $\triangleright$
(Prieštaros būdu.)

Tarkime $S \vdash C$ ir C – nėra įvykdomas, bet aibė S nėra prieštaringa.

Jei aibė S nėra prieštaringa, tai yra tokia interpretacija ν, su kuria visi aibės S disjunktai bus teisingi:

$$\forall D\,(D \in S) : \nu(D) = 1$$

Parodysime, jog visi iš aibės S išvedami disjunktai yra teisingi su ta pačia interpretacija ν.

Įrodymas matematinės indukcijos metodu, pagal išvedimo ilgį l:

1. Jei $l = 1$, tai $D_1 \in S \implies \nu(D_1) = 1$, nes visi aibės S disjunktai yra teisingi su interpretacija ν.

2. Tarkime, kad $\forall i\,(i < m),\ \nu(D_i) = 1$.

3. Panagrinėkime $\nu(D_m)$:

   - Jei $D_m \in S$, tai $\nu(D_m) = 1$.

- Jei $D_m \notin S$, tai D_m gautas iš kažkokių D_a ir D_b pagal atkirtos taisyklę. Pagal matematinės indukcijos prielaidą $\nu(D_a) = \nu(D_b) = 1$, nes $a, b < m$. Kadangi

$$\frac{D_a \qquad D_b}{D_m}\ \text{(AT)}$$

tai $D_a = D'_a \vee p$ ir $D_b = D'_b \vee \neg p$, o $D_m = D'_a \vee D'_b$.
  - Jei $\nu(p) = 1$, tai $\nu(D'_b) = 1$(, nes $\nu(D'_b \vee \neg p) = 1$) $\implies \nu(D_m) = \nu(D'_b \vee D'_a) = 1$.
  - Jei $\nu(\neg p) = 1$, tai $\nu(D'_a) = 1$(, nes $\nu(D'_a \vee p) = 1$) $\implies \nu(D_m) = \nu(D'_b \vee D'_a) = 1$.

Naudodami matematinę indukciją gavome, kad visi išvedami disjunktai yra teisingi su interpretacija ν. Tada ir $\nu(C) = 1 \implies$ prieštara, nes disjunktas C nėra įvykdomas.

$\triangleleft$

Tuščias disjunktas nėra įvykdomas (yra tapačiai klaidingas).

Žymėjimas: $\square$ – tuščias disjunktas.

Jei iš aibės $S \vdash \square$, tai aibė S – prieštaringa.

7 T. *Jei disjunktų aibė S yra prieštaringa, tai iš jos galima išvesti tuščią disjunktą $S \vdash \square$.*

Proof $\triangleright$
(Matematinės indukcijos būdu pagal skirtingų kintamųjų kiekį n aibėje S.)

Bazė Jei $n = 1$, tai aibė S gali būti: $\underbrace{\{p\},\ \{\neg p\},}_{\text{nėra prieštaringos}}\ \{p, \neg p\}$.

$S = \{p, \neg p\}$ – prieštaringa aibė:

1. p (iš S)
2. $\neg p$ (iš S)
3. $\square$ (pagal AT iš 1 ir 2)

Jei aibėje S yra vienas kintamasis, tai jai (aibei S) teisinga: jei S – prieštaringa, tai iš jos išvedama $\square$.

Prielaida Tarkime, jei aibėje yra $n < m$ skirtingų kintamųjų, tai jei S – prieštaringa, tai $S \vdash \square$.

Indukcinis žingsnis Tegu disjunktų aibė S turi $n = m$ skirtingų kintamųjų.

Padalinkime aibės S disjunktus į 3 aibes (grupes):

S_p priklauso tie aibės S disjunktai, kurie neturi literos p.

S_p^+ priklauso tie aibės S disjunktai, kurie turi literą p (be neigimo).

S_p^- priklauso tie aibės S disjunktai, kurie turi literą $\neg p$ (su neigimu).

Tada: $S = S_p \cup S_p^+ \cup S_p^-$.

Nagrinėkime aibę $S' = S_p \cup \mathrm{at}(S_p^+, S_p^-)$, kur $\mathrm{at}(S_p^+, S_p^-)$ – yra disjunktų aibė, kuri yra gauta pritaikius AT visiems disjunktams iš aibių S_p^+ ir S_p^- pagal kintamąjį p. Tegu:

$$\begin{aligned} S_p^+ &= \{C_1 \vee p, C_2 \vee p, \ldots, C_v \vee p\} \qquad \text{ir} \\ S_p^- &= \{D_1 \vee \neg p, D_2 \vee \neg p, \ldots, D_r \vee \neg p\} \end{aligned}$$

tada

$$\begin{aligned} \mathrm{at}(S_p^+, S_p^-) = \{&C_1 \vee D_1, C_1 \vee D_2, \ldots, C_1 \vee D_r \\ &C_2 \vee D_1, C_2 \vee D_2, \ldots, C_2 \vee D_r \\ &\ldots \\ &C_v \vee D_1, C_v \vee D_2, \ldots, C_v \vee D_r\} \end{aligned}$$

Aibėje S' yra $< m$ kintamųjų (nes nėra p), todėl aibei S' galioja prielaida: jei S' – prieštaringa $\implies S' \vdash \square$. Parodysime, jog S ir S' arba abi kartu įvykdomos, arba neįvykdomos:

1. Jei S – įvykdoma, tai $\exists \nu : \nu(D) = 1, \forall D \, (D \in S)$. Taip pat ir visi disjunktai iš aibės S_p yra įvykdomi su ta pačia interpretacija ν, nes $S_p \subset S$. Taip pat visi disjunktai iš aibės $\mathrm{at}(S_p^+, S_p^-)$ yra įvykdomi su interpretacija ν, nes jie yra išvesti iš aibės S pritaikius AT (žr. 6 teiginio įrodymą). Taigi su ν teisingi visi aibės S' disjunktai, tai yra S' – įvykdoma.

2. Jei S' – įvykdoma, tai yra interpretacija ν, su kuria visi aibės S' disjunktai yra teisingi. Kadangi aibėje S' nėra kintamojo p, tai interpretacija ν jam nepriskiria nei 0, nei 1. Papildome interpretaciją ν apibrėždami ar kintamasis p yra teisingas ar klaidingas, taip kad su interpretacija ν visi aibės S disjunktai būtų teisingi.

   2.1. Jei yra toks $i \, (i = 1, 2, \ldots, v)$, su kuriuo $\nu(C_i) = 0$, tada apibrėžiame $\nu(p) = 1$, tada $\nu(C_1 \vee p) = \nu(C_2 \vee p) = \cdots = \nu(C_v \vee p) = 1$
   Kadangi visi aibės $\mathrm{at}(S_p^+, S_p^-)$ disjunktai yra teisingi su interpretacija ν, tai $\nu(C_i \vee D_1) = \cdots = \nu(C_i \vee D_r) = 1$, bet $\nu(C_i) = 0$, todėl $\nu(D_1) = \nu(D_2) = \cdots = \nu(D_r) = 1 \implies \nu(D_1 \vee \neg p) = \cdots = \nu(D_r \vee \neg p) = 1$
   Gavome, kad su ν visi S_p^+, S_p^- ir S_p yra teisingi $\implies S$ – įvykdoma.

