<u>AT uždavinių analizė</u>

(būtinoji teorija + pratybos + uždavinių sprendimai)

[Matematinės logikos pagrindai]

Lekt. A.Birštuno dėstomo kurso „Algoritmų teorija“ praktinis taikymas uždavimų sprendimui.

Parengė Kęstutis Matuliauskas
2010 m. gegužės 28 d.

Užrašams

1 leidimas – 2009.06.09
2 leidimas – 2009.09.09
3 leidimas – 2010.05.08
4 leidimas – 2010.05.28

Pratarmė

A. Birštuno skaitomas kursas – „*Algoritmų teorija*“, šiuo leidiniu, yra praplečiamas aukštųjų mokyklų studentams svarbia medžiaga apie praktinį turimų žinių panaudojimą, sprendžiant matematinės logikos uždavinius. Šis leidinys remiasi doc. S. Norgėlos knygoje „Logika ir Dirbtinis intelektas“ išdėstyta medžiaga, bei lekt. A. Birštuno paskaitų ir pratybų metu išdėstyta medžiaga. Dalis informacijos yra paimta iš internetinių šaltinių, bei kitų aukštųjų mokyklų literatūros leidinių. Šio leidinio esmė - konkretus, struktūrizuotas konspektas, kurį būtų lengva suprasti, kuris nebūtų apkrautas per dideliu kiekiu informacijos, o informacija jame būtų dėsningai išskirstyta į temas, kurių kiekviena klasifikuota pagal tam tikrus, su ta tema susijusius, dėsningumus. Prie kiekvienos temos yra pateikiama visa būtinoji teorija, reikalinga žinoti, norint išspręsti leidinyje pateiktus, bei panašius matematinės logikos uždavinius.

Turinys

ĮVADAS .... **- 3 -**

UŽDAVINIŲ SPRENDIMAI .... **- 5 -**

PAGRINDINIAI LOGIKOS UŽDAVINIAI .... - 5 -

Pratybos nr.1. Hilberto tipo teiginių skaičiavimas .... *- 6 -*

Pratybos nr.2. Sekvekcinis skaičiavimas G .... *- 9 -*

Pratybos nr.3. Rezoliucijų metodas .... *- 12 -*

Pratybos nr.4-1. Turingo mašinų variantai .... *- 15 -*

Pratybos nr.4-2. Baigtiniai automatai .... *- 23 -*

Pratybos 6. Porų numeravimas .... *- 32 -*

REKURSYVIOSIOS FUNKCIJOS/AIBĖS .... - 35 -

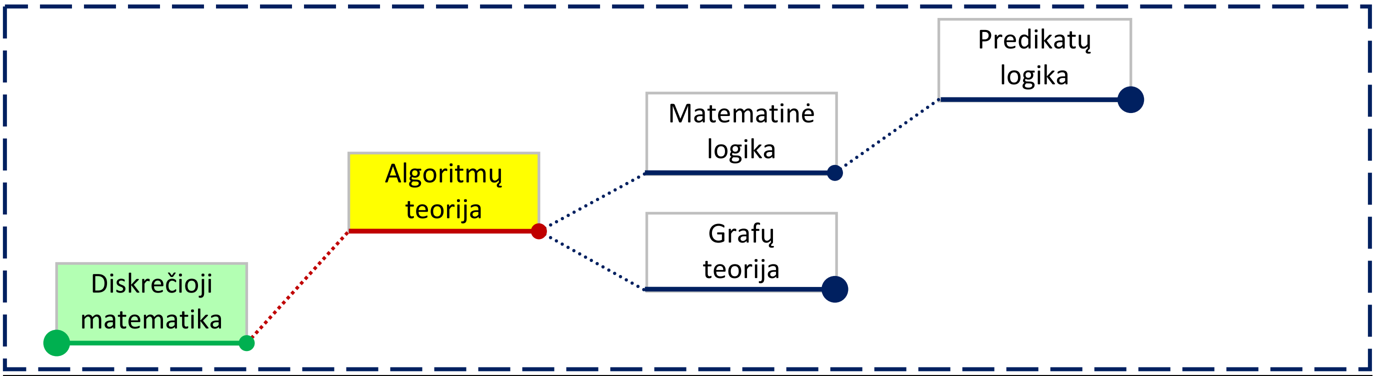
Pratybos 7. Primityviai rekursyviosios funkcijos .... *- 38 -*

Pratybos 8. Skaičiavimas Ackermann funkcijomis .... *- 40 -*

Pratybos 9. Minimizacijos operatorius .... *- 43 -*

Pratybos 10. Rekursyvios ir rekursyviai skaičios aibės .... *- 45 -*

TEIGINIŲ ĮRODYMAI .... **- 51 -**



Uždavinių sprendimai

Pagrindiniai logikos uždaviniai

Pratybos nr.1. Hilberto tipo teiginių skaičiavimas

Aksiomos:

1.1. A-> (B->A)
1.2. (A->(B->C))->((A->B) -> (A->C))

&
2.1. A&B -> A
2.2. A&B -> B
2.3. (A->B) -> ((A->C) -> (A->(B & C)))

V
3.1. A -> (A V B)
3.2. B -> (A V B)
3.3. (A->C) -> ((B->C) -> ((A V B) -> C))

¬
4.1. (A->B) -> (¬B -> ¬A)
4.2. A -> ¬¬A
4.3. ¬¬A -> A

Ir Modus Ponens(toliau – MP) taisyklė: $\frac{A \quad A\to B}{B}$

Paprasčiau:
Iš aksiomos A, ir formulės, kuri susideda iš A->B. B yra bet kokia formulė(ne aksioma), gauname formulę B.

Ženklų reikšmės:

Ženkliukas „|-" žymi, kad formulė F yra tapačiai teisinga. Žymime |-F.
Ženkliukas „¬" žymi neigimą. Invertuoja teiginio/prielaidos reikšmes.
Ženkliukas „/\" yra ekvivalentus ženkliukui „&" ir žymi loginę daugybą(konjunkciją). Teiginys teisingas tik tuomet kaip abu teiginiai A ir B yra teisingi.
Ženkliukas „V" žymi loginę sudėtį(disjunkciją). Klaidingas tik tuomet, kaip abu A ir B yra klaidingi.
Ženkliukas „->" žymi loginę išvadą(implikaciją). Klaidingas tik tuomet, kai iš teisingos prielaidos(A) seka klaidinga išvada(B).

Eiliškumas:

1.Neigimas (**¬**)
2.Loginė daugyba (**&**).
3.Loginė sudėtis (**V**).
4.Kitos operacijos.

Teorija:

Dedukcijos teorema *(žymime D.T.)* :

$$\Gamma \vdash A -> B \Leftrightarrow \Gamma, A \vdash B$$

Pastabos:

1. Ženkliukas "Γ"(gama) žymi prielaidas, raidės A,B formules. A – prielaida, B – išvada.
2. Dedukcijos teoremą galime taikyti tol, kol mūsų formulėje yra implikacija(**->**).

Uždavinys nr.1:

2.f) Naudojantis dedukcijos teorema(D.T.) įrodyti Hilberto tipo teiginiu skaiciavime:

$\vdash (p \to (q \to r)) \to ((p \& q) \to r)$.

Pastaba.*(Pavyzdys):* *$\vdash$(p & q) -> (p V q) reikštų: „Iš prielaidos,'jeigu abu p ir q teiginiai yra teisingi', galime daryti išvadą kad 'bent vienas iš teiginių – p arba q, yra teisingas'.*

Sprendimas:

<1>. Jeigu mūsų duotoje sąlygoje randame implikaciją, taikome dedukcijos teoremą.
Pritaikome dedukcijos teoremą:

D.T.: $\Gamma \vdash A \to B \Leftrightarrow \Gamma, A \vdash B$

$A = (p \to (q \to r))$, $B = ((p \& q) \to r)$

* A raide dedukcijos teoremoje pažymime formulę „$(p \to (q \to r))$“, B raide – „$((p \& q) \to r)$“.

<2>. Dedukcijos teoremą taikome tol, kol yra išorinių implikacijų išvadoje. Gauname:

1) $\vdash (p \to (q \to r)) \to ((p \& q) \to r)$

   $\Updownarrow$ **D.T.**

2) $p \to (q \to r) \vdash (p \& q) \to r$

   $\Updownarrow$ **D.T.**

3) $p \to (q \to r), p \& q \vdash r$

! Daugiau dedukcijos teoremos nebetaikome, nes mūsų formulėje(išvadoje) nebeliko implikacijų($\to$)

Vadinasi reikia imti formulę:

„ $p \to (q \to r), p \& q \vdash r$ “

ir ją įrodinėti Hilberto tipo teiginių skaičiavime(išskirti punktus, rasti aksiomas, bei pritaikyti MP taisyklę).

<3>. Mums reikia įrodyti, kad formulė(išvada) „**F** = r“ yra tapačiai teisinga*(plačiau apie tapatų teisingumą skaitykite – „Pratybos nr.3. Rezoliucijų metodas“).*

**Šį kartą mūsų išvada F labai paprasta, todėl iškart aišku kad formulė tikrai turės tapataus teisingumo variantą.*

Vadinasi ieškome aksiomos, kuri užsibaigtų mūsų ieškoma formule F.
Rezultate, jei pasiseks, turėtume gauti formulę „iš prielaidos(-ų) seka išvada“.
Rezultate gauti formulę: „***r***“

//Susižymime prielaidas:

1. $p \to (q \to r)$ ----- prielaida,

2. **p & q** ----- prielaida,

// Panaudojame 2.1. aksiomą ((A&B) – A)

3. p & ? $\to$ p ----- 2.1. aksioma: **A – p, B – ?**

Pastaba. Į klaustuko vietą rašome bet ką, pvz. B– (p & q) . Tačiau žiūrime kas būtų prasmingiausia(padėtų lengviau išspręsti). Be to, negalime rašyti jau kartą panaudoti kintamojo, t.y. negalime vietoje B statyti to pačio ką įstatėme, kaip B*(ar bet kokio kito, jeigu yra)* kintamojo.

// kas šiuo atveju yra B – spėjame, kas mūsų manymų yra geriausia. Šiuo atveju mes norėsime pasinaudoti prielaida ,2', todėl geriausiai tiktų **B – q**:

3. p & q $\to$ p ----- 2.1. aksioma: **A – p, B – q**

// Pritaikome Modus Ponens taisyklę

4. p ----- Pagal MP iš 2 ir 3 // (t.y. **p & q** ir **p & q → p** ⟹ ~~p & q →~~ p)

// Toliau taikome MP taisyklę:

5. q → r ----- Pagal MP iš 4 ir 1

// Įsistatome į 2.2. aksiomą „ (A&B) → B “ , B – q. O kas yra A – spėjame, kas tinka geriausiai – šiuo atveju geriausiai tinka A – p, nes yra tokia prielaida „2.“

6. p & q → q ----- 2.2. aks. A – p, B – q

// Dar kartą taikome MP taisyklę:

7. q ----- pagal MP iš 2 ir 6

// Ir galiausiai, paskutinį kartą pritaikę MP taisyklę, gauname tai ką turėjome įrodyti H.T.T. skaičiavime – mūsų formulės išvadą **F =r**

8. r ----- pagal MP iš 7 ir 5.

Atsakymas:

Įrodyta, kad iš prielaidų „p **→** (q → r)“ ir „p **&** q“ tikrai seka mūsų ieškota išvada „r“.

~~~~~~

Uždavinys nr.2:

2.f) **NESINAUDOJANT** dedukcijos teorema(D.T.) įrodyti Hilberto tipo teiginiu skaiciavime:

((p V q) & w) & p ⊢ q V w .

Sprendimas:

<0>. Imame ((p V q) & w) & p ⊢ **q V w** geltonai pažymėtą dalį ir pritaikome 3.2 aksiomą (ji baigiasi mūsų ieškoma formule F = q V w)

1. w → (q V w)

2. (p V q) & w → **w**

<1>. Taikome 2.1 aksiomą(A & B → A). Šiuo atveju mūsų A yra 2-ame punke geltonai pažymėta formulė:

3. (((p V q) & w) & **?**) → **(p V q) & w**

<2>. Šiuo atveju spėjame kas yra B.Manome, kad geriausiai tiktų „w“.

3. (((p V q) & w) & **p**) → (p V q) & w

4. ((pVq) & w) & p --- prielaida

// Du kartus taikome MP taisyklę

5. (p V q) & **w** --- pagal MP iš 3 ir 4.

6. **w** --- pagal MP iš 5 ir 2.

// Ir gauname rezultatą paskutinį kartą pritaikę MP taisyklę:

7. q V w --- pagal MP iš 6 ir 1.

Atsakymas:

Įrodėme, kad iš prielaidos „((p V q) & w) & p“ tikrai seka mūsų ieškota išvada „q V w“.
~~~~~~

Pratybos nr.2. Sekvekcinis skaičiavimas G

Teorija:

1. **Sekvencija ir jos antecedentas ir sukcedentas.** Sekvencija vadiname reiškinį $A_1, ..., A_n \vdash B_1, ..., B_m$: čia $A_i (i = 1..n)$ bei $B_i (i = 1..m)$ yra formulės, o $n + m > 0$. Sekvencijoje $\Gamma \vdash \Delta \Gamma$ - **antecedentas**, o Δ - **sukcedentas**.
2. **Sekvencinis skaičiavimas G.** Sekvencijos išvedimu sekvenciniame skaičiavime G vadiname medį, kurio visose galinėse viršūnėse (lapuose) yra aksiomos, likusiose viršūnėse - formulės gautos pagal kurią nors sekvencinio skaičiavimo taisyklę iš tiesiogiai virš jų medyje esančių formulių, ir šaknyje esanti sekvencija lygi pradinei.
3. **Jei sekvencija** $A_1, ..., A_n \vdash B_1, ..., B_m$ yra išvedama, tai reiškia, kad iš prielaidų $A_1, ..., A_n$ seka bent viena išvada iš $B_1, ..., B_m$.
4. Sakysime, kad sekvencija **S** ($\Gamma \vdash \Delta$) yra išvedama skaičiavime **G**, jei galima sukonstruoti tokį medį(grafą), kuriame:
   1) Visose medžio mazguose yra sekvekcija .
   2) Šaknyje yra sekvencija $\Gamma \vdash \Delta$.
   3) Jei viršūnės S_1 ir S_2 yra viršūnės **S** vaikai, tai sekvencija esanti viršūnėje S yra gauta pagal kažkurią taisyklę iš sekvencijų, esančių viršūnėse S_1 ir S_2.
   4) Visuose medžio lapuose(galinėse viršūnėse) yra sekvencijos aksiomos.

Sekvencija turi **VIENĄ** aksiomą, ir daug taisyklių.
Sekvencijos aksioma: $A, F, B \vdash C, F, D$
,kur A, B, C ir D - formuliu aibes(bet koks kiekis formulių)
, F - kažkokia konkreti formulė ar kintamasis.
Kad dėtume + prie lapo(viršūnėje), jame turi būti dvi vienodos formulės po vieną kiekvienoje pusėje.

Teorema:

Atkirtos taisyklė (žymime A.T.):

$$\frac{\neg p \vee C_1 \qquad p \vee C_2}{C_1 1 \vee C_2}$$

Pastabos:

1. Atkirtos taisyklę per vieną operaciją galime taikyti tik vienai kintamųjų porai.
2. $p \vee p \vee C_1 \Rightarrow p \vee C_1$

Sekvencinio skaičiavimo G taisyklės:

Pilnos taisyklės:

Pusė / Taisyklė	*Dešinėje*	*Kairėje*
Implikacija	$(\vdash\to)\ \dfrac{\Gamma_1, \mathbf{A}, \Gamma_2 \vdash \Delta_1, \mathbf{B}, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \mathbf{A} \to \mathbf{B}, \Delta_2}$	$(\to\vdash)\ \dfrac{\Gamma_1, \Gamma_2 \vdash \mathbf{A} \qquad \Gamma_1, \mathbf{B}, \Gamma_2 \vdash \Delta_1, \Delta_2}{\Gamma_1 \mathbf{A} \to \mathbf{B} \vdash \Delta_1, \Delta_2}$
Konjunkcija	$(\vdash \&)\ \dfrac{\Gamma_1, \Gamma_2 \vdash \Delta_1, \mathbf{A}, \Delta_2 \qquad \Gamma_1, \Gamma_2 \vdash \Delta_1, \mathbf{B}, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \mathbf{A} \to \mathbf{B}, \Gamma_2}$	$(\& \vdash)\ \dfrac{\Gamma_1 \mathbf{A}, \mathbf{B}, \Gamma_2 \vdash \Delta_1, \Delta_2}{\Gamma_1 \mathbf{A} \& B, \Gamma_2 \vdash \Delta_1, \Delta_2}$
Disjunkcija	$(\vdash \vee)\ \dfrac{\Gamma_1, \Gamma_2 \vdash \Delta_1, \mathbf{A}, \mathbf{B}, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \mathbf{A} \vee \mathbf{B}, \Delta_2}$	$(\vee \vdash)\ \dfrac{\Gamma_1, \mathbf{A}, \Gamma_2 \vdash \Delta_1, \Delta_2 \qquad \Gamma_1, \mathbf{B}, \Gamma_2 \vdash \Delta_1, \Delta_2}{\Gamma_1, \mathbf{A} \vee \mathbf{B}, \Gamma_2 \vdash \Delta_1, \Delta_2}$
Neigimas	$(\vdash \neg)\ \dfrac{\Gamma_1, \mathbf{A}, \Gamma_2 \vdash \Delta_1, \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \neg\mathbf{A}, \Delta_2}$	$(\neg\vdash)\ \dfrac{\Gamma_1, \Gamma_2 \vdash \Delta_1, \mathbf{A}, \Delta_2}{\Gamma_1, \neg\mathbf{A}, \Gamma_2 \vdash \Delta_1, \Delta_2}$

* Čia $\Gamma_1, \Gamma_2, \Delta_1, \Delta_2$- bet kokio ilgio formulių sekos. Arba tušti elementai.

** Taisyklės vardiklis – sąlyga, skaitiklis – rezultatas.

Supaprastintos taisyklės:

Supaprastinta lentelė	
$(\vdash\to)\ \dfrac{\mathbf{A} \vdash \mathbf{B}}{\vdash \mathbf{A} \to \mathbf{B}}$	$(\to\vdash)\ \dfrac{\vdash \mathbf{A} \qquad \mathbf{B} \vdash}{\mathbf{A} \to \mathbf{B} \vdash}$
$(\vdash \&)\ \dfrac{\vdash \mathbf{A} \qquad \vdash \mathbf{B}}{\vdash \mathbf{A} \to \mathbf{B}}$	$(\& \vdash)\ \dfrac{\mathbf{A}, \mathbf{B} \vdash}{\mathbf{A} \& B \vdash}$
$(\vdash \vee)\ \dfrac{\vdash \mathbf{A}, \mathbf{B}}{\vdash \mathbf{A} \vee \mathbf{B}}$	$(\vee \vdash)\ \dfrac{\mathbf{A} \vdash \qquad \mathbf{B} \vdash}{\mathbf{A} \vee \mathbf{B} \vdash}$
$(\vdash \neg)\ \dfrac{\mathbf{A} \vdash}{\vdash \neg\mathbf{A}}$	$(\neg\vdash)\ \dfrac{\vdash \mathbf{A}}{\neg\mathbf{A} \vdash}$

Minimalistinė lentelė
$(\vdash\to)\ \mathbf{A} \vdash \mathbf{B}$
$(\vdash \&)\ \vdash \mathbf{A} \quad \vdash \mathbf{B}$
$(\vdash \vee)\ \vdash \mathbf{A}, \mathbf{B}$
$(\vdash \neg)\mathbf{A} \vdash$
$(\to\vdash)\ \vdash \mathbf{A} \quad \mathbf{B} \vdash$
$(\& \vdash)\ \mathbf{A}, \mathbf{B} \vdash$
$(\vee \vdash)\ \mathbf{A} \vdash \quad \mathbf{B} \vdash$
$(\neg\vdash) \vdash \mathbf{A}$

Uždavinys:

Ar formulė išvedama sekvenciniame skaičiavime G:

1.e) (p V (¬(r -> ¬q))) -> (¬q V p) ⊦ p V (r -> ¬q)

Sprendimas:

(p V (¬(r -> ¬q))) -> (¬q V p) ⊦ p V (r -> ¬q)

----------------*(Taikome „ ⊦V“ taisyklę, čia A – geltonai, B – žaliai)*-------------------- **(⊦ V)**

(p V (¬(r -> ¬q))) -> (¬q V p) ⊦ p, r -> ¬q

----------------*(Taikome „-> ⊦“ taisyklę, čia A – geltonai, B – žaliai)*-------------------- **(⊦ ->)**

(p V (¬(r -> ¬q))) -> (¬q V p), r ⊦ p, ¬q

---------------------------*(Taikome „ ⊦¬“ taisyklę, čia A – geltonai)*------------------------ **(⊦ ¬)**

(p V (¬(r -> ¬q))) -> (¬q V p), r, q ⊦ p

--------------------*(Taikome „ ⊦¬“ taisyklę, čia A – geltonai, B – žaliai)*-------------------- **(-> ⊦)**

// Pagal taisyklę, mūsų medis skyla į dvi atšakas, tad toliau sprendžiame kiekvieną šaką atskirai

r, q ⊦ p, p V (¬(r -> ¬q)) **;** ¬q V p, r, q ⊦ p

(⊦ V) --- --- **(V ⊦)**

r, q ⊦ p, p, ¬(r -> ¬q) **;** ¬q, r, q ⊦ p ; p, r, q ⊦ p

(⊦ ¬) ------------------------------------- ------------------------ **(¬ ⊦)**

r, q, r -> ¬q ⊦ p, ~~p~~ **;** r, q ⊦ p, q ;

(-> ⊦) ---

r, q ⊦ r, p **;** r, q, ¬q ⊦ p

// Iš prielaidos „**r**“ išplaukia išvada „**r**“ --------------------------- **(¬ ⊦)**

// Vadinasi įrodyta r, q ⊦ p, q

// P.S. kadangi „**p, p**“ dubliuojasi,

// vieną „**p**“ galima nubraukti

Visos viršūnės yra sekvencijos aksiomos.

