Stanislovas Norgėla

MATEMATINĖS LOGIKOS PAGRINDAI

Mokomasis leidinys

Vilnius, 2025

UDK 16(075.8)

Turinys

Pratarmė

Knygoje nagrinėjamos formalios logikos pagrindinės savybės bei skaičiavimai: sekvencinis, semantinių lentelių, rezoliucijų, natūralios dedukcijos. Supažindinama su pirmos eilės predikatų logikos taikymais formalioms aksiominėms teorijoms (Peano aritmetika, Zermelo-Frenkelio aibių teorija). Skyriuje *loginis programavimas* aprašyti logikos rezultatai, kuriais remiantis sukurtos matematikos teoremų išvedimų paieškai skirtos *programavimų kalbos*.

Aprašomos ir aukštesnės eilės logikos bei nagrinėjamos jų galimybės matematikos teoremų automatizuotai įrodymų paieškai. Knygos tikslas - supažindinti su naudojamais matematinės logikos rezultatais programuojant *gebėjimą kompiuteriui mąstyti*.

Knyga skirta *informatikos* bei *programų sistemų* specialybių aukštųjų mokyklų studentams. Ji bus naudinga klausantiems *matematinės logikos*, *algoritmų teorijos*, *diskrečios matematikos* bei *dirbtinio intelekto* kursus.

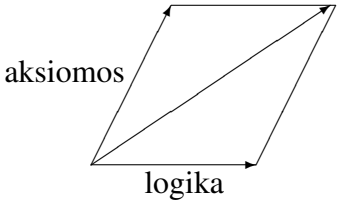
Yra daug uždavinių ir visiems jiems pateikti sprendimai.

Autorius

Įvadas

Matematika susideda iš daugelio matematinių teorijų.

Matematinę teoriją galima pavaizduoti lygiagretainiu, kurio viena kraštinė yra tos teorijos aksiomų sistema, o kita logika (išvedimo taisyklės).



Lygiagretainio įstrižainė simbolizuoja matematinę teoriją – aibę visų įrodomų teorijoje teiginių.

Kintant aksiomų aibei (fiksuotoje logikoje), kinta ir teorija. Gauname kitokią aibę įrodomų teiginių. Kintant logikai (aksiomos fiksuotos) taip pat kinta teorija. Taip pat, suprantama, pasikeis teorija, jei pakeisim ir aksiomas, ir logiką.
Aksiomų samprata bėgant amžiams keitėsi. Euklido „Pradmenyse“ aksiomomis vadinami akivaizdūs teiginiai. Vėliau aksiomomis pradėta vadinti teiginius, kurie autorių (sugalvojusių naujas aksiomas) nuomone įdomūs, sukuriantys naujas neprieštaringas matematines teorijas.
Euklidas (apie 365 – 275 m. per. Kr.) savo veikale Pradmenys suformulavo tokias geometrijos aksiomas:
1. Per du nesutampančius taškus visada galima nubrėžti tiesę.
2. Atkarpą, jungiančią du taškus visada galima pratęsti į abi puses ir gauti tiesę.
3. Apie bet kokį tašką galima apibrėžti bet kokio spindulio apskritimą.
4. Visi statūs kampai tarpusavyje lygūs.
5. Jeigu tiesei (A) kertant dvi kitas tieses (B, C) vienoje pusėje susiformuoja kampai, nelygūs dviems statiems, tai tas dvi tieses (B, C) pratęsus neribotai, jos susikirs toje pusėje, kur vidinių kampų suma mažesnė už du stačiuosius.

Pakeitus penktąjį postulatą Riemanno aksioma gaunama *Riemanno neeuklidinė geometrija*. *Riemanno aksioma*. Per tašką, nesantį tiesėje, taško ir tiesės plokštumoje nėra nei vienos tiesės, einančios per tašką ir nekertančios turimos tiesės.

Pakeitus penktąjį Euklido geometrijos postulatą Lobačevskio aksioma gaunama *Lobačevskio neeuklidinė geometrija*.

Lobačevskio aksioma. Per tašką, nesantį tiesėje, taško ir tiesės plokštumoje galime nubrėžti mažiausiai dvi skirtingas tieses, nekertančias turimos tiesės.

Abiem atvejais pakito aksiomos, o logika liko ta pati. Ilgą laiką buvo manoma, kad yra vienintelė logika, kurią naudojame kaip šnekamojoje kalboje, taip ir matematikoje, fizikoje bei kitose mokslo šakose. Tik dalis logikos taisyklių aprašytos išreikštiniu pavidalu. Nežinoma tiksliai kokiais logikos dėsniais naudojasi matematikai. Matematinis mąstymas susiformavo savaime per tiksliųjų mokslų pamokas mokykloje.

Vokiečių matematiko ir logiko G. Frege's (1848-1925) dėka logikoje naudojamas simbolis

$\vdash$. Sakinį *tvirtinama, kad teisingas F* jis rašydavo $\vdash F$. Dabar tas simbolis naudojamas kaip *įrodomas* (*išvedamas*) *F*.

Turime teiginį (teoremą) *F*. Galimi keturi atvejai:

1. $\vdash \boldsymbol{F}$ **ir** $\nvdash \neg \boldsymbol{F}$, įrodomas *F* ir neįrodomas $\neg F$;
įrodoma, kad teiginys teisingas, ir neįrodoma, kad jis klaidingas,

2. $\nvdash \boldsymbol{F}$ **ir** $\vdash \neg \boldsymbol{F}$, neįrodomas *F* ir įrodomas $\neg F$;
neįrodoma, kad teiginys teisingas, o įrodoma, kad jis klaidingas,

3. $\nvdash \boldsymbol{F}$ **ir** $\nvdash \neg \mathbf{F}$, neįrodomi nei *F*, nei $\neg F$;
negalime įrodyti, kad teiginys teisingas, negalime įrodyti ir kad jis klaidingas,

4. $\vdash \boldsymbol{F}$ **ir** $\vdash \neg \boldsymbol{F}$, įrodomas *F* ir įrodomas $\neg F$; galime įrodyti, kad teiginys teisingas ir kad jis yra klaidingas.

Seniai matematikai suprato, kad ketvirtas atvejis matematikoje, kaip ir bet kurioje kitoje mokslo teorijoje, nepriimtinas. *Nes iš prieštaravimo išplaukia bet kuris teiginys*. Todėl vieninteliu reikalavimu bet kuriai matematikos teorijai tapo *neprieštaringumo įrodymas*.

O kaip su trečiuoju atveju? O kaip galima juo pasinaudoti, jei matematikoje iki XX amžiaus vidurio *įrodymas* buvo netiksli, tik intuityviai suprantama sąvoka? Nebuvo aišku *kas yra įrodymas*. Tik suteikus sąvokai *įrodymas* tikslią matematinę prasmę, galima kelti klausimą *įrodomas F, ar ne*? Taigi, ištisus šimtmečius matematikų mąstysenoje trečio atvejo nebuvo. Buvo tik du pirmieji atvejai. Galiojo trečiojo negalimo dėsnis. Koks bebūtų *F, arba įrodomas F, arba neįrodomas F*.

Sakau „iki XX mažiaus vidurio“, bet matematikos teorijos su *trečio negalimo dėsniu* sėkmingai vystomos toliau. Kodėl? Palikti susikurtą matematikos rojų, nors daugeliu atveju ir iliuzinį, nelengva.

Jei yra trečias atvejis, tai prie aksiomų sistemos galima prijungti bet kurį, arba *F*, arba $\neg F$. Jei matematinė teorija buvo neprieštaringa, tai naujosios irgi bus neprieštaringos.

Pavyzdžiui, teiginys *egzistuoja begalinė aibė* (turima omenyje natūraliųjų skaičių aibė) matematikoje neįrodomas. Tai žinomų logikų B.Russel ir A.Whitehead praeito šimtmečio pradžioje gautas rezultatas. Gaunamos dvi matematinės teorijos. Vienoje yra tik baigtinės aibės, kitoje atsiranda ir nauja begalinė aibė. Abi neprieštaringos. Kurią norite, tą ir pasirinkite. Skonio reikalas. Informatikai pasirinko pirmąją. Ir nieko neatsitiko. Uždirba pragyvenimui ir su tokia matematika.

Kokios dar yra logikos?

Pradėsim iš toli, nuo XIX amžiaus pabaigos. Iki 19 amžiaus pabaigos plačiai buvo paplitus nuomonė, kad pasaulis sukurtas pagal matematikos dėsnius. Tam pasitarnavo ir Koperniko, Keplerio, Galilėjaus darbai. Matematikos tvirtinimų (teoremų) pagrindiniu teisingumo kriterijumi buvo jų atitikmuo realaus pasaulio reiškiniams t.y. stebėjimų ir eksperimentų rezultatams.

Matematika buvo efektyvus būdas fizikos, astronomijos ir kitų taikomųjų uždavinių sprendimui.

Nebuvo domimasi matematinių sąvokų bei įrodymų patikimumu, griežtumu. Įrodymai dažnai rėmėsi geometrine intuicija. Dažniausiai tai buvo tik hipotezės, o ne įrodymai. Įrodymais tebuvo laikomi samprotavimai, *kuriais jus tiek įtikino, kad jūs pasiruošęs įtikinti ir kitus*. Dar didesnį netikrumą matematikoje įnešė vokiečių matematiko G.Kantoro aprašyta aibių teorija su savo paradoksais.

Galbūt didžiausią įtaką tuometinės matematikos raidai padarė vokiečių matematiko **D.Hilberto** (1862-1943) idėjos. Jis matematikos išėjimo iš krizės sprendimu laikė matematikos formalizavimą. Samprotavimai matematikoje turi būti *prasmingi ir patikimi*. Kiekvienoje matematikos teorijoje turi būti dviejų rūšių teiginiai. Aksiomos (teiginiai laikomi teisingais be įrodymo) ir teiginiai išvedami iš aksiomų naudojant **logiką**. Būtina išreikštiniu pavidalu aprašyti logikos dėsnius naudojamus matematiniuose įrodymuose ir atsakyti į klausimą **kas yra įrodymas**. Turime funkciją, jei mokame, žinodami argumento reikšmes, apskaičiuoti funkcijos reikšmes. Tai yra egzistuoja algoritmas funkcijai apskaičiuoti. **Kas yra algoritmas**?

Po A.Puankare mirties (1912 m.) vokiečių matematikas D.Hilbertas buvo pripažintas pasaulio matematikų lyderiu. Turėjo didelę įtaka jauniems matematikams besirenkantiems mokslinių darbų temas. Žymaus matematiko idėjos apie matematikos formalizavimą sudomino daugelį. Pirmas XX amžiaus pusmetis buvo *matematinės logikos aukso laikotarpis*.

Tai kaip atsirado kompiuteriai? Kam kilo idėjos jį sukurti? Kompiuterių atsiradimo istorija yra geras pavyzdys kaip matematikams sprendžiant problemą, lyg tai labai abstrakčią, nieko bendro neturinčią su taikymais, netikėtai gaunami dideli atradimai. Tad *kas yra įrodymas, kas yra algoritmas*?

Iš kur kilo žodis *algoritmas*? Tai sulotynintas arabų (kai kurie šaltiniai nurodo, kad persų) matematiko Al Chorezmio (*ca* 783-850) vardas. Penktame amžiuje dešimt skaitmenų 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 bei dešimtainę skaičiavimo sistemą sugalvojo Indijos matematikai. Tik po maždaug 300 metų dešimtainę sistemą perėmė arabai. Al Chorezmi išgarsėjo savo knyga, kurioje aprašė veiksmus su skaičiais, perimtais iš Indijos. Naujoji pozicinė skaičiavimo sistema (tuo metu Europoje buvo naudojami romėniški skaitmenys) greitai paplito pasaulyje, o jo knyga tapo daugelio žmonių parankine knyga. Skaitmenis pradėta vadinti arabiškais. Knygoje buvo daug taisyklių rinkinių, kuriuos taikant po baigtinio žingsnių skaičiaus gaunamas rezultatas.

Algoritmo sąvoka buvo tikslinama dviem būdais:

1) kuriama idealizuota (matematinė) skaičiavimo mašina,

2) aprašoma algoritmiškai apskaičiuojamų funkcijų klasė.

Po daugelio metų - 1934-1936 m. ir viena, ir kita kryptimi dirbantys mokslininkai gavo daug formalizmu bei idėjomis skirtingų algoritmo sąvokos patikslinimų. Pirmosios krypties

žinomiausiais darbais tapo A.Turingo ir E.Posto aprašytosios mašinos. Antrosios krypties –

K.Gödelio, A.Churcho bei S.Kleene aprašytosios funkcijų klasės.

Church -Turingo tezė. *Algoritmiškai apskaičiuojamų funkcijų aibė sutampa su Turingo mašina apskaičiuojamomis funkcijomis*. Tezė buvo paskelbta 1936 metais. *Teze* vadinama todėl,

kad tai tvirtinimas, kuriuo, reikia tikėti, bet įrodyti negalima. Negalima įrodyti dviejų aibių lygybės, nes, viena - tai matematiškai tiksli funkcijų klasė, kita - intuityvi, netiksli, skirtingų žmonių skirtingai suprantama *algoritmiškai apskaičuojamų funkcijų klasė*.

Kodėl tikima teze? Pagrindiniai argumentai yra du:

1. Visų pasiūlytų, skirtingomis idėjomis aprašytų algoritmiškai apskaičiuojamų funkcijų klasės sutampa tarpusavyje ir su idealizuotų skaičiavimo mašinų apskaičiuojamomis funkcijų klasėmis.

2. Nėra žinomas joks apskaičiuojamos funkcijos pavyzdys, kuris nebūtų apskaičiuojamas Turingo mašina.

*Turingo mašina apskaičiuojamų funkcijų aibė yra **skaičioji***.

*Visų algoritmų aibė yra **skaičioji***.

*Funkcijų, apskaičiuojamų šiuolaikiniais kompiuteriais aibė, sutampa su apskaičiuojamų Turingo mašina aibe ir yra **skaičioji***.

Taigi atsirado galimybė įrodymams visiškai kitokio tipo teoremoms: Jei *f(x) neapskaičiuojama Turingo mašina, tai funkcija neapskaičiuojama algoritmiškai*. Tai reiškia, kad nėra algoritmo f(x) apskaičiavimui. Niekas nemoka jos apskaičiuoti dabar (t.y. žinodami argumento reikšmes nemokame apskaičiuoti funkcijos reikšmių) ir *nemokės niekada*.

Turingas laikomas *informatikos mokslo tėvu*. 2007 metų birželio 23-čią dieną Anglijoje, Sackville parke greta Mančesterio universiteto atidengta memorialinė skulptūra skirta A.Turingui. Jis pavaizduotas sėdintis ant suolelio. Memorialinėje lentoje parašyta

Alan Mathison Turing, 1912-1954, Father of Computer Science, Mathematician, Logician.

O ką bendro turi algoritmo sąvokos apibrėžimas su kompiuteriais?

O gi, kad apskaičiuotum bet kurią funkciją, tereikia mokėti nedaug ir labai paprastų veiksmų. Visų pirma atkreipiu dėmesį, kad nagrinėjamos tik funkcijos apibrėžtos natūraliųjų skaičių aibėje su reikšmėmis taip pat natūraliųjų skaičių aibėje. Jos gali būti ir dalinės, t.y. su kai kuriomis reikšmėmis neapibrėžtos. Apibendrinti tokias funkcijas iki racionaliųjų skaičių aibės nesudėtinga. Visas sudėtingumas – aprašyti algoritmiškai apskaičiuojamas funkcijas natūraliųjų skaičių aibėje. O, jei sugalvosit apskaičiuoti funkciją, kai argumentas iracionalus skaičius, tai nesuspėsit iki mirties užrašyti vien tik argumento reikšmės.

Pabandysiu paaiškinti algoritmiškai apskaičiuojamas funkcijas. Pateiksiu porą pavyzdžių.

Pirmas pavyzdys. Tarkime mokame pridėti vienetuką. T.y. mokame apskaičiuoti funkciją $n+1$. Tuomet $n+m$ galima apskaičiuoti taip: $n+1+1+...+1$ (reikia pridėti *m* vienetukų). Tereikia mokėti kaip užbaigti vienetukų pridėjimą. Tuo tikslu apibrėžiamas rekursijos operatorius. $n \cdot m = n + n + \ldots + n$ (reikia *m* kartų pridėti *n*, o sudėti du skaičius jau mokame pridėdami vienetuką).

$$n \times m = n + n + \ldots + n.$$

Kad apskaičiuotum bet kurią funkciją tereikia turėti funkciją $n+1$ ir kelis operatorius: kompozicijos (iš funkcijų mokėti sudaryti jų kompoziciją), rekursijos ir minimizacijos. Jei tokiais veiksmais neįmanoma jūsų sugalvotos funkcijos apskaičiuoti su duota argumento reikšme, tai nei jūs, nei kas nors kitas niekada jos ir nemokės apskaičiuoti.

Antras pavyzdys. Visas algoritmiškai apskaičiuojamas funkcijas galima apskaičiuoti tokiu būdu: tarkime funkcijos $f(x)$ argumento reikšmė *n*, tuomet pradiniais algoritmo skaičiavimo duomenimis imamas skaičius 2^n. Algoritmo skaičiavimo rezultatas bus $2^{f(n)}$. Apskaičiavimui bet kurios funkcijos (suprantama, jei ji algoritmiškai apskaičiuojama) tereikia mokėti atlikti tik tokius veiksmus: $\times 2, \times 3, \times 5, :30$ (dauginti iš 2, 3 ir 5, bei dalinti iš 30). Be abejo, tuos veiksmus reikia surašyti tam tikra tvarka (parašyti programą). Jei $f(n)$ neapibrėžta (su argumento reikšme *n*), tai rezultato paieška užsiciklins (sakoma, dirbs be galo ilgai).

Suprantama, kad kai buvo aprašyti veiksmai, kurių pakanka paskaičiavimui bet kurios algoritmiškai apskaičiuojamos funkcijos reikšmėms, tai daugeliui kilo didelė pagunda realizuoti techniškai. Tie iš pradžių gremėzdiški aparatai ir buvo pavadinti *kompiuteriais*.

Matematikoje kai kurioms naudojamoms sąvokoms reikalingas patikslinimas. Pavyzdžiui, ką reiškia *duota skaičių seka*? Su patikslinimu keičiasi ir mąstymo taisyklės (**logika**). Nors aksiomos tos pačios, **matematinė teorija** gaunama kitokia. Gauname **konstruktyviąją matematiką**, kurioje naudojama jau kitokia logika – **intuicionistinė logika**.

Intuityviai, kai sakome duota seka $a_1, a_2, a_3, \ldots$ lyg tai turime suprasti, kad *egzistuoja algoritmas* (mokame, žinome) kaip užrašyti pirmuosius *n* sekos narius. Čia *n* bet kuris natūralusis skaičius.

Jei to nemokame, tai ką reiškia *duota begalinė seka*? Kad mokame 17, gal 20, parašyti pirmųjų sekos narių, o toliau nežinome kokie sekos nariai?

Ką reiškia *duotas iracionalus skaičius a*? Tą patį ką ir duota seka. Koks bebūtų *n*, mokame (egzistuoja algoritmas) parašyti *pirmuosius n skaitmenis*. Tik tuomet ir galima tvirtinti, kad *egzistuoja iracionalusis skaičius a*.

Pavyzdžiui, egzistuoja iracionalusis skaičius $\sqrt{2}$. Nes žinomas algoritmas kaip jį apskaičiuoti. Juo ir nustatau, kad, pavyzdžiui,

jei $n = 3$, tai $\sqrt{2} = 1,41$,
jei $n = 7$, tai $\sqrt{2} = 1,414213$,
jei $n = 11$, tai $\sqrt{2} = 1,4142125624$.

Panašiai su e, π, su šaknimis iš 3, 5, 6, ... T.y. egzistuoja be galo daug iracionalių skaičių. O tiksliau? O tiksliau, tiek, kiek egzistuoja algoritmų.

O kaip su **logika**?

D.Hilbertas buvo universalus matematikas. Dideli jo nuopelnai daugelyje matematikos šakų: geometrijoje, funkcionalinėje analizėje, diferencialiniame skaičiavime, skaičių teorijoje, matematinėje fizikoje, matematinėje logikoje, algebroje. Jis „matė visą matematiką“, o ne kurią nors vieną sritį. Pastebėjo, kad visose matematikos šakose mąstymo taisyklės vienos ir tos pačios. Jų baigtinis skaičius. Aprašė jas. Dabar tai vadinama **pirmos eilės logika**. Įrodė jos neprieštaringumą. Pilnumo jam nepavyko įrodyti. Daugiau kaip po dešimties metų Hilberto loginės sistemos (predikatų logikos skaičiavimo) pilnumą įrodė K.Gödelis.

Po K.Gödelio įrodymo buvo gautas naujas instrumentas tyrimams matematikoje. Apibrėžta tiksli sąvoka *kas yra įrodymas*. Atsirado galimybė:

a) įrodyti, kad kai kurie teiginiai matematikoje nei išvedami, nei paneigiami,

b) suprogramuoti teoremų išvedimo paiešką (iš pradžių buvo tik teoriniai samprotavimai, o vėliau atsirado daug realizuojančių įrodymus programų).

Atsirado nauja matematikos šaka – ***matematinė logika*** *su naujais problemų matematikoje sprendimų metodais*.

Skaitome knygą apie Arvydo gyvenimą. Laikome, kad viskas ir tik tai kas parašyta yra teisinga. T.y. į knygos teiginius žiūrime kaip į aksiomas (laikome, kad turime neprieštaringą sistemą). Norime išsiaiškinti ar *Arvydas buvo Vokietijoje*. Knygoje randame tik vieną sakinį apie jo kelionę į užsienį *Vasarą Arvydas atostogavo užsienyje*. Tvirtinti, kad jis buvo Vokietijoje negalime (nenurodyta kokioje šalyje buvo). Lygiai taip pat negalime tvirtinti, kad jis nebuvo Vokietijoje. Buvo užsienyje. Galbūt Vokietijoje.

Vadinasi, negalime įrodyti nei teiginio *Arvydas buvo Vokietijoje*, nei jo neigimo.
Apibrėžus tiksliai logikos taisykles, kuriomis naudojasi matematikai, įrodyta, kad panaši situacija yra ir matematikoje. Kiekvienoje matematinėje teorijoje (jei ji nėra triviali) galima suformuluoti be galo daug teiginių, kurie nei įrodomi, nei paneigiami. Ne triviali, jei, pavyzdžiui, teorija naudoja aritmetiką arba aibių teoriją pilna apimtimi.

Logikos taisyklės irgi pasirodo paprastos. Paaiškinsiu naudodamas vokiečių matematiko G.Gentzeno predikatų logikos skaičiavimu (tai perdarytas Hilberto skaičiavimas). Jos beveik atitinka tą mąstymą, kuris susiformuoja per matematikos pamokas.

Parašysime keletą jų.

1.Jei reikia įrodyti $\vdash F \wedge G$ (*F* ir *G*), tai reikia įrodyti kaip $\vdash F$, taip ir $\vdash G$. Įrodymą suskaidome į du. Tai užrašoma taisykle (ji skaitoma iš apačios į viršų; apačioje, tai ką turime, viršuje – į ką transformuojame įrodymą).

$$(\vdash \wedge)\ \ \frac{\vdash F \quad \vdash G}{\vdash F \wedge G}$$

2. Jei reikia įrodyti $\vdash F \leftrightarrow G$ (*F* tada ir tiktai tada, kai *G*), turime taisyklę

$$(\vdash\leftrightarrow)\quad \frac{F \vdash G \quad G \vdash F}{\vdash F \leftrightarrow G}$$

(duota *F*, įrodyti *G*; duota *G*, įrodyti *F*).

Su teiginių logikos taisyklėmis viskas aišku. Jas matematikai išmoksta intuityviai, o logikai bei informatikai užrašo taisyklėmis (logikos formulėmis). Jų teisingumą galima patikrinti teisingumo lentelėmis. Ir jos galioja bet kurios eilės logikoje (nebūtinai pirmosios).

O kaip su predikatų logika? Kaip matematikai įrodo, kad tvirtinimas pavidalo $\forall x P(x)$ yra teisingas (tarkime *x* natūralieji skaičiai)? Juk reikia įrodyti, kad $P(0), P(1), P(2), \ldots$ yra teisingi. Bet jų yra be galo daug. Kaip matematikai pagauna begalybę? Kaip išmokyti kompiuterį spręsti tokio pavidalo teoremas?

Prisiminkime kad ir Pitagoro teoremą. Teoremos įrodymų yra labai daug. Bet daugumos esmė tokia: *tarkime a, b, c yra* ***kurie nors*** *stataus trikampio kraštinių ilgiai*. Parodysime, kad $c^2 = a^2 + b^2$ (nesigilinkime į detales: suprantama, *c* yra įžambinė; teisingas ir priešingas teiginys).

Dar kartą: reikia įrodyti, kad su visais ***x, y, z***, o įrodant peršokame prie kurių nors *a, b, c*. Duodama suprasti, kad *a, b, c* yra konkretūs kraštinių ilgiai. Bet juk jie konkretūs būtų, jei tai būtų skaičiai, pavyzdžiui 3, 4, 5. Bet *a, b, c* yra kintamieji, o ne skaičiai. Jie lygiai tokie patys kintamieji kaip ir *x, y, z*, tik pažymėti kitomis raidėmis.

Nelabai aišku, kodėl toks perėjimas visiems moksleiviams ir studentams suprantamas, nekyla abejonių. Bet jame glūdi labai svarbi mintis: $\forall x P(x)$ įrodomas (***visiems x***), jei **įrodomas šablonas** $P(a)$ (***kuriam nors a***). Tai reiškia, kad pakeitęs įrodyme a bet kuriuo konkrečiu skaičiumi, pavyzdžiui 25, pakaks kartoti tą patį įrodymą $P(25)$ kaip ir su a (su kai kuriais konkrečiais skaičiais galbūt įrodymas bus dar trumpesnis). Taigi, perėjimas yra grynai *sintaksinis*. Tai pastebėję logikai naudoja taisyklę:

$$\frac{\vdash P(a)}{\vdash \forall x P(x)}$$

Suvaržytuosius kintamuosius žymint *x, y, z*, o laisvuosius - *a, b, c*. Nesunku išmokyti kompiuterį mąstyti.

Bet, sakysite, jei matematikai tik taip *pagauna begalybę*, tai gali taip būti, kad visi $P(0), P(1), P(2), \ldots$ teisingi, o šablonas $P(a)$ neįrodomas. Tuomet neįrodoma ir $\forall x P(x)$. Taip, tai tiesa. Tarkime jus norite įrodyti teoremą $\forall x P(x)$. Jei *a* lyginiai skaičiai – mokate įrodyti. Jei *a* dalosi iš 3 taip pat mokate, bet įrodymas jau kitoks. Jei *a* dalosi iš 5, tai įrodymas dar kitoks, negalit jų apjungti į vieną, reikia skaidyti į atvejus. Ir taip toliau. Bet skirtingų atvejų bus be galo daug. $\forall x P(x)$ bus neįrodomas (reikia šablono).

Matematinė indukcija yra atskiras šablono atvejis. Ji taikoma, kai turime aibę, kurioje įvesta griežta tvarka ir yra funkcija nurodanti pagal kiekvieną *n* sekantį (paskesnįjį) narį.

Turbūt pastebėjote, kad logikos taisyklės yra sintaksinės. Žiūrima tik į formulės pavidalą, o ne į turinį. Pavyzdžiui, jei reikia rasti išvedimą $\vdash F \wedge G$, tai bus ieškomi $\vdash F, \vdash G$ išvedimai. Kas aprašoma formulėmis *F, G* mus nedomina. Tai galima palyginti su užduotimi: vieno litro talpos butelius sudėkite į viršutinę lentyną, 0,75 litro talpos butelius į vidurinę lentyną, o 0,5 litro talpos butelius į apatinę lentyną. Atliksime darbą atsižvelgdami tik į butelių talpą (nustatom pagal etiketes ant butelių), o butelių turinys tos pačios talpos buteliuose gali būti įvairus (galbūt kai kurie tušti).

Pavyzdžiui, Zermelo-Fränkelio aibių teorija *ZF* aprašoma pirmos eilės predikatų logikos formulėmis, kuriose tėra tik simbolis $\emptyset$ ir du dviviečiai predikatai $=, \in$. $\emptyset$ tėra simbolis, tenkinantis tam tikras aksiomas, kurį kaip norit, taip ir interpretuokit – tuščia aibė, o gal ir ne. Kompiuteriui, kaip ir logikui, tas pats. Yra aksiomos, taisyklės bei tikslas (teorema, kurią reikia įrodyti). Žiūrima į sintaksę ir ieškomas teoremos įrodymas.

Elementarioje (formalioje loginėje teorijoje) plokštumos geometrijoje pradinėmis sąvokomis yra taškai, žymima juos raidėmis *a, b, c, . . .*, tiesė - du taškai, dvi raidės $ab, ac, bc, \ldots$. Ir nereikia jokios sampratos, kas yra taškas ar tiesė. Parašyta ac, vadinasi tai *tiesė* (o ką reiškia tiesė, nesvarbu) per taškus *a* ir *c*. Be abejo, yra aksiomos. Bet jos tarp simbolių. Pavyzdžiui, $ab = ba$.
Taip išvengiama neaiškių apibrėžimų: *taškas – dydžio neturintis geometrijos objektas, kurį apibūdina jo padėtis*.
Vadinasi, atkarpa nuo 0 iki 1, susidedanti iš *objektų, neturinčių dydžio*, turi dydį (ilgį) lygų 1 (*formalioje geometrijoje pabrėžiama, kad visos figūros plokštumoje yra plokštumos taškų poaibis*). O atkarpos dydis, nuo 0 iki 2, lygus 2. Nors tarp abejas atkarpas sudarančių objektų, neturinčių dydžių, yra bijekcija.

Rinkimo aksioma – jei aibės elementais yra aibės (baigtinės ar begalinės), tai visada galima parinkti po elementą iš kiekvienos netuščios aibės ir sudaryti naują *atstovų aibę*.

Ši aksioma plačiai naudojama matematikoje. Keletas rezultatų, kuriuose buvo panaudota rinkimo aksioma:
1. G.Kantoras 1887 metais įrodė, kad bet kurioje begalinėje aibėje yra skaitusis poaibis.
2. Bet kurioje aprėžtoje realiųjų skaičių aibėje yra konverguojanti seka.
3. Realiųjų skaičių konstravimas naudojant Peano aritmetiką.

Bet teiginys, vadinamas rinkimo aksioma, įrodomas atvejais, kai aibė yra baigtinė arba skaičioji, ir *nei įrodomas, nei paneigiamas* atveju, kai aibė yra kontinuumo galios (tai įrodyta tik XX amžiuje). Todėl teiginys ir vadinamas aksioma (laikomas teisingu be įrodymo). Kaip matome, G.Kantoras naudojo rinkimo aksiomą dar iki to, kai buvo gautas rezultatas, kad *ji nei įrodoma, nei paneigiama* (todėl ir prieštaravimo negavo). Lygiai taip pat turi teisę egzistuoti neprieštaringa matematinė teorija, kurioje teiginys (dabar vadinamas rinkimo aksioma) yra klaidingas. Kurią norite, tą ir rinkitės. Įrodyta, kad jei aibių teorija ZF nėra prieštaringa, tai galima gauti dvi neprieštaringas teorijas: viena yra ZF su rinkimo aksioma, kita - ZF su rinkimo aksiomos neigimu.

O ką renkamės? XVII a. gyvenusio olandų filosofo Spinoza pastebėjimas kaip žmonės nustato teiginių teisingumą:
Šnekamojoje kalboje ***teiginio paprastumas*** *lemia, kad jis nebyliai priimamas kaip teisingas.*

Daugelis žmonių nepakenčia dviprasmybių ir todėl patikima greitai, o abejojimas ateina lėtai ir yra neįprasta būsena. Teiginys priimamas kaip teisingas, jei jo ***supratimui nereikia daug pastangų****. Dauguma nereikalauja teiginio įrodymo, nes tai prieštarauja įgimtam polinkiui laikyti teisinga tai, ką greitai perprantame.*

D.Hilbertas įrodymams kėlė *patikimumo reikalavimus*. Vienas tokių buvo *niekada negalima tvirtinti apie objekto egzistavimą, jei nenurodytas jo konstravimo būdas*.

Jei įrodymais pripažinsime tik patikimus (pakeisime mąstymo taisykles), tai gausim kitokią matematiką.
Teorema. *Egzistuoja du tokie iracionalieji a, b, kad* a^b *yra racionalusis skaičius*.

Įrodymas. Tarkime $\sqrt{2}^{\sqrt{2}}$ yra racionalus skaičius. Tuomet parinkę $a = b = \sqrt{2}$, gauname, kad $a^b = \sqrt{2}^{\sqrt{2}}$yra racionalusis skaičius, o a, b – iracionalieji skaičiai. Priešingu atveju, jei $\sqrt{2}^{\sqrt{2}}$ yra iracionalus, imkime $a = \sqrt{2}^{\sqrt{2}}$, o $b = \sqrt{2}$. Tada $a^b = (\sqrt{2})^2 = 2$, t.y. a^b yra racionalus skaičius. Teorema įrodyta.
Tai kam vis dėlto lygūs a ir b? Nors ir įrodėme egzistavimą tokių skaičių, bet kam jie lygūs, nežinome. Toks įrodymas vadinamas nekonstruktyviuoju įrodymu. Įrodoma, kad egzistuoja koks nors objektas, bet iš įrodymo neišplaukia algoritmas, kaip jį rasti. Šios teoremos atveju galima rasti kitą, konstruktyvų jos įrodymą. Jis yra ilgesnis ir sudėtingesnis. Įrodyta, kad $a = \sqrt{2}^{\sqrt{2}}, b = \sqrt{2}$.

Teorema (Bolzano-Weierstrasso). *Iš kiekvienos aprėžtos skaičių sekos galima išrinkti konverguojantį posekį*.

Ar galima aprašyti algoritmą, kuriuo remiantis rastumėm konverguojantį posekį? Deja, toks neegzistuoja.
Nėra algoritmo, kuris pagal bet kurią duotą (t.y. duota programa generuojanti sekos narius) *begalinę seką iš intervalo [0,2], nurodytų, kuriame iš intervalų [0,1), [1,2] yra be galo daug sekos narių*.

Reziumė.
Matematinė teorija panaši į loginį žaidimą, *kuriame fiksuotos aksiomos (žaidžiantysis galbūt pats dalį jų sugalvojo) ir logika (kad ir neapibrėžta taisyklėmis, intuityvi). Baigtinių aibių atvejais daugelis teorijų turi taikymus realiame pasaulyje. Kaip ir bet kuris žaidimas,* ***neprieštaringa matematikos teorija turi teisę egzistuoti***.

Egzistuoja skaitinė aibė, kuri nėra nei skaičioji, nei kontinuumo galios.
1963 m. amerikietis P. Cohen įrodė, kad šį teiginį negalima (aksiominėje aibių teorijoje) nei įrodyti, nei paneigti. O tai reiškia, kad, jei prie aksiomų sistemos ZF prijungsime teiginį *egzistuoja skaitinė aibė, kuri nėra nei skaičioji, nei kontinuumo galios*, tai turėsime naują įdomų loginį žaidimą (matematinę teoriją), nes šio teiginio neigimas neįrodomas. **Tereikia tik norinčių žaisti**.

Kai pavargsti nuo *loginio žaidimo*, įjungi kompiuterį, prisiminęs, kad reikia paskaičiuoti funkcijos lokalųjį minimumą ir grįžti iš *realiųjų skaičių aibių* į *realųjį gyvenimą* – **baigtinę racionaliųjų skaičių aibę**.

1 Skyrius
TEIGINIŲ LOGIKA

1.1 Loginės išvados

Sąvoką *dirbtinis intelektas* 1955 m. sugalvojo amerikietis J. McCarthy, kartu su M. Minsky įkūręs pirmąją **dirbtinio intelekto** laboratoriją. Jis yra ir programavimo kalbos LISP autorius. *Intelektas* - gebėjimas logiškai mąstyti. Ką kompiuteris turi mokėti, kad galėtum sakyti jog tai dirbtinis intelektas, kad *kompiuteris mąsto*? Diferenciavimas, veiksmų su matricomis atlikimas nėra dirbtinio intelekto požymiai. Iš pat pradžių susiformavo dvi problemų sritys, kurias suprogramavus galima sakyti, kad tai dirbtinio intelekto programos: *žaidimas šachmatais ir matematikos teoremų įrodymas*. Mus domins antroji problema. Vėliau atsirado ir kitokio tipo uždavinių. Bet keletą dešimtmečių po 1955 metų tik šių dviejų problemų sprendimas dominavo straipsniuose (ypač *naujų matematikos teoremų įrodymo paieška naudojant kompiuterius*) apie dirbtinį intelektą.

Netgi XIX amžiaus pabaigoje teoremų įrodymų stilius vis dar labai skyrėsi nuo šiuolaikinių. Neretai įrodymuose pagrindiniu argumentu buvo *tai akivaizdu*. Įrodymai nebuvo formalūs. Įrodymais buvo siekiama įtikinti, o ne pagrįsti logiškai, kad teiginys yra teisingas. Dar prisidėjo nesusipratimai su aibių teorija, sukėlusiai matematikoje krizę dėl joje aptiktų paradoksų.

Esant tokiai situacijai, XX amžiaus pradžioje žymus vokiečių matematikas Davidas Hilbertas paskelbė programą *kitokiai matematikai* kurti. Pagrindinius Hilberto programos reikalavimus **idealiai matematikai** galima nusakyti taip:

• Kiekviena matematikos teorija turi būti aksiomatizuojama (*visos matematikos aksiomatizavimas*). Kiekvienoje matematikos teorijoje turi būti išvardinti teiginiai laikomi teisingais be įrodymo (aksiomos).

• Matematikos teiginiai turi būti parašyti tikslia formalia kalba.

• Visi teisingi teiginiai turi būti formaliai įrodomi.

• Funkcijos apskaičiuojamos.

• Įrodymai apie *idealius objektus* (tokius, pavyzdžiui, kaip begalinės aibės) turi būti gaunami nenaudojant pačių idealių objektų.

Kaip matome, vienas iš D.Hilberto programos tikslų - *matematikos teiginiai turi būti pa-*

rašyti tikslia formalia kalba. Gerokai anksčiau vokiečių matematikas G.Leibnicas (1646-1716) jau buvo iškėlęs idėją sukurti universalią visai matematikai kalbą ir ta kalba formalizuoti matematinius įrodymus. Taip buvo norima išspręsti vis pasitaikančią problemą, kai vienas „įrodo“ teoremą, kitas „įrodo“ jos neigimą, o kuris teisus neįmanoma nustatyti. Jei turime funkciją parašytą formalia kalba, pavyzdžiui $y = \sin x$), tai bet kurios šalies, nepriklausomai nuo jo gimtosios kalbos, matematikas vienareikšmiškai supranta kas tai per funkcija.

20 amžiaus pradžioje jau buvo susiformavus nuomonė dėl kalbos, priimtiniausios laikyti universalia, tai **matematinės logikos formulių kalba**.

Knygoje aprašysime mąstymo taisykles, kurias naudoja matematikai. Turėdami jas galima kurti ir programas. Kompiuteris modeliuos matematiko mąstyseną. Taisyklės beveik tos pačios kaip ir naudojamos šnekamojoje kalboje. Tik matematikoje vartojamos tikslesnės sąvokos ir yra kai kurios specifinės taisyklės (pavyzdžiui, matematinės indukcijos). Kaip galima nustatyti mąstymo taisykles? Paaiškinsiu pavyzdžiu. Jūs stebite žaidimą (tarkime šachmatų) ir norite nustatyti žaidimo taisykles. Matydami žaidžiančiuosius jus pastebite: „ėjimai atliekame paeiliui“, „žaidimas pradedamas baltomis figūromis“ ir t.t. Stebėdami ir analizuodami šachmatų partijas jus galite aprašyti žaidimo taisykles. Panašiai ir su matematikų mąstymu. Tik tų taisyklių aprašymui prireikė poros tūkstančių metų. Nelengva buvo jas nustatyti.

Turime du samprotavimus:

Jei skaičius dalosi iš šešių, tai jis dalosi iš dviejų
Skaičius nesidalo iš dviejų

Vadinasi, skaičius nesidalo iš šešių

Jei trikampis status, tai įžambinės ilgio kvadratas yra lygus jo statinių ilgių kvadratų sumai
Įžambinės ilgio kvadratas nelygus statinių ilgių kvadratų sumai
---
Vadinasi, trikampis nėra status

Virš brūkšnio parašėme **prielaidas** (tai kas yra duota), o apačioje brūkšnio **išvadą**. Jei abi prielaidos teisingos, tai išvada irgi teisinga. Samprotavimų turiniai skirtingi, o mąstymo forma ta pati:

Jei p, tai q
Nėra q

Vadinasi, nėra p

Čia p, q teiginiai (tvirtinamojo pobūdžio sakiniai, kurių atžvilgiu prasmingas klausimas, teisingas jis ar klaidingas). Ar tokia mąstymo forma yra logikos dėsnis? Taip, tai logikos dėsnis. Nesvarbu kokie būtų p, q teiginiai (šnekamosios kalbos ar iš matematikos srities). Štai ir susipažinom su vienu logikos dėsniu.

Logines jungtis „*jei*, ..., *tai* ...“, „*nėra*“ bei kitas parašytas lietuvių kalba pakeisime simboliais naudojamais tiksliuose moksluose ir vadinsime juos **loginėmis operacijomis**. Teiginius žymime mažosiomis lotyniškomis raidėmis, bei raidėmis su indeksais. Dažniausiai naudosime raides p, q, r, s, taip pat su indeksais: $p_0, q_0, r_0, s_0, p_1, q_1, r_1, s_1, \ldots$. Jie gali būti teisingi (tuomet sakysime, kad jų reikšmė *t*) arba klaidingi (tuomet sakysime, kad jų reikšmė *k*). Jei teiginys p teisingas, tai rašysime $p = t$. Jei teiginys klaidingas, tai $p = k$. Į teiginius žiūrime kaip į

kintamuosius (nebent tai yra konkretus tvirtinamojo pobūdžio sakinys), kurių reikšmių aibe yra $\{t,k\}$. Mes juos vadinsime **loginiais kintamaisiais**.

Apibrėšime šias logines operacijas: *neigimą, konjunkciją, disjunkciją, griežtąją disjunkciją, implikaciją, ekvivalentumą*, o vėliau *Shefferio funkciją* ir *Peirco operaciją*. Logines operacijas pritaikę teiginiams, gauname *sudėtinius teiginius*. Loginius kintamuosius bei sudėtinius teiginius vadinsime **formulėmis**. Formules rašysime didžiosiomis lotyniškomis raidėmis.

Konjunkcija žymima simboliu $\wedge$. Užrašą $p \wedge q$ skaitome „ p ir q“.

Disjunkcija žymima simboliu $\vee$. Užrašą $p \vee q$ skaitome „p arba q“.

Griežtoji disjunkcija žymima simboliu $\dot{\vee}$. Užrašą $p\dot{\vee}q$ skaitome „arba p, arba q“.

Implikacija žymima $\rightarrow$. Užrašą $p \rightarrow q$ skaitome „jei p, tai q“ arba „iš p išplaukia q“.

Ekvivalentumas žymimas $\leftrightarrow$. Užrašą $p \leftrightarrow q$ skaitome „p ekvivalentus q“, arba „p ir q ekvivalentūs“.

Aprašytųjų formulių reikšmės nusakomos **teisingumo lentele**:

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p\dot{\vee}q$	$p \rightarrow q$	$p \leftrightarrow q$
t	t	k	t	t	k	t	t
t	k	k	k	t	t	k	k
k	t	t	k	t	t	t	k
k	k	t	k	k	k	t	t

Iš pateiktos lentelės matome, kad:

• formulės $\neg p$ reikšmė priešinga teiginio p reikšmei: jei teiginio p reikšmė lygi k, tai formulės $\neg p$ reikšmė lygi t, o jei teiginio p reikšmė yra k, tai formulės $\neg p$ reikšmė yra *t*,

• formulė $p \wedge q$ teisinga tada ir tiktai tada, kai abu teiginiai p, q teisingi,

• formulė $p \vee q$ klaidinga tada ir tiktai tada, kai abu teiginiai p, q klaidingi,

• formulė $p\dot{\vee}q$ teisinga tada ir tiktai tada, kai teisingas tik vienas iš teiginių p, q,

• formulė $p \rightarrow q$ klaidinga tada ir tiktai tada, kai teiginys p teisingas, o teiginys q klaidingas,

• formulė $p \leftrightarrow q$ teisinga tada ir tiktai tada, kai abu teiginiai p, q teisingi arba abu klaidingi.

Toks keistas implikacijos apibrėžimas reikalauja paaiškinimo.

Netikslus atsakymas būtų toks. *Paaiškinimas pavyzdžiu.* Sakinys *jei* $x < y$, *tai* $x < y + 8$ laikomas teisingu bet kurioms x, y reikšmėms. Galima parinkti tokius natūraliuosius x, y, kad tvirtinimas $x < y$ (t.y. p) būtų klaidingas, o $x < y + 8$ (t.y. q) vienu atveju būtų teisingas, kitu klaidingas (pavyzdžiui, $x = 6, y = 3; x = 10, y = 1$), bet nėra tokių x, y, kad p būtų teisingas, o q klaidingas. Formule $p \rightarrow q$ mes netvirtiname, kad abu p ir q būtinai teisingi. Mes

tik nurodome, kad jei p teisingas, tai būtinai ir q bus teisingas.

Jei skaičius lyginis, tai jis dalijasi iš 2.

Tikslesnis atsakymas būtų toks. 1879 metais vokiečių matematikas G.Frege, (kai dar nebuvo žinoma, kad logikos dėsnius galima nustatyti ir teisingumo lentelėmis) sukūrė skaičiavimą, kuriame aksiomomis (formulėmis) tiksliai aprašė kaip naudojama implikacija matematikoje. Tik praėjus šešeriems metams Ch.S.Peirc'as sugalvojo teisingumo lenteles. Aprašytoji teisingumo lentelė užpildyta remiantis G.Frege's aksiomomis *implikacijai*. Ji tokia, kokią ir siūlau jums naudotis. Esmė tame, kad matematikai įrodymuose naudoja *trečio negalimo dėsnį*. Kiekvienas teiginys matematikoje buvo (ir tebėra) laikomas arba teisingu, arba klaidingu, nors praeitame šimtmetyje įrodyta, kad matematikoje yra be galo daug teiginių, kurių negalima nei įrodyti, nei paneigti. Pateiktas implikacijos apibrėžimas naudojamas matematikos teoremų įrodymuose su trečio negalimo dėsniu.

Tikslus atsakymas būtų toks. Susitaikykite su tokiu **keistu** implikacijos apibrėžimu.

Nors susitaikė ne visi. Implikaciją $p \to q$ siūloma interpretuoti taip: $p \to q$ įrodomas, jei turint p įrodymą galima rasti q įrodymą. Kalbant apie teiginius naudojamos ne sąvokos *teisingas, klaidingas o įrodomas, neįrodomas*. Šiuo atveju loginės operacijos neaprašomos teisingumo lentelėmis. Gaunamos kitokios mąstymo taisyklės. Tokia logika vadinama *intuicionistine (konstruktyviąja)*. Ji dažniausiai naudojama *informatikų*. Šioje logikoje nėra vietos tvirtinimui, kad *iš klaidingo teiginio išvedamas bet koks*.

Knygoje pagrindinį dėmesį skirsime klasikinei logikai, bet aprašysime ir konstruktyvią.

Loginių kintamųjų apibrėžimo aibe kai kada patogu laikyti aibę $\{1, 0\}$, t.y. t keisti vienetu, o k keisti nuliu.

Iš loginių kintamųjų, loginių operacijų ir skliaustų konstruojame sudėtingesnius reiškinius ir vadiname juos *formulėmis*. Jei neigimas yra prieš loginį kintamąjį, tai skliaustų nerašome. Formulių pavyzdžiai:

$(p \vee q) \to \neg(\neg p \vee q)$, $\neg p$, $\neg(p \vee q)$, $(p_1 \leftrightarrow p_2) \vee \neg\neg(p_1 \wedge (p_3 \vee \neg p_1))$.

XVII a. gyvenusio olandų filosofo Spinoza pastebėjimas kaip žmonės nustato teiginių teisingumą:

Šnekamojoje kalboje teiginio paprastumas lemia, kad jis nebyliai priimamas kaip teisingas. Daugelis žmonių nepakenčia dviprasmybių ir todėl patikima greitai, o abejojimas ateina lėtai ir yra neįprasta būsena. Teiginys priimamas kaip teisingas, jei jo supratimui nereikia daug pastangų. Dauguma nereikalauja teiginio įrodymo, nes tai prieštarauja įgimtam polinkiui laikyti teisinga tai, ką greitai perprantame.

Turime formulę F: $\neg p \to (p \vee (q \leftrightarrow \neg r))$. Ar ji teisinga? Atsakymas priklausys nuo to kokias reikšmes priskirsime kintamiesiems p, q, r. Jei $p = q = t$, $r = k$, tai $F = t$ (nesunku tai paskaičiuoti naudojantis loginių operacijų teisingumo lentelėmis). Sakysime, kad su **interpretacija** $p = q = t, r = k$ formulė F teisinga. Jei $p = q = r = k$, tai su šia interpretacija formulė

F klaidinga. O kiek iš viso yra formulės interpretacijų, jei joje n skirtingų loginių kintamųjų. Iš viso bus 2^n. Nagrinėjamoje formulėje F yra trys kintamieji. Taigi, iš viso yra $2^3 = 8$ interpretacijos. Išrašysime jas visas ir paskaičiuosime formulės reikšmes. Gauname **formulės** F **teisingumo lentelę**.

p	q	r	$\neg p$	$\neg r$	$q \leftrightarrow \neg r$	$p \vee (q \leftrightarrow \neg r)$	$\neg p \rightarrow (p \vee (q \leftrightarrow \neg r))$
t	t	t	k	k	k	t	t
t	t	k	k	t	t	t	t
t	k	t	k	k	t	t	t
t	k	k	k	t	k	t	t
k	t	t	t	k	k	k	k
k	t	k	t	t	t	t	t
k	k	t	t	k	t	t	t
k	k	k	t	t	k	k	k

Matome, kad su kai kuriomis interpretacijomis formulė teisinga, o su kai kuriomis – formulė klaidinga.

Dvi formules F, G, kuriose yra vienodi loginiai kintamieji, vadiname ekvivalenčiosiomis (*rašome* $F \equiv G$), *jei su bet kuria interpretacija galioja: arba abi formulės teisingos, arba abi klaidingos*.

Šis ekvivalentumo simbolis $\equiv$ nenaudojamas kaip loginė operacija (knygoje tik $\leftrightarrow$ yra ekvivalentumo loginė operacija).

Pavyzdžiui, formulės $p \rightarrow q$, $\neg q \rightarrow \neg p$ yra ekvivalenčios (t.y. $p \rightarrow q \equiv \neg q \rightarrow \neg p$).

p	q	$\neg p$	$\neg q$	$p \rightarrow q$	$\neg q \rightarrow \neg p$
t	t	k	k	t	t
t	k	k	t	k	k
k	t	t	k	t	t
k	k	t	t	t	t

Stulpeliai atitinkantys $p \rightarrow q$ ir $\neg q \rightarrow \neg p$ reikšmėms sutampa. Ekvivalenčių formulių pavyzdžiai:

$$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r,$$
$$p \vee (q \vee r) \equiv (p \vee q) \vee r.$$

Formulių ekvivalentumas nustatomas sudarant jų teisingumo lenteles. Kaip matome, formulių reikšmės nuo suskliaudymo tvarkos nepriklauso, jei jos yra pavidalo $F_1 \wedge \ldots \wedge F_n$ arba $F_1 \vee \ldots \vee F_n$. Todėl tokio pavidalo formulėse skliaustus praleisime.

Aprašėm formulių ekvivalentumo nustatymo metodą **naudojantis teisingumo lentelėmis**. Teisingumo lentelėmis retai naudojamasi dėl metodo sudėtingumo. Jis tinka tik atvejais, kai formulėje nedaug kintamųjų. 2^n yra greitai auganti funkcija.

Informaciją (žinias, duomenis) apie tai kas duota (t.y. kokios yra prielaidos) bei ką reikia įrodyti pateikiama kompiuteriui formulių pavidalu. Duomenų užrašymas formulėmis vadinamas **formalizavimas**. Formalizavimo tikslas – paversti matematikos teiginius tiksliais matematiniais objektais.

Panagrinėsime kelis pavyzdžius. **Formalizuoti**:
1. *Elena pavargusi arba serga. Jei pavargusi, tai ji irzli. Ji nėra irzli.* Vadinasi, *ji serga*.

Turimą informaciją ir tai ką norime sužinoti aprašysime formulėmis. Teiginius žymime mažosiomis lotyniškomis raidėmis:

p – Elena pavargusi.
s – Elena serga.
i – Elena irzli.

Užduotį užrašysime **sekvencija**, reiškiniu pavidalo $A_1, A_2, \ldots, A_n \vdash B$.

$A_1, A_2, \ldots, A_n, B$ yra formulės. Simbolis $\vdash$ vadinamas *išvedimo* (*įrodymo*) simboliu. Jo kairėje rašome prielaidas, o dešinėje išvadą (tai, ką reikia įrodyti).

Pavyzdį parašysime tokia sekvencija (užduoties parašymas *sekvencija* ir yra užduoties *formalizavimas*):

$$p \vee s, p \to i, \neg i \vdash s$$

Prielaidų tvarka nesvarbi. Kairėje nuo simbolio $\vdash$ esančias formules galima rašyti bet kuria tvarka.

2. *Tyrimu įrodyta*:
a) *kaltas bent vienas iš A, B, C*,
b) *jei A kaltas, tai ir C kaltas*,
c) *jei C nekaltas, tai ir B nekaltas*.
Klausiama: *ar kaltas C*?

Žymėsime:
a – kaltas A,
b – kaltas B,
c – kaltas C.

Užduoties formalizavimas. Ji užrašoma sekvencija:

$$a \vee (b \vee c), a \to c, \neg c \to \neg b \vdash c$$

3. *Jei skaičius dalijasi iš 2 ir 3, tai jis dalijasi ir iš 6. Duotas skaičius nesidalija iš 2, bet dalijasi iš 3. Vadinasi, skaičius nesidalija iš 6*.

Žymime:
p_2 – skaičius dalijasi iš 2 ir 3,
p_3 – skaičius dalijasi iš 3,
p_6 – skaičius dalijasi iš 6.
Užduotis užrašoma sekvencija

$$(p_2 \wedge p_3) \to p_6, \neg p_2 \wedge p_3 \vdash \neg p_6$$

Pastaba. Įprastinėje kalboje konjunkcija reiškiama ne tik žodžiu *ir*. Kai kuriais atvejais loginiu požiūriu jungčiai *ir* lygiaverčios šios jungtys: „*o*“, „*bet*“, „*tačiau*“, „*nors*“. Mūsų pavyzdyje buvo „bet“.

Dar vienas pavyzdys. *Atskrido pulkas žvirblių ir varnų. Visi žvirbliai nutūpė ant elektros laidų* ($ž$), **o** *visos varnos ant medžių viršūnių* (v). Toks teiginys užrašomas formule $ž \wedge v$.

4. *Jei keturkampio priešingosios kraštinės lygios ir lygiagrečios, tai keturkampis yra lygiagretainis. Duoto keturkampio priešingosios kraštinės lygios arba lygiagrečios*. Vadinasi, *keturkampis nėra lygiagretainis*.

Žymime:
p – keturkampio priešingosios kraštinės lygios,
q – keturkampio priešingosios kraštinės lygiagrečios,
l – keturkampis yra lygiagretainis.

Užduotis užrašoma sekvencija

$$(p \wedge q) \to l, p \vee q \vdash \neg l$$

5. *Nagrinėjamasis erdvinis kūnas yra ritinys, kūgis arba rutulys. Jei kūnas yra ritinys arba kūgis, tai jis gautas sukant tam tikro pavidalo daugiakampį apie vieną jo kraštinių . Kūnas nėra nei kūgis, nei rutulys. Vadinasi, jis yra gautas sukant daugiakampį apie vieną jo kraštinę ir yra ritinys*.

Žymime:
r – kūnas yra ritinys,
g – kūnas yra kūgis,
l – kūnas yra rutulys,
d – kūnas gautas sukant tam tikro pavidalo daugiakampį apie vieną jo kraštinių.

Užduotis užrašoma sekvencija

$$r \vee (g \vee l), (r \vee g) \to d, \neg g \wedge \neg l \vdash d \wedge r.$$

Užduotis formalizavom. Dabar laikas jas spręsti. Tik neskubėkite. Užduotys nėra sudėtingos ir tikiu, kad jus jau žinot atsakymus. Tikslas – aprašyti **mąstymo taisykles**. Turėdami jas, galėsime suprogramuoti panašaus tipo uždavinių sprendimą. Juk yra uždavinių ir žymiai sudėtingesnių, kurių aprašymui reikia daug formulių, o sprendimui daug laiko. Lengvesniais uždaviniais norime tik paaiškinti **pačią idėją** kaip galima tą darbą perkelti kompiuteriui ant pečių.

Šiek tiek žinių iš istorijos.

1955 ir 1956 metų bėgyje, t.y. praėjus dešimčiai metų po pirmųjų kompiuterių pasirodymo, amerikiečiai A. Newellis, H. Simonas ir J.C. Shaw'as sukūrė programą Logic Theorist, kuria įrodė 38 iš 52 teiginių logikos teoremų aprašytų tritomyje Principia Mathematica. Viešas programos demonstravimas įvyko 1956 m. rugpjūčio 9 dieną. Tai buvo pirmoji ***dirbtinio intelekto programa*** *pasaulyje. M.Davisas sukūrė panašią programą anksčiau, bet spaudoje jos aprašymas pasirodė šiek tiek vėliau negu Logic Theorist. Abi programos naudojo kai kuriuos*

specifinius metodus, tinkamus tik nagrinėjamiems uždaviniams spręsti ir todėl neturėjo įtakos vėlesniems darbams iš automatinio teoremų išvedimo srities. Automatinio teoremų išvedimo pradininku laikomas kinų kilmės amerikietis X. Wangas. IBM kompiuteriu 1957 m. jis įrodė apie 400 teiginių logikos ir pirmosios eilės logikos su lygybės predikatu dėsnių, aprašytų tritomyje Principia Mathematica.

Jau mokame formalizuoti matematikos užduotis. T.y. aprašyti tai, kas yra duota, ir tai, ką reikia įrodyti, teiginių logikos formulėmis. Dabar pateiksime loginių taisyklių rinkinį, kuriuo naudojantis ir tikrinsime ar sekvencijos išvedamos.

Žinau, kad jus **nežinote**, kad teiginių logikos skaičiavimo taisykles jau **žinote**. Teisingo loginio mąstymo pagrindai susiformuoja jau mokykloje per tiksliųjų mokslų pamokas. Jau žinote taisykles, tik niekas jums nepateikė jų formaliu (išreikštiniu) pavidalu. Pavyzdžiui, reikia įrodyti teiginį *A ir B*. Be abejo žinote, kad įrodymas skaidomas į du: įrodyti *A*, įrodyti *B*. Tai aprašoma taisykle:

$$\frac{\vdash F \quad \vdash G}{\vdash F \wedge G}$$

Tai, ką reikia įrodyti, rašome brūkšnio apačioje. Įrodymas skaidomas į du.

Parašysime pilną sąrašą taisyklių. Jas aprašė vokiečių matematikas G.Gentzenas. Taisyklių rinkinį pavadino **sekvenciniu skaičiavimu**. Visų pirma pateiksime tikslų sekvencijos apibrėžimą.

Apibrėžimas. *Sekvencija vadiname reiškinį pavidalo* $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$. *Reiškinyje* $A_i (i = 1, \ldots, n)$ *bei* $B_i (i = 1, \ldots, m)$ *yra formulės. Sekvencijoje turi būti bent viena formulė, todėl* $n + m \neq 0$.

$\Gamma, \Gamma_1, \Gamma_2, \Delta, \Delta_1, \Delta_2$ žymi baigtines formulių sekas (jos gali būti ir tuščios). Sekvencijoje $\Gamma \vdash \Delta$

Pastaba *dėl sekvencijos* $A, \Gamma \vdash \Delta, B$ rašymo. Šį užrašą reikia suprasti taip: *antecedente* yra sąrašas formulių (gali būti ir viena formulė *A*, tuomet Γ būtų tuščias sąrašas), tarp kurių yra formulė *A* (ji, suprantama, nebūtinai pirma tame sąraše). Analogiškai ir su formule *B* sukcedente.

Kodėl sukcedente sekvencijos apibrėžime yra daugiau kaip viena formulė? Matematikams tai neįprasta. Štai kodėl: pradinėje sekvencijoje, kaip taisyklė, yra *viena formulė*. Bet skaičiavimo eigoje atsiranda ir daugiau (toks jau Gentzeno skaičiavimas).

Sekvencinis skaičiavimas G.

Aksiomos: $F, \Gamma \vdash \Delta, F$

Loginių operacijų taisyklės:

$$(\neg \vdash) \quad \frac{\Gamma, \vdash \Delta, F}{\neg F, \Gamma \vdash \Delta} \qquad\qquad (\vdash \neg) \quad \frac{F, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg F}$$

$(\neg \vdash)$ ir $(\vdash \neg)$ yra taisyklių vardai (neigimo antecedente, neigimo sukcedente)

$$(\to\vdash)\ \frac{\Gamma \vdash \Delta, F \quad G, \Gamma \vdash \Delta}{F \to G, \Gamma \vdash \Delta} \qquad (\vdash\to)\ \frac{F, \Gamma \vdash \Delta, G}{\Gamma \vdash \Delta, F \to G}$$

$$(\wedge\vdash)\ \frac{F, G, \Gamma \vdash \Delta}{F \wedge G, \Gamma \vdash \Delta} \qquad (\vdash\wedge)\ \frac{\Gamma \vdash \Delta, F \quad \Gamma \vdash \Delta, G}{\Gamma \vdash \Delta, F \wedge G}$$

$$(\vee\vdash)\ \frac{F, \Gamma \vdash \Delta \quad G, \Gamma \vdash \Delta}{F \vee G, \Gamma \vdash \Delta} \qquad (\vdash\vee)\ \frac{\Gamma \vdash \Delta, F, G}{\Gamma \vdash \Delta, A \vee B}$$

$$(\leftrightarrow\vdash)\ \frac{\Gamma \vdash \Delta, F, G \quad F, G, \Gamma \vdash \Delta}{F \leftrightarrow G, \Gamma \vdash \Delta} \qquad (\vdash\leftrightarrow)\ \frac{F, \Gamma \vdash \Delta, G \quad G, \Gamma \vdash \Delta, F}{\Gamma \vdash \Delta, F \leftrightarrow G}$$

Taisyklė vadinama **leistina**, jei, prijungę ją prie sekvencinio skaičiavimo taisyklių, išvedamų sekvencijų aibė nekinta.

Kam leistinos taisyklės reikalingos? Tiesiog turėjimas tokių taisyklių (be kurių galima ir apsieiti) kai kuriais atvejais supaprastina išvedimą.

Leistinos sekvenciniame skaičiavimui taisyklės:

Silpninimas

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, F} \qquad \frac{\Gamma \vdash \Delta}{F, \Gamma \vdash \Delta}$$

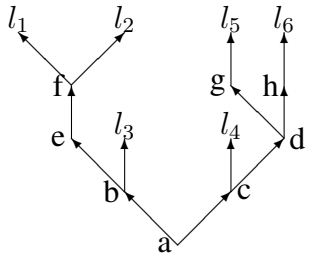
Jei išvedimo metu gauname formules, kurios nebebus naudojamos tolimesnėje išvedimo eigoje, tai galima jas pašalinti.

Pakeitę kurioje nors taisyklėje Γ, Δ, F, G konkrečiomis formulėmis, gauname **taisyklės taikymą**.

Taisyklėse yra sekvencijos. Virš brūkšnio (jų gali būti ne daugiau kaip dvi) vadinamos **prielaidomis**, o esančios žemiau brūkšnio **išvada** (ji visada viena). Pavyzdžiui, taisyklėje $(\wedge\vdash)$ prielaida yra sekvencija $F, G, \Gamma \vdash \Delta$, o išvada yra $F \wedge G, \Gamma \vdash \Delta$.

Apibrėžimas. *Sekvencijos išvedimu sekvenciniame skaičiavime G vadiname baigtinį medį, kurio visose **galinėse viršūnėse yra aksiomos**. Visose likusiose viršūnėse - sekvencijos, gautos pagal kurią nors sekvencinio skaičiavimo taisyklę iš tiesiogiai virš jų medyje esančių sekvencijų. Šaknyje esanti sekvencija sutampa su pradine*

Išvedimas vaizduojamas orientuotu grafu. Kryptis – iš apačios į viršų.



a,b,c,d,,e,f,g,h, $l_1, \ldots l_6$ - sekvencijos. *a* - pradinė sekvencija (medžio šaknis), $l_1 \ldots l_6$ - galinės viršūnės (lapai).

Spręsime anksčiau aprašytus penkis uždavinius.

1. *Elena pavargusi (p) arba serga (s). Jei pavargusi, tai ji irzli (i). Ji nėra irzli.* Vadinasi, *ji serga*

$$p \vee s, p \to i, \neg i \vdash s$$

Sekvencijos išvedimo paieška (neužmirškite, *iš apačios į viršų*):
Sprendimas:
Virš aksiomos rašysime $\oplus$, virš nesėkmės klaustuką. Formules, kurioms taikomos taisyklės, yra paryškintos.

$$\dfrac{\dfrac{\dfrac{\dfrac{\oplus}{p \vdash i, s, p} \qquad \dfrac{\oplus}{p, i \vdash i, s}}{p, \mathbf{p} \to \mathbf{i} \vdash i, s} \qquad \dfrac{\oplus}{s, p \to i \vdash i, s}}{\mathbf{p} \vee \mathbf{s}, p \to i \vdash i, s}}{p \vee s, p \to i, \neg \mathbf{i} \vdash s}$$

Atsakymas: *išvada yra teisinga.*

Pradinei formulei turėjome galimybę taikyti kurią nors iš taisyklių $(\vee \vdash), (\to \vdash), (\neg \vdash)$. Išvados teisingumas nepriklauso nuo pasirinkimo. Galėjome taikyti bet kurią. Taikėme $\neg \vdash$.

Primename, kad sekvencijoje formulių tvarka kaip antecedente, taip ir sukcedente nėra fiksuota. Todėl, pavyzdžiui, jei reikia įrašyti formulę antecedente, tai jus galite ją įrašyti tarp esančių antecedente formulių bet kurioje pozicijoje (pirmoje vietoje, paskutinėje, ar bet kurioje kitoje).

Sekvencija yra tam tikros abėcėlės žodis. Sakoma, kad sekvencijoje yra *n* loginės operacijos *įeičių*, jei ji aptinkama (einant iš kairės į dešinę) *n* kartų. Pavyzdžiui, sekvencijoje

$$p \vee (q \to (r \wedge p)) \vdash q \to r, p \wedge s, s \to (p \wedge r)$$

yra trys $\to$ įeitys, viena įeitis $\vee$ ir trys $\wedge$. Iš viso yra 7 loginių operacijų įeitys.
Pritaikius taisyklę gauname sekvenciją (*arba dvi*) *kurioje viena loginių operacijų įeitimi mažiau. Vadinasi, jei sekvencijoje n loginių operacijų įeičių, tai po ne daugiau kaip n žingsnių pasibaigs išvedimo paieška. Visada gausime atsakymą. Jei sekvencija išvedama, tai atsakymas teigiamas, jei ne, tai neigiamas*.

2. *Tyrimu įrodyta:*

- *kaltas bent vienas iš A, B, C,*
- *jei A kaltas, tai ir C kaltas,*
- *jei C nekaltas, tai ir B nekaltas.*

Raide *a* žymime teiginį *kaltas A*, raide *b* - *kaltas B*, raide *c* - *kaltas C*.

$$a \vee (b \vee c), a \to c, \neg c \to \neg b \vdash c$$

Sprendimas:

$$\dfrac{\dfrac{\dfrac{}{a, \neg c \to \neg b \vdash c, a}\,\oplus}{\mathbf{a} \vee (\mathbf{b} \vee \mathbf{c}), \neg c \to \neg b \vdash c, a}\qquad \dfrac{\dfrac{\dfrac{C}{b, \neg \mathbf{c} \to \neg \mathbf{b} \vdash c, a}\qquad \dfrac{}{c, \neg c \to \neg b \vdash c, a}\,\oplus}{\mathbf{b} \vee \mathbf{c}, \neg c \to \neg b \vdash c, a}}{a \vee (b \vee c), \mathbf{a} \to \mathbf{c}, \neg c \to \neg b \vdash c}\qquad \dfrac{\dfrac{}{a \vee (b \vee c), c, \neg c \to \neg b \vdash c}\,\oplus}{a \vee (b \vee c), \mathbf{a} \to \mathbf{c}, \neg c \to \neg b \vdash c}$$

C sudaro dvi sekvencijos: $b, \neg b \vdash c, a$ ir $b \vdash c, a, \neg c$. Jų išvedimai:

$$\dfrac{\dfrac{}{b \vdash c, a, b}\,\oplus}{b, \neg \mathbf{b} \vdash c, a}\qquad\qquad \dfrac{\dfrac{}{b, c \vdash c, a}\,\oplus}{b \vdash c, a, \neg \mathbf{c}}$$

Atsakymas: *išvada teisinga.*

3. *Jei skaičius dalijasi iš 2 ir 3, tai jis dalijasi ir iš 6. Duotas skaičius nesidalija iš 2, bet dalijasi iš 3.* Vadinasi, *skaičius nesidalija iš 6.*

$$(p_2 \wedge p_3) \to p_6, \neg p_2 \wedge p_3 \vdash \neg p_6$$

Sprendimas:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{p_3, p_6 \vdash p_2, p_2 \wedge p_3 \qquad \dfrac{?}{p_6, p_3, p_6 \vdash p_2}}{(\mathbf{p_2} \wedge \mathbf{p_3}) \to \mathbf{p_6}, p_3, p_6 \vdash p_2}}{(p_2 \wedge p_3) \to p_6, \neg \mathbf{p_2}, p_3, p_6 \vdash}}{(p_2 \wedge p_3) \to p_6, \neg p_2, p_3 \vdash \neg \mathbf{p_6}}}{(p_2 \wedge p_3) \to p_6, \neg \mathbf{p_2} \wedge \mathbf{p_3} \vdash \neg p_6}$$

Matome, kad *ne visose galinėse viršūnėse bus aksiomos*, todėl nutraukiame išvedimo paiešką.
Atsakymas: *išvada klaidinga. Iš duotų prielaidų neišplaukia, kad skaičius nesidalija iš 6.*

Duomenų kompiuteriui nepakanka, kad darytų išvadą *skaičius nesidalija iš 6.* Per mažai duomenų apie dalybą. Mes žinome daugiau, o kopiuteris ne.