   2.2. Jei nėra tokio $i \, (i = 1, 2, \ldots, v)$, su kuriuo $\nu(C_i) = 0$, tada $\nu(C_1) = \nu(C_2) = \cdots = \nu(C_v) = 1$, tada ir $\nu(C_1 \vee p) = \nu(C_2 \vee p) = \cdots = \nu(C_v \vee p) = 1$.
   Apibrėžkime $\nu(p) = 0$, tada $\nu(D_1 \vee \neg p) = \nu(D_2 \vee \neg p) = \cdots = \nu(D_r \vee \neg p) = 1$.
   Gavome, kad su ν visi S_p^+, S_p^- ir S_p yra teisingi $\implies S$ – įvykdoma.

   Gavome, kad jeigu S' įvykdoma, tai ir S yra įvykdoma.

Aibėje S' yra $< m$ kintamųjų, todėl jei S' prieštaringa, tai $S' \vdash \square$ (prielaida). S' – prieštaringa tada ir tik tada, kai S – prieštaringa, todėl jei S – prieštaringa, tai iš S galima išvesti $\square$.

Išvada: $S \vdash \square \iff S$ – prieštaringa.

$\lhd$

4.1. Rezoliucijų metodas

Jei norime patikrinti ar iš prielaidų $A_1, A_2, \ldots, A_n$ seka išvada F:

$$
\begin{aligned}
A_1, A_2, \ldots, A_n \vdash F &\iff \text{aibė } B = \{A_1, A_2, \ldots, A_n, \neg F\} \text{ – prieštaringa} \\
&\iff \text{disjunktų aibė (gauta paverčiant B aibės elementus į NKF)} \\
&\quad\; S = \{D_1, D_2, \ldots, D_m\} \text{ – prieštaringa} \\
&\iff S \vdash \square \text{ pagal disjunktų dedukcinę sistemą.}
\end{aligned}
$$

5. TURINGO MAŠINOS IR JŲ VARIANTAI

9 Ap. (Determinuota vienajuostė Turingo mašina). Ketvertas $< \Sigma, Q, F, \delta >$, kur:

Σ – baigtinė aibė – abėcėlė. (Jei nepaminėta kitaip: $\Sigma = \{0, 1, b\}$.)

Q – baigtinė aibė – būsenų aibė. ($Q = \{q_0, q_1, \ldots, q_n\}$, kur q_0 – pradinė būsena.)

F – galutinių būsenų aibė ($F \subset Q$).

δ – perėjimų funkcija. ($\delta : Q \times \Sigma \to Q \times \Sigma \times \{K, N, D\}$)

10 Ap. (Apibrėžta *TM*). Sakysime, jog *TM* su pradiniais duomenimis X yra apibrėžta, jei pradžioje į duomenų juostą įrašius žodį X, *TM* po baigtinio žingsnių kiekio patenka į vieną iš galutinių būsenų.

11 Ap. (Turingo mašina apskaičiuoja funkciją). Sakysime, kad *TM* M apskaičiuoja funkciją $f(x_1, x_2, \ldots, x_n)$, jei egzistuoja toks kodavimas abėcėlės Σ simboliais $\text{cod}(x_1, x_2, \ldots, x_n) = \tilde{x}$, kuriam teisinga:

1. jei $f(x_1, x_2, \ldots, x_n) = y$ (funkcija apibrėžta), tada *TM* M su pradiniais duomenimis $\text{cod}(x_1, x_2, \ldots, x_n)$ yra apibrėžta ir baigus darbą, į dešinę nuo skaitymo galvutės yra žodis $\text{cod}(y)$;
2. jei $f(x_1, x_2, \ldots, x_n)$ – neapibrėžta, tada *TM* M su pradiniais duomenimis $\text{cod}(x_1, x_2, \ldots, x_n)$ irgi yra neapibrėžta.

12 Ap. (Determinuota m-juostė *TM*). Ketvertas $< \Sigma, Q, F, \delta >$, kur:

Σ – baigtinė aibė – abėcėlė.

Q – baigtinė būsenų aibė.

F – galutinių būsenų aibė. ($F \subset Q$)

δ – perėjimų funkcija:

$$\delta : Q \times \underbrace{\Sigma \times \Sigma \times \cdots \times \Sigma}_{m} \to$$
$$Q \times \underbrace{\Sigma \times \Sigma \times \cdots \times \Sigma}_{m} \times \underbrace{\{K, D, N\} \times \{K, D, N\} \times \cdots \times \{K, D, N\}}_{m}$$

13 Ap. (Nedeterminuota Turingo mašina). Turingo mašina, kurios perėjimų funkcija yra daugiareikšmė.

5.1. Baigtiniai automatai

14 Ap. (Baigtinis automatas). Vienajuostė determinuota Turiningo mašina, kurios perėjimų funkcija yra $\delta(q_i, a) = (q_j, a, D)$ ir kuri (*TM*) baigia darbą tada, kai pasiekia pirmąją tuščią ląstelę.

15 Ap. (Baigtinio automato kalba). Aibė žodžių, su kuriais automatas baigia darbą vienoje iš galutinių būsenų.

8 T. *Baigtinė aibė yra baigtinio automato kalba.*

9 T. *Baigtinio automato kalbos A papildinys $\bar{A}$ irgi yra baigtinio automato kalba.*

10 T. *Jei A_1 ir A_2 yra baigtinio automato kalbos, tai ir $A_1 \cup A_2$, bei $A_1 \cap A_2$ yra baigtinio automato kalbos.*

Proof ▷
Grafų G_1 ir G_2 Dekarto sandauga yra grafas, kurio viršūnės yra visos įmanomos poros (q_i, q_j), kur q_i yra G_1 viršūnė, o q_j – G_2 viršūnė. Iš viršūnės (q_i, q_j) eis briauna į viršūnę (q_k, q_l) su simboliu a, jei grafe G_1 eina briauna iš viršūnės q_i į viršūnę q_k su simboliu a ir grafe G_2 eina briauna iš viršūnės q_j į viršūnę q_l su simboliu a.

$F_{A \cup B}$ – visos viršūnės (q_i, q_j), kur $q_i \in F_A$ arba $q_j \in F_B$.

$F_{A \cap B}$ – visos viršūnės (q_i, q_j), kur $q_i \in F_A$ ir $q_j \in F_B$.

Baigtinis automatas, kurio perėjimų funkcija vaizduojama grafu $G_1 \times G_2$, pradinė būsena yra (q_0, q_0) ir kurio galutinių būsenų aibė yra $F_{A \cup B}$, turės kalbą $A_1 \cup A_2$. Atitinkamai, baigtinis automatas, kurio galutinių būsenų aibė yra $F_{A \cap B}$, turės kalbą $A_1 \cap A_2$.

◁

11 T. *Baigtinių automatų kalbų*

1. *konkatenacija* $:= \{uv : u \in A_1, v \in A_2\}$*, kur* A_1*,* A_2 *– automato kalbos;*

2. *iteracija* $:= \{u_1 u_2 \dots u_k : u_i \in A, i = 1, 2, \dots, k\}$*, kur* A *– automato kalba;*

3. *atspindys* $:= \{a_1 a_2 \dots a_n : a_n a_{n-1} \dots a_1 \in A\}$*, kur* A *– automato kalba*

irgi yra baigtinių automatų kalbos.

5.2. Algoritmų sudėtingumas

Žymėjimas:

$i(v)$ – žodžio v ilgis;

$t(v)$ – žingsnių kiekis, kurį atlieka *TM*, jei pradinių duomenų juostoje yra žodis v.

$s(v)$ – panaudotų ląstelių kiekis, kurį sunaudojo *TM*, jei pradinių duomenų juostoje buvo žodis v.