Pratybos nr.3. Rezoliucijų metodas

Logikos operatorių reikšmės:

p	q	¬p	p V q	p & q	p → q	p ⇔ q	p ⊕ q	p / q
t	t	k	t	t	t	t	k	k
t	k	k	t	k	k	k	t	t
k	t	t	t	k	t	k	t	t
k	k	t	k	k	t	t	k	t

Teorija:

1. ***Disjunktų dedukcinė sistema*** patikrina ar disjunktų aibė prieštaringa.

2. ***Rezoliucijų metodas*** patikrina ar iš prielaidų seka atitinkama išvada.

3. **Aibė** yra **prieštaringa**, jei su ∀ aibės interpretacija ∃ klaidinga aibės formulė.

Pvz. B = { p&q, p-> ¬q , p V q }:

Interpretacija

p	q	p&q	p -> ¬ q	p V q
0	0	0	1	0
0	1	0	1	1
1	0	0	1	1
1	1	1	0	1

Išvada: formulių aibė B yra prieštaringa.
(Todėl kad kiekvienoje lentelės eilutėje*(interpretacijoje),* ∃ klaidinga aibės formulė).

Pastaba: Jeigu ∃-tų tokia aibės eilutė, su kuria visos aibės formulės būtų teisingos(t.y. ∃ eilutė 1 1 1), tai formulių aibė B būtų neprieštaringa.

2.**Litera** –tai kintamasis, arba kintamasis su neigimu. *(pvz. p, ¬q, w – literos)*

3.**Disjunktas** – tai literų disjunkcija, t.y. formulė pavidalo: $l_1 \vee ... \vee l_s$, kur l_i – literos. *(pvz.* $p \vee \neg q \vee \neg r$ *)*

4.**Horno disjunktas** – tai disjunktas, kuriame yra ne daugiau kaip viena neigiama įeitis. *(pvz.* $p \vee q \vee \neg r$ *)*

Formulės:

Atkirtos taisyklė *(žymime A.T.)*:

$$\frac{\neg p \vee C_1 \qquad p \vee C_2}{C_1 \vee C_2}$$

Pastabos:

1. Atkirtos taisyklę per vieną operaciją galime taikyti tik vienai kintamųjų porai.
2. $p \vee p \vee C_1 \Rightarrow p \vee C_1$
3. Bandant išvesti tuščią disjunktą, iš disjunktų aibės S galime išbraukti visus disjunktus į kuriuos įeina kintamasis, ir tas pats kintamasis su neigimu, nes jei nauja gauta aibė S‘ bus prieštaringa, tai ir S – prieštaringa.
4. Bandant išvesti tuščią disjunktą, tą padaryti yra kiek paprasčiau, jei nagrinėjama disjunktų aibė S, susideda tik iš Horno disjunktų.

5. Normalinė konjunkcija forma:

Normalinė konjunkcija forma*(žymime NKF)* – tai formulė pavidalo:

D_1 & D_2 & ... & D_k;   (Čia D_i – disjunktas.)

Veiksmų seka, norint paprastai ir greitai gauti iš formulės jos NKF:

1. Eliminuoti visas logines operacijas, išskyrus **V**, **&** ir **¬** , t.y. panaikinti **→**, **↔**, **⊕** ir **/**:
   $\boldsymbol{A} \to \boldsymbol{B} \sim \neg\boldsymbol{A} \vee \boldsymbol{B}$
   $\boldsymbol{A} \leftrightarrow \boldsymbol{B} \sim (\boldsymbol{A} \to \boldsymbol{B}) \,\&\, (\boldsymbol{B} \to \boldsymbol{A})$
   $\boldsymbol{A} \oplus \boldsymbol{B} \sim \neg(\boldsymbol{A} \leftrightarrow \boldsymbol{B})$
   $\boldsymbol{A} \,/\, \boldsymbol{B} \sim \neg(\boldsymbol{A} \,\&\, \boldsymbol{B})$
2. Įkeliame **¬** iki kintamųjų, naudodamiesi De Morgano dėsniais:
   $\neg(\boldsymbol{A} \vee \boldsymbol{B}) \sim \neg\boldsymbol{A} \,\&\, \neg\boldsymbol{B}$
   $\neg(\boldsymbol{A} \,\&\, \boldsymbol{B}) \sim \neg\boldsymbol{A} \vee \neg\boldsymbol{B}$
3. Taikome distribūtyvumo dėsnius NKF gauti:
   $\boldsymbol{A} \vee (\boldsymbol{B} \,\&\, \boldsymbol{C}) \sim (\boldsymbol{A} \vee \boldsymbol{B}) \,\&\, (\boldsymbol{A} \vee \boldsymbol{C})$

Rezoliucijų metodas:

1. Turime formulių aibę: **A_1, A_2, ..., A_n** ⊢ **F** (iš prielaidų $\Gamma = \{A_1, \ldots\}$ seka išvada $\Delta\{F\}$) ir norime patikrinti ar šį iš prielaidų **Γ** tikrai seka išvada **F**, todėl į formulių aibę **B**, išvadą rašome su neigimu:

2. $B = \{ A_1, A_2, \ldots, A_n, \neg F \}$

3. Naudodami elementarius matematinės logikos formulių pertvarkius sukonstrukuojame **NKF**. Konjunkcijas pakeitę į kablelius, sukonstruojame disjungtų aibę **S**:
$S = \{ D_1, D_2, \ldots, D_m \}$

4. Jei iš aibės **S***(naudojant A.T.)* išvedamas tuščias disjunktas☐, tai vadinasi aibė S yra prieštaringa: **S** ⊢☐. Kitu atveju aibė **S** – nėra prieštaringa.

5. a) Jeigu disjunktų aibė S yra prieštaringa, tai ir formulių aibė B yra prieštaringa. Todėl iš prielaidų **Γ** SEKA išvada **F**.

5. b) Jeigu disjunktų aibė S nėra prieštaringa, tai ir formulių aibė B nėra prieštaringa. Todėl iš prielaidų **Γ** NESEKA išvada **F**.

Pastaba. Jeigu formulių sekoje, bandant išvesti tuščią disjunktą iš aibės S, pradedame gauti tuos disjunktus, kuriuos jau turime, tai vadinasi disjunktų aibė S – nėra prieštaringa.

Uždavinys:

Rezoliucijų metodu patikrinti ar iš prielaidų seka išvada:

3.d) $\boldsymbol{p} \to (\boldsymbol{q} \,\&\, r), (q \vee t) \to \neg r \vdash p \to (\neg t \vee u)$

Sprendimas:

1. Prielaidas ir išvadą surašome į formulių aibę B. Išvadą rašome su neigimu (t.y. tarsime kad teisingas yra faktas – „iš prielaidų NESEKA išvada“):

$\boldsymbol{B} = \{\boldsymbol{p} \to (\boldsymbol{q} \,\&\, r),\ (q \vee t) \to \neg r,\ \neg(\boldsymbol{p} \to (\neg \boldsymbol{t} \vee \boldsymbol{u}))\ \}$

2. Taikome elementarius pertvarkymus, kad kiekvienai formulių aibės B formulei gautume jai ekvivalenčią NKF.

$\mathbf{1)}\ \boldsymbol{p} \to (\boldsymbol{q} \,\&\, \boldsymbol{r}) \sim \neg \boldsymbol{p} \vee (\boldsymbol{p} \,\&\, q) \sim (\neg \boldsymbol{p} \vee \boldsymbol{q}) \,\&\, (\neg p \vee r)$

$\mathbf{2)}\ (\boldsymbol{q} \vee \boldsymbol{t}) \to \neg \boldsymbol{r} \sim \neg(\boldsymbol{q} \vee \boldsymbol{t}) \vee \neg \boldsymbol{r} \sim (\neg \boldsymbol{q} \,\&\, \neg t) \vee \neg r \sim \neg r \vee (\neg q \,\&\, \neg t) \sim (\neg \boldsymbol{r} \vee \neg \boldsymbol{q}) \,\&\, (\neg \boldsymbol{r} \vee \neg \boldsymbol{t})$

$\mathbf{3)}\ \neg(\boldsymbol{p} \to (\neg \boldsymbol{t} \vee \boldsymbol{u})) \sim \neg(\neg \boldsymbol{p} \vee (\neg \boldsymbol{t} \vee \boldsymbol{u})) \sim \boldsymbol{p} \,\&\, \neg(\neg \boldsymbol{t} \vee \boldsymbol{u}) \sim \boldsymbol{p} \,\&\, (\boldsymbol{t} \,\&\, \neg u) \sim \boldsymbol{p} \,\&\, t \,\&\, \neg u$

3. Mūsų naujoji formulių aibė B:

$\boldsymbol{B} = \{\ (\neg \boldsymbol{p} \vee \boldsymbol{q}) \,\&\, (\neg p \vee r),\ (\neg \boldsymbol{r} \vee \neg \boldsymbol{q}) \,\&\, (\neg r \vee \neg t),\ \boldsymbol{p} \,\&\, t \,\&\, \neg u\ \}$

4. Iš aibės B sukuriame disjunktų aibę S, keisdami visas konjunkcijas į kablelius (visos konjunkcijos yra išorinės):

$\boldsymbol{S} = \{\ (\neg \boldsymbol{p} \vee \boldsymbol{q}), (\neg \boldsymbol{p} \vee \boldsymbol{r}),\ (\neg \boldsymbol{r} \vee \neg \boldsymbol{q}), (\neg \boldsymbol{r} \vee \neg \boldsymbol{t}),\ \boldsymbol{p}, \boldsymbol{t}, \neg \boldsymbol{u}\ \}$

5. Sunumeruojame mūsų disjunktų aibės S formules:

$\boldsymbol{S} = \{\underset{1}{\neg \boldsymbol{p} \vee \boldsymbol{q}},\ \underset{2}{\neg \boldsymbol{p} \vee \boldsymbol{r}},\ \underset{3}{\neg \boldsymbol{r} \vee \neg \boldsymbol{q}},\ \underset{4}{\neg \boldsymbol{r} \vee \neg \boldsymbol{t}},\ \underset{5}{\boldsymbol{p}},\ \underset{6}{\boldsymbol{t}},\ \underset{7}{\neg \boldsymbol{u}}\}$

6. Bandome išvesti tuščią disjunktą iš aibės S taikydami atkirtos taisyklę:

8. $\neg p \vee \neg r$ *- pagal A.T. iš 1 ir 3* **(¬p ∨ q** ir **¬r ∨ ¬q** $\Longrightarrow$ **¬p** $\cancel{\vee q}$ **∨ ¬r** $\cancel{\vee \neg q}$ **)**

9. $\neg p \vee \neg q$ *- pagal A.T. iš 2 ir 3*

10. $\neg r$ *- pagal A.T. iš 4 ir 6*

11. r *- pagal A.T. iš 2 ir 5*

12. $\square$ *- pagal A.T. iš 10 ir 11*

7. Vadinasi iš disjunktų aibės S išvedamas tuščias disjunktas:

$S \Longrightarrow \square$

8. Todėl disjunktų aibės **S** – prieštaringa.

9. Kadangi **S** – prieštaringa, tai ir formulių aibių **B** prieštaringa.

10. Kadangi **B** – prieštaringa, tai reiškia jog iš prielaidų $\boldsymbol{\Gamma}$ seka išvada **F**.

Atsakymas:

iš prielaidų seka išvada.

Pratybos nr.4-1. Turingo mašinų variantai

Teorija:

Norint formaliai nagrinėti algoritmus kaip matematinius objektus, buvo pasiūlyti įvairūs skaičiavimo modeliai: Turing'o mašinos (beje, apibrėžtos gerokai anksčiau už realių kompiuterių atsiradimą), RAM (tiesioginės kreipties į atmintį) mašinos, dalinės rekursyviosios funkcijos, Markovo algoritmai.

Determinuota Turing'o mašina interpretuojama kaip mašina, turinti begalinę juostą, suskirstytą į ląsteles,ir palei juostą slenkančią skaitymo bei rašymo galvutę, kuri fiksuotu laiko momentu mato vieną juostos ląstelę ir gali perskaityti, koks abėcėlės ∑ ženklas ten įrašytas, vietoje jo įrašyti kitą ženklą bei pasislinkti per vieną ląstelę į kairę arba į dešinę, arba likti stovėti toje pat vietoje. Aibė {K,N,D} yra galimų galvutės postūmių aibė, kur K atitinka postūmį į kairę, D atitinka postūmį į dešinę, ir N reiškia, kad galvut˙e niekur nejuda.

Vienas iš Turing'o mašinos abėcėlės ženklų vadinamas tuščiu ("blank")ir žymimas b (b ∈ ∑). Jis yra skirtas tuščioms ląstelėms žymėti ir negali būti naudojamas pradinių duomenų bei tarpinių rezultatų, su kuriais dirba Turing'o mašina, kodavimui. Duomenys tarp savęs atskiriami vienu ar daugiau tuščių ženklų ar kitais sutartiniais abėcėlės ženklais. Dešiniau nuo paskutinių duomenų visa juosta vėl užpildyta tuščiais ženklais.

Vienajuostė Tiuringo mašina

Apibrėžimas. **Turing'o mašina**(toliau TM) *(arba* **1-juoste determinuota** Turing'o mašina*) vadiname rinkinį*

$$M = < \Sigma, Q, \delta, q_0, F>$$

kur:

- **∑** = $\{a_1, \ldots, a_n\}$ *yra baigtinė aibė, vadinama* abėcėle*; [0,1,...,b], kur b – tuščias elementas ‚blank'*
- **Q** = $\{q_0, q_1, \ldots, q_k\}$ *yra baigtinė* būsenų aibė;
- **δ** : $Q \times \Sigma \rightarrow Q \times \Sigma \times \{K,N,D\}$ *yra dalinė (ne visur apibrėžta)* perėjimų funkcija*;*

(K – į kairę, D – į dešinę, N – nejudėti, likti vietoje)

- $q_0 \in Q$ – pradinė būsena*; ir*
- **F** $\subseteq Q$ *yra* galutinių būsenų aibė.

δ – yra komanda, su reikšme, pvz. „$\delta(q_0,0) = (q_1, 1, D)$".

Ji reiškia, kad mašina būdama būsenoje„q_0" ir sutikusi skaičiu „**0**", turi pereiti į būseną „q_2", vietoje skaičiaus „**0**" įrašyti skaičių „**1**", ir pasislinkti per vieną simbolį **į dešinę pusę**.

Vienajuostė "vienos juostos - vienos galvutės" Tiuringo mašiną dar vadinama **standartine Tiuringo mašina**.

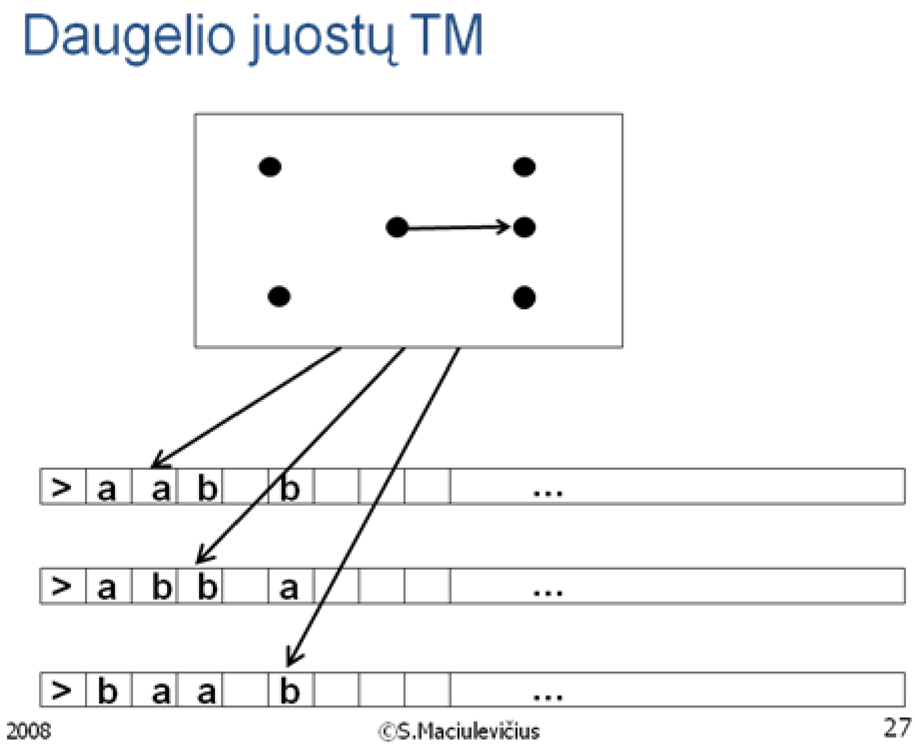
---

k-juostų Tiuringo mašina

Daugelio juostų TM turi keletą juostų ir keletą galvučių – po vieną kiekvienai juostai. Kiekviename žingsnyje visos galvutės skaito po vieną simbolį, priklausomai nuo perskaitytos aibės ir vidinės automato būsenos pakeičia simbolius ir perstumia galvutes į kairę ar dešinę.
Parodoma, kad tokia k-juostė TM gali būti sumodeliuota standartine TM, skaičiuojančia tą pačią funkciją.

Naudojant *k* juostų, Tiuringo mašiną taip pat galima aprašyti lygiai taip pat. Skirsis tik vienintelė komandos (**δ**) funkcija:

- **δ** : $Q \times \Sigma^{k} \rightarrow Q \times (\Sigma \times \{K,N,D\})^{k}$ - *yra dalinė (ne visur apibrėžta)* perėjimų funkcija*;*



Trijuostėje tiuringo mašinoje išskriamos 3 – duomenų, darbinė ir rezultatų juostos.
Pavyzdys:
3-juostės TM komandos pavyzdys:

$\delta(q_0,0,1,b) = (q_2,\quad 0,1,1,\quad K,D,N)$

// Šią komandą įvykdome, jeigu esame q0 būsenoje ir sutikome 0 – pirmoje juostoje, „1“ – antroje, ir
// „b“ – trečioje juostoje.
// Po komandos įvykdymo, būsena bus pakeista į q2, į 1-ąją juostą bus įrašytas „0“, į 2-ąją – „1“, į 3-iąją – „1“.
// 1-os juostos galvutė bus pastumta per vieną poziciją į kairę pusę,
// 2-os juostos galvutė – per vieną poziciją į dešinę pusę, 3-ios juostos galvutė – nejudės niekur(liks vietoje).

Pastaba:
TM begalybę žymi be galo dirbanti mašina, t.y. ji neturi patekti į galutinę būseną, o būdame ne galutinėje būsenoje gali vykdyti begalinį ciklą nejudant vietoje.

Deterministinė ir nedeterministinė Tiuringo mašinos

Apibrėžimas. Jei kiekvienai simbolio ir būsenos porai yra daugiausiai viena reikšmė veiksmų lentelėje, Tiuringo mašina vadinama *deterministine*, priešingu atveju – *nedeterministine*.
Deterministinės turingo mašinos komandos pavyzdys:

$\delta(q_0,0) = (q_2, 1, D)$

~~$\delta(q_0,0) = (q_3, 0, D)$~~ **// Šios eilutės naudoti negalime, jeigu siekiame kad mūsų mašina būtų deterministinė.**
Šią mašiną padarome nedeterministine, priskirdami daugiau kaip vieną reikmšmę veiksmų lentelėje:

$\delta(q_0,0) = (q_2, 1, D)$

$\delta(q_0,0) = (q_3, 0, D)$

Pastaba. Nedeterminuotos Turing’o mašinos paprastai naudojamos ne bet kokioms funkcijoms apskaičiuoti, o tik taip vadinamų *egzistavimo problemų* sprendimui, t.y. kai reikia tik atsakyti, ar sprendinys egzistuoja ar ne.

Tiuringo Mašinų uždavinys:

1) Parašyti 3-juoste Turingo mašina su abecele ***∑*** *=* ***{0, 1, b}****, kuri skaiciuoja funkcija (laikome, kad argumentas* ***x*** *atskirtas nuo argumento* ***y*** *lygiai vienu simboliu* ***b****):*
a) ***f(x, y) = x + y***

1. Pirmiausia mums reikia suprasti šios TM algoritmą.
2. Tada parašysime sprendimą vienajuostei determinuotąjai Turingo mašinai.
3. Užrašyti rezultatą vienajuosei TM.
4. Tada parašysime sprendimą trijuostei determinuotąjai Turingo mašinai.
5. Užrašyti rezultatą trijuostei TM.