O kas gautusi, jei prielaidą *jei skaičius dalijasi iš 2 ir 3, tai jis dalijasi ir iš 6* pakeistume kita *skaičius dalijasi iš 2 ir 3 tada ir tiktai tada, kai jis dalijasi iš 6.*

Nagrinėsime sekvenciją

$$(p_2 \wedge p_3) \leftrightarrow p_6, \neg p_2 \wedge p_3 \vdash \neg p_6$$

Sprendimas:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p_2, p_3, p_6, p_3, p_6 \vdash p_2}\,\oplus}{\mathbf{p_2} \wedge \mathbf{p_3}, p_6, p_3, p_6 \vdash p_2} \qquad \dfrac{\dfrac{}{p_3, p_6 \vdash p_2, p_2 \wedge p_3, p_6}\,\oplus}{(\mathbf{p_2} \wedge \mathbf{p_3}) \leftrightarrow \mathbf{p_6}, p_3, p_6 \vdash p_2}}{(p_2 \wedge p_3) \leftrightarrow p_6, \neg \mathbf{p_2}, p_3, p_6 \vdash}}{(p_2 \wedge p_3) \leftrightarrow p_6, \neg p_2, p_3 \vdash \neg \mathbf{p_6}}}{(p_2 \wedge p_3) \leftrightarrow p_6, \neg \mathbf{p_2} \wedge \mathbf{p_3} \vdash \neg p_6}$$

Atsakymas: *išvada teisinga.*

4. *Jei ketrukampio priešingosios kraštinės lygios (p) ir lygiagrečios (q), tai keturkampis yra lygiagretainis (l). Duoto keturkampio priešingosios kraštinės lygios ir lygiagrečios.* Vadinasi, *keturkampis nėra lygiagretainis.*

$$(p \wedge q) \to l, p \vee q \vdash \neg l$$

Sprendimas:

$$\dfrac{\dfrac{p, l \vdash p \wedge q \quad \dfrac{?}{l, p, l \vdash}}{(\mathbf{p} \wedge \mathbf{q}) \to \mathbf{l}, p, l \vdash} \quad \dfrac{q, l \vdash p \wedge q \quad \dfrac{?}{l, q, l \vdash}}{(\mathbf{p} \wedge \mathbf{q}) \to \mathbf{l}, q, l \vdash}}{(p \wedge q) \to l, \mathbf{p} \vee \mathbf{q}, l \vdash}}{(p \wedge q) \to l, p \vee q \vdash \neg \mathbf{l}}$$

Atsakymas: *išvada klaidinga.*

Kas blogai šiuo atveju? *Duoto keturkampio priešingosios kraštinės lygios **arba** lygiagrečios.*

Prisiminkime disjunkcijos teisingumo lentelę:

p	q	$p \vee q$
t	t	t
t	k	t
k	t	t
k	k	k

Disjunkcija teisinga trimis atvejais. $p \vee q$ juk teisinga ir kai *arba* $p = t$, *arba* $q = t$. Vadinasi, pagal prielaidą duoto keturkampio kraštinės *gali būti* lygios ir lygiagrečios, tuomet jis būtų lygiagretainis.

Pastaba. Galima nerašyti pasikartojančių formulių antecedente (sukcedente).

5. *Nagrinėjamasis erdvinis kūnas yra ritinys (r), kūgis (g) arba rutulys(l). Jei kūnas yra ritinys arba kūgis, tai jis gautas sukant tam tikro pavidalo daugiakampį apie vieną jo kraštinių (d). Kūnas nėra nei kūgis, nei rutulys.* Vadinasi, *jis yra gautas sukant daugiakampį apie vieną jo kraštinę ir yra ritinys*.

$$r \vee (g \vee l), (r \vee g) \to d, \neg g \wedge \neg l \vdash d \wedge r$$

Sprendimas:

Raide *S* žymime sekvenciją $r, (r \vee g) \to d \vdash d \wedge r, g, l$.

$$\dfrac{\dfrac{\dfrac{\dfrac{S \quad \dfrac{\dfrac{\oplus}{g, (r \vee g) \to d \vdash d \wedge r, g, l} \quad \dfrac{\oplus}{l, (r \vee g) \to d \vdash d \wedge r, g, l}}{\mathbf{g} \vee \mathbf{l}, (r \vee g) \to d \vdash d \wedge r, g, l}}{\mathbf{r} \vee (\mathbf{g} \vee \mathbf{l}), (r \vee g) \to d, \vdash d \wedge r, g, l}}{r \vee (g \vee l), (r \vee g) \to d, \neg \mathbf{l} \vdash d \wedge r, g}]}{r \vee (g \vee l), (r \vee g) \to d, \neg \mathbf{g}, \neg l \vdash d \wedge r}}{r \vee (g \vee l), (r \vee g) \to d, \neg \mathbf{g} \wedge \neg \mathbf{l} \vdash d \wedge r}$$

Sekvencijos *S* išvedimas:

$$\dfrac{\dfrac{\oplus}{r, (r \vee g) \to d \vdash r, g, l} \quad \dfrac{\dfrac{\dfrac{\oplus}{r \vdash d, g, l, r, g}}{r \vdash d, g, l, \mathbf{r} \vee \mathbf{g}} \quad \dfrac{\oplus}{r, d \vdash d, g, l}}{r, (\mathbf{r} \vee \mathbf{g}) \to \mathbf{d} \vdash d, g, l}}{r, (r \vee g) \to d \vdash \mathbf{d} \wedge \mathbf{r}, g, l}$$

Atsakymas: *išvada teisinga.*

Sekvencinio skaičiavimo **taisyklės** yra **sintaksinės**. Žiūrima tik į formulės pavidalą ir visiškai nesvarbus turinys, ką ta formule nusakome. Pavyzdžiui, jei turime sekvenciją pavidalo $\vdash p \wedge q$, tai bus ieškomi sekvencijų $\vdash p, \vdash q$ išvedimai. Kas aprašoma teiginiais *p, q* mus nedomina. Tai galima palyginti su užduotimi: *vieno litro butelius sudėkite į viršutinę lentyną, 0,75 litro talpos butelius į vidurinę lentyną, o 0,5 litro talpos butelius į apatinę lentyną.* Atliksime darbą atsižvelgdami tik į butelių talpą, o butelių turinys tos pačios talpos buteliuose gali būti įvairus.

Pastabos dėl formalizavimo:
gal . . . , gal . . . . Motina dukteriai: *pas tave buvo atėjus draugė, gal Ieva (i), gal Vaiva (v).* Teiginys užrašomas *griežta disjunkcija*: $i \dot{\vee} v$.

Gal nepatinka, kad išvedimo paieška vykdoma *iš apačios į viršų*? Sunku nustatyti kiek palikti lape vietos, kad tilptų išvedimas? Rašykite, jei patinka *iš viršaus į apačią* arba dar kuriuo nors kitu jums priimtinesniu būdu.

Kodėl knygoje nerašoma? Todėl, kad G.Gentzenas taip aprašė skaičiavimą ir visuose moksliniuose straipsniuose naudojama tik paieška iš apačios į viršų.

Išvedimą G.Gentzenas vaizdavo medžiu, kurio šaknis formulė, kurią reikia įrodyti, o galinėse viršūnėse (lapuose) aksiomos. O juk medžio viršūnės aukščiau negu šaknis. Bet studentams taip rašyti nebūtina. Rašykite taip, kaip jums patogiau.

Kai kada teiginių logikos formulės tapatų teisingumą daug lengviau patikrinti naudojantis *teisingumo lentelėmis*, negu ieškoti jos išvedimo skaičiavime. Bet sudėtingesnių logikų (predikatų, deskriptyviosios, modalumo, laiko ir kt.) formulių tapačiam teisingumui nustatyti teisingumo lentelių metodas **iš viso nenaudojamas**, nes interpretacijų skaičius yra begalinis arba jį pakankamai sudėtinga aprašyti.

Šiek tiek iš istorijos apie loginių operacijų žymėjimą

Pirmasis matematinės logikos vadovėlis pasaulyje *Principia Mathematika* (autoriai B.Russellas ir A.Whiteheadas) pasirodė 1910 metais. Kitas vadovėlis išleistas tik po 18 metų. 1928 m. išleista D.Hilberto ir W.Ackermanno knyga *Grundzüge der theoretischen Logik.* Vėliau išėjo D.Hilberto ir P.Bernayso knyga *Grundlagen der Mathematik* (pirmasis tomas 1934 m., o antrasis - 1939 m.). Dauguma tose knygose loginėms operacijoms žymėti įvestų simbolių naudojami iki šiolei.

Loginės operacijos	B.Russellas A.Whiteheadas	D.Hilbertas P.Bernaysas
neigimas	$\sim p$	$\bar{p}$
konjunkcija	$p \cdot q$	$p \,\&\, q$
disjunkcija	$p \vee q$	$p \vee q$
implikacija	$p \supset q$	$p \to q$
ekvivalentumas	$p \equiv q$	$p \sim q$

Simbolius $\sim$, $\vee$, $\equiv$, $\supset$ 1908 m. sugalvojo B.Russellas, o $\cdot$ konjunkcijai žymėti $-$ D.Boole'is. Simbolių $\neg$, $\wedge$ autoriumi yra A.Heytingas (1929 m.), simbolio $\&$ $-$ M. Schönfinkelis (1924 m.), simbolio $\to$ $-$ D.Hilbertas (1917 m.), o simbolio $\leftrightarrow$ $-$ A.Tarskis (1940 m.).

1.2 Loginių operacijų aibių pilnumas

O kaip nustatomas formulių ekvivalentumas sekvencijų skaičiavimu? Formulės *F, G ekvivalenčios tada ir tiktai tada, jei įrodoma sekvencija* $\vdash F \leftrightarrow G$.

Pavyzdys. $p \vee q, \neg p \to q$ ekvivalenčios, nes skaičiavime išvedama sekvencija $\vdash (p \vee q) \leftrightarrow (\neg p \to q)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\oplus}}{p \vdash q, p}\quad\dfrac{\dfrac{}{\oplus}}{q, \neg p \vdash q}}{p, \neg \mathbf{p} \vdash q}\quad\dfrac{\dfrac{\dfrac{}{\oplus}}{p \vdash p, q}\quad\dfrac{\dfrac{}{\oplus}}{q \vdash p, q}}{\vdash p, q, \neg \mathbf{p}}}{\mathbf{p} \vee \mathbf{q}, \neg p \vdash q}\quad\dfrac{}{\neg \mathbf{p} \to \mathbf{q} \vdash p, q}}{p \vee q \vdash \neg \mathbf{p} \to \mathbf{q}}\quad\dfrac{}{\neg p \to q \vdash \mathbf{p} \vee \mathbf{q}}}{\vdash (p \vee q) \leftrightarrow (\neg p \to q)}$$

Pageidautina išmokti mintinai dažnai naudojamas ekvivalenčių formulių poras. Išvardinsime jas.

Nr.	Ekvivalenčios formulės	Pavadinimas
1	$\neg(p \wedge q) \equiv \neg p \vee \neg q$	Neigimo įkėlimas į skliaustus (De Morgano dėsnis)
2	$\neg(p \vee q) \equiv \neg p \wedge \neg q$	Neigimo įkėlimas į skliaustus (De Morgano dėsnis)
3	$p \to q \equiv \neg p \vee q$	Implikacijos eliminavimas
4	$(p \wedge q) \vee r \equiv (p \vee r) \wedge (q \vee r)$	Distributyvumo dėsnis
5	$(p \vee q) \wedge r \equiv (p \wedge r) \vee (q \wedge r)$	Distributyvumo dėsnis
6	$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$	Ekvivalentumo eliminavimas

O kai kurios ekvivalenčių formulių poros akivaizdžios ir lengvai įsiminamos.

Pavyzdžiui:

$$\neg\neg p \equiv p,$$

$$p \wedge q \equiv q \wedge p,$$

$$p \vee q \equiv q \vee p.$$

Knygoje nagrinėjamos šešios loginės operacijos: $\neg, \wedge, \vee, \dot{\vee}, \to, \leftrightarrow$. Bet pakanka ir dviejų: $\neg, \wedge$. Likusias keturias galima išreikšti operacijomis $\neg, \wedge$. Aibė $\{\neg, \wedge\}$ vadinama **pilnąja**. Loginių operacijų aibės, kuriomis galima išreikšti nagrinėjamas logines operacija, vadinamos **pilnosiomis**.

Įrodysime, kad aibė $\{\neg, \wedge\}$ yra **pilna**.

Išreikškime operacijas naudodamiesi ekvivalenčiomis formulėmis 1 - 6, savybe $\neg\neg p \equiv p$ bei savybe, kad, jei $A \equiv B$ ir $B \equiv C$, tai ir $A \equiv C$:

- $p \vee q \equiv \neg(\neg p \wedge \neg q)$.

- $p \dot{\vee} q \equiv (p \vee q) \wedge \neg(p \wedge q)$. Mes retai naudosimės šia ekvivalenčių formulių pora. Todėl jos ir neįtraukėme į sąrašą pageidautinų žinojimui mintinai.
  $(p \vee q) \wedge \neg(p \wedge q) \equiv \neg(\neg p \wedge \neg q) \wedge \neg(p \wedge q)$.

- $p \to q \equiv \neg p \vee q \equiv \neg(p \wedge \neg q)$.

- $p \leftrightarrow q \equiv (p \to q) \wedge (q \to p) \equiv \neg(p \wedge \neg q) \wedge \neg(q \wedge \neg p)$.

Pilnomis aibėmis yra, pavyzdžiui, ir $\{\neg, \vee\}, \{\neg, \to\}$. (žr. 28 - 29 psl. knygoje [1]).

Kaip matome, pateiktuose pavyzdžiuose aibės susideda iš dviejų elementų. Ar galima rasti aibę su vienu elementu? Tarp aprašytųjų loginių operacijų tokios vienos nėra. Buvo sugalvotos specialiai (atitinkančių loginių jungčių joms šnekamojoje kalboje nėra) tam tikslui operacijos. Pirmasis tokią operaciją 1880 m. sugalvojo Ch.Peirce'as. O vėliau - 1913 m. kitą operaciją aprašė H.M.Shefferis (mes ją vadiname **Shefferio funkcija**). **Peirce'o operaciją** žymėsime $\uparrow$ o Shefferio funkciją - $|$. Jos nusakomos tokiomis lentelėmis:

p	q	$p \uparrow q$
t	t	k
t	k	k
k	t	k
k	k	t

p	q	$p \mid q$
t	t	k
t	k	t
k	t	t
k	k	t

Kompiuterių architektūros kursuose įprasta Peirc'o funkciją žymėti NOR, o Sheffer'io NAND. Jos sudaro pilnąsias aibes. Ekvivalenčių formulių poros:

$$\neg p \equiv p \uparrow p, \qquad p \vee q \equiv (p \uparrow q) \uparrow (p \uparrow q),$$

$$\neg p \equiv p \mid p, \qquad p \wedge q \equiv (p \mid q) \mid (p \mid q).$$

Matematinėje logikoje dažniausiai nagrinėjamos formulės, kuriose yra tik loginės operacijos $\neg, \wedge, \vee, \to$, nors pakaktų ir mažesnio skaičiaus (tiesiog, turint daugiau loginių operacijų paprasčiau, patogiau formalizuoti užduotis). Mes kai kada prijungsime dar ir $\leftrightarrow$.

Formulės gali būti *tapačiai teisingos, tapačiai klaidingos* ar *įvykdomos*.

Apibrėžimas. *Formulė F vadinama **tapačiai teisinga**, jei ji teisinga su bet kuria interpretacija*.

Įrodyta, kad formulė *F **tapačiai teisinga*** tada ir tiktai tada, kai išvedama sekvencija $\vdash F$.

Matematinėje logikoje taip pat naudojamos sąvokos **tautologija**, **logikos dėsnis**. Tai sąvokos *tapačiai teisinga formulė* sinonimai. Dauguma matematinės logikos uždavinių susiveda į kurios nors formulės tapataus teisingumo nustatymą, t.y. jai atitinkančios sekvencijos išvedimą.

Formulės *F* tapačiam teisingumui nustatyti jau žinome du būdus: a) naudojant teisingumo lentelę (paskutiniame stulpelyje turėtų būti tik reikšmės *t*), b) nustatyti ar $\vdash F$ išvedama sekvenciniame skaičiavime. *Jei išvedama sekvenciniame skaičiavime, tai F tapačiai teisinga, jei ne, tai F nėra tapačiai teisinga.*

Įrodysime, kad $F = (p \to \neg q) \leftrightarrow (q \to \neg p)$ yra tapačiai teisinga sudarydami jos teisingumo lentelę:

p	q	$\neg p$	$\neg q$	$p \to \neg q$	$q \to \neg p$	F
t	t	k	k	k	k	t
t	k	k	t	t	t	t
k	t	t	k	t	t	t
k	k	t	t	t	t	t

Apibėžimas. *Formulė vadinama tapačiai klaidinga, jei ji klaidinga su bet kuria interpretacija.*

$p \wedge \neg p$ yra tapačiai klaidingos formulės pavyzdys. Kai kuriose matematinės logikos taikymuose naudojama loginė konstanta *k* (*melas*). *k* yra prieštaringo tvirtinimo (tapačiai klaidingos formulės) simbolis. Mokslinėje literatūroje ši konstanta žymima simboliu $\perp$ arba žodžiu FALSE. Naudojant šią loginę konstantą formulėse galima eliminuoti neigimo operaciją, nes $\neg p \equiv p \to \perp$. Dažnai literatūroje sekvencijos su tuščiu sukcedentu $\Gamma \vdash$ rašomos taip $\Gamma \vdash \perp$.

Tapačiai klaidingų formulių savybės:

- *F* tapačiai klaidinga tada ir tiktai tada, kai $\neg F$ tapačiai teisinga formulė,

- Tarkime, turime formules $G, F \to G$. Jei *F* tapačiai klaidinga formulė, o *G* - kuri nors formulė, tai $F \to G$ yra tapačiai teisinga formulė,

- *F* tapačiai klaidinga tada ir tiktai tada, kai **išvedama sekvencija** $\vdash \neg F$,

- *F* tapačiai klaidinga tada ir tiktai tada, kai **išvedama sekvencija** $F \vdash$.

Apibrėžimas. *Formulė vadinama **įvykdoma**, jei atsiras tokia interpretacija, su kuria ji teisinga.*

Tapačiai teisinga formulė yra teisinga su bet kuria interpretacija, todėl ji įvykdoma. Turbūt savaime aišku, kad ne visos įvykdomos formulės yra tapačiai teisingos (pavyzdžiui, $p \to q$).

Apibrėžimas. *Sakome, kad formulė F yra formulių aibės $A = \{F_1, \ldots, F_n\}$ **loginė išvada**, jei su bet kuria interpretacija, su kuria visos aibės A formulės teisingos, teisinga yra ir F.*

Vadinasi, F yra formulių aibės $\{F_1, \ldots, F_n\}$ loginė išvada tada ir tiktai tada, kai

$$(F_1 \wedge \ldots \wedge F_n) \to F$$

yra tapačiai teisinga formulė.

Savo ruožtu $(F_1, \ldots, F_n) \to F$ yra tapačiai teisinga tada ir tiktai tada, kai išvedama sekvencija

$$F_1, \ldots, F_n \vdash F.$$

1.3 Semantinių lentelių metodas

Turbūt pastebėjote, kad vykdant sekvencijos išvedimo paiešką tenka perrašinėti (kartoti) daug formulių. Pavyzdžiui, taikant taisyklę $(\vdash \vee)$

$$\frac{\Gamma \vdash \Delta, F, G}{\Gamma \vdash \Delta, F \vee G}$$

teks perrašyti ir visas formules esančias sąrašuose Γ, Δ. Kaip to išvengti? Aišku, galima pasinaudoti *silpninimo* taisykle ir išbraukti formules, kurių nereikės ateityje. Visų pirma, yra sunku nustatyti, kurių nereikės. Ir vis tiek neapsieisime be dalies formulių perrašinėjimo. 1972 metais amerikiečių logikas M.Fittingas aprašė sekvencinio skaičiavimo variantą be formulių kartojimo ir pavadino jį **semantinių lentelių metodu**. Aprašysime šiuolaikinį, 1994 metais pasirodžiusį spaudoje, F.Massacci patobulintą *lentelių metodą*.

Vietoje sekvencijos $F_1, \ldots, F_n \vdash G_1, \ldots, G_m$ nagrinėjama $F_1, \ldots, F_n, \neg G_1, \ldots, \neg G_m \vdash$. Taisykles juk galima perrašyti taip, kad formulių, išvedimo paieškos metu, nebūtų sukcedente. T.y. išvedime aptinkamos tik sekvencijos pavidalo $\Gamma \vdash$. Bet tuomet simbolis $\vdash$ iš viso nereikalingas.

Kaip perrašomos sekvencinio skaičiavimo taisyklės, kai sukcedentas tuščias? Paaiškinsime pavyzdžiu. Tarkime, sekvencijai $\Gamma, \neg(F \to G) \vdash$ norime taikyti taisyklę, kuria panaikintumėme implikacijos įeitį, išlaikant sukcedentą tuščią.

Galima *įsivaizduoti* tokią veiksmų seką:
1. Perkeliame $F \to G$ į sukcedentą: $\Gamma \vdash F \to G$.
2. Taikome sekvencinio skaičiavimo taisyklę $(\vdash\to) : \Gamma, F \vdash G$.
3. Perkeliame G į antecedentą: $\Gamma, F, \neg G \vdash$.

Dabar galima parašyti ir taisyklę su tuščiu sukcedentu:

$$\frac{\Gamma, F, \neg G \vdash}{\Gamma, \neg(F \to G) \vdash}$$

Išvedimas lentelių metodu vykdomas *iš viršaus į apačią*. Kadangi sukcedentas tuščias, tai nerašomas išvedimo simbolis $\vdash$.

Semantinio lentelių metodo taisyklės

Konjunkcijos taisyklės:

$$\frac{F \wedge G}{\begin{array}{c} F \\ G \end{array}} \qquad \frac{\neg(F \vee G)}{\begin{array}{c} \neg F \\ \neg G \end{array}} \qquad \frac{\neg(F \to G)}{\begin{array}{c} F \\ \neg G \end{array}}$$

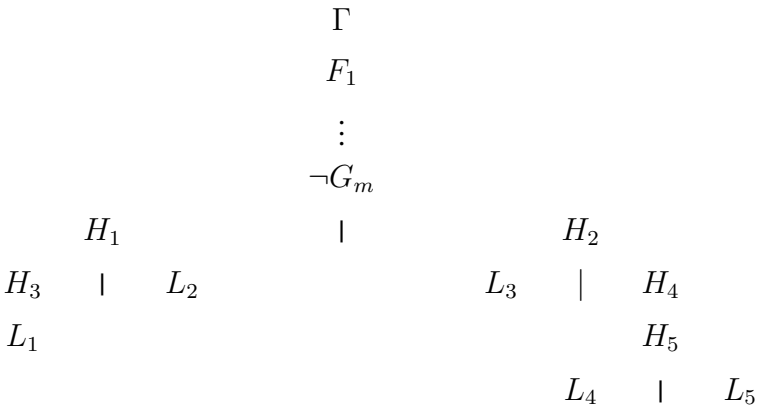
Disjunkcijos taisyklės:

$$\frac{F \vee G}{F \mid G} \qquad \frac{\neg(F \wedge G)}{\neg F \mid \neg G} \qquad \frac{F \to G}{\neg F \mid G}$$

Dvigubo neigimo taisyklė:

$$\frac{\neg\neg F}{F}.$$

Išvedimo paieška pradedama užduoties parašymu formulių seka Γ. Jei užduotis parašyta sekvencija $F_1, \ldots, F_n \vdash G_1, \ldots, G_m$, tai pradinė formulių seka $\Gamma = F_1, \ldots, F_n, \neg G_1, \ldots, \neg G_m$. Parašomos visos sekos Γ formulės (bet kuria tvarka) iš viršaus į apačią. Taisyklės gali būti taikomos bet kuriai aukščiau esančiai formulei. Išvedimo paieška iliustruojam tokiu brėžiniu:



Brėžinyje, formulės pažymėtos raidėmis $L_1, \ldots, L_5$, yra išvedimo paieškos medžio galinės viršūnės. *Šaka* vadiname formulių seką prasidedančią pirmąja formule ir besibaigiančią kuria nors galine viršūne. Pavyzdžiui, seka formulių $F_1, \ldots, \neg G_m, H_1, H_3, L_1$ yra šaka. Brėžinyje yra penkios šakos. Jei šakoje yra dvi formulės, viena iš jų yra kitos neigimas (t.y. šakoje yra dvi formulės pavidalo $A, \neg A$), tai šaka vadinama *uždarąja*. Jei išvedimo paieškos medyje visos šakos *uždaros*, tai *medis* yra formulių užduoties Γ *išvedimas lentelių metodu*.

Gal bus lengviau suprasti lentelių metodo taisykles, jei priminsime kai kurias ekvivalenčias formules (jų išvedimui panaudojome ekvivalenčias formules iš sąrašo 1 - 6):

$$\neg(p \vee q) \equiv \neg p \wedge \neg q$$
$$\neg(p \to q) \equiv \neg(\neg p \vee q) \equiv p \wedge \neg q$$
$$\neg(p \wedge q) \equiv \neg p \vee \neg q$$

Išvedimų pavyzdžiai:

1. Įrodysime implikacijos tranzityvumą: $\vdash (p \to q) \to ((q \to r) \to (p \to r))$.

Sprendimas:

$$\neg((p \to q) \to ((q \to r) \to (p \to r)))$$

Neigimas dviejų formulių $p \to q, (q \to r) \to (p \to r)$ implikacijos.

$$p \to q$$
$$\neg((q \to r) \to (p \to r))$$

Dar kartą ta pati taisyklė.

$$q \to r$$
$$\neg(p \to r)$$

Ir dar kartą ta pati taisyklė.

$$p$$
$$\neg r$$

Taikome formulei $q \to r$ taisyklę. Gauname dvi šakas.

$$\neg q \qquad | \qquad r$$

Taikome formulei $p \to q$ taisyklę $\oplus$

Gauname dvi šakas.

$$\neg p \qquad | \qquad q$$
$$\oplus \qquad\qquad \oplus$$

2. *Jei važiuosiu savaitgalį pas tėvus* (v), *tai neišlaikysiu algebros egzamino. Jei neišmokės šiomis dienomis stipendijos, tai važiuosiu pas tėvus. Algebros egzaminą išlaikiau* (a). Vadinasi, *išmokėjo stipendiją laiku* (s).

Formalizavimas: $v \to \neg a, \neg s \to v, a \vdash s$.

Patikrinsim ar išvada teisinga semantinių lentelių metodu. Jei sekvencija išvedama, tai išvada teisinga. Jei ne, tai išvada klaidinga.

Galima būsimą uždavinio rezultatą paaiškinti ir taip: jei sekvencija išvedama, tai formulė $((v \to \neg a) \wedge (\neg s \to v) \wedge a) \to s$ yra tapačiai teisinga. Jei ne, tai formulė nėra tapačiai teisinga (ji yra arba įvykdoma, arba tapačiai klaidinga). Taigi, ar išvada teisinga, galima patikrinti ir naudojant teisingumo lentelę.

$$v \to \neg a, \neg s \to v, a, \neg s$$
$$v \to \neg a$$
$$\neg s \to v$$
$$a$$
$$\neg s$$
$$\neg v \qquad | \qquad \neg a$$
$$s \qquad | \qquad v \qquad \oplus$$
$$\oplus \qquad\qquad \oplus$$

Atsakymas: *išvada teisinga*.

3. *Jei 25 nesidalija iš 3, tai ir 15 nesidalija iš 3. 15 dalijasi iš 3.* Vadinasi, *25 dalijasi iš 3*.

Žymime:
p - 15 dalijasi iš 3,
q - 25 dalijasi iš 3.

Užduotis užrašoma sekvencija $\neg q \to \neg p, p \vdash q$.

Sprendimas semantinių lentelių metodu:

$$\begin{array}{c} \neg q \to \neg p, p, \neg q \\ \neg q \to \neg p \\ p \\ \neg q \\ \neg\neg q \quad | \quad \neg p \\ \oplus \qquad\qquad \oplus \end{array}$$

Atsakymas: *išvada teisinga*.

Išvada teisinga, o tvirtinimas *25 dalijasi iš 3* klaidingas. Nieko nuostabaus. *Išvada teisinga* reiškia, kad *samprotauta teisingai*. Vadinasi, ne visos prielaidos buvo teisingos.

4. Geometrijos užduotis. *Duota tiesė kertanti plokštumą. Jei ji statmena dvejoms tiesėms a, b plokštumoje* (p) *ir tos tiesės nėra lygiagrečios* ($\neg q$), *tai ji statmena ir bet kuriai tiesei toje plokštumoje* (r). Vadinasi, *jei tiesė statmena dvejoms tiesėms plokštumoje* (p) *ir nestatmena kuriai nors tiesei plokštumoje* ($\neg r$), *tai tiesės a, b lygiagrečios* (q).

Užduotis užrašoma sekvencija $(p \wedge \neg q) \to r \vdash (p \wedge \neg r) \to q$.

Sprendimas semantinių lentelių metodu:

$$\begin{array}{c} (p \wedge \neg q) \to r, \neg((p \wedge \neg r) \to q) \\ (p \wedge \neg q) \to r \\ \neg((p \wedge \neg r) \to q) \\ p \wedge \neg r \\ \neg q \\ p \\ \neg r \\ \neg(p \wedge \neg q) \quad | \quad r \\ \neg p \quad | \quad \neg\neg q \qquad \oplus \\ \oplus \qquad \oplus \end{array}$$

5. *Jei trikampis yra daugiakampis* (d), *tai jis mažiausiai kraštinių turintis daugiakampis* (m). *Jei trikampis yra mažiausiai kraštinių turintis daugiakampis, tai atkarpa nėra daugiakampis* ($\neg a$). Vadinasi, *trikampis yra daugiakampis*.

Užduotis užrašoma sekvencija $d \to m, m \to \neg a \vdash d$.

Sprendimas:

$$
\begin{array}{ccccccc}
 & & & d \to m, m \to \neg a, \neg d & & & \\
 & & & d \to m & & & \\
 & & & m \to \neg a & & & \\
 & & & \neg d & & & \\
 & \neg m & & | & & \neg a & \\
\neg d & | & m & & \neg d & | & m \\
? & & \oplus & & ? & & ?
\end{array}
$$

Atsakymas: *išvada klaidinga.*

Aprašėme antrąjį skaičiavimą. Vėliau bus aprašytas *rezoliucijų metodas* bei *natūralioji dedukcija*. Šie keturi skaičiavimai yra pagrindiniai (dažniausiai naudojami) kaip teoriniuose darbuose, taip ir logikos taikymuose.

Kuo semantinių lentelių metodas *geresnis* už kitus? Žiūrint koks yra kriterijus *gerumui* nustatyti.

Vieta	Studentams lengviausia išmokti	Tinkamiausias suprogramavimui
1	Sekvencinis skaičiavimas	Rezoliucijų metodas
2	Semantinių lentelių metodas	Semantinių lentelių metodas
3	Rezoliucijų metodas	Natūralioji dedukcija
4	Natūralioji dedukcija	Sekvencinis skaičiavimas

1.4 Pratimai

1. Sudarykite teisingumo lentelę ir nustatykite formulės tipą: *tapačiai teisinga, tapačiai klaidinga* ar *įvykdoma*:
a) $(p \to q) \wedge (\neg(\neg p \vee r) \to (q \wedge \neg r))$,
b) $((p \wedge q) \to \neg r) \vee ((q \to r) \vee (\neg p \to q))$,
c) $((p \to q) \to (q \to p)) \wedge (\neg p \wedge q)$.

2. Išreikškite disjunkcija ir neiginiu šias formules:
a) $p \wedge (q \to p)$,
b) $(p \to q) \to (q \vee p)$,
c) $p \to (q \to (p \wedge q))$,
d) $(\neg p \to \neg q) \to (q \to p)$.

3. Išreikškite Peirc'o funkcija logines operacijas $\to, \wedge$.

4. Išreikškite Schefferio funkcija $\vee, \to$.

5. Naudodamiesi *sekvenciniu skaičiavimu* nustatykite ar formulės tapačiai teisingos:
a) $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$,
b) $(\neg p \to \neg q) \to ((\neg p \to q) \to p)$.

6. Naudodamiesi *sekvenciniu skaičiavimu*, nustatykite ar formulė tapačiai klaidinga: $((p \wedge q) \to r) \to ((p \wedge \neg r) \to r)$

7. Naudodamiesi *sekvenciniu skaičiavimu*, nustatykite ar samprotavimuose daroma teisinga išvada:
a) Jei važiuosiu troleibusu ir troleibusas pavėluos, tai aš pavėluosiu į paskaitas. Jei pavėluosiu į paskaitas, tai neišspręsiu užduoties. Jei neišspręsiu užduoties, tai neišlaikysiu testo. Vadinasi, jei nevažiuosiu troleibusu, tai nepavėluosiu į paskaitą ir išlaikysiu testą.
b) *A* ir *B* nutarė pirkti sklypus. Jei sklypą pirks *A*, tai reiškia, kad jis pigus. Jei jis brangus, tai perka *B*. *B* nepirko sklypo. Vadinasi, pirks *A*.

8. Naudodamiesi *semantinių lentelių metodu*, nustatykite ar formulė tapačiai teisinga:
$(p \vee (q \wedge r)) \to ((p \vee q) \wedge (p \vee r))$.

9. Naudodamiesi *semantinių lentelių metodu*, nustatykite ar formulė
$\neg((\neg p \wedge \neg q) \vee (p \vee q))$
tapačiai klaidinga.

10. Naudodamiesi *semantinių lentelių metodu*, nustatykite ar samprotavimuose daroma teisinga išvada:
a) Į vakarėlį žadėjo ateiti Arvydas arba Aurimas (bent vienas iš jų). Greičiausiai bus Arvydas, o Elena ne. O, jei bus Elena, tai būtinai ir Tomas. Vadinasi, vakarėlyje bus Arvydas ir Tomas.
b) Netiesa, kad Tomas buvo Ispanijoje ar Portugalijoje. Jei jis būtų išvažiavęs, kaip kad jam buvo siūloma, į Prancūziją, tai darbo klausimais būtų nuvažiavęs ir į Ispaniją. Vadinasi, Tomas nevažiavo į Prancūziją.
c) Jei Aurimas laiku būtų pranešęs apie gautą telegramą, tai Elena su Tomu būtų laiku išskridę. Žinoma, kad bent vienas iš jų pavėlavo į lėktuvą. Vadinasi Aurimas laiku nepranešė.

2 Skyrius
PIRMOS EILĖS LOGIKA

2.1 Formalizavimas

Tarkime *D* kuri nors netušia aibė.

n-viečiu ($n = 1, 2, 3, \ldots$) ***predikatu*** *aibėje D vadiname vienareikšmę n argumentų funkciją, kurios apibrėžimo sritis yra aibė D, o reikšmių aibė - t, k.*

Predikatus žymime didžiosiomis lotyniškomis raidėmis (kartais su indeksais). Juos vadinsime **atominėmis formulėmis**. Skliaustuose dažniausiai nurodome ir vietų (argumentų) skaičių:

$$P(a,\ b,\ c),\ Q(a_1, a_2, \ldots, a_n),\ R(a),\ \ldots$$

Aibės *D* elementus vadinsime **individinėmis konstantomis**. Prisiminkime, kad logikoje yra dar ir **loginės konstantos** *t, k*.

Pavyzdžiai:

1. *D* yra visų tiesių plokštumoje aibė. Predikatų *pavyzdžiai*:
P(*a*) yra teisingas tada ir tiktai tada, kai *tiesė a eina per koordinačių pradžią*.
Q(*a, b*) teisingas tada ir tiktai tada, kai *tiesės a, b lygiagrečios*.
R(*a, b*) teisingas tada ir tiktai tada, kai *tiesės a, b statmenos*

Čia *P*(*a*) vienvietis predikatas, *Q*(*a, b*), *R*(*a, b*) dviviečiai predikatai. Kintamuosius *a, b* vadinsime **laisvaisiais individiniais kintamaisiais**. Pakeitę juos *kuriais nors* aibės *D* elementais gauname teiginį, kuris gali būti teisingas arba klaidingas (priklauso nuo pasirinktų elementų). Priminsime, kad logikoje yra dar ir **loginiai kintamieji**.

Laisvuosius individinius kintamuosius žymėsime raidėmis a, b, c, d, e, $a_1, a_2, a_3, \ldots$

2. Individinių konstantų aibė yra natūraliųjų skaičių aibė $N = \{0, 1, 2, \ldots\}$. Predikatų *pavyzdžiai*:

P(*a*) teisingas tada ir tiktai tada, kai *a yra lyginis*.
Q(*a*) teisingas tada ir tiktai tada, kai *a pirminis skaičius*.
R(*a, b*) teisingas tada ir tiktai tada, kai *a dalijasi iš b*.
V(*a, b*) teisingas tada ir tiktai tada, kai $a \leq b$.

S(*a, b, c*) teisingas tada ir tiktai tada, kai $a + b = c$.

3. *D* - grafo viršūnių aibė.

L(*a, b*) teisingas tada ir tiktai tada, kai *yra lankas iš viršūnės a į b*.

4. *D* - informatikos specialybės studentų grupė.

P(*a*) teisingas tada ir tiktai tada, kai *a yra mergina*.
Q(*a*) teisingas tada ir tiktai tada, kai *studentas a išlaikė algoritmų teorijos egzaminą*.
R(*a, b*) teisingas tada ir tiktai tada, kai *a ir b per loginio programavimo pratybas sėdi viename suole*.

Iš pavyzdžių turbūt supratote, kad sąvokos *predikatas* (kai vietų skaičius ≥ 2) ir *sąryšis* matematikoje naudojami kaip sinonimai.

Kodėl sakome *vienvietis, dvivietis* predikatas, o ne *vieno, dviejų argumentų* predikatas. Tai yra todėl, kad predikatais pradėta vadinti teiginius, kurių *kai kuriose vietose* buvo kintamieji. Pavyzdžiui, *x ir y yra kaimynai*. Šiame teiginyje yra *du kintamieji*. Taigi, jis *dvivietis* predikatas. Tik pakeitę kintamuosius žmonių pavardėmis gauname *teiginį*, kurio atžvilgiu galime kelti klausimą, teisingas teiginys ar klaidingas. Taip ir išliko iki šiolei sąvokos *vienvietis, dvivietis, n-vietis predikatas*.

Matematikoje naudojami teiginiai aprašomi ir predikatų logikos formulėmis. Jų aprašymui naudosime predikatus, logines operacijas ir du **kvantorius** (lot. *quantum* - kiek). Tai **bendrumo kvantorius** (žymime $\forall$) bei **egzistavimo kvantorius** (žymime $\exists$). Iš kur kilo tokie žymėjimai? 19 amžiaus pabaigoje bei 20 amžiaus pirmoje pusėje matematinėje logikoje dominavo vokiškai kalbantys mokslininkai. Ženklas $\exists$ yra vokiečių kalbos *Existieren* apversta pirmoji raidė, o $\forall$ - žodžio *Alle* apversta pirmoji raidė. Kadangi panašiai rašomi žodžiai *egzistuoja, kiekvienas* (*visi*) ir anglų kalba (Exist, All), tai tokie žymėjimai tiko visiems. Kvantorius naudojame tik kartu su individiniais kintamaisiais (po kvantoriaus rašomas individinis kintamasis) ir vadiname **kvatoriniais kompleksais**. Užrašą $\forall x$ skaitome "kiekvienam *x*", "kiekvienam *x* teisinga", "kad ir koks būtų *x*", o $\exists x$ - "egzistuoja *x*", "egzistuoja *x*, su kuriuo teisinga", "yra toks *x*".

Prisiminkime antrą pavyzdį. Jame individinių konstantų aibė *D* yra natūraliųjų skaičių aibė $N = \{0, 1, 2, \ldots\}$. Naudodamiesi šio pavyzdžio predikatus, parašysime formules su kvantoriais.

P(*a*) teisingas tada ir tiktai tada, kai *a yra lyginis*. Formule $\forall x P(x)$ tvirtinama, kad *visi natūralieji skaičiai yra lyginiai*. Ji klaidinga.

Kaip pastebėjote, prirašėme ne tik kvantorinį kompleksą $\forall x$, bet ir predikate laisvąjį kintamąjį *a* pakeitėme kintamuoju *x*.

Jei kintamasis aptinkamas formulėje kvantoriniame komplekse, tai jis vadinamas **suvaržytuoju kintamuoju**. Mūsų pavyzdyje *x* yra suvaržytas kintamasis.

Suvaržytuosius kintamuosius žymėsime $x, y, z, u, v, w, x_1, x_2, x_3, \ldots$

Individinius kintamuosius skirstome į **laisvuosius** ir **suvaržytuosius**. Formulėse juos žymime skirtingomis raidėmis.

Formule $\exists x P(x)$ tvirtinama, kad *egzistuoja natūralieji lyginiai skaičiai* arba *tarp natūraliųjų skaičių yra lyginiai*. Ji teisinga.

Nesunku matyti, kad, jei kurį nors suvaržytąjį kintamąjį pakeisime kuriuo nors kitu (nauju) neįeinančiu į formulę, tai formulės reikšmė nepasikeis. Pavyzdžiui, formulių $\forall x P(x), \forall y P(y), \forall z P(z)$ prasmės bei jų reikšmės (jos visos klaidingos) nekinta.

V(*a, b*) teisingas tada ir tiktai tada, kai $a \leq b$.

Formule $\forall x \forall y V(x, y)$, t.y. formule $\forall x \forall y (x \leq y)$ tvirtinama, kad *su bet kuriais natūraliaisiais skaičiais x, y teisinga nelygybė* $x \leq y$. Tai netiesa. Pavyzdžiui, tvirtinimas $5 \leq 3$ yra klaidingas.

Formule $\exists x \exists y (x \leq y)$ tvirtinama, kad *egzistuoja natūralieji skaičiai x, y, su kuriais nelygybė teisinga*. Tai tiesa.

Jei pervardinsime individinius kintamuosius kitais, kurių nėra formulėje, gauname tą patį tvirtinimą. Pavyzdžiui, formulę $\exists x \exists y (x \leq y)$ galima pakeisti $\exists u \exists v (u \leq v)$.

Jei sukeisime vietomis du egzistavimo kvantorius vietomis, tai taip pat turėsime tą patį tvirtinimą $\exists u \exists v (u \leq v) \equiv \exists v \exists u (u \leq v)$ (*egzistuoja natūralieji skaičiai u, v, su kuriais nelygybė* $u \leq v$ *teisinga*). Sukeisti kvantorinius kompleksus galima ne tik šiuo konkrečiu atveju. Tai galioja bet kurioms formulėms.

Turime poras ekvivalenčių formulių:

$$\forall x \forall y F \equiv \forall y \forall x F,$$
$$\exists x \exists y F \equiv \exists y \exists x F.$$

Tai nesunku atsiminti. O kaip, jei kvantoriniai kompleksai prasideda skirtingais kvantoriais? Tuomet, deja, sukeisti negalima. Pavyzdžiui, formulės $\forall x \exists y (x \leq y)$ ir $\exists y \forall x (x \leq y)$ turi skirtingą prasmę. Pirmąja tvirtinama, kad *kiekvienam natūraliajam skaičiui x galima rasti ne mažesnį y*. Tai teisingas teiginys. O antrąja tvirtinama *egzistuoja toks natūralusis skaičius, kad visi natūralieji yra už jį mažesni arba lygūs*. T.y., tvirtinama, kad *natūraliųjų skaičių aibėje egzistuoja didžiausias*. Toks tvirtinimas klaidingas.

Reiks prisiminti, kad

$$\forall x \exists y F \not\equiv \exists y \forall x F.$$

Nustebsite, bet predikatų logikoje yra nedaug ekvivalenčių formulių, kurias vertėtų žinoti mintinai. Be to, jas lengva įsiminti. Jūs ir patys galėtumėt jas įrodyti. Pateiksime dar porą tokių.

Prisiminkime ketvirtą pavyzdį: *D* - informatikos specialybės studentų grupė. Predikatas *Q*(*a*) teisingas tada ir tiktai tada, kai *studentas x išlaikė algoritmų teorijos egzaminą*.

Teiginys *ne visi išlaikė algoritmų teorijos egzaminą* užrašomas formule $\neg\forall x Q(x)$. Bet, jei *ne visi studentai išlaikė egzaminą*, tai tą patį teiginį galima pasakyti ir taip: *yra grupėje bent vienas studentas neišlaikęs egzamino*. O šis teiginys užrašomas formule $\exists x\neg Q(x)$.

Teiginį *netiesa, kad grupėje yra bent vienas studentas išlaikęs egzaminą* (formulė $\neg\exists x Q(x)$) galima nusakyti ir kitais žodžiais: *visa grupė neišlaikė egzamino* (formulė $\forall x\neg Q(x)$)

Gavome dar dvi poras ekvivalenčių formulių. Kad neužmirštumėt, parašysiu jas lentelėje (atkreipkit dėmesį, kad pirmoji pora yra *neekvivalenčių* formulių, nesuklyskite).

$$\begin{array}{|c|}\hline \forall x\exists y F \not\equiv \exists y\forall x F \\ \neg\forall x Q(x) \equiv \exists x\neg Q(x) \\ \neg\exists Q(x) \equiv \forall x\neg Q(x) \\ \hline\end{array}$$

Parašysim kai kurias grafų savybes naudodami trečio pavyzdžio predikatą.

D = V - grafo viršūnių aibė. Predikatas *L(a, b)* teisingas tada ir tiktai tada, kai *grafe yra lankas iš viršūnės a į b.*

a) Teiginys *grafas refleksyvus* užrašomas formule $\forall x L(x,x)$.

b) Teiginys *grafas irefleksyvus* užrašomas formule $\forall x\neg L(x,x)$.

c) Teiginys *totalusis grafas* (grafas vadinamas totaliuoju, jei iš kiekvienos viršūnės išeina bent vienas lankas) užrašomas formule $\forall x\exists y L(x,y)$.

d) Teiginys *grafas simetrinis* (jei yra lankas iš *x* į *y*, tai grafe yra ir lankas iš viršūnės *y* į viršūnę *x*) užrašomas formule $\forall x\forall y(L(x,y)\to L(y,x))$.

e) Teiginys *grafas tranzityvus* (jei grafe yra lankas iš viršūnės *x* į *y* ir iš *y* į viršūnę *z*, tai grafe bus ir lankas iš viršūnės *x* į *z*) $\forall x\forall y\forall z((L(x,y)\wedge L(y,z))\to L(x,z))$.

f) Teiginys *grafas euklidinis* (jei bet kokiems $v, y, z \in V$, iš to, kad yra lankas iš *x* į *y* ir yra lankas iš *x* į *z*, išplaukia, kad grafe bus ir lankas iš *y* į *z)* užrašomas formule $\forall x\forall y\forall z((L(x,y)\wedge L(x,z))\to L(y,z))$.

Suformuluosime keletą grafų savybių. Ne visos yra akivaizdžios. Kitame skyrelyje, kai aprašysime predikatų logikos skaičiavimus, mes įrodysime pirmuosius du teiginius. Šiame skyrelyje supažindiname tik su teiginių aprašymu predikatų logikos formulėmis.

Aprašome sekvencijomis samprotavimus:

1. Jei grafas simetrinis ir tranzityvus, tai jis euklidinis.
$\forall x\forall y(L(x,y)\to L(y,x)), \forall x\forall y\forall z((L(x,y)\wedge L(y,z))\to L(x,z)) \vdash \forall x\forall y\forall z((L(x,y)\wedge L(x,z))\to L(y,z))$

2. Grafas simetrinis, tranzityvus ir totalusis tada ir tiktai tada, kai jis refleksyvus ir euklidinis.
$\vdash [[\forall x\forall y(L(x,y)\to L(y,x))\wedge\forall x\forall y\forall z((L(x,y)\wedge L(y,z))\to L(x,z))]\wedge\forall x\exists y L(x,y)]\leftrightarrow$

$\forall x L(x,x) \wedge \forall x \forall y \forall z((L(x,y) \wedge L(x,z)) \rightarrow L(y,z))$.

Kaip pastebėjote, naudojame ir skliaustus [,]. Taip lengviau matoma formulės struktūra. Dažniausiai apsieisime be skliaustų [,].

Teiginiai pavidalo (*S, P* yra vienviečiai predikatai)

Visi S yra P,
Kai kurie S yra P,
Nė vienas S nėra P,
Kai kurie S nėra P

formalizuojami tokiomis predikatų logikos formulėmis (plačiau apie tokio tipo teiginius skaityk 132 - 135 psl. knygoje [1]):

$\forall x(S(x) \rightarrow P(x))$,
$\exists x(S(x) \wedge P(x))$,
$\forall x(S(x) \rightarrow \neg P(x))$,
$\exists x(S(x) \wedge \neg P(x))$.

Pavyzdžiui, teiginiai: a) *visi informatikos studentai* (*S*) *išlaikė algebros* (*A*) *egzaminą*, b) *kai kurie informatikos studentai* (*S*) *neišlaikė diskrečiosios matematikos* (*DM*) *egzamino* užrašomi formulėmis:

a) $\forall x(S(x) \rightarrow A(x))$,
b) $\exists x(S(x) \wedge \neg DM(x))$.

Predikatų logika yra tinkama kalba matematikos teiginių aprašymui. Turime nurodyti individinių konstantų aibę ir joje apibrėžtus predikatus.

Individinių konstantų aibė: R (realiųjų skaičių aibė).
Predikatai:
1. *Z*(*a*) yra teisingas tada ir tiktai tada, kai *a* yra sveikasis skaičius.
2. Lygybės predikatas $a = b$.
3. Predikatai $a < b, a > b$.

Naudodami šiuos predikatus aprašysime teiginius (jie nebūtinai teisingi).

a) *Sveikųjų skaičių aibėje egzistuoja mažiausias skaičius*:

$$\exists x(Z(x) \wedge \forall y(Z(y) \rightarrow x < y)).$$

b) *Koks bebūtų sveikasis skaičius, atsiras didesnis bei mažesnis už jį sveikieji skaičiai*:

$$\forall x(Z(x) \rightarrow [\exists y(Z(y) \wedge y > x) \wedge \exists z(Z(z) \wedge z < x)]).$$

c) *Atsiras vienintelis realusis skaičius, su kuriuo formulė F*(*x*) *yra teisinga*:

$$\exists x F(x) \wedge \forall x \forall y((F(x) \wedge F(y)) \rightarrow x = y)$$

arba

$$\exists x \forall y (F(y) \to x = y).$$

d) *Egzistuoja ne daugiau kaip n individinių konstantų* (konstantomis šiame pavyzdyje yra realieji skaičiai), *su kuriomis teisinga formulė F(x)*:

$$\exists x_1 \exists x_2 \ldots \exists x_n (\forall y F(y) \to (x_1 = y \vee x_2 = y \vee \ldots \vee x_n = y)).$$

Pastaba. Dar trys šio pavyzdžio užduotys yra pratimų skyrelyje (1a, 1b, 1c).

2.2 Išvedimo paieška

Šiame skyrelyje susipažinsim su *sekvenciniu skaičiavimu* bei *semantinių lentelių metodu* predikatų logikai. Taip pat su sąvokomis *įvykdoma, tapačiai teisinga, tapačiai klaidinga formulė.*

Praeitame skyrelyje išmokom aprašyti teiginius predikatų logikos formulėmis, kai duota individinių konstantų aibė ir joje apibrėžti predikatai. Tokios formulės vadinamos **pirmos eilės logikos** formulėmis. O dabar pabandykim spręsti priešingą uždavinį.

Turim formulę $\forall x \exists y (P(x, y) \wedge \neg P(x, x))$ (2.1)

Kokia jos prasmė? Ką norima ja pasakyti? Atsakymo paieška susiveda į tokios aibės ir joje apibrėžto dviviečio predikato paiešką, kad duota formulė būtų teisinga. Ieškosime **formulės sprendinio.** Jame yra du nežinomieji: *individinių konstantų aibė* ir *dvivietis predikatas.* Formulė turi ne vieną sprendinį. Pateiksime du:

1. Individinių konstantų aibė $N = \{0, 1, 2, \ldots\}, P(x, y) = x < y$.

$$\forall x \exists y (x < y \wedge \neg (x < x)) = t.$$

2. Individinių konstantų aibė *D* yra *visų tiesių plokštumoje* aibė. $P(x, y) = t$ tada ir tiktai tada, kai tiesė *x* statmena tiesei *y* (predikatas $x \perp y$).

$$\forall x \exists y (x \perp y \wedge \neg (x \perp x)) = t.$$

Sprendinys (pora: netuščia aibė ir predikatas) logikoje vadinamas formulės **modeliu**. O bet kuri pora (netuščia aibė ir joje apibrėžtas predikatas) - **struktūra.**

*Formulė vadinama **įvykdoma**, jei yra struktūra, kurioje ji teisinga. T.y., jei ji turi modelį.*

*Formulė vadinama **tapačiai teisinga** (tautologija), jei ji teisinga bet kurioje struktūroje.*

*Formulė vadinama **tapačiai klaidinga**, jei ji neturi modelio. T.y., jei ji klaidinga bet kurioje struktūroje.*

Formulėse gali būti ir laisvieji kintamieji. Tarkime, turime formulę $F(a_1, \ldots, a_n)$ su laisvaisiais kintamaisiais $a_1, \ldots, a_n$. Kintamiesiems priskiriamos struktūros *kurios nors individinės konstantos*. Formulės teisingumo nustatymas priklauso nuo užduoties:

a) formulė $F(a_1, \ldots, a_n)$ *įvykdoma*, jei *galima rasti* tokias individines struktūros konstantas, kad ji būtų teisinga, t.y. $F(a_1, \ldots, a_n)$ *ekvivalenti* formulei $\exists x_1 \ldots \exists x_n F(x_1, \ldots, x_n)$.

b) formulė $F(a_1, \ldots, a_n)$ *tapačiai teisinga,* jei teisinga su *bet kuriomis* individinėmis struktūros konstantomis, t.y.$F(a_1, \ldots, a_n)$ *ekvivalenti* formulei $\forall x_1 \ldots \forall x_n F(x_1, \ldots, x_n)$.

Pavyzdžiui, turime formulę $\forall x(P(a) \vee \neg P(x))$ ir struktūrą, kurioje individinių konstantų aibe yra natūraliųjų skaičių aibė, o predikatas *P* teisingas, jei argumento reikšmė *yra lyginis skaičius.* Formulė *įvykdoma*, nes, jei kintamajam *a* priskirsime kurį nors lyginį skaičių (pavyzdžiui, $a = 4$), tai $\forall x(P(4) \vee \neg P(x)) = t$. Nustatant įvykdomumą, ji ekvivalenti formulei $\exists y \forall x(P(y) \vee \neg P(x))$ be laisvųjų kintamųjų. Formule su vienu laisvuoju kintamuoju *nusakome aibę.* Jos elementais yra tos individinės konstantos, su kuriomis formulė teisinga.
Formulę be laisvųjų kintamųjų vadiname ***uždara***.
Uždara formule gauname *teiginį, kuris teisingas arba klaidingas*. Formule $P(a)$ nusakome lyginių skaičių aibę (individinių konstantų poaibį). O formule $\exists x P(x)$ parašėme teiginį, kuris gali būti teisingas (taip yra šiuo atveju) arba klaidingas.

Matematiniai reiškiniai skirstomi į teisingus su visais elementais iš apibrėžimo srities, klaidingus su visais elementais, bei teisingus (įvykdomus) su kai kuriais elementais. Tvirtinimas $(x+y)^2 = x^2 + 2xy + y^2$ yra teisingas su visais realiaisiais skaičiais. $x^2 + 5x - 6 = 0$ teisingas su kai kuriais skaičiais. Tvirtinimas $x + y = 0 \wedge 2x + 2y - 3 = 0$ klaidingas su visais realiaisiais skaičiais.

Norint įrodyti, kad formulė įvykdoma, pakanka rasti bent vieną struktūrą, kurioje ji teisinga. (2.1) formulė įvykdoma. O kaip įrodyti, kad formulė tapačiai teisinga, t.y. įrodyti, kad ji teisinga su bet kuriais formulės predikatais, apibrėžtais joje. *Teisingumo lentelių* metodui, nagrinėtam teiginių logikoje, analogo nėra predikatų logikoje. Struktūrų yra be galo daug. XX amžiaus pirmoje pusėje buvo aprašyti keli formulių tapačiai teisingumo nustatymo metodai. Tai buvo vieni iš žymiausių praeito amžiaus matematikų pasiekimų. Mes supažindinsime šiame skyrelyje su dviem formulės tapačiai teisingumo nustatymo būdais: *sekvenciniu skaičiavimu predikatų logikai* ir *semantinių lentelių metodu predikatų logikai.* Vėliau bus aprašytas *rezoliucijų metodas* bei *natūralioji dedukcija.*

Primename, kad ***sekvencija*** *vadiname reiškinį* $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$; *čia* $A_i (i = 1, \ldots, n)$ *bei* $B_i (i = 1, \ldots, m)$ *yra formulės*. Sekvencijoje turi būti bent viena formulė, todėl $n + m \neq 0$.

Sekvencinis predikatų logikos skaičiavimas gaunamas papildžius teiginių logikos *sekvencinį skaičiavimą* keturiomis kvantorinėmis taisyklėmis. Jų vardai: $(\vdash \forall), (\vdash \exists), (\forall \vdash), (\exists \vdash)$.

Tarkime, norime įrodyti formulę pavidalo $\forall x F(x)$, kai duota, kad individinių konstantų aibė natūralieji skaičiai. Taigi, reikia įrodyti, kad $F(0), F(1), F(2), F(3), \ldots$ yra teisingos formulės. Betgi jų be galo daug. Kaip galima tuomet jas visas įrodyti? Kaip *pagaunama begalybė*, kaip baigtiniu samprotavimu įrodoma be galo daug formulių? Sakysite, tokiu atveju matematikai naudoja *matematinę indukciją*. Taip, daugeliui uždavinių tai tinkamas metodas. Bet ne visiems. Tinkamesnė idėja rasta kitokio tipo matematikos teoremų įrodymuose. Pavyzdžiui, Pitagoro teoremos. ***Kokios bebūtų x, y,z*** (*stataus trikampio kraštinės, z - įžambinė*) *galioja lygybė* $z^2 = x^2 + y^2$. Yra daug skirtingų šios teoremos įrodymų. Kai kuriuose yra tokie samprotavimai:

tarkime ***a, b, c kurios nors*** *stataus trikampio kraštinės* (*c - įžambinė*), *įrodysime, kad* $c^2 = a^2 + b^2$.

Pereinama nuo suvaržytųjų kintamųjų *x, y, z* $\forall x \forall y \forall z(z^2 = x^2 + y^2)$ prie laisvųjų kintamųjų $c^2 = a^2 + b^2$. **Įrodomas šablonas**. *Begalybė pagaunama šablonu.* Jei paimsime konkrečius realiuosius skaičius - kurio nors trikampio kraštinių ilgius, tai, pakeitę įrodyme kintamuosius *a, b, c* gausime Pitagoro teoremos įrodymą konkretaus stataus trikampio atveju. Šią įrodymų idėją perėmė ir logikai. Pasirodo tai universalus metodas tinkamas visiems uždaviniams aprašomiems **pirmos eilės logikos formulėmis**.

Taisyklė ($\vdash \forall$) tokia:

$$\frac{\vdash F(\beta)}{\vdash \forall x F(x)}$$

Čia β kuris nors naujas, neįeinantis į apatinę sekvenciją, *laisvasis kintamasis.*

Perėjimas nuo *kokie bebūtų x, y, z* prie *kurių nors a, b, c* nėra tik žodžių žaismas.

Todėl šioje knygoje žymime laisvuosius ir suvaržytuosius kintamuosius skirtingomis raidėmis, kad jos skirtųsi vizualiai (sunku rasti kitą tokį vadovėlį pasaulyje). Primenu:

Laisvuosius kintamuosius žymime *a, b, c, d, e,* $a_1, a_2, a_3, \ldots$

Suvaržytuosius kintamuosius žymime $x, y, z, u, v, w, x_1, x_2, x_3, \ldots$

Pastaba. Patogumo dėlei galite laisvuosius kintamuosius žymėti ir **kitomis raidėmis**, bet jos vis tiek turi skirtis nuo *suvaržytųjų kintamųjų* $x, y, z, u, v, w, x_1, x_2, x_3, \ldots$

Pavyzdys.
Įrodysime, kad formulė $\forall x \forall y((F(x) \to F(y)) \to (\neg F(y) \to \neg F(x)))$ yra tapačiai teisinga.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{F(a) \vdash F(b), F(a)}\oplus \qquad \dfrac{}{F(b), F(a) \vdash F(b)}\oplus}{\mathbf{F(a)} \to \mathbf{F(b)}, F(a) \vdash F(b)}}{F(a) \to F(b), \neg \mathbf{F(b)}, F(a) \vdash}}{F(a) \to F(b), \neg F(b) \vdash \neg \mathbf{F(a)}}}{F(a) \to F(b) \vdash \neg \mathbf{F(b)} \to \neg \mathbf{F(a)}}}{\vdash (F(a) \to F(b)) \to (\neg F(b) \to \neg F(a))}}{\vdash \forall y((F(a) \to F(y)) \to (\neg F(y) \to \neg F(a)))}}{\vdash \forall x \forall y((F(x) \to F(y)) \to (\neg F(y) \to \neg F(x)))}$$

Jei individinių konstantų aibė baigtinė, tai predikatų logikos formules galima transformuoti į teiginių logikos formules.

Pavyzdys.
Studentams *A, B, C, D* teko perlaikyti egzaminą. Pažymėkime *I*(*a*) - *studentas a išlaikė egzaminą*.

Tuomet tvirtinimas *visi studentai išlaikė egzaminą* parašomas formule $\forall x I(x)$. Ji yra ekvivalenti teiginių logikos formulei.

$$\forall x I(x) \equiv I(A) \wedge I(B) \wedge I(C) \wedge I(D)$$

Tvirtinimas *kai kurie studentai išlaikė egzaminą* parašomas formule $\exists x I(x)$. Ji ekvivalenti teiginių logikos formulei.

$$\exists x I(x) \equiv I(A) \vee I(B) \vee I(C) \vee I(D)$$

Išsiaiškinom, kad formulės pavidalo $\forall x F(x)$ įrodomos, jei rasime įrodymą šablono $F(a)$. O jeigu visos formulės (tarkime, individinių konstantų aibė yra natūralieji skaičiai) $F(0), F(1), F(2), \ldots$ neįrodomos, tai gautusi, kad visos formulės $F(0), F(1), F(2), \ldots$ *teisingos*, o formulė $\forall x F(x)$ *neįrodoma*? Taip, tai tiesa.

Laikas aprašyti taisyklę $(\vdash \exists)$.

$$\frac{\vdash F(\gamma), \exists x F(x)}{\vdash \exists x F(x)}$$

Čia γ *kuris nors laisvasis* kintamasis iš apatinės sekvencijos. O, jei apatinėje sekvencijoje nėra laisvųjų kintamųjų, tai γ yra *naujas laisvasis* kintamasis.

Taikydami taisyklę keičiame kintamąjį x. Jeigu ne tą pasirinkome, tai nieko tokio, formulė $\exists x F(x)$ kartojama ir galėsite x pakeisti kitu (gal dabar pasiseks parinkti tinkamą).

Turime sekvenciją $\vdash F$. Ji įrodoma tada ir tiktai tada, kai išvedama $\neg F \vdash$.

Sekvencija $\vdash \neg F$ įrodoma tada ir tiktai tada, kai išvedama $F \vdash$.

T.y., perkeliant formulę iš vienos (nuo išvedimo simbolio) pusės į kitą, prirašomas neigimas (arba panaikinamas, jei ji prasidėjo neigimu, nes $\neg\neg F \equiv F$). Todėl taisyklė $(\forall \vdash)$ panaši į $(\vdash \exists)$, o $\exists \vdash$ į $(\vdash \forall)$. Juk $\neg \forall x Q(x) \equiv \exists x \neg Q(x)$ ir $\neg \exists Q(x) \equiv \forall x \neg Q(x)$.

Pirmos eilės logikos predikatų skaičiavimas.

Aksiomos: $F, \Gamma \vdash \Delta, F$

Loginių operacijų taisyklės:

$$(\neg \vdash) \quad \frac{\Gamma, \vdash \Delta, F}{\neg F, \Gamma \vdash \Delta} \qquad (\vdash \neg) \quad \frac{F, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg F}$$

$$(\wedge \vdash) \quad \frac{F, G, \Gamma \vdash \Delta}{F \wedge G, \Gamma \vdash \Delta} \qquad (\vdash \wedge) \quad \frac{\Gamma \vdash \Delta, F \quad \Gamma \vdash \Delta, G}{\Gamma \vdash \Delta, F \wedge G}$$

$$(\vee \vdash) \quad \frac{F, \Gamma \vdash \Delta \quad G, \Gamma \vdash \Delta}{F \vee G, \Gamma \vdash \Delta} \qquad (\vdash \vee) \quad \frac{\Gamma \vdash \Delta, F, G}{\Gamma \vdash \Delta, F \vee G}$$

$$(\to\vdash)\ \frac{\Gamma\vdash\Delta, F \quad G, \Gamma\vdash\Delta}{F\to G, \Gamma\vdash\Delta} \qquad (\vdash\to)\ \frac{F, \Gamma\vdash\Delta, G}{\Gamma\vdash\Delta, F\to G}$$

$$(\leftrightarrow\vdash)\ \frac{\Gamma\vdash\Delta, F, G \quad F, G, \Gamma\vdash\Delta}{F\leftrightarrow G, \Gamma\vdash\Delta} \qquad (\vdash\leftrightarrow)\ \frac{F, \Gamma\vdash\Delta, G \quad G, \Gamma\vdash\Delta, F}{\Gamma\vdash\Delta, F\leftrightarrow G}$$

Kvantorinės taisyklės:

$$(\exists\vdash)\ \frac{F(\beta), \Gamma\vdash\Delta}{\exists x F(x), \Gamma\vdash\Delta} \qquad (\vdash\exists)\ \frac{\Gamma\vdash\Delta, F(\gamma), \exists x F(x)}{\Gamma\vdash\Delta, \exists x F(x)}$$

$$(\forall\vdash)\ \frac{F(\gamma), \forall x F(x), \Gamma\vdash\Delta}{\forall x F(x), \Gamma\vdash\Delta} \qquad (\vdash\forall)\ \frac{\Gamma\vdash\Delta, F(\beta)}{\Gamma\vdash\Delta, \forall x F(x)}$$

$$(\exists\ldots\exists\vdash)\ \frac{F(\beta_1, \beta_2, \ldots, \beta_n), \Gamma\vdash\Delta}{\exists x_1\exists x_2\ldots\exists x_n F(x_1, x_2, \ldots, x_n), \Gamma\vdash\Delta}$$

$$(\vdash\forall\ldots\forall)\ \frac{\Gamma\vdash F(\beta_1, \beta_2, \ldots, \beta_n)}{\Gamma\vdash\forall x_1\forall x_2, \ldots\forall x_n F(x_1, x_2, \ldots, x_n)}$$

Raide C žymime aibę laisvųjų kintamųjų, esančių apatinėje sekvencijoje.

$(\exists\vdash), (\vdash\forall), (\exists\ldots\exists\vdash), (\vdash\forall\ldots\forall)$ taisyklėse $\beta, \beta_1, \beta_2, \ldots, \beta_n \notin C$. $\beta, \beta_1, \beta_2, \ldots, \beta_n$ yra *nauji* laisvieji kintamieji.

Taisyklėse $(\vdash\exists)$ ir $(\forall\vdash)$ $\gamma\in C$. Jei C tuščia aibė, tai γ naujas laisvasis kintamasis.

Leistinos sekvenciniam skaičiavimui taisyklės.
Silpninimas:

$$\frac{\Gamma\vdash\Delta}{\Gamma\vdash\Delta, F} \qquad \frac{\Gamma\vdash\Delta}{F, \Gamma\vdash\Delta}$$

Jei išvedimo paieškos metu gauname formules, kurios nebebus naudojamos tolimesnėje išvedimo paieškos eigoje, tai galima jas pašalinti.

Pjūvio taisyklė:

$$\frac{\Gamma\vdash\Delta, F \qquad F, \Gamma\vdash\Delta}{\Gamma\vdash\Delta}$$

Visi formulės F laisvieji kintamieji turi būti ir apatinėje sekvencijoje.

Pavyzdžiai:

1. $\forall x P(a, a, x)\to\exists y\exists z P(y, b, z)\vdash\forall x P(x, x, b)$.
$C = \{a, b\}$.

2. $\forall x Q(x, x)\vee\neg\exists y P(y)\vdash\exists x\exists y Q(x, y)\wedge\forall x\neg P(x)$.
$C = \emptyset$.

Įrodyta:

- sekvencija $\vdash F$ įrodoma tada ir tiktai tada, kai ***F* tapačiai teisinga,**

- sekvencija $F \vdash$ bei $\vdash \neg F$ išvedamos tada ir tiktai tada, kai ***F* tapačiai klaidinga,**

- sekvencija $F_1, \ldots, F_n \vdash G_1, \ldots, G_m$ išvedama tada ir tiktai tada, kai $(F_1 \wedge \ldots \wedge F_n) \to (G_1 \vee \ldots \vee G_m)$ **tapačiai teisinga formulė**.

Taisyklės $(\vdash \exists)$ **baigtinumo savybė**. Taikant šią taisyklę, γ renkamės iš laisvųjų kintamųjų apatinėje sekvencijoje (jų gali būti tik baigtinis skaičius).

Jei nagrinėjamos formulės tik su predikatiniais kintamaisiais, t.y. formulėse nėra nei konkrečių predikatų, nei konkrečių individinių konstantų, tai tokia logika vadinama **grynąja**.

Taisyklės $(\vdash \exists)$ *baigtinumo savybė* galioja **grynąjai logikai** bei daugeliui šioje knygoje nagrinėjamų formalių teorijų, pavyzdžiui, grynąjai logikai su lygybės predikatu. Bet ne visoms. Negalioja aibių teorijai *ZF* bei **antros eilės logikos** formalioms teorijoms.

Pateikta pastaba galioja ir taisyklei $(\forall \vdash)$. Juk, jei reikia įrodyti $\vdash \exists x P(x)$, tai tas pats, kas įrodyti$\neg \exists x P(x) \vdash$. O tai ekvivalentu sekvencijos $\forall x \neg Q(x) \vdash$ įrodymui.

Formulių išvedimų pavyzdžiai (sekvencijų neatskirsime brūkšniais; vietomis komentuosime išvedimą).

1. Nors $\forall x \exists y F(x,y) \not\equiv \exists y \forall x F(x,y)$, bet $\exists y \forall x F(x,y) \to \forall x \exists y F(x,y)$ yra tapačiai teisinga formulė.