16 Ap. (Turingo mašinos M sudėtingumas laiko atžvilgiu). Funkcija: $T_M(n) = \max\{t(v) : i(v) = n\}$.

17 Ap. (Turingo mašinos M sudėtingumas atminties atžvilgiu). Funkcija: $S_M(n) = \max\{s(v) : i(v) = n\}$.

18 Ap. (Turingo mašinos kalba). Žodžių, su kuriais *TM* baigia darbą galutinėje būsenoje, aibė.

19 Ap. (Problema (aibė) išsprendžiama su *TM*). Sakysime, jog problema (aibė) A yra išsprendžiama su Turingo mašina, jei yra tokia Turingo mašina, kurios kalba yra A.

Sakysime, kad *TM* sudėtingumas atminties atžvilgiu yra $f(n)$, jei $\exists c\,(c > 0) : S_M(n) = cf(n), \forall n$.

Sakysime, kad *TM* sudėtingumas laiko atžvilgiu yra $f(n)$, jei $\exists c\,(c > 0) : T_M(n) = cf(n), \forall n$.

20 Ap. (Sudėtingumo klasės).

- $DTIME(f(n))$ – sudėtingumo klasė, kuriai priklauso visi uždaviniai, kuriems egzistuoja juos sprendžianti daugiajuostė determinuota *TM*, kurios sudėtingumas laiko atžvilgiu yra $f(n)$.
- $NTIME(f(n))$ – sudėtingumo klasė, kuriai priklauso visi uždaviniai, kuriems egzisutoja juos sprendžianti daugiajuostė nedeterminuota *TM*, kurios sudėtingumas laiko atžvilgiu yra $f(n)$.
- $DSPACE(f(n))$ – sudėtingumo klasė, kuriai priklauso visi uždaviniai, kuriems egzistuoja juos sprendžianti daugiajuostė determinuota *TM*, kurios sudėtingumas atminties atžvilgiu yra $f(n)$.
- $NSPACE(f(n))$ – sudėtingumo klasė, kuriai priklauso visi uždaviniai, kuriems egzsituoja juos sprendžianti daugiajuostė nedeterminuota *TM*, kurios sudėtingumas atminties atžvilgiu yra $f(n)$.

21 Ap. (Sudėtingumo klasės).

- $L = DSPACE(\log n)$
- $NL = NSPACE(\log n)$
- $P = DTIME(n^k)$, kur k – konstanta.
- $NP = NTIME(n^k)$, kur k – konstanta.
- $PSPACE = DSPACE(n^k)$, kur k – konstanta.
- $EXP = DTIME(2^{n^k})$, kur k – konstanta.

12 T. $L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq EXP$ *ir* $NL \neq PSPACE$ *ir* $P \neq EXP$.

5.3. Porų numeravimas

$$\overbrace{\underbrace{(0;0)}_{0}}^{\Sigma=0}, \overbrace{\underbrace{(0;1)}_{1}, \underbrace{(1;0)}_{2}}^{\Sigma=1}, \overbrace{\underbrace{(0;2)}_{3}, \underbrace{(1;1)}_{4}, \underbrace{(2;0)}_{5}}^{\Sigma=2}, \underbrace{(0;3)}_{6}, \underbrace{(1;2)}_{7}, \ldots$$

22 Ap. (Poros numeris *Kantaro* numeracijoje). Funkcija $\alpha_2(x, y)$. Kairiojo nario funkcija yra $\pi_2^1(n)$, o dešiniojo $\pi_2^2(n)$, kur n yra porai priskirtas numeris.

13 T. *Poros* $(x; y)$ *numeris:*

$$\alpha_2(x, y) = \frac{(x+y)^2 + 3x + y}{2}$$

Pastaba.

$$\pi_2^1(n) = n \dot{-} \frac{1}{2}\left[\frac{[\sqrt{8n+1}] + 1}{2}\right]\left[[\sqrt{8n+1}] - 1\right]$$

23 Ap. (Kantaro funkcijos). $\alpha_n(x_1, x_2, \ldots, x_n)$ – rinkinio $x_1, x_2, \ldots, x_n$ numeris Kantaro numeracijoje. $\pi_n^i(K)$ – K-ojo rinkinio iš n elementų i-asis narys.

Kantaro funkcijos rekurentinė išraiška:

$$\begin{cases} \alpha_2(x, y) = \frac{(x+y)^2+3y+x}{2} \\ \alpha_n(x_1, x_2, \ldots, x_n) = \alpha_2(x_1, \alpha_{n-1}(x_2, x_3, \ldots, x_n)) \end{cases}$$

Pastaba.

```
>>> def alpha(*numbers):
...     if len(numbers) == 2:
...         x, y = numbers
...         return ((x+y)**2 + 3*x + y)/2
...     elif len(numbers) > 2:
...         return alpha(numbers[0], alpha(*numbers[1:]))
...     else:
...         raise Exception('Netinkamas argumentų kiekis!')
>>> alpha(1,0,1,2)
436.0
```

1 Pvz. Šis pavyzdys iliustruoja kaip į kiekvieną n-argumentų funkciją galima žiūrėti kaip į 1-argumento funkciją, kur argumentas yra atitinkamo rinkinio numeris. Šiuo atveju $f(x, y)$ yra 2 argumentų funkcija, o ją atitinkanti vieno argumento funkcija yra $g(z)$ (iš esmės abi jos skaičiuoja tą patį).

$$\begin{aligned} f(x, y) &= 3x + y \\ g(z) &= 3\pi_2^1(z) + \pi_2^2(z) \\ f(x, y) &= g(\alpha_2(x, y)) = 3x + y \end{aligned}$$

5.4. Baigtinumo problema

24 Ap. (Standartinė *TM*)**.** Tokia *TM*, kuri:

1. vienajuostė determinuota;
2. $\Sigma = \{0, 1, b\}$;
3. kai baigia darbą būdama galutinėje būsenoje, juostoje yra tik atsakymas, ir skaitymo galvutė žiūri į pirmąjį iš kairės netuščią simbolį;
4. $|F| = 1$.

14 T. *Kiekvienai* TM *egzistuoja standartinė* TM*, kuri skaičiuoja tą pačią funkciją.*

15 T. *Standartinių* TM *aibė yra skaiti.*

Pastaba. Visas standartines *TM* galima sunumeruoti:

$$\begin{array}{rcccccc} \text{Turingo mašina:} & T_0, & T_1, & T_2, & T_3, & T_4, & \dots \\ \text{Jos skaičiuojama funkcija:} & \varphi_0(x), & \varphi_1(x), & \varphi_2(x), & \varphi_3(x), & \varphi_4(x), & \dots \end{array}$$

25 Ap. (Baigtinumo problema)**.** Ar yra toks algoritmas, kuris ℕ skaičių porai (m, n) pasakytų, ar *TM* su numeriu m (T_m) ir pradiniais duomenimis n baigia darbą, ar ne.