TM uždavinio sprendimas(vienajuostei TM):

1. Sprendimo algoritmas:

Dvejetainio sumatoriaus algoritmas yra puikiai aprašytas Stasio Maciulevičiaus knygoje „*Kompiuterių teorija*“:

5.3.3.Dvejetainių skaičių sumatorius

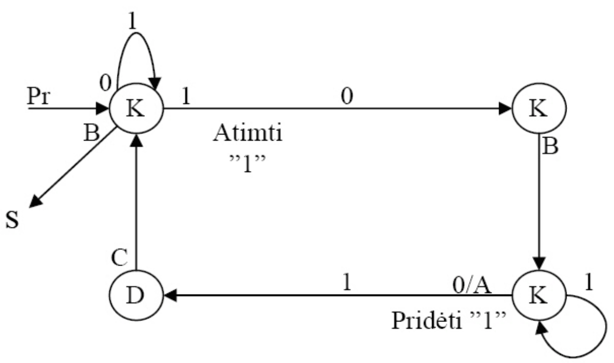
1.Reikia sudėti du skaičius *m* ir *n*, atvaizduotus dvejetainiu kodu (5.3.7 pav.).

2.Skaičiai turi vienodą skilčių skaičių, suma turi būti patalpinta pirmojo skaičiaus vietoje.

…	A	A	0	1	1	0	1	0	1	B	0	0	1	1	1	0	0	C	…
						m								*n*				Δ	

(5.3.7 pav. Pradinė juosta)

Dvejetainio sumatoriaus (T.M.) būsenų diagrama(5.3.8 pav.):



(čia S raidė žymi tą patį ką ir N – nejudėti, likti vietoje, Pr – pradinė būsena, B – „blank“ simbolis, K – postūmis į kairę, D – postūmis į dešinę pusę)

3.Mašinos darbo algoritmas labai paprastas – iš skaičiaus ***n*** atimamas **-1** ir prie ***m*** pridedamas 1. Procesas kartojamas, kol ***n*=0**. Po darbo juostoje (5.3.9 pav.) vietoje skaičiaus m suformuojamas rezultatas – ***m*+*n***.

…	A	A	1	0	1	0	0	0	1	B	1	1	1	1	1	1	1	C	…
						m+n				Δ									

5.3.9 Skaičiavimų rezultato juosta

2. TM uždavinio sprendimas(vienajuostei TM):

*Įdomumo dėlei – galime ir šiek tiek kitoks algortimas. Vietoje atimties, skaičių **n** galime tiesiog invertuoti, ir **pridedinėti** po **+1** tol, kol dešiniausias iš skaičiaus **n** kairės pusės **0** ženklas pavirs į vienetą.*

0. Pradedame ties tašku **C**:

...	A	A	0	1	1	0	1	0	1	B	0	0	1	1	1	0	0	C	...
						m								n				Δ	

0. Sutinkame ženklą „**blank**“ ir judame į kairę pusę:
$\delta(q_0,b) = (q_1,b,K)$;
$\delta(q_0,0) = (q_{99},b,N)$; // Klaida, pereiname į klaidos būseną ir sustojame
$\delta(q_0,1) = (q_{99},b,N)$; // Klaida, pereiname į klaidos būseną ir sustojame

1. Iš skaičiaus „**n**“ atimame **-1**.
a)Jeigu skaičiaus n paskutinis skaičius yra „**0**“, paskutinį skaitmenį keičiame į „**1**“ ir važiuojame į kairę pusę(keisdami visus nulius į vienetus(**0**->**1**)) tol, kol surandame pirmą „**1**“.
b)Jeigu skaičiaus n paskutinis skaičius yra „**1**“, paskutinį skaitmenį keičiame į „**0**“ (atimame **-1**) ir pereiname prie skaičiaus „**m**“.

1.a) Radome, kad paskutinis skaičiaus „**n**“ skaitmuo yra „**1**“, keičiame jį į „**0**“(atimame **-1**), nustamtome būseną „**q_3**“ ir einame į kairę pusę ieškodami „**blank**“ simbolio (**peršokame iškart į 4 etapą**).
$\delta(q_1,1) = (q_3,1,K)$;

1.b) Radome, kad paskutinis skaičiaus „**n**“ skaitmuo yra „**0**“, todėl įjungiame atimties režimą(nustatome būseną „**q_2**“) ir ieškome pirmojo vieneto.
$\delta(q_1,0) = (q_2,1,K)$;
$\delta(q_1,b) = (q_{99},b,N)$; // Klaida, pereiname į klaidos būseną ir sustojame

2.a) Antras nuo dešinės arba tolimesnis skaičiaus „**n**“ skaitmuo yra „**0**“, todėl jį keičiame į „**1**“ ir toliau važiuojame į kairę.
$\delta(q_2,0) = (q_2,1,K)$;

2.b) Keitėme skaičiaus „**n**“ skaičius iš „**0**“ į „**1**“ tol, kol galiausiai priėjome pabaigą(skaičiaus „**m**“ pradžią) – radome simbolį „**blank**“. (**Δ** – žymi esamą galvutės poziciją)

...	A	A	1	0	1	0	0	0	1	B	1	1	1	1	1	1	1	C	...
						$m+n$				Δ									

Vadinasi, mes jau suskaičiavome sumą. Keičiame būseną į „**q_9**“. (**peršokame prie etapo nr.11**)
$\delta(q_2,b) = (q_{14},b,K)$;

3. Puiku, radome „**1**“. Jį keičiame į „**0**“, išjungiame atimties režimą(nustatome būseną „**q_3**“), ir einame į kairę pusę ieškodami „blank“ simbolio.
$\delta(q_2,1) = (q_3,0,K)$;

4.a) Ieškome... (tai ne „**blank**“ simboliai, todėl paiešką tęsiame nekeisdami būsenos)
$\delta(q_3,0) = (q_3,0,K)$;
$\delta(q_3,1) = (q_3,1,K)$;

4.b) Puiku, radome „**blank**“ simbolį, įjungiame sumatoriaus režimą(nustatome būseną „q_{11}“), ir einame į toliau į kairę pusę.
$\delta(q_3,b) = (q_{11},b,K)$;

5.a) Radome, kad paskutinis skaičiaus „**m**“ skaitmuo yra „**0**“, keičiame jį į „**1**“(prie skaičiaus m pridedame **+1**), nustatome būseną į grįžimo, ir pakeičiame galvutės kryptį iš „**K**“ į „**D**“. (**peršokame iškart į 7 etapą**)
$\delta(q_{11},0) = (q_{13},1,D)$;
$\delta(q_{11},b) = (q_{99},b,N)$; // Klaida, pereiname į klaidos būseną ir sustojame

5. b) Radome, paskutinis skaičiaus „**m**“ skaitmuo yra „**1**“, todėl keičiame jį į „**0**“(prie skaičiaus „**m**“ pridedame **+1**), įjungiame pridėties režimą(nustatome būseną „q_{12}“) ir keičiame vienetus į nulius(**1**->**0**) tol, kol sutinkame **nulį**.
$\delta(q_{11},1) = (q_{12},0,K)$; // Paskutinis yra 1, keičiame į 0.
$\delta(q_{12},1) = (q_{12},0,K)$; // Keičiame visus 1->0, kol sutikinėjame tik vienetus.
$\delta(q_{12},b) = (q_{99},b,N)$; // Klaida, pereiname į klaidos būseną ir sustojame

6. Sutikome „**0**“, keičiame jį į „**1**“, išjungiame sumavimo režimą(nustatome būseną „q_{13}“) ir keičiame galvutės kryptį(grįžtame atgal į skaičiaus pradžią).
$\delta(q_{12},0) = (q_{13},1,D)$;

7. Ieškome skaičiaus m pradžios – „blank“ simbolio.
//$\delta(q_{13},0) = (q_{13},0,D)$; // variantas nereikalinga, nes praktiškai tokia situacija neįmanoma – nulio sutikti negalime.
$\delta(q_{13},1) = (q_{13},1,D)$; // Ieškome skaičiaus pradžios

8. Puiku, radome skaičiaus „**m**“ pradžią. Pereiname į grįžties būseną - einame į skaičiaus „**n**“ pradžią.
$\delta(q_{13},b) = (q_4,b,D)$;

9. Jau esame skaičiuje „**n**“, jeigu sutinkame „**0**“ arba „**1**“ – judame į dešinę pusę toliau. Jeigu sutinkame simbolį „**blank**“ – vadinasi radome skaičiaus „**n**“ pradžią.
$\delta(q_4,0) = (q_4,0,D)$; // Ieškome... (tai ne „blank“ simbolis, todėl ieškome toliau)
$\delta(q_4,1) = (q_4,1,D)$; // Ieškome... (tai ne „blank“ simbolis, todėl ieškome toliau)

10. Puiku, radome skaičiaus „**n**“ pradžią – „**blank**“ simbolį. Vadinasi ciklas pilnai apsisuko, ir grįžtame į buseną „q_1“ ir keičiame galvutės kryptį į **K**. (peršokame į 2 etapą).
$\delta(q_4,b) = (q_1,b,K)$;

11. (pabaigos etapas nr.1) Einame per skaičių „**m**“, (dabar tai jau skaičius tapęs „**m+n**“), ieškodami „**blank**“ simbolio – skaičiaus „**m+n**“ pradžios (paveikslėlyje jį žymi raidė **A**, **Δ** – žymi galvutės poziciją po to kai ji patenka į galutinę būseną – „q_{15}“).

$\delta(q_{14},0) = (q_{14},0,K)$;   // Ieškome... (tai ne „blank“ simbolis, todėl ieškome toliau)

$\delta(q_{14},1) = (q_{14},1,K)$;   // Ieškome... (tai ne „blank“ simbolis, todėl ieškome toliau)

12. (pabaigos etapas nr.2) Puiku, radome skaičiaus „**m+n**“ pabaigą, nustatome galutinę būseną į „q_{15}“, bei galvutę perkeliame į pirmąjį skaičiaus „**m+n**“ skaitmenį.

$\delta(q_{14},b) = (q_{15},b,D)$;

…	A	A	1	0	1	0	0	0	1	B	1	1	1	1	1	1	1	C	…
			Δ			$m+n$													

Atsakymas:

TM abėcėlė:

∑ = {0, 1, b}

TM baigtinė būsenų aibė Q yra:

Q = $\{q_0,q_1,q_2,q_3,q_4,q_{11},q_{12},q_{13},q_{14},q_{15},q_{99}\}$

TM galutinių būsenų aibė **F** yra:

F = $\{q_{15},q_{99}\}$

Patogiausia komandas užrašyti lentele, kurioje stulpeliai sunumeruoti abėcėlės ∑ simboliais, o eilutės – būsenomis:

	0	1	b
q_0	q_{99},b,N	q_{99},b,N	q_1,b,K
q_1	q_2,1,K	q_3,1,K	q_{99},b,N
q_2	q_2,1,K	q_3,0,K	q_{14},b,K
q_3	q_3,0,K	q_3,1,K	q_{11},b,K
q_4	q_4,0,D	q_4,1,D	q_1,b,K
q_{11}	q_{13},1,D	q_{12},0,K	q_{99},b,N
q_{12}	q_{13},1,D	q_{12},0,K	q_{99},b,N
q_{13}	------	q_{13},1,D	q_4,b,D
q_{14}	q_{14},0,K	q_{14},1,K	q_{15},b,D

<u>TM uždavinio sprendimas(3-juostei TM):</u>

Sprendimo Algoritmas:

susidedame du skaičius į dvi atskiras juostas ir nueiname į tų skaičių galą. Po to judėdami į kairę, po 1 skaitmetį darome sumavimą tarp 1-os ir 2-os juostų, rezultatą rašydami į 3-iąją juostą. Jeigu sumavimo rezultatas yra didesnis nei +1, patenkame į būseną „**mintyse +1**“. Iš šios būsenos grįžtame, kai skaičiaus mintyse nebelieka.

Sprendimas:

1. Pradedame ties pirma IŠ KAIRĖS <u>netuščia</u> lastele būsenoje q_0 ir TM: juostoje nr. 1:

...	A	A	0	1	1	0	1	0	1	B	0	0	1	1	1	0	0	C	...
						m								*n*					

2. Perkeliame skaičių X į 2-ąją juostą*(pirmoje juostoje judame į dešine tol, kol pasiekiame tuščią ląstelę b)*:

$\delta(q_0,\ 0, b, b) = (q_0,\ b, 0, b,\ K,K,N);$
$\delta(q_0,\ 1, b, b) = (q_0,\ b, 1, b,\ K,K,N);$
$\delta(q_0,\ b, b, b) = (q_1,\ b, b, b,\ K,N,N);$

3. Su pirmąja juosta nuvažiuokime į galą. Pasiekus pabaigą, 1-ojoje ir 2-oje juostoje skaitymo galvutes nustatykime ties pirma netuščia ląstele IŠ DEŠINĖS:

$\delta(q_1,\ 0, b, b) = (q_1,\ 0, b, b,\ K,N,N);$
$\delta(q_1,\ 1, b, b) = (q_1,\ 1, b, b,\ K,N,N);$
$\delta(q_1,\ b, b, b) = (q_2,\ b, b, b,\ D,D,N);$

4. Sudėkime du skaičius – skaičių X, esantį 2-oje juoste, ir skaičių Y – esantį pirmoje juostoje. Jeigu rezultatas yra 1+1, tai patenkame į specifinę būseną q_3 (atvejis - „+1 mintyse“):

$\delta(q_2,\ 0, 0, b) = (q_2,\ 0, 0, 0,\ D,D,D);$
$\delta(q_2,\ 0, 1, b) = (q_2,\ 0, 1, 1,\ D,D,D);$
$\delta(q_2,\ 0, b, b) = (q_2,\ 0, b, 0,\ D,N,D);$
$\delta(q_2,\ 1, 0, b) = (q_2,\ 1, 0, 1,\ D,D,D);$
$\delta(q_2,\ 1, 1, b) = (q_3,\ 1, 1, 0,\ D,D,D);$
$\delta(q_2,\ 1, b, b) = (q_2,\ 1, b, 1,\ D,K,N);$
$\delta(q_2,\ b, 0, b) = (q_2,\ b, 0, 0,\ N,D,N);$
$\delta(q_2,\ b, 1, b) = (q_2,\ b, 1, 1,\ N,D,N);$
$\delta(q_2,\ b, b, b) = (q_4,\ b, b, b,\ N,N,N);$

5. Jeigu patekome į specifinę būseną „+1 mintyse“:

$\delta(q_3,\ 0, 0, b) = (q_2,\ 0, 0, 1,\ D,D,D);$ *// **BIN:** 0+0+1 = 1, **grįžtame** atgal į būseną **q_2** – „<u>**mintyse nieko nėra**</u>“*
$\delta(q_3,\ 0, 1, b) = (q_3,\ 0, 1, 0,\ D,D,D);$ *// **BIN:** 0+1+1 = 0, toliau liekame būsenoje **q_3** – „+1 mintyse“*
$\delta(q_3,\ 0, b, b) = (q_2,\ 0, b, 0,\ D,N,D);$ *// **BIN:** 0+0+1 = 1, **grįžtame** atgal į būseną **q_2** – „<u>**mintyse nieko nėra**</u>“*
$\delta(q_3,\ 1, 0, b) = (q_3,\ 1, 0, 1,\ D,D,D);$ *// **BIN:** 1+0+1 = 0, toliau liekame būsenoje **q_3** – „+1 mintyse“*
$\delta(q_3,\ 1, 1, b) = (q_3,\ 1, 1, 0,\ D,D,D);$ *// **BIN:** 1+1+1 = 1, toliau liekame būsenoje **q_3** – „+1 mintyse“*
$\delta(q_3,\ 1, b, b) = (q_3,\ 1, b, 1,\ D,K,N);$ *// **BIN:** 1+0+1 = 0, toliau liekame būsenoje **q_3** – „+1 mintyse“*
$\delta(q_3,\ b, 0, b) = (q_2,\ b, 0, 0,\ N,D,N);$ *// **BIN:** 0+0+1 = 1, **grįžtame** atgal į būseną **q_2** – „<u>**mintyse nieko nėra**</u>“*
$\delta(q_3,\ b, 1, b) = (q_2,\ b, 1, 1,\ N,D,N);$ *// **BIN:** 0+1+1 = 0, **grįžtame** atgal į būseną **q_2** – „<u>**mintyse nieko nėra**</u>“*
$\delta(q_3,\ b, b, b) = (q_4,\ b, b, b,\ N,N,N);$ *// **BIN:** 0+0+1 = 1, patenkame į **galutinę** būseną „**q_4**“*

Atsakymas:

1. Aprašykime būsenų aibę:

Būsena	Būsenos Aprašymas
q_0	Perkeliame skaičių X iš pirmos juostos į antrąją
q_1	Pasiekiame pirmos juostos pabaigą(kur pasibaigia antrasis skaičius)
q_2	[MINTYSE: +0] Sudėkime 2 skaičius: X – iš antros juostos ir Y – iš pirmos juostos
q_3	[MINTYSE: +1] Sudėkime 2 skaičius: X – iš antros juostos ir Y – iš pirmos juostos
q_4	Galutinė būsena

2. TM abėcėlė:

$\sum$ = **{0, 1, b}**

3. TM baigtinė būsenų aibė Q yra:

Q = **{q_0,q_1,q_2,q_3,q_4 }**

4. TM galutinių būsenų aibė **F** yra:

F = {q_4 }

5. Įdomumo ir išsamumo dėlei, aprašykime šią daugiajuostę TM, lentele:

	0,0,b	0,1,b	0,b,b	1,0,b	1,1,b	1,b,b	b,0,b	b,1,b	b,b,b
q_0	---	---	q_0,b,0,b,K,K,N	---	---	q_0,b,1,b,K,K,N	---	---	q_1,b,b,b,K,N,N
q_1	---	---	q_1,0,b,b,K,N,N	---	---	q_1,1,b,b,K,N,N	---	---	q_2,b,b,b,D,D,N
q_2	q_2,0,0,0,D,D,D	q_2,0,1,1,D,D,D	q_2,0,b,0,D,N,D	q_2,1,0,1,D,D,D	q_3,1,1,0,D,D,D	q_2,1,b,1,D,K,N	q_2,b,0,0,N,D,N	q_2,b,1,1,N,D,N	q_4,b,b,b,N,N,N
q_3	q_2,0,0,1,D,D,D	q_3,0,1,0,D,D,D	q_2,0,b,0,D,N,D	q_3,1,0,1,D,D,D	q_3,1,1,0,D,D,D	q_3,1,b,1,D,K,N	q_2,b,0,0,N,D,N	q_2,b,1,1,N,D,N	q_4,b,b,b,N,N,N

Pratybos nr.4-2. Baigtiniai automatai

Teorija:

Skirtumas nuo TM. Nuo Tiuringo mašinų baigtiniai automatai(toliau B.A.) skiriasi tuo, kad sustoja ne patekę į specialią būseną, o sutikus „b“ (tuščią) ženklą.

Vaizdavimas. B.A. dažniausiai vaizduojamas orientuotu grafu. O būsena, į kurią patenkama visais nenumatytais atvejais, vadinama „juodąja skylė“.

17. Baigtinio automato kalba. Baigtinio automato kalba vadinsime aibę, abėcėlės Σ, žodžių su kuriais baigtinis automatas patenka į galutinę būseną.

Komandos formatas:

Apibrėžimas. **Baigtiniu automatu** *vadiname vienajuostės TM rinkinį:*

$$M = < \Sigma, Q, \delta, q_0, F>$$

kur:

- **∑** = $\{a_1, \dots, a_n\}$ *yra baigtinė aibė, vadinama* abėcėle; *[0,1,...,b], kur b – tuščias elementas ‚blank‘*
- **Q** = $\{q_0, q_1, \dots, q_k\}$ *yra baigtinė* būsenų aibė;
- **δ** : $Q \times \Sigma \to Q \times \Sigma$ *yra* perėjimų funkcija, pvz. „$\delta(q_0,0) = (q_1, 1)$“;
- $q_0 \in Q$ – pradinė būsena; *ir*
- **F** $\subseteq Q$ *yra* galutinių būsenų aibė.

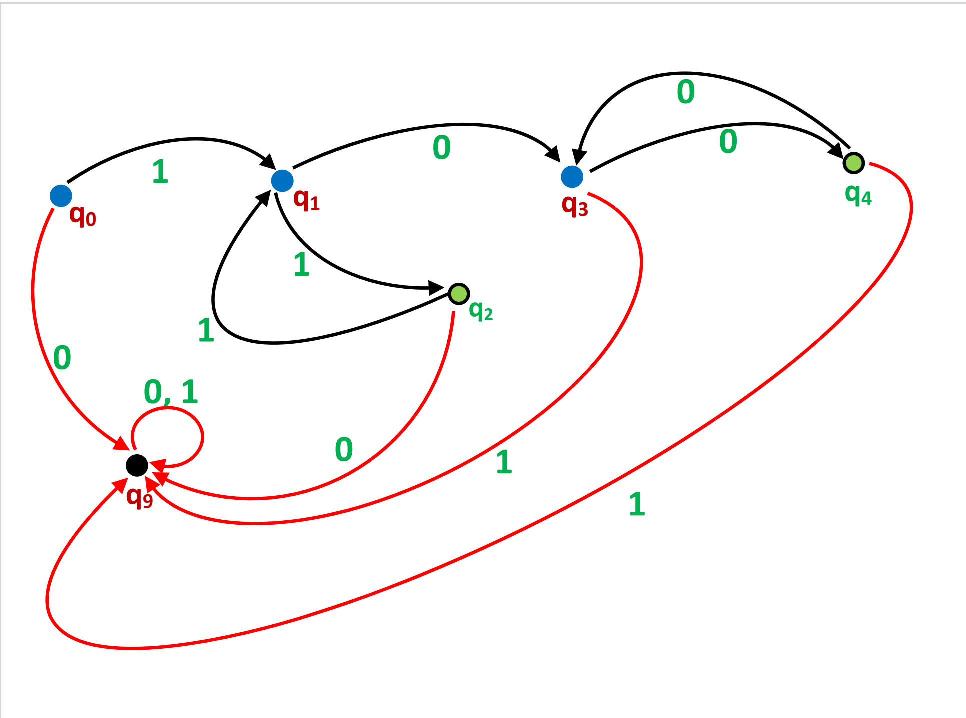
Uždavinys nr.1.

