Loogika

Mõtlemisest tõestamiseni

Loogika

Mõtlemisest tõestamiseni

Tõnu Tamme, Tanel Tammet, Rein Prank

Tartu Ülikooli Kirjastus

Raamat valmis Eesti Teadusfondi ja Avatud Eesti Fondi toetusel.

Keeletoimetaja Leelo Jago
Kaane kujundanud Lemmi Koni

Autorite postiaadressid: *tqnu@cs.ut.ee* (Tõnu Tamme)
tammet@cs.chalmers.se (Tanel Tammet)
prank@cs.ut.ee (Rein Prank)

ISBN 9985-56-231-3

Kaas: “Postimehe” reprokeskus / Tõravere Trükikoda
Trükk: OÜ Greif
Köide: Tartu Ülikooli Kirjastuse trükikoda
Tellimus nr. 29

Autorid kasutasid küljendamiseks süsteemi LaTeX ning trükikirju Times, MathTime ja LucidaSans-Typewriter.

Eessõna

Maailma esimest raamatut predikaatarvutuse kohta — Begriffsschrift, 1879 — *alustab autor ja arvutuse looja, Gottlob Frege, sõnadega:* Teadusliku tõe tunnetus käib reeglina läbi mitmed tõsikindluse astmed … kindlaim tõestusviis on ilmselt puhtloogiline, mis kõigist asjade eriomadustest kõrvale vaadates tugineb ainuüksi nendele seadustele, millel põhineb kogu tunnetus üldse. *Oma* mõistekirja — *sisuliselt, nüüdisaegses tähenduses identsusega kõrgemat järku predikaatarvutuse — vajalikkust põhjendab ta võrdluses tavakeelega:* Püüdes rangeimal viisil järgida neid nõudeid *(tõestuskäigu augutus)*, leidsin ma takistusena ees keele, mis … mida keerulisemaks muutusid seosed, seda vähem lubas saavutada mu eesmärkide nõutavat täpsust … Seost minu mõistekirja ja tavakeele vahel arvan võivat kõige selgemaks teha mikroskoobi võrdluses silmaga. Viimane on kasutuse ulatuse, liikuvuse tõttu, millega suudab end kohandada kõige erinevamate asjaoludega, suuresti üle mikroskoobist … Niipea aga kui teaduslikud eesmärgid püstitavad kõrgeid nõudmisi eraldusvõime teravusele, osutub silm ebapiisavaks. Mikroskoop on just sedalaadi eesmärkideks täiuslikult kohandatud, aga just seetõttu kõigiks teisteks kasutuskõlbmatu. Nii ongi see mõistekiri kindlateks teaduslikeks eesmärkideks leiutatud abivahend, mida sellepärast ära ei tule põlata, et ta teistsugusteks ei kõlba.

Käesolev eestikeelne loogikaraamat peab siis — lubades endale analoogiat — olema see teaduslikku täpsust võimaldav mikroskoop tänapäevaselt täiustatud kujul; ehk küll esmajoones kooli-otstarbeks mõteldud.

Mida temaga teha saab? Kellele teda vaja läheb — sest nagu

üteldud, kõigeks ja kõigile ta ju ei kõlba? Muidugi ja esmajoones on see raamat matemaatikutele, sest matemaatika, täpsemalt, aritmeetika aluste uurimiseks, oligi mõttekiri esialgselt mõteldud. Kuid loogika on arenenud ja matemaatika koos temaga ning loogika kasutus ei piirdu enam matemaatika alustega, loogika ise on iseseisev, omaenda sisemiste impulsside ajendil arenev teadus, mille otsesed praktilised rakendusvõimalused, näiteks programmeerimises ja tehisintellektis, on hästi teada. Loogika loomisega pani aga Frege muu hulgas aluse täiesti uuele mõtlemisviisile, filosofeerimismoodusele, mille üheks tunnuseks on see, mida tänapäeval nimetatakse lingvistiliseks pöördeks.

Lingvistilises pöördes *sündinud filosoofia, olgu ta spetsiaalselt keele-, matemaatika- või vaimufilosoofia, ei ole viljeldav ilma loogilise aparatuuri abita.*

Tänapäeva filosoofilt nõutav matemaatiline ettevalmistus ei ole mahult väiksem füüsikult nõutavast ettevalmistusest. Kuid vajaminevad erialad on erinevad: filosoofi matemaatiline pagas haarab esmajoones matemaatilist loogikat.

Arvan, et just filosoofide rõõm käesoleva raamatu ilmumisest on suurim.

On muidugi loogikapeatükke, mis huvitab filosoofe enam, ja neid, mis vähem. Peale üldise lause- ja predikaatarvutuse, mis iseendastmõistetavalt kuulub hariduse juurde, on filosoofi erilisse huvipiirkonda haaratud näiteks mudelite teooria, mitmesugused intensionaalsed loogikad (modaal- ja ajaloogikad), intuitsionistlik loogika, semantika erinevad probleemid (Tarski tõeteooria) jpm. Kuid humanitaarteadustegi vajadus loogika järele on kasvanud. Filosoofia lingvistiline pööre *tõmbab kaasa ka lingvistika. Küllap ei ole lingvistide rõõm selle raamatu valmimise üle palju väiksem filosoofide omast, kuigi nende spetsiifilisi huve on ehk materjalivalikus vähem silmas peetud. Kuid raamat niisugusel kujul, nagu talle just on antud, ei olegi mõeldud erialade rahuldamiseks, oma olulises osas sisaldab ta (ka väljaspool klassikalise loogika peatükki) klassikalist materjali, mida niihästi matemaatik, filosoof, lingvist, kui mõne teisegi eriala asjatundja — miks mitte füüsikki — p e a b valdama.*

24.11.96

Madis Kõiv

Sisukord

Eessõna **v**

1. Sissejuhatus **1**
1.1. Loogika aine 1
1.2. Loogika ajalugu 19

I Klassikaline loogika **53**

2. Loogika põhimõisted **55**
2.1. Põhilised loogikaseadused 55
2.2. Vasturääkivus ja mittevasturääkivus 57
2.3. Arutlus ja järeldus 58
2.4. Arutluste kehtivus ja korrektsus 59
2.5. Lausete loogiline tõesus ja väärus 61
2.6. Lausete ekvivalentsus 62
2.7. Matemaatilised seosed 62
2.8. Ülesanded 64

3. Lausearvutus **65**
3.1. Lausete analüüs 65
3.2. Lausearvutuse tehted 68
3.3. Lausearvutuse süntaks ja semantika 76
3.4. Lausevormid ja arutlusvormid 80
3.5. Lausearvutuses mitteväljendatavad seosed 83
3.6. Tõeväärtustabelid 85
3.7. Tautoloogia, kontradiktsioon ja kontingentne lause . . . 88
3.8. Ülesanded 92

4. Predikaatarvutus **94**
4.1. Predikaadid ja konstandid 95
4.2. Kvantorid 98
4.3. Predikaatarvutuse süntaks 100
4.4. Tarski semantika 102
4.5. Loogiline ruut 108
4.6. Süllogismid 112
4.7. Korduv kvantifitseerimine 114
4.8. Signatuur ja interpretatsioon. Struktuuri ja struktuuride klassi teooria 115
4.9. Lõplikud struktuurid 121
4.10. Reaalarvude järjestuse elementaarteooria 123
4.11. Ülesanded 126

5. Klassikalised tuletusreeglid **130**
5.1. Tuletusreeglid 131
5.2. Konditsionaalne tõestus 140
5.3. Kaudne tõestus 144
5.4. Predikaatarvutuse tuletusreeglid 146
5.5. Formaalne ja mitteformaalne tõestus 154
5.6. Aksioom ja mudel 157
5.7. Aksiomatiseerimisteooria 164
5.8. Ülesanded 165

6. Tuletussüsteemid **168**
6.1. Hilberti tüüpi tuletus 168
6.2. Loomulik tuletus 172
6.3. Gentzeni sekventsiaalne arvutus 174
6.4. Tõesuspuu 179

II Matemaatiline loogika **185**

7. Matemaatiline induktsioon **187**
7.1. Induktiivne tõestus 187
7.2. Duaalsus 189

8. Lausearvutuse täielikkus **192**
8.1. Lausearvutuse korrektsus ja mittevasturääkivus 193
8.2. Lausearvutuse täielikkus 193

9. Mittelahenduvad probleemid **196**
9.1. Turingi masin 197
9.2. Mittelahenduvad probleemid 201
9.3. Predikaatarvutuse mittelahenduvus 208

10. Aritmeetika mittetäielikkus **213**
10.1. Formaalne aritmeetika 213
10.2. Gödeli teoreemid aritmeetika mittetäielikkusest 217

11. Kõrgemat järku loogika **221**
11.1. Aritmeetika kui teist järku teooria 221
11.2. Teist järku loogika 222
11.3. Kompaktsusteoreem 223
11.4. Väljendatavus teist järku loogikas 224
11.5. Tüübiteooria 226
11.6. Matemaatiline teooria 228
11.7. Rakendused arvutiteaduses ja programmeerimises 229
11.8. Ülesanded 230

III Mitteklassikaline loogika **233**

12. Modaalloogika **235**
12.1. Modaalsed operaatorid 235
12.2. Modaalloogika seadused 237
12.3. Võimalike maailmade semantika 239
12.4. Tuletused modaalloogikas $S5$ 241
12.5. Kanooniline tõlge predikaatloogika keelde 244
12.6. Erinevad tuletussüsteemid 245
12.7. Modaalne predikaatloogika 247

13. Intuitsionistlik loogika **249**
13.1. Sissejuhatus 249
13.2. Kaks tõestuse näidet 252
13.3. Intuitsionistlik loogikaseoste mõistmine 254
13.4. Intuitsionistlik predikaatarvutus 256
13.5. Kripke mudelid 258
13.6. Rekursiivne realiseeritavus. Nelsoni teoreem 261
13.7. Ülesanded 265

14. Hägusloogika **266**
14.1. Soriitide paradoksid 266
14.2. Hägusloogika semantika 268

15. Mittemonotoonne loogika **273**
15.1. McCarthy näide 273
15.2. Näite formaalne käsitlus 276
15.3. Suletud maailm 280
15.4. Piiramine 283
15.5. Vaikimisi-loogika 284
15.6. Autoepisteemiline loogika 286
15.7. Ülesanded 286

16. Lineaarloogika **287**
16.1. Tegevuste ja situatsioonide kordumatus 287
16.2. Kaks konjunktsiooni ja disjunktsiooni 289
16.3. Lineaarne eitus 290
16.4. Sekventsiaalne tuletus 291
16.5. Lineaarloogika väljendusvõimsus 293
16.6. Ülesanded 294

IV Loogika rakendusi programmeerimises ja tehisintellektis **295**

17. Automaatne teoreemitõestamine **297**
17.1. Eesmärgid 298
17.2. Sissejuhatus resolutsioonimeetodisse 299
17.3. Resolutsioonimeetodi strateegiad ja rakendused 327
17.4. Näiteid ja rakendusi 335

18. Loogiline programmeerimine ja Horni disjunktid **346**
18.1. Prolog 348

19. Funktsionaalne programmeerimine ja võrdussüsteemid **366**
19.1. Deduktsioon ja reduktsioon: termiteisendussüsteemid . . 367
19.2. Lambda-arvutus 376
19.3. Kombinatoorne loogika 382

Ülesannete vastused **387**

Kirjandus **390**

Indeks **395**

1. Sissejuhatus

Puhta loogika eesmärk on olla õige kõigis võimalikes maailmades, mitte ainult selles veidersegases vaevarikkas maailmas, kuhu juhus meid on heitnud. Loogik peab eneses alal hoidma teatud annuse jumalikkust: ta ei tohi alanduda selleni, et teha järeldusi enese ümber nähtust.
— B. Russell, *Sissejuhatus matemaatilisse filosoofiasse*

Kui loogika oleks olemas isegi juhul, kui maailma ei oleks, siis kuidas saab loogika olemas olla olukorras, kus maailm on olemas?
— L. Wittgenstein, *Tractatus Logico-Philosophicus*

1.1. Loogika aine

Esimene küsimus, mis meil esitada tuleb, on küsimus loogika ainest. Mis asi on *loogika*, ja mida loogikateadus uurib? Sageli jaotatakse loogika aine mitmesse eraldiseisvasse gruppi: matemaatiline loogika, filosoofiline loogika, dialektiline loogika jne. Meie raamat käsitleb peamiselt *matemaatilist* ehk formaalset loogikat, mis on tänapäeva loogikateaduse tuum ja arenenuim osa. Seejuures on filosoofiline loogika matemaatilise loogikaga väga tihedas suhtes ning üht ei saa olla ilma teiseta: matemaatiline loogika on alati

ka filosoofiline loogika, ning nö. mittematemaatiline filosoofiline loogika on samas oluline matemaatilise loogika jaoks.

Tuleme tagasi loogika aine juurde. Lühidalt ja robustselt öeldes uurib loogika *mõtlemise* kõige fundamentaalsemaid aspekte. Mis asi mõtlemine õieti on ja mis on tema fundamentaalsed aspektid? Mõtlemist uurivad paljud distsipliinid, näiteks psühholoogia, ning miks mitte ka ajalugu ja kirjandusteadus. Loogika eripäraks on uurida, *mida üldse saab mõelda* ja mida mõelda ei saa — võiksime öelda, et loogika uurib puhast ehk täielikult abstraheeritud mõtlemist.

Loogika, mõtlemise ja maailma suhete skaalal on mitu vastaspoolust: ühel pool seisab loogika kui jumaliku, paratamatu ning maailmast sõltumatu tõe uurimine, teiselt poolt aga ei pääse loogika asjaolust, et loogikud on konkreetsed inimesed reaalses maailmas ning neile omased fundamentaalsed mõtlemisprintsiibid ei pruugi olla needsamad, mis kujutletavatel kosmoses hõljuvatel ülimõtlejatel.

1.1.1. Mõtlemine

Vaimne tegevus ehk ajutegevus ei koosne kunagi ainult sihipärasest mõtlemisest: suurem osa inimajust tegeleb igasuguste muude, samuti väga oluliste asjadega, nagu silmanärvidest saabunud kujutise analüüs, kõne süntees, assotsiatsioonide otsimine mälust, instinktimehhanismide järgimine jne. jne. Isegi juhul, kui seda kõike peaks koordineerima meile tundmatu mittemateriaalne jumalik vaim, jääb nimetatud vaim ikkagi ainult üheks vaimse tegevuse komponendiks. Tegelik mõtlemisprotsess, nagu me enda peal kogeme, koosneb tuhandetest protsessidest, mida mõtleja reeglina ei teadvusta — teadvus on vaimse tegevuse jäämäe pisike veepinnale ulatuv nurk.

Igasugune vaimne tegevus nagu ka igasugune protsess üldse, ei ole kindlasti mitte alati mõtlemisprotsess. Poolunes lebamist, toidu ja veini nautimist ning mälestustes uitamist ei saa nimetada mõtlemiseks selle sõna üldkasutatavas tähenduses. Sõnaga

mõtlemine tähistatakse enamasti sihipärast vaimset tegevust mingi kuigivõrd selgelt kirjeldatava probleemi lahendamiseks.

Ei ole võimalik öelda, mis asi on *mõte* või defineerida selgelt mõtlemist kui niisugust. Mida me saame teha, on tuua näiteid nii mõtlemisest kui sellisest vaimsest tegevusest, mida mõtlemiseks nimetada ei saa. Ma võin näiteks mõelda, et “praegu sajab siinsamas õues vihma”, “ma kaalun vähem kui sada kilo” või et “kaks pluss kaks on neli”. Ma võin mõelda:

- *Kui kõik inimesed on surelikud ja kui mina olen inimene, siis ma olen surelik.*

Aga ma ei saa mõelda:

- *Kui kõik inimesed on surelikud ja kui mina olen inimene, siis ma ei ole surelik.*

Samamoodi ei saa mõelda:

- *Sellest, et ma olen, järeldub, et ma ei ole,*

- *Praegu sajab siinsamas õues vihma ja ei saja vihma.*

Viimased kolm väidet on mõeldamatud ehk absurdsed. Need väited on kahtlemata olemas kui laused, mida saab lugeda ja *mille üle* saab mõelda, aga neid endid mõelda ei saa: taoliste lausete ütlemine või uskumine on küll vaimne tegevus, aga mitte mõtlemine selle sõna üldkasutatavas tähenduses. Ma ei saa tunnetada, et need laused võiksid olla tõesed.

“Mõtlemise” sõna laiemalt mõistes võib rääkida “õigest” ja “valest” mõtlemisest, kuid selle hinnaks on sõna tähenduse hägustumine: kui ma saan mõelda “A on ja A ei ole” ning “neli on seesama mis viis”, miks mitte siis juba igasugust vaimset tegevust mõtlemiseks nimetada. Kuna loogika seisukohast pole meie terminoloogiline probleem kriitiline, jätame siinkohal keelefilosoofilise arutelu katki.

1.1.2. Induktsioon ja deduktsioon

Laias laastus saab mõtlemismehhanisme jagada kahte põhirühma: üldistuste ja järelduste tegemine.

Õppimine ehk üldistuste tegemine ehk *induktsioon*

Märgates, et paljud asjad, millega me kokku puutume, esinevad kas enamasti või alati koos, üldistame selle kokkusattumuse sageli *reegliks*. Laps jätab kiiresti meelde, et küünlaleegi puudutamisele järgneb valuaisting: paarist “kokkusattumusest” on ta moodustanud enda jaoks reegli, s.t. ära õppinud, et leegi puudutamine teeb haiget. Keeleõppimine käib enamvähem samamoodi: kui ema ja isa ütlevad “söök” enamasti sellises olukorras, kus parajasti süüa pakutakse, või kui nad “söök” öeldes toidule osutavad, jätab laps mõne aja pärast meelde seose söögi enda ja sõna “söök” vahel.

Enamikul igapäevaselt õpitud reeglitel on paraku omadus erandlikes olukordades mitte kehtida: reeglitel on reeglina *erandid*. Vanemas eas õpib laps oma üllatuseks, et leegi puudutamine ei olegi alati valus: kui sõrm leegist hästi kiiresti läbi vuhistada (eriti veel siis, kui ta eelnevalt märjaks teha), ei saa üldsegi haiget. Igaüks teab, et lindudel on omadus lennata, aga mitte alati: pingviinid ja jaanalinnud reeglina ei lenda, samuti ei lenda surnud linnud. Erandina erandist lendab jäämerest kõrgele kaldale väljahüppav pingviin ikkagi paar sekundit. Ja nii edasi ja nii edasi: reeglitel on erandid, eranditel on erandid, ning viimastel omakorda erandid, kuni lõpmatuseni.

Üldistuste tegemine ehk induktsioon on seega mõtlemisprotsess, mis ei anna mingeid kindlaid teadmisi. Üldistuste edukus on *statistiline*: mida sagedamini selliselt leitud reegel kehtib, seda parem, aga ei maksa loota, et ta *alati* kehtib. Sellele vaatamata on õppimine ja üldistamine inimese ning muu elusloodu jaoks ilmselt kõige suurema tähtsusega mõtlemisprotsess üldse. Nii linnas kui metsas on kiire reageerimine olulisem kui pikk aeganõudev mõtisklemine.

Kokkuvõtteks: induktsioon kui statistilisi ning paratamatult ebakindlaid tulemusi andev mõtlemismehhanism ei kuulu formaalloogika uurimissfääri. Tasub aga meelde jätta, et ülalkasutatud sõnal “induktsioon” on ka teine tähendus “matemaatilise induktsiooni” näol. Viimane on täpne mõiste, mis annab alati õigeid resultaate: sellisena kuulub ta loogika olulisemate uurimisobjektide hulka.

Reeglite rakendamine ehk järelduste tegemine ehk *deduktsioon*

Järeldamise mõistel ja järelduste tegemisel on loogikas fundamentaalne koht: loogikat huvitav mõtlemisprotsess on reeglina suunatud lihtsatest faktidest või väidetest keerulisemate järeldamisele. Seetõttu on suur osa loogikas sagedamini kasutatavatest reeglitest esitatud järelduse vormis: ühe või mitme väite tõesusest järeldub hoopis uus väide. Loogikareeglite kasutamist uute väidete järeldamiseks nimetatakse sageli nende väidete *tuletamiseks* ehk *tõestamiseks*.

Inimene ja muu elusloodus on keeruliste järelduste tegemise jaoks palju halvemini kohastunud kui uute umbkaudsete reeglite õppimisele. Järelduste tegemine võtab palju aega, kuna enamikus olukordades on meil kasutada suur hulk reegleid ning neid reegleid saab kombineerida väga mitmel viisil. Tulemuseni püüdlemine sarnaneb labürindis ekslemisega: mida sügavamat ja keerulisemat järeldust me teha püüame, seda rohkem on teel võimalikke eksiradu, mis tulemuseni ei vii, aega aga raiskavad küll.

Erinevalt induktsioonist garanteerib õigete reeglite rakendamine õigetele faktidele alati ka õige tulemuse. Takistuseks on ainult eelnevalt mainitud suur ajakulu ning probleem, kas meie reeglid ja faktid ise on õiged, samuti, kas neid reegleid ja fakte on piisavalt palju. Valedelt faktidelt ning ekslike reeglite abil ei ole võimalik teha õigeid järeldusi. Suur osa loogikat ongi seetõttu pühendatud *kindlasti õigete reeglite* otsimisele.

1.1.3. Mõtlemise paratamatud aspektid

Loogika uurib mõtlemise *paratamatuid* aspekte ehk seda, mis üldse teeb mõtlemisest mõtlemise ehk õige mõtlemise. Enamik meie vaimsest tegevusest ilmselt ei ole paratamatu, vaid sõltub tujust, ilmast, haridusest ja muudest taolistest pooljuhuslikest teguritest. Sestap on oluline eraldada kasvõi väga väike hulk mõtlemise komponente, ilma milleta kuidagi läbi ei saa ja mis on kindlasti, alati ja igal juhul õiged. Kui me niisugused algkomponendid oleme kord eraldanud, saame neid loodetavasti kasutada üha suuremate paratamatult õige mõtlemise konstruktsioonide ehitamiseks.

Mõtlemise fundamentaalsete aspektide eraldamisega tegi teadaolevalt algust vanakreeka filosoof Aristoteles. Raamatus "Loogika" loetleb ta hulga mõtlemise kaheldamatult õigeid reegleid, mida kasutades saab veenduda küllalt keeruliste mõtlemiskonstruktsioonide õigsuses.

Küsime nüüd, millised võiksid olla need õige mõtlemise baaskomponendid. Esiteks tegeleb loogika kui teadus mõtlemise selliste meetoditega, mida saab *väidetena* kirja panna. Uurides mõtlemise fundamentaalseid aspekte, ei jõuaks me kuigi kaugele, kui ühtteist vahel selgelt kirja ei saaks panna.

Väited, mis on paratamatult õiged, on seda sageli oma ehituse tõttu. Näiteks on väide "Kui praegu sajab vihma, siis praegu sajab vihma" õige sõltumata sellest, kas praegu ka tegelikult vihma sajab või ei. "Praegu sajab vihma" asemele võiksime uuritavas väites samahästi kirjutada "Kaks pluss kaks on neli" või "Kaks pluss kaks on viis", uuritav väide jääks ikka tõeseks. Kuna meid huvitab esmajoones väidete üldine ehitus, siis kasutame nüüdsest kokkulepet, et need väite osad, mida võib suvaliselt asendada, tähistatakse suurte tähtedega, mida nimetatakse *lausemuutujateks*.

Niisiis on alati ehk tautoloogiliselt õige järgmine:

- Kui väide A on õige, siis on väide A õige

ehk lühemalt ja peaaegu ekvivalentselt

- A-st järeldub A.

Nagu öeldud, tähistab lausemuutuja A siin suvalist väidet: A sisust või õigsusest meie suurema väite õigsus ei sõltu. Viimane on õige oma ehituse ehk vormi ehk struktuuri tõttu.

Samamoodi on oma ehituse tõttu õiged

- Kui A ja B, siis A;
- Ei ole tõsi, et A ja mitte A;
- Juba vanas Kreekas tuntud järeldusreegel *modus ponens*: "Kui A-st järeldub B, ning A on tõsi, siis on ka B tõsi".

Muutujad A ja B tähistavad siin jällegi suvalisi väiteid.

Toodud näidetes moodustasime väiteid sidesõnade "ja", "ei", ning "järeldub" ehk "kui … siis" abil. Sellised sidesõnad vastavad ilmsesti mõtlemise tõeliselt fundamentaalsetele kategooriatele, ilma milleta me mõtlemist ette kujutada ei oska. Lisaks taolistele sidesõnadega märgitud kategooriatele sisaldab mõtlemine ilmselt veel hulga teisigi keeles väljendatavaid kategooriaid. Eriti olulised paistavad olevat konstruktsioonid, mida saab moodustada "kõigi", "olemasolemise" ja "omaduse" abil. Näide triviaalsest tõest (eeldusel, et mingid asjad on üldse olemas):

- Kui kõigil asjadel on omadus P, siis on olemas asi, millel on omadus P.

Viimane väide on õige konstruktsiooni tõttu, P sisust sõltumata: P võib olla omadus "olla punane kala" või omadus "kas mitte olla arv või olla viiest suurem arv", see asja ei muuda. Samas ilmselt *ei ole tõsi* väide

- Kui on olemas asi, millel on omadus P, siis on kõigil asjadel omadus P.

Vaatame järeldust kahest eeldusest:

1. eeldus: iga koer on imetaja.

2. eeldus: mõned neljajalgsed on koerad.

Järeldus: mõned neljajalgsed on imetajad.

See tuletus on õige oma konstruktsiooni tõttu, sõltumata sõnade nagu *neljajalgne*, *koer* ja *imetaja* sisust. Viimased võib välja vahetada:

1. eeldus: iga anarhist on vabaabielu pooldaja.

2. eeldus: mõned valitseva partei liikmed on anarhistid.

Järeldus: mõned valitseva partei liikmed on vabaabielu pooldajad.

Tuletuse struktuuri võib seega esitada muutujate x, y ja z abil ning tuletus on õige sõltumata fraasidest, millega neid muutujaid asendada:

1. eeldus: iga x on y.

2. eeldus: mõni z on x.

Järeldus: mõni z on y.

1.1.4. Väidete formaalne esitus

Loogikas uuritavad tuletused ei koosne enamasti ühe reegli ühekordsest rakendamisest, vaid mitme reegli mitmekordsest rakendamisest: tuletus ehk tõestus koosneb lihtsatest osadest, kuid moodustab tervikuna omaette keerulise struktuuri.

Keerulise tuletusstruktuuri esitamine inimkeelsete lausetena muudab esimese väga suureks, kohmakaks ja raskesti mõistetavaks. Loogikas on seetõttu kasutusel spetsiaalsed *formaalsed keeled*, mis on harilike keeltega võrreldes lühemalt kirjapandavad. Mis kõige tähtsam: formaalsed keeled väldivad harilikus keeles esinevaid mitmetähenduslikkusi. Üks näide: 6. sajandil e.m.a. elanud Lüüdia kuningas Kröösus uuris Delfi oraaklilt järele, kas tasub minna sõjakäigule Pärsia vastu. Oraakel ütles: “Kui alustad sõda Pärsiaga, hävitad võimsa kuningriigi”. Oraaklilt julgustust saanud Kröösus ründaski Pärsiat, kuid kaotas. Hävis nimelt tema enda kuningriik!

Lihtsustatuse tõttu on formaalsed keeled harilike keeltega võrreldes palju vaesemad ning võimaldavad edasi anda ainult väga

piiratud ampluaaga väiteid. Üks lihtsamaid formaalseid keeli on lausearvutuse keel. Viimases saab väiteid moodustada lihtväiteid tähistavatest tähtedest loogilise tähendusega sidesõnu (ja, või, ei, kui … siis) tähistavate sümbolitega (&, $\vee$, $\neg$, $\Rightarrow$).

Mainitud olulistel sidesõnadel endilgi ei ole harilikus keeles täpset tähendust. “Või” esineb sageli tähenduses “kas see või teine”. Lause “Ta on punapäine või meessoost” on harilikus keelekasutuses kohatu. Klassikalise loogika jaoks on formaalne väide $A \vee B$ õige siis ja ainult siis, kui vähemalt üks väidetest A ja B on õige; seejuures võivad ka mõlemad korraga õiged olla.

Loomuliku keele järeldussuhe erineb samuti formaalsest. Lause “Kuu peal on lehmi, järelikult olen ma kolm meetrit pikk” on loomuliku keele mõttes väär, kuna sõna “järelikult” ei sobi panna asjade vahele, mis ilmselt pole omavahel põhjuslikus seoses. Klassikalise loogika järeldussuhe $A \Rightarrow B$ on aga õige siis ja ainult siis, kui kas B on õige või A on vale: teiste sõnadega, $B \vee \neg A$. Kui kuu peal lehmi pole, siis on lause “’Kuu peal on lehmi’ $\Rightarrow$ ’ma olen kolm meetrit pikk”’ formaalselt õige, A ja B põhjuslik suhe pole seejuures oluline.

Lisaks mainitud klassikalisele loogikale on hulk mitteklassikalisi loogikaid, kus väidete tõesus ja loogikatehete nagu $\vee$ ja $\Rightarrow$ täpne tähendus on defineeritud hoopis teisiti. Mitmed mitteklassikalised loogikad püüavad tabada elementaarsete loogikatehete igapäevast tähendust, näiteks järeldussuhte põhjuslikku iseloomu.

Näiteid: Lause “Kui ’A ja B’, siis A” pannakse kirja kui $(A \,\&\, B) \Rightarrow A$. Ülalmainitud järeldusreegli saab kirja panna kui

$$((A \Rightarrow B) \,\&\, A) \Rightarrow B$$

Vaatame järgmisena olukorda, kus meil on teada, et kehtivad järgmised kolm lausearvutuse keeles esitatud väidet:

(1) A

(2) $A \Rightarrow B$

(3) $B \Rightarrow C$

Esimesest kahest saab järeldusreegliga tuletada formaalselt väite B,

ning B ja väide (3) annavad järeldusreegli abil lõpuks väite C. Viimase tuletussammu formaalseks läbiviimiseks asendasime järeldusreeglis muutujad A ja B muutujatega B ning C.

Enamik loogikaharusid kasutab lausearvutuse keele rikastatud variante, mis lubavad kirja panna märksa keerulisemaid väiteid, kui võimaldab puhas lausearvutus. Olulisem neist rikkamatest formaalsetest keeltest on *predikaatarvutuse keel*, milles saab rääkida objektidest, nende omadustest ja omavahelistest suhetest.

Keerulisemad formaalsed keeled võimaldavad väljendada veel samasust, paratamatust, võimalikkust, teadmist ja aega, suhtuda väidetesse kui objektidesse, kasutada vaikimisireegleid jne. Põhimõtteliselt on võimalik konstrueerida kuitahes keerulisi ja väljendusrikkaid formaalseid keeli, kuid fikseerimata ning pidevalt areneva loomuliku keelega võrreldes jääb mistahes formaalne keel alati piiratuks.

Kuivõrd loogika uurib mõtlemise fundamentaalseid ja abstraktseid omadusi, siis on formaalse keele piiratus loogika seisukohast kasulik. Formaalses keeles moodustatud tuletus ehk tõestus on selgepiiriline matemaatika vahenditega uuritav objekt, ning enamikku formaalset loogikat nimetatakse uurimismeetodi järgi sageli matemaatiliseks loogikaks.

1.1.5. Tõeste lausete tuletamisalgoritm

Formaalsete keelte kasutamisel loogikas pole motivatsiooniks mitte ainult nende keelte lihtsus ja tuletuste selge ning ühemõtteline struktuur. Formaalseid keeli kasutatakse ju laialdaselt ka loogikast väljaspool, näiteks arvutite programmeerimisel: kõik programmeerimiskeeled on formaalsed keeled.

Loogikas kasutatakse selliseid formaalseid keeli, mille jaoks on võimalik konstrueerida *algoritm* (s.o. selged, ühemõttelised, mehhaaniliselt järgitavad juhised) õigete lausete konstrueerimiseks, s.t. iga niisuguse keele K jaoks konstrueeritakse algoritm M, millega saab kontrollida, kas suvaline keeles K kirjutatud väide on õige või ei. Taoline algoritm esitatakse enamasti loogikareeglite koguna.

Keerulises formaalses keeles kirjutatud keerulised väited võivad väljendada ka olulisi ning üpris keerulisi probleeme. Mehhaanilise, arvutiga teostatava kontrolli võimalus võib tunduda väga ahvatlev, kuid siin tuleb arvestada, et keeruliste lausete õigsuse kontroll on väga töömahukas ülesanne ning et enamiku tõeliselt huvitavate väidete (kasvõi enamiku senilahendamata matemaatikaprobleemide) õigsuse automaatne kontrollimine ei ole tänapäeva arvutitele jõukohane. Põhimõtteliselt on automaatne kontroll küll võimalik, kuid see võib võtta arvutil aega sadu miljardeid aastaid, s.t. sageli pole mingit lootust seda tegelikult kasutada. See ei tähenda muidugi, et lootust üldse pole: mõnda liiki suhteliselt lihtsamate väidetega saab arvuti mõistliku aja jooksul hakkama.

Väidete õigsuse automaatsest kontrollimisest rääkides tuleb vahet teha *õigete* väidete õigsuse tõestamisel ja *valede* väidete mitteõigsuse tõestamisel. Esimene probleem (õigete väidete õigsuse tõestamine) on algoritmiga mehhaniseeritav, teine probleem (valede väidete mitteõigsuse tõestamine) aga pole keerulisemate formaalsete keelte (näiteks predikaatarvutuse keel) jaoks mehhaniseeritav. Viimasel juhul suudavad algoritmid osa väidete jaoks näidata, et need pole õiged, kuid nad ei suuda näidata seda *kõigi* valede väidete jaoks, vaid jäävad halvemal juhul igavesti tööle, mingitki resultaati andmata.

Niisuguse asümmeetria põhjuseks on asjaolu, et keerulisemate formaalsete keelte jaoks saab kirjutada algoritmi kõigi *tõeste väidete* tuletamiseks, mitte aga kõigi valede väidete tuletamiseks. Tõeste väidete tuletamise algoritmi skeem on üldjuhul järgmine: alustatakse lõplikust hulgast elementaarsetest baastõdedest ehk aksioomidest. Aksioomidele rakendatakse tuletusreegleid, mis kasutavad teadaolevaid ehk juba tuletatud tõeseid väiteid. Sel viisil saadakse uued tõesed väited, mida saab nüüd omakorda kasutada tuletusreeglite eeldustena. Kui meid huvitav väide V on tõene, siis tähendab see, et nimetatud väide taolise protsessi käigus ka ükskord tuletatakse. Kui V aga pole tõene, siis seda muidugi ei tuletata ning meil oleks vaja mingil viisil *tõestada*, et väidet V antud tuletusalgoritmiga tuletada ei saa. Üldjuhul pole aga või-

malik anda algoritmi, mis sellise tõestuse iga V jaoks alati leida suudaks.

Kokkuvõtteks: loogika uurib selliseid formaalseid keeli, mille jaoks suudetakse kirja panna selles keeles kirjutatud õigete väidete tuletamise algoritm. Loogika kasutatav keel käib alati paaris mehaanilise mõtlemise mehhanismiga; keelest ja tuletamismehhanismist koosnevat paari nimetatakse *teooriaks* ehk *arvutuseks*. Arvutust, mille iga lause jaoks saab algoritmiliselt lahendada, kas ta on tõene või väär, nimetatakse *lahenduvaks* (näide: lausearvutus), ülejäänuid nimetatakse *mittelahenduvaks* (näide: predikaatarvutus).

1.1.6. Lihtsatest väidetest ehitatakse keerulisi

Meie näited olid siiamaani triviaalsed ja lugejal võib tekkida kahtlus, kas selliste triviaalsuste uurimine saab öelda midagi olulist mõtlemise või üldse millegi kohta. Vastuseks ütleme, et loogika alustab teadlikult triviaalsustest ning mida triviaalsematest, seda parem: nende õigsuse suhtes ei teki kellelgi mingeid kahtlusi. Oluline on, et loogika ei jää triviaalsuste juurde pidama, vaid näitab, kuidas neist konstrueerida üha keerulisemaid ja sisukamaid väiteid. Viimaste õigsus on sageli kõike muud kui triviaalne (tegu võib olla näiteks keeruliste teoreemidega matemaatikas), aga iga üksik samm tema konstruktsioonis on triviaalselt arusaadav ja õige.

Sugulussidemed

Võtame esimeseks näitevaldkonnaks sugulussidemete kontrollimise ülesande. Et midagi kontrollida, peab meil kõigepealt olema kuskilt võtta andmebaas, kus hulga inimeste jaoks kirjas, kes on tema ema ja kes on isa. Sellise andmebaasi saab kirja panna kui suure väite stiilis “Leena Tammik on Jaan Tammiku ema ja Mihkel Tammik on Jaan Tammiku isa ja Jaan Tammik on Mart Tammiku isa ja Maia Kask on Ants Kase ema ja Mihkel Tammik on Ants Kase isa ja ...”. Predikaatarvutuse formaalses keeles märgitakse asjaolu,

et objektidel x ja y on suhe P järgmiselt: $P(x, y)$. Sugulaste andmebaas näeb predikaatarvutuse keeles välja niimoodi:

```
ema(Leena_Tammik,Jaan_Tammik) &
isa(Mihkel_Tammik,Jaan_Tammik) &
isa(Jaan_Tammik,Mart_Tammik) &
ema(Maia_Kask,Ants_Kask) &
isa(Mihkel_Tammik,Ants_Kask) &
...
```

Vanaisaks ja vendadeks olemise saab defineerida järgmiste väidetega:

- “x on y vanaisa siis ja ainult siis, kui on olemas selline z, et x on z isa ja z on y isa või z on y ema”.

  Predikaatarvutuse keeles: $\forall x \forall y$ (vanaisa(x,y) $\Leftrightarrow$ $\exists z$ (isa(x,z) & (isa(z,y) $\vee$ ema(z,y)))). Fraas $\forall x$ tähendab “iga x-i jaoks kehtib”, $\exists x$ tähendab “on olemas selline x, et ...”.

- “x on y vend siis ja ainult siis, kui kas on olemas selline z, et z on x isa ja y isa või kui on olemas selline u, et u on x ema ja y ema”.

  Predikaatarvutuse keeles: $\forall x \forall y$ (vend(x,y) $\Leftrightarrow$ $\exists z$ (isa(z,x) & isa(z,y)) $\vee$ $\exists u$ (ema(u,x) & ema(u,y))).

Äsjatoodud definitsioonides tähistavad väikesed tähed x, z, y ja u suvalisi inimesi.

Nüüd saame leida kindlasti õigeid vastuseid küsimustele sugulussidemete kohta, mis tulenevad meie andmebaasist. Küsimusele “Kas Mihkel Tammik on Mart Tammiku vanaisa?” jaatava vastuse andmiseks tuletame väite “Mihkel Tammik on Mart Tammiku vanaisa” ehk predikaatarvutuse keeles vanaisa(Mihkel_Tammik,Mart_Tammik). Tuletamise juures kasu-

tame mehhaaniliselt, ilma omapoolsete lisaarutlusteta meile etteantud andmebaasi ning loogika elementaarselt tõeseid väiteid. Selleks asendame vanaisasuhet defineerivas väites muutuja x nimega Mihkel_Tammik, muutuja y nimega Mart_Tammik ja muutuja z nimega Jaan_Tammik. Vanaisa definitsiooni kahepoolsest järeldussuhtest $\Leftrightarrow$ läheb meil tarvis ainult vasakpoolset $\Leftarrow$. Seega saame vanaisareegli meie jaoks vajaliku esialgse erikuju

```
vanaisa(Mihkel_Tammik,Mart_Tammik) ⇐
isa(Mihkel_Tammik,Jaan_Tammik) &
(isa(Jaan_Tammik,Mart_Tammik) ∨
ema(Jaan_Tammik,Mart_Tammik)).
```

Lauseosa `(isa(Jaan_Tammik,Mart_Tammik) ∨ ema(Jaan_Tammik,Mart_Tammik))` tuletamiseks piisab sellest, kui tõestada `isa(Jaan_Tammik,Mart_Tammik)`. Reegli lõpliku erikujuna kasutame järgmist:

```
vanaisa(Mihkel_Tammik,Mart_Tammik) ⇐
isa(Mihkel_Tammik,Jaan_Tammik) &
isa(Jaan_Tammik,Mart_Tammik).
```

Viimased kaks väidet on meie sugulussidemete andmebaasis olemas, ning järeldusreegliga saame seetõttu tuletada `vanaisa(Mihkel_Tammik,Mart_Tammik)`.

Võib muidugi küsida, kust peaksime teadma, et z-ks sobib nimelt Jaan_Tammik. Sellelaadsetele küsimustele vastuse leidmine on nimelt see, mis tegeliku tõestamise väga töömahukaks muudab. Üldiselt ei ole muud töökindlat varianti kui kõigi võimaluste mehhaaniline järeleproovimine, s.t. kuni tulemus käes pole, tuleb katseliselt asendada z nii Mihkel, Mart, Jaan, Leena, Maia Tammiku kui ka Ants Kasega.

Tuumajaama kontrollsüsteem

Võtame teiseks näitevaldkonnaks praktilise probleemi tuumaelektrijaamadega. Sellistes suurtes jaamades on keerukad mitmekordselt dubleeritud juhtimis- ja kaitsesüsteemid, mis peavad garanteerima,

et kunagi ei juhtuks õnnetusi. Iga niisuguse süsteemi saab kirja panna suure hulga elementaarsete väidetena. Küsimuse kaitsesüsteemide endi korrektsuse kohta saab püstitada kui küsimuse, et kas õnnetus on töökorras kaitsesüsteemi korral välistatud. Sellisele küsimusele kindlasti õige vastuse leidmine on väga oluline ja samas väga keeruline ning siin kasutataksegi sageli arvutite abi. Arvutiga püütakse tuletada väide "Juhtimissüsteemi korrektsust kinnitav väide järeldub juhtimissüsteemi kirjeldavate väidete hulgast", kasutades abivahenditena ainult elementaarseid, kindlasti tõeseid väiteid.

Mõistagi ei garanteeri kaitsesüsteemide korrektsuse tõestamine, et kaitsesüsteemi hoopiski välja ei lülitata, misjärel õnnetus võib juhtuda kõigele vaatamata. Selle vältimiseks oleks vaja kirjeldada detailselt kõigi inimeste ja ehk koguni kogu maailma olekut ja võimalikku käitumist. Kui teoreetiliselt võiks niisugune kirjeldus olla ehk isegi võimalik, siis praktiliselt on seda võimatu konstrueerida, kasutamisest rääkimata: kirjeldus oleks lootusetult suur.

Täisarvudega tegelev matemaatika

Võtame kolmandaks näitevaldkonnaks harilike täisarvudega tegeleva matemaatika. Nimetame sellist sorti matemaatikat aritmeetikaks. Aritmeetika valdkonnas defineeritakse liitmis- ja korrutamistehted ning hakatakse seejärel teoreeme tõestama. Lihtsaimad teoreemid on harilikud arvutusülesanded, nagu

- "Kas $2 * 15 = 25$?"
- "Kas $(3 + 4) * 7 = 85$?",

keerulisemad aga pärivad arvude ja tehete üldiste omaduste järele, nagu

- "Kas iga arvu x ja arvu y jaoks kehtib $x + y = y + x$?"
- "Kas algarve on lõpmatu hulk?"

- Fermat' nn. suur teoreem, mida keegi pole veel tõestada suutnud: "iga kahest suurema täisarvu x jaoks kehtib väide: ei leidu selliseid nullist suuremaid täisarve u, v ja w, et kehtiks $u^x + v^x = w^x$"

jne. jne. Aritmeetika on piiramatult keeruline: arvude kohta saab esitada lõpmatult palju küsimusi ehk tõestust ootavaid teoreeme, ja lahendatud on sellest lõpmatust hulgast vaid mõned tuhanded.

Aritmeetika alused, näiteks liitmis- ja korrutamistehted, saab selgete väidetena suhteliselt kergesti kirja panna. Samuti saab kirja panna iga huvipakkuva aritmeetikateoreemi. Teoreemi tõestamiseks tuleb teda kirjeldav väide tuletada aritmeetika aluseid kajastavate väidete hulgast, kasutades selle juures vaid elementaarselt tõeseid väiteid. Kui me niisuguse tuletuse oleme leidnud, on teoreem ilma igasuguse kahtlusevarjuta õige.

1.1.7. Aksioomid, reeglid ja mittetäielikkus

Niisiis on loogikute üks ülesanne eraldada mõtlemise kirjapandavaid baaskomponente: oma struktuuri tõttu paratamatult tõeseid väiteid ehk *aksioome* ja teadaolevalt tõestest väidetest uute väidete moodustamise elementaarseid *reegleid*.

Kõigi selliste mõtlemise baaskomponentide olulisim omadus on, et nende tõesust *ei saa tõestada*: me lihtsalt teame ehk usume, et nad on tõesed. Enamasti usume seda sellepärast, et need baaskomponendid on niivõrd lihtsad ja nende tõesus näib olevat paratamatu. Ei saa ju kahelda väites "Kui A, siis A", samas ei saa seda väidet ka kuidagi tõestada: tõestus peaks kusagilt pihta hakkama, aga kust? Tõestuste alguspunktid ning uute väidete tuletamise reeglid ongi meie ainus kindel pind, mõtlemise alus, milles kahelda ei saa.

Küsime nüüd, kui palju selliseid uskumist nõudvaid baaskomponente üldse olemas on ehk kui palju neid mõtlemise juures vaja läheb?

Vastus sõltub sellest, mis sorti valdkonnast me mõelda soovime. Võib muidugi öelda, et soovime mõelda kõigest, aga

vaatame esialgu ülalpool toodud kolme näidet: sugulussidemete andmebaas, tuumajaama kaitsesüsteem ja aritmeetika.

Esimese kahe oluliseks omaduseks on *lõplikkus*. Tõepoolest, olgu sugulaste andmebaasis või mitu miljonit nime, nende hulk on ikkagi lõplik. Samamoodi tuumajaamaga: kuitahes keeruline kaitsesüsteem ka ei oleks, sellegipoolest on tegu lõpliku hulga torude, juhtmete, andurite ja muude komponentidega, mis konkreetsel viisil kokku pandud. Niisuguste lõplike andmebaaside või süsteemide üle arutlemise jaoks on loogika *täielik*: põhimõtteliselt on võimalik kogu süsteem hulga väidetega kirjeldada ning seejärel saab väikese hulga standardsete loogikareeglitega tuletada iga õige väite nimetatud süsteemi kohta.

Erinevalt esimesest kahest on meie kolmanda näite — aritmeetika — näol tegemist lõpmatu süsteemiga: täisarve on ju lõpmatult palju. Kas lõpmatut süsteemi saab üldse kirjeldada lõpliku hulga väidetega? Teatud piirini saab. Defineerime näiteks täisarvude hulga:

- "0 on täisarv"
- "Kui x on täisarv, siis $x + 1$ on täisarv"

Definitsioonis kasutasime sümboleid 0 ja $+1$, millele me mingit aprioorset tähendust ei omista. Oluline on võimalus konstrueerida definitsiooni abil lõpmatu hulk objekte:

$$\{0, 0+1, 0+1+1, 0+1+1+1, \ldots\}$$

Aritmeetika jaoks tuleb meil defineerida täisarvude hulga järgmine fundamentaalne omadus (matemaatilise induktsiooni reegel): "Kui arvul 0 on omadus P ja kui väitest, et arvul x on omadus P, järeldub alati, et ka arvul $x + 1$ on omadus P, siis on kõigil arvudel omadus P". Induktsioonireeglit ei saa tuletada loogika baasväidetest: me võtame induktsioonireegli enda uueks täisarvude kohta kehtivaks baasväiteks. Tähendab, me *usume*, et induktsioonireegel on matemaatiliselt õige. Miks me peaksime sellise reegli tõesust uskuma? Sellepärast, et kui tema üle mõnda aega mõelda, siis saame aru, et see reegel on paratamatult õige, samamoodi

nagu me saame aru, et väide “Kui A, siis A” on paratamatult õige. Erinevalt viimasest on induktsioonireegel aga keerulisem ning tema paratamatu õigsus ei tundugi enam elementaarne. Mis juhtub, kui meil tekib vajadus samamoodi baasreeglitena uskuma hakata induktsioonireeglist veel palju keerulisemaid matemaatilisi väiteid?

Vaatame esialgu aritmeetika piiratud varianti, nn. Presburgeri aritmeetikat. Viimases defineeritakse üksainus tehe — liitmine — ja kirja on võimalik panna mitmesuguseid teoreeme liitmise kohta. Saab näidata, et niisugune ainult liitmist sisaldav aritmeetika on lõpliku hulga väidetega Θ täielikult aksiomatiseeritav, s.t. iga matemaatiliselt õige teoreem selles aritmeetikas on loogiliselt tuletatav Θ-s sisalduvatest baasväidetest, sellele vaatamata et täisarve on lõpmatult palju.

Liigume nüüd edasi piiranguteta aritmeetika juurde. Piirangute kaotamiseks piisab korrutamise lubamisest: nimelt saab liitmise ja korrutamise abil defineerida ka teised tuntud aritmeetikatehted.

Olgu meil hulk aritmeetika aluseid kirjeldavaid baasväiteid Γ. Kas iga aritmeetikateoreemi, mis *on tegelikult tõene*, saab loogikareeglitega tuletada Γ-st? Kui jah, siis on Γ aritmeetika jaoks täielik aksioomide kogu. Kui ei, siis ilmselt on Γ-st midagi vajalikku puudu.

Kolmekümnendatel aastatel tõestas Kurt Gödel enamikule kaasaja loogikutele ootamatult ühe praegu kuulsaima loogikateoreemi üldse: teoreemi mittetäielikkusest. Nimetatud teoreem näitab, et aritmeetikat ei saa taandada loogikale. Konkreetselt: ei ole olemas lõplikku baasväidete kogu Γ, millest saaks tuletada kõiki aritmeetikateoreeme. Ükskõik kui palju baasväiteid aritmeetika kohta me ka Γ-sse ei võtaks, alati leidub matemaatiliselt õigeid aritmeetikateoreeme, mida sellest Γ-st tuletada ei saa: nende jaoks tuleb hulk uusi tõeseid baasväiteid Σ lihtsalt juurde võtta. Seda saab muidugi teha: võtame uue baasväidete hulga Σ lisaks ning saame vajaliku teoreemi tuletada. Paraku aga leidub seejärel ikkagi aritmeetikateoreeme, mille jaoks ei piisa ka uuest täiendatud baasväidete hulgast. Jne. jne.: baasväidete hulk ei saa kunagi kõigi õigete aritmeetikateoreemide tuletamiseks piisavalt suureks.

Kui juba aritmeetikat ei saa lõpliku hulga baasväidetega aksiomatiseerida, siis loomulikult ei saa seda teha ka enamiku teiste matemaatikaharude jaoks. Samuti ei saa aksiomatiseerida palju muid, matemaatikast täiesti erinevaid mõtlemisvaldkondi, mis tegelevad lõpmatute struktuuridega.

See ei tähenda samas, et loogikavahendid oleksid aritmeetika või muude keerulisemate valdkondade juures kasutud: üldjuhul piisab meile huvi pakkuvate väidete tõestamiseks siiski suhteliselt väikesest hulgast harilikest elementaaraksioomidest. Juhud, kus neist ei piisa, on küll olemas, kuid praktikas väga haruldased.

Loogikateaduse uurimisvaldkonnaks ei ole mitte niivõrd keeruliste loogiliste tuletuste mehhaaniline sooritamine kui mõtlemise fundamentaalsete meetodite ja piiride uurimine. Gödeli teoreem mittetäielikkusest on üks loogikateaduse resultaatide ilusamaid näiteid.

Mida mittetäielikkus ehk aksiomatiseerimise võimatus meile ütleb? Ilmselt seda, et enamikku tõdesid (me mõtleme siinjuures ka absoluutseid, paratamatuid, matemaatilisi tõdesid) ei saa tuletada ühestki väikesest konkreetsest baasväidete hulgast. Mõtlemise jaoks ei ole kindlat lõplikku alust, millest kõik muu loogiliselt tuleneb. Mida keerulisemaid väiteid me tõestada tahame, seda suurema hulga ja keerulisemate baasväidete tõesust peame uskuma. Maailm on tõepoolest väga ebakindel: mõtlemise baas on tõestusteta uskumine.

1.2. Loogika ajalugu

1.2.1. Antiikloogika

Aristotelese-eelne periood

Keskajast pärineva legendi järgi leiutas loogika Kreeka filosoof Parmenides (5. sajand e.m.a.) Egiptuses kaljurüngaste vahel. Tegelikult Parmenides loogikat siiski ei leiutanud, s.t. arutluste struktuuri kui sellist ta ei käsitlenud. Küll aga *põhjendas* Par-

menides oma filosoofilisi vaateid pikkade arutlustega. Oletatakse, et Parmenidese väitlusmeetod pärineb kokkupuutest Pythagorase koolkonna matemaatikaga.

Parmenidese õpilane Zenon Eleast (5. sajand e.m.a.) astus oma õpetajast sammu kaugemale ja põhjendas Parmenidese vaateid kuulsate Zenoni paradokside ehk apooriatega, milles ta justkui näitas, et ruumi ja liikumist pole tegelikkuses olemas. Kuulus apooria, mis demonstreerib liikumise võimatust: te väidate, et liikumine on olemas, seega saab kõndida mööda teed punktist A punkti B. Enne punkti B jõudmist peate läbima pool teed, s.o. punkti C. Liikudes punktist C edasi B suunas peate jällegi läbima pool teed C ja B vahel, s.o. punkti D. Liikudes punktist D edasi B suunas peate jällegi läbima pool teed jne. jne. Seega jääb teil alati läbida pool mingist teelõigust ja te ei jõua kunagi punkti B. Zenoni paradokside loogiline struktuur oli väite põhjendamine väite vastandist absurdsete järelduste tuletamise teel: nn. *reductio ad absurdum*.

Väidete põhjendamise ning ümberlükkamise kunsti arendasid edasi varased retoorikud ja sofistid. Tuntumad neist olid 5. sajandil e.m.a. elanud Gorgias, Hippias, Prodicus ja Protagoras. Sofistid ei tegelnud küll väitluse tehnika uurimisega, kuid nad nõudsid näiteks, et moraalipõhimõtteid tuleb ratsionaalselt põhjendada.

Sofistide traditsiooni kandsid edasi Sokrates (470–399 e.m.a.) ja Platon (428/427–348/347 e.m.a.). Platoni kirjutistes ilmub esimest korda tähelepanek väitluse struktuurist kui iseseisvast uurimisobjektist.

Aristoteles

Loogika kui väitluse struktuuri uuriva omaette teaduse rajas Platoni õpilane Aristoteles (384–322 e.m.a.). *Sofistlike vastuväidete* lõpus kinnitab Aristoteles oma teerajaja rolli ise: “Deduktsiooni alal ei olnud varasemast olemas üldse mitte midagi.”

Aristotelese loogikaalased kirjutised koosnevad kuuest teosest koondnimetusega *Organon* (Tööriist). Aristotelese jaoks ei olnud loogika mitte üks teoreetilistest teadustest (need olid füüsika, matemaatika ja metafüüsika), vaid tööriist kõigi teaduste jaoks.

Aristotelese väljatöötatud spetsiifilist loogikasüsteemi nimetatakse sageli *mõisteloogikaks* ehk *terminite loogikaks*. Vaatleme väiteskeemi “Kui iga β on α ja iga γ on β, siis iga γ on α”. α, β ja γ on siin *muutujad*, s.t. formaalsed kohatäitjad. Iga väide, mis nimetatud struktuuri sobib, on korrektne *süllogism*. Muutujate asemele saab paigutada konkreetseid mõisteid või nimesid. Asendades α sõnaga “asi”, β sõnaga “loom” ja γ sõnaga “koer”, saame: “Kui iga loom on asi ja iga koer on loom, siis iga koer on asi”.

Aristoteles võttis loogikas kasutusele muutujad: see alusidee võimaldas luua süsteemi, mis jäi loogika vundamendiks Euroopas kuni 18. sajandini.

Aristotelese loogika käsitleb väiteid, mis koosnevad järgmistest grammatilistest komponentidest: (1) kvantor (“kõik”, “mõni”, “mitte ükski”), (2) subjekt, (3) koopula, (4) eitus, (5) predikaat. Niisugusi väiteid nimetatakse kategoorilisteks väideteks ja nad jagunevad kaheksaks liigiks:

1. “Iga β on α”.
2. “Mitte ükski β pole α”.
3. “Mõni β on α”.
4. “Mõni β ei ole α”.
5. “β on α”.
6. “β ei ole α”.
7. “x on α”, kus x tähistab konkreetset indiviidi. Näiteks: “Sokrates on surelik”.
8. “x ei ole α”, kus x tähistab konkreetset indiviidi.

Liike 1–4 peeti teistest olulisemaks. Üldjuhul ignoreeris süllogistika liike 7 ja 8. Liigid 5 ja 6 arvati ekvivalentseks liikidega 3 ja 4. Keskajal hakati liike 1–4 nimetama tähtedega: A, E, I, O.

Aristoteles defineeris *süllogismi* kui väitluse, kus mingitest etteantud väidetest (eeldustest) järeldub paratamatult uus väide. Konkreetselt vaatles ta ainult selliseid süllogisme, milles olid antud kaks eeldust, nendest tuletati üks järeldus ja nii eeldused kui järeldus olid kategoorilised väited.

Taolisi süllogisme saab moodustada neljal viisil (neli *figuuri*), millest Aristoteles käsitles ainult kolme esimest. Tõestamaks süllogistiliselt, et α on γ, tuleb leida selline β, et kehtiks üks järgmistest variantidest:

1. β on α ja γ on β.
2. α on β ja γ on β.
3. β on α ja β on γ.
4. α on β ja β on γ.

Nelja väidete liiki A, E, I, O saab nelja figuuri abil kombineerida kokku 256 viisil ja ainult 24 nendest viisidest annavad õige süllogismi: 6 tükki iga figuuri jaoks.

Keskaja skolastilises traditsioonis oli Aristotelese loogikal tähtis koht ja nimetatud 24 õige süllogismi jaoks leiutati järgmised meelespidamist hõlbustavad nimed (rühmitatud figuuride kaupa):

1. *Barbara*, *Celarent*, *Darii*, *Ferio*, *Barbari*, *Celaront*.
2. *Cesare*, *Camestres*, *Festino*, *Baroco*, *Cesaro*, *Camestrop*.
3. *Darapti*, *Disamis*, *Datisi*, *Felapton*, *Bocardo*, *Ferison*.
4. *Bramantip*, *Camenes*, *Dimaris*, *Fesapo*, *Fresison*, *Camenop*.