16 T. *Baigtinumo problema neišsprendžiama.*
Proof ▷
(Prieštaros būdu.)

Tarkime, jog egzistuoja toks algoritmas, kuris porai $(m; n)$ pasako, ar *TM* T_m su duomenimis n yra apibrėžta, ar ne. Tada egzistuoja tokia algoritmiškai apskaičiuojama funkcija:

$$g(\alpha_2(x, y)) = \begin{cases} 1, & \text{jei } \varphi_x(y) < \infty \text{ (apibrėžta),} \\ 0, & \text{jei } \varphi_x(y) = \infty \text{ (neapibrėžta).} \end{cases}$$

Tada yra *TM*, kuri apskaičiuoja funkciją $g(z)$. Tada yra *TM*, kuri apskaičiuoja funkciją:

$$f(x) = \begin{cases} 1, & \text{jei } g(\alpha_2(x, x)) = 0, \\ \infty, & \text{jei } g(\alpha_2(x, x)) = 1. \end{cases}$$

TM skaičiuojančią $g(z)$ pažymėkime M_1.

$$\begin{aligned} M_1 : Q_1 &- \text{būsenų aibė;} \\ F_1 &- \text{galutinių būsenų aibė.} \end{aligned}$$

Sukonstruokime *TM* M_2 skaičiuojančią $f(x)$:

Kai M_1 baigia darbą, tai galimi du variantai:

1. *TM* yra galutinėje būsenoje $q_F \in F_1$ ir juostoje yra vienintelis simbolis 1, į kurį ir žiūri skaitymo galvutė.

2. *TM* yra galutinėje būsenoje $q_F \in F_1$ ir juostoje yra vienintelis simbolis 0, į kurį ir žiūri skaitymo galvutė.

TM M_2 tokia pati, kaip ir M_1, tik $Q_2 = Q_1 \cup \{q^*\}$, $F_2 = \{q^*\}$, q_F – nėra galutinė būsena ir:

$$\delta(q_F, 0) = (q^*, 1, N) \to f(x) \text{ – apibrėžta;}$$
$$\delta(q_F, 1) = (q_F, 1, N) \to f(x) \text{ – neapibrėžta.}$$

Kadangi yra *TM* (M_2), kuri apskaičiuoja funkciją $f(x)$, tai yra toks numeris l, kad $T_l = M_2$. T_l apskaičiuoja $\varphi_l(x)$, todėl $f(x) = \varphi_l(x)$. Panagrinėkime $\varphi_l(l)$:

1. Jei $\varphi_l(l) < \infty$, tai $g(\alpha_2(l, l)) = 1 \implies f(l) = \infty \implies \varphi_l(l) = \infty$;

2. Jei $\varphi_l(l) = \infty$, tai $g(\alpha_2(l, l)) = 0 \implies f(l) = 1 \implies \varphi_l(l) < \infty$.

Gauname prieštarą, todėl neegzistuoja toks algoritmas, kuris porai $(m; n)$ pasakytų, ar *TM* T_m su pradiniais duomenis n baigia darbą, ar ne.

$\triangleleft$

26 Ap. (Aibės charakteringoji funkcija). Aibės A charakteringoji funkcija yra:

$$\chi_A(x) = \begin{cases} 1, & \text{jei } x \in A \\ 0, & \text{jei } x \notin A \end{cases}$$

27 Ap. (Rekursyvi aibė). Sakysime, kad aibė yra rekursyvi, jei jos charakteringoji funkcija yra visur apibrėžta rekursyvi funkcija.

Pastaba. (Rice teorema.) Jei aibė X yra vieno argumento dalinai rekursyvių funkcijų aibė ir nesutampa nei su $\varnothing$, nei su visa vieno argumento dalinai rekursyvių funkcijų aibe (yra poaibis), tai aibė $A = \{x : \varphi_x \in X\}$ nėra rekursyvi.

2 Pvz. X – funkcijų, kurios visada grąžina 1 aibė: $X = \{f(x) = 1\}$. Tada $A = \{x : \varphi_x \in X\}$ nėra rekursyvi $\implies$ nėra algoritmo, kuris pasakytu ar $x \in A$.

6. λ-SKAIČIAVIMAS

28 Ap. (λ–skaičiavimo termas)**.**

- Jei u – kintamasis, tai u yra termas.
- Jei E_1 ir E_2 yra termai, tai ir (E_1E_2) yra termas.
- Jei E yra termas, o x – kintamasis, tai ir $\lambda x.E$ irgi yra termas.

Pastaba. Kintamuosius žymime mažosiomis raidėmis.

Pastaba. Skliaustai būtini: $(xy)z \neq x(yz)$!

Kintamojo įeitis terme yra jo pasitaikymas jame.

3 Pvz. Jei turime termą:

$$(\lambda x.((yz)(\lambda z.((zx)y)))(x(\lambda y.(xy)))),$$

tai:

$$(\lambda \underbrace{x}_{1}.((yz)(\lambda z.((z\underbrace{x}_{2})y)))(\underbrace{x}_{3}(\lambda y.(\underbrace{x}_{4}y)))),$$

sunumeruotos yra kintamojo x įeitys.

Kintamojo įeitis x yra suvaržyta, jei patenka į $\lambda x.$ veikimo sritį. Kitaip x įeitis yra laisva.

Pastaba. $\lambda x.E$ terme $\lambda x.$ veikimo sritis yra termas E.

Pastaba. $\lambda x.xu = (\lambda x.x)u \neq \lambda x.(xu)$

Termai E_1 ir E_2 yra α-ekvivalentūs, jei E_2 yra gautas iš E_1, jame visas laisvas kažkurio kintamojo x įeitis pakeitus nauju kintamuoju.

Termas $\lambda x.E_1$ yra α-ekvivalentus termui $\lambda y.E_2$, jei termas E_2 yra gautas iš termo E_1, jame visas laisvas x įeitis pakeitus nauju kintamuoju y.

29 Ap. (Redeksas ir jo santrauka)**.** Termas pavidalo $(\lambda x.E)Y$, kur E ir Y yra termai, vadinamas redeksu. Redekso $(\lambda x.E)Y$ santrauka yra termas $E[{}^{Y}/_{x}]$ – termas E, kuriame visos laisvos kintamojo x įeitys yra pakeistos termu Y.

4 Pvz. Redekso $(\lambda x.\underbrace{(ux))}_{E}\underbrace{(zz)}_{Y}$ santrauka yra termas $u(zz)$.

5 Pvz. Redekso $(\lambda x.((ux)(\lambda x.(xy))))(zz)$ santrauka yra termas $(u(zz))(\lambda x.(xy))$.

30 Ap. (β-redukcija)**.** λ-skaičiavimo termo E β-redukcija vadinama termų seka $E_1 \triangleright E_2 \triangleright E_3 \triangleright E_4 \triangleright \cdots \triangleright E_k \triangleright$, kur $E_1 = E$, ir E_{i+1} yra gautas iš E_i jame pirmąjį iš kairės redeksą pakeitus jo santrauka.

31 Ap. (Normalinis termas). Termas, kuriame nėra redeksų. Termas yra nenormalizuojamas, jeigu jo β-redukcija yra begalinė.

32 Ap.. λ-skaičiavimo loginės konstantos yra $\underbrace{\lambda x.\lambda y.y}_{klaidinga}$ ir $\underbrace{\lambda x.\lambda y.x}_{teisinga}$.

Žymėjimas: $\lambda x.\lambda y.y = 0$ ir $\lambda x.\lambda y.x = 1$.

33 Ap.. λ-skaičiavimo natūrinis skaičius k yra $\lambda f.\lambda x.(f^k x)$, kur $f^k x = \underbrace{f(f(\ldots f(f}_{\text{k kartų}} x)\ldots))$.

Žymėjimas: Skaičius 2 žymimas $\underline{2} = \lambda f.\lambda x(f^2 x) = \lambda f.\lambda x(f(fx))$

6 Pvz. $(x1)\underline{1} = (x(\lambda x.\lambda y.x))(\lambda f.\lambda x.(fx))$

34 Ap.. Sakysime, kad termas E apibrėžia dalinę funkciją $f(x_1, x_2, \ldots, x_n)$, jei:

- jei $f(K_1, K_2, \ldots, K_n) = K$, tai termas $(\ldots((EK_1)K_2)\ldots)K_n$ redukuojamas (β-redukcijoje) į termą K;
- jei $f(K_1, K_2, \ldots, K_n) = \infty$, tai termas $(\ldots((EK_1)K_2)\ldots)K_n$ yra neredukuojamas.