Primename, išvedimas *iš apačios į viršų.*

$$\oplus$$
$$F(b,a), \forall x F(x,a) \vdash \exists y F(b,y), F(b,a)$$
$$C = \{a,b\}. \text{ Taikome } (\vdash \exists).$$
$$F(b,a), \forall x F(x,a) \vdash \exists y F(b,y)$$
$$C = \{a,b\}. \text{ Taikome } (\forall \vdash) \text{ (galėjom taikyti } (\vdash \exists)).$$
$$\forall x F(x,a) \vdash \exists y F(b,y)$$
$$\forall x F(x,a) \vdash \forall x \exists y F(x,y)$$

Yra pasirinkimas. Galim taikyti arba $(\exists \vdash)$, arba $(\vdash \forall)$. Abiem atvejais suvaržytuosius kintamuosius turim keisti *naujais laisvaisiais* kintamaisiais. Kas yra nuostabu šiame skaičiavime, kad nuo taisyklių pasirinkimo tvarkos priklauso tik išvedimo sudėtingumas (išvedimo medžio aukštis). Jei formulė tapačiai teisinga, tai rasim jos išvedimą nepriklausomai nuo taisyklių pasirinkimo. *Rekomendacija greitesniam išvedimui*: jei yra galimybė, taikome kurią nors iš $(\exists \vdash), (\vdash \forall)$.

$$\exists y \forall x F(x,y) \vdash \forall x \exists y F(x,y)$$
$$\vdash \exists y \forall x F(x,y) \to \forall x \exists y F(x,y)$$

2. Jei grafas refleksyvus, tai jis totalusis. $\forall x L(x,x) \vdash \forall x \exists y L(x,y)$.

$$\oplus$$
$$\forall L(x,x), L(a,a) \vdash L(a,a), \exists y L(a,y)$$
$$C = \{a\}$$
$$\forall x L(x,x), L(a,a) \vdash \exists \mathbf{y}\mathbf{L}(\mathbf{a},\mathbf{y})$$
$$C = \{a\}$$
$$\forall \mathbf{x}\mathbf{L}(\mathbf{x},\mathbf{x}) \vdash \exists y L(a,y)$$
$$\forall x L(x,x) \vdash \forall \mathbf{x} \exists \mathbf{y}\mathbf{L}(\mathbf{x},\mathbf{y})$$

3. Jei grafas simetrinis ir tranzityvus, tai jis euklidinis.
$\forall x \forall y (L(x,y) \rightarrow L(y,x)), \forall x \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z)) \vdash \forall x \forall y \forall z ((L(x,y) \wedge L(x,z)) \rightarrow L(y,z))$.

Pažymėkime F_1 formulę $\forall x \forall y (L(x,y) \rightarrow L(y,x))$,
F_2 - $\forall y (L(a,y) \rightarrow L(y,a))$,
G_1 - $\forall x \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z))$,
G_2 - $\forall y \forall z ((L(b,y) \wedge L(y,z)) \rightarrow L(b,z))$,
G_3 - $\forall z ((L(b,a) \wedge L(a,z)) \rightarrow L(b,z))$.

Išvedimo medis (iš apačios į viršų):

[1] $L(b,a), L(a,b), L(a,c) \vdash L(b,c), L(b,a)$.
[2] $L(b,a), L(a,b), L(a,c) \vdash L(b,c), L(a,c)$.
Taikome $(\vdash \wedge)$ taisyklę. Gauname dvi ([1], [2]) aksiomas.
[2] $L(b,a), L(a,b), L(a,c) \vdash L(b,c), \mathbf{L}(\mathbf{b},\mathbf{a}) \wedge \mathbf{L}(\mathbf{a},\mathbf{c})$
[1] $L(b,a), L(b,c), L(a,b), L(a,c) \vdash L(b,c)$.
Taikome $(\rightarrow \vdash)$ taisyklę. Gauname dvi [1], [2] sekvencijas. [1] sekvencija yra aksioma. Tęsiame antros ([2]) sekvencijos išvedimą.
$L(b,a), (\mathbf{L}(\mathbf{b},\mathbf{a}) \wedge \mathbf{L}(\mathbf{a},\mathbf{c}) \rightarrow \mathbf{L}(\mathbf{b},\mathbf{c})), L(a,b), L(a,c) \vdash L(b,c)$.
$L(b,a), \forall \mathbf{z}((\mathbf{L}(\mathbf{b},\mathbf{a}) \wedge \mathbf{L}(\mathbf{a},\mathbf{z})) \rightarrow \mathbf{L}(\mathbf{b},\mathbf{z})), \mathbf{G_2}, L(a,b), L(a,c) \vdash L(b,c)$.
$L(b,a), \forall \mathbf{y} \forall \mathbf{z}((\mathbf{L}(\mathbf{b},\mathbf{y}) \wedge \mathbf{L}(\mathbf{y},\mathbf{z})) \rightarrow \mathbf{L}(\mathbf{b},\mathbf{z})), \mathbf{G_1}, L(a,b), L(a,c) \vdash L(b,c)$.
[1] $L(b,a), \forall \mathbf{x} \forall \mathbf{y} \forall \mathbf{z}((\mathbf{L}(\mathbf{x},\mathbf{y}) \wedge \mathbf{L}(\mathbf{y},\mathbf{z})) \rightarrow \mathbf{L}(\mathbf{x},\mathbf{z})), L(a,b), L(a,c) \vdash L(b,c)$.
[2] $\forall x \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z)), L(a,b), L(a,c) \vdash L(b,c), L(a,b)$.
Taikome $(\rightarrow \vdash)$ ir *silpninimo* taisykles. Gauname dvi [1], [2] sekvencijas. [2] sekvencija yra aksioma. Tęsiame pirmos ([1]) sekvencijos išvedimą.
$\mathbf{L}(\mathbf{a},\mathbf{b}) \rightarrow \mathbf{L}(\mathbf{b},\mathbf{a}), \mathbf{F_2}, \forall x \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z)), L(a,b), L(a,c) \vdash L(b,c)$.
Taikome $(\forall \vdash)$ ir *silpninimo* taisykles.
$\forall \mathbf{y}(\mathbf{L}(\mathbf{a},\mathbf{y}) \rightarrow \mathbf{L}(\mathbf{y},\mathbf{a})), \mathbf{F_1}, \forall x \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z)), L(a,b), L(a,c) \vdash L(b,c)$.
$C = \{a,b,c\}$.
$\forall \mathbf{x} \forall \mathbf{y}(\mathbf{L}(\mathbf{x},\mathbf{y}) \rightarrow \mathbf{L}(\mathbf{y},\mathbf{x})), \forall x \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z)), L(a,b), L(a,c) \vdash L(b,c)$.
$\forall x \forall y (L(x,y) \rightarrow L(y,x)), \forall x \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z)), \mathbf{L}(\mathbf{a},\mathbf{b}) \wedge \mathbf{L}(\mathbf{a},\mathbf{c}) \vdash L(b,c)$.
$\forall x \forall y (L(x,y) \rightarrow L(y,x)), \forall x \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z)) \vdash (\mathbf{L}(\mathbf{a},\mathbf{b}) \wedge \mathbf{L}(\mathbf{a},\mathbf{c})) \rightarrow \mathbf{L}(\mathbf{b},\mathbf{c})$.
Taikome $(\vdash \forall \ldots \forall)$.
$\forall x \forall y (L(x,y) \rightarrow L(y,x)), \forall x \forall y \forall z ((L(x,y) \wedge L(y,z)) \rightarrow L(x,z)) \vdash \forall \mathbf{x} \forall \mathbf{y} \forall \mathbf{z}((\mathbf{L}(\mathbf{x},\mathbf{y}) \wedge \mathbf{L}(\mathbf{x},\mathbf{z})) \rightarrow \mathbf{L}(\mathbf{y},\mathbf{z}))$.

Loginių operacijų bei kvantorinių taisyklių $(\exists \vdash), (\vdash \forall), (\exists \ldots \exists \vdash), (\vdash \forall \ldots \forall)$ taikymas "automatiškas". Yra dvi taisyklės $(\vdash \exists), (\forall \vdash)$, kurias taikant tenka dar **pagalvoti** kurį laisvąjį

kintamąjį pasirinkti. Nuo pasirinkimo priklauso išvedimo medžio aukštis.

Įrodyta, kad **tapačiai teisingų pirmos eilės predikatų logikos formulių aibė neišsprendžiama.** Tai reiškia, kad nėra galimybių nustatyti ar formulė tapačiai teisinga, ar ne.

Tai ką duoda sekvencinis predikatų skaičiavimas? Jei formulė tapačiai teisinga sekvenciniame skaičiavime **yra medis** - formulės įrodymas. Tai kame problema? Mes, kaip įprasta, nežinome ar duota formulė tapačiai teisinga, ar ne. Vykdydami paiešką ir nerasdami įrodymo nežinome **kada baigti paiešką.** Jei tapačiai teisinga, tai yra įrodymas - baigtinis medis (ir galim tęsti jo paiešką), bet, jei nėra tapačiai teisinga, tai paieška gali tęstis be galo ilgai.

Jei formulė *F* tapačiai teisinga, tai turime sekvencinį skaičiavimą, kuriame sekvencija $\vdash F$ įrodoma. Jei formulė tapačiai klaidinga, tai įrodoma sekvencija $\vdash \neg F$. O, jei formulė nėra nei tapačiai teisinga, nei tapačiai klaidinga? Tuomet apskritai negalima sugalvoti tokio skaičiavimo, kuriame būtų įrodomos tokio tipo formulės ir tik jos. Kodėl? Tarkime, sugalvojome tokį skaičiavimą (pažymėkime jį *H*), kuriame įrodomos formulės, kurios nėra nei tapačiai teisingos, nei tapačiai klaidingos (įrodomos tiktai tokio tipo formulės). Tuomet būtų ir algoritmas, kuriuo galima nustatyti ar formulės tapačiai teisingos, ar ne. Paleiskime lygiagrečiai tris programas: pirmąja ieškome sekvencijos $\vdash F$ įrodymo, antrąja sekvencijos $\vdash \neg F$ įrodymo, o trečiąja ieškome formulės *F* įrodymo skaičiavime *H*. O dabar galite ramiai sėdėti ir laukti rezultato. Po baigtinio žingsnių skaičiaus kompiuteris baigs darbą. Viena iš trijų programų paskelbs jums rezultatą. Deja, įrodyta, kad *tapačiai teisingų predikatų logikos formulių aibė neišsprendžiama.*

Nei tapačiai teisingų, nei tapačiai klaidingų, nei įvykdomų (arba įvykdomų, bet ne tapačiai teisingų) formulių aibės nėra algoritmiškai išsprendžiamos.

Semantinių lentelių metodas predikatų logikai

Semantinių lentelių metodas susideda iš *konjunkcijos, disjunkcijos taisyklių, dvigubo neigimo taisyklės* (jas jau aprašėme anksčiau) ir *kvantorinių taisyklių.*

Kvantorinės taisyklės:

$$\frac{\forall x F(x)}{F(\gamma)} \qquad \frac{\neg\exists x F(x)}{\neg F(\gamma)} \qquad \frac{\exists x F(x)}{F(\beta)} \qquad \frac{\neg\forall x F(x)}{\neg F(\beta)}$$

Kvantorinėse taisyklėse β, γ yra laisvieji kintamieji, kuriems keliami tie patys reikalavimai kaip ir sekvenciniame skaičiavime. Šaka vadinama *uždaraja*, jei joje yra dvi kurios nors formulės pavidalo *G*, $\neg G$, t.y. viena kitos neiginys. Šaka, kuri nėra uždara, vadinama atviraja. Medis vadinamas *uždaruoju*, jei visos jo šakos uždaros. Uždaras medis su šaknimi $\neg F$ vadinamas formulės *F išvedimu semantinių lentelių metodu.*

Pavyzdys:

Įrodysime sekvenciją $\vdash (\forall x P(x) \to \exists x Q(x)) \to \exists x (P(x) \to Q(x))$.

$$\neg[(\forall x P(x) \to \exists x Q(x)) \to \exists x (P(x) \to Q(x))]$$

$$\forall x P(x) \to \exists x Q(x) \qquad (1)$$
$$\neg\exists x(P(x) \to Q(x)) \qquad (2)$$

iš (1) gauname

$$\neg\forall x P(x) \quad | \quad \exists x Q(x)$$
$$\neg P(a) \qquad\qquad Q(a)$$

iš (2) gauname

$$\neg(P(a) \to Q(a)) \qquad\qquad \neg(P(a) \to Q(a))$$
$$P(a) \qquad\qquad P(a)$$
$$\neg Q(a) \qquad\qquad \neg Q(a)$$
$$\oplus \qquad\qquad \oplus$$

Pastaba. Kuo skiriasi *įrodymas* nuo *išvedimo*? Tarkime, turime pradinę sekvenciją $\Gamma \vdash F$ ir, naudodamiesi predikatų skaičiavimo taisyklėmis sukonstravome baigtinį medį, kurio visuose lapuose yra aksiomos. Kas tai, *įrodymas* ar *išvedimas*? Jei prielaidų sąrašas Γ pradinėje sekvencijoje yra tuščias, tai *įrodymas*. Jei prielaidų sąrašas Γ netuščias, tai *išvedimas*. Tos sąvokos tokia prasme tebevartojamos Hilberto skaičiavimuose (juos aprašysime vėliau). Kituose skaičiavimuose dažniausiai vartojamas *išvedimas*, suprantant, kad prielaidų sąrašas Γ gali būti ir tuščias. Terminas *įrodymas* vartojamas tada, kai norima *pabrėžti*, kad prielaidų sąrašas tuščias.

2.3 Normaliosios priešdėlinės formos

Kai norima suprogramuoti išvedimo paiešką sekvenciniu skaičiavimu ar semantinių lentelių metodu, dažniausiai formulės transformuojamos į tinkamesnį programavimui pavidalą, vadinamą *priešdėline normaliąja forma.*

Formulės F ***normaliąja priešdėline forma*** *vadiname jai ekvivalenčią formulę pavidalo P*(*M*); *čia M - formulė, kurioje nėra kvantorių* (*bekvantorė formulė dar vadinama* ***matrica***) *o P yra kvantorių seka, vadinama formulės F* ***prefiksu***.

Formulės

$$\forall x\exists y(P(x) \wedge Q(y)),$$
$$\forall x\forall y\forall z((L(x,y) \wedge L(y,z)) \to L(x,z)),$$
$$\forall x\forall y\forall z((F(x) \to F(y)) \to (\neg F(z) \to \neg F(x))).$$

yra normaliosios priešdėlinės formos. Pirmosios prefiksas yra $\forall x\exists y$, o $P(x) \wedge Q(y)$ yra matrica.

Formulės

$$\exists x A(x) \wedge \exists x B(x),$$
$$\forall x(\exists y P(x,y) \to \forall z \neg P(z,x)),$$
$$(\exists x A(x) \wedge \forall x(A(x) \to \exists y A(y))) \to \forall x A(x).$$

nėra normaliosios priešdėlinės formos.

Parodysime, kaip bet kuri predikatų logikos formulė transformuojama į jai ekvivalenčią normaliosios priešdėlinės formos.

Paaiškinsime atvejui, kai transformuojamoje formulėje iš loginių operacijų *yra tik* $\neg, \wedge, \vee$ (nėra $\rightarrow, \leftrightarrow, \dot{\vee}$). Be to, laikysime, kad, jei formulėje yra $\exists x$ (arba $\forall x$) įeitis, tai ji vienintelė (čia *x* kuris nors suvaržytas kintamasis). Tai pasiekiama *pervardijant* kintamuosius. Pervardijama remiantis ekvivalenčiomis formulėmis: $\exists x A(x) \equiv \exists y A(y), \forall x A(x) \equiv \forall y A(y)$ (čia *y* - naujasis individinis kintamasis, neaptinkamas formulėje *A*(*x*)).

Transformuojant naudojamės:

- 2.1 skyrelyje aprašytomis ekvivalenčiomis formulėmis:

$$\neg\forall x A(x) \equiv \exists x \neg A(x),$$
$$\neg\exists x A(x) \equiv \forall x \neg A(x).$$

- Formulėmis, kurių ekvivalentiškumą nesunku įrodyti (laikome, kad formulėje *B* nėra kintamojo *x* įeičių):

$$\forall x A(x) \wedge B \equiv \forall x (A(x) \wedge B),$$
$$\forall x A(x) \vee B \equiv \forall x (A(x) \vee B),$$
$$\exists x A(x) \wedge B \equiv \exists (A(x) \wedge B),$$
$$\exists A(x) \vee B \equiv \exists x (A(x) \vee B).$$

Pavyzdžiai:

1. $\exists x P(x) \vee \exists y (\forall x Q(x, y) \wedge P(y))$.
Pervardijam kintamuosius.
$\exists x P(x) \vee \exists y (\forall z Q(z, y) \wedge P(y))$.
Yra pasirinkimas. Galima iškelti prieš visą formulę $\exists x$ arba $\exists y$. Galim iškelti $\forall z$ prieš formulę $(Q(z, y) \wedge P(y))$ (*bet negalima* $\forall z$ *iškelti prieš* $\exists y (Q(z, y) \wedge P(y))$). Taigi, atsakymas nėra vienareikšmis. Jei nėra kokių nors papildomų reikalavimų normaliosios priešdėlinės formos prefiksui, tai galima rinktis bet kurį variantą.

$\exists x P(x) \vee \exists y (\forall z Q(z, y) \wedge P(y)) \equiv \exists x P(x) \vee \exists y \forall z (Q(z, y) \wedge P(y)) \equiv$
$\exists x \exists y (P(x) \vee \forall z (Q(z, y) \wedge P(y))) \equiv \exists x \exists y \forall z (P(x) \vee (Q(z, y) \wedge P(y)))$.

2. $\exists x \neg\forall y P(x, y) \wedge \forall x (P(x, x) \vee \exists y Q(x, y)) \equiv \exists x \neg\forall y P(x, y) \wedge \forall z (P(z, z) \vee \exists u Q(z, u)) \equiv$
$\exists x \exists y \neg P(x, y) \wedge \forall z (P(z, z) \vee \exists u Q(z, u)) \equiv \exists x \exists y \neg P(x, y) \wedge \forall z \exists u (P(z, z) \vee Q(z, u)) \equiv$
$\forall z (\exists x \exists y \neg P(x, y) \wedge \exists u (P(z, z) \vee Q(z, u))) \equiv \forall z \exists x (\exists y \neg P(x, y) \wedge \exists u (P(z, z) \vee Q(z, u))) \equiv$
$\forall z \exists x \exists u (\exists y \neg P(x, y) \wedge (P(z, z) \vee Q(z, u))) \equiv \forall z \exists x \exists u \exists y (\neg P(x, y) \wedge (P(z, z) \vee Q(z, u)))$.

Priminsime, kad **bendru atveju** nėra algoritmo nustatymui ar formulė tapačiai teisinga. Sekvencija $\vdash F$ išvedama tada ir tiktai tada, kai *F* tapačiai teisinga. Tai reiškia, kad atsiras baigtinis išvedimo medis, jei formulė tapačiai teisinga. Bet, kada nutraukti išvedimo paiešką? Jei nėra tapačiai teisinga, tai išvedimo paieška gali tęstis be galo ilgai.

Atskiru atveju, kai formulių normaliosios priešdėlinės formos yra pavidalo

$$\forall x_1 \forall x_2 \ldots \forall x_m \exists y_1 \exists y_2 \ldots \exists y_n M,$$

egzistuoja algoritmas, kuriuo nustatoma ar formulė tapačiai teisinga, ar ne (*m, n* bet kurie natūralieji skaičiai).

Pritaikius sekvencijai $\vdash \forall x_1 \forall x_2 \ldots \forall x_m \exists y_1 \exists y_2 \ldots \exists y_n M(x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n)$ taisyklę ($\vdash \forall \ldots \forall$), gauname $\exists y_1 \exists y_2 \ldots \exists y_n M(a_1, a_2, \ldots, a_m, y_1, y_2, \ldots, y_n)$. Kitais žingsniais y_i keisime kintamaisiais a_j. Iš viso galimų variantų baigtinis skaičius. Po baigtinio žingsnių skaičiaus baigsis išvedimo paieška. Gausime arba išvedimo medį (formulė tapačiai teisinga), arba medį, kuris nėra išvedimas (formulė nėra tapačiai teisinga).

Pavyzdys.
Transformuokime

$$\forall x \exists y A(x, y) \vee \forall y (B(y, y) \wedge \exists x A(y, x))$$

į normaliąją priešdėlinę formą.

Sprendimas:

$$\begin{gathered}
\forall x \exists y A(x, y) \vee \forall y (B(y, y) \wedge \exists x A(y, x)) \equiv \\
\forall v \exists u A(v, u) \vee \forall y (B(y, y) \wedge \exists x A(y, x)) \equiv \\
\forall v \exists u A(v, u) \vee \forall y \exists x (B(y, y) \wedge A(y, x)) \equiv \\
\forall v \forall y (\exists u A(v, u) \vee \exists x (B(y, y) \wedge A(y, x))) \equiv \\
\forall v \forall y (\exists u A(v, u) \vee \exists u (B(y, y) \wedge A(y, u))) \equiv \\
\forall v \forall y \exists u (A(v, u) \vee (B(y, y) \wedge A(y, u)))
\end{gathered}$$

Formulės prefiksas yra $\forall v \forall y \exists u$. Sakoma, kad *prefikso forma* $\forall\forall\exists$, t.y. $\forall^2\exists$. Matricoje (t.y formulėje $A(v, u) \vee \exists u (B(y, y) \wedge \exists x A(y, u))$) yra trijų individinių kintamųjų įeitys. Išvedimo paiešką atliekame tokiu būdu:

1. taikome taisyklę ($\vdash \forall\forall$); pakeičiame *v, y* kintamaisiais *a, b,*

2. taikome taisyklę ($\vdash \exists$) du kartus; pirmą kartą *u* keičiame į *a*, antruoju į *b*,

3. taikome *silpninimo* taisyklę (išbraukiame visas formules, prasidedančias kvantoriais),

4. gavome teiginių logikos formulę, kurią mokame patikrinti ar ji tapačiai teisinga.

2.4 Algoritmai

Idealioje matematikoje, sutinkamai su Hilberto programa, reikėtų nagrinėti tik funkcijas, kurias *mokame apskaičiuoti*. T.y. nagrinėti tik funkcijas, kurių apskaičiavimui žinome *būdą, metodą, algoritmą*. Intuityvi algoritmo sąvoka paprasta ir visiems suprantama. Bet, jeigu norime įrodyti, kad kurios nors problemos sprendimui neegzistuoja algoritmo, tai intuityvios sąvokos nepakaks. Reikia turėti *tikslų algoritmo sąvokos matematinį apibrėžimą*. Tuomet galėsim įrodyti, kad kažkuri tai funkcija nėra algoritmiškai apskaičiuojama. Nemokame mes jos apskaičiuoti, nemokės ir niekas ateityje. T.y. visų galimų algoritmų aibėje nėra ją apskaičiuojančio. Hilberto idėjos sudomino daugelį žymių matematikų ir buvo pasiūlyta daug **algoritmo** bei *algoritmiškai*

apskaičiuojamų funkcijų klasių apibrėžimų.

Buvo nagrinėjamos tik *funkcijos, kurių apibrėžimo sritimi yra natūraliųjų skaičių aibė*, o *reikšmių sritimi, kuris nors natūraliųjų skaičių poaibis*. 1931 metais K.Gödelis pirmasis aprašė apskaičiuojamų funkcijų klasę. A.Church'as, naudodamas kitus metodus, 1936 m. irgi aprašė apskaičiuojamų funkcijų klasę, kuri, pasirodo, sutampa su Gödelio klase. A.Church'as aprašytąją klasę pavadino *rekursyviosiomis funkcijomis*. Jis pirmasis paskelbė tezę (dabar vieni ją vadina Church'io teze, kiti Church'io - Turingo teze), kad *rekursyviųjų funkcijų klasė sutampa su apskaičiuojamų* (apibrėžtomis natūraliųjų skaičių aibėje) *funkcijų aibe*. Tais pačiais metais S.Kleene apibendrino A.Church'io klasę, nagrinėdamas ir dalines rekursyviąsias funkcijas.

Rekursyviųjų funkcijų klasė yra algoritmiškai apskaičiuojamų funkcijų klasė. O kas yra algoritmas? Į tą klausimą atsakė E.Postas (1936 m.) ir A.Turingas (1937 m.), beveik vienu metu aprašę algoritmus, remdamiesi A.Church'io ir S.Kleene rezultatais, matematinių mašinų pavidalu. Abiejų mašinos nedaug skyrėsi ir vėliau pradėta jas vadinti vienu vardu - *Turingo mašinos*.

Tai kas gi vis tiktai yra algoritmas? *Egzistuoja* ***algoritmas*** *funkcijai apskaičiuoti* tada ir tiktai tada, kai egzistuoja ją apskaičiuojanti **Turingo mašina**. *Turingo mašina ir yra algoritmas*. O kodėl? Jau vien todėl, kad dar niekas nesugalvojo algoritmo, kurio nebūtų galima redukuoti į Turingo mašiną. Turingo mašinomis apskaičiuojamų funkcijų klasė sutampa su rekursyviųjų funkcijų klase.

1945 m. Dž.Ekerto ir DŽ.Moklio kompiuteris ENIAC sukurtas pagal Turingo aprašytą matematinės mašinos modelį. Tai vienas pirmųjų kompiuterių pasaulyje. *Šiuolaikiniai kompiuteriai atitinka Turingo mašinų modelius.* ***Jei ne Hilberto programa, kažin ar mes turėtumėm šiandien kompiuterius.***

Šios, mokslinėje bei mokomojoje literatūroje, naudojamos sąvokos yra **sinonimai** (jomis nusakoma ta pati funkcijų klasė):

- **apskaičiuojamųjų funkcijų** (computable functions) **klasė** (aibė),
- **algoritmiškai apskaičiuojamųjų funkcijų klasė**,
- **rekursyviųjų funkcijų klasė**,
- **suprogramuojamų funkcijų klasė**.

Aprašysime vieną iš pagrindinių apskaičiuojamų funkcijų klasės formalizmų - rekursyviąsias funkcijas.

Šiame skyrelyje supažindinsim ir su 1953 metais amerikiečio Wang Hao aprašytais operatoriniais algoritmais. Tai dar vienas *algoritmo apibrėžimas*.

Tai istorija. O dabar visi jau žino ir dar vieną algoritmo sąvokos apibrėžimą. Juk mokykloje yra informatikos pamokos. Jose supažindinama su kuria nors programavimo kalba. **Algoritmas**, kuriai nors užduočiai spręsti, **yra programa**.

Tada kyla klausimas, o kam reikia žinoti Turingo mašinas, Wang Hao algoritmus, rekursyviąsias funkcijas? Todėl, kad algoritmas, kuriai nors užduočiai spręsti, *gali būti*, bet ***gali ir nebūti***. Kai norima įrodyti, kad ***nėra algoritmo***, dažniausiai naudojamasi Turingo mašinomis, Wang Hao algoritmais bei rekursyviosiomis funkcijomis. Tai yra *paprasčiausias kelias įrodymui*, kad nėra algoritmo.

Kaip suprasti, kad nėra algoritmo funkcijai *f* apskaičiuoti? Studentui lengviausia būtų pasakyti (bet nelengva jam bus suprasti), kad funkcijos *f* ***neįmanoma suprogramuoti***. Tai reiškia, kad tarp visų kompiuteriui programų (nepriklausomai nuo programavimo kalbos) nėra apskaičiuojančios *f*. Pavyzdžiai tokių funkcijų bus kituose skyriuose.

Nagrinėjamos tik funkcijos apibrėžtos natūraliųjų skaičių aibėje su reikšmėmis kurio nors natūraliųjų skaičių netuščiame poaibyje. Apibendrinti tokias funkcijas, iki racionaliųjų skaičių aibės, nesudėtinga. Visas sudėtingumas - aprašyti algoritmiškai apskaičiuojamas funkcijas natūraliųjų skaičių aibėje. O, jei sugalvosit apskaičiuoti funkciją, kai argumentas iracionalus skaičius, tai nespėsit iki mirties užrašyti vien tik argumento reikšmės.

Priminsime, kad aibė *A vadinama skaičiąja*, jei yra abipusiai vienareikšmė atitiktis (bijekcija) tarp aibių *A* ir $N^- = \{1, 2, 3, \ldots\}$ (arba tarp *A* ir $N = \{0, 1, 2, 3, \ldots\}$; naudojamės tuo apibrėžimu, kuris patogesnis nagrinėjamu atveju). Paaiškinsime, kodėl aibė, visų kompiuteriui programų, yra skaičioji. Aibė visų žodžių, pavyzdžiui, abėcėlėje $\{a, b, c\}$, yra skaičioji. Visus žodžius galima išvardinti taip vadinama *leksikografine tvarka*:

$$a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, \ldots$$

Vienodo ilgio žodžius rašome panašia tvarka, kaip kad jie rašomi žodynuose. Jei pirmosios raidės sutampa, tai lyginame pagal antrąsias raides. Jei ir antrosios sutampa, tai pagal trečiąsias ir t.t. Kaip matote, visų pirma parašėme žodžius ilgio 1 (žodžiai iš vienos raidės), po to, ilgio 2 ir t.t. Remiantis šia idėja galima išvardinti (t.y. įrodyti, kad visų žodžių aibė skaičioji) ir visus žodžius bet kurios baigtinės abėcėlės, įvedę griežtą tvarką tarp abėcėlės raidžių.

O kas gi yra programa kuria nors programavimo kalba? Kai kuriuose vadovėliuose aprašomos jose nagrinėjamų programavimo kalbų abėcėlės. Pavyzdžiui, Paskalio kalbos (ar dar prisimenate tokią?) abėcėlė aprašyta vadovėlyje "V.Tumasonis. *Paskalio programavimo kalba.* Vilnius. Mokslas. 1983". Abėcėlės yra baigtinės. Programa, tai *žodis baigtinėje abėcėlėje.* Vadinasi, visų programų aibė (ji begalinė) yra skaičioji, Turbūt nenustebinsiu sakydamas, kad apskaičiuojamų funkcijų Paskal kalba aibė sutampa su apskaičiuojamų Java kalba. O apskaičiuojamos Java kalba sutampa su apskaičiuojamomis primityviomis (iš pirmo žvilgsnio) vienajuostėmis Turingo mašinomis.

Aišku kas yra *skaičioji aibė*. Su ta sąvoka supažindinama ne viename universiteto paskaitų kurse. O kokia aibė vadinama **rekursyviai skaičiąja**? Aibė *A* vadinama *rekursyviai skaičiąja*, jei **egzistuoja algoritmiškai apskaičiuojama** abipusiai vienareikšmė atitiktis (bijekcija) tarp aibių *A* ir natūraliųjų skaičių aibės *N* (t.y., *jei galima suprogramuoti bijekciją*). Plačiau skaitykite vadovėlyje [1].

Tikiuosi sudominau ir norite žinoti bent vieną skaičiosios, bet ne rekursyviai skaičiosios aibės pavyzdį. Tokią aibę jau minėjau šioje knygoje. *Aibė pirmos eilės predikatų logikos formulių, kurios nėra nei tapačiai teisingos, nei tapačiai klaidingos*. Formulės yra žodžiai tam tikroje baigtinėje abėcėlėje. Formulių aibė skaičioji, tiksliau, ji yra rekursyviai skaičioji (pastaruosius du teiginius siūlyčiau aiškintis per logikos užsiėmimus su dėstytoju). *Aibė pirmos eilės predikatų logikos formulių, kurios nėra nei tapačiai teisingos, nei tapačiai klaidingos yra begalinis skaičiosios aibės* (*visų formulių aibės*) *poaibis, todėl sudaro skaičiąją aibę*. Bet, deja, įrodyta, kad *ji nėra rekursyviai skaičioji*.

Turbūt nelabai supratote.
Šiame skyrelyje nagrinėjamos funkcijos, kurių apibrėžimo aibė yra natūraliųjų skaičių. Pateiktame pavyzdyje kalbama apie bijekciją tarp *formulių aibės* ir N^-. Formulės yra žodžiai tam tikroje baigtinėje abėcėlėje. Jas galima sunumeruoti natūraliaisiais skaičiais. Yra žinoma įvairių numeravimo būdų. Dažniausiai naudojamas tapusia klasika *formulių Gödelio numeracija* (apie ją vėliau, kitame skyriuje).

Kuo skiriasi sąvoka *indukcija* nuo *rekursijos*? Tarkime, reikia įrodyti, kad tvirtinimas *P*(*n*) teisingas su $n = 0, 1, 2, \ldots$

Iš pradžių įrodome teisingumą *P*(0), o paskui *P*(1), *P*(2), *P*(3), ... remiantis tuo, kad, *jei P*(*i*) *teisingas, tai ir P*($i + 1$) *teisingas,* tai **indukcija**.

Jei iš pradžių įrodome teisingumą *P*(0), o paskui *P*(1), *P*(2), *P*(3), ... remiantis tuo, kad, *jei* $P(m)(m \leq i)$ *teisingi, tai ir P*($i + 1$) *teisingas,* tai **rekursija**.

Pradinėmis arba **bazinėmis** pasirinktos tokios funkcijos, kurios niekam nekelia abejonių, kad jas mokame apskaičiuoti.

Pradinės funkcijos. Konstanta 0, paskesniojo nario funkcija *s*(*x*) ir projekcijos funkcijos $pr_p^i(x_1, \ldots, x_p) = x_i(p \geq 1; 1 \leq i \leq p)$.

$s(x)$ yra Peano apibrėžta ir taip literatūroje žymima *paskesnio nario funkcija* $s(x) = x + 1$. Rekursyviųjų funkcijų teorijoje tokia jos prasmė paaiškėja tik apibrėžus kompozicijos operatorių. Nes kol kas neturime vieneto.

Algoritmiškai apskaičiuojamų funkcijų klasė nusakoma remiantis tokia idėja: aprašomos pradinės funkcijos, kurios *savaime aišku* yra apskaičiuojamos, ir trys operatoriai (taisyklės), kaip iš turimų apskaičiuojamų funkcijų gauti kitas.

Kompozicijos (superpozicijos) operatorius. Tarkime, turime $(k + 1)$ funkciją:

$$f(y_1, \ldots, y_k), g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n).$$

Iš jų, panaudojus *kompozicijos* (*superpozicijos*) *operatorių*, gaunama *n* argumentų funkcija $f(g_1, \ldots, g_k)$. Pritaikę kompozicijos operatorių funkcijoms *s*(*x*) ir 0, gauname *s*(0); pritaikę

funkcijoms *s*(*x*) ir *s*(0), gauname $s(s(0))$. Taigi, galima gauti funkcijas (termus) *s*(0), $s(s(0))$, $s(s(s(0)))$ ir t.t. Juos žymime (suteikiame vardus) 1, 2, 3 ir t.t. Tiktai **žymime**, o **turime** *s*(0), $s(s(0))$, $s(s(s(0)))$ ir t.t.

Primityviosios rekursijos operatorius. Jį taikysime dvejoms funkcijoms. Tarkime, yra dvi funkcijos $g(x_1, \ldots, x_{n-1}), h(x_1, \ldots, x_{n+1})$. Viena jų $(n-1)$ argumento, o antroji - $(n+1)$ argumento (štai kur praverčia projekcijos funkcijos; jų dėka galime *fiktyviai* padidinti funkcijų argumentų skaičių). Apibrėžiame naują *n* argumentų funkciją tokia schema:

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

Atveju, kai $n = 1$, schema tokia (*a* - konstanta):

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

Atveju, kai $n = 2$, schema bus tokia:

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

Minimizavimo operatorius. Taikome jau turimai *n* argumentų $(n = 1, 2, \ldots)$ funkcijai *f*. Apibrėžiame naują, taip pat *n* argumentų funkciją $g(x_1, \ldots, x_n)$. Jos reikšmė lygi mažiausiam *y*, su kuriuo $g(x_1, \ldots, x_{n-1}, y) = x_n$. Naujai gauta funkcija privalo būti algoritmiškai apskaičiuojama. Todėl, nusakydami naują funkciją *g*, privalome nurodyti ir algoritmą, kaip bus ieškomas mažiausias *y*:

- jei $f(x_1, \ldots, x_{n-1}, 0) = x_n$, tai funkcijos *g* reikšmė lygi 0; jei ne, tai tikriname ar $f(x_1, \ldots, x_{n-1}, 1) = x_n$,

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

Suprantama, kad tokia aprašytoji mažiausio *y* paieška gali tęstis be galo ilgai. Tuo atveju sakoma, kad su duotomis argumentų reikšmėmis *funkcija neapibrėžta*, arba jos *reikšmė* lygi *begalybei*.

Minimizacijos operatoriumi nauja gauta funkcija *g* žymima taip:

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

Pati mažiausia aibė, kuriai priklauso pradinės funkcijos ir kuri uždara kompozicijos bei primityvios rekursijos atžvilgiu, vadinama ***primityviai rekursyvių funkcijų aibe (klase)***. Ji žymima *PR*.

Pati mažiausia aibė, kuriai priklauso pradinės funkcijos ir kuri uždara kompozicijos, primityvios rekursijos bei minimizavimo atžvilgiu, vadinama ***dalinių rekursyviųjų funkcijų aibe (klase)***. Ji žymima *DR*.

*Visur apibrėžta dalinė rekursyvioji funkcija vadinama **bendraja rekursyviąja funkcija**.* Ji žymima *BR*.

Pavyzdžiai:

1. $x + n \in PR$, nes $s(s(\dots s(x))\dots) = x + n$. Funkcija gauta panaudojus tik kompozicijos operatorių $(n-1)$ kartą pradinei funkcijai $s(x)$.

2. Skirtumo funkcija $x - y \in DR$, nes $x - y = \mu_z(y + z = x)$ Minimizavimo operatoriumi aprašome funkcijas nusakytas "neišreikštiniu būdu".

Minimizavimo operatoriumi patogu aprašyti atvirkštines funkcijas. $f^{-1}(x) = \mu_y(f(y) = x)$ (toks mažiausias *y*, su kuriuo $f(y) = x$). Pavyzdžiui, duota funkcija $y = x^2$. Jai atvirkštinė yra $\mu_y(y^2 = x)$, t.y. $x = \sqrt{y}$.

Daugiau uždavinių (8 - 11) rasite pratimų skyrelyje. O jų atsakymus ir sprendimus tam skirtame skyriuje.

*Netuščia aibė **rekursyviai skaiti**, jei ji sutampa su kurios nors primityviai rekursyvios funkcijos reikšmių aibe.*

*Egzistuoja rekursyviai skaičios **neišsprendžiamos** (nėra rekursyvi, charakteringossios funkcijos neįmanoma suprogramuoti) aibės.*

Wang Hao algoritmai

Wang Hao algoritmas vadinamas *programa*. Programa yra baigtinė *komandų* seka. Komandos užrašomos tokiu pavidalu

$$i : (f; m; j).$$

Čia *i* yra komandos numeris (natūralusis skaičius) programoje, *f* - komandos veiksmas. Komandos veiksmai yra tik pavidalo: $\times 2, \times 3, \times 5, : 30$. *m, n* yra komandų numeriai. *m* yra komandos, į kurią bus pereinama (jei f apibrėžta), numeris. Jei f neapibrėžta, pereinama į komandą numeriu *j*.

Kadangi daugyba yra visada apibrėžtas veiksmas, tai daugybos atveju užrašas paprastesnis

$$i : (f; m).$$

Yra dar komanda $0 :$ *stop*. Jei pereiname į komandą numeriu nulis, tai programa baigia darbą. Programos darbo rezultatas - paskutinės komandos, prieš pereinant į *stop*, duomenys. Jei nepatenkama į komandą *stop*, tai laikoma, kad programos darbo rezultatas *neapibrėžta* (su duotais pradiniais duomenimis).

Išvardinsime visų komandų pavidalus:

$$0 : \mathit{stop}, \quad i : (\times 2; m), \quad i : (\times 3; m), \quad i : (\times 5; m), \quad i : (: 30; m; j). \qquad (2.2)$$

Čia $i > 0$.

Įrodyta, kad *kokia bebūtų dalinė rekursyvioji funkcija f(n), galima rasti tokią programą, kad pritaikę ją duomenims* 2^n, *po baigtinio žingsnių skaičiaus gausime* $2^{f(n)}$, *jei f(n) apibrėžta. Programa dirbs be galo ilgai, jei f(n) neapibrėžta.*

Duomenys programoje yra pavidalo $2^l \cdot 3^u \cdot 5^v$; čia *l, u, v* - natūralieji skaičiai. Pradiniais duomenimis dažniausiai būna 2^n.

Pavyzdžiai.

Pradiniai duomenys 2^n (*n* - kuris nors natūralusis skaičius). Parašykite Wang Hao programą:

1. Po baigtinio žingsnių skaičiaus programa baigia darbą. Darbo rezultatas 2^{n+3}.

Atsakymas: 0 : *stop,* 1: $(\times 2; 2)$, 2 : $(\times 2; 3)$, 3 : $(\times 2; 0)$.

2. Jei $n > 0$, tai darbo rezultatas 2^{n-1}. Jei $n = 0$, tai programa dirba be galo ilgai.

Atsakymas: 0 : *stop*, 1 : $(\times 3; 2)$, 2 : $(\times 5; 3)$, 3 : $(: 30; 0; 4)$, 4 : $(\times 2; 4)$.

Pastaba. Tikiuosi supratote, kad komanda $i : (: 2; m)$ leistina (gaunama naudojant taisykles (2.2)). Jei norite padalinti iš 2, tai dauginkite iš 3 ir 5, dalinkite iš 30. Tas pats ir su dalyba iš 3 arba 5.

Todėl komandų sąrašą (2.2) papildysime naujomis komandomis:

$$i : (: 2; m; j), i : (: 3; m; j), i : (: 5; m; j).$$

Įrodykite, kad komandos $i : (: 6; m; j), i : (: 10; m; j), i : (: 15; m; j)$ taip pat leistinos.

3. Darbo rezultatas 5^n.

Atsakymas: 0 : *stop*, 1: $(: 2; 2; 0)$, 2 : $(\times 5; 1)$.

Reziumė.

Pagrindiniai šio skyrelio teiginiai, kuriais vėliau naudosimės:

- Yra apskaičiuojamos funkcijos (rekursyvios, suprogramuojamos) ir neapskaičiuojamos. Matematikai neskirsto taip funkcijų. Sąvoka **funkcija** matematikoje suprantama kaip *kuri nors funkcija* (apskaičiuojama arba neapskaičiuojama).

- Yra apskaičiuojamos aibės (rekursyviai skaičiosios, suprogramuojamos) ir neapskaičiuojamos. Natūraliųjų skaičių begalinis poaibis *A suprogramuojamas*, jei turime tokią kompiuteriui programą Π, kad koks bebūtų *n*, po baigtinio žingsnių skaičiaus programa baigia

darbą ir rezultatas yra kuris nors aibės *A* elementas. Sąraše $\Pi(0), \Pi(1),$
$\Pi(2), \ldots$ yra visi (ir tiktai jie) aibės *A* elementai. Kai kurie elementai sąraše gali būti ne vieną kartą.
Matematikai neskirsto taip aibių. Sąvoka **aibė** suprantama kaip bet kuris *įsivaizduojamas* poaibis. Jis gali būti suprogramuojamas, gali ir nebūti.

- Egzistuoja suprogramuojami begaliniai natūraliųjų skaičių poaibiai, kurių papildiniai (iki natūraliųjų skaičių aibės) nėra suprogramuojami.

2.5 Pratimai

1. Formalizuoti teiginius naudojant lygybės predikatą = (individinių konstantų aibė *R*):

a) egzistuoja ne mažiau kaip du elementai, su kuriais formulė *F*(*a*) yra teisinga,

b) egzistuoja lygiai du elementai, su kuriais formulė *F*(*a*) teisinga,

c) egzistuoja ne daugiau kaip du elementai, su kuriais formulė *F*(*a*) teisinga.

2. *D* kuri nors netuščia aibė. Predikatas $a \in E$ (*E* - kuris nors aibės *D* poaibis) teisingas tada ir tiktai tada, kai aibės *D* elementas *a* priklauso aibei *E*. *A, B, C* yra aibės *D* poaibiai. Predikatų logikos kalba parašykite (formalizuokite) teiginius:

a) $A = B \cap C$, b) $A = B \cup C$, c) $A = \emptyset$, d) $A = D$, e) *A* yra aibės *B* papildinys.

3.*D* yra taškų ir tiesių, kurioje nors plokštumoje, aibė. Joje apibrėžti predikatai:
T(*a*) yra teisingas tada ir tiktai tada, kai *a* yra taškas,
I(*a*) teisingas tada ir tiktai tada, kai *a* yra tiesė,
$a = b$ teisingas tada ir tiktai tada, kai *a, b* yra arba sutampantys *taškai*, arba sutampančios *tiesės*,
$a \in b$ teisingas tada it tiktai tada, kai *a* yra taškas, *b* - tiesė ir *taškas yra tiesėje*.
Parašykite formulėmis teiginius:

a) per bet kuriuos du taškus galima nubrėžti tiesę,

b) per bet kuriuos skirtingus taškus galima nubrėžti vienintelę tiesę,

c) skirtingos (nesutampančios) tiesės *a, b* yra lygiagrečios,

d) kiekvienoje tiesėje yra bent du skirtingi taškai,

e) egzistuoja susikertančios tiesės,

f) bet kurios dvi tiesės arba sutampa, arba nesusikerta.

4. Naudodami 3 užduoties predikatus, parašykite formule Riemanno aksiomą.

Riemanno aksioma. Per tašką, nesantį tiesėje, nėra (negalima nubrėžti) nei vienos tiesės nekertančios turimos tiesės.

5. Naudodami 3 užduoties predikatus, parašykite formule Lobačevskio aksiomą.

Lobačevskio aksioma. Per tašką, nesantį tiesėje, galime nubrėžti mažiausiai dvi skirtingas tieses, nekertančias turimos tiesės.

6. $D = \{1, 2, 3\}$. Aibėje *D* apibrėžtas predikatas *P*(*x, y*) nusakytas lentele:

$x \setminus y$	1	2	3
1	*t*	*t*	*k*
2	*t*	*k*	*t*
3	*k*	*k*	*t*

Nustatyti ir paaiškinti kurios iš formulių teisingos:

a) $\exists x P(x,x), \forall x P(x,x), \exists x \neg P(x,x), \forall x \neg P(x,x)$,

b) $\exists x \exists y (P(x,y) \to \neg P(y,x)), \forall x \forall y (P(x,y) \to \neg P(y,x)), \forall x \exists y (P(x,y) \to \neg P(y,x)), \exists x \forall y (P(x,y) \to \neg P(y,x))$.

7. Individinių konstantų aibė $D = \{A, B, C\}$. Transformuoti predikatų logikos formules į ekvivalenčias teiginių logikos formules:

a) $\forall x P(x) \to \exists x Q(x)$, b) $\forall x \exists y (P(x) \wedge Q(y))$, c) $\forall x \exists y P(x,y)$.

8. Raskite bent vieną struktūrą, kurioje formulė klaidinga:

a) $\forall x \forall y (P(x,y) \to P(y,x)) \to \forall x P(x,x)$,

b) $\exists x (P(x) \to Q(x))$,

c) $(\exists x P(x) \to \exists x Q(x)) \to \forall x (P(x) \to Q(x))$,

d) $\exists x (P(x) \wedge Q(x)) \leftrightarrow (\exists x P(x) \wedge \exists x Q(y))$,

e) $\forall x (P(x) \vee Q(x)) \leftrightarrow (\forall x P(x) \vee \forall x Q(x))$.

9. Išveskite sekvenciniame predikatų logikos skaičiavime:

a) $\exists x P(x,x) \vdash \exists x \exists y P(x,y)$,

b) $\exists x P(x) \to \forall x Q(x) \vdash \forall x (P(x) \to Q(x))$,

c) $\exists x (P(x) \wedge Q(x)) \vdash \neg \forall x (P(x) \to \neg Q(x))$.

10. Išveskite semantinių lentelių metodu:

a) $\vdash (\exists x P(x) \vee \exists x Q(x)) \leftrightarrow \exists x (P(x) \vee Q(x))$,

b) $\vdash (\forall x P(x) \wedge \forall x Q(x)) \leftrightarrow \forall x (P(x) \wedge Q(x))$.

11. Transformuokite į normaliąją priešdėlinę formą:

a) $\forall x \exists y A(x,y) \to \forall y (B(y,y) \wedge \exists x A(y,x))$,

b) $\forall x (P(x,x) \wedge \exists y \neg P(x,y)) \vee \forall x (Q(x,x) \wedge \exists z (P(z,x) \vee \forall u \neg P(x,u)))$,

c) $\exists x \forall y P(x,y) \to \forall x \exists y P(y,x)$.

12. Raskite formulės normaliąją priešdėlinę formą su minimaliu kvantorinių kompleksų skaičiumi prefikse:

a) $\exists x \forall y P(x,y) \vee \exists x \forall y Q(x,y)$,

b) $\forall x \exists y \forall z P(x,y,z) \wedge \forall u \exists v \forall w Q(u,v,w)$,

c) $\forall x \exists y P(x,y) \vee (\forall u \exists z Q(u,z) \wedge \exists v \forall w R(v,w))$.

13. Parašykite įvykdomą formulę struktūroje, kurios individinių konstantų aibėje:

a) ne mažiau kaip du elementai, b) ne mažiau kaip trys elementai.

14. Įrodykite, kad funkcijos yra primityviai rekursyvios:

a) $x + y$, b) $x \cdot y$, c) $n!$, kai $0! = 1$, d) $x \dot{-} 1$, e) $x \dot{-} u$,

f) $sg(x)$, g) $max(x,y)$, h) $min(x,y)$,i) $|x - y|$,

j) funkcija $u(x_1, \ldots, x_n, x_{n+1})$ primityviai rekursyvi; įrodykite, kad ir $f(x_1, \ldots, x_n, x_{n+1})$ yra primityviai rekursyvi:

$$f(x_1, \ldots, x_n, x_{n+1}) = \sum_{i=0}^{x_{n+1}} u(x_1, \ldots, x_n, i).$$

15. Kokią funkciją $f(x, y)$ apskaičiuojame primityviosios rekursijos schema:

a) $g(x) = x$, $h(x, y, z) = z$,

b) $g(x) = x$, $h(x, y, z) = x + z$,

c) $g(x) = x$, $h(x, y, z) = x + y$.

16. Raskite funkcijos atvirkštinę:

a) $y = x$.

b) $y = x \dot{-} 1$.

c) $y = 2^x$.

17. Pradiniai duomenys 2^n $(n > 0)$. Parašykite Wang Hao programą, kuri visada baigia darbą, o darbo rezultatas:

a) jei *n* dalijasi iš 4, tai 3; priešingu atveju 5,

b) $2 \cdot 3^n$.

18. Parašykite Wang Hao programą. Pradiniai duomenys $2 \cdot 3^n$ $(n > 0)$. Programa visada baigia darbą, o darbo rezultatas 2^{2^n}.

3 Skyrius
FORMALIOS AKSIOMINĖS TEORIJOS

3.1 Termai

Predikatų logikos kalba patogi matematinių samprotavimų formalizavimui. Formulėse naudojame predikatus. Matematikoje naudojamos ir funkcijos. Logikoje galima be jų apsieiti. Pavyzdžiui, vietoje $a + b$ pakanka nagrinėti predikatą $S(a, b, c)$, kuris teisingas tada ir tiktai tada, kai *c* lygus $a + b$. Bet, turint funkcijas, suprantamiau (patogiau, formulės trumpesnės) aprašomi matematiniai teiginiai. Be to, kai kurios išvedimo sistemos (pavyzdžiui, rezoliucijų metodas) pritaikytas darbui su formulėmis, kuriose yra funkciniai simboliai. Papildę kalbą funkcijomis gauname **logiką su funkciniais simboliais.** Funkcijos, kaip ir predikatai, apibrėžiami vienoje ir toje pačioje aibėje - *individinių konstantų aibėje*. Tik predikatų reikšmių aibe yra *t, k*, o funkcijų - individinių konstantų aibė.

Taigi, naudojamos dviejų rūšių funkcijos. Pirmosios vadinamos *predikatais*, o antrosios, tiesiog *funkcijomis.*
Yra sąvokos *predikatinis kintamasis* ir *funkcinis simbolis*. Jas vartojame, kai kalbama ne apie konkrečius predikatus ar funkcijas.

Pavyzdžiui, *nagrinėjame funkciją f*(*a, b*) *su apibrėžimo ir reišmių aibe N*. Mes neturime konkrečios funkcijos. Nurodome tik, kad kalba eina apie *kažkurią dviejų argumentų funkciją* su apibrėžimo ir reikšmių aibe *N*. Dažniausiai tokiu atveju sakoma *turime funkcinį simbolį f*(*a, b*). Nors galima sakyti ir funkcija, jei iš konteksto aišku, kad tai yra konkreti funkcija.

Termo apibrėžimas.
1. *Individinė konstanta yra termas.*
2. *Individinis laisvasis kintamasis yra termas.*
3. *Jei f yra n-vietis funkcinis simbolis* (*funkcija*) ir $t_1, \ldots, t_n$ *- termai, tai* $f(t_1, \ldots, t_n)$ taip pat yra termas.

Pastaba. Termo apibrėžimas skiriasi nuo apibrėžimo knygoje [1] (138 psl.). Termu mūsų knygoje, iš kintamųjų, gali būti tik *laisvasis.*

Pavyzdžiai: Individinių konstantų aibė $D = \{0, 1\}, f$ - vienvietis funkcinis simbolis.
Termų pavyzdžiai: $0, 1, a, b, c, f(a), f(0), f(1), f(f(0)), f(f(1)), f(f(f(c)))$.

Individinių konstantų aibė $D = \{0\}$. Funkcijos $s(a), a + b, a \cdot b$.
Termų pavyzdžiai: $0, b, (a + 0) \cdot s(0), s(s(b)) + ((c + 0) \cdot s(s(s(0))))$.

Formulės apibrėžimas.

1. *Jei P yra n-vietis predikatas (predikatinis kintamasis) ir* $t_1, \ldots, t_n$ *- termai, tai* $P(t_1, \ldots, t_n)$ *yra formulė*. Ji vadinama **atomine formule**.

2. *Jei F, G - formulės, tai* $(\neg F), (F \vee G), (F \wedge G), (F \leftrightarrow G), (F \rightarrow G)$ *- irgi formulės.*

3. *Jei F - formulė ir a - laisvasis kintamasis, tai koks bebūtų suvaržytas kintamasis x, neįeinantis į formulę F, reiškiniai* $(\exists x F(a/x)), (\forall x F(a/x))$ *yra irgi formulės.* Čia $F(a/x)$ žymi reiškinį, gautą pakeitus formulėje *F* visas laisvojo kintamojo *a* įeitis kintamuoju *x*.

Formulės konstruojamos pradedant atominėmis. Galutinės formulės išorinius skliaustus galima nerašyti.

Teorijos, kuriose visos aksiomos užrašytos predikatų logikos formulėmis ir įrodymai vykdomi naudojant predikatų logikos skaičiavimą, vadinamos **formaliosiomis aksiominėmis teorijomis.**

Formaliosios aksiominės teorijos nusakomos: a) *signatūra* (nurodomos teorijos *konstantos, predikatai* ir *funkcijos*), b) *teorijos aksiomų sąrašu*, užrašytu predikatų logikos formulėmis, kuriose konstantos, **konkretūs** predikatai ir funkcijos yra tik iš nurodytųjų signatūroje, c) *predikatų logika* (*aksiomų ir taisyklių sąrašas*).

Formali aksiominė teorija ≡ signatūra + teorijos aksiomos + predikatų logika

Teorijos pavyzdys. Tarkime, turime aibę, kurioje apibrėžta binarinė operacija $*$ (dviejų argumentų funkcija). Be to, aibė uždara operacijos $*$ atžvilgiu (jei *a, b* priklauso aibei, tai ir $a*b$ turi priklausyti). Tokia aibė su operacija (algebrinė struktūra) vadinama *grupe*, jei tenkina dar ir tam tikras savybes (jos nusakomos aksiomų pavidalu). Likusios *grupės* savybės išvedamos naudojant predikatų logiką.

Priminsime, kad predikatų logikoje naudojame kintamuosius (jie nepriklauso signatūrai). Jie fiksuoti ir priklauso predikatų logikos kalbai:

laisvieji kintamieji $a, b, c, d, e, a_1, a_2, a_3, \ldots$

suvaržytieji kintamieji $x, y, z, u, v, w, x_1, x_2, x_3, \ldots$

Deja, grupių teorijoje naudojama *konstanta* žymima raide *e*. Todėl grupių teorijoje laisvieji predikatų logikos kintamieji yra be raidės *e*:

laisvieji kintamieji $a, b, c, d, a_1, a_2, a_3, \ldots$

Grupių teorija.

Signatūra:

- *konstantos*: *e* (kairysis neutralusis grupės elementas).
- *predikatai*: $a = b$. Vienas dvivietis predikatas (lygybės predikatas).
- *funkcijos*: $a * b$, *inv*(*a*) (kairysis atvirkštinis elementas). Viena dviejų, o kita vieno argumento funkcijos.

Teorijos aksiomos:

1. $\forall x(e * x = x)$

2. $\forall x(inv(x) * x = e)$

3. $\forall x \forall y \forall z((x * y) * z = x * (y * z))$ Asociatyvumo savybė.

Pavyzdžiai:

a) grupių teorijos termai: $e, a, b, e * a, inv(inv(a)), ((inv(b) * e) * inv(e),$

b) parašyti formule teiginį: *kiekvienas kairysis atvirkštinis elementas yra ir dešinysis atvirkštinis elementas* (atsakymas: $\forall x(x * inv(x) = e)$,

c) parašyti formule teiginį: *bet kuris elementas lygus neutraliajam tada ir tiktai tada, kai jis lygus algebrinei operacijai su savimi* (atsakymas: $\forall x(x * x = x \leftrightarrow x = e)$).

Dar vienas formalios aksiominės teorijos pavyzdys.

Signatūra:

- *konstantos* (apibrėžimo aibė): 0, 1, 2, 3, ...
- *predikatai*: = Vienas lygybės predikatas.
- *funkcijos*: +, $\cdot$ Dvi dviejų argumentų funkcijos (sudėtis, daugyba).

Užduotys:
1. Parašyti formulę, su vienu laisvuoju kintamuoju *a*, teisingą tada ir tiktai tada, kai:

a) *a* yra lyginis skaičius,

b) *a* yra pirminis skaičius.

Atsakymai:
a) $\exists x(a = x + x)$ arba $\exists x(a = 2 \cdot x)$,
b) $\neg(a = 0) \wedge \neg(a = 1) \wedge \forall x \forall y[(a = x \cdot y) \rightarrow ((x = 1) \vee (y = 1))]$.

2. Kokie termai yra formulėje $\exists x(a = 2 \cdot x)$?

Atsakymas: yra tik du termai: *a*, 2.
Reiškinys $2 \cdot x$ *nėra termas*. Kai kuriuose vadovėliuose laikoma, kad termuose gali būti ir suvaržytieji kintamieji, kai kuriuose - kintamieji gali būti tik laisvieji. Mes prisilaikome pastarojo reikalavimo.

O kaip tada gavome formulę $\exists x(a = 2 \cdot x)$? Ją gavome prisilaikydami žingsnių, aprašytų formulės apibrėžime:

$a = 2 \cdot b$ pagal 1-ą žingsnį (= yra predikatas, o *a*, $2 \cdot b$ - termai),
$\exists x(a = 2 \cdot x)$ pagal 3-ią žingsnį (*b* - laisvasis kintamasis, *x* - suvaržytasis kintamasis, neįeinantis į formulę $a = 2 \cdot b$).

Priminsime, kad *formulė vadinama* ***įvykdoma****, jei yra struktūra, kurioje ji teisinga. T.y., jei ji turi* ***modelį****.*

Užduotis. Ar įvykdoma formulė $\forall x \exists y(P(x, y) \wedge P(x, f(x)))$ struktūroje, kurioje $D = \{2, 3\}$ yra individinių konstantų aibė. Duota vieno argumento funkcija *f* ($f(2) = 3; f(3) = 2$) bei dvivietis predikatas *P*, nusakomas lentele:

a	*b*	$P(a, b)$
2	2	*k*
2	3	*t*
3	2	*t*
3	3	*k*

Atsakymas: *taip, įvykdoma* (struktūra yra modelis), nes, jei:
a) $x = 2$, tai *y* pasirenkam lygų 3, $f(2) = 3$. Su šiomis reikšmėmis formulė teisinga.
b) $x = 3$, tai *y* pasirenkam lygų 2, $f(3) = 2$. Su šiomis reikšmėmis formulė teisinga.

Formali aksiominė lygybės predikato teorija.
Signatūra:

- konstantos (apibrėžimo aibė *D*): bet kuri netuščia baigtinė ar begalinė aibė.

- predikatai: =

- *Aksiomos*:
  1. $\forall x(x = x)$,
  2. $\forall x \forall y(x = y \to y = x)$,
  3. $\forall x \forall y \forall z((x = y \wedge y = z) \to x = z)$.

Šios teorijos formulėse yra tik predikatas =. Formulė *G* išvedama teorijoje, jei išvedama sekvencija $A_1, A_2, A_3 \vdash G$. Čia $A_i (i = 1, 2, 3)$ yra trys teorijos aksiomos.

Matematinės teorijos aprašymas formalios aksiominės teorijos pavidalu suteikia kitokias išvedimo paieškos galimybes. Formalus išvedimas grynai *sintaksinis*. Neatsižvelgiama į išvedime gaunamų formulių prasmę. Todėl išvedimų paieškai patogu naudoti kompiuterius. Naudojant loginį išvedimą pereinama nuo tesingų išvadų prie teisingų prielaidų **nesinaudojant intuicija, patirtimi ar kitais išoriniais faktoriais**.

Pavyzdys. Parodysime, kad formulė $\forall x \exists y \forall z \exists u(u = z \wedge (\neg(x = z) \vee x = y))$ išvedama teorijoje. Iš aksiomų, šios formulės išvedimui, mums tereikės tik pirmosios.
Raide *F* išvedime žymime formulę $\exists y \forall z \exists u(u = z \wedge (\neg(a = z) \vee a = y))$,
o raide *G* - formulę $\exists u(u = b \wedge (\neg(a = b) \vee a = a))$.

$$\dfrac{\dfrac{\dfrac{}{b = b, \forall x(x = x) \vdash b = b, F, G}\oplus}{\forall \mathbf{x}(\mathbf{x} = \mathbf{x}) \vdash b = b, F, G} \quad \dfrac{\dfrac{\dfrac{}{a = a, \forall x(x = x) \vdash \neg(a = b), a = a, F, G}\oplus}{\forall \mathbf{x}(\mathbf{x} = \mathbf{x}) \vdash \neg(a = b), a = a, F, G}}{\forall x(x = x) \vdash \neg(\mathbf{a} = \mathbf{b}) \vee \mathbf{a} = \mathbf{a}, F, G}}{\forall x(x = x) \vdash (\mathbf{b} = \mathbf{b} \wedge (\neg(\mathbf{a} = \mathbf{b}) \vee \mathbf{a} = \mathbf{a})), F, G}$$
$$\dfrac{\forall x(x = x) \vdash (\mathbf{b} = \mathbf{b} \wedge (\neg(\mathbf{a} = \mathbf{b}) \vee \mathbf{a} = \mathbf{a})), F, G}{\forall x(x = x) \vdash \exists \mathbf{u}(\mathbf{u} = \mathbf{b} \wedge (\neg(\mathbf{a} = \mathbf{b}) \vee \mathbf{a} = \mathbf{a})), F}$$
$$\dfrac{\forall x(x = x) \vdash \exists \mathbf{u}(\mathbf{u} = \mathbf{b} \wedge (\neg(\mathbf{a} = \mathbf{b}) \vee \mathbf{a} = \mathbf{a})), F}{\forall x(x = x) \vdash \forall \mathbf{z} \exists \mathbf{u}(\mathbf{u} = \mathbf{z} \wedge (\neg(\mathbf{a} = \mathbf{z}) \vee \mathbf{a} = \mathbf{a})), F}$$
$$\dfrac{\forall x(x = x) \vdash \forall \mathbf{z} \exists \mathbf{u}(\mathbf{u} = \mathbf{z} \wedge (\neg(\mathbf{a} = \mathbf{z}) \vee \mathbf{a} = \mathbf{a})), F}{\forall x(x = x) \vdash \exists \mathbf{y} \forall \mathbf{z} \exists \mathbf{u}(\mathbf{u} = \mathbf{z} \wedge (\neg(\mathbf{a} = \mathbf{z}) \vee \mathbf{a} = \mathbf{y}))}$$
$$\dfrac{\forall x(x = x) \vdash \exists \mathbf{y} \forall \mathbf{z} \exists \mathbf{u}(\mathbf{u} = \mathbf{z} \wedge (\neg(\mathbf{a} = \mathbf{z}) \vee \mathbf{a} = \mathbf{y}))}{\forall x(x = x) \vdash \forall \mathbf{x} \exists \mathbf{y} \forall \mathbf{z} \exists \mathbf{u}(\mathbf{u} = \mathbf{z} \wedge (\neg(\mathbf{x} = \mathbf{z}) \vee \mathbf{x} = \mathbf{y}))}$$

Teorijoje įrodomos tik tokios lygybės savybės, kurios nepriklauso nuo aibės *D*.

Formali aksiominė teorija vadinama pilna, jei, kokia bebūtų teorijos **uždara formulė** *F*, įrodoma arba *F*, arba $\neg F$.

Formali aksiominė lygybės predikato teorija **nėra pilna**. Pavyzdžiui, uždara formulė $\forall x \forall y(x = y)$ nėra įrodoma, nes struktūroje iš dviejų elementų (kai aibėje *D* yra du skirtingi elementai) ji klaidinga. Jos neigimas $\exists x \exists y \neg(x = y)$ yra klaidingas aibėje iš vieno elemento. Teorijoje įrodomos tik formulės teisingos bet kurioje struktūroje (tapačiai teisingos formulės). Matematikoje retai nagrinėjamos teorijos yra *pilnos*. Todėl kelia nuostabą, kai daugelis matematikų stebisi, kad formali Peano aritmetika yra nepilna.

Jei teorija pilna ir efektyviai aksiomatizuojama, tai ji išsprendžiama. Jei Peano aritmetika būtų pilna, tai egzistuotų algoritmas, kuriuo galėtumėm patikrinti bet kurią uždarą aritmetikos formulę, ar ji išvedama. *Teorija T efektyviai aksiomatizuojama, jei egzistuoja algoritmas, kuriuo atpažistamos teorijos T aksiomos*.

Papildysime predikatų logikos skaičiavimą nauja aksioma ir dvejomis taisyklėmis.

Aksioma: $\Gamma \vdash \Delta, t = t$. Čia *t* (pirmoji žodžio *termas* raidė) yra kuris nors termas.

Taisyklės lygybės predikatui:

$$\frac{t = r, \Gamma^{t=r} \vdash \Delta^{t=r}}{t = r, \Gamma \vdash \Delta} \qquad \frac{t = r, \Gamma^{r=t} \vdash \Delta^{r=t}}{t = r, \Gamma \vdash \Delta}$$

Pirmoji taisyklė tvirtina, kad, jei turim antecedente lygybę $t = r$, tai visose sekvencijos formulėse termą *t* keičiame jam lygiu *r*, o antroji - visose sekvencijos formulėse keičiame *r* jam lygiu *t*.

Gautasis skaičiavimas vadinamas **pirmos eilės predikatų logikos sekvenciniu skaičiavimu su lygybės predikatu**. Formaliose teorijoje dažniausiai naudojamas skaičiavimas su lygybės predikatu.

Pavyzdys. Sekvencijos $\vdash \forall x \forall y \forall u \forall v((u = y \wedge u = v \wedge F(x, y)) \rightarrow F(u, v))$ išvedimas skaičiavime su lygybės predikatu. Čia *F* bet kurios formalios teorijos formulė.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\oplus}{a = b, c = d, F(b, d) \vdash F(b, d)}}{a = b, \mathbf{c} = \mathbf{d}, F(b, c) \vdash F(b, d)}}{\mathbf{a} = \mathbf{b}, c = d, F(a, c) \vdash F(b, d)}}{a = b \wedge c = d \wedge F(a, c) \vdash F(b, d)}}{\vdash (a = b \wedge c = d \wedge F(a, c)) \rightarrow F(b, d)}}{\vdash \forall x \forall y \forall u \forall v((u = y \wedge u = v \wedge F(x, y)) \rightarrow F(u, v)}$$

Formali aksiominė griežtos tvarkos teorija.
Signatūra:

- konstantos (apibrėžimo aibė*D*): bet kuri netuščia baigtinė ar begalinė aibė.
- predikatai: <,
- funkcijos: nėra
- *Aksiomos*:
  1. $\forall x \neg(x < x)$,
  2. $\forall x \forall y \forall z((x < y \wedge y < z) \rightarrow x < z)$.

Pavyzdys. Parašysime sekvencijos $\forall x \forall y(x < y \rightarrow \neg(y < x))$. išvedimą. Taikydami $(\vdash \forall)$ taisyklę, formulės nekartosime, nes jos nebereiks (t.y. taikome ir silpninimo taisyklę).

$$\dfrac{\dfrac{\dfrac{\oplus}{a<b,b<a\vdash a<a,a<b} \qquad \dfrac{\oplus}{a<b,b<a\vdash a<a,b<a}}{a<b,b<a\vdash a<a,\mathbf{a}<\mathbf{b}\wedge\mathbf{b}<\mathbf{a}} \qquad \dfrac{\oplus}{a<a,a<b,b<a\vdash a<a}}{(\mathbf{a}<\mathbf{b}\wedge\mathbf{b}<\mathbf{a})\to\mathbf{a}<\mathbf{a},a<b,b<a\vdash a<a}$$

$$\begin{array}{c}
(\mathbf{a}<\mathbf{b}\wedge\mathbf{b}<\mathbf{a})\to\mathbf{a}<\mathbf{a},a<b,b<a\vdash a<a \\ \hline
\neg(\mathbf{a}<\mathbf{a}),(a<b\wedge b<a)\to a<a,a<b,b<a\vdash \\ \hline
\neg(a<a),\forall\mathbf{z}((\mathbf{a}<\mathbf{b}\wedge\mathbf{b}<\mathbf{z})\to\mathbf{a}<\mathbf{z}),a<b,b<a\vdash \\ \hline
\neg(a<a),\forall\mathbf{y}\forall\mathbf{z}((\mathbf{a}<\mathbf{y}\wedge\mathbf{y}<\mathbf{z})\to\mathbf{a}<\mathbf{z}),a<b,b<a\vdash \\ \hline
\neg(a<a),\forall\mathbf{x}\forall\mathbf{y}\forall\mathbf{z}((\mathbf{x}<\mathbf{y}\wedge\mathbf{y}<\mathbf{z})\to\mathbf{x}<\mathbf{z}),a<b,b<a\vdash \\ \hline
\forall\mathbf{x}\neg(\mathbf{x}<\mathbf{x}),\forall x\forall y\forall z((x<y\wedge y<z)\to x<z),a<b,b<a\vdash \\ \hline
\forall x\neg(x<x),\forall x\forall y\forall z((x<y\wedge y<z)\to x<z),a<b\vdash\neg(\mathbf{b}<\mathbf{a}) \\ \hline
\forall x\neg(x<x),\forall x\forall y\forall z((x<y\wedge y<z)\to x<z)\vdash\mathbf{a}<\mathbf{b}\to\neg(\mathbf{b}<\mathbf{a}) \\ \hline
\forall x\neg(x<x),\forall x\forall y\forall z((x<y\wedge y<z)\to x<z)\vdash\forall\mathbf{x}\forall\mathbf{y}(\mathbf{x}<\mathbf{y}\to\neg(\mathbf{y}<\mathbf{x}))
\end{array}$$

Teorija **nepilna**. Teorijoje neįrodoma formulė $\exists x\exists y(x<y)$, nes egzistuoja aibės su apibrėžta griežta daline tvarka, kurioje visi elementai nepalyginami. Neįrodomas ir formulės neigimas $\neg\exists x\exists y(x<y)\equiv\forall x\forall y\neg(x<y)$, nes egzistuoja aibės su griežta daline tvarka, kurioje yra bent du palyginami elementai.