Täishäälikute järjekord nimes määrab kategooriliste väidete järjekorra süllogismi moodustamisviisis, eeldustega alustades ja järeldusega lõpetades.

Süllogismid *Baroco* ja *Bocardo* on tõestatavad ainult kaudselt, *reductio ad absurdum*'iga. Kõik teised süllogismid figuurides 2–4 saab taandada figuurile 1. Süllogismi esitäht ja nimes esinevad tähed *s*, *p*, *m* ja *c* määravad taandamismeetodi:

- Esitäht määrab, missugusele esimese figuuri süllogismile süllogism taandub. Näiteks *Felapton* taandub *Ferio*'le.
- Kui *s* ei ole viimane täht, siis tähendab see "Taanda väide lihtsalt", *p* vastupidi, ütleb: "Taanda väide nn. juhusliku meetodi abil".
- *S* või *p* viimaste tähtedena ütlevad: "Taanda järeldus lihtsalt" või "Taanda järeldus nn. juhusliku meetodi abil".
- *C* mitte-esitähena märgib, et süllogism ei taandu ning tuleb tõestada *reductio ad absurdum*'iga.

Süllogismi nimed jätavad ütlemata, millisesse figuuri süllogism kuulub; viimase tarvis leiutati keskajal meelespidamist hõlbustavad värsid.

Peale mainitud kategooriliste väidete huvitasid Aristotelest ka *modaalsed* väited: "α on kindlasti β" ja "α on võibolla β".

Tänapäeva kontekstis saab Aristotelese loogikat vaadelda kui predikaatarvutuse oluliselt piiratud, lahenduvat varianti. Tasub tähele panna, et Aristoteles ei tegelnud lausete loogilistest sidesõnadest (*ja*, *või*, *ei*, *järeldub*) moodustuva struktuuri — lausearvutuse — uurimisega.

Theophrastus

Pärast Aristotelest jätkas Ateena koolkonna juhtimist Theophrastus Ersesusest (371–286 e.m.a.). Kõik Theophrastuse teosed on kaduma läinud, kuid sellegipoolest on tema ning ta kaasaegse ja kolleegi Eudemuse tegevusest küllalt palju teada.

Theophrastus jätkas Aristotelese uuringuid modaalsuste alal, tõi Aristotelese süllogistikasse uue väidete tüübi ja kombineeris mõisteloogikat lausearvutuse elementidega, vaadeldes süllogisme nagu "Kui α, siis β; kui β siis γ; järelikult, kui α siis γ".

Eukleidese õpilased ja stoikud

Kreeka geomeetri Eukleidese (430–360 e.m.a.) õpilased Diodorus Cronus (4. saj. e.m.a.) ja Philon tegelesid loogiliste mõistatustega. Väidetavalt avastasid nad nn. valetaja paradoksi: “Ma ütlen, et ma praegu valetan. Kas minu väide on õige või vale?”

Sarnaselt Eukleidese õpilastega ning viimastelt mõjutusi saades tegelesid stoikud paradokside, absurdi ja vasturääkivustega. Peamiseks huviobjektiks oli Aristotelese jaoks varju jäänud lausearvutus. Stoikud uurisid, kuidas saab loogiliste sidesõnade (*ja*, *ei*, *või*, *kui* … *siis*) abil lihtsamatest lausetest keerulisemaid kokku panna ja kuidas näidata selliselt moodustatud lausete õigsust.

Erinevalt Aristotelesest, kes kasutas oma loogikas muutujatena kreeka tähti, tarvitasid stoikud muutujatena numbreid. Näiteks: “Kas esimene või teine; mitteteine; järelikult esimene”. Kuulsaim loogik stoikute seast, Chrysippus (279–206 e.m.a.), kinnitas, et järgmised viis väiteskeemi on elementaarsed ehk mittetõestatavad ning kõik õiged väiteskeemid on nendest tuletatavad (lisatud on tõlge nüüdisaegse lausearvutuse keelde):

1. Kui esimene, siis teine; esimene; järelikult teine. $((A \Rightarrow B) \,\&\, A) \Rightarrow B$.

2. Kui esimene, siis teine; mitteteine; järelikult mitteesimene. $((A \Rightarrow B) \,\&\, \neg B) \Rightarrow \neg A$.

3. Mitte korraga esimene ja teine; esimene; järelikult mitteteine. $(\neg(A \,\&\, B) \,\&\, A) \Rightarrow \neg B$.

4. Kas esimene või teine; esimene; järelikult mitteteine. $(((A \vee B) \,\&\, \neg(A \,\&\, B)) \,\&\, A) \Rightarrow \neg B$.

5. Kas esimene või teine; teine; järelikult mitteesimene. $(((A \vee B) \,\&\, \neg(A \,\&\, B)) \,\&\, B) \Rightarrow \neg A$.

Neist viiest aksioomist tuletas Chrysippus veel suure hulga õigeid väiteskeeme. Paraku on ekslik Chrysippuse kinnitus, et kõik õiged väiteskeemid on nimetatud viiest tuletatavad.

Keskaegsed ja hilisemad loogikatekstid kommenteerisid nii Aristotelese mõisteloogikat kui stoikute lausearvutust. Pärast Chrysippust kirjutati Kreekas küll mitu kommentaari Aristotelese ja Chrysippuse teostele, aktiivne loogikauurimine aga jäi soiku.

1.2.2. Keskaegne loogika

Kreeka loogikatraditsiooni kanti ladinakeelsesse maailma hulga tõlgete abil, olulisemad olid seejuures Cicero (106–43 e.m.a.), Marius Victorinuse (4. sajand) ja Martianus Capella (5. sajand) tõlked. Keskaja mõjukaimaks autoriks peetakse Boethiust (480–524/525), kes tõlkis ladina keelde Aristotelese senitõlkimata teosed ning kirjutas neile kommentaare ja lisandusi. Enne 12. sajandit Euroopas iseseisvat loogikaalast uurimistööd ei tehtud, ning kuni 12. sajandini olid Boethiuse teosed olulisim loogikaalane kirjandus Euroopas.

Araabia loogika

Stoikute ja 12. sajandil Euroopas toimunud loogika taassünni vahele jääva perioodi tähtsamad loogikaalased teosed on loodud araabia maailmas. Umbkaudu 850. aastaks olid Aristotelese ja Porphyriose peateosed tõlgitud araabia keelde ning Bagdadi koolkonnas hakati kirjutama loogika originaalteoseid. Peamised 9. sajandi autorid olid al-Kindi (805–873) ja Abu Nasr al-Farabi (870–950). Suure osa Bagdadi koolkonnast moodustasid nestoriaanlikud ja jakobiitlikud kristlased.

1050. aastaks oli Bagdadi koolkond lagunenud, üldse kahanes 11. sajandil loogika populaarsus araabia maailmas. Erandiks on Ibn Sina ehk Avicenna (980–1037). Erinevalt varasematest autoritest kirjutas Avicenna uurimusi iseseisvate originaaltekstidena, mitte kui kommentaare Aristotelese tõlgetele. Seejuures kritiseeris Avicenna Bagdadi koolkonda Aristotelese pimeda järgimise eest. Avicenna tööd jätkas teoloog al-Ghazali ehk Algazel (1058–1111).

12. sajandi olulisim araabia loogik oli Hispaanias elanud Ibn Rushd ehk Averroes (1126–1198). Bagdadi koolkonna eeskujul

kirjutas Averroes Aristotelese teoste kommentaare ning nende mõjukuse tõttu viidati Averroesele hiljem sageli lihtsalt kui Kommentaatorile. Averroese järel algas läänepoolsetes islamimaades loogika allakäik, põhjuseks islami fundamentalismi negatiivne suhtumine loogikasse ja filosoofiasse. Idapoolsetes islamimaades (näiteks Pärsias) suhtuti loogikasse al-Ghazali eeskujul pigem kui tööriista, mida muuseas saab oma filosoofiavastastes rünnakutes kasutada ka islami teoloogia. Seetõttu jätkati Pärsias käsiraamatute ja kommentaaride kirjutamist, iseseisvate loogikauurimustega aga ei tegeldud.

Loogika taastärkamine Euroopas

Canterbury Anselm (1033–1109) kasutas oma tuntud teoloogiaalastes teostes antiikajast tuntud loogikameetodeid ning rajas sellega eriti katoliikluses hiljem väga tähtsaks muutunud teoloogilise traditsiooni.

Otseselt loogikaga Anselm palju ei tegelnud, küll aga tegi seda Peter Abelard (1079–1142). Boethiusest ja Anselmist mõjutatud Abelard tõstis oma kommentaaridega Aristotelese ning Porphyriose klassikalistele tekstidele keskaegse loogikauurimise senisest palju kõrgemale tasemele.

Aristotelese muidu suhteliselt väheoluline teos *Sofistlikud vastuväited* sobis hästi 12. sajandi teoloogiasse: tegu oli väikese kataloogiga praktilises arutluskäigus tehtavatest loogikavigadest ning nende vältimisest. Suurem hulk 12. ja 13. sajandi loogikakirjandust (nn. uus loogika) on pühendatud *Sofistlike vastuväidete* edasiarendamisele ning rakendustele teoloogias ja füüsikas.

13. sajandi tuntumateks teosteks on olemasoleva loogikakirjanduse kokkuvõtted ja olulisemad autorid on Petrus Hispanus — hilisem paavst Johannes XXI — ning John duns Scotus (1266–1308). Kuulsaim autor on kahtlemata katoliikliku teoloogia alustala Aquino Thomas (1225–1274), kes vaatamata hulgale ratsionaalsetele jumalatõestustele printsiibil, et miski ei saaks olemas olla, kui poleks alguspunkti, kirjutas otseselt loogikast siiski ainult kaks vähetähtsat teost.

Keskaegse loogika hiilgeajaks peetakse 14. sajandi esimest poolt ja keskusteks Oxfordi ning Pariisi ülikooli. Esimeses kirjutas William Ockhamist (1285–1349), tuntud kui “Ockhami habemenoa” printsiibi autor, mõjuka *Summa logicae*. Samuti Oxfordiga seotud Walter Burley teos *De puritate artis logicae* oli Ockhami suhtes ülikriitiliselt meelestatud. Mõjukaim selleaegne Pariisi loogik oli Jean Buridan.

1.2.3. Loogika pärast renessanssi: 16. sajandist 19. sajandi keskpaigani

Renessanss tähendas loogika jaoks lahtiütlemist keskaegsest, otse Aristoteleselt lähtuvast skolastilisest loogikast. Hakkas sündima uus, matemaatikaga sidet leidev sümbolloogika. Viimasest jooksebki veelahe antiik- ja keskaegse ning tänapäeva loogika vahel.

Õhus olevad, siin-seal väljendatud ideed täpsest, universaalsest sümbolkeelest ning selle keele abil arutlemise matematiseerimisest ja mehhaniseerimisest ei jõudnud enne 19. sajandi keskpaiku aga mingisugustegi praktiliste tulemusteni peale lihtsate, ebasüstemaatiliste sümbolsüsteemide loomise.

16. ja 17. sajand

Renessanssi seostatakse kreeka-rooma klassikute ausse tõstmisega, kuid samas oli renessansiaegsete kirjanike hulgas kombeks põlata Aristotelesest lähtuvat, möödunud sajandite “steriilset” skolastilist loogikat. Martin Lutherit olevat ärritanud mistahes viide Aristotelesele.

Esimese mitteladinakeelse loogikateose Euroopas kirjutas 1555. aastal Petrus Ramus ehk Pierre de la Rame. *Dialectique* ja hilisem ladinakeelne *Dialecticae libri duo* ründasid skolastilist loogikat ning tegelesid kategooriliste süllogismide lihtsustamisega, lähtumata seejuures Aristotelesest. Ramuse meetoditele oli üles ehitatud Arnauld’ ja Pierre Nicole’i 1662. aastal avaldatud ning *Port-Royali loogika* nime all tuntuks saanud ja laialt kasutatud teos *La logique ou l’art de penser*. Port-Royali loogika sisaldab Blaise

Pascalist mõjustatud diskussiooni definitsioonidest, kus eristatakse *nominaalseid* ja *reaalseid* definitsioone. Filosoofiline diskussioon nominalismi ja realismi üle oli tuline juba 14. sajandil. Seejuures rõhutab Pascal matemaatiliste definitsioonide nominaalset ja pragmaatilist iseloomu.

Skolastilise loogika traditsioon siiski säilis, peamiselt muidugi katoliiklikes ülikoolides ning katoliiklikes riikides: Hispaanias ja Itaalias.

Kolmanda loogikasuuna rajajaks sai Hispaania seikleja ning müstik Ramón Lull (1235–1315). Aastal 1274 ilmunud *Ars magna, generalis et ultima* püüab esitada kontseptsioone sümbolite keeles ning tuletada väiteid variantide kombineerimise teel. Lulli ideed ning nendega kaudselt seotud kabalistlikud müüdid mõjutasid hilisemaid suurkujusid Pascali ja Leibnizit.

17. sajand oli aeg, mil tänu sümbolite ja sümbolkeele kasutuselevõtule tehti väga suuri edusamme matemaatikas. Samasuguseid sümbolsüsteeme püüti luua ka loogika jaoks.

Leibniz

17. sajandi universaalne suurkuju Gottfried Wilhelm Leibniz (1646–1716) lõi 1680. aastatel loogikasüsteemi, mis on vägagi sarnane George Boole'i süsteemiga aastast 1847. Ometigi peetakse matemaatilise ja sümbolloogika rajajaks just Boole'i, mitte Leibnizit. Üksikud erandid välja arvatud, polnud Leibnizi loogikaalastel ideedel ning avastustel järgneva kahe sajandi jooksul praktiliselt mõju.

Mõjutatuna nii Ramón Lulli ideedest kui matemaatika arengust püstitas Leibniz ülesande luua universaalne sümbolkeel (*lingua characteristica universalis*) ja seda keelt kasutav nn. arutlemise aritmeetika (*calculus rationator*), mille abil saaks algoritmiliselt või mehhaaniliselt tuletada uusi tõeseid väiteid ja kontrollida arutluste korrektsust. Leibniz oletas, et niisuguseid tuletusi ja kontrolle saaks teha spetsiaalse masinaga.

18. sajand ning 19. sajandi algus

Leibniz ei olnud ainus, kes taolisi eesmärke püstitas. Jakob Bernoulli (1654–1705), hiljem ka Gottfried Ploucquet (1716–1790) ning matemaatikud Johann Heinrich Lambert (1728–1777) ja Leonhard Euler (1707–1783) pakkusid välja sama laadi ideid. Paraku ei suutnud ei Leibniz ega ükski teine mainitutest konstrueerida loogika jaoks vähegi rahuldavat sümbolkeelt, arutlemise aritmeetikast rääkimata.

Gottfried Ploucquet ehitas Leibnizi ideedel, kuid mitte Leibnizi süsteemil baseeruva lausearvutuse sümbolsüsteemi. Ploucquet tõi sisse kvantorid "iga ..." ja "on olemas ...", tõsi küll, suhteliselt kohmakal ja piiratud viisil. Johann Heinrich Lambert konstrueeris, tõenäoliselt Leibnizist sõltumatult, Leibnizi süsteemiga sarnase sümboolse loogikasüsteemi, kus ta tõi olulise uuendusena sisse matemaatikast tuttava funktsiooni mõiste ja kasutas seda mitmekohaliste suhete (nagu näiteks "*A* on *B* isa") tähistamiseks.

Ülimõjukad Saksa filosoofid Immanuel Kant (1724–1804) ja Georg Wilhelm Friedrich Hegel (1770–1831) tegelesid muu hulgas samuti sümbolloogikaga, kuid üpris vähesel määral. *Puhta mõistuse kriitikas* viitab Kant loogikale kui lõpetatud, valmis tehtud konstruktsioonile. Kanti loengud loogikast tegelevad peaaegu ainult loogika ajalooga. Hegel kinnitab oma monumentaalses teoses *Loogika*, et senised loogikauuringud on olnud tehnilised manipulatsioonid ning asub seejärel uurima loogika sisu vastandades seda vormile. Ei Kant ega Hegel tegelnud kuigivõrd sümboolse, matemaatilise loogikaga. Sellegipoolest on Kanti ja Hegeli fundamentaalsed filosoofilised teosed puhta mõistuse analüüsidega väga olulised loogika, sh. sümbolloogika hilisema arengu jaoks.

1.2.4. Tänapäeva loogika algus

Tänapäeva loogikale panid aluse George Boole'i, Augustus de Morgani, Gottlob Frege ja teatud mõttes ka Georg Cantori tööd 19. sajandi keskel ning teisel poolel. Päris tänapäeva loogikast saab rääkida küll alles 20. sajandi 40. aastatest, alusmõisted

ja printsiibid olid aga loodud juba 19. sajandi lõpuks. Kaasaegsele loogikale pandi alus seega märgatavalt hiljem kui kaasaegsele matemaatikale.

George Boole ja Augustus de Morgan

Inglise matemaatiku George Boole'i (1815–1864) kaks peamist loogika-alast tööd on *Loogika matemaatiline analüüs* aastast 1847 ja *Mõtlemise reeglid* aastast 1854. Eriti esimene neist avaldas suurt mõju loogika järgnevale arengule. Nimelt rakendas Boole värskeid ideid matemaatilisest algebrast otse loogikale, ehitades üles *loogika algebra*, mida sageli nimetataksegi Boole'i algebraks. Kaugemaks eesmärgiks pidas Boole nagu Leibnizki loogika keele arendamist ja mõtlemise aritmeetika ehitamist. Erinevalt Leibnizist ja teistest varasematest loogikutest andis Boole süsteemse, matemaatilise kuju niisuguse keele baasfragmendile — lausearvutusele.

Nagu öeldud, annab Boole'i algebra lausearvutusele süstemaatilise, kuid mitte veel rangelt aksiomaatilise kuju. Samuti ei jõua Boole lausearvutusest kaugemale, suhteid ja omadusi kirjeldava predikaatarvutuse juurde — seda teeb 1879. aastal Frege.

Kõigi asjade klassi (nimetades seda *universumiks*) tähistas Boole numbriga 1. Tühja klassi tähistas number 0. Kõrvutiseisvad avaldised (näiteks AB) tähistasid avaldiste ühisosa ehk *ja*-tehet. $A + B$ tähistas avaldiste ühendust ehk *või*-tehet, paraku aga piirangutega: A-l ja B-l ei tohtinud olla ühisosa — kui neil oli mõni ühine element, siis oli $A+B$ defineerimata. $A-B$ tähistas B elementide eemaldamist A-st. Loogika algebra reeglid on Boole'il järgmised: $1A = A$, $0A = 0$, $A + 0 = A$, $A + 1 = 1$ (ainult juhul kui $A = 0$: Boole kartis reeglit $1 + 1 = 1$ — ühisosaga hulkade summat polnud tal ju määratletud; aga juba 1860. aastatel teisendasid Peirce ja Jevons $+$-tehte harilikuks *või*-tehteks, aktsepteerides reeglit $1 + 1 = 1$), $A + B = B + A$, $AB = BA$, $AA = A$ (kuid mitte $A + A = A$, jällegi ühisosaga hulkade summa määramatuse tõttu), $(AB)C = A(BC)$, $A(B + C) = AB + AC$ ja $A + (BC) = (A + B)(A + C)$.

Samaaegselt Boole'i esimese teosega avaldas Augustus de Morgan (1806–1871) raamatu *Formaalloogika*. De Morgani formaalne süsteem on oma sisu poolest Boole'i omaga küllalt sarnane, vormilt aga teistsugune ja kohmakas. Erinevalt Boole'ist ei olnud de Morgan ka kuigi huvitatud loogika matematiseerimisest. De Morgani käsitluse juures on olulisemad järgmised punktid:

- Kõigi asjade ehk universumi asemel kasutas de Morgan vabalt määratavat arutluse universumi mõistet (*universe of discourse*).
- De Morgan rõhutas Aristotelese loogika piiratust, järeldustehte olulisust ja järeldamise ning tinglike väidete fundamentaalset osa loogika formaliseerimisel.

Sajandi viimasel kolmandikul töötasid Boole'i ja de Morgani süsteemide arendamise ning kombineerimise kallal ameerika filosoof ja loogik Charles Sanders Peirce (1839–1914) ning saksa matemaatik Ernst Schröder.

Frege

Saksa matemaatiku Gottlob Frege (1848–1925) 1879. aastal avaldatud lühikest teost *Kontseptuaalne notatsioon* ("Begriffsschrift") võib julgelt nimetada 19. sajandi olulisemaks loogikaraamatuks. Selles raamatus esitab Frege kogu tänapäeva loogika fundamentaalseima süsteemi, *esimest järku predikaatarvutuse*. Predikaatarvutus baseerub lausearvutusel, predikaatidel ja kvantoritel "iga x jaoks kehtib ..." ning "on olemas selline x, et ...", võimaldades kirjeldada asjade omadusi ja suhteid. Teatud mõttes on võimalik Wittgensteini parafraseerides öelda, et kõike, mida saab rangelt kirjeldada, saab kirjeldada predikaatarvutuse keeles. Viimane väide ei tähenda, et predikaatarvutuse keel on alati kõige praktilisem, selgem või mugavam meetod ükskõik mille rangeks kirjeldamiseks, vaid et teoreetiliselt on iga range kirjeldus predikaatarvutuse abil kirja pandav.

Frege antud konkreetne predikaatarvutuse esitus on raskesti loetav ja hoopis teistsugune, kui praegusaegsed esitused. Põhimõtted on sellegipoolest samad. Ka ei esitanud Frege oma süsteemi

aksiomaatilisel kujul ega tuletanud kaasaegse predikaatarvutuse jaoks olulisi metateoreeme ehk süsteemi ennast, tema võimalusi ja puudujääke käsitlevaid tulemusi: nendeni jõuti alles 20. sajandi esimesel kolmandikul.

Frege tähtsus ei piirdu ainult predikaatarvutuse loomisega. Pärast 1879. aastat kirjutas ta sarja mõjukaid artikleid, alustades kõigepealt Boole'i kritiseerimisega (Frege ei olnud teadlik Peirce'i ja Schröderi parandustest ning täiendustest Boole'i algebrale). Frege kindel seisukoht oli, et kogu matemaatika saab taandada elementaarsetele loogikareeglitele, s.t. loogikareeglitega saab tuletada ükskõik millise tõese matemaatikateoreemi. Viimases ei olnud Fregel päriselt õigus, kuid tema seisukoha ümberlükkamiseni jõuti alles 1931. aastal Gödeli teoreemiga mittetäielikkusest. Vastandina Inglismaal levinud nominalistlikele ja empiristlikele ideedele toetas Frege Saksamaal tavapärasemaid realistlikke vaateid. Frege seisukoht matemaatika ja loogika vahekorrast arenes hiljem oluliseks filosoofilise loogika suunaks nimega *logitsism*.

Frege süsteemi ja logitsistlikud vaated võtsid oma töös aluseks 20. sajandi alguse mõjukaimad loogikud Bertrand Russell ja Alfred North Whitehead; seejuures ei kasutanud nad aga Frege kirjaviisi, vaid pigem Boole'i kirjaviisi edasiarendust. Frege filosoofilised ideed avaldasid hiljem olulist mõju David Hilbertile ja Ludwig Wittgensteinile ning on aktuaalsed ka praegu.

Georg Cantor

Taani päritolu ja St. Peterburgis sündinud Saksa matemaatik Georg Cantor (1845–1918) loogikaga ei tegelnud. Teda ja Richard Dedekindi peetakse *hulgateooria* rajajaks. 20. sajandil muutus Cantori hulgateooria peaaegu kogu matemaatika baasiks: nimetatud teooria olulisus seisneb *lõpmatute hulkade* käsitlemises. Cantor näitas, et lõpmatud hulgad pole sugugi kõik sama "suured" ehk ühesuguse võimsusega, vaid et lõpmatus peidab endas kirjeldamatult keerulist struktuuri erineva "suurusega" lõpmatustest.

19. sajandi viimastel aastatel märkas Cantor, et tema näiliselt selge ja vastuvaidlematu hulgateooria lubab tuletada vastuolulisi

väiteid ehk paradokse. *Cantori paradoks* on järgmine: Vaatleme *kõigi hulkade hulka* ja tähistame selle tähega M. Cantori teoreemi järgi on suvalise hulga X kõigi alamhulkade hulga võimsus (see tähendab lõpmatu hulga puhul tema "suurust") suurem kui X-i võimsus. Seega on ka M-i kõigi alamhulkade hulga võimsus suurem kui M-i võimsus. Teisest küljest aga, kuna M on kõigi hulkade hulk, peab M sisaldama elemendina iga oma alamhulka, seega ei saa M-i võimsus olla väiksem kui tema kõigi alamhulkade hulga võimsus (sest viimase kõik elemendid on ka M-i elemendid). Paradoksi põhjuseks on meie hüpotees, et eksisteerib abstraktne kõigi hulkade hulk M. Selgub, et matemaatikas võib piiranguteta abstraheerimine viia vastuoludeni.

Paradokside avastamine hulgateoorias sundis matemaatikuid suhtuma kogu matemaatika-aparatuuri kriitiliselt ja ettevaatlikult. Matemaatika aluste kriitika ja hulgateooria ise muutusid 20. sajandi loogika üheks peamiseks komponendiks ning tõukejõuks.

1.2.5. Loogika 20. sajandil

19. sajandil ei aktsepteeritud loogikat akadeemilistes ringkondades täisväärtusliku teadusena. Sajandilõpu edusammud kulmineerusid aga suurejooneliselt 1900. aasta augustikuus Pariisis järjestikku peetud esimese rahvusvahelise filosoofiakongressi ja teise rahvusvahelise matemaatikakongressiga, kus loogika, filosoofia ja matemaatika lähenemine sai uue, olulise tõuke. Matemaatikakongressil esitas David Hilbert kuulsad 23 fundamentaalprobleemi, mis suunasid loogika arengut 20. sajandi esimesel poolel. Mainitud kongresside ning Hilberti ja Russelli töö tulemusel muutus loogika akadeemiliselt aktsepteeritud distsipliiniks ja hakkas avaldama mõju nii filosoofia kui matemaatika arengule.

Sajandi esimesel kolmandikul oli loogika areng seotud peamiselt matemaatika aluste uurimisega, mille käigus kujunes kolm praeguseni olulist loogilis-filosoofilist koolkonda: *logitsism*, *formalism* ja *intuitsionism*. Gödeli ning Churchi negatiivsed resultaadid 1930. aastatel hakkasid aga vähendama matemaatikute huvi loo-

gika vastu ning praegusajal ei ole puhta matemaatikaga tegelejate seas loogika kuigivõrd tuntud või huvipakkuv ala.

Paralleelselt matemaatikute huvi vähenemisega muutus loogika üha olulisemaks analüütilistele filosoofidele — sajandi keskpaiga suurkujudest nimetame Carnapit, Łukasiewiczit, Wittgensteini ja Kripket. Nimetatud arenguga seoses hakati välja töötama mitmesuguseid uusi *mitteklassikalisi loogikaid*.

Elektronarvutite leiutamine sajandi keskel ja majanduse, teaduse ning ühiskonna süvenev arvutiseerimine andsid loogikateadusele uue võimsa tõuke. Viimaste kümnendite jooksul on loogika areng olnud üha rohkem seotud arvutiteadusega ning vastupidi. Loogika ja teoreetiline arvutiteadus on muutunud üksteisest sõltuvaks ning mitme valdkonna puhul raskesti eristatavateks. Tehisintellektiteaduse problemaatika kaudu tuleb neile kolmanda olulise komponendina juurde analüütiline filosoofia.

Logitsism: Russell ja Whitehead

Inglise filosoof ja loogik Bertrand Russell (1872–1970) oli ebaharilikult laia huvisfääriga mõtleja. Aastal 1950 pälvis Russell oma esseistika eest Nobeli auhinna, Russell oli tuntud ka kui patsifismi propageerija.

Loogika ajaloos seostub Russelli nimi filosoofi ja matemaatiku Alfred North Whiteheadiga (1861–1947): Russell ja Whitehead avaldasid aastatel 1910–1913 kolmeosalise suurteose *Principia Mathematica*, mis võttis kokku Frege, Cantori ja Peano hiljutised tulemused ning arendas neid kaugeleulatuvalt edasi. Sellest sai sajandi esimese poole mõjukaim loogikaraamat, mis on oluline loogika- ja filosoofiatekst praegugi. *Principia*'t läbiv filosoofiline liin — *logitsism* — on Leibnizist ja Fregest lähtuv ning hiljem Gödeli ümber lükatud püüdlus tuletada kogu matemaatika otse loogikast — niisugusel ühemõttelisel kujul sõnastas logitsismi teesi esimesena Russell.

Matemaatika tuletamist loogikast alustasid Russell ja Whitehead täisarvude teooriast ehk aritmeetikast. 19. sajandi lõpus postuleeris Giuseppe Peano järgmised aritmeetika baastõed, millest loodeti tuletada aritmeetika ning seejärel ehitada sinna peale matemaatiline analüüs, algebra ja muud matemaatikaharud:

1. 0 on täisarv.
2. Kui x on täisarv, siis x-ile järgnev arv $(x + 1)$ on samuti täisarv.
3. Kahel erineval täisarvul x ja y on erinevad järgnevad arvud $(x \neq y \Rightarrow (x + 1 \neq y + 1))$.
4. 0 ei järgne ühelegi arvule.
5. Matemaatilise induktsiooni printsiip. Eeldame, et mingi väide A kehtib arvu 0 kohta. Kui asjaolust, et väide A kehtib täisarvu x kohta, saab tuletada, et A kehtib ka arvu $x + 1$ kohta, siis kehtib A kõigi täisarvude kohta.

Induktsiooniprintsiibi sõnastamisel kasutas Peano mõistet *väide* ehk *omadus*, täpsustamata, mis keeles ja kuidas selliseid väiteid kirja panna. Seetõttu ongi tegemist *postulaatidega*, mitte aga range *aksiomaatikaga*. Frege väitis, et tal õnnestus Peano postulaadid range, hulgateoorial põhineva aksiomaatikana kirja panna. Russell demonstreeris vastuseks, et Frege aksiomaatika on vastuoluline, s.t. sellest saab tuletada ka valesid väiteid.

Järgnev *Russelli paradoks* sarnaneb Cantori paradoksiga, kuid on viimasest lihtsam. Moodustame kõigi selliste hulkade hulga, mis ei sisalda iseennast. Tähistame selle hulga tähega T. Küsime nüüd, kas T sisaldab iseennast. Oletame, et sisaldab $(T \in T)$. T definitsiooni järgi (T on selliste hulkade hulk, mis iseennast ei sisalda) ei saa T sel juhul iseennast sisaldada $(T \notin T)$. Saime vastuolu eeldusega, et $T \in T$. Oletame nüüd vastupidi, et T ei sisalda iseennast $(T \notin T)$. Definitsiooni järgi peaks aga T sel juhul T-d sisaldama $(T \in T)$, mis on vastuolus eeldusega $T \notin T$. Seega saime vastuolu mõlemal võimalikul juhul.

Russelli paradoksi võib vaadata kui antiikajast tuttava *valetaja paradoksi* modifikatsiooni. Russell pakkus lisaks välja järgmise modifikatsiooni. Külas on habemeajaja, kes ajab kõigi nende

külaelanike habet, kes endal ise habet ei aja. Küsime nüüd, kas habemeajaja ajab endal habet.

Russell ja Whitehead osutasid, et hulgateooria paradoksid ja vastuolud tekivad nõiaringi põhimõttel ning otsisid nõiaringi vältimise meetodit. *Principia Mathematica's* pakutud lahendus on nn. *lihtne tüüpide teooria*: igale objektile omistatakse teatud tüüp ja hulki saab moodustada ainult moodustatavast hulgast madalamat tüüpi hulkadest. Nii tekib tüüpide hierarhia. Siis ei saa ka kõigi hulkade hulka üldse moodustada.

Logitsistlike põhimõtete optimistlike kandjatena ei esitanud Russell ja Whitehead *Principia's* kordagi küsimust, mil viisil võiks nende ehitatud loogikasüsteem olla piiratud või kas on võimalik tõestada selle süsteemi mittevastuolulisust ning universaalsust.

Formalism: Hilbert

Kahekümnenda sajandi esimese kolmandiku jooksul oli David Hilbert (1862–1943) Saksamaa ehk kõige mõjukam matemaatik. Muuhulgas peetakse tema teeneks funktsionaalanalüüsi rajamist. Matemaatika ja loogika arengut mõjutas tugevasti 1900. aasta matemaatikakongressil Hilberti pakutud 23 peamise senilahendamata matemaatikaprobleemi loetelu: osa nimetatud probleemidest on praegusajaks lahendatud, osa probleemide jaoks on näidatud, et Hilberti kujutatud lahendust ei saagi olla, osa probleeme on siiani lahendamata.

Loogikuna soovis Hilbert nagu logitsistidki formaliseerida matemaatika alused range aksiomaatikana, millest saaks siis tuletada kõik matemaatikateoreemid. Erinevalt logitsistidest väitis Hilbert, et matemaatilise mõtlemise objektideks on formaliseerimisel kasutatud sümboolika ja formaalselt esitatud väited ise, mitte aga selle sümboolika kujutatav matemaatiline sisu, ja kuna matemaatika ei ole looduslik objekt, siis ei saa matemaatikast mõelda teisiti kui abstraktsioonide ehk sümbolite süsteemist.

Sajandialgust matemaatika aluste kriisi, sh. paradokside teket lõpmatuse mõiste käsitamisel püüdis Hilbert lahendada järgmisel viisil:

1. Matemaatika alused tuleb esitada loogika keeles, range aksiomaatikana.

2. Tuleb tõestada, et nimetatud aksiomaatika ei ole vastuoluline, s.t. temast ei ole võimalik tuletada korraga mingit väidet A ja sellesama väite eitust $\neg A$.

Selle nn. *Hilberti programmi* täitmise korral oleksid matemaatika aluste õigsuse probleemid mõistagi lahendatud.

Hilberti printsiibiks matemaatika aksiomatiseerimisel oli *finitism*: vastuolude puudumise tõestamise võimalikkus eeldas konstrueeritava aksiomaatika lõplikku, finiitset iseloomu, s.t. sümbolitega ei lubatud tähistada lõpmatuid objekte. Lõpmatus oli keelatud abstraktsioon: kõik tuletamissammud, kõik tõestused ja aksioomid pidid olema lõplikud. Vastasel korral ei oleks võimalik teooria mittevastuolulisust usaldusväärselt analüüsida.

1920. aastal alustas Hilbert koos silmapaistvatest loogikutest koosneva rahvusvahelise grupiga (Wilhelm Ackermann, Paul Bernays, John von Neumann ja Jacques Herbrand) nimetatud metamatemaatilise programmi täitmist. Algust tehti aritmeetika aksiomatiseerimisega Peano postulaatide baasil. 1924–1925 tõestas Ackermann, et oluline alamhulk aritmeetikast on mittevastuoluline, kuid terve aritmeetika jaoks ei suudetud tõestust leida. 1931. aastal näitas Gödel, et Hilberti programm on põhimõtteliselt teostamatu, kuid Hilbert ise ei aktsepteerinud Gödeli resultaatide sellist negatiivset tähendust kunagi.

Intuitsionism: Brouwer ja Heyting

Kolmandat matemaatikale kindla vundamendi rajanud koolkonda nimetatakse *intuitsionismiks* ehk *konstruktivismiks*. Ülalloetletud kolme koolkonna hulgast on just intuitsionismil oluline koht praegusaja loogikas, iseäranis teoreetilise arvutiteadusega seotud osades. Praegusaja matemaatikute hulgas ei ole intuitsionism seevastu kuigi populaarne.

Lühidalt ja robustselt öeldes on intuitsionismi põhiprintsiip välistatud kolmanda reegli mittetunnustamine: intuitsionist ei akt-

septeeri klassikalise loogika reeglit "A või mitte-A" ($A \vee \neg A$) ega sellega samaväärset eituse eitamise reeglit "mitte-mitte-A on sama, mis A" ($(\neg\neg A) \Leftrightarrow A$).

Hollandi matemaatikud Luitzen Egbertus Jan Brouwer (1881–1966) ja Arend Heyting (1898–1980) otsisid väljapääsu hulgateooria paradoksidest nõudmisega, et mingi matemaatilise objekti olemasolu tõestamiseks tuleb see objekt konstrueerida või näidata, mil viisil täpselt seda objekti konstrueerida saab. Näiteks ei ole teada mingit meetodit kõigi hulkade hulga konstrueerimiseks, järelikult ei ole intuitsionisti seisukohalt sellise hulga olemasolu tõestatud. Samamoodi nõuavad intuitsionistid, et "A või B" tõestamiseks tuleb tõestada eraldi kas A või B: üldisest tõestusest, et vähemalt üks neist kehtib, ei piisa. Seetõttu ei aktsepteeri intuitsionistid reeglit $A \vee \neg A$, sest üldjuhul pole ju teada, kas mingil konkreetsel juhul on tõestatav A või $\neg A$. Intuitsionistlik loogika ei erine klassikalisest loogikast mitte ainult olemasolukvantori ning *või*- ja *ja*-tehete piiratuse poolest: järgmine klassikalise loogika jaoks tõene ja ainult järeldustehet sisaldav väide (nn. Peirce'i reegel) ei ole intuitsionistlikult tõestatav: $((A \Rightarrow B) \Rightarrow A) \Rightarrow A$.

Intuitsionismi filosoofiline tuum on abstraktsete platooniliste tõdede mitteaktsepteerimine: intuitsionisti jaoks on tõde ainult see, mille jaoks on konstrueeritud tõestus. Sellest ka koolkonna nimi — *intuitsionism* tähistab intuitiivselt selget ja arusaadavat matemaatikat, vastandina näiteks Hilberti formalismile.

Harilikud loogikatehted saavad klassikalisest loogikast hoopis erineva tähenduse. Näiteks on väide $A \vee B$ klassikaliselt õige siis, kui A või B või mõlemad on õiged, ning väide $A \Rightarrow B$ on õige siis, kui kas A on vale või B on õige (või mõlemad korraga). Intuitsionistlikult on $A \vee B$ tõestatav (intuitsionistid ei räägi abstraktsest õigsusest) siis, kui kas A või B (või mõlemad) on tõestatav(ad), ning me teame, milline neist on tõestatav. Väide $A \Rightarrow B$ on tõestatav siis, kui A tõestusest saab alati konstrueerida B tõestuse. Eitus $\neg A$ on tõestatav siis, kui A tõestusest saab konstrueerida vastuolu.

Seni kuni on tegemist konkreetsete lõplike hulkadega, ei erine intuitsionistlik matemaatika klassikalisest. Vahe tekib lõpmatute hulkade, näiteks täisarvude käsitlemisel. Klassikalise loogika järgi kehtib järgmine intuitsionistlikult mittekehtiv väide: üks kahest, kas igal arvul on omadus P või on olemas arv a, millel seda omadust ei ole. Kuidas saaks suvalise omaduse P jaoks kindlaks teha, kas igal arvul on see omadus P? Ei kuidagi, sest arve on lõpmatult palju ja seetõttu ei ole põhimõtteliselt võimalik neid kõiki kontrollida.

Intuitsionistliku koolkonna alguspunktiks on L. E. J. Brouweri 1907. aastal ilmunud ja oletatavasti Kantist mõjutatud doktoridissertatsioon. Dissertatsiooni ilmumise järel võttis Brouwer ette mammutprojekti matemaatika ülesehitamiseks intuitsionistlikest printsiipidest, s.o. piirangutest lähtudes. Intuitsionistlike tõestuste leidmine on üldjuhul keerulisem kui klassikaliste tõestuste leidmine, kuid sellegipoolest suutis Brouwer neid tõestusi leida ja sel viisil näidata, et suur hulk olulist matemaatikat kehtib ka intuitsionistlikust vaatepunktist. Aastal 1918 avaldas Brouwer intuitsionistliku hulgateooria, 1919 intuitsionistliku mõõduteooria ja 1923 funktsiooniteooria.

Kuigi Brouwer rajas intuitsionistliku koolkonna ja ehitas üles suure hulga intuitsionistlikku matemaatikat, ei formaliseerinud ta *intuitsionistlikku loogikat*. Sellega sai 1930. aastal hakkama Arend Heyting. Intuitsionistlik loogika on klassikalisega väga sarnane, erinevus on (konkreetsest esitusest sõltuvalt) vaid paaris lisapiirangus ehk mõne klassikalise aksioomi keelamises.

Loogikute hulgas on intuitsionism olnud suur ja oluline koolkond Brouwerist kuni tänase päevani. Intuitsionismile annab praktilise rakendusliku väärtuse asjaolu, et mingi objekti olemasolu intuitsionistlik tõestus annab alati algoritmi selle objekti tegelikuks konstrueerimiseks. Seetõttu saab intuitsionistlikku loogikat kasutada arvutiprogrammide *automaatseks sünteesimiseks*.

Formaalne süsteem: süntaks, reeglid ja semantika

Russell ja Whitehead nimetasid *loogikaks* muuhulgas nii hulgateooriat kui endaleiutatud tüübiteooriat. Loogika tähendus oli neil küllaltki laialivalguv. Poola päritolu USA matemaatiku ja

loogiku Alfred Tarski (1902–1983) ning Saksa-Austria-USA filosoofi, loogilise positivisti Rudolf Carnapi (1891–1970) tööd tõid udusesse pilti vajalikku selgust. Kuigi loogika tähendusväli on jätkuvalt üpris avar ja ka kitsamas mõttes ei ole loogikud sugugi ühel nõul, millist formaalset süsteemi võib loogikaks nimetada ja millist mitte, ollakse enam-vähem ühel meelel, et iga formaalne loogikasüsteem peab sisaldama kolme komponenti: *süntaksit, tuletamisreegleid ja semantikat*.

- *Süntaks* on reeglite süsteem, mis määrab loogika vaadeldavate väidete *keele*, s.o. milline väide üldse on nimetatud loogika väide ja milline mitte, sõltumata tõesusest ja tuletatavusest.

- *Tuletamisreeglite süsteem* on iga loogika kõige olulisem osa. Reeglid jaotuvad eeldusteta *aksioomideks* ja juba tuletatud väidetest *uute väidete tuletamise reegliteks*. Reeglite süsteem määrab, millised väited on antud loogikas tuletatavad, s.o. “õiged”.

- *Semantika* annab formaalsetele loogikaväidetele tähenduse. Formaalse loogikasüsteemi semantika määrab kõigepealt, missugust osa maailmast väited kirjeldavad. Sageli vaadeldakse seejuures mingit lihtsat matemaatilist süsteemi, näiteks täisarve, kuna ükskõik mida muud, kasvõi inimesi, saab objektide nummerdamise teel esitada täisarvudena. Seejärel määratakse loogika elementaarväidete tõesustingimused antud maailmaosas. Lõpuks näidatakse, kuidas leida keerukate väidete tõesus komponentideks olevate elementaarväidete tõesusest. Seesugust kolmest komponendist koosnevat semantikat nimetatakse antud loogika *mudeliks*.

Tarski uuris 1920.–1930. aastatel keele, sh. loogika formaalse keele — objektkeele — ja keelest kõnelemise keele — metakeele — vahekordi, pannes sellega aluse semantika ja mudelite praegusaegsele käsitlusele. Muu hulgas näitas Tarski, et üheski formaalses keeles endas ei ole võimalik väljendada selle keele

lausete tõesust. Tõepoolest, oletame, et mingi formaalne keel võimaldab väljendada selle keele lausete tõesust. Siis saab selles keeles kirja panna väite, mille sisu on järgmine: “See lause on vale”. Jõudsime antiikajast tuttava valetaja paradoksini; nimetatud lause ei saa olla ei tõene ega väär.

Mistahes formaalse keele lausete tõesuse väljendamiseks tuleb appi võtta nimetatud keelest väljendusrikkam formaalne keel.

Aastatel 1915–1920 tõestasid Löwenheim ja Skolem nn. Löwenheimi-Skolemi teoreemi, millele ülalpool sai juba kaudselt viidatud: semantikas kasutatava maailma osana piisab alati lihtsalt täisarvude vaatlemisest. Teisisõnu: ükskõik kui keerulise ainevaldkonna kirjeldamiseks me mingit formaalset keelt kasutame, objektide nummerdamise teel saab selle valdkonna alati esitada täisarvudena. Teatavasti kasutatakse matemaatikas ja hulgateoorias täisarvude hulgast veel “suuremaid” lõpmatuid hulki; selliste lõpmatute hulkade kõiki omadusi ei saagi ühegi lõpliku keele vahenditega kirjeldada, s.t. nende struktuur võib olla sõna otseses mõttes kirjeldamatult keeruline.

Täielikkus, mittetäielikkus ja Kurt Gödel

Nagu Albert Einstein füüsikas nii on Austria-USA loogik Kurt Gödel (1906–1978) üks väheseid, keda võib ülepakkumist kartmata geeniuseks nimetada.

1930. aastal tõestas Gödel, et kaasaegse loogika baaskeel, Fregest lähtuv ja Russelli, Whiteheadi, Hilberti, Tarski, Gentzeni töödes kaasaegse kuju saanud esimest järku predikaatarvutus on *täielik*: iga tegelikult õige väide, mida saab predikaatarvutuses kirja panna, on predikaatarvutuse formaalsete reeglitega tõestatav. Siinkohal toetub õigsuse mõiste Tarski rajatud semantika ja mudelite teoorial.

Esmapilgul tundub teoreem täielikkusest olevat vastuolus järgmises lõigus vaadeldava teoreemiga formaalse aritmeetika mittetäielikkusest, kuid see vastuolu on ainult näiline: predikaatarvutuse keeles ei saa otseselt kirja panna matemaatilise induktsiooni printsiipi, mis nö. loob ehk genereerib täisarvud koos kõigi nende

omadustega. Kõike, mida üldse kirja panna saab, saab teha seda otseselt või kaudselt predikaatarvutuse vahenditega; täisarvude kõigi omaduste hulka ei saa aga üldse lõplikul viisil kirja panna.

Gödeli kõige kuulsam resultaat on varem mitmel korral mainitud mittetäielikkuse teoreem, avaldatud 1931. aastal: Peano aritmeetika postulaatide range aksiomatiseerimine annab formaalse teooria, millest ei saa tuletada kõiki tegelikult tõeseid aritmeetikaväiteid.

Tõestuse alusidee on tuntud valetaja paradoks: kas väide "Ma praegu valetan" on tõene või mitte? Lihtne arutlus näitab, et see ei saa olla kumbagi.

Koostame nüüd sellise aritmeetilise väite A, mis ütleb, et seesama A ei ole tõestatav (see väide ei ütle, et A ei ole tõsi!). Siis ei saa väide A ise olla vale. Tõepoolest, kui A oleks vale, siis A sisu kohaselt peaks A olema tõestatav. Kuna me valesid väiteid tõestada ei saa, siis peabki A olema õige. Kuna A on õige, peab kehtima see, mida A väidab: A pole tõestatav. Tõepoolest, kui A oleks tõestatav, siis oleks A sisu ("A ei ole tõestatav") vale, see on aga, nagu näidatud, võimatu. Kokkuvõtteks, A on õige, aga ei A ega A eitus pole tõestatavad.

Äsjatoodud mitteformaalne arutlus muidugi veel aritmeetika mittetäielikkust ei tõesta. Gödel kodeeris mittetäielikkuse tõestamiseks formaalse aksiomaatika aritmeetikasse. Nimelt saab kogu nimetatud formaalse süsteemi ja kõik väited esitada aritmeetika enda teoreemidena, s.t. teoreemidena täisarvude kohta. Seega õnnestub kirja panna aritmeetikateoreem A, mille sisuline tähendus formaalses süsteemis on, et seesama teoreem A ei ole aritmeetika aksiomaatikast tõestatav.

Teoreem A on tegelikult tõene, aga kasutatud formaalsest süsteemist endast teda tuletada ei saa, tarvis on lisada aksioome. Nende lisamise korral ilmnevad uued tõesed, aga mittetõestatavad teoreemid, mille tõestamiseks on tarvis jälle lisada uusi aksioome jne. jne.

Sellest näiliselt ainult aritmeetikasse puutuvast spetsiifilisest teoreemist järeldub, et ühtegi piisavalt keerulist matemaatilist süsteemi ei saa lõpliku hulga aksioomidega täielikult aksiomatiseerida. Nii ei saa lõpliku aksiomaatika abil aksiomatiseerida ühtegi lõpmatust sisaldavat, piisavalt keerulist süsteemi, olgu siis tegemist matemaatilise või matemaatikavälise süsteemiga.

Mittetäielikkuse tõestamine andis sisuliselt surmahoobi Hilberti formalistlikule ja Russelli logitsistlikule kogu matemaatika lõpliku aksiomatiseerimise programmile ning kahandas lõppkokkuvõttes matemaatikute huvi loogika vastu. Nii praktilisest kui ka filosoofilisest seisukohast tuleb mittetäielikkuse kasutamisse suhtuda siiski ettevaatlikult. Esiteks on väga suured ja huvipakkuvad alamhulgad matemaatikat ning matemaatikaväliseid süsteeme siiski lõplikult aksiomatiseeritavad, s.t. loogika ja formaalne aksiomaatika kui praktiline tööriist ei kaota sugugi oma tähtsust: lihtsalt ei saa loota, et mingi lõplik hulk aksioome suudaks kirjeldada absoluutselt kõike. Teiseks ei ole näha fundamentaalset vahet inimese ja formaalse aksiomaatika teoreetiliste võimaluste vahel: ka inimene on piiratud, nii ruumis kui ajas lõplik ega suuda samuti kõike kirjeldada ja lahendada. Inimese võimaluste kohta öeldakse vahel, et inimene suudab teha vigu ja saab just seepärast lahendada üks-kõik milliseid probleeme; täringu lisamise teel saab ka formaalse süsteemi panna vigu tegema ning selline täringuga täiendatud, juhuslikke vigu tegev süsteem suudab samuti "lahendada" mistahes probleemi. Mis puutub inimese probleemilahendamise võimetesse ja intuitsiooni, siis tuleb alati meeles pidada, et tegu ei pruugi olla jumalikest sfääridest lähtuva otsese abiga, vaid närvivõrgust paralleelprotsessorina töötava inimaju tundmatute ja teadvustamata protsessidega, mille nagu ka kivi kukkumise kohta ei saa kuidagi öelda, et tegu on millegi *a priori* loogikavälisega.

Gödeli enda filosoofilised vaated kaldusid matemaatilisse platonismi: ta suhtus täisarvudesse kui objektidesse, mis on tegelikult olemas ning millel on kindlad omadused sõltumata sellest, kas ja millises ulatuses me neid omadusi aksiomatiseerida ja tõestada suudame.

Predikaatarvutuse täielikkus ja aritmeetika ning keerulisemate süsteemide mittetäielikkus on küll kõige kuulsamad, kuid kaugeltki mitte ainsad 20. sajandi loogika fundamentaalsetest teoreemidest, mille autoriks on Gödel. Oma elu jooksul jõudis Gödel publitsee-

rida aukartustäratava hulga artikleid ning käsitleda peaaegu kõiki loogikavaldkondi.

Automaadid, programmeerimine ning lahenduvus

Tuhandeid aastaid on lisaks sõrmedele kasutatud arvutamisel abiks arvutuspulki, arvelaudu jm. Selliseid abivahendeid ei saa nimetada *arvutusmasinateks*, sest arvutamise meetodid pidid olema ikkagi kasutaja peas. Esimesed mehhaanilised arvutusmasinad leiutati Euroopas 17. sajandil, tuntumad neist on Prantsuse filosoofi Blaise Pascali liitmise-lahutamise masin ning Leibnizi masin, mis suutis ka korrutada, jagada ja ruutjuuri arvutada. 1822. a. ehitas inglane Charles Babbage esimese *programmeeritava arvuti* prototüübi: Babbage'i masinale sai kasutaja anda vabalt ette meetodeid, mida mehhaaniliselt järgides jõudis masin soovitud tulemuseni.

Esimesi programmeeritavaid elektronarvuteid hakati ehitama 1930. aastate lõpus ja 1940. alguses nii Saksamaal, Inglismaal kui ka USAs. Paar aastat varem oli loogikutel ja matemaatikutel tekkinud aktiivne huvi arvutite programmeerimise ja algoritmide ning nende üldise teooria vastu.

Teooria jaoks pole tegelikult töötada suutvat arvutit vaja, piisab abstraktselt kujutatavast arvutist, mis suudab täita etteantud operatsioone. Sellised abstraktsed arvutid ja nende programmeerimise teooria lõid teineteisest sõltumatult ameeriklane Alonzo Church (1903–1995) ja inglane Alan Turing (1912–1954) (hiljem oli Turing mõnda aega küll Churchi õpilane).

Aastatel 1935–1937 kirjutas Turing artikli, kus ta kirjeldas väga lihtsat abstraktset arvutit, nn. *Turingi masinat*. Arvuti koosnes lõputult pikast lindist (mälu), kirjutavast-lugevast peast ja seda kirjutuspead juhtivat programmi sisaldavast tabelist. Turingi veendumuse kohaselt sai Turingi masinaga arvutada kõike sedasama, mida ükskõik millise teistsuguse või suurema arvutiga: keerulisemat arvutit saab Turingi masin programmiliselt simuleerida. Tegu on niisiis *universaalse arvutiga*, nagu on universaalsed ka tegelikult eksisteerivad arvutid.

Turingi eesmärk oli uurida, mida üldse põhimõtteliselt arvutada

saab, ning mida ei saa ehk teisisõnu, missuguste lahendust omavate probleemide jaoks eksisteerib lahendamise *algoritm*. Selleks piisas Turingi veendumust mööda Turingi masina kui universaalse masina teoreetiliste võimaluste uurimisest. Turing tõestas, et tema masinaga ei ole alati võimalik otsustada, kas suvaline predikaatarvutuse keeles kirjutatud väide on õige ja seega predikaatarvutuse reeglitest mehhaaniliselt tuletatav või ei. Predikaatarvutus *ei ole lahenduv*. Mittelahenduvus laieneb predikaatarvutuselt muidugi kõigile süsteemidele, mille kaudu saab predikaatarvutust esitada. Predikaatarvutusest oluliselt lihtsamad loogikasüsteemid on sageli lahenduvad. Juba enne Turingit oli teada, et näiteks klassikaline lausearvutus on lahenduv. On olemas algoritmid, mis suudavad (kui neile piisavalt aega anda) iga lausearvutuse keeles kirjutatud väite kohta öelda, kas see väide on õige ja tuletatav või vale ega ole tuletatav.

Lahenduvuse takistuseks on väited, mis *pole tuletatavad*: ei saa olla olemas algoritmi, mis suudaks mittetuletatava predikaatarvutuse väite puhul alati õigesti otsustada, et seda väidet tuletada ei saa ja otsingu võib katki jätta. Tuletatavate väidete puhul aga suudab Turingi masin teoreetiliselt alati leida tuletuse. Muidugi võib mittetriviaalsete väidete tuletuste leidmiseks kuluda sageli rohkem aega kui universumil vanust.

1936. aastal esitas Alonzo Church minimaalsete vahenditega algoritmikirjutamis- ehk programmeerimiskeele: *lambda-arvutuse*. Lambda-arvutus erineb kardinaalselt Turingi masinast, kuid teoreetiliselt on neil samasugused arvutamisvõimed: üks on teises algoritmiliselt simuleeritav. Churchi loodud lambda-arvutus on tänapäeva *funktsionaalsete programmeerimiskeelte* eellane, ning tema teoreetiline tähtsus praegusaegses loogikas ja teoreetilises arvutiteaduses on erinevalt Turingi masina omast väga suur. Analoogiliselt Turingiga tõestas Church, et lambda-arvutuse abstraktse masinaga ei ole võimalik alati otsustada, kas suvaline formaalse aritmeetika teoreem on formaalse aritmeetika reeglitest tuletatav või mitte.

Churchi nime mainitakse kõige sagedamini seoses *Churchi teesiga*. Nimelt väitis Church, et ükskõik millise algoritmi saab

esitada lambda-arvutusega. Sellest teesist tuleneb, et ükskõik millise algoritmi saab esitada ka Turingi masinaga või iga tegelikult kasutatava programmeerimiskeelega. Tuleb tähele panna, et Churchi tees ei ole teoreem ega ka kindlasti õige. Nimelt ei ole võimalik täpselt defineerida loomulikus keeles kasutatava sõna *algoritm* sisu, näiteks ei saa me kindlad olla, et ühel hetkel ei leita mingit uut ja hoopis eriskummalist, senitundmatut ja ettekujutamatut meetodit, mida siiski algoritmiliseks võib pidada. See on küll äärmiselt ebatõenäone — siiamaani pole kellelgi tekkinud tõsist soovi või võimalust Churchi teesi kuidagi kahtluse alla seada.

Peale Churchi ja Turingi tegeles lahenduvuse problemaatikaga veel päris palju loogikuid: probleemiklasside lahenduvuse uurimine on 20. sajandi loogika klassikaline temaatika ning praegusajal töötatakse selle kallal aktiivselt. Ainult lahenduvuse probleemidega tegelevat spetsiaalset uurimisvaldkonda nimetatakse *algoritmi-* ehk *rekursiooniteooriaks*.

Üks suuremaid valdkondi loogikaga seotud lahenduvuse uuringutes on predikaatarvutuse lahenduvate ja mittelahenduvate klasside selgitamine. Nagu öeldud, pole predikaatarvutus tervikuna lahenduv. Kui aga predikaatarvutuse keelt sobivalt piirata, võime saada sellise piiratud võimalustega süsteemi, mis on lahenduv. Juba aastatel 1915–1919, s.t. enne Churchi teesi ja lahenduvuse täpse mõiste sissetoomist tõestasid Löwenheim ja Skolem, et kui lubada predikaatarvutuses ainult objektide omaduste, mitte aga suhete kirjeldamist, s.o. kasutada ainult ühekohalisi predikaate, siis selline loogikasüsteem on lahenduv.

Mis puutub loogika ja arvutiteaduse suhetesse, siis need on märksa laiemad kui algoritmi- ja rekursiooniteooria problemaatika. Viimane pakub peamist huvi loogikasiseselt ja filosoofilisest vaatepunktist, praktilise arvutiteaduse seisukohalt on lahendamatuse eri astmeid ja lahendamatuse struktuuri uurivad algoritmiteooria sfäärid suhteliselt vähem huvitavad. Iga probleemiklassi lahendatavuse/mittelahendatavuse probleem ise on arvutiteaduse jaoks siiski väga oluline, sest lahendatavuse tõestamine annab "kõrvalproduktina" enamasti kaasa lahendusalgoritmi enda.

Loogika ja analüütiline filosoofia

Analüütiline filosoofia tekkis sajandivahetuse Inglismaal reaktsioonina seni domineerinud, Hegelilt pärit idealismile. Alusepanijateks peetakse filosoofe Bertrand Russelli ja George Edward Moore'i (1873–1958). Russell nimetas oma vaateid *loogiliseks atomismiks* ning need vaated kujunesid koostöös Wittgensteiniga viimase varasemal tegevusperioodil.