17 T. *Kiekvienai algoritmiškai apskaičiuojamai funkcijai egzistuoja ją apibrėžiantis termas.*

7. PRIMITYVIAI REKURSYVIOS FUNKCIJOS

Kompozicijos operatorius: sakysime, kad funkcija $f(x_1, x_2, \ldots, x_n)$ yra gauta iš funkcijų $g_i(x_1, x_2, \ldots, x_n)\,(i = 1, 2, 3, \ldots, m)$ ir funkcijos $h(x_1, x_2, \ldots, x_m)$, jei $f(x_1, x_2, \ldots, x_n) = h(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$.

35 Ap.. Sakysime, jog funkcija $f(x_1, x_2, \ldots, x_n)$ yra gauta pagal primityviosios rekursijos operatorių iš $g(x_1, x_2, \ldots, x_{n-1})$ ir $h(x_1, x_2, \ldots, x_n, x_{n+1})$, jei teisinga:

$$f(x_1, x_2, \ldots, x_{n-1}, 0) = g(x_1, x_2, \ldots, x_{n-1}) \text{ ir}$$
$$f(x_1, x_2, \ldots, x_{n-1}, y+1) = h(x_1, x_2, \ldots, x_{n-1}, y, f(x_1, x_2, \ldots, x_{n-1}, y))$$

36 Ap. (Primityviai rekursyvių funkcijų aibė (*PR*)). Primityviai rekursyvių funkcijų aibė sutampa su aibe, kuriai priklauso bazinės funkcijos: $0, s(x) = x+1, pr^i_m(x_1, x_2, \ldots, x_m) = x_i$ ir kuri yra uždara kompozicijos ir primityviosios rekursijos operatorių atžvilgiu.

Visos *PR* funkcijos yra apibrėžtos su visais natūraliaisiais skaičiais.

Sakysime, kad $\mathbb{N}$ skaičių poaibis yra primityviai rekursyvus, jei jo charakteringoji funkcija priklauso *PR*.

18 T. *Kantaro funkcijos $\alpha_n(x_1, x_2, \ldots, x_n)$ ir $\pi^i_n(K)$ yra primityviai rekursyvios.*

19 T. *Jei funkcija $g(x_1, x_2, \ldots, x_n) \in$ PR, tai ir funkcija $f(x_1, x_2, \ldots, x_n) = \sum_{i=0}^{x_n} g(x_1, x_2, \ldots, x_n) \in$ PR*

20 T. *Jei $f_i(x_1, x_2, \ldots, x_n) \in$ PR ir $\alpha_j(x_1, x_2, \ldots, x_n) \in$ PR $(i = 1, 2, \ldots, s, s{+}1;\ j = 1, 2, \ldots, s)$, tada ir funkcija:*

$$h(x_1, x_2, \ldots, x_n) = \begin{cases} f_1(x_1, x_2, \ldots, x_n), & \textit{jei } \alpha_1(x_1, x_2, \ldots, x_n) = 0, \\ f_2(x_1, x_2, \ldots, x_n), & \textit{jei } \alpha_2(x_1, x_2, \ldots, x_n) = 0, \\ \ldots & \\ f_s(x_1, x_2, \ldots, x_n), & \textit{jei } \alpha_s(x_1, x_2, \ldots, x_n) = 0, \\ f_{s+1}(x_1, x_2, \ldots, x_n), & \textit{kitu atveju} \end{cases}$$

yra primityviai rekursyvi.

Pastaba. Su bet kuriuo rinkiniu $(x_1, x_2, \ldots, x_n)$ ne daugiau nei viena $\alpha_i(x_1, x_2, \ldots, x_n)$ $(i = 1, 2, \ldots, s)$ gali būti lygi 0.

37 Ap. (Iteracijos operatorius). Sakysime, kad funkcija $f(x)$ yra gauta pagal iteracijos operatorių iš funkcijos $g(x)$, jei teisinga:

$$\begin{cases} f(0) & = 0 \\ f(y+1) & = g(f(y)) \end{cases}$$

21 T. *Visų vieno argumento PR funkcijų aibė sutampa su aibe, kuriai priklauso funkcijos* $s(x) = x+1$, $q(x) = x \dot{-} [\sqrt{x}]^2$ *ir kuri yra uždara kompozicijos, sudėties ir iteracijos operatorių atžvilgiu.*

7.1. Dalinai rekursyvios funkcijos

38 Ap. (Minimizacijos operatorius). Sakysime, kad funkcija $f(x_1, x_2, \ldots, x_n)$ gauta pagal minimizacijos operatorių iš funkcijos $g(x_1, x_2, \ldots, x_n)$, jei:

$$f(x_1, x_2, \ldots, x_n) = \mu_y(g(x_1, x_2, \ldots, x_{n-1}, y) = x_n),$$

tai yra pati mažiausia natūrali y reikšmė, kuriai teisinga $g(x_1, x_2, \ldots, x_{n-1}, y) = x_n$.

39 Ap. (Dalinai rekursyvių funkcijų aibė (DR)). Dalinai rekursyvių funkcijų aibė sutampa su aibe, kuriai priklauso bazinės funkcijos $0, s(x) = x + 1, pr_n^i(x_1, x_2, \ldots, x_n) = x_i$ ir kuri yra uždara kompozicijos, primityviosios rekursijos ir minimizacijos operatorių atžvilgiu.

Pastaba. Jei $f(x_1, x_2, \ldots, x_n) = \mu_y(g(x_1, x_2, \ldots, x_{n-1}, y) = x_n)$, tai $f(x_1, x_2, \ldots, x_n)$ reikšmė apskaičiuojama remiantis tokiu algoritmu:

$$\texttt{for (y = 0; } g(x_1, x_2, \ldots, x_{n-1}, y) \neq x_n\texttt{; y++);}$$

$$f(x_1, x_2, \ldots, x_n) = \begin{cases} y, & \text{jei pavyko rasti } y \text{ arba} \\ \infty, & \text{jei } y \text{ neegzistuoja, arba skaičiuojant buvo gauta neapibrėžtis.} \end{cases}$$

40 Ap. (Bendrųjų rekursyviųjų funkcijų aibė (BR)). Aibė sudaryta iš visų DR funkcijų, kurios yra apibrėžtos su visais natūraliaisiais argumentais.