Sekvencinis akičiavimas su funkciniais simboliais.

Aksiomos: $F,\Gamma\vdash\Delta,F$

Loginių operacijų taisyklės tos pačios kaip ir visuose sekvenciniuose skaičiavimuose.

Kvantorinės taisyklės:

$$(\exists\vdash)\quad\frac{F(\beta),\Gamma\vdash\Delta}{\exists xF(x),\Gamma\vdash\Delta}\qquad\qquad(\vdash\exists)\quad\frac{\Gamma\vdash\Delta,F(\gamma),\exists xF(x)}{\Gamma\vdash\Delta,\exists xF(x)}$$

$$(\forall\vdash)\quad\frac{\forall xF(x),F(\gamma),\Gamma\vdash\Delta}{\forall xF(x),\Gamma\vdash\Delta}\qquad\qquad(\vdash\forall)\quad\frac{\Gamma\vdash\Delta,F(\beta)}{\Gamma\vdash\Delta,\forall xF(x)}$$

Raide *C* žymime apatinės sekvencijos visų *termų* aibę. **Neužmirškite**, kad, sutinkamai su knygoje vartojamu termo apibrėžimu, termuose negali būti suvaržytųjų kintamųjų įeičių. *K* - teorijos konstantų aibė (kai išvedimo paieška vykdoma formalioje aksiominėje teorijoje).

$(\exists\vdash)$ ir $(\vdash\forall)$ taisyklėse $\beta\notin C$. β yra *naujas* laisvasis kintamasis.
Taisyklėse $(\vdash\exists)$ ir $(\forall\vdash)$ $\gamma\in K\cup C$. Jei *C* tuščia aibė, tai $\gamma\in K$. Jei $K\cup C$ tuščia aibė, tai γ yra naujas laisvasis kintamasis.

Pavyzdys.

Įrodysime sekvenciją (vis tenka priminti, kad įrodymas *iš apačios į viršų*)
$f(a)=g(b),b=g(a),\forall x(P(f(x))\wedge Q(g(x)))\vdash\exists x(P(g(x))\wedge Q(x))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{f(a)=g(b),b=g(a),P(g(b)),Q(g(a))\vdash P(g(b))}\oplus}{\mathbf{f(a)}=\mathbf{g(b)},b=g(a),P(f(a)),Q(g(a))\vdash P(g(b))}}{f(a)=g(b),b=g(a),P(f(a)),Q(g(a))\vdash \mathbf{P(g(b))}\wedge\mathbf{Q(b)}}}{f(a)=g(b),b=g(a),P(f(a)),Q(g(a))\vdash \exists\mathbf{x}(\mathbf{P(g(x))}\wedge\mathbf{Q(x)})}}{f(a)=g(b),b=g(a),\mathbf{P(f(a))}\wedge\mathbf{Q(g(a))}\vdash \exists x(P(g(x))\wedge Q(x))}}{f(a)=g(b),b=g(a),\forall\mathbf{x}(\mathbf{P(f(x))}\wedge\mathbf{Q(g(x))})\vdash \exists x(P(g(x))\wedge Q(x))}$$

$$\dfrac{\dfrac{\dfrac{}{f(a)=g(b),b=g(a),P(f(a)),Q(b)\vdash Q(b)}\oplus}{f(a)=g(b),\mathbf{b}=\mathbf{g(a)},P(f(a)),Q(g(a))\vdash Q(b)}}{f(a)=g(b),b=g(a),P(f(a)),Q(g(a))\vdash \mathbf{P(g(b))}\wedge\mathbf{Q(b)}}$$

3.2 Gödelio numeriai

Jau rašėme, kad matematikoje naudojama sąvoka *funkcija*, o logikoje galima apsieiti ir be funkcijų. Praplėsdami logikos kalbą funkcijomis padarome ją labiau įprasta ir suprantamesne matematikams.

Struktūra susideda iš predikatų ir jų apibrėpimo aibės *D*. Nagrinėsime ir funkcijas, kurių apibrėžimo aibėmis yra ta pati aibė *D*. O reikšmių aibėmis - *D* poaibiai.
Naudojant funkcinius simbolius bei termus, matematikos teiginius galima užrašyti įprastesne matematikams kalba.

Pavyzdžiai:

Struktūra: $D = R$. *Predikatai*: $<, \leq, >$. *Funkcijos*: $-, |a|, f$.

Užduotis. Parašyti predikatų logikos formule teiginius (kintamuosius rašyti įprastais matematikoje simboliais):

1. Funkcijos $f(x)$ tolydumo taške x_0 apibrėžimas.

Atsakymas: $\forall\epsilon\exists\delta\forall x((\epsilon > 0 \wedge \delta > 0 \wedge |x - x_0| < \delta) \rightarrow |f(x) - f(x_0)| < \epsilon)$.

2. Taškas x_0 yra funkcijos $f(x)$ lokalus minimumas.

Atsakymas: $\exists\delta\forall x((\delta > 0 \wedge |x - x_0| < \delta) \rightarrow f(x) \leq f(x_0))$.

Jei aibės *D* poaibis nusakomas kuriuo nors vienviečiu predikatu, pavyzdžiui $N(a)$, teisingas tada ir tiktai tada, kai *a* natūralusis skaičius, tai vietoje $N(a)$ galima rašyti $a \in N$. Suprantama, jei struktūroje yra predikatas $\in$.

Struktūra. *D*: *R* ir jos poaibis *N*. *Predikatai*: $<, \leq, >, \in$. *Funkcijos*: $-, |a|$.

3. Skaitinės sekos $a_1, a_2, a_3, \ldots$ riba yra skaičius *a.*

Atsakymas: $\forall\epsilon\exists n_0\forall n((\epsilon > 0 \wedge n_0 \in N \wedge n \in N \wedge n > n_0) \rightarrow |a_0 - a| < \epsilon)$.

4. Koši kriterijus. Skaitinės sekos $a_1, a_2, a_3, \ldots$ riba yra skaičius *a.*

Atsakymas: $\forall\epsilon\exists n_0\forall n\forall m((\epsilon > 0 \wedge n_0 \in N \wedge n \in N \wedge m \in N \wedge n > n_0 \wedge m > n_0) \rightarrow |a_0 - a_m| < \epsilon)$.

Predikatų logikos formulių aibė yra skaičioji. Skaičiosiomis yra ir jos poaibiai. Tame tarpe, ir tapačiai teisingų, tapačiai klaidingų bei įvykdomų formulių aibės. Formulėms galima priskirti numerius (natūraliuosius skaičius) ir nagrinėti ne formulių aibes, o natūraliųjų skaičių poaibius. Pirmasis tuo pasinaudojo K.Gödelis, nagrinėdamas formaliąją aritmetiką. Kiekvienam aritmetikos formulės simboliui *e* priskiriamas numeris (natūralusis skaičius) *g*(*e*). Formulei susidedančiai iš *n* simbolių $e_1e_2e_3\ldots e_n$ priskiriamas skaičius $2^{e_1} \cdot 3^{e_2} \cdot 5^{e_3} \cdot \ldots \cdot p_n^{e_n}$. Čia p_n yra *n*-tasis pirminis skaičius. Kiekvienai formulei priskiriamas vienintelis skaičius. Skirtingos formulės turi skirtingus numerius. Be to, pagal natūralųjį skaičių galima nustatyti, ar jis yra kurios nors formulės numeris. Jei skaičius yra kurios nors formulės numeris, tai mokame vienareikšmiškai parašyti tą formulę. Toks formulių numeravimo metodas vadinamas Gödelio numeravimu, o skaičiai, priskirti formulėms, vadinami *Gödelio numeriais.*

Pavyzdžiui, parašysime aritmetikos formulių kodavimą, kai jos užrašomos naudojant simbolius:

- konstantas: 0, 1
- funkcijas: $+, \cdot$
- predikatą: =
- logines operacijas: $\neg, \wedge, \vee, \rightarrow$
- kvantorius: $\forall, \exists$
- skliaustus: (,)
- individinius kintamuosius $x_0, x_1, x_2, \ldots$

Kiekvienam simboliui priskiriamas *Gödelio numeris*:
$g(0) = 3, g(1) = 5, g(+) = 7, g(\cdot) = 9, g(=) = 11, g(\neg) = 13, g(\wedge) = 15, g(\vee) = 17,$
$g(\rightarrow) = 19, g(\forall) = 21, g(\exists) = 23, g(() = 25, g()) = 27, g(x_i) = 29 + 2i (i = 0, 1, 2, \ldots)$

Tuomet formulei $\exists x_0(x_0 + 0 = 1)$ priskiriamas numeris

$2^{23} \cdot 3^{29} \cdot 5^{25} \cdot 7^{29} \cdot 11^7 \cdot 13^3 \cdot 17^{11} \cdot 19^5 \cdot 23^{27}$.

Grynosios logikos su lygybės predikatu ir funkciniais simboliais formulių užtašymui pakanka baigtinio skaičiaus simbolių:

- predikatas =
- loginės operacijos $\neg, \wedge, \vee, \rightarrow$
- kvantoriai $\forall, \exists$
- skliaustai: (,)
- kablelis ,
- predikatinis kintamasis *P*
- funkcinis simbolis *f*
- laisvasis kintamasis *a*
- suvaržytas kintamasis *x*
- abėcėlė kintamųjų indeksams $0, 1, 2, \ldots, 9$

Kiekviena formulė yra žodis baigtinėje abėcėlėje $\{=, \neg, \wedge, \ldots, 9\}$. Pavyzdžiui, į formulę $\neg P_{17}(a_{13}, f_{21}(a_5))$ žiūrime kaip į žodį $\neg 17(a13, f21(a5))$ duotoje baigtinėje abėcėlėje (joje yra 24 simboliai). *n* raidžių žodžiui priskirsime skaičių $2^{e_1} \cdot 3^{e_2} \cdot 5^{e_3} \cdot \ldots \cdot p_n^{e_n}$. Čia p_n yra *n*-tasis pirminis skaičius, e_i yra *i*-tosios raidės žodyje numeris, t.y. raidės vieta (iš kairės į dešinę) baigtinėje abėcėlėje $(1 \leq e_i \leq 24)$.

Vietoje kurio nors formulių poaibio, atsirado galimybė, nagrinėti poaibio formulių numerių aibę, t.y. natūraliųjų skaičių poaibį.

Nagrinėjame funkcijas, kurių apibrėžimo aibė yra *N*, o reikšmių - kuris nors *N* poaibis. Tokių funkcijų yra kontinuumas.

Apibrėžimas. Funkcija $f(a)$ vadinama **apskaičiuojama**, jei ją galima suprogramuoti.

Apskaičiuojamų funkcijų aibė yra **skaičioji**, kadangi programa, bet kurioje programavimo kalboje, yra baigtinis žodis tos programavimo kalbos abėcėlėje.

Vadinasi, **tik dalis funkcijų**, nagrinėjamų matematikoje, yra apskaičiuojamos.

Pažymėkime raide *T* visų *tapačiai teisingų* predikatų logikos formulių *Gödelio numerių aibę*; raide *K* - visų *tapačiai klaidingų* formulių *Gödelio numerių aibę*; raide *I* - visų *įvykdomų* formulių *Gödelio numerių aibę*.

Įrodyta, kad, nei funkcija $f_T(n)$

$$f_T(n) = \begin{cases} 1, & \text{jei } n \in T \\ 0, & \text{priešingu atveju} \end{cases}$$

nei analogiškos funkcijos $f_K(n)$, $f_I(n)$ **nėra apskaičiuojamos**. Tai reiškia, kad neįmanoma jų suprogramuoti, t.y. tarp visų galimų programų nėra tokių, kuriomis apskaičiuojamos $f_T(n)$, $f_K(n)$, $f_I(n)$.

Matematikoje įrodyta, kad bet kuris begalinis natūraliųjų skaičių poaibis yra skaitusis. Tai reiškia, kad tarp natūraliųjų skaičių aibės ir aibių *T, K, I* yra bijekcijos (abipusiai vienareikšmės atitiktys) $f : N \to T$, $g : N \to K$, $h : N \to I$. Klausimas tik, ar funkcijos (bijekcijos) *f, g, h* apskaičiuojamos. Nė viena iš jų nėra apskaičiuojama

O kaip su funkcija

$$F_T(n) = \begin{cases} 1, & \text{jei } n \in T \\ \infty, & \text{priešingu atveju} \end{cases}$$

bei analogiškomis $F_K(n)$, $F_I(n)$? **Įrodyta**, kad funkcijos $f_T(n)$, $F_K(n)$ **apskaičiuojamos**, o funkcija $F_I(n)$ **nėra apskaičiuojama**.

Remiantis šiais rezultatais, sukurti **predikatų logikos skaičiavimai**, kuriuose įrodomos **tapačiai teisingos formulės** ir tiktai jos. Bet nėra tokio predikatų skaičiavimo, kuriame būtų įrodomos *įvykdomos formulės* ir tiktai jos.

Tie patys rezultatai galioja ir predikatų logikai su lygybės predikatu bei su funkciniais simboliais.

Jei manoma, kad formulė nėra tapačiai teisinga, tai dažniausiai ieškoma struktūra, kurioje formulė klaidinga.

Pavyzdys. Rasti struktūrą, kurioje formulė $\forall x \exists y(P(y, f(x) \wedge P(y, f(f(x))))$ klaidinga.

Atsakymas. $D = N$. Predikatą $P(a, b)$ keičiame lygybe, o funkcinį simbolį $f(a)$ - funkcija $a + 1$.
$\forall x \exists y(y = x + 1 \wedge y = (x + 1) + 1)$.

Pagrindinės pirmos eilės grynosios logikos savybės:

1. Sekvencija $\vdash F$ išvedama tada ir tiktai tada, kai *F* tapačiai teisinga formulė.

2. *Kompaktiškumas.* Formulių aibė *M* įvykdoma tada ir tiktai tada, kai įvykdomas kiekvienas jos baigtinis poaibis.

3. *Löwenheimo-Skolemo teorema.* Jei formulė įvykdoma, tai ji įvykdoma ir numeruojamoje aibėje (baigtinėje arba skaičiojoje).

3.3 Elementarioji geometrija

Pirmasis geometrijos aksiomas aprašė D.Hilbertas 1899 metais. Buvo 40 aksiomų. Vėliau 1959 metais A.Tarskis su 12 aksiomų aprašė geometrijos dalį, kurioje nenaudojama aibių teorija ir kurios aksiomos aprašomos pirmos eilės logikos su lygybės predikatu formulėmis. Pavadino *elementariąja geometrija*. 1983 metais išleista knyga, kurioje aprašyta elementari geometrija su 11 aksiomų. Tai yra ilgo A.Tarskio darbo, pradėto 1926 metais su geometrijos aksiomatizavimu, rezultatas. Gavosi graži pirmos eilės logikos su lygybės predikatu teorija aprašanti dalį geometrijos.

Be to, A.Tarskis įrodė:

- *Elementarioji geometrija yra pilna*. Kokia bebūtų uždara formulė parašyta elementarios geometrijos signatūroje, arba ji, arba jos neigimas įrodomas elementarios geometrijos teorijoje.
- *Elementarioji geometrija išsprendžiama*. Egzistuoja algoritmas, kuriuo, kokia bebūtų uždara formulė, parašyta elementarios geometrijos signatūroje, nustatoma ar formulė įvykdoma.

Nagrinėjamoje teorijoje predikatų apibrėžimo aibė - plokštumos taškų aibė. Yra du predikatai: 1) simboliu $\cong$ nusakome keturvietį predikatą. Formule $ab \cong cd$ tvirtiname, kad atkarpos *ab* ir *cd* yra vienodo ilgio. 2) Triviečiu predikatu $B(a, b, c)$ teigiama, kad taškas *b* yra tarp taškų *a, c* tiesėje, einančioje per taškus *a, c*.

Dar nuo Euklido laikų matematikai pateikia vis kitokius pradinių matematikos sąvokų *taškas, tiesė,* apibrėžimus.

Bet niekas nesugalvojo geresnio būdo, kaip pateikti apibrėžimus aksiomų pavidalu. Toks sąvokų apibrėžimo būdas yra tikslus.

Elementarios geometrijos signatūra:

- konstantos (apibrėžimo aibė): R^2
- predikatai: $\cong, B$
- funkcijos: *nėra*

Elemntarios geometrijos aksiomos:

Aksiomose nėra laisvųjų kintamųjų. Paprastumo dėlei kai kada **bendrumo** kvantorių nerašome. Vietoje *B*(*a, b, c*) *rašysime, praleisdami skliaustus* (taip rašė A.Tarskis), *Babc*.

1. $ab \cong ba$

2. $ab \cong cc \to a = b$

3. $(ab \cong cd \wedge ab \cong rs) \rightarrow cd \cong rs$
Trečia aksioma aprašome predikato $\cong$ tranzityvumą.

4. $Baba \rightarrow a = b$

5. $\exists x(Bdax \wedge ax \cong bc)$
Tiesėje *da* galima nubrėžti atkarpą *bc*, priešinga kryptimi, negu kad iš *a* patenkame į *d*.

6. $(Badc \wedge Bebc) \rightarrow \exists x(Bdxe \wedge Bbxa)$
Dvi iškylaus keturkampio *adbe* įstrižainės susikerta viename taške.

7. $(\neg(a = b) \ \wedge \ Babc \ \wedge \ Ba_1b_1c_1 \ \wedge \ ab \cong a_1b_1 \ \wedge \ bc \cong b_1c_1 \ \wedge \ ad \cong a_1d_1 \ \wedge \ bd \cong b_1d_1) \rightarrow cd \cong c_1d_1$
Ar galima kalbėti apie kampą, neturint sąvokos *kampas* predikato? Pasirodo, kai kuriais atvejais, galima. Turim keturias atkarpas *ab, bc, ad, bd* ir brėžinį, kuriame yra joms lygios atkarpos $a_1b_1, b_1c_1, a_1d_1, b_1d_1$ tuomet išplaukia, kad $cd \cong c_1d_1$.

8. $\exists x\exists y\exists z(\neg Bxyz \wedge \neg Byzx \wedge \neg Bzxy)$
Egzistuoja trys taškai nepriklausantys vienai ir tai pačiai tiesei. Šia aksioma nustatome, kad nagrinėjamų taškų aibė nėra vienos tiesės taškų aibė.

9. $\exists x\exists y\exists z((\neg(d = e) \ \wedge \ ad \cong ae \ \wedge \ bd \cong be \ \wedge \ cd \cong ce) \rightarrow (Bxyz \vee Byzx \vee Bzxy))$
Trys taškai *a, b, c,* vienodai nutolę nuo taškų *d, e* yra vienoje tiesėje. Šia aksioma tvirtinama, kad nagrinėjamų taškų aibė nėra daugiamatės (tame tarpe, trimatės) erdvės taškų aibė.

10. $(Bade \wedge Bbdc \wedge \neg(d = e)) \rightarrow \exists x\exists y(Babx \ \wedge \ Bacy \ \wedge \ Byex)$.

Euklido aksioma.

11. Bet kurioms formulėms elementarios geometrijos signatūroje $C(a)$ ir $D(b)$ su laisvaisiais kintamaisiais *a, b*, teisinga formulė
$\exists u\forall x\forall y((C(x) \wedge D(y)) \rightarrow Buxy) \rightarrow \exists v\forall x\forall y((C(x) \wedge D(y)) \rightarrow Bxvy)$.
Jei taškai, su kuriais *C* teisinga yra tarp fiksuoto *u* ir bet kurio taško, su kuriuo *D* teisinga, tai atsiras taškas *v* atskiriantis *C* taškus nuo *D* taškų.

Geometrijoje yra daug pirminių (neapibrėžiamų) sąvokų. Priimta matematikoje jas tik paaiškinti, kad suprastumėt, intuityviai. Elementarioje geometrijoje pradinėmis sąvokomis yra **taškas, tiesė, atkarpa**. Enciklopedijose įprasta lyg tai pateikti pradinių sąvokų apibrėžimus, bet jie tik daugiau sukelia neaiškumų. *Taškas*: *dydžio neturintis geometrijos objektas, kurį apibūdina jo padėtis*. Bet juk atkarpa [3;5] sudaryta iš taškų (dydžio neturinčių geometrijos objektų) ir turi dydį, jos ilgis 2. Atkarpos [3;5] ir [6;9] yra skirtingų ilgių, nors tarp jas sudarančių taškų yra bijekcija.

Kaip išsisuko A.Tarskis geometrijoje su pradinėmis sąvokomis? *Taškas* yra raidė. jos žymimos *a, b, c,...* arba raide su indeksu.

Atkarpa yra dvi raidės. *Tiesė* - trejetas taškų (kurių kai kurie, arba visi, gali ir sutapti), pažymėtų raide *B*. Nieko geresnio ir negalima sugalvoti kaip apibrėžti pradines sąvokas (mūsų

atveju atkarpa, tiesė) aksiomomis. 11 aksiomų ir yra *atkarpų* ir *tiesių* tikslus apibrėžimas. Atkarpai nepriskiriamas skaičius, jo ilgis. Bet jas galima palyginti. Nustatyti kuri ilgesnė. Elementariąją geometriją sudaro visi teiginiai, užrašyti formulių pavidalu, kurie išvedami naudojant pirmos eilės logiką su lygybės predikatu iš 11 aksiomų. Ši teorija išsprendžiama, yra parašytos programos kompiuteriui, kurios į bet kokią užklausą, parašytą formulės pavidalu, atsakys ar ji teisinga (ir pateiks įrodymą), ar ne. Toks formalus aprašymas kainuoja. Tenka netgi ir akivaizdžius teiginius įrodyti, nes joje teisinga tiktai tai, kas yra įrodyta.

Įrodysime, kad $aa \cong bb$, t.y. įrodysime, kad visi taškai turi vienodą ilgį. Panaudosime 2-ą ir 5-tą aksiomas.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\oplus}{Bdae, aa \cong bb \vdash aa = bb, ae \cong bb} \qquad \dfrac{\dfrac{\oplus}{a = e, Bdae, aa \cong bb \vdash aa = bb}}{\mathbf{a} = \mathbf{e}, Bdae, ae \cong bb \vdash aa = bb}}{F\mathbf{ae} \cong \mathbf{bb} \to \mathbf{a} = \mathbf{e}, Bdae, ae \cong bb \vdash aa = bb}}{ae \cong bb \to a = e, \mathbf{Bdae} \wedge \mathbf{ae} \cong \mathbf{bb}) \vdash aa = bb}}{\forall \mathbf{x} \forall \mathbf{y} \forall \mathbf{z}(\mathbf{xy} \cong \mathbf{zz} \to \mathbf{x} = \mathbf{y}), Bdae \wedge ae \cong bb) \vdash aa = bb}}{\forall x \forall y \forall z(xy \cong zz \to x = y), \exists \mathbf{v}(\mathbf{Bdav} \wedge \mathbf{av} \cong \mathbf{bb}) \vdash aa = bb}}{\forall x \forall y \forall z(xy \cong zz \to x = y), \forall \mathbf{x} \forall \mathbf{y} \forall \mathbf{z} \forall \mathbf{u} \exists \mathbf{v}(\mathbf{Buxv} \wedge \mathbf{xv} \cong \mathbf{yz}) \vdash aa = bb}}{\forall x \forall y \forall z(xy \cong zz \to x = y), \forall x \forall y \forall z \forall u \exists v(Buxv \wedge xv \cong yz) \vdash \forall \mathbf{x} \forall \mathbf{y}(\mathbf{xx} = \mathbf{yy})}$$

Kai kurios įrodytos teoremos (jomis, per pratybas, galima naudotis, kaip duotomis):
1. $Babc \to Bcba$,
2. $(Babd \wedge Bbcd) \to Babc$,
3. $(Babc \wedge Badc) \to (Babd \vee Badb)$,
4. $(\neg(a = b) \wedge Babc \wedge Babd) \to (Bacd \vee Badc)$.

Ketvirtoji formulė buvo laikoma aksioma. Bet pavyko ją įrodyti ir sumažinti aksiomų skaičių. Jos įrodymas ganėtinai sudėtingas.

Kiti naudojami elementarioje geometrijoje predikatai yra išvestiniai. Juos galima užrašyti logikos formulėmis su pradiniais $\cong, B$ predikatais.

Pavyzdžiai:

- $ab \leq cd$ užrašomas formule $\exists x(ab \cong cx \wedge Baxd)$,
- Teiginys *taškas c priklauso tiesei nubrėžtai per taškus a, b, žymimas* $c \in ab$ ir užrašomas formule $Bcab \vee Bacb \vee Babc$,
- Teiginys *tiesė, einanti per taškus a, b, lygiagreti tiesei, einančiai per taškus c, d, žymimas* $ab \| cd$ ir užrašomas formule $\neg(a = b) \wedge \neg(c = d) \wedge \forall x \neg(x \in ab \wedge x \in cd)$.

3.4 Peano aritmetika

Matematinius teiginius norime paversti tiksliais matematiniais objektais. Tuo tikslu matematinėje logikoje kuriamos dirbtinės formalios kalbos. Aprašysime dar vieną jų - *formalios arit-*

metikos kalbą.

Italų matematikas G.Peano 1889 metais paskelbė aksiomų sistemą natūraliųjų skaičių aibei apibūdinti. Vėliau, 1900 metais, pataisė ją. Tiksliau, tik pirmąją aksiomą, kuria nulis dabar jau laikomas natūraliuoju skaičiumi.

1. Elementas, žymimas 0, vadinamas *nuliu* yra natūralusis skaičius.

2. Kiekvienam natūraliajam skaičiui *n* egzistuoja paskesnysis skaičius $s(n)$.

3. 0 nėra jokio natūralaus skaičiaus paskesniuoju.

4. Natūralieji skaičiai lygūs, jei jų paskesnieji skaičiai sutampa.

5. Jei *S* yra natūraliųjų skaičių savybė ir:
a) nulis tenkina savybę *S,*
b) jei natūralusis skaičius tenkina savybę *S*, ir taip pat ją tenkina jo paskesnysis skaičius,
tuomet visi natūralieji skaičiai tenkina savybę S.

1901 metais Dedekindas aprašė pusiau aksiominį aritmetikos (t.y. tik dalis aksiomų buvo aprašytos formulių pavidalu) variantą ir pavadino tą sistemą *Peano aritmetika*. Vėliau buvo paskelbti spaudoje keli aksiomų sistemų variantai, parašyti tik predikatų logikos kalba, ir visi jie buvo vadinami *Peano vardu*. Aprašysime vieną jų.

Peano aritmetikos (sutrumpintai PA) signatūra:

- konstantos: 0
- predikatai: $=$
- funkcijos: $s, +, \cdot$

Yra tik vienas dvivietis lygybės predikatas, trys funkcijos: $s(a) = a + 1, a + b$ (sudėtis), $a \cdot b$ (daugyba) bei viena konstanta 0 (nulis).

Aritmetikos aksiomos.

1. $\forall x \neg(s(x) = 0)$

2. $\forall x \forall y(s(x) = s(y) \rightarrow x = y)$

3. $\forall x(x + 0 = x)$

4. $\forall x \forall y(x + s(y) = s(x + y))$

5. $\forall x(x \cdot 0 = 0)$

6. $\forall x \forall y(x \cdot s(y) = (x \cdot y) + y)$

7. $(A(0) \wedge \forall x A(x) \to A(s(x)))) \to \forall x A(x)$

7-a aksioma yra *matematinės indukcijos* aksioma. Ją rašysime taisyklės pavidalu.

Indukcijos taisyklė:

$$\frac{\Gamma \vdash \Delta, A(0) \quad \Gamma \vdash \Delta, \forall x(A(x) \to A(s(x)))}{\Gamma \vdash \Delta, \forall x A(x)}$$

čia *A* - kuri nors PA formulė su laisvuoju kintamuoju *a*, formulėje $A(a)$.

Teiginys, užrašytas PA formule *F*, teisingas, jei išvedama sekvencija $P_1, P_2, \ldots, P_6 \vdash F$ predikatų skaičiavime su funkciniais simboliais ir lygybės predikatu bei indukcijos taisykle. $P_1, P_2, \ldots, P_6$ yra aritmetikos aksiomos.

Visų aksiomų nerašysime antecedente. Jei kurią nors naudosime, tai paryškintai parašysime ją, bei iš jos gautą formulę antecedente.

Išvedimų *pavyzdžiai*:

1. $\vdash \forall x(x = 0 \vee \exists y(x = s(y)))$.

$$\dfrac{\dfrac{\dfrac{}{\vdash 0 = 0, \exists y(0 = s(y))}\oplus}{\vdash \mathbf{0} = \mathbf{0} \vee \exists \mathbf{y}(\mathbf{0} = \mathbf{s}(\mathbf{y}))} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{a = 0 \vee \exists y(a = s(y)) \vdash s(a) = 0, s(a) = s(a)}\oplus}{a = 0 \vee \exists y(a = s(y)) \vdash s(a) = 0, \exists \mathbf{y}(\mathbf{s}(\mathbf{a}) = \mathbf{s}(\mathbf{y}))}}{a = 0 \vee \exists y(a = s(y)) \vdash \mathbf{s}(\mathbf{a}) = \mathbf{0} \vee \exists \mathbf{y}(\mathbf{s}(\mathbf{a}) = \mathbf{s}(\mathbf{y}))}}{\vdash (a = 0 \vee \exists y(a = s(y))) \to (s(a) = 0 \vee \exists y(s(a) = s(y)))}}{\vdash \forall x(x = 0 \vee \exists y(x = s(y)))(taikymas - indukcija)}$$

Robinsono aritmetika gaunama išbraukus iš PA matematinės indukcijos aksiomą ir prijungus naują aksiomą $\forall x(x = 0 \vee \exists y(x = s(y)))$. Kaip matome (pirmas pavyzdys), ji įrodoma PA formulė, bet neįrodoma Robinsono aritmetikoje.

2. $\forall x(0 + x = x)$.

$$\dfrac{\dfrac{\dfrac{}{\mathbf{aks}.(\mathbf{5}), \mathbf{0} + \mathbf{0} = \mathbf{0} \vdash 0 + 0 = 0}\oplus}{\vdash 0 + 0 = 0} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{0 + s(a) = s(a), 0 + a = a \vdash s(a) = s(a)}\oplus}{0 + s(a) = s(0 + a), \mathbf{0} + \mathbf{a} = \mathbf{a} \vdash 0 + s(a) = s(a)}}{\mathbf{aks}.(\mathbf{6}), \mathbf{0} + \mathbf{s}(\mathbf{a}) = \mathbf{s}(\mathbf{0} + \mathbf{a}), 0 + a = a \vdash 0 + s(a) = s(a)}}{0 + a = a \vdash 0 + s(a) = s(a)}}{\vdash 0 + a = a \to 0 + s(a) = s(a)}}{\forall x(0 + x = x)(taikymas - indukcija)}$$

3. $\vdash \forall x(0 + x = x + 0)$.

$$\dfrac{\dfrac{}{\vdash 0+0=0+0}\oplus \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{s(a)+0=s(a)\vdash s(a)=s(a)}\oplus}{\mathbf{aks.(5)},\mathbf{s(a)+0=s(a)}\vdash s(a)=s(a)+0}}{\vdash s(a)=s(a)+0}}{\mathbf{silpn.},\mathbf{0+a=a}\vdash s(0+a)=s(a)+0}}{\mathbf{aks.(5)},\mathbf{a+0=a}0+a=a+0\vdash s(0+a)=s(a)+0}}{\mathbf{silp.},\mathbf{0+s(a)=s(0+a)},0+a=a+0\vdash s(0+a)=s(a)+0}}{\mathbf{aks.(6)},\mathbf{0+s(a)=s(0+a)},0+a=a+0\vdash 0+s(a)=s(a)+0}}{0+a=a+0\vdash 0+s(a)=s(a)+0}}{\vdash 0+a=a+0\to 0+s(a)=s(a)+0}}{\vdash \forall x(0+x=x+0)(taikymas-indukcija)}$$

1931 m. K.Gödelis įrodė dvi svarbias teoremas.

1 teorema. *Egzistuoja tokia PA formulė F, kad nei* $\vdash F$, *nei* $\vdash \neg F$ *neįrodomos Peano aritmetikoje. T.y. PA aritmetika nepilna.*

2 teorema. *Tvirtinimas* **Peano aritmetika neprieštaringa** *aprašomas uždara* (*be laisvųjų kintamųjų*) *PA formule* (*pažymėkime ją F*). *Bet nei* $\vdash F$, *nei* $\vdash \neg F$ *neįrodomi Peano aritmetikoje.*

K.Gödelis kodavo formules bei jų įrodymus natūraliaisiais skaičiais (žr. skyrelį *Gödelio numeriai*). O natūraliuosius skaičius *k* formalioje aritmetikoje užrašydavo termu, vadinamu **numeralu** $\underline{k} = s(s(\ldots s(0))\ldots)$. Numerale yra *k* funkcijos *s* įeičių.

Jis įrodė, kad teorijoje PA yra tokia formulė Prf($\underline{m}, \underline{n}$), kuri teisinga tada ir tiktai tada, kai $\underline{n}$ yra kurios nors formulės kodas, o $\underline{m}$ yra jos išvedimo kodas. Taigi, $\underline{m}$ - formulės kodas, $\underline{n}$ - išvedimo kodas.

Prisiminkime pirmąją aksiomą $\forall\neg(s(x) = 0)$. Jei $x = 0$, tai turime teisingą PA formulę $\neg(s(0) = 0)$, t.y. teiginį $1 \neq 0$. Jei Peano aritmetikoje būtų įrodomas teiginys $1 = 0$, t.y. formulė $s(0) = 0$, tai PA būtų prieštaringa. Kiekvienai formulei yra priskirtas kodas. Tarkime formulės $s(0) = 0$ kodas $\underline{k}$.

Taigi, jei PA prieštaringa, tai joje įrodoma sekvencija $\vdash \exists y Prf(\underline{k}, y)$. Jei ji neprieštaringa, tai įrodoma sekvencija $\vdash \neg Prf(\underline{k}, y)$. Pagal antrąją K.Gödelio teoremą *nė viena iš šių sekvencijų nėra PA įrodoma.*

Nepavyks PA nepilnumo išvengti papildžius aksiomų sąrašą naujomis formulėmis. K.Gödelis įrodė, kad bet kurioje matematikos teorijoje, kurioje yra PA, neprieštaringumas negali būti įrodytas tos teorijos metodais.

Daugelis matematikų laiko, kad Hilberto svajonės apie idealią matematiką žlugo, Peano aritmetika *nėra pilna*. Nors, knygos autoriaus nuomone, tokia kritika abejotina. Jei PA būtų pilna, tai ji būtų ir išsprendžiama. Pasirodytų, kad matematikos pasaulis yra gana primityvus.

Kaip įrodomas aksiominių formalių teorijų neprieštaringumas?

Matematinė teorija *neprieštaringa*, jei nėra tokios teorijos formulės *F*, kad abi *F* ir $\neg F$ įrodomos.

Formalių aksiominių matematinių teorijų neprieštaringumo įrodymo būdai:

- *Papildant nagrinėjamą teoriją* (pažymėkime *T*) naujomis aksiomomis ar taisyklėmis (gautą naują teoriją pažymėkime *S*).

  Jei teorija *T* yra *S* poaibis ir *S* neprieštaringa, tai *T* taip pat neprieštaringa. Šiuo atveju gauname **sąlyginį neprieštaringumą**: *T* neprieštaringa tik su sąlyga, kad *S* neprieštaringa. O gal, savo ruožtu, įrodomas ir *S* tik sąlyginis neprieštaringumas. Ir taip gali tęstis be galo ilgai.

  Gal nėra *absoliutaus neprieštaringumo*. Priimta laikyti teoriją *T neprieštaringa* (absoliutus neprieštaringumas), jei egzistuoja konstruktyvus (naudojant intuicionistinę logiką) teorijos *S* neprieštaringumo įrodymas. Taip suprantama, kad *įrodymas priimtinas ir pakankamai įtikinantis*.

- *Randant teorijos modelį,* kurio neprieštaringumą mokame įrodyti.

- *Analizuojant visus galimus įrodymus* (*metamatematika*).

- *Redukuojant į neprieštaringą arba sąlyginai neprieštaringą teoriją.*

Panagrinėsime kai kuriuos Peano aritmetikos, pirmos eilės teorijos, neprieštaringumo įrodymus.

1) PA aksiomų sąrašas papildomas transfinitine indukcijos aksioma. Transfinitinė indukcija aprašoma antros eilės logikoje, bet neaprašoma jokia pirmos eilės logikos formule. PA teorija yra poaibis *PA + transfinitinė indukcija* teorijos. Naujoje teorijoje įrodomas PA neprieštaringumas. Bet ar pati naujoji teorija neprieštaringa? Deja, to iki šiolei nepavyko įrodyti.

Įrodytas **sąlyginis neprieštaringumas:**

jei *PA + transfinitinė indukcija* neprieštaringa, **tai** *PA neprieštaringa.*

2) PA aksiomų sistema papildoma Carnapo taisykle

$$\frac{\vdash F(0) \quad \vdash F(1) \quad \vdash F(2) \ldots}{\vdash \forall x F(x)}$$

Įrodyta, kad naujoje teorijoje PA yra pilna ir neprieštaringa.
Įrodytas **sąlyginis neprieštaringumas**.

Prisiminkime predikatų logikos išvedimo taisyklę

$$\frac{\vdash P(a)}{\vdash \forall x P(x)}$$

Priminsime, kad $\forall x P(x)$ įrodomas, jei **įrodomas šablonas** $P(a)$ (čia *a*) naujas laisvasis kintamasis). Tai reiškia, kad pakeitus *a* bet kuriuo konkrečiu skaičiumi, pavyzdžiui 25, pakaks kartoti tą patį įrodymą *P*(25), kaip ir su *a* (su kai kuriais konkrečiais skaičiais galbūt įrodymas bus trumpesnis).

Matematikoje tik taip ir *pagaunama begalybė.* Gali taip būti, kad visi $P(0), P(1), P(2), \ldots$ teisingi, o šablonas $P(a)$ neįrodomas. Tuomet neįrodoma ir $\forall x P(x)$.

Taigi, iš PA pilnumo, prijungus Carnapo taisyklę, išplaukia, kad Peano aritmetikoje yra be galo daug teiginių $F_i(n)$, kurie teisingi su visais natūraliaisiais skaičiais, bet formulės $\forall v F_i(n)$ neįrodomos. Ir tik dėl to aritmetika nepilna.

3) Prijungsim prie PA aksiomų vienvietį predikatą *P*, apibrėžtą natūraliųjų skaičių aibėje. $g(F)$ žymi uždaros formulės *F* Gödelio numerį. $P(g(F))$ teisingas tada ir tiktai tada, kai *F* yra teisinga. Toks predikatas neaprašomas jokia PA formule. Tokioje teorijoje įrodomas PA neprieštaringumas. Vadinasi, įrodomas **sąlyginis neprieštaringumas**.

4) Nagrinėjimo objektai yra ne formulės, o formulių įrodymai. Tokia teorija vadinama *įrodymų teorija* arba *metamatematika*. Tik analizuojant įrodymus, bei juos redukuojant į paprastesnius, G.Genzenui pavyko įrodyti, kad visų išvedamų formulių aibėje nėra tokių dviejų įrodymų, kad vienas baigtusi kuria nors formule *F*, o kitas jos neigimu. Vieni matematikai tai vadina **absoliučiu neprieštaringumo** įrodymu, kiti **sąlyginio neprieštaringumo** įrodymu.

Yra aprašyta ir daugiau PA neprieštaringumo įrodymų. Todėl nebeliko abejonių, kad PA yra **absoliučiai neprieštaringa.**

Pirmos eilės predikatų logikos absoliutus neprieštaringumas įrodytas naudojant interpretaciją, kurioje individinių konstantų aibėje tik vienas elementas (žr., pavyzdžiui, [2]). Tokioje interpretacijoje predikatų logikos formulės transformuojamos į teiginių logikos formules, o tokios logikos **absoliutus neprieštaringumas** niekam nekelia abejonių.

Panašiai įrodomas ir **grupių teorijos neprieštaringumas**. Aprašomas modelis, kuriame neįrodoma bent viena formulė.

Euklido geometrijos neprieštaringumas įrodomas transformuojant ją į aritmetiką.

3.5 Definavimas

Aibė $A \subseteq N^n$ (čia $n \geq 1$) vadinama **aritmetine**, jei yra tokia Robinsono aritmetikos formulė $F(a_1, \ldots, a_n)$ (priminsime, kad Robinsono ir Peano aritmetikų signatūros sutampa) su laisvaisiais kintamaisiais $a_1, \ldots, a_n$, kad, kokie bebūtų natūralieji skaičiai $k_1, \ldots, k_n$

$$\langle k_1, \ldots, k_n \rangle \in A \text{ tada ir tiktai tada, kai } F(\underline{k_1}, \ldots, \underline{k_n}) = t.$$

Sakoma, kad aibė *A* **definuojama** (aprašoma) formule *F*. Čia $\underline{k_1}, \ldots, \underline{k_n}$ yra numeralai (termai pavidalo $s(s(\ldots s(0))\ldots)$).

Pavyzdžiui, aibė $\{\langle a, b\rangle \mid a < b\}$ definuojama formule $\exists x(a + x = b \wedge \neg(x = 0))$.

Funkcija vadinama **aritmetine**, jei jos apibrėžimo aibė yra natūraliųjų skaičių aibė, o reikšmių aibė *E* yra kuris nors natūraliųjų skaičių poaibis.

Aritmetinės funkcijos $b = f(a_1, \ldots, a_n)$ **grafiku** vadiname aibę vektorių

$$G = \{\langle k_1, \ldots, k_n, k_{n+1}\rangle \mid k_{n+1} = f(k_1, \ldots, k_n)\}.$$

Aritmetinė **funkcija definuojama** Robinsono aritmetikoje, jei joje definuojamas funkcijos grafikas.

Įrodyta, kad **aritmetinė funkcija definuojama Robinsono aritmetikoje tada ir tiktai tada, kai ji rekursyvioji**.
Parodysime tik kaip definuojamos bazinės funkcijos bei tos, kurios gaunamos taikant superpozicijos ar minimizavimo operatorius:

- Bazinė funkcija $b = a + 1$ definuojama formule $b = s(a)$.
- Bazinė funkcija 0 definuojama formule $b = 0$.
- Bazinė funkcija $pr_i(a_1, \ldots, a_n) = a_i \ \ (1 \leq i \leq n)$ definuojama formule $b = 0 \cdot a_1 + \ldots + 0 \cdot a_{i-1} + a_i + 0 \cdot a_{i+1} + \ldots + 0 \cdot a_n$.
- Tarkime, turime $(k + 1)$ funkciją

  $$f(b_1, \ldots, b_k), g_1(a_1, \ldots, a_n), \ldots, g_k(a_1, \ldots, a_n). \qquad (3.1)$$

  Iš jų, panaudojus *superpozicijos operatorių*, gaunama *n* argumentų funkcija $b = f(g_1, \ldots, g_k)$. Tarkime, kad funkcijos (3.1) definuojamos formulėmis

  $$F(b_1, \ldots, b_k, b_{k+1}), G_1(a_1, \ldots, a_n, a_{n+1}), \ldots, G_k(a_1, \ldots, a_n, a_{n+1}).$$

  Tuomet funkcijų superpozicija definuojama aritmetikos formule

  $$\exists x_1 \ldots \exists x_k(G_1(a_1, \ldots, a_n, x_1) \wedge \ldots \wedge G_k(a_1, \ldots, a_n, x_k) \wedge F(x_1, \ldots, x_k, b)).$$

- Minimizacijos operatoriumi nauja gauta funkcija *g* aprašoma taip:

  $$g(a_1, \ldots, a_n) = \mu_y(f(a_1, \ldots, a_{n-1}, y) = a_n.$$

  Tarkime, kad funkcija *f* definuojama formule $F(a_1, \ldots, a_n, b)$. Tuomet funkcija *g* definuojama formule

  $$F(a_1, \ldots, a_n, 0) \vee \exists x[F(a_1, \ldots, a_n, x) \wedge \forall y(y < x \rightarrow \neg F(a_1, \ldots, a_n, y))]$$

PA definuojamų funkcijų aibė sutampa su Robinsono aritmetikoje definuojamomis funkcijomis. Vadinasi, aritmetinė funkcija definuojama Peano aritmetikoje tada ir tiktai tada, kai ji rekursyvioji.
Robinsono aritmetikoje neįrodomos daugelis funkcijų savybių, įrodomų Peano aritmetikoje. Pavyzdžiui, formulė $\forall x \forall y(x + y = y + x)$ įrodoma Peano aritmetikoje, bet neįrodoma Robinsono aritmetikoje.

Įrodyta, kad *Peano aritmetika* ekvivalenti (izomorfiška) aibių teorijai *ZF be begalybės aksiomos.*

Įdomi Peano aritmetikos uždarų formulių pavidalo $\exists x_1 \dots \exists x_n(F = G)$ klasė. Čia *F, G* bekvantorės PA formulės. Ji yra:

- pilna (teisingų, aprašytojo pavidalo, formulių atžvilgiu),
- neišsprendžiama.

A.Tarskis laikomas matematinės logikos pradininku Jungtinėse Valstijose. Jis sukūrė pirmąją JAV matematinės logikos mokyklą (Berklyje, Kalifornijos universitete). A.Tarskis 1956 metais įrodė bendresnę, nei K.Gödelio rezultatas apie aritmetikos nepilnumą, teoremą.

A.Tarskio teorema. *Pirmos eilės aritmetikos* ***teisingų formulių*** *aibė* (*tiksliau, tų formulių Gödelio numerių aibė*) *nėra aritmetinė*.

Neegzistuoja aksiomų sistemos, parašytos pirmos eilės logikos formulėmis, kurioje būtų išvedamos visos natūraliųjų skaičių savybės.

3.6 Antros eilės logika

Predikatų logikos formulės skirstomos į aibes (logikos formulių eiles) $L_1, L_2, L_3, \dots$ Pateiksime du skirtingus *logikos eilių* apibrėžimus. Pirmuoju apibrėšime savybes, kurias tenkina įvykdomos $L_i(i = 1, 2, \dots)$ formulės, o antruoju - kokio pavidalo formulės patenka į L_i $(i = 1, 2, \dots)$. Šiame skyrelyje nagrinėsime **grynosios logikos formules**, t.y. logikos formules, kuriose nėra nei *konkrečių predikatų,* nei *konkrečių funkcijų*, nei *konkrečių konstantų* (gali būti *predikatiniai kintamieji, funkciniai simboliai, individiniai kintamieji*).

Pirmasis apibrėžimas.

Tarkime, *F* kuri nors predikatų logikos formulė. Tikriname, ar ji *įvykdoma*. Ieškome struktūros, kurioje ji būtų teisinga. Ieškome konstantų aibės *D* bei joje apibrėžtų predikatų, funkcijų. Kiekviena *įvykdoma formulė* patenka bent į vieną iš šių aibių:

$\mathbf{L_1}$: *D* yra *baigtinė* arba *skaičioji.* **Pirmos eilės predikatų logika**. Arba tiesiog **pirmos eilės logika.**

$\mathbf{L_2}$: *D* yra baigtinė, skaičioji arba *kontinuumo galios.* **Antros eilės logika.**
Retai kuriame nors matematinės logikos vadovėlyje (skirtame aukštųjų mokyklų studentams) iš viso užsimenama apie *antros eilės logiką.* Todėl studentai, kaip taisyklė, terminą *logika* sutapatina su *pirmos eilės logiką.*

$\mathbf{L_3}$: *D* yra baigtinė, skaičioji, kontinuumo arba 2^c *galios.* **Trečios eilės logika.**

Tikiuosi supratote logikų klasifikaciją. Dar yra logikos $\mathbf{L_4}, \mathbf{L_5}, \ldots$

$$L_1 \subset L_2 \subset L_3 \subset L_4 \subset L_5 \subset \ldots$$

Kadangi *tapačiai teisingos* formulės taip pat yra įvykdomos, tai ta pati klasifikacija galioja ir **tapačiai teisingoms** formulėms.

Antrasis apibrėžimas. Klasės nusakomos *pagal formulių pavidalą.*

Priminsime, kad formulės *F* Gödelio numerį žymime $g(F)$.

$\mathbf{L_1}$. **Pirmos eilės logika**. Formulėse kvantoriais gali būti suvaržyti tik individiniai kintamieji.

Tarkime $A \subset L_1$ yra *visų **įrodomų** pirmos eilės logikos* formulių aibė. Deja, tokios algoritmiškai apskaičiuojamos funkcijos φ_A **neegzistuoja**:

$$\varphi_A(g(F)) = \begin{cases} 1, & \text{jei } F \in A \\ 0, & \text{priešingu atveju} \end{cases}$$

Tarp visų galimų kompiuteriui programų nėra tokios, kuri apskaičiuotų φ_A.

Šiek tiek nukrypsiu. Nagrinėsime funkciją

$$f(n) = \begin{cases} 1, & \text{jei skaičiuje } \pi \text{ yra lygiai n greta stovinčių 7} \\ 0, & \text{priešingu atveju} \end{cases}$$

Lygiai *n* suprantama taip: pavyzdžiui, skaičiuje $3, \ldots 27774 \ldots$ yra lygiai 3 greta stovintys septynetukai. Tokia funkcija neapskaičiuojama. Bet, aišku, kad funkcija $h(n)$ apskaičiuojama

$$h(n) = \begin{cases} 1, & \text{jei skaičiuje } \pi \text{ yra lygiai n greta stovinčių 7} \\ \infty, & \text{priešingu atveju} \end{cases}$$

Pavyzdžiui, $h(2)$ reikšmę galima taip rasti: ieškome $\ldots x77y \ldots$ (*x* ir *y* nelygūs 7) peržiūrėdami skaičių π iš kairės į dešinę. Jei randame $x77y \ldots$, tai spausdiname 1. Jei nėra, tai kompiuteris ieško (dirbs) "be galo ilgai".

Panaši situacija yra ir pirmos eilės logikoje.

Tarkime $A \subset L_1$ yra *visų **įrodomų** pirmos eilės logikos* formulių aibė. Tokia funkcija δ_A algoritmiškai apskaičiuojama (1929 metais įrodė K.Gödelis):

$$\delta_A(g(F)) = \begin{cases} 1, & \text{jei } F \in A \\ \infty, & \text{priešingu atveju} \end{cases}$$

Jei problema užrašoma pirmos eilės grynosios logikos formule F su funkciniais simboliais, tai galima parašyti tokią programą, kad, jei F įrodoma, tai po baigtinio žingsnių skaičiaus kompiuteris baigs darbą ir jums pateiks jos įrodymą tokiu pavidalu, kad bus sunku nustatyti, įrodė tai kompiuteris ar žmogus. Jei problema neįrodoma, tai kompiuteris gali ieškoti įrodymo be galo ilgai (teorinis variantas).

Savo laiku buvo kilęs didelis džiaugsmas tokiu rezultatu. Jei žinome kurią nors neišspręstą problemą ir mokame ją aprašyti pirmos eilės logikos formule, tai įrodymo paiešką galbūt galima pavesti kompiuteriui. Pasipylė *automated theorem prover*'ių (programų skirtų pirmos eilės grynosios logikos formulių įrodymų paieškai) rašymas.

Bet, pasirodo, viskas yra sudėtingiau. Neužtenka pakeisti predikatinius kintamuosius ir funkcinius simbolius konkrečiais predikatais ir funkcijomis, t.y. pirmos eilės formule užrašyti kurios nors formalios aksiominės teorijos problemą. Yra dar ir papildomi reikalavimai matematinėms teorijoms. Be to, kompiuteris reikalauja daug atminteis ir dirba ilgai netgi su nesudėtingais uždaviniais. Techninės priežastys: *atmintis* ir *laikas*. Vis tik kompiuteriu pavyko įrodyti daug teoremų. Bet absoliuti dauguma ir taip buvo žinomos.

Šioje knygoje yra keletas **formalių aksiominių teorijų**, kurios naudoja **pirmos eilės logiką**, pavyzdžių:

1. lygybės predikato teorija,

2. griežtos tvarkos teorija,

3. grupių teorija,

4. Peano aritmetika,

5. aibių teorija *ZF*.

Paminėsime dar vieną matematikos teoriją, kurios problemoms spręsti naudojama pirmos eilės logika (tokių teorijų yra daug; knygoje aprašome tik kelias). Tai **sveikųjų skaičių žiedas**

Signatūra:

- konstantos: 0, 1
- predikatai: $a = b$. Vienas dvivietis predikatas (lygybės predikatas)
- funkcijos: $+, -, \times$. Viena vienvietė funkcija $-$, kuri kiekvienam skaičiui a priskiria jam priešingą $-a$ ir dvi dvivietės funkcijos $+, \times$.

$\mathbf{L_2}$. **Antros eilės logika.** *Formulėse kvantoriais gali būti suvaržyti ne tik individiniai kintamieji, bet ir predikatiniai kintamieji bei funkciniai simboliai*.

Antros eilės logikoje yra formulės, kurios skiriasi vizualiai nuo pirmos eilės.

Formulių pavyzdžiai.

1. Savybių dvireikšmiškumo principas $\forall P \forall n(P(n) \vee \neg P(n))$ yra antros eilės formulė.

2. $\forall x(\exists y P(x,y) \to P(x, f(x)))$ yra pirmos eilės formulė (ji yra, tuo pačiu ir antros, trečios ir t.t. eilės formulė; bet, priimta, vardą pagal pavidalą, priskirti žemiausiai eilei).

3. $\exists f \forall x(\exists y P(x,y) \to P(x, f(x)))$ yra antros eilės logikos formulė, nes funkcija f suvaržyta kvantoriumi $\exists$.

4. Įrodyta, kad antros eilės logikos formulė

$$\neg\exists z \exists f \forall P[(P(z) \wedge \forall x(P(x) \to P(f(x))) \to \forall x P(x)]$$

įvykdoma kontinuumo galios aibėje ir neįvykdoma jokioje numeruojamoje (baigtinėje arba skaičiojoje) aibėje.

5. Turime neorientuotą be kilpų grafą. *D* - viršūnių aibė. Joje apibrėžtas dvivietis predikatas *E* (briauna). Teiginys *grafe galima rasti dvi tokias netuščias viršūnių aibes A, B, kad nėra briaunos jungiamčios kurią nors A viršūnę su kuria nors B viršūne* užrašomas antros eilės formule

$$\exists A \exists B[(\exists A(x) \wedge \exists x B(x) \wedge \forall x \forall y(A(x) \wedge B(y))) \to \neg E(x,y)].$$

Tarkime $A \subset L_2$ yra *visų įrodomų antros eilės logikos* formulių aibė. Deja, tokios algoritmiškai apskaičiuojamos funkcijos φ_A neegzistuoja ($F \in L_2$):

$$\varphi_A(g(F)) = \begin{cases} 1, & \text{jei } F \in A \\ 0, & \text{priešingu atveju} \end{cases}$$

Neegzistuoja ir tokios (palyginkite su atitinkama funkcija pirmos eilės logikai)

$$\delta_A(g(F)) = \begin{cases} 1, & \text{jei } F \in A \\ \infty, & \text{priešingu atveju} \end{cases}$$

Tarkime $\beta(F)$ kuri nors algoritmiškai apskaičiuojama (t.y. ją galima suprogramuoti) funkcija apibrėžta su visomis antros eilės formulėmis (reikšmių aibė $\{0,1\}$). Tuomet aibė $B_\beta = \{F \mid jei\ \beta(F) = 1\}$ yra griežtas aibės *A* (priminsiu, kad aibė *A* yra *visų įrodomų antros eilės logikos* formulių aibė) poaibis $B_\beta \subset A$, arba griežtas viršaibis $A \subset B_\beta$.

*Antros eilės logika vadinama **standartine**, jei joje nagrinėjamų funkcijų ir predikatų apibrėžimų aibė yra viena ir ta pati aibė D, o funkcijų reikšmių aibė E yra D poaibis.*

Šioje knygoje nagrinėjami tik standartiniai dviejų *formalių aksiominių teorijų* pavyzdžiai, kurie naudoja *antros eilės logiką.* Tai aibių teorija *ZFC* ir antros eilės aritmetika (dar vadinama

analize).

1955 metais suomių logikas J.Hintikka įrodė, kad kokia bebūtų formulė F aukštesnės nei antros eilės logikos (*t.y. trečios, ketvirtos ir t.t.*) *galima rasti tokią antros eilės formulę G, kad arba abi įvykdomos, arba abi neįvykdomos.*

Ehrenfeucht-Fraisse žaidimas.

Du žaidėjai: *A, B.*

Du grafai: G_1 ir G_2.

Ėjimų skaičius $k \in N$.

Žaidimo taisyklės. n-tasis ėjimas ($n = 1, 2, \ldots, k$*).* Jo metu abu žaidėjai pasirenka po grafo viršūnę. Iš pradžių žaidėjas *A* pasirenka kurį nors grafą ir jame viršūnę. Po to, žaidėjas *B,* žinodamas *A* pasirinkimą, kitame grafe pasirenka kurią nors viršūnę.

Po *k* ėjimų žaidimas stabdomas.

Tarkime, pirmame grafe pasirinktos $a_1, \ldots, a_k$, o antrame $b_1, \ldots, b_k$ (kai kurios viršūnės gali sutapti). Laimi žaidėjas *B*, jei pografiai sudaryti iš viršūnių $a_1, \ldots, a_k$ ir $b_1, \ldots, b_k$ yra izomorfiniai. Priešingu atveju laimi žaidėjas *A*.

Ar egzistuoja žaidėjui *B* laimėjimo strategija? Tai priklauso nuo duotų grafų.

Teorema. *Šeimos S grafų negalima aprašyti* ***pirmos eilės logikos formule****, jei kiekvienam n* (*ėjimų skaičius*) *atsiras tokie sistemoje S du grafai, kad žaidėjui B egzistuoja laimėjimo strategija.*

Grafų šeimų *S* pavyzdžiai:

a) Eulerio grafai,

b) plokštieji grafai,

c) medžiai.

Grafų teorijos teiginiams formalizuoti dažniausiai naudojami vienviečiai antros eilės predikatiniai kintamieji.

Duotas neorientuotas be kilpų grafas (V, E). Čia *V* - viršūnių aibė, *E* - dvivietis predikatas su apibrėžimo aibe *V*. $E(a, b) = t$ tada ir tiktai tada, kai grafe yra briauna jungianti *a, b*. Vienviečiais predikatais nusakome aibės *V* poaibius.

Pavyzdys. Teiginys *egzistuoja pografis, kurio visos viršūnės izoliuotos,* užrašomas formule:

$$\exists X \forall x (X(x) \to \forall y \neg E(x, y)).$$

Aukštesnės eilės logikos formulėmis paprasčiau ir suprantamiau galima aprašyti matematikos teiginius. Ją dažniausiai tik ir naudoja matematikai teiginių užrašymams ir teoremų įrodymams. Bet paprastumas kainuoja.

Antros eilės logikos tapačiai teisngų formulių aibė ***nėra rekursyviai skaiti****. Tai reiškia, kad nėra skaičiavimo, kuriame bet kuri formulė būtų išvedama tada ir tiktai tada, kai ji tapačiai teisinga.*

Dar kartą priminsime, kad pirmos eilės logika yra griežtas antros eilės logikos poaibis. Antros eilės logikoje aprašomi kai kurie teiginiai, kurių neįmanoma aprašyti pirmos eilės logikos formule. Pavyzdžiui, parašysime formulę, kuria tvirtinama, kad struktūros aibė *D* yra begalinė:

$$\exists y \exists f \forall X ((X(y) \wedge \forall x (X(x) \to X(f(x)))) \to \forall x X(x))$$

Priminimas.
1. Formule su vienu laisvuoju kintamuoju $F(n)$ nusakome kažkurį tai *apibrėžimo aibės D poaibį.* Jam priklauso tie *n*, su kuriais $F(n)$ teisinga.
2. *Uždara formulė* (formulė be laisvųjų kintamųjų) yra *teiginys*. Ji yra arba teisinga, arba klaidinga.
3. Formali aksiominė teorija vadinama *pilnąja*, jei kokia bebūtų jos signatūroje parašyta *uždara formulė F*, įrodoma arba F, arba $\neg F$.
4. Peano aritmetikoje yra be galo daug tokių uždarų formulių, kad neįrodomos nei jos, nei jų neiginiai.

3.7 Antros eilės formalioji aritmetika

Peano aritmetikoje įrodoma tik dalis, netgi žinomų, įrodytų matematikoje, natūraliųjų skaičių savybių. Papildysime aritmetikos aksiomų sąrašą nauja antros eilės aksioma. Antros eilės teorijoje įrodomos visos pirmos eilės teorijos formulės bei daug naujų, neįrodomų pirmos eilės aritmetikoje. Yra įvairių antros eilės formaliosios aritmetikos teorijų (žr. knygoje [4]). Aprašysim dažniausiai naudojamą variantą - *standartinę antros eilės aritmetiką Z2*, kurią 1939 metais aprašė D.Hilbertas ir P.Bernays knygoje *Grundlagen der Mathematik* antrame tome.

Antros eilės aritmetikos Z2 signatūra:

- konstantos: 0
- predikatai: $=$
- funkcijos: $s, +, \cdot$

Yra tik vienas dvivietis lygybės predikatas, trys funkcijos: $s(a) = a + 1, a + b$ (sudėtis), $a \cdot b$ (daugyba) bei viena konstanta 0 (nulis).

Aritmetikos aksiomos.

1. $\forall x \neg(s(x) = 0)$

2. $\forall x \forall y(s(x) = s(y) \to x = y)$

3. $\forall x(x + 0 = x)$

4. $\forall x \forall y(x + s(y) = s(x + y))$

5. $\forall x(x \cdot 0 = 0)$

6. $\forall x \forall y(x \cdot s(y) = (x \cdot y) + y)$

7. $\forall F[(F(0) \wedge \forall x F(x) \to F(s(x)))) \to \forall x F(x)$

8. $\exists X \forall x(X(x) \leftrightarrow F(x))$

8-ta aksioma vadinama *supratimo aksioma* (axiom of comprehension). Čia F bet kuri *Z2* formulė, kurioje nėra *X* įeičių. Aksioma tvirtinama, kad egzistuoja natūraliųjų skaičių aibė X, sudaryta iš visų tokių natūraliųjų *a* su kuriais $F(a) = t$.

Teorijoje *Z2 individinių kintamųjų reikšmės* yra natūralieji skaičiai, o *predikatinių kintamųjų* - natūraliųjų skaičių poaibiai.

Tarkime *A* vienvietis predikatas, nusakantis kurį nors natūraliųjų skaičių poaibį. Pritaikę formulei $\neg A$ supratimo aksiomą, gauname, kad egzistuoja aibės *A* papildinys: $\exists X \forall x(X(x) \leftrightarrow \neg A(x))$.

Formulių įrodymai teorijoje *Z2*, kaip taisyklė, vykdomi naudojant *antros eilės* ***intuityvią logiką***. Teorija *Z2* pakankamai "stipri". Joje įrodomi praktiškai visi žinomi matematikos teiginiai, kuriuos galima aprašyti *Z2* kalboje (signatūroje).

Iš teorijos *Z2* , taikant apribojimus *supratimo* arba *indukcijos aksiomoms*, gaunamos naujos penkios teorijos, tarp kurių yra toks ryšys:

$$RCA_0 \subset WKL_0 \subset ACA_0 \subset ATR_0 \subset \Pi^1_1 - CA_0 \subset Z2.$$

Tai reiškia, kad, jei *F* įrodoma, pavyzdžiui, teorijoje RCA_0, tai ji įrodoma ir visose išvardintose. Taip pat yra formulės, įrodomos RCA_0, bet neįrodomos pirmos eilės aritmetikoje. Indeksu nulis nurodome, kad nagrinėjamoji teorija yra griežtas teorijos *Z2* poaibis.

Aprašysime vieną antros eilės aritmetikos fragmentą.

Antros eilės aritmetika $\mathbf{RCA_0}$.

Teorijos RCA_0 (recursive comprehension axiom) signatūrą sudaro PA signatūra, papildyta predikatu <.

Aksiomomis yra Robinsono aritmetikos aksiomos plius dar keturios:

$\forall x \neg (x < 0)$,

$\forall x \forall y (x < s(y) \leftrightarrow (x < y \vee x = y))$,

$(F(0) \wedge \forall x (F(x) \rightarrow F(s(x)))) \rightarrow \forall x F(x)$ (indukcijos aksioma),

$\forall x (f(x) \leftrightarrow G(x)) \rightarrow \exists X \forall x (x < t \rightarrow (X(x) \leftrightarrow F(x)))$ (supratimo aksioma).

Formulės *F, G* tenkina tam tikras sąlygas. Aprašysime jas:

- formulėse nėra kvantorinių kompleksų, suvaržančių aibių kintamuosius, t.y. kvantorinių kompleksų pavidalo $\forall X, \exists X$.
- visi kvantoriniai kompleksai, suvaržantys *individinius* formulių kintamuosius yra *apriboti termais.* T.y. visos formulės (poformuliai) yra pavidalo $\forall x (x < t \rightarrow H(x)), \exists x (x < t \rightarrow H(x))$. *t* yra termas, kriame nėra *x* įeičių. Tokios formulės vadinamos *apribotomis*,
- formulės $F(x)$ abejose (indukcijos ir supratimo) aksiomose priklauso formulių klasei $\sum_1^0$. Jai priklauso *apribotos* aritmetikos formulės pavidalo $\exists x_1 \ldots \exists x_n H(x_1, \ldots, x_n, a)$ su laisvuoju kintamuoju *a*. Įrodyta, kad tokiomis formulėmis aprašomos rekursyviai skaičiosios natūraliųjų skaičių aibės.

Apribotų formulių pavyzdys.

$$\exists x \exists y \exists z ((x < 2^a \wedge y < (x + a) \wedge z < (x + y)) \rightarrow H(x, y, z, a))$$

Vadinasi, **indukcijos aksioma teorijoje** RCA_0 taikoma tik formulėms aprašančioms rekursyviai skaičiąsias aibes.

Formulė *G* supratimo aksiomoje priklauso aritmetikos klasei Π_1^0. Jai priklauso *apribotos* aritmetikos formulės pavidalo $\forall x_1 \ldots \forall x_n H(x_1, \ldots, x_n, a)$. Įrodyta, kad, jei funkcija priklauso abejoms klasėms (tenkina sąlygą $\forall x (F(x) \leftrightarrow G(x))$), tai ja nusakoma aibė yra išsprendžiama (rekursyvi).

Vadinasi, **supratimo aksioma teorijoje** RCA_0 **taikoma tik išsprendžiamoms (rekusyvioms) aibėms.**

Kadangi įrodymai antros eilės logikoje dažniausiai tik intuityvūs, tai praktikoje netikrinama sąlyga $\forall x (F(x) \leftrightarrow G(x))$. Stengiamasi kitokiais būdais įrodyti, kad formulės, kuriai leidžiama taikyti supratimo aksiomą, nusako išsprendžiamą natūraliųjų skaičių aibę.

Teorija RCA_0 dar vadinama apskaičiuojamos matematikos (computable mathematics) teorija.

Teoremų, įrodomų teorijoje RCA_0 ir neįrodomų pirmos eilės aritmetikoje pavyzdžiai:

- Jei kurios nors formalios aksiominės teorijos uždarų formulių aibė turi skaitųjį modelį, tai ta formulių aibė neprieštaringa.
- Jei funkcija f tolydi atkarpoje [0;1] ir $f(0) < 0 < f(1)$, tai egzistuoja toks x $(0 < x < 1)$, kad $f(x) = 0$.

Pavyzdys formulės įrodomos antros eilės aritmetikoje ir neįrodomos teorijoje RCA_0:

Königo lema. Bet kuriame medžio pavidalo grafe, kurio kiekvienos viršūnės laipsnis baigtinis, egzistuoja begalinis kelias.

3.8 Aksiominė aibių teorija ZFC

Kadangi naivi Kantoro aibių teorija pasirodė esanti prieštaringa, tai išeitis, turėti aibių teoriją be paradoksų, buvo rasta aprašant ją formalios aksiominės teorijos pavidalu. Buvo sukurta ne viena aksiominė aibių teorija. Labiausiai žinoma yra *Zermelo-Fränkelio* aibių teorija. Joje žingsnis po žingsnio, pradedant tuščia aibe $\emptyset$ (taigi, teorija gaunama "iš nieko") konstruojamos kitos: $\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \ldots$ Aksiomos aprašomos naudojant tik du predikatus: $=, \in$. Formulė $x \in y$ suprantama kaip *x yra aibės y elementas*. Kintamaisiais $x, y, z, \ldots$ žymime ir aibes, ir jų elementas (aibių elementais yra kitos aibės). Nagrinėjamos aibių teorijos dalį aksiomų aprašė 1908 m. Ernst Zermelo, o likusias 1922 m. Abraham Fränkel ir Toralf Skolem.

Pradėsime nuo *ZF* aibių teorijos aprašymo, kurią papildę dar viena aksioma, gausime aksiominę aibių teoriją *ZFC*. Atkreipiame dėmesį, kad *ZF* aksiomos aprašomos pirmos eilės logikos formulėmis.

Zermelo-Fränkelio aibių teorijos ZF signatūra:

- konstantos: $\emptyset$
- predikatai : $=, \in$
- funkcijos *nėra*

Ar nestebina jūsų šios teorijos paprastumas? Teorijai, kuri yra visos matematikos pagrindas, sukurti tereikia *tuščios aibės* ir dviejų, visiems suprantamų, predikatų.

Aksiomos:
1. $\forall y \forall z(\forall x(x \in y \leftrightarrow x \in z) \rightarrow y = z)$.
Jei aibės susideda iš vienodų aibių (elementų), tai jos lygios.

2. $\forall x \neg(x \in \emptyset)$.
Tuščios aibės aksioma.
Formulę $a = \emptyset$ galima parašyti ir taip $\forall x \neg(x \in a)$.

3. $\forall x \forall y \exists z(z \in u \leftrightarrow (z = x \vee z = y))$.
Poros aksioma.
Jei *a* ir *b* aibės, tai $c = \{a, b\}$ taip pat aibė. *ZF* teorijoje įrodoma, kad taip nusakoma aibė *u* yra vienintelė. Netuščioms aibėms aprašyti naudojame skliaustus $\{, \}$. Tvarka tarp jos elementų nesvarbi. 3-ia aksioma nusakoma aibė, kurioje neįvesta tvarka (nesutvarkyta aibė). Kaip matome, teiginys $c = \{a, b\}$ užrašomas formule $\forall z(z \in c \leftrightarrow (z = a \vee z = b))$.
Teiginys $c = \{a\}$ formule $\forall z(z \in c \leftrightarrow z = a)$.

4. $\forall z \exists y \forall x(x \in y \leftrightarrow \exists u(u \in z \wedge x \in u))$.
Sumos aksioma.
Visų aibės z elementų sąjunga yra irgi aibė. Sakoma, kad y yra aibės z elementų sąjunga ir žymima $\cup z$
Pavyzdžiui, $z = \{\{\emptyset, \{\emptyset\}\}, \{\{\{\emptyset\}\}\}\}$. Tuomet $y = \{\emptyset, \{\emptyset\}\} \cup \{\{\{\emptyset\}\}\} = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}$.