Tänapäeva analüütilise filosoofia kõige radikaalsemaks ja ehk kõige mõjukamaks suunaks kujunes Hume'ist ja Russelli-Whiteheadi loogikast inspireeritud loogilise positivismi ehk loogilise empirismi koolkond. Vormiliselt pandi loogilise positivismi koolkonnale alus 1926. aastal Moritz Schlicki seminarides Viini ülikoolis ning koolkond püsis Viini ringi nimetuse all kuni 1938. aastani. Viini ringi olulise liikmena võib nimetada Austria-USA semantikut ja filosoofi Rudolf Carnapit (1891–1970).

Loogilise positivismi peamiseks tulemuseks on filosoofia kui sellise rolli muutumine. Loogilised positivistid nõudsid, et filosoofia peab olema teaduslik ja keeruliste maailmapiltide asemel tuleb produtseerida selget mõtlemist. Kuigi suurem osa filosoofiat seepärast veel eriti teaduslikuks ei muutunud, hakkasid ometigi tekkima teaduslikkust oluliseks pidavad koolkonnad. Filosoofia taustsüsteem nihkus.

Viini ringiga tihedalt seotud Austria-Inglise filosoof Ludwig Wittgenstein (1889–1951) kirjutas oma *Loogilis-filosoofilises traktaadis* (selle traktaadi uurimine oli Viini ringi algusaastate üks põhitegevusi): "Filosoofia eesmärk on mõtlemise loogiline selgitamine. Filosoofia on tegevus, mitte teooria. Filosoofiline teos koosneb peamiselt selguse toomisest. Filosoofia tulemus pole mitte hulk filosoofilisi väiteid, vaid väidete selgus."

Loogiliste positivistide olulisemad teesid kannavad endas Russelli-Whiteheadi logitsistliku programmi vaimu (praegusajal on muidugi kerge nimetada seda programmi naiivoptimistlikuks) ja üldistavad selle universaalseks eesmärgiks mujalgi peale matemaatika:

1. Mõtestatud tekst koosneb kas (a) loogika ja matemaatika formaalsetest väidetest või (b) konkreetsete teadusharude fakte esitavatest lausetest.

2. Igasugusel fakti esitaval väitel on sisu ainult siis, kui on võimalik öelda, kuidas selle väite kehtivust kontrollida.

3. Metafüüsilised väited, mis ei lange punktide 1 ja 2 alla, on sisutud.

4. Kõik moraali, esteetikat ja religiooni käsitlevad väited on mittekontrollitavad ja mõttetud.

Niisiis oli loogiliste positivistide võtmeküsimuseks väidete kontrollitavus ehk *verifitseerimine*. Ajapikku selgus, et usaldusväärne verifitseerimine pole üldjuhul võimalik ei loodusteadustes ega sageli ka mitte matemaatikas (Gödeli resultaadid mittetäielikkusest), ning loogiline positivism kaotas oma aktuaalse (aga mitte ajaloolise) tähenduse.

Loogiliste positivistide kolm põhihuvi olid loogika, keel ja taju. Russell alustas loogikaga, jätkas tajuprobleemidega ja lõpetas semantikauurijana; Carnap alustas tajust, jätkas semantikaga ja lõpetas loogikuna. Kesksel kohal neist kolmest probleemiringist seisis keel ja keelefilosoofia.

Protsessi sajandivahetuse logitsistlikust optimismist neljakümnendate aastate nö. küpsema suhtumiseni illustreerib hästi Ludwig Wittgensteini vaadete areng. Wittgensteini vaated ja looming jaotuvad selgelt kaheks perioodiks, kus põhiküsimused jäävad samaks, kuid neile lähenemise teed, metoodika ja lootused muutuvad kardinaalselt.

Mitmekülgselt andekas Wittgenstein alustas teadlasekarjääri 1908. aastal Manchesteri ülikoolis aeronautika üliõpilasena, hakkas aga varsti huvi tundma matemaatika aluste vastu. Russelli 1903. aastal ilmunud raamat *Matemaatika põhimõtted* avaldas talle sügavat mõju ja 1911. aastal läks Wittgenstein Cambridge'i, kus tal õnnestus hakata õppima Russelli juures. Hiljem erakuna Norras töötades ja Esimese maailmasõja ajal suurtükiväe

ohvitserina rindel teenides kirjutas Wittgenstein märkmeid, millest sai kokku 1921. aastal avaldatud lühike (75 lehekülge), aga äärmiselt mõjukas *Loogilis-filosoofiline traktaat* (*Tractatus Logico-Philosophicus*). *Tractatus*'e ja kogu Wittgensteini loomingu põhiteema on keele ja väljendatavuse piirid: kuidas on võimalik keele olemasolu; kuidas on võimalik teistele midagi *öelda*; miks nad öeldust *aru saavad*? Wittgenstein rõhutab *Tractatus*'es keele piiratust ("On asju, millest ei saa rääkida"), kuid suhtub samas ülemäära optimistlikult loogika keele võimalustesse. Keelte paljus, väitis Wittgenstein, viib eksiteele; tarvis on otsida fundamentaalset, täpset, ühist nimetajat. Keel ja öeldavus piirneb *Tractatus*'es enamvähem sellega, mida saab öelda *loogika formaalses keeles*. Maailm on analoogiliselt logitsistide arusaamaga, *a priori* korrastatud.

Tractatus'e kirjutamise järel lõpetas Wittgenstein ajutiselt filosoofiaga tegelemise, töötas aastaid kolkaküla kooliõpetajana, seejärel aedniku ja arhitektina. 1929. aastal pöördus Wittgenstein tagasi Cambridge'i, seekord õppejõuna, ning alustas filosoofiauuringuid, nn. hilise Wittgensteini perioodi. Wittgensteini hilist perioodi kokkuvõttev teos *Philosophische Unterschungen* (*Philosophical Investigations*) ilmus 1953. aastal, pärast Wittgensteini surma.

Investigations võtab maailma korrastatuse ning keele ja mõtlemise täpsuse suhtes *Tractatus*'ega vastupidise hoiaku. Keele ja mõtlemise mitmekesisuse taga ei ole universaalset ühist nimetajat. Keel on praktiline tööriist, mis tekib ja kannab tähendust ainult tegelikus kommunikatsiooniprotsessis. Wittgensteini lemmiknäide on sõna *mäng*. Ta veenab lugejat, et mänguks nimetatavatel tegevustel pole ühist tuuma, pole mingit loogilist põhjust, miks neid kõiki just mängudeks nimetatakse, teisisõnu, loomuliku keele sõnu ei saa defineerida. Sõnade kasutus ongi sõnade tähendus. Defineerida ei saa ka selliseid filosoofia jaoks fundamentaalseid sõnu nagu *teadmine*, *väide*, *reegel*, *põhjus* jne. Mingi sõnaga tähistatavate objektide ühise essentsi asemel tuleb rääkida nende perekondlikust sarnasusest. "Mitteräägitavaid" asju pole.

Wittgensteini varase perioodi ideoloogia ühtib oma põhiloomult selleaegse loogika ideestiku ja ühe universaalse formalismi

ning aksiomaatika ootustega, mille abil saaks kõike kirjeldada ning aksiomatiseerida. Wittgensteini hiline periood haakub Gödeli negatiivsete tulemuste järel saabunud murranguga loogikas: maailma lihtsa loogilise aluse lootus oli kadunud. Varane Wittgenstein sobis logitsistide ning loogiliste positivistide programmi, hiline enam mitte: Bertrand Russell hindas kõrgelt *Tractatus*'t, kuid suhtus negatiivselt *Investigations*'isse.

Logitsistika ja loogilise positivismi allakäigu järel jätkusid loogika ja filosoofia vastasmõjud, kuid ilma seniste ekstsessideta. Palju fundamentaalseid loogikaprobleeme ja teoreeme pakub iseseisvat filosoofilist huvi. Otsesed kokkupuutepunktid on loogikal ja analüütilisel filosoofial endiselt keelefilosoofia pinnal, kus loogika oma kaugelearendatud teooriaga semantikast ja mudelitest esindab matemaatiliselt täpset äärmust, andes filosoofiale seega kindla punkti, millele vajaduse korral toetuda. Ühist huvi pakuvad *mitteklassikalised loogikad* (lisaks intuitsionistlikule on neist olulisemad modaalne, relevantne, lineaarne, episteemiline ja mittemonotoonne loogika), mis formaliseerivad spetsiaalset tüüpi arutlusi.

Näiteks formaliseerib modaalloogika arutlusi, kus kasutatakse *võimalikkuse* ja *paratamatuse* mõisteid. Paratamatu on näiteks väide "$2 + 2 = 4$", võimalik aga väide "Prantsusmaa on vabariik". Võime ette kujutada maailma, kus viimane väide ei kehti (neid on ajaloos juba olnud), kuid ei suuda ette kujutada maailma, kus esimene väide ei kehti. 1950. aastate lõpul ehitas modaalsele loogikale range *võimalike maailmade semantika* Ameerika semantik Saul Kripke. Richard Montague kasutas modaalset loogikat loomuliku keele semantika analüüsiks (Montague grammatikad), katoliiklik loogik Bochenski aga religiooniprobleemide analüüsiks.

Analüütilise tõesuse (s.o. loogiline tõesus laiemas mõttes) küsimused on üks olulisemaid filosoofilisi küsimusi, mille uurimise juures on loogikal tähtis koht. Siinkohal tuleb mainida Willard Van Orman Quine'i (s. 1908) ja nn. *funktsionalistliku koolkonna* rajanud ja sellest hiljem lahti ütelnud Hilary Putnami (s. 1926) nime, kes tegelevad nii puhta loogika kui ka filosoofiaga.

Quine'i ja Putnami seisukohalt ei ole analüütilistel ja empiirilistel väidetel mingit fundamentaalset vahet. Analüütilisteks

nimetatakse väiteid, mille tõesus või mittetõesus tuleneb loogiliselt definitsioonidest, nagu väide "Onupoeg on meessoost" ja matemaatikateoreemid. Empiiriliste väidete tõeväärtus sõltub ümbritsevast maailmast, tüüpiliseks näiteks on siin füüsika- ja muude loodusteaduste väited. Quine püüab seejuures näidata, et teadus saab põhimõtteliselt hakkama puhtalt loogikal baseeruva keelega ning *intensionaalseid määratlusi*, nagu *tähendus* ja *paratamatu õigsus* pole teaduskeeles vaja kasutada.

Loogikaga on seotud ka Ameerika lingvisti Avram Noam Chomsky (s. 1928) tööd ning kaudselt kogu strukturaalne lingvistika.

Mahukaima koostööpinnase loogikale ja filosoofiale on viimaste kümnendite jooksul andnud tehisintellektiteadus ja kognitiivsusteadus (*cognitive science*). Neis valdkondades uuritakse informatsiooni esitamise ja vaimse tegevuse viise päris praktilisel eesmärgil, ehitamaks arvutiprogramme, mis suudaksid lahendada keerulisi, intellekti nõudvaid ülesandeid. Selle uurimistöö käigus areneb samas ettekujutus inimese enda vaimsetest protsessidest. Loogika jaoks on siin fundamentaalseks probleemiks igapäevaste, tavaliselt ebakindlate ja umbmääraste teadmiste esitamine ning arutlemise juures kasutamine, introspektsioon, kommunikatsiooniprobleemid ja eksitustest ning väärinformatsioonist tekkinud segadustest ülesaamine.

I

Klassikaline loogika

2. Loogika põhimõisted

Armastus ja loogika on kaks asja, mis hoiavad maailma koos.
— Linnar Priimägi, “Ars et vita”, Eesti Televisioon, 1995

2.1. Põhilised loogikaseadused

Iga loogilise teooria aluseks on põhitõed. Niisuguseid baastõdesid nimetatakse loogikaseadusteks.

Samasusseadus on üks tähtsamaid formaalloogilisi seadusi:

- Mitte ühegi lause sisu ei muutu arutluse käigus.

$$p \supset p.$$

S.t. lause on alati ja igal pool iseendaga identne.

Ilma samasusseaduseta ei ole arutluste õigsust praktiliselt võimalik kontrollida.

Vasturääkivusseadus väidab, et

- Ükski lause ei saa olla iseendaga vastuolus.

$$\neg(p \,\&\, \neg p).$$

S.t. lause ei saa olla korraga tõene ja väär.

Me ei saa lubada, et arutlus sisaldab väiteid, mis on omavahel vastuolus.

Kui soovitakse mingit mõttekäiku kritiseerida, siis viidatakse tavaliselt selles mõttekäigus esinevatele väidetele, mis on vastuolus üldtuntud tõdedega või mõttekäigu osadega.

Vasturääkivusi sisaldav arutlus ei saa olla korrektne. Vasturääkivusseadus ei luba näiteks sellist olukorda, kus samaaegselt sajab vihma ega saja vihma.

Mõned vanakreeka filosoofid väitsid, et tõeseid ja vääraid lauseid ei ole üldse olemas. Aristoteles sõnastas neile vastukaaluks ***välistatud kolmanda seaduse***:

- Iga lause on kas tõene või väär, kolmandat võimalust ei ole.

$$p \vee \neg p.$$

  S.t. Ei saa olla olukorda, kus ei lause ega selle eitus pole tõene.

Neid väärtusi, mida laused võivad omandada, nimetatakse *tõeväärtusteks*. Välistatud kolmanda seadus väidab, et lausetel on ainult kaks tõeväärtust: *tõene* või *väär*. Sellisel juhul on tegemist klassikalise kahevalentse loogikaga.

Välistatud kolmanda seadus võimaldab vaatluse alt välja jätta sellised arutluste osad, mis ei ole ei tõesed ega väärad. Näiteks võib tuua hüüd- ja küsilaused:

Milline ilm on täna?

Tore!

Välistatud kolmanda seadust kasutatakse sageli vastuväitelisel tõestamisel: me eeldame, et kehtib vaadeldava väite eitus ja tuletame sellest vastuolu; järelikult peab väide olema tõene.

Küllaldase aluse seaduse formuleeris saksa filosoof Gottfried Wilhelm Leibniz:

- Ühtki lauset ei saa pidada tõeseks ega vääraks ilma küllaldase aluseta.

Küllaldase aluse seadust kasutatakse näiteks kohtupraktikas. Me ei saa öelda, et inimene on mingis kuritöös süüdi, kui meil ei ole ühtegi seda kinnitavat fakti. Inimese õigeksmõistmiseks ei ole aga vaja mingeid tõendeid. Selles mõttes ei kasutata küllaldase aluse seadust kohtutes sümmeetriliselt.

Küllaldase aluse seadus nõuab oma väidete põhjendamist. Näiteks matemaatikud peavad põhjendatuks ainult tõestatud väiteid. Küllaldase aluse seadus on mingil määral vastuolus välistatud kolmanda seadusega, kuna vastuväiteline (kaudne) tõestus ei tarvitse väidet küllaldaselt põhjendada.

2.2. Vasturääkivus ja mittevasturääkivus

Loogika uurib lauseid (väiteid) ja nendevahelisi seoseid. Lausete hulga korral huvitab meid küsimus, kas väited on omavahel kooskõlas või mitte.

Definitsioon 2.1. Lausete hulk on *mittevasturääkiv* ehk kooskõlaline $\Leftrightarrow$ selle hulga kõik laused saavad olla korraga tõesed.[1]

Vastasel juhul nimetatakse lausete hulka *vasturääkivaks* ehk vastuoluliseks.

Vaatame järgmist lausete hulka:

Igaüks, kellele Mari naeratab, on Marist sisse võetud.

Mari ei naerata.

Jüri on Marisse armunud.

Need laused saavad olla samaaegselt tõesed. Kujutame näiteks ette, et Jüri on Marisse kõrvuni armunud. Mari on aga tõsine, sest tal on ka teine kavaler ja ta ei tea veel, kumma kasuks otsustada.

[1] Sümbolit $\Leftrightarrow$ loetakse “siis ja ainult siis, kui” või “parajasti siis, kui”. Selle sümboli kasutamine muudab uute mõistete defineerimise mugavamaks.

Kui aga Mari lõpuks Jürile naeratab, siis on selge, et Jüri on veel rohkem Marist sisse võetud.

Kui me aga lisame vaadeldavale lausete hulgale lause

Mari naeratab Jürile,

siis on tulemuseks vasturääkiv lausete hulk. See ei ole ju võimalik, et Mari on samaaegselt tõsine ja naeratab Jürile. Ka laused

Eesti ja Venemaa piiriks on Tartu rahuga määratud piir;

Eesti ja Venemaa piiriks on riikidevaheline ajutine kontrolljoon

on vasturääkivad, kuna nad on vastuolus samasusseadusega.

2.3. Arutlus ja järeldus

Suulises ja kirjalikus tekstis on tihti lõigud, kus midagi põhjendatakse. Meil on vaja selgitada oma põhimõtteid teistele või ka lihtsalt iseendale. Selline selgitus sisaldab peale selgitatava printsiibi ka eeldusi, millele me selgituses otse või kaudselt tugineme. Näiteks saab jäätise söömist põhjendada väitega

Ma pean jäätist sööma, sest jäätise söömine mõjub hästi minu näonahale.

Eraldades selgitusest kõik *järelduse* põhjendamiseks vajalikud laused, saame abstraktse konstruktsiooni, mida nimetatakse *arutluseks* e. mõttekäiguks.

Definitsioon 2.2. *Arutlus* (või *argument* või *tõestus*) on lausete hulk, milles üks lause on valitud kui *järeldus* ja ülejäänud laused on selle lause *eeldusteks*.

Vaatleme tekstilõiku

Kui Maril on hea tuju, siis on Jüri õnnelik. Maril on täna hea tuju. Järelikult on Jüri täna oma õnne tipus.

See on tüüpiline tõestus, kus kaks esimest lauset põhjendavad viimast lauset. Järelduse äratundmist lihtsustab *võtmesõna järelikult* kasutamine.

Järelduse kindlakstegemine ei tarvitse alati olla nii lihtne. Ent kui järeldus on leitud, siis saab arutluse esitada *standardkujul*, milleks on tabel, kus joone kohal on eeldused ja joone all järeldus. Meie näitearutluse üks võimalikest standardkujudest on selline:

Maril on täna hea tuju.

Kui Maril on hea tuju, siis on Jüri õnnelik.

Jüri on täna oma õnne tipus.

Ülesanded

1. Leidke raamatutest ja ajalehtedest arutlusi ning märkige neis ära eeldused ja järeldus. Püüdke hinnata, kas tegemist on deduktiivsete või induktiivsete järeldustega. Kas argumenteerivat mõtlemisvormi kasutatakse sageli?

2.4. Arutluste kehtivus ja korrektsus

Kuidas otsustada, kas arutluse eeldused põhjendavad järeldust või mitte?

Definitsioon 2.3. Arutlus *kehtib* $\Leftrightarrow$ ei ole võimalik, et tema eeldused on tõesed ja järeldus on väär.

Tähistades arutluse eelduste loetelu tähega Γ ja järelduse tähega p, paneme arutluse kehtivuse lühidalt kirja kujul

$$\Gamma \models p$$

ning ütleme, et eelduste hulgast Γ *järeldub loogiliselt* lause p.

Eelmises punktis toodud arutlus kehtib. Selles veendumiseks oletame, et vaadeldava arutluse eeldused on tõesed. Siis on Maril täna hea tuju. Teisest eeldusest järeldub, et Jüri on täna õnnelik. See aga tähendab, et järeldus ei saa olla väär.

Vasturääkivate eeldustega arutlus on alati kehtiv, sest pole võimalik, et tema eeldused on korraga tõesed. Vasturääkivad on näiteks arutluse

Tallinn on Eesti Vabariigi pealinn.

Tallinn ei ole Eesti Vabariigi pealinn.

Tartu asub Soome lahe põhjakaldal.

eeldused. Järelikult see arutlus kehtib.

Ülesanded

1. *** Selgitage, miks me ei sõnastanud arutluse kehtivuse definitsiooni kujul

   - *Arutlus kehtib ⇔ kui arutluse eeldused on tõesed, siis on ka tema järeldus tõene.*

Arutluse kehtivus väljendab eelduste ja järelduse loogilist seost. Arutluse tegeliku paikapidavuse hindamiseks kasutatakse korrektsuse mõistet.

Definitsioon 2.4. Arutlus on *korrektne* ⇔ ta on kehtiv ja kõik tema eeldused on tõesed.

Korrektse arutluse järeldus on tõene. Näiteks arutlus

Tallinn on Eesti Vabariigi pealinn.

Eesti Vabariigi pealinn asub Eesti põhjaosas.

Tallinn asub Eesti põhjaosas.

on korrektne, sest tema mõlemad eelduslaused on tõesed. Järgmine arutlus ei ole aga korrektne, kuigi kõik tema väited on tõesed:

Tallinn on Eesti Vabariigi pealinn.

Eesti Vabariigi pealinn asub Eesti põhjaosas.

Tartu ei ole Eesti pealinn.

Tallinn asub Tartust põhja pool.

See, et Tartu ei ole Eesti pealinn, ei põhjenda piisavalt seda, et Tallinn asub Tartust põhja pool.

2.5. Lausete loogiline tõesus ja väärus

Üksikuid lauseid võib jagada tõesteks, vääradeks ja kontingentseteks.

Definitsioon 2.5.

- Lause on *loogiliselt tõene* (e. samaselt tõene) ⇔ ei ole võimalik, et lause on väär.
- Lause on *loogiliselt väär* (e. samaselt väär) ⇔ ei ole võimalik, et lause on tõene.
- Kõiki ülejäänud lauseid nimetatakse *kontingentseteks*.

Loogikuid huvitavad kõige rohkem loogiliselt tõesed laused, sest need on vankumatud tõed. On selge, et loogiliselt tõese lause eitus on alati loogiliselt väär. Kehtib ka vastupidine väide: loogiliselt väära lause eitus on loogiliselt tõene. Enamus lauseid on siiski kontingentsed ja nende tõesus sõltub kontekstist. Vaatleme lauseid

Elu on elu.

Tööpäev kestab reedel kella poole viieni.

Jüri on ja ei ole mees.

Esimene lause on samasusseaduse põhjal loogiliselt tõene (kuna ta ei saa olla väär). Teine lause on kontingentne, sest ta võib olla mingis kontekstis tõene, mingis teises kontekstis aga väär. Viimane lause on aga vasturääkivusseaduse põhjal loogiliselt väär.

2.6. Lausete ekvivalentsus

Kahe lause korral saab rääkida nende samaväärsusest.

Definitsioon 2.6. Kaks lauset on *loogiliselt ekvivalentsed* $\Leftrightarrow$ ei ole võimalik, et üks lausetest on tõene ja teine väär.

Loogiliselt ekvivalentsed on näiteks laused:

Päike paistab ja linnud laulavad.

Linnud laulavad ja päike paistab.

Järgmiste lausete korral võib ette kujutada olukorda, kus üks lausetest on tõene ja teine väär:

Karu magab talveund.

Karule meeldib magada.

Näiteks võib karu magada vastu tahtmist. Järelikult ei saa neid lauseid lugeda loogiliselt ekvivalentseteks.

Teiselt poolt on aga kõik loogiliselt tõesed laused omavahel ekvivalentsed.

2.7. Matemaatilised seosed

Näitame, et lausetevahelised tähtsamad loogilised seosed saab avaldada vasturääkivuse omaduse kaudu.

Tähistagu

$$\Gamma \models p$$

lause p loogilist järeldumist lausete hulgast Γ,

$$p \models q$$

lause q loogilist järeldumist lausest p,

$$\models p,$$

lause p loogilist tõesust,

$$p = q$$

lausete p ja q loogilist ekvivalentsust ja sümbol

$$\Leftrightarrow$$

väljendit "parajasti siis, kui".

Lause 2.7. *Lausete hulgast Γ järeldub loogiliselt lause $p \Leftrightarrow$ hulk $\Gamma \cup \{\neg p\}$ on vasturääkiv.*

Tõestus. $\Rightarrow$. $\Gamma \models p$ tähendab, et kui Γ laused on tõesed, siis ei saa p olla väär. Järelikult ei saa kõik hulga $\Gamma \cup \{\neg p\}$ laused olla tõesed, sest kui Γ laused on tõesed, siis on $\neg p$ väär.

$\Leftarrow$. Oletame, et hulk $\Gamma \cup \{\neg p\}$ on vasturääkiv. Kui kõik Γ laused on tõesed, siis ei saa $\neg p$ olla tõene, s.t. p ei saa olla väär. Seega kehtib $\Gamma \models p$. □

Lause 2.8.

a) Laused p ja q on loogiliselt ekvivalentsed $\Leftrightarrow$ $p \models q$ ja $q \models p$.

b) Laused p ja q on loogiliselt ekvivalentsed $\Leftrightarrow$ hulgad $\{p, \neg q\}$ ja $\{\neg p, q\}$ on vasturääkivad.

Tõestus. Punkt b järeldub eelmise lause põhjal punktist a. Tõestame punkti a.

$\Rightarrow$. Olgu laused p ja q loogiliselt ekvivalentsed: $p = q$. See tähendab, et nende lausete tõeväärtused ei saa olla erinevad. Kui üks lausetest p ja q on tõene, siis ei saa teine lause olla väär. Järelikult $p \models q$ ja $q \models p$.

$\Leftarrow$. Olgu $p \models q$ ja $q \models p$. See tähendab, et kui p on tõene, siis ei saa q olla väär ja vastupidi. Järelikult ei saa lausete p ja q tõeväärtused olla erinevad ning kehtib $p = q$. □

Lause 2.9. *Lause p on loogiliselt tõene $\Leftrightarrow$ hulk $\{\neg p\}$ on vasturääkiv. Lause p on loogiliselt väär $\Leftrightarrow$ hulk $\{p\}$ on vasturääkiv. Lause p on kontingentne $\Leftrightarrow$ mitte kumbki hulkadest $\{p\}$ ja $\{\neg p\}$ ei ole vasturääkiv.*

Tõestus. Tõestame esimese lause. Ülejäänud tõestused jätame lugejale (ülesanded 2.7.1 ja 2.7.2).

$\Rightarrow$. Olgu $\models p$. Seega ei saa p olla väär ning $\neg p$ ei saa olla tõene. Järelikult on hulk $\{\neg p\}$ vasturääkiv.

$\Leftarrow$. Olgu hulk $\{\neg p\}$ vasturääkiv. Siis ei saa $\neg p$ olla tõene ning p ei saa olla väär. Järelikult kehtib $\models p$. □

Analoogiliselt saab näidata, et lausete vahelised põhilised loogilised seosed saab taandada teatavate keerukamate lausete loogilisele tõesusele. Selleks on aga vaja eelnevalt sisse tuua lausearvutuse tehted.

Ülesanded

1. Tõestage lauses 2.9 väide loogilise vääruse kohta.
2. Tõestage lauses 2.9 väide kontingentsuse kohta.
3. Tõestage, et vasturääkivast lausete hulgast saab moodustada kehtiva arutluse, valides järelduseks suvalise selles esineva lause eituse.

2.8. Ülesanded

1. Viige tekstilõigud arutluse standardkujule:
   (a) *Ei saa olla tühjust; sest see, mis on tühi, pole midagi, ja mis pole midagi, see ei saa olla.* (Melissus)
   (b) *Esimese inimese sünnitasid mingid teised olendid; sest teised olendid hakkavad ennast varsti ise hooldama ning ainult inimene vajab pikaajalist hooldamist. Seega poleks esimene inimene ellu jäänud, kui see oleks olnud tema esialgne eluvorm.* (Anaximenes)

3. Lausearvutus

> *Milli laused on nõtked, tõelised süntaksiimed, nad on plastilised, neisse näib mahtuvat kogu inimkogemus, kuid ka sellel osal, mis ei mahu, pole väljaspool sõnaringi halb olla.*
>
> — Toomas Raudam

Arutlusi on võimalik mitmeti analüüsida. Kõige lihtsamaks mooduseks on arutluskäigu jaotamine propositsioonideks e. *lauseteks*. Propositsioonidest saab omakorda eraldada osalauseid. Niisugusel analüüsil põhinebki *lausearvutus*, mida nimetatakse vahel ka lauseloogikaks või Boole'i loogikaks. Inglise loogik George Boole (1815–1864) kirjeldas eelmisel sajandil lausearvutuse (ehk Boole'i algebra) matemaatilisi omadusi. Sellega viis ta arutluste analüüsi kvalitatiivselt uuele tasemele.

Kõigi inimarutluste analüüsimiseks ei piisa tavaliselt klassikalisest lausearvutusest ning appi tuleb võtta kas predikaatarvutus või mitteklassikaline loogika. Lausearvutuse tehted on aga leidnud laialdast kasutamist arvutite konstrueerimisel ja tarkvara loomisel.

3.1. Lausete analüüs

Me käsitleme *lauset* e. propositsiooni kui otsustuse (väite, mõtte) väljendusvormi. Iga lause on kas tõene või väär.

Tavaliselt saab arutlust jaotada lihtsamateks lauseteks, mille omavahelised seosed määravad arutluse kehtivuse. Näiteks arutlus

Jüri õpib täna või ta kukub homme eksamil läbi.
Jüri ei õpi täna.

Jüri kukub homme eksamil läbi.

koosneb lausetest

1. *Jüri õpib täna.*
2. *Jüri kukub homme eksamil läbi.*
3. *Jüri ei õpi täna.*
4. *Jüri õpib täna või ta kukub homme eksamil läbi.*

Nende lausete võrdlemisel selgub, et kolmas lause on esimese lause **eitus**. Neljas lause on aga saadud kahe esimese lause ühendamisel sidesõnaga **või**. Nii moodustuvad lausete vahelised *loogilised seosed*.

Neist lausetest saab analoogiliselt moodustada ka teisi seoseid:

*Jüri kukub läbi, **sest** ta ei õpi.*
*Jüri ei õpi, **kuid** ta ei kuku läbi.*
*Kui Jüri õpib, **siis** ta ei kuku läbi.*
*Jüri kukub läbi, **kui** ta ei õpi.*
*Jüri õpib **ja** ta kukub läbi.*
*Jüri kukub läbi **isegi siis, kui** ta õpib.*

Niisugused seosed on üldjuhul loogilised. Lausearvutuse seoste korral määravad osalausete tõeväärtused *täielikult* kogu lause tõeväärtuse, osalausete konkreetne sisu ei ole aga tähtis. Sellisel juhul ütlevad matemaatikud, et lause tõeväärtus on tema osalausete tõeväärtuste *funktsioon*.

Definitsioon 3.1. *Lausearvutuse tehteks* nimetatakse niisugust lausetes kasutatavat seost, mille tõeväärtus on tema osalausete tõeväärtuste funktsioon (ehk *Boole'i funktsioon*).

Vaatleme näiteks lauseid

*Jüri õpib **või** Jüri vaatab televiisorit.*
*Jüri õpib **ja** Jüri vaatab televiisorit.*
*Jüri **ei** õpi.*

Tähistades tõeväärtuse *tõene* tähega t ja tõeväärtuse *väär* tähega v saame nendes lausetes kasutatavate seoste väärtused erinevatel osalausete tõeväärtustel esitada tabelitena:

ja $(t, t) = t$	**või** $(t, t) = t$	**ei** $(t) = v$
ja $(t, v) = v$	**või** $(t, v) = t$	**ei** $(v) = t$
ja $(v, t) = v$	**või** $(v, t) = t$	
ja $(v, v) = v$	**või** $(v, v) = v$	

Matemaatilises mõttes on seoste tabelid Boole'i funktsioonide graafikud. Seose *ja* põhjal moodustatud lause on tõene ainult siis, kui tema mõlemad osalaused on tõesed, s.t. Jüri õpib ja vaatab televiisorit korraga. Seos *või* on aga tõene, kui vähemalt üks tema osalausetest on tõene. Seos *ei* muudab tõese lause vääraks ja vastupidi.

Lausetes esinevad tihti ka seosed, mida ei saa kirjeldada Boole'i funktsioonidega. Selliste seoste korral ei ole võimalik selgitada, kuidas osalausete tõeväärtused määravad kogu lause tõeväärtuse. Me ei oska näiteks määrata lause

*Neljapäeval oli maavärin, **sest** Kuu ja Päike olid ühel joonel*

tõeväärtust, kuna on raske otsustada, kas lause teine pool põhjendab piisavalt tema esimest poolt või mitte.

3.2. Lausearvutuse tehted

3.2.1. Konjunktsioon

Seost *ja* nimetatakse lauseloogilisel kasutamisel *konjunktsiooniks*. Konjunktsioonil on osalausete samadel tõeväärtustel alati sama tõeväärtus. Sõltuvalt kontekstist tähistatakse konjunktsiooni erinevate sümbolitega. Levinumad tähistused on

$p \,\&\, q$ (ampersand),

$p \wedge q$,

$p \,.\, q$ (punkt),

p *and* q,

kus p ja q on suvalised (osa)laused.

Konjunktsiooni sisaldavad näiteks laused

Jüri õpib ja Mari õpib.

Jüri õpib. Mari õpib.

Jüri ja Mari õpivad.

Jüri õpib, kuid ka Mari õpib.

Antud juhtudel nimetatakse *konjunktsiooniks* nii lauses esinevat loogilist seost kui ka lauset ennast. Konjunktsiooni sisaldava lause saab alati ümber sõnastada, kasutades seost *nii ... kui ka*:

Nii Jüri õpib kui ka Mari õpib.

Laused

Jüri ja Mari õpivad koos.

1 ja 1 annavad kokku 2

ei sisalda aga konjunktsiooni, sest neid lauseid ei saa selliselt ümber sõnastada.

Lausearvutuse seoseid on mugav kirjeldada *tõeväärtustabelite* abil, mis meenutavad jälle funktsioonide graafikuid:

Jüri õpib	*Mari õpib*	*Jüri õpib* & *Mari õpib*
t	*t*	*t*
t	*v*	*v*
v	*t*	*v*
v	*v*	*v*

3.2.2. Eitus

Sõnad *ei*, *pole* ja *mitte* viitavad lauses harilikult *eituseseosele*. Eitus esineb näiteks lausetes

Jüri ei õpi täna.

Jüri pole mitte midagi õppinud.

Ma ei saa mitte midagi aru.

Eitust tähistatakse tavaliselt sümbolitega

$\neg p$,

$\sim p$ (tilde),

$\overline{p}$ (ülakriips),

not p

ja tema tõeväärtustabel näeb välja niimoodi:

p	$\neg p$
t	*v*
v	*t*

3.2.3. Disjunktsioon

Disjunktsiooni tunneb lausetes ära sidesõnade *või* ja *ehk* järgi. Disjunktsioon esineb näiteks lauses

Jüri õpib või Jüri vaatab televiisorit.

Disjunktsiooni sisaldava lause saab ümber sõnastada väitena kujul

Kas Jüri õpib või Jüri vaatab televiisorit või ta teeb mõlemat tegevust korraga.

Disjunktsiooni tähistatakse sümbolitega

$$p \vee q,$$

$$p \text{ or } q$$

ning ta on on väär ainult siis, kui tema mõlemad osalaused on väärad:

p	q	$p \vee q$
t	t	t
t	v	t
v	t	t
v	v	v

Sõna *või* kasutatakse tekstis ka välistavas tähenduses. Näiteks lauses

Sa sööd kõik ära või sa saad karistada

peetakse silmas ainult ühe mainitud tegevuse võimalikkust.

Välistav disjunktsioon (ehk mitteekvivalents) on väär ka juhul, kui tema mõlemad osalaused on tõesed:

p	q	$p \vee q$
t	t	v
t	v	t
v	t	t
v	v	v

ja seda tähistatakse arvutiprogrammides

$$p \text{ xor } q. \text{ (exclusive or)}$$

Välistava disjunktsiooni tehet kasutatakse näiteks arvutigraafikas esemete liikumise kujutamiseks.

Välistava disjunktsiooni tõeväärtustabeli saab konstrueerida ka hariliku disjunktsiooni, konjunktsiooni ja eituse tehete abil:

S	K	$S \vee K$	$S \,\&\, K$	$\neg(S \,\&\, K)$	$(S \vee K) \,\&\, \neg(S \,\&\, K)$
t	t	t	t	v	v
t	v	t	v	t	t
v	t	t	v	t	t
v	v	v	v	t	v

Järelikult on laused

$$S \text{ xor } K$$

ja

$$(S \vee K) \,\&\, \neg(S \,\&\, K)$$

lausearvutuse seisukohast sama tähendusega ning nende Boole'i funktsioonid langevad kokku:

$$S \text{ xor } K = (S \vee K) \,\&\, \neg(S \,\&\, K).$$

Tõlkides need valemid tagasi eesti keelde, saame aga laused

*Tõsi on **nii** see, et sa sööd kõik ära **või** sa saad karistada, **kui ka** see, et **pole** õige, et sa sööd kõik ära **ja** sind karistatakse.*

*On kaks võimalust: sa **kas** sööd kõik ära **või** sa saad karistada.*

3.2.4. Implikatsioon

Implikatsioon väidab, et üks tema osalausetest on tema teise osalause eeltingimuseks. *Implikatsioonitehte* tunneb lausetest ära väljendi *kui ... siis* või sellega samaväärse seose järgi. Implikatsioon esineb näiteks lauses

Kui praegu on kevad, siis vahetult enne seda oli talv.

Selle lause saab eesti keeles kirjutada ka lihtsamal kujul:

Kevadele eelneb talv.

Implikatsiooni tähistatakse harilikult sümbolitega

$p \supset q$ (hobuseraud),

$p \to q$ (nool),

$p \Rightarrow q$ (järeldumine),

if p *then* q.

Implikatsioon on väär ainult siis, kui tema *eeldus* on tõene ja *järeldus* on väär:

p	q	$p \supset q$
t	t	t
t	v	v
v	t	t
v	v	t

Implikatsiooni korral tuleb tähele panna, et ta on tõene ka juhul, kui tema eeldus on väär. Vaadeldes meie näitelauset paneme tähele, et see on tõene ka siis, kui parasjagu ei ole kevad. Ent asendades selles "talve" "sügisega", saame lause

Kui praegu on kevad, siis vahetult enne seda oli sügis,

mis väära eelduse korral on väär.

Eesti keeles kasutatakse sõnapaari *kui ... siis* sageli ka teistes tähendustes. Näiteks lauses

Kui Toomas teeb hoolsalt trenni, siis Peeter tegeleb rohkem ajaviitmisega

on tegemist pigem konjunktsiooniga. Me ei ole vahel mingi seose

implikatiivse tõlgendamisega nõus ka sellepärast, et selle seose eeldus on väär. Mõned inimesed loevad näiteks lauset

Juhul kui Pariis on Inglismaa pealinn, olen mina Eesti president

vääraks, kuna nad ei saa kunagi Eesti presidendiks. Niisugune tingimuslik seos ei tarvitse aga olla lausearvutuses *väljendatav*, s.t. sellisele seosele ei tarvitse vastata mitte ükski Boole'i funktsioon.

Implikatsiooni saab avaldada eelmiste lausearvutuse tehete kaudu kas kujul

$$p \supset q = \neg p \vee q$$

või

$$p \supset q = \neg(p \,\&\, \neg q).$$

Viimane valem väidab, et ei ole võimalik situatsioon, kus implikatsiooni eeldus on tõene ja järeldus on väär.

3.2.5. Ekvivalents

Ekvivalentsiseost kasutatakse loomulikus keeles vähe, matemaatikas aga laialdaselt. Ekvivalentsiseosele viitavad tekstis väljendid

- *siis ja ainult siis, kui*
- *parajasti siis, kui*

või lihtsalt mõlemapoolset järeldumist tähistav sümbol $\Leftrightarrow$. Ekvivalentsi saab kasutada mõistete defineerimiseks. Tüüpilise ekvivalentsiga on tegemist näiteks definitsioonis

Ristkülik on ruut $\Leftrightarrow$ *tema küljed on võrdsed.*

Definitsioonides ei kasutata tavaliselt ekvivalentsiseose keelelisi vasteid, vaid lugeja peab ise taipama, millal on tegemist definitsiooniga, millal mitte. Näiteks lauset

Ruut on ristkülik, mille küljed on võrdsed

võib vaadelda kui ruudu definitsiooni. Matemaatikud eelistavad kasutada *definitsioonides* kokkulepitud keelekonstruktsioone:

*Me **ütleme**, et ristkülik on ruut, kui tema küljed on võrdsed.*

*Ruuduks **nimetatakse** ristkülikut, mille küljed on võrdsed.*

Tähtis on see, et me ei defineeriks mõistet liiga laialt ega liiga kitsalt. Näiteks ei sobi antud juhul sõnastus

Ruut on nelinurk, mille küljed on võrdsed,

sest selle järgi oleks ka romb ruut.

Ekvivalentsi tähistatakse sümbolitega

$p \equiv q$ (ekvivalents),

$p \sim q$ (sarnasus),

$p \leftrightarrow q$,

$p \Leftrightarrow q$ (vastastikune järeldumine),

p *iff* q (*if and only if*)

ning ta on tõene ainult siis, kui tema osalausetel on samad tõeväärtused:

p	q	$p \equiv q$
t	t	t
t	v	v
v	t	v
v	v	t

Ekvivalentsi saab avaldada nii implikatsiooni kui ka konjunktsiooni, disjunktsiooni ja eituse kaudu:

$$p \equiv q = (p \supset q)\ \&\ (q \supset p),$$
$$p \equiv q = (p\ \&\ q) \vee (\neg p\ \&\ \neg q),$$

s.t. laused on ekvivalentsed, kui nad järelduvad vastastikku teineteisest, või nad on samaaegselt tõesed või samaaegselt väärad.

Ülesanded

1. Tähistagu

   A – *Sajab vihma* E – *On suvi*

   B – *Sajab lund* F – *On talv*

   C – *Sajab rahet* G – *On külm*

   D – *Sajab* H – *On soe*

   Kirjutada valemite abil:

   (a) *Sajab vihma ja külmetab.*

   (b) *Sajab vihma või lund ja on külm.*

   (c) *Sajab vihma, kuid ei saja lund.*

   (d) *Kui külmetab, siis sajab lund.*

   (e) *Ei saja vihma ega lund.*

   (f) *Kui sajab lund, siis on talv.*

   (g) *Iga sadu on vihmasadu, lumesadu või rahesadu.*

   (h) *Kui sajab vihma, siis ei saja lund ja on suvi.*

   (i) *Kui on külm või sajab lund, siis ei saja vihma.*

   (j) *Suvel ei saja korraga vihma ja rahet.*

   (k) *Suvel sajab vihma või rahet, talvel lund.*

2. Lugeda, kasutades eelmise ülesande tähistusi:

   (a) $E \vee F \supset D$;

   (b) $D \supset B\ \&\ F \vee A\ \&\ E$;

   (c) $E\ \&\ G\ \&\ D \supset C$;

   (d) $E\ \&\ B \supset G$;

   (e) $F\ \&\ H\ \&\ D \supset A$.

3. Näidake, et implikatsiooni saab avaldada eituse ja konjunktsiooni kaudu:

   (a) $p \supset q = \neg p \vee q$;

   (b) $p \supset q = \neg(p\ \&\ \neg q)$.

4. Näidake, et ekvivalentsi saab avaldada kujul
   (a) $p \equiv q = (p \supset q)\ \&\ (q \supset p)$;
   (b) $p \equiv q = (p\ \&\ q) \vee (\neg p\ \&\ \neg q)$.
5. ***Matemaatikud nimetavad väite $p \Leftrightarrow q$ tõestamisel tõestuse $p \Rightarrow q$ osa *tarvilikkuseks* ja tõestuse $p \Leftarrow q$ osa *piisavuseks*. Selgitage, miks on tõestuse osadel sellised nimed.

3.3. Lausearvutuse süntaks ja semantika

Loogikud on omavahel kokku leppinud, kuidas loogikavalemeid kirjutada. Selleks määratakse valemites kasutatavad sümbolid ja valemite õigekirjareeglid. Sellist kirjeldust nimetatakse keele *süntaksiks*. Täpselt samuti toimitakse loomulike keelte korral. Eesti keele jaoks on näiteks "Õigekeelsussõnaraamatus" fikseeritud tähestik, sõnavara ja grammatika. Lausearvutuse *tähestiku* moodustavad kolme tüüpi sümbolid:

- lausemuutujate sümbolid:
  $A, B, C, \ldots, A_1, A_2, \ldots$ (suured tähed)
- loogiliste tehete sümbolid: $\neg\ \&\ \vee\ \supset\ \equiv$
- kirjavahemärgid: ()

Anname järgnevalt lausearvutuse valemite e. *lausete* induktiivse definitsiooni. Selles defineeritakse kõigepealt atomaarsed laused, mis koosnevad ainult ühest lausemuutujast ja seejärel defineeritakse atomaarsete lausete baasil liitlaused, milles on sümboleid rohkem kui üks.

Definitsioon 3.2 (lausearvutuse süntaks). Defineerime *lausearvutuse valemid* e. *laused* induktiivselt:

1. *Atomaarne valem* e. *aatom* koosneb ainult ühest lausemuutujast. Iga atomaarne valem on lausearvutuse valem.

2. Kui p on lausearvutuse valem, siis $\neg p$ on lausearvutuse valem.
3. Kui p ja q on lausearvutuse valemid, siis $(p\ \&\ q)$, $(p \vee q)$, $(p \supset q)$ ja $(p \equiv q)$ on lausearvutuse valemid.
4. Lausearvutuse valemi välimised sulud võib ära jätta.
5. Muid lausearvutuse valemeid ei ole.

Sageli lisatakse lausearvutuse tähestikku ka *loogilised konstandid* t (tõene) ja v (väär) ning tunnistatakse neid kui atomaarseid valemeid. Kõiki vaadeldava valemi konstrueerimise käigus tekkivaid valemeid nimetatakse aga selle valemi *alamvalemiteks* (e. *osavalemiteks*). Viimasena valemisse paigutatud tehet nimetatakse valemi *peatehteks*.

Näide 3.3. Järgmine formaalne lause on lausearvutuse valem:

$$(A \vee B)\ \&\ \neg(A\ \&\ B).$$

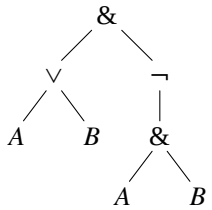
Selles veendumiseks näitame definitsiooni punktide kaupa, kuidas see valem on konstrueeritud.

1. A on atomaarne valem (definitsiooni 1. punkti põhjal).
2. B on atomaarne valem (1).
3. $(A \vee B)$ on valem (3), sest A on valem ja B on valem.
4. $(A\ \&\ B)$ on valem (3), sest A ja B on valemid.
5. $\neg(A\ \&\ B)$ on valem (2), sest $(A\ \&\ B)$ on valem.
6. $((A \vee B)\ \&\ \neg(A\ \&\ B))$ on valem (3), sest $(A \vee B)$ on valem ja $\neg(A\ \&\ B)$ on valem.
7. $(A \vee B)\ \&\ \neg(A\ \&\ B)$ on valem (4), sest valemi välimised sulud võib ära jätta.

Selle valemi peatehteks on (sulgudeväline) konjunktsioon ja tema alamvalemid on

$$A,\ B,\ A \vee B,\ A \,\&\, B,\ \neg(A \,\&\, B),$$
$$(A \vee B) \,\&\, \neg(A \,\&\, B).$$

Valemi ehitust kirjeldab hästi süntaksipuu:



Ülesanded

1. *Kirjeldage mõnda eesti keele süntaksireeglit.

Loogiliste tehete *prioriteedid* võimaldavad vähendada valemites kasutatavate sulgude arvu. Kõige kõrgema prioriteediga on eitusetehe ja kõige madalama prioriteediga ekvivalentsitehe. Teiste tehete prioriteet fikseeritakse tavaliselt nii, et see kahaneb järgmises tehete loetelus vasakult paremale:

$$\neg \;\&\; \vee \;\supset \equiv$$

Tehete prioriteetide abil saab taastada valemi puuduvad sulud. Selleks vaadatakse iga loogilise tehte korral valem vasakult paremale läbi ja taastatakse vajaduse korral selle tehte sulud. Sulgude taastamist alustatakse kõige kõrgema prioriteediga tehtest ja korratakse seda protsessi iga järgneva tehtega uuesti. Lause

$$A \,\&\, \neg B \supset A \vee C$$

korral saadakse näiteks tulemuseks süntaktiliselt korrektne lausearvutuse valem

$$(A \,\&\, \neg B) \supset (A \vee C).$$

Valemit

$$A \,\&\, B \vee A \,\&\, C$$

tõlgendatakse aga valemina

$$(A \,\&\, B) \vee (A \,\&\, C).$$

Assotsiatiivsusreegel ütleb, et konjunktsiooni korral annavad sulgude erinevad paigutused samaväärse tulemuse:

$$((A \,\&\, B) \,\&\, C) = (A \,\&\, (B \,\&\, C)).$$

Sellepärast võib üksteisele järgnevate konjunktsioonide korral sulud ära jätta:

$$A \,\&\, B \,\&\, C.$$

Puuduvate sulgude taastamiseks kasutatakse kas konjunktsiooni vasakpoolse assotsiatiivsuse reeglit või parempoolse assotsiatiivsuse reeglit. *Vasakpoolse assotsiatiivsuse reegli* kasutamisel on kõige kõrgema prioriteediga vasakpoolseim konjunktsioon:

$$(A \,\&\, B) \,\&\, C,$$

parempoolse assotsiatiivsuse reegel annab aga tulemuseks lause

$$A \,\&\, (B \,\&\, C).$$

Assotsiatiivsusreegel kehtib küll konjunktsiooni ja disjunktsiooni korral, kuid ta ei kehti implikatsiooni korral:

$$((A \supset B) \supset C) \neq (A \supset (B \supset C)).$$

Seepärast on avaldise

$$A \supset B \supset C$$

kasutamisel tarvis eelnevalt kokku leppida, kumba assotsiatiivsusreeglit me kasutame.

Lausearvutuse süntaks andis lausearvutuse valemite õigekirjareeglid. *Semantikareeglid* ütlevad aga, kuidas alamvalemite tõeväärtuste põhjal määrata kogu valemi tõeväärtust. Antud juhul tähendab semantika sisuliselt lausearvutuse tehete tõeväärtustabelite sõnalist selgitamist.

Definitsioon 3.4 (lausearvutuse semantika). Olgu p ja q lausearvutuse valemid.

1. Iga lausemuutuja väärtuseks on kas tõeväärtus *tõene* (t) või tõeväärtus *väär* (v).

2. Valem $\neg p$ on tõene $\Leftrightarrow$ p on väär.

3. $p \,\&\, q$ on tõene $\Leftrightarrow$ p ja q on mõlemad tõesed.

   $p \vee q$ on tõene $\Leftrightarrow$ vähemalt üks valemitest p ja q on tõene.

   $p \supset q$ on tõene $\Leftrightarrow$ p on väär või q on tõene.

   $p \equiv q$ on tõene $\Leftrightarrow$ p ja q tõeväärtused on samad.

Süntaksit ja semantikat ei tohi omavahel segi ajada. Vaatame näiteks võrdust

$$A = \neg\neg A.$$

Semantiliselt see võrdus kehtib: kui A on tõene, siis $\neg A$ on väär ja $\neg\neg A$ on tõene; A vääruse korral saame analoogiliselt, et $\neg\neg A$ on väär. Järelikult on need valemid loogiliselt ekvivalentsed. Süntaktiliselt kehtib aga mittevõrdus

$$A \neq \neg\neg A,$$

sest need valemid koosnevad erinevatest sümbolitest.

Ülesanded

2. Leidke valemi peatehe ja loetlege kõik alamvalemid:
   (a) $\neg A \,\&\, H$;
   (b) $\neg(A \,\&\, H)$;
   (c) $K \supset (\neg K \supset K)$.
3. Tõestage assotsiatiivsusreeglid:
   (a) $(A \,\&\, B) \,\&\, C = A \,\&\, (B \,\&\, C)$;
   (b) $(A \vee B) \vee C = A \vee (B \vee C)$;
   (c) $^*(A \supset B) \supset C \neq A \supset (B \supset C)$.

3.4. Lausevormid ja arutlusvormid

Loogikas eristatakse *objektkeelt* ja *metakeelt*. Selles peatükis selgitatakse näiteks eesti keeles, mis on lausearvutus. Antud juhul

on objektkeeleks lausearvutuse keel, metakeeleks aga eesti keel. Kui me räägime eesti keeles eesti keelest, siis langevad objektkeel ja metakeel kokku. Kõige lihtsam moodus mõlema keele lausete eristamiseks on erineva tähestiku kasutamine. Kuna lausearvutuse valemites puuduvad väiketähed, siis viitab iga valemites esinev väiketäht sellele, et me selgitame parasjagu eesti keeles lausearvutuse üldiseid omadusi.

Loogikas kasutatakse tihti väljendeid

- Iga valem kujul $p \vee \neg(p \,\&\, q)$ on loogiliselt tõene.

Valem $p \vee \neg(p \,\&\, q)$ sisaldab väiketähti p ja q, mis ei kuulu lausearvutuse tähestikku. Sarnaselt lausemuutujatega võib tähti p ja q nimetada *metamuutujateks*. Väiketähti sisaldavaid metakeelseid lauseid nimetatakse *lausevormideks*. Lausevormidest saab asenduste abil genereerida lausearvutuse valemeid, mida nimetatakse vaadeldava lausevormi *erikujudeks* (ehk *erijuhtudeks*). Me lubame ainult selliseid asendusi, kus asendatakse korraga mingi metamuutuja kõik esinemised. Matemaatikud nimetavad selliseid asendusi *substitutsioonideks*.

Lausevormidest saab asenduste teel moodustada lõpmatult palju erikujusid. Näiteks lausevormist $p \vee \neg(p \& q)$ saab genereerida järgmised erikujud:

Erikuju	Substitutsioon
$A \vee \neg(A \,\&\, C)$	$\{p/A, q/C\}$
$B \vee \neg(B \,\&\, A)$	$\{p/B, q/A\}$
$(B \,\&\, C) \vee \neg((B \,\&\, C) \,\&\, A)$	$\{p/B \,\&\, C, q/A\}$
$(B \vee \neg C) \vee \neg((B \vee \neg C) \,\&\, C)$	$\{p/B \vee \neg C, q/C\}$

(p/A tähistab niisugust substitutsiooni, kus kõik metamuutuja p esinemised asendatakse lausemuutujaga A.) Lausevormi ja tema erikujude vahel ei ole üksühest vastavust. Näiteks valem $A \vee \neg(A \,\&\, C)$ on erikuju järgmistest lausevormidest:

p	$\{p/A \vee \neg(A \,\&\, C)\}$
$p \vee q$	$\{p/A, q/\neg(A \,\&\, C)\}$
$p \vee \neg q$	$\{p/A, q/A \,\&\, C\}$
$p \vee \neg(q \,\&\, r)$	$\{p/A, q/A, r/C\}$
$p \vee \neg(p \,\&\, r)$	$\{p/A, r/C\}$

Analoogiliselt lausevormidega saab rääkida ka *arutlusvormidest*. Näiteks arutlusvormist

$$\frac{p \vee q \qquad \neg p}{q}$$

saab substitueerides genereerida erisuguseid kehtivaid arutlusi. Substitutsioon $\{p/A, q/B\}$ teisendab selle arutlusvormi lausearvutuse arutluseks

$$\frac{A \vee B \qquad \neg A}{B}$$

Toodud arutlusvorm on klassikaline disjunktiivse süllogismi *tuletusreegel*.

Ülesanded

1. Me eristame keemilisi elemente ja neid tähistavaid sõnu. Millised järgmistest lausetest on tõesed?
   (a) *Vask on vask.*
   (b) *“Vask” on vase nimetus.*
   (c) *Keemiline sümbol “Cu” tähistab “vaske”.*
   (d) *“Vask” on metall.*
2. Millised valemid on lausevormi $\neg p \supset q$ erikujud?
   (a) $A \supset B$;
   (b) $\neg A \supset \neg B$;

   (c) $\neg(A \supset B)$;
   (d) $\neg\neg(A \supset B) \supset (C \supset D)$.

3.5. Lausearvutuses mitteväljendatavad seosed

Keeles on tihti ka seoseid, mis ei ole lausearvutuse tehted.

Definitsioon 3.5. *Seose* kasutus on *tõeväärtusfunktsiooniline*, kui teda sisaldava lause tõeväärtuse määravad täielikult tema osalausete tõeväärtused.

Seos *sest* ei ole tavaliselt lausetes tõeväärtusfunktsiooniline. Võtame näiteks kolm tõest lauset:

- *Konstantin Päts oli president 1940. a.*
- *Päts sai presidendiks 1938. a.*
- *Päts oli Ajutise Valitsuse peaminister 1918. a.*

Nendest tõestest lausetest saab seosega *sest* moodustada nii tõese liitlause

> *Päts oli president 1940. a., sest ta sai presidendiks 1938. a.*

kui ka väära liitlause

> *Päts oli Ajutise Valitsuse peaminister 1918. a., sest ta sai presidendiks 1938. a.*

Viimane lause on väär, sest tema teine osalause ei põhjenda esimest osalauset.

Seose *enne kui* tõeväärtusfunktsioonilisuse lükkavad ümber laused

Mart Laar sai peaministriks enne, kui Edgar Savisaar sai peaministriks.

Mart Laar sai peaministriks enne, kui Mart Siimann sai peaministriks.

Esimene nendest lausetest on väär, teine on aga tõene, kuigi nende lausete osalaused on tõesed. Analoogiliselt saab näidata, et järgmised keeles kasutatavad seosed ei ole tõeväärtusfunktsioonilised:

- *On hästi teada, et ...*
- *Andres usub, et ...*
- *On tõenäone, et ...*
- *On paratamatu, et ...*
- *On võimalik, et ...*

Nende seoste tähistamiseks tuuakse lausearvutuse intensionaalsetes laiendites[1] (modaalloogikas, uskumiste loogikas jne.) sisse erisümbolid. Kahjuks selgub ka, et seos *kui ... siis* ei ole tihti lausetes tõeväärtusfunktsiooniline. Näiteks väljendi *Kui oleks ... siis oleks* kasutamisel on raske kindlaks teha, kas lause on tõene või väär:

Kui Mart Laar oleks suutnud rublaskandaali vältida, siis oleks ta edasi peaminister.

Ülesanded

1. Näidake, et järgmised seosed ei ole lausearvutuse tehted:
   (a) *Andres usub, et ;*
   (b) *On paratamatu, et ;*
   (c) *On võimalik, et .*

[1] *Intensionaalseteks* nimetatakse loogikaid, milles ei kehti Leibnizi võrduse printsiip. Loogikaid, kus see printsiip kehtib, nimetatakse *ekstensionaalseteks.*

3.6. Tõeväärtustabelid

Tõeväärtustabel võimaldab süstemaatiliselt läbi vaadata lausearvutuse valemi alamvalemite kõikvõimalikud väärtustused. Vaatame näiteks arutlust

Jüri loeb ajalehte või Jüri vaatab televiisorit.
Jüri ei loe ajalehte.

Jüri vaatab televiisorit.