22 T. *PR* $\subseteq$ *BR* $\subseteq$ *DR*

Pastaba. Žinomos primityviai rekursyvios funkcijos:

- bazinės funkcijos:

$$\begin{aligned} &0, \\ &s(x) = x + 1, \\ &pr_n^i(x_1, x_2, \ldots, x_n) = x_i; \end{aligned}$$

- vieno argumento PR funkcijų bazinės funkcijos:

$$\begin{aligned} &s(x) = x + 1, \\ &q(x) = x \dot{-} \left[\sqrt{x}\right]^2; \end{aligned}$$

- kitos žinomos PR funkcijos:

$$\operatorname{sg}(x) = \begin{cases} 1, & \text{jei } x > 0 \\ 0, & \text{jei } x = 0 \end{cases},$$

$$\overline{\operatorname{sg}}(x) = \begin{cases} 0, & \text{jei } x > 0 \\ 1, & \text{jei } x = 0 \end{cases},$$

$$x \dot{-} y = \begin{cases} x - y, & \text{jei } x > y \\ 0, & \text{kitu atveju} \end{cases},$$

$$\begin{aligned} x + y&, \\ x \cdot y&, \\ |x - y|&, \\ [x/y]&. \end{aligned}$$

Pastaba. Žinomos dalinai rekursyvios funkcijos:

- dalinis skirtumas:

$$x - y = \begin{cases} x - y, & \text{jei } x \geq y \\ \infty, & \text{kitu atveju} \end{cases},$$

- dalinė dalyba:

$$x/y = \begin{cases} x/y, & \text{jei } x \text{ dalosi iš } y \\ \infty, & \text{kitu atveju} \end{cases},$$

- dalinė šaknis:

$$\sqrt{x} = \begin{cases} \sqrt{x}, & \text{jei } x \text{ yra natūralaus skaičiaus kvadratas} \\ \infty, & \text{kitu atveju} \end{cases}.$$

7 Pvz. Tarkime, kad turime funkciją

$$g(x, y, z) = (s(z) \dot{-} x) \cdot \operatorname{sg}(y \dot{-} z).$$

Iš jos, pritaikę minimizacijos operatorių, galime gauti funkciją:

$$\begin{aligned} f'(x_1, x_2, x_3) &= \mu_y(g(x_1, x_2, y) = x_3) \\ &= \mu_y((s(y) \dot{-} x_1) \cdot \operatorname{sg}(x_2 \dot{-} y) = x_3). \end{aligned}$$

Dabar galime apibrėžti naują funkciją $f(x, y)$:

$$\begin{aligned} f(x, y) &= f'(x, y, x) \\ &= \mu_z((s(z) \dot{-} x) \cdot \operatorname{sg}(y \dot{-} z) = x). \end{aligned}$$

Apskaičiuokime $f(1,3)$:

1. Įsistatome reikšmes:

$$f(1,3) = \mu_z((s(z) \dot{-} 1) \cdot \mathrm{sg}(3 \dot{-} z) = 1).$$

2. Tikriname $z = 0$:

$$(s(0) \dot{-} 1) \cdot \mathrm{sg}(3 \dot{-} 0) = (1 \dot{-} 1) \cdot \mathrm{sg}(3) = 0 \cdot 1 = 0 \neq 1$$

3. Tikriname $z = 1$:

$$(s(1) \dot{-} 1) \cdot \mathrm{sg}(3 \dot{-} 1) = (2 \dot{-} 1) \cdot \mathrm{sg}(2) = 1 \cdot 1 = 1$$

Taigi gavome, kad $f(1,3) = 1$.

7.2. Rekursyviai skaičiosios aibės

41 Ap. (Rekursyviai skaiti aibė)**.** Aibė A yra rekursyviai skaiti, jei sutampa su kažkurios DR funkcijos apibrėžimo sritimi.

42 Ap. (Rekursyviai skaiti aibė)**.** Netuščia aibė A yra rekursyviai skaiti, jei sutampa su kažkurios PR funkcijos reikšmių sritimi.

43 Ap. (Rekursyviai skaiti aibė)**.** Aibė A yra rekursyviai skaiti, jei $\exists$ tokia PR funkcija $f(a, x)$, kad lygtis $f(a, x) = 0$ turi sprendinį tada ir tik tada, kai $a \in A$.

23 T. *Visi trys (41, 42 ir 43) rekursyvios aibės apibrėžimai yra ekvivalentūs netuščios aibės atžvilgiu.*
Proof $\triangleright$
(Dalinis 42 $\implies$ 41, 42 $\implies$ 43 ir 43 $\implies$ 42)

(42 $\implies$ 41) Tarkime, kad aibė A yra rekursyviai skaiti, nes $\exists$PR funkcija $h(x)$ tokia, kad A sutampa su h reikšmių sritimi (42 ap.):

$$A = \{h(0), h(1), h(2), h(3), \dots\}.$$

Imkime funkciją $f(x) = \mu_y(h(y) = x)$. Ji yra dalinai rekursyvi, nes yra gauta pagal minimizacijos operatorių iš dalinai rekursyvios funkcijos $h(\in \mathrm{PR} \subseteq \mathrm{DR})$.

- Jei $a \in A$, tai $\exists i (i \in \mathbb{N})$, kad $h(i) = a \implies f(a) = \mu_y(h(y) = a) < \infty$ – apibrėžta.
- Jei $a \notin A$, tai $\forall j (j \in \mathbb{N}) : h(j) \neq a \implies f(a) = \mu_y(h(y) = a) = \infty$ – neapibrėžta.

Aibė A sutampa su funkcijos $f(x)$ apibrėžimo sritimi ir $f(x) \in \mathrm{DR} \implies$ 41 ap.

(42 $\implies$ 43) Tarkime, jog aibė A sutampa su PR funkcijos $h(x)$ reikšmių sritimi (42 ap.). Tai yra:

$$A = \{h(0), h(1), h(2), h(3), \dots\}.$$

Imkime funkciją $f(a,x) = |h(x) - a|$. $f(a,x) \in$ PR, nes gauta iš PR funkcijų pritaikius kompozicijos operatorių. Lygtis $f(a,x) = 0$ yra $|h(x) - a| = 0$.

- Jei $a \in A$, tai $\exists$ toks i, kad $h(i) = a \implies |h(x) - a| = 0$, turi sprendinį $x = i$.
- Jei $a \notin A$, tai $\forall j (j \in \mathbb{N}) : h(j) \neq a \implies |h(x) - a| > 0$ – lygtis neturi sprendinių.

Lygtis $f(a,x) = 0$, turi sprendinį tada ir tik tada, kai $a \in A \implies$ 43 ap.

(43 $\implies$ 42) Tarkime, jog $a \in A$ tada ir tik tada, kai lygtis $f(a,x) = 0$, kur $f(a,x) \in$ PR, turi sprendinį. Imkime funkciją

$$h(t) = \pi_2^1(t) \cdot \overline{\text{sg}}(f(\pi_2^1(t), \pi_2^2(t))) + d \cdot \text{sg}(f(\pi_2^1(t), \pi_2^2(t))),$$

kur d – bet koks aibės A elementas. $h(t) \in$ PR, nes gauta iš žinomų PR funkcijų pritaikius kompozicijos operatorių. Parodysime, kad $h(t)$ reikšmių sritis sutampa su aibe A:

- Jei $a \in A$, tai lygtis $f(a,x) = 0$ turi sprendinį $x = u$. Pažymėkime $t = \alpha_2(a,u)$. Tada

$$\begin{aligned} h(t) &= h(\alpha_2(a,u)) \\ &= a \cdot \overline{\text{sg}}(\underbrace{f(a,u)}_{=0}) + d \cdot \text{sg}(\underbrace{f(a,u)}_{=0}) \\ &= a \cdot \overline{\text{sg}}(0) + d \cdot \text{sg}(0) \\ &= a \end{aligned}$$

- Jei $t = \alpha_2(a,v)$, kad $f(a,v) \neq 0$, tai

$$\begin{aligned} h(t) &= h(\alpha_2(a,v)) \\ &= a \cdot \overline{\text{sg}}(\underbrace{f(a,v)}_{>0}) + d \cdot \text{sg}(\underbrace{f(a,v)}_{>0}) \\ &= a \cdot 0 + d \cdot 1 \\ &= d \end{aligned}$$

Gavome, kad $h(t)$ su bet kokiu t įgyja reikšmę $\in A$ ir $h(t)$ įgyja visas reikšmes iš aibės A. Kadangi $h(t)$ reikšmių sritis yra aibė A ir $h(t) \in$ PR, tai gavome 42 apibrėžimą.