5. Rasti duotos kalbos baigtinį automatą.

d) A= $\{1^{2n+1}0^{2m}$, n=0,1,...,m, m=0,1,2,... }

Sprendimas:

Mūsų baigtinio automato kalba bus A = { 1, 111, 11111, 1111111 , ..., 100, 11100, 10000, 111110000, ... }



Pastaba I. Būsena q_9 – B.A. “juodoji skylė“.

Pastaba II. Kai B.A. grafas tampa labai didelis, vietoje visų raudonai pažymėtų kelių(kelių į „juodąją skylę“) galime nebraižyti, vietoje to užrašydami sakinį „Visi kiti veda į q_9“.

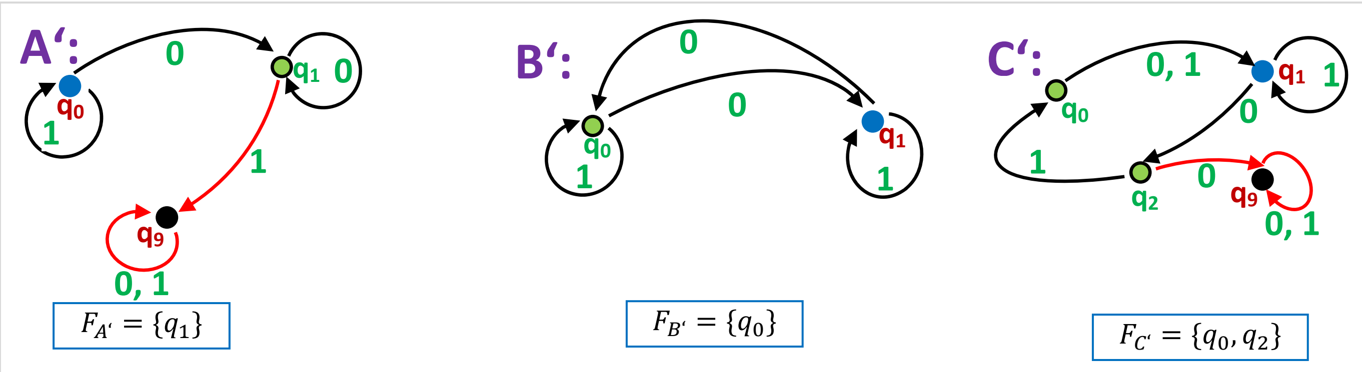
Pastaba III. B.A. „Juodoji skylė nėra įtraukiama į galutinių būsenų aibę.

Atsakymas.
$F = \{ q_2, q_4\}$
(B.A. galutinių būsenų aibė)

Uždavinys nr.2.

A, B, C – B.A. kalbos. A‘,B‘, C‘ – šių kalbų B.A. grafai. Rasti grafų A‘×B‘, bei A‘×B‘×C‘ dekarto sandaugą, bei parodyti, kad jei A,B,C – B.A. kalbos, tai ir A∪B, A⋂B, A∪B∪C, A⋂B⋂C – taip pat B.A. kalbos.

Duota:

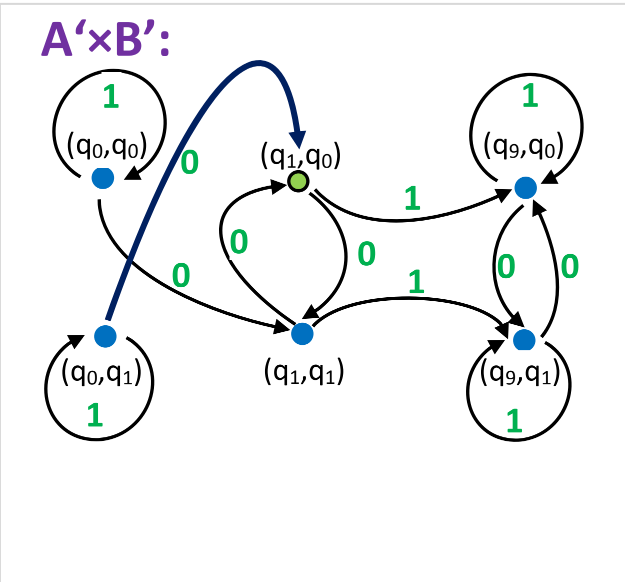


Apibr.: grafų G_1 ir G_2 dekarto sandauga vadiname grafą, kurio viršūnės yra visos įmanomos viršūnių porų kombinacijos iš grafo G_1 ir G_2 viršūnių aibių su žyme **a**, t.y.: $\{ \forall(q_i, q_j), q_i \rightarrow (a) \rightarrow q_j : q_i \in G_1, q_j \in G_2 \}$.

Svarbi pastaba: Ir paprasto B.A. grafo, ir Dekarto sandaugos B.A. grafo atveju visuomet iš kiekvienos viršūnės išeis NE MAŽIAU KAIP 2 lankai(su parametrais a=0 ir a=1) arba 1 lankas su parametru(a=0,1).

Sprendimas:

1) Lankas, su žyme „**a**“, tarp grafo $G_1 \times G_2 \times \dots \times G_r$ viršūnių $(q_{i_1}, q_{i_2}, \dots, q_{i_m})$ ir $(q_{j_1}, q_{j_2}, \dots, q_{j_m})$ ∃ t.t.t., jei ∃ lankai su žyme a, iš q_{i_k} į q_{j_k} grafe G_k, čia $k = 1, 2, \dots, r$.



Atsakymas:

$F_{A' \cap B'} = \{(q_1, q_0)\}$

$F_{A' \cup B'} = \{(q_0, q_0), (q_1, q_0), (q_1, q_1), (q_9, q_0)\}$

Žodis Z ∈ $(A \cap B)$ t.t.t., jei Z ∈ A ***ir*** Z ∈ B

Žodis Z ∈ $(A \cup B)$ t.t.t., jei Z ∈ A ***arba*** Z ∈ B .

2) Norėdami surasti grafą ***A‘×B’×C’***, tai galime padaryti 3 budais:
   - ***a)*** Išsirašyti visas galimas orientuotas jungtis tarp viršūnių. Tuomet beliks apsibraukti tuos variantus, kurie ∃ grafuose.
     *[**BŪDAS A:** Įprastas tipinio uždavinio sprendimo laikas: ~1 val. 20min.]*
   - ***b)*** Susidaryti lentelę, ir susižymėti „+“ bei „-“ ženklais, atitinkamai, jei lankas ∃arba ∄.
     *[**BŪDAS B:** Įprastas tipinio uždavinio sprendimo laikas: ~55 min.]*
   - ***c)*** **PAPRASČIAUSIAS BŪDAS:** pasinaudoti duomenimis iš jau anksčiau apskaičiuoto grafo **D’= A‘×B’** , ir skaičiuoti Dekarto sandaugą grafams **D‘** ir **C‘**, t.y. **E’= D‘×C’**.
     *[**BŪDAS C:** Įprastas tipinio uždavinio sprendimo laikas: ~15 min.]*

Parašysime sprendimą keliais būdais:

a) **Išrašykime visas galimas jungtis tarp viršūnių.** Paprastumo dėlei būseną ***q_i*** žymėkime tiesiog skaičiu „***i***“, o lanką su kryptini į skaičimi „***x***“ žymėkime „***>[x]>***“ bei „***<[x]<***“:

000>[0]>000; 000>[0]>001; 000>[0]>002; 000>[0]>009;
000>[1]>000; 000>[1]>001; 000>[1]>002; 000>[1]>009;

000>[0]>010; 000>[0]>011; 000>[0]>012; 000>[0]>019;
000>[1]>010; 000>[1]>011; 000>[1]>012; 000>[1]>019;
...
111>[0]>110; 111>[0]>111; 111>[0]>112; 111>[0]>119;
111>[1]>110; 111>[1]>111; 111>[1]>112; 111>[1]>119;

000<[0]<000; 000<[0]<001; 000<[0]<002; 000<[0]<009;
000<[1]<000; 000<[1]<001; 000<[1]<002; 000<[1]<009;

000<[0]<010; 000<[0]<011; 000<[0]<012; 000<[0]<019;
000<[1]<010; 000<[1]<011; 000<[1]<012; 000<[1]<019;
...
111<[0]<110; 111<[0]<111; 111<[0]<112; 111<[0]<119;
111<[1]<110; 111<[1]<111; 111<[1]<112; 111<[1]<119;

Taip išrašę visus variantus, apsibrauksime tuos elementus, kur ∃ kelias grafuose, bei pažymėsime tai naujajame grafe.

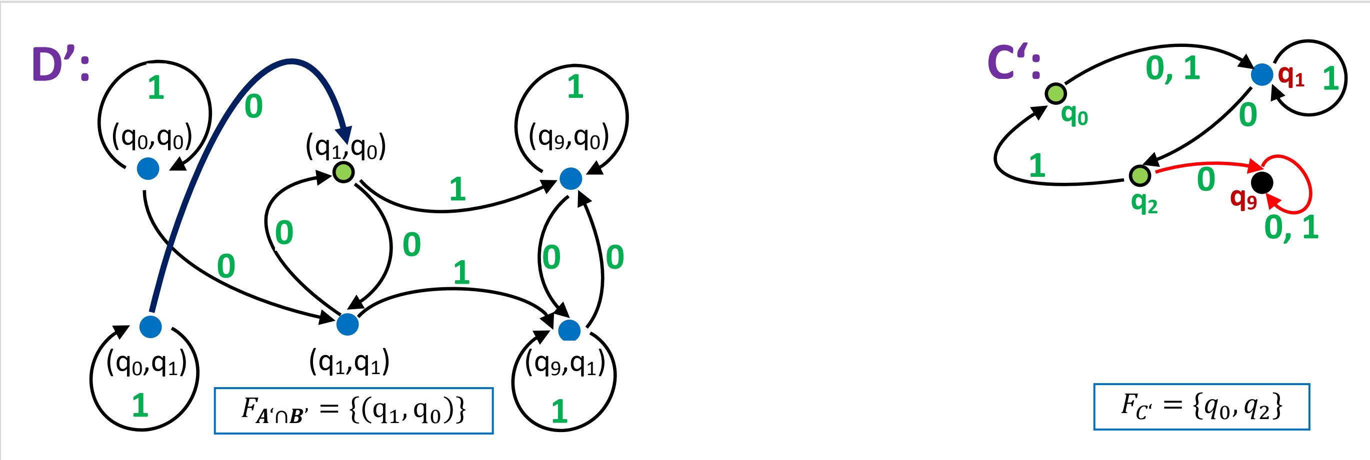
Tačiau visų variantų išrašymo procesas gali pareikalauti nemažai laiko.

b) **Sudarykime lentelę** *(jei lankas yra – žym. „+“, jei ne – žym. „-“)***:**

		q_X, q_Y, q_0						q_X, q_Y, q_1						q_X, q_Y, q_2						q_X, q_Y, q_9					
		X=0 Y=0	X=0 Y=1	X=1 Y=0	X=1 Y=1	X=9 Y=0	X=9 Y=1	X=0 Y=0	X=0 Y=1	X=1 Y=0	X=1 Y=1	X=9 Y=0	X=9 Y=1	X=0 Y=0	X=0 Y=1	X=1 Y=0	X=1 Y=1	X=9 Y=1	X=9 Y=0	X=0 Y=0	X=0 Y=1	X=1 Y=0	X=1 Y=1	X=9 Y=0	X= 9 Y= 1
q_0, q_0, q_0	0 →																								
	0 ←																								
	1 →																								
	1 ←																								
q_0, q_0, q_1	0 →																								
	0 ←																								
	1 →																								
	1 ←																								
...	...												...												
q_9, q_1, q_9	0 →																								
	0 ←																								
	1 →																								
	1 ←																								

Lentelė yra išties didelė, todėl spręsti lentelės būdu, trijų ar daugiau grafų Dekarto sandaugų uždavinius gali būti sudėtinga.

c) Nubražykime grafą E‘ – grafų D‘ ir C‘ Dekarto sandauga: **E’= D‘×C’**, kur **D’= A‘×B’** .



Sprendimo algoritmas:

1) Susirašykime visus lankus iš grafo C':

??0→(0)→??1	??0→(1)→??1
??1→(0)→??2	??1→(1)→??1
??2→(0)→??9	??2→(1)→??0
??9→(0)→??9	??9→(1)→??9

2) Susirašykime visus lankus iš grafo $D' = A' \times B'$:

00?→(0)→11?	01?→(0)→10?	10?→(0)→11?	11?→(0)→10?	90?→(0)→91?	91?→(0)→90?
00?→(0)→11?	…	…	…	…	…
00?→(0)→11?					
00?→(0)→11?					

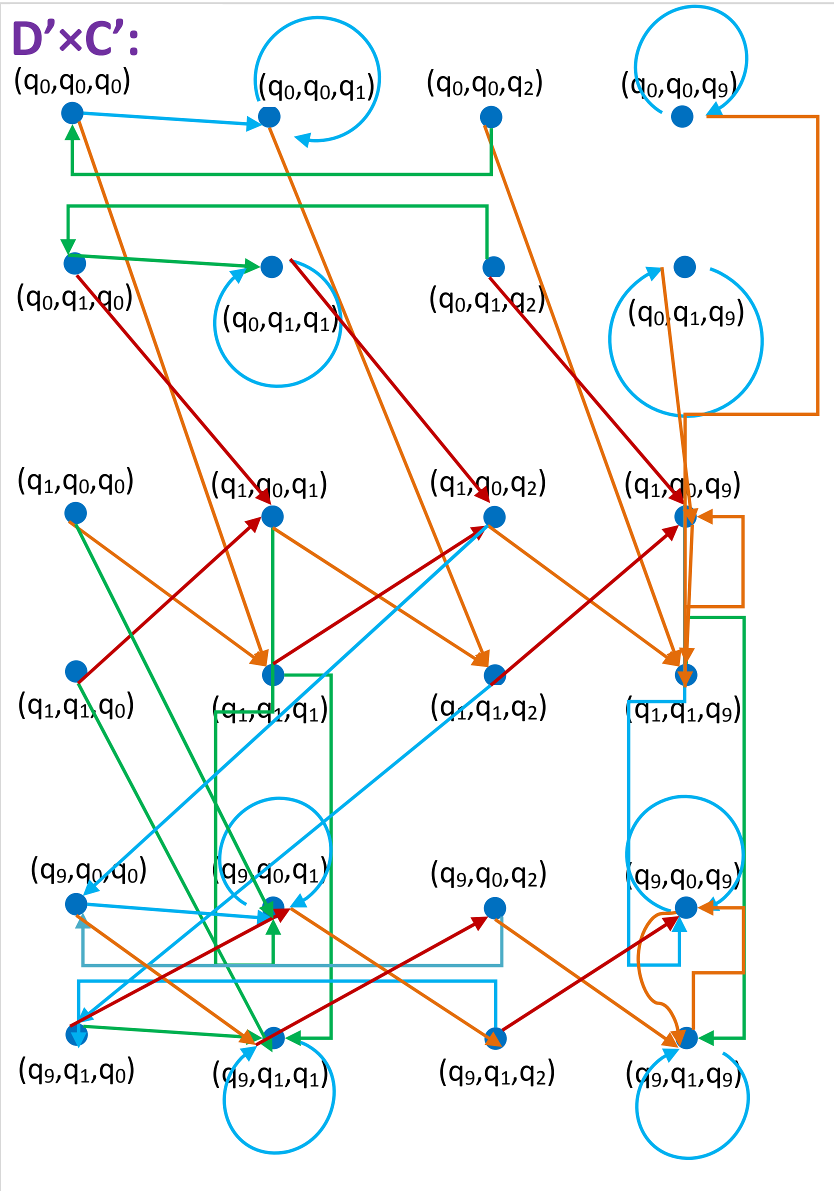
00?→(1)→00?	01?→(1)→01?	10?→(1)→90?	11?→(1)→91?	90?→(1)→90?	91?→(1)→91?
…	…	…	…	…	…

3) Sudarykime bendrą lankų $E' = D' \times C'$ lentelę, imdami ir grupuodami duomenis iš **1)** ir **2)** punktų:

000→(0)→111	010→(0)→101	100→(0)→111	110→(0)→101	900→(0)→911	910→(0)→901
001→(0)→112	011→(0)→102	101→(0)→112	111→(0)→102	901→(0)→912	911→(0)→902
002→(0)→119	012→(0)→109	102→(0)→119	112→(0)→109	902→(0)→919	912→(0)→909
009→(0)→119	019→(0)→109	109→(0)→119	119→(0)→109	909→(0)→919	919→(0)→909

000→(1)→001	010→(1)→011	100→(1)→901	110→(1)→911	900→(1)→901	910→(1)→911
001→(1)→001	011→(1)→011	101→(1)→901	111→(1)→911	901→(1)→901	911→(1)→911
002→(1)→000	012→(1)→010	102→(1)→900	112→(1)→910	902→(1)→900	912→(1)→910
009→(1)→009	019→(1)→019	109→(1)→909	119→(1)→919	909→(1)→909	919→(1)→919

4) Nubrėžkime grafą E‘=D’×C’, kur **D’= A‘×B‘** . Kad būtų paprasčiau, lankus su $\boldsymbol{a} = \mathbf{0}$ žymėkime **ORANŽINE** arba **BORDINE**, o $\boldsymbol{a} = \mathbf{1}$ – **MĖLYNA** arba **ŽALIA** spalva:



Atsakymas:

$$F_{A'\cap B'\cap C'} = \{$$
$$(\boldsymbol{q_1}, \boldsymbol{q_0}, \boldsymbol{q_0}), (\boldsymbol{q_1}, \boldsymbol{q_0}, \boldsymbol{q_2})$$
$$\},$$

$$F_{A'\cup B'\cup C'} = \{$$
$(q_0, \boldsymbol{q_0}, q_0)$, $(q_0, \boldsymbol{q_0}, q_1)$, $(q_0, \boldsymbol{q_0}, \boldsymbol{q_2})$, $(q_0, \boldsymbol{q_0}, q_9)$,
$(\boldsymbol{q_1}, \boldsymbol{q_0}, q_0)$, $(\boldsymbol{q_1}, \boldsymbol{q_0}, q_1)$, $(\boldsymbol{q_1}, \boldsymbol{q_0}, \boldsymbol{q_2})$, $(\boldsymbol{q_1}, \boldsymbol{q_0}, q_9)$,
$(\boldsymbol{q_1}, q_1, q_0)$,$(\boldsymbol{q_1}, q_1, q_1)$, $(\boldsymbol{q_1}, q_1, \boldsymbol{q_2})$, $(\boldsymbol{q_1}, q_1, q_9)$,
$(q_9, \boldsymbol{q_0}, q_0)$,$(q_9, \boldsymbol{q_0}, q_1)$, $(q_9, \boldsymbol{q_0}, \boldsymbol{q_2})$, $(q_9, \boldsymbol{q_0}, q_9)$
$$\}.$$

1) Žodis $Z \in (\boldsymbol{A} \cap \boldsymbol{B} \cap \boldsymbol{C})$ t.t.t., jei:
$\boldsymbol{Z \in A}$, $\boldsymbol{Z \in B}$ ***ir*** $\boldsymbol{Z \in C}$;
2) Žodis $Z \in (\boldsymbol{A} \cup \boldsymbol{B} \cup C)$ t.t.t., jei :
$Z \in A$ ***arba*** $Z \in B$, ***arba*** $Z \in C$.

Pratybos nr.5. λ-skaičiavimas

Teorija:

λ-skaičiavimas operuoja λ-skaičiavimo termais.

Termo pavyzdys - **x, y, x_3** . Šis kintamųjų rinkinys yra vadinamas termu.

29.1. Kintamasis yra termas.

29.2. Jei E_1, E_2 - termai, tai (E_1E_2) irgi termas (aplikacija).

29.3. Jei *x* - kintamasis, *E* - termas, tai λ*x*.*E* irgi termas (abstrakcija).

30. Termo redeksas ir jo santrauka. Termas, atrodantis kaip „**(λ*x*.*E*)*Y***“ vadinamas **redeksu**, o „***E*[*Y* / *x*]**“ - jo **santrauka**, kur E, Y – termai.

30.1. Apibrėžimas. Jei termas E = x, tai kintamojo x įeitis terme E yra laisva.

31. Termo β-redukcija. Termo β-redukcija vyksta tokiu būdu:

1. Ieškomas pirmas iš kairės redeksas ir jis pakeičiamas jo santrauka. Tai vadinama **redukcijos žingsniu** ir žymima simboliu ▷

2. Peržiūrime gautąjį termą, vėl iš kairės į dešinę ir, jei randame redeksą, keičiame jį jo santrauka.

32. Normalinis termas. Nenormalizuojamas termas.

32.1. Termas vadinamas **normaliniu**, jei neįmanoma jo daugiau redukuoti. t.y. jame nebėra redeksų.

32.2. Termas vadinamas **nenormalizuojamu**, jei jo neįmanoma redukuoti į normalinį termą. Pavyzdžiui, termas **(λ*x*.(*xx*))(λ*x*.(*xx*))** nenormalizuojamas:
$(\lambda x.(xx))(\lambda x.(xx)) \triangleright (\lambda x.(xx))(\lambda x.(xx)) \triangleright$...