Selle arutluskäigu saab lausearvutuses kirja panna kujul

$$\frac{A \vee T \qquad \neg A}{T}$$

mis on järgmise arutlusvormi erikuju:

$$\frac{p \vee q \qquad \neg p}{q}$$

Arutluse kehtivuse kindlakstegemiseks tuleb kontrollida, kas selle lausemuutujatele saab anda sellised tõeväärtused, et arutluse eeldused on tõesed ning järeldus väär. Kõigi võimaluste läbi vaatamiseks *väärtustame* arutlusvormis esinevad metamuutujad p ja q kõikvõimalikel erinevatel viisidel tõeväärtustega *tõene* (t) ja *väär* (v). Muutujate süstemaatilisel väärtustamisel on mõistlik valida tõeväärtuste komplekte tähestikulises järjekorras. Kahe muutuja korral saame järgmise tõeväärtuskomplektide järjestuse:

$$tt,\ tv,\ vt,\ vv.$$

Kanname need tõeväärtuste komplektid tabeli ridadesse ja arvutame seejärel igas reas arutlusvormi valemite tõeväärtused:

p	q	$p \vee q$	$\neg p$	q	
t	t	t	v	t	
t	v	t	v	v	
v	t	t	t	t	← arutlus kehtib
v	v	v	t	v	

Arutlusvormi eelduslaused on tõesed ainult tabeli kolmandas reas. Kuna ka järeldus q on kolmandas reas tõene, siis vaadeldav arutlusvorm kehtib. Esialgne arutlus kehtib samuti, sest ta on selle kehtiva arutlusvormi erikuju. Saab näidata (ülesanne 3.6.4), et iga kehtiv arutlus on mingi kehtiva arutlusvormi erikuju. Järelikult ei olegi vaja defineerida arutluste kehtivust variantide läbivaatamise kaudu, piisab vaid arutlusvormide kehtivuse defineerimisest ning arutluse nimetamisest *kehtivaks*, kui ta on mingi kehtiva arutlusvormi erikuju.

Kui tabeli täitmise käigus selgub, et me oleme probleemi juba lahendanud, siis võib tõeväärtustabeli täitmise pooleli jätta. Sellisel juhul saame *osalise tõeväärtustabeli*. Kuna eeldus $\neg p$ on tabeli esimestes ridades väär, siis oleks piisanud tõeväärtustabeli kahe viimase rea täitmisest:

p	q	$p \vee q$	$\neg p$	q
t	t		v	
t	v		v	
v	t	t	t	t
v	v	v	t	v

Toodud tõeväärtustabelis on neli rida. Kui valemis on n lausemuutujat, siis on tõeväärtustabelis 2^n rida. Näiteks valemi

$$A \,\&\, \neg(B \supset (A \vee C))$$

korral saame joonisel 3.1 toodud tõeväärtustabeli. Selle tabeli kohale on eelnevalt märgitud tehete sooritamise järjekord, mille määravad lausearvutuse süntaksi- ja semantikareeglid. Esmalt tehakse kõige sisemistes sulgudes asuv disjunktsioonitehe. Seejärel

			4	3	2	1
A	B	C	A	$\&\,\neg(B$	$\supset\,(A$	$\vee\, C))$
t	t	t	v	v	t	t
t	t	v	v	v	t	t
t	v	t	v	v	t	t
t	v	v	v	v	t	t
v	t	t	v	v	t	t
v	t	v	v	t	v	v
v	v	t	v	v	t	t
v	v	v	v	v	t	v

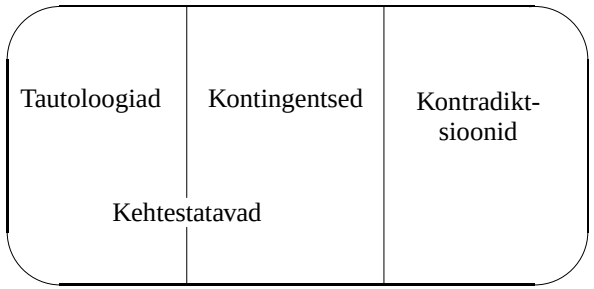
Joonis 3.1. Tõeväärtustabeli näide.

leitakse implikatsiooni ja eituse väärtused. Alles siis saab arvutada konjunktsiooni väärtuse.

Tõeväärtustabeli täitmisel on võimalikud ka variatsioonid. Vahel kirjutatakse tõeväärtustabelisse valemi alla ka lausemuutujate tõeväärtused. Samuti võib varieerida tabeli täitmise suunda: horisontaalsel täitmisel täidetakse tabel ridade kaupa, vertikaalsel täitmisel veergude kaupa.

Ülesanded

1. Arvutage valemi tõeväärtustabel:
$$A \,\&\, D \vee \neg C \vee D.$$
2. Näidake tõeväärtustabeli abil, et valem on loogiliselt tõene:
$$A \supset (B \supset A).$$
3. Kontrollige tõeväärtustabeli abil järgmise arutluse kehtivust.
Kui Einsteini mehhaanika on õige, siis Newtoni mehhaanika ei saa olla õige. Einsteini mehhaanika on õige parajasti siis, kui ruum on mitteeukleidiline. Ruum on mitteeukleidiline või Newtoni mehhaanika on õige.



Joonis 3.2. Lausearvutuse lausete liigitus.

4. *Tõestage, et iga kehtiv arutlus on mingi kehtiva arutlusvormi erikuju.

3.7. Tautoloogia, kontradiktsioon ja kontingentne lause

Lausearvutuse valemeid liigitatakse järgmiselt:

- *Loogiliselt tõene* valem e. *tautoloogia* on kõigil lausemuutujate väärtustustel tõene.
- *Loogiliselt väär* valem e. *kontradiktsioon* on kõigil lausemuutujate väärtustustel väär.
- Ülejäänud valemeid nimetatakse *kontingentseteks*.

Lausearvutuse valemite liigituse skeem on antud joonisel 3.2. Valemit, mis on mingil lausemuutujate väärtustusel tõene, nimetatakse *kehtestatavaks*. Enamus lausearvutuse valemeid ei ole loogiliselt tõesed ega loogiliselt väärad, vaid kontingentsed, s.t. võivad olla nii tõesed kui ka väärad. Kontingentsed on näiteks valemid

$$A \vee B,$$
$$A \,\&\, (\neg B \equiv D).$$

$$\begin{array}{l} \neg(A \,\&\, \neg(B \supset (A \vee C))) \\ v \quad t \\ \quad t \quad t \\ \qquad\quad t\ v \quad v \\ \qquad\qquad\ v \quad v \end{array}$$

Joonis 3.3. Kiirkontrolli meetodi rakendamise näide.

Jooniselt 3.1 on näha, et valem

$$A \,\&\, \neg(B \supset (A \vee C))$$

on kontradiktsioon, mistõttu tema eitus

$$\neg(A \,\&\, \neg(B \supset (A \vee C)))$$

on tautoloogia. Tihti saab valemi loogilist tõesust ja väärust ka lihtsamini kontrollida. Selleks kasutatakse *kiirkontrolli meetodit*, mille rakendamise näide on toodud joonisel 3.3. Me eeldame, et see tautoloogia on mingil lausemuutujate väärtustusel väär, ja saame sellest semantikareeglite abil vastuolu. Kuna valem koos eitusega on väär, siis on sulgudes olev konjunktiivne avaldis tõene. Seega on konjunktsiooni mõlemad alamvalemid tõesed. Siis on aga implikatsioon väär, millest saame, et implikatsiooni esimene pool on tõene, teiseks pooleks olev disjunktsioon aga väär. Disjunktsioon on väär ainult siis, kui tema mõlemad alamvalemid on väärad. Nüüd on kõigil atomaarsetel valemitel tõeväärtused. Nende ülevaatamisel selgub, et A on samaaegselt tõene ja väär, mis on vastuolus samasusseadusega. Järelikult ei saa vaadeldav valem olla väär, mistõttu ta on kõigil lausemuutujate väärtustustel tõene, s.t. ta on tautoloogia.

Kiirkontrolli meetodit saab kasutada ka loogilise vääruse näitamiseks. Kontingentse lause korral annab see meetod vastuolu asemel vaadeldava lause kaks lubatavat väärtustust. Näiteks valemi $A \vee B$ korral saame kiirkontrolli meetodi abil väärtustused

$A \vee B$		$A \vee B$
$t\ t\ t$	ja	$v\ v\ v$

Kiirkontrolli meetod võimaldab leida *kontranäiteid*, s.t. selliseid lausemuutujate väärtustusi, mis lükkavad mingi hüpoteesi ümber. Näiteks lause tautoloogilisuse näitamiseks kontrollitakse, kas leidub selline väärtustus, millel lause on väär. Valemi $A \vee B$ korral saime kontranäite nii väitele, et see valem on kontradiktsioon, kui ka väitele, et valem on tautoloogia.

Tautoloogiad on loogikas väga olulised, sest nad väljendavad vankumatuid tõdesid, mis peavad paika igas olukorras. Tautoloogiad on näiteks valemid

$$A \vee \neg A,$$

$$A \supset A,$$

$$\neg(A \,\&\, \neg A),$$

$$A \supset (B \supset A).$$

Esimene nendest tautoloogiatest vastab välistatud kolmanda seadusele, teine samasusseadusele ja kolmas vasturääkivusseadusele. Tautoloogilised lausevormid kujutavad endast *tuletusreegleid*. Kirjutades neljanda tautoloogia lausevormina

$$p \supset (q \supset p),$$

saame sobiva asenduse teel tautoloogia

$$C \supset (A \supset C),$$

mille abil saab loogiliselt *järeldada*

$$C \models A \supset C,$$

sest implikatsiooni vasaku poole tõesusest järeldub tema parema poole tõesus. Seega võib loogikas kasutada reeglit, mis lubab võtta tõese järeldusega implikatsiooni eelduseks suvalise valemi:

$$\frac{p}{q \supset p}$$

Me mainisime loogikamõistetevaheliste matemaatilise seoste kirjeldamise juures (punktis 2.7), et lauseloogika tehetega saab lausetevahelised põhilised loogilised seosed taandada tautoloogilisuse kontrollimisele. Kuna järgmised väited kehtivad ka predikaatloogikas, siis kasutame nendes *tautoloogia* mõiste asemel *loogilise tõesuse* mõistet.

Lause 3.6. *Lausete hulk* $\{p_1, p_2, \ldots, p_n\}$ *on vasturääkiv* $\Leftrightarrow$ *valem* $\neg(p_1 \,\&\, p_2 \,\&\, \ldots \,\&\, p_n)$ *on loogiliselt tõene.*

Lause 3.7. $\{p_1, p_2, \ldots, p_n\} \models q \Leftrightarrow$ *valem* $(p_1 \& p_2 \& \ldots \& p_n) \supset q$ *on loogiliselt tõene.*

Lause 3.8. *Laused* p *ja* q *on loogiliselt ekvivalentsed* $\Leftrightarrow$ $p \equiv q$ *on loogiliselt tõene.*

Lause 3.9. p *on loogiliselt väär* $\Leftrightarrow \neg p$ *on loogiliselt tõene.* p *on kontingentne* $\Leftrightarrow$ *mitte kumbki lausetest* p *ja* $\neg p$ *ei ole loogiliselt tõene.*

Ülesanded

1. Tõestage, et tautoloogiast (kui tautoloogilisest vormist) asenduse teel saadud lause on samuti tautoloogia.
2. Tõestage laused:
   (a) 3.6;
   (b) 3.7;
   (c) 3.8;
   (d) 3.9.

3.8. Ülesanded

1. Tähistagu

   A – *Üliõpilane Rebane on kodus*

   B – *Üliõpilane Rebane on loengul*

   C – *Üliõpilane Rebane konspekteerib*

   D – *Üliõpilane Rebane magab*

   E_1 – *On esimene loeng*

   E_2 – *On teine loeng*

   E_3 – *On kolmas loeng*

   F – *Rebane mõtleb lõunasöögist*

   G – *Rebasel on raske*

   Lugeda:

   (a) $E_1 \supset D$;

   (b) $E_2 \supset A \,\&\, D \vee B \,\&\, D$;

   (c) $E_3 \supset B \,\&\, F$;

   (d) $B \supset D \vee C \,\&\, G$;

   (e) $\neg B \supset \neg G$;

   (f) $C \vee F \supset D$.

2. Kirjutada valemina, kasutades eelmise ülesande tähistusi:

   (a) *Rebane magab auditooriumis ainult teise loengu ajal.*

   (b) **Kui Rebasel on loengul alati raske, siis on tal raske ainult loengul.*

   (c) **Kui Rebane on iga loengu ajal kohal ja konspekteerib, siis ta magab ainult kodus.*

3. Formaliseerige arutlus ja kontrollige selle kehtivust:

   Kui ma lähen sisseoste tegema, siis ma ostan kas riideid või toitu. Ma kulutan raha ainult siis, kui ma ostan riideid või toitu. Ma ei ole raha kulutanud. Seepärast ma ei käinud sisseoste tegemas.

4. Formaliseerige tekstilõik lausearvutuse arutlusena:

   Kui ese on kullast, siis on selle keemiline number 79. Mart usub, et ese on kullast. Järelikult usub Mart, et eseme metalli keemiline number on 79.

5. Tõestage samasused:

   (a) $A \vee B = B \vee A$;

   (b) $\neg p \,\&\, \neg q = \neg(p \vee q)$;

   (c) $(p \,\&\, q) \,\&\, r = p \,\&\, (q \,\&\, r)$.

6. Kontrollige tõeväärtustabeli abil, kas lausevorm on tautoloogiline, kontradiktoorne või kontingentne:

   (a) $(p \vee q) \,\&\, (\neg p \,\&\, \neg q)$;

   (b) $(p \supset r) \supset (q \supset r)$;

   (c) $(p \,\&\, q) \supset (p \vee r)$.

7. Näidake järgmiste argumentide kehtivust või mittekehtivust, kasutades tõeväärtustabeleid või tuues kontranäiteid:

   (a)
   $$\begin{array}{l} B \equiv \neg A \\ A \\ \hline B \end{array}$$

   (b)
   $$\begin{array}{l} \neg(A \,\&\, B) \\ A \\ \hline \neg B \end{array}$$

   (c)
   $$\begin{array}{l} ((A \vee B) \vee C) \supset D \\ \neg B \\ \hline D \end{array}$$

8. Kontrollige, kas lausevormid on ekvivalentsed:

   (a) $p \,\&\, q$ ja $\neg(\neg p \vee \neg q)$;

   (b) $p \vee q$ ja $(\neg p \supset \neg q)$;

   (c) $p \,\&\, \neg q$ ja $\neg(p \supset q)$.

4. Predikaatarvutus

Elu on suur müsteerium. Andku taevas, et seda iial narmasteks ei harutata, elu saladust ei avastata.
— Linda Rummo

Ma olen loonud mitte millestki veidra uue universumi.
— János Bolyai (1802–1860)

Mis on inimene looduses? Lõpmatusega võrreldes mitte midagi, kuid kõik võrreldes mitte millegagi, ta on midagi, mis on mitte millegi ja kõige vahel.
— Blaise Pascal (1623–1662), *Pensées*, 1670

Predikaatarvutuse loojaks peetakse saksa matemaatikut Gottlob Freget (1848–1925). Lausearvutuses piirdub lausete analüüs nende lihtosalausete eraldamisega. *Predikaatarvutus* e. predikaatloogika võimaldab lauseid detailsemalt analüüsida, põhinedes lihtlausete traditsioonilisel jaotamisel lauseliikmeteks. Näiteks lause

Priit sööb prääniku

koosneb alusest, öeldisest ja sihitisest.

Predikaatarvutus on lausearvutusest üldisem, sest lausete detailsem analüüs võimaldab kehtivaks kuulutada ka selliseid arutlusi, mis lausearvutuse seisukohast seda ei ole. Näiteks arutlus

Kõik linnud lendavad.
Part on lind.

Part lendab

kehtib predikaatarvutuses. Lausearvutuses ta aga ei kehti, sest tema lausetel puuduvad osalaused.

Predikaatloogika on välja kasvanud Aristotelese loogikast, kelle *süllogistika* põhineb ka sellisel lauseanalüüsil. Kuid Aristoteles (384–322 e.m.a.) ja tema järeltulijad kasutasid tänapäevasest erinevat lausete üleskirjutusviisi. Algebralise üleskirjutusviisi kasutuselevõtt eelmise sajandi lõpus võimaldas arendada süstemaatiliselt predikaatloogikat, mistõttu see on Aristotelese süllogistika kui uurimistehnika teadusest praktiliselt välja tõrjunud. Alles on jäänud vaid tähtsamad süllogismi moodused ja lausete analüüsimise metoodika.

4.1. Predikaadid ja konstandid

Vaatame standardset lauset

Jüri armastab Marit.

See lause koosneb kahest nimisõnast ja ühest tegusõnast, kusjuures tegusõna *armastab* seostab omavahel nimisõnade *Jüri* ja *Mari* poolt tähistatavad objektid. Me võime käsitleda sõna *armastab* kui kahe objekti vahelist seost. Sellisel juhul me jätame lausesse nimisõnade asemele augud või asendame nad *indiviidmuutujatega*:

- _____ armastab . . . . . .
- x armastab y.

Indiviidmuutuja tähistab suvalist objekti meie fikseeritud objektide hulgast ehk *universumist*. Indiviidmuutujaid tähistatakse väiketähtedega tähestiku lõpust: $x, y, z, \ldots$

Niisugust auklikku lauset, mis sisaldab ainult objektidevahelist seost, mitte objekte endid, nimetatakse *predikaadiks*. Predikaadi

seotavaid objekte nimetatakse tema *argumentideks*. Meie näite korral on tegemist 2-kohalise (ehk *binaarse*) predikaadiga *armastab*. Tavaliselt nimetatakse kõik kasutatavad predikaadid ümber nii, et nende nimed koosnevad ainult ühest tähest. Selleks on mugav võtta predikaati määrava tegusõnafraasi esimene täht. Antud juhul sobib selleks sümbol A sõnast *armastab*. Kui tähestiku tähtedest tuleb puudus, siis võetakse predikaatide tähistamiseks kasutusele ka alaindeksitega tähed. Ülaindeksit kasutatakse harilikult predikaadi argumentide arvu (ehk *aarsuse*) näitamiseks: kas predikaat määrab 1-, 2- või enamakohalise seose. Loomulikult on võimalikud ka 0-kohalised seosed, mis on tavalised lausearvutuse laused. Näiteks võib sümboliga $A^{(0)}$ tähistada kogu vaadeldava lause

Jüri armastab Marit.

Seda lauset saab käsitleda ka kui 1-kohalist (ehk *unaarset*) predikaati J, mis seostab Jüri tema armastatava objektiga:

- *Jüri armastab* . . . . . .
- *Jüri armastab* y.

Objekte tähistatakse *konstantsümbolitega*. Konstantsümboliks valitakse tavaliselt vaadeldava objekti nime esimene täht. Tähistame näiteks *Jüri* sümboliga j ja *Mari* sümboliga m. Sellisel moel saame predikaatloogika keeles lauseid lühemalt kirja panna. Predikaatarvutuse lause esimeseks sümboliks on selle lausega kirjeldatava seose predikaatsümbol, millele järgnevad (kas sulgudes või ilma) tema argumentide konstantsümbolid. Predikaatsümbolitena kasutatakse tavaliselt tähestiku suurtähti ja konstantsümbolitena väiketähti. (Ent tihti on hoopis vastupidi: predikaate tähistatakse väiketähtedega ja konstantsümboleid suurtega. Viimane moodus on vastavuses ka pärisnimede kirjutamise reeglitega, mille kohaselt alustatakse pärisnimesid suurtähega.)

Näiteks saab eespool toodud lause kirjutada predikaatloogikas kujul

$$Ajm.$$

Sõltuvalt predikaadi argumentide arvust saab sama lauset üles kirjutada ka teistel kujudel:

$$Jm,$$
$$A^{(0)}.$$

Isegi juhul, kui predikaatide argumentide arvud on fikseeritud, saab antud lauset erinevatel viisidel formaliseerida (kasutades erisuguseid üleskirjutamisreegleid):

$$A(j, m),$$
$$a(\textit{Jüri}, \textit{Mari}),$$
$$\textit{armastab}\ (\textit{jüri}, \textit{mari}).$$

Kui predikaatsümbol eelneb oma argumentide tähistustele, siis öeldakse, et lause on *prefikskujul*. Kahekohaliste predikaatide korral kasutatakse tihti ka *infikskuju*, milles predikaatsümbol asub oma argumentide vahel:

$$\textit{Jüri armastab Marit},$$
$$j\ A\ m.$$

Keeleliselt on võimalik konstrueerida ka 3-kohalisi predikaate. Näiteks, tähistades *Peetri*, *suve*, *talve*, *Kanaari saared* ja *Otepää* vastavalt sümbolitega p, s, t, k ja o ning *puhkamise seose* sümboliga P, saame laused

Peeter puhkab suvel Kanaari saartel.

Peeter puhkab talvel Otepääl

esitada predikaatloogikas kujul

$$Ppsk,$$
$$Ppto.$$

Ainult ühte predikaati sisaldavaid predikaatarvutuse lauseid nimetatakse *atomaarseteks lauseteks* e. *aatomiteks*. Atomaarsetest

lausetest saab lausearvutuse tehete abil moodustada *liitlauseid*. Näiteks saab lause

Jüri armastab ennast ja Marit

ümber sõnastada liitlausena

Jüri armastab ennast ja Jüri armastab Marit,

millele vastab predikaatloogika lause

$$Ajj \,\&\, Ajm.$$

Tähistades *tervisespordiga tegelemise* ja *haigeks jäämise* predikaadid sümbolitega T ja H ning *Andrese* sümboliga a, saame lause

Kui Andres tegeleb tervisespordiga, siis ta ei jää haigeks

formaliseerida kujul

$$Ta \supset \neg Ha.$$

4.2. Kvantorid

Predikaatloogika laused ei tohi üldjuhul sisaldada indiviidmuutujaid. Kui me soovime väita midagi **kõigi** objektide kohta vaadeldavast universumist, siis me toome sisse *üldisuskvantori*. Näiteks lause

Jüri armastab kõiki

formaliseeritakse valemina, kus armastamise predikaadi teise argumendi kohale kirjutatakse indiviidmuutuja ja seotakse see üldisuskvantoriga $\forall x$:

$$\forall x\, Ajx.$$

Sümbol $\forall$ on tagurpiditäht A (ingliskeelsest sõnast *all*). Saadud lauset loetakse: "**Iga** objekt x on selline, et objekt j on temaga seoses A" või "Iga x korral kehtib Ajx".

Eksistentsikvantori (ehk olemasolukvantori) abil väidetakse, et **mingi** objekt vaadeldavast universumist rahuldab antud tingimust. Näiteks predikaatloogika lause

$$\exists x\, Ajx.$$

ütleb meile, et

Jüri armastab kedagi (või midagi).

Eksistentsikvantoris $\exists x$ esinev sümbol $\exists$ on ümberpööratud täht E (ingliskeelsest sõnast *exist*) ja me loeme eespool toodud formaalset lauset: "**Leidub** objekt x, millega objekt j on seoses A" või "Leidub x, nii et kehtib Ajx". Eksistentsikvantorit sisaldav lause on tõene nii juhul, kui antud tingimust rahuldab ainult üks objekt, kui ka juhul, kui selliseid objekte on palju. Sellepärast on seda lauset õigem lugeda: "Leidub vähemalt üks x, mis rahuldab tingimust Ajx".

Ülesanded

1. Formaliseerige laused:
   (a) **Kõik on valge või must.*
   (b) *Kõik on valge või kõik on must.*
   (c) *Fido on koer.*
   (d) **Igaüks armastab kedagi.*
   (e) *On olemas keegi, keda kõik armastavad.*
   (f) *Kui keegi armastab kedagi, siis John armastab ennast.*
   (g) **Mitte keegi peale Johni ei armasta iseennast.*
   (h) *Igaüks armastab kas ennast või mõnda naist.*
   (i) *Kui keegi ei suudle Johni, siis Mary teeb seda.*
   (j) *New Yorki elanikud armastavad oma kodulinna.*
   (k) *Kui John ei armasta New Yorki, siis ta ei ela seal.*
   (l) *Kõik on hea, mis hästi lõpeb.*
2. Formuleerige eelmises ülesandes toodud lausete eitused.

4.3. Predikaatarvutuse süntaks

Predikaatarvutuse süntaksi kirjelduse saame lausearvutuse tähestiku ja süntaksireeglite täiendamise teel. *Predikaatarvutuse tähestik* koosneb järgmistest märkidest ja sümbolitest:

- predikaatsümbolid:
  $A, B, C, \ldots, A_1, A_2, \ldots$ (suurtähed)
- indiviidmuutujad: $x, y, z, \ldots, x_1, y_1, z_1, \ldots$
  (tähestiku viimased tähed)
- indiviidkonstantide sümbolid:
  $a, b, c, \ldots, v, a_1, a_2, \ldots$
  (tähestiku esimesed tähed)
- loogiliste tehete sümbolid: $\neg\ \&\ \vee\ \supset\ \equiv\ \exists\ \forall$
- kirjavahemärgid: ()

Me eeldame, et iga predikaatsümboli argumentide arv on fikseeritud. *Indiviidtermid* on indiviidkonstantide sümbolid ja indiviidmuutujad.

Defineerime kõigepealt kõige lihtsamad *predikaatarvutuse valemid* — atomaarsed valemid — ja laiendame seejärel valemite klassi loogiliste tehete abil.

Definitsioon 4.1 (predikaatarvutuse süntaks). Atomaarne valem *e.* aatom *on kas kujul L, kus L on 0-kohaline predikaatsümbol (lausemuutuja), või kujul $Pt_1 \ldots t_n$ (või $P(t_1, \ldots, t_n)$), kus P on n-kohaline predikaatsümbol ja $t_1, \ldots, t_n$ on indiviidtermid.*

1. *Atomaarne valem on* valem.
2. *Kui p on valem, siis $\neg p$ on valem.*
3. *Kui p ja q on valemid, siis $(p \,\&\, q)$, $(p \vee q)$, $(p \supset q)$ ja $(p \equiv q)$ on valemid.*

4. *Kui p on valem ja v on indiviidmuutuja, siis on $\forall v\ p$ ja $\exists v\ p$ on valemid.*
5. *Valemi välimised sulud võib ära jätta.*

Näide 4.2. Avaldis

$$Ajx \,\&\, \forall x\ Axx$$

on predikaatarvutuse valem, sest

1. Ajx on atomaarne valem (definitsiooni 1. punkti põhjal).
2. Axx on atomaarne valem (1).
3. $\forall x\ Axx$ on valem (4), sest Axx on valem ja x on indiviidmuutuja.
4. $(Ajx \,\&\, \forall x\ Axx)$ on valem (3), sest Ajx ja $\forall x\ Axx$ on valemid.
5. $Ajx \,\&\, \forall x\ Axx$ on valem (5).

Muutuja esinemine on valemis *seotud*, kui ta esineb kvantoris või kvantorile alluvas avaldises. Kui muutuja esinemine ei ole valemis seotud, siis see muutuja esinemine on *vaba*.

Muutuja on valemis *seotud*, kui kõik tema esinemised on valemis seotud. Vastasel juhul on muutuja valemis *vaba*.

Valem on *kinnine*, kui kõik tema muutujad on seotud. Vastasel juhul nimetatakse valemit *lahtiseks*. Näiteks muutuja x esimene esinemine valemis

$$Ajx \,\&\, \forall x\ Axx$$

on vaba, kuid ülejäänud kolm muutuja x esinemist on seotud. Järelikult on muutuja x vaadeldavas valemis vaba ja see valem on lahtine.

Vabu muutujaid sisaldavate lausete tõeväärtuste kindlakstegemine võib tekitada raskusi. Sellepärast jätame vabu muutujaid sisaldavad valemid vaatluse alt välja. *Lauseks* nimetatakse predikaatarvutuse valemit, milles ei ole vabu muutujaid. Näiteks avaldis

$$\exists x\ (Ajx \,\&\, \forall x\ Axx)$$

on predikaatarvutuse lause, sest kõik temas esinevad muutujad on seotud.

Ülesanded

1. Leidke valemitest seotud ja vabad muutujad:
   (a) $\forall x\, Px \vee Qxy$;
   (b) $\forall y\, (Qx \supset \forall z\, Pyz)$;
   (c) $\forall x \neg(Px \supset \exists y \forall z\, Qxyz)$.

4.4. Tarski semantika

Predikaatarvutuse lause tõesuse kontrollimiseks tuleb läbi vaadata selle lause kõikvõimalikud tähendused. Lausearvutuse korral piisab tõeväärtustabeli arvutamisest, sest selle iga rida vastab mingile võimalikule olukorrale. Predikaatarvutuse korral ei ole asi nii lihtne, kuna predikaadi tõeväärtus sõltub tema argumentidest.

Tähistame Jüri tähega j ja naiseks olemise predikaadi tähega N. Siis on lause

$$Nj$$

väär, sest Jüri ei ole naissoost. Kuid me võime ette kujutada olukorda, kus j tähistab hoopis Marit, ning sellises interpretatsioonis on vaadeldav lause tõene. Järelikult on see lause kontingentne.

Erinevate interpretatsioonide konstrueerimise hõlbustamiseks tuuakse predikaatarvutusse signatuuri mõiste. Predikaatarvutuse *signatuur* sisaldab kõiki kasutatava tähestiku predikaat- ja konstantsümboleid. Signatuuri sümboleid nimetatakse vahel ka *mitteloogilisteks sümboliteks*, sest nende tähendus ei ole loogikas fikseeritud ning seda on vaja eraldi täpsustada. Kõiki ülejäänud tähestiku sümboleid nimetatakse *loogilisteks sümboliteks*. Loogilised sümbolid on näiteks loogiliste tehete sümbolid ja indiviidmuutujad. Tavaliselt käsitletakse loogilise sümbolina ka objektidevahelise võrduse seost $=$.

Olgu meil fikseeritud mingi predikaatarvutuse tähestik, $Const$ on kõigi selle indiviidkonstantide sümbolite loetelu ja $Pred$ kõigi kasutatavate predikaatsümbolite loetelu. Sellise predikaatarvutuse *signatuuri* σ saab defineerida kui nende kahe loetelu ühendi:

$$\sigma = \langle Const; Pred \rangle.$$

Eespool toodud näite korral võib signatuuriks võtta ainult selles nimetatud objektide ja predikaatide tähistused:

$$\sigma = \langle j, m; N \rangle,$$

kus Mari on tähistatud tähega m.

Vaadeldava predikaatarvutuse *interpretatsiooniks* I nimetatakse loetelu, mis koosneb mittetühjast hulgast U_I, mida nimetatakse *universumiks* e. interpretatsiooni kandjaks, ja selle predikaatarvutuse signatuuri kõigi elementide interpretatsioonidest universumis U_I:

$$I = \langle U_I; Const_I; Pred_I \rangle,$$

kus $Const_I$ on saadud loetelust $Const$ iga konstantsümboli c asendamisel universumi U_I mingi elemendiga c_I ja $Pred_I$ on saadud predikaatsümbolite loetelust $Pred$ iga predikaatsümboli P asendamisel tema *ekstensiooni* e. *tõehulgaga* P_I universumis U_I.

Meie näite korral on mõistlik valida universumiks kahest inimesest koosnev hulk

$$H = \{\text{Jüri}, \text{Mari}\}$$

ja interpretatsiooniks loetelu

$$I = \langle H; j_I, m_I; N_I \rangle,$$

kus signatuuri sümboleid interpreteeritakse “loomulikul” viisil:

$$j_I = \text{Jüri},$$
$$m_I = \text{Mari},$$
$$N_I = \{\text{Mari}\} \subseteq H.$$

Predikaadi Nx ekstensioon koosneb antud juhul ainult ühest elemendist, sest hulgas H ei ole rohkem naisi.

Kuna predikaatarvutuse lausetes ei ole vabu muutujaid, siis on iga lause tema tähestikku sisaldava interpretatsiooni korral kas

tõene või väär. Kui valem p on interpretatsioonis I tõene, siis me kirjutame

$$I \models p,$$

aga vastasel juhul kirjutame

$$I \not\models p.$$

Valemite hulga L *mudeliks* nimetatakse interpretatsiooni I, milles kõik L laused on tõesed:

$$I \models p \quad (p \in L).$$

Näiteks meie kirjeldatud interpretatsioon I on lause

$$Nm$$

mudel, sest Mari on naissoost.

Binaarse predikaadi ekstensiooniks on universumi elementide paaride hulk, mis määrab, milliste argumendipaaride korral on predikaat tõene. Näiteks kahest valemist koosneva hulga

$$L = \{\forall x\, Ajx, Amj\}$$

mudeliks sobib interpretatsioon

$$I = \langle\{\text{Jüri, Mari}\};\, j_I, m_I;\, A_I\rangle,$$

$$j_I = \text{Jüri},$$

$$m_I = \text{Mari},$$

$$A_I = \{(\text{Mari, Jüri}), (\text{Jüri, Mari}), (\text{Jüri, Jüri})\},$$

milles Jüri armastab nii ennast kui ka Marit ja Mari armastab ainult Jürit, sest kõik L laused on selles interpretatsioonis tõesed.

Me võime eeldada, et interpretatsioonis on piisavalt indiviidide konstantsümboleid, nii et nendega saab ära tähistada kõik universumi elemendid. Konstantsümboliteks sobivad näiteks universumi elementide nimede esitähed. Olgu $p\{x/c\}$ valem, mis on saadud valemist p muutuja x kõigi vabade esinemiste *asendamisel* sümboliga c. Kui x on valemi p ainus vaba muutuja ja c on universumi mingile elemendile vastav konstantsümbol, siis on $p\{x/c\}$ predikaatarvutuse kinnine valem ehk lause. Näiteks, tehes asenduse $\{x/m\}$ valemis

$$Ajx,$$

saame lause

$$Ajm.$$

Asenduse $\{y/j\}$ puhul valem ei muutu, sest ta ei sisalda muutujat y vabalt.

Üldisuskvantor väidab, et tema kvantifitseeritav valem on universumi kõigi elementide korral tõene, s.t. see valem on tõene iga asenduse korral, milles me asendame kvantoris esineva muutuja universumi mingit objekti tähistava konstantsümboliga. Eksistentsikvantori tõesuseks piisab, et valem oleks tõene vähemalt ühe asenduse korral.

Anname nüüd predikaatarvutuse valemite semantika formaalse definitsiooni.

Definitsioon 4.3 (predikaatarvutuse semantika). *Olgu antud interpretatsioon I, mille universum on U_I.*

1. *Kinnine atomaarne valem $Pt_1 \ldots t_n$ on* interpretatsioonis I tõene *siis ja ainult siis, kui konstantide $t_1, \ldots, t_n$ interpretatsioonide korteež $(t_{1I}, \ldots, t_{nI})$ kuulub predikaadi P ekstensiooni:*
$$I \models Pt_1 \ldots t_n \Leftrightarrow (t_{1I}, \ldots, t_{nI}) \in P_I.$$
2. $I \models \neg p \Leftrightarrow I \not\models p.$
3. $I \models p \,\&\, q \Leftrightarrow I \models p$ *ja* $I \models q.$
   $I \models p \vee q \Leftrightarrow I \models p$ *või* $I \models q.$
   $I \models p \supset q \Leftrightarrow I \not\models p$ *või* $I \models q.$
   $I \models p \equiv q \Leftrightarrow (I \models p$ *ja* $I \models q)$ *või* $(I \not\models p$ *ja* $I \not\models q).$
4. $I \models \forall x\, p \Leftrightarrow$ *iga* $c \in U_I$ *korral* $I \models p\{x/c\}.$
   $I \models \exists x\, p \Leftrightarrow$ *leidub* $c \in U_I$, *nii et* $I \models p\{x/c\}.$

Näide 4.4. Leheküljel 104 kirjeldatud interpretatsiooni I korral kehtib

$$I \models \forall x\, Ajx \,\&\, Amj,$$

sest

1. $I \models Ajj$ (definitsiooni 1. punkti põhjal), sest $(j_I, j_I) \in A_I$.
2. $I \models Ajm$ (1), sest $(j_I, m_I) \in A_I$.
3. $I \models \forall x\, Ajx$ (4), sest $I \models Ajj$ ja $I \models Ajm$.
4. $I \models Amj$ (1), sest $(m_I, j_I) \in A_I$.
5. $I \models \forall x\, Ajx \,\&\, Amj$ (3), sest $I \models \forall x\, Ajx$ ja $I \models Amj$.

Vaadeldes valemi võimalike tähendustena kõiki tema signatuuri interpretatsioone, saame nüüd defineerida predikaatarvutuse valemite loogilise tõesuse. Me ütleme, et valem p on (predikaatarvutuses) *loogiliselt tõene*, kui ta on tõene igas (tema tähestiku) interpretatsioonis, ja kirjutame

$$\models p.$$

Valem p on (predikaatarvutuses) *loogiliselt väär*, kui ta on igas interpretatsioonis väär, s.t.

$$\models \neg p.$$

Kõiki ülejäänud valemeid nimetatakse *kontingentseteks*. Loogiliselt tõeseid ja kontingentseid valemeid nimetatakse *kehtestatavateks*.

Predikaatarvutuse valemite loogilise tõesuse definitsiooni saab ümber sõnastada matemaatilise väitena.

Lause 4.5. $\models p \Leftrightarrow$ *iga interpretatsiooni* I *korral kehtib* $I \models p$.

Defineerime interpretatsioonide abil ka loogilise järelduvuse mõiste. Me ütleme, et valemite hulgast Γ *järeldub (predikaatarvutuses) loogiliselt* valem p, kui iga Γ mudel on ka p mudel, ning kirjutame

$$\Gamma \models p.$$

Kui eelduste hulk on tühi ($\Gamma = \emptyset$), langevad loogilise järelduvuse ja loogilise tõesuse mõisted kokku:

$$\models p \Leftrightarrow \emptyset \models p.$$

Näitame nüüd, et kõik tautoloogiatest asenduste teel saadavad predikaatarvutuse valemid on loogiliselt tõesed.

Lause 4.6. *Kõik lausearvutuse tautoloogiate erikujud on (predikaatarvutuses) loogiliselt tõesed.*

Tõestus. Olgu meile antud mingi tautoloogia. Moodustame selle tautoloogia erikuju, asendades lausemuutujad predikaatarvutuse valemitega. Lausemuutujaid asendavad valemid on igas interpretatsioonis kas tõesed või väärad. Kuna tautoloogia on lausemuutujate suvaliste väärtuste korral tõene, siis on ka tema erikuju igas interpretatsioonis tõene. □

Näiteks on valem

$$\forall x\, Px \vee \neg \forall x\, Px$$

loogiliselt tõene, sest ta on saadud tautoloogiast $A \vee \neg A$ asenduse $\{A/\forall x\, Px\}$ teel.

Predikaatarvutuses saab moodustada ka selliseid loogiliselt tõeseid valemeid, mis ei ole tautoloogiate erikujud. Valem

$$\forall x\, (Px \supset \exists x\, Px)$$

ei ole tautoloogia erikuju, sest seda ei saa esitada lausearvutuse tehteid sisaldava liitlausena. Samas on see valem loogiliselt tõene. Selles võib veenduda, valides valemi suvalise interpretatsiooni I ja asendades kvantorialuses avaldises muutuja x interpretatsiooni I universumi mingit elementi tähistava konstantsümboliga c. On kaks võimalust. Kui $I \not\models Pc$, siis definitsiooni 4.3 kolmanda punkti põhjal

$$I \models Pc \supset \exists x\, Px.$$

Vaadeldav implikatsioon on interpretatsioonis I tõene ka juhul, kui $I \models Pc$, sest siis $I \models \exists x\, Px$. Järelikult kehtib iga interpretatsiooni I korral

$$I \models \forall x\, (Px \supset \exists x\, Px),$$

mis tõestabki valemi loogilise tõesuse.

Ülesanded

1. Väljendage predikaatarvutuses kvantor *see*:
   See tüdruk, kes meile lehvitab, on tuttava näoga.

2. Tõestage kvantorite duaalsusreeglid:
   (a) $\neg\forall x\ p = \exists x\ \neg p$;
   (b) $\neg\exists x\ p = \forall x\ \neg p$;
   (c) $\forall x\ p = \neg\exists x\ \neg p$;
   (d) $\exists x\ p = \neg\forall x\ \neg p$.

4.5. Loogiline ruut

Vaatleme lauseid, mis sisaldavad ainult ühte kvantorit. *Üldjaatav lause* väidab mingi omaduse olemasolu kõigil antud liiki objektidel. Üldjaatav on näiteks lause

- *Kõik inimesed naeratavad.*

Me saame lause ümber sõnastada:

> *Iga inimene on sellise omadusega, et ta naeratab.*
>
> *Iga objekt on selline, et kui ta on inimene, siis ta naeratab.*

Tähistades naeratamist väljendava predikaadi tähega N ja inimeseks olemise predikaadi tähega I, saab vaadeldava üldjaatava lause formaliseerida predikaatarvutuse valemina

$$\forall x\,(Ix \supset Nx).$$

Tuleb tähele panna, et implikatsiooni kasutamine ei ole siin sugugi juhuslik. Näiteks konjunktsiooni kasutamise korral on lausel

$$\forall x\,(Ix \,\&\, Nx).$$

esialgsest hoopis erinev mõte:

> *Iga objekt on selline, et ta on inimene ja ta naeratab.*

Me saame aga ette kujutada olukorda, kus objektide universum koosneb nii inimestest kui ka loomadest. Sellise näite korral on esialgne lause juhul, kui kõik inimesed tõepoolest naeratavad, tõene. Konjunktsiooni sisaldav lause on aga kindlasti väär, sest loomad võivad küll naeratada, kuid nad ei ole inimesed.

Üldjaatavast lausest on lihtne moodustada *üldeitav lause*. Selleks eitatakse kõigi antud liiki objektide korral vaadeldava omaduse kehtivust. Sellisel teel saame lause

> *Iga inimene on sellise omadusega, et ta ei naerata,*

mis on samaväärne lausega

- *Ükski inimene ei naerata*

ning see formaliseeritakse kujul

$$\forall x\,(Ix \supset \neg Nx).$$

Rääkides mitte kõigist antud liiki objektidest, vaid ainult osast objektidest, saab sõnastada vaadeldava lause *osajaataval* ja *osaeitaval* kujul:

- *Mõni inimene naeratab.*
- *Mõni inimene ei naerata.*

Sellised laused teisendatatakse predikaatarvutuse valemiteks, kasutades eksistentsikvantorit ja konjunktsiooni:

$$\exists x\,(Ix \,\&\, Nx),$$
$$\exists x\,(Ix \,\&\, \neg Nx).$$

Viimast lauset võib samuti vaadelda kui üldjaatava lause eitust:

$$\neg\forall x\,(Ix \supset Nx).$$

Predikaatarvutuse semantikareeglite abil saab aga näidata, et need formaliseeritud valemid on ekvivalentsed. Analoogiliselt võib üldeitavat lauset käsitleda kui osajaatava lause eitust:

$$\neg\exists x\,(Ix \,\&\, Nx).$$

Lause liik	Liigi tähis	Näidislause	Lause formalisatsioon
Üldjaatav	A	*Kõik inimesed naeratavad.*	$\forall x\,(Ix \supset Nx)$
Üldeitav	E	*Ükski inimene ei naerata.*	$\forall x\,(Ix \supset \neg Nx)$
Osajaatav	I	*Mõni inimene naeratab.*	$\exists x\,(Ix \,\&\, Nx)$
Osaeitav	O	*Mõni inimene ei naerata.*	$\exists x\,(Ix \,\&\, \neg Nx)$

Tabel 4.1. Kvantifitseeritud lausete liigitus.

	Jaatav lause	Eitav lause
Üldine lause	A: Iga I on N	E: Ükski I pole N
Osaline lause	I: Mõni I on N	O: Mõni I pole N

Tabel 4.2. Kvantifitseeritud lausete omadused.

Tabel 4.1 kirjeldab tähtsamad kvantifitseeritud lausete liigid ja annab nende levinumad tähistused. Tabelis 4.2 esitatakse kvantifitseeritud lausete põhilised omadused.

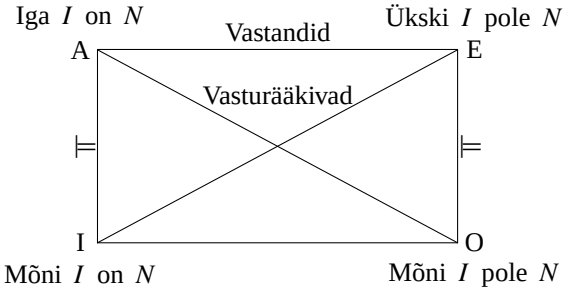
Olgu antud kaks suvalist lauset. Kui need laused ei saa olla korraga tõesed, siis me ütleme, et nad on teineteise *vastandid*. Vastandlikkuse erijuhuks on sellised laused, mille tõeväärtused on alati erinevad, nii et üks lause on teise lause eitus. Selliseid lauseid nimetatakse *vasturääkivateks* e. *kontradiktsioonideks*. Vasturääkivad on näiteks valemid

$$A \,\&\, B \quad \text{ja} \quad \neg(A \,\&\, B).$$

Valemid

$$A \,\&\, B \quad \text{ja} \quad \neg A \,\&\, \neg B$$

on vastandid, sest nad ei saa olla samaaegselt tõesed. Kui valem $A \,\&\, B$ on tõene, siis on A ja B mõlemad tõesed, mistõttu valem



Joonis 4.1. Loogiline ruut.

$\neg A \,\&\, \neg B$ on väär. Valem $\neg A \,\&\, \neg B$ on tõene, kui A ja B on mõlemad väärad. Kuid siis on $A \,\&\, B$ väär.

Me saame anda ka vastandlikkuse ja vasturääkivuse matemaatilised kirjeldused.

Lause 4.7. *Valemid p ja q on vastandid* $\Leftrightarrow$ $p \models \neg q$ *ja* $q \models \neg p$. *Valemid p ja q on vasturääkivad* $\Leftrightarrow$ $p = \neg q$.

Joonisel 4.1 toodud *loogilise ruudu* abil kirjeldatakse kvantifitseeritud lausete liikide vahelised põhilised loogilised seosed.

Ruudu langeva diagonaali lausete vasturääkivus annab meile seosed

$$A = \neg O$$
$$\text{ja} \qquad O = \neg A$$

ehk

$$\forall x\,(Ix \supset Nx) = \neg\exists x\,(Ix \,\&\, \neg Nx)$$
$$\text{ja} \qquad \exists x\,(Ix \,\&\, \neg Nx) = \neg\forall x\,(Ix \supset Nx).$$

Ruudu tõusva diagonaali otspunktide vasturääkivus annab võrdused

$$I = \neg E$$
$$\text{ja} \qquad E = \neg I$$

ehk lahtikirjutatult

$$\exists x\,(Ix \,\&\, Nx) = \neg\forall x\,(Ix \supset \neg Nx)$$

ja $$\forall x\,(Ix \supset \neg Nx) = \neg\exists x\,(Ix \,\&\, Nx).$$

Jättes kõrvale objektide filtreerimise (predikaadiga Ix), saame siit tuletatada kvantorite *duaalsusreeglid* (ülesanne 4.4.2, lk. 108).

Loogilise ruudu korral tuleb veel märkida, et A- ja E-liiki laused on teineteise vastandid, kuid nad pole omavahel vasturääkivad, sest nad võivad olla korraga väärad. I- ja O-liiki laused ei ole aga vastandid, sest nad ei välista teineteist. Ruudu külgedel asuvad laused on seotud loogilise järeldumise seose kaudu:

$$\mathrm{A} \models \mathrm{I},$$

$$\mathrm{E} \models \mathrm{O}.$$

Ülesanded

1. Tõestage A- ja E-liiki lausete vastandlikkus.
2. Tõestage, et $\exists x\,(Ix \,\&\, \neg Nx) = \neg\forall x\,(Ix \supset Nx)$.
3. Tõestage, et $\forall x\,(Ix \supset Nx) \models \exists x\,(Ix \& Nx)$ (kui Ix kehtib vähemalt ühe objekti x korral).

4.6. Süllogismid

Süllogismid on kahe eeldusega kehtivad arutlused. Süllogismi eeldustes on kolm mõistet, kusjuures üks mõistetest esineb mõlemas eelduses. Sõltuvalt mõistete omavahelisest paiknemisest eeldustes eristatakse nelja tüüpi *süllogismifiguure*. Näitame, kuidas predikaatarvutuse vahenditega saab hinnata süllogismide kehtivust.

Vaatleme arutlust

Kõik kassid on hallid.
Mõni koduloom on kass.

Mõni koduloom on hall.

Tähistame mõisted *kass*, *hall* ja *koduloom* vastavalt sümbolitega K, H ja L. Me saame kirjutada selle arutluse predikaatarvutuse keeles kujul

$$\frac{\forall x\,(Kx \supset Hx)\qquad \exists x\,(Lx \,\&\, Kx)}{\exists x\,(Lx \,\&\, Hx)}$$

See arutlus on kehtiv, sest teise eelduse põhjal leidub tema interpretatsiooni universumis niisugune objekt, mis muudab predikaadid Lx ja Kx on tõeseks. Arutluse esimene eeldus väidab aga, et antud objekti korral on ka Hx tõene.

Toome nüüd näite mittekehtivast süllogismist. Tähistame arutluses

Kõik pardid lendavad.
Kõik haned lendavad.

Kõik haned on pardid

mõisted *part*, *lendama* ja *hani* vastavalt sümbolitega P, L ja H. Tulemuseks on predikaatarvutuse arutlus

$$\frac{\forall x\,(Px \supset Lx)\qquad \forall x\,(Hx \supset Lx)}{\forall x\,(Hx \supset Px)}$$

See arutlus ei kehti, sest leidub niisugune interpretatsioon I (nn. *kontranäide*), mis lükkab tema kehtivuse ümber. Interpretatsiooni I universum koosneb kahest objektist: hanest Hans ja pardist Piilu. Sellisel juhul on arutluse eeldused tõesed, sest mõlemad linnud lendavad, aga järeldus on väär, sest Hans ei ole part.

4.7. Korduv kvantifitseerimine

Lausetes kasutatakse tihti mitut kvantorit. Näiteks lause

Keegi armastab kedagi

formaliseeritud kujus on kaks eksistensikvantorit: üks kvantor selle inimese kohta, kes armastab, ja teine kvantor inimese kohta, keda ta armastab:

$$\exists x \exists y\ Axy.$$

Kui saadav valem on esialgsega ekvivalentne, võib valemi ees olevad kvantorid omavahel ära vahetada. Toodud lauses on eksistentsikvantorite vahetus lubatud:

$$\exists y \exists x\ Axy.$$

Selles veendumiseks kujutame endale ette, et esialgne lause on interpretatsioonis I tõene. Siis kehtib mingite konstantide c_I ja d_I korral

$$(c_I, d_I) \in A_I.$$

Ent siis on ka vahetatud kvantoritega lause interpretatsioonis I tõene. Kui esialgne lause on interpretatsioonis I väär, siis on seda ka teine lause. Järelikult on esialgne lause loogiliselt ekvivalentne lausega

Kedagi armastatakse

ning me võime lause ees olevad kaks eksistentsikvantorit omavahel ära vahetada.

Sama kehtib ka üldisuskvantori korral, sest kahe üldisuskvantoriga algavad laused on kas samaaegselt tõesed või samaaegselt väärad:

$$\forall x \forall y\ p = \forall y \forall x\ p.$$

Selgub aga, et kahte eri tüüpi kvantorit ei tohi omavahel ära vahetada:

$$\forall x \exists y\ Exy \neq \exists y \forall x\ Exy.$$

Selleks vaatleme interpretatsiooni, mille universumiks on kõigi inimeste hulk ja predikaat Exy tähendab "x ema on y". Vasakpoolne lause on selles interpretatsioonis tõene, sest igal inimesel on ema, parempoolne lause on aga väär, sest ükski inimene pole kõigi inimeste ema. Erinevat tüüpi kvantorite korral kehtib loogiline järeldumine:

$$\exists y \forall x\ p \models \forall x \exists y\ p.$$

Seega võib üldisuskvantori tõsta eksistentsikvantori ette.

Ülesanded

1. Näidake, et üldisuskvantori võib tõsta eksistentsikvantori ette:

$$\exists y \forall x\ Axy \models \forall x \exists y\ Axy.$$

2. Tõlkige predikaatarvutuse keelde:
   (a) *Kõik vastasid kõigile küsimustele.
   (b) Mitte keegi ei vastanud kõigile küsimustele.
   (c) Vastati igale küsimusele.
   (d) Igaüks vastas vähemalt ühele küsimusele.
   (e) *Mõned ei vastanud ühelegi küsimusele.
   (f) Kõik peale Fredi vastasid vähemalt ühele küsimusele.

4.8. Signatuur ja interpretatsioon. Struktuuri ja struktuuride klassi teooria

Kui analüüsida matemaatiliste teooriate kirjapanemise keelt, siis näeme, et lihtsamal juhul piisab meie vajadusteks paragrahvis 4.4 esitatud *signatuuri* mõiste täiendamisest veel ühe sümbolite kategooria võrra. Predikaat- ja konstantsümbolitele tuleks lisada sümbolid antud teoorias uuritavate funktsioonide tähistamiseks — nn. *funktsionaalsümbolid*. Esimest järku keele *signatuur* ongi kolmest sümbolite hulgast koosnev kolmik $\langle C, F, P\rangle$. Näiteks aritmeetika esimest järku teooria nn. formaalne aritmeetika kasutab tavaliselt signatuuri $\sigma = \langle 0; ', +, \cdot; =\rangle$, kus sümbol $'$ tähistab järgmise arvu leidmise funktsiooni $f(x) = x + 1$. Muidugi eeldatakse, et

signatuuri defineerimisel fikseeritakse iga funktsionaalsümboli ja predikaatsümboli jaoks ka argumentide arv. Aritmeetika signatuuris on funktsionaalsümbol $'$ ühekohaline, teised funktsionaalsümbolid ja predikaatsümbol $=$ kahekohalised. Kõik kolm sümbolite hulka võivad üldiselt olla nii lõplikud kui lõpmatud. Konstant- ja funktsionaalsümbolite hulgad võivad mõnikord olla ka tühjad. Predikaatsümbolite hulk peab sisaldama vähemalt ühe elemendi, et oleks võimalik moodustada valemeid.

Valemi mõiste defineerimisel skeem eriti ei muutu. Ainult predikaatide argumentide kohale lubatakse lisaks konstantidele ja muutujatele nüüd kirjutada ka terme: muutujatest ja konstantsümbolitest funktsionaalsümbolite abil moodustatud avaldisi. Nii saab formaalses aritmeetikas näiteks moodustada valemi

$$\forall x \exists y \exists z\, [x + 0''' = y \,\&\, x + y' = z].$$

Tavaliselt konstrueeritakse esimest järku keele signatuur mingist vaatluse all olevast matemaatilisest struktuurist lähtudes nii, et ta sisaldaks teoorias formuleeritavate väidete jaoks vajalikke tähiseid. Sealjuures võidakse ka signatuur defineerida formaalselt kitsamana ja toetuda sellele, et mõnda predikaati saab väljendada signatuuri sümbolite kaudu. Näiteks kui universumiks on naturaalarvude hulk $\mathbb{N} = \{0, 1, \ldots\}$, siis saame predikaadi $x < y$ väljendada valemiga

$$\neg(x = y) \,\&\, \exists z\, (y = x + z)$$

ülaltoodud signatuuris, predikaadi $x - y = z$ aga valemiga $x = y + z$. Nii võime vältida suure hulga sümbolite lisamist ja saavutada, et signatuur ja valemite hulk antud keeles oleks teoreetilise uurimistöö jaoks piisavalt kompaktne.

Edasi on selge, et ühe ja sama keele abil on võimalik rääkida erinevatest struktuuridest. Näiteks sellesama aritmeetika signatuuri valemitega võime väljendada ka fakte täisarvude, ratsionaalarvude või reaalarvude universumi kohta. Algebras on ka kombeks seada ühele ja samale tehtemärgile vastavusse erinevaid (isegi samal hulgal defineeritud) operatsioone. Muidugi võib universumi või signatuuri sümbolite tähenduse varieerimine kaasa tuua seda, et

mingi valem on ühes interpreteeringus tõene ja teises väär. Naturaalarvudel väär valem $\forall x \exists y\, [x = y']$ on täisarvudel ilmselt tõene. Kokku järeldame aga siit, et võib osutuda otstarbekaks ajaloolisele vastupidine liikumissuund, kus fikseeritakse keel, vaadeldakse selle keele valemeid eri struktuuridel ja saadakse vastavalt valemite tõesusele või väärusele nende struktuuride teooriad.

Definitsioon 4.8. Olgu $\sigma = \langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ signatuur. Signatuuri *interpretatsiooniks* nimetatakse paari $\langle M, I_\sigma \rangle$, kus M on suvaline mittetühi hulk (*interpretatsiooni kandja*), mida mõistetakse indiviidide piirkonnana, ja I_σ on interpreteeriv kujutus, mis seab

1) igale konstantsümbolile $c \in \mathcal{C}$ vastavusse elemendi $\overline{c} \in M$;

2) igale n-kohalisele funktsionaalsümbolile $f^{(n)} \in \mathcal{F}$ vastavusse hulgal M defineeritud n muutuja funktsiooni $\overline{f}\colon M^n \to M$;

3) igale n-kohalisele predikaatsümbolile $P \in \mathcal{P}$ vastavusse hulgal M defineeritud n-kohalise predikaadi $\overline{P}\colon M^n \to \{t, v\}$, kusjuures võrdusmärgile seatakse vastavusse võrduspredikaat.

Näide 4.9. Olgu $\sigma = \langle 0, 1, 2, \ldots; +, \cdot; =, < \rangle$. Anname sellele signatuurile kolm interpretatsiooni:

a) Olgu $M = \mathbb{N}$ ja signatuuri sümbolid interpreteeritud standardsel viisil;

b) Olgu $M = \mathbb{R}$ ja signatuuri sümbolid interpreteeritud standardsel viisil;

c) Vaatleme Boole'i interpretatsiooni. Olgu $M = \{0, 1\}$. Interpreteerime signatuuri sümboleid järgmiselt:

$$\overline{c} = \begin{cases} 0, & \text{kui } c \text{ on paarisarv,} \\ 1, & \text{kui } c \text{ on paaritu arv,} \end{cases}$$

korrutamismärki mõistame tavalisel viisil, märki $+$ aga kui liitmist arvu 2 jäägiklassiringis, s.t.

$+$	0	1
0	0	1
1	1	0

(tavalist liitmist ei saa siin kasutada, kuna M ei ole liitmise suhtes kinnine hulk); predikaatsümboleid $=$ ja $<$ mõistame tavalisel viisil.