$\triangleleft$

Pastaba. Rekursyvių ir rekursyviai skaičių aibių skirtumas:

- A – rekursyvi, jei
$$\exists\, \chi_A(a) = \begin{cases} 1, & \text{jei } a \in A \\ 0, & \text{jei } a \notin A \end{cases} \in \mathrm{BR}$$
(Funkcija visur apibrėžta.)
- A – rekursyviai skaiti, jei
$$\exists\, \chi_A(a) = \begin{cases} 1, & \text{jei } a \in A \\ \infty, & \text{jei } a \notin A \end{cases} \in \mathrm{DR}$$
(Funkcija ne visur apibrėžta.)

Rekursyviai skaičių aibių savybės:

1. Rekursyvi aibė A yra ir rekursyviai skaiti:
$$\exists\, \chi_A(a) = \begin{cases} 1, & \text{jei } a \in A \\ 0, & \text{jei } a \notin A \end{cases},$$
tada aibė A sutampa su DR funkcijos $f(a) = \chi_A(a) - 1$ apibrėžimo sritimi. Pagal 41 apibrėžimą A yra rekursyviai skaiti.

2. Baigtinė aibė yra ir rekursyvi ir rekursyviai skaiti:
$$A = \{a_1, a_2, \ldots, a_n\} \text{ – baigtinė aibė.}$$
Tada, jos charakteringoji funkcija:
$$\chi_A(a) = \overline{\mathrm{sg}}(|a - a_1|) + \overline{\mathrm{sg}}(|a - a_2|) + \ldots + \overline{\mathrm{sg}}(|a - a_n|).$$

24 T. *Jeigu aibė A yra rekursyviai skaiti, bet A nėra rekursyvi, tai jos papildinys $\overline{A}$ nėra nei rekursyvi, nei rekursyviai skaiti aibė.*
Proof $\triangleright$

1. Tarkime, kad $\overline{A}$ yra rekursyviai skaiti.

2. Tada pagal 42 apibrėžimą yra PR funkcija $g(x)$, tokia kad $\overline{A} = \{g(0), g(1), g(2), \ldots\}$

3. Kadangi aibė A yra rekursyviai skaiti (duota), tai pagal 42 apibrėžimą yra PR funkcija $f(x)$ tokia, kad $A = \{f(0), f(1), f(2), \ldots\}$.

4. Panagrinėkime funkciją $h(x) = \mu_z(|f(z) - x| \cdot |g(z) - x| = 0)$. $h(x)$ yra BR, nes gauta iš PR funkcijų taikant kompozicijos, bei minimizacijos operatorius ir yra apibrėžta su visais x ($x \in \mathbb{N}$), nes $A \cup \overline{A} = \mathbb{N}$.

- jei $x \in A$, tada $\exists z : f(z) = x \implies |f(z) - x| = 0$,
- jei $x \notin A \implies x \in \overline{A}$, tada $\exists z : g(z) = x \implies |g(z) - x| = 0$.

5. - Jei $x \in A$, tada $h(x) = z$, kad $f(z) = x$.
   - Jei $x \in \overline{A}$, tada $h(x) = z$, kad $g(z) = x$ ir $f(z) \neq x$.

6. Tada aibės A charakteringoji funkcija būtų:

$$\begin{aligned}\chi_A(x) &= \overline{\text{sg}}(|f(h(x)) - x|) \\ &= \begin{cases} \overline{\text{sg}}(|f(z) - x|), & \text{kur } f(z) = x, \text{ jei } x \in A \\ \overline{\text{sg}}(|f(z) - x|), & \text{kur } f(z) \neq x, \text{ jei } x \notin A \end{cases}, \\ &= \begin{cases} \overline{\text{sg}}(0), & \text{jei } x \in A \\ \overline{\text{sg}}(y + 1), & \text{jei } x \notin A \end{cases},\end{aligned}$$

Čia su $y + 1$ norėta pasakyti, kad tai kažkas > 0. Taip ir bus, jei y yra bet koks natūralusis skaičius.

$$= \begin{cases} 1, & \text{jei } x \in A \\ 0, & \text{jei } x \notin A \end{cases}.$$

7. $\chi_A(x) \in$ BR, nes gauta iš BR ir PR funkcijų pritaikius kompozicijos operatorių.

8. Kadangi $\chi_A(x) \in$ BR, tai A – rekursyvi $\implies$ prieštara $\implies$ $\overline{A}$ – nėra rekursyviai skaiti aibė.

$\triangleleft$

7.3. Ackermann funkcijos.

Ackerman funkcijos yra:

$$\begin{aligned} B_0(a, x) &= a + x \\ B_1(a, x) &= a \cdot x \\ B_2(a, x) &= a^x \\ B_n(a, x + 1) &= B_{n-1}(a, B_n(a, x)) \\ B_n(a, 0) &= 1, \text{ kai } n \geq 2 \end{aligned}$$

Jos yra visur apibrėžtos ir algoritmiškai apskaičiuojamos funkcijos, todėl yra bendrai rekursyvios funkcijos.

Ackermann funkcijos variantas, kai $a = 2$, yra funkcija $A(n, x) = B_n(2, x)$. Jam teisingos lygybės ir nelygybės:

1. $A(n+1, x+1) = B_{n+1}(2, x+1) = B_n(2, B_{n+1}(2, x)) = A(n, A(n+1, x))$
2. $A(0, x) = x + 2$
3. $A(1, 0) = 0$
4. $A(n, 0) = 1$, kai $n \geq 2$
5. $A(n, x) \geq 2^x$, kai $n \geq 2$ ir $x \geq 1$
6. $A(n+1, x) > A(n, x)$, kai $n, x \geq 2$
7. $A(n, x+1) > A(n, x)$, kai $n, x \geq 2$
8. $A(n+1, x) \geq A(n, x+1)$, kai $n, x \geq 2$

44 Ap. (Mažoruojanti funkcija). Sakysime, kad funkcija $f(x)$ yra mažoruojama[1] funkcijos $h(x)$, jei yra toks $n_0 \in \mathbb{N}$, kad $f(x) < h(x)$ su visais $x > n_0$.

25 T. *Jei $f(x)$ yra vieno argumento PR funkcija, tai egzistuoja toks n $(n \in \mathbb{N})$, kad*

$$f(x) < A(n, x), \text{ kai } x > 2.$$

26 T. *Egzistuoja tokios funkcijos, kurios yra bendrai rekursyvios, bet nėra primityviai rekursyvios.*

Proof $\triangleright$
Parodysime, kad funkcija $h(x) = A(x, x)$ yra BR, bet nėra PR.

$h(x) \in$ BR, nes ji yra visur apibrėžta, algoritmiškai apskaičiuojama funkcija.

Įrodysim, kad $h(x)$ mažoruoja bet kokią PR vieno argumento funkciją.

Tegu $f(x) \in$ PR. Iš 25 teiginio: $\exists n$ $(n \in \mathbb{N})$, kad $f(x) < A(n, x)$, kai $x > 2$.

Tuomet $f(n+x) < A(n, n+x) < A(n+x, n+x) = h(n+x) \implies \exists n_0$ $(n_0 \in \mathbb{N})$, kad $f(x) < h(x)$, kai $x > n_0 \implies f(x)$ yra mažoruojama funkcijos $h(x)$.