5. $\forall z \exists y \forall x(x \in y \rightarrow (x \in z \wedge A(x)))$.
Poaibio aksioma $c \subseteq b$.
$A(a)$ - kuri nors formulė signatūroje $\{=, \in\}$ su vienu laisvu kintamuoju a. Kaip matome, tiksliau būtų sakyti, kad turime poaibių aksiomų schemą, nes aibių $A(a)$ yra be galo daug. z yra aibė visų elementų *a*, su kuriais *A* teisinga. Aksioma tvirtinama, kad egzistuoja z poaibis *y*, kurio elementais (aibėmis) yra tie *x*, su kuriais $A(x)$ teisinga. Kai kuriose aksiominėse teorijose vietoje implikacijos 5-toje aksiomoje naudojama lygybė, nes įrodoma, kad kiekvienai $A(a)$ egzistuoja vienintelis poaibis *y*.

6. $\exists x(\emptyset \in x \wedge \forall y(y \in x \rightarrow y \cup \{y\} \in x))$.
Begalybės aksioma.
Aksioma nusakomas natūraliųjų skaičių aibės egzistavimas. Aksiomą galima parašyti ir be $y \cup \{y\}$, tik ji bus ilgesnė ir sunkiau suprantama:
$\forall w(\forall z(\neg(z \in w) \rightarrow w \in x)) \wedge \forall y(y \in x \rightarrow \forall u(\forall v(v \in u \leftrightarrow (v \in y \vee v = y)) \rightarrow u \in x))$.
Naudojantis šia aksioma gaunamos aibės
$\{\emptyset, \{\emptyset\}\}$, $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$, $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\}$ ir t.t.
Į tokio tipo aibes patogu, sprendžiant užduotis, žiūrėti kaip į natūraliuosius skaičius. Pažymėjus tuščią aibę 0, aibę $\{\emptyset\}$ - 1, $\{\emptyset, \{\emptyset\}\}$ - 2 ... gaunama natūraliųjų skaičių aibė.

7. $\forall x \exists y \forall x(x \in y \leftrightarrow x \subseteq z)$.

Ši aksioma dar vadinama *laipsnio aksioma*. Šia aksioma apibrėžiamas poaibių aibės egzistavimas. *y* žymimas 2^z. Taip gaunamos kontinuumo galios ir dar didesnių galių aibės.
Laipsnio aksiomą galima parašyti ir taip

$$\forall z \forall x (x \in 2^z \leftrightarrow x \subseteq z).$$

8. $\forall x(\neg(x = \emptyset) \rightarrow \exists y(y \in x \wedge y \cap x = \emptyset))$.
Parašysime ją pirmos eilės logikos formulės pavidalu:
$\forall x(\exists z(z \in x) \rightarrow \exists y(y \in x \wedge \forall w \neg(w \in x \wedge w \in y)))$.
Bet kurioje aibėje nėra begalinės susitraukiančių aibių sekos.
Seka $x_1, x_2, x_3, \ldots$ vadinama susitraukianti, jei jos nariai tenkina sąlygas: $x_2 \in x_1, x_3 \in x_2, x_4 \in x_3, \ldots$
Ši aksioma naudojama kaip indukcijos aksioma.

9. $\forall z \exists y \forall x(x \in y \leftrightarrow \exists v((v \in z \wedge G(u, v) = t) \rightarrow x = u))$.
Keitimo aksioma.
Aibė *y* gaunama iš *z*, pakeitus kiekvieną aibės *z* elementą *v* tokiu elementu *u*, su kuriuo $G(u, v)$ yra teisinga ($G(u, v) = t$). Čia *t* - loginė konstanta *teisinga*. $G(u, v)$ yra kuri nors *ZF* formulė su dviem laisvais kintamaisiais. Be to, reikalaujama, kad kiekvienam *u* egzistuotų vienintelis *v*. Turime *aksiomų schemą*, nes tik pakeitę $G(u, v)$ konkrečia formule gausime aksiomą. Bet taip jau įprastą logikoje, kad, jei iš konteksto aišku, kad tai schema, tai rašomas tik žodis *aksioma*.

Turbūt pastebėjote, kad aksiomoje 7 naudojame ne tik predikatus iš signatūros. Tuo siekiama, kad aksioma būtų parašyta trumpesne ir lengviau suprantama formule. Ją galima perrašyti ir formulėmis tik su predikatais iš signatūros.

Aibių teorija ZFC.

Prie aksiomų 1-9 prijungiama dar viena aksioma:

10. $\forall z \exists f \forall x((x \in z \wedge \neg(x = \emptyset)) \rightarrow f(x) \in x)$.
Rinkimo aksioma.
Kokia bebūtų aibė *z*, atsiras tokia funkcija *f* išrenkanti iš bet kurios aibės *z* netuščio elemento *x* vienintelį elementą $f(x) \in x$.
Funkcija *f* suformuojama nauja aibė - visų aibės *z* elementų (netuščių elementų) *atstovų aibė.*

Pastaba. Funkcijai *f* nekeliami jokie apribojimai. Kitaip tariant, ji gali būti ir **algoritmiškai neapskaičiuojama.**

10 aksioma yra antros eilės formulė, o 1-9 aksiomomis apibrėžta aibių teorija yra *pirmos eilės formali aksiominė teorija.*

Kokias formules vadiname teisingomis teorijoje *ZFC*? Tai formulės, kurios išvedamos iš teorijos aksiomų. Jei išvedama sekvencija $A_1, A_2, \ldots, A_{10} \vdash F$ antros eilės predikatų logikoje, tai *F* priklauso teisingų teorijos *ZFC* formulių aibei ($A_i \ \ (i = 1, 2, \ldots, 10)$ yra *ZFC* aksiomos).

Rinkimo aksioma ekvivalenti Zermelo aksiomai bei Corno lemai.

Zermelo aksioma. Bet kurioje aibėje galima įvesti visišką tvarką (galima apibrėžti visiškos tvarkos binarinį sąryšį).

Corno lema. Jei netuščioje aibėje apibrėžtas dalinės tvarkos sąryšis ir bet kuri grandinė turi vienintelį rėžį, tai aibėje egzistuoja bent vienas maksimalus elementas.

Pavyzdžiai. Pavyzdžių įrodymui naudosime pirmos eilės predikatų logikos skaičiavimą su lygybės predikatu.

1. Įrodyti, kad aibės $\{x\}$ ir $\{x, x\}$ yra lygios su bet kuriomis aibėmis *x*.

Pažymėkime raide *u* aibę $\{x\}$, o raide *v* aibę $\{x, x\}$.

$u = \{x\}$, $v = \{x, x\}$.

Remiantis 3-ia aksioma $u = \{x\}$ užrašoma formule $\forall x \exists u \forall z (z \in u \leftrightarrow z = x)$.

Remiantis 3-ia aksioma $v = \{x, x\}$ užrašoma formule $\forall x \exists v \forall z (z \in v \leftrightarrow (z = x \vee z = x))$. Formulę galim supaprastinti $\forall x \exists v \forall z (z \in v \leftrightarrow z = x)$.

Reikia įrodyti, kad tos formulės ekvivalenčios, t.y. jomis nusakomos aibės yra lygios:
a) $\forall x \exists u \forall z (z \in u \leftrightarrow z = x) \vdash \forall x \exists v \forall z (z \in v \leftrightarrow z = x)$ ir
b) $\forall x \exists v \forall z (z \in v \leftrightarrow z = x) \vdash \forall x \exists u \forall z (z \in u \leftrightarrow z = x)$

Įrodysime tik atvejį *a*. Antrasis atvejis įrodomas panašiai. Įrodymas grynai sintaksinis. Taikant taisykles $(\forall \vdash), (\vdash \exists)$ formulių nekartosime, nes matosi, kad šio uždavinio sprendimui to nereikia.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall z(z \in b \leftrightarrow z = a) \vdash \forall z(z \in v \leftrightarrow z = a)}\oplus}{\forall z(z \in b \leftrightarrow z = a) \vdash \exists \mathbf{v} \forall \mathbf{z}(\mathbf{z} \in \mathbf{v} \leftrightarrow \mathbf{z} = \mathbf{a})}}{\exists \mathbf{u} \forall \mathbf{z}(\mathbf{z} \in \mathbf{u} \leftrightarrow \mathbf{z} = \mathbf{a}) \vdash \exists v \forall z(z \in v \leftrightarrow z = a)}}{\forall \mathbf{x} \exists \mathbf{u} \forall \mathbf{z}(\mathbf{z} \in \mathbf{u} \leftrightarrow \mathbf{z} = \mathbf{x}) \vdash \exists v \forall z(z \in v \leftrightarrow z = a)}}{\forall x \exists u \forall z (z \in u \leftrightarrow z = x) \vdash \forall \mathbf{x} \exists \mathbf{v} \forall \mathbf{z}(\mathbf{z} \in \mathbf{v} \leftrightarrow \mathbf{z} = \mathbf{x})}$$

2. Parodysime, kad 1-ojoje aksiomoje $\forall y \forall z (\forall x (x \in y \leftrightarrow x \in z) \rightarrow y = z)$ sukeitus implikacijos prielaidą su išvada, gauname įrodomą formulę $\forall y \forall z (y = z \rightarrow \forall x (x \in y \leftrightarrow x \in z))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a = b \vdash c \in b \leftrightarrow c \in b}\oplus}{\mathbf{a} = \mathbf{b} \vdash c \in a \leftrightarrow c \in b}}{a = b \vdash \forall \mathbf{x}(\mathbf{x} \in \mathbf{a} \leftrightarrow \mathbf{x} \in \mathbf{b})}}{\vdash a = b \to \forall x(x \in a \leftrightarrow x \in b)}}{\vdash \forall z(a = z \to \forall x(x \in a \leftrightarrow x \in z))}}{\vdash \forall y \forall z(y = z \to \forall x(x \in y \leftrightarrow x \in z))}$$

3. Įrodysime, kad, kokia bebūtų aibė *x*, aibių teorijoje neįrodoma $x \in x$, t.y. įrodoma $\forall x \neg(x \in x)$. Įrodymui pakanka 8-tos aksiomos.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a \in a \vdash \neg(a \in a), a \in a}\oplus \quad \dfrac{}{a \in a \vdash \neg(a \in a), a \in a}\oplus}{a \in a \vdash \neg(a \in a), \mathbf{a} \in \mathbf{a} \wedge \mathbf{a} \in \mathbf{a}}}{a \in a, \neg(\mathbf{a} \in \mathbf{a} \wedge \mathbf{a} \in \mathbf{a}) \vdash \neg(a \in a)}}{a \in a, \forall \mathbf{w} \neg(\mathbf{w} \in \mathbf{a} \wedge \mathbf{w} \in \mathbf{a}) \vdash \neg(a \in a)}}{\mathbf{a} \in \mathbf{a} \wedge \forall \mathbf{w} \neg(\mathbf{w} \in \mathbf{a} \wedge \mathbf{w} \in \mathbf{a}) \vdash \neg(a \in a)} \quad \dfrac{\dfrac{\dfrac{}{a \in a \vdash a \in a}\oplus}{\vdash \neg(\mathbf{a} \in \mathbf{a}), a \in a}}{\vdash \neg(a \in a), \exists \mathbf{z}(\mathbf{z} \in \mathbf{a})}}{\exists \mathbf{y}(\mathbf{y} \in \mathbf{a} \wedge \forall \mathbf{w} \neg(\mathbf{w} \in \mathbf{a} \wedge \mathbf{w} \in \mathbf{y})) \vdash \neg(a \in a)}}{\exists \mathbf{z}(\mathbf{z} \in \mathbf{a}) \to \exists \mathbf{y}(\mathbf{y} \in \mathbf{a} \wedge \forall \mathbf{w} \neg(\mathbf{w} \in \mathbf{a} \wedge \mathbf{w} \in \mathbf{y})) \vdash \neg(a \in a)}}{\forall \mathbf{x}(\exists \mathbf{z}(\mathbf{z} \in \mathbf{x}) \to \exists \mathbf{y}(\mathbf{y} \in \mathbf{x} \wedge \forall \mathbf{w} \neg(\mathbf{w} \in \mathbf{x} \wedge \mathbf{w} \in \mathbf{y}))) \vdash \neg(a \in a)}}{\forall x(\exists z(z \in x) \to \exists y(y \in x \wedge \forall w \neg(w \in x \wedge w \in y))) \vdash \forall \mathbf{x} \neg(\mathbf{x} \in \mathbf{x})}$$

Priminsime du apibrėžimus:

Uždara formulė F (t.y. formulė be laisvųjų kintamųjų) yra *teiginys*. Ji yra arba teisinga, arba klaidinga.
Formali matematinė teorija vadinama *pilnąja*, jei kokia bebūtų jos signatūroje parašyta *uždara formulė F*, įrodoma arba *F*, arba $\neg F$.

Aibių teorija ZC vadinama teorija, kurios aksiomomis yra *ZFC* aksiomos 1, 5, 6, 7 (priminsiu šias aksiomas: 1) aibių lygybės, 5) poaibio, 6) begalybės, 7) laipsnio) ir rinkimo aksioma 10. Daugeliu atveju matematikos teoremų įrodymui pakanka teorijos *ZC*.

Aibių teorija IZF vadinama teorija, kurios aksiomomis yra *ZF* aksiomos, o logika *pirmos eilės intuicionistinė*. Įrodyta, kad $IZF \subset ZF \subset ZFC$.

Pažymėkime raide *R rinkimo aksiomą.* Jei ji yra teorijos aksiomų sąraše, rašysime *R*, o reiškiniu $\neg R$, jei yra jos neigimas. Analogiškai, raide *K* žymėsime kontinumo hipotezę.

Priminsiu kontinuumo hipotezę. *Bet kuris realiųjų skaičių poaibis arba yra skaitusis, arba kontinuumo galios.*

1937 metų K.Gödelio bei 1963 metų P.Koeno darbais įrodyta, kad yra keturios skirtingos aksiominės formalios aibių teorijos:

1. ZF + R + K,
2. ZF + R + $\neg K$,
3. ZF + $\neg R$ + K,
4. ZF + $\neg R$ + $\neg K$.

Kuri patinka, tą ir laikykite matematikos pagrindu. Skonio reikalas. O jei jums dar nepatinka ir klasikinė logika, rinkitės intuicionistinę (turėsite dar vieną skirtingą matematikos sampratą).

Kodėl nenagrinėjama matematika, kurioje nėra rinkimo aksiomos, o yra *kontinuumo hipotezės neigimas*? Manote tokia teorija prieštaringa? Jei jums tai pavyktų įrodyti, tai įrodytumėte jau 100 metų egzistuojančią problemą - teorija *ZF* prieštaringa. Visos keturios teorijos surištos viena virve. Jei kuri nors iš jų prieštaringa, tai prieštaringos ir likusios trys.

Keletas pastebėjimų apie aibių teoriją ZF.

- Ji yra aksiominė formalioji *pirmos eolės logikos* matematinė teorija.
- *Neišsprendžiama*. Nėra algoritmo, kuriuo būtų galima bet kurią teorijos formulę patikrinti ar ji išvedama.
- Ji *nepilna.*
- *Neprieštaringumas neįrodytas.*
- *Išvedamų formulių aibė rekusyviai skaiti*.

Į matematikos nagrinėjamus objektus galima žiūrėti kaip į aibes. Teoremas įmanoma suformuluoti kaip teiginius apie aibes. O įrodymus formalizuoti išvedimais aibių teorijoje ZFC. Taip ***ZFC tampa visos matematikos pagrindu***. Bet jau praėjo apie 100 metų, kai aprašyta aibių teorija *ZFC*, bet vis dar neįrodytas jos neprieštaringumas.

Šiuolaikinė matematika pagrįsta **tikėjimu**, kad *ZFC* neprieštaringa.

Vienas iš teiginių įrodytų teorijoje *ZFC*: *rutulį trimatėje Euklido erdvėje galima taip suskaidyti į baigtinį skaičių nepersikertančių aibių, kad, perstumiant aibes, galima gauti du rutulius to paties, kaip ir pradinio, skersmens.*

3.9 Induktyvūs apibrėžimai

Aibes galima nusakyti įvairiais būdais. Pavyzdžiui, *lyginių natūraliųjų skaičių aibė L*. Matematikai greičiausiai rašytų $L = \{n \mid \exists k \in N, n = 2k\}$, logikai formule $\exists x S(x, x, n)$ (čia $S(a, b, c)$ predikatas apibrėžtas natūraliųjų skaičių aibėje ir teisingas tada ir tiktai tada, kai $a + b = c$). Sakysite, kokia čia problema (dalijasi iš dviejų, ar ne). O jei nagrinėjamoje teorijoje nėra dalybos, bet yra sudėtis ir atimtis? Programuojant teoremų išvedimo paiešką, pageidautina, kad iš aprašymo matytusi ir algoritmo žingsniai kaip patikrinti ar *elementas priklauso* aprašomai

aibei. Progaramuojant dažnai naudojami *induktyvūs apibrėžimai*.

Apibrėžiant aibę *A induktyviai* nurodoma:

- **Bazė (pradiniai duomenys.)** Pateikiamas sąrašas aibės *A* elementų.
- **Taisyklės (indukcija)**, kuriomis naudojantis generuojami (gaunami) visi likusieji aibės *A* elementai ir tiktai jie.

Pavyzdžiai.

1. Lyginių natūraliųjų skaičių aibė *L*.

Bazė. $0 \in L$.

Taisyklės. Jei $n \in L$, tai $n + 2$ taip pat priklauso *L*.

Turint tokį apibrėžimą, aišku kaip patikrinti ar *n* yra lyginis. Jei $n = 0$, tai skaičius lyginis. Jei $n > 0$ tai apskaičiuojame $n - 2$. Jei gavom 0, tai *n* lyginis. Jei neigiamas (kai kuriose teorijose rezultatas būtų *neapibrėžta*), tai nelyginis, o, jei teigiamas, tai iš jo atimam 2. Ir kartojam aprašytus žingsnius.

2. Peano aritmetikos natūraliųjų skaičių aibė *PN*.

Bazė. $0 \in PN$.

Taisyklės. Jei $n \in PN$, tai $s(n)$ taip pat priklauso aibei *PN*.

Aprašėme aibę $PN = \{0, s(0), s(s(0)), s(s(s(0))), \ldots$

3. Sudėtis Peano aritmetikoje $+(n, m)$.
Bazė. $+(0, m) = m$.

Taisyklės. $+(s(n), m) = s(+(n, m))$.

3.10 Baigtinumo logika

Teiginys *egzistuoja begalinė aibė* (turima omenyje natūraliųjų skaičių aibę) matematikoje neįrodomas. Tai žinomų logikų B.Russel ir A.Whitehead praeito šimtmečio pradžioje paskelbtas rezultatas.

Priminsime studentams, kad matematikoje tai yra **susitarimas (aksioma)**: *egzistuoja begalinė aibė*.

Tarkime, turime dviem būdais nusakytą funkcijos f apibrėžimo aibę.

1a. Funkcija f apibrėžta bet kurioje baigtinėje natūraliųjų skaičių aibėje.

1b. Funkcija f apibrėžta natūraliųjų skaičių aibėje.

Ar pastebėjote skirtumą tarp 1a ir 1b? Vargu.

Tarkime, turime predikatų logikos formulę F.

2a. Formulė F įvykdoma bet kurioje baigtinėje natūraliųjų skaičių aibėje.

2b. Formulė F įvykdoma natūraliųjų skaičių aibėje.

Pažymėkime *TB* - tapačiai teisingų bet kurioje baigtinėje aibėje (t.y. teisingų bet kurioje struktūroje su baigtine individinių konstantų aibe) predikatų logikos formulių aibę.

Pažymėkime T - aibę predikatų logikos formulių tapačiai teisingų bet kurioje numeruojamoje (t.y. baigtinėje arba skaičiojoje) aibėje. T - pirmos eilės predikatų logikos tapačiai teisingų formulių aibė.

Prisiminkime formulę (žr. 124-125 psl. knygoje [2])

$$\forall x \exists y P(x,y) \wedge \forall x \neg P(x,x) \wedge \forall x \forall y \forall z((P(x,y) \wedge P(y,z)) \rightarrow P(x,z)).$$

Ši formulė įvykdoma natūraliųjų skaičių aibėje ir neįvykdoma jokioje baigtinėje natūraliųjų skaičių aibėje. Taigi, galima rasti tokią formulę, kad teiginys 2a klaidingas, o 2b teisingas. Šis pavyzdys rodo, kad T yra griežtas poaibis *TB*.

$$T \subset TB.$$

Turime matematiką *be begalinės aibės egzistavimo* ir su ja. Kuri matematika sudėtingesnė? Dauguma tvirtins, kad susitarus dėl *begalinės aibės egzistavimo* matematika pasidarė sudėtingesnė.

Logikos požiūriu *begalinės aibės egzistavimo* aksioma pavertė matematiką ne sudėtingesne o **paprastesne, lengvesne**, nes

- aibė *TB* (žr. 182-184 psl. knygoje [2]) nėra rekursyviai skaičioji. Nėra jokio baigtinumo logikos taisyklių sąrašo, naudojantis kuriomis galima būtų išvesti visas *TB* formules (aibė *TB* nėra rekursyviai skaičioji).
  Pastaba. Primenu, kad aibė A vadinama *rekursyviai skaičiąja*, jei **egzistuoja algoritmiškai apskaičiuojama** abipusiai vienareikšmė atitiktis (bijekcija) tarp aibių A ir N (t.y. **galima suprogramuoti bijekciją).**

- aibė T yra rekursyviai skaičioji. Yra aprašyta baigtinė *pirmos eilės logikos* taisyklių aibė, kuri modeliuoja matematiko mąstymą (pavyzdžiui, *natūralioji dedukcija*).

Taigi, prijungdami begalinės aibės egzistavimo aksiomą **matematikai palengvino savo gyvenimą**.

Kuo skiriasi vienas nuo kito teiginiai 1a ir 1b. O gi tuo, kad 1b naudoja *begalinės aibės egzistavimo aksiomą.*

Pastaba. Prisiminkime, kad $L_1 \subset L_2$. Kai sakoma, kad uždavinių klasė aprašoma antros eilės logikos formulėmis, tai turima omenyje, kad *dalis uždavinių* gali būti aprašoma pirmos eilės logikos formulėmis. Bet yra uždavinių toje klasėje, kurie aprašomi antros eilės logikos formulėmis ir neaprašomi jokiomis pirmos eilės logikos formulėmis.

Išbraukime begalybės aksiomą (ji yra 6-toji aksiomų sąraše) iš aibių teorijos *ZFC*. Gauname naują formaliąją teoriją. Įdomu tai, kad Peano aritmetika yra **ZFC ir be begalybės aksiomos** poaibis. Tokioje aibių teorijoje natūralieji skaičiai 0, 1, 2 ir t.t prilyginami termams $\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \ldots$ Apibrėžiamas predikatas $N(a)$, kuriuo nusakome, kad termas *a* yra natūralusis skaičius. Paskesnio nario funkcija $s(a)$ apibrėžiama aibe, kurios elementais yra visi termo *a* elementai ir pats termas *a*. Apibrėžiamos sudėties ir daugybos operacijos bei įrodomos (teorijoje *ZFC be begalybės aksiomos*) visos Peano aritmetikos aksiomos.

Vadinasi, kaip formalioji teorija *ZF*, taip ir *ZFC be begalybės aksiomos* yra *nepilnos*, nes joms priklauso Peano aritmetika.

3.11 Antros eilės logikos skaičiavimas

Kiekviena predikatų logikos formule nusakome aibę tų struktūrų, su kuriomis formulė teisinga.

Paaiškinsime pavyzdžiais.

1.Pirmos eilės logikos formule su lygybės predikatu

$G_n = \exists_1 \ldots \exists x_n(\neg(x_1 = x_2) \wedge \ldots \wedge \neg(x_1 = x_n) \wedge \neg(x_2 = x_3) \wedge \ldots \wedge \neg(x_2 = x_n) \wedge \ldots \wedge \neg(x_{n-1} = x_n)$

nusakome struktūras, kurių individinių konstantų aibėse elementų $\geq n$.

2. Formule $G_n \wedge \neg G_{n+1}$ nusakome *baigtines struktūras*, kuriose lygiai *n* elementų.

Kaip matome, pirmos eilės logikos formulėmis galima nusakyti ir baigtines struktūras. Bet, deja, ne visas. Pavyzdžiui, *individinių konstantų aibė baigtinė* nenusakomas jokia pirmos eilės logikos formule, bet gali būti aprašytas *antros eilės logikos formule.*

Idėja tokia: jei baigtinė aibė $\{a_1, a_2, \ldots, a_n\}$ nėra tuščia, tai bet kurioje jos elementų sekoje, kurioje daugiau kaip *n* narių $\{a_{i_1}, a_{i_2}, \ldots, a_{i_{n+1}} \ldots, \}$ atsiras bent du lygūs elementai. Tai užrašoma formule

$\forall P([(\forall x \forall y \forall z((P(x,y) \wedge P(y,z)) \to P(x,z)) \wedge \forall x \exists y P(x,y)] \to \exists x P(x,x))$.

Antros eilės logika suteikia platesnes galimybes aprašant matematikos teiginius. Formulė (paskesnio nario formulė) $n = m + 1$ užrašoma naudojant tik nelygybės predikatą $<$ pirmos eilės logikos formule

$(m < n) \wedge \neg(\exists x((m < x) \wedge (x < n)))$.

Bet $m < n$ neužrašomas jokia pirmos eilės logikos formule naudojant tik *paskesnio nario formulę*.

$(n = m + 1) \vee \exists P(P(m) \wedge [\forall x \forall y((P(x) \wedge y = x + 1) \to (P(y) \vee n = y + 1)])$.

Standartinės antros eilės logikos su lygybės predikatu skaičiavimo fragmentas.

Pastaba. Nėra tokio sekvencinio skaičiavimo, kuriame būtų išvedamos tapačiai teisingos antros eilės formulės ir tiktai jos. Literatūroje aprašomais skaičiavimais sudaromos galimybės išvesti **tik dalį** tapačiai teisingų antros eilės formulių.

Sintaksė.

- **Suvaržytuosius individinius** kintamuosius žymėsime $x, y, z, u, v, w, x_1, x_2, x_3, \ldots$
- **Laisvuosius individinius** kintamuosius dažniausiai žymėsime $a, b, c, d, e, a_1, a_2, a_3, \ldots$ Galima žymėti ir kitomis raidėmis ar žodžiais. Svarbiausia, kad jie skirtusi nuo suvaržytųjų kintamųjų.
- **Suvaržytuosius predikatinius** kintamuosius žymėsime $X, Y, Z, U, X_1, X_2, X_3, \ldots$
- **Laisvuosius predikatinius** kintamuosius žymėsime $P, Q, R, P_1, P_2, P_3, \ldots$
- Predikatus ir formules žymėsime $F, G, A, B, C, F_1, F_2, F_3, \ldots$

Aksiomos, loginių operacijų bei silpninimo taisyklės yra tos pačios, kaip ir grynosios logikos predikatų skaičiavimo su lygybės predikatu.

Kvantorinės taisyklės individiniams kintamiesiems išlieka tos pačios. O apribojimai keičiasi.

- $(\exists \vdash), (\vdash \forall), (\exists \ldots \exists \vdash), (\vdash \forall \ldots \forall)$ taisyklėse $\beta, \beta_1, \beta_2, \ldots$ yra nauji laisvieji kintamieji.
- taisyklėse $(\vdash \exists)$ ir $(\forall \vdash)$ γ yra **kuris nors** laisvasis kintamasis.

Naujos taisyklės:

Kvantorinės taisyklės predikatiniams kintamiesiems:

$$(\exists \vdash)\ \frac{F(P), \Gamma \vdash \Delta}{\exists X F(X), \Gamma \vdash \Delta} \qquad (\vdash \forall)\ \frac{\Gamma \vdash \Delta, F(P)}{\Gamma \vdash \Delta, \forall X F(X)}$$

Čia *P* **naujas laisvasis** predikatinis kintamasis, kurio vietų skaičius sutampa su *X*.

$$(\forall \vdash)\ \frac{F(G), \forall X F(X), \Gamma \vdash \Delta}{\forall X F(X), \Gamma \vdash \Delta} \qquad (\vdash \exists)\ \frac{\Gamma \vdash \Delta, F(G), \exists X F(X)}{\Gamma \vdash \Delta, \exists X F(X)}$$

Čia $G(a_1, \ldots, a_n)$ **kuris nors** predikatas (arba laisvasis predikatinis kintamasis), kurio laisvųjų kintamųjų skaičius *n* sutampa su $X(b_1, \ldots, b_n)$. Keičiant *X* predikatu *G* visi laisvieji *G* kintamieji pakeičiami atitinkamais *X įeities* laisvaisiais kintamaisiais; a_1 keičiamas į $b_1, \ldots, a_n$ keičiamas į b_n.

Pavyzdys. Formulėje *F* yra dviviečio predikato *X* įeitys: $X(a,b), X(b,a), X(b,b)$ ir *X* keičiamas dviviečiu predikatu $c \leq d$. Tuomet $X(a,b)$ keičiame į $a \leq b$, $X(b,a)$ keičiame į $b \leq a$, $X(b,b)$ keičiame į $b \leq b$.

Čia *G* **kuris nors** predikatas. O tai reiškia, kad jis gali būti ir *neapskaičiuojamas.* Jis gali būti bet koks *įsivaizduojamas* predikatas. Tokie samprotavimai būdingi matematikoje. Tai viena iš priežasčių kodėl neįmanoma aprašyti antros eilės logikos skaičiavimo, kuriame būtų išvedamos tapačiai teisingos antros eilės logikos formulės ir tiktai jos.

Tai kaip tada aprašyti *G*? Dažniausiai tokia kliūtis apeinama taip: apribojama šiose taisyklėse galimų predikatų *G* aibę.
Pavyzdžiui, **mūsų skaičiavime reikalavimai** predikatui *G* tokie: *G yra pirmos eilės logikos su lygybės predikatu formulė* (predikatas). Tuomet formulė susideda iš laisvųjų predikatinių kintamųjų (dažniausiai naudojami tie, kurie įeina į apatinę sekvenciją) ir lygybės predikato.

Tarkime, *P, Q* - predikatinai kintamieji. Kokios eilės formulė $\exists x(P(x) \vee Q(x)) \leftrightarrow (\exists x P(x) \vee \exists x Q(x))$? Sakysite pirmos. Taip, jus teisus (per egzaminą taip ir reikia atsakyti). Bet, formulėje *P, Q* žymi **bet kuriuos** predikatus. Todėl formulė $\exists x(P(x) \vee Q(x)) \leftrightarrow (\exists x P(x) \vee \exists x Q(x))$ išvedama tada ir tiktai tada, kai išvedama antros eilės formulė $\forall P \forall Q(\exists x(P(x) \vee Q(x)) \leftrightarrow (\exists x P(x) \vee \exists x Q(x)))$. Jos deduktyviai ekvivalenčios. Praeito šimtmečio pirmoje pusėje logikai matematikai neskirstė formulių į pirmos ir antros eilės.

Pavyzdžiai.
Taikant taisyklę $(\vdash \exists)$ formulės nekartosime, jei matome, kad jos daugiau neprisireiks. Išvedimuose aišku kurioms formulėms taikomos taisyklės. Todėl formulių paryškinimo nėra.

1. $A(a), B(a) \vdash \exists X \exists Y \exists x(X(x) \wedge Y(x))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A(a), B(a) \vdash A(a)}\oplus \quad \dfrac{}{A(a), B(a) \vdash B(a)}\oplus}{A(a), B(a) \vdash A(a) \wedge B(a)}}{A(a), B(a) \vdash \exists x(A(x) \wedge B(x))}}{A(a), B(a) \vdash \exists Y \exists x(A(x) \wedge Y(x))}}{A(a), B(a) \vdash \exists X \exists Y \exists x(X(x) \wedge Y(x))}$$

2. $\vdash \exists X \exists Y \forall x \forall y(X(x,y) \to Y(y,x))$.

$$\oplus$$
$$\overline{a = b \vdash a = a}$$
$$\overline{a = b \vdash b = a}$$
$$\overline{\vdash a = b \to b = a}$$
$$\overline{\vdash \forall y(a = y \to y = a)}$$
$$\overline{\vdash \forall x \forall y(x = y \to y = x)}$$
$$\overline{\vdash \exists Y \forall x \forall y(x = y \to Y(y, x))}$$
$$\vdash \exists X \exists Y \forall x \forall y(X(x, y) \to Y(y, x))$$

3. $\vdash (a = b) \to \forall X(X(a) \to X(b))$.

$$\oplus$$
$$\overline{a = b, P(b) \vdash P(b)}$$
$$\overline{a = b \vdash P(b) \to P(b)}$$
$$\overline{a = b \vdash P(a) \to P(b)}$$
$$\overline{a = b \vdash \forall X(X(a) \to X(b))}$$
$$\vdash (a = b) \to \forall X(X(a) \to X(b))$$

Deja, formulė $\forall X(X(a) \to X(b)) \to (a = b)$ yra antros eikės logikos tapačiai teisinga, bet neįrodoma aprašytame skaičiavime.

Matematikai teoremų įrodymui dažniausiai naudoja antros eilės logiką **(intuityvią antros eilės logiką)**. Kaip matome, logikai *ne ką ir tegali jiems padėti*. Patarimas, kad taikant taisyklę ($\vdash \exists$) (arba ($\forall \vdash$)) pasirenkamas **kuris nors** predikatas, nelabai guodžiantis. Tik *žinantis matematinę teoriją*, kurios ieškomas išvedimas, gali numatyti kuo reikia keisti kintamuosius taisyklėse ($\vdash \exists$), ($\forall \vdash$). Yra sukurta *interaktyvių* teoremų išvedimo paieškos sistemų, kai taisyklės ($\vdash \exists$), ($\forall \vdash$) taikomos ne automatiškai, o *predikatą ar termą parenka vartotojas*, išmanantis nagrinėjamą matematinę teoriją.
Daugelis žinomų logikų (pavyzdžiui, D.Hilbertas, G.Gentzenas) laikė įrodymus, naudojant antros eilės logiką, **nepatikimais**. Ir ne veltui. Daugumos matematinių teorijų, naudojančių *intuityvią logiką,* vis dar **neįrodytas neprieštaringumas**.

Aukštesnių eilių (trečios, ketvirtos ir t.t.) logikos formulių įvykdomumas redukuojamas į antros eilės logikos formulių įvykdomumą.

3.12 Apie matematiką

Yra įvairių matematikos teorijų bei teoremų skaidymų į klases pagal sudėtingumą. Palyginsime tik tas formalias aksiomines teorijas, kurios buvo nagrinėtos šioje knygoje:

Robinsono aritmetiką (RA), *Peano aritmetiką* (PA), *aritmetiką su rekursyvia supratimo aksioma* RCA_0, *antros eilės aritmetiką* (Z2), *formalią aibių teoriją* (ZF), *formalią aibių teoriją su parinkimo aksioma* (ZFC).

Šias teorijas sieja toks ryšys:

$$\text{RA} \subset \text{PA} \subset \text{RCA}_0 \subset \text{Z2} \subset \text{ZF} \subset \text{ZFC}.$$

Yra žinoma, kad kiekvienam matematikos teiginiui *F* egzistuoja toks analogas, teiginys *G*, priklausantis bent vienai iš mūsų aprašytųjų formalių aksiominių teorijų. *F* ir *G* arba abu įrodomi, arba ne. Kitaip tariant, visa matematika *redukuotina* į aprašytąsias formalias teorijas.

Jei teorema *F* įrodoma kurioje nors iš išvardintų šešių teorijų, tai ji įrodoma ir visose viršaibėse. Kiekvienoje teorijoje (išskyrus RA, pagal mūsų klasifikavimą) egzistuoja teoremos neįrodomos jokiose teorijose poaibėse. Teorija ZFC neturi viršaibio. Ji vadinama *visos matematikos pagrindu*.

Pažvelkime į teorijas, naudojantis kitais kriterijais:

Teorija	**Naudojama logika**	**Ar įrodytas teorijos neprieštaringumas?**
RA	Pirmos eilės	Taip
PA	Pirmos eilės	Taip
RCA_0	Antros eilės	Taip
Z2	Antros eilės	Ne
ZF	Pirmos eilės	Ne
ZFC	Antros eilės	Ne

Nesuklydau. *ZF* yra pirmos eilės teorija.

B.Russel: *Matematikai naudojasi jiems suteikta laisve aksiomų pasirinkimui. Bet nebeįmanoma atskirti **laisvės** nuo **viskas leistina**.*

A.Puankare: *Bet kurios mokslo teorijos esmė - sąlyginiai susitarimai. Bet jiems keliama viena privaloma sąlyga - **įrodyti**, kad teorija **netampa prieštaringa.***

Vienas iš Hilberto programos punktų, visos matematikos aksiomatizavimas, įvykdytas. Dabar labai aiškiai matosi, ką sutarta, be įrodymo, laikyti teisingais teiginiais (aksiomomis).

Priminsime kai kurių teorijų aksiomas.

1. RA (Robinsono aritmetika).
7-ios aksiomos. Jomis nusakomos aritmetinės funkcijos: sudėtis ir daugyba natūraliųjų skaičių aibėje. Robinsono aritmetikoje definuojamos visos rekursyviosios funkcijos ir atvirkščiai, kiekviena Robinsono aritmetikos funkcija yra rekursyvioji.
Kiekvieną *RA* funkciją galima *suprogramuoti*. Teorija *neprieštaringa.*

2. PA (Peano aritmetika).
7-ios RA aksiomos $+$ **indukcijos aksioma** $(\mathbf{F(0)} \wedge \forall \mathbf{x}(\mathbf{F(x)} \rightarrow \mathbf{F(s(x))})) \rightarrow \forall \mathbf{x F(x)}$.
Prijungę indukcijos aksiomą gauname teoriją, kurioje įrodomos ir rekursyviųjų funkcijų savybės. Pavyzdžiui, $\forall x \forall y(x + y = y + x)$.
Kiekvieną *PA* funkciją galima *suprogramuoti.* Teorija *neprieštaringa.*

3. RCA_0 (aritmetika su *supratimo* aksioma).
7-ios RA aksiomos + indukcijos aksioma + supratimo aksioma $\exists \mathbf{X} \forall \mathbf{x}(\mathbf{X(x)} \leftrightarrow \mathbf{F(x)})$.

Aksiomose visi formulių kvantoriniai kompleksai, suvaržantys *individinius* formulių kintamuosius, *apriboti termais.* Aritmetikos formulės $F(a)$ (su vienu laisvu kintamuoju) su kuriomis *F* teisinga, aibė yra rekursyviai skaiti. Taigi, indukcija taikoma tik rekursyviai skaičioms aibėms, o supratimo aksioma tik rekursinėms (išsprendžiamoms) aibėms.
Kiekvieną RCA_0 funkciją galima *suprogramuoti.* Teorija *neprieštaringa*.

4. Z2 (antros eilės aritmetika).
7-ios RA aksiomos + indukcijos aksioma + supratimo aksioma be apribojimų.
Taikydami supratimo aksiomą (axiom of comprehension) gaunamos ir aibės, kurios nei rekursinės, nei rekursyviai skaičiosios. Nes, pagal šią aksiomą rekursyviai skaičiųjų aibių papildiniai išvedami šioje teorijoje. Šia aksioma į apyvartą išleidžiamos ir nesuprogramuojamos funkcijos. Remiantis supratimo aksioma gaunamos ir begalinės aibės. Kitaip tariant, šiai teorijai nereikia begalybės aksiomos.
Suprogramuoti galima **ne visas** Z2 funkcijas. Teorijos **neprieštaringumas neįrodytas**.

5. Z2C (antros eilės aritmetika su parinkimo aksioma.)
Z2 aksiomos + parinkimo aksioma.
Laikoma, kad šioje teorijoje galima įrodyti absoliučią daugumą matematikos teoremų. *Parinkimo aksioma* ekvivalenti *Zermelo aksiomai*: bet kurioje aibėje galima įvesti visišką tvarką. Pavyzdžiui, matematikoje laikoma (išplaukia iš parinkimo aksiomos), kad egzistuoja visiška tvarka (<) realiųjų skaičių aibėje *R*.

Reziumė.
Tokia matematika (su supratimo ir parinkimo aksiomomis) sukurta dar tada, kai nebuvo žinoma nei apie rekursyviąsias funkcijas, nei apie kompiuterius.

Dabar, kai matematika yra aksiomatizuota, tapo aišku kokie susitarimai (teiginiai laikomi teisingais be įrodymo, aksiomomis), naudojami įrodymuose, išveda iš suprogramuojamų funkcijų klasės. Tai: 1) **begalybės**, 2) **supratimo be apribojimų** bei 3) **parinkimo** aksiomos. **Visose teorijose**, naudojančiose bent vieną iš šių aksiomų, **neprieštaringumas neįrodytas iki šiolei.**
Ar šios aksiomos tik nėra fantazijos vaisius? Sunku pasakyti. Teiginiai aprašyti supratimo bei parinkimo aksiomomis yra teisingi baigtinių aibių atveju. Bet, toks jau įprotis, kad tai, kas galioja baigtinėms aibėms, norisi pratęsti ir begalinėms. O ten kitoks pasaulis, su kita valdžia ir kitokiais įstatymais.
Klausimas: ar kiekvieną intuityvų įrodymą galima transformuoti į formalia logika pagrįstą įrodymą?

3.13 Pratimai

1. Rasti antros eilės logikos sekvencijos išvedimą:
a) $\vdash (\exists x P(x) \wedge \neg \forall x P(x)) \rightarrow \exists X(\exists x X(x) \wedge \exists x \neg X(x))$,
b) $\exists X \exists x \neg X(x), \forall X \forall x X(x) \vdash$,
c) $\vdash \forall x(a = x) \rightarrow \forall X(X(a) \rightarrow \forall x X(x))$.

4 Skyrius
LOGINIS PROGRAMAVIMAS

Šiame skyriuje aprašysime logikos rezultatus, kuriais remiantis sukurtos loginio (žinomiausias atstovas Prolog) bei teoremų išvedimo paieškai skirtos (žinomiausias atstovas Coq) programavimo kalbos.

4.1 Prieštaringos aibės

Uždavinius ankstesniuose skyriuose užrašydavome sekvencijų $A_1, \ldots, A_n \vdash B$ pavidalu. Ji išvedama tada ir tiktai tada, kai $(A_1 \wedge \ldots \wedge A_n) \rightarrow B$ tapačiai teisinga formulė. Aprašysime kitokį metodą, kuriuo naudojantis paprasčiau suprogramuoti išvedimo paiešką. Reikės nustatyti ar *formulių aibė prieštaringa*. Remiantis šiuo metodu sukurta programavimo kalba *Prolog* (PROgramming in LOGic). Šiame skyrelyje metodą iliustruosime teiginių logikos formulėmis, kuriose yra tik loginės operacijos $\neg, \wedge, \vee, \rightarrow$, o kitame aprašysime predikatų logikos formulėms.

Sakoma, kad formulių aibė $\{A_1, \ldots, A_n\}$ prieštaringa, jei, kokia bebūtų interpretacija, atsiras bent viena klaidinga formulė.

Pavyzdžiai:

1. $\{p \wedge q, \neg p, r \rightarrow p$. Aibėje trys formulės. Ji prieštaringa, nes, kai $p = t$, antroji formulė klaidinga, o kai $p = k$, pirmoji klaidinga.

2. $\{p \rightarrow q, q \rightarrow r, r \rightarrow p\}$. Aibė neprieštaringa. Pakanka nurodyti bent vieną interpretaciją, su kuria visos formulės teisingos. Kai $p = q = r = t$, visos formulės teisingos.

3. $\{\neg((p \wedge q) \rightarrow p)\}$. Aibėje viena formulė. Aibė prieštaringa, kai joje viena formulė, tik tuo atveju, kai formulė yra tapačiai klaidinga. Šiuo atveju aibė prieštaringa.

Atkreipiame dėmesį, kad aibė $\{A_1, \ldots, A_n\}$ prieštaringa tada ir tiktai tada, kai formulė $A_1 \wedge \ldots \wedge A_n$ tapačiai klaidinga (kablelis aibėje atitinka konjunkciją formulėje). Primename, kad formulė *F* tapačiai teisinga tada ir tiktai tad, kai $\neg F$ tapačiai klaidinga. 1930 m. prancūzų logikas J.Herbrand'as aprašė algoritmą, kuriuo greičiau nustatomas formulės $\neg F$ tapačiai klaidingumas, negu *F* tapačiai teisingumas žinomais metodais. 1931 m. J.Herbrand'as žuvo, būdamas 23 metų, bekopdamas į Alpių viršūnę.

Aprašysime aibės $\{A_1, \ldots, A_n\}$ prieštaringumo nustatymo metodą. Bet prieš tai, aibės formules transformuosime į *disjunktų* aibes.

Loginį kintamąjį bei loginio kintamojo neigimą vadiname **litera**. Pavyzdžiui, $p, \neg q, \neg r$ yra literos.

*Literų disjunkciją $l_1 \vee l_2 \vee \ldots \vee l_n$ vadiname **disjunktu**, o skaičių n - jo ilgiu. **Konjunktu** vadiname literų konjunkciją.*

Pavyzdžiai:

Turime 6 formules

$$p \vee \neg r, \;\; p \wedge \neg r, \;\; p \vee \neg q \vee r, \;\; q, \;\; \neg p, \;\; r \wedge p \wedge q.$$

1, 3, 4 ir 5 formulės yra disjunktai. 2, 4, 5 ir 6 formulės yra konjunktai. Kaip matome, 4 ir 5 formulės yra ir disjunktai, ir konjunktai.

Kadangi $l \vee l \equiv l, l \wedge l \equiv l$ (l - kuri nors litera), tai laikysime, kad disjunktuose ir konjunktuose yra tik po vieną skirtingų literų įeitį. Pavyzdžiui, vietoje konjunkto $p \wedge \neg r \wedge \neg q \wedge p \wedge \neg q \wedge \neg r$, nagrinėsime jam ekvivalentų $p \wedge \neg q \wedge \neg r$.

Formules pavidalo $D_1 \wedge D_2 \wedge \ldots \wedge D_n$ (čia $D_i \;\; (i = 1, \ldots, n)$ - disjunktai) vadiname **normaliąja konjunkcine forma** (*NKF*).

Pavyzdžiai:

$$(\neg p \vee \neg q) \wedge (p \vee \neg q \vee \neg r) \wedge \neg p \wedge (p \vee \neg p),$$
$$p \vee \neg q \vee r \text{ (šiuo atveju } n = 1),$$
$$\neg p \wedge \neg q \wedge \neg r.$$

Paaiškinsime kaip kiekvieną formulę galima transformuoti į normaliąją konjunkcinę formą. Tuo tikslu aprašysime du transformavimo būdus: *transformavimas naudojantis teisingumo lentele* ir *transformavimas naudojantis ekvivalenčiomis formulėmis.*

1. *Transformavimas į NKF naudojantis teisingumo lentelėmis.* Tarkime yra formulė $F(p_1, \ldots, p_n)$ su *n* loginiais kintamaisiais. Sudarome teisingumo lentelę. Joje bus 2^n interpretacijų. Kiekvienai interpretacijai, su kuria *F* klaidinga, priskiriame po disjunktą $l_1 \vee l_2 \vee \ldots \vee l_n$:

$$l_i = \neg p_i, \text{ jei } p_i = t,$$
$$l_i = p_i, \text{ jei } p_i = k.$$

Jei sudarytumėm $l_1 \vee l_2 \vee \ldots \vee l_n$ teisingumo lentelę, tai joje būtų tik viena interpretacija su kuria ji klaidinga. Būtent ta, su kuria *F* klaidinga, ir pagal kurią suformavom disjunktą. Konjunkcija tokių disjunktų yra formulė normaliosios konjunkcinės formos ekvivalenti formulei $F(p_1, \ldots, p_n)$. Jei *F* tapačiai teisinga, tai jos NKF laikome $p_i \vee \neg p_i$. Tai susitarimo reikalas. Kai kuriuose vadovėliuose reikalaujama, kad *F* nebūtų tapačiai teisinga (tik tuomet ją galima transformuoti į NKF).

Pavyzdys.

p	q	r	$F(p,q,r)$
t	t	t	t
t	t	k	k
t	k	t	k
t	k	k	t
k	t	t	k
k	t	k	t
k	k	t	t
k	k	k	t

Formulė klaidinga su trimis interpretacijomis:

1) $p = q = t, r = k$,

2) $p = r = t, q = k$,

3) $q = r = t, p = k$.

Pirmajai interpretacijai priskiriame disjunktą $\neg p \vee \neg q \vee r$, antrąjai - $\neg p \vee q \vee \neg r$, trečiąjai - $p \vee \neg q \neg r$. . Formulės NKF yra

$$(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (p \vee \neg q \vee \neg r).$$

Formulės $F(p_1, \ldots, p_n)$ **normalioji konjunkcinė forma** $D_1 \wedge D_2 \wedge \ldots \wedge D_n$ vadinama **tobuląja**, jei kiekviename disjunkte $D_i \ \ (i = 1, \ldots, n)$ yra arba p_j, arba $\neg p_j \ \ (j = 1, \ldots, n)$.

Atkreipiame dėmesį, kad transformavimo metodas, *naudojantis teisingumo lentelę,* suformuoja tobulą normaliąją formą, išskyrus atvejį, kai formulė yra tapačiai teisinga.

2. *Transformavimas į NKF naudojantis ekvivalenčiomis formulėmis.*

Transformavimas nusakomas keturiais algoritmo žingsniais: eliminavimas, neigimo įkėlimas į skliaustus, distributyvumo dėsnio taikymas, prastinimas.

Eliminavimas. Kadangi nagrinėjame formules, kuriose yra tik loginės operacijos $\neg, \wedge, \vee, \rightarrow$, o formulėse normaliosios konjunkcinės formos gali būti tik operacijos $\neg, \wedge, \vee$, tai, visų pirma, reikia eliminuoti implikaciją (jei ji yra nagrinėjamoje formulėje). Naudojamės ekvivalentumu $p \rightarrow q \equiv \neg p \vee q$.

Neigimo įkėlimas į skliaustus. Šio žingsnio tikslas - gauti formulę, kurioje neigimas būtų tik prieš loginius kintamuosius. Naudojamės ekvivalentumais

$$\neg(p \wedge q) \equiv \neg p \vee \neg q,$$
$$\neg(p \vee q) \equiv \neg p \wedge \neg q,$$
$$\neg\neg p \equiv p.$$

Distributyvumo dėsnio taikymas. Taikant ekvivalentumą

$$(p \wedge q) \vee r \equiv (p \vee r) \wedge (q \vee r)$$

galima gauti formulę, kuri ir yra NKF.

Prastinimas. Taikome ekvivalentumus:

$$p \vee p \equiv p,$$
$$p \wedge p \equiv p.$$

Jei yra poformuliai pavidalo $p \wedge \neg p, p \vee \neg p$, tai keičiame konstantomis *k, t* ir jas eliminuojame. Prastinti galima ir po pirmojo bei antrojo žingsnių.

Pavyzdys. $(p \vee q) \rightarrow (r \vee \neg p)$.

Sprendimas.

Elimminuojame implikaciją: $\neg(p \vee q) \vee (r \vee \neg p)$.

Įkeliame neigimą į skliaustus: $(\neg p \wedge \neg q) \vee (r \vee \neg p)$.

Taikome distributyvumo dėsnį: $(\neg p \vee r \vee \neg p) \wedge (\neg q \vee r \vee \neg p)$.

Prastiname: $(\neg p \vee r) \wedge (\neg q \vee r \vee \neg p)$.

Formulės gali turėti ne vieninteles normaliąsias konjunkcines formas. Gautoji NKF $(\neg p \vee r) \wedge (\neg q \vee r \vee \neg p)$ nėra vienintelė. Pavyzdžiui, $(\neg p \vee r \vee \neg q) \wedge (\neg p \vee r \vee q) \wedge (\neg q \vee r \vee \neg p)$ taip pat yra tos pačios formulės $(p \vee q) \rightarrow (r \vee \neg p)$ NKF.

Grįšim prie pagrindinio tikslo - aibės prieštaringumo nustatymo metodo. Tik dabar nagrinėsime aibes, kurios elementais yra disjunktai. Užduotis (problemas) įprasta užrašyti sekvencijų $A_1, \ldots, A_n \vdash B$ pavidalu, t.y. nurodant prielaidas (tai, kas yra duota $A_1, \ldots, A_n$) ir tai, ką reikia įrodyti (formulė **B**). Sekvencija įrodoma tada ir tiktai tada, kai $(A_1 \wedge \ldots \wedge A_n) \rightarrow B$ tapačiai teisinga formulė. O ji tapačiai teisinga tada ir tiktai tada, kai jos neigimas $\neg((A_1 \wedge \ldots \wedge A_n) \rightarrow B)$ yra tapačiai klaidinga formulė.

$$\neg((A_1 \wedge \ldots \wedge A_n) \rightarrow B) \equiv$$
$$\neg(\neg(A_1 \wedge \ldots \wedge A_n) \vee B) \equiv$$
$$A_1 \wedge \ldots \wedge A_n \wedge \neg B.$$

Formulė $A_1 \wedge \ldots \wedge A_n \wedge \neg B$ tapačiai klaidinga tada ir tiktai tada, kai formulių aibė $\{A_1, \ldots, A_n, \neg B\}$ prieštaringa.

*Prieštaringumo nustatymo algoritmo **pirmas žingsnis** - transformuoti formules $A_1, \ldots, A_n, \neg B$ į NKF ir iš jų suformuoti disjunktų aibę.*

Sekvencijos $A_1, \ldots, A_n \vdash B$ transformavimo į disjunktų aibę pavyzdžiai:

1. $p \rightarrow (r \wedge q), \neg p \rightarrow \neg q \vdash q \rightarrow r$.

Sprendimas.

Transformuojame prielaidas (antecedente esančias formules) į NKF:

$$p \to (r \wedge q) \equiv \neg p \vee (r \wedge q) \equiv (\neg p \vee r) \wedge (\neg p \vee q),$$

$$\neg p \to \neg q \equiv p \vee \neg q.$$

Transformuojame išvados neigimą (sukcedente esančios formulės neigimą):

$$\neg(q \to r) \equiv \neg(\neg q \vee r) \equiv q \wedge \neg r.$$

Dabar galim suformuoti disjunktų aibę (pažymėkime ją raide *S*).

Atsakymas. $S = \{\neg p \vee r,\ \neg p \vee q,\ p \vee \neg q,\ q,\ \neg r\}$.

2. $(p \to r) \to (q \to r) \vdash (p \wedge q) \to r$.

Sprendimas:

$(p \to r) \to (q \to r) \equiv \neg(\neg p \vee r) \vee (\neg q \vee r) \equiv (p \wedge \neg r) \vee (\neg q \vee r) \equiv (p \vee \neg q \vee r) \wedge (\neg r \vee \neg q \vee r) \equiv$
(formulė $\neg r \vee r$ ekvivalenti (lygi) konstantai *t*)
$\equiv (p \vee \neg q \vee r) \wedge (\neg q \vee t) \equiv$
(formulė pavidalo $F \vee t$ yra teisinga)
$\equiv (p \vee \neg q \vee r) \wedge t \equiv$
(formulė $F \wedge t \equiv F$)
$\equiv p \vee \neg q \vee r$.

$$\neg((p \wedge q) \to r) \equiv \neg(\neg(p \wedge q) \vee r) \equiv p \wedge q \wedge \neg r.$$

Atsakymas: $S = \{p \vee \neg q \vee r, p, q, \neg r\}$.

3. Nagrinėjame funkciją *f* apibrėžtą intervale [*a, b*]. *Jei funkcija diferencijuojama* (*d*), *tai ji tolydi* (*l*) *ir aprėžta* (*a*). Vadinasi, *jei funkcija nediferencijuojama, tai ji nėra tolydi arba ji nėra aprėžta intervale.*
Aprašysime samprotavimą disjunktų aibe, kurios prieštaringumą reikia nustatyti.

$$d \to (l \wedge a) \vdash (\neg d \to \neg l) \vee (\neg d \to \neg a).$$

Sprendimas:
$d \to (l \wedge a) \equiv \neg d \vee (l \wedge a) \equiv (\neg d \vee l) \wedge (\neg d \vee a)$.

$\neg((\neg d \to \neg l) \vee (\neg d \to \neg a)) \equiv \neg((d \vee \neg l) \vee (d \vee \neg a)) \equiv \neg(d \vee \neg l) \wedge \neg(d \vee \neg a) \equiv \neg d \wedge l \wedge \neg d \wedge a \equiv \neg d \wedge l \wedge a$.

Atsakymas: $S = \{\neg d \vee l, \neg d \vee a, \neg d, l, a\}$.

Pavyzdžiais paaiškinome kaip sudaromos disjunktų aibės. O kaip nustatyti ar disjunktų aibė prieštaringa? 1965 metais amerikiečių logikė J.Robinson aprašė prieštaringumo nustatymo algoritmą, kurį pavadino **rezoliucijų metodu.** Greitai jis tapo populiarus, pagrindiniu metodu prieštaringumo nustatymui. Šiuo metodu remiasi daugelis sukurtų loginio programavimo kalbų - *Prolog, Molog* ir kitos.

Paaiškinsime rezoliucijų metodą ir panaudosime jį prieštaringumo nustatymui aprašytuose pavyzdžiuose. Tarkime trijų kintamųjų *p, q, r* formulė *F* tapačiai klaidinga. Tuomet jos teisingumo lentelė tokia:

p	q	r	$F(p,q,r)$
t	t	t	k
t	t	k	k
t	k	t	k
t	k	k	k
k	t	t	k
k	t	k	k
k	k	t	k
k	k	k	k

Formulės $F(p,q,r)$ tobula NKF:

$(\neg p \vee \neg q \vee \neg r) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (p \vee \neg q \vee \neg r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee \neg r) \wedge (p \vee q \vee r)$.

Disjunktų aibė:

$S = \{\neg p \vee \neg q \vee \neg r,\ \neg p \vee \neg q \vee r,\ \neg p \vee q \vee \neg r,\ \neg p \vee q \vee r,\ p \vee \neg q \vee \neg r,\ p \vee \neg q \vee r,\ p \vee q \vee \neg r,\ p \vee q \vee r$.

Tobulos NKF kiekviename disjunkte yra visi trys kintamieji *p, q, r.* Tapačiai klaidingos formulės NKF turi $2^3 = 8$ disjunktus.

Prastinsime aibės *S* disjunktus naudodami ekvivalentumą $(C \vee p) \wedge (C \vee \neg p) \equiv C$. Čia *C* kuris nors disjunktas (gali būti ir tuščias; tuščią disjunktą žymėsime $\square$), o *p* kuris nors loginis kintamasis. T.y. disjunktų porą $C \vee p, C \vee \neg p$ keičiame disjunktu *C*. Pavyzdžiui, aibėje *S* išbraukiame disjunktus $\neg p \vee \neg q \vee \neg r, \neg p \vee \neg q \vee \neg r$ ir pakeičiame juos disjunktu $\neg p \vee \neg q$. Jei *C* yra tuščias disjunktas, tuomet turime kurį nors kintamąjį ir jo neigimą, pavyzdžiui, $q, \neg q$. Porą $q, \neg q$ keičiame tuščiu disjunktu $\square$. $q \wedge \neg q \equiv \square$. Kadangi taip pat $q \wedge \neg q \equiv k$, tai, turbūt supratote, kad rezoliucijų metode tuščio disjunkto simboliu žymima loginė konstanta *k* (*klaidinga*).

Vadinasi, jei turite tapačiai klaidingą formulę, tai, parašę pagal tobulą NKF disjunktų aibę ir, *naudodami tik prastinimo taisyklę*, gausite tuščią disjunktą $\square$.

Sudarykime iš aibės *S* keturias poras disjunktų $(\neg p \vee \neg q \vee \neg r,\ \neg p \vee \neg q \vee r),\ (\neg p \vee q \vee \neg r,\ \neg p \vee q \vee r),\ (p \vee \neg q \vee \neg r,\ p \vee \neg q \vee r),\ (p \vee q \vee \neg r,\ p \vee q \vee r)$

Taikome prastinimo taisyklę poroms atžvilgiu vieno ir to paties kintamojo *r*. Gauname keturius disjunkus. Iš jų sudarome dvi poras $(\neg p \vee \neg q, \neg p \vee q), (p \vee \neg q, p \vee q)$.

Šį kartą taikome prastinimo taisyklę poroms atžvilgiu vieno ir to paties kintamojo q. Gauname du disjunkus $\neg p, p$. Taikydami prastinimo taisyklę, iš jų gauname tuščią disjunktą.

Tarkime, turime trijų kintamųjų tobulą formulės NKF. Jei joje aštuoni skirtingi disjunktai, tai ji tapačiai klaidinga, o, jeigu mažiau, tai ji nėra tapačiai klaidinga (priminsime, kad tapačiai teisinga formulė taip pat yra ir įvykdoma). Ir, bendru atveju, jei formulės su *n* kintamaisiais tobuloje NKF mažiau nei 2^n skirtingų disjunktų, tai ji įvykdoma. Taigi, jei turime tobulą NKF, tereikia suskaičiuoti kiek turime disjunktų ir nustatysime ar ji tapačiai klaidinga.

Bendru atveju, kai disjunktų aibė *S* sudaryta ne tik iš kurios nors formulės tobulos NKF disjunktų, taikoma ne aprašytoji *prastinimo taisyklė*, o išvedimo taisyklė. Ji vadinama **atkirtos taisykle**. Ji taikoma dviem disjunktams, o rezultatas - vienas disjunktas. Taisyklė tokia:

$$\frac{C_1 \vee p \vee C_2, \quad C_3 \vee \neg p \vee C_4}{C_1 \vee C_2 \vee C_3 \vee C_4}$$

Disjunktus žymime $C, C_0, C_1, \ldots$ Tuščias disjunktas žymimas $\Box$. Taisyklę sutrumpintai žymėsime AT. Kaip matome, taisyklę galima taikyti tik tuo atveju, kai viename disjunktų yra kurio nors loginio kintamojo įeitis, o antrajame - jo neigimas, Taisyklėje toks kintamasis pažymėtas raide *p*. Kadangi literų tvarka disjunktuose nesvarbi $(p \vee q \equiv q \vee p)$, tai atkirtos taisyklę rašysime taip:

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

Primename, kad disjunktus nagrinėjame tik pavidalo, kuriuose bet kurio kintamojo įeitis yra tik viena. Pavyzdžiui, vietoje $\neg p \vee q \vee \neg p \vee q$ nagrinėjame $\neg p \vee q$.

Sakome, kad disjunktas C išvedamas iš disjunktų aibės S (žymime $S \vdash C$), *jei yra tokia baigtinė disjunktų seka* $C_1, \ldots, C_n$, *kurioje kiekvienas* C_i $(i = 1, \ldots, n)$ *arba priklauso aibei S, arba gautas iš kairėje jo stovinčių disjunktų pagal atkirtos taisyklę. Be to,* $C_n = C$.

Teiginių logikos **rezoliucijų** metodo žingsniai:

1. formulės ar sekvencijos transformavimas į disjunktų aibę,

2. tuščio disjunkto paieška naudojant atkirtos taisyklę.

Išspręsime tris aukščiau aprašytus pavyzdžius. Skliaustuose nurodome, kaip gautas disjunktas, t.y., ar jis priklauso pradinei aibei *S*, ar gautas pagal atkirtos taisyklę (AT).

1. $S = \{\neg p \vee r, \ \neg p \vee q, \ p \vee \neg q, \ q, \ \neg r\}$. Parodysime, kad $S \vdash \Box$.

$p \vee \neg q(S), \ q(S), \ p(AT), \ \neg p \vee r(S), \ r(AT), \ \neg r(S), \ \Box(AT)$.

Disjunktų išvedimai aprašomi ir kitokiais būdais. Naudokite tą, kuris jums priimtinesnis.

Išvedimas kaip *taisyklių taikymo seka*:

$$\frac{p \vee \neg q \;\; q}{p} \qquad \frac{\neg p \vee r \;\; p}{r} \qquad \frac{r \;\; \neg r}{\square}$$

Atsakymas: išvada teisinga.

*Įrodyta, kad formulių aibė **prieštaringa** tada ir tiktai tada, kai $S \vdash \square$.*

2. $S = \{p \vee \neg q \vee r, p, q, \neg r\}$.
Neišvedamas tuščias disjunktas, nes taikydami atkirtos taisyklę, galim gauti tik šiuos naujus disjunktus: $p \vee r, p \vee \neg q$. Tarp išvedamų disjunktų nėra tuščio.

Suprogramavus paiešką rezoliucijų metodu, generuojami išvedami disjunktai. Jei randamas tuščias disjunktas, tai aibė prieštaringa ir išvada teisinga. Skirtingų išvedamų disjunktų yra baigtinis skaičius. Jei tarp jų nėra tuščio, vadinasi *aibė nėra prieštaringa*, ji įvykdoma.

Tuščias disjunktas neišvedamas tada ir tiktai tada, kai aibė *S* įvykdoma. Dažniausiai, kad aibė įvykdoma, pavyksta greičiau randant interpretaciją, su kuria visi aibės disjunktai teisingi. Pratybų metu, kai paieška vykdoma ne kompiuteriu, galima ir taip įrodyti aibės įvykdomumą.

Mūsų atveju, kai $p = q = t, r = k$ visos aibės formulės teisingos.

Atsakymas: *išvada klaidinga*.

3. $S = \{\neg d \vee l, \neg d \vee a, \neg d, l, a\}$. Su interpretacija $d = k, l = a = t$ visos aibės *S* formulės teisingos. *Aibė nėra prieštaringa*.

Atsakymas: *išvada klaidinga.*

Atkreipiame dėmesį, kad iš aibės $\{p \vee q, \neg p \vee \neg q\}$ *neišvedamas tuščias disjunktas*. Ji įvykdoma, pavyzdžiui, su interpretacija $p = t, q = k$.

Disjunktai pavidalo $\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n \vee q$, $\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n$, q vadinami **Horno disjunktais.** Pirmąjį galima užrašyti formule pavidalo $(p_1 \vee p_2 \vee \ldots \vee p_n) \rightarrow q$. Čia $p_1, p_2, \ldots, p_n, q$ yra loginiai kintamieji.

Užduotis, parašytas Horno disjunktais, lengviau suprogramuoti.

Formulės pavidalo $D_1 \wedge D_2 \wedge \ldots \wedge D_m$ vadinamos **Horno formulėmis**, jei D_i $(i = 1, \ldots, m)$ yra Horno disjunktai

4.2 Rezoliucijų metodas

Preitame skyrelyje aprašėme rezoliucijų metodą teiginių logikai. Šiame skyrelyje aprašysime rezoliucijų metodą predikatų logikai. Metodas taikomas tam tikro pavidalo formulių aibei. Visų

pirma paaiškinsime kas tai per aibė ir kaip į ją transformuoti sekvenciją $A_1, \ldots, A_n \vdash B$, kurioje $A_1, \ldots, A_n, B$ predikatų logikos formulės. Priminsime, kad sekvencija $A_1, \ldots, A_n \vdash B$ išvedama tada ir tiktai tada, kai aibė $\{A_1, \ldots, A_n, \neg B\}$ prieštaringa. Rezoliucijų metodu nustatysime ar aibė $\{A_1, \ldots, A_n, \neg B\}$ prieštaringa. Tik aibės formules prieš tai transformuosime į tam tikrą pavidalą.

Pirmas žingsnis. Visos aibės formulės transformuojamos į normaliąsias priešdėlines formas.

Antras žingsnis. Skulemizavimas. Egzistavimo kvantorių eliminavimas. Jį taikome normaliosios priešdėlinės formos formulėms.

Formule $\forall x \exists y P(x, y)$ tvirtiname, kad kiekvienai *x* reikšmei galima rasti tokią *y* reikšmę (t.y. *y* yra funkcija nuo *x*), kad $P(x, y) = t$. Tai reiškia, kad $\forall x \exists y P(x, y)$ įvykdoma tada ir tiktai tada, kai įvykdoma $\forall x P(x, f(x))$. Šioje formulėje nebėra egzistavimo kvantoriaus.

Formulė $\forall x \exists y P(x, y)$ įvykdoma, pavyzdžiui, struktūroje:

1. natūraliųjų skaičių aibė yra individinių konstantų aibė,
2. $P(x, y) = x < y$.

Formulė $\forall x P(x, f(x))$ taip pat teisinga šioje struktūroje. Tereikia konkrečiai nurodyti kaip parenkamas $f(x)$ (vietoje *y*), turint *x* reikšmę. Pavyzdžiui, funkcija *f* galėtų būti $x + 1$.

Paaiškinsime egzistavimo kvantorių eliminavimą bendru atveju. Turime formulę normaliosios priešdėlinės formos $Q_1 x_1 \ldots Q_m x_m G(x_1, \ldots, x_m)$.

Ieškome prefikse pirmojo, iš kairės į dešinę, egzistavimo kvantoriaus. Tarkime, pirmuoju yra Q_i, t.y. $Q_i = \exists$. Vadinasi $Q_1 = Q_2 = \ldots = Q_{i-1} = \forall$. Išbraukiame prefikse $Q_i x_i$. Formulės matricoje visas x_i įeitis keičiame funkcija *. f* yra naujas, nepriklausantis nagrinėjamai formulei, funkcinis simbolis. Gauname formulę

$$\forall x_1 \ldots \forall x_{i-1} Q_{i+1} x_{i+1} \ldots Q_m x_m G(x_1, \ldots, x_{i-1}, f(x_1, \ldots, x_{i-1}), x_{i+1}, \ldots, x_m).$$

Taip keičiami visi formulės egzistavimo kvantoriai.

Pavyzdžiai.

Skulemizuoti formules:

1. $F = \forall x \exists y \forall z \exists u G(x, y, z, u)$. Formulėje yra $\exists y$ ir $\exists u$. Eliminuosime šiuos kvantorinius kompleksus. *y* pakeisime $f(x)$, o *u* - $g(x, z)$. Čia *f, g* nauji funkciniai simboliai. *y* pakeisime $f(x)$, nes prieš $\exists y$ prefikse yra vienas kvantorinis kompleksas prasidedantis $\forall$, t.y. kvantorinis kompleksas prasidedantis bendrumo kvantoriumi. O prieš $\exists u$ - du $\forall x, \forall z$. Gavome skulemizuotą formulę.

$$\forall x \forall z G(x, f(x), z, g(x, z)).$$

2. $F = \exists x_1 \exists x_2 \forall x_3 \forall x_4 \exists x_5 G(x_1, x_2, x_3, x_4, x_5)$. Formulėje yra $\exists x_1, \exists x_2, \exists x_5$.

x_1 keičiame konstanta *a*,

x_2 keičiame konstanta *b*,

x_5 - $f(x_3, x_4)$.

x_1, x_2 keičiame kuriais nors naujais (jų neturi būti formulėje *F*) laisvaisiais kintamaisiais (galėjome parinkti ir kitus, nebūtinai *a, b*, iš laisvųjų kintamųjų sąrašo). Priminsime, kad suvaržytuosius kintamuosius žymime $x, y, z, u, v, w, x_1, x_2, x_3, \ldots$, o laisvuosius kintamuosius $a, b, c, d, e, a_1, a_2, \ldots f$ - naujas funkcinis simbolis. Gauname skulemizuotą formulę:

$$\forall x_3 \forall x_4 G(a, b, x_3, x_4, f(x_3, x_4)).$$

Bet kuri normaliosios priešdėlinės formos formulė F deduktyviai ekvivalenti skulemizuotai F, t.y. F ir jai skulemizuota abi yra išvedamos, arba abi nėra išvedamos.