Ülesanded

1. Leida järgmiste valemite tõeväärtused igaühes kolmest eelmises näites vaadeldud interpretatsioonist:
   (a) $0 = 2$;
   (b) $1 + 2 = 3$;
   (c) $1 + 5 = 2$;
   (d) $\forall x (x < x + 1)$;
   (e) $\forall x \exists y (x < y)$;
   (f) $\forall x \exists y (y < x)$;
   (g) $\forall x \forall y [x < y \supset \exists z (x < z \,\&\, z < y)]$;
   (h) $\forall x \forall y \forall z (x + y = x + z \supset y = z)$;
   (i) $\forall x \forall y \exists z (x + z = y)$.

Olgu M struktuur signatuuris σ. Struktuuri M esimest järku teooriaks (M elementaarteooriaks) nimetatakse kõigi sellel struktuuril tõeste σ valemite hulka:

$$\mathrm{Th}\,(M) = \{\, A \mid A \text{ on } \sigma \text{ valem ja } M \models A \,\}.$$

Näiteks ülaltoodud valem $\forall x \exists y \exists z\, [x + 0''' = y \,\&\, x + y' = z]$ kuulub naturaalarvude esimest järku teooriasse, aga valem $\forall x \exists y\, [x = y']$ ei kuulu, sest $x = 0$ jaoks sellist y väärtust ei leidu.

Struktuure $M1$ ja $M2$ nimetatakse *elementaarselt ekvivalentseteks*, kui nende elementaarteooriad ühtivad. On küllalt ilmne, et isomorfism on selleks piisav tingimus. Tarvilik ta aga ei ole.

Ka esimest järku keelte jaoks on tähtsal kohal samaselt tõesed valemid ja valemite samaväärsus. *Samaselt tõesteks* nimetatakse valemeid, mis on tõesed oma signatuuri kõigis interpretatsioonides (vabade muutujate kõigil väärtustel, kui valem pole kinnine). Valemeid A ja B signatuuris σ nimetatakse *loogiliselt samaväärseteks*, kui kõigis signatuuri σ interpretatsioonides on A tõeväärtus võrdne B tõeväärtusega.

On selge, et mingit otse definitsioonil põhinevat samaselt tõesuse või loogilise samaväärsuse kontrolli algoritmi (nagu lausearvutuses) siin olla ei saa, sest tuleks kontrollida lõpmata palju

interpretatsioone. Osal lihtsatel juhtudel saab küll näidata, et piisab lõpliku hulga interpretatsioonide vaatlemisest. Aga see on nii ainult mõne väga piiratud signatuuri puhul, näiteks ainult võrdusmärgist koosnev signatuur ja signatuurid, mis koosnevad ainult unaarsetest predikaatsümbolitest.

Matemaatikas on oluline ka üks vahepealse üldisuse astmega tõesuse mõiste. Tihti uuritakse mitte ühte struktuuri, vaid mingit struktuuride klassi ja eesmärgiks on tõestada väiteid, mis kehtivad igas sellesse klassi kuuluvas struktuuris. Nii tehakse näiteks algebras ja funktsionaalanalüüsis, vaadeldes poolrühmi, rühmi, ringe, korpusi, vektorruume, meetrilisi või topoloogilisi ruume jne.

Olgu fikseeritud mingi signatuur σ ja olgu K mingi struktuuride hulk, mis koosneb struktuuridest signatuuris σ. Struktuuride klassi K *elementaarteooriaks* nimetame signatuuri σ kõigi selliste valemite hulka, mis on tõesed igal klassi K kuuluval struktuuril.

Näiteks rühmateooria jaoks on tihti kasutusel signatuur $\langle e; \cdot; = \rangle$, kus sümbol e tähistab rühma ühikelementi ja $\cdot$ (korrutamis-) tehet. Rühmateooriasse kuuluvad valemid, mis on tõesed igal rühmal. Näiteks valem

$$\forall x \forall y\, [\forall u\, [x \cdot u = u] \,\&\, \forall u\, [u \cdot y = u] \supset x = y]$$

(vasak- ja parempoolse ühikelemendi ühtimine) kuulub teooriasse, sest ta kehtib igas rühmas, aga kommutatiivsuse aksioom

$$\forall x \forall y\, [x \cdot y = y \cdot x]$$

ei kuulu, sest on nii kommutatiivseid (Abeli rühmi) kui ka mittekommutatiivseid rühmi.

Palju matemaatikas uuritavaid struktuuride klasse on muidugi ise esimest järku keele vahenditega (aksiomaatiliselt) defineeritud, aga see pole elementaarteooria mõiste jaoks oluline. Näiteks võib klass K koosneda ka kõigist lõpliku kandjaga struktuuridest signatuuris σ.

Matemaatikuid huvitab, milliste meetoditega saab kindlaks teha, kas mingi valem on antud struktuuril või struktuuride klassil tõene või väär. Sellele probleemile võib leiduda kahte sorti positiivne lahendus. Lihtsamal juhul võib struktuuri (struktuu-

ride klassi) teooria *lahenduda* algoritmiliselt, s.t. leidub algoritm, millega saab kontrollida suvalise valemi tõesust sellel struktuuril (struktuuride klassis). Lahenduvad näiteks iga Boole'i algebra (sealhulgas ka suvalise hulga alamhulkade algebra) teooria, kõigi Boole'i algebrate, Abeli rühmade jpm. teooriad. Mitu teist teooriat aga ei lahendu, kusjuures algebralises mõttes ei tarvitse erinevus "lähimast" lahenduvast juhust olla suur: rühmateooria, distributiivsete võrede teooria.

Teiseks võimalikuks positiivseks lahenduseks on teooria aksiomatiseeritavus (täieliku ja korrektse aksiomaatikaga, kus on tuletatavad parajasti teooriasse kuuluvad valemid). Kui teooria on aksiomatiseeritav, siis saab ühelt poolt iga teooriasse kuuluva valemi jaoks konstrueerida nendest aksioomidest lähtuva "mehhaaniliselt kontrollitava" tõestuse. Teiselt poolt on aga võimalik ka algoritm, mis genereerib kõikvõimalikud tõestused selles süsteemis, s.t. teoreemide leidmine on automatiseeritav (vähemalt põhimõtteliselt, s.t. arvutusaja kulu arvestamata).

Kui tegemist on ühe struktuuri teooriaga, siis need kaks juhtu ühtivad. Iga kinnine valem on vaadeldaval struktuuril kas tõene või väär. Viimasel juhul on muidugi tema eitus tõene. Valemi tõeväärtuse leidmiseks võime siis käivitada tõestuste generaatori ja oodata, kas saame uuritava valemi või tema eituse tõestuse. Kui aga on aksiomatiseeritud mingi struktuuride klassi teooria, siis generaator annab meile otsuse ainult nende valemite korral, mis on kõigil vaadeldavatel struktuuridel tõesed või kõigil väärad. Aga tavaliselt pole uuritavasse klassi kuuluvad struktuurid kõik elementaarselt ekvivalentsed ja leidub valemeid, mis mõnel struktuuril on tõesed ja mõnel väärad (nagu kommutatiivsuse seadus rühmadel). Sellise valemi kohta me nii vastust ei saa.

Ülesanded

2. Olgu interpretatsiooni kandja $M = \mathbb{N}$ ja signatuur $\sigma = \langle 0, 1; +, \cdot; = \rangle$. Väljendada predikaadid:
   (a) $x = 2$;
   (b) $y = 4x + 3$;
   (c) $z = x/y$;
   (d) $x \leq y$;
   (e) x on paarisarv;
   (f) x jagub arvuga y;
   (g) x on täisruut;
   (h) z on x/y täisosa.

4.9. Lõplikud struktuurid

Uurime kõigepealt pisut lõplikke struktuure. Selgub, et küllalt loomulikel eeldustel on siin elementaarteooria lahenduv.

Teoreem 4.10. *Olgu σ lõplik signatuur ja $\mathcal{M}$ lõpliku kandjaga struktuur signatuuris σ. Siis struktuuri $\mathcal{M}$ elementaarteooria on lahenduv.*

Tõestus. Olgu $\sigma = \langle C, F, P \rangle$, kus

$$C = \{c_1, \ldots, c_k\},$$
$$F = \{f_1, \ldots, f_l\},$$
$$P = \{P_1, \ldots, P_m\}.$$

Olgu interpretatsiooni kandjaks hulk $M = \{m_1, \ldots, m_n\}$ ja signatuuri sümbolid interpreteeritud hulga M elementidega $c_1, \ldots, c_k$, hulgal M defineeritud funktsioonidega $f_1, \ldots, f_l$ ja predikaatidega $P_1, \ldots, P_m$. Funktsioonid f_i ja predikaadid P_j on lõpliku määramispiirkonnaga ja neid saab esitada lõplike tabelitena.

Olgu A kinnine valem signatuuris σ. Tema tõeväärtuse leidmiseks asendame A kvantoriteta valemiga laiendatud keeles, kus σ sümbolitele on lisatud tähised hulga M elementide $m_1, \ldots, m_n$ tähistamiseks. Selleks kirjutame seestpoolt alates iga osavalemi $\forall x\, B(x)$ asemele konjunktsiooni $B(m_1)\ \&\ \ldots\ \&\ B(m_n)$ ja iga osavalemi $\exists x\, B(x)$ asemele disjunktsiooni $B(m_1) \vee \ldots \vee B(m_n)$. Saadud valemis pole enam ka muutujaid, s.t. termid on koostatud ainult M elementide tähistest. Edasi arvutame funktsioonide tabelite järgi kõigi termide väärtused, predikaatide tabelite järgi kõigi

atomaarsete valemite tõeväärtused ja lõpuks saame lausearvutuse tehete tulemusena kogu valemi tõeväärtuse. □

Ülesanded

1. Kui signatuur on lõpmatu (sisaldades näiteks iga $n \geq 1$ jaoks ühe n muutuja predikaadi), siis võib struktuuri esimest järku teooria universumi lõplikkusest hoolimata olla mittelahenduv. Konstrueerida selline signatuur ja tema mittelahenduva teooriaga interpretatsioon.

Teoreem 4.11. *Kui signatuur sisaldab võrdusmärki, siis iga $k \geq 1$ jaoks leidub*

1) valem, mis on tõene parajasti nendes interpretatsioonides, mille kandjas on vähemalt k elementi;

2) valem, mis on tõene parajasti nendes interpretatsioonides, mille kandjas pole rohkem kui k elementi;

3) valem, mis on tõene parajasti nendes interpretatsioonides, mille kandjas on täpselt k elementi.

Tõestus. Esimest tingimust rahuldab valem

$$\exists x_1 \ldots \exists x_k \, [\& \neg x_i = x_j],$$

teist tingimust valem

$$\exists x_1 \ldots \exists x_k \forall y \, [y = x_1 \vee \ldots \vee y = x_k],$$

kolmandat nende konjunktsioon. □

Lõplike struktuuride esimest järku teooriate lahenduvus võib meid viia mõttele kasutada valemite samaselt tõesuse kontrollimiseks kontrollimist lõplikel struktuuridel. Järgmine teoreem näitab, et sellest ei piisa.

Teoreem 4.12. *Leidub kehtestatav valem, mis on igal lõplikul struktuuril väär.*

Tõestus. Vaatleme valemit

$$A = \forall x \neg P(x, x) \,\&\, \forall x \forall y \forall z \, [P(x, y) \,\&\, P(y, z) \supset P(x, z)]$$
$$\&\, \forall x \exists y \, P(x, y).$$

See valem on kehtestatav, sest ta on tõene, kui võtta interpretatsiooni kandjaks naturaalarvude hulk ja mõista $P(x, y)$ kui $x < y$. Oletame vastuväiteliselt, et A on tõene lõplikul hulgal M, mis sisaldab n elementi. Olgu $m_1 \in M$. Kasutades konjunktsiooni kolmanda liikme tõesust, saame konstrueerida jada $m_1, m_2, \ldots, m_{n+1}$, nii et kehtib $P(m_1, m_2), P(m_2, m_3), \ldots, P(m_n, m_{n+1})$. Konjunktsiooni teise liikme (transitiivsuse) tõttu kehtib

$$i < j \Rightarrow P(m_i, m_j)$$

ja järelikult konjunktsiooni esimese liikme tõttu

$$i < j \Rightarrow m_i \neq m_j,$$

s.t. elemendid $m_1, \ldots, m_{n+1}$ peavad kõik olema erinevad, mis pole M elementide arvu tõttu võimalik. □

4.10. Reaalarvude järjestuse elementaarteooria

Toome siin ühe pisut tõsisema näite elementaarteooria lahenduvuse kohta. Eelmise paragrahvi esimese teoreemi tõestasime uuritavast valemist kvantorite elimineerimise teel. Põhiidee on sama ka siin, aga elimineerimise mehhanism oluliselt keerulisem.

Vaatleme signatuuri, mis koosneb kahest kahekohalisest predikaatsümbolist $=$ ja $<$. Kui võtta interpretatsiooni kandjaks reaalarvude hulk ja interpreteerida sümbolit $<$ tavalisel viisil, moodustab tõeste valemite hulk reaalarvude järjestuse elementaarteooria.

Teoreem 4.13. *Reaalarvude järjestuse elementaarteooria on lahenduv.*

Tõestus. Märgime kõigepealt, et vaadeldavas interpretatsioonis on tõesed järgmised valemid:

(1) $\forall x \neg (x < x)$ — range järjestuse aksioomid,
(2) $\forall x \forall y \forall z\, (x < y \,\&\, y < z \supset x < z)$,
(3) $\forall x \forall y\, (x < y \vee x = y \vee y < x)$ — järjestuse lineaarsus,
(4) $\forall x \forall y\, (x < y \supset \exists z\, (x < z \,\&\, z < y)$ — tihedus,
(5) $\forall x \exists y \exists z\, (y < x \,\&\, x < z)$ — pole esimest ega viimast elementi.
Konstrueerime nüüd algoritmi kinniste valemite tõesuse kontrolliks.

1. etapil elimineerime tehtemärgid $\supset$ ja $\equiv$, avaldades need eituse, disjunktsiooni ja konjunktsiooni kaudu.

2. etapil elimineerime üldisuskvantorid, avaldades need eituse ja olemasolukvantori kaudu.

3. etapil elimineerime saadud valemist olemasolukvantorid. Seda võimaldab järgmine lemma.

Lemma 4.14. *Iga osavalemi $\exists x_0\, B$, kus B on kvantoriteta valem, võime asendada sama tõeväärtusega kvantoriteta valemiga.*

Näitame, kuidas saada selline kvantoriteta valem.

3a. Viime valemis B eitused vahetult atomaarsete valemite ette (läbi konjunktsioonide ja disjunktsioonide).

3b. Atomaarsete valemite eitused asendame (3) põhjal disjunktsioonidega, kirjutades

$$\neg x = y \quad \text{asemele} \quad x < y \vee y < x,$$
$$\neg x < y \quad \text{asemele} \quad x = y \vee y < x.$$

3c. Järelejäänud kvantorialune valem sisaldab ainult & ja $\vee$. Disjunktsioone läbi korrutades saame atomaarsete valemite konjunktsioonide disjunktsiooni, s.t.

$$B = t \Leftrightarrow \bigvee_{i=1}^{k} B_i.$$

3d. Disjunktsiooni saame olemasolukvantori alt välja tuua:

$$\exists x_0 \bigvee_{i=1}^{k} B_i(x_0) = \bigvee_{i=1}^{k} \exists x_0\, B_i(x_0).$$

3e. Elimineerime kvantori igast valemist $\exists x_0\, B_i(x_0)$ eraldi. Kõigepealt saame kvantori alt välja tuua need konjunktsiooni B_i liikmed, mis ei sisalda muutujat x_0:

$$\exists x_0\, B_i(x_0) = C_i \,\&\, \exists x_0\, D_i(x_0).$$

3f. Vaatleme $\exists x_0\, D_i(x_0)$. Siin $D_i(x_0)$ on konjunktsioon atomaarsetest valemitest kujul $x_0 < x_j$, $x_0 = x_j$ ja $x_j < x_0$.

3f1. Tõesed liikmed $x_0 = x_0$ võime välja jätta. Kui leidub liige $x_0 < x_0$, siis on konjunktsioon kohe väär.

3f2. Kui konjunktsioonis leidub liige kujul $x_0 = x_j$, kus $j \neq 0$, siis $\exists x_0\, D_i(x_0) = D_i(x_j)$ ja kvantor on elimineeritud.

3f3. Kui konjunktsioonis on ainult ühesuunalisi võrratusi, s.o. $D_i(x_0)$ on kujul

$$\&_j\, (x_j < x_0) \quad \text{või kujul} \quad \&_j\, (x_0 < x_j),$$

siis on $\exists x_0\, D_i(x_0)$ valemi (5) põhjal tõene.

3f4. Kui konjunktsioonis leidub nii liikmeid kujul $x_j < x_0$ kui ka kujul $x_0 < x_k$, siis väidab $\exists x_0\, D_i(x_0)$, et x_0 saab paigutada kõigist elementidest x_j paremale ja kõigist elementidest x_k vasakule. Tiheduse (4) tõttu on see võimalik parajasti siis, kui on tõene valem

$$\&_{\{\, j,k \,|\, \text{leidub liige } x_j < x_0 \text{ ja leidub liige } x_0 < x_k \,\}}\, (x_j < x_k),$$

s.t. võime $\exists x_0\, D_i(x_0)$ asendada selle valemiga, kus pole kvantorit ega x_0.

Sellega on lemma tõestatud ja saame seestpoolt väljapoole elimineerida kõik kvantorid. □

Järeldus 4.15. *(teoreemist ja tõestusest). Ratsionaalarvude järjestuse teooria on elementaarselt ekvivalentne reaalarvude järjestuse teooriaga ja samuti lahenduv.*

Tõestuses me ei kasutanud mingeid muid reaalarvude järjestuse struktuuri omadusi kui ainult valemite (1)–(5) tõesust. Aga need valemid kehtivad ka ratsionaalarvude järjestuse kohta ja seega

arvutab konstrueeritud algoritm samuti valemi tõeväärtuse ratsionaalarvudel. Muuhulgas oleme saanud näite mitteisomorfsetest elementaarstruktuuridest. Üksühest vastavust nende struktuuride vahel ei saa olla sel põhjusel, et ratsionaalarvude hulk on loenduv, aga reaalarvude hulk kontiinumi võimsusega.

Järeldus 4.16. *Valemid (1)–(5) kujutavad endast täielikku aksiomaatikat, sest nende kehtimisest järeldub antud signatuuri iga valemi tõeväärtus.*

4.11. Ülesanded

1. Näidake arutluste kehtivust:

   (a) $\forall x\,\neg(Fx \,\&\, Gx)$
   Fa

   $\neg Ga$

   (b) $\neg\exists x\,(Px \,\&\, Qx)$
   $\exists x\,(Px \,\&\, Rx)$

   $\exists x\,(Rx \,\&\, \neg Qx)$

   (c) $\forall x\,(Px \supset Qx)$
   Pa
   Ra

   $\exists x\,(Rx \,\&\, Qx)$

2. (C. H. Papadimitriou) Olgu meil kasutada binaarne predikaat $canfool(p,t)$ (p-d saab narritada ajal t). Abraham Lincoln on ütelnud

   > You can fool some people all of the time, and all of the people some of the time, but you cannot fool all of the people all of the time.

   (Te saate alati narritada mõnda inimest ja mõnikord kõiki inimesi, kuid te ei saa alati narritada kõiki inimesi.)

   (a) Formaliseerige Lincolni ütluse pessimistlik variant (leidub inimene, keda saab alati narritada) ja optimistlik variant (sellist inimest ei leidu, kuid alati on olemas inimene, keda saab narritada).

   (b) Tõestage, et üks nendest variantidest järeldub teisest.

3. Formaliseerige koolimatemaatika väited. (Allikad: III, IV ja V klassi õpikud, 1987–1990.)

   (a) *Summa ei muutu, kui muudame liidetavate järjekorda.

   (b) *Kui summast lahutame ühe liidetava, siis saame teise liidetava.

   (c) Korrutis ei muutu, kui muudame tegurite järjekorda.

   (d) *Kui üks tegur on võrdne 1-ga, siis korrutis on võrdne teise teguriga.

   (e) Kui üks tegur on võrdne nulliga, siis korrutis on võrdne nulliga.

   (f) Paarisarvule järgneb ja eelneb paaritu arv.

   (g) Summa korrutamiseks arvuga võime korrutada liidetavad eraldi ja korrutised liita.

   (h) Selle asemel, et lahutada arvust summa, võin enne lahutada ühe liidetava ja siis teise liidetava.

   (i) Kui arvu ristsumma jagub 3-ga, siis arv jagub 3-ga, vastasel juhul mitte.

4. Formaliseerige koolimatemaatika väited. (Allikas: V klassi õpik. Nurk, Telgmaa, 1990.)

   (a) Läbi iga kahe punkti võib tõmmata sirge ja sealjuures ainult üks kord.

   (b) Läbi antud punkti saab antud sirgele joonistada ainult ühe ristsirge.

   (c) Kui kaks sirget tasandil on risti ühe ja sama sirgega, siis need kaks sirget on paralleelsed.

(d) (Paralleelide aksioom.) Väljaspool sirget asetsevat punkti läbib ainult üks sirge, mis on paralleelne antud sirgega.

5. Väljendada predikaadid signatuuris $\sigma = \langle 0, 1; +, \cdot; = \rangle$ eeldusel, et universumiks on reaalarvude hulk $\mathbb{R}$.

   (a) $x \geq 0$;
   (b) $x \leq y$;
   (c) $x < y$;
   (d) $x \in \mathbb{N}$.

6. Väljendada predikaadid signatuuris $\sigma = \langle ; ; =, \subseteq \rangle$ eeldusel, et universumiks on naturaalarvude hulga kõigi osahulkade hulk $\mathcal{P}(\mathbb{N})$.

   (a) $X = \emptyset$;
   (b) $X = \mathbb{N}$;
   (c) $X \cup Y = Z$;
   (d) $X \cap Y = Z$;
   (e) $X \cup (Y \cap Z) = V$;
   (f) $X \cap Y = \emptyset$;
   (g) $\overline{X} = Y$;
   (h) $X \setminus Y = Z$;
   (i) $|X| = 1$;
   (j) $|X| = 2$.

7. Väljendada predikaadid signatuuris $\sigma = \langle ; \cup, \cap ; = \rangle$ eeldusel, et universumiks on naturaalarvude hulga kõigi osahulkade hulk $\mathcal{P}(\mathbb{N})$.

   (a) $X = \emptyset$;
   (b) $X = \mathbb{N}$;
   (c) $X \subseteq Y$;
   (d) $\overline{X} = Y$;

   (e) $X \setminus Y = Z$.

8. Olgu fikseeritud hulk M ja olgu $A, B, C \subseteq M$, $D \subseteq M \times M$. Tähistagu $A(x)$, $B(x)$ ja $C(x)$ vastavalt $x \in A$, $x \in B$ ja $x \in C$, $D(x, y)$ aga $(x, y) \in D$. Signatuur koosnegu predikaatsümbolitest A, B, C, D ja võrdusmärgist. Väljendada:

   (a) $A = B$;
   (b) $A \subseteq B$;
   (c) $A \subset B$;
   (d) $A \cup B = C$;
   (e) $A \cap B = C$;
   (f) $A \setminus B = C$;
   (g) $\overline{A} = B$;
   (h) $A = \emptyset$;
   (i) $A = M$;
   (j) $A \cap B = \emptyset$;
   (k) $|A| = 1$;
   (l) $|A| = 2$;
   (m) $A \times B \subseteq D$;
   (n) $D \subseteq A \times B$;
   (o) $A \times B = D$.

9. Selgitage, kuidas saab predikaatarvutusse tuua kvantorid *vähe*, *palju* ja *enamus*.

5. Klassikalised tuletusreeglid

> *Siin on selge, et kuri on karjas, kui pildis oleva ebareeglipärasuse leidmine põhineb automatiseerunud seosteahela veatul käivitumisel.*
>
> — Andreas Walden

> *Bentham oskas asju koost lahti võtta, ta ei uskunud üldistusi, igal juhul mitte enne, kui nad tema meetodi abil tõestust polnud leidnud. Tema eesmärk, konstrueerida selline keel, mis oleks oma lihtsuses ja läbipaistvuses arusaadav kõigile, jäi küll saavutamata, kuid selle poole püüdmine tundub tänapäeval lausa kangelaslik.*
>
> — Toomas Raudam

Arutluste kehtivuse kontrollimiseks on palju võimalusi. Eelnevalt nägime, kuidas seda saab teha kõigi variantide läbivaatamise teel. Lausearvutuse korral on selleks süstemaatiline meetod — tõeväärtustabelid. Nende koostamine on aga väga töömahukas. Kui arutlus sisaldab 3 lausemuutujat, siis tuleb tõeväärtustabelis täita 8 rida. n muutuja korral on aga vaja täita 2^n rida. Mõnikord aitab hädast välja kiirkontrolli meetod, kuid arutluste kehtivuse kontrollimine on halvimal juhul ikka eksponentsiaalse ajakeerukusega. Me ütleme sellisel juhul, et "lausearvutuse lausete loogilise tõesuse kontrollimise probleem *lahendub raskesti*".

Alonzo Church (1903–1995) tõestas, et predikaatarvutuse korral on analoogiline probleem mitte raskesti lahenduv, vaid üldse mittelahenduv. Sellepärast kasutatakse arutluste kehtivuse kontrollimiseks meetodeid, mille abil on lootust lahendada probleem reaalse ajaga. Kasutatakse mitmesuguseid formaalseid tuletussüsteeme, millest efektiivsemaks peetakse tõesuspuude meetodit ja resolutsioonimeetodit. Paljud eelistavad aga nende asemel mõne loomuliku tuletuse süsteemi kasutamist.

5.1. Tuletusreeglid

Suur osa tuletusreeglitest on pärit Aristotelese loogikast. Seepärast kasutatakse nende reeglite tähistamiseks sageli ladinakeelseid nimetusi.

Vaatleme näiteks lausearvutuses arutlust

> *Kui Anul on hea tuju, siis on Peeter õnnelik.*
>
> *Kui Peeter on õnnelik, siis Anu ja Peeter mängivad kabet või joovad teed.*
>
> *Anul on hea tuju, kuid Anu ja Peeter ei mängi kabet.*
>
> ---
>
> *Anu ja Peeter joovad koos teed.*

Selle arutluse saab formaalselt kirja panna kujul

$$\begin{array}{ll} 1. & A \supset P \\ 2. & P \supset (K \vee T) \\ 3. & A \,\&\, \neg K \\ \hline \vdash & T \end{array}$$

kus lausearvutuse valemid on nummerdatud ja lausemuutujad on tähistatud vastavate sõnade esitähtedega. Selle arutluse korral on lihtne näha, miks ta kehtib:

4.	A	(3. lause põhjal)	*Anul on hea tuju.*
5.	P	(1. ja 4. põhjal)	*Peeter on õnnelik.*
6.	$K \vee T$	(2,5)	*Anu ja Peeter mängivad kabet või joovad teed.*
7.	$\neg K$	(3)	*Anu ja Peeter ei mängi kabet.*
8.	T	(6,7)	*Anu ja Peeter joovad koos teed.*

Arutluse kehtivuse kontrollimiseks oleks tulnud täita 16-realine tõeväärtustabel. Praegu piisas aga 8-realisest tuletusest. Me kasutasime selles näites mitut *tuletusreeglit*. Neljanda ja seitsmenda lause saamiseks kasutasime *lihtsustamisreeglit*

$$\frac{p \,\&\, q}{p} \quad \text{või} \quad \frac{p \,\&\, q}{q} \quad (\textit{Simp}).$$

Viienda ja kuuenda lause saamiseks kasutasime reeglit *modus ponens*:

$$\frac{p \supset q,\ p}{q} \quad (\textit{MP}).$$

Kaheksanda valemi saime aga *disjunktiivse süllogismi reegli* abil:

$$\frac{p \vee q, \neg p}{q} \quad \text{või} \quad \frac{p \vee q, \neg q}{p} \quad (\textit{DS}).$$

Siit selguvad tuletustele iseloomulikud jooned. *Tuletus* on lausete jada, kus iga lause on kas eeldus või saadud temale jadas eelnevatest lausetest mingi tuletusreegli abil. Kui eelduste hulgast Γ saab tuletada järelduse p, siis pannakse see lühidalt kirja kujul

$$\Gamma \vdash p.$$

Nimetame ka tuletusreeglite tähtsamad omadused:

- *Korrektsus*. Iga tuletusreegel on kehtiv arutlusvorm, s.t. ta säilitab oma argumentide tõesust.

- *Kompaktsus*. Reegleid ei tohi olla liiga palju. Neid peab olema lihtne meelde jätta.
- *Tuletussüsteemi täielikkus*. Reeglite abil peab saama tuletada kõik kehtivad (lausearvutuse või predikaatarvutuse) arutlused.

Tuletussüsteemi nimetatakse vahel ka *formaalseks aksiomaatiliseks süsteemiks* või lihtsalt *arvutuseks*. Me oleme siiamaani nimetanud lauseloogikat ja predikaatloogikat samuti arvutusteks, silmas pidades nende süsteemide algebralisi omadusi, mis võimaldavad tunnistada teatavad laused omavahel ekvivalentseteks ning sooritada lausete samasusteisendusi ja asendusi.

Esimene teadaolev tuletussüsteem on pärit Eukleideselt (u. 365–300 e.m.a.), kes esitas aksiomaatilise süsteemi elementaargeomeetria jaoks. Saksa matemaatik David Hilbert (1862–1943) püüdis aga sajandivahetusel formaliseerida kogu kaasaegset matemaatikat alates hulgateooriast ja algebrast ning lõpetades matemaatilise analüüsiga. Kurt Gödel (1906–1978) näitas 1931. aastal, et naturaalarvulist aritmeetikat ei saa esitada formaalse aksiomaatilise süsteemina. Ta tõestas, et iga piisavalt võimas matemaatiline teooria on *mittetäielik*, s.t. selles teoorias leidub tõene valem, mis ei ole formaalses süsteemis tuletatav.

Ka Aristotelese *süllogismid* sobivad sellesse skeemi. Nad on lihtsalt kahe eeldusega kehtivad tuletused. *Süllogismi moodused* on meie mõistes tuletusreeglid. Tuntumad neist on *modus ponens* ja *disjunktiivse süllogismi reegel*.

Tabelites 5.1 ja 5.2 on loetletud kasutatavamad lausearvutuse tuletus- ja asendusreeglid.

Asendusreegleid, mis esitatakse võrduste kujul, võib samuti vaadelda kui teatud tüüpi tuletusreegleid. Erinevalt tavalistest tuletusreeglitest, mis toimivad ainult ühes suunas: ülemisest valemist saab tuletada alumise valemi, saab asendusreegleid rakendada nii vasakpoolsest valemist parempoolse valemi tuletamiseks kui ka vastupidi. Asendusreeglite korral on võrduse kahel pool olevad valemid loogiliselt ekvivalentsed. Seepärast võib neid reegleid kasutada ka valemite osade asendamiseks.

Reegel	Lühend	Reegli nimi
$\frac{p \,\&\, q}{p}$, $\frac{p \,\&\, q}{q}$	Simp	Lihtsustamisreegel
$\frac{p, q}{p \,\&\, q}$	Conj	Konjunktsioonireegel
$\frac{p \supset q,\ p}{q}$	MP	*Modus ponens*
$\frac{p \supset q, \neg q}{\neg p}$	MT	*Modus tollens*
$\frac{p \vee q, \neg p}{q}$, $\frac{p \vee q, \neg q}{p}$	DS	Disjunktiivne süllogism
$\frac{p \supset q,\ q \supset r}{p \supset r}$	HS	Hüpoteetiline süllogism
$\frac{p}{p \vee q}$, $\frac{q}{p \vee q}$	Add	Lisamisreegel
$\frac{p \supset q,\ r \supset s,\ p \vee r}{q \vee s}$	CD	Konstruktiivne dilemma

Tabel 5.1. Levinumad lausearvutuse tuletusreeglid.

Teoreem 5.1 (asendusteoreem). *Kui valemid p ja q on loogiliselt ekvivalentsed ja valem r on saadud valemist s mõnede valemi p esinemiste asendamise teel valemiga q, siis on valemid r ja s loogiliselt ekvivalentsed.*

Reegel	Lühend	Reegli nimi
$p \,\&\, q = q \,\&\, p$	Comm	Kommutatiivsus
$p \vee q = q \vee p$	Comm	
$p \,\&\, (q \,\&\, r) = (p \,\&\, q) \,\&\, r$	Assoc	Assotsiatiivsus
$p \vee (q \vee r) = (p \vee q) \vee r$	Assoc	
$p \,\&\, (q \vee r) = (p \,\&\, q) \vee (p \,\&\, r)$	Dist	Distributiivsus
$p \vee (q \,\&\, r) = (p \vee q) \,\&\, (p \vee r)$	Dist	
$p = p \,\&\, p$	Red	Liiasusreeglid
$p = p \vee p$	Red	
$p = \neg\neg p$	DN	Kahekordse eituse reegel
$\neg(p \,\&\, q) = \neg p \vee \neg q$	DeM	De Morgani seadused
$\neg(p \vee q) = \neg p \,\&\, \neg q$	DeM	
$p \supset q = \neg p \vee q$	CE	Implikatsioonireegel
$p \supset q = \neg q \supset \neg p$	Contra	Ümberpööramisreegel
$(p \,\&\, q) \supset r = p \supset (q \supset r)$	Exp	Väljaviimisreegel
$p \equiv q = (p \supset q) \,\&\, (q \supset p)$	Bic	Ekvivalentsireeglid
$p \equiv q = p \,\&\, q \vee \neg p \,\&\, \neg q$	Bic	

Tabel 5.2. Tuntumad lausearvutuse asendusreeglid.

Tõestus. Valemite p ja q loogiline ekvivalentsus tähendab, et nendel valemitel on kõigis võimalikes situatsioonides sama tõeväärtus. Järelikult ei saa ka valemid r ja s loogiliselt erineda. □

Esitame nüüd mõned tuletuste näited. Vaatleme arutlust

Kui ma teen tervisejooksu, siis ma ei jää haigeks.

Ma hakkan haigeks jääma.

Ma ei tee tervisejooksu.

Selle järelduse tuletamiseks toome sisse kahekordse eituse ja rakendame seejärel reeglit *modus tollens*:

1.	$T \supset \neg H$	eeldus	*Kui ma teen tervisejooksu, siis ma ei jää haigeks.*
2.	H	eeldus	*Ma hakkan haigeks jääma.*
3.	$\neg\neg H$	2 DN	*Pole õige, et ma ei jää haigeks.*
4.	$\neg T$	1,3 MT	*Ma ei tee tervisejooksu.*

Toome näite asendusreeglite erisugusest kasutamisest. Selles tuletuses kasutatakse kommutatiivsust kui tuletusreeglit:

1.	$(A \vee B) \supset C$	
2.	$B \vee A$	
3.	$A \vee B$	2 Comm
4.	C	1,3 MP

Järgmise tuletuse kolmandas reas rakendatakse aga asendusteoreemi:

1.	$(A \vee B) \supset C$	
2.	$B \vee A$	
3.	$(B \vee A) \supset C$	1 Comm
4.	C	2,3 MP

Assotsiatiivsusreeglid võimaldavad konjunktsioonide ja disjunktsioonide alamvalemeid suvaliselt grupeerida ja ümber järjestada. Seepärast võib paljudel juhtudel sulud üldse ära jätta.

Teoreem 5.2 (ümbertõstmisteoreem). *Kehtivad järgmised samaväärsused:*

$$p_1 \,\&\, \ldots \,\&\, p_n = p_{m_1} \,\&\, \ldots \,\&\, p_{m_n},$$

$$p_1 \vee \ldots \vee p_n = p_{m_1} \vee \ldots \vee p_{m_n},$$

kus $m_1, \ldots, m_n$ on indeksite jada $1, \ldots, n$ suvaline ümberjärjestus (ehk permutatsioon) ja võrduse mõlemale poole jäävates valemites võib kasutada suvalist (lubatud) sulgude paigutust.

Näiteks järgmises tuletuses on kõik esimesest valemist tuletatud valemid esialgse valemiga ekvivalentsed:

1.	$(A \vee B) \vee (C \vee D)$	
2.	$A \vee (B \vee (C \vee D))$	1 Assoc
3.	$(B \vee (C \vee D)) \vee A$	2 Comm
4.	$((B \vee C) \vee D) \vee A$	3 Assoc
5.	$(B \vee C) \vee (D \vee A)$	4 Assoc
6.	$(C \vee B) \vee (D \vee A)$	5 Comm

Asendusreeglitega saab viia kõik lausearvutuse valemid üheselt määratud standardkujule: kas täielikule disjunktiivsele normaalkujule või täielikule konjunktiivsele normaalkujule.

Definitsioon 5.3. *Literaalideks* nimetatakse atomaarseid valemeid ja nende eitusi. *Elementaardisjunktsiooniks* nimetatakse literaalide disjunktsiooni ning *elementaarkonjunktsiooniks* literaalide konjunktsiooni. Valem on *disjunktiivsel normaalkujul* (DNK), kui ta on elementaarkonjunktsioonide disjunktsioon:

$$\bigvee_{i=1}^{m} l_{i_1} \,\&\, \ldots \,\&\, l_{i_{m_i}},$$

ning valem on *konjunktiivsel normaalkujul* (KNK), kui ta on elementaardisjunktsioonide konjunktsioon:

$$\bigwedge_{i=1}^{m} l_{i_1} \vee \ldots \vee l_{i_{m_i}}.$$

Lausemuutujaid $X_1, \ldots, X_n$ sisaldav valem on *täielikul disjunktiivsel normaalkujul*, kui ta on disjunktiivsel normaalkujul, mille iga elementaarkonjunktsioon sisaldab kõiki lausemuutujaid $X_1, \ldots, X_n$, s.t. iga elementaarkonjunktsioon on kujul

$$X_1^{\alpha_1} \,\&\, \ldots \,\&\, X_n^{\alpha_n},$$

kus $\alpha_1, \ldots, \alpha_n \in \{t, v\}$ ja

$$X^t \stackrel{\text{def}}{=} X \quad \text{ja} \quad X^v \stackrel{\text{def}}{=} \neg X,$$

ning valem on *täielikul konjunktiivsel normaalkujul*, kui ta on konjunktiivsel normaalkujul, mille iga elementaardisjunktsioon on kujul

$$X_1^{\alpha_1} \vee \ldots \vee X_n^{\alpha_n}.$$

Näiteks valem $\neg A \,\&\, (B \vee C)$ on konjunktiivsel normaalkujul, ent ta ei ole täielikul konjunktiivsel normaalkujul ega disjunktiivsel normaalkujul. Valem $\neg A \,\&\, B$ on aga nii konjunktiivsel normaalkujul kui ka täielikul disjunktiivsel normaalkujul.

Teoreem 5.4. *Iga lausearvutuse valemi saab viia disjunktiivsele (konjunktiivsele) normaalkujule, s.t. valemi saab teisendada temaga ekvivalentseks disjunktiivsel (konjunktiivsel) normaalkujul olevaks valemiks.*

Tõestus. Valemite normaalkujule viimiseks sobib algoritm

1. Kaotame implikatsiooni- ja ekvivalentsireeglite abil valemist implikatsioonid ja ekvivalentsid.
2. Viime eitused De Morgani seaduste ja kahekordse eituse reegli abil vahetult lausemuutujate ette.
3. Rakendame distributiivsusreegleid, kuni kõik konjunktsioonid on disjunktsioonide all (disjunktsioonid on konjunktsioonide all).

□

Näiteks valem $(X \equiv Y) \supset Z$ teisendatakse disjunktiivsele normaalkujule järgmiselt:

$$\begin{aligned}
(X \equiv Y) \supset Z &= (X \,\&\, Y \vee \neg X \,\&\, \neg Y) \supset Z \\
&= \neg(X \,\&\, Y \vee \neg X \,\&\, \neg Y) \vee Z \\
&= (\neg(X \,\&\, Y) \,\&\, \neg(\neg X \,\&\, \neg Y)) \vee Z \\
&= ((\neg X \vee \neg Y) \,\&\, (\neg\neg X \vee \neg\neg Y)) \vee Z \\
&= ((\neg X \vee \neg Y) \,\&\, (X \vee Y)) \vee Z \\
&= ((\neg X \vee \neg Y) \,\&\, X \vee \\
&\quad (\neg X \vee \neg Y) \,\&\, Y) \vee Z \\
&= \neg X \,\&\, X \vee \neg Y \,\&\, X \vee \neg X \,\&\, Y \vee \\
&\quad \neg Y \,\&\, Y \vee Z.
\end{aligned}$$

Disjunktiivsest (konjunktiivsest) normaalkujust on mõistlik välja jätta loogiliselt väärad elementaarkonjunktsioonid (loogiliselt tõesed elementaardisjunktsioonid), mis sisaldavad mingit aatomit ja selle eitust:

$$\ldots = \neg Y \,\&\, X \vee \neg X \,\&\, Y \vee Z.$$

Samuti kustutatakse korduvad elementaarkonjunktsioonid (elementaardisjunktsioonid).

Kui eesmärgiks on täieliku DNK leidmine, siis tuleb mittetäielikke elementaarkonjunktsioone täiendada neis puuduvate lausemuutujatega. Selleks korrutame igat elementaarkonjunktsiooni iga temas puuduva muutuja T korral loogiliselt tõese valemiga $T \vee \neg T$ ja kasutame distributiivsusreeglit. Täielike normaalkujude elementaarkonjunktsioonid ja -disjunktsioonid järjestatakse tihti ka tähestikuliselt:

$$\begin{aligned}
\ldots &= \neg Y \,\&\, X \,\&\, (Z \vee \neg Z) \vee \neg X \,\&\, Y \,\&\, (Z \vee \neg Z) \vee \\
&\quad Z \,\&\, (X \vee \neg X) \,\&\, (Y \vee \neg Y) \\
&= \neg Y \,\&\, X \,\&\, Z \vee \neg Y \,\&\, X \,\&\, \neg Z \vee \neg X \,\&\, Y \,\&\, Z \vee \\
&\quad \neg X \,\&\, Y \,\&\, \neg Z \vee Z \,\&\, X \,\&\, Y \vee Z \,\&\, X \,\&\, \neg Y \vee \\
&\quad Z \,\&\, \neg X \,\&\, Y \vee Z \,\&\, \neg X \,\&\, \neg Y \\
&= X \,\&\, Y \,\&\, Z \vee X \,\&\, \neg Y \,\&\, Z \vee X \,\&\, \neg Y \,\&\, \neg Z \vee
\end{aligned}$$

$$\neg X \,\&\, Y \,\&\, Z \vee \neg X \,\&\, Y \,\&\, \neg Z \vee \neg X \,\&\, \neg Y \,\&\, Z.$$

Valemite täielikud normaalkujud on liikmete järjekorra täpsusega üheselt määratud.

Ülesanded

1. *Teisendage valem $(X \equiv Y) \supset Z$ täielikule konjunktiivsele normaalkujule.
2. Tõestage arutluste kehtivus:

   (a) $C \supset A$
   $A \supset S$
   $S \supset \neg D$
   D
   ———
   $\neg C$

   (b) $D \supset (R \vee C)$
   $\neg R$
   $\neg C$
   ———
   $\neg D$

5.2. Konditsionaalne tõestus

Kui meil on vaja tuletada implikatsiooni sisaldav lause, siis on üheks võimaluseks kasutada *konditsionaalse tõestuse* reeglit. Võtame implikatsiooni vasakul pool oleva valemi hüpoteesiks ja püüame sellest lähtudes tõestada implikatsiooni paremal pool olevat valemit. Konditsionaalse e. *tingimusliku* tõestuse reegli (ka *implikatsiooni sissetoomisreegli*) CP saab kirja panna kujul

$$\frac{\Gamma,\, p \vdash q}{\Gamma \vdash p \supset q} \quad \text{(CP)},$$

kus Γ tähistab tuletuse eelduste hulka.

n.	p	hüpotees
	$\dots$	
m.	q	
m+1.	$p \supset q$	n–m CP

Joonis 5.1. Konditsionaalse tõestuse skeem.

Konditsionaalse tõestuse skeem on antud joonisel 5.1. Tuletuse osa alates n-ndast valemist kuni m-nda valemini sõltub hüpoteesist p ja on seepärast märgitud vasakul püstjoonega. Seda osa tuletusest nimetatakse *alamtuletuseks*. Alamtuletuse lõppedes kaotavad tema valemid kehtivuse ning neid ei tohi enam tuletuses kasutada. Alamtuletuse hüpotees kaob samuti tuletuse eelduste hulgast ära. Alamtuletuses endas võib aga kasutada kõiki sellest väljapoole jäävaid tuletuse valemeid. Kasutades alamtuletuse valemeid väljastpoolt saab näiteks tuletada vastuolulise järelduse:

1.	A	eeldus
2.	$\neg A$	hüpotees
3.	$A \,\&\, \neg A$	1,2 Conj

Konditsionaalse tõestuse reegli kasutamist õigustab fakt, et valemid

$$(l_1 \,\&\, l_2 \,\&\, \dots \,\&\, l_n) \supset (p \supset q)$$

ja $$(l_1 \,\&\, l_2 \,\&\, \dots \,\&\, l_n \,\&\, p) \supset q$$

on loogiliselt ekvivalentsed.

Toome näite konditsionaalse tõestuse reegli rakendamise kohta. Vaatleme arutlust

Kui ostan laserplaadimängija, siis pean oma kulutusi kärpima või jään võlgadesse.

Kui ostan laserplaate, siis ma ei kärbi oma kulutusi.

Kui ostan CD-mängija, siis ostan plaate.

Kui ostan laserplaadimängija, siis jään võlgadesse.

See arutlus tõestatakse konditsionaalse tõestuse reegliga järgmiselt:

1.	$L \supset (K \vee V)$	eeldus
2.	$P \supset \neg K$	eeldus
3.	$L \supset P$	eeldus
4.	$\vert L$	hüpotees
5.	$\vert K \vee V$	1,4 MP
6.	$\vert P$	3,4 MP
7.	$\vert \neg K$	2,6 MP
8.	$\vert V$	5,7 DS
9.	$L \supset V$	4–8 CP

Konditsionaalse tuletuse saab tihti asendada tuletusega, kus seda reeglit ei kasutata. Näiteks saab vaadeldava arutluse tõestada ka ilma konditsionaalse tõestuse reeglita:

1.	$L \supset (K \vee V)$	eeldus
2.	$P \supset \neg K$	eeldus
3.	$L \supset P$	eeldus

4.	$L \supset \neg K$	2,3 HS
5.	$L \supset (\neg\neg K \vee V)$	1 DN
6.	$L \supset (\neg K \supset V)$	5 CE
7.	$(L \,\&\, \neg K) \supset V$	6 Exp
8.	$(\neg K \,\&\, L) \supset V$	7 Comm
9.	$\neg K \supset (L \supset V)$	8 Exp
10.	$L \supset (L \supset V)$	4,9 HS
11.	$(L \,\&\, L) \supset V$	10 Exp
12.	$L \supset V$	11 Red

Konditsionaalse tõestuse reeglit on vaja tuletussüsteemi täielikkuse jaoks, kuna muidu ei saaks mõnesid loogiliselt tõeseid valemeid formaalselt tuletada. Ilma konditsionaalse tõestuse reeglita ei saa näidata, et

$$\vdash A \supset A.$$

Tõepoolest, peale konditsionaalse tõestuse reegli ei ole meie tuletussüsteemis ühtegi teist reeglit, mille abil seda valemit saaks tuletada. Reegli CP abil on aga asi lihtne:

1.	$\vert A$	hüpotees
2.	$A \supset A$	1 CP

Selliseid valemeid, mida saab tuletada ilma ühtegi eeldust kasutamata, nimetatakse *teoreemideks*. Järelikult on valem $A \supset A$ (lausearvutuse ja predikaatarvutuse) teoreem.

Pikemate arutluskäikude korral võib alamtõestusi olla mitu. Siis võib üks alamtuletus olla teise alamtuletuse sees:

1.	A	hüpotees
2.	B	hüpotees
3.	$\neg\neg A$	1 DN
4.	A	3 DN
5.	$B \supset A$	2–4 CP
6.	$A \supset (B \supset A)$	1–5 CP

Kahekordse eituse sissetoomine ja väljaviimine on selles tuletuses vajalikud, sest meil puudub reegel, mis võimaldab valemit A dubleerida.

Ülesanded

1. Tõestage, et

   (a) $p \supset q = p \supset (p \,\&\, q)$;

   (b) $p \supset q \neq p \vee q$.

5.3. Kaudne tõestus

Kaudse tõestamise korral eeldatakse, et arutluse eeldused on tõesed ja järeldus on väär (s.t. järelduse eitus on tõene) ning saadakse sellest vastuolu. Seega peab järeldus olema tõene. Sellist tõestusmeetodit nimetatakse ka *vastuväiteliseks tõestamiseks* (*reductio ad absurdum*). Kaudne tõestus on matemaatikas levinud, sest seda on sageli lihtsam teha kui otsest tõestust. Võtame näiteks väite "Tühihulk sisaldub igas hulgas H": $\emptyset \subseteq H$. Selle väite vastuväitelisel tõestamisel eeldatakse, et tühihulgas $\emptyset$ leidub element, mis ei kuulu hulka H. See on aga vastuolus tühihulga definitsiooniga, mis ütleb, et temas ei ole ühtegi elementi. Järelikult on väide $\emptyset \subseteq H$ tõene.

Kaudse tõestuse skeem on toodud joonisel 5.2. Tõestuse n-indas reas tuuakse sisse hüpotees $\neg p$ ning alustatakse alamtõestust. Kui m-ndas reas saadakse vastuolu $q \,\&\, \neg q$, siis võib $(m+1)$-s reas tuletada *kaudse tõestuse reegliga* IP hüpoteesi $\neg p$

n.	$\neg p$	hüpotees
	$\ldots$	
m.	$q \,\&\, \neg q$	
m+1.	p	n–m IP

Joonis 5.2. Kaudse tõestuse skeem.

ilma eituseta. *Vastuoluks* sobib suvalise valemi ja selle eituse konjunktsioon. Mõnes tuletussüsteemis tuuakse vastuolu tähistamiseks sisse erisümbol (näiteks $\perp$).

Kaudse tõestuse reegli saab ülejäänud reeglitega asendada. Näiteks joonisel 5.2 toodud kaudsest tuletusest saab teha tavalise tuletuse:

n.	$\neg p$	hüpotees
	$\ldots$	
m.	$q \,\&\, \neg q$	
m+1.	q	Simp
	$\neg q$	Simp
	$q \vee p$	Add
	p	DS
	$\neg p \supset p$	CP
	$\neg\neg p \vee p$	CE
	$p \vee p$	DN
	p	Red

Järelikult ei ole kaudse tõestuse reeglit tuletussüsteemi täielikkuse jaoks tarvis.

Toome nüüd näite kaudse tõestuse reeglit sisaldavast tuletusest. Arutlus

Kui riigikogu ratifitseerib seaduseelnõu ja president selle kinnitab, siis see muutub seaduseks.

Riigikogu ratifitseeris seaduseelnõu.

See ei muutunud seaduseks.

President ei kinnitanud seaduseelnõud

tõestatakse kaudse tõestuse reegli abil järgmiselt:

1.	$(R \,\&\, P) \supset S$	
2.	R	
3.	$\neg S$	
4.	$\neg\neg P$	hüpotees
5.	P	4 DN
6.	$R \,\&\, P$	2,5 Conj
7.	S	1,6 MP
8.	$S \,\&\, \neg S$	3,7 Conj
9.	$\neg P$	4–8 IP

5.4. Predikaatarvutuse tuletusreeglid

Predikaatarvutuses kehtivad kõik lausearvutuse tuletusreeglid, kuid siin on ka omad spetsiifilised reeglid. Kvantorite sissetoomiseks ja eemaldamiseks tuleb valemid viia kujule, kus kvantorid asuvad valemi vasakus servas. Me ütleme, et sellised valemid on *prefikskujul* e. *preeneks-normaalkujul*.

Kvantorite ettetoomisreeglid on koondatud tabelisse 5.3. *Kvantorite eitamisreeglid* võimaldavad viia eitusesümbolid kvantorialustesse avaldistesse. *Kvantorite distributiivsusreeglitega* saab kvantorite alla viia konjunktsioonid ja disjunktsioonid. Kaks esimest kvantorite distributiivsusreeglit on asendusreeglid, sest nende abil

saab teha mõlemasuunalisi tuletusi, kaks viimast on aga tuletusreeglid, sest nendega saab tuletada ainult vasakpoolsest valemist parempoolse valemi, vastupidine tuletus ei kehti. Näiteks ei saa kvantorite distributiivsusreeglit

$$\frac{\exists x\,(p(x) \,\&\, q(x))}{\exists x\; p(x) \,\&\, \exists x\, q(x)} \quad \text{(QD)}$$

kasutada teistpidi: eksistentsikvantori valemi ette toomiseks. Võtame näiteks interpretatsiooni, kus $p(x)$ tähendab, et inimene on naissoost ja $q(x)$, et inimene on meessoost. Siis on tuletusreegli alumine lause tõene, sest leidub nii naisi kui ka mehi, kuid ülemine lause on väär, sest ükski inimene ei ole korraga nais- ja meessoost.

Kvantorite järjekorra muutmiseks kasutatakse *kvantorite järjestusreegleid*. *Kvantorite liigutamisreeglites* eeldatakse, et x ei esine valemites p ja q vabalt (x võib aga esineda vabalt valemites $p(x)$ ja $q(x)$).

Kui kvantoreid ei saa ühegi reegliga valemi ette tuua, siis tuleb mõni kvantifitseeritud muutuja asendada selles valemis mitteesineva muutujaga. Kvantifitseeritud *muutuja asendamine* on lubatud, kui me asendame muutuja korraga nii kvantoris kui ka kvantorialuses avaldises. Muutujate asendamisega saab viia prefikskujule ka eespool toodud distributiivsusreegli alumise valemi:

$$\begin{aligned} \exists x\; p(x) \,\&\, \exists x\, q(x) &= \exists x\; p(x) \,\&\, \exists y\, q(y) \\ &= \exists x \exists y\, (p(x) \,\&\, q(y)). \end{aligned}$$

Muutujate asendamisel on oluline, et asendav muutuja ei esineks valemis, sest muidu saaksime mittekehtiva tuletuse. Näiteks valemist

$$\exists x \exists y\; Sxy$$

ei tohi asenduse teel moodustada valemit

$$\exists x \exists x\; Sxx \qquad (= \exists x\; Sxx),$$

sest esialgse valemi tõesusest ei järeldu sugugi, et valem Sxy on tõene x ja y võrdsetel väärtustel. See võimaldab meil konstrueerida

Reegel	Lühend	Reegli nimi
$\neg\forall x\ p(x) = \exists x \neg p(x)$	QN	Kvantorite eitamisreeglid
$\forall x\ p(x) = \neg\exists x \neg p(x)$	QN	
$\neg\forall x \neg p(x) = \exists x\ p(x)$	QN	
$\forall x \neg p(x) = \neg\exists x\ p(x)$	QN	
$\forall x\ (p(x)\ \&\ q(x)) =$ $\forall x\ p(x)\ \&\ \forall x\ q(x)$	QD	Kvantorite distributiivsusreeglid
$\exists x\ (p(x) \vee q(x)) =$ $\exists x\ p(x) \vee \exists x\ q(x)$	QD	
$\forall x\ p(x) \vee \forall x\ q(x) \Rightarrow$ $\forall x\ (p(x) \vee q(x))$	QD	
$\exists x\ (p(x)\ \&\ q(x)) \Rightarrow$ $\exists x\ p(x)\ \&\ \exists x\ q(x)$	QD	
$\forall x\forall y\ p(x, y) = \forall y\forall x\ p(x, y)$	QI	Kvantorite järjestusreeglid
$\exists x\exists y\ p(x, y) = \exists y\exists x\ p(x, y)$	QI	
$\exists x\forall y\ p(x, y) \Rightarrow \forall y\exists x\ p(x, y)$	QI	
$p \supset \forall x\ q(x) = \forall x\ (p \supset q(x))$	QM	Kvantorite liigutamisreeglid (p ja q ei sisalda muutujat x vabalt)
$p \supset \exists x\ q(x) = \exists x\ (p \supset q(x))$	QM	
$\forall x\ p(x) \supset q = \exists x\ (p(x) \supset q)$	QM	
$\exists x\ p(x) \supset q = \forall x\ (p(x) \supset q)$	QM	
$p\ \&\ \forall x\ q(x) = \forall x\ (p\ \&\ q(x))$	QM	
$p\ \&\ \exists x\ q(x) = \exists x\ (p\ \&\ q(x))$	QM	
$p \vee \forall x\ q(x) = \forall x\ (p \vee q(x))$	QM	
$p \vee \exists x\ q(x) = \exists x\ (p \vee q(x))$	QM	

Tabel 5.3. Kvantorite ettetoomisreeglid.

interpretatsiooni, kus esimene valem on tõene, aga teine valem väär. Näiteks interpreteerides predikaati Sxy kui seost $x \neq y$.

Teoreem 5.5. *Iga predikaatarvutuse valemi saab viia prefikskujule, s.t. valemi saab teisendada temaga ekvivalentseks prefikskujul olevaks valemiks.*

Tõestus. Teeme induktsiooni predikaatarvutuse valemi definitsiooni järgi:

1. Atomaarsed valemid ongi prefikskujul, sest nad ei sisalda ühtegi kvantorit.
2. Valemi $\neg p$ saab viia prefikskujule, sest me saame induktsiooni põhjal tuua kvantorid valemi p algusesse ja tõstame nad seejärel kvantorite eitamisreeglite abil eituse ette.
3. Valemid $(p\ \&\ q)$, $(p \vee q)$, $(p \supset q)$ ja $(p \equiv q)$ saab viia prefikskujule, sest induktsiooni põhjal saab tõsta kvantorid valemite p ja q algusesse. Seejärel asendame ekvivalentsid ning kasutame kvantorite distributiivsus- ja liigutamisreegleid ning kvantifitseeritud muutujate asendamist.
4. Valemid $\forall v\ p$ ja $\exists v\ p$ saab induktsiooni põhjal viia prefikskujule, sest kvantorid saab tõsta valemi p algusesse.