Kadangi bet kokia vieno argumento primityviai rekursyvi funkcija yra mažoruojama funkcijos $h(x)$, tai funkcija $h(x) \notin$ PR.

$\triangleleft$

[1] Žr. DLKŽ.

7.4. Universaliosios funkcijos

45 Ap. (Universalioji funkcija). Tarkime, kad A yra kuri nors n argumentų funkcijų aibė. Funkciją $F(x_0, x_1, \ldots, x_n)$ vadinsime aibės A universaliąja, jei

$$A = \{F(0, x_1, x_2, \ldots, x_n), F(1, x_1, x_2, \ldots, x_n), F(2, x_1, x_2, \ldots, x_n), \ldots\},$$

tai yra, jei $F(i, x_1, x_2, \ldots, x_n) \in A$ $(i = 0, 1, 2, \ldots)$ ir nesvarbu, kokia būtų $f(x_1, x_2, \ldots, x_n) \in A$, atsiras bent vienas toks natūralusis skaičius i, kad $f(x_1, x_2, \ldots, x_n) = F(i, x_1, x_2, \ldots, x_n)$.

27 T.

1. *Visų n argumentų PR funkcijų universalioji $F(i, x_1, x_2, \ldots, x_n) \notin$ PR.*

2. *Visų n argumentų BR funkcijų universalioji $F(i, x_1, x_2, \ldots, x_n) \notin$ BR.*

Proof $\triangleright$

1. Tegu $F(i, x_1, x_2, \ldots, x_n) \in$ PR. Imkime funkciją $g(x_1, x_2, \ldots, x_n) = F(x_1, x_1, x_2, \ldots, x_n) + 1$. $g(x_1, x_2, \ldots, x_n) \in$ PR, nes gauta iš PR funkcijų $F(i, x_1, x_2, \ldots, x_n)$ ir $(x + y)$ pritaikius kompozicijos operatorių. Kadangi $g(x_1, x_2, \ldots, x_n)$ yra n argumentų PR funkcija, tai $\exists j\ (j \in \mathbb{N})$ toks, kad $F(j, x_1, x_2, \ldots, x_n) = g(x_1, x_2, \ldots, x_n)$. Tuomet paimkime $x_1 = x_2 = \cdots = x_n = j$, tada $F(j, j, j, \ldots, j) = g(j, j, \ldots, j) = F(j, j, j, \ldots, j) + 1 \implies F(j, j, j, \ldots, j)$ – neapibrėžta $\implies$ prieštara, nes visos PR funkcijos yra apibrėžtos su visais argumentais, todėl mūsų prielaida, jog $F(i, x_0, x_1, \ldots, x_n) \in$ PR yra neteisinga.

2. Tegu $F(i, x_1, x_2, \ldots, x_n) \in$ BR. Imkime funkciją $g(x_1, x_2, \ldots, x_n) = F(x_1, x_1, x_2, \ldots, x_n) + 1$. $g(x_1, x_2, \ldots, x_n) \in$ BR, nes gauta iš BR funkcijų $F(i, x_1, x_2, \ldots, x_n)$ ir $(x + y)$ pritaikius kompozicijos operatorių. Kadangi $g(x_1, x_2, \ldots, x_n)$ yra n argumentų BR funkcija, tai $\exists j\ (j \in \mathbb{N})$ toks, kad $F(j, x_1, x_2, \ldots, x_n) = g(x_1, x_2, \ldots, x_n)$. Tuomet paimkime $x_1 = x_2 = \cdots = x_n = j$, tada $F(j, j, j, \ldots, j) = g(j, j, \ldots, j) = F(j, j, j, \ldots, j) + 1 \implies F(j, j, j, \ldots, j)$ – neapibrėžta $\implies$ prieštara, nes visos BR funkcijos yra apibrėžtos su visais argumentais, todėl mūsų prielaida, jog $F(i, x_0, x_1, \ldots, x_n) \in$ BR yra neteisinga.

$\triangleleft$

28 T. *Visų vieno argumento PR funkcijų aibei egzistuoja universalioji funkcija, kuri yra BR.*
Proof $\triangleright$
Pagal 21 teiginį visas vieno argumento PR funkcijas galime išreikšti per bazines $s(x)$, $q(x)$ ir sudėties, kompozicijos, bei iteracijos operatorius. Remiantis išraiška per $s(x)$ ir $q(x)$, visoms PR vieno argumento funkcijoms suteiksime numerius tokiu būdu:

- $n(s(x)) = 1$
- $n(q(x)) = 3$

Jei $n(f(x)) = a$ ir $n(g(x)) = b$, tai

- $h(x) = f(x) + g(x) \implies n(h(x)) = 2 \cdot 3^a \cdot 5^b$
- $h(x) = f(g(x)) \implies n(h(x)) = 4 \cdot 3^a \cdot 5^b$
- $h(x) = f(x)^I \implies n(h(x)) = 8 \cdot 3^a$

Tarkime, turime funkciją $f_n(x)$, kurios numeris yra n. Pažymėkime $F(n,x) = f_n(x)$. Tada:

$$F(n,x) = \begin{cases} f_a(x) + f_b(x), & \text{jei } n = 2 \cdot 3^a \cdot 5^b \\ f_a(f_b(x)), & \text{jei } n = 4 \cdot 3^a \cdot 5^b \\ f_a(f_n(x-1)), & \text{jei } n = 8 \cdot 3^a \text{ ir } x > 0 \\ 0, & \text{jei } n = 8 \cdot 3^a \text{ ir } x = 0 \\ s(x), & \text{jei } n = 1 \\ q(x), & \text{jei } n = 3 \end{cases}$$

$F(n,x)$ yra algoritmiškai apskaičiuojama. Tada visų vieno argumento PR universalioji yra:

$$D(n,x) = \begin{cases} F(n,x), & \text{jei } n \text{ – kažkurios funkcijos numeris} \\ 0, & \text{kitu atveju} \end{cases}.$$

$D(n,x) \in$ BR, nes yra algoritmiškai apskaičiuojama ir visur apibrėžta.

$\lhd$

29 T. *Visų n argumentų PR funkcijų universalioji yra funkcija:*

$$D^{n+1}(x_0, x_1, x_2, \ldots, x_n) = D(x_0, \alpha_n(x_1, x_2, \ldots, x_n))$$

46 Ap.. DR n argumentų funkcijos grafikas yra taškų, sudarytų iš $(n+1)$ elemento, aibė:

$$G = \{(x_1, x_2, \ldots, x_n, y) : f(x_1, x_2, \ldots, x_n) = y\}$$

PAKEITIMAI

d1c7c2e7bbc0c819cba92a03b60662aedf107324		
Vytautas Astrauskas	2015-06-16 15:39:37 +0200	Smulkus pataisymas.
10c4be5f50afda2a368fda677afb63eef6b87382		
Vytautas Astrauskas	2013-01-03 21:38:33 +0100	Smulkūs formatavimo pataisymai.
a00590681359d0b4b111595d4ee4f5f79b2a822e		
Vytautas Astrauskas	2013-01-03 21:16:51 +0100	Pataisymai pagal lekt. dr. Adomo Birštuno pastabas.
f0fec6a1ce78a20cb06f83c6e5daf56c0e810238		
Vytautas Astrauskas	2013-01-03 18:03:25 +0100	Pridėtas pakeitimų sąrašas.
96c6758732a27c45a27ffa7c80a4b8f63b76fb99		
Vytautas Astrauskas	2013-01-03 17:45:04 +0100	Atnaujintas šablonas.