33. λ-skaičiavimo loginės konstantos:

0 - λx.λy.y *(loginis TRUE / teisinga)*
1 - λx.λy.x *(loginis FALSE / klaidinga)*

natūralusis skaičius **k**(Church skaičius):
$\underline{0}$ = **λf. λx.x**
$\underline{1}$ = **λf. λx.f(x)**
$\underline{2}$ = **λf. λx.(f(f(x)))**
$\underline{k}$ = **λf. λx.(f^kx)**

Įsidėmėtina:
$\underline{0} = 0$
$\underline{1} \neq 1$
$\lambda z.zux = (\lambda z.z)u)x$
$(xy)u \neq x(yu)$
$\lambda x.\lambda y.(yz) = \lambda x.(\lambda y.(yz))$

Kas NĖRA termai:

λx.λy.	– ne termas *(tuščia termo λy galiojimo sritis)*
*λx**y***	– ne termas
*λ**ux**.z*	– ne termas (nex „ux“ – nėra kintamasis
*(λu.(ux))**λ**z*	– ne termas
*[**E_1**(λx.E)]Y*	– ne termas

Uždavinys nr.1:

*1) Funkcija **¬x** yra definuojama termu **λx.((x0)1)**. Apskaičiuoti: **¬1** .*

Sprendimas:

1. Taigi mūsų formulė atrodo taip:

f(x) = ¬x

2. Įsistatę gauname:

f(1) = ¬1

3. Mūsų termas **E = λx.((x0)1)** .

// λ-skaičiavime visus skaičius kuriuos apibrėžia ieškoma mūsų funkcija, statome į dešinę pusę paeiliui, pvz. Jeigu turime funkciją „f(x,y,z) = x+2y+3z” ir turime termą E, kuris apibrėžia mūsų funkciją, tai į mūsų termo formulę įstatome skaičius 0, 1 bei 2 tokiu būdu: **[** [**[** (λx.E)(__0__) **]** (__1__)] (__2__) **]** (čia E gali būti pvz. **((x0)1)**). Sprendžiama paeiliui – pirmiausia β-redukcija taikoma geltonam blokui, po to iš to kas liko, einama prieš žalio, ir galiausiai prie raudono bloko.

4. Įsistatome skaičių „1“(kaip ženklą Y, paryšintas raudonai) į mūsų formulę ir padarome redeksą (λx.E)Y

[λx.((x**0**)**1**)]**1** ⊳

// Šioje formulėje yra 3 loginės konstantos: 0, 1 ir 1. Galėjime iš karto tiesiog išsireikšti visas logines
// konstantas(0 ir 1) termais ir gauti: „ *⊳ [λx.((x **λx.λy.y**) **λx.λy.x**)] (**λx.λy.x**)* “, bet aiškumo ir paprastumo dėlei, to
// galime ir nedaryti, jeigu nereikia daryti spec. operacijų su veikimo sritimo (todėl pakeiskime tik
// vidinius 0 ir 1 termais).

6. Taikome β-redukciją(pirmas iš kairės **(λx.E)Y** junginys), keičiame visus nesuvaržytus(laisvus) „x‘us“ <u>terme E</u>, termu Y. Pabrėžiu tai, kad žiūrime būtent ir TIK į termą E.

Pastaba dėl suvaržymo. Šį kartą žiūrime į x‘us, nes mūsų redeksas prasideda λ**x**. , jeigu prasidėtų, pvz. λ**y** , tada dėl suvaržymo tikrintume visus y‘us.

Įsistatom termą Y vietoje (L) x‘o

$$\rhd [\underline{\lambda x.}\underbrace{(\,(x\,(\lambda x.\lambda y.y))\,(\lambda x.\lambda y.x)\,)}_{\text{Termas E}}]\,\underbrace{1}_{Y} \;\rhd\; [\cancel{\lambda x}.(\,(\,1\,(\lambda x.\lambda y.y))\,(\lambda x.\lambda y.x)\,)]\,\cancel{1} =$$

Laisvas (L)

7. Įsistatome vietoje „**1**“ jo reikšmę „**λx.λy.x**“. O naujo rasto redekso terme E, keičiame laisvąjį(nesuvaržytą) x‘ą termu Y(įsistatome).

$$= (\ (\underline{\lambda x.}\underbrace{\lambda y.x}_{\text{Termas E}})\ \underbrace{(\lambda x.\lambda y.y)}_{Y}\ )\ (\lambda x.\lambda y.x)\ \rhd\ (\cancel{(\lambda x.}\lambda y.\ (\lambda x.\lambda y.y)\cancel{)}\ \cancel{(\lambda x.\lambda y.y)})\ (\lambda x.\lambda y.x) =$$

Laisvas (L)

---

8. Įsistatome vietoje „**1**“ jo reikšmę „**λx.λy.x**“. O naujo rasto redekso terme E, keičiame laisvąjį(nesuvaržytą) x‘ą termu Y(įsistatome). Tuomet numetame išbrauktą tekstą ir keičiame termą „λx.λy.y“ jį atitinkančia konstanta „0“.

Suvaržytas (S) S

$$= (\ \underline{\lambda y.}\ \underbrace{(\lambda x.\lambda y.y)}_{\text{Termas E}}\ )\ \underbrace{(\lambda x.\lambda y.x)}_{Y}\ \rhd\ \cancel{(\ \lambda y.}\ (\lambda x.\lambda y.y)\ \cancel{)\ (\lambda x.\lambda y.x)} = \mathbf{0}$$

Uždavinys nr.2:

3) *Funkcija **x&y** yra definuojama termu **λx. λy.((xy)x)**. Apskaičiuoti **1&1**.*

Sprendimas:

Suskaičiuojame matematiškai:

f(x,y) = x & y => f(1,1) = 1&1 => 1

Suskaičiuojame λ-skaičiavimu:

1. *„x“ įeitis yra suvaržyta, jei patenka į „λx“ veikimo sritį. Šiuo atveju, terme E yra tik „λy“ veikimo sritis, todėl mūsų abu „x‘ai“ yra laisvi(nesuvaržyti).*

*Žaliai - termas E, geltonai – termas Y. **E**, **Y** ir **λx** sudaro **redeksą**.*

λx. λy.((xy)x) (1) (1) ⊳

Abu laisvi (L)

2. *“λx“ pasinaikina, „Y“ reikšmė įsistatato vietoje laisvų „x“.*

Gauname redekso santrauką:

⊳ λy.((1y)1) (1) =

3.1. *Įsistatome konstantos „1“ termą ir atliekame β-redukciją naujai rastam redeksui.*

Žaliai – E, raudonai – Y. „λy" pasinaikina, vietoje „y“ įsistatome „Y“ reikšmę.

3.2. *Galiausiai, vietoje „1“ termo įstatome jo konstantą „1“.*

$= \lambda y.((\,(\lambda x.\lambda y.x)y)1)\,(1) \;\rhd\; (\,\underbrace{(\lambda x.\lambda y.x)}_{\text{Termas E}}1)1 \Leftrightarrow ((\underbrace{1}_{\text{E}})1)1$

Laisvas (L)

*Kadangi „**λx.λy.x**“ yra termas E, o „**λx.λy.x = 1**“, tai savo formulę**(neformaliai)** galime parašyti taip:*

((1)1)1 ⇔ ((E)1)1 ⇔ (E1)1 **= 1**

Iš čia iškart matome, kad galime pritaikyti 35-ąjį apibrėžimą:

> $f(k_1,\dots,k_n) = k$ **(k ∈ N)** išplaukia, kad **$(\dots((Ek_1)k_2)\dots)k_n$** redukuojamas į normalinį termą **k**, o jei $f(k_1,\dots,k_n)$ neapibrėžta, tai **$(\dots((Ek_1)k_2)\dots)k_n$** nenormalizuojamas.

*Taigi, rėmėmės kad aibėje **$((E)k_1)k_2$** termas E redukuojamas į termą k. Iš to gavome, kad atsakymas yra „**1**“, ką ir reikėjo gauti.*

Pratybos 6. Porų numeravimas

Teorija:

Kantoro numeracija: Poros (x,y) numeris Kantoro numeracijoje yra funkcija $\boldsymbol{\alpha_2(x,y) = n}$.
N-tosios poros kairiojo nario funkcija yra $\boldsymbol{\pi_2^1(n)} = \mathbf{x}$.
N-tosios poros dešiniojo nario funkcija yra $\boldsymbol{\pi_2^2(n)} = \mathbf{y}$.

25 apibrėžimas. F-ja $\boldsymbol{\pi_n^i(k)}$ – gražina k-tojo rinkinio i-tąjį elementą(*čia n – elementų skaičius rinkinyje*).
O funkcija $\boldsymbol{\alpha_n(x_1, \dots, x_n)}$ gražina rinkinio $(x_1, \dots, x_n)$ numerį.

Lema(16 teiginys): Poros (x,y) numeris $\alpha_2(x,y)$ apskaičiuojamas naudojant funkciją:

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

Kairysis poros (x, y) narys randamas pagal formulę:

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

Trejetų, ketvertų ir t.t. numeravimas:
Bet kurio n-dedamųjų vektoriaus numeris apibrėžiamas rekursija:
$\alpha_n(x_1, \dots, x_n) = \alpha_2(x_1, \alpha_{n-1}(x_2, \dots, x_n))$
Pvz.: $\alpha_3(x_1, x_2, x_3) = \alpha_2(x_1, \alpha_2(x_2, x_3))$

Jei $\alpha_n(x_1, \dots, x_n) = \boldsymbol{k}$, tai $\pi_n^i(\boldsymbol{k}) = x_i$
Pvz.: $\alpha_7(x_1, \dots, x_4, \dots, x_7) = \mathbf{9}$, tai $\pi_7^4(\mathbf{9}) = x_4$

Pastaba. *Visi rinkiniai ir poros pradedamos numeruoti nuo nulio.*

Porų numeracijos pradžia. Kiekvienos porų grupės vienos poros elementų suma yra vienetu mažesnė nei tos porų grupės elementų kiekis. *Žemiau patekta pradinė dvejetų(poroje 2 elementai) numeracija:*

Pora:	(0,0),	(0,1)(1,0),		(0,2)(1,1)(2,0),			(0,3)(1,2)(2,1)(3,0),				(0,4)(1,3)(2,2)...		
Poros nr.:	0	1	2	3	4	5	6	7	8	9	10	11	12

Jei {...,(x,y),(u,v),...}, tai galioja:
(x+y < u+v) arba (x+y=u+v ir x<u)

Jei $\alpha_n(x_1, \dots, x_n) = \boldsymbol{x}$, tai $\pi_n^i(\mathbf{x}) = x_i$ randamas taip:

$x_n = \pi_2^Q(\pi_2^2(\pi_2^2(\dots(\pi_2^2(\mathbf{x}))\dots)))$, *čia* $Q = 2$, *jei tai paskutinis vektoriaus elementas, kitu atveju* $Q = 1$.

Pvz.: 4-elementų vektoriuje trečiasis elementas apskaičiojamas taip:
$x_3 = \pi_2^1(\pi_2^2(\pi_2^2(\pi_2^2(\mathbf{x}))))$

Uždavinys:

1) *Apskaičiuoti vektoriaus (1,3,0,2) numerį, naudojantis Kantoro numeracija.*
2) *Aprašyti šio vektoriaus elementus per* π.

Sprendimas:

1. Skaičiuojame „$\alpha_4(1,3,0,2)$“, taikydami formulę „$\alpha_n(x_1, \dots, x_n) = \alpha_2(x_1, \alpha_{n-1}(x_2, \dots, x_n))$“:
   $\alpha_4(1,3,0,2) = \alpha_2(1, \boldsymbol{\alpha_3(3,0,2)})$
2. Skaičiuojame „$\alpha_3(3,0,2)$“:
   $\boldsymbol{\alpha_3(3,0,2)} = \alpha_2(3, \boldsymbol{\alpha_2(0,2)})$
3. Apskaičiuojame „$\alpha_2(0,2)$“:
   $\boldsymbol{\alpha_2(0,2)} = ((0+2)^2 + 3*0 + 2) / 2 = (4+0+2) / 2 = \mathbf{3}$
4. Pradedame rekursiją. Į 2-ąją eilutę įsistatome apskaičiuotą „$\alpha_2(0,2)$“ reikšmę „**3**-*etą*“ ir apskaičiuojame „$\alpha_3(3,0,2)$“:
   $\boldsymbol{\alpha_3(3,0,2)} = \alpha_2(3, \boldsymbol{\alpha_2(0,2)}) = \alpha_2(3, \mathbf{3}) = ((3+3)^2 + 3*3 + 3) / 2 = (36 + 9 + 3) / 2 = \mathbf{24}$
5. Tęsiame rekursiją. Į 1-ąją eilutę įsistatome apskaičiuotą „$\alpha_3(3,0,2)$“ reikšmę „**24**“ ir apskaičiuojame „$\alpha_4(3,0,2)$“:
   $\alpha_4(1,3,0,2) = \alpha_2(1, \boldsymbol{\alpha_3(3,0,2)}) = \alpha_2(1, \mathbf{24}) = ((1+24)^2 + 3*1 + 24) / 2 = (625 + 3 + 24) / 2 = \mathbf{326}$
6. Belieka tik aprašyti vektoriaus elementus per π:
   $\pi_4^{\mathbf{1}}(\mathbf{326}) = 1$
   $\pi_4^{\mathbf{2}}(\mathbf{326}) = 3$
   $\pi_4^{\mathbf{3}}(\mathbf{326}) = 0$
   $\pi_4^{\mathbf{4}}(\mathbf{326}) = 2$

Atsakymas:

Vektoriaus **(1,3,0,2)** numeris aibėje yra **326**‘as, o aprašyti elementai yra: $\pi_4^{\mathbf{1}}(\mathbf{326}) = 1$, $\pi_4^{\mathbf{2}}(\mathbf{326}) = 3$, $\pi_4^{\mathbf{3}}(\mathbf{326}) = 0$, $\pi_4^{\mathbf{4}}(\mathbf{326}) = 2$.

Rekursyviosios funkcijos/aibės

(7-12 pratybos)

Trumpiniai:

PR – primityviai rekursyvi(-ios) **BR** – bendra rekursyvi(-ios) **DR** – dalinai rekursyvi(-ios) **R.S.** – rekursyviai skaiti(-ios) **B.F.** – bazinės funkcijos	**op./oper.** – operatorius*(kompozicijos, PR arba iteracijos)* **n-arg.** – n-argumentų **f-ja/f-jos** – funkcija(-ios) **t.t.t.** – tada ir tik tada

Simboliai:

$A \subseteq B$, $B \supseteq A$ – poaibis(*A≤B*) $A \subset B$, $B \supset A$ – tikrasis poaibis(*A<B*) $A \cup B$ – sąjunga(*A+B*) $A \cap B$ – sankirta(*A*B*) $A \setminus B$ – skirtumas(A-B) Σ (a,b,c) – suma(a+b+c)	$\exists$ – egzistuoja $\nexists$ – neegzistuoja $\forall$ – kiekvienam $\in$ – priklauso $\notin$ – nepriklauso $\equiv$ – ekvivalentus

D_f – apibrėžimo sritis(*daugeliu atveju tai x'o reikšmės*)

E_f – reikšmių sritis(*daugeliu atveju tai y'o reikšmės*)

Teorija:

Churcho tezė. Algoritmiškai apskaičiuojamų funkcijų aibė(klasė) sutampa su rekursyviųjų funkcijų aibe(klase).

Bazinės funkcijos(B.F.): konstanta 0, paskesnio nario funkcija s(x), bei projekcijų f-jos pr_n^i $(x_1, ..., x_n)$.

Operatoriai naudojami B.F.: Iš bazinių naujos funkcijos gaunamos pritaikius kompozicijos ir/ar primityviosios rekursijos(PR) operatorius.

Formalioji sistema. Ją sudaro bazinės funkcijos ir operatoriai, kurie taikomi turimoms funkcijoms.

Rekursyviosios funkcijos. Tai funkcijos, kurias galima gauti formalioje sistemoje.

Pastaba. Rekursyviai skaičios gali būti **TIK AIBĖS**, bet ne funkcijos.

PR funkcijų aibė. Pati mažiausia aibė, kuriai priklauso bazinės funkcijos ir kuri uždara kompozicijos ir PR atžvilgiu.

DR funkcijų aibė. Pati mažiausia aibė, kuriai priklauso bazinės funkcijos ir kuri uždara kompozicijos, PR bei minimizavimo atžvilgiu.

BR funkcijų aibė. Tai visur apibrėžtų DR funkcijų aibė.

Iš apibrėžimų išplaukia: **PR** $\subseteq$ **BR** $\subseteq$ **DR**, tačiau $\exists$ tokios BR funkcijos, kurios nėra PR funkcijos, todėl: **PR** $\subset$ **BR** $\subset$ **DR**.

Įsidėmėtinos PR funkcijos:

Bazinės f-jos: 0 ; s(x) = x+1 ; $pr_n^i(x_1, \ldots, x_n) = x_i$; $q(x) = x \dot{-} [\sqrt{x}]^2$

Operacijos: x + y ; $x \dot{-} y$; x*y ; k*x *(k – konstanta)* ; |x – y| ; [x/y]

Ženklo f-jos: sg(x) ; $\overline{sg}(x)$

Cantaro f-jos: $\alpha_n(x_1, \ldots, x_n)$; $\pi_n^i(\boldsymbol{x})$

Bei kompozicijos ir PR operatoriai.

Pastaba: *Daugyba/kvadratas yra PR, nes jas galime išreikšti per sudėti x*k = {x + x + …. (k-kartų)}*

Pastaba 2. Jeigu f(x) = sg(….) ir visos **sg()** viduje naudojamos funkcijos yra **PR**, tai ir pati f-ja f(x) yra **PR**.

Įsidėmėtinos DR funkcijos:

Dalinė atimtis: $x - y = \begin{cases} \mathrm{x-y}, & \text{kai } \mathrm{x \geq y} \\ \infty, & \text{kitu atveju} \end{cases}$

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

Dalinė šaknis: $\sqrt{x} = \begin{cases} \sqrt{x}, & \text{jei x yra } \mathbb{N} \text{ skaičiaus kvadratas} \\ \infty, & \text{kitu atveju} \end{cases}$

Bei minimizacijos operatorius.

Pastaba. Jeigu reikia parodyti kad kokia tai nors f-ja yra **DR**, tai vadinasi įrodome kad ji:

a) arba gauta iš f-jos **x–y** *(1 proc. įrodymo atvejų)*
b) arba gauta naudojantis minimizacijos operatoriumi *(99 proc. įrodymo atvejų)*

Pratybos 7. Primityviai rekursyviosios funkcijos.

Operatorių paaiškinimai:

5+[x/y] – [] operatorius reiškia kad bus gražinta tik sveikoji dalis.
rest(x, y) – rest() operatorius grąžina rezultato(skaičiaus) liekaną.
$\dot{-}$ - atimities su tašku operatorius reiškia, kad:

$$x \dot{-} y = \begin{cases} x - y, \text{kai } x \geq y \\ 0, \text{kitu atveju} \end{cases}$$

Ženklo funkcijos:

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

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

Ekvivalentumo apibrėžimas(≡):

Dvi formulės A($p_1, ..., p_n, q_1, ..., q_s$) ir B($p_1, ..., p_n, r_1, ..., r_u$) vadinamos ekvivalenčiomis(≡), jeigu su bet kuria aibės {$p_1, ..., p_n, q_1, ..., q_s, r_1, ..., r_u$} interpretacija **v** galioja lygybė **v**(A) = **v**(B).

Pavyzdžiai:

p -> q ≡ ¬p V q
¬¬p ≡ p

Teorija:

Visas 1-argumento PR f-jas galime išreikšti per bazines funkcijas (**B.F.**):

1. **0** ;
2. **s(x)** = **x+1** ;
3. $pr_n^i(x_1, ..., x_n) = x_i$; *(i = 1, 2, …., n)*
4. **q(x)** = $\mathbf{x \dot{-} [\sqrt{x}]^2}$;

x	q(x)
0	0
1	0
2	1
3	2
4	0
5	1
6	2
7	3
8	4

x	q(x)
9	0
10	1
11	2
12	3
13	4
14	5
15	6
16	0

Ir **3** operatoriai:
Sudėties: f(x) + g(x)
Kompozicijos: f(g(x))
Iteracijos: $f(x) = g(x)^I$, čia pvz. galime rašyti: $f(x) = g(x)^I$ ⇔ f(0) = 0 ; f(y+1) = g(f(y))

Taip pat turime du **PR** operatorius(funkcijas):
Jeigu:
1. **f(x,y)** = **h(x, y-1, f(x, y-1))** ; ir
2. **f(x,0)** = **g(x)** ;
tai funkcija „**f**“ yra gauta pagal **PR** operatorių iš funkcijų „**g**“ ir „**h**“.

Pastaba: Gali būti ir daugiau kintamųjų, pvz.: **g(x, y)**, tada **f(x, y, 0)** = **g(x, y)** .