□

Toome nüüd valemite prefikskujule viimise kohta paar näidet. Valem $\exists x\ Fx \supset \forall y\ Gy$ viiakse prefikskujule järgmiselt:

$$\begin{aligned} \exists x\ Fx \supset \forall y\ Gy &= \forall y\ (\exists x\ Fx \supset Gy) \\ &= \forall y\forall x\ (Fx \supset Gy). \end{aligned}$$

Lause "Kellegile meeldivad kõik raamatud", mille võib formaliseerida valemina

$$\exists x\ (Ix\ \&\ \forall y\ (Ry \supset Mxy)),$$

üheks võimalikuks prefikskujuks on

$$\exists x\forall y\ (Ix\ \&\ (Ry \supset Mxy)).$$

Prefikskujul valemid saab edasi teisendada *Skolemi normaalkujule*. Selleks eemaldatakse valemist kõik eksistentsikvantorid ja asendatakse nendes esinevad muutujad *Skolemi funktsioonidega*, mille argumentideks võetakse kõik eksistentsikvantorile eelnevates üldisuskvantorites esinevad muutujad. Kui eksistentsikvantorile ei eelne ühtegi üldisuskvantorit, siis nimetatakse Skolemi funktsiooni *Skolemi konstandiks*. Eelmise näite lause saab viia Skolemi normaalkujule, asendades inimese, kellele meeldivad kõik raamatud, Skolemi konstandiga c:

$$\forall y\,(Ic \,\&\, (Ry \supset Mcy)).$$

Lause "Igal inimesel on ema" formalisatsiooni

$$\forall x \exists y\, E(x, y)$$

saab aga teisendada Skolemi normaalkujule, asendades muutuja y Skolemi funktsiooniga $e(x)$. Funktsioon $e(x)$ seab igale inimesele vastavusse tema ema:

$$\forall x\, E(x, e(x)).$$

Skolemi funktsioone kasutatakse teoreemide automaatsel tõestamisel.

Me oleme nüüd valmis selleks, et tutvustada kvantorite sissetoomis- ja eemaldamisreegleid. Eeldame, et valemid on prefikskujul. Idee on selles, et me eemaldame kõigepealt kvantorid, teeme valemiga sobivad teisendused ja asetame seejärel kvantorid valemi ette tagasi.

Kvantorite sissetoomis- ja eemaldamisreeglid on koondatud tabelisse 5.4. *Üldisuskvantori eemaldamisreegel* võimaldab üldisuskvantoris esineva muutuja x asendada suvalise konstantsümboliga c. Seejuures tuleb valemis p asendada korraga kõik x vabad esinemised. Sellise asenduse tulemus märgitakse üles kujul $p\,\{x/c\}$ ja öeldakse, et see on valemi p *x/c-erijuht*. Üldisuskvantori sissetoomisreegli

$$\frac{p\,\{x/v\}}{\forall x\; p} \quad \text{(UG)}$$

Reegel	Lühend	Reegli nimi	Tingimus
$\dfrac{\forall x\; p}{p\,\{x/c\}}$	UI	Üldisuskvantori eemaldamine	c on suvaline konstant
$\dfrac{p\,\{x/v\}}{\forall x\; p}$	UG	Üldisuskvantori sissetoomine	v on spetsiaalne "konstantmuutuja"
$\dfrac{\exists x\; p}{p\,\{x/w\}}$	EI	Eksistentsikvantori eemaldamine	w on uus konstantsümbol (mis ei tohi esineda järelduses)
$\dfrac{p\,\{x/c\}}{\exists x\; p}$	EG	Eksistentsikvantori sissetoomine	

Tabel 5.4. Kvantorite sissetoomis- ja eemaldamisreeglid.

korral on tähtis, et konstant v oleks suvaline. Teda on kasulik vaadelda kui konstantmuutujat. Üldisuskvantori kaotamisel asendatakse sellised konstandid, mida me soovime hiljem üldistada, spetsiaalse *konstantmuutujaga* v, mis tähistab suvalist vaadeldava universumi objekti. Kuna v on asendamise hetkest peale suvaline konstant, siis tohib teda hiljem uuesti üldistada. Kui meil on tuletuses tarvis rohkem kui ühte sellist konstantmuutujat, siis kirjutame sümboli v koos alaindeksiga: $v_1, v_2, \ldots$

Kasutame kirjeldatud kvantorite eemaldamise ja sissetoomise meetodit järgmise arutluse kehtivuse tõestamiseks:

$$\frac{\begin{array}{l}\textit{Kõik esmakursuslased on tudengid}\\ \textit{Kõigil tudengitel on õpinguraamat}\end{array}}{\textit{Kõigil esmakursuslastel on õpinguraamat}}$$

Toome tuletuses sisse konstantmuutuja v ja seome selle lõpus uuesti üldisuskvantoriga:

1.	$\forall x\,(Ex \supset Tx)$	eeldus
2.	$\forall x\,(Tx \supset \tilde{O}x)$	eeldus
3.	$Ev \supset Tv$	1 UI
4.	$Tv \supset \tilde{O}v$	2 UI
5.	$Ev \supset \tilde{O}v$	3,4 HS
6.	$\forall x\,(Ex \supset \tilde{O}x)$	5 UG

Eksistentsikvantori eemaldamisreeglis asendatakse muutuja tuletuses varem mitte esineva konstantsümboliga w. *Eksistentsikvantori sissetoomisreegel*

$$\frac{p\,\{x/c\}}{\exists x\,p}\quad \text{(EG)}$$

lubab asendada valemis esineva konstandi c muutujaga x ning siduda selle eksistentsikvantoriga. Näiteks lause "Jaan armastab ennast" ehk

$$Ajj$$

korral saab eksistentsikvantori sisse tuua kolmel viisil:

$$\exists x\,Ajx,$$
$$\exists x\,Axj$$
$$\text{või}\quad \exists x\,Axx,$$

sest

$$Ajj = Ajx\,\{x/j\} = Axj\,\{x/j\} = Axx\,\{x/j\}.$$

Toome nüüd mõned näited eksistentsikvantori eemaldamise ja sissetoomise kohta. Arutluse

Mõni kass lööb nurru

Leidub nurrulööja

tõestamiseks toome nurruva kassi tähistamiseks sisse konstantsümboli w ja eemaldame selle tuletuse lõpus eksistentsikvantori sissetoomise reegliga:

1.	$\exists x\,(Kx \,\&\, Nx)$	eeldus
2.	$Kw \,\&\, Nw$	1 EI
3.	Nw	2 Simp
4.	$\exists x\,Nx$	3 EG

Järgmises näites demonstreeritakse, et juhul, kui me ei kasuta eksistentsikvantori eemaldamisel unikaalset konstanti, saab teha vigase järelduse:

1.	$\exists x\,(Kx \,\&\, Sx)$		*Mõni kass on sõbralik*
2.	$\exists x\,(Rx \,\&\, Sx)$		*Mõni koer on sõbralik*
3.	$Kw \,\&\, Sw$	1 EI	
4.	$Rw \,\&\, Sw$	2 EI	
5.	Kw	3 Simp	
6.	Rw	4 Simp	
7.	$Kw \,\&\, Rw$	5,6 Conj	
8.	$\exists x\,(Kx \,\&\, Rx)$	7 EG	*Mõni kass on koer*

Tõestame lõpuks keerulisema arutluse, kus kasutatakse nii üldisus- kui ka eksistentsikvantoreid:

Keegi armastab Katit

Igaüks, kes kedagi armastab, on õnnelik

Kati on inimene

Keegi on õnnelik

Arutluse tõestus on selline:

1.	$\exists x\,(Ix \,\&\, Axk)$	eeldus
2.	$\forall x\,((Ix \,\&\, \exists y\,(Iy \,\&\, Axy)) \supset \tilde{O}x)$	eeldus
3.	Ik	eeldus

4.	$Iw \,\&\, Awk$	1 EI
5.	Awk	4 Simp
6.	$Ik \,\&\, Awk$	3,5 Conj
7.	$\exists y\,(Iy \,\&\, Awy)$	6 EG
8.	$(Iw \,\&\, \exists y\,(Iy \,\&\, Awy)) \supset \tilde{O}w$	2 UI
9.	Iw	4 Simp
10.	$Iw \,\&\, \exists y\,(Iy \,\&\, Awy)$	7,9 Conj
11.	$\tilde{O}w$	8,10 MP
12.	$Iw \,\&\, \tilde{O}w$	9,11 Conj
13.	$\exists x\,(Ix \,\&\, \tilde{O}x)$	12 EG

Ülesanded

1. Tooge näide üldisuskvantori sissetoomisreegli tingimuse vajalikkuse kohta.
2. Konstrueerige ülesande 4.11.1 arutluste tuletused.

5.5. Formaalne ja mitteformaalne tõestus

Matemaatika on teiste teaduste hulgas erilisel kohal, sest matemaatikud püüavad oma väiteid põhjalikult tõestada. Tõestuste formaalsuse tase varieerub olenevalt tõestajast ja eeldatavast lugejast alates formaalsest predikaatarvutuse tõestusest kuni paarisõnalise selgituseni.

Tõestame näiteks hulkade võrduse omaduse

$$X = Y \;\Leftrightarrow\; X \subseteq Y \quad \text{ja} \quad Y \subseteq X.$$

Selleks tuleb eelnevalt defineerida omaduses kasutatavad seosed. Kaks hulka on võrdsed, kui nad koosnevad ühtedest ja samadest elementidest:

$$X = Y \Leftrightarrow \forall x\,(x \in X \Leftrightarrow x \in Y).$$

Hulk X on hulga Y *osahulk* siis ja ainult siis, kui iga X element kuulub hulka Y:

$$X \subseteq Y \Leftrightarrow \forall x\,(x \in X \Rightarrow x \in Y).$$

Nende seoste definitsioonid saab predikaatarvutuses esitada kujul

$$\forall X \forall Y\,(X = Y \equiv \forall x\,(x \in X \equiv x \in Y)),$$
$$\forall X \forall Y\,(X \subseteq Y \equiv \forall x\,(x \in X \supset x \in Y)).$$

Meil on vaja tõestada, et

$$\forall X \forall Y\,(X = Y \equiv (X \subseteq Y \,\&\, Y \subseteq X)).$$

Tõestus. Esitame vaadeldava omaduse formaalse tõestuse:

1.	$\forall X \forall Y\,(X = Y \equiv \forall x\,(x \in X \equiv x \in Y))$	$=$ def.
2.	$V_1 = V_2 \equiv \forall x\,(x \in V_1 \equiv x \in V_2)$	1 UI
3.	$V_1 = V_2 \equiv \forall x\,((x \in V_1 \supset x \in V_2)\,\&\, (x \in V_2 \supset x \in V_1))$	2 Bic
4.	$V_1 = V_2 \equiv (\forall x\,(x \in V_1 \supset x \in V_2)\,\&\, \forall x\,(x \in V_2 \supset x \in V_1))$	3 QD
5.	$V_1 = V_2 \equiv (V_1 \subseteq V_2 \,\&\, V_2 \subseteq V_1)$	4 $\subseteq$ def.
6.	$\forall X \forall Y\,(X = Y \equiv (X \subseteq Y \,\&\, Y \subseteq X))$	5 UG

□

Tõestame veel, et osahulga seos on transitiivne:

$$X \subseteq Y \quad \text{ja} \quad Y \subseteq Z \;\Rightarrow\; X \subseteq Z,$$

millele vastab predikaatarvutuse valem

$$\forall X \forall Y \forall Z\,((X \subseteq Y \,\&\, Y \subseteq Z) \supset X \subseteq Z).$$

Tõestus. Formaalne predikaatarvutuse tõestus koosneb üheteistkümnest tõestussammust:

1.	$V_1 \subseteq V_2 \,\&\, V_2 \subseteq V_3$	hüpotees
2.	$\forall x\,(x \in V_1 \supset x \in V_2) \,\&\, \forall x\,(x \in V_2 \supset x \in V_3)$	1 $\subseteq$ def.
3.	$\forall x\,((x \in V_1 \supset x \in V_2) \,\&\, (x \in V_2 \supset x \in V_3))$	2 QD
4.	$(v \in V_1 \supset v \in V_2) \,\&\, (v \in V_2 \supset v \in V_3)$	3 UI
5.	$v \in V_1 \supset v \in V_2$	4 Simp
6.	$v \in V_2 \supset v \in V_3$	4 Simp
7.	$v \in V_1 \supset v \in V_3$	5,6 HS
8.	$\forall x\,(x \in V_1 \supset x \in V_3)$	7 UG
9.	$V_1 \subseteq V_3$	8 $\subseteq$ def.
10.	$(V_1 \subseteq V_2 \,\&\, V_2 \subseteq V_3) \supset V_1 \subseteq V_3$	1–9 CP
11.	$\forall X \forall Y \forall Z\,((X \subseteq Y \,\&\, Y \subseteq Z) \supset X \subseteq Z)$	10 UG

□

Matemaatikud ei esita tavaliselt tõestusi formaalselt. Nad eeldavad, et kuulajad on loogikareeglitega piisavalt kursis, ning esitavad tõestuse skemaatiliselt ehk *mitteformaalselt*. Tõestuse skeemi kehtivust saab igaüks ise kontrollida.

Tõestus (mitteformaalne). Olgu X, Y ja Z suvalised hulgad, nii et $X \subseteq Y$ ja $Y \subseteq Z$. Olgu $x \in X$ suvaline element. Siis $x \in Y$, sest $X \subseteq Y$, ja $x \in Z$, sest $Y \subseteq Z$. Seega $x \in X \Rightarrow x \in Z$ ja kehtib $X \subseteq Z$. □

Tõestus (lühem mitteformaalne). Oletame, et $x \in X$. Siis $x \in Y$ ja $x \in Z$. □

Tõestus (veel lühem). See on ilmne. □

Ülesanded

1. Tõestage hulgateoorias formaalselt ja mitteformaalselt:

$$X = Y \Rightarrow Y = X.$$

5.6. Aksioom ja mudel

Mitmed matemaatika distsipliinid on üles ehitatud aksiomaatiliselt:

- Rühma-, ringi-, võre-, vektorruumide ja muude algebraliste süsteemide teooriad algebras.
- Meetriliste, topoloogiliste, Hilberti, Banachi jms. ruumide teooriad funktsionaalanalüüsis.
- Geomeetria alused.
- Zermelo-Fraenkeli hulgateooria jne.

Tavaliselt esitatakse aine aksiomaatilise käsitluse puhul järgmise skeemi kohaselt:

1. Fikseeritakse mingi hulk teoorias uuritavaid objekte, nendel defineeritud funktsioone ja seoseid ning nende tähistamise sümboolika.
2. Teatud hulk väiteid peetakse tõesteks *a priori* (ilma tõestuseta). Neid väiteid nimetatakse selle teooria *aksioomideks*.
3. Teooria arendamine seisneb *teoreemide tõestamises*. *Teoreemid* on väited, mida saab tõestada ainult aksioome kasutades. Väidete mugavamaks sõnastamiseks võidakse olemasolevate mõistete baasil defineerida uusi.

Filosoofilises mõttes võib suhtumine aksioomidesse olla üsna mitmesugune. Eukleides mõtles geomeetriast nähtavasti kui teadusest, mis uurib tegelikult eksisteeriva ruumi omadusi. Tema geomeetria aksioomid pidid kujutama piisavalt lihtsaid väiteid, mille tõesus reaalses ruumis on ilmne. Järgmiste, enamasti ka keerulisema struktuuriga väidete tõesus vajas aga juba tõestamist.

Aksiomaatiliselt esitatud tänapäeva loodusteaduste, näiteks elektrodünaamika ja kvantmehhaanika korral, on suhtumine aksioomidesse relatiivne. Aksioomid on siin väited, mis on tõesed

sedavõrd, kuivõrd nad on vaatluse või eksperimendiga kontrollitud. Teooria aksiomaatiline korrastamine võimaldab selgitada mingi väidete hulga, mida tuleb kontrollida. Aksioomidest tõestamise teel tuletatud seadused on aga siis juba garanteeritult tõesed.

Tänapäeva matemaatika aksiomaatilised teooriad defineerivad aksioomidega oma uurimisobjekti. Näiteks algebras nimetatakse rühmadeks neid struktuure, mis rahuldavad rühma aksioome. Sellises teoorias tõestatud teoreemide kohta teame aga lihtsalt seda, et nad on tõesed seal, kus kehtivad aksioomid (näiteks kõigil rühmadel).

Loogikat huvitab toodud skeemis põhiliselt kolmas punkt: teoreemide tõestamine. Matemaatikaga tegelejal tuleb tihti kokku puutuda küsimusega, kas antud tekst T on väite V tõestus või mitte.

Iga tõestus koosneb sammudest. Igal sammul esitatakse järjekordne väide ja seletus, millistest eespool olevatest väidetest see järeldub (ajutiselt võidakse esitada ka oletusi, mida siis nendest omakorda järeldusi tehes kontrollitakse). Selle, kas uus väide tõepoolest on viidatud väidete järeldus, otsustame oma isiklikku tõestussammu intuitsiooni kasutades. Iga samm tõestuses peab meie jaoks olema ilmne, silmnähtav.

Matemaatika areng on näidanud, et sellisest käsitlusest ei piisa. Vaikimisi eeldatakse siin, et järeldumise intuitsioon on kõigil inimestel ühesugune ja vastab objektiivsele tegelikkusele. Neist kahest eeldusest viimase paikapidavuse kohta vaevalt mingit lõplikku otsust teha saab. Aga tegelikult ei kehti ka esimene. Vähemalt kogu matemaatilise analüüsi ajalugu näitab, et juba maailma juhtivate matemaatikute vahel on olnud vaidlusi. Tõestusmeetodid, mis on vastuvõetavad ühtedele matemaatikutele, pole seda teiste jaoks. Matemaatilises loogikas on selle probleemi uurimiseks tekkinud *formaliseeritud aksiomaatilise teooria*formaliseeritud aksiomaatiline teooria mõiste.

Formaliseeritud aksiomaatilises teoorias fikseeritakse peale aksioomide ilmutatud kujul ka *tuletusreeglid*: lõplik hulk tõestussammu tegemiseks lubatud viise, s.t. võimalusi järeldada väidetest

$p_1, p_2, \ldots, p_n$ väide q. Selleks tuleb kõigepealt fikseerida keel, milles väiteid kirjutatakse. Järelikult saame formaliseeritud aksiomaatiliste teooriate jaoks järgmise üldskeemi.

1. Fikseeritakse teooria tähestik ja valemite keel väidete üleskirjutamiseks.

2. Valemite hulga teatud alamhulga elemente nimetatakse *aksioomideks* ja arvatakse antud teoorias *a priori* tõesteks. Enamasti nõutakse, et aksioomide hulk oleks *lahenduv*, s.t. peab leiduma algoritm, mis teeb kindlaks, kas mingi valem on aksioom või ei. Tihti on aksioomide hulk lihtsalt lõplik.

3. Fikseeritakse lõplik hulk tuletusreegleid kujul

$$\frac{p_1, p_2, \ldots, p_n}{q} ,$$

mille põhjal arvatakse valem q *vahetult tuletatavaks* valemitest $p_1, p_2, \ldots, p_n$. Iga reegel peab olema *lahenduv*, s.t. tema jaoks peab leiduma algoritm, mis teeb suvaliste valemite $p_1, p_2, \ldots, p_n$ ja q korral kindlaks, kas valem q järeldub antud reegliga valemitest $p_1, p_2, \ldots, p_n$ või mitte.

Näide 5.6.

1. Koosnegu teooria $\mathcal{M}$ tähestik sümbolitest I ja * ning olgu valemiteks need sõnad, mis sisaldavad parajasti ühte sümbolit *.

2. Ainsaks aksioomiks olgu valem *.

3. Olgu teoorial üks tuletusreegel

$$\frac{p}{\mathrm{I}p\mathrm{I}} .$$

On kerge näha, et teoorias $\mathcal{M}$ on tuletatavad parajasti need valemid, kus vasakul ja paremal on ühesugune arv sümboleid I.

Tuletamine igas formaalses teoorias on mäng sümbolitega. Selle mängu reegleid nimetatakse *teooria süntaksiks*. Peale selle võib teooria valemitele olla antud mingisugune *sisu*, s.t. võib olla defineeritud, mida iga valem tähendab. Valemite tähendust nimetatakse vastava keele *semantikaks*.

Näiteks esimest järku keele korral defineerib iga interpretatsioon ühe semantika selle keele valemitele. Semantikat võib defineerida ka mingi interpretatsioonide klassi kaudu, seades igale valemile vastavusse väite, et see valem on tõene kõigis antud klassi kuuluvates interpretatsioonides (näiteks kõigil rühmadel või kõigil lõpmatutel mudelitel). Lausearvutusvalemitele võime seada vastavusse väite, et see valem on samaselt tõene.

Definitsioon 5.7. Teooriat $\mathcal{T}$ nimetatakse *korrektseks* semantika $\mathcal{S}$ suhtes, kui iga teoorias $\mathcal{T}$ tuletatav valem on semantikas $\mathcal{S}$ tõene.

Definitsioon 5.8. Teooriat $\mathcal{T}$ nimetatakse *täielikuks* semantika $\mathcal{S}$ suhtes, kui iga semantikas $\mathcal{S}$ tõene valem on tuletatav teoorias $\mathcal{T}$.

Jätkame näidet teooriaga $\mathcal{M}$. Tähistagu $m*n$ valemit, kus sümbolit I on vasakul m korda ja paremal n korda ($m, n \geq 0$). Defineerime teooria valemitele kolm semantikat:

- $\mathcal{S}_1$: $m*n$ tähendab $m = n$.
- $\mathcal{S}_2$: $m*n$ tähendab $m \leq n$.
- $\mathcal{S}_3$: $m*n$ tähendab $m < n$.

Eespool toodud tuletatavuse kirjeldusest järeldub, et teooria $\mathcal{M}$ on korrektne ja täielik semantika $\mathcal{S}_1$ suhtes. Semantika $\mathcal{S}_2$ suhtes on ta korrektne, aga mitte täielik, sest $m < n$ korral ei saa valemit $m*n$ tuletada. Täielikkuse saavutamiseks võiksime lisada teise tuletusreegli

$$\frac{p}{p\mathrm{I}}\ .$$

Semantika $\mathcal{S}_3$ suhtes ei ole teooria $\mathcal{M}$ ei korrektne ega täielik. Reegli lisamisel saadud teooria on küll täielik, aga ei ole korrektne, sest tuletatavad on ka valemid $m*n$, kus $m = n$.

Ülesanded

1. Konstrueerige korrektne teooria semantika $\mathcal{S}_3$ jaoks.

Paljudel juhtudel on alguses olemas teooria sisuline pool, s.t. uuritavad objektid, mille kohta mingi hulk fakte on juba teada. Teooria arendamiseks või korrastamiseks valitakse valemite keel, mis võimaldab tähistada valdkonnas esinevaid põhilisi konstante ja funktsioone ning väljendada teoorias kasutatavaid põhilisi predikaate. Selle keele semantika on juba algusest peale olemas. Aksioomid ja reeglid proovitakse valida nii, et formaalne aksiomaatiline teooria tuleks korrektne ja täielik, s.t. tuletatavaks osutuksid parajasti need valemid, mis on semantikas tõesed. Matemaatilise loogika uurimused on näidanud, et paljudel tähtsatel juhtudel ei ole see võimalik (näiteks naturaalarvude aritmeetika korral). On selge, et mittekorrektne formalisatsioon, kus saaks tõestada ka vääri valemeid, ei ole üldse vastuvõetav. Seega jääb üle rahulduda paratamatusega, et mõnda tõest valemit ei saa formaalses teoorias tuletada, ja püüda jätta mittetuletatavad tõed teooria ääremaadele.

Formaalsete aksiomaatiliste teooriate aluseks võetakse tavaliselt kas lausearvutuse või predikaatarvutuse aksiomaatiline teooria, millele lisatakse vaadeldava teooria aksioomid. Seepärast on formaalsetel aksiomaatilistel teooriatel üldjuhul samad tuletusreeglid ning nad erinevad üksteisest ainult aksioomide poolest.

Olgu σ mingi signatuur ja T suvaline valemite hulk signatuuris σ. Me võime vaadelda hulka T kuuluvaid valemeid kui aksioome.

Signatuuri σ interpretatsiooni M nimetame aksioomide hulga T *mudeliks*, kui iga valem hulgast T on interpretatsioonis M tõene.

Näiteks kui $\sigma = \langle e; \cdot; = \rangle$ ja hulk T koosneb viiest rühmateooria aksioomist

G1. $\forall x \forall y \forall z\, [(x \cdot y) \cdot z = x \cdot (y \cdot z)]$,
G2. $\forall x\, [x \cdot e = x]$,
G3. $\forall x\, [e \cdot x = x]$,
G4. $\forall x \exists u\, [x \cdot u = e]$,

G5. $\forall x \exists u\,[u \cdot x = e]$,
siis hulga T mudeliteks on parajasti kõik rühmad. Kui lisada veel kuues aksioom
G6. $\forall x \forall y\,[x \cdot y = y \cdot x]$,
siis on mudeliteks parajasti kõik kommutatiivsed (Abeli) rühmad.

Mudeli definitsiooni järgi võib aksioomide hulgaks olla suvaline valemite hulk. Väga paljudel juhtudel tegeldakse lõplike aksioomide hulkadega. Näiteks kõik klassikalised algebralised süsteemid defineeritakse lõpliku aksioomisüsteemi abil. Aga näiteks peatükis 10 vaadeldav naturaalarvude Peano aksiomaatika on lõpmatu. Kui me tahame tuletatavust aksioomidest vaadelda kui kontrollitavat tõestusviisi, peab aksioomide hulk olema algoritmiliselt lahenduv, s.t. meil peab olema algoritm tõestuses esinevate viidete "See on tõene, sest ta on aksioom" kontrollimiseks. Teoreetilistes konstruktsioonides vaatleme aga mõnikord ka halvema struktuuriga aksioomide hulki.

Esimest järku teooria on aksiomaatiline teooria, mille valemite keele moodustavad mingi signatuuri valemid, aksioomideks on predikaatloogika aksioomid ja selle teooria omaaksioomid ning tuletusreegliteks predikaatloogika tuletusreeglid. Kui signatuur sisaldab võrdusmärki, arvatakse aksioomide hulka kuuluvateks ka võrduse aksioomid.

On selge, et esimest järku teooria määratakse tema signatuuri ja omaaksioomidega. Teooria mudeliks nimetame iga struktuuri, kus kehtivad teooria kõik omaaksioomid. Siinkohal me ei pea rääkima predikaatloogika ja võrduse aksioomidest, sest need kehtivad igal struktuuril.

Öeldakse, et kinnine valem A signatuuris σ järeldub loogiliselt aksioomidest T, kui A on tõene T igas mudelis. Millises vahekorras on aksioomidest sellises semantilises mõttes järelduvad valemid ja teoorias T tuletatavad valemid?

Teoreem 5.9 (korrektsuseteoreem). *Kui mingi valem on teoorias T tuletatav, siis on ta tõene teooria T omaaksioomide igas mudelis.*

Sellel on lihtne põhjendus predikaatloogika korrektsuse kaudu. Fikseerime mudeli M. Olgu valem B tuletatav. Tema tuletus on lõplik objekt ja seal saab olla kasutatud ainult lõplik arv teooria T aksioome. Olgu need $A_1, \ldots, A_n$. Kasutades n korda deduktsiooniteoreemi, saame, et predikaatloogikas (juba ilma T aksioomideta) saab tuletada implikatsiooni $A_1 \,\&\, \ldots \,\&\, A_n \supset B$. Järelikult peab see valem olema samaselt tõene. Siis on ta mudelis M tõene, samuti kui aksioomid $A_1, \ldots, A_n$. Implikatsiooni definitsioonist saame, et tõene on ka B.

Teoreem näitab, et struktuuride klassi defineerimine (näiteks algebras, funktsionaalanalüüsis jne.) võimaldab matemaatikas tõestada ühekorraga teoreeme, mis kehtivad paljudel struktuuridel. Kehtib ka teoreem 5.10.

Teoreem 5.10 (täielikkuseteoreem). *Iga valemit, mis on tõene teooria T omaaksioomide igas mudelis, saab tuletada teoorias T.*

Küsime nüüd, millistel valemite hulkadel T leidub mudel. On selge, et vasturääkival hulgal, millest saab tuletada mingi valemi ja ka selle valemi eituse, mudelit olla ei saa, sest mudelil peaksid siis teoreemi 5.9 põhjal olema tõesed kaks teineteist eitavat valemit. Seega saab mudeli olemasoluga tõestada aksiomaatika mittevasturääkivust. Näiteks algebraõpikutes kontrollitakse tavaliselt pärast rühmateooria aksioomide tutvustamist, et ainult ühikelemendist e koosnev struktuur rahuldab rühmaaksioome. Sellega on näidatud, et rühmateooria uurimisobjekt on olemas.

Osutub, et mittevasturääkivus on ka piisav mudeli olemasoluks, s.t. kehtib teoreem 5.11.

Teoreem 5.11 (mudeliteoreem). *Igal mittevasturääkival esimest järku teoorial leidub mudel.*

Mudel konstrueeritakse vaadeldava signatuuri kinnistest (s.t. vabade muutujateta) termidest ja valemitest.

Paljud aksioomide süsteemid on matemaatikas tegelikult definitsiooni rollis ja matemaatikud on aksioome fikseerides küllalt vabad. Näiteks on avastatud, et tehetel arvude aritmeetikas,

maatriksitel jm. objektidel on mingid omadused: kommutatiivsus, assotsiatiivsus jne. Järgmise sammuna võetakse signatuuri üks, kaks või enam tehtemärki ja aksioomideks nende tehete mingid olulised omadused. Kõiki saadud aksioome rahuldavaid struktuure nimetatakse rühmadeks, ringideks vms. Rühma-, ringi- vms. teooria spetsialistid arendavad nüüd teooriat, tõestades aksioomidest järelduvaid teoreeme. Tihti avastatakse, et sellise skeemi järgi postuleeritud omadused on veel muudel praktilist või teoreetilist huvi pakkuvatel struktuuridel ja saadud teoreeme saab rakendada.

5.7. Aksiomatiseerimisteooria

Vaatleme nüüd olukorda, kus on fikseeritud mingi struktuuride klass või üksik struktuur mingis signatuuris ja matemaatikud soovivad omada selle klassi/struktuuri teooria aksiomaatikat, et tõestada teoreeme mingil kindlal baasil. Kas elementaarteooriat (s.t. klassil/struktuuril tõeste valemite hulka) saab alati aksiomatiseerida? Päris triviaalses mõttes muidugi, sest võime võtta aksioomideks kõik tõesed valemid. Aga sellise aksiomatiseerimisega me ei võida midagi, kui aksioomiks olemise kontroll on sama keeruline kui tõesuse kontroll. Milliste omadustega aksiomaatika on üldse kasutamiseks kõlblik?

Kõigepealt on selge, et aksiomaatika peab olema *korrektne*, s.t. aksioomidest peab saama tuletada ainult tõeseid valemeid. Kui tuletatav on ka mõni väär valem, siis ei anna aksioomidest tuletamine mingit tõesuse garantiid. Nagu juba nägime, piisab korrektsuseks iga aksioomi tõesusest kõigil vaadeldavatel struktuuridel. Vastupidine omadus — *täielikkus*, s.t. kõigi tõeste valemite tuletatavus — on küll samuti soovitav, kuid mitte absoluutselt vajalik. Matemaatikas on kasutusel ka mittetäielikke aksiomaatikaid. Sellise puudusega on isegi üldtuntud Peano aksiomaatika naturaalarvude jaoks. Küll aga peab aksioomide hulk olema algoritmiliselt *lahenduv*, et tõestuse lugeja saaks kontrollida, kas tõestusse ilma põhjendamata sisse toodud väited ikka on aksioomid.

Miks ollakse sunnitud kasutama mittetäielikke aksioomide süsteeme? Algoritmiteooria terminites väljendades toob aksioomide hulga lahenduvus kaasa tuletatavate valemite hulga *rekursiivse loetletavuse*, s.t. leidub algoritm, mis genereerib kõik tuletatavad valemid. Algoritmiteoorias tõestatakse ka, et naturaalarvude hulgal tõeste signatuuri $\langle 0; +, \cdot; = \rangle$ valemite hulk pole loenduv ja järelikult pole olemas naturaalarvude aritmeetika korrektset ja täielikku aksiomaatikat. Sellegipoolest saab Peano aksioomidest tuletada näiteks kõik standardsete ülikooli matemaatikakursuste ülesehitamiseks vajalikud aritmeetikateoreemid. Kuni seitsmekümnendate aastateni kirjutati loogikaõpikutes koguni, et tuletatavad on üldse kõik "tegeliku matemaatika" teoreemid ning loogikutele teadaolevad mittetuletatavad tõesed valemid on kunstlikult konstrueeritud. Praeguseks on välja selgitatud, et juba mõned eelmise sajandi arvuteooria tulemused asuvad väljaspool Peano aritmeetikat.

5.8. Ülesanded

1. On teada, et
$$A \vdash B.$$
   (a) Mida me võime öelda A kehtestatavuse, samaselt tõesuse, samaselt vääruse ja temast B järeldumise kohta?
   (b) Mida me võime öelda B kehtestatavuse, samaselt tõesuse, samaselt vääruse ja temast A järeldumise kohta?

2. Andke järgmistele argumentide otsesed tõestused. (Allikas: Platon, *Phaedo*.)
   (a) Kui Sokrates ei usuks, et kui ta sureb, läheb ta jumalate juurde, toimiks ta valesti mitte surma vastu olles. Kui ta seda usub, ei toimiks ta valesti mitte surma vastu olles. Seega, ta kas usub seda ja ei toimi valesti, mitte surma vastu olles, või ta ei usu seda ja toimib valesti mitte surma vastu olles.

(b) Kui keha on hingele takistuseks, siis on hingel parem kehast vaba olla. Kui see on tõsi, siis ei tule surma karta. Aga, kui kõrgeimate asjade teadmisel saab hing keha poolt petetud, siis on keha hingele takistuseks. Kuid hing on keha poolt sel viisil petetud. Seega, surma ei tule karta.

(c) Kui me tunnistame, et mõned asjad on võrdsed ja mõned ebavõrdsed, siis me peame teadma, mis võrdsus ise on. Aga, kui mitte midagi meie meeltes ei ole võrdsus ise, siis me kas ei tea, mis võrdsus ise on või ei omanda me seda teadmist meelte kaudu. Kui me ei omanda seda teadmist meelte kaudu, siis me sünnime koos mingi teadmisega. Kuna me tunnistame, et mõned asjad on võrdsed ja teised ebavõrdsed, ja kuna miski meie meeltes pole võrdsus ise, siis järelikult me sünnime koos mingi teadmisega.

3. (C. H. Papadimitriou) Olgu meil kasutada konstantsümbolid me ja $mybaby$ ning binaarne predikaatsümbol $loves$. Kirjutage valem p, mis vastab kõnekäänule

Everybody loves my baby,
my baby loves nobody but me ...

(Kõik armastavad minu beebit, minu beebi armastab ainult mind ...)

(a) Näidake, et $p \models mybaby = me$;

(b) Tõestage, et $p \vdash mybaby = me$;

(c) Püüdke kirjutada valem p hoolikamalt, nii et ei tekiks selliseid vastuolusid.

4. Kurjuse probleem. Tõestage, et järgmised väited on vasturääkivad:

- Kui kurjus on olemas, siis kas jumal ei taha seda ära hoida või ei saa seda ära hoida.
- Kui jumal on kõikvõimas, siis ta saab kurjust ära hoida.
- Kui jumal on hea, siis ta tahab kurjust ära hoida.

- Kui jumal eksisteerib, siis on ta kõikvõimas ja hea.
- Kurjus on olemas.
- Jumal on olemas.

Mis sellest järeldub?

5. Võtame kasutusele naiste loogika aksioomi (A. Kolmogorov):

$$\frac{A \supset B \quad B \text{ on meeldiv}}{A}$$

Tõestage, et naiste loogika on vasturääkiv. Mida on selleks vaja minimaalselt eeldada?

6. Tuletussüsteemid

> *Mis täna tõestatakse, oli millalgi vaid unistuseks.*
> — William Blake (1757–1827), *Taeva ja põrgu abielu*

Kirjeldame selles peatükis lühidalt formaalseid tuletussüsteeme, nagu Hilberti tüüpi tuletused, loomulikud ja sekventsiaalsed tuletused ning tõesuspuud. Resolutsiooni kui programmeerimises ja tehisintellektis enim kasutatava tuletussüsteemi üksikasjalise kirjelduse toome raamatu viimases osas.

6.1. Hilberti tüüpi tuletus

Ajalooliselt hakati esimesena kasutama Hilberti tüüpi tuletussüsteeme, milles on suhteliselt palju aksioome ja vähe tuletusreegleid.

Vaatleme lihtsat tuletussüsteemi H, milles on ainult kaks lausearvutuse tehet: eitus ja implikatsioon. Ülejäänud tehted defineeritakse nende kaudu:

$$p \,\&\, q \stackrel{\text{def}}{=} \neg(p \supset \neg q),$$
$$p \vee q \stackrel{\text{def}}{=} \neg p \supset q,$$
$$p \equiv q \stackrel{\text{def}}{=} (p \supset q) \,\&\, (q \supset p).$$

Tuletussüsteemis on kolm aksioomiskeemi:

(H1) $p \supset (q \supset p)$;
(H2) $(p \supset (q \supset r)) \supset ((p \supset q) \supset (p \supset r))$;
(H3) $(\neg p \supset \neg q) \supset ((\neg p \supset q) \supset p)$.

Ainsaks tuletusreegliks on *modus ponens*:

$$\frac{p \quad p \supset q}{q}\ (MP).$$

Sellisel tuletussüsteemil on rohkem teoreetiline tähendus, sest tema kohta on lihtne tõestada matemaatilisi omadusi; temaga tuletamine on aga küllalt tülikas, kuna me ei saa kasutada paljusid tuttavaid tuletusreegleid.

Näitame, et süsteemis H on tuletatav teoreem $p \supset p$.

Lause 6.1. $\vdash_H p \supset p$.

Tõestus.

1.	$(p \supset ((p \supset p) \supset p)) \supset ((p \supset (p \supset p)) \supset (p \supset p))$	H2
2.	$p \supset ((p \supset p) \supset p)$	H1
3.	$(p \supset (p \supset p)) \supset (p \supset p)$	1,2 MP
4.	$p \supset (p \supset p)$	H1
5.	$p \supset p$	3,4 MP

□

Tuletamise muudab keerukaks see, et aksioomiskeemidesse on väga raske leida vajalikke substitutsioone. Näiteks lause 6.1 tõestuses on määrava tähtsusega aksioomiskeemi (H2) sobiva erikuju leidmine.

Kuidas aga tõestada konjunktsiooni kommutatiivsusreeglit

$$p \,\&\, q \supset q \,\&\, p,$$

s.t. $\vdash_H \neg(p \supset \neg q) \supset \neg(q \supset \neg p)$?

Hilberti tüüpi tuletussüsteemides tuletatakse tõestest valemitest uusi tõeseid valemeid. Järgnevalt tõestame deduktsiooniteoreemi, mille abil saab sisse tuua ka hüpoteese.

Teoreem 6.2 (deduktsiooniteoreem (Herbrand, 1930)). *Kehtib*

$$\Gamma, p \vdash_H q \;\Rightarrow\; \Gamma \vdash_H p \supset q,$$

kus Γ *on suvaline valemite loetelu ning* p *ja* q *on suvalised valemid.*

Tõestus. Olgu $q_1, \ldots, q_n = q$ tuletus, mis kasutab eeldustena ainult Γ elemente ja valemit p. Näitame induktsiooniga, et $\Gamma \vdash p \supset q_i$ $(1 \le i \le n)$. Valem q_1 on kas Γ element, aksioom või valem p. Esimesel kahel juhul saame aksioomiskeemi (H1) erikuju $q_1 \supset (p \supset q_1)$, millest tuletame *modus ponens*'iga $\Gamma \vdash p \supset q_1$. Kui $q_1 = p$, kasutame lauset 6.1.

Eeldame nüüd, et $\Gamma \vdash p \supset q_k$ $(k < i)$, ja näitame, et $\Gamma \vdash p \supset q_i$. Valem q_i on kas Γ element, aksioom, valem p või järeldub *modus ponens*'iga valemitest q_j ja q_m, kus $j, m < i$ ja $q_m = q_j \supset q_i$. Esimesed kolm juhtu tõestame samamoodi, kui $i = 1$ korral. Viimasel juhul on meil juba tõestatud, et $\Gamma \vdash p \supset q_j$ ja $\Gamma \vdash p \supset (q_j \supset q_i)$. Aksioomiskeemi (H2) põhjal kehtib

$$\vdash (p \supset (q_j \supset q_i)) \supset ((p \supset q_j) \supset (p \supset q_i)).$$

Rakendades sellele valemile kaks korda reeglit *modus ponens* saame $\Gamma \vdash p \supset q_i$. Võttes $i = n$ saamegi soovitud tulemuse. □

Näiteks piisab $p \supset q,\ q \supset r \vdash p \supset r$ tõestamiseks tuletusest

1.	$p \supset q$	hüpotees
2.	$q \supset r$	hüpotees
3.	p	hüpotees
4.	q	1,3 MP
5.	r	2,4 MP

Kleene kasutab lausearvutuse jaoks järgmist Hilberti tüüpi tuletussüsteemi K, kus $p \equiv q \stackrel{\text{def}}{=} (p \supset q)\,\&\,(q \supset p)$:

(K1) $p \supset (q \supset p)$;
(K2) $(p \supset q) \supset ((p \supset (q \supset r)) \supset (p \supset r))$;

(K3) $p \,\&\, q \supset p$;
(K4) $p \,\&\, q \supset q$;
(K5) $p \supset (q \supset p \,\&\, q)$;
(K6) $p \supset p \vee q$;
(K7) $q \supset p \vee q$;
(K8) $(p \supset r) \supset ((q \supset r) \supset (p \vee q \supset r))$;
(K9) $(p \supset q) \supset ((p \supset \neg q) \supset \neg p)$;
(K10) $\neg\neg p \supset p$.

Süsteemi ainsaks tuletusreegliks on jällegi *modus ponens*. Selles süsteemis on juba võimalik tõestada deduktsiooniteoreemiga, et $\vdash_K p \,\&\, q \supset q \,\&\, p$:

1.	$p \,\&\, q$	hüpotees
2.	$p \,\&\, q \supset p$	K3
3.	p	1,2 MP
4.	$p \,\&\, q \supset q$	K4
5.	q	1,4 MP
6.	$q \supset (p \supset (q \,\&\, p))$	K5
7.	$p \supset (q \,\&\, p)$	5,6 MP
8.	$q \,\&\, p$	3,7 MP
9.	$p \,\&\, q \supset q \,\&\, p$	1–8 deduktsiooniteoreem

Predikaatarvutuse korral lisanduvad Hilberti süsteemile H aksioomiskeemid

(H4) $\forall x\, p(x) \supset p\{x/t\}$;
(H5) $\forall x\,(p \supset q) \supset (p \supset \forall x\, q)$.

Term t peab olema aksioomiskeemis (H4) muutuja x *asendamiseks vaba*, s.t. term t ei tohi sisaldada muutujaid, mis osutuvad substitutsiooni tagajärjel valemis $p\{x/t\}$ seotuks. Aksioomiskeemis (H5) ei sisalda valem p muutujat x vabalt.

Modus ponens'ile lisandub *üldistamisreegel*

$$\frac{p}{\forall x\, p}\ (Gen)$$

ning eksistentsikvantor defineeritakse valemiga

$$\exists x\ p \stackrel{\text{def}}{=} \neg\forall x \neg p.$$

Ülesanded

1. Tõestage deduktsiooniteoreemiga:

$$p \supset (q \supset r),\ q \vdash_H p \supset r.$$

2. Tõestage $\vdash_K p \supset p$.
3. Tõestage, et süsteemis K kehtib deduktsiooniteoreem.
4. Konstrueerige interpretatsioonid, mis näitavad, et aksioomiskeemides (H4) ja (H5) ei või kitsendavaid lisatingimusi ära jätta.

6.2. Loomulik tuletus

Gentzeni tüüpi *loomuliku tuletuse* süsteemides on iga loogilise tehte jaoks sissetoomisreegel ja eemaldamisreegel. Gentzeni tüüpi on näiteks tuletussüsteem ND, milles on kaks aksioomiskeemi

$$p \vdash p \quad \text{(hüpotees)},$$
$$\vdash p \vee \neg p \quad (\textit{Tertium non datur})$$

ja tabelis 6.1 toodud tuletusreeglid.

Loomulikes tuletustes käsitletakse hüpoteese reeglite osadena. Siis on hüpoteesid ümbritsetud nurksulgudega ning on reeglitega seotud punktiirjoonte abil. Sümboliga $\bot$ tähistatakse suvalist vastuolulist olukorda, mille põhjuseks on mingi lause ja selle eituse samaaegne tõesus.

Üldisuskvantori sissetoomisreeglis $(\forall I)$ ja eksistentsikvantori eemaldamisreeglis $(\exists E)$ peab olema täidetud tingimus $(*)$, mis nõuab, et term a peab olema muutuja x asendamiseks valemis p vaba.

Loomuliku tuletuse süsteemide matemaatiliste omaduste tõestamine on üldiselt keerulisem kui Hilberti tüüpi süsteemide korral, kuid nad on praktiliseks tõestuseks tunduvalt mugavamad.

Sissetoomisreegel	Eemaldamisreegel
$\dfrac{\bot}{p}$ *Ex falso quodlibet*	
$\dfrac{p \quad q}{p \,\&\, q}\ (\&I)$	$\dfrac{p \,\&\, q}{p}\quad\dfrac{p \,\&\, q}{q}\ (\&E)$
$\dfrac{p}{p \vee q}\quad\dfrac{q}{p \vee q}\ (\vee I)$	$\dfrac{p \vee q \quad \begin{matrix}[p]\\ \vdots\\ r\end{matrix} \quad \begin{matrix}[q]\\ \vdots\\ r\end{matrix}}{r}\ (\vee E)$
$\dfrac{\begin{matrix}[p]\\ \vdots\\ q\end{matrix}}{p \supset q}\ (\supset I)$	$\dfrac{p \quad p \supset q}{q}\ (\supset E)$
$\dfrac{\begin{matrix}[p]\\ \vdots\\ \bot\end{matrix}}{\neg p}\ (\neg I)$	$\dfrac{p \quad \neg p}{\bot}\ (\neg E)$
$\dfrac{(p \supset q) \,\&\, (q \supset p)}{p \equiv q}\ (\equiv I)$	$\dfrac{p \equiv q}{(p \supset q) \,\&\, (q \supset p)}\ (\equiv E)$
$\dfrac{p\{x/a\}}{\forall x\ p}\ (\forall I)^*$	$\dfrac{\forall x\ p}{p\{x/t\}}\ (\forall E)$
$\dfrac{p\{x/t\}}{\exists x\ p}\ (\exists I)$	$\dfrac{\exists x\ p \quad \begin{matrix}[p\{x/a\}]\\ \vdots\\ q\end{matrix}}{q}\ (\exists E)^*$

Tabel 6.1. Loomuliku tuletuse süsteemi reeglid.

Tihti kirjutatakse valemi tõesust mõjutavate eelduste ja hüpoteeside numbrid temast vasakule ja tuletusreegli rakendamise järel kustutatakse selle reegli hüpoteeside numbrid saadud valemit mõjutavate väidete numbrite loendist. Selles mõttes võib igat loomuliku tuletuse rida käsitleda kui kehtivat arutlust.

Lause 6.3. $\vdash_{ND} p \,\&\, q \supset q \,\&\, p$.

Tõestus.

1	1.	$p \,\&\, q$	hüpotees
1	2.	p	$1\ \&E$
1	3.	q	$1\ \&E$
1	4.	$q \,\&\, p$	$2{,}3\ \&I$
	5.	$p \,\&\, q \supset q \,\&\, p$	$1{,}4\ \supset I$

Hüpotees $p \,\&\, q$ sõltub ainult iseendast. Sellepärast on temast vasakule kirjutatud tema enda number. Ka valemite p, q ja $q \,\&\, p$ tõesus sõltub ainult hüpoteesi $p \,\&\, q$ tõesusest. Kuna valemi $p \,\&\, q \supset q \,\&\, p$ tuletamisel kasutatakse hüpotees $p \,\&\, q$ ära, siis ei sõltu tema tõesus enam ühestki eeldusest ega hüpoteesist. □

6.3. Gentzeni sekventsiaalne arvutus

Oma 1935. aasta artiklis tegi saksa matemaatik Gerhard Gentzen (1909–1945) loomuliku tuletuse süsteemide defineerimise järel veel ühe sammu edasi, muutes peale tuletuste kuju ka nende objektide kuju, mille abil väiteid üles kirjutatakse ja mida tuletatakse. Valemite asemel on vaatluse all *sekventsid* — read kujul

$$A_1, \ldots, A_m \Rightarrow B_1, \ldots, B_n,$$

kus $m, n \geq 0$ ja $A_1, \ldots, A_m$ ning $B_1, \ldots, B_n$ on valemid. Sekventsi vasakpoolset osa $A_1, \ldots, A_m$ nimetatakse *antetsedendiks* ja parempoolset osa $B_1, \ldots, B_n$ *suktsedendiks.*

Sekventsile omistatakse sama tähendus kui valemile $A_1 \,\&\, \ldots \,\&\, A_m$ ning $B_1 \vee \ldots \vee B_n$, s.t. väide "Valemitest A_1, … ja A_m järeldub, et kehtib B_1, … või B_n". Kui $m = 0$, siis tähendab see lihtsalt paremal oleva disjunktsiooni kehtimist. Kui $n = 0$, siis mõistetakse tühja disjunktsiooni vastuoluna. Näiteks paljusid matemaatikateoreeme, mis väidavad, et kui on tõene eelduste konjunktsioon, siis kehtib ka teoreemi väide, on loomulik kirja panna sekventsina, mille antetsedent koosneb mitmest valemist, aga suktsedent ainult ühest. Gentzen võttis mitut valemit sisaldava suktsedendiga sekventsid kasutusele sümmeetria saavutamiseks. Nende lubamine osutub samaväärseks üleminekuga intuitsionistlikult arvutuselt klassikalisele.

Kirjeldame nüüd Gentzeni arvutust.

- *Keel.* Tavaliselt formuleeritakse sekventsiaalarvutus sekventside jaoks, milles valemid sisaldavad loogilisi seoseid &, $\vee$, $\supset$, $\neg$, $\forall$ ja $\exists$.
- *Aksioomideks* loetakse sekventse kujul $A \Rightarrow A$.
- *Tuletusreegleid* on kahte liiki: loogilised (seoste sissetoomise) ja struktuursed.

Reeglite üleskirjutuses tähistavad suured ladina tähed valemeid, täht x seotud muutujat, a vaba muutujat, t termi (mis muidugi peab olema vaba vastavale kohale asendamiseks), kreeka tähed komadega eraldatud valemite jadasid. Reeglites ($\Rightarrow\forall$) ja ($\exists\Rightarrow$) ei ole muutuja a vabalt Γ ja Θ valemites.

Põhilised tuletamisvahendid on koondatud loogilistesse reeglitesse. Iga seose jaoks on Gentzeni süsteemis selle seose antetsedenti ja suktsedenti sissetoomise reegel. See võimaldab saada sekventse, mis sisaldavad vastavat seost vasakul või paremal. Antetsedenti sissetoomise reeglid näitavad, kuidas saab tõestada teoreeme, mille eeldus on konjunktsioon, disjunktsioon jne. Suktsedenti sissetoomise reeglid annavad vahendi juhuks, kui teoreemi väide on konjunktsioon, disjunktsioon jne. Oma sisu poolest formaliseerib iga reegel mingit tuntud tõestamisvõtet. Näiteks disjunktsiooni

$$\frac{A, \Gamma \Rightarrow \Theta}{A \,\&\, B, \Gamma \Rightarrow \Theta}, \frac{B, \Gamma \Rightarrow \Theta}{A \,\&\, B, \Gamma \Rightarrow \Theta}\ (\&\Rightarrow) \qquad \frac{\Gamma \Rightarrow \Theta, A \quad \Gamma \Rightarrow \Theta, B}{\Gamma \Rightarrow \Theta, A \,\&\, B}\ (\Rightarrow\&)$$

$$\frac{A, \Gamma \Rightarrow \Theta \quad B, \Gamma \Rightarrow \Theta}{A \vee B, \Gamma \Rightarrow \Theta}\ (\vee\Rightarrow) \qquad \frac{\Gamma \Rightarrow \Theta, A}{\Gamma \Rightarrow \Theta, A \vee B}, \frac{\Gamma \Rightarrow \Theta, B}{\Gamma \Rightarrow \Theta, A \vee B}\ (\Rightarrow\vee)$$

$$\frac{\Gamma \Rightarrow \Theta, A \quad B, \Delta \Rightarrow \Lambda}{A \supset B, \Gamma, \Delta \Rightarrow \Theta, \Lambda}\ (\supset\Rightarrow) \qquad \frac{A, \Gamma \Rightarrow \Theta, B}{\Gamma \Rightarrow \Theta, A \supset B}\ (\Rightarrow\supset)$$

$$\frac{\Gamma \Rightarrow \Theta, A}{\neg A, \Gamma \Rightarrow \Theta}\ (\neg\Rightarrow) \qquad \frac{A, \Gamma \Rightarrow \Theta}{\Gamma \Rightarrow \Theta, \neg A}\ (\Rightarrow\neg)$$

$$\frac{A(x/t), \Gamma \Rightarrow \Theta}{\forall x\, A(x), \Gamma \Rightarrow \Theta}\ (\forall\Rightarrow) \qquad \frac{\Gamma \Rightarrow \Theta, A(x/a)}{\Gamma \Rightarrow \Theta, \forall x\, A(x)}\ (\Rightarrow\forall)$$

$$\frac{A(x/a), \Gamma \Rightarrow \Theta}{\exists x\, A(x), \Gamma \Rightarrow \Theta}\ (\exists\Rightarrow) \qquad \frac{\Gamma \Rightarrow \Theta, A(x/t)}{\Gamma \Rightarrow \Theta, \exists x\, A(x)}\ (\Rightarrow\exists)$$

Tabel 6.2. Loogilised reeglid.

antetsedenti toomise reegel ($\vee\Rightarrow$) tähendab, kui eeldustest Γ ja valemist A järeldub Θ ning eeldustest Γ ja valemist B järeldub Θ, siis järeldub Θ juba eeldustest ja disjunktsioonist $A \vee B$.

Peale *lõikereegli* (Cut) on struktuursed reeglid oma sisult küllalt triviaalsed. Sekventsiga esitatud lauses lubatakse lisada eeldusi või väite alternatiive, eemaldada eelduste ja väite alternatiivide liigseid eksemplare, muuta eelduste või alternatiivide järjekorda. Lõikereegel formaliseerib matemaatikas tihti kasutatava strateegia, kus tõestuse ühes osas saadakse mingi lemma C ja teises osas saadakse lemmat C kasutades teoreem. Eespool näidetes vaadeldud konjunktsiooni kommutatiivsuse teoreemi on sekventsina loomulik kirja panna kujul $P \,\&\, Q \supset Q \,\&\, P$ ja tema tõestus koosneb Gentzeni süsteemis kolmest sammust:

$$\frac{\Gamma \Rightarrow \Theta}{C, \Gamma \Rightarrow \Theta} \qquad \frac{\Gamma \Rightarrow \Theta}{\Gamma \Rightarrow \Theta, C}$$

$$\frac{C, C, \Gamma \Rightarrow \Theta}{C, \Gamma \Rightarrow \Theta} \qquad \frac{\Gamma \Rightarrow \Theta, C, C}{\Gamma \Rightarrow \Theta, C}$$

$$\frac{\Delta, C, D, \Gamma \Rightarrow \Theta}{\Delta, D, C, \Gamma \Rightarrow \Theta} \qquad \frac{\Gamma \Rightarrow \Lambda, C, D, \Theta}{\Gamma \Rightarrow \Lambda, D, C, \Theta}$$

$$\frac{\Delta \Rightarrow \Lambda, C \quad C, \Gamma \Rightarrow \Theta}{\Delta, \Gamma \Rightarrow \Lambda, \Theta}\ (\text{Cut})$$

Tabel 6.3. Struktuursed reeglid.

$$\cfrac{\cfrac{Q \Rightarrow Q}{P \,\&\, Q \Rightarrow Q}\ (\&\Rightarrow) \qquad \cfrac{P \Rightarrow P}{P \,\&\, Q \Rightarrow P}\ (\&\Rightarrow)}{P \,\&\, Q \Rightarrow Q \,\&\, P}\ (\Rightarrow\&)$$

Olulisem kui sammude arv on aga asjaolu, et tuletus on leitav ainult rutiinseid samme kasutades. Strateegia seisneb siin tuletuspuu ehitamises alt üles, s.t. sihtsekventsist aksioomide poole. Igal sammul püütakse lahutada sekventsis mingi valem osavalemiteks, kasutades selle valemi peatehte sissetoomise reeglit. Lausearvutuse jaoks ongi see strateegia koos mittetautoloogiliste sekventside vältimisega piisav. Näiteks ülaltoodud tõestuses me ei saanud sihtsekventsile rakendada kumbagi reeglitest ($\&\Rightarrow$), sest sekventsid $P \Rightarrow Q \,\&\, P$ ja $Q \Rightarrow Q \,\&\, P$ pole samaselt tõesed ja neid pole lootust tuletada. Sellegipoolest saab tuletamist alustada antetsedendi käsitlemisega, tuues sinna valemi $P \,\&\, Q$ teise eksemplari ja kaotades seejärel kummastki konjunktsioonist ühe liikme. Predikaatloogikaga on olukord muidugi keerulisem, sest kvantorreeglites pole ülespoole liikudes termi valik ühene, samuti pole olemas algoritmi mittetuletatavate harude vältimiseks.

Toome siin ka ühe predikaattuletuse näite, tuletades $\Rightarrow \forall x\, P(x) \vee \exists x \neg P(x)$. Märgime, et kvantorireeglite rakendamise järjekord on oluline, sest reeglit ($\Rightarrow\forall$) saab rakendada ainult

siis, kui teised valemid sekventsis ei sisalda vaba muutujat a.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(a) \Rightarrow P(a)}{\Rightarrow P(a), \neg P(a)}\,(\Rightarrow\neg)}{\Rightarrow P(a), \exists x\, \neg P(x)}\,(\Rightarrow\exists)}{\Rightarrow \exists x\, \neg P(x), P(a)}}{\Rightarrow \exists x\, \neg P(x), \forall x\, P(x)}\,(\Rightarrow\forall)}{\Rightarrow \exists x\, \neg P(x), \forall x\, P(x) \vee \exists x\, \neg P(x)}\,(\Rightarrow\vee)}{\Rightarrow \forall x\, P(x) \vee \exists x\, \neg P(x), \exists x\, \neg P(x)}}{\Rightarrow \forall x\, P(x) \vee \exists x\, \neg P(x), \forall x\, P(x) \vee \exists x\, \neg P(x)}\,(\Rightarrow\vee)$$
$$\Rightarrow \forall x\, P(x) \vee \exists x\, \neg P(x)$$

Gentzen tõestas kõige tähtsama tulemusena oma arvutuse kohta, et *lõikereegli* võib süsteemist välja jätta, sest tema rakendused saab igast tuletusest *elimineerida*. Saadud arvutuses kehtib *alamvalemi omadus*: igal tuletussammul on iga valem reegli ülemises sekventsis (ülemistes sekventsides) alumise sekventsi mingi valemi osavalem. Automaatse tõestamise jaoks tähendab see, et tuletuse otsimispuus on hargnemistegur tõkestatud. Tõestuste teoorias saab aga väga lihtsalt tõestada lõikevaba arvutuse mittevasturäkivust: näiteks lihtsaim sekvents $\Rightarrow$ ei saa olla tuletatav, sest ükski aksioom ei koosne tema valemite osavalemitest. Ka paljude konkreetsete valemite (s.o. kujul $\Rightarrow A$ olevate sekventside) mittetuletatavust on Gentzeni tüüpi arvutuse korral lihtne näidata, sest on olemas ainult väike arv reegleid, mille rakendamise tulemusena selline sekvents saab tekkida.