Trečias žingsnis. Skulemizuotų formulių transformavimas į NKF (normaliąją konjunkcinę formą).

Skulemizuotosios $\forall x_1 \forall x_2 \ldots \forall x_n G$ formulėje *G* nėra kvantorių. Bekvantorė formulė *G* dar vadinama **matrica**, o $\forall x_1 \forall x_2 \ldots \forall x_n$ vadinamas formulės **prefiksu.**

Transformuojame matricas į NKF. Gauname formules pavidalo

$$\forall x_1 \forall x_2 \ldots \forall x_n (D_1 \wedge D_2 \wedge \ldots \wedge D_m).$$

Čia $D_i \ \ (i = 1, \ldots, m)$ yra disjunktai.

Ketvirtas žingsnis. Bendrumo kvantorių eliminavimas. Prisiminkime, kad

$$\forall x(A(x) \wedge B(x)) \equiv \forall x A(x) \wedge \forall x B(x).$$

Formulėse pavidalo $\forall x_1 \forall x_2 \ldots \forall x_n (D_1 \wedge D_2 \wedge \ldots \wedge D_m)$ įkeliame prefiksą į skliaustus

$$\forall x_1 \forall x_2 \ldots \forall x_n D_1 \wedge \forall x_1 \forall x_2 \ldots \forall x_n D_2 \wedge \ldots \wedge \forall x_1 \forall x_2 \ldots \forall x_n D_m$$

ir pervardijame suvaržytuosius kintamuosius taip, kad skirtinguose disjunktuose būtų skirtingi suvaržytieji kintamieji. Po to, išbraukiame prefiksus. Jie nereikalingi, nes visi suvaržytieji kintamieji yra suvaržyti tik bendrumo kvatoriais ir todėl *kvantorių rašymas yra perteklinis*. Po ketvirto žingsnio gauname aibę formulių pavidalo $D'_1 \wedge D'_2 \wedge \ldots \wedge D'_m$.

Penktas žingsnis. Disjunktų aibės sudarymas.

Disjunktų aibės *S* elementais yra visų gautų formulių pavidalo $D'_1 \wedge D'_2 \wedge \ldots \wedge D'_m$ disjunktai $D'_i \ \ (i = 1, \ldots, m)$. disjunktų aibėje skirtinguose disjunktuose yra skirtingi suvaržytieji kintamieji.

Pavyzdžiai.

a) Transformuoti sekvenciją $\forall x \neg P(x) \vee \exists x Q(x) \vdash \exists x (P(x) \to Q(x))$ į disjunktų aibę.

Transformuosime $\{\forall x \neg P(x) \vee \exists x Q(x), \neg \exists x (P(x) \to Q(x))\}$
Atliekame transformavimo algoritmo žingsnius:

1. Transformuojame $\forall x \neg P(x) \vee \exists x Q(x), \neg \exists x (P(x) \to Q(x))$ į normaliąją priešdėlinę formą. $\forall x \neg P(x) \vee \exists x Q(x) \equiv \forall x \neg P(x) \vee \exists y Q(y) \equiv$ (jei yra pasirinkimas - iškelti egzistavimo ar bendrumo kvantorių, rekomenduojama iškelti egzistavimo kvantorių) $\equiv \exists y \forall x (\neg P(x) \vee Q(y))$.

$\neg \exists x (P(x) \to Q(x)) \equiv \forall x \neg (P(x) \to Q(x))$.

2. Skulemizavimas.

$\exists y \forall x (\neg P(x) \vee Q(y))$. Gauname formulę $\forall x (\neg P(x) \vee Q(a))$.

Formulė $\forall x \neg (P(x) \to Q(x))$ jau yra skulemizuota.

3. Transformuojame matricą į NKF.

Formulė $\forall x (\neg P(x) \vee Q(a))$ jau yra NKF formos.

$\forall x \neg (P(x) \to Q(x)) \equiv \forall x \neg (\neg P(x) \vee Q(x)) \equiv \forall x ((P(x) \wedge \neg Q(x))$.

4. Eliminuojame bendrumo kvantorius.

$\forall x (\neg P(x) \vee Q(a))$ keičiame į $\neg P(x) \vee Q(a)$.

$\forall x (P(x) \wedge \neg Q(x)) \equiv \forall x P(x) \wedge \forall x \neg Q(x) \equiv \forall y P(y) \wedge \forall z \neg Q(z) \equiv P(y) \wedge \neg Q(z)$.

Atsakymas: *Sekvencijos disjunktų aibė* $\{\neg P(x) \vee Q(a), P(y), \neg Q(z)\}$.

b) Ar tapačiai teisinga formulė $\forall x \exists y \forall z (P(y, z) \to P(x, y))$?

Formulė tapačiai teisinga, jei išvedama sekvencija $\vdash \forall x \exists y \forall z (P(y, z) \to P(x, y))$.

Transformuosime $\{\neg \forall x \exists y \forall z (P(y, z) \to P(x, y))\}$ į disjunktų aibę.

Atliekame transformavimo algoritmo žingsnius:

1. Transformuojame į normaliąją priešdėlinę formą:

$\neg \forall x \exists y \forall z (P(y, z) \to P(x, y)) \equiv \exists x \forall y \exists z \neg (P(y, z) \to P(x, y))$.

2. Skulemizavimas: $\exists x \forall y \exists z \neg (P(y, z) \to P(x, y))$ deduktyviai ekvivalenti $\forall y \neg (P(y, f(y)) \to P(a, y))$.

3. Transformuojame matricą į NKF:

$$\forall y \neg (P(y, f(y)) \to P(a, y)) \equiv \forall y \neg (\neg P(y, f(y)) \vee P(a, y)) \equiv \forall y (P(y, f(y)) \wedge \neg P(a, y)).$$

4. Eliminuojame bendrumo kvantorių:

$$\forall y (P(y, f(y)) \wedge \neg P(a, y)) \equiv \forall y P(y, f(y)) \wedge \forall y \neg P(a, y) \equiv P(y, f(y)) \wedge \neg P(a, x).$$

Atsakymas: $S = \{P(y, f(y)), \neg P(a, x)\}$.

Žemiau rezoliucijų metodu nustatysime ar aibė *S* prieštaringa, t.y. ar tapačiai teisinga formulė $\forall x \exists y \forall z (P(y, z) \to P(x, y))$.

Įrodėm kaip formulių aibės transformuojamos į tam tikrą formulių pavidalą - disjunktų aibę. J.A.Robinson aprašytas rezoliucijų metodas remiasi dviem fundamentaliais logikos rezultatais: Herbrand'o universumu ir kompaktiškumo teorema. Paaiškinsime juos.

Herbrand'o universumas.

Jau buvo aiškinta, kad jei nagrinėjame formulę $\forall x G(x)$ baigtinėje individinių konstantų aibėje $D = \{a_1, \ldots, a_m\}$, tai

$$\forall x G(x) \equiv G(a_1) \wedge \ldots \wedge G(a_m).$$

T.y. tokiu atveju galima apsieiti be bendrumo kvantoriaus. Jį mokame eliminuoti ir predikatų logikos formulę galime transformuoti į teiginių logikos formulę. O ar bet kuriai formulei pavidalo $\forall x_1 \forall x_2 \ldots \forall x_n G$ (formulėje G nėra kvantorių įeičių) egzistuoja tokia individinių konstantų aibė *D*?

Nagrinėjame normaliosios priešdėlinės formos su funkciniais simboliais formules. Prancūzų logikas J.Herbrand'as įrodė, kad bet kuriai skulemizuotai formulei $F = \forall x G(x)$ visada galima rasti tokią termų aibę *H*, kad formulė įvykdoma tada ir tiktai tada, kai ji įvykdoma aibėje *H*. Aibė *H* dar vadinama *Herbrand'o universumu*. Deja, aibė *H* ne visada baigtinė.

Herbrand'o universumas konstruojamas tokiu būdu:

1. Visi laisvieji kintamieji, priklausantys formulei *F*, priklauso ir universumui *H*. Jei formulėje *F* nėra laisvųjų kintamųjų, tai aibės *H* elementu yra *a* (vadinasi, aibė *H* negali būti tuščia).

2. Jei f^n yra *n*-vietis funkcinis simbolis, priklausantis *F*, ir $t_1, \ldots, t_n$ yra *H* elementai, tai $f(t_1, \ldots, t_n)$ taip pat priklauso universumui *H*.

Pavyzdžiai.

1. $\forall x \forall y (P(x, c) \vee \neg Q(b, y))$. $H = \{b, c\}$.

2. $\forall x \forall y \forall z(P(x,y) \wedge Q(y,z))$. $H = \{a\}$.

3. $\forall x(P(x, f(f(a))) \rightarrow Q(b,a))$. $H = \{a, b, f(a), f(b), f(f(a)), f(f(b)), f(f(f(a))),$ $f(f(f(b))), \ldots\}$.

4. $\forall x(P(x, f(g(a))) \vee \neg Q(b,a))$. $H = \{a, b, f(a), f(b), g(a), g(b), f(f(a)), f(f(b)), f(g(a)),$ $f(g(b)), g(f(a)), g(f(b)), \ldots\}$.

Pirmame ir antrame pavyzdžiuose aibė *H* baigtinė. O tokiu atveju jau mokame eliminuoti kvantorius:

1. $\forall x \forall y(P(x,a) \vee \neg Q(b,y)) \equiv \forall y(P(a,a) \vee \neg Q(b,y)) \wedge \forall y(P(b,a) \vee \neg Q(b,y)) \equiv$ $\equiv (P(a,a) \vee \neg Q(b,a)) \wedge (P(a,a) \vee \neg Q(b,b)) \wedge (P(b,a) \vee \neg Q(b,a)) \wedge (P(b,a) \vee \neg Q(b,b))$.

2. $\forall x \forall y \forall z(P(x,y) \wedge Q(y,z)) \equiv \forall y \forall z(P(a,y) \wedge Q(y,z)) \equiv \forall z(P(a,a) \wedge Q(a,z)) \equiv$ $P(a,a) \wedge Q(a,a))$.

Begalinės *H* atveju (3 pavyzdys) formulei $\forall x(P(x, f(f(a))) \rightarrow Q(b,a))$ priskiriamas begalinis reiškinys (tai nėra formulė, nes formulės yra baigtiniai reiškiniai).

Formulę $\forall x(P(x, f(f(a))) \rightarrow Q(b,a))$ pažymėkime $G(x)$. Tuomet formulei $\forall x G(x)$ priskiriame begalinį reiškinį

$$G(a) \wedge G(b) \wedge G(f(a)) \wedge G(f(b)) \wedge G(f(f(a))) \wedge G(f(f(b))) \wedge \ldots$$

Užrašysime jį begalinės aibės pavidalu (konjukciją pakeisime kableliu)

$$\{G(a), G(b), G(f(a)), G(f(b)), G(f(f(a))), G(f(f(b))), \ldots\}.$$

Jei *G* yra NKF, tai turėsime **teiginių logikos disjunktų aibę**. Aibė įvykdoma tada ir tiktai tada, kai formulė $\forall x(P(x, f(f(a))) \rightarrow Q(b,a))$ įvykdoma. Vadinasi, aibė prieštaringa tada ir tiktai tada, kai formulė $\forall x(P(x, f(f(a))) \rightarrow Q(b,a))$ tapačiai klaidinga (neįvykdoma). Taigi, Herbrand'ui pavyko įrodyti, kad pagal bet kurią predikatų logikos formulę *F*, galima rasti tokią teiginių logikos disjunktų aibę, kad formulė *F* tapačiai klaidinga tada ir tiktai tada, kai *teiginių logikos formulių aibė prieštaringa*.

Kompaktiškumo teorema. Jei teiginių logikos formulių aibė prieštaringa, tai egzistuoja jos baigtinis prieštaringas poaibis.

Tarkime kuri nors begalinė teiginių logikos formulių aibė $\{G_1, \mathbf{G_2}, G_3, \mathbf{G_4}, \mathbf{G_5}, G_6, \ldots\}$ prieštaringa. Pagal kompaktiškumo teoremą, egzistuoja baigtinis prieštaringas poaibis. Tarkime tik $\{G_2, G_4, G_5\}$ yra prieštaringas tarp pirmų penkių jos elementų. Mes to nežinome. Kaip jį rasti? Galima naudoti tokį algoritmą:

1. Tikriname ar $\{G_1\}$ prieštaringa. Kadangi tai teiginių logikos formulė, tai patikrinti galima ir teisingumo lentele. Nėra prieštaringa. Aibę papildome antruoju elementu.

2. Tikriname ar $\{G_1, \mathbf{G_2}\}$ prieštaringa. Nėra. Aibę papildome trečiuoju elementu.

3. Tikriname ar $\{G_1, \mathbf{G_2}, G_3\}$ prieštaringa. Nėra. Aibę papildome ketvirtuoju elementu.

4. Tikriname ar $\{G_1, \mathbf{G_2}, G_3, \mathbf{G_4}\}$ prieštaringa. Nėra. Aibę papildome penktuoju elementu.

5. Tikriname ar $\{G_1, \mathbf{G_2}, G_3, \mathbf{G_4}, \mathbf{G_5}\}$ prieštaringa. Taip. Aibei atitinkančios sekvencijos išvada teisinga. Baigiame paiešką.

Turbūt prisimenate, kad, jei *F* tapačiai klaidinga, tai ir $F \wedge G$ tapačiai klaidinga su bet kuria formule *G*.

Aprašytuoju algoritmu visad rasite baigtinį prieštaringą poaibį, jeigu toks nagrinėjamoje aibėje egzistuoja. Jei baigtinio prieštaringo poaibio nėra, tai paieška bus begalinė.

Be to, toks algoritmas nėra efektyvus (reikalauja daug kompiuterio atminties ir atlieka daug žingsnių, be dalies kurių galėtumėm ir apsieiti). J.Robinson aprašius rezoliucijų metodą, jis greitai paplito po pasaulį ir laikomas vienu efektyviausiu prieštaringumo nustatymo algoritmu.

Rezolucijų metodas taikomas predikatų logikos formulių aibei, kurios visi elementai skulemizuoti disjunktai. O taisyklė yra tik viena. Ji vadinama **rezoliucijos taisykle**. Ji taikoma dviem disjunktams, rezultatas - vienas disjunktas. Priminsime atkirtos taisyklę teiginių logikos formulėms

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

Panaši taisyklė yra ir predikatų logikos formulėms, bet vadinama *rezoliucijos taisykle*. Tik vietoje loginių kintamųjų yra *atominės formulės.* Priminsime apibrėžimą.

Jei P yra n-vietis predikatas (*predikatinis kintamasis*) *ir* $t_1, \ldots, t_n$ *- termai, tai* $P(t_1, \ldots, t_n)$ *yra formulė* (*ji vadinama* ***atomine formule***).

Norint taikyti rezoliucijos taisyklę vienoje prielaidoje (disjunkte) turime turėti atominę formulę, o kitoje tokios pačios (jai lygios) neigimą. Jei atominės formulės nėra lygios, tai, kai kuriais atvejais, rezoliucijų metode galima jas suvienodinti.

Pavyzdžiui, $P(x)$ ir $P(y)$ yra skirtingos atominės formulės. Bet, jei laikysime $x = y$ (pervardinsime), tai gausime jas lygias. Tiksliau, jei *pakeisime y* į *x* (arba *x* į *y*). Apibrėšime *keitinio* sąvoką.

Keitiniu vadiname reiškinį pavidalo $(t_1/x_1, t_2/x_2, \ldots, t_n/x_n)$; čia t_i *- termai*. Keitinio prasmė tokia - formulėje visų kintamųjų $x_1, x_2, \ldots, x_n$ įeitys keičiamos atitinkamais termais $t_1, t_2, \ldots, t_n$. Keitinius žymime raidėmis α, σ. Formulę *F*, kurioje visos kintamojo x_i $(i = 1, \ldots, n)$ įeitys pakeistos termu t_i. T.y. $\alpha = (t_1/x_1, t_2/x_2, \ldots, t_n/x_n)$). Žymima $F\alpha$.

Atkreipiame dėmesį, kad rezoliucijų metode naudojamuose termuose gali būti ir kintamųjų *x, y, z* ir t.t įeitys, t.y. kintamųjų, kuriais įprasta žymėti suvaržytuosius. Tai yra todėl, kad rezoliucijų metode naudojamuose disjunktuose kvantoriai nerašomi.

Keitinys α *vadinamas formulių F, G* ***unifikatoriumi****, jei* $F\alpha = G\alpha$.

Pavyzdžiui, $P(x)$ ir $P(y)$ yra skirtingos atominės formulės, bet keitiniu $\alpha = (x/y)$ mes jas suvienodiname (unifikavome) $P(x)\alpha = P(y)\alpha$. Šiuo keitiniu tik antroje formulėje *y* pakeitėme į *x*. $P(x)\alpha = P(x), P(y)\alpha = P(x)$.

Tarkime *A, B* - atominės formulės, $A \vee C_1, \neg B \vee C_2$ - disjunktai ir α toks keitinys, kad $A\alpha = B\alpha$. Tuomet **rezoliucijos taisyklė** yra tokia

$$\frac{(A \vee C_1)\alpha \quad (\neg B \vee C_2)\alpha}{(C_1 \vee C_2)\alpha}$$

Bet kurios literos (atominės formulės arba jos neigimo) įeitis bet kuriame disjunkte yra *tik viena.* Pavyzdžiui, $A \vee A \vee \neg B \vee \neg B$ laikysime lygiu $A \vee \neg B$ ir nagrinėsime pastarąjį.

Sakoma, kad iš disjunktų aibės S išvedamas tuščias disjunktas, jei egzistuoja baigtinė disjunktų seka $E_1, \ldots, E_r$, tenkinanti sąlygas:

1) $E_r = \square$,

2) kiekviena E_i priklauso aibei S arba gauta iš kairėje jos esančių disjunktų pagal rezoliucijos taisyklę.

Teorema. *Disjunktų aibė S prieštaringa tada ir tiktai tada, kai iš jos išvedamas tuščias disjunktas*.

Turėjome dvi sekvencijas, kurias transformavome į disjunktų aibes. Naudodamiesi jomis baigsime spręsti užduotis.

a) Rezoliucijų metodu nustatysime ar išvedama sekvencija $\forall x \neg P(x) \vee \exists x Q(x) \vdash \exists x(P(x) \to Q(x))$.

Ją transformavome į disjunktų aibę $\{\neg P(x) \vee Q(a), P(y), \neg Q(z)\}$. Ieškome tuščio disjunkto išvedimo iš aibės *S*.

1. $\neg P(x) \vee Q(a)$ prielaida,

2. $P(y)$ prielaida,

3. $Q(a)$ pagal rezoliucijos taisyklę iš 1, 2 su keitiniu $\sigma = (x/y)$,

4. $\neg Q(z)$ prielaida,

5. $\square$ pagal rezoliucijos taisyklę iš 3, 4 su keitinius $\sigma = (a/z)$.

Atsakymas: *sekvencija išvedama.*

Sprendimą galima rašyti ir taisyklės taikymų seka:

$$\frac{\neg P(x) \vee Q(a) \quad P(y)}{Q(a)}\ \sigma = (x/y) \qquad\qquad \frac{Q(a) \quad \neg Q(z)}{\square}\ \sigma = (a/z)$$

b) Ar tapačiai teisinga formulė $\forall x \exists y \forall z (P(y,z) \to P(x,y))$?

Užduotį transformavome į aibės $S = \{P(y, f(y)), \neg P(a,x)\}$ prieštaringumo nustatymą. Naudosime rezoliucijų metodą.

$$\frac{P(y,f(y)) \quad \neg P(a,x)}{\square}\ \sigma(a/y, f(a)/x)$$

Atsakymas: *formulė tapačiai teisinga*.

4.3 Natūralioji dedukcija

Sakoma, kad logikos dėsnių aibė aprašyta *natūraliu* loginiu skaičiavimu, jei jis tenkina dvi sąlygas. Visų pirma, skaičiavimas yra aksiominės matematinės teorijos pavidalo, t.y. jis aprašomas kaip aksiomų ir taisyklių rinkinys, kuriame kaip aksiomos, taip ir taisyklės yra akivaizdžios, savaime suprantomos ir naudojamos matematiniuose įrodymuose. B.Russellas ir A.N.Whiteheadas buvo pirmieji aprašę savo tritomyje *Principia Mathematika* logikos dėsnių aibę kaip aksiominę teoriją. Be to, ta dirbtinė, formali sistema turi būti artima šnekamosios kalbos struktūrai. Tokias dedukcijos sistemas nepriklausomai vienas nuo kito 1934 m. aprašė lenkas S.Jaskowskis ir vokietis G.Gentzenas. Pavadino jas **natūraliomis dedukcijomis** (pirmieji panaudojo šį terminą), nes perėjimai nuo prielaidų prie išvadų geriausiai modeliuoja šnekamosios bei mokslininkų vartojamus išvedimuose samprotavimus.

Sekvencinis skaičiavimas irgi modeliuoja matematiko mąstyseną. Kuo jis neįtiko Gentzenui?
Pagrindinės priežąstys yra dvi:

1. Matematiniuose įrodymuose naudojama ir tokia taisyklė (o sekvenciniame nėra):

$$\frac{\Gamma, \neg F \vdash}{\Gamma \vdash F}$$

Išvedimo paieška dažniausiai vykdoma iš apačios į viršų. *Mąstymo schema tokia*: norime įrodyti *F*. Tariame, kad *F* klaidinga (t.y. perkeliam į prielaidų sąrašą *F* su neigimu) ir *įrodome prieštaravimą* (prieštaravimo atitikmuo - sukcedente nėra formulių, t.y. jis tuščias, kai kuriuose skaičiavimuose tai žymima $\perp$).

2. Matematikai nenaudoja tokios taisyklės

$$\frac{\Gamma \vdash \Delta, F, G}{\Gamma \vdash \Delta, F \vee G}$$

Matematiniuose įrodymuose visada yra *ne daugiau kaip viena formulė* (viena arba sukcedentas tuščias).

Šiuo skaičiavimu dar labiau priartėta prie matematiko mąstymo modeliavimo. Įrodomų sekvencijų aibė natūraliąja dedukcija sutampa au aibe įrodomų sekvenciniame skaičiavime.

Paminėsiu porą atveju, kai natūraliosios dedukcijos skaičiavimas pranašesnis už sekvencinį.

1. Norima įrodyti teoremą (dažniausiai naudojant kompiuterį) ir pateikti matematikams suprantama logine kalba (natūraliosios dedukcijos skaičiavimas modeliuoja matematiko mąstyseną, o sekvencinis ne),

2. Pavyko rasti ryšį tarp mąstymo (natūralioji dedukcija) ir algoritmiškai apskaičiuojamų funkcijų (λ-skaičiavimas). Juo naudojantis sukurtos programavimo kalbos, kuriomis parašytų programų nebereikia verifikuoti. Tai padaro pati programavimo kalbos aplinka.

Natūraliosios dedukcijos skaičiavimas.

Aksioma: $\Gamma, F \vdash F$

Loginių operacijų taisyklės:

$\to$ įvedimas $\to$ eliminavimas

$$\frac{\Gamma, F \vdash G}{\Gamma \vdash F \to G} \qquad \frac{\Gamma \vdash F \quad \Delta \vdash F \to G}{\Gamma, \Delta \vdash G}$$

Pastaba. Jei Γ ir Δ sutampa, tai taisyklė $\to$ *eliminavimas* atrodo šitaip:

$$\frac{\Gamma \vdash F \quad \Gamma \vdash F \to G}{\Gamma \vdash G}$$

$\wedge$ įvedimas $\wedge_1$ eliminavimas $\wedge_2$ eliminavimas

$$\frac{\Gamma \vdash F \quad \Delta \vdash G}{\Gamma, \Delta \vdash F \wedge G} \qquad \frac{\Gamma \vdash F \wedge G}{\Gamma \vdash F} \qquad \frac{\Gamma \vdash F \wedge G}{\Gamma \vdash G}$$

Pastaba. Jei Γ ir Δ sutampa, tai taisyklė $\wedge$ *įvedimas* atrodo šitaip:

$$\frac{\Gamma \vdash F \quad \Gamma \vdash G}{\Gamma \vdash F \wedge G}$$

$\vee_1$ įvedimas $\vee_2$ įvedimas $\vee$ eliminavimas

$$\frac{\Gamma \vdash F}{\Gamma \vdash F \vee G} \qquad \frac{\Gamma \vdash G}{\Gamma \vdash F \vee G} \qquad \frac{\Gamma \vdash F \vee G \quad \Delta, F \vdash H \quad \Delta_1, G \vdash H}{\Gamma, \Delta, \Delta_1 \vdash H}$$

Pastaba. Jei Γ, Δ ir Δ_1 sutampa, tai taisyklė $\vee$ *eliminavimas* atrodo šitaip:

$$\frac{\Gamma \vdash F \vee G \quad \Gamma, F \vdash H \quad \Gamma, G \vdash H}{\Gamma \vdash H}$$

$\neg$ įvedimas $\neg_1$ eliminavimas $\neg_2$ eliminavimas

$$\frac{\Gamma, F \vdash}{\Gamma \vdash \neg F} \qquad \frac{\Gamma \vdash F \quad \Delta \vdash \neg F}{\Gamma, \Delta \vdash} \qquad \frac{\Gamma, \neg F \vdash}{\Gamma \vdash F}$$

Pastaba. Jei Γ ir Δ sutampa, tai taisyklė $\neg_1$ *eliminavimas* atrodo šitaip:

$$\frac{\Gamma \vdash F \quad \Gamma \vdash \neg F}{\Gamma \vdash}$$

silpninimas

$$\frac{\Gamma \vdash G}{\Gamma, F \vdash G} \qquad \frac{\Gamma \vdash}{\Gamma \vdash F}$$

Čia Γ, Δ, Δ_1 yra baigtinės formulių sekoa (gali būti ir tuščios). *F, G, H* - formulės. Priminsime, kad sekvencijų antecedente esančių formulių tvarka nesvarbi. Taisyklėse viršuje brūkšnio yra *sekvencija prielaida*, kuriai taikoma taisyklė. Apačioje brūkšnio, taisyklės taikymo rezultatas - *išvada*. Kai kuriuose vadovėliuose tuščiame sukcedente rašomas simbolis $\bot$ (klaidinga). Pavyzdžiui, vietoje $\Gamma \vdash$ rašoma $\Gamma \vdash \bot$.

Formulių įrodymų pavyzdžiai.

1. $\vdash \neg(F \vee G) \to (\neg F \wedge \neg G)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg(F \vee G), F \vdash F}\oplus}{H, F \vdash \mathbf{F} \vee \mathbf{G}} \quad \dfrac{}{H, F \vdash \neg(F \vee G)}\oplus}{\neg(F \vee G), F \vdash}}{\neg(F \vee G) \vdash \neg\mathbf{F}} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\neg(F \vee G), G \vdash G}\oplus}{H, G \vdash \mathbf{F} \vee \mathbf{G}} \quad \dfrac{}{H, G \vdash \neg(F \vee G)}\oplus}{\neg(F \vee G), G \vdash}}{\neg(F \vee G) \vdash \neg\mathbf{G}}}{\neg(F \vee G) \vdash \neg\mathbf{F} \wedge \neg\mathbf{G}}}{\vdash \neg(F \vee G) \to (\neg F \wedge \neg G)}$$

Čia, kai kuriose sekvencijose, H žymi formulę $\neg(F \vee G)$.

2. $\vdash (F \wedge G) \to (G \wedge F)$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{F \wedge G \vdash F \wedge G}\oplus}{F \wedge G \vdash \mathbf{G}} \quad \dfrac{\dfrac{}{F \wedge G \vdash F \wedge G}\oplus}{F \wedge G \vdash \mathbf{F}}}{F \wedge G \vdash \mathbf{G} \wedge \mathbf{F}}}{\vdash (F \wedge G) \to (G \wedge F)}$$

3. $\vdash \neg\neg F \to F$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\neg F, \neg F \vdash \neg F}\oplus \quad \dfrac{}{\neg\neg F, \neg F \vdash \neg\neg F}\oplus}{\neg\neg F, \neg F \vdash}}{\neg\neg F \vdash \mathbf{F}}}{\vdash \neg\neg F \to F}$$

Priminsime, kad *laisvuosius kintamuosius* žymime $a, b, c, d, e, a_1, a_2, a_3, \ldots$

Suvaržytuosius kintamuosius žymime $x, y, z, u, v, w, x_1, x_2, x_3, \ldots$

Vadovėliuose natūraliosios dedukcijos kvantorinės taisyklės dažniausiai nusakomos tokiu būdu:

$\exists$ įvedimas $\qquad\qquad$ $\exists$ eliminavimas

$$\frac{\Gamma \vdash F(\gamma)}{\Gamma \vdash \exists x F(x)} \qquad\qquad \frac{\Gamma \vdash \exists x F(x) \quad \Delta, F(\beta) \vdash G}{\Gamma, \Delta \vdash G}$$

$\forall$ įvedimas $\qquad\qquad$ $\forall$ eliminavimas

$$\frac{\Gamma \vdash F(\beta)}{\Gamma \vdash \forall x F(x)} \qquad\qquad \frac{\Gamma \vdash \forall x F(x)}{\Gamma \vdash F(\gamma)}$$

Raide *C* žymime aibę laisvųjų kintamųjų taisyklėse:

1. $\exists$ *įvedimas* - visi laisvieji kintamieji, esantys apatinėje sekvencijoje,
2. $\forall$ *įvedimas* - visi laisvieji kintamieji, esantys apatinėje sekvencijoje,
3. $\exists$ *eliminavimas* - visi laisvieji kintamieji, esantys formulėse $\Gamma, \exists x F(x), \Delta, G$,
4. $\forall$ *eliminavimas* - visi laisvieji kintamieji, esantys viršutinėje sekvencijoje.

Taisyklėse $\beta \notin C$, β yra *naujas* laisvasis kintamasis, o γ taisyklėse žymi kintamąjį priklausantį *C*. $(\gamma \in C)$. Jei *C* tuščia aibė, tai γ naujas laisvasis kintamasis.

Pastaba dėl taisyklių pavadinimų. Taisyklės skaitomos *iš viršaus į apačią* (virš brūkšnio prielaida, apačioje išvada). Taisyklėje $\rightarrow$ *įvedimas* virš brūkšnio nėra loginės operacijos $\rightarrow$, o apačioje ji atsiranda (įvedama $\rightarrow$). Taisyklėje $\rightarrow$ *eliminavimas* virš brūkšnio vienoje sekvencijoje yra loginė operacija $\rightarrow$, o apatinėje jos nebėra (vyksta loginės operacijos *šalinimas, eliminavimas*).

Kai kada formulių išvedimai natūralioje dedukcijoje vykdomi *iš apačios į viršų*, kaip ir sekvenciniame skaičiavime. Tuomet patogu naudoti sekvencinio skaičiavimo kvantorines taisykles $(\exists \vdash), (\forall \vdash)$. Parodysime, kad šios sekvencinio skaičiavimo taisyklės yra leistinos natūraliosios dedukcijos skaičiavime:

$$(\exists \vdash) \ \frac{\Gamma, F(\beta) \vdash G}{\Gamma, \exists x F(x) \vdash G} \qquad\qquad (\forall \vdash) \ \frac{\Gamma, \forall x F(x), F(\gamma) \vdash G}{\Gamma, \forall x F(x) \vdash G}$$

Taisyklę $(\exists \vdash)$ modeliuojame tokius išvedimu:

$$\dfrac{\dfrac{\dfrac{\oplus}{\Gamma, \exists x F(x) \vdash \exists x F(x)} \quad \dfrac{?}{\Gamma, \exists x F(x), F(\beta) \vdash G}}{\text{taikome } \exists \text{ eliminavimas taisyklę}}}{\Gamma, \exists x F(x) \vdash G}$$

Gavome, kad $\Gamma, \exists x F(x) \vdash G$ išvedama, jei išvedama $\Gamma, F(\beta) \vdash G$, t.y. gavome tą patį, jei taikytumėm sekvencinio skaičiavimo taisyklę $(\exists \vdash)$.

Taisyklę $(\forall \vdash)$ modeliuojame tokiu būdu:

$$\dfrac{\dfrac{\dfrac{}{\Gamma, \forall x F(x) \vdash \forall x F(x)}\,\oplus}{\Gamma, \forall x F(x), \neg G \vdash \forall x F(x)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{?}{\Gamma, \forall x F(x), F(\gamma) \vdash G}}{\Gamma, \forall x F(x), \neg G, F(\gamma) \vdash G}\,\text{silpninimas} \qquad \dfrac{}{\Gamma, \forall x F(x), \neg G, F(\gamma) \vdash \neg G}\,\oplus}{\Gamma, \forall x F(x), \neg G, F(\gamma) \vdash}\,\text{taikome } \neg_1}{\Gamma, \forall x F(x), \neg G \vdash \neg \mathbf{F}(\gamma)}}{\Gamma, \forall x F(x), \neg G \vdash}$$

$$\Gamma, \forall x F(x) \vdash \mathbf{G}$$

Gavome, kad $\Gamma, \forall x F(x) \vdash G$ išvedama, jei išvedama $\Gamma, \forall x F(x), F(\gamma) \vdash G$, t.y. gavome tą patį, jei taikytumėm sekvencinio skaičiavimo taisyklę $(\forall \vdash)$.

Pavyzdžiai.

1. $\exists x(A(x) \to B(x)) \vdash \forall x A(x) \to \exists x B(x)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A(a) \to B(a), \forall x A(x), A(a) \vdash A(a)}\,\oplus \qquad \dfrac{}{A(a) \to B(a), \forall x A(x), A(a) \vdash A(a) \to B(a)}\,\oplus}{A(a) \to B(a), \forall x A(x), A(a) \vdash \mathbf{B(a)}}}{A(a) \to B(a), \forall x A(x), A(a) \vdash \exists \mathbf{x} \mathbf{B(x)}}}{A(a) \to B(a), \forall \mathbf{x} \mathbf{A(x)} \vdash \exists x B(x)}}{A(a) \to B(a) \vdash \forall \mathbf{x} \mathbf{A(x)} \to \exists \mathbf{x} \mathbf{B(x)}}}{\exists \mathbf{x}(\mathbf{A(x)} \to \mathbf{B(x)}) \vdash \forall x A(x) \to \exists x B(x)}$$

2. $A(a) \to \exists x B(x) \vdash \exists x(A(a) \to B(x))$. Atkreipiu dėmesį. Išvedimo pradžia - trečiąsis išvedimo paieškos medis.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, B(b), A(a) \vdash B(b)}\,\oplus}{\Gamma, B(b) \vdash \mathbf{A(a)} \to \mathbf{B(b)}}}{\Gamma, B(b) \vdash \exists \mathbf{x}(\mathbf{A(a)} \to \mathbf{B(x)})} \qquad \dfrac{}{\Gamma, B(b) \vdash \neg \exists x(A(a) \to B(x))}\,\oplus}{\Gamma, B(b) \vdash}$$

Antrosios sekvencijos išvedimas.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, \neg A(a), A(a), \neg B(a) \vdash A(a)}\,\oplus \qquad \dfrac{}{\Gamma, \neg A(a), A(a), \neg B(a) \vdash \neg A(a)}\,\oplus}{\Gamma, \neg A(a), A(a), \neg B(a) \vdash}}{\Gamma, \neg A(a), A(a) \vdash \mathbf{B(a)}}}{\Gamma, \neg A(a) \vdash \mathbf{A(a)} \to \mathbf{B(a)}}}{\Gamma, \neg A(a) \vdash \exists \mathbf{x}(\mathbf{A(a)} \to \mathbf{B(x)})} \qquad \dfrac{}{\Gamma, \neg A(a) \vdash \neg \exists x(A(a) \to B(x))}\,\oplus}{\Gamma, \neg A(a) \vdash}$$

Pirmosios sekvencijos išvedimas.

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{pirma sekven.}}{\Gamma, \neg A(a) \vdash}}{\Gamma \vdash \mathbf{A(a)}} \quad \dfrac{\oplus}{\Gamma \vdash A(a) \to \exists x B(x)}}{\Gamma \vdash \exists \mathbf{x}\mathbf{B}(\mathbf{x})} \quad \dfrac{\dfrac{\text{antra sekven.}}{\Gamma, B(b) \vdash}}{\Gamma, \exists \mathbf{x}\mathbf{B}(\mathbf{x}) \vdash} \Big/ \Gamma \vdash \neg\exists \mathbf{x}\mathbf{B}(\mathbf{x})}{\text{Pažymėkime raide } \Gamma \text{ sąrašą formulių } A(a) \to \exists x B(x), \neg\exists x(A(a) \to B(x))}$$
$$\dfrac{A(a) \to \exists x B(x), \neg\exists x(A(a) \to B(x)) \vdash}{A(a) \to \exists x B(x) \vdash \exists \mathbf{x}(\mathbf{A}(\mathbf{a}) \to \mathbf{B}(\mathbf{x}))}$$

Natūraliosios dedukcijos predikatų logikos skaičiavimas su funkciniais simboliais. Taisyklės tokios pačios, tik vietoje laisvųjų kintamųjų naudojame termus ir kitokie reikalavimai individiniams kintamiesiems β, γ kvantorinėse taisyklėse (žr. skyrelį *termai*).

4.4 Intuicionistinė natūralioji dedukcija

Tarkime *F* kuri nors formulė. Galimi keturi atvejai:

1. $\vdash F$ ir $\nvdash \neg F$. Įrodoma *F* ir neįrodoma $\neg F$; įrodoma, kad formulė teisinga, ir neįrodoma, kad ji klaidinga.

2. $\nvdash F$ ir $\vdash \neg F$. Neįrodoma *F* ir įrodoma $\neg F$; neįrodoma, kad formulė teisinga, o įrodoma, kad ji klaidinga.

3. $\nvdash F$ ir $\nvdash \neg F$. Neįrodomos nei *F*, nei $\neg F$; negalime įrodyti, kad formulė teisinga, negalime įrodyti ir kad ji klaidinga.

4. $\vdash F$ ir $\vdash \neg F$. Įrodoma *F* ir įrodoma $\neg F$; galime įrodyti, kad formulė teisinga ir kad ji klaidinga.

Ketvirtas atvejis bet kurioje mokslo teorijoje nepriimtinas. *Nes iš prieštaravimo išplaukia bet kuris teiginys.* Todėl būtinas reikalavimas bet kuriai mokslo teorijai yra *neprieštaringumo įrodymas.*

Trečią atvejį yra prasmė nagrinėti tik teorijose, kuriose yra tiksliai apibrėžta *kas yra įrodymas*. Tiksliau, teorijose, kuriose įrodymai gaunami naudojant loginius skaičiavimus. Kadangi *pirmas pilnas ir neprietaringas* loginis skaičiavimas aprašytas tik XX amžiaus pirmoje pusėje, tai iki to laiko visuose matematiniuose įrodymuose buvo laikomasi, kad trečias atvejis negalimas. T.y., koks bebūtų tvirtinimas *F*, teisinga formulė $F \vee \neg F$ (trečiojo negalimo dėsnis). Loginiai skaičiavimai, kuriuose yra aksioma $F \vee \neg F$, arba tokia formulė įrodoma, vadinami *klasikiniais loginiais skaičiavimais* (klasikinė logika).

Šiame skyrelyje nagrinėsime loginį skaičiavimą, kuriame formulė $F \vee \neg F$ nėra įrodoma. Tokie skaičiavimai vadinami *intuicionistiniais* (*konstruktyviais*). Tokius skaičiavimus dažniausiai

naudoja informatikai. Konstruktyvioje matematikoje samprotaujama naudojant intuicionistinės logikos taisykles. Nes tvirtinimai pavidalo $F \vee \neg F$ laikomi įrodomais tik tuo atveju, jei įrodoma, kuris iš $F, \neg F$ yra teisingas. Aprašysime vieną intuicionistinės logikos skaičiavimų variantą - *intuicionistinę natūraliąją dedukciją*.

Intuicionistinėje logikoje daug kas keičiasi. Tiesa, sintaksė ta pati. Formulių apibrėžimai tie patys (pagal pavidalą klasikinės ir intuicionistinės logikos formulės nesiskiria). Semantika kitokia. Kalbant apie teiginius dažniausiai naudojamos ne sąvokos *teisingas, klaidingas*, o *įrodomas, neįrodomas* (*prieštaringas*). Nes kitokia teisingumo samprata:

1. $F \wedge G$ įrodomas, jei įrodomi kaip *F*, taip ir *G*,
2. $\neg F$ įrodoma, jei įrodoma $F \to \bot$, *F* prieštaringa,
3. $F \to G$ įrodoma, jei, turint *F* įrodymą, mokame rasti *G* įrodymą,
4. $F \vee G$ įrodoma, jei *F* arba *G* įrodomi,
5. $\exists x F(x)$ įrodoma, jei nurodome algoritmą, kuriuo galima rasti tokį *a*, su kuriuo $F(a)$ įrodomas,
6. $\forall x F(x)$ įrodoma, jei, kokį bepaimtumėm elementą *a* iš apibrėžimo aibės, įrodoma $F(a)$.

Intuicionistinėje logikoje loginės operacijos neaprašomos teisingumo lentelėmis. Formulių įvykdomumui, tapačiai teisingumui ar tapačiai klaidingumui naudojamos Kripke struktūros arba intuicionistiniai loginiai skaičiavimai. Visų pirma susipažinsim su Kripke struktūromis.

Apibrėžimas. *Intuicionistinės teiginių logikos formulės $F(p_1, \ldots, p_n)$ Kripke's struktūra vadiname trejetą $\Phi = (D, R, V)$; **D** - kuri nors netuščia aibė, vadinama **galimų pasaulių aibe**, **R** - apibrėžtas aibėje **D** binarinis refleksyvus ir tranzityvus predikatas, vadinamas **pasaulių sąryšiu**, **V** - **aibė interpretacijų** pasauliuose (kiekvienam pasauliui prikiriamas kuris nors loginių kintamųjų aibės $\{p_1, \ldots, p_n\}$ poaibis tenkinantis sąlygą: jei, kuriam nors pasauliui i priskiriamas poaibis **A**, pasauliui j - **B** ir $R(i, j) = t$, tai $A \subseteq B$).*

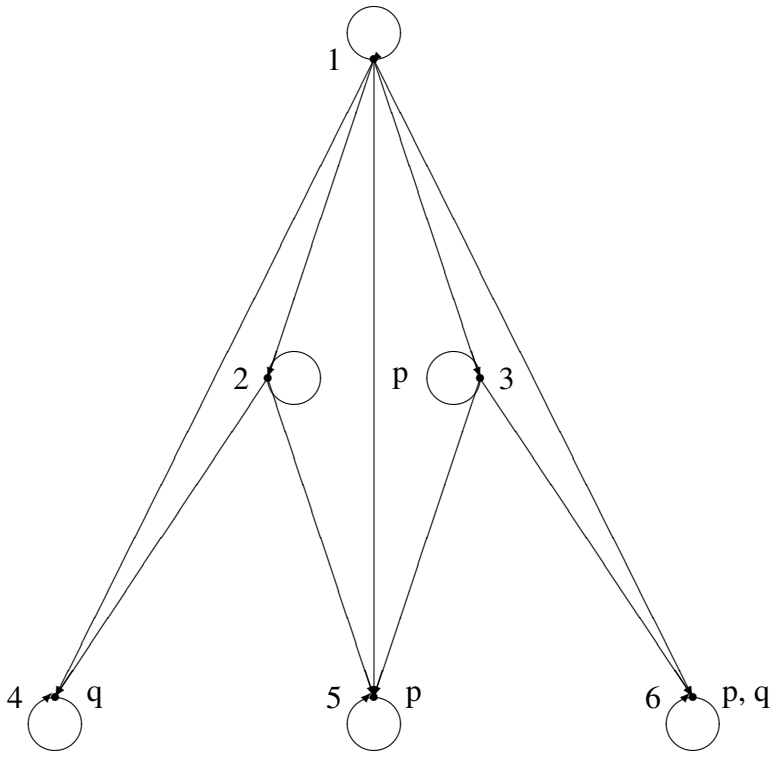
Pateiksime *Kripke's struktūros pavyzdį* formulei, kurioje yra loginiai kintamieji *p, q*. Struktūrą nusakysime orientuotu grafu. *Galimi pasauliai* - grafo viršūnės. *Pasaulių sąryšiai* žymimi lankais. Greta viršūnių rašome jai priskirto poaibio elementus - loginius kintamuosius. Jei poaibis tuščias, tai greta viršūnės nėra loginių kintamųjų.

Pavyzdys.

Galimų pasaulių aibė $D = \{1, 2, 3, 4, 5, 6\}$.

Pasaulių sąryšių aibė $R = \{(1,1), (2,2), (3,3), (4,4), (5,5), (6,6), (1,2), (1,3), (1,4), (1,5), (1,6), (2,4), (2,5), (3,5), (3,6)\}$.

Interpretacijų aibė *V*:
viršūnei (pasauliui) 3 priskirta aibė $\{p\}$, viršūnei 4 - $\{q\}$, 5 - $\{p\}$, 6 - $\{p, q\}$, likusioms viršūnėms priskirtos tuščios aibės.



Tarkime Φ yra Kripke's struktūra formulei $F(p_1, \ldots, p_n)$. Formulės *F* struktūros Φ pasaulyje *s įvykdomumo sąryšis* $\Phi, s \Vdash F$ apibrėžiamas tokiu būdu:

1. jei *F* yra loginis kintamasis $p_i \in \{p_1, \ldots, p_n\}$, tai $\Phi, s \Vdash F$ (t.y. *F* įvykdoma struktūros Φ pasaulyje *s*), jei p_i yra aibės, priskirtos viršūnei *s*, elementas;
2. jei $F = G \wedge H$, tai $\Phi, s \Vdash F$ tada ir tiktai tada, kai $\Phi, s \Vdash G$ ir $\Phi, s \Vdash H$;
3. jei $F = G \vee H$, tai $\Phi, s \Vdash F$ tada ir tiktai tada, kai $\Phi, s \Vdash G$ arba $\Phi, s \Vdash H$;
4. jei $F = G \to H$, tai $\Phi, s \Vdash F$ tada ir tiktai tada, jei kiekviename pasaulyje *m*, į kurį galima patekti (yra kelias) iš *s*, tenkinama sąlyga: jei $\Phi, m \Vdash G$, tai ir $\Phi, m \Vdash H$;
5. jei $F = \neg G$, tai $\Phi, s \Vdash F$ tada ir tiktai tada, jei $\Phi, s \nVdash G$ kiekviename pasaulyje *m*, į kurį galima patekti iš *s*.

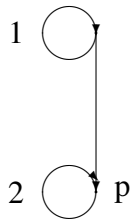
Formulė *F* vadinama *intuicionistinės logikos dėsniu*, jei kiekvienos Kripke's struktūros Φ kiekviename pasaulyje *s* teisinga $\Phi, s \Vdash F$.

Pavyzdys. Struktūra Φ:

Galimų pasaulių aibė $D = \{1, 2\}$.

Pasaulių sąryšių aibė $R = \{(1,1), (2,2), (1,2)\}$.

Interpretacijų aibė *V*:
viršūnei (pasauliui) 2 priskirta aibė $\{p\}$, o viršūnei 1 - tuščia aibė.



Nagrinėjame formulių $p, \neg p$ įvykdomumą viršūnėje (pasaulyje) 1. $\Phi, 1 \nVdash p$, nes priskirtoje viršūnei 1 aibėje nėra *p* (priskirtoji aibė tuščia).

Taip pat $\Phi, 1 \nVdash \neg p$, nes tam, kad būtų $\Phi, 1 \Vdash p$, visose viršūnėse, į kurias galime patekti iš pirmos viršūnė, turėtų nebūti *p* (t.y. visose viršūnėse, į kurias galima patekti iš pirmos, turi būti neįvykdoma *p*). Bet taip nėra. Galima patekti į viršūnę 2 (jai priskirtoje aibėje yra *p*).

Vadinasi $\Phi, 1 \nVdash p \vee \neg p$. Radome struktūrą ir joje pasaulį, kuriame neįvykdoma $p \vee \neg p$. Taigi, $p \vee \neg p$ nėra intuicionistinės logikos dėsnis.

Kripke's struktūros naudojamos, kaip taisyklė, kai norima įrodyti, kad formulė neįvykdoma.

Aprašysime natūraliosios dedukcijos skaičiavimą. Jame išvedamos formulės yra tapačiai teisingos remiantis Kripke's semantika. Jos vadinamos intuicionistinės logikos dėsniais.

Intuicionistinės natūraliosios dedukcijos teiginių skaičiavimas.

Aksioma: $\Gamma, F \vdash F$

Loginių operacijų taisyklės:

$\to$ įvedimas $\quad$ $\to$ eliminavimas

$$\frac{\Gamma, F \vdash G}{\Gamma \vdash F \to G} \qquad \frac{\Gamma \vdash F \quad \Gamma \vdash F \to G}{\Gamma \vdash G}$$

$\wedge$ įvedimas $\quad$ $\wedge_1$ eliminavimas $\quad$ $\wedge_2$ eliminavimas

$$\frac{\Gamma \vdash F \quad \Gamma \vdash G}{\Gamma \vdash F \wedge G} \qquad \frac{\Gamma \vdash F \wedge G}{\Gamma \vdash F} \qquad \frac{\Gamma \vdash F \wedge G}{\Gamma \vdash G}$$

$\vee_1$ įvedimas $\quad$ $\vee_2$ įvedimas $\quad$ $\vee$ eliminavimas

$$\frac{\Gamma \vdash F}{\Gamma \vdash F \vee G} \qquad \frac{\Gamma \vdash G}{\Gamma \vdash F \vee G} \qquad \frac{\Gamma \vdash F \vee G \quad \Gamma, F \vdash H \quad \Gamma, G \vdash H}{\Gamma \vdash H}$$

$\neg$ įvedimas $\quad$ $\neg$ eliminavimas

$$\frac{\Gamma, F \vdash}{\Gamma \vdash \neg F} \qquad \frac{\Gamma \vdash F \quad \Gamma \vdash \neg F}{\Gamma \vdash}$$

Struktūrinės taisyklės:

silpninimas

$$\frac{\Gamma \vdash G}{\Gamma, F \vdash G} \qquad\qquad \frac{\Gamma \vdash}{\Gamma \vdash F}$$

Čia Γ yra baigtinės formulių sekos (gali būti ir tuščios). *F, G, H* - formulės. Sekvencijų antecedente esančių formulių tvarka nesvarbi.

Pavyzdžiai.

1. $\neg(p \vee \neg p) \vdash \neg\neg p$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg(p \vee \neg p), \neg p \vdash \neg p}\oplus}{\neg(p \vee \neg p), \neg p \vdash \mathbf{p} \vee \neg \mathbf{p}} \quad \dfrac{}{\neg(p \vee \neg p), \neg p \vdash \neg(p \vee \neg p)}\oplus}{\text{taikoma taisyklė } \neg \quad \text{eliminavimas}}}{\neg(p \vee \neg p), \neg p \vdash}}{\neg(p \vee \neg p) \vdash \neg\neg \mathbf{p}}$$

2. $\vdash p \to (q \to (p \wedge q))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p, q \vdash p}\oplus \quad \dfrac{}{p, q \vdash q}\oplus}{p, q \vdash \mathbf{p} \wedge \mathbf{q}}}{p \vdash \mathbf{q} \to (\mathbf{p} \wedge \mathbf{q})}}{\vdash p \to (q \to (p \wedge q))}$$

3. $\vdash p \to \neg\neg p$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p, \neg p \vdash p}\oplus \quad \dfrac{}{p, \neg p \vdash \neg p}\oplus}{p, \neg p \vdash}}{p \vdash \neg\neg p}}{\vdash p \to \neg\neg p}$$

Intuicionistiniame sekvenciniame skaičiavime (žr. knygą [1]) yra *prastinimo taisyklė* (formulės kartojimo taisyklė), kuri būtina.

$$\frac{\Gamma, F, F \vdash \Delta}{\Gamma, F \vdash \Delta}$$

Mūsų skaičiavime jos nėra. Ji jame nereikalinga. Nes formulės, loginių ir kvantorinių operacijų taisyklėmis, nenutrinamos antecedente. Tai galima atlikti tik struktūrine taisykle, jei matome, kad išvedimo paieškai ji nereikalinga.

Formulių išvedimų paieška intuicionistiniame *natūraliosios dedukcijos* skaičiavime nėra lengva. Žymiai lengvesnė paieška ituicionistiniame *sekvenciniame* skaičiavime (žr. priedą B).

Deja, ne visuose informatikos ir matematikos taikymuose galime naudotis intuicionistiniu sekvenciniu skaičiavimu. Kai kuriuose taikymuose (pavyzdžiui, funkcinio programavimo kalbose) tenka naudoti intuicionistinį natūraliosios dedukcijos skaičiavimą. Literatūroje kai kada naudojamas ir iutuicionistinis natūraliosios dedukcijos variantas, kuriame nėra neigimo. $\neg p$ keičiamas formule $p \to k$ (čia *k* - loginė konstanta). Šis variantas pateiktas priede C.

Kvantorinis intuicionistinės logikos variantas gaunamas papildžius intuicionistinį natūralios dedukcijos teiginių skaičiavimą kvantorinėmis taisyklėmis, aprašytomis praeitame skyrelyje *natūralioji dedukcija*.

Užduotis pamąstymui. Formulė $p \vee \neg p \vee \neg\neg p$ neįrodoma intuicionistinėje logikoje (žr. šio skyriaus pratimą 9c). Bet, jei žinome, kad $p \vee \neg p$ neįrodoma, tai $\neg(p \vee \neg p) \vdash \neg\neg p$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg(p \vee \neg p), \neg p \vdash \neg p}\oplus}{\neg(p \vee \neg p), \neg p \vdash p \vee \neg p} \quad \dfrac{}{\neg(p \vee \neg p), \neg p \vdash \neg(p \vee \neg p)}\oplus}{\neg(p \vee \neg p), \neg p \vdash}}{\neg(p \vee \neg p) \vdash \neg\neg p}$$

4.5 Lambda skaičiavimas

Kai kada skirtingi objektai žymimi vienodai. Pavyzdžiui, $x^2 + 3$ gali būti termas, gali būti ir funkcija. Jei funkciją rašysime $\lambda x.x^2 + 3$, suprasdami, kad funkcijoje argumentu yra *x*, tai ji sintaksiškai skirsis nuo termo. Jei norime paskaičiuoti funkcijos reikšmę taške 5, tai rašome $(\lambda x.x^2 + 3)5$. Argumentas nebūtinai rašomas skliaustuose. Panašiai yra ir matematikoje. Pavyzdžiui, *sin x.* Funkcijos reikšmė bus $5^2 + 3$.

Lambda skaičiavimas įdomus tuo, kad daugelio kintamųjų funkcijos skaičiavimą redukuojame į vieno argumento funkcijos skaičiavimą. Paaiškinsiu pavyzdžiu. Turime dviejų argumentų funkciją $f(x, y) = x + y$.

$\lambda x.x + y$ yra vieno argumento *x* funkcija. $\lambda y.x + y$ irgi yra vieno argumento *y* funkcija. O kaip skaičiuojama funkcijos $\lambda x.(\lambda y.x + y)$ reikšmė? Tarkime, $x = 5, y = 2$. Skaičiuojame vieno argumento funkcijos $(\lambda x.(\lambda y.x + y))5$ reikšmę. Ji lygi $\lambda y.5 + y$. Dabar skaičiuojame kitos vieno argumento funkcijos $\lambda y.5 + y$ reikšmę taške 2. $(\lambda y.5 + y)2 = 7$. Dviejų argumentų funkcijos $f(5, 2)$ reikšmę paskaičiavom naudodami tik vieno argumento funkcijas.

Kintamuosius žymėsime mažomis lotyniškomis raidėmis.

$\lambda-$skaičiavimo termo apibrėžimas.

1. *Kintamasis yra termas.*

2. *Jei M, N yra termai, tai* (*MN*) *taip pat termas* (*aplikacija*).

3. *Jei x yra kintamasis, M - termas, tai* $\lambda x.M$ *taip pat termas* (*abstrakcija*).

Išorinius skliaustus įprasta praleisti. Bet, deja, lambda skaičiavime dažnai praleidžiami ne tik išoriniai. Reikia atsiminti, kad termas *MNE* lygus ((*MN*)*E*). Termas $\lambda x.M$ lygus $(\lambda x.(M))$. Termas $\lambda x.\lambda y.\lambda z.M$ lygus $(\lambda x.(\lambda y.(\lambda z.M)))$. Nepatyrusiam dažnai skliaustai sukelia painiavą. Bet, ką padarysi, taip jau rašoma.

Kaip matome, nei λ, nei λx nėra termai. Terme *MN M* gali būti funkcija, o *N* argumentas.

Termų pavyzdžiai:

$$(xy)z$$
$$(\lambda x.yx)z$$
$$(\lambda x.\lambda y.xyz)uv$$

Kaip ir predikatų logikos formulėse, taip ir λ-skaičiavimo termuose, kintamųjų įeitys (išskyrus tas, kurios yra tiesiogiai simbolio λ dešinėje) skirstomos į laisvąsias ir suvaržytąsias. Predikatų logikos formulėse, bent jau šioje knygoje, laisvieji ir suvaržytieji kintamieji žymimi skirtingomis raidėmis. Deja, laisvieji ir suvaržytieji kintamieji lambda skaičiavime žymimi, taip jau priimta, tomis pačiomis raidėmis. Todėl reikalingas apibrėžimas, kada kintamojo įeitis laikoma laisvąja, o kada suvaržytąja.

1. *Jei termas E yra kintamasis x, t.y.* $E = x$, *tai kintamojo x įeitis terme E yra laisva.*

2. *Jei termas E sintaksiškai lygus termui* (*MN*), *tai visos laisvosios x įeitys termuose M, N yra laisvosios ir terme E.*

3. *Jei termas E sintaksiškai lygus termui* $\lambda y.M$ *ir* $x \neq y$, *tai visos laisvosios x įeitys terme M yra laisvosios ir terme E, o jei* $x = y$ *tai terme E nėra laisvųjų x įeičių.*

Pavyzdžiui, terme $x((\lambda x.xy)x)$ pirmoji, ketvirtoji *x* įeitys ir *y* įeitis yra laisvosios. Trečioji *x* įeitis yra suvaržyta.

Jei terme nėra laisvųjų kintamųjų įeičių, tai jis vadinamas *uždaruoju*. Pavyzdžiui, termas $\lambda x.\lambda y.(xyx)$ yra uždaras.

Termai *M, N* vadinami α-ekvivalenčiais, jei

1. *M* yra gautas iš *N*, pervardijus visas kurio nors kintamojo laisvąsias įeitis naujais, neįeinančiais į *N* kintamaisiais.

2. *M* gautas iš *N*, pakeitus kurio nors termo $\lambda x.E$ įeitį termu $\lambda z.E'$. Čia *E'* gautas iš *E*, pervardijus visas laisvąsias *x* įeitis terme *E* kintamuoju *z*, kuris yra naujas, neįeinantis į *E* kintamasis.

λ-skaičiavime du termai laikomi lygiais, jei jie yra α-ekvivalentūs. Sakoma, kad lambda skaičiavime nagrinėjami termai su tikslumu iki α-ekvivalentumo. Pavyzdžiui, termai $(\lambda x.x)(\lambda y.xy)$ ir $(\lambda z.z)(\lambda y.xy)$ yra α-ekvivalentūs.

Terme *E* kintamąjį *x* pakeitę termu *Y*, gauname naują termą *E'* ir žymime $E' = E[Y/x]$.

Apibrėžimas. *Termas* $(\lambda x.E)Y$ *vadinamas* ***redeksu****, o* $E[Y/x]$ *- jo santrauka.*

Lambda skaičiavime galima apskaičiuoti visas rekursyviąsias funkcijas. Apskaičiuojant funkciją vykdomi tam tikri algoritmo žingsniai (vadinami redukcijos žingsniais). Funkcijos (ji nusakoma lambda skaičiavimo termu) apskaičiavimo žingsniai vadinami termo β-redukcija. Ji vykdoma tokius būdu: ieškomas pirmas iš kairės redeksas ir jis keičiamas jo santrauka. Tai vadinama redukcijos žingsniu ir žymima $\beta \triangleright$. Gautasis termas vėl peržiūrimas iš kairės į dešinę ir, jei randame redeksą, keičiame jį jo santrauka. Jei terme daugiau nebėra redeksų, t.y. neįmanoma jo daugiau redukuoti, tai jis vadinamas **normalizuotu**, Termas, į kurį redukavome pradinį, vadinamas *normaliąja forma*.

Pavyzdžiai.

$$(\lambda x.x)y \ \beta \triangleright y,$$

$$(\lambda x.y)z \ \beta \triangleright y,$$

$$(\lambda x.x(xy))v \ \beta \triangleright v(vy).$$

Termas vadinamas *nenormalizuojamu*, jei jo neįmanoma redukuoti į normaliąją formą.

Pavyzdžiui, termas $(\lambda x.(xx))(\lambda x.(xx))$ nenormalizuojamas:

$$(\lambda x.(xx))(\lambda x.(xx)) \ \beta \triangleright \ (\lambda x.(xx))(\lambda x.(xx)) \ \beta \triangleright \ldots$$

$E^k M$ žymime termą $E(\ldots(E(EM))\ldots)$; čia *E* kartojamas *k* kartų. Jei $k = 0$, tai $E^k M = M$.

λ-skaičiavimo natūraliuoju skaičiumi (*Church'o skaičiumi*) *vadiname termą* $\underline{k} = \lambda f.\lambda x.(f^k x)$.

Dalinė funkcija $g(x_1, \ldots, x_n)$ *definuojama λ-skaičiavimo termu E, jei, kokie bebūtų natūralieji* $k_1, \ldots, k_n$ *iš to, kad* $g(x_1, \ldots, x_n) = k$*, išplaukia, kad* $(\ldots((E\underline{k_1})\underline{k_2})\ldots)\underline{k_n}$ *redukuojamas į normalųjį termą* $\underline{k}$*. Jei* $g(x_1, \ldots, x_n)$ *neapibrėžta, tai* $(\ldots((E\underline{k_1})\underline{k_2})\ldots)\underline{k_n}$ nenormalizuojamas.

λ-skaičiavimas yra dar vienas algoritmiškai apskaičiuojamų funkcijų formalizmas. Kiekvieną dalinę rekursyviąją funkciją galima definuoti λ-skaičiavimo termu.

Atrodytų, kad dalinės funkcijos apskaičiavimas λ-skaičiavime yra ganėtinai paprastas. Apskaičiuojant tereikia tik kai kuriuos kintamuosius *pervardinti* (α-redukcija) arba kintamąjį *pakeisti termu* (β-redukcija). Deja, studentams tai sunkiai sekasi. Pagrindinė priežastis - painiava su skliaustais. Gal geriau būtų nukrypti nuo tikslaus termo apibrėžimo ir naudoti laužtinius [,] bei figurinius $\{, \}$ skliaustus, esant susitarimamas dėl termų lygybės:

a) termas *MNE* lygus $((MN)E$,
b) termas λxM lygus $(\lambda x.(M))$,
c) termas $\lambda x.\lambda y.\lambda zM$ lygus $(\lambda x.(\lambda y.(\lambda z.M)))$.

Loginėms konstantoms *t, k* priskiriami tokie λ-skaičiavimo termai:

$$t = \lambda x.\lambda y.x$$
$$k = \lambda x.\lambda y.y$$

Loginių operacijų definavimas termais:

neigimas termu $\lambda x.((x0)1)$
konjunkcija termu $\lambda x.\lambda y.((xy)x)$
disjunkcija termu $\lambda x.\lambda y.((xx)y)$.

Reiškinys *if B then P else Q* nusakomas termu *BPQ* (čia *P, Q* yra kurie nors λ-skaičiavimo termai, o *B* - termas nusakantis loginę konstantą).

Pavyzdžiui, jei termu *B* nusakome loginę konstantą *t*, tai

if t then P else Q =
if $\lambda x.\lambda y.x$ *then P else Q =*
$(\lambda x.\lambda y.x)PQ\ \beta\ \triangleright$
$(\lambda y.P)Q\ \beta\ \triangleright$ *P.*

4.6 Termų tipizavimas

Formulėse aptinkami skirtingi objektai: individinės konstantos, funkcijos, predikatai. Funkcijos apibrėžimo sritimi gali būti natūraliųjų, sveikųjų ar dar kuri nors aibė. Kad būtų aiškiau kokius objektus naudojame formulėse, kokios jų apibrėžimo bei reikšmių aibės, jiems priskiriami reiškiniai, vadinami *tipais*. Prisiminkime įprastą matematikoje užrašą: $f : N \to R$, kuriuo nurodome, kad nagrinėjama funkcija apibrėžta natūraliųjų skaičių aibėje, o realiųjų skaičių aibė yra jos reikšmių aibė. Sakome, kad funkcija yra tipo $N \to R$. Taigi, reiškinys $N \to R$ yra tipas. *Tipas* yra simbolis nusakantis aibę objektų, tenkinančių tam tikras sąlygas. Mūsų atveju, tipu $N \to R$ nusakome aibę funkcijų, apibrėžtų natūraliųjų skaičių aibėje, kurių reikšmės yra realieji skaičiai. Užrašu $f : N \to R$ nurodome, kad funkcija *f* priklauso tai klasei (yra jos elementas).

Jei, pavyzdžiui, individinėms kontantoms bei kintamiesiems priskirsime tipą ι (žymėsime $x : \iota$, o loginiams kintamiesiems ir konstantoms *t, k* tipą o (t : o, k : o), tai predikatui *P* bus priskiriamas tipas $\iota \to o$. Žymime $P : \iota \to o$, nes predikatas apibrėžtas individinių konstantų aibėje (jų tipas ι) su reikšmėmis aibėje $\{t, k\}$. Loginių operacijų $\vee, \to, \wedge$ tipas būtų $o \to (o \to o)$, nes į logines operacijas galima žiūrėti kaip į dviejų argumentų funkcijas, kurių apibrėžimo ir reikšmių aibės yra $\{t, k\}$.

Nagrinėsime teiginių logikos formules virš aibės $\{\neg, \to\}$ (t.y formules, kuriose loginės operacijos gali būti tik iš aibės $\{\neg, \to\}$ ir jų išvedimus *intuicionistinės natūraliosios dedukcijos*

skaičiavime. Vietoje $\neg p$ naudosime $p \to k$ (čia *k* yra loginė konstanta *klaidinga*). Konstantai *k* priskirsime tipą $\bot$. ($k : \bot$).

Loginių operacijų pora $\{\neg, \to\}$ sudaro pilną sistemą *klasikinėje logikoje* (t.y. kiekvienai teiginių logikos formulei galima rasti jai ekvivalenčią, kurioje tėra tik loginės operacijos iš aibės $\{\neg, \to\}$), bet nesudaro pilnos *intuicionistinėje logikoje*.

λ-skaičiavimo termams priskirsime *tipus - teiginių logikos formules*. Kadangi vietoje $\neg p$ naudojame $p \to k$, tai nagrinėjamose formulėse iš loginių operacijų gali būti tik *implikacijos* įeitys ir išvedimo paieškai tereikia tik intuicionistinės (konstruktyvios) natūraliosios dedukcijos skaičiavimo *implikacijos taisyklių*.

Aprašysime kaip λ-skaičiavimo termams priskiriami tipai. Priskirimas pradedamas nuo kintamųjų. Kintamieji, kuriems priskirti tipai, vadinami *tipizuotais kintamaisiais*. Sąrašas tipizuotų kintamųjų Γ vadinamas *kontekstu*. Kontekstas susideda iš reiškinių pavidalo $x : A$ (*x* yra λ-skaičiavimo kintamasis, *A* - tipas (teiginių logikos formulė)). Be to, kiekvienas kintamasis yra tik vieno tipo.

Termų tipizavimas, esant duotam *termo kintamųjų kontekstui*, vykdomas prisilaikant tokių taisyklių:

Aksioma: $\Gamma \vdash x : A, \quad \text{jei} \quad x : A \in \Gamma$

Taisyklės:

$\to$ šalinimas

$$\frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash (MN) : B}$$

$\to$ įvedimas

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : A \to B}$$

Čia *M, N* yra λ-skaičiavimo termai.

Taisyklės skaitomos *iš viršaus į apačią*. Virš brūkšnio *duota* sekvencija, o apatinėje - *išvada*. Priminsime kodėl tokie taisyklių pavadinimai. Virš brūkšnio yra $\to$, o apatinėje nėra ($\to$ *šalinimas*). Virš brūkšnio nėra $\to$, o apatinėje atsiranda ($\to$ *įvedimas*).

λ-skaičiavimo tipizuoti termai formuojami iš jau turimų dvejomis taisyklėmis: aplikacija ($\to$ *šalinimas*) bei abstrakcija ($\to$ *įvedimas*).

Pavyzdys.

Antecedente surašomi tipizuoti suvaržytieji kintamieji, o sukcedente, taikant taisykles (*iš viršaus į apačią*) ieškomas duoto termo tipas (tipizuojame termą).

Tarkime, $\Gamma = \{x : p, y : p\}$. Tipizuojame termą $\lambda y.\lambda x.x$ (priskirsime duotam termui tipą).

$$x : p, y : p \vdash x : p$$
$$y : p \vdash \lambda x : p.x : p \to p$$
$$\vdash \lambda y : p.\lambda x : p.x : p \to (p \to p)$$

Atsakymas: $\lambda y : p.\lambda x : p.x : p \to (p \to p)$.

Tipų teorijoje priimta vietoje $A \to (B \to C)$ rašyti $A \to B \to C$. Tiksliau,

$$A_1 \to A_2 \to A_3 \to \ldots \to A_n$$

suprantamas kaip

$$A_1 \to (A_2 \to (A_3 \to \ldots \to A_n))\ldots).$$

Intuicionistinės natūraliosios dedukcijos aksiomomis yra sekvencijos pavidalo $\Gamma, A \vdash A$. Yra dvi taisyklės implikacijai. Jos panašios į termų tipizavimo taisykles:

$\to$ šalinimas

$$\frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}$$

$\to$ įvedimas

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$$

Curry-Howardo izomorfizmas: *egzistuoja abipusiai vienareikšmė atitiktis tarp tipizuotų termų ir formulių, išvedamų intuicionistinėje natūralioje dedukcijoje.*

Atitiktis:

formulė - tipas

išvedimas - programa (tipizuotas termas)

Pavyko susieti *mąstymą* (intuicionistinės natūraliosios dedukcijos skaičiavimą) su *algoritmiškai apskaičiuojamomis funkcijomis* (λ-skaičiavimo tipizuotais termais).

Aplikacija (*uv*) tarp dviejų tipizuotų termų $u : A, v : B$ galima tik atveju, jei *u* tipas yra pavidalo $B \to C$, t.y. $A = B \to C$. Todėl ne visi λ-skaičiavimo termai tipizuojami. Pavyzdžiui, termas $\lambda x.xx$ nėra tipizuojamas, nes aplikacijai (*xx*) neįmanoma priskirti tipo.

Įrodyta, kad *visi tipizuoti termai yra normalizuojami.*

Pavyzdžiais paaiškinsime Curry-Howardo izomorfizmą.