Uždavinys:

*1) Funkcija **f(x, y)** yra gauta iš **PR** funkcijų **g(x)** = pr_3^2**(x, s(x), s(s(x)))** ir **h(x, y, z)** = **x · (y + 1) ∸ z** pritaikius primityviosios rekursijos operatorių. Rasti: **f(1, 3)**.*

Sprendimas:

0. Visų pirma reikia žinoti kas yra tie **PR** operatoriai. *(Teorija)*

*// F-jos „**f(x, y)**" y'o reikšmę – 3-etą išreiškiame kaip sumą „**2+1**" ir įsistatome į funkciją „**h**" pagal teoremą: „f(x,y) = h(x, y-1, f(x, y-1))".*

1. f(1, **3**) = f(1, **2**+1) = h(1, **2**, f(1, **2**))

*// Toliau paeiliui rekursiškai išsireiškiame vienetu mažesnes f-jos „**f(x, y)**" y'o reikšmes:*

2. f(1, **2**) = f(1, **1**+1) = h(1, **1**, f(1, **1**))

3. f(1, **1**) = f(1, **0**+1) = h(1, **0**, f(1, **0**))

*// Na o galiausiai priėję prie „**f(1, 0)**" jau galime ir formulę „**f(x,0)** = **g(x)**" pritaikyti:*

4. f(1, 0) = g(1)

*// Įsistatome 1-etą į „**g(x)**" formulę:*

5. f(1, 0) = g(1) = $\mathbf{pr_3^2}$**(1, s(1), s(s(1)))**

*// Apskaičiuojame „**s(1)**" remdamiesi B.F. nr.2 formule:*

6. s(1) = 1+1 = 2

*// Įsistatome į f-ją „**s(s(1))**" funkcijos "**s(1)** = **2**" reikšmę ir paskaičiuojame „**s(2)**" remdamiesi B.F. nr.2:*

7. s(s(1)) = s(2) = 2+1 = 3

*// Įsistatome į „pr_3^2(1, s(1), s(s(1)))" formulę apskaičiuotas „**s(1)**" ir „**s(s(1))**" reikšmes:*

8. f(1,0) = g(1) = pr_3^2(1, s(1), s(s(1))) = $\mathbf{pr_3^2}$**(1, 2, 3)**

// Apskaičiuojame „pr_3^2(1, 2, 3)" remdamiesi B.F. nr.3 (dvejetas yra antras elementas aibėje).

9. pr_3^2(1, 2, 3) = 2

// Papildome 8-ąją (kartu ir 4-ąją) eilutę:

10. f(1, 0) = g(1) = pr_3^2(1, s(1), s(s(1))) = pr_3^2(1, 2, 3) = **2**

*// Papildome 3-iąją eilutę suskaičiuota „**f(1, 0)**" reikšme:*

11. f(1, 1) = f(1, 0+1) = h(1, 0, f(1, 0)) = **h(1, 0, 2)**

*// Įsistatome į „**h(x, y, z)** = **x · (y + 1) ∸ z**" formulę „**x**", „**y**" ir „**z**" reikšmes ir apskaičiuojame „**h(1, 0, 2)**":*

12. h(1, 0, 2) = **1 · (0 + 1) ∸ 2 = 1 · 1 ∸ 2 = 1∸2**

*// Remiamės „**∸**" operatoriaus paaiškinimu ir apskaičiuojame „**1∸2**":*

13. h(1, 0, 2) = 1 · (0 + 1) ∸ 2 = 1 · 1 ∸ 2 = 1∸2 = **0**

*// Papildome 11-ąją (kartu ir 3-iąją) eilutę suskaičiuota „**h(1, 0, 2)** = **0**" reikšme:*

14. f(1, **1**) = f(1, **0**+1) = h(1, **0**, f(1, **0**)) = h(1, 0, 2) = **0**

------------- [Pradedame rekursiją] --------------

*// Į 2-ąją eilutę įsistatome apskaičiuotą „**f(1, 1)** = **0**" reikšmę:*

15. f(1, **2**) = f(1, **1**+1) = h(1, **1**, f(1, **1**)) = **h(1, 1, 0)**

*// Apskaičiuojame „**h(1, 1, 0)**":*

16. h(1, 1, 0) = **1 · (1 + 1)** ∸ 0 = 1 **· 2** ∸ 0 = **2**∸ 0 = **2**

*// Į 15-ąją (kartu ir 2-ąją) eilutę įsistatome apskaičiuotą „**h(1, 1, 0)** = **2**" reikšmę:*

17. f(1, **2**) = f(1, **1**+1) = h(1, **1**, f(1, **1**)) = h(1, 1, 0) = **2**

*// Į 1-ąją eilutę įsistatome apskaičiuotą „**f(1, 2)** = **2**"reikšmę ir paskaičiuojame „**h(1, 2, 2)**":*

18. f(1, **3**) = f(1, **2**+1) = h(1, **2**, f(1, **2**)) = **h(1, 2, 2) = 1 · (2 + 1) ∸ 2 = 1 · 3 ∸ 2 = 3 ∸ 2 = 1**

Atsakymas:

Taigi gavome atsakymą, kad: **f(1, 3)** = **1** .

Pratybos 8. Skaičiavimas Ackermann funkcijomis

33. Ackermann funkcijų lygybės ir savybės:

1. A(0, x) = x + 2 ;
2. A(1, 0) = 0 ;
3. A(y, 0) = 1, *kai y ≥ 2* ;
4. A(y+1, x+1) = A(y, A(y+1, x)) ;
5. A(n, x) ≥ 2^x, *kai n ≥ 2, x ≥ 1*;
6. A(n+1, x) ≥ A(n, x) + 1 ;
7. A(n, x+1) > A(n, x), *kai n,x ≥ 1* ;
8. A(n+1, x) ≥ A(n, x+1) .

Apibrėžiame funkcijas $B_n(a, x)$ kai $a \geq 2$:

$B_0(a, x) = a + x$,
$B_1(a, x) = a * x$,
$B_2(a, x) = a^x$.

Tai didėjančios funkcijos. $B_i(a, x) < B_j(a, x)$, kai $i < j$, pradedant kažkuriuo tai x_0.

Jos tenkina tokias lygybes:

$B_1(a, 1) = a$ $\quad$ $B_1(a, x + 1) = B_0(a, B_1(a, x))$,
$B_2(a, 1) = a$ $\quad$ $B_2(a, x + 1) = B_1(a, B_2(a, x))$

Pratęskime jas ($n \geq 2$):

$B_{n+1}(a, 1) = a$ $\quad$ $B_{n+1}(a, x + 1) = B_n(a, B_{n+1}(a, x))$

Tarkime, kad $B_{n+1}(a, 0) = 1$, kai $n \geq 1$.

Ackermann funkcijos variantu, kai $a = 2$, vadiname $A(n, x) = B_n(2, x)$.

34. Apibrėžimas. Egzistuoja **BR**, bet ne **PR** funkcija **h(x)** = **A(x, x)** ∈ **BR**, bet **h(x)** = **A(x, x)** ∉ **PR**

Uždavinys:

*1) Apskaičiuoti: A**(5, 3)**.*

Sprendimas:

// Parašome kintamuosius x,y kaip sumas x+1 ir y+1 ir taikome 33.4 formulę „A(y+1, x+1) = A(y, A(y+1, x))“:

1. A(5,3) = A(4+1, 2+1) = A(4, **A(5,2)**)

// Apskaičiuojame „A(5,2)“:

2. A(5,2) = A(4+1, 1+1) = A(4, **A(5,1)**)

// Apskaičiuojame „A(5,1)”. Tą galime padaryti dviem būdais:

a) Naudoti „A(n,x) = $B_n(2,x)$“ formulę

b) Tęsti esamą skaičiavimą taikant 33.1-33.4 formules

*// **Būdas A**. Taikome Ackermann funkcijos variantą, kai a = 2 : „A(n,x) = $B_n(2,x)$“*

3. A(5,1) = $B_5(2,1)$ = **2**

*// **Būdas B**. Toliau tęsiame esamą skaičiavimą:*

5.1. A(5, 1) = A(5+1, 0+1) = A(4, **A(5,0)**)

// Apskaičiuojame „A(5,0)“ taikydami 33.3 formulę „A(y, 0) = 1“, kai y ≥ 2:

3.2. A(5,0) = 1

*// Papildome 3-iąją eilutę apskaičiuoto A(5,0) reikšme „**1**“, bei tą pačią procedūrą kartojame:*

3.3. A(5,1) = A(5+1, 0+1) = A(4, **A(5,0)**) = *A(4,1) = A(3+1, 0+1) = A(3, **A(4,0)**) = **A(3,1)** = A(2+1, 0+1) = A(2, **A(3,0)**) = **A(2,1)** = A(1+1, 0+1) = A(1, **A(2,0)**)* = A(1,1) = A(0+1, 0+1) = A(0, **A(1,0)**)

*// Pritaikome 33.2 formulę „A(1,0) = **0**”, įsitatome reikšmę „0“ vietoje „A(1,0)“ į 5-ąją eilutę. Tada gavę naująją funkciją - „A(0,0)“, apskaičiuojame ją remdamiesi formule 33.1 - „A(0, x) = x + 2“:*

3.4. A(5,1) = <...> = A(1,1) = A(0+1, 0+1) = A(0, **A(1,0)**) = A(0,0) = **0** + 2 = **2**

// Na o dabar grįžtame atgal prie 2-osios eilutės, į kurią įsistatome apskaičiuotą „A(5,1)“ reikšmę „2“. Ir toliau tęsiame skaičiavimą:

4. A(5,2) = A(4+1, 1+1) = A(4, **A(5,1)**) = A(4,2) = A(3+1, 1+1) = A(3, **A(4,1)**)

// Kadangi „A(4,1)“ reikšmę jau kartą apskaičiavome 3.3-3.4. eilutėse – „2“ tai ją tiesiog įsistatome ir skaičiuojame toliau.

// Pastaba: „A(4,1)“ reikšmę taip pat galėjo apskaičiuoti pasinaudoję formule „A(n,x) = $B_n(2,x)$“.

5. A(5,2) = <...> = A(4,2) = A(3+1, 1+1) = A(3, **A(4,1)**) = A(3, 2) = A(2, A(3,1)) = A(2, 2) = $B_2(2, 2)$ = **4**

// Dabar įsistatome gautą rezultatą – „4“ į 1-ąją eilutę(rekursija):

6. A(5,3) = A(4+1, 2+1) = A(4, **A(5,2)**) = A(4, **4**) = A(3, **A(4,3)**)

// Apskaičiuojame „A(4,3)“:

7. A(4,3) = A(3, **A(4,2)**) = A(3,**4**) = A(2, **A(3,3)**)

// Suskaičiuojame „A(3,3)“ reikšmę. „A(3,2)“ buvo apskaičiuotas 5-oje eilute ir gautas atsakymas „4“:

8. A(3,3) = A(2, **A(3,2)**) = A(2, **4**) = $B_2(2,4) = 2^4 =$ **16**

// Įsistatome gautą „A(3,3)“ atsakymą – „16“ į 7-ąją eilutę(rekursija) ir apskaičiuojame „A(4,3)“:

9. A(4,3) = A(3, **A(4,2)**) = A(3,**4**) = A(2, **A(3,3)**) = A(2, **16**) = $B_2(2, 16) = \mathbf{2^{16}}$

// Dabar įsistatome gautą rezultatą – „2^{16}“ į 6-ąją(buvusią 1-ąją) eilutę(rekursija) ir gauname atsakymą:

10. A(5,3) = A(4+1, 2+1) = A(4, **A(5,2)**) = A(4, **4**) = A(3, **A(4,3)**) = A(3, $\mathbf{2^{16}}$) = $B_2(2, \mathbf{2^{16}}) = 2^{2^{16}} = \mathbf{2^{65536}}$

Atsakymas:

Ackermann funkcijos A**(5, 3)** rezultatas yra $\mathbf{2^{65536}}$.

Pratybos 9. Minimizacijos operatorius

Teorija:

41. Minimizacijos operatorius

Tarkime, yra *n* argumentų funkcija *f*. Apibrėžiame naują, taip pat *n* argumentų, funkciją, $g(x_1,...,x_n)$, kurios reikšmė lygi mažiausiam *y*, su kuriuo $f(x_1,...,x_{n-1},y) = x_n$.

Įrodyta: jeigu *f* yra net primityviai rekursyvi, *g* gali ir nebūti algoritmiškai apskaičiuojama funkcija. Todėl nusakydami naują funkciją *g*, mes privalome nurodyti ir metodą kaip ieškoti mažiausio *y*:

Jei $f(x_1,...,x_{n-1},0) = x_n$, tai funkcijos *g* reikšmė lygi 0, jei ne, tai tikriname, ar $f(x_1,...,x_{n-1},1) = x_n$.
Jei $f(x_1,...,x_{n-1},1) = x_n$, tai funkcijos *g* reikšmė lygi *1*, jei ne, tai tikriname ar $f(x_1,...,x_{n-1},2) = x_n$ ir t.t.

Funkcija *g* gali būti ir dalinė, t.y. su kai kuriomis argumentų reikšmėmis ji gali būti ir neapibrėžta, nes, pavyzdžiui, tokio *y*, tenkinančio aprašytąją lygybę, gali ir nebūti. Tačiau gali ir būti toks *m*, kad $f(x_1,...,x_{n-1},m) = x_n$, bet, jei su kuriuo nors $i < m$ funkcija $f(x_1,...,x_{n-1},i)$ neapibrėžta, tai ir *g* bus neapibrėžta.

Sakysime, kad *g* gauta pritaikius minimizacijos operatorių funkcijai *f*, ir žymime

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

Naudojantis minimizavimo operatoriumi, gaunama dalinę skirtumo funkcija $x - y = \mu_y(y + z = x)$.

Funkcija $f(x) = \mu_y(y - (x + 1) = 0)$ neapibrėžta su jokiu $x \in N$, nors kiekvienam *x* atsiras mažiausias *y*. Jis lygus *x+1*.

Apibrėžimas.

f(x_1,..., x_n) yra gauta pagal Minimizacijos operatorių iš **g(x_1,..., x_n)**,
jei **f(x_1, x_2, ..., x_n)** = **μ_y(g(x1,x2, …, xn-1, y) = x_n)**.
*Čia f-ja **g(x_1,x_2, …, x_{n-1}, y)** grąžina patį mažiausią **y** ∈ ℕ tenkinantį lygybę.*

Uždavinys:

Duota:

$f(x,y) = \mu_z(\, sg(\, z \dot{-} y\,) + \overline{s}(\, sg(x \dot{-} z)\,) = y)$

Rasti:

$f(2,3) = ?$

Sprendimas:

1. Įsistatome reikšmes:
2. $f(2,3) = \mu_z(sg(\, z \dot{-} 3\,) + s(\overline{sg}(\, 2 \dot{-} z\,)\,)=3)$
3. Tikriname visas z reikšmes paeiliui didėjimo tvarka, ieškodami pačio mažiausio z, su kuriuo teisinga lygybė.
   Pradedame tikrinti nuo 0:

   jei **z** = **0**, tai $sg(\,0 \dot{-} 3\,) + s(\,\overline{sg}(\,2 \dot{-} 0\,)\,) = 0 + s(0) = 1 \neq 3$

   jei **z** = **1**, tai $sg(\,1 \dot{-} 3\,) + s(\,\overline{sg}(\,2 \dot{-} 1\,)\,) = 0 + s(0) = 1 \neq 3$

   jei **z** = **2**, tai $sg(\,2 \dot{-} 3\,) + s(\,\overline{sg}(\,2 \dot{-} 2\,)\,) = 0 + s(1) = 2 \neq 3$

   jei **z** = **3**, tai $sg(\,3 \dot{-} 3\,) + s(\,\overline{sg}(\,2 \dot{-} 3\,)\,) = 0 + s(1) = 2 \neq 3$

   jei **z** = **4**, tai $sg(\,4 \dot{-} 3\,) + s(\,\overline{sg}(\,2 \dot{-} 4\,)\,) = 1 + s(1) = 1 + 2 = 3 \neq 3 \Longrightarrow f(2,3) = 4$

   Skaičius **4** yra pats mažiausias z su kuriuo teisinga lygybė.

> **Pastaba.** Jeigu tinkrindami gautume su viena z reikšme atsakymą „neapibrėžtą“, tai su visomis už šį skaičių didesnėmis reikšmėmis nebetikriname.

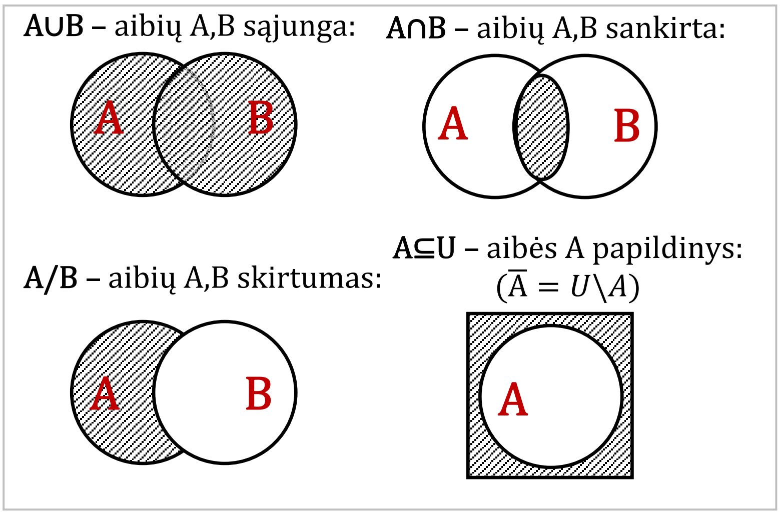
Atsakymas.

Skaičius **4** yra pats mažiausias z su kuriuo teisinga lygybė; $f(2,3) = 4$.

Pratybos 10. Rekursyvios ir rekursyviai skaičios aibės

Aibių tipai:

1. Tušios($\emptyset$) – **0**,
2. Baigtinės – {**0**, **1**, **...**, **x**} ∈ ($\mathbb{N} \setminus +\infty \cup 0$),
3. Natūraliųjų skaičių($\mathbb{N}$) – **ω**,
4. Sveikųjų skaičių($\mathbb{Z}$) – **π**,
5. Racionaliųjų skaičių($\mathbb{Q}$) – **η**,
6. Realiųjų skaičių($\mathbb{R}$) – **λ**.



***Rekursyvios*/*R.S.* aibės A charakteristinės funkcijos apibrėžimas*(bendru atveju)*:**

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

***Rekursyviai skaičios* aibės A charakteristinės funkcijos apibrėžimas *taip pat gali būti*:**

$$\chi_A(x) = \begin{cases} \mathbf{1}, & jei\ x \in A, \\ \infty, & jei\ x \notin A. \end{cases}$$

Teorija:

Rekursyvi aibė. Aibė A vadinama rekursyviąja, jei jos charakterigoji f-ja χ_A yra kuri nors visur apibrėžta rekursyvioji funkcija.

Kalbant apie aibes, vietoje (ne)rekursyvi aibė, dažnai sakome (ne)išsprendžiama aibė.

Apibrėžimas. Kiekvienas $\mathbb{N}$ aibės poaibis yra **skaitusis** arba **baigtinis**.

Apibrėžimas. ∃ begaliniai $\mathbb{N}$poaibiai, kurie nėra rekursyviai skaitūs.

***Apibrėžimas.* Rekursyviai skaiti(R.S.) aibė:**

I. Aibė yra rekursyviai skaiti, jei ji sutampa su kurios nors **DR** f-jos $\mathbf{D}_f$.
II. Aibė yra rekursyviai skaiti, jei ji sutampa su kurios nors **PR** f-jos $\mathbf{E}_f$.
III. Aibė A yra rekursyviai skaiti, jei ∃ tokia PR f-ja f(a,x), kad **f(a,x)** = **0**, turi sprendinį **x** t.t.t., kai **a ∈ A.**

25 teiginys. Visi 3 rekursyviai skaičios aibės apibrėžimai yra ekvivalentūs.

Rekursyvių ir rekursyviai skaičių aibių savybės:

I. Jei aibė A rekursyvi, tai ji ir rekursyviai skaiti.
II. Baigtinė aibė A = {$a_1, a_2, \ldots, a_n$} yra ir rekursyvi ir rekursyviai skaiti.
III. Jei aibė A yra R.S., bet nėra rekursyvi, tai jos papildinys $\bar{A}$ nėra nei rekursyvi, nei R.S. aibė.

Kitos savybės:

IV. Tuščia aibė yra rekursyviai skaiti.
V. Naturaliųjų skaičių aibė yra rekursyviai skaiti.
VI. Baigtinio skaičiaus rekursyviai skaičių aibių sąjunga ir sankirta yra rekursyviai skaičios aibės.

Pastebėjimas. Jei aibė A skaiti, ir abipusę vienareikšmę atitiktį su $\mathbb{N}$ galime nusakyti kuria nors PR f-ja h(x) (A = h(0), h(1), ...}), tai A taip pat ir rekursyviai skaiti.

33 teiginys. ∃ rekursyviai skaičios, bet ne rekursyvios aibės.

PR f-jų aibė. Pati mažiausia aibė, kuriai priklauso bazinės f-jos ir kuri uždara kompozicijos ir PR atžvilgiu.

DR f-jų aibė. Pati mažiausia aibė, kuriai priklauso bazinės f-jos ir kuri uždara kompozicijos, PR bei minimizavimo atžvilgiu.

BR f-jų aibė. Tai visur apibrėžtų DR funkcijų aibė.

Aibių sąjunga(AUB). Tai aibė, sudaryta iš visų **skirtingų** šių aibių elementų(priklauso bent vienai iš aibių).

Pvz.: {1, 2, 5} ∪ {0, 1, 3, 4, 5, 6} = {0, 1, 2, 3, 4, 5, 6}
Pastebėkime, kad tų pačių elementų du kartus nerašome.

Aibių sankirta(A ∩ B). Tai aibė, sudaryta tik iš tų šių aibių elementų, kurie priklauso abejoms aibėms.
Pvz.: {1, 2, 5} ∩ {0, 1, 3, 4, 5, 6} = {1, 5}

Aprėžta funkcija. Funkcija f(x) yra aprėžta(apibrėžta) (funkcijų reikšmių) intervale I, jei ∃ egzistuoja tokie skaičiai m ir M, kad visiems x iš I galioja nelygybė **m** ≤ f(x) ≤ **M**.