Puhtas predikaatloogikas kandub alamvalemi omadus kahjuks edasi ainult nendesse rakendustesse, kus aksioomid on lihtsa ehitusega valemid. Tüüpiline negatiivne näide on naturaalarvude aritmeetika, kus induktsioonireeglil alamvalemi omadust ei ole ja valemi $\forall x\, A(x)$ tõestamiseks tuleb tõestada valemid $A(0)$ ja $\forall x\, [A(x) \supset A(x+1)]$, millest viimane on oluliselt keerulisema ehitusega kui esialgne. See teeb tuletuse automaatse otsingu aritmeetikas väga raskeks.

6.4. Tõesuspuu

Tõesuspuu on välja kasvanud *Bethi tabelimeetodist*, kus tõesed ja väärad valemid koondatakse omaette tabelitesse, millel võivad omakorda olla alamtabelid. Kuna väära valemi eitus on tõene, siis me võime väärade valemite tabelid ära kaotada, tõstes väärade valemite eitused tõeste valemite tabelitesse. Tulemuseks on hargnemistega puu, mis püüab kehtestatava valemi korral säilitada mingis harus tema tõesust.

Tõesuspuude tuletusmeetod põhineb vastuväitelisel tõestamisel, sest sellega kontrollitakse, kas on võimalik olukord, kus arutluse eeldused on tõesed ja järeldus on väär (või kas lause saab olla loogiliselt väär). Eesmärgiks on esialgsete valemite *dekompositsioon* võimalikult lihtsateks valemiteks — literaalideks, s.t. atomaarseteks valemiteks ja nende eitusteks. Dekompositsiooni aluseks on semantikareeglid. Näiteks konjunktsiooni semantika

$$p \,\&\, q = t \Leftrightarrow p = t \quad \text{ja} \quad q = t$$

põhjal on konjunktsiooni dekompositsiooni tulemuseks tema mõlemad liikmed: p ja q. De Morgani seaduste põhjal kehtib

$$\neg(p \,\&\, q) = \neg p \vee \neg q,$$

s.t. konjunktsiooni eitus on samaväärne eituste disjunktsiooniga. Disjunktsioonile vastab puus hargnemine, sest tema tõesuseks piisab ainult tema ühe alamvalemi tõesusest. Selliselt saame tõesuspuude tuletusreeglid, mis on koondatud tabelisse 6.4[1]. Üldisuskvantori dekompositsioonireeglis ($\forall D$) on a suvaline vaadeldavas puuharus esinev konstant (või mingi fikseeritud konstant a_0, kui harus pole veel ühtegi konstanti). Eksistentsikvantori dekompositsioonireeglis ($\exists D$) valitakse konstandiks c puuharus mitteesinev (uus) konstant. Tavaliselt piisab valemite ühekordsest dekompositsioonist ning kasutatud valemid märgistatakse sümboliga $\surd$. Kuid üldisuskvantori dekompositsioonireeglit ($\forall D$) on võimalik kasu-

[1] Vahel nimetatakse disjunktsiooni dekompositsioonireegli sarnaseid hargnemistega reegleid β-reegliteks ja konjunktsiooni dekompositsioonireegli tüüpi järjestikuseid reegleid α-reegliteks.

	$\neg\neg p \quad (\neg\neg D)$ p
$p \,\&\, q \quad (\&D)$ p q	$\neg(p \,\&\, q) \quad (\neg\&D)$ $\neg p \qquad \neg q$
$p \vee q \quad (\vee D)$ $p \qquad q$	$\neg(p \vee q) \quad (\neg\vee D)$ $\neg p$ $\neg q$
$p \supset q \quad (\supset D)$ $\neg p \qquad q$	$\neg(p \supset q) \quad (\neg\supset D)$ p $\neg q$
$p \equiv q \quad (\equiv D)$ $p \qquad \neg p$ $q \qquad \neg q$	$\neg(p \equiv q) \quad (\neg\equiv D)$ $p \qquad \neg p$ $\neg q \qquad q$
$\forall x\ p \quad (\forall D)$ $p\,\{x/a\}$	$\neg\forall x\ p \quad (\neg\forall D)$ $\exists x \neg p$
$\exists x\ p \quad (\exists D)$ $p\,\{x/c\}$	$\neg\exists x\ p \quad (\neg\exists D)$ $\forall x \neg p$

Tabel 6.4. Tõesuspuude tuletusreeglid.

tada järgemööda kõigi vaadeldavas puuharus esinevate konstantide korral, mistõttu selle reegli rakendamisel jäetakse esialgne valem märgistamata.

Puuharu, milles on mingi aatom koos iseenda eitusega, nimetatakse *suletud haruks* ning tähistatakse sümboliga $\times$. Puuharu, mis pole suletud, kuid mis on lõpetatud, s.t. kõik temas esinevad valemid on lahutatud dekompositsioonireeglite abil literaalideks, nimetatakse *avatud lõpetatud haruks*. Puud nimetatakse *suletuks*, kui tema kõik harud on suletud, ja *avatuks*, kui temas leidub avatud lõpetatud haru.

Tähistagu $\vdash_{TT} p$ valemi $\neg p$ tõesuspuu suletust.

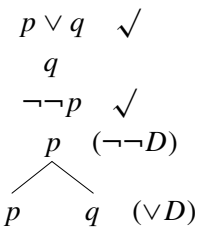
Lause 6.4. $\vdash_{TT} p \,\&\, q \supset q \,\&\, p$.

Tõestus. Kontrollime vastuväiteliselt, kas valemi $p \,\&\, q \supset q \,\&\, p$ eitus võib olla tõene:

1.	$\neg(p \,\&\, q \supset q \,\&\, p) \quad \surd$	eeldus
2.	$p \,\&\, q \quad \surd$	1 $\neg\supset D$
3.	$\neg(q \,\&\, p) \quad \surd$	1 $\neg\supset D$
4.	p	2 $\&D$
5.	q	2 $\&D$
6.	$\neg q \qquad \neg p$ $\times \qquad \times$	3 $\neg\&D$

Saadud suletud tõesuspuu näitab, et valemi $p \,\&\, q \supset q \,\&\, p$ eitus annab vastuolu. Järelikult on see valem tõene. □

Kuna tõesuspuude meetod on kergesti programmeeritav, sobib see hästi nii teoreemide tõestamiseks kui ka valemite hulkade mudelite konstrueerimiseks. Näiteks saab tõesuspuudega leida kontranäite mittekehtivale arutlusele $p \vee q, q \models \neg p$:



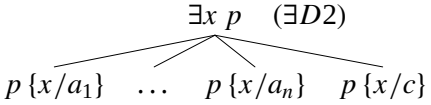
Selles puus on kaks avatud lõpetatud haru, mille literaalid annavad mõlemal juhul väärtustuse $\{p, q\}$ ($p = t$ ja $q = t$). See lükkab vaadeldava arutluse kehtivuse ümber.

Lausearvutusvalemite korral on tõesuspuud alati lõplikud, sest dekompositsioonireeglit rakendatakse igale valemile maksimaalselt üks kord. Kuna üldisuskvantori dekompositsioonireeglit saab korduvalt rakendada, võivad predikaatarvutuse valemite tõesuspuud osutuda lõpmatuks. Näiteks valemi $\forall x \exists y\, Pxy$ tõesuspuu

1.	$\forall x \exists y\, Pxy$		eeldus
2.	$\exists y\, Pay$	$\surd$	1 $\forall D$
3.	Pab		2 $\exists D$
4.	$\exists y\, Pby$	$\surd$	1 $\forall D$
5.	Pbc		4 $\exists D$
6.	$\exists y\, Pcy$	$\surd$	1 $\forall D$
7.	Pcd		6 $\exists D$
	$\ldots$		

ainus haru on lõpmatu, sest eksistentsikvantori dekompositsioonireegel lisab pidevalt uue konstandi ning üldisuskvantori dekompositsioonireegliga genereeritakse selle konstandi abil uus eksistentsikvantoriga valem. Tulemuseks on tõesuspuu, milles puudub avatud lõpetatud haru.

Tõesuspuude tuletusmeetodit saab täiustada, võttes eksistentsikvantori dekompositsioonireegli ($\exists D$) asemel kasutusele reegli



mis püüab muutujat x asendada kõigi puuharus olevate konstantidega $a_1, \ldots, a_n$ ning toob seejärel sisse harus puuduva konstandi c. Selle reegliga saame valemi $\forall x \exists y\, Pxy$ jaoks lõpmatu hargneva puu

1.	$\forall x \exists y\, Pxy$		eeldus
2.	$\exists y\, Pay$	$\surd$	1 $\forall D$
3.	Paa Pab		2 $\exists D2$
	$\ldots$		

milles esinevad ka avatud lõpetatud harud. Näiteks valemiga Paa lõppev haru genereerib valemi $\forall x \exists y\, Pxy$ mudeli, milles on ainult üks element a ning Paa on tõene. Ent valemite korral, millel on ainult lõpmatud mudelid, ei ole ka täiustatud reeglite korral võimalik piirduda tõesuspuu lõpliku fragmendi läbivaatamisega.

Ülesanded

1. ***Konstrueerige valemite hulk, millel on ainult lõpmatud mudelid.

Kuna tõesuspuus on võimalik dekompositsioonireegleid rakendada erinevas järjekorras, siis antakse tihti ka soovitused, milliseid reegleid rakendada esimesena. Tõesuspuude mõõtmeid aitab vähendada *strateegia*, mille korral sooritatakse alguses kõik järjestikused tuletusreeglid ning seejärel kõik hargnevad reeglid. Kvantorireeglitest tuleks viimaseks jätta reegli ($\forall D$) rakendamine.

Saab tõestada, et tõesuspuude meetod on *täielik*, s.t. kõik, mis on predikaatarvutuses loogiliselt järelduv, on ka tõesuspuudega tuletatav ja vastupidi:

$$\Gamma \models p \Leftrightarrow \Gamma \vdash_{TT} p.$$

Puuharusid võib sulgeda ka mitte ainult vasturääkivate literaalide A ja $\neg A$ esinemisel, vaid ka vasturääkivate valemite p ja $\neg p$ korral. Mõlemad tuletusmeetodid on ekvivalentsed, sest valemid p ja $\neg p$ ei saa olla samaaegselt tõesed. Kahjuks ei anna selline tuletusmeetodi täiustus meile avatud lõpetatud puuharu põhjal automaatselt atomaarsete valemite kehtestavat väärtustust.

II

Matemaatiline loogika

7. Matemaatiline induktsioon

Matemaatika on mäng, mida mängitakse lihtsate reeglite järgi, kirjutades paberile mõttetuid sümboleid.

— David Hilbert (1862–1943)

7.1. Induktiivne tõestus

Matemaatiline induktsioon on võimas matemaatiline meetod, mis võimaldab opereerida lõpmatute hulkadega. Ta aitab meil koostada lõpmatuid objektide klasse ning seejärel saab temaga tõestada, et sellistel klassidel on kindlad abstraktsed omadused.

Näide 7.1. Naturaalarvude hulk

$$\mathbb{N} = \{0, 1, 2, \ldots\}$$

defineeritakse induktiivselt kahe punktiga:

1. 0 on naturaalarv.
2. Kui n on naturaalarv, siis on $n + 1$ naturaalarv.

Esimese lausega defineeritakse *induktsiooni baas* ja teisega määratakse *induktsiooni samm*.

Näide 7.2. Lausearvutuse süntaksi definitsioonis lk. 76 on induktsiooni baasiks kõik atomaarsed valemid, mis koosnevad ainult lausemuutujast, ja induktsiooni sammuga kirjeldatakse järkjärgult lausearvutuse liitlaused.

Induktsiooniga on mugav tõestada lausearvutuse (kui lõpmatu valemite hulga) matemaatilisi omadusi. Tõestame näiteks sellise "lihtsa" teoreemi.

Teoreem 7.3. *Igas lausearvutuse valemis on vasakpoolsete sulgude arv võrdne parempoolsete sulgude arvuga.*

Tõestus. Tõestame teoreemi induktsiooniga üle kõigi lausearvutusvalemite. Teoreem kehtib atomaarsete valemite korral, sest neis sulud puuduvad. Teoreem kehtib ka valemite korral, mis on saadud atomaarsetest valemitest ühe lausearvutuse tehte lisamise teel, sest olemasolevatest valemitest uue valemi konstrueerimisel lisatakse ühed paarissulud või ei lisata ühtegi sulgu. Sellised on näiteks valemid

$$\neg A,\ (A \supset B),\ (A \,\&\, B).$$

Kasutades lausearvutuse süntaksi definitsiooni teist korda, saame tõestada, et teoreem kehtib kõigi lausearvutusvalemite korral, milles on kaks loogilist tehet:

$$\neg\neg A,\ \neg(A \vee B),\ (A \vee \neg B),\ ((A \equiv B) \supset C), \ldots$$

Suurendades niimoodi pidevalt valemites esinevate loogiliste tehete arvu, tõestame (lõpmatu tõestuse) tulemusena, et teoreem kehtib kõigi lausearvutusvalemite korral.

Kuna lõpmatut tõestust ei ole mõtet kirjutada, siis tõestatakse kahes etapis. Kõigepealt tõestatakse *induktsiooni baas*:

- Atomaarsetes lausearvutusvalemites on vasakpoolsete sulgude arv võrdne parempoolsete sulgude arvuga,

ning seejärel tõestatakse *induktsiooni samm*:

- Kui igas lausearvutusvalemis, kus on kuni k loogilist tehet, on vasakpoolsete sulgude arv võrdne parempoolsete sulgude arvuga, siis on ka kõigis $k+1$ loogilise tehtega lausearvutusvalemites vasakpoolsete sulgude arv võrdne parempoolsete sulgude arvuga.

Induktsiooni baas ja samm tõestatakse lausearvutuse süntaksi definitsiooni põhjal. Teoreemi väite tõestus saadakse siit aga *matemaatilise induktsiooni printsiibil*, sest teoreemi väide järeldub loogiliselt induktsiooni baasi ja induktsiooni sammu väidetest. □

Matemaatilise induktsiooni meetod töötab ainult selliste hulkade korral, mida saab grupeerida järjestatud rühmade jadaks. Esimesel induktsiooni sammul tõestatakse, et esimese rühma kõik liikmed on antud omadusega. Igal järgmisel induktsiooni sammul tõestatakse, et ka eelmistele vahetult järgneva rühma elementidel on vaadeldav omadus. Näiteks eelmise teoreemi korral oli rühmitamise aluseks valemis esinevate loogiliste tehete arv.

Ülesanded

1. Tõestage, et induktsiooni baas, induktsiooni samm ja teoreemi väide moodustavad teoreemis 7.3 kehtiva arutluse. Millise loogika arutlus see on?
2. Tõestage matemaatilise induktsiooniga, et kõik inimesed on kas mees- või naissoost.

7.2. Duaalsus

Lausearvutusvalemites kasutatakse tavaliselt eitus-, konjunktsiooni-, disjunktsiooni-, implikatsiooni- ja ekvivalentsitehteid. Sellisel juhul öeldakse, et valemid on (tehete) *baasis* $\{\neg, \&, \vee, \supset, \equiv\}$. Kui kasutatakse ainult kolme tehet: eitust, konjunktsiooni ja disjunktsiooni, on valemid baasis $\{\neg, \&, \vee\}$. Niisugustel valemitel on olemas ka duaalsed valemid. Valemi *p duaalne valem* p' saadakse, kui kõik konjunktsioonid asendatakse disjunktsioonidega ja kõik disjunktsioonid konjunktsioonidega ning kõik atomaarsed valemid nende eitustega. Tabelis 7.1 on toodud mõned näited valemitest ja nende duaalsetest valemitest.

Valem p	Duaalne valem p'
A	$\neg A$
$((A \vee F) \,\&\, G)$	$((\neg A \,\&\, \neg F) \vee \neg G)$
$\neg((A \vee \neg B) \vee (\neg A \,\&\, \neg B))$	$\neg((\neg A \,\&\, \neg\neg B) \,\&\, (\neg\neg A \vee \neg\neg B))$

Tabel 7.1. Näiteid valemitest ja nende duaalsetest valemitest.

Järgmine teoreem väidab, et valem ja tema duaalne valem on üksteisele vasturääkivad.

Teoreem 7.4 (duaalsus). *Olgu p valem baasis $\{\neg, \&, \vee\}$ ja p' tema duaalne valem. Siis kehtib*

$$p = \neg p'.$$

Tõestus. Induktsiooni baas. Olgu p atomaarne valem. Siis saame kahekordse eituse reegli ja duaalsuse definitsiooni põhjal

$$p = \neg\neg p = \neg p'.$$

Induktsiooni samm.

1. Olgu $p = \neg q$. Induktsiooni eelduse põhjal kehtib $q = \neg q'$. Siit saame aga

   $$p = \neg q = \neg\neg q' = \neg p',$$

   sest duaalsuse definitsiooni põhjal kehtib $p' = \neg q'$.

2. Olgu $p = q \,\&\, r$. Induktsiooni põhjal kehtib

   $$q = \neg q', \quad r = \neg r'$$

   ning duaalsuse definitsioonist saame

   $$p' = q' \vee r'.$$

   De Morgani seadused annavad nüüd

   $$\begin{aligned} p = q \,\&\, r &= \neg(\neg q \vee \neg r) = \neg(\neg\neg q' \vee \neg\neg r') \\ &= \neg(q' \vee r') = \neg p'. \end{aligned}$$

3. $p = q \vee r$. Selle punkti jätame tõestada ülesandes 7.2.2. □

Järeldus 7.5. *Kui valemid p ja q on loogiliselt ekvivalentsed, siis on ka nende duaalsed valemid p' ja q' loogiliselt ekvivalentsed:*

$$p = q \Rightarrow p' = q'.$$

Palju asendusreegleid saab duaalsusteoreemi ja asenduste abil omavahel teisendada. Konjunktsiooni kommutatiivsusest võib sellisel teel saada näiteks disjunktsiooni kommutatiivsuse:

$$p \,\&\, q = q \,\&\, p \Rightarrow p \vee q = q \vee p.$$

Ülesanded

1. *Näidake, et iga valem baasis $\{\&, \vee\}$ on tõene, kui kõik tema lausemuutujad on tõesed. Järelikult ei saa eitust selles baasis väljendada.

2. Tõestage disjunktsiooni osa duaalsusteoreemis.

8. Lausearvutuse täielikkus

Formaalse aksiomaatilise süsteemi täielikkus tähendab tõdemust, et selles süsteemis langevad tõesuse ja tuletatavuse mõiste kokku. Ideaalis peaks iga loogilise süsteemi korral leiduma ka vastav täielik tuletussüsteem. Praktikas tekitab loogiliste süsteemide semantika ja süntaktiliste tuletusreeglite vastavusse seadmine küllalt suuri probleeme: kas ei ole konstrueeritud tuletusreeglitel selget tähendust või ei osata kirjeldatud semantikat esitada tuletusreeglite abil. Näiteks tekivad tehisintellektisüsteemides kasutatavate praktiliste tuletussüsteemide täielikkuse tagamisel ületamatud probleemid. Klassikaliste lause- ja predikaatarvutuse täielikkus on aga väljaspool kahtlust. Järgnevalt tõestame Hilberti tüüpi tuletussüsteemi H täielikkust klassikalise lausearvutuse jaoks. Tõestust saab laiendada traditsiooniliste vahenditega ka predikaatarvutusele.

Täielikkuseteoreemi tõestus koosneb kahest osast:

- *Korrektsuse* osas näidatakse, et iga tuletatav valem on loogiliselt tõene.
- *Täielikkuse* osas näidatakse, et kehtib ka vastupidine: iga loogiliselt tõest valemit saab tuletada.

Me tõestame, et Hilberti tüüpi tuletussüsteem H on lausearvutuse jaoks korrektne ja täielik:

$$\vdash_H p \Leftrightarrow \models p.$$

8.1. Lausearvutuse korrektsus ja mittevasturääkivus

Teoreem 8.1 (korrektsus). *Iga lausearvutusvalemi $p \in L_H$ korral*

$$\vdash_H p \Rightarrow \models p.$$

Tõestus. Olgu $q_1, \ldots, q_n = p$ valemi p tuletus formaalses süsteemis H. Näitame induktsiooniga, et iga valem q_i $(1 \leq i \leq n)$ on tautoloogia.

Kui q_i on aksioom, on lihtne kontrollida, et ta on tautoloogia. Oletame, et valem q_i on saadud valemitest q_j ja q_m *modus ponens*'iga, kus $j, m < i$ ja $q_m = q_j \supset q_i$. Induktsiooni eelduse põhjal on valemid q_j ja q_m tautoloogiad. Oletame vastuväiteliselt, et q_i ei ole tautoloogia. Siis leidub nende valemite selline väärtustus α, et $\alpha(q_j) = t$, $\alpha(q_j \supset q_i) = t$ ja $\alpha(q_i) = v$. Siit järeldub aga, et $\alpha(q_j \supset q_i) = v$, mis annab vastuolu. Võttes $i = n$ saame, et valem p on tautoloogia. □

Definitsioon 8.2. Aksiomaatilist teooriat $\mathcal{T}$ nimetatakse *vasturääkivaks*, kui leidub selline valem p, nii et teoorias $\mathcal{T}$ saab tuletada p ja $\neg p$.

Teoreem 8.3. *Lausearvutus H on mittevasturääkiv.*

Tõestus. Oletame, et mingi valemi p korral saab süsteemis H tuletada nii p kui ka $\neg p$. Korrektsuseteoreemi põhjal on nad mõlemad siis tautoloogiad, mis ei ole võimalik. □

8.2. Lausearvutuse täielikkus

Tähistagu p^t valemit p ja p^v valemit $\neg p$. Vahetult saab kontrollida, et iga $\alpha \in \{t, v\}$ korral

$$p^\alpha = t \Leftrightarrow p = \alpha.$$

Süsteemi H täielikkuse tõestuses kasutame järgmist lemmat.

Lemma 8.4. *Olgu $X_1, \ldots, X_n$ valemis p esinevad lausemuutujad ja olgu α suvaline p väärtustus. Siis kehtib*

$$X_1^\alpha, \ldots, X_n^\alpha \vdash_H p^\alpha.$$

Et lemma väitest aru saada, vaatleme kõigepealt järgmist näidet. Olgu $p = X \supset \neg Y$. Vaatleme väärtustusi (t, t) ja (t, v). Nendele väärtustustele vastavad lemmas hüpoteeside loetelud X, Y ja $X, \neg Y$. Esimesel väärtustusel on p väär ja järelikult väidab lemma, et $X, Y \vdash_H \neg(X \supset \neg Y)$. Teisel juhul on p tõene ja siis peab olema tuletatav $X, \neg Y \vdash_H X \supset \neg Y$. Lemmat võime mõista ka nii, et valemi tõeväärtus on süsteemis H arvutatav, kui kasutada andmetena väärtustele vastavaid hüpoteese.

Tõestus. Tõestame lemma induktsiooniga valemi p ehituse järgi. Olgu selles m loogilist seost.

Juhul kui $m = 0$, on p aatomaarne valem. On selge, et kui $\alpha(p) = t$, siis $p \vdash_H p$, ja kui $\alpha(p) = v$, siis $\neg p \vdash_H \neg p$.

Eeldame, et lemma väide kehtib iga $j < m$ korral. Näitame, et see kehtib ka valemite korral, milles on m loogilist seost. Selleks vaatame läbi kõikvõimalikud variandid.

1. Olgu valem p kujul $q \supset r$ ja olgu $X_1, \ldots, X_k$ temas esinevad lausemuutujad. Kuna valemites q ja r on loogiliste seoste arv väiksem kui m, siis saame induktsiooni eelduse põhjal $X_1^\alpha, \ldots, X_k^\alpha \vdash_H q^\alpha$ ja $X_1^\alpha, \ldots, X_k^\alpha \vdash_H r^\alpha$.

1a. Kui $\alpha(q) = \alpha(r) = t$, siis $q^\alpha = q$ ja $r^\alpha = r$. Kuna $\alpha(q \supset r) = t$, siis $p^\alpha = q \supset r$. Eelduse põhjal kehtib $X_1^\alpha, \ldots, X_k^\alpha \vdash_H r$ ja aksioomiskeem $(H1)$ annab $\vdash_H r \supset (q \supset r)$. Siit saame *modus ponens*'iga $X_1^\alpha, \ldots, X_k^\alpha \vdash_H q \supset r$, s.t. $X_1^\alpha, \ldots, X_k^\alpha \vdash_H p^\alpha$.

1b. Kui $\alpha(q) = t$ ja $\alpha(r) = v$, siis $q^\alpha = q$ ja $r^\alpha = \neg r$. Nüüd on $\alpha(q \supset r) = v$ ja $p^\alpha = \neg(q \supset r)$. Eelduse põhjal kehtivad $X_1^\alpha, \ldots, X_k^\alpha \vdash_H q$ ja $X_1^\alpha, \ldots, X_k^\alpha \vdash_H \neg r$. Saab tõestada, et $\vdash_H q \supset (\neg r \supset \neg(q \supset r))$. Rakendades sellele valemile kaks korda *modus ponens*'it, saame $X_1^\alpha, \ldots, X_k^\alpha \vdash_H \neg(q \supset r)$, s.t. $X_1^\alpha, \ldots, X_k^\alpha \vdash_H p^\alpha$.

1c. Kui $\alpha(q) = v$, siis $q^\alpha = \neg q$ ja suvalise r väärtustuse korral $\alpha(q \supset r) = t$, mistõttu $p^\alpha = q \supset r$. Eelduse põhjal kehtib $X_1^\alpha, \ldots, X_k^\alpha \vdash_H \neg q$. Saab tõestada, et $\vdash_H \neg q \supset (q \supset r)$. Siit saame *modus ponens*'iga $X_1^\alpha, \ldots, X_k^\alpha \vdash_H q \supset r$, s.t. $X_1^\alpha, \ldots, X_k^\alpha \vdash_H p^\alpha$.

2. Olgu valem p kujul $\neg q$. Kuna valemis q on loogiliste seoste arv väiksem kui m, siis saame induktsiooni eelduse põhjal $X_1^\alpha, \ldots, X_k^\alpha \vdash_H q^\alpha$.

2a. Kui $\alpha(q) = t$, siis $q^\alpha = q$, $\alpha(p) = v$ ja $p^\alpha = \neg p$. Kuna $\vdash_H q \supset \neg\neg q$, siis saame *modus ponens*'iga $X_1^\alpha, \ldots, X_k^\alpha \vdash_H \neg\neg q$. Kuid $\neg\neg q = p^\alpha$, mis annabki $X_1^\alpha, \ldots, X_k^\alpha \vdash_H p^\alpha$.

2b. Kui $\alpha(q) = v$, siis $q^\alpha = \neg q$, $\alpha(p) = t$ ja $p^\alpha = p$. Eelduse põhjal kehtib $X_1^\alpha, \ldots, X_k^\alpha \vdash_H \neg q$, mis annabki $X_1^\alpha, \ldots, X_k^\alpha \vdash_H p^\alpha$.

Sellega on kõik variandid läbi vaadatud ja lemma tõestatud. □

Teoreem 8.5 (täielikkus). *Iga lausearvutusvalemi $p \in L_H$ korral*

$$\models p \Rightarrow \vdash_H p.$$

Tõestus. Olgu p tautoloogia ja olgu $X_1, \ldots, X_n$ temas esinevad lausemuutujad. Iga väärtustuse α korral kehtib $p^\alpha = p$. Lemmast 8.4 saame, et $X_1^\alpha, \ldots, X_n^\alpha \vdash_H p$. Järelikult $X_1^\alpha, \ldots, X_{n-1}^\alpha, X_n \vdash_H p$ ja $X_1^\alpha, \ldots, X_{n-1}^\alpha, \neg X_n \vdash_H p$.

Deduktsiooni teoreemi järgi saame $X_1^\alpha, \ldots, X_{n-1}^\alpha \vdash_H X_n \supset p$ ja $X_1^\alpha, \ldots, X_{n-1}^\alpha \vdash_H \neg X_n \supset p$. On võimalik tõestada, et $\vdash_H (X_n \supset p) \supset ((\neg X_n \supset p) \supset p)$, kust saame *modus ponens*'iga $X_1^\alpha, \ldots, X_{n-1}^\alpha \vdash_H p$. Korrates seda protsessi $n-1$ korda, saamegi tulemuseks $\vdash_H p$. □

Ülesanded

1. Tõestage, et
   (a) $\vdash_H q \supset (\neg r \supset \neg(q \supset r))$;
   (b) $\vdash_H \neg q \supset (q \supset r)$;
   (c) $\vdash_H q \supset \neg\neg q$;
   (d) $\vdash_H (q \supset p) \supset ((\neg q \supset p) \supset p)$.

9. Mittelahenduvad probleemid

> *Matemaatikast ei ole võimalik aru saada. Temaga saab vaid kohaneda.*
>
> — John von Neumann (1903–1957)

Matemaatikas on olnud alati palju lahendamata probleeme, s.t. ülesandeid, mida mitte keegi ei ole suutnud lahendada. Näiteks võib tuua Fermat' suure teoreemi:

- Võrrandil $x^n + y^n = z^n$ ei ole $n > 2$ korral positiivseid täisarvulisi lahendeid.

Prantsuse matemaatik Pierre de Fermat (1601–1665) kirjutas omal ajal raamatu servale, et ta teab, kuidas seda väidet tõestada. Tõestuse kirjapanekuks kulus matemaatikutel mitusada aastat. Abiks võeti ka arvutid, kuid pingsast otsimisest hoolimata ei õnnestunud leida ühtegi arvude kombinatsiooni, mis oleks Fermat' teoreemi ümber lükanud. Alles 1994. a. esitas Ameerika matemaatik Andrew Wiles Fermat' teoreemi piisavalt üksikasjalise tõestuse.

Selliste mitteolemasolu teoreemide tõestused on alati matemaatikat kaunistanud. "Negatiivsete" tulemuste tõestamine on tavaliselt tunduvalt keerulisem kui ümberlükkamine, sest tõestamiseks tuleb välja mõelda üha uusi keerukamaid meetodeid.

Matemaatika ajaloost on teada ka teisi analoogilisi "negatiivseid" tulemusi. Tuntumad neist on

- ringi kvadratuuri probleem[1],
- arvu $\sqrt{2}$ irratsionaalsus.

Samamoodi on võimalik uurida ka inimvõimete piire. Võttes kasutusele inimtegevuse abstraktsed mudelid, saab tõestada, et on ülesandeid, mida ei suuda lahendada isegi kõige targem inimene. Järelikult pole ka inimene täiuslik.

9.1. Turingi masin

Arvutatavuse mõistet on võimalik mitmeti täpsustada. Me võime näiteks sooritada arvutusi kitsal paberlindil või ruudulisel paberil. Paberlindi korral peab otsustama, kas lindil on algus ja lõpp või saab seda tarviduse korral lõpmatult jätkata ning kas lindi lahtritel on aadressid (nagu majadel) või peame ise arvet pidama, kus me parasjagu asume. Selgub, et arvutatavuse mõiste sisu jääb ikka samaks, ükskõik kuidas me neile küsimustele vastame. Kui me valime mingi uue arvutatavuse formalisatsiooni, siis saab alati hoolika kontrollimise teel tõestada, et see on eelmiste formalisatsioonidega *ekvivalentne*, s.t. neile vastab täpselt sama arvutatavate funktsioonide klass.

Seda ameerika matemaatiku Alonzo Churchi (1903–1995) ning inglise matemaatiku ja loogiku Alan Mathison Turingi (1912–1954) sõnastatud väidet ei saa tõestada, vaid see formuleeritakse *Churchi teesina* (e. *Turingi teesina*):

- Mingi formalisatsiooniga arvutatavate funktsioonide klass langeb kokku funktsioonide klassiga, mis on üleüldse inimeste või arvutite poolt suvaliste efektiivsete meetoditega arvutatavad, s.t. iga intuitiivselt arvutatav funktsioon on ka meie formalisatsiooni mõttes arvutatav.

[1] Konstrueerida sirkli ja joonlaua abil antud ringiga pindvõrdne ruut. Saksa matemaatik Carl Louis Ferdinand von Lindemann (1852–1939) tõestas 1882. a., et see ülesanne on lahendamatu.

Churchi teesi kehtivust kinnitab tõsiasi, et kellelgi ei ole seda õnnestunud ümber lükata ning erinevad arvutatavuse mõiste formalisatsioonid on osutunud tõepoolest ekvivalentseteks.

Turing võttis arvutuste tegemiseks kasutusele nn. *Turingi masina*, mis sooritab arvutusi mõlemas suunas lõpmatul lahtriteks jaotatud lindil. Selle lindi kõik lahtrid, välja arvatud lõplik arv lahtreid, on arvutuste alguses tühjad, s.t. nad on täidetud spetsiaalse tühikusümboliga s_0. Igasse lindi lahtrisse on kirjutatud üks sümbol lõplikust *linditähestikust* $s_0, s_1, \ldots, s_n$ ning igal arvutussammul vaatleb inimene või mehhaaniline arvuti mingit kindlat lindi lahtrit. Ta oskab lugeda selles lahtris olevat sümbolit, sinna kirjutada ja liikuda vasakule või paremale. Masin on igal ajahetkel ühes lõplikust arvust *seisunditest* $q_1, \ldots, q_m$.

Linti võib endale ette kujutada kui lõpmatut raudteed, mis on jaotatud liipritega lahtriteks. Raudteerööbastel liigub vagun, mida liigutab selle sees asuv mees, kes loeb liiprite vahelt sümboleid, kustutab neid ja kirjutab sinna uusi sümboleid vastavalt tema käes olevatele lehtedele, kuhu on kantud instruktsioonid, mida igas seisundis teha.

Instruktsioon sõltub nii arvutuste seisundist kui ka vaadeldavast sümbolist. Arvutaja sooritab ühe $(n+4)$-st võimalikust *tegevusest*:

- peatab arvutamise;
- liigub ühe lahtri võrra paremale (R);
- liigub ühe lahtri võrra vasakule (L);
- kirjutab vaadeldavasse lahtrisse sümboli s_0;
- ...
- kirjutab vaadeldavasse lahtrisse sümboli s_n.

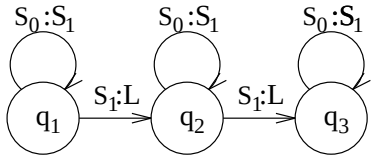
Peale nähtava tegevuse sooritab mees ka varjatud tegevuse: ta määrab järgmise instruktsiooni (seisundi).

Niisuguseid instruktsioonide kogumeid nimetataksegi *Turingi masinateks*. Turingi masinaid võib esitada kas tabelitena, graafidena või käsunelikute hulkadena.

	s_0	s_1
q_1	$s_1 q_1$	$L q_2$
q_2	$s_1 q_2$	$L q_3$
q_3	$s_1 q_3$	

$$\begin{cases} q_1 s_0 s_1 q_1 \\ q_1 s_1 L q_2 \\ q_2 s_0 s_1 q_2 \\ q_2 s_1 L q_3 \\ q_3 s_0 s_1 q_3 \end{cases}$$

Joonis 9.1. Turingi masina kaks esitust.



Joonis 9.2. Turingi masina graaf.

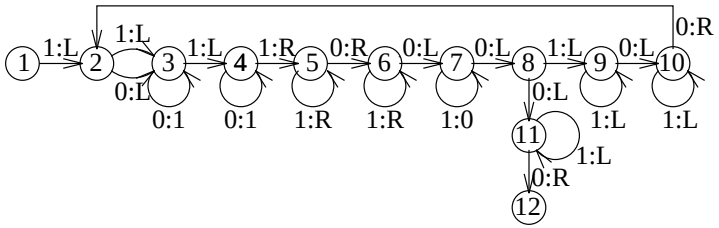
Näide 9.1. Kirjeldame Turingi masinat, mis kirjutab tühjale lindile kolm ühte: $s_1 s_1 s_1$.

See masin on esitatud joonisel 9.1 tabelina ja käsunelikute hulgana ning joonisel 9.2 graafina. Käsk $q_i s_j s_k q_l$ ütleb järgmist:

- Kui Turingi masinat interpreteeriv seade on seisundis q_i ning vaatleb sümbolit s_j, siis ta kirjutab lahtrisse sümboli s_k ja läheb üle seisundisse q_l,

Toodud Turingi masin kirjutab vaadeldavasse ruutu sümboli s_1, liigub vasakule, kirjutab s_1, liigub veelkord vasakule, kirjutab s_1 ja peatub, sest puudub täidetav instruktsioon. Me eeldame, et masin alustab tööd vähima indeksiga seisundis, milleks on antud juhul seisund q_1.

Turingi masina tööd on mõistlik kirjeldada kui arvutusprotsessi konfiguratsioonide jada. *Arvutuse konfiguratsioonis* näidatakse lindi hetkeseis, seadme seisund ja vaadeldav lahter ajahetkel. Olgu Turingi masinat interpreteeriv seade arvutusprotsessi alguses seisundis q_1 ja olgu kõigisse lindi lahtritesse kirjutatud sümbol s_0.



Joonis 9.3. Kahekordistav Turingi masin.

Samastades seisundid ja sümbolid nende indeksitega ($q_1 = 1$, $q_2 = 2$, $q_3 = 3$, $s_0 = 0$, $s_1 = 1$), saame näites 9.1 kirjeldatud Turingi masina korral järgmise konfiguratsioonide jada:

$$\begin{array}{cccccc} 0000000 & 0000100 & 0000100 & 0001100 & 0001100 & 0011100\,, \\ \overset{\triangle}{1} & \overset{\triangle}{1} & \overset{\triangle}{2} & \overset{\triangle}{2} & \overset{\triangle}{3} & \overset{\triangle}{3} \end{array}$$

kus üleval on näidatud lindi seis ja vaadeldava lahtri alla on kirjutatud Turingi masinat interpreteeriva seadme seisundi number selles konfiguratsioonis.

Näide 9.2. Joonisel 9.3 on Turingi masin, mis kahekordistab ühtede arvu lindil.

Näide 9.3. Eelmises kahes näites toodud Turingi masinate ühendamise teel saame konstrueerida masina, mis kirjutab kõigepealt tühjale lindile n ühte ja kahekordistab siis ühtede arvu. Järelikult kirjutab see Turingi masin tühjale lindile $2n$ ühte.

Edaspidi huvitavad meid Turingi masinad, millega saab arvutada täisarvulisi funktsioone. Võtame linditähestikuks hulga $\{0, 1\}$ ning kujutame arve ühendsüsteemis.

Definitsioon 9.4. Turingi masin on *standardkonfiguratsioonis n*, kui lindil on n järjestikust ühte ja ülejäänud lint on tühi (s.t. koosneb nullidest), kusjuures lugemis-kirjutamispea vaatleb vasakpoolseimat ühte.

Tähistame sümboliga **P** positiivsete täisarvude hulga $\{1, 2, \ldots\}$.

Definitsioon 9.5. Me ütleme, et Turingi masin *arvutab funktsiooni* $f\colon \mathbf{P} \to \mathbf{P}$, kui ta, alustades tööd vähima numbriga seisundis standardkonfiguratsioonis n ($n \in \mathbf{P}$), lõpetab töö standardkonfiguratsioonis $f(n)$. Kui Turingi masin ei lõpeta tööd või peatub mittestandardses konfiguratsioonis, siis $f(n) = \perp$ (mittemääratud).

Näitest 9.2 selgub, et funktsiooni $f(n) = 2n$ saab Turingi masinatega arvutada.

Ülesanded

1. Konstrueerige Turingi masin, mis modelleerib teie igapäevaseid tegemisi.
2. Leidke funktsiooni $f(n) = 2n$ arvutamiseks Turingi masin, mille seisundite arv on väiksem kui 12 (näites 9.2).

9.2. Mittelahenduvad probleemid

Definitsioon 9.6. Olgu T Turingi masin ja f tema arvutatav funktsioon. Turingi masina T *produktiivsus* on tema poolt tühjale lindile kirjutatud arv, s.t.

$$\begin{cases} f(0), & \text{kui } f(0) \text{ on määratud;} \\ 0, & \text{muidu.} \end{cases}$$

Defineerime funktsiooni

$$p(n) \stackrel{\text{def}}{=} \text{kõige produktiivsema } n\text{-seisundise Turingi masina produktiivsus,}$$

millega saab sõnastada *virga kopra probleemi*.

- Konstrueerida Turingi masin tähestikus $\{0, 1\}$, mis arvutab funktsiooni $p(n)$.

Näitame, et virga kopra probleemi ei saa Turingi masinatega lahendada, s.t. ei leidu Turingi masinat, mis arvutaks produktiivsusfunktsiooni $p(n)$. Tõestame alustuseks paar abitulemust.

Lemma 9.7. $p(1) = 1$.

Tõestus. Lemma tõestatakse kõigi variantide läbivaatamisega, võttes aluseks Turingi masinate esituse graafide kujul. Üheseisundise Turingi masina graafis võib olla maksimaalselt kaks noolt.

0 noolega Turingi masina graafe on ainult üks. Kuna see masin peatub kohe algseisundis, siis on tema produktiivsus 0.

Mitmesuguseid 2 noolega Turingi masina graafe on kokku 16. Need Turingi masinad ei peatu kunagi, sest nad saavad alati täita mingit tegevust.

1 noolega Turingi masina graafe on kokku 8. Need Turingi masinad, mille noole sümboliks on 1, peatuvad kohe. Masinad, mille noolel on märgend 0:R, 0:L või 0:0, ei peatu, sest nad töötavad lõpmatus tsüklis. Viimase 1 noolega Turingi masina, mille noolel on märgend 0:1, produktiivsus on 1. Järelikult $p(1) = 1$. □

Lemma 9.8. $p(47) \geq 100$.

Tõestus. Ühendame kolm Turingi masinat (samastades nende alg- ja lõpptipud):

1. Masin, mis kirjutab tühjale lindile 25 ühte (selleks piisab 25 seisundist).
2. Masin, mis kahekordistab ühtede arvu (milleks võtame 12-seisundise Turingi masina näitest 9.2).
3. Masin, mis kahekordistab ühtede arvu (12 seisundit).

Tulemuseks on 47-seisundine Turingi masin, mis kirjutab tühjale lindile 100 ühte. □

Lemma 9.9. $p(n+1) > p(n)$.

Tõestus. Ülesanne 9.2.1. □

Lemma 9.10. $p(n+11) \geq 2n$.

Tõestus. Ühendame omavahel kaks Turingi masinat:

1. Masin, mis kirjutab tühjale lindile n ühte (n seisundit).
2. Masin, mis kahekordistab ühtede arvu (12 seisundit).

Saadud $(n+11)$-seisundise Turingi masina produktiivsus on $2n$. □

Teoreem 9.11. *Virga kopra probleem on mittelahenduv.*

Tõestus. Tõestame teoreemi vastuväiteliselt. Oletame, et leidub Turingi masin *VK*, mis arvutab funktsiooni $p(n)$. Olgu sellel masinal k seisundit. Teoreemi tõestamiseks ühendame omavahel kolm Turingi masinat:

1) masin, mis kirjutab tühjale lindile n ühte (n seisundit);

2) *VK* (k seisundit);

3) *VK* (k seisundit).

Tulemuseks on $(n+2k)$-seisundine Turingi masin produktiivsusega $p(p(n))$. Seega kehtib iga arvu n korral

$$p(n+2k) \geq p(p(n)).$$

Lemma 9.9 põhjal on $p(n)$ kasvav funktsioon, mistõttu

$$n+2k \geq p(n).$$

Tehes asenduse {n/n+11}, saame

$$n+11+2k \geq p(n+11),$$

mis annab lemma 9.10 põhjal võrratuse

$$n+11+2k \geq 2n.$$

Asendus {n/12+2k} annab nüüd vastuolu, sest võrratus

$$11+2k \geq 12+2k$$

on samaväärne võrratusega

$$0 \geq 1.$$

Järelikult ei ole Turingi masinat *VK* olemas, mistõttu funktsioon $p(n)$ ei ole Turingi mõttes arvutatav. Churchi teesi põhjal võime väita, et produktiivsusfunktsioon $p(n)$ on mitte ainult Turingi mõttes mittearvutatav, vaid seda ei saa arvutada mitte ühegi mehhaanilise arvutusvahendiga. □

Ülesanded

1. Tõestage lemma 9.9.
2. **Tõestage, et $p(2) = 4$.

Peatumisprobleem on algoritmiliselt mittelahenduva ülesande teine klassikaline näide. Selle tõestamiseks kasutatakse tavaliselt nn. *diagonaliseerimismeetodit*. Laiendame kõigepealt Turingi masinatel arvutatavate funktsioonide mõiste mitme muutuja funktsioonidele.

Definitsioon 9.12. Me ütleme, et Turingi masin *M arvutab funktsiooni* $f_M: \mathbf{P}^m \to \mathbf{P}$, kui ta, alustades tööd vähima numbriga seisundis standardkonfiguratsioonis $n_1 0 n_2 0 \ldots 0 n_m$ ($n_i \in \mathbf{P}$), lõpetab töö standardkonfiguratsioonis $f_M(n_1, n_2, \ldots, n_m)$. Kui M ei lõpeta tööd või peatub mittestandardses konfiguratsioonis, siis $f_M(n_1, \ldots, n_m) = \perp$ (mittemääratud).

Definitsioon 9.13. Funktsioon $f(x_1, \ldots, x_m)$ on *Turingi mõttes arvutatav*, kui leidub seda arvutav Turingi masin.

Näiteks joonisel 9.3 toodud Turingi masin arvutab funktsiooni $f(n) = 2n$, mistõttu funktsioon $f(n) = 2n$ on Turingi mõttes arvutatav.

Kõik Turingi masinad saab esitada lõpmatu loeteluna[2]

$$M_1, M_2, M_3, \ldots,$$

[2]Sellist Turingi masinate efektiivset loetelu nimetatakse Turingi masinate *Gödeli numeratsiooniks*.

mis annab meile omakorda kõigi Turingi mõttes arvutatavate 1 muutuja funktsioonide loetelu

$$f_1, f_2, f_3, \ldots,$$

kus Turingi masin M_n arvutab funktsiooni f_n.

Sõnastame nüüd Turingi masinate *peatumisprobleemi*:

- Leida efektiivne protseduur Turingi masinate peatumise kontrollimiseks.

Ülesanded

3. *Pakkuge välja Turingi masinate efektiivse loetlemise moodus.
4. **Näidake, et virga kopra probleemi mittelahenduvusest järeldub Turingi masinate peatumisprobleemi mittelahenduvus.
5. Peatumisprobleemi korral on võimalik vaadelda kahte erijuhtu:
   - *Endal peatumise probleem* $M_n(n)\downarrow$? (Kas Turingi masin M_n peatub sisendil n?)
   - *Üldine peatumisprobleem*: $M_m(n)\downarrow$?

   Näidake, et endal peatumise probleemi mittelahenduvusest järeldub üldise peatumisprobleemi mittelahenduvus.

Kuigi peatumisprobleemi mittelahenduvus järeldub virga kopra probleemi mittelahenduvusest, pakub peatumisprobleemi tõestus ka iseseisvat huvi.

Teoreem 9.14 (peatumisprobleemi mittelahenduvus). *Ei leidu Turingi masinat, mis lahendaks Turingi masinate peatumisprobleemi.*

Tõestus. Oletame, et leidub Turingi masin, mis arvutab üldise peatumisprobleemi *karakteristliku funktsiooni*[3]

[3]Karakteristlikes funktsioonides tähistab tavaliselt 1 tõesust ja 0 väärust. Kuna me kasutasime Turingi masinates naturaalarvude hulga $\mathbb{N}$ asemel positiivsete täisarvude hulka $\mathbf{P} = \{1, 2, \ldots\}$, siis asendasime siin väärtused 1 ja 0 arvudega 2 ja 1.

$$h(x,y) = \begin{cases} 2, & \text{kui } M_x(y)\downarrow; \\ 1, & \text{kui } M_x(y)\uparrow. \end{cases}$$

Seda funktsiooni arvutava Turingi masina põhjal saab konstrueerida uue Turingi masina, mis arvutab funktsiooni

$$g(x) = \begin{cases} \perp, & \text{kui } h(x,x) = 2; \\ 1, & \text{kui } h(x,x) = 1 \end{cases}$$

ehk noolte keeles

$$g(x) = \begin{cases} \uparrow, & \text{kui } M_x(x)\downarrow; \\ \downarrow, & \text{kui } M_x(x)\uparrow. \end{cases}$$

Funktsioon $g(x)$ puudub *konstruktsiooni tõttu* Turingi mõttes arvutatavate 1 muutuja funktsioonide loetelust

$$f_1, f_2, f_3, \ldots,$$

sest oletus, et mingi arvu m korral $g = f_m$, annab vastuolu

$$f_m(m)\downarrow \Leftrightarrow f_m(m)\uparrow.$$

Järelikult ei ole peatumisprobleemi karakteristlik funktsioon $h(x,y)$ Turingi mõttes arvutatav. □

Diagonaliseerimise[4] ideed selgitab järgmine tabel, kus m-nda rea n-ndas veerus on funktsiooni f_m väärtus kohal n.

$$\begin{array}{cccccc} f_1(1) & f_1(2) & f_1(3) & \ldots & f_1(m) & \ldots \\ f_2(1) & f_2(2) & f_2(3) & \ldots & f_2(m) & \ldots \\ f_3(1) & f_3(2) & f_3(3) & \ldots & f_3(m) & \ldots \\ \ldots & \ldots & \ldots & \ddots & \ldots & \ldots \\ f_m(1) & f_m(2) & f_m(3) & \ldots & f_m(m) & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ddots \end{array}$$

[4]Saksa matemaatik Georg Cantor (1845–1918) tõestas diagonaliseerimismeetodiga, et lõigus $[0,1]$ on reaalarve rohkem kui naturaalarve hulgas $\mathbb{N}$. Ta tõestas samuti, et iga hulga korral on tema kõigi osahulkade hulgas rohkem elemente kui hulgas endas. Kõigi hulkade hulga korral saame siit Cantori paradoksi.

Funktsioon g erineb igast arvutatavast funktsioonist f_m *diagonaalelemendil* m:

$$g(m) \neq f_m(m).$$

Me tõestasime, et mitte ükski Turingi masin ei suuda lahendada peatumisprobleemi kogu Turingi masinate lõpmatu loetelu jaoks korraga. Churchi teesiga saab selle tulemuse üle kanda kõigile efektiivsetele algoritmidele, s.t. mitte ükski algoritm ei suuda lahendada peatumisprobleemi kõigi Turingi masinate ja kõigi nende sisendandmete jaoks korraga. Küll võib peatumisprobleem lahenduda iga konkreetse Turingi masina korral. Näiteks joonisel 9.3 toodud kahekordistav Turingi masin peatub igal sisendi $n \in \mathbf{P}$ korral.

Turingi masinate efektiivsetel loeteludel on peale peatumisprobleemi mittelahenduvuse veel teisigi huvitavaid omadusi. Üheks neist on loetelude universaalsete funktsioonide arvutatavus.

Definitsioon 9.15. Funktsioonide loetelu $f_1, f_2, \ldots$ *universaalne funktsioon* on funktsioon

$$f(m,n) = f_m(n) \qquad (m,n \in \mathbf{P}).$$

Teoreem 9.16 (loetlemisteoreem). *Kõigi Turingi mõttes arvutatavate* 1 *muutuja funktsioonide loetelu* $f_1, f_2, \ldots$ *universaalne funktsioon on arvutatav.*

Ka seda teoreemi on võimalik tõestada nii formaalselt, konstrueerides antud Turingi masinate loetelu jaoks konkreetse universaalse Turingi masina, kui ka mitteformaalselt, selgitades mismoodi selle masina töö peaks põhimõtteliselt toimuma ning viidates seejärel Churchi teesile.

Ülesanded

6. Andke loetlemisteoreemi mitteformaalne tõestus.

7. **Näidake, et universaalne funktsioon $f(m,n)$ ei ole laiendatav kõikjal määratud arvutatavaks funktsiooniks.

9.3. Predikaatarvutuse mittelahenduvus

Alonzo Church tõestas, et predikaatarvutus ei ole lahenduv, s.t. pole võimalik kontrollida, kas suvaliselt valitud predikaatarvutuse valem on loogiliselt tõene või mitte.

Teoreem 9.17 (Church). *Esimest järku loogika on mittelahenduv.*

Tõestus. Teoreemi tõestamiseks taandame üldise peatumisprobleemi esimest järku loogika *tõesuse probleemile*, s.t. näitame, et kui esimest järku loogikas on tõesuse probleem lahenduv, siis on ka peatumisprobleem lahenduv.

Olgu meil Turingi masin T, mille sisendiks on arv n. Konstrueerime Turingi masina T kirjelduse põhjal predikaatarvutuse valemite hulga Δ ja valemi h, nii et

$$\Delta \models h \Leftrightarrow T(n)\downarrow.$$

Me interpreteerime neid valemeid loomulikul viisil, nii et nad kirjeldavad Turingi masina T tööd sisendil n:

- Valem h väidab, et Turingi masin T peatub sisendil n: $T(n)\downarrow$.
- Valemite hulk Δ ütleb, et masina sisendiks on arv n ja masin töötab vastavalt oma kirjeldusele.

Eeldame, et masin T alustab tööd ajahetkel 0, vaadeldes parasjagu 0-ndat lahtrit. Eeldame, et Turingi masina lint on vasakule ja paremale lõpmatu ning linditähestik võib peale numbrite 0 ja 1 sisaldada ka teisi sümboleid. Lepime veel kokku, et enne tööle asumist ja pärast peatumist ei ole masina seisund, vaadeldav lindi positsioon ning loetav sümbol määratud.

Toome predikaatarvutusse uued predikaatsümbolid

$$Q_i \quad (1 \le i \le r),$$
$$S_j \quad (0 \le j \le m),$$

mis vastavad Turingi masina T seisundite hulgale $Q = \{q_1, q_2, \ldots, q_r\}$ ja linditähestikule $\Sigma = \{s_0, s_1, \ldots, s_m\}$. Nii saame esimest järku loogika signatuuris $\langle 0; '; \{Q_i\}, \{S_j\}, =, < \rangle$, kus $\{Q_i\}$ ja $\{S_j\}$ tähistavad vastavaid predikaatsümbolite loetelusid indeksite i ja j järgi.

Selle signatuuri loomulikuks interpretatsiooniks on struktuur

$$I = \langle \mathbb{Z}; 0; +1; \{\overline{Q}_i\}, \{\overline{S}_j\}, =, < \rangle,$$

mis koosneb täisarvude hulgast $\mathbb{Z}$ tema loomulikus järjestuses, täisarvust null ja ühe liitmise tehtest ning predikaatsümbolite Q_i ja S_j interpretatsioonidest, mis vastavad masina T kirjeldusele:

- $I \models Q_i(t, x) \Leftrightarrow$ masin T on ajahetkel t seisundis q_i ning vaatleb x-ndat lahtrit;
- $I \models S_j(t, x) \Leftrightarrow$ ajahetkel t on x-ndas lahtris sümbol s_j.

Kirjeldame nüüd valemite hulka Δ. Iga kirjutamiskäsu $q_i s_j s_k q_l$ kohta leidub temas valem

$$\begin{aligned}\forall t \forall x \forall y \{&[Q_i(t, x) \,\&\, S_j(t, x)] \supset \\ &[Q_l(t', x) \,\&\, S_k(t', x) \,\&\, (y \neq x \supset \\ &(S_0(t, y) \supset S_0(t', y)) \,\&\, \ldots \,\& \\ &(S_m(t, y) \supset S_m(t', y)))]\},\end{aligned}$$

s.t. käsu tulemusel muutub ainult vaadeldava lahtri sisu ning kõik ülejäänud lahtrid jäävad samaks.

Iga paremale liikumise käsu $q_i s_j R q_l$ kohta on hulgas Δ valem

$$\begin{aligned}\forall t \forall x \forall y \{&[Q_i(t, x) \,\&\, S_j(t, x)] \supset \\ &[Q_l(t', x') \,\&\, (S_0(t, y) \supset S_0(t', y)) \,\&\, \ldots \,\& \\ &(S_m(t, y) \supset S_m(t', y))]\}\end{aligned}$$

ning iga vasakule liikumise käsu $q_i s_j L q_l$ kohta valem

$$\forall t \forall x \forall y \{[Q_i(t, x') \,\&\, S_j(t, x')] \supset$$
$$[Q_l(t', x) \,\&\, (S_0(t, y) \supset S_0(t', y)) \,\&\, \ldots \,\&$$
$$(S_m(t, y) \supset S_m(t', y))]\}.$$

Hulgas Δ on ka valem, mis ütleb, et ajahetkel 0 on masin algseisus:

$$Q_1(0, 0) \,\&\, S_1(0, 0) \,\&\, S_1(0, 0') \,\&\, \ldots \,\&\, S_1(0, 0^{(n-1)})\&$$
$$\forall y\,[(y \neq 0 \,\&\, y \neq 0' \,\&\, \ldots \,\&\, y \neq 0^{(n-1)}) \supset S_0(0, y)],$$

s.t. algseisus on lindil n ühte ja ülejäänud lahtrites on nullid. Kui $n = 0$, siis on viimane valem kujul

$$Q_1(0, 0) \,\&\, \forall y\, S_0(0, y).$$

Hulgas Δ on veel seoste $'$ ja $<$ definitsioonid:

$$\forall z \exists x\,(z = x') \,\&\, \forall z \forall x \forall y\,(z = x' \,\&\, z = y' \supset x = y),$$
$$\forall x \forall y \forall z\,(x < y \,\&\, y < z \supset x < z)\&$$
$$\forall x \forall y\,(x' = y \supset x < y) \,\&\, \forall x \forall y\,(x < y \supset x \neq y).$$

On selge, et interpretatsioonis I on kõik Δ valemid tõesed, s.t. interpretatsioon I on valemite hulga Δ mudeliks.