Kai termas yra pavidalo $\lambda x_1.\lambda x_2.\ldots.\lambda x_n.M$ ir terme *M* nėra λ įeičių, tai *M* vadinamas termo *kūnu*.

1. *Užduotis*. Tipizuoti termą $\lambda x.\lambda y.\lambda z.(yz)(xz)$, kai duotas kontekstas $\Gamma = \{x : p \to (q \to r), y : p \to q, z : p\}$.

Prisilaikysime tokios strategijos: visų pirma (tipizavimą vykdome **iš viršaus į apačią** taikydami *termų tipizavimo taisykles*) taikysime tik taisyklę $\to$ *šalinimas* tol, kol rasime termo *kūno tipą*. O po to, taikydami tik taisyklę $\to$ *įvedimas*, rasime *duoto termo tipą.*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash x : p \to (q \to r) \quad \Gamma \vdash z : p}{\Gamma \vdash xz : q \to r} \quad \dfrac{\Gamma \vdash y : p \to q \quad \Gamma \vdash z : p}{\Gamma \vdash yz : q}}{x : p \to (q \to r), y : p \to q, z : p \vdash (xz)(yz) : r}}{x : p \to (q \to r), y : p \to q \vdash \lambda z : p.(xz)(yz) : p \to r}}{x : p \to (q \to r) \vdash \lambda y : p \to q.\lambda z : p.(xz)(yz) : (p \to q) \to p \to r}}{\vdash \lambda x : p \to (q \to r).\lambda y : p \to q.\lambda z : p.(xz)(yz) : (p \to (q \to r)) \to (p \to q) \to p \to r}$$

Atsakymas:
$\lambda x : p \to (q \to r).\lambda y : p \to q.\lambda z : p.(xz)(yz) : (p \to (q \to r)) \to (p \to q) \to p \to r$

Pagal Curry-Howardo izomofizmą *formulė yra tipas*. Pažiūrėkite kaip paprastai gaunamas formulės $(p \to (q \to r)) \to (p \to q) \to p \to r$ išvedimas intuicionistinės natūraliosios dedukcijos skaičiavime. Tereikia termo tipizavimo išvedime palikti tik termų tipus (pas mus jie yra formulės).

Išvedime raide Δ žymime aibę formulių $p \to (q \to r), p \to q, p$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Delta \vdash p \to (q \to r)}\oplus \quad \dfrac{}{\Delta \vdash p}\oplus}{\Delta \vdash q \to r} \quad \dfrac{\dfrac{}{\Delta \vdash p \to q}\oplus \quad \dfrac{}{\Gamma \vdash p}\oplus}{\Delta \vdash q}}{p \to (q \to r), p \to q, p \vdash r}}{p \to (q \to r), p \to q \vdash p \to r}}{p \to (q \to r) \vdash (p \to q) \to p \to r}}{\vdash (p \to (q \to r)) \to (p \to q) \to p \to r}$$

2. *Užduotis.* Raskite formulės $(p \to q) \to ((p \to q) \to q) \to q$ išvedimą intuicionistinės natūraliosios dedukcijos skaičiavime ir, remiantis išvedimu, sukonstruokite formulei atitinkantį tipizuotą termą.

Išvedimas intuicionistinės natūraliosio dedukcijos skaičiavime (išvedimas *iš apačios į viršų)*:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p \to q, (p \to q) \to q \vdash (p \to q) \to q}\oplus \quad \dfrac{}{p \to q, (p \to q) \to q \vdash p \to q}\oplus}{p \to q, (p \to q) \to q \vdash q}}{p \to q \vdash ((p \to q) \to q) \to q}}{\vdash (p \to q) \to ((p \to q) \to q) \to q}$$

Pagal aksiomų (išvedimo galinių sekvencijų) antecedentuose esančias formules sudarome kontekstą. Tarkime, *u, v* λ-skaičiavimo kintamieji (t.y. kintamųjų vardus pasirenkame savo nuožiūra). Priskirkime jiems tipus:

$$u : p \to q, v : (p \to q) \to q.$$

Eidami *iš viršaus į apačią* priskirkime visoms formulėms, aptinkamoms išvedime, termus. Naudojame termų tipizavimo taisykles.

$$\dfrac{\dfrac{\dfrac{u : p \to q, v : (p \to q) \to q \vdash v : (p \to q) \to q \quad u : p \to q, v : (p \to q) \to q \vdash u : p \to q}{u : p \to q, v : (p \to q) \to q \vdash vu : q}}{u : p \to q \vdash \lambda v : (p \to q) \to q.vu : ((p \to q) \to q) \to q}}{\vdash \lambda u : u : p \to q.\lambda v : (p \to q) \to q.vu : (p \to q) \to ((p \to q) \to q) \to q}$$

Atsakymas: $\lambda u : u : p \to q.\lambda v : (p \to q) \to q.vu : (p \to q) \to ((p \to q) \to q) \to q.$

Termų tipizavimo taisykles galima papildyti ir formulėms, kuriose gali būti konjunkcija, disjunkcija. Taip pat galima aprašyti tipizavimo taisykles ir natūraliosio dedukcijos intuicionistiniam predikatų skaičiavimui.

Paaiškinsime tik konjunkcijos atvejui. Parodysime kaip randami termai, kai tipai yra intuicionistinės natūraliosios dedukcijos formulės virš aibės $\{\neg, \to, \wedge\}$. Klasikinėje logikoje $\wedge$ išreiškiamas per $\neg, \to$. Pavyzdžiui, taip: $p \wedge q \equiv \neg(p \to \neg q)$. Deja, intuicionistinėje logikoje konjunkcija neišreiškiama per $\neg, \to$. Reikia aprašyti kaip gaunami termai, kai taikomos intuicionistinės logikos konjunkcijos taisyklės. Jos yra trys:

$\wedge$ įvedimas $\qquad$ $\wedge_1$ eliminavimas $\qquad$ $\wedge_2$ eliminavimas

$$\dfrac{\Gamma \vdash F \quad \Gamma \vdash G}{\Gamma \vdash F \wedge G} \qquad \dfrac{\Gamma \vdash F \wedge G}{\Gamma \vdash F} \qquad \dfrac{\Gamma \vdash F \wedge G}{\Gamma \vdash G}$$

Termų tipizavimas esant duotam *termų kintamųjų kotekstui* vykdomas prisilaikant tokių taisyklių:

$$\dfrac{\Gamma \vdash M : F \quad \Gamma \vdash N : G}{\Gamma \vdash (M, N) : F \wedge G} \qquad \dfrac{\Gamma \vdash M : F \wedge G}{\Gamma \vdash fstM : F} \qquad \dfrac{\Gamma \vdash M : F \wedge G}{\Gamma \vdash sndM : G}$$

Čia *M, N* - termai, *F, G* - formulės. O (MN), $fstM$, $sndM$ nusakomi taip:

$$(M, N) = \lambda z.zMN$$

$fstM = Mt$; $\quad$ Tarkime, $M = KN \quad fst(K, N) = (\lambda z.zKN)t = \lambda x.\lambda y.xKN = K$

$sndM = Mk$; $\quad$ Tarkime, $M = KN \quad snd(KN) = (\lambda z.zKN)k = \lambda x.\lambda y.yKN = N$

Priminsime, kad $t = \lambda x.\lambda y.x$, $k = \lambda x.\lambda y.y$.

Du paprastosios tipų teorijos taikymai:

1. Natūraliosios dedukcijos predikatų logikos skaičiavimus, naudojant kintamuosius (individinius, funkcinius, predikatinius) su tipais, sudaroma galimybė nagrinėti matematikos teiginius, kurių įrodymui reikalinga **aukštesnės eilės logika**.

2. Į λ-skaičiavimą galima žiūrėti kaip į paprastą programavimo kalbą. λ-skaičiavimo tipizuotas termas definuoja programą, kurios specifikacija aprašoma formule (tipu). Formulės įrodymas *užtikrina programos korektiškumą*.

Aukštesnės eilės logika

Loginiai skaičiavimai skirstomi pagal eiles į pirmos, antros, trečios ir. t.t. Visų *aprašomų skaičiavimų* taisyklės panašios. Skiriasi tik kvantorinės taisyklės. Taikant jas skirtingos eilės logikose parenkami skirtingo tipo kintamieji bei konstantos. Naudojant tipus loginiams reiškiniams galima skirtingų eilių skaičiavimus apjungti į vieną. Toks skaičiavimas (jis gali būti sekvencinis, natūraliosios dedukcijos ar Hilberto tipo) vadinamas *aukštesnės eilės logikos* (Higher-Order Logic) skaičiavimu.

Paaiškinsime tik kaip apjungiami pirmos ir antros eilės logikų skaičiavimai su funkciniais simboliais.

Paprastų tipų abės *T* **apibrėžimas**:

1. ι yra individinių konstantų bei kintamųjų tipas,
2. o yra loginių konstantų $\{t, k\}$ bei loginių kintamųjų tipas,
3. jei τ, σ yra tipai, tai $\tau \to \sigma$ irgi yra tipas (funkcijos iš τ į σ tipas).

Loginėms operacijoms priskiriami tipai:

1. neigimui priskiriamas tipas $o \to o$; neigimas yra funkcija apibrėžta loginių konstantų aibėje su reikšmėmis irgi loginių konstantų aibėje,
2. $\wedge, \vee, \to, \leftrightarrow$ priskiriamas tipas $o \to (o \to o)$. Žiūrima kaip į dviejų argumentų funkcijas. Daugelio argumentų funkcijas galima redukuoti į vieno argumento funkcijų skaičiavimą.

Individinėms konstantoms ir kintamiesiems bei funkciniams simboliams ir funkcijoms priskiriami tipai iš aibės

$$T_f = \{\iota, \iota \to \iota, \iota \to (\iota \to \iota), (\iota \to \iota) \to \iota, \ldots\}.$$

Pavyzdžiui, jei *x* individinis kintamasis, tai jo tipas ι (žymima x^{ι}). Jei *x* yra vieno argumento funkcija (tipas $\iota \to \iota$), tai žymima $x^{\iota \to \iota}$. Jei *x* dviejų argumentų funkcija, tai tipas $\iota \to (\iota \to \iota)$. Jei funkcija yra dviejų vieno argumento funkcijų kompozicija, tai jos tipas $(\iota \to \iota) \to \iota$.

Loginėms konstantoms ir kintamiesiems bei predikatiniams kintamiesiems ir predikatams priskiriami tipai iš aibės

$$T_P = \{o, \iota \to o, \iota \to (\iota \to o), (\iota \to \iota) \to o, \ldots\}.$$

Pavyzdžiui, vienviečiam predikatiniam kintamajam ar predikatui priskiriamas tipas $\iota \to o$, kai argumentu yra individinis kintamasis ar konstanta. Jei vienviečio pradikato argumentu yra vieno argumento funkcija, tai tipas $(\iota \to \iota) \to o$.

Dabar pirmos ir antros eilės logikos taisykles galime apjungti. Pavyzdžiui, natūraliosios dedukcijos taisyklę

$\forall$ įvedimas
$$\frac{\Gamma \vdash F(\beta)}{\Gamma \vdash \forall x F(x)}$$

β yra *naujas* laisvasis kintamasis.

Galima pakeisti taisykle

$$\frac{\Gamma \vdash F(y^\tau)}{\Gamma \vdash \forall x^\tau F(x^\tau)}$$

Čia y^τ yra naujas individinis, predikatinis kintamasis ar funkcinis simbolis (priklausomai nuo tipo τ).

Jei $\tau = \iota$, visose taikomose kvantorinėse išvedimo taisyklėse, tai turim pirmos eilės logiką. Priešingu atveju turime antros eilės logiką.

Taip galima apjungti visų eilių logikų skaičiavimus (ne tik pirmos ir antros eilės) į vieną. Toks skaičiavimas vadinamas *aukštesnės eilės logikos skaičiavimu*.

4.7 Pratimai

1. Transformuoti į normaliąją konjunkcinę formą naudojant teisingumo lenteles:

a) $p \to q$.

b) $p \to (q \wedge r)$.

2. Transformuoti į normaliąją konjunkcinę formą naudojant ekvivalenčias formules:

a) $p \wedge (q \to (\neg p \vee r))$,

b) $\neg(p \to (q \vee r)) \wedge ((q \to r) \to p)$,

c) $((p \wedge q) \vee r) \wedge (p \to (\neg q \to (p \vee r)))$.

3. Išveskite rezoliucijų metodu:

a) $r \to p, r \to (p \to q) \vdash r \to q$,

b) $p \wedge (q \vee (p \to q)) \vdash q$,

c) $(p \wedge r) \vee (q \wedge r) \vdash (p \vee q) \wedge r$.

4. Kurios yra Horno formulės:

a) $(p \wedge q) \to r$,

b) $((p \wedge q \wedge r) \to s) \wedge \neg q$,

c) $((p \wedge q \wedge \neg r) \to s) \wedge q$,

d) $((p \wedge q) \to r) \wedge s \wedge r$?

5. Raskite sekvencijų išvedimus rezoliucijų metodu:

a) $\exists x Q(x), \forall x(Q(x) \to (P(x) \vee R(x))) \vdash \exists x(\neg P(x) \to R(x))$,

b) $\exists x(P(x) \to Q(x)), \forall x(Q(x) \to R(x)) \vdash \forall x P(x) \to \exists x R(x)$,

c) $\forall x \neg M(x), \neg \exists x \forall y(\neg M(x) \wedge \neg P(x, y)) \vdash \forall x \exists y P(x, y)$.

6. Rezoliucijų metodu nustatykite ar formulė
$\exists x \forall y \forall z \exists v((P(x) \vee \neg P(y)) \wedge (P(z) \vee \neg P(v)))$ tapačiai teisinga?

7. Rezoliucijų metodu nustatykite ar formulė
$\forall x(P(x) \to P(f(x))) \wedge P(a) \wedge \neg P(f(f(f(a))))$ tapačiai klaidinga?

8. Išveskite natūraliosios dedukcijos klasikiniame skaičiavime formules:

a) $\neg F \wedge \neg G \vdash \neg(F \vee G)$,

b) $F \to (G \to H) \vdash (F \wedge G) \to H$,

c) $F \to G \vdash \neg G \to \neg F$,

d) $A(a) \to \forall x B(x) \vdash \forall x(A(a) \to B(x))$,

e) $\vdash \forall x(A(x) \to B(a)) \to \exists x(A(x) \to B(a))$,

9. Naudodami Kripke's struktūras įrodykite, kad šios formulės nėra intuicionistinės logikos dėsniai:

a) $(p \to q) \to (\neg p \vee q)$,

b) $\neg(p \wedge q) \to (\neg p \vee \neg q)$,

c) $p \vee \neg p \vee \neg\neg p$.

10. Raskite sekvencijų išvedimus intuicionistiniame natūraliosios dedukcijos skaičiavime:

a) $\vdash \neg(p \wedge \neg p)$,

b) $\vdash (\neg p \vee q) \to (p \to q)$,

c) $\vdash (\neg p \wedge \neg q) \to \neg(p \vee q)$,

d) $\vdash \neg\neg\neg p \to \neg p$.

11. Redukuokite termus:

a) $(\lambda x.\lambda y.y(xxy))fg$,

b) $(\lambda y.yzy)(\lambda x.cd)$,

c) $\lambda x.\lambda y.(\lambda z.zv)yx$,

d) $(\lambda x.((\lambda y.xy)u))(\lambda v.v)$,

e) $\lambda f.\lambda x.(\lambda x.fx)((\lambda x.fx)x)$,

f) $(\lambda x.\lambda y.yx)zv$,

g) $(\lambda x.(\lambda y.yx)z)v$,

h) $(\lambda x.\lambda y.x(xy))fu$,

i) $\lambda x.\lambda y.x((\lambda x.\lambda y.x(x(xy)))xy)$.

12. Kurios kintamųjų *x, y, z* įeitys laisvos?

a) $(\lambda x.((\lambda y.xy)z))(\lambda z.z)(\lambda z.zy)yx$,

b) $\lambda z.\lambda x.(\lambda x.zx)((\lambda x.fx)z(\lambda x.\lambda y.yzx)$,

c) $(\lambda x.(\lambda y.yx)z)x(\lambda x.\lambda y.z(xy))y$.

13. Tipizuoti termą $\lambda x.\lambda y.\lambda z.y(xz)$, kai duotas kontekstas $\Gamma = \{x : p \to q, y : q \to r, z : p\}$.

14. Rasti termą, kurio tipas yra *s*, naudojant tik taisyklę $\to$ *šalinimas.* Kontekstas $\Gamma = \{x : p, u : p \to q, y : p \to (p \to r), z : r \to (q \to s)\}$

15. Naudojant tipizavimo taisykles įrodyti, kad termo $\neg((\leftrightarrow\ ((\vee q)r))p)$ tipas o. Kontekstas $\Gamma = \{p : o, q : o, \neg : o \to o, \vee : o \to (o \to o), \leftrightarrow : o \to (o \to o)\}$.

16. Raskite λ-termo $\lambda x.x\lambda y.y\lambda z.x\lambda u.z$ tipo $((((p \to q) \to p) \to p) \to q) \to q$ išvedimą.

5 Skyrius
PRIEDAI

A. Kai kurios išsprendžiamos predikatų logikos klasės

Nagrinėjame normaliosios priešdėlinės formos pirmos eilės predikatų logikos formules, t.y. formules pavidalo $Q_1x_1Q_2x_2\dots Q_nx_nG$; čia *G* - formulė, kurioje nėra kvantorių (bekvantorė formulė, dar vadinama *matrica*), o $Q_1x_1Q_2x_2\dots Q_nx_n$ vadiname formulės *prefiksu* ($Q_i \in \{\forall, \exists\}\ \ (i = 1, \dots, n)$). *Prefikso forma* vadiname žodį $Q_1Q_2\dots Q_n$.

Pavyzdžiai.

1. Formulės $\forall x\exists y(P(x) \wedge Q(y))$ prefiksas $\forall x\exists y$, o prefikso forma $\forall\exists$.

2. Formulės $\forall x\forall y\forall z((L(x, y) \wedge L(y, z)) \to L(x, z))$ prefiksas $\forall x\forall y\forall z$, o prefikso forma $\forall\forall\forall$ (sutrumpintai žymėsime $\forall^3$).

3. Formulės $\forall x\forall y\exists z\exists v((F(x) \to F(y)) \to (\neg F(z) \to \neg F(v)))$ prefiksas $\forall x\forall y\exists z\exists v$, o prefikso forma $\forall\forall\exists\exists$ (sutrumpintai žymėsime $\forall^2\exists^2$).

Formulės vadinamos *baigiai įvykdomos*, jei jos įvykdomos, kurioje nors baigtinėje individinių konstantų aibėje. Kai kuriais atvejais, pagal formulių pavidalą, galima nurodyti tokį natūralųjį *n*, kad formulė įvykdoma tada ir tiktai tada, kai ji įvykdoma kurioje nors individinių konstantų aibėje susidedančioje iš ne daugiau kaip *n* elementų. Tokių formulių aibės (klasės) yra išsprendžiamos *pagal įvykdomumą* - egzistuoja algoritmas, kuriuo mokame nustatyti ar formulės įvykdomos.

Išvardinsime gerai žinomas išsprendžiamų baigiai įvykdomų formulių klases:

1. Klasė formulių, kuriose yra tik vienviečiai predikatiniai kintamieji.

2. Klasė formulių, kurių prefiksų forma yra pavidalo $\exists^m\forall^n\ \ (m = 0, 1, 2, \dots; n = 0, 1, 2\dots$ ir, suprantama, $m + n \neq 0$).

3. Klasė formulių, kurių prefiksų forma yra pavidalo $\exists^m\forall\exists^n$ $(m = 0, 1, 2, \dots; n = 0, 1, 2, \dots)$.

4. Klasė formulių, kurių prefiksų forma yra pavidalo $\exists^m\forall^2\exists^n$ $(m = 0, 1, 2, \dots; n = 0, 1, 2, \dots)$.

Naudojama ir sąvoka *išsprendžiamumas pagal įrodomumą* (tapačiai teisingumo nustaty-

mas). Tuomet išsprendžiamos klasės perrašomos taip: $\forall$ keičiamas egzistavimo kvantoriumi $\exists$, o $\exists$ keičiama bendrumo kvantoriumi $\forall$. Pavyzdžiui, ketvirtoji formulių klasė *pagal įrodomumą* aprašoma taip:

klasė formulių, kurių prefiksų forma yra pavidalo $\forall^m\exists^2\forall^n$ $(m = 0, 1, 2, \ldots; n = 0, 1, 2, \ldots)$.

Pagal įrodomumą reiškia, kad kalbame apie formules, kurios yra sekvencijos sukcedente: egzistuoja algoritmas, kuriuo galima nustatyti ar sekvencija $\vdash F$ išvedama, kai formulės prefikso forma $\forall^m\exists^2\forall^n$.

B. Intuicionistinis sekvencinis skaičiavimas

Aksiomos: $\Gamma, F \vdash F$

Struktūrinės taisyklės:

Silpninimas

$$\frac{\Gamma \vdash}{\Gamma \vdash F} \qquad \frac{\Gamma \vdash \Delta}{F, \Gamma \vdash \Delta}$$

Prastinimas

$$\frac{F, F, \Gamma \vdash \Delta}{F, \Gamma \vdash \Delta}$$

Loginių operacijų taisyklės:

$$(\neg \vdash)\ \frac{\Gamma \vdash F}{\neg F, \Gamma \vdash \Delta} \qquad (\vdash \neg)\ \frac{F, \Gamma \vdash}{\Gamma \vdash \neg F}$$

$$(\wedge \vdash)\ \frac{F, G, \Gamma \vdash \Delta}{F \wedge G, \Gamma \vdash \Delta} \qquad (\vdash \wedge)\ \frac{\Gamma \vdash F \quad \Gamma \vdash G}{\Gamma \vdash F \wedge G}$$

$$(\vee \vdash)\ \frac{F, \Gamma \vdash \Delta \quad G, \Gamma \vdash \Delta}{F \vee G, \Gamma \vdash \Delta} \qquad (\vdash \vee)\ \frac{\Gamma \vdash F}{\Gamma \vdash F \vee G} \quad \text{arba} \quad \frac{\Gamma \vdash G}{\Gamma \vdash F \vee G}$$

$$(\to\vdash)\ \frac{\Gamma \vdash F \quad G, \Gamma \vdash \Delta}{F \to G, \Gamma \vdash \Delta} \qquad (\vdash\to)\ \frac{F, \Gamma \vdash G}{\Gamma \vdash F \to G}$$

Kvantorinės taisyklės:

$$(\exists \vdash)\ \frac{F(\beta), \Gamma \vdash \Delta}{\exists x F(x), \Gamma \vdash \Delta} \qquad (\vdash \exists)\ \frac{\Gamma \vdash F(\gamma)}{\Gamma \vdash \exists x F(x)}$$

$$(\forall \vdash)\ \frac{F(\gamma), \Gamma \vdash \Delta}{\forall x F(x), \Gamma \vdash \Delta} \qquad (\vdash \forall)\ \frac{\Gamma \vdash F(\beta)}{\Gamma \vdash \forall x F(x)}$$

Pjūvio taisyklė:

$$\frac{\Gamma_1 \vdash F \quad F, \Gamma_2 \vdash \Delta}{\Gamma_1, \Gamma_2 \vdash \Delta}$$

Δ žymi vieną formulę arba tuščią formulių aibę.
Raide *C* žymime aibę laisvųjų kintamųjų, esančių apatinėje sekvencijoje.
$(\exists \vdash)$ ir $(\vdash \forall)$ taisyklėse $\beta \notin C$. β yra *naujas* laisvasis kintamasis.
Taisyklėse $(\vdash \exists)$ ir $(\forall \vdash)$ $\gamma \in C$. Jei *C* tuščia aibė, tai γ naujas laisvasis kintamasis.

Kvantorinėje taisyklėje $(\forall \vdash)$ nėra formulės $\forall x F(x)$ kartojimo. Reikalui esant formulę galima kartoti remiantis *prastinimo* (kai *išvedimas iš apačios į viršų*) taisykle.

Sekvencija $\vdash F \vee \neg F$ intuicionistiniame skaičiavime išvedama tik tuo atveju, kai išvedama viena iš sekvencijų $\vdash F$, $\vdash \neg F$, o sekvencija $\vdash F \to \neg\neg F$ išvedama.

Pavyzdžiai. Išvedimuose nesunku matyti kurioms formulėms taikomos taisyklės. Todėl formulių paryškinimo nėra.

Sekvencijų $\vdash \neg(F \wedge \neg F)$ ir $\vdash \neg\neg\neg \to \neg F$ išvedimai:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{F \vdash F}\,\oplus}{F, \neg F \vdash}}{F \wedge \neg F \vdash}}{\vdash \neg(F \wedge \neg F)}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{F \vdash F}\,\oplus}{\neg F, F \vdash}}{F \vdash \neg\neg F}}{\neg\neg\neg F, F \vdash}}{\neg\neg\neg F \vdash \neg F}}{\vdash \neg\neg\neg \to \neg F}$$

Intuicionistiniame skaičiavime sekvencija $\vdash \neg\neg F \to F$ neišvedama, nebent būtų išvedama $\vdash F$.

Intuicionistinėje logikoje teisingi *teiginiai*:

1. Išvedamų sekvencijų aibė yra griežtas poaibis išvedamų klasikiniame sekvenciniame skaičiavime, t.y., jei kuri nors sekvencija $\Gamma \vdash \Delta$ išvedama intuicionistiniame skaičiavime, tai ji išvedama ir klasikiniame skaičiavime.

2. Intuicionistiniame skaičiavime galima apsieiti be pjūvio taisuklės, t.y. pjūvio taisyklė šiame skaičiavime turi *leistinos taisyklės* statusą.

Pjūvio taisyklė leistina, o be *prastinimo taisyklės* apsieiti negalime. Be jos, pavyzdžiui, sekvencija $\vdash \neg\neg(F \vee \neg F)$ neišvedama (žr. knygoje [1]).

Pratimai.

Išveskite sekvencijas (atsakymai ir sprendimai knygos pabaigoje):

1. $\vdash \neg\neg(F \to G) \to (F \to \neg\neg G)$,

2. $\vdash (H \to F) \to ((H \to (F \to G)) \to (H \to G))$,

3. $\vdash \exists x \exists y(\neg\neg(\neg\neg F(x) \to F(y)))$,

4. $\vdash \forall x \exists y((\neg\neg F(x) \to F(y)) \to (\neg\neg(F(x) \to F(y)) \to (F(x) \to F(y))))$,

5. $\exists x F(x) \vee \exists x G(x) \vdash \exists x(F(x) \vee G(x))$.

C. Intuicionistinės natūraliosios dedukcijos skaičiavimas be neigimo operacijos

Aprašysime dažnai literatūroje aptinkamą intuicionistinės natūraliosios dedukcijos variantą, kuriame nenaudojamas neigimo simbolis. Neigimas išreiškiamas implikacija ir logine konstanta *klaidinga*: $\neg F \equiv F \to \bot$. Tokiame skaičiavime nėra taisyklių neigimo operacijai. Nenaudojamas tuščias sukcedentas. Tuščias sukcedentas laikomas lygiu konstantai $\bot$. T.y. vietoje $\Gamma \vdash$ rašoma $\Gamma \vdash \bot$.

Aksiomos. $\Gamma, F \vdash F$.

Taisyklės.

$$(\to I)\ \frac{\Gamma, F \vdash G}{\Gamma \vdash F \to G} \qquad (\to E)\ \frac{\Gamma \vdash F \to G \quad \Gamma \vdash F}{\Gamma \vdash G}$$

$$(\wedge I)\ \frac{\Gamma \vdash F \quad \Gamma \vdash G}{\Gamma \vdash F \wedge G} \qquad (\wedge E)\ \frac{\Gamma \vdash F \wedge G}{\Gamma \vdash F} \qquad \frac{\Gamma \vdash F \wedge G}{\Gamma \vdash G}$$

$$(\vee I)\ \frac{\Gamma \vdash F}{\Gamma \vdash F \vee G} \quad \frac{\Gamma \vdash G}{\Gamma \vdash F \vee G} \qquad (\vee E)\ \frac{\Gamma \vdash F \vee G \quad \Gamma, F \vdash H \quad \Gamma, G \vdash H}{\Gamma \vdash H}$$

$$(\bot E)\ \frac{\Gamma \vdash \bot}{\Gamma \vdash F}$$

Pavyzdys. $\vdash (F \to \neg G) \to (G \to \neg F)$.
Įrodysime formulei $\vdash (F \to \neg G) \to (G \to \neg F)$ ekvivalenčią formulę $\vdash (F \to (G \to \bot)) \to (G \to (F \to \bot))$.

Išvedime raide Γ žymime formulių seką $F \to (G \to \bot), G$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, F \vdash G}\oplus \quad \dfrac{\dfrac{}{\Gamma, F \vdash F}\oplus \quad \dfrac{}{\Gamma, F \vdash F \to (G \to \bot)}\oplus}{\Gamma, F \vdash G \to \bot}}{\text{implikacijos eliminavimas } (\to E)}}{\Gamma \vdash \mathbf{F} \to \bot}}{F \to (G \to \bot) \vdash \mathbf{G} \to (\mathbf{F} \to \bot)}}{\vdash (F \to (G \to \bot)) \to (G \to (F \to \bot))}$$

Jei sekvencijoje nėra neigimo, tai ji išvedama taip pat kaip ir intuicionistiniame natūraliosios dedukcijos skaičiavime, aprašytame skyrelyje 4.4.

Pratimai.

Išveskite sekvencijas (atsakymai ir sprendimai knygos pabaigoje):

1. $\neg F \to \neg\neg\neg F$,

2. $\vdash (F \to G) \to (\neg G \to \neg F)$.

D. Davido Hilberto skaičiavimas

D.Hilbertas (1862 - 1943) buvo universalus matematikas. Dideli jo nuopelnai daugelyje matematikos šakų: geometrijoje, funkcionalinėje analizėje, diferancialiniame skaičiavime, skaičių teorijoje, matematinėje fizikoje, matematinėje logikoje, algebroje. Jis "matė visą matematiką", o ne kurią nors vieną sritį. Pastebėjo, kad visose matematikos šakose mąstymo taisyklės vienos ir tos pačios. Aprašė jas. Vėliau buvo įrodyta kad jos sudaro pilną (formulė įrodoma tada ir tiktai tad, kai ji tapačiai teisinga) ir neprieštaringą loginį skaičiavimą. Tai buvo pirmasis tokios tipo loginis skaičiavimas. Juo apimama dalis matematikoje naudojamų taisyklių. Vėliau, remiantis skaičiavimu, aprašyti kiti žinomi loginiai skaičiavimai (natūralioji dedukcija, sekvencinis skaičiavimas, semantinių lentelių metodas, rezoliucijų metodas).

Teiginių skaičiavime yra 11 aksiomų schemų ir viena taisyklė. Formulės įrodymo paieška ganėtinai sudėtinga ir todėl logikos taikymuose retai naudojama. Yra sukurta daug *patogesnių išvedimo paieškai* loginių skaičiavimų. Bet tai yra *pirmasis pilnas ir neprieštaringas* loginis skaičiavimas (dažniausiai naudojamas teoriniuose darbuose). Juk tik jo dėka ir atsirado patogesni skaičiavimai.

Aksiomos (*A, B, C* - bet kurios formulės):

1.1 $A \to (B \to A)$,

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

2.1 $(A \wedge B) \to A$,

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

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

3.1 $A \to (A \vee B)$,

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

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

4.1 $(A \to B) \to (\neg B \to \neg A)$,

4.2 $A \to \neg\neg A$,

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

Šios 1.1 - 4.3 aksiomos dar vadinamos *aksiomų schemomis*. Skaičiavime yra be galo daug aksiomų. Jos gaunamos iš aksiomų schemų keičiant *A, B, C* bet kuriomis formulėmis.

Vienintelė teiginių skaičiavimo taisyklė yra *modus ponens* (MP):

$$\frac{A \quad A \to B}{B}$$

čia *A* ir *B* - bet kurios formulės.

Įrodymu teiginių skaičiavime vadiname baigtinę formulių seką, kurioje kiekviena formulė yra arba aksioma, arba gauta iš prieš ją esančių formulių pagal modus ponens taisyklę. Sakome, kad formulė A įrodoma teiginių skaičiavime (žymime $\vdash A$), *jei galime rasti įrodymą, kurio paskutinis narys A.*

Tarkime, norime įrodyti formulę *F.* Kurioje nors aksiomų schemoje pavyko pakeisti *A, B, C* taip, kad:

- gautoji aksioma yra pavidalo $G_1 \to (G_2 \to (\dots (G_n \to F) \dots))$,
- $G_1, G_2, \dots, G_n$ yra aksiomos.

Tuomet *F* įrodymas atrodys šitaip:

1. $G_1 \to (G_2 \to (\dots (G_n \to F) \dots))$ aksioma,

2. G_1 aksioma,

3. $G_2 \to (\dots (G_n \to F) \dots))$ yra gauta pagal MP taisyklę iš 1 ir 2 formulių,

4. G_2 aksioma,

5. $G_3 \to (\dots (G_n \to F) \dots))$ gauta pagal MP taisyklę iš 3 ir 4 formulių,

. . .,

$n + 2$. $G_n \to F$ gauta pagal MP taisyklę iš n ir $n + 1$ formulių,

G_n aksioma,

$n + 4$. F gauta pagal MP taisyklę iš $n + 2$ ir $n + 3$ formulių.

Tarkime, nepavyko gauti tokios aksiomos $G_1 \to (G_2 \to (\dots (G_n \to F) \dots))$, kurioje visos formulės $G_1, G_2, \dots, G_n$ yra aksiomos (tarkime, G_i nėra aksioma). Tuomet ieškome tokių formulių *A, B, C* keitinių ir tokios aksiomų schemos, kad $H_1 \to (H_2 \to (\dots (H_m \to G_i) \dots))$ bei $H_1, H_2, \dots, H_m$ būtų aksiomos. Taikydami taisyklę MP rasime G_i išvedimą.

Jei kuri nors iš $H_1, H_2, \dots, H_m$ nėra aksioma, tai vėl ieškome aksiomų chemos ir mums tinkamo keitinio formulėms *A, B, C* ir t.t.

Pavyzdžiai.

1. Parodysime, kad formulė $A \to A$ t.y. sekvencija $\vdash A \to A$, kurioje prielaidų sąrašas tuščias, yra įrodoma.

Pastaba. Šiuo pavyzdžiu pradedamas išvedimų Hilberto skaičiavime demonstravimas beveik visose studentams skirtose matematinės logikos knygose, tame tarpe ir [1].

Formulėje yra tik viena loginė operacija - *implikacija*. Todėl formulės įrodymui pakaks tik aksiomų schemų, kuriomis nusakomos implikacijos savybės, t.y. aksiomų schemų 1.1 ir 1.2. Nors implikacija ir įeina į visas aksiomų schemas, bet likusiose aksiomų schemose nusakomos *kitų loginių operacijų savybės* naudojant implikaciją.

Aksiomų schemoje 1.2 atliekame keitimą: keičiame *B* į $A \to A$, *C* į *A*. Gauname aksiomą

$$(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A)).$$

Pažymėkime raide G_1 formulę $A \to ((A \to A) \to A)$,

raide G_2 formulę $A \to (A \to A)$,

raide *F* mūsų tikslą, formulę $A \to A$.

Gauta aksioma yra pavidalo $G_1 \to (G_2 \to F)$. G_1 bus aksioma, kai aksiomų schemoje 1.1 pakeisime *B* formule $A \to A$. G_2 taip pat bus aksioma, kai aksiomų schemoje 1.1 pakeisime *B* formule *A*.

Taigi, formulės $A \to A$ įrodymas Hilberto skaičiavime yra toks:

1. $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$ aksioma 1.2 $(B - A \to A; C - A)$,

2. $A \to ((A \to A) \to A)$ aksioma 1.1 $(b - A \to A)$,

3. $(A \to (A \to A)) \to (A \to A)$ MP 1 ir 2 formulėms,

4. $A \to (A \to A)$ aksioma 1.1 $(B - A)$,

5. $A \to A$ MP 3 ir 4 formulėms.

2. Rasime formulės $\neg\neg\neg\neg A \to A$ įrodymą.

Šioje formulėje yra dvi loginės operacijos - *neigimas* ir *implikacija*. Todėl formulės įrodymui turi pakakti aksiomų schemų, kuriomis aprašomos neigimo bei implikacijos savybės - 1.1, 1.2.

Paaiškinsime, kokias implikacijos savybes nusakome aksiomomis 1.1, 1.2.

Jei *F* yra aksioma arba prielaida (duota formulė be įrodymo), tai Hilberto skaičiavime įrodoma formulė $M \to F$ su bet kuria formule *M*. T.y., jei turime *F*, tai leidžiama prirašyti

prieš implikaciją bet kurią formulę *M*. *Neužmirškite to.* Tai dažnai naudojama įrodymuose savybė.

1. F aksioma (arba prielaida),

2. $F \to (M \to F)$ aksioma 1.1 ($A - F; B - M$(kuri nors formulė)),

3. $M \to F$ MP 1 ir 2 formulėms.

Aksioma 1.2 nusakome *implikacijos tranzityvumą.* Dažniausiai implikacijos tranzityvumas nusakomas teiginiu *jei* $A \to B$ *ir* $B \to C$, *tai* $A \to C$. T.y., jei įrodomos formulės $A \to B$ ir $B \to C$, tai įrodoma ir $A \to C$. Hilberto skaičiavime yra *savotiška* tranzityvumo aksioma

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

Tik nesuklyskite naudodami aksiomų schemą 1.2. Formules žyminčios raidės joje atitinka raidėms teiginyje *jei* $A \to B$ *ir* $B \to C$, *tai* $A \to C$. Tiesa, yra formulė $A \to (B \to C)$. Bet juk formulė $B \to C$ duota, o priekyje tereikia prirašyti *A* (panaudojus aksiomą 1.1) ir gauti $A \to (B \to C)$.

Taigi, turime (duota) $A \to B$ *ir* $B \to C$. Kaip įrodyti $A \to C$?

Tranzityvumo įrodymo šablonas:

1. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$ aksioma 1.2,

2. $(B \to C) \to (A \to (B \to C))$ aksioma 1.1 ($A - B \to C; B - A$),

3. $B \to C$ *duota* (*turima*) *formulė* (tai yra aksioma, prielaida ar išvedimo eigoje gauta formulė),

4. $A \to (B \to C)$ MP 3 ir 2 formulėms,

5. $(A \to B) \to (A \to C)$ MP 4 ir 1 formulėms,

6. $A \to B$ *turima formulė*,

7. $A \to C$ MP 6 ir 5 formulėms.

Grįžtame prie formulės $\neg\neg\neg\neg A \to A$ įrodymo.

Iš 4.3 aksiomų schemos gauname formules $\neg\neg\neg\neg A \to \neg\neg A, \neg\neg A \to A$. Telieka pasinaudoti implikacijos tranzityvumu (aksioma 1.2). $A - \neg\neg\neg\neg A; B - \neg\neg A; C - A$.

Formulės $\neg\neg\neg\neg A \to A$ *įrodymas*:

1. $\neg\neg\neg\neg A \to \neg\neg A$ aksioma 4.3 ($A - \neg\neg A$),

2. $\neg\neg A \to A$ aksioma 4.3,

3. $(\neg\neg\neg\neg A \to (\neg\neg A \to A)) \to ((\neg\neg\neg\neg A \to \neg\neg A) \to (\neg\neg\neg\neg A \to A))$ aksioma 1.2 $(A - \neg\neg\neg\neg A; B - \neg\neg A; C - A)$,

4. $(\neg\neg A \to A) \to (\neg\neg\neg\neg A \to (\neg\neg A \to A))$ aksioma 1.1 $(A - \neg\neg A; B - \neg\neg\neg\neg A)$,

5. $\neg\neg\neg\neg A \to (\neg\neg A \to A)$ MP 2 ir 4 formulėms,

6. $(\neg\neg\neg\neg A \to \neg\neg A) \to (\neg\neg\neg\neg A \to A)$ MP 5 ir 3 formulėms,

7. $\neg\neg\neg\neg A \to A$ MP 1 ir 6 formulėms.

Sekvencijose prielaidų sąrašas gali būti ir netuščias. Tuomet sakoma, *išvedimai iš prielaidų*. Išvedimai beveik tokie patys, kaip ir be prielaidų. Tik galima naudotis prielaidomis kaip aksiomomis t.y. *duotomis formilėmis,* kurių nereikalaujame įrodymo.

3. Sekvencijos $A \to C, A \to (C \to B) \vdash A \to B$ išvedimas:

1. $A \to C$ prielaida,

2. $A \to (C \to B)$ prielaida,

3. $(A \to (C \to B)) \to ((A \to C) \to (A \to B))$ aksioma 1.2 $(B - C; C - B)$,

4. $(A \to C) \to (A \to B)$ MP 2 ir 3 formulėms,

$A \to B$ MP 1 ir 4 formulėms.

Konjunkcijos įrodymo šablonas.

Tarkime *duotos* formulės *B, C*. Reikia įrodyti $B \wedge C$. Aksiomą, prielaidą ar išvedimo eigoje gautą formulę vadiname *duota* (*turima*) formule. Naudosime aksiomą 2.3 $(A \to B) \to ((A \to C) \to (A \to (B \wedge C)))$. Vietoje *A* reikia imti kurią nors turimą formulę. Pavyzdžiui, 4.2 $A \to \neg\neg A$.

1. B duota formulė,

2. C duota formulė,

3. $((A \to \neg\neg A) \to B) \to (((A \to \neg\neg A) \to C) \to ((A \to \neg\neg A) \to (B \wedge C)))$ aksioma 2.3,

4. $B \to ((A \to \neg\neg A) \to B)$ aksioma 1.1 $(A - B; B - A \to \neg\neg A)$,

5. $(A \to \neg\neg A) \to B$ MP 1 ir 4 formulėms,

6. $((A \to \neg\neg A) \to) \to C) \to ((A \to \neg\neg A) \to (B \wedge C))$ MP 5 ir 3 formulėms,

7. $C \to ((A \to \neg\neg A) \to C)$ aksioma 1.1 $(A - C; B - A \to \neg\neg A)$,

8. $(A \to \neg\neg A) \to C$ MP 2 ir 7 formulėms,

9. $(A \to \neg\neg A) \to (B \wedge C)$ MP 8 ir 6 formulėms,

10. $A \to \neg\neg A$ aksioma 4.2,

11. $B \wedge C$ MP 10 ir 9 formulėms.

Pratimai. Išveskite sekvencijas (atsakymai ir sprendimai knygos pabaigoje):

1. $\vdash (A \vee B) \to (B \vee A)$,

2. $\vdash A \to \neg(\neg A \wedge \neg B)$,

3. $\vdash (A \wedge B) \to (A \wedge (B \vee C))$,

4. $A \to (B \to C), \neg C, A \vdash \neg B$,

5. $(A \vee B) \to C, (C \vee D) \to E, A \vdash E$,

Įrodyta, kad Hilberto skaičiavime $\Gamma \vdash A \to B$ tada ir tiktai tada, kai $\Gamma, A \vdash B$ (*dedukcijos teorema*). Kitaip tariant, jei sukcedente yra formulė pavidalo $A \to B$, tai A galima perkelti į prielaidų sąrašą. Daugeliu atveju taikant dedukcijos teoremą gaunami trumpesni išvedimai. Vietoje vienos taisyklės (modus ponens) dabar turime tris. Gauname dar dvi *leistinas taisykles*, t.y. taisykles, be kurių galima ir apsieiti, bet su jomis kai kuriais atvejais išvedimai tampa trumpesni.

$$\frac{\Gamma, F \vdash G}{\Gamma \vdash F \to G} \qquad \frac{\Gamma \vdash F \to G}{\Gamma, F \vdash G} \qquad \frac{\Gamma \vdash F \to G \quad \Gamma \vdash F}{\Gamma \vdash G}$$

Turbūt pastebėjote, kad pirma ir trečia yra *natūraliosios dedukcijos* taisyklės. Štai kaip ir atsirado natūralioji dedukcija. Hilberto skaičiavimas, kuriame viena taisyklė ir daug aksiomų schemų buvo perdarytas į skaičiavimą, kuriame viena aksioma ir daug taisyklių. Tai *natūralioji dedukcija*, kuriai šioje knygoje skyrėme daug dėmesio. O prasidėjo Hilberto skaičiavimo transformavimas į natūraliąją dedukciją nuo *dedukcijos teoremos*.

Pavyzdys. Įrodyti $\vdash (A \to B) \to ((B \to C)\neg(A \to C))$ naudojant dedukcijos teoremą.

1. $A \to B \vdash (B \to C) \to (A \to C)$ (taikėme dedukcijos teoremą),

2. $A \to B, B \to C \vdash A \to C$ (taikėme dedukcijos teoremą),

3. $A \to B, B \to C, A \vdash C$ (taikėme dedukcijos teoremą),

Taigi, $\vdash (A \to B) \to ((B \to C)\neg(A \to C))$ įrodoma tada ir tiktai tada, kai išvedama $A \to B, B \to C, A \vdash C$.

Įrodysime $A \to B, B \to C, A \vdash C$.

4. $A \to B$ prielaida,

5. A prielaida,

6. B gauta pagal MP 5 ir 4 formulėms,

7. $B \to C$ prielaida,

8. C gauta pagal MP 6 ir 7 formulėms.

Taikant dedukcijos teoremą išvedimas gavosi trivialus. Nesinaudojant dedukcijos teorema, išvedimas formulės $(A \to B) \to ((B \to C)\neg(A \to C))$ yra žymiai sudėtingesnis.

Pratimas. Išveskite sekvencijas naudojant dedukcijos teoremą (sprendimas knygos pabaigoje):

6. $\vdash (A \to (B \to C)) \to (B \to (A \to C))$,

Sąvokų žodynas

Abėcėlė baigtinė netuščia aibė.

- žodis abėcėlėje A 1. Jei a yra abėcėlės A elementas, tai, a yra žodis abėcėlėje A.
2. Jei x yra žodis abėcėlėje A ir $a \in A$, tai xa yra taip pat žodis abėcėlėje A.

Aibė

- begalinė aktuali aibė su kuria operuojama kaip su duotu, užbaigtu, realiai egzistuojančiu objektu. *Pavyzdžiui, iracionaliųjų skaičių aibė.*

- begalinė potenciali aibės elementai gaunami tęsiant procesą neribotai (t.y. bet kurį baigtinį skaičių kartų) pagal nurodytas taisykles.
Pavyzdys:
1. *Nulis yra natūralusis skaičius.*
2. *Jei n natūralusis skaičius, tai* $n + 1$ *irgi natūralusis skaičius.*

- kontinuumo galios jei yra bijekcija tarp nagrinėjamos aibės ir realiųjų skaičių aibės.

- numeruojamoji jei ji baigtinė arba skaičioji.

- rekursyvi (išsprendžiama) jei jos charakteringąją funkciją galima suprogramuoti.

- skaičioji jei yra bijekcija tarp nagrinėjamos aibės ir natūraliųjų skaičių aibės.

- skaičioji rekursyviai jei bijekciją galima suprogramuoti.

Formulių aibė

- išsprendžiama jei galima suprogramuoti jos charakteringąją funkciją.

- įvykdoma jei galima rasti struktūrą (interpretaciją), kurioje visos aibės formulės teisingos.

- prieštaringa jei, kokia bebūtų struktūra (interpretacija), aibėje yra bent viena klaidinga formulė.

Formali aksiominė teorija

- efektyviai aksiomatizuojama jei joje išvedamų formulių Gödelio numerių aibė rekursyviai skaičioji. **Kitas apibrėžimas:** jei egzistuoja algoritmas atpažistantis teorijos aksiomas.

- išsprendžiama jei egzistuoja algoritmas, kuriuo, kokia bebūtų teorijos formulė, galima nuastatyti, ar ji išvedama teorijoje.

- kategoriška jei bet kurie du teorijos modeliai yra izomorfiniai.

- kompaktiška kokia bebūtų teorijos uždarų formulių aibė M (ji gali būti ir begalinė), jei kiekvienas jos baigtinis poaibis įvykdomas, tai įvykdoma ir aibė M.

- korektiška įrodomos formulės yra teisingos teorijoje (nebūtinai visos teisingos yra įrodomos, bet, jei įrodomos, tai teisingos).

- neprieštaringa jei teorijoje įrodoma kuri nors formulė, tai neįrodomas jos neigimas.

- pilna kiekviena teorijos formulė arba jos neigimas įrodomi teorijoje.

- pilna atžvilgiu . . . *Pavyzdžiui*, teorija **pilna atžvilgiu tapačiai teisingų formulių**. Teorijoje išvedamos visos tapačiai teisingos formulės ir tiktai jos.

Įrodymas (išvedimas)

- intuityvus jei išvedime nesinaudojama vien tik formaliu loginiu skaičiavimu.

- normalinis įrodymas, kuriame nenaudojama pjūvio taisyklė.

Logika

- grynoji loginis skaičiavimas su tuščia signatūra, t.y. formulėse nėra nei konkrečių konstantų, nei konkrečių predikatų, nei konkrečių funkcijų.

- vieninga hipotezė (požiūris), kad žmogaus mąstymui būdinga vienintelė logika.

Matematika

- finitinė konstruktyvioji matematika; įrodymams naudoja intuicionistinę (konstruktyviąją) logiką.

Uždavinių sprendimai ir atsakymai

1 skyrius. Teiginių logika

1a. $F = (p \to q) \wedge (\neg(\neg p \vee r) \to (q \wedge \neg r))$.

p	q	r	$\neg p$	$\neg r$	$\neg p \vee r$	$q \wedge \neg r$	$\neg(p \vee r)$	$\neg(\neg p \vee r) \to (q \wedge \neg r)$	$p \to q$	F
t	t	t	k	k	t	k	k	t	t	t
t	t	k	k	t	k	t	t	t	t	t
t	k	t	k	k	t	k	k	t	k	k
t	k	k	k	t	k	k	t	k	k	k
k	t	t	t	k	t	k	k	t	t	t
k	t	k	t	t	t	t	k	t	t	t
k	k	t	t	k	t	k	k	t	t	t
k	k	k	t	t	t	k	k	t	t	t

Atsakymas: *formulė įvykdoma.*

1b. $F = ((p \wedge q) \to \neg r) \vee ((q \to r) \vee (\neg p \to q))$.

p	q	r	$\neg p$	$\neg r$	$p \wedge r$	$(p \wedge q) \to \neg r$	$q \to r$	$\neg p \to q$	$((q \to r) \vee (\neg p \to q)$	F
t	t	t	k	k	t	k	t	t	t	t
t	t	k	k	t	t	t	k	t	t	t
t	k	t	k	k	k	t	t	t	t	t
t	k	k	k	t	k	t	t	t	t	t
k	t	t	t	k	k	t	t	t	t	t
k	t	k	t	t	k	t	k	t	t	t
k	k	t	t	k	k	t	t	k	t	t
k	k	k	t	t	k	t	t	k	t	t

Atsakymas: *formulė tapačiai teisinga.*

1c. $((p \to q) \to (q \to p)) \wedge (\neg p \wedge q)$.

p	q	$\neg p$	$\neg p \wedge q$	$p \to q$	$q \to p$	$(p \to q) \to (q \to p)$	F
t	t	k	k	t	t	t	k
t	k	k	k	k	t	t	k
k	t	t	t	t	k	k	k
k	k	t	k	t	t	t	k

Atsakymas: *formulė tapačiai klaidinga.*

2a. $p \wedge (q \to p) \equiv p \wedge (\neg q \vee p) \equiv \neg(\neg p \vee \neg(\neg q \vee p))$.

2b. $(p \to q) \to (q \vee p) \equiv (\neg p \vee q) \to (q \vee p) \equiv \neg(\neg p \vee q) \vee (q \vee p)$.

2c. $p \to (q \to (p \wedge q)) \equiv p \to (q \to \neg(\neg p \vee \neg q)) \equiv p \to (\neg q \vee \neg(\neg p \vee \neg q)) \equiv \neg p \vee (\neg q \vee \neg(\neg p \vee \neg q))$.

2d. $(\neg p \to \neg q) \to (q \to p) \equiv (\neg p \to \neg q) \to (\neg q \vee p) \equiv (p \vee \neg q) \to (\neg q \vee p) \equiv \neg(p \vee \neg q) \vee (\neg q \vee p)$.

3. Pasinaudosime jau aprašytomis formulėmis $\neg p \equiv p \uparrow p$ ir $p \vee q \equiv (p \uparrow q) \uparrow (p \uparrow q)$.

$p \to q \equiv \neg p \vee q \equiv (p \uparrow p) \vee q \equiv ((p \uparrow p) \uparrow q) \uparrow ((p \uparrow p) \uparrow q)$.

$p \wedge q \equiv \neg(\neg p \vee \neg q) \equiv \neg((p \uparrow p) \vee (q \uparrow q)) \equiv \neg(((p \uparrow p) \uparrow (q \uparrow q)) \uparrow ((p \uparrow p) \uparrow (q \uparrow q))) \equiv [(((p \uparrow p) \uparrow (q \uparrow q)) \uparrow ((p \uparrow p) \uparrow (q \uparrow q)))] \uparrow [(((p \uparrow p) \uparrow (q \uparrow q)) \uparrow ((p \uparrow p) \uparrow (q \uparrow q)))]$.

4. Pasinaudosime ekvivalenčiomis formulėmis $\neg p \equiv p \mid p$ ir $p \wedge q \equiv (p \mid q) \mid (p \mid q)$.
$p \vee q \equiv \neg(\neg p \wedge \neg q) \equiv \neg((p \mid p) \wedge (q \mid q)) \equiv \neg(((p \mid p) \wedge (q \mid q)) \mid ((p \mid p) \wedge (q \mid q))) \equiv [((p \mid p) \wedge (q \mid q)) \mid ((p \mid p) \wedge (q \mid q))] \mid [((p \mid p) \wedge (q \mid q)) \mid ((p \mid p) \wedge (q \mid q))]$.

$p \to q \equiv \neg p \vee q \equiv \neg(p \wedge \neg q) \equiv \neg(p \wedge (q \mid q)) \equiv \neg(((p \mid (q \mid q)) \mid (p \mid (q \mid q))) \equiv [((p \mid (q \mid q)) \mid (p \mid (q \mid q)))] \mid [((p \mid (q \mid q)) \mid (p \mid (q \mid q)))]$.

5a.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p \to (q \to r), p \vdash r, p}\oplus \quad \dfrac{\dfrac{}{q, p \vdash r, p}\oplus \quad \dfrac{\dfrac{}{q, p \vdash r, q}\oplus \quad \dfrac{}{r, q, p \vdash r}\oplus}{\mathbf{q} \to \mathbf{r}, q, p \vdash r}}{\mathbf{p} \to (\mathbf{q} \to \mathbf{r}), q, p \vdash r}}{p \to (q \to r), \mathbf{p} \to \mathbf{q}, p \vdash r}}{p \to (q \to r), p \to q \vdash \mathbf{p} \to \mathbf{r}}}{p \to (q \to r) \vdash (\mathbf{p} \to \mathbf{q}) \to (\mathbf{p} \to \mathbf{r})}}{\vdash (p \to (q \to r)) \to ((p \to q) \to (p \to r))}$$

Atsakymas: *formulė tapačiai teisinga.*

5b.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg p \to \neg q, p \vdash p}\oplus \quad \dfrac{\dfrac{}{q, p \vdash p}\oplus \quad \dfrac{\dfrac{}{q \vdash p, q}\oplus \quad \dfrac{}{\neg \mathbf{q}, q \vdash p}}{q \vdash p, \neg \mathbf{p}}}{\neg \mathbf{p} \to \neg \mathbf{q}, q \vdash p}}{\neg p \to \neg q \vdash p, \neg \mathbf{p}}}{\neg p \to \neg q, \neg \mathbf{p} \to \mathbf{q} \vdash p}}{\neg p \to \neg q \vdash (\neg \mathbf{p} \to \mathbf{q}) \to \mathbf{p}}}{\vdash (\neg p \to \neg q) \to ((\neg p \to q) \to p)}$$

Atsakymas: formulė tapačiai teisinga.

6. Formulė tapačiai klaidinga, jei išvedama sekvencija $F \vdash$ (arba sekvencija $\vdash \neg F$).

$$\dfrac{\dfrac{}{\vdash (p \wedge q) \to r} \quad \dfrac{\dfrac{}{\vdash p \wedge \neg r} \quad \dfrac{?}{r \vdash}}{(\mathbf{p} \wedge \neg \mathbf{r}) \to \mathbf{r} \vdash}}{((p \wedge q) \to r) \to ((p \wedge \neg r) \to r) \vdash}$$

Atsakymas: *formulė nėra tapačiai kladinga.*

7a. Jei važiuosiu troleibusu (v) ir troleibusas pavėluos (p), tai aš pavėluosiu į paskaitas (s). Jei pavėluosiu į paskaitas, tai neišspręsiu užduoties ($\neg n$). Jei neišspręsiu užduoties, tai neišlaikysiu testo ($\neg t$). Vadinasi, jei nevažiuosiu troleibusu, tai nepavėluosiu į paskaitą ir išlaikysiu testą.

Reikia patikrinti ar išvedama sekvencija $(v \wedge p) \to s, s \to \neg n, \neg n \to \neg t \vdash \neg v \to (\neg s \wedge t)$.

Atsakymas: *išvada klaidinga* (*sekvencija neišvedama*).

7b. *A* ir *B* nutarė pirkti sklypus. Jei sklypą pirks *A* (*a*), tai reiškia, kad jis pigus (*p*). Jei jis brangus ($\neg p$), tai perka *B* (*b*). *B* nepirko sklypo. Vadinasi, pirks *A*.

Reikia patikrinti ar išvedama sekvencija $a \to p, \neg p \to b, \neg b \vdash a$.

$$\dfrac{\dfrac{\dfrac{\dfrac{?}{p \vdash a, b}}{p \vdash a, b, \neg \mathbf{p}} \quad \dfrac{\dfrac{?}{p \vdash a, b}}{p \vdash a, b, \neg \mathbf{p}}}{\mathbf{a} \to \mathbf{p} \vdash a, b, \neg p} \quad \dfrac{\oplus}{a \to p, b \vdash a, b}}{a \to p, \neg \mathbf{p} \to \mathbf{b} \vdash a, b}}{a \to p, \neg p \to b, \neg \mathbf{b} \vdash a}$$

Atsakymas: *išvada klaidinga*.

8. Nustatysime *semantinių lentelių metodu* ar formulė $(p \vee (q \wedge r)) \to ((p \vee q) \wedge (p \vee r))$ tapačiai teisinga, t.y. ar išvedama sekvencija $\vdash (p \vee (q \wedge r)) \to ((p \vee q) \wedge (p \vee r))$.

$$\begin{array}{c}
\vdash (p \vee (q \wedge r)) \to ((p \vee q) \wedge (p \vee r)) \\
p \vee (q \wedge r) \\
\neg((p \vee q) \wedge (p \vee r))
\end{array}$$

$$\begin{array}{ccc}
\neg(p \vee q) & | & \neg(p \vee r) \\
\neg p & & \neg p \\
\neg q & & \neg r
\end{array}$$

$$\begin{array}{ccc}
p & | & q \wedge r \\
\oplus & & q \\
 & & r \\
 & & \oplus
\end{array}
\qquad\qquad
\begin{array}{ccc}
p & | & q \wedge r \\
\oplus & & q \\
 & & r \\
 & & \oplus
\end{array}$$

Atsakymas: *formulė tapačiai teisinga*.

9. Naudodamiesi *semantinių lentelių metodu*, nustatykite ar formulė
$\neg((\neg p \wedge \neg q) \vee (p \vee q))$
tapačiai klaidinga.

$$\neg((\neg p \wedge \neg q) \vee (p \vee q))$$
$$\neg(\neg p \wedge \neg q)$$
$$\neg(p \vee q)$$

$\neg\neg p$	$\neg\neg q$
$\neg p$	$\neg p$
$\neg q$	$\neg q$
$\oplus$	$\oplus$

Atsakymas: *formulė tapačiai klaidinga.*

10a. Į vakarėlį žadėjo ateiti Arvydas (*a*) arba Aurimas (*r*) (bent vienas iš jų). Greičiausiai bus Arvydas, o Elena (*e*) ne. O, jei bus Elena, tai būtinai ir Tomas (*t*). Vadinasi, vakarėlyje bus Arvydas ir Tomas.

Nustatysime ar išvedama sekvencija $a \vee r, a \wedge \neg e, e \to t \vdash a \wedge t$.

$$a \vee r, a \wedge \neg e, e \to t, \neg(a \wedge t)$$
$$a \vee r$$
$$a \wedge \neg e$$
$$e \to t$$
$$\neg(a \wedge t)$$
$$a$$
$$\neg e$$

$\neg a$				$\neg t$	
$\oplus$			$\neg e$	|	t
		a	|	r	$\oplus$
		?		?	

Atsakymas: *išvada klaidinga.*

10b. Netiesa, kad Tomas buvo Ispanijoje (*i*) ar Portugalijoje (*p*). Jei jis būtų išvažiavęs, kaip kad jam buvo siūloma, į Prancūziją (*r*), tai darbo klausimais būtų nuvažiavęs ir į Ispaniją. Vadinasi, Tomas nevažiavo į Prancūziją.

Nustatysime ar išvedama sekvencija $\neg(i \vee p), r \to i, r$.

$$\neg(i \vee p), r \to i, r$$
$$\neg(i \vee p)$$
$$r \to i$$

$$
\begin{array}{ccc}
 & r & \\
 & \neg i & \\
 & \neg p & \\
\neg r & | & i \\
\oplus & & \oplus
\end{array}
$$

Atsakymas: *išvada teisinga.*

10c. Jei Aurimas laiku būtų pranešęs apie gautą telegramą (a), tai Elena (e) su Tomu (t) būtų laiku išskridę. Žinoma, kad ent vienas iš jų pavėlavo į lėktuvą. Vadinasi Aurimas laiku nepranešė.

Nustatysime ar išvedama sekvencija $a \to (e \wedge t), \neg e \vee \neg t \vdash \neg a$.

$$
\begin{array}{ccccc}
 & & a \to (e \wedge t), \neg e \vee \neg t, a & & \\
 & & a \to (e \wedge t) & & \\
 & & \neg e \vee \neg t & & \\
 & & a & & \\
\neg a & & | & & e \wedge t \\
\oplus & & & & e \\
 & & & & t \\
 & & & \neg e \quad | & \neg t \\
 & & & \oplus & \oplus
\end{array}
$$

Atsakymas: *išvada teisinga.*

2 skyrius. Pirmos eilės logika

1a. $\exists x \exists y(\neg(x = y) \wedge F(x) \wedge F(y))$

1b. $\exists x \exists y(\neg(x = y) \wedge \forall z(F(z) \leftrightarrow (z = x \vee z = y)))$.

1c. $\forall x \forall y \forall z([\neg(x = y) \wedge \neg(x = z) \wedge \neg(y = z)] \to \neg[F(x) \wedge F(y) \wedge F(z)])$.

2a. $\forall x(x \in A \leftrightarrow (x \in B \wedge x \in C))$

2b. $\forall x(x \in A \leftrightarrow (x \in B \vee x \in C))$

2c. $\neg \exists x(x \in A)$

2d. $\forall x(x \in A \leftrightarrow x \in D)$.

2e. $\forall x(x \in A \leftrightarrow \neg(x \in B))$

3 Pastaba. Nenusiminkite, jei gausite ne tokį atsakymą. Tą patį teiginį galima parašyti skirtingomis bet ekvivalenčiomis formulėmis.

3a. $\forall x \forall y([T(x) \wedge T(y)] \to \exists z[I(z) \wedge x \in z \wedge y \in z])$.

3b. $\forall x \forall y([T(x) \wedge T(y) \wedge \neg(x = y)] \to \exists z[I(z) \wedge x \in z \wedge y \in z \wedge \forall u((I(u) \wedge x \in u \wedge y \in u) \to z = u)])$.

3c. $I(a) \wedge I(b) \wedge \forall x(T(x) \to \neg(x \in a \wedge x \in b))$.

3d. $\forall x[I(x) \to \exists y \exists z(T(y) \wedge T(z) \wedge y \in x \wedge z \in x \wedge \neg(y = z))]$.

3e. $\exists x \exists y[I(x) \wedge I(y) \wedge \neg(x = y) \wedge \exists z(T(z) \wedge z \in x \wedge z \in y)]$.

3f. $\forall x \forall y[(I(x) \wedge I(y)) \to (x = y \dot{\vee} \forall z(T(z) \to \neg(z \in x \wedge z \in y))]$.

4. $\forall x \forall y([T(x) \wedge I(y) \wedge \neg(x \in y)] \to \forall z[(I(z) \wedge x \in z) \to \exists u(T(u) \wedge u \in y \wedge u \in z)])$.

5. $\forall x \forall y([T(x) \wedge I(y) \wedge \neg(x \in y)] \to \exists u \exists v[I(u) \wedge I(v) \wedge \neg(u = v) \wedge x \in u \wedge x \in v \wedge \forall z((T(z) \wedge z \in y) \to (\neg(z \in u) \wedge \neg(z \in v)))])$

6a. $\exists P(x, x) = t$, nes, pavyzdžiui, $P(1, 1) = t$,
$\forall x P(x, x) = k$, nes ne su visais $x \in \{1, 2, 3\}$ $P(x, x)$ yra teisingas. $P(2, 2) = k$,
$\exists x \neg P(x, x) = t$, nes $P(2, 2) = k$; tuomet $\neg P(2, 2) = t$ ir $\exists x \neg P(x, x) = t$,
$\forall x \neg P(x, x) = k$, nes $\forall x \neg P(x, x)$ būtų teisingas, jei su visais $x \in \{1, 2, 3\}$ $\neg P(x, x)$ būtų teisingas, t.y. $P(x, x)$ būtų klaidingas. Bet, pavyzdžiui, $P(3, 3) = t$.

6b. $\exists x \exists y(P(x, y) \to \neg P(y, x)) = t$, nes, kai $x = 3, y = 1, P(3, 1) = k, \neg P(1, 3) = t$,
$k \to t = t$, $\forall x \forall y(P(x, y) \to \neg P(y, x)) = k$, nes ne su visomis *x,y* reikšmėmis formulė teisinga.

Formulė $\forall x \forall y(P(x, y) \to \neg P(y, x))$ klaidinga, kai $x = y = 1$.

Formulė $\forall x \exists y(P(x, y) \to \neg P(y, x))$ bus teisinga, jei kiekvienam $x \in \{1, 2, 3\}$ galim rasti tokias *y* reikšmes, kad formulė $P(x, y) \to \neg P(y, x)$ būtų teisinga. Kadangi *su visais x*, tai reikia peržiūrėti visas *x* reikšmes. Pradedam iš eilės.

$x = 1$. Ar galim rasti tokią *y* reikšmę, kad formulė būtų teisinga? Galim, $y = 3$.
$x = 2$. Imam $y = 2$. Tuomet $P(2, 2) = k, \neg P(2, 2) = t, k \to t = t$.
$x = 3$. Imam $y = 1$. Tuomet $P(3, 1) = k, \neg P(1, 3) = t, k \to t = t$.

Taigi, įrodėm, kad $\forall x \exists y(P(x, y) \to \neg P(y, x)) = t$.

Ir paskutinis atvejis formulei $\exists x \forall y(P(x, y) \to \neg P(y, x))$. Ar rasim tokią *x* reikšmę aibėje $\{1, 2, 3\}$, kad, kokį bepaimtumėm *y* iš $\{1, 2, 3\}$, formulė būtų teisinga? Peržiūrim visus variantus.

$x = 1$. Ar su visomis *y* reikšmėmis formulė teisinga? T.y. ar visos formulės $P(1, 1) \to$

$\neg P(1,1), P(1,2) \to \neg P(2,1), P(1,3) \to \neg P(3,1)$ teisingos? Ne, formulė $P(1,1) \to \neg P(1,1) = k$. Vadinasi, variantas, kai $x = 1$, netinka. Peržiūrėsim likusius galimus variantus.

$x = 2$. Ar su visomis y reikšmėmis formulė teisinga? T.y. ar visos formulės $P(2,1) \to \neg P(1,2), P(2,2) \to \neg P(2,2), P(2,3) \to \neg P(3,2)$ teisingos? Ne, formulė $P(2,1) \to \neg P(1,2) = k$. Vadinasi, variantas, kai $x = 2$, netinka. Peržiūrėsim paskutinį variantą.

$x = 3$. Ar su visomis y reikšmėmis formulė teisinga? T.y. ar visos formulės $P(3,1) \to \neg P(1,3), P(3,2) \to \neg P(2,3), P(3,3) \to \neg P(3,3)$ teisingos? Ne, formulė $P(3,3) \to \neg P(3,3) = k$.

Vadinasi, formulė $\forall x \exists y (P(x,y) \to \neg P(y,x))$ yra klaidinga.

7a. $\forall x P(x) \to \exists x Q(x) \equiv (P(A) \wedge P(B) \wedge P(C)) \to (Q(A) \vee Q(B) \vee Q(C))$.

7b. $\forall x \exists y (P(x) \wedge Q(y)) \equiv \exists y (P(A) \wedge Q(y)) \wedge \exists y (P(B) \wedge Q(y)) \wedge \exists y (P(C) \wedge Q(y)) \equiv$
$\equiv [P(A) \wedge (Q(A) \vee Q(B) \vee Q(C))] \wedge [P(B) \wedge (Q(A) \vee Q(B) \vee Q(C))] \wedge$
$[P(C) \wedge (Q(A) \vee Q(B) \vee Q(C))]$.

7c. $\forall x \exists y P(x,y) \equiv \exists y P(A,y) \wedge \exists y P(B,y) \wedge \exists y P(C,y) \equiv [P(A,A) \vee P(A,B) \vee P(A,C)] \wedge$
$[P(B,A) \vee P(B,B) \vee P(B,C)] \wedge [P(C,A) \vee P(C,B) \vee P(C,C)]$.

8. Pastaba. Yra daug struktūrų, kuriuose duotos formulės klaidingos. Todėl jūsų rasta struktūra gali skirtis nuo čia aprašytųjų.

8a. Individinių konstantų aibė susideda iš *visų plokštumos tiesių.* $P(a,b)$ teisingas tada ir tiktai tada, kai *tiesė a statmena tiesei b*.

8b. Individinių konstantų aibė susideda iš vieno elemento $D = \{A\}$. $P(A) = t, Q(A) = k$.

8c. Individinių konstantų aibė susideda iš dviejų elementų $D = \{A, B\}$. $P(A) = t, P(B) = t, Q(A) = t, Q(B) = k$.

8d, 8e. Individinių konstantų aibe yra natūraliųjų skaičių aibė. $P(a) = t$ tada ir tiktai tada, kai a lyginis skičius. $Q(a) = t$ tada ir tiktai tada, kai a nelyginis skaičius.