T.y. su bet kuria x'o reišme, funkcijos f(x) rezultatas nekerta intervalo [m,M] rėžių.

Churcho tezė. Algoritmiškai apskaičiuojamų funkcijų klasė sutampa su rekursyviųjų funkcijų klase.

Churcho tezė(2-a formuluotė). Jeigu funkcija yra apskaičiuojame **TM**, ji yra daliniai rekursyvi(**DR**).

Apibrėžimas. Funkcija f(x) yra **BR**, jei ji yra apbrėžta su visais argumentais(visur apibrėžta).

Uždavinys nr.1:

11) Aibės A_1, ..., A_n yra rekursyvios. Įrodykite, kad jų sąjunga bei sankirta yra taip pat rekursyvios aibės.

Sprendimas:

1. Čia visų pirma remsimės 1-ąja rekursyviųjų ir R.S. aibių savybe – *„Jei aibė A rekursyvi, tai ji ir rekursyviai skaiti“*. Taigi galime teigti kad mūsų aibės A_1, ..., A_n yra ir rekursyviai skaičioms.
   Pastaba. *Tačiau, atvirkštiniu atveju to teigti negalėtume, nes prieštarautume teoremai, sakančiai, kad „∃ rekursyviai skaičios, bet ne rekursyvios aibės“.*

2. Dabar prisiminkime III-iąją *„Aibė A yra R.S., jei ∃ tokia PR f-ja f(a,x), kad **f(a,x) = 0**, turi sprendinį **x** t.t.t., kai **a ∈ A**“*. Vadinasi, galime padaryti išvadą, kad egzistuoja tokios PR f-jos $f_i(a,x)$, kad $f_i(a,x)$ = ***0*** sprendinį turi tiktais tokiu atveju, kai ***a*** yra viena iš aibės A_i elementų reikšmė($a \in A_i$).

3. Taip pat prisiminkime taisyklę, kad Visas 1-argumento PR f-jas galime išreikšti per bazines funkcijas bei sudėties, kompozicijos, iteracijos bei PR operatorius.

4. Bei sugrįžkime prie porų numeravimo ir **Kantoro numeracijos funkcijų:**
   Poros (x,y) numeris Kantoro numeracijoje yra funkcija $\alpha_2(x, y) = n$.
   N-tosios poros kairiojo nario funkcija yra $\pi_2^1(n) = x$.
   N-tosios poros dešiniojo nario funkcija yra $\pi_2^1(n) = y$.

5. Taigi, remdamiesi 2-4 punktais, aibių A_1, ..., A_n sąjungos atveju konstruojame tokią PR f-ją:
   $f(a,x) = f_1(a, \pi_n^1(x)) * f_2(a, \pi_n^2(x)) * \ldots * f_n(a, \pi_n^n(x))$
6. Su reikšmėmis $x_1^0, x_2^0, \ldots, x_n^0$ lygybės $f(a, x_i^0) = 0$, *kur* $i = 1, 2, \ldots, n$, galioja t.t.t., kai ∃ toks ***a***, kuris priklauso visų A_1, ..., A_n aibių sąjungai($a \in A_1 \cup A_2 \cup \ldots \cup A_n$).
   Pvz.: *Jeigu* $x^0 = \alpha_n(x_1^0, x_2^0, \ldots, x_n^0)$, *tada* $f(a, x^0) = 0$.

7. Remdamiesi 2-4 punktų apibrėžimais, aibių A_1, ..., A_n sankirtos atveju konstruojame tokią PR f-ją:
   $g(a,x) = g_1(a, \pi_n^1(x)) + g_2(a, \pi_n^2(x)) + \ldots + g_n(a, \pi_n^n(x))$
8. Su reikšmėmis $x_1^0, x_2^0, \ldots, x_n^0$ lygybės $g(a, x_i^0) = 0$, *kur* $i = 1, 2, \ldots, n$, galioja t.t.t., kai ∃ toks ***a***, kuris priklauso visų A_1, ..., A_n aibių sankritai($a \in A_1 \cap A_2 \cap \ldots \cap A_n$).
   Pvz.: *Jeigu* $x^0 = \alpha_n(x_1^0, x_2^0, \ldots, x_n^0)$, *tada* $g(a, x^0) = 0$.

9. ***Teiginys įrodytas.***

Uždavinys nr.1:

E) Parodyti, kad $f(x,y,z) = (z \dot{-} (x+y))! \in PR.$

Sprendimas:

1) Tariam, kad $f(x,y,z)$ gauta iš funkcijų g ir h naudojant PR operatorių, todėl ir $f \in PR$:
$$f(x,y,0) = g(x,y)$$
$$f(x,y,z+1) = h(x,y,z,f(x,y,z))$$

2) Kadangi $f(x,y,0) = (0 \dot{-} (x+y))! = 0! = 1$, todėl $g(x,y) = 1.$

3) $g(x,y) \in PR$, nes gauta iš bazinės $s(x)$:
$$g(x,y) = s(0) = 0 + 1 = 1$$

4) $$f(x,y,z+1) = f(x,y,z) \cdot \left(\left((z+1) \dot{-} (x+y)\right) + \overline{sg}\left(\overbrace{(z+1) \dot{-} (x+y)}^{\begin{cases} 1 \;, & jei = 0 \\ 0 \;, & jei > 0 \end{cases}} \right) \right)$$

jei $f(x,y,z)$
jei $f(x,y,z) \sim 0!$, tai **1**,
jei $f(x,y,z) \sim 1!$, tai **1**,
jei $f(x,y,z) \sim 2!$, tai **2**,
jei $f(x,y,z) \sim 3!$, tai **6**,
jei $f(x,y,z) \sim 4!$, tai **24**,
jei $f(x,y,z) \sim 5!$, tai **120**;

$\left((z+1) \dot{-} (x+y)\right) + \overline{sg}\left((z+1) \dot{-} (x+y)\right)$
jei $f(x,y,z) \sim 0!$, tai *0+1*=**1**,
jei $f(x,y,z) \sim 1!$, tai *1+0*=**1**,
jei $f(x,y,z) \sim 2!$, tai *0+0*=**2**,
jei $f(x,y,z) \sim 3!$, tai *0+0*=**3**,
jei $f(x,y,z) \sim 4!$, tai *0+0*=**4**,
jei $f(x,y,z) \sim 5!$, tai *5+0*=**5**;

5) $$h(x,y,z,w) = w \cdot \left(\left((z+1) \dot{-} (x+y)\right) + \overline{sg}\left((z+1) \dot{-} (x+y)\right) \right)$$

Atsakymas:

$s(x) \in PR$ – nes bazinė,
$g \in PR$ – nes gauta iš bazinės $s(x) \in PR$,
$f \in PR$ – nes išreikšta per **PR** operatorių,
$h \in PR$ – nes gauta iš žinomų PR funkcijų $\boldsymbol{x \cdot y}$, $\boldsymbol{x + y}$, $\boldsymbol{\overline{sg}(x)}$, $\boldsymbol{x \dot{-} y}$ pritaikius kompozicija.

Uždavinys nr.3:

Duota: *f(x,y)* = $\begin{cases} 1, & jei\ x-y>0 \\ 2, & jei\ x-y=0 \\ \infty, & kitu\ atveju \end{cases}$

Kuris teiginys apie duotą funkciją yra teisingas:

a) *f(x,y) ∈ DR, bet f(x,y) ∉ BR.*
b) *f(x,y) ∈ PR, bet f(x,y) ∉ BR.*
c) *f(x,y) ∈ BR ir f(x,y) ∈ DR.*
d) *f(x,y) yra R.S., bet f(x,y) ∉ DR.*
e) *f(x,y) yra R.S. ir f(x,y) ∈ PR.*
f) *f(x,y) yra R.S., ir f(x,y) ∈ PR, f(x,y) ∈ BR, f(x,y) ∈ DR.*
g) *f(x,y) nėra nei R.S., nei PR, nei BR, nei DR.*

Sprendimas:

1. Funkcija $\boldsymbol{x} - \boldsymbol{y} \in \mathrm{DR}$, todėl ji yra apskaičiuojama TM.
   O kadangi ji yra apskaičiuojama TM, tai pagal Chuch‘o tezę „f-ja yra DR, jei ji apskaičiuojama TM. Taigi variantai *d)* ir *g)* iškart atkrenta, t.y. tie kur vienas iš teiginių yra *f(x,y) ∉ DR*.
2. Funkcija $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) \notin \mathrm{BR}$, nes ji nėra nėra visur apibrėžta, t.y. turime atsakymą ∞. Tačiau tai yra DR funkcija. Taigi variantai *c)* , *f)* ir *g)(dar kartą)* išsibraukia.
3. Jeigu funkcija nėra BR, o iš PR, BR ir DR apibrėžimų išplaukia, kad **PR ⊂ BR ⊂ DR**. Tai atkrenta ir variantai su PR. Šiuo atveju toks variantas yra *b)*.
4. Rekursyviai skaiti gali būti TIK AIBĖ, bet ne funkcija, todėl variantai *d)*, *e)* ir *f)(dar kartą)* atkrenta.
5. *a)*, ~~*b)*~~, ~~*c)*~~, ~~*d)*~~, ~~*e)*~~, ~~*f)*~~, ~~*g)*~~. Taigi liko variantas ***a)*** – jis ir yra teisingas atsakymas.

Teiginių įrodymai

4 teiginys. (Dedukcijos teorema) $\Gamma \vdash A \to B$ tada ir tik tada, kai $\Gamma, A \vdash B$.

6 teiginys. Jei disjunktų dedukcinėje sistemoje iš aibės **S** galima išvesti disjunktą C ir jis nėra įvykdomas, tai aibė **S** – prieštaringa.

7 teiginys. Jei disjunktų aibė **S** – prieštaringa, tai iš **S** išvedamas tuščias disjunktas$\square$.

10 teiginys. Jei $\mathbf{A_1}$,$\mathbf{A_2}$ – B.A. kalbos, tai $A_1 \cup A_2$, $A_1 \cap A_2$ – taip pat **B.A.** kalbos.

16 teiginys. Baigtinumo problema neišsprendžiama.

18 teiginys. Standartinių TM aibė yra skaiti.

23 teiginys. Visi 3 R.S. aibės apibrėžimai yra ekvivalentūs. *(tik dalinis įrodymas)*

24 teiginys. Jei aibė **A** yra R.S., bet nėra rekursyvi, tai jos papildinys $\overline{A}$ nėra, nei rekursyvi, nei R.S. aibė.

26 teiginys. ∃ **BR**, bet ne **PR** funkcijos.

27 teiginys. a) Visų n-arg. **PR** f-jų universalioji $F(x_0, x_1, ..., x_n) \notin$ **PR**.
b) Visų n-arg. **BR** f-jų universalioji $F(x_0, x_1, ..., x_n) \notin$ **BR**.

28 teiginys. Visų 1-arg. **PR** funkcijų aibe ∃ universalioji **BR** klasė.

29 teiginys. $D^{n+1}(x_0, . . . , x_n) = D(x_0, \alpha_n(x_1, . . . , x_n))$ yra visų n-arg. **PR** funkcijų universalioji.

30 teiginys. Visų n-arg. **DR** funkcijų aibe ∃ universalioji $\widetilde{D}^{n+1}(x_0, x_1, ..., x_n)$.

31 teiginys. Visų n-arg. **DR** funkcijų aibės universalioji $\widetilde{D}^{n+1}(x_0, x_1, ..., x_n)$ neturi pratęsimo.

32 teiginys. ∃ rekursyvios, bet ne rekursyviai skaičios aibės.

4 teiginys. *(Dedukcijos teorema)* $\Gamma \vdash A\text{->}B$ tada ir tik tada, kai $\Gamma, A \vdash B$.

Įrodymas.

(prielaida) Tarkim, kad $B_1, \ldots, B_m$ $(B_m = B)$ yra f-lės išvedimas iš prielaidų **Γ**, **A**. Juo remiantis bandysime išvesti f-lę **A → B** iš prielaidų **Γ.** B_i $(1 \le i \le m)$ pakeitę į **A → B**$_i$, sudarome seką:

$$A \to B_1, \ldots, A \to B_i, \ldots, A \to B_m \qquad \textit{(žym.: sek}_1\textit{)}$$

Ši seka nebūtinai yra išvedimas, t.y. ∀ sekos narys nebūtinai yra aksioma, prielaida iš **Γ**, ar gautas pagal **MP** iš kairėje stovinčių. ∀ **sek**$_1$ narį keičiame tokia seka, kad gautoji seka būtų f-lės **A → B** išvedimas iš **Γ**.

Nagrinėjame **A →** B_i. Galimi atvejai:

6) **B**$_i$ – aksioma
7) **B**$_i$ – prielaida iš **Γ**.
8) **B**$_i$ = **A**.
9) **B**$_i$ gautas pagal **MP** iš jau esančių. *(j,k < i)*

∀ atvejį nagrinėjame atskirai:

1) **A →** B_i pakeiskime **A →** B_i įrodymu: B_i *(aks.)*, $B_i \to (A \to B_i)$ *(1.1. aks.)*, $A \to B_i$ *(pagal MP)*.
2) **A →** B_i pakeiskime analogiška seka *(tik šiuo atveju B_i – prielaida)*.
3) **A → A** keiskime jos įrodymu H.T.T.S.
4) Tik jei $i \ge 3$. Kadangi B_i gautas pagal **MP** iš B_j ir B_k, tai $B_k = B_j \to B_i$. $A \to B_j$ keiskime seka:
   - [7] $(A \to (B_j \to B_i)) \to ((A \to B_j) \to (A \to B_i))$ – 1.2. aks.
   - [8] $(A \to B_j) \to (A \to B_i)$ – pagal MP iš [7] ir sek$_1$ nario $A \to B_k$.
   - [9] $A \to B_i$ – pagal MP iš [8] ir sek$_1$ nario $A \to B_j$.

Po šių keitimų, vietoje ∀ f-lės $A \to B_i$ *(i=1, …, m)* gauname $A \to B_m = A \to B$ išvedimą.

Teiginys įrodytas.

6 teiginys. Jei disjunktų dedukcinėje sistemoje iš aibės **S** galima išvesti disjunktą C ir jis nėra įvykdomas, tai aibė **S** – prieštaringa.

Reikia žinoti.

Kas yra logikos formulių aibės interpretacija.

Įrodymas.

(1 prielaida) Tarkim $C_1, \ldots, C_s = C$ yra disjunkto **C** išvedimas iš aibės **S** ir **S** – įvykdoma. Vadinasi, ∃ interpretacija $\boldsymbol{v}$, su kuria visi aibės **S** disjunktai teisingi.

Išvedimo ilgis – f-lių, esančių išvedimo sekoje, skaičius. Taikydami indukciją pagal išvedimo ilgį s, parodysime, kad su ta pačia interpretacija v ir C yra teisingas.

Jei $\boldsymbol{s} = \mathbf{1}$, tai $\boldsymbol{C_1} \in S$ ir todėl $\boldsymbol{v}(\boldsymbol{C_1}) = \boldsymbol{t}$.

(2 prielaida) Tarkim, kad ∀ disijunktas $\boldsymbol{C_i}$ $(i < m)$, tenkina: $\boldsymbol{v}(\boldsymbol{C_i}) = \boldsymbol{t}$. Parodysim, kad ir $\boldsymbol{v}(\boldsymbol{C_m}) = \boldsymbol{t}$.

$\boldsymbol{C_m} \in$ **S** *(tada $\boldsymbol{v}(\boldsymbol{C_m} = \boldsymbol{t})$)* arba gautas pagal A.T. iš kairėje esančių disjunktų(žym. C_j, C_k).

(3 prielaida) Tarkim $C_j = p \vee C'_j$, $C_k = \neg p \vee C'_k$ ir $C_m = C'_j \vee C'_k$. Pagal indukcijos prielaidą abu C_j, C_k teisingi su interpretacija $\boldsymbol{v}$. Galimi atvejai:

a) Jei $\boldsymbol{v}(\boldsymbol{p}) = \boldsymbol{t}$, tai $\boldsymbol{v}(\boldsymbol{C'_k}) = \boldsymbol{t} \xrightarrow{todėl} \boldsymbol{v}(\boldsymbol{C_m}) = \boldsymbol{t}$

b) Jei $\boldsymbol{v}(\boldsymbol{p}) = \boldsymbol{k}$, tai $\boldsymbol{v}(\boldsymbol{C'_j}) = \boldsymbol{t} \xrightarrow{todėl} \boldsymbol{v}(\boldsymbol{C_m}) = \boldsymbol{t}$

t.y. C_m įvykdoma su ta pačia interpretacija $\boldsymbol{v}$.

Gavome: jei S – įvykdoma, tai ir C – įvykdomas. Jei S ⊢ C ir C nėra įvykdomas, tai aibė S – prieštaringa.

Teiginys įrodytas.

<u>7 teiginys.</u> Jei disjunktų aibė **S** – prieštaringa, tai iš S išvedamas tuščias disjunktas$\square$.

<u>Įrodymas.</u>

Pažymėkime loginių kintamųjų esančių aibėje S kiekį l. Įrodysime mat. indukcijos principu. Indukcijos bazė $(l = 1)$ yra vieno iš pavidalų: $\{p\}, \{\neg p\}, \{p, \neg p\}$. Tik 3-iu atveju aibė - prieštaringa ir išvedamas $\square$.

Jei aibėje yra disjunktas pavidalo $p \vee \neg p \vee C$, tai jis visada bus teisingas, todėl aibė S prieštaringa bus t.t.t., kai jį išbraukus aibė bus prieštaringa. Laikykime, kad tokiu disjunktų nėra.

(1 prielaida) Tarkime kai aibėje yra $l < k$ kintamųjų ir ji prieštaringa, tai iš jos išvedamas$\square$. Įrodysime, kad kai $l=k$ ir aibė prieštaringa, iš jos taip pat išvedamas$\square$.

Tegul *p* yra koks nors loginis kintamasis, tenkinantis sąlygas: aibėje S yra disjunktas su p ir aibėje S yra disjunktas su $\neg p$, bet be p. Jei tokio p neatsirastų, tai aibė būtų įvykdoma.

Imkime aibę S_p, į kurią įeina visi disjunktai, turintys *p* bei $\neg p$. Suskaidome ją į dvi dalis: S_p^+ ir S_p^- taip, kad pirmojoje būtų visi su *p*, o antrojoje visi su $\neg p$.

Taikome atžvilgiu p atkirtos taisyklę, imdami vieną disjunktą iš S_p^+, o kitą - iš S_p^- . Visų, tokiu būdu gautų, disjunktų aibę pažymėkime $at(S_p)$. Aibės $at(S_p)$ disjunktuose nėra p (kartu ir $\neg p$). Parodysime, kad aibė S įvykdoma t.t.t., kai įvykdoma:

$$A = (S - S_p) \cup at(S_p).$$

Iš esmės aibė A yra ta pati S, tik jos disjunktuose nebėra p (bei $\neg p$).

1. *(2 prielaida)* Tarkime, S įvykdoma. Tuomet visi disjunktai iš $at(S_p)$ taip pat įvykdomi, nes gauti iš įvykdomų disjunktų, pritaikius atkirtos taisyklę. $S - S_p$ įvykdoma, kadangi yra įvykdomos aibės poaibis. Be to, abi aibės įvykdomos su viena ir ta pačia interpretacija. Taigi A įvykdoma.
2. *(3 prielaida)* Tarkime, aibė A įvykdoma. Vadinasi, yra interpretacija $\boldsymbol{v}$, su kuria visi disjunktai iš A teisingi. Parodysime, kad $\boldsymbol{v}$ galima pratesti taip, t.y. priskirti kintamajam p tokia reikšmę, kad būtų ivykdoma S_p, o kartu bus ivykdoma ir S.

Tegul $S^+ = \{C_1 \vee p, ..., C_m \vee p\}$, o $S^- = \{D_1 \vee \neg p, ..., D_n \vee \neg p\}$.

Tuomet: $at(S_p) = \begin{cases} C_1 \cup D_1, ..., C_1 \cup D_n \\ ... \\ C_m \cup D_1, ..., C_1 \cup D_n \end{cases}$

1. *(4 prielaida)* Tarkime, ∃ toks *i*, kad $v(C_i) = k$ $(1 <= i <= m)$. Tuomet $v(D_j)=t$ $(j = 1..n)$ yra teisingi, todėl p galime priskirti reikšmę t.
2. *(5 prielaida)* Tarkime, $v(C_i) = t (j = 1..m)$, tada p galime priskirti reikšmę k.

Gavome, kad aibė S yra įvykdoma tada ir tik tada, kai įvykdoma A, t.y. aibė S prieštaringa tik kai A prieštaringa. Kadangi aibėje A mažiau nei k elementų, tai jai galioja indukcijos prielaida.

Teiginys įrodytas.

10 teiginys. Jei $\mathbf{A_1}$,$\mathbf{A_2}$ – B.A. kalbos, tai $A_1 \cup A_2$, $A_1 \cap A_2$ – taip pat B.A. kalbos.

Įrodymas.

(prielaida) Tarkim, kad žodžių aibės **A** ir **B** yra B.A. $\boldsymbol{\psi_A} = < \boldsymbol{G_A}, \boldsymbol{F_A} >$ir $\boldsymbol{\psi_B} = < \boldsymbol{G_B}, \boldsymbol{F_B} >$, kalbos.

10) Pavaizdavus šiuos B.A. orientuotais grafais(viršūnes vaizduojant B.A. būsenomis, o perėjimų funkcijas - lankais tarp viršūnių), bei tarus, kad:

1-ojo B.A. ψ_1: būsenų aibė yra $Q_A = \{q_0, \dots, q_c, \dots, q_i, \dots, q_m\}$, galutinių būsenų aibė $F_A = \{q_c, q_i, \dots\}$,

2-ojo B.A. ψ_2: būsenų aibė yra $Q_B = \{q_0, \dots, q_d, \dots, q_j, \dots, q_n\}$, galutinių būsenų aibė $F_B = \{q_d, q_j, \dots\}$

5) Galime sudaryti abiejų B.A. būsenų aibių Dekarto sandaugą (**žym.: $Q_{A\times B} = Q_A \times Q_B$**) *(Pagal apibrėžimą, dviejų aibių A ir B dekarto sandauga yra visų įmanomų porų (m,n) aibė, kur 1-asis poros elementas yra iš aibės A, o 2-asis – iš aibės B)*:

$$Q_{A\times B} = \left\{ \begin{array}{c} (q_0,q_0), \dots, (q_0,q_d), \dots, (q_0,q_j), \dots, (q_0,q_n), \\ \dots \\ (q_c,q_0), \dots, (q_c,q_d), \dots, (q_c,q_j), \dots, (q_c,q_n), \\ \dots \\ (q_i,q_0), \dots, (q_i,q_d), \dots, (q_i,q_j), \dots, (q_i,q_n), \\ \dots \\ (q_m,q_0), \dots, (q_m,q_d), \dots, (q_m,q_j), \dots, (q_m,q_n) \end{array} \right\}$$

6) Tarkim, kad grafas $\mathbf{G_{A\times B}}$ yra grafų $\mathbf{G_A}$ ir $\mathbf{G_B}$ dekarto sandauga: $\mathbf{G_{A\times B}} = \mathbf{G_A} \times \mathbf{G_B}$. Tuomet grafo $\mathbf{G_{A\times B}}$ viršūnės bus aibės $\mathbf{Q_{A\times B}}$ elementai. Taigi, jei $\mathbf{G_A}$ turi **m** viršūnių, o $\mathbf{G_B}$ – **n**, tai $\mathbf{G_{A\times B}}$ turi **m** × **n** viršūnių. Lankas $\boldsymbol{l_{A\times B}} = (\boldsymbol{q_c}, \boldsymbol{q_d}) \xrightarrow{v} (\boldsymbol{q_i}, \boldsymbol{q_j})$, $v \in \Sigma$, tarp dviejų grafo $\mathbf{G_{A\times B}}$ viršūnių bus t.t.t., jeigu ∃ lankai: $\boldsymbol{l_A} = (\boldsymbol{q_c}) \xrightarrow{v} (\boldsymbol{q_i})$ ir $\boldsymbol{l_B} = (\boldsymbol{q_b}) \xrightarrow{v} (\boldsymbol{q_j})$.

7) Tarkim, kad $\boldsymbol{\psi_{A\times B}} = < \boldsymbol{G_A} \times \boldsymbol{G_B}, \boldsymbol{F_C} >$. Tuomet, jei:
   1. $(\boldsymbol{q_i}, \boldsymbol{q_j}) \in \boldsymbol{F_C}$ t.t.t., kai: $\boldsymbol{q_i} \in \boldsymbol{F_A}$ ***arba*** $\boldsymbol{q_j} \in \boldsymbol{F_B}$, tai B.A. $\boldsymbol{\psi_{A\times B}}$ kalba yra $\boldsymbol{A} \cup \boldsymbol{B}$;
   2. $(\boldsymbol{q_i}, \boldsymbol{q_j}) \in \boldsymbol{F_C}$ t.t.t., kai: $\boldsymbol{q_i} \in \boldsymbol{F_A}$ **ir** $\boldsymbol{q_i} \in \boldsymbol{F_B}$, tai B.A. $\boldsymbol{\psi_{A\times B}}$ kalba yra $\boldsymbol{A} \cap \boldsymbol{B}$.

Teiginys įrodytas.

Pastaba: Norėdami aiškiau suvokti – kas yra grafų Dekarto sandauga, peržvelkite 4-ųjų pratybų „b“ dalies – *„Baigtiniai Automatai“*, uždavinį su grafų Dekarto sandauga.

16 teiginys. Baigtinumo problema neišsprendžiama.

Įrodymas.

(1 prielaida) Tarkime, kad ∃ algoritmas išsprendžiantis baigtinumo problemą. Vadinasi yra tokia **TM**, kuri su pradiniais duomenimis $\boldsymbol{\alpha_2(x, y)}$ po baigtinio skaičiaus žingsnių baigs darbą, juostoję įrašiusi **0** arba **1** ir ties ja bus skaitymo galvutė. Tuomet:

$$\boldsymbol{g}\left(\alpha_2(\mathbf{x}, \mathbf{y})\right) = \begin{cases} 1, & jei\ \boldsymbol{\varphi_x}(y) < \infty \\ 0, & jei\ \boldsymbol{\varphi_x}(y) = \infty \end{cases}$$

yra BR f-ja ir atsiras TM paskaičiuojanti DR f-ją Ψ(x):

$$\boldsymbol{\Psi}(\mathbf{x}) = \begin{cases} 1, & jei\ g\left(\alpha_2(\mathrm{x}, \mathrm{y})\right) = 0 \\ \infty, & jei\ g\left(\alpha_2(\mathrm{x}, \mathrm{y})\right) = 1 \end{cases}$$

Ją gauname taip: ∀ galutinę būseną q_i išbraukiame iš F aibės, o perėjimų f-ją papildome:

$$\delta(\mathrm{q_i}, 1) = (\mathrm{q_i}, 1, \mathrm{D})$$
$$\delta(\mathrm{q_i}, \mathrm{b}) = (\mathrm{q_i}, 1, \mathrm{D})$$
$$\delta(\mathrm{q_i}, 0) = \left(q_{k_i}, 1, \mathrm{N}\right) \quad , čia\ q_{k_i} - \ kurios\ nors\ naujos\ būsenos, jas\ priskiriame\ F\ aibei.$$

(2 prielaida) Tarkime, kad l yra TM, apskaičiuojančios Ψ, numeris. Tuomet l bus ir Ψ numeris,t.y. $\Psi = \varphi_l$.

Aiškinamės, ar $\Psi(l) < \infty$. Iš 1 prielaidos gauname prieštarą:

1) Jei $\boldsymbol{\Psi(l)} < \infty$, tai $\boldsymbol{g}\left(\alpha_2(\boldsymbol{l}, \boldsymbol{l})\right) = \mathbf{0}$ ir $\boldsymbol{\varphi_l(l)} = \infty$, t.y. $\boldsymbol{\Psi(l)} = \infty$.
2) Jei $\boldsymbol{\Psi(l)} = \infty$, tai $\boldsymbol{g}\left(\alpha_2(\boldsymbol{l}, \boldsymbol{l})\right) = \mathbf{1}$ ir $\boldsymbol{\varphi_l(l)} = \infty$, t.y. $\boldsymbol{\Psi(l)} < \infty$.

Teiginys įrodytas.

18 teiginys. Standartinių TM aibė yra skaiti.

Įrodymas.

∀ standartinę **TM** galime užrašyti žodžiu iš abėcėlės $\mathrm{A} = \{0, 1, \mathrm{b}, \mathrm{q}, 2, \ldots, 9, \delta, =, (,), \mathrm{K}, \mathrm{D}, \mathrm{N}\} \cup \{,\}$. Baigtinės abėcėlės visų žodžių aibė yra skaičioji. Taigi žodžiai, sudarantys kurios nors TM perėjimų f-jas, yra begalinės skaičios aibės $\mathbf{A^*}$ poaibis, kuris yra skaitus, nes skaičios aibės poaibis yra skaitus arba baigtinis. ***(past.:** žvaigždute* žymima abėcėlės A žodžių aibė)*

Teiginys įrodytas.

23 teiginys. Visi 3 rekursyviai skaičios aibės apibrėžimai yra ekvivalentūs. *(tik dalinis įrodymas)*

I. Aibė A yra R.S., jei ji sutampa su kurios nors DR f-jos D_f.
II. Aibė A yra R.S., jei ji sutampa su kurios nors PR f-jos E_f.
III. Aibė A yra R.S., jei ∃ tokia PR f-ja $f(a, x)$, kad $f(a, x) = 0$ turi sprendinį t.t.t., kai $a \in A$.

Įrodymas *(II → III).*

(prielaida) Tarkime A – R.S. pagal II apibr.. Tada A sutampa su PR f-jos h(x) E_f. Tada $|h(x) - a| = 0$ yra PR ir turi sprendinį t.t.t., kai $a \in A$. Vadinasi A – R.S. pagal III apibr..

Teiginys įrodytas.

24 teiginys. Jei aibė ***A*** yra R.S., bet nėra rekursyvi, tai jos papildinys $\overline{\boldsymbol{A}}$ nėra nei rekursyvi nei R.S. aibė.

Įrodymas.

*(prielaida)*Tarkime priešingai $\bar{A}$ – R.S.. A – R.S. *(duota)*. Pagal II R.S. aibės apibr. ∃ tokios PR f-jos f(x) ir g(x), kad $A = \{f(0), f(1), \dots\}$ ir $\bar{A} = \{g(0), g(1), \dots\}$.

Imame f-ją $h(x) = \mu_z(|f(z) - x| * |g(z) - x| = 0)$, kuri yra **BR**, nes apibrėžta $\forall x \in \mathbb{N}$, o $A \cup \bar{A} = \mathbb{N}$.

Jei $\boldsymbol{x} \in \boldsymbol{A}$ $(\boldsymbol{x} \notin \overline{\boldsymbol{A}})$, tai $h(x) = z$, kuriam teisinga, kad $f(z) = x$

Jei $\boldsymbol{x} \in \overline{\boldsymbol{A}}$ $(\boldsymbol{x} \notin \boldsymbol{A})$, tai $h(x) = z$, kuriam teisinga, kad $g(z) = x$, $f(z) \neq x$

Tada aibės A charakteringoji BR f-ja:

$$\chi_A(x) = \overline{sg}(|f(h(x) - x|) = \begin{cases} \overline{sg}(\overbrace{|f(z) - x|}^{=0}) & , \ jei\ x \in A \\ \overline{sg}(\underbrace{|f(z) - x|}_{>0}) & , \ jei\ x \notin A \end{cases} = \begin{cases} 1, jei\ x \in A \\ 0, jei\ x \notin A \end{cases}$$

Iš šio char. f-jos apibr. išplaukia, kad A – rekursyvi, o pagal sąlygą: A – nėra rekursyvi. Gauta prieštara.

Teiginys įrodytas.

26 teiginys. ∃ **BR**, bet ne **PR** funkcija $(h(x) = A(x,x) \in BR,\ bet\ \notin PR)$.

Reikia žinoti.

F-ja $f(x)$ yra mažoruojama f-jos $h(x)$, jei ∃ toks $\mathbf{x_0}$, nuo kurio su visais $x \geq x_0$ galioja: $f(x) < h(x)$.

Įrodymas.

Parodysim, kad f-ja $\boldsymbol{h(x)}$ mažoruoja ∀ 1-arg. **PR** f-ją, todėl pati nėra **PR**.

Pirmiausia - kad ir kokia būtų $\boldsymbol{f(x)}$, ∃ toks n, su kuriuo $\boldsymbol{f(x)}$ bus mažoruojama f-jos $\boldsymbol{A(n,x)}$:

1. $s(x) < 2^x = A(2,x)$, $x = 2, 3, \dots$
2. $q(x) < s(x) = 2^x < A(2,x)$
3. *(prielaida)* Tarkim $\mathbf{f(x)} < \boldsymbol{A(n_1,x)}$ ir $\mathbf{g(x)} < \boldsymbol{A(n_2,x)}$, bei $\boldsymbol{n = n_1 + n_2}$.
   Tada $\mathbf{f(x)} < \boldsymbol{A(n,x)}$ ir $\mathbf{g(x)} < \boldsymbol{A(n,x)}$
4. Pritaikome sudėties operatorių:
   $$f(x) + g(x) < 2 \cdot A(n,x) < 2 \cdot 2^{A(n,x)} \leq 2^{A(n+1,x)} \leq A(n, A(n+1,x)) = A(n+1, x+1) \leq A(n+2, x)$$
5. Pritaikome kompozicijos operatorių:
   $$f(g(x)) \leq A(n, g(x)) \leq A(n, A(n+1,x)) = A(n+1, x+1) \leq A(n+2, x)$$
6. Panašiai gaunamas įvertis ir iteracijos operatoriaus atveju

Parodėm, kad ∀ PR klasės f-jai f(x) ∃n: f(x) < A(n,x). Parodysime kad f(x) mažoruojama h(x):

$$f(n+x) < A(n, n+x) \leq A(n+x, n+x) = h(n+x) \qquad \text{, čia } f(x) < A(n,x)$$

Taigi, $\forall f(x) \in PR$ yra mažoruojama $h(x) \in BR$. Todėl $h(x) \notin PR, bet \in BR$. Vadinasi $PR \subset BR$.

Teiginys įrodytas.

27 teiginys. a) Visų n-arg. **PR** f-jų universalioji $F(x_0, x_1, ..., x_n) \notin$ **PR**.
b) Visų n-arg. **BR** f-jų universalioji $F(x_0, x_1, ..., x_n) \notin$ **BR**.

Įrodymas a).

(prielaida) Tarkim $F(x_0, x_1, ..., x_n) \in$ **PR**, tada imam f-ją $g(x_1, ..., x_n) = F(x_1, x_1, ..., x_n)+1 \in$ **PR**, nes gauta iš ***x+1*** ir ***F***, pritaikius kompozicijos operatorių, bei ∃ toks $i \in \mathbb{N}$, kad $F(i, x_1, ..., x_n) = g(x_1, ..., x_n)$, nes F – universalioji.

Tada paėmę $x_1 = ... = x_n = i$: $F(i, i, ..., i) = g(i, ..., i)$ ir $F(i, ..., i)+1 = g(i, ..., i)$ **=>** $F(i, i, ..., i) = F(i, ..., i)+1$ ir F ∈ **PR***(apibrėžta visur)*. Gauta prieštara => F ∉ **PR**.

Teiginys įrodytas.

Įrodymas b). Visur vietoje **PR** įsistatyti **BR**, įrodoma analogiškai kaip ir *a)* atveju.

28 teiginys. Visų 1-arg. **PR** funkcijų aibe ∃ universalioji **BR** klasė.

Įrodymas.

Visas 1-arg. PR f-jas galime išreikšti per bazines s(x) ir q(x), taikant sudėties, kompozicijos ir iteracijos operatorius. Todėl ∀ PR f-jai galime priskirti numerį. f(x) numerį žymėsime n(f(x)), kur n – f-jos numeris.

$\boldsymbol{n(f(x))}$ žym. $\boldsymbol{f_n(x)}$	jei $n(f(x)) = a$ ir $n(g(x)) = b$, tai:
$n(s(x)) = 1$	$n(f(x)+g(x)) = 2*3^a*5^b$
$n(q(x)) = 3$	$n(f(g(x))) = 4*3^a*5^b$
	$n(f(x)^I) = 8*3^a$

Apibrėžiame **2-arg.** f-ją $\boldsymbol{F(n,x) = f_n(x)}$, t.y. $\boldsymbol{F(n,x)}$ lygi 1-arg. f-jai, kurios numeris yra n:

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

Iš apibr. matome, kad ∀ f-ja turi numerį, bet ne vienintelį. Be to ∃ ℕ skaičių, kurių neatitinka jokios f-jos.

Dabar apibrėžiame 1-arg. PR f-jų universaliąją:

$$D(n,x) = \begin{cases} F(n,x) , & jei\ n\ yra\ kuris\ nors\ f-jos\ numeris \\ 0, & priešingu\ atveju \end{cases}$$

$D(n,x) \in BR$, nes yra visur apibrėžta, tačiau pagal „27tg.a)" dalį, $D(n,x) \notin PR$.

Teiginys įrodytas.

29 teiginys. $D^{n+1}(x_0, . . . , x_n) = D(x_0, \alpha_n(x_1, . . . , x_n))$ yra visų n-arg. **PR** funkcijų universalioji.

Su kiekvienu fiksuotu x_0, funkcija $D(x_0,\alpha_n(x_1,...,x_n))$ yra PR. Taip pat, jei $g(x_1,...,x_n)$ yra kuri nors n-arg. PR f-ja, tai tokia yra ir **1-arg.** $\boldsymbol{f(x)} = g(\pi_n^1(x), ..., \pi_n^n(x))$. Tad atsiras toks natūralusis x_0, kad $f(x) = D(x_0,x)$.

Skaičius x_0 ir yra $g(x_1,...,x_n)$ numeris, nes

$$f(\alpha_n(x_1, ..., x_n)) = g(\pi_n^1(\alpha_n(x_1, ..., x_n)), ..., \pi_n^n(\alpha_n(x_1, ..., x_n))) = g(x_1, ..., x_n)$$

Teiginys įrodytas.

30 teiginys. Visų **n**-arg. **DR** funkcijų aibe ∃ universalioji $\widetilde{D}^{n+1}(x_0, x_1, ..., x_n)$.

Įrodymas. Bet kurios **DR** f-jos grafikas yra R.S. aibė. Taigi ∃ tokia **PR** f-ja $g(x_1, ..., x_n, y, z)$, kad $(x_1, ..., x_n, y) \in$ **A** t.t.t., kaip ∃ toks **z**, kad $g(x_1, ..., x_n, y, z) = 0$.
(prielaida) Tarkim, kad $t = \alpha_2(y,z)$. Tada $(x_1, ..., x_n, y, z) \in$ **A** t.t.t., kai ∃ toks **t**, kad $g(x_1, ..., x_n, \pi_2^1(t), \pi_n^1(t)) = 0$. Pažymėkime ją $F(x_1, ..., x_n, t)$.
Kad ir kokia būtų **DR** f-ja $f(x_1, ..., x_n)$, atsiras tokia **PR** $F(x_1, ..., x_n, t)$,kad

$f(x_1, ..., x_n) = \pi_2^1(\mu_t(F(x_1, ..., x_n, t) = 0))$ *{INF: 31.1}*

DR n-arg. f-jų universalioji $\widetilde{D}^{n+1}$ gaunama taip:

$\widetilde{D}^{n+1}(x_0, x_1, ..., x_n) = \pi_2^1(\mu_t(D^{n+2}(x_0, x_1, ..., x_n, t) = 0))$

Iš tikrųjų ši f-ja su ∀ fiksuotu x_0 yra **DR**. Tačiau, jei $f(x_1, ..., x_n)$ yra kuri nors **DR** f-ja, tai ∃ tokia **PR** $F(x_1, ..., x_n, t)$ kuriai galioja *{INF: 31.1}*. Tarkim jos numeris yra i, tada:

$\widetilde{D}^{n+1}(i, x_1, ..., x_n) = \pi_2^1(\mu_t(D^{n+2}(i, x_1, ..., x_n, t) = 0))$

Teiginys įrodytas.

31 teiginys. Visų **n**-arg. **DR** funkcijų aibės universalioji $\widetilde{D}^{n+1}(x_0, x_1, ..., x_n)$ neturi pratęsimo.

Įrodymas. Imam $V(x) = \overline{sg}\widetilde{D}^{n+1}(x, x, ..., x)$. Jei V(x) apibrėžta su kuriuo nors x_0, tai jos reikšmė yra 1 arba 0.
(prielaida) Tarkim V(x) turi pratęsimą W(x). Į ją(1-arg) galime žiūrėti kaip į n-arg f-ją:

$W(x_1) = pr_n^1(W(x_1), x_2 ..., x_n)$

Atsiras tos a, kad $\widetilde{D}^{n+1}(a, x_1, ..., x_n) = W(x_1)$. Ji visur apibrėžta. Imam $x_1 = ... = x_n = a$.
$W(x)$ yra $V(x) = \overline{sg}\widetilde{D}^{n+1}(x, x, ..., x)$ pratęsimas. Gaunam prieštarą $W(a) = \overline{sg}\ W(a)$. Taigi V(x) neturi pratęsimo.
Tarkim, kad $\widetilde{D}^{n+1}(x_0, x_1, ..., x_n)$ turi pratęsimą $P(x_0, x_1, ..., x_n)$. Tada $\overline{sg}P(x, x, ..., x)$ būtų V(x) pratęsimas, o tokios tarp BR f-jų nėra.
Teiginys įrodytas.

32 teiginys. ∃ rekursyvios, bet ne rekursyviai skaičios aibės.

Įrodymas. Imam universaliąją 1-arg. f-joms $\widetilde{D}^2(x_1,x_2)$. F-ja $V(x) = \overline{sg}\widetilde{D}^2(x,x)$ turi savybes:

1) V(x) yra DR.
2) V(x) neturi pratęsimo.
3) V(x) reikšmių aibė yra {0,1}.

Lygties V(x) = 0 sprendinių aibė yra R.S., nes sutampa su DR f-jos $\mu_z(V(x) + z = 0)$ D_f. Jei ji būtų rekursyvi, t.y. atsirastų tokia BR κ(x), kad:

$$\kappa(x) = \begin{cases} \mathbf{1}, & jei\ V(x) = 0, \\ \mathbf{0}, & kitu\ atveju. \end{cases}$$

tai $\overline{sg}\ \kappa(x)$, būtų V(x) pratęsimas. O tai prieštarauja 2-ai f-jos V(x) savybei.

Teiginys įrodytas.