9a.

$$\dfrac{\dfrac{\dfrac{\dfrac{\oplus}{P(a,a) \vdash P(a,a), \exists y P(a,y), \exists x \exists y P(x,y)}}{P(a,a) \vdash \exists \mathbf{y} \mathbf{P}(\mathbf{a},\mathbf{y}), \exists x \exists y P(x,y)}}{P(a,a) \vdash \exists \mathbf{x} \exists \mathbf{y} \mathbf{P}(\mathbf{x},\mathbf{y})}}{\exists \mathbf{x} \mathbf{P}(\mathbf{x},\mathbf{x}) \vdash \exists x \exists y P(x,y)}$$

9b.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\oplus}{P(a)\vdash Q(a),P(a),\exists xP(x)}}{P(a)\vdash Q(a),\exists\mathbf{x}\mathbf{P}(\mathbf{x})}}{\vdash\mathbf{P}(\mathbf{a})\to\mathbf{Q}(\mathbf{a}),\exists xP(x)}}{\vdash\forall\mathbf{x}(\mathbf{P}(\mathbf{x})\to\mathbf{Q}(\mathbf{x})),\exists xP(x)}\quad\dfrac{\dfrac{\dfrac{\dfrac{\oplus}{Q(a),\forall xQ(x),P(a)\vdash Q(a)}}{\forall\mathbf{x}\mathbf{Q}(\mathbf{x}),P(a)\vdash Q(a)}}{\forall xQ(x)\vdash\mathbf{P}(\mathbf{a})\to\mathbf{Q}(\mathbf{a})}}{\forall xQ(x)\vdash\forall\mathbf{x}(\mathbf{P}(\mathbf{x})\to\mathbf{Q}(\mathbf{x}))}}{\exists\mathbf{x}\mathbf{P}(\mathbf{x})\to\forall\mathbf{x}\mathbf{Q}(\mathbf{x})\vdash\forall x(P(x)\to Q(x))}$$

9c.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\oplus}{P(a),Q(a),\forall x(P(x)\to\neg Q(x))\vdash P(a)}\quad\dfrac{\dfrac{\oplus}{P(a),Q(a),\forall x(P(x)\to\neg Q(x))\vdash Q(a)}}{P(a),Q(a),\neg\mathbf{Q}(\mathbf{a}),\forall x(P(x)\to\neg Q(x))\vdash}}{P(a),Q(a),\mathbf{P}(\mathbf{a})\to\neg\mathbf{Q}(\mathbf{a}),\forall x(P(x)\to\neg Q(x))\vdash}}{P(a),Q(a),\forall\mathbf{x}(\mathbf{P}(\mathbf{x})\to\neg\mathbf{Q}(\mathbf{x}))\vdash}}{P(a),Q(a)\vdash\neg\forall\mathbf{x}(\mathbf{P}(\mathbf{x})\to\neg\mathbf{Q}(\mathbf{x}))}}{\mathbf{P}(\mathbf{a})\wedge\mathbf{Q}(\mathbf{a})\vdash\neg\forall x(P(x)\to\neg Q(x))}}{\exists\mathbf{x}(\mathbf{P}(\mathbf{x})\wedge\mathbf{Q}(\mathbf{x}))\vdash\neg\forall x(P(x)\to\neg Q(x))}$$

10a. $\vdash(\exists xP(x)\vee\exists xQ(x))\leftrightarrow\exists x(P(x)\vee Q(x))$

$$\neg[(\exists xP(x)\vee\exists xQ(y))\leftrightarrow\exists x(P(x)\vee Q(x))]$$

$$\begin{array}{cc|cc}
\multicolumn{2}{c|}{\neg(\exists xP(x)\vee\exists xQ(x))} & \multicolumn{2}{c}{\exists xP(x)\vee\exists xQ(x)}\\
\multicolumn{2}{c|}{\exists x(P(x)\vee Q(x))} & \multicolumn{2}{c}{\neg\exists x(P(x)\vee Q(x))}\\
\multicolumn{2}{c|}{P(a)\vee Q(a)} & \exists xP(x) & \exists xQ(x)\\
P(a) & Q(a) & P(a) & Q(a)\\
\neg\exists xP(x) & \neg\exists xP(x) & \neg(P(a)\vee Q(a)) & \neg(P(a)\vee Q(a))\\
\neg\exists xQ(x) & \neg\exists xQ(x) & \neg P(a) & \neg P(a)\\
\neg P(a) & \neg Q(a) & \neg Q(a) & \neg Q(a)\\
\oplus & \oplus & \oplus & \oplus
\end{array}$$

10b. $\vdash(\forall xP(x)\wedge\forall xQ(x))\leftrightarrow\forall x(P(x)\wedge Q(x))$.

$$\neg[(\forall xP(x)\wedge\forall xQ(x))\leftrightarrow\forall x(P(x)\wedge Q(x))]$$

$$\begin{array}{cc|cc}
\multicolumn{2}{c|}{\neg(\forall xP(x)\wedge\forall xQ(x))} & \multicolumn{2}{c}{\forall x(P(x)\wedge Q(x))}\\
\multicolumn{2}{c|}{\forall x(P(x)\wedge Q(x))} & \multicolumn{2}{c}{\neg\forall x(P(x)\wedge Q(x))}\\
\neg\forall xP(x) & \neg\forall xQ(x) & \multicolumn{2}{c}{\neg(P(a)\wedge Q(a))}\\
\neg P(a) & \neg Q(a) & \multicolumn{2}{c}{\forall xP(x)}\\
P(a)\wedge Q(a) & P(a)\wedge Q(a) & \multicolumn{2}{c}{\forall xQ(x)}\\
P(a) & P(a) & \multicolumn{2}{c}{P(a)}\\
Q(a) & Q(a) & \multicolumn{2}{c}{Q(a)}\\
\oplus & \oplus & \neg P(a) & \neg Q(a)\\
 & & \oplus & \oplus
\end{array}$$

11a. $\forall x \exists y A(x,y) \to \forall y (B(y,y) \wedge \exists x A(y,x))$

Eliminuojame implikaciją $\neg \forall x \exists y A(x,y) \vee \forall y (B(y,y) \wedge \exists x A(y,x))$.

Yra pasirinkimas tarp veiksmų, kuriuos galima atlikti. Todėl atsakymas nėra vienareikšmis. Parašysime vieną iš galimų sprendimų variantų.

Iškeliame kvantorinius kompleksus prieš neigimą $\exists x \forall y \neg A(x,y) \vee \forall y (B(y,y) \wedge \exists x A(y,x))$.

Pervardijame kintamuosius $\exists x \forall y \neg A(x,y) \vee \forall z (B(z,z) \wedge \exists u A(z,u))$.

$\exists x \forall y \neg A(x,y) \vee \forall z (B(z,z) \wedge \exists u A(z,u)) \equiv \exists x \forall y \neg A(x,y) \vee \forall z \exists u (B(z,z) \wedge A(z,u)) \equiv$

$\exists x (\forall y \neg A(x,y) \vee \forall z \exists u (B(z,z) \wedge A(z,u))) \equiv \exists x \forall y (\neg A(x,y) \vee \forall z \exists u (B(z,z) \wedge A(z,u))) \equiv$

$\exists x \forall y \forall z (\neg A(x,y) \vee \exists u (B(z,z) \wedge A(z,u))) \equiv \exists x \forall y \forall z \exists u (\neg A(x,y) \vee (B(z,z) \wedge A(z,u)))$.

11b. $\forall x (P(x,x) \wedge \exists y \neg P(x,y)) \vee \forall x (Q(x,x) \wedge \exists z (P(z,x) \vee \forall u \neg P(x,u)))$.

Pervardijame kintamuosius $\forall x (P(x,x) \wedge \exists y \neg P(x,y)) \vee \forall w (Q(w,w) \wedge$

$\exists z (P(z,w) \vee \forall u \neg P(w,u)))$.

Iškeliame kvantorinius kompleksus

$\forall x (P(x,x) \wedge \exists y \neg P(x,y)) \vee \forall w (Q(w,w) \wedge \exists z (P(z,w) \vee \forall u \neg P(w,u))) \equiv$

$\forall x \exists y (P(x,x) \wedge \neg P(x,y)) \vee \forall w (Q(w,w) \wedge \exists z \forall u (P(z,w) \vee \neg P(w,u))) \equiv$

$\forall x \exists y (P(x,x) \wedge \neg P(x,y)) \vee \forall w \exists z \forall u (Q(w,w) \wedge (P(z,w) \vee \neg P(w,u))) \equiv$

$\forall x \exists y \forall w \exists z \forall u [(P(x,x) \wedge \neg P(x,y)) \vee (Q(w,w) \wedge (P(z,w) \vee \neg P(w,u)))]$.

11c. $\exists x \forall y P(x,y) \to \forall x \exists y P(y,x)) \equiv \neg \exists x \forall y P(x,y) \vee \forall x \exists y P(y,x) \equiv$

$\forall x \exists y \neg P(x,y) \vee \forall x \exists y P(y,x) \equiv \forall x \exists y \neg P(x,y) \vee \forall z \exists u P(u,z) \equiv$

$\forall x \exists y \forall z \exists u (\neg P(x,y) \vee P(u,z))$.

12. Naudosimės ekvivalenčiomis formulėmis

$$\forall x A(x) \wedge \forall x B(x) \equiv \forall x (A(x) \wedge B(x)),$$
$$\exists x A(x) \vee \exists x B(x) \equiv \exists x (A(x) \vee B(x)).$$

12a. $\exists x \forall y P(x,y) \vee \exists x \forall y Q(x,y) \equiv \exists x(\forall y P(x,y) \vee \forall y Q(x,y)) \equiv$ pervardijame $\equiv$

$\exists x(\forall y P(x,y) \vee \forall z Q(x,z)) \equiv \exists x \forall y \forall z(P(x,y) \vee Q(x,z))$.

12b. $\forall x \exists y \forall z P(x,y,z) \wedge \forall u \exists v \forall w Q(u,v,w) \equiv$ pervardijame $\equiv$

$\forall x \exists y \forall z P(x,y,z) \wedge \forall x \exists v \forall w Q(x,v,w) \equiv \forall x(\exists y \forall z P(x,y,z) \wedge \exists v \forall w Q(x,v,w)) \equiv$

$\forall x \exists y \exists v(\forall z P(x,y,z) \wedge \forall w Q(x,v,w)) \equiv$ pervardijame $\equiv$

$\forall x \exists y \exists v(\forall z P(x,y,z) \wedge \forall z Q(x,v,z)) \equiv \forall x \exists y \exists v \forall z(P(x,y,z) \wedge Q(x,v,z))$.

12c. $\forall x \exists y P(x,y) \vee (\forall u \exists z Q(u,z) \wedge \exists v \forall w R(v,w)) \equiv$

$\forall x \exists y P(x,y) \vee \exists v(\forall u \exists z Q(u,z) \wedge \forall w R(v,w)) \equiv$ pervardijame $\equiv$

$\forall x \exists y P(x,y) \vee \exists v(\forall u \exists z Q(u,z) \wedge \forall u R(v,u)) \equiv$

$\forall x \exists y P(x,y) \vee \exists v \forall u(\exists z Q(u,z) \wedge R(v,u)) \equiv \forall x \exists y P(x,y) \vee \exists v \forall u \exists z(Q(u,z) \wedge R(v,u)) \equiv$

$\forall x(\exists y P(x,y) \vee \exists v \forall u \exists z(Q(u,z) \wedge R(v,u))) \equiv$ pervardijame $\equiv$

$\forall x(\exists y P(x,y) \vee \exists y \forall u \exists z(Q(u,z) \wedge R(y,u))) \equiv \forall x \exists y(P(x,y) \vee \forall u \exists z(Q(u,z) \wedge R(y,u))) \equiv$

$\forall x \exists y \forall u \exists z(P(x,y) \vee (Q(u,z) \wedge R(y,u)))$.

13a. $\exists x(P(x) \wedge Q(x)) \wedge \exists x(\neg P(x) \wedge \neg Q(x))$.

13b. $\exists x(P(x) \wedge Q(x)) \wedge \exists x(\neg P(x) \wedge \neg Q(x)) \wedge \exists x(\neg P(x) \wedge Q(x))$.

14a. $x + y$.

Reikia rasti tokias dvi primityviai rekursyvias funkcijas $g(x), h(x,y,z)$, kad

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

Kadangi $x + 0 = x$, tai $g(x) = pr_1^1(x)$.

Ieškant $h(x,y,z)$, reikia išreikšti $x + (y + 1)$ per ankstesnę (vienetu mažesnę) funkciją (ją ir žymime raide z). $x + (y + 1) = (x + y) + 1 = s(z)$.

Atsakymas: $g(x) = pr_1^1(x), h(x, y, z) = s(z)$.

14b. $x \cdot y$.

Reikia rasti tokias dvi primityviai rekursyvias funkcijas $g(x), h(x, y, z)$, kad

$x \cdot 0 = g(x)$, $x \cdot (y + 1) = h(x, y, x \cdot y)$.
Kadangi $x \cdot 0 = 0$, tai $g(x) = 0$.

Ieškant $h(x, y, z)$, reikia išreikšti $x \cdot (y + 1)$ per ankstesnę (y vienetu mažesnis) funkciją (ją ir žymime raide z). $x \cdot (y + 1) = x \cdot y + x = z + x$. 14a užduotyje jau įrodėm, kad sudėtis primityviai rekursyvi funkcija, todėl ieškoma funkcija ir yra lygi $h(x, y, z)$.

Atsakymas: $g(x) = 0, h(x, y, z) = z + x$.

14c. $n!$, kai $0! = 1$

Kadangi funkcija vieno argumento, tai reikia rasti tokią konstantą a ir primityviai rekursyvią funkciją $h(y, z)$, kad

$0! = a$,
$(y + 1)! = h(y, y!)$.

Taigi, $a = 1 = s(0)$, $(y + 1)! = y! \cdot (y + 1) = z \cdot (y + 1)$.

Atsakymas: $a = s(0), h(y, z) = z \cdot (y + 1)$.

14d. $x \dot{-} 1$.

$0 \dot{-} 1 = 0$. $a = 0$.
$(y + 1) \dot{-} 1 = y$.

Atsakymas: $a = 0, h(y, z) = y$.

14e. $x \dot{-} u$.

$x \dot{-} 0 = x = pr_1^1(x)$.
$x \dot{-} (y + 1) = (x \dot{-} y) \dot{-} 1 = z \dot{-} 1$.

Atsakymas: $g(x) = pr_1^1(x), h(x, y, z) = z \dot{-} 1$.

14f. $sg(x)$.

$sg(0) = 0$. Taigi, konstanta (žr. užduotį 14c) a lygi nuliui.
$sg(y + 1) = 1 = s(0)$.

Atsakymas: $a = 0, h(y, z) = s(0)$.

14g. $max(x, y)$.
Atsakymas: $max(x, y) = (x \dot{-} y) + y$.

14h. $min(x, y)$

Atsakymas: $min(x, y) = x \dot{-} (x \dot{-} y)$.

14i. $|x - y|$.

Atsakymas: $|x - y| = (x \dot{-} y) + (y \dot{-} x)$.

14j. Funkcija $u(x_1, \ldots, x_n, x_{n+1})$ primityviai rekursyvi; įrodykite, kad ir $f(x_1, \ldots, x_n, x_{n+1})$ yra primityviai rekursyvi:

$$f(x_1, \ldots, x_n, x_{n+1}) = \sum_{i=0}^{x_{n+1}} u(x_1, \ldots, x_n, i).$$

Sprendimas:

$f(x_1, \ldots, x_n, 0) = u(x_1, \ldots, x_n, 0)$,

$f(x_1, \ldots, x_n, y+1) = \sum_{i=0}^{y} u(x_1, \ldots, x_n, i) + u(x_1, \ldots, x_n, y+1) = f(x_1, \ldots, x_n, y) + u(x_1, \ldots, x_n, y+1) = z + u(x_1, \ldots, x_n, y+1)$.

Atsakymas: $g(x_1, \ldots, x_n) = u(x_1, \ldots, x_n, 0)$, $h(x_1, \ldots, x_{n-1}, y, z) = z + u(x_1, \ldots, x_n, y+1)$.

15a. $g(x) = x$, $h(x, y, z) = z$.

Skaičiuosime $f(x, y)$ su reikšmėmis $y = 0, 1, 2, 3, \ldots$ kol pastebėsime dėsningumą.

$f(x, 0) = g(x) = x$,

$f(x, 1) = h(x, 0, f(x, 0)) = f(x, 0) = x$,

$f(x, 2) = h(x, 1, f(x, 1)) = f(x, 1) = x$,

$f(x, 3) = h(x, 2, f(x, 2)) = f(x, 2) = x$,

Turbūt užtenka skaičiuoti. Tikiuosi pastebėjote dėsningumą.

Atsakymas: $f(x, y) = x$.

15b. $g(x) = x$, $h(x, y, z) = x + z$.

$f(x, 0) = g(x) = x$,

$f(x,1) = h(x,0,f(x,0)) = x + z = x + f(x,0) = x + x = 2x,$

$f(x,2) = h(x,1,f(x,1)) = x + z = x + f(x,1) = x + 2x = 3x,$

$f(x,3) = h(x,2,f(x,2)) = x + z = x + f(x,2) = x + 3x = 4x.$

Atsakymas: $f(x,y) = x(y+1)$.

15c. $g(x) = x$, $h(x,y,z) = x + y$.

$f(x,0) = g(x) = x,$

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

$f(x,2) = h(x,1,f(x,1)) = x + y = x + 1,$

$f(x,3) = h(x,2,f(x,2)) = x + y = x + 2,$

$f(x,4) = h(x,3,f(x,3)) = x + y = x + 3.$

Atsakymas: $f(x,y) = x + (y \dot{-} 1)$.

16a. Priminsime teoriją: $f^{-1}(x) = \mu_y(f(y) = x)$ (toks mažiausias *y*, su kuriuo $f(y) = x$).

Atsakymas: $x^{-1} = \mu_y(y = x)$.

16b. Atsakymas: $(x \dot{-} 1)^{-1} = \mu_y(y \dot{-} 1 = x)$.

16c. $(2^x)^{-1} = \mu_y(2^y = x)$.

17a. Atsakymas: $0 : stop$, $1 : (:2;2;3)$, $2 : (:2;1;4)$, $3 : (\times 3;0)$, $4 : (\times 5;0)$.

17b. Atsakymas: $0 : stop$, $1 : (:2;2;3)$, $2 : (\times 3;1)$, $3 : (\times 2;0)$.

18. Atsakymas: $0 : stop$, $1 : (:3;2;0)$, $2 : (:2;3;4)$, $3 : (\times 5;2)$, $4 : (:5;5;1)$, $5 : (\times 2;6)$, $6 : (\times 2;4)$.

3 Skyrius. Formalios aksiominės teorijos

1a.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P(a) \vdash P(a), \exists x P(x), P(b)}\oplus}{P(a) \vdash \exists \mathbf{x}\mathbf{P}(\mathbf{x}), P(b)} \quad \dfrac{\dfrac{\dfrac{}{P(a), P(b) \vdash \exists x \neg P(x), P(b)}\oplus}{P(a) \vdash \neg \mathbf{P}(\mathbf{b}), \exists x \neg P(x), P(b)}}{P(a) \vdash \exists \mathbf{x} \neg \mathbf{P}(\mathbf{x}), P(b)}}{P(a) \vdash \exists \mathbf{x}\mathbf{P}(\mathbf{x}) \wedge \exists \mathbf{x} \neg \mathbf{P}(\mathbf{x}), P(b)}}{P(a) \vdash \exists \mathbf{X}(\exists \mathbf{x}\mathbf{X}(\mathbf{x}) \wedge \exists \mathbf{x} \neg \mathbf{X}(\mathbf{x})), P(b)}}{P(a) \vdash \exists X(\exists x X(x) \wedge \exists x \neg X(x)), \forall \mathbf{x}\mathbf{P}(\mathbf{x})}}{P(a), \neg \forall \mathbf{x}\mathbf{P}(\mathbf{x}) \vdash \exists X(\exists x X(x) \wedge \exists x \neg X(x))}}{\exists \mathbf{x}\mathbf{P}(\mathbf{x}), \neg \forall x P(x) \vdash \exists X(\exists x X(x) \wedge \exists x \neg X(x))}}{\exists \mathbf{x}\mathbf{P}(\mathbf{x}) \wedge \neg \forall \mathbf{x}\mathbf{P}(\mathbf{x}) \vdash \exists X(\exists x X(x) \wedge \exists x \neg X(x))}}{\vdash (\exists x P(x) \wedge \neg \forall x P(x)) \rightarrow \exists X(\exists x X(x) \wedge \exists x \neg X(x))}$$

1b.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P(a), \forall x P(x), \forall X \forall x X(x) \vdash P(a)}\oplus}{\forall \mathbf{x}\mathbf{P}(\mathbf{x}), \forall X \forall x X(x) \vdash P(a)}}{\forall \mathbf{X} \forall \mathbf{x}\mathbf{X}(\mathbf{x}) \vdash P(a)}}{\neg \mathbf{P}(\mathbf{a}), \forall X \forall x X(x) \vdash}}{\exists \mathbf{x} \neg \mathbf{P}(\mathbf{x}), \forall X \forall x X(x) \vdash}}{\exists \mathbf{X} \exists \mathbf{x} \neg \mathbf{X}(\mathbf{x}), \forall X \forall x X(x) \vdash}$$

1c.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a = b, \forall x (b = x), P(b) \vdash P(b)}\oplus}{\mathbf{a} = \mathbf{b}, \forall x (a = x), P(a) \vdash P(b)}}{\forall \mathbf{x}(\mathbf{a} = \mathbf{x}), P(a) \vdash P(b)}}{\forall x (a = x), P(a) \vdash \forall \mathbf{x}\mathbf{P}(\mathbf{x})}}{\forall x (a = x) \vdash \mathbf{P}(\mathbf{a}) \rightarrow \forall \mathbf{x}\mathbf{P}(\mathbf{x})}}{\forall x (a = x) \vdash \forall \mathbf{X}(\mathbf{X}(\mathbf{a}) \rightarrow \forall \mathbf{x}\mathbf{X}(\mathbf{x}))}}{\vdash \forall x (a = x) \rightarrow \forall X (X(a) \rightarrow \forall x X(x))}$$

4 Skyrius Loginis programavimas

1a.

p	q	$p \to q$
t	t	t
t	k	k
k	t	t
k	k	t

Formulė klaidinga tik su viena interpretacija: $p = t, q = k$. Todėl ir NKF bus tik iš vieno disjunkto: $\neg p \vee q$.

Atsakymas: $\neg p \vee q$.

1b.

p	q	r	$q \wedge r$	$p \to (q \wedge r)$
t	t	t	t	t
t	t	k	k	k
t	k	t	k	k
t	k	k	k	k
k	t	t	t	t
k	t	k	k	t
k	k	t	k	t
k	k	k	k	t

Yra trys interpretacijos, su kuriomis formulė klaidinga. NKF formulėje bus trys disjunktai.

Atsakymas: $(\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r)$.

2a. $p \wedge (q \to (\neg p \vee r)) \equiv$ (eliminuojame implikaciją) $p \wedge (\neg q \vee (\neg p \vee r)) \equiv p \wedge (\neg q \vee \neg p \vee r)$.

Atsakymas: $p \wedge (\neg q \vee \neg p \vee r)$.

2b. $\neg(p \to (q \vee r)) \wedge ((q \to r) \to p) \equiv$ (eliminuojame implikaciją) $\neg(\neg p \vee (q \vee r)) \wedge (\neg(\neg q \vee r) \vee p) \equiv$ (įkeliame neigimą į skliaustus) $p \wedge \neg q \wedge \neg r \wedge ((q \wedge \neg r) \vee p) \equiv$ (taikome distributyvumo dėsnį) $p \wedge \neg q \wedge \neg r \wedge (q \vee p) \wedge (\neg r \vee p)$.

Atsakymas: $p \wedge \neg q \wedge \neg r \wedge (q \vee p) \wedge (\neg r \vee p)$.

2c. $((p \wedge q) \vee r) \wedge (p \to (\neg q \to (p \vee r))) \equiv$ (eliminuojame implikaciją) $((p \wedge q) \vee r) \wedge (\neg p \vee q \vee p \vee r) \equiv$ (taikome distributyvumo dėsnį) $(p \vee r) \wedge (q \vee r) \wedge (\neg p \vee q \vee p \vee r) \equiv$ (kadangi $\neg p \vee p$ lygi konstantai *t*, gauname) $(p \vee r) \wedge (q \vee r) \wedge (t \vee q \vee r) \equiv$ (kadangi $t \vee F$ (čia *F* bet kuri formulė) lygi *t*) $(p \vee r) \wedge (q \vee r) \wedge t \equiv$ (kadangi $t \wedge F$ ekvivalenti *F*) $(p \vee r) \wedge (q \vee r)$.

Pastaba. Jei turime formulę, kurioje yra loginė konstanta, tai ją eliminuojame naudojantis teisingumo lentele. Pavyzdžiui, $F \to k \equiv \neg F$, nes , kai $F = t$, gauname *k*, kai $F = k$, gauname *t*.

Atsakymas: $(p \vee r) \wedge (q \vee r)$.

3a. $r \to p, r \to (p \to q) \vdash r \to q$.

$S = \{\neg r \vee p, \neg r \vee \neg p \vee q, r, \neg q\}$.

$$\frac{\neg r \vee \neg p \vee q \quad r}{\neg p \vee q} \qquad \frac{\neg p \vee q \quad \neg q}{\neg p} \qquad \frac{\neg p \quad \neg r \vee p}{\neg r} \qquad \frac{\neg r \quad r}{\square}$$

3b. $p \wedge (q \vee (p \to q)) \vdash q$.

$S = \{p, q \vee \neg p, \neg q\}$.

$$\frac{p \quad q \vee \neg p}{q} \qquad \frac{q \quad \neg q}{\square}$$

3c $(p \wedge r) \vee (q \wedge r) \vdash (p \vee q) \wedge r$.

$(p \wedge r) \vee (q \wedge r) \equiv (p \vee (q \wedge r)) \wedge (r \vee (q \wedge r)) \equiv (p \vee q) \wedge (p \vee r) \wedge (r \vee q) \wedge (r \vee r) \equiv (p \vee q) \wedge (p \vee r) \wedge (r \vee q) \wedge r$.

$\neg((p \vee q) \wedge r) \equiv \neg(p \vee q) \vee \neg r \equiv (\neg p \wedge \neg q) \vee \neg r \equiv (\neg p \vee \neg r) \wedge (\neg q \vee \neg r)$.

$S = \{p \vee q, p \vee r, r \vee q, r, \neg p \vee \neg r, \neg q \vee \neg r\}$

$$\frac{p \vee q \quad \neg q \vee \neg r}{p \vee \neg r} \qquad \frac{p \vee \neg r \quad r}{p} \qquad \frac{p \quad \neg p \vee \neg r}{\neg r} \qquad \frac{\neg r \quad r}{\square}$$

4. *a, d* - Horno disjunktai, *b, c* - ne, nes *b* formulėje yra $\neg q$ įeitis, o *c* formulėje - $\neg r$ įeitis.

5a. Nustatysime ar aibė $\{\exists x Q(x), \forall x(Q(x) \to (P(x) \vee R(x))), \neg\exists x(\neg P(x) \to R(x))\}$ prieštaringa.

Jei ji prieštaringa, tai sekvencija $\exists x Q(x), \forall x(Q(x) \to (P(x) \vee R(x))) \vdash \exists x(\neg P(x) \to R(x))$ išvedama. O tai reiškia, kad *išvada* $\exists x(\neg P(x) \to R(x))$ *yra teisinga* su prielaidomis $\exists x Q(x), \forall x(Q(x) \to (P(x) \vee R(x)))$.

Transformuojame į normaliąją priešdėlinę formą:

$\{\exists x Q(x), \forall x(Q(x) \to (P(x) \vee R(x))), \forall x\neg(\neg P(x) \to R(x))\}$.

Skulemizuojame:

$\{Q(a), \forall x(Q(x) \to (P(x) \vee R(x))), \forall x\neg(\neg P(x) \to R(x))\}$.

Transformuojame į NKF:

$\{Q(a), \forall x(\neg Q(x) \vee P(x) \vee R(x)), \forall x(\neg P(x) \wedge \neg R(x))\}.$

Bendrumo kvantorių eliminavimas:

$\{Q(a), \forall x(\neg Q(x) \vee P(x) \vee R(x)), \forall x \neg P(x) \wedge \forall x \neg R(x))\},$
$\{Q(a), \forall x(\neg Q(x) \vee P(x) \vee R(x)), \forall y \neg P(y) \wedge \forall z \neg R(z))\},$
$\{Q(a), \neg Q(x) \vee P(x) \vee R(x), \neg P(y) \wedge \neg R(z)\}$

Disjunktų aibė: $S = \{Q(a), \neg Q(x) \vee P(x) \vee R(x), \neg P(y), \neg R(z)\}$

Naudojantis rezoliucijų taisykle randame tuščio disjunkto išvedimą:

$$\frac{Q(a) \quad \neg Q(x) \vee P(x) \vee R(x)}{P(a) \vee R(a)} \ \sigma = (a/x)$$

$$\frac{P(a) \vee R(a) \quad \neg P(y)}{R(a)} \ \sigma = (a/y)$$

$$\frac{R(a) \quad \neg R(z)}{\square} \ \sigma = (a/z)$$

Atsakymas: *išvada teisinga.*

5b. Nustatysime ar aibė $\{\exists x(P(x) \rightarrow Q(x)), \forall x(Q(x) \rightarrow R(x)), \neg(\forall x P(x) \rightarrow \exists x R(x))\}$ prieštaringa.

Transformuojame į normaliąją priešdėlinę formą:

$\{\exists x(P(x) \rightarrow Q(x)), \forall x(Q(x) \rightarrow R(x)), \neg(\neg \forall x P(x) \vee \exists x R(x))\},$
$\{\exists x(P(x) \rightarrow Q(x)), \forall x(Q(x) \rightarrow R(x)), \forall x P(x) \wedge \forall x \neg R(x)\},$
$\{\exists x(P(x) \rightarrow Q(x)), \forall x(Q(x) \rightarrow R(x)), \forall y P(y) \wedge \forall z \neg R(z)\},$
$\{\exists x(P(x) \rightarrow Q(x)), \forall x(Q(x) \rightarrow R(x)), \forall y \forall z(P(y) \wedge \neg R(z))\}.$

Skulemizuojame:

$\{P(a) \rightarrow Q(a), \forall x(Q(x) \rightarrow R(x)), \forall y \forall z(P(y) \wedge \neg R(z))\}.$

Transformuojame į NKF:

$\{\neg P(a) \vee Q(a), \forall x(\neg Q(x) \vee R(x)), \forall y \forall z(P(y) \wedge \neg R(z))\}.$

Bendrumo kvantorių eliminavimas:

$\{\neg P(a) \vee Q(a), \neg Q(x) \vee R(x), P(y) \wedge \neg R(z)\}.$

Disjunktų aibė:

$S = \{\neg P(a) \vee Q(a), \neg Q(x) \vee R(x), P(y), \neg R(z)\}$.

Naudojantis rezoliucijų taisykle randame tuščio disjunkto išvedimą:

$$\frac{\neg P(a) \vee Q(a) \quad P(y)}{Q(a)} \ \sigma = (a/y)$$

$$\frac{Q(a) \quad \neg Q(x) \vee R(x)}{R(a)} \ \sigma = (a/x)$$

$$\frac{R(a) \ \ \neg R(z)}{\square} \ \sigma = (a/z)$$

Atsakymas: *išvada teisinga.*

5c. Nustatysime ar aibė $\{\forall x \neg M(x), \neg\exists x \forall y(\neg M(x) \wedge \neg P(x,y)), \neg\forall x \exists y P(x,y)\}$ prieštaringa.

Transformuojame į normaliąją priešdėlinę formą:

$\{\forall x \neg M(x), \forall x \exists y \neg(\neg M(x) \wedge \neg P(x,y)), \exists x \forall y \neg P(x,y)\}$.

Skulemizuojame:

$\{\forall x \neg M(x), \forall x \neg(\neg M(x) \wedge \neg P(x, f(x))), \forall y \neg P(a,y)\}$.

Transformuojame į NKF:

$\{\forall x \neg M(x), \forall x(M(x) \vee P(x, f(x))), \forall y \neg P(a,y)\}$.

Bendrumo kvantorių eliminavimas. Disjunktų aibė:

$\{\neg M(x), M(z) \vee P(z, f(z)), \neg P(a,y)\}$.

Naudojantis rezoliucijų taisykle ieškome tuščio disjunkto išvedimo:

$$\neg M(x) \quad M(z) \vee P(z, f(z))$$

Kokį pasirinkti keitinį? Galima $\sigma = (t/x, t/z)$ (čia *t* bet kuris termas iš nagrinėjamos sekvencijos Herbrand'o universumo). Galima ir $\sigma = (x/x, x/z)$, t.y. kintamojo *x* nekeičiame. Tokiu atveju keitinį galima rašyti ir taip $\sigma = (x/z)$, nes jis yra *bendriausias unifikatorius*.

Unifikatorius σ vadinamas **bendriausiuoju** reiškiniams R, R', jei, bet kuriam reiškinių R, R' unifikatoriui α, egzistuoja toks β, kad α yra *lygus* β *ir* σ *kompozicijai* $\alpha = \sigma(\beta)$.

Keičiame *z* į *x*. Vėliau, jei reiks, galėsime tuščio disjunkto išvedime kintamąjį *x* pakeisti bet kuriuo kitu termu iš Herbrand'o universumo. O vietoje *t* negalėtumėme. Sunku nusakyti, kai uždaviniai sudėtingi, kurį keitinį pasirinkti, kokio prisireiks ateityje, todėl imamas universalus, *bendriausias keitinys.* Mūsų atveju, tai $\sigma = (x/z)$.

$$\frac{M(x) \quad M(z) \vee P(z, f(z))}{P(z, f(z))} \ \sigma = (x/z)$$

$$\frac{P(z, f(z)) \quad \neg P(a, y)}{\square} \ \sigma = (a/z, f(a)/y)$$

Atsakymas; *išvada teisinga.*

6. Formulė tapačiai teisinga tada ir tiktai tada, jei išvedama sekvencija $\vdash \exists x \forall y \forall z \exists v((P(x) \vee \neg P(y)) \wedge (P(z) \vee \neg P(v)))$. Sekvencija išvedama tada ir tiktai tada, kai aibė $\{\neg(\exists x \forall y \forall z \exists v((P(x) \vee \neg P(y)) \wedge (P(z) \vee \neg P(v))))\}$ prieštaringa.

Transformuojame į normaliąją priešdėlinę formą:

$\{\forall x \exists y \exists z \forall v \neg((P(x) \vee \neg P(y)) \wedge (P(z) \vee \neg P(v)))\}$

Skulemizuojame: $\{\forall x \forall v \neg((P(x) \vee \neg P(f(x))) \wedge (P(g(x)) \vee \neg P(v)))\}$.

Transformuojame į NKF:

$\{\forall x \forall v((\neg P(x) \wedge P(f(x))) \vee (\neg P(g(x)) \wedge P(v)))\}$,
$\{\forall x \forall v((\neg P(x) \vee \neg P(g(x))) \wedge (\neg P(x) \vee P(v)) \wedge (P(f(x)) \vee \neg P(g(x))) \wedge (P(f(x)) \vee P(v)))\}$.

Bendrumo kvantorių eliminavimas. Disjunktų aibė:

$\{\neg P(x) \vee \neg P(g(x)), \neg P(y) \vee P(v), P(f(z)) \vee \neg P(g(z)), P(f(u)) \vee P(m)\}$.

Naudojantis rezoliucijų taisykle randame tuščio disjunkto išvedimą:

$$\frac{\neg P(y) \vee P(v) \quad P(f(u)) \vee P(m)}{P(v)} \ \sigma = (v/m, f(u)/y)$$

$$\frac{P(v) \quad \neg P(x) \vee \neg P(g(x))}{\neg P(x)} \ \sigma = (g(x)/v)$$

$$\frac{\neg P(x) \quad P(v)}{\square} \ \sigma = (x/v)$$

Atsakymas: *formulė tapačiai teisinga.*

7. Formulė tapačiai klaidinga tada ir tiktai tada, kai išvedama sekvencija $\forall x(P(x) \to P(f(x))) \wedge P(a) \wedge \neg P(f(f(f(a)))) \vdash$. Sekvencija išvedama tada ir tiktai tada, kai aibė $\{\forall x(P(x) \to P(f(x))) \wedge P(a) \wedge \neg P(f(f(f(a))))\}$ prieštaringa.

Pakeiskime konjunkciją tarp formulių kableliu:

$\{\forall x(P(x) \to P(f(x))), P(a), \neg P(f(f(f(a))))\}$.

Aibės formulės normaliosios priešdėlinės formos ir skulemizuotos.
Transformuojame į NKF:

$\{\forall x(\neg P(x) \vee P(f(x))), P(a), \neg P(f(f(f(a))))\}$.

Eliminuojame bendrumo kvantorius ir gauname disjunktų aibę:

$\{\neg P(x) \vee P(f(x)), P(a), \neg P(f(f(f(a))))\}$.

Naudojantis rezoliucijų taisykle randame tuščio disjunkto išvedimą:

$$\frac{\neg P(x) \vee P(f(x)) \quad P(a)}{P(f(a))}\ \sigma = (a/x)$$

$$\frac{P(f(a)) \quad \neg P(x) \vee P(f(x))}{P(f(f(a)))}\ \sigma = (f(a)/x)$$

$$\frac{P(f(f(a))) \quad \neg P(x) \vee P(f(x))}{P(f(f(f(a))))}\ \sigma = (f(f(a))/x)$$

$$\frac{P(f(f(f(a)))) \quad \neg P(f(f(f(a))))}{\square}\ \sigma = \emptyset$$

Atsakymas: *formulė tapačiai klaidinga.*

8a. Visų pirma įrodysime, kad sekvencija $Sk = \neg F \wedge \neg G, F \vee G, G \vdash$ išvedama. Ją panaudosime sekvencijos $\neg F \wedge \neg G \vdash \neg(F \vee G)$ išvedime.

$$\dfrac{\dfrac{\dfrac{}{\neg F \wedge \neg G, F \vee G, G \vdash \neg F \wedge \neg G}\oplus}{\neg F \wedge \neg G, F \vee G, G \vdash \neg G} \quad \dfrac{\dfrac{}{\neg F \wedge \neg G, F \vee G, G \vdash G}\oplus}{}}{\neg F \wedge \neg G, F \vee G, G \vdash}$$

$$\dfrac{\dfrac{}{\neg F \wedge \neg G, F \vee G \vdash F \vee G}\oplus \quad \dfrac{\dfrac{\dfrac{}{\neg F \wedge \neg G, F \vee G, F \vdash \neg F \wedge \neg G}\oplus}{\neg F \wedge \neg G, F \vee G, F \vdash \neg F} \quad \dfrac{\dfrac{}{\neg F \wedge \neg G, F \vee G, F \vdash F}\oplus}{}}{\neg F \wedge \neg G, F \vee G, F \vdash} \quad \dfrac{}{Sk}}{\neg F \wedge \neg G, F \vee G \vdash}}{\neg F \wedge \neg G \vdash \neg(F \vee G)}$$

8b. Iš pradžių rasime sevencijos $Sk = F \to (G \to H), F \wedge G \vdash G \to H$ išvedimą. Ją panaudosime sekvencijos $F \to (G \to H) \vdash (F \wedge G) \to H$

$$\dfrac{\dfrac{}{F \to (G \to H), F \wedge G \vdash F \to (G \to H)}\oplus \quad \dfrac{\dfrac{}{F \to (G \to H), F \wedge G \vdash F \wedge G}\oplus}{F \to (G \to H), F \wedge G \vdash F}}{F \to (G \to H), F \wedge G \vdash G \to H}$$

$$\dfrac{\dfrac{}{Sk} \quad \dfrac{\dfrac{}{F \to (G \to H), F \wedge G \vdash F \wedge G}\oplus}{F \to (G \to H), F \wedge G \vdash G}}{F \to (G \to H), F \wedge G \vdash H}}{F \to (G \to H) \vdash (F \wedge G) \to H}$$

8c.

$$\dfrac{\dfrac{}{F \to G, \neg G, F \vdash \neg G}\,\oplus \qquad \dfrac{\dfrac{}{F \to G, \neg G, F \vdash F \to G}\,\oplus \quad \dfrac{}{F \to G, \neg G, F \vdash F}\,\oplus}{F \to G, \neg G, F \vdash G}}{\dfrac{F \to G, \neg G, F \vdash}{\dfrac{F \to G, \neg G \vdash \neg F}{F \to G \vdash \neg G \to \neg F}}}$$

8d.

Iš pradžių įrodome sekvenciją $Sk = \Gamma \vdash \neg\forall x B(x)$ Čia Γ yra formulių seka. $\Gamma = A(a) \to \forall x B(x), A(a), \neg B(b)$

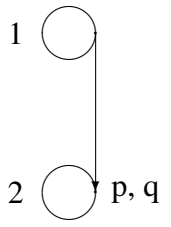
$$\dfrac{\dfrac{\dfrac{}{\Gamma, \forall x B(x), B(b) \vdash B(b)}\,\oplus \quad \dfrac{}{\Gamma, \forall x B(x), B(b) \vdash \neg B(b)}\,\oplus}{\Gamma, \forall x B(x), B(b) \vdash}}{\dfrac{\Gamma, \forall x B(x) \vdash}{\Gamma \vdash \neg\forall x B(x)}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash A(a)}\,\oplus \quad \dfrac{}{\Gamma \vdash A(a) \to \forall x B(x)}\,\oplus}{\Gamma \vdash \forall x B(x)} \quad \dfrac{}{Sk}}{A(a) \to \forall x B(x), A(a), \neg B(b) \vdash}}{A(a) \to \forall x B(x), A(a) \vdash B(b)}}{A(a) \to \forall x B(x) \vdash A(a) \to B(b)}}{A(a) \to \forall x B(x) \vdash \forall x(A(a) \to B(x))}$$

8e.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x(A(x) \to B(a)), A(a) \to B(a) \vdash A(a) \to B(a)}\,\oplus}{\forall x(A(x) \to B(a)) \vdash A(a) \to B(a)}}{\forall x(A(x) \to B(a)) \vdash \exists x(A(x) \to B(a))}}{\vdash \forall x(A(x) \to B(a)) \to \exists x(A(x) \to B(a))}$$

9a.



Turim refleksyvų ir tanzityvų orientuotą grafą. Yra lankas iš 1-os viršūnės į 2-ąją.

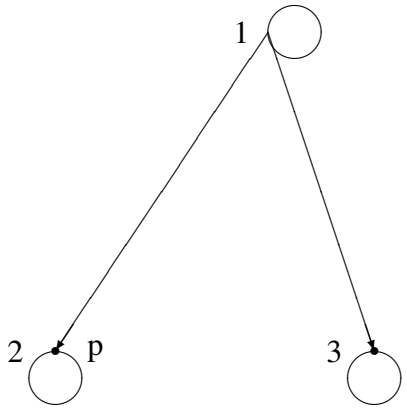
Pirmoje viršūnėje $\Phi, 1 \nVdash p$, nes priskirtoje viršūnei 1 aibėje nėra *p* (priskirtoji aibė yra tuščia). Taip pat ir $\Phi, 1 \nVdash q$.

Priminsime, kad formulė $F \to G$ įvykdoma viršūnėje *s* tada ir tiktai tada, kai *G* įvykdoma kiekvienoje viršūnėje, kurioje įvykdoma *F* (jei į tą viršūnę galima patekti iš *s*). Kadangi *p* įvykdoma 2-oje viršūnėje (į ją galima patekti iš 1-os), kurioje įvykdoma ir *q* tai $\Phi, 1 \vDash p \to q$.

$\Phi, 1 \nVdash \neg p$, nes tam, kad būtų įvykdoma $\neg p$, reikia kad visose viršūnėse, į kurias galima patekti iš 1-os, nebūtų *p*. Bet 2-oje viršūnėje yra *p*. $\Phi, 1 \nVdash q$, todėl $\Phi, 1 \nVdash \neg p \vee q$.

Gavome, kad 1-oje viršūnėje $\Phi, 1 \vDash p \to q$ ir $\Phi, 1 \nVdash \neg p \vee q$. Todėl $\Phi, 1 \nVdash (p \to q) \to (\neg p \vee q)$.

9b. Turim refleksyvų ir tranzityvų grafą Φ. Yra lankai iš 1-os viršūnės į 2-ąją ir 3-iąją bei visose viršūnėse kilpos.



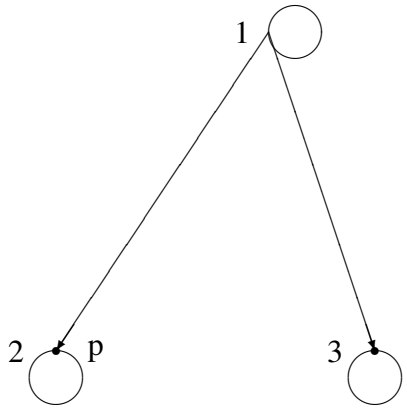
Visose viršūnėse neįvykdoma $p \wedge q$, todėl visose viršūnėse įvykdoma $\neg(p \wedge q)$. Taigi, turime, kad $\Phi, 1 \vDash \neg(p \wedge q)$.

$\Phi, 1 \nVdash \neg p$, nes tam, kad būtų įvykdoma $\neg p$, reikia kad visose viršūnėse, į kurias galima patekti iš 1-os, nebūtų *p*. Bet 2-oje viršūnėje yra *p*. Panašiai įrodoma, kad $\Phi, 1 \nVdash \neg q$ (3-ioje viršūnėje yra *q*). Todėl $\Phi, 1 \nVdash \neg p \vee \neg q$.

Vadinasi, $\Phi, 1 \nVdash \neg(p \wedge q) \to (\neg p \vee \neg q)$.

Radom Kripke's struktūrą ir joje pasaulį (viršūnę 1), kuriame formulė klaidinga. Vadinasi, $\neg(p \wedge q) \to (\neg p \vee \neg q)$ nėra intuicionistinės logikos dėsnis (nėra tapačiai teisinga pagal Kripke's semantiką).

9c.



1-oje viršūnėje *p* neįvykdomas, nes aibei 1 priskirtų kintamųjų aibė tuščia. Taigi, $\Phi, 1 \nVdash p$.

1-oje viršūnėje neįvykdomas ir $\neg p$. samprotaujama taip pat, kaip ir pavyzdžiuose 9a, 9b. Taigi, $\Phi, 1 \nVdash \neg p$.

1-oje viršūnėje neįvykdoma ir $\neg\neg p$. Tam kad būtų įvykdoma $\neg\neg p$ ($\neg\neg p = \neg(\neg p)$), reikia, kad visose viršūnėse $\Phi, i \nVdash \neg p$ $(i = 1, 2, 3)$. Bet $\neg p$ įvykdoma viršūnėje 3. Todėl $\Phi, 1 \nVdash \neg\neg p$.

Priminsime apibrėžimą

$\neg A$ *įvykdoma* viršūnėje *s*, jei visose viršūnėse, į kurias galima patekti iš *s, neįvykdoma A*.

Vadinasi, $\Phi, 3 \vDash \neg p$, nes iš 3-ios viršūnės galima patekti tik į 3-iąją viršūnę, o jai priskirtoje aibėje nėra p.

Taigi, struktūros Φ pirmoje viršūnėje neįvykdoma $p \vee \neg p \vee \neg\neg p$. Todėl $p \vee \neg p \vee \neg\neg p$ nėra intuicionistinės logikos dėsnis.

10a.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p \wedge \neg p \vdash p \wedge \neg p}\,\oplus}{p \wedge \neg p \vdash p} \quad \dfrac{\dfrac{}{p \wedge \neg p \vdash p \wedge \neg p}\,\oplus}{p \wedge \neg p \vdash \neg p}}{p \wedge \neg p \vdash}}{\vdash \neg(p \wedge \neg p)}$$

10b.
Iš pradžių įrodome sekvenciją *Sk*. $Sk = \neg p \vee q, p, \neg p \vdash q$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg p \vee q, p, \neg p \vdash p}\,\oplus}{\neg p \vee q, p, \neg p \vdash} \quad \dfrac{\dfrac{}{\neg p \vee q, p, \neg p \vdash \neg p}\,\oplus}{\neg p \vee q, p, \neg p \vdash}}{\neg p \vee q, p, \neg p \vdash q}$$

$$\dfrac{\dfrac{\dfrac{\oplus}{\neg p \vee q, p, \vdash \neg p \vee q} \quad \overline{Sk} \quad \dfrac{\oplus}{\neg p \vee q, p, q \vdash q}}{\neg p \vee q, p \vdash q}}{\dfrac{\neg p \vee q \vdash p \to q}{\vdash (\neg p \vee q) \to (p \to q)}}$$

10c. Įrodysime sekvencijas $Sk1 = \neg p \wedge \neg q, p \vee q, p \vdash$, $Sk2 = \neg p \wedge \neg q, p \vee q, q \vdash$ ir, remdamiesi jomis, įrodysime $\vdash (\neg p \wedge \neg q) \to \neg(p \vee q)$.

$$\dfrac{\dfrac{\dfrac{\oplus}{\neg p \wedge \neg q, p \vee q, p \vdash p} \quad \dfrac{\oplus}{\neg p \wedge \neg q, p \vee q, p \vdash \neg p \wedge \neg q}}{\neg p \wedge \neg q, p \vee q, p \vdash \neg p}}{\neg p \wedge \neg q, p \vee q, p \vdash}$$

$$\dfrac{\dfrac{\dfrac{\oplus}{\neg p \wedge \neg q, p \vee q, q \vdash q} \quad \dfrac{\oplus}{\neg p \wedge \neg q, p \vee q, q \vdash \neg p \wedge \neg q}}{\neg p \wedge \neg q, p \vee q, q \vdash \neg q}}{\neg p \wedge \neg q, p \vee q, q \vdash}$$

$$\dfrac{\dfrac{\dfrac{\oplus}{\neg p \wedge \neg q, p \vee q \vdash p \vee q} \quad \overline{Sk1} \quad \overline{Sk2}}{\neg p \wedge \neg q, p \vee q \vdash}}{\dfrac{\neg p \wedge \neg q \vdash \neg(p \vee q)}{\vdash (\neg p \wedge \neg q) \to \neg(p \vee q)}}$$

10d.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\oplus}{\neg\neg\neg p, p, \neg p \vdash p} \quad \dfrac{\oplus}{\neg\neg\neg p, p, \neg p \vdash \neg p}}{\neg\neg\neg p, p, \neg p \vdash}}{\neg\neg\neg p, p \vdash \neg\neg p} \quad \dfrac{\oplus}{\neg\neg\neg p, p \vdash \neg\neg\neg p}}{\neg\neg\neg p, p \vdash}}{\dfrac{\neg\neg\neg p \vdash \neg p}{\vdash \neg\neg\neg p \to \neg p}}$$

11. Priminsime, kad:

termas ***MNE*** lygus $((MN)E)$,

termas $\lambda x.M$ lygus $(\lambda x.(M))$,

termas $\lambda x.\lambda y.\lambda z.M$ lygus $(\lambda x.(\lambda y.(\lambda z.M)))$,

redeksas yra pavidalo $(\lambda x.E)Y$. Redeksus sprendimuose *pabrauksime*.

Pastabos:
1. Nebūtina rašyti visų skliaustų, jei jums aišku kaip redukuoti,
2. Nors λ-skaičiavime naudojami tik skliaustai (,), bet, kad būtų aiškiau (metodiniais sumetimais), kartais naudosime ir laužtinius [,].

11a. $(\lambda x.\lambda y.y(xxy))fg.$

$(\lambda x.\lambda y.y(xxy))fg = \underline{[(\lambda x.\lambda y.y(xxy))f]}g \quad \beta \triangleright \quad \underline{\lambda y.y(ffy)]g} \quad \beta \triangleright \quad g(ffg).$

11b. $(\lambda y.yzy)(\lambda x.cd).$

$(\lambda y.yzy)(\lambda x.cd) \quad \beta \triangleright \quad (\lambda x.cd)z(\lambda xcd) = \underline{[(\lambda x.cd)z]}(\lambda x.cd) \quad \beta \triangleright \quad cd(\lambda x.cd).$

11c. $\lambda x.\lambda y.(\lambda z.zv)yx.$

$\lambda x.\lambda y.(\lambda z.zv)yx = \lambda x.\lambda y.[[\underline{(\lambda z.zv)y}]x] \quad \beta \triangleright \quad \lambda x.\lambda y.[[yv]x].$

11d. $(\lambda x.((\lambda y.xy)u))(\lambda v.v)$

$(\lambda x.((\lambda y.xy)u))(\lambda v.v) \quad \beta \triangleright \quad \underline{(\lambda y.(\lambda v.v)y)u} \quad \beta \triangleright \quad \underline{(\lambda v.v)u} \quad \beta \triangleright \quad u.$

11e. $\lambda f.\lambda x.(\lambda x.fx)((\lambda x.fx)x)$

$\lambda f.\lambda x.(\lambda x.fx)((\lambda x.fx)x) \quad \alpha \triangleright$ (pervardijimas) $\lambda f.\lambda x.(\lambda u.fu)(\underline{(\lambda v.fv)x}) \quad \beta \triangleright$
$\lambda f.\lambda x.\underline{(\lambda u.fu)(fx)} \quad \beta \triangleright \quad \lambda f.\lambda x.f(fx).$

11f. $(\lambda x.\lambda y.yx)zv.$

$(\lambda x.\lambda y.yx)zv = \underline{[(\lambda x.\lambda y.yx)z]}v \quad \beta \triangleright \quad \underline{[\lambda y.yz]v} \quad \beta \triangleright \quad vz.$

11g. $(\lambda x.(\lambda y.yx)z)v.$

$\underline{(\lambda x.(\lambda y.yx)z)v} \quad \beta \triangleright \quad \underline{(\lambda y.yv)z} \quad \beta \triangleright \quad zv.$

Ši užduotis įdomi tuo, kad yra du redeksai pradiniame terme $(\lambda x.(\lambda y.yx)z)v$ ir $(\lambda y.yx)z$. Kuri rinktis? Reikia rinktis pirmą iš kairės, t.y. $(\lambda x.(\lambda y.yx)z)v$. Nors šiuo atveju ir nesvarbu. Abiem atvejais gautumėt tą patį rezultatą.

11h. $(\lambda x.\lambda y.x(xy))fu.$

$(\lambda x.\lambda y.x(xy))fu = \underline{[(\lambda x.\lambda y.x(xy))f]}u \quad \beta \triangleright \quad \underline{[\lambda y.f(fy)]u} \quad \beta \triangleright \quad f(fu).$

11i. $\lambda x.\lambda y.x((\lambda x.\lambda y.x(x(xy)))xy).$

$\lambda x.\lambda y.x((\lambda x.\lambda y.x(x(xy)))xy) \quad \alpha \triangleright \quad \lambda x.\lambda y.x((\lambda u.\lambda v.u(uv)))xy) = \lambda x.\lambda y.x(\underline{[(\lambda u.\lambda v.u(u(uv)))x]}y)$
$\beta \triangleright \quad \lambda x.\lambda y.x(\underline{[\lambda v.x(x(xv))]y}) \quad \beta \triangleright \quad \lambda x.\lambda y.x(x(x(xy))).$

12a. Atsakymas: x -trečia, y - trečia, ketvirta, z - pirma.

12b Atsakymas: nėra laisvųjų kintamųjų, nes

$\lambda z.\lambda x.(\lambda x.zx)((\lambda x.fx)z)(\lambda x.\lambda y.yzx) = \lambda z.[\lambda x.(\lambda x.zx)((\lambda x.fx)z)(\lambda x.\lambda y.yzx)]$.

12c. Atsakymas: x - trečia, y - penkta, z - pirma, antra.

13. Tipizuoti termą $\lambda x.\lambda y.\lambda z.y(xz)$, kai duotas kontekstas $\Gamma = \{x : p \to q, y : q \to r, z : p\}$.

Nuoroda. Kai termas yra pavidalo $\lambda x_1.\lambda x_2.\ldots.\lambda x_n.M$ ir terme *M* nėra λ įeičių, tai *M* vadinamas termo *kūnu*. Prisilaikysime tokios stategijos: visų pirma taikysime tik taisyklę $\to$ *šalinimas* tol, kol rasime termo *kūno tipą*. o po to, taikydami tik taisyklę $\to$ *įvedimas*, rasime *duoto termo tipą*. Tipo nustatymas vyksta *iš viršaus į apačią*.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash y : q \to r \quad \dfrac{\Gamma \vdash x : p \to q \;\; \Gamma \vdash z : p}{\Gamma \vdash xz : q}}{x : p \to q, y : q \to r, z : p \vdash y(xz) : r}}{x : p \to q, y : q \to r \vdash \lambda z : p.y(xz) : p \to r}}{x : p \to q \vdash \lambda y : q \to r.\lambda z : p.y(xz) : (q \to r) \to p \to r}}{\vdash \lambda x : p \to q.\lambda y : q \to r.\lambda z : p.y(xz) : (p \to q) \to (q \to r) \to p \to r}$$

Atsakymas: $\lambda x : p \to q.\lambda y : q \to r.\lambda z : p.y(xz) : (p \to q) \to (q \to r) \to p \to r$.

14. Rasti termą, kurio tipas yra s, naudojant tik taisyklę $\to$ *šalinimas*. Kontekstas $\Gamma = \{x : p, u : p \to q, y : p \to (p \to r), z : r \to (q \to s)\}$

Termo paieška vykdoma *iš viršaus į apačią*.

$$\dfrac{\Gamma \vdash z : r \to (q \to s) \quad \dfrac{\dfrac{\Gamma \vdash y : p \to (p \to r) \;\; \Gamma \vdash x : p}{\Gamma \vdash yx : p \to r} \;\; \Gamma \vdash x : p}{\Gamma \vdash (yx)x : r}}{\Gamma \vdash z((yx)x) : q \to s}$$

$$\dfrac{\Gamma \vdash u : p \to q \quad \Gamma \vdash x : p}{\Gamma \vdash ux : q}$$

iš dviejų gautų sekvencijų $\Gamma \vdash z((yx)x) : q \to s \;\; \Gamma \vdash ux : q$ ir gauname termą, kurio tipas s.

$$\dfrac{\Gamma \vdash z((yx)x) : q \to s \;\; \Gamma \vdash ux : q}{\Gamma \vdash (z((yx)x))(ux) : s}$$

Atsakymas: $(z((yx)x))(ux) : s$.

15.

$$\dfrac{\dfrac{\dfrac{}{\neg : o \to o}\oplus \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\leftrightarrow : o \to (o \to o)}\oplus \quad \dfrac{\dfrac{\dfrac{}{\vee : o \to (o \to o)}\oplus \quad \dfrac{}{q : o}\oplus}{\vee q : o \to o}}{(\vee q)r : o} \quad \dfrac{}{r : o}\oplus}{\leftrightarrow ((\vee q)r) : o \to o}}{(\leftrightarrow ((\vee q)r))p : o} \quad \dfrac{}{p : o}\oplus}{\neg((\leftrightarrow ((\vee q)r))p) : o}$$

16. Pažymėkime:

$A = (((p \to q) \to p) \to p) \to q$,

$B = (p \to q) \to p$.

Visų pirma išvesime $C = x : A, y : B, z : p \vdash x\lambda u.z : q$.

$$\dfrac{\dfrac{}{x : A, y : B, z : p \vdash x : A}\oplus \quad \dfrac{\dfrac{}{x : A, y : B, z : p, u : ((p \to q) \to p) \to p \vdash z : p}\oplus}{x : A, y : B, z : p \vdash \lambda u.z : ((p \to q) \to p) \to p}}{x : A, y : B, z : p \vdash x\lambda u.z : q}$$

$$\dfrac{\dfrac{}{x : A \vdash x : A}\oplus \quad \dfrac{\dfrac{\dfrac{}{x : A, y : B \vdash y : B}\oplus \quad \dfrac{C}{x : A, y : B \vdash \lambda z.x\lambda u.z : p \to q}}{x : A, y : B \vdash y\lambda z.x\lambda u.z : p}}{x : A \vdash \lambda y.y\lambda z.x\lambda u.z : ((p \to q) \to p) \to p}}{x : A \vdash x\lambda y.y\lambda z.x\lambda u.z : q}$$

$$\vdash \lambda x.x\lambda y.y\lambda z.x\lambda u.z : ((((p \to q) \to p) \to p) \to q) \to q$$

5 Skyrius. Priedai

B priedas

1.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{F \vdash F}\oplus \quad \dfrac{}{F, G \vdash G}\oplus}{F, F \to G \vdash G}}{F, \neg G, F \to G \vdash}}{F, \neg G \vdash \neg(F \to G)}}{\neg\neg(F \to G), F, \neg G \vdash}}{\neg\neg(F \to G), F \vdash \neg\neg G}}{\neg\neg(F \to G) \vdash F \to \neg\neg G}}{\vdash \neg\neg(F \to G) \to (F \to \neg\neg G)}$$

2.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{H \to (F \to G), H \vdash H}\oplus \quad \dfrac{\dfrac{}{F, H \vdash H}\oplus \quad \dfrac{\dfrac{}{F, H \vdash F}\oplus \quad \dfrac{}{F, G, H \vdash G}\oplus}{F, F \to G, H \vdash G}}{F, H \to (F \to G), H \vdash G}}{H \to F, H \to (F \to G), H \vdash G}}{H \to F, H \to (F \to G) \vdash H \to G}}{H \to F \vdash ((H \to (F \to G)) \to (H \to G))}}{\vdash (H \to F) \to ((H \to (F \to G)) \to (H \to G))}$$

3.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{F(a), \neg\neg F(a) \vdash F(a)}\oplus}{F(a) \vdash \neg\neg F(a) \to F(a)}}{\neg(\neg\neg F(a) \to F(a)), F(a) \vdash}}{\neg(\neg\neg F(a) \to F(a)) \vdash \neg F(a)}}{\neg(\neg\neg F(a) \to F(a)), \neg\neg F(a) \vdash F(a)}}{\neg(\neg\neg F(a) \to F(a)) \vdash \neg\neg F(a) \to F(a)}}{\neg(\neg\neg F(a) \to F(a)), \neg(\neg\neg F(a) \to F(a)) \vdash}}{\neg(\neg\neg F(a) \to F(a)) \vdash}}{\vdash \neg\neg(\neg\neg F(a) \to F(a))}}{\vdash \exists y(\neg\neg(\neg\neg F(a) \to F(y)))}}{\vdash \exists x \exists y(\neg\neg(\neg\neg F(x) \to F(y)))}$$

4.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\neg F(a) \to F(a), \neg\neg(F(a) \to F(a)), F(a) \vdash F(a)}\oplus}{\neg\neg F(a) \to F(a), \neg\neg(F(a) \to F(a)) \vdash F(a) \to F(a)}}{\neg\neg F(a) \to F(a) \vdash \neg\neg(F(a) \to F(a)) \to (F(a) \to F(a))}}{\vdash (\neg\neg F(a) \to F(a)) \to (\neg\neg(F(a) \to F(a)) \to (F(a) \to F(a)))}}{\vdash \exists y((\neg\neg F(a) \to F(y)) \to (\neg\neg(F(a) \to F(y)) \to (F(a) \to F(y))))}}{\vdash \forall x \exists y((\neg\neg F(x) \to F(y)) \to (\neg\neg(F(x) \to F(y)) \to (F(x) \to F(y))))}$$

5.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{F(a)\vdash F(a)}\oplus}{F(a)\vdash F(a)\vee G(a)}}{F(a)\vdash \exists x(F(x)\vee G(x))}}{\exists xF(x)\vdash \exists x(F(x)\vee G(x))} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{G(a)\vdash G(a)}\oplus}{G(a)\vdash F(a)\vee G(a)}}{G(a)\vdash \exists x(F(x)\vee G(x))}}{\exists xG(x)\vdash \exists x(F(x)\vee G(x))}}{\exists xF(x)\vee \exists xG(x)\vdash \exists x(F(x)\vee G(x))}$$

C priedas

1. $\neg F\to\neg\neg\neg F$ keičiame į $\vdash (F\to\bot)\to(((F\to\bot)\to\bot)\to\bot)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{F\to\bot,(F\to\bot)\to\bot\vdash F\to\bot}\oplus \quad \dfrac{}{F\to\bot,(F\to\bot)\to\bot\vdash (F\to\bot)\to\bot}\oplus}{F\to\bot,(F\to\bot)\to\bot\vdash\bot}}{F\to\bot\vdash((F\to\bot)\to\bot)\to\bot}}{\vdash(F\to\bot)\to(((F\to\bot)\to\bot)\to\bot)}$$

2. $\vdash (F\to G)\to(\neg G\to\neg F)$ keičiame į $\vdash (F\to G)\to((G\to\bot)\to(F\to\bot)$.

Įrodyme raide Γ žymime formulių seką $F\to G, G\to\bot, F$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma\vdash F}\oplus \quad \dfrac{}{\Gamma\vdash F\to G}\oplus}{\Gamma\vdash G} \quad \dfrac{}{\Gamma\vdash G\to\bot}\oplus}{\Gamma\vdash\bot}}{F\to G,G\to\bot\vdash F\to\bot}}{F\to G\vdash(G\to\bot)\to(F\to\bot)}}{\vdash(F\to G)\to((G\to\bot)\to(F\to\bot))}$$

D priedas

1. $\vdash (\mathbf{A} \vee \mathbf{B}) \to (\mathbf{B} \vee \mathbf{A})$

1. $(A \to (B \vee A)) \to ((B \to (B \vee A)) \to ((A \vee B) \to (B \vee A)))$ aksioma 3.3 $(C - B \vee A)$

2. $A \to (B \vee A)$ aksioma 3.2 (A - B; B - A)

3. $(B \to (B \vee A)) \to ((A \vee B) \to (B \vee A))$ MP (2, 1)

4. $B \to (B \vee A)$ aksioma 3.1 $(A - B; B - A)$

5. $(A \vee B) \to (B \vee A)$ MP(4,3)

2. $\vdash \mathbf{A} \to \neg(\neg\mathbf{A} \wedge \neg\mathbf{B})$

1. $(\neg A \wedge \neg B) \to \neg A$ aksioma 2.1 $(A - \neg A; B - \neg A)$

2. $((\neg A \wedge \neg B) \to \neg A) \to (\neg\neg A \to \neg(\neg A \wedge \neg B))$ aksioma 4.1 $(A - \neg A \wedge \neg B; B - \neg A)$

3. $\neg\neg A \to \neg(\neg A \wedge \neg B)$ MP(1,2)

4. $A \to \neg\neg A$ aksioma 4.2

Norėdami iš 4 ir 3 formulių gauti $A \to \neg(\neg A \wedge \neg B)$, panaudosime implikacijos tranzityvumą aprašančią aksiomų schemą 1.2 bei aksiomų schemą 1.1.

5. $(A \to (\neg\neg A \to \neg(\neg A \wedge \neg B))) \to ((A \to \neg\neg A) \to (A \to \neg(\neg A \wedge \neg B)))$

aksioma 1.2 $(B - \neg\neg A; C - \neg(\neg A \wedge \neg B))$

6. $(\neg\neg A \to \neg(\neg A \wedge \neg B)) \to (A \to (\neg\neg A \to \neg(\neg A \wedge \neg B)))$

aksioma 1.1 $(A - \neg\neg A \to \neg(\neg A \wedge \neg B); B - A)$

7. $(A \to \neg\neg A) \to (A \to \neg(\neg A \wedge \neg B))$ MP (3,6)

8. $A \to \neg(\neg A \wedge \neg B)$ MP (4,7)

3. $\vdash (\mathbf{A} \wedge \mathbf{B}) \to (\mathbf{A} \wedge (\mathbf{B} \vee \mathbf{C}))$.

1. $((A \wedge B) \to A) \to (((A \wedge B) \to (B \vee C)) \to ((A \wedge B) \to (A \wedge (B \vee C)))))$.

aksioma 2.3 $(A - A \wedge B; B - A; C - B \vee C)$

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

3. $((A \wedge B) \to (B \vee C)) \to ((A \wedge B) \to (A \wedge (B \vee C))))$ MP (2,1)

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

5. $B \to (B \vee C)$ aksioma 3.1

6. $((A \wedge B) \to (B \to (B \vee C))) \to (((A \wedge B) \to B) \to ((A \wedge B) \to (B \vee C)))$

aksioma 1.2 $(A - A \wedge B; C - B \vee C)$

7. $(B \to (B \vee C)) \to ((A \wedge B) \to (B \to (B \vee C)))$

aksioma 1.2 $(A - B \to (B \vee C); B - A \wedge B)$

8. $(A \wedge B) \to (B \to (B \vee C))$ MP (5,7)

9. $((A \wedge B) \to B) \to ((A \wedge B) \to (B \vee C))$ MP (8,6)

10.$(A \wedge B) \to (B \vee C)$ MP (4,9)

11. $(A \wedge B) \to (A \wedge (B \vee C))$ MP (10,3)

4. $\mathbf{A} \to (\mathbf{B} \to \mathbf{C}), \neg\mathbf{C}, \mathbf{A} \vdash \neg\mathbf{B}$.

1. $A \to (B \to C)$ prielaida

2. A prielaida

3. $B \to C$ MP (2,1)

4. $(B \to C) \to (\neg C \to \neg B)$ aksioma 4.1 $(A - B; B - C)$

5. $\neg C \to \neg B$ MP (3,4)

6. $\neg C$ prielaida

7. $\neg B$ MP (6,5)

5. $(\mathbf{A} \vee \mathbf{B}) \to \mathbf{C}, (\mathbf{C} \vee \mathbf{D}) \to \mathbf{E}, \mathbf{A} \vdash \mathbf{E}$.

1. $A \to (A \vee B)$ aksioma 3.1

2. A prielaida

3. $A \vee B$ MP (2,1)

4. $(A \vee B) \to C$ prielaida

5. C MP (3,4)

6. $C \to (C \vee D)$ aksioma 3.1 $(A - C; B - D)$

7. $C \vee D$ MP (5,6)

8. $(C \vee D) \to E$ prielaida

9. E MP (7,8)

6. $\vdash (\mathbf{A} \to (\mathbf{B} \to \mathbf{C})) \to (\mathbf{B} \to (\mathbf{A} \to \mathbf{C}))$

1. $A \to (B \to C) \vdash B \to (A \to C)$ pagal dedukcijos teoremą

2. $A \to (B \to C), B \vdash A \to C$ pagal dedukcijos teoremą

3. $A \to (B \to C), B, A \vdash C$ pagal dedukcijos teoremą

Taigi, $\vdash (A \to (B \to C)) \to (B \to (A \to C))$ įrodoma tada ir tiktai tad, kai įrodoma $A \to (B \to C), B, A \vdash C$.

Įrodysime $A \to (B \to C), B, A \vdash C$.

4. A prielaida

5. $A \to (B \to C)$ prielaida

6. $B \to C$ MP (4,5)

7. B prielaida

8. C MP (7,6)

Naudota literatūra

1. S.Norgėla, *Logika ir dirbtinis intelektas*, Vilnius: TEV, 2007.

2. S.Norgėla, *Matematinė logika*, Vilnius: TEV, 2004.

3. G.S.Boolos, J.P.Burgess, R.C.Jeffrey, *Computability and Logic*, Cambridge University Press, 2007.

4. S.G.Simpson, *Subsystems of Second Order Arithmetic*, Cambridge University Press, NY, 2009.

S.Norgėla, *Matematinė logika informatikams* (*paskaitų konspektai*).
https://sites.google.com/view/stanislovasnorgela

Rekomenduojama literatūra

1. M.Ben Ari, *Mathematical logic for computer science*, Springer, 2008.

2. U.Schöning, *Logic for computer science*, Birkhäuser Boston, 2012.