Valem h väidab, et Turingi masin T peatub sisendil n. Kuna Turingi masin peatub ainult juhul, kui mingi seisundi ja sümboli paari korral on masina tegevus määramata, siis võib valemiks h võtta disjunktsiooni üle kõigi selliste seisundi-sümboli paaride:

$$\bigvee_{\text{puudub tegevus } q_i s_j \ldots} \exists t \exists x\,(Q_i(t, x) \,\&\, S_j(t, x)).$$

Mittepeatuvate Turingi masinate korral (mille tegevus kirjeldatud kõigi seisundi-sümboli paaride korral) võib valemiks h võtta loogiliselt väära valemi

$$\neg\, 0 = 0.$$

Näitame, et

$$\Delta \models h \Leftrightarrow T(n)\downarrow.$$

Oletame, et $\Delta \models h$. Kuna interpretatsioon I on valemite hulga Δ mudeliks, siis kehtib $I \models \Delta$ ja seega $I \models h$. Valemi h tõesus on aga samaväärne Turingi masina T peatumisega sisendil n.

Oletame nüüd, et $T(n)\downarrow$ ja lõpetamise momendil s on masin seisundis q_i, vaadeldes p-ndas lahtris sümbolit s_j. Kirjeldame arvutuse konfiguratsiooni ajahetkel s valemiga

$$m_s = Q_i(0^{(s)}, 0^{(p)}) \,\&\, S_{j_1}(0^{(s)}, 0^{(p_1)}) \,\&\, \ldots \,\&$$
$$S_j(0^{(s)}, 0^{(p)}) \,\&\, \ldots \,\&\, S_{j_v}(0^{(s)}, 0^{(p_v)})\&$$
$$\forall y\,[y \neq 0^{(p_1)} \,\&\, \ldots \,\&\, y \neq 0^{(p)} \,\&\, \ldots \,\&$$
$$y \neq 0^{(p_v)} \supset S_0(0^{(s)}, y)],$$

kus $p_1, \ldots, p, \ldots, p_v$ on lindi järjestikuste lahtrite numbrite jada ja $s_{j_1}, \ldots, s_j, \ldots, s_{j_v}$ nendes lahtrites asuvate sümbolite jada. Näiteks algseisu käsitlev valem hulgast Δ kirjeldab arvutuste alghetke konfiguratsiooni m_0. Kuna arvutuste käigus saab liikuda ka 0-ndast lahtrist vasakule, siis tuleb kokku leppida, kuidas me viitame negatiivse numbriga lahtritele. Lahtrile numbriga $p = -q$ $(q > 0)$ võib näiteks viidata valemitega kujul

$$Q_i(x, 0^{(p)}) = \exists z\,(Q_i(x, z) \,\&\, z^{(q)} = 0).$$

Matemaatilise induktsiooniga saab tõestada (ülesanne 9.3.1), et

$$\Delta \models m_s,$$

kus m_s on suvaline arvutuse konfiguratsioon ajahetkel $s \geq 0$.

See valem kehtib igal arvutussammul. Järelikult kehtib ta ka juhul, kui g on konfiguratsioon, milles Turingi masin T peatub:

$$\Delta \models g.$$

Kuna Turingi masin T on lõpetamise momendil s seisundis q_i ja vaatleb p-ndas lahtris sümbolit s_j, siis

$$g \models Q_i(0^{(s)}, 0^{(p)}) \,\&\, S_j(0^{(s)}, 0^{(p)}).$$

Seega

$$g \models \exists t \exists x\,(Q_i(t, x) \,\&\, S_j(t, x)),$$

millest järeldub $g \models h$ ja $\Delta \models h$. □

Ülesanded

1. **Tõestage, et

$$\Delta \models m_s,$$

kus m_s on suvaline arvutuse konfiguratsioon ajahetkel s ($s \geq 0$).

2. **Konstrueerige valemite hulk Δ ja valem h Turingi masina

$$\{q_1 1 R q_1, q_1 0 L q_2, q_2 1 0 q_2\}$$

jaoks, kui $n = 0$ ja $n = 2$. Näidake, et $\Delta \models h$, kui $n = 0$.

10. Aritmeetika mittetäielikkus

Inimsoo saatuslik kalduvus loobuda mõtlemast asjade üle, mille suhtes ei valitse enam kahtlusi, on põhjustanud poole tema vigadest.

— John Stuart Mill (1806–1873)

Inimene ei saa aru ei olematusest, millest ta esile kerkib, ega lõpmatusest, mis teda alla neelab.

— Blaise Pascal (1623–1662)

Kuna geomeetria ja algebra on intuitiivselt kõige paremini mõistetavad matemaatikaharud, siis on neid püütud korduvalt formaliseerida. Tuntud on geomeetria aksiomaatiline ülesehitus Eukleidese (u. 365–300 e.m.a.) poolt ja naturaalarvude kirjeldus Dedekindi (1831–1916) poolt, mis sai tuntuks *Peano aksiomaatikana*. Aritmeetika all mõistetakse üldiselt abstraktset struktuuri, mis koosneb kõigi naturaalarvude hulgast $\mathbb{N}$ ja selle elementaartehetest: liitmisest, korrutamisest jms.

10.1. Formaalne aritmeetika

Olgu aritmeetika keele L_N signatuur $\Sigma_N = \langle 0; ', +, \cdot; = \rangle$. Kirjeldame *formaalse aritmeetika* kui esimest järku teooria *N Peano aksioomidega*:

(P1) $\forall x \forall y\,(x' = y' \supset x = y)$.

(P2) $\forall x\,(x' \neq 0)$.

(P3) $\forall x\,(x \neq 0 \supset \exists y\,(x = y'))$.

(P4) $\forall x\,(x + 0 = x)$.

(P5) $\forall x \forall y\,(x + y' = (x + y)')$.

(P6) $\forall x\,(x \cdot 0 = 0)$.

(P7) $\forall x \forall y\,(x \cdot y' = (x \cdot y) + x)$.

(P8) $P(0) \,\&\, \forall x(P(x) \supset P(x')) \supset \forall x\, P(x)$ – *induktsiooniaksioom* (P on suvaline predikaat)

Aksioom (P1) ütleb, et järgmise elemendi leidmise tehe $'$ on ühene e. injektiivne. Aksioomid (P2) ja (P3) väidavad, et leidub ainult üks esimene element, nimelt 0. Aksioomid (P4)–(P7) defineerivad induktiivselt liitmis- ja korrutamistehte. Induktsiooniaksioom (P8) formaliseerib aga matemaatilise induktsiooni printsiibi: alustades elemendist 0 on tehtega $'$ võimalik jõuda iga naturaalarvuni.

Formaalsel aritmeetikal on *standardinterpretatsioon*

$$\mathcal{N} = \langle \mathbb{N}, 0, +1, +, \cdot \rangle,$$

mis koosneb naturaalarvude hulgast $\mathbb{N}$ ja selle harilikest tehetest. Kõiki teisi teooria N mudeleid, mis ei ole mudeliga $\mathcal{N}$ isomorfsed, nimetatakse aritmeetika *mittestandardseteks mudeliteks*.

Selgub, et formaalne aritmeetika on mittetäielik ja mittelahenduv. Põhjuseks on ühelt poolt esimest järku predikaatarvutuse piiratus ja teiselt poolt formaalse aritmeetika suur väljendusvõimsus. Norra matemaatik Thoralf Skolem (1887–1963) tõestas 1934. a., et aritmeetikal leidub loenduv mittestandardne mudel, mis sisaldab lisaks tavalistele arvudele ka lõpmatult suuri arve. Gödel formaliseeris aga 1931. a. aritmeetikas *valetaja paradoksi*, näidates aritmeetika sisemist vastuolulisust.

Tähistame formaalsed naturaalarvud *sümbolarvudega*.

$$\overline{0} = 0, \quad \overline{1} = 0', \quad \ldots \quad , \quad \overline{m} = 0\overbrace{''\ldots'}^{m}.$$

Formaalses aritmeetikas saab tõestada paljud naturaalarvude hulga omadused. Tõestame näiteks sümbolarvude hulga lõpmatuse.

Lause 10.1. *Teooria N kõik mudelid on lõpmatud.*

Tõestus. Näitame, et $m \neq n \Rightarrow \ \vdash_N \overline{m} \neq \overline{n}$ $(m, n \in \mathbb{N})$. Olgu $m \neq n$. Siis kehtib kas $m < n$ või $n < m$. Olgu $m < n$.

1.	$\overline{m} = \overline{n}$	hüpotees
2.	$0\overbrace{''\ldots'}^{m} = 0\overbrace{''\ldots'}^{n}$	sümbolarvude definitsioon
3.	$0 = 0\overbrace{''\ldots'}^{n-m}$	2 m korda (P1), UI ja MP
4.	$t' = 0$	3 $t = \overline{n - m - 1}$, $=$ sümmeetria
5.	$t' \neq 0$	aksioom (P2)
6.	$t' = 0 \,\&\, t' \neq 0$	4,5 Conj
7.	$\overline{m} \neq \overline{n}$	1–6 IP

Analoogilise tõestuse saab konstrueerida ka $n < m$ korral. Järelikult on erinevatele arvudele vastavad objektid erinevad. Arve on aga loenduv hulk. □

Teoorias N saab kirjeldada järjestusseosed:

$$x < y \stackrel{\text{def}}{=} \exists w\,(w \neq 0 \,\&\, x + w = y),$$
$$x \leq y \stackrel{\text{def}}{=} x < y \vee x = y.$$

Definitsioon 10.2. Naturaalarvuline *predikaat* $R(x_1, \ldots, x_n)$ on *aritmeetikas väljendatav* (e. *aritmeetiline predikaat*), kui leidub valem $\mathcal{A}(x_1, \ldots, x_n) \in L_N$, nii et iga $k_1, \ldots, k_n \in \mathbb{N}$ korral

$$R(k_1, \ldots, k_n) = t \Leftrightarrow \mathcal{N} \models \mathcal{A}(\overline{k_1}, \ldots, \overline{k_n}).$$

Näiteks seost $x_1 = x_2$ väljendab valem $x_1 = x_2$, sest võrdust interpreteeritakse alati objektide samasusena:

$$k_1 = k_2 \Rightarrow \mathcal{N} \models \overline{k_1} = \overline{k_2},$$
$$k_1 \neq k_2 \Rightarrow \mathcal{N} \models \overline{k_1} \neq \overline{k_2}.$$

Seost $x_1 < x_2$ väljendab aga valem $\exists w\,(w \neq 0 \,\&\, x_1 + w = x_2)$, kuna

$$k_1 < k_2 \Rightarrow \mathcal{N} \models \exists w\,(w \neq 0 \,\&\, \overline{k_1} + w = \overline{k_2}),$$
$$k_1 \not< k_2 \Rightarrow \mathcal{N} \models \neg\exists w\,(w \neq 0 \,\&\, \overline{k_1} + w = \overline{k_2}).$$

Defineerime kvantori *leidub täpselt üks* valemiga

$$\exists_1 x\; p(x) \stackrel{\text{def}}{=} \exists x\; p(x) \,\&\, \forall x \forall y\,(p(x) \,\&\, p(y) \supset x = y).$$

Sarnaselt aritmeetikas väljendatavate predikaatidega saab defineerida ka aritmeetikas väljendatavad funktsioonid.

Definitsioon 10.3. Naturaalarvuline *funktsioon* $f(x_1, \ldots, x_n)$ on *aritmeetikas väljendatav*, kui leidub valem $\mathcal{A}(x_1, \ldots, x_n, x_{n+1}) \in L_N$, nii et iga $k_1, \ldots, k_{n+1} \in \mathbb{N}$ korral

1) kui $f(k_1, \ldots, k_n) = k_{n+1}$, siis $\mathcal{N} \models \mathcal{A}(\overline{k_1}, \ldots, \overline{k_n}, \overline{k_{n+1}})$;
2) $\mathcal{N} \models \exists_1 x_{n+1}\, \mathcal{A}(\overline{k_1}, \ldots, \overline{k_n}, x_{n+1})$.

Näiteks *nullfunktsioon*

$$z(x) = 0$$

on aritmeetikas väljendatav valemiga

$$\mathcal{A}(x) \stackrel{\text{def}}{=} x_1 = x_1 \,\&\, x_2 = 0.$$

Olgu $z(k_1) = k_2$, s.t. $k_2 = 0$. Siis saame võrduse omaduste põhjal

$$\mathcal{N} \models \overline{k_1} = \overline{k_1} \,\&\, 0 = 0,$$
$$\mathcal{N} \models \exists_1 x_2 (x_1 = x_1 \,\&\, x_2 = 0),$$

sest võrdustest $x_2 = 0$ ja $y = 0$ järeldub $x_2 = y$. Ka *successor-funktsioon*

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

on aritmeetikas väljendatav. Selleks sobib näiteks valem

$$\mathcal{A}(x) \stackrel{\text{def}}{=} x_2 = x_1'.$$

Kurt Gödel (1906–1978) näitas, et kõik arvutatavad naturaalarvulised funktsioonid ja predikaadid on aritmeetikas väljendatavad. Ta tõi selleks sisse β-funktsiooni, mis võimaldab aritmeetikas väljendada väiteid naturaalarvude lõplike jadade kohta.

Teoreem 10.4 (Gödel). *Iga kõikjal määratud arvutatav funktsioon on aritmeetikas väljendatav.*

Järeldus 10.5. *Iga arvutatav predikaat on aritmeetikas väljendatav.*

Ülesanded

1. Tõestage, et aritmeetikas on väljendatav *successor*-funktsioon.
2. Näidake, et funktsioon $f(x_1, \ldots, x_n)$ on aritmeetikas väljendatav $\Leftrightarrow$ tema graafik $f(x_1, \ldots, x_n) = y$ on aritmeetikas väljendatav.
3. Põhjendage, miks on aritmeetikas väljendatav mitte kuskil määratud funktsioon $\perp(x)$.

10.2. Gödeli teoreemid aritmeetika mittetäielikkusest

Matemaatika ajaloost on teada mitmeid näiteid "intuitiivselt selgetest" teooriatest, milles hiljem avastatakse ebatäpsusi või vastuolusid. Eukleidese geomeetriat peeti kuni 19. sajandini ainuvõimalikuks geomeetriaks. Arvati, et selle aksioomide kehtivus on väljaspool kahtlust. Siis aga seati *paralleelide aksioom* kahtluse alla

ning saksa matemaatik Georg Friedrich Bernhard Riemann (1826–1866), vene matemaatik Nikolai Lobatševski (1793–1856) ja ungari matemaatik János Bolyai (1802–1860) konstrueerisid pärast selle kõrvalejätmist mitu *mitteeukleidilist geomeetriat*.

Samasugune olukord tekkis eelmisel sajandivahetusel ka Cantori hulgateoorias. Selles intuitiivselt lihtsas ja selges teoorias avastati mitu vastuolu, millest kuulsamad on Russelli ja Cantori paradoksid.

Mitme sajandi vältel arvati, et Newtoni mehhaanika kirjeldab ideaalselt kehade liikumist ja vastastikuseid mõjusid. Kuid Einsteini relatiivsusteooria näitas, et kehade massi ja kiirust saab käsitleda ka Newtonist erinevalt: keha aeglasel liikumisel on selle mass konstantne, kuid valguse kiirusele lähedastel kiirustel keha mass väheneb. Seda kinnitavad ka füüsikute poolt mikromaailmas tehtud katsed.

Aritmeetika täielikkuse probleem kasvas välja *Hilberti programmist*. Saksa matemaatik David Hilbert (1862–1943) püüdis selle sajandi alguses matemaatilistes tõestustes "korda luua". Ta tahtis esitada kogu olemasoleva matemaatika ühtse formaalse aksiomaatilise süsteemina, mille *finiitseid tõestusi* tunnustaksid kõik matemaatikud. See süsteem pidi loomulikult sisaldama ka tavalist naturaalarvulist aritmeetikat. Hilbert pidas tarvilikuks tõestada rangelt nii selle süsteemi mittevasturääkivus kui ka täielikkus.

Definitsioon 10.6. Me ütleme, et teooria N on *mittetäielik*, kui leidub lause $p \in L_N$, nii et

$$\nvdash_N p \quad \text{ja} \quad \nvdash_N \neg p.$$

Saksa loogik Kurt Gödel (1906–1978) tõestas 1931. aastal, et Hilberti programm on sisemiselt vastuoluline, mistõttu seda ei saagi põhimõtteliselt ellu viia. Ta näitas, et kui aritmeetikat sisaldav formaalne teooria ei ole vasturääkiv, siis ta ei ole täielik. Seega saab selle teooria tulemusi erinevalt interpreteerida ning ei ole olemas ühtset kogu olemasolevat matemaatikat hõlmavat formaalset teooriat.

Oletame, et vaadeldaval teoorial leidub mudel. Suvalise kinnise valemi p korral on selles mudelis tõene kas valem p ise või tema eitus $\neg p$. Kuna see teooria käsitleb kogu olemasolevat matemaatikat, siis tema mittetäielikkus tähendab, et leidub niisugune matemaatiline väide, mida ei saa rangelt tõestada ega ka ümber lükata.

Me võime võtta konstrueeritava teooria aksioomideks aritmeetika kõigi tõeste valemite hulga

$$\mathcal{T} = \{\, p \in L_N \mid \mathcal{N} \models p \,\}.$$

Saadud teooriat nimetatakse standardinterpretatsiooni $\mathcal{N}$ *elementaarteooriaks* $\varepsilon(\mathcal{N})$. See teooria on küll täielik, kuid tema aksioomide hulk ei ole enam lahenduv. Aksioomide hulga lahenduvus ja tuletusreeglite hulga lõplikkus on aga praktikas kasutatavate teooriate iseenesestmõistetavad omadused.

Teoreem 10.7 (Gödeli esimene teoreem aritmeetika mittetäielikkusest). *Iga lahenduv mittevasturääkiv aksiomaatiline teooria N, mis sisaldab Peano aksioome, on mittetäielik, s.t.*

$$\mathcal{P} = \{\, p \in L_N \mid \vdash_N p \,\} \subsetneq \mathcal{T}.$$

Tõestus. Olgu $\{M_i\}$ Turingi masinate Gödeli numeratsioon. Defineerime predikaadi

$$T(e, x, y) \stackrel{\text{def}}{=} \text{Turingi masin } M_e \text{ peatub sisendil } x \text{ täpselt } y \text{ sammu jooksul.}$$

Predikaat $T(e, x, y)$ on Churchi teesi põhjal arvutatav. Järelikult on ta aritmeetikas väljendatav. Kuna teooria N on Peano aksiomaatika üldistus, siis on $T(e, x, y)$ väljendatav ka teoorias N ja leidub seda väljendav valem $\mathcal{A}(e, x, y) \in L_N$. Nüüd saab teoorias N valemiga

$$\mathcal{B}(e, x) \stackrel{\text{def}}{=} \exists y\, \mathcal{A}(e, x, y)$$

väljendada predikaadi $M_e(x)\!\downarrow$ (Turingi masin M_e peatub sisendil x). Oletame, et teooria N on täielik. Siis kas

$$\vdash_N \mathcal{B}(\overline{m}, \overline{n}) \quad \text{või} \quad \vdash_N \neg\mathcal{B}(\overline{m}, \overline{n}).$$

Kuna lahenduva teooria teoreemide hulk on *rekursiivselt loetletav*, siis lahendab loetlemisalgoritm ka üldise peatumisprobleemi:

$$\vdash_N \mathcal{B}(\overline{m},\overline{n}) \Rightarrow M_m(n)\downarrow,$$
$$\vdash_N \neg\mathcal{B}(\overline{m},\overline{n}) \Rightarrow M_m(n)\uparrow \quad \text{(ei peatu)}.$$

Üldine peatumisprobleem on aga mittelahenduv, mistõttu teooria N ei saa olla täielik. □

Gödeli teine teoreem aritmeetika mittetäielikkusest väidab, et kui aritmeetika on mittevasturääkiv, siis ei saa temas tuletada valemit, mis väljendab tema enda mittevasturääkivust. Seega peab aritmeetika mittevasturääkivuse tõestamiseks võtma kasutusele temast üldisema teooria. Näiteks võib proovida tõestada formaalse aritmeetika mittevasturääkivust formaalses hulgateoorias. Kuid siis oleks vaja tõestada ka formaalse hulgateooria mittevasturääkivus, milleks tuleb konstrueerida veel üldisem formaalne teooria.

Ülesanded

1. Näidake, et formaalse aritmeetika kõigi teoreemide hulk on rekursiivselt (s.t. algoritmiliselt) loetletav.

11. Kõrgemat järku loogika

11.1. Aritmeetika kui teist järku teooria

Esimest järku loogikas tohib kvantifitseerida ainult indiviidmuutujaid. Matemaatiliste teooriate esitamisel kvantifitseeritakse sageli ka funktsioone ja predikaate. Võtame näiteks pidevusaksioomi[1]

> *Igal ülalt tõkestatud reaalarvude hulgal on olemas ülemine raja.*

Reaalarvude hulgad kujutavad endast reaalarvudel defineeritud predikaate. Järelikult väidab pidevusaksioom, et igal reaalarvulisel predikaadil leidub minimaalne ülemine tõke.

Ka induktsiooniaksioomis

$$P0 \,\&\, \forall x\,(Px \supset Px') \supset \forall x\, Px$$

on P suvaline predikaat ning tema sidumine üldisuskvantoriga $\forall P$ annab meile teist järku predikaatloogika aksioomi. Kui me käsitleme formaalset aritmeetikat kui esimest järku teooriat, siis annab induktsiooniaksioom iga predikaadi Px korral uue aksioomi, mistõttu aksioomide hulk on lõpmatu. Naturaalarvude hulga korral on võimalik konstrueerida kontinuaalne arv erinevaid predikaate Px.

[1] G. Kangro. *Matemaatiline analüüs I.* Tallinn, 1982, lk. 17.

Esimest järku formaalsel aritmeetikal on lisaks naturaalarvude hulgale ka mittestandardsed mudelid. Formaalne aritmeetika on aga teist järku teooriana lõplikult aksiomatiseeritav ning tema ainsaks mudeliks on aritmeetika standardinterpretatsioon.

11.2. Teist järku loogika

Lubades esimest järku loogika keeles kasutada lisaks indiviidmuutujatele ka lause-, predikaat- ja funktsionaalmuutujaid, mida võib siduda üldisus- ja eksistentsikvantoritega, on lihtne üle minna esimest järku loogikast *teist järku loogikasse*. Teist järku loogika semantika ei erine oluliselt esimest järku loogika omast: n-kohalist predikaatmuutujat püütakse asendada kõikvõimalike n-kohaliste seostega. mis on defineeritavad vaadeldava interpretatsiooni universumil, ning vabu muutujaid käsitletakse selliselt, nagu nad oleksid seotud üldisuskvantoriga.

Kõrgemat järku loogikas on paljusid mõisteid tunduvalt lihtsam defineerida kui esimest järku loogikas. Näiteks saab *Leibnizi võrduse printsiibi*, mis väidab, et kui kahel objektil on ühed ja samad omadused, siis võib need objektid samastada, esitada definitsioonina

$$x = y \stackrel{\text{def}}{=} \forall P\,(Px \equiv Py).$$

Definitsiooni aluseks on tõdemus, et erinevaid objekte on võimalik alati semantiliselt eristada, sest me saame defineerida abstraktse omaduse, mis on ühel objektil täidetud, teisel aga ei ole. Tegelikult on Leibnizi võrduse printsiip esitatav ka esimest järku loogikas. Selleks võtame kasutusele binaarse predikaadi app, mis rakendab oma esimest argumenti teisele argumendile, ja käsitleme predikaatsümboleid kui indiviidmuutujaid sobivas esimest järku teoorias:

$$\forall P\,(app(P, x) \equiv app(P, y)).$$

Ent kõrgemat järku loogika võimaldab esitada sellised väited palju lihtsamal ja loomulikumal kujul.

11.3. Kompaktsusteoreem

Esimest järku predikaatarvutuse täielikkuse teoreemist järeldub vahetult mitu olulist tulemust, nagu Skolemi-Löwenheimi teoreem ja kompaktsusteoreem. Me nimetame teooriaks suvalist valemite hulka. Teooria mudel on interpretatsioon, milles on tõesed kõik selle teooria valemid. Interpretatsioon on lõplik, kui tema universum on lõplik.

Teoreem 11.1 (Skolemi-Löwenheimi teoreem, 1915). *Kui esimest järku teoorial leidub lõpmatu mudel, siis leidub tal ka lõplik või loenduv mudel.*

Seda teoreemi nimetatakse vahel ka Skolemi-Löwenheimi laskumisteoreemiks vastandina Skolemi-Löwenheimi tõusmisteoreemile.

Teoreem 11.2 (Skolemi-Löwenheimi tõusmisteoreem). *Kui esimest järku teoorial leidub lõpmatu mudel, siis leidub tal mudel ka iga võimsuse korral, mis on selle mudeli võimsusest suurem.*

Nimetatud teoreemidest järeldub, et esimest järku loogikas ei saa lõpmatuid võimsusi eristada. Kuna esimest järku predikaatarvutuses langevad tuletatavuse ja loogilise järelduvuse mõisted kokku, siis kehtib veel järgmine teoreem.

Teoreem 11.3 (kompaktsusteoreem). *Kui* $\Gamma \models p$*, siis leidub lõplik hulk* $\Delta \subseteq \Gamma$*, nii et* $\Delta \models p$*.*

Vahel nimetatakse kompaktsusteoreemiks tema järeldust.

Järeldus 11.4. *Kui teooria* Γ *igal lõplikul alamteoorial leidub mudel, siis leidub mudel ka teoorial* Γ*.*

Ülesanded

1. Tõestage kompaktsusteoreem ja selle järeldus.

11.4. Väljendatavus teist järku loogikas

Järgnevalt näitame kompaktsusteoreemiga esimest järku loogika piiratust.

Teoreem 11.5 (Henkin, 1949). *Kui teoorial Γ leidub kuitahes suur lõplik mudel, siis leidub tal ka lõpmatu mudel, s.t. esimest järku loogikas ei saa lõplikkust ja lõpmatust eristada.*

Tõestus. Olgu Γ esimest järku teooria, millel leidub kuitahes suur lõplik mudel. Teooria Γ' saamiseks lisame teooriale Γ kõikvõimalikud valemid $\sigma_{\geq n}$ $(n = 1, 2, \ldots)$, mis väidavad, et leidub vähemalt n erinevat elementi[2]. Kuna teooria Γ' igal lõplikul alamteoorial leidub mudel, on kompaktsusteoreemi põhjal mudel ka tal endal. Valemite $\sigma_{\geq n}$ tõesuse tõttu on see mudel lõpmatu. Kuna $\Gamma \subseteq \Gamma'$, siis on saadud mudel ka esialgse teooria Γ mudel. □

Teist järku loogikas saab eristada lõplikke ja lõpmatuid hulki. Selleks sobib näiteks valem

$$\sigma_{<\aleph_0} \stackrel{\text{def}}{=} \forall f\,(\mathrm{Inj}(f) \supset \mathrm{Surj}(f)),$$

s.t. iga injektiivne funktsioon on samal ajal ka sürjektiivne, kus

$$\mathrm{Inj}(f) \stackrel{\text{def}}{=} \forall x \forall y\,(f(x) = f(y) \supset x = y),$$
$$\mathrm{Surj}(f) \stackrel{\text{def}}{=} \forall x \exists y\,(f(y) = x).$$

Ülesanded

1. Põhjendage, miks valem $\sigma_{<\aleph_0}$ on kõigis lõplikes interpretatsioonides tõene ja kõigis lõpmatutes interpretatsioonides väär.

Järgnevalt näitame, et esimest järku aritmeetikal leiduvad loenduvad *mittestandardsed mudelid*.

[2] Näiteks $\sigma_{\geq n} \stackrel{\text{def}}{=} \exists x_1 \ldots \exists x_n (x_1 \neq x_2 \,\&\, x_1 \neq x_3 \,\&\, \ldots \,\&\, x_{n-1} \neq x_n)$.

Teoreem 11.6 ***(Skolemi teoreem aritmeetika mittestandardsetest mudelitest, 1934).*** *Olgu Γ esimest järku teooria, mille signatuuris on sümbolid 0 ja $'$ (successor-funktsioon) ning mille mudeliks on aritmeetika standardinterpretatsioon $\mathcal{N}$. Siis leidub teoorial Γ loenduv mudel $\mathcal{M}$, mis pole mudeliga $\mathcal{N}$ isomorfne.*

Teooriaks Γ võib võtta näiteks kõik formaalse aritmeetika (teooria N) teoreemid või kõik aritmeetika standardinterpretatsioonis $\mathcal{N}$ tõesed valemid (s.t. mudeli $\mathcal{N}$ *elementaarteooria*)

$$\varepsilon(\mathcal{N}) = \{\, p \mid \mathcal{N} \models p \,\}.$$

Tõestus. Olgu Γ loetletud omadustega teooria. Teooria Γ' saamiseks lisame Γ signatuuri uue konstandi c ja aksioomid $c \neq \overline{n}$ $(n = 0, 1, 2, \ldots)$, kus $\overline{n}$ tähistab n-ndat sümbolarvu. Teooria Γ' igal lõplikul alamteoorial on olemas mudel, milleks võib võtta standardinterpretatsiooni $\mathcal{N}$, käsitledes konstanti c esimese arvuna, mille kohta vaadeldavas lõplikus alamteoorias puudub aksioom $c \neq \overline{n}$. Kompaktsusteoreemi ja Skolemi-Löwenheimi teoreemi põhjal leidub teoorial Γ' loenduv mudel $\mathcal{M}$. Et konstantsümbolit c ei saa interpreteerida ühegi naturaalarvuna, siis pole mudelid $\mathcal{M}$ ja $\mathcal{N}$ omavahel isomorfsed. □

Ülesanded

2. Tõestage, et formaalse aritmeetika mudelid $\mathcal{N}$ ja $\mathcal{M}$ ei ole omavahel isomorfsed.

Erinevalt esimest järku loogikast on teist järku formaalse aritmeetika mudel üheselt määratud. Selleks võib kasutada näiteks valemit

$$\varphi_N \stackrel{\text{def}}{=} \varphi_{suc} \,\&\, \forall x\, Nx,$$

kus

$$\varphi_{suc} \stackrel{\text{def}}{=} \forall x\,(x' \neq 0) \,\&\, \forall x \forall y\,(x' = y' \supset x = y),$$
$$Nx \stackrel{\text{def}}{=} \forall P\,(P0 \,\&\, \forall u\,(Pu \supset Pu') \supset Px),$$

s.t. sümbolarvud $\overline{n}$ ($n = 0, 1, 2, \ldots$) tähistavad erinevaid arve ning puuduvad sümbolitega tähistamata arvud.

Toome etteruttavalt veel ühe esimest ja teist järku predikaatarvutuse erinevuse: kui esimest järku intuitsionistlikus loogikas ei saa kvantoreid ja lausearvutustehteid üksteise kaudu avaldada, siis teist järku intuitsionistlikus loogikas saab defineerida need üldisuskvantori ja implikatsiooni kaudu:

$$\begin{aligned}
\bot\,(= v) &= \forall X\, X,\\
\neg p &= p \supset \bot,\\
p \,\&\, q &= \forall X\, ((p \supset (q \supset X)) \supset X),\\
p \vee q &= \forall X\, ((p \supset X) \supset ((q \supset X) \supset X)),\\
\exists x\, p &= \forall X \forall x\, ((p \supset X) \supset X),\\
\exists P\, p &= \forall X \forall P\, ((p \supset X) \supset X),
\end{aligned}$$

kus suured tähed tähistavad lause- ja predikaatmuutujaid ning $\bot$ vastuolu.

11.5. Tüübiteooria

Kui me lubame kasutada predikaate ka predikaatide ja funktsioonide argumentidena, saame tulemuseks *tüüpidega predikaatarvutuse*. Defineerime näiteks predikaadi

$$Pred(P) \stackrel{\text{def}}{=} P(P),$$

mis ütleb, et predikaat P on iseendale rakendatav.

Ülesanded

1. *Kirjeldage predikaadid P ja Q, nii et $Pred(P)$ on tõene ja $Pred(Q)$ on väär.

Küsime nüüd, kas predikaati $\neg Pred$ saab iseendale rakendada või mitte? Kui $\neg Pred(\neg Pred)$ on tõene, on predikaadi $Pred$ definitsiooni tõttu tõene ka lause $Pred(\neg Pred)$. Saime vastuolu.

Oletame, et $\neg Pred(\neg Pred)$ on väär. Siis on definitsiooni tõttu väär ka lause $Pred(\neg Pred)$, mis annab jälle vastuolu.

See Russelli avastatud paradoks näitab, et predikaatsümbolite kasutamisel predikaatide argumentavaldistes ilma ühegi kitsendusteta võivad tekkida vastuolud. Paradokside vältimiseks toodi sisse predikaatide, funktsioonide ja indiviidide tüübid:

- Boole'i konstandid t ja v on *tüüpi* o.
- Indiviidmuutujad ja indiviidkonstandid on tüüpi ι.
- Kui σ ja τ on tüübid, siis $\sigma \to \tau$ on *funktsioonitüüp*, mis seab σ-tüüpi objektidele vastavusse τ-tüüpi objektid.

Näiteks tüüp $\iota \to \iota$ tähistab indiviididel määratud funktsioone ning tüüp $\iota \to o$ indiviididel määratud ühekohalisi predikaate.

Niisugust tüübiteooriat kasutatakse näiteks funktsionaalse programmeerimise süsteemides, kus igal predikaadil, funktsioonil ja indiviidil on fikseeritud tüüp. Tüüpide põhjal defineeritakse ka objektide tasemed:

- Indiviidmuutujad, indiviidkonstandid ja Boole'i konstandid on *tasemega* 0.
- Funktsiooni tase on ühe võrra suurem tema argumentide maksimaalsest tasemest.

Näiteks esimest ja teist järku predikaatarvutus on tasemega 1. Predikaatidele ja funktsioonidele tüüpide ja tasemete omistamise kaudu saame rääkida ka kolmandat, neljandat jne. järku loogikast (s.t. *lõplikku järku loogikatest*). On selge, et tüübiteooria kõrvaldab loogikast Russelli paradoksi sarnased vastuolud. Predikaati ei ole sellisel juhul võimalik iseendale rakendada, sest siis peaks selle predikaadi aste olema samaaegselt iseendast ühe võrra suurem. Loogikat, kus iseendale viitamine ei ole definitsioonides lubatud, nimetatakse *predikatiivseks*. Huvi pakub ka *mittepredikatiivne* loogika. Eriti seoses rakendustega andmebaaside teoorias ning rekursiivsete programmide ja induktiivsete struktuuride analüüsimisel.

Tüübiteooria kaudu on loomulik sisse tuua ka indiviidide *liigid* e. sordid. Selleks võtame kasutusele mitu erisugust universumit, tähistades nende elemendid erinevate 0-taseme tüüpidega ι_1, ι_2, ... Tulemuseks on *mitmeliigiline* e. mitmesordiline predikaatarvutus. Näiteks programmeerimiskeeltes võivad protseduuride ja funktsioonide argumendid olla kas mõnda süsteemselt määratud — täisarvud, reaalarvud, sümbolid — või kasutaja defineeritud tüüpi — hulgad, kirjed vms.

11.6. Matemaatiline teooria

Esimest järku loogikal on rida omadusi, mis teist järku loogikal puuduvad:

1. Esimest järku loogika tõeste valemite hulka saab rekursiivselt loetleda. Ent teist järku loogika tõeste valemite hulka ei saa rekursiivselt loetleda, sest see hulk ei ole mitte üheski kõrgemat järku aritmeetikas kirjeldatav.

2. Esimest järku loogika on *kompaktne*, aga teist järku loogika ei ole kompaktne. Selle põhjuseks on aritmeetika standardinterpretatsiooni ühene kirjeldatavus teist järku loogikas.

3. Esimest järku loogikas kehtivad *Skolemi-Löwenheimi laskumisteoreem* ja *tõusmisteoreem*. Teist järku loogikas nad aga ei kehti, sest seal saab eristada loenduvaid ja kontiinumi võimsusega hulki.

Sellepärast tekib õigustatud küsimus:

- Kas teist järku loogikat võib üldse pidada loogikaks?

Võib-olla on teist järku loogika vaid matemaatiline abstraktsioon. Ühelt poolt on tal olemas loogikale iseloomulikud tunnused: süntaks, semantika ja tuletusteooria. Filosoofilises mõttes võib aga loogikat käsitleda kui teadust *a priori* tõdedest, mis ei sõltu praktikast ega meie subjektiivsusest. Seepärast peab Quine teist järku loogikat pigem matemaatiliseks teooriaks.

Samas on selge, et teist järku loogikal on matemaatilistes arutlustes tähtis koht. Põhjalikumalt on uuritud kõrgemat järku hulgateooriat ja analüüsi. Sajandi alguses käivitunud *Hilberti programmi* sisuks oli mittepredikatiivsete tõestusmeetodite põhjendamine predikatiivsete vahenditega. Gödeli teoreemid formaalse aritmeetika mittetäielikkusest näitasid aga, et Hilberti ettevõtmine oli liiga optimistlik. Sellegipoolest õnnestus muuta mitu näiliselt mittepredikatiivset meetodit oluliselt konstruktiivsemaks.

11.7. Rakendused arvutiteaduses ja programmeerimises

Universaalsete programmeerimiskeelte aluseks on protseduurid, mis on kõrgemat järku funktsioonid. Juba Algol ja Ada kasutasid täiel määral kõrgemat järku loogikate tüübistruktuuri. Eriti arenenud on objektorienteeritud keelte tüübisüsteem. Tüübid on üheaegselt nii andmete kirjeldused, mida saab kasutada programmide vigade avastamiseks, kui ka iseseisvad andmed.

Konstruktiivse loogika ja funktsionaalse programmeerimise vahel avastatud seosed on *programmide automaatse sünteesi* aluseks, kus programmid genereeritakse nende formaalsetest tõestustest kõrgemat järku loogikas.

Kuna juba esimest järku loogika on algoritmiliselt mittelahenduv, on mittelahenduv ka teist järku loogika. Sellepärast uuritakse arvutiteaduses intensiivselt tema alamloogikaid nagu *monaadilist teist järku loogikat*, milles võib kasutada kõrgemat järku muutujatena ainult ühekohalisi predikaate (s.t. hulki). Ilma funktsionaalsümboliteta monaadiline teist järku loogika lahendub nagu ka monaadiline esimest järku loogika.

11.8. Ülesanded

1. (G. Forbes) Formaliseerige laused, kasutades teist järku kvantoreid:

   (a) Platonit ja Sokratest ei ole võimalik eristada.

   (b) Igal kahel inimesel on midagi ühist.

   (c) Mõnel inimesel ei ole teiste inimestega midagi ühist.

   (d) Kui miski on omane igale kahele inimesele, siis on igal kahel inimesel mingi ühine omadus.

2. **Formaliseerige väited. Püüdke hinnata kasutatava loogika järku. (Allikas: G. Kangro. *Matemaatiline analüüs I*. Tallinn, 1982)

   (a) Kui igale reaalarvule x hulgast X on üheselt vastavusse seatud reaalarv y, siis öeldakse, et hulgal X on defineeritud *funktsioon*.

   (b) Arv A on funktsiooni f piirväärtus kohal a parajasti siis, kui funktsiooni f määramispiirkonnas iga a-le läheneva jada $\{x_n\}$ puhul, kus $x_n \neq a$, kehtib seos $\lim\limits_{n\to\infty} f(x_n) = A$.

   (c) Kui üksteisesse sisestatud lõikude jadas lõikude pikkused lähenevad nullile, siis neil lõikudel on parajasti üks ühine punkt, mis kujutab nende lõikude otspunktide jada piirväärtust.

   (d) Igast tõkestatud jadast saab eraldada koonduva osajada.

   (e) Kui funktsioonid f ja g on pidevad kohal a, siis ka funktsioonid $f(x) + g(x)$, $f(x) - g(x)$, $f(x)\,g(x)$, $f(x)/g(x)$ on pidevad kohal a, kusjuures jagatise korral eeldame, et $g(a) \neq 0$.

   (f) Lõigus pidev funktsioon on selles lõigus tõkestatud.

   (g) Lõigus pideval funktsioonil on selles lõigus ekstremaalsed väärtused.

   (h) Joon $y = f(x)$ on kumer (nõgus) piirkonnas X parajasti siis, kui $f'(x)$ kahaneb (kasvab) monotoonselt piirkonnas X.

   (i) Rida nimetame *koonduvaks*, kui selle rea osasummade jada koondub, ja *hajuvaks*, kui osasummade jada hajub.

III

Mitteklassikaline loogika

12. Modaalloogika

Loogikat ei saa rakendada tegelikule maailmale.
— Marvin Lee Minsky (s. 1927)

Modaalloogikat tunti juba Aristotelese ajal, tänapäeva modaalloogika rajas ameerika loogik ja filosoof Clarence Irving Lewis (1883–1964). Ta kirjeldas alates 1912. a. modaalloogika jaoks mitu aksiomaatilist süsteemi. Modaalloogika on tihedalt seotud mitme tehisintellektis, arvutiteaduses ja filosoofias kasutatava loogikaga nagu *temporaalne*, *deontiline* ja *episteemiline* loogika. Temporaalloogikaga saab näiteks tõestada programmide korrektsust, teadmisi ja uskumisi käsitleva episteemilise loogikaga saab aga formaalselt kirjeldada erisuguseid inimmõtlemise tüüpe.

12.1. Modaalsed operaatorid

Modaalloogika on klassikalise loogika üldistus: temas kehtivad kõik klassikalise loogika seadused, lisaks tuuakse veel *võimalikkuse* ja *paratamatuse* modaalsed operaatorid $\Diamond$ ja $\Box$, mida kasutatakse sarnaselt eitusega. Erinevalt eitusest ei ole modaalsed operaatorid tõeväärtusfunktsioonilised, s.t. väidete $\Diamond p$ ja $\Box p$ tõeväärtused ei ole väite p tõeväärtusega täielikult määratud.

Vaatleme lauset

On võimalik, et ma olen Eestimaa kuningas.

See lause on *episteemilisel* tõlgendamisel väär, sest ma *tean*, et ma ei ole kuningas. Ent seda saab tõlgendada ka lausena

On võimalik ette kujutada olukorda, kus ma olen Eestimaa kuningas,

mis on tõene, sest asjad oleksid võinud areneda niimoodi, et Eesti on kuningriik ning mina olen selle riigi kuningas.

Modaalloogikas nimetatakse võimalikke olukordi, milliseks asjad oleksid võinud areneda, *võimalikeks maailmadeks*. See terminoloogia on pärit saksa filosoofilt ja matemaatikult Gottfried Wilhelm Leibnizilt (1646–1716).

Mida tähendab mingi olukorra võimalikkus? Võimalik on loomulikult kõik see, mis kehtib tegelikult. Kuid mis on võimatu? Selle asemel, et rääkida mingi olukorra eetilisest, juriidilisest, tehnilisest vms. võimatusest, kasutame neist üldisemat *loogilise mittevõimalikkuse* mõistet. Kõik, mis on loogiliselt mittevõimalik, on mittevõimalik ka teistes tähendustes.

Mingi olukorra loogilist mittevõimalikkust saab põhjendada ainult loogikareeglitega, kusjuures *loogikat* tuleb käsitleda võimalikult üldiselt. Näiteks loogiliselt ei ole võimalik ette kujutada ümmargust nelinurka ega ka naissoost venda. Tavaliselt öeldakse, et loogiliselt mittevõimalik on see, millest järeldub vastuolu. Niisiis on iga väide, mille väärus on tõestatud, loogiliselt *mittevõimalik*, ning iga tõestatud väide on *paratamatult tõene*, sest ta ei saa kuidagi olla väär[1]. Paratamatult tõene on näiteks lause

Kolmnurgal on kolm nurka.

Kui lause pole paratamatult väär, on ta loogiliselt *võimalik*. Võimalik on näiteks lause

[1]Sellise mittevõimalikkuse definitsiooniga tuleb olla ettevaatlik, sest *Gödeli teoreem aritmeetika mittetäielikkusest* kinnitab väärade lausete olemasolu, mille väärus ei tarvitse olla formaalselt tõestatav.

Lennart Meri oli Eesti president 1996. aastal.

See lause pole aga paratamatult tõene, sest Riigikogu oleks võinud valida presidendiks ka alternatiivse kandidaadi.

Filosoofilises analüüsis kasutatakse modaalseid operaatoreid küllalt sageli. Näiteks *vaba tahte probleemi* korral kerkib küsimus:

- Kas inimene oleks võinud konkreetses situatsioonis käituda teisiti?

Kui inimene poleks saanud teisiti käituda, on meie tunne tegutsemise vabadusest vaid illusoorne.

12.2. Modaalloogika seadused

Niisiis me laiendasime klassikalist loogikat võimalikkuse ja paratamatuse seostega. Selgus, et lause on mittevõimalik siis ja ainult siis, kui ta on paratamatult väär:

$$\neg\Diamond p = \Box\neg p.$$

Järelikult saab modaalsed tehted avaldada teineteise kaudu:

$$\Diamond p = \neg\Box\neg p,$$
$$\Box p = \neg\Diamond\neg p.$$

On selge, et iga tõene lause on võimalik, ning et iga paratamatu lause peab olema tõene:

$$p \models \Diamond p,$$
$$\Box p \models p.$$

Neid seadusi ei saa aga ümber pöörata, sest siis langeksid võimalikkuse ja paratamatuse mõisted kokku.

Oletame, et mingi väide järeldub loogiliselt mingist paratamatult tõesest väitest. Siis on ka järeldus paratamatult tõene. Selles veendumiseks eeldame, et q järeldub loogiliselt väitest p, väide p on paratamatult tõene, ent q ei ole paratamatult tõene. Kuna q

ei ole paratamatult tõene, on ta mingis olukorras väär. Loogilise järeldumise tõttu on ka tema eeldus p selles olukorras väär. See on aga vastuolus p paratamatu tõesusega. Pannes tähele, et q *loogilist järeldumist* väitest p võib väljendada lausega $\Box(p \supset q)$, saab tõestatud seaduse esitada kujul

$$\Box(p \supset q) \models \Box p \supset \Box q.$$

Ülesanded

1. *Näidake, et selle seaduse pöördseadus ei kehti.

Kuna kõik, mis on loogiliselt tõene, on ka paratamatult tõene, siis kehtib *paratamatuse sissetoomise* reegel:

$$\text{Kui } \models p, \text{ siis } \models \Box p.$$

Toodud seadused ja reeglid on aluseks tuletussüsteemile T, mis on modaalloogika süsteemidest üks nõrgemaid.

Põhjus, miks Lewis tõi loogikasse *paratamatuse* mõiste, oli tema rahulolematus tavalise *implikatsiooniga*. Näiteks lause

> *Kui kolmnurgal on neli nurka, siis värvid sina endal huuli*

on implikatsiooni eelduse vääruse tõttu tõene, kuid tema osalausete vahel puudub tajutav loogiline seos. Lewis defineeris *range implikatsiooni*, mis on tõene ainult siis, kui ei ole võimalik, et tema eeldus on tõene ning järeldus on väär:

$$\Box(p \supset q) = \neg\Diamond(p \,\&\, \neg q).$$

Kahjuks tekivad range implikatsiooni korral samasugused paradoksid nagu tavalise implikatsiooni korral[2]:

[2]Tautoloogiaid kujul

$$p \supset (q \supset p),$$
$$\neg p \supset (p \supset q),$$
$$(p \supset q) \vee (q \supset p)$$

nimetatakse *implikatsiooni paradoksideks* (MacColl, 1906). Nende valemite paradoksaalsus seisneb selles, et nad on tõesed suvalise valemi q korral.

$$\Box p \supset \Box(q \supset p),$$
$$\Box\neg p \supset \Box(p \supset q),$$

s.t. range implikatsiooniga saab suvalisest lausest järeldada paratamatu lause ning mittevõimalikust lausest järeldub rangelt ükskõik milline lause.

12.3. Võimalike maailmade semantika

Modaalsete operaatorite tähenduse kirjeldamiseks võttis USA loogik ja filosoof Saul Aaron Kripke (s. 1940) 1963. a. kasutusele *võimalike maailmade semantika*, kus kõikvõimalikke ettetulevaid olukordi käsitletakse kui terviklikke maailmade süsteeme. Iga maailm on vaadeldava signatuuri sümbolite (lausemuutujate, predikaatsümbolite jne.) üks võimalik interpretatsioon. Kuna modaalloogikas võib kasutada ka tavalisi modaliseerimata lauseid, siis kuulutatakse üks maailmadest *tegelikuks maailmaks* ning lause, mis on selles maailmas tõesed, arvatakse tõesteks ka kogu maailmade süsteemis.

Me ütleme, et $\Box p$ on tõene, kui p kehtib kõigis võimalikes maailmades, ja $\Diamond p$ on tõene, kui p kehtib mõnes võimalikus maailmas. Idee on selles, et *võimalik maailm* on täielik alternatiivne viis, kuidas asjad oleksid võinud olla.

Definitsioon 12.1. Modaalse lauseloogika valemite hulga *interpretatsioon* on võimalike maailmade hulk $\mathcal{W}$, mille iga maailm w omistab igale selles hulgas esinevale lausemuutujale mingi tõeväärtuse ja mille üks maailm on tähistatud kui tegelik maailm w^*.

Sellist modaalloogilist süsteemi nimetatakse ka *standardseks modaalloogikaks* ehk süsteemiks $S5$. Me käsitleme Lewise teisi modaalloogilisi süsteeme nagu T ja $S0$–$S4$ põgusalt natuke hiljem.

Tähistades väljenditega $w[p]$ ja $\mathcal{W}[p]$ valemi p tõeväärtused maailmas w ja intepretatsioonis $\mathcal{W}$ ning väljendiga $w\colon p \mapsto t$ tõeväärtuse, mille maailm w omistab valemile p, saame anda modaalse lauseloogika semantikareeglid:

1. $w[p] = t \Leftrightarrow w\colon p \mapsto t$ (p on lausemuutuja).
2. $w[\Box p] = t \Leftrightarrow (\forall u \in \mathcal{W})\ u[p] = t$.
3. $w[\Diamond p] = t \Leftrightarrow (\exists u \in \mathcal{W})\ u[p] = t$.
4. $\mathcal{W}[p] = t \Leftrightarrow w^*_{\mathcal{W}}[p] = t$,

kus $w \in \mathcal{W}$ on suvaline võimalik maailm, $w^*_{\mathcal{W}}$ on interpretatsiooni $\mathcal{W}$ tegelik maailm ning lausearvutustehete semantika on sama, mis lauseloogikas.

Me ütleme nagu tavaliselt, et valem on *loogiliselt tõene*, kui ta on tõene kõigis interpretatsioonides, ning arutlus *kehtib*, kui ei leidu interpretatsiooni, milles tema eeldused on tõesed ja järeldus väär.

Võimalike maailmade süsteeme saab kasutada *eksijärelduste* ümberlükkamiseks.

Näide 12.2. $\Diamond A \,\&\, \Diamond B \not\models_{S5} \Diamond(A \,\&\, B)$.

Valime interpretatsiooni $\mathcal{W} = \{w^*, u\}$ selliselt, et A ja B on vasturääkivad laused, mis ei saa olla samaaegselt tõesed:

•	•
w^*	u
$A \mapsto v$	$A \mapsto t$
$B \mapsto t$	$B \mapsto v$

Selles interpretatsioonis on eeldus $\Diamond A \,\&\, \Diamond B$ tõene, sest

1. $u[A] = t$ (definitsiooni 1. punkti põhjal).
2. $w^*[\Diamond A] = t$ (3), sest $u[A] = t$.
3. $w^*[B] = t$ (1).
4. $w^*[\Diamond B] = t$ (3), sest $w^*[B] = t$.
5. $w^*[\Diamond A \,\&\, \Diamond B] = t$ (konjunktsiooni semantika põhjal).
6. $\mathcal{W}[\Diamond A \,\&\, \Diamond B] = t$ (4), sest $w^*[\Diamond A \,\&\, \Diamond B] = t$.

Järeldus $\Diamond(A \,\&\, B)$ on aga väär:

1. $w^*[A] = v$ (definitsiooni 1. punkti põhjal).
2. $w^*[A \,\&\, B] = v$ (konjunktsiooni semantika põhjal).
3. $u[B] = v$ (1).
4. $u[A \,\&\, B] = v$ (konjunktsiooni semantika põhjal).
5. $w^*[\Diamond(A \,\&\, B)] = v$ (3), sest $w^*[A \,\&\, B] = v$ ja $u[A \,\&\, B] = v$.
6. $\mathcal{W}[\Diamond(A \,\&\, B)] = v$ (4), sest $w^*[\Diamond(A \,\&\, B)] = v$.

Ülesanded

1. Tõestage väited:
   (a) $\Diamond\neg A \not\models_{S5} \neg\Diamond A$;
   (b) $\Box(A \vee B) \not\models_{S5} \Box A \vee \Box B$;
   (c) $\Box\Diamond(A \supset B) \not\models_{S5} \Box(\Diamond A \supset B)$.
2. *Näidake, et $\models_{S5} \Box(p \vee \neg p)$, kuid $\not\models_{S5} \Box p \vee \Box\neg p$.

12.4. Tuletused modaalloogikas $S5$

Kirjeldame modaalloogika tuletusreeglid. Selleks lisame lausearvutuse loomuliku tuletuse süsteemi ND modaalsete operaatorite sissetoomis- ja eemaldamisreeglid.

Paratamatuse sissetoomisreeglis ($\Box I$) ja võimalikkuse eemaldamisreeglis ($\Diamond E$) peab olema täidetud tingimus ($*$), mis nõuab, et valemeid p, $\Diamond p$ ja q mõjutavad eeldused ja hüpoteesid (välja arvatud hüpotees p võimalikkuseoperaatori eemaldamisel) peavad olema *täielikult modaliseeritud*, s.t. nende kõik lausemuutujad peavad olema seotud kas võimalikkuse- või paratamatuseoperaatoriga.

Toome nüüd mõned tuletuste näited.

Sissetoomisreegel	Eemaldamisreegel
$\dfrac{p}{\Box p}\ (\Box I)^*$	$\dfrac{\Box p}{p}\ (\Box E)$
$\dfrac{p}{\Diamond p}\ (\Diamond I)$	$\dfrac{\Diamond p \quad \begin{array}{c}[p]\\ \vdots\\ q\end{array}}{q}\ (\Diamond E)^*$

Tabel 12.1. Loomuliku tuletussüsteemi modaalsed reeglid.

Näide 12.3. Tõestame, et $\Box(A \,\&\, B) \vdash_{S5} \Diamond B$:

1	1.	$\Box(A \,\&\, B)$	eeldus
1	2.	$A \,\&\, B$	1 $\Box E$
1	3.	B	2 $\&E$
1	4.	$\Diamond B$	3 $\Diamond I$

Tõestamiseks kasutame loomuliku tuletuse süsteemi, milles iga valemi ees on tema eelduste ja hüpoteeside numbrite loend. Antud juhul sõltub valemite tõesus ainult eelduse $\Box(A \,\&\, B)$ tõesusest.

Näide 12.4. Tõestame $\Box(p \supset q) \vdash_{S5} \Box p \supset \Box q$:

1	1.	$\Box(p \supset q)$	eeldus
2	2.	$\Box p$	hüpotees
1	3.	$p \supset q$	1 $\Box E$
2	4.	p	2 $\Box E$
1,2	5.	q	3,4 $\supset E$
1,2	6.	$\Box q$	5 $\Box I$ (sest $\Box(p \supset q)$ ja $\Box p$ on täielikult modaliseeritud)
1	7.	$\Box p \supset \Box q$	2,6 $\supset I$

Näide 12.5. Järgmiseks tõestame seaduse $\neg\Diamond p = \Box\neg p$. Selleks tuletame kõigepealt vasakpoolsest valemist parempoolse valemi:

1	1.	$\neg\Diamond p$	eeldus
2	2.	p	hüpotees
2	3.	$\Diamond p$	2 $\Diamond I$
1,2	4.	$\bot$	1,3 $\neg E$
1	5.	$\neg p$	2,4 $\neg I$
1	6.	$\Box\neg p$	5 $\Box I$ (sest 1. valem on täielikult modaliseeritud)

Seejärel tuletame parempoolsest valemist $\Box\neg p$ valemi $\neg\Diamond p$:

1	1.	$\Box\neg p$	eeldus
2	2.	$\Diamond p$	hüpotees
3	3.	p	hüpotees
1	4.	$\neg p$	1 $\Box E$
1,3	5.	$\bot$	3,4 $\neg E$
1,2	6.	$\bot$	2,3,5 $\Diamond E$ (sest 1. ja 5. valem on täielikult modaliseeritud)
1	7.	$\neg\Diamond p$	2,6 $\neg I$

Seaduse $\neg\Diamond p = \Box\neg p$ saamiseks tuleb nüüd kasutada implikatsiooni ja ekvivalentsi sissetoomisreegleid.

Paratamatuse sissetoomis- ja võimalikkuse eemaldamisreeglites esinevad kitsendused võivad mõnel juhul muuta otsesed tuletused võimatuks.

Näide 12.6. $A \vdash_{S5} \Box\Diamond A$ ei saa tuletada kujul

1	1.	A	eeldus
1	2.	$\Diamond A$	1 $\Diamond I$
1	3.	$\Box\Diamond A$	2 $\Box I$

sest lause A ei ole täielikult modaliseeritud, mistõttu 3. reas ei tohi paratamatuse sissetoomisreeglit rakendada. Võimalik on aga tuletus

1	1.	A	eeldus
1	2.	$\Diamond A$	1 $\Diamond I$
3	3.	$\Diamond A$	hüpotees
3	4.	$\Box\Diamond A$	3 $\Box I$ (sest 3. valem on täielikult modaliseeritud)
–	5.	$\Diamond A \supset \Box\Diamond A$	3,4 $\supset I$
1	6.	$\Box\Diamond A$	2,5 $\supset E$

kus kasutatakse kõigepealt implikatsiooni sissetoomisreeglit ning seejärel implikatsiooni eemaldamisreeglit.

Ülesanded

1. Konstrueerige tuletused:
   (a) $\Diamond A \supset \Box B \vdash_{S5} \Box A \supset \Box B$;
   (b) $\Box B \vdash_{S5} \Box(A \supset B)$;
   (c) $\Diamond(A \,\&\, \Diamond B) \vdash_{S5} \Diamond A \,\&\, \Diamond B$;
   (d) $\Diamond A \vee \Diamond B \vdash_{S5} \Diamond(A \vee B)$;
   (e) $\Diamond\Box A \vdash_{S5} A$;
   (f) $\vdash_{S5} \Diamond\Box A \supset \Box\Diamond A$.

12.5. Kanooniline tõlge predikaatloogika keelde

Modaalse lauseloogika laused saab tõlkida *monaadilise predikaatarvutuse* keelde, milles kasutatakse mitteloogilistest sümbolitest ainult ühekohalisi predikaatsümboleid. Selleks vaatleme paratamatust kui üldisuskvantorit ja võimalikkust kui eksistentsikvantorit üle võimalike maailmade universumi. Me vajame ainult ühte indiviidmuutujat, milleks võtame muutuja w. Tõlkimise käigus asendame lausemuutujad $A, B, \ldots$ atomaarsete valemitega $Aw, Bw, \ldots$. Näiteks modaalloogika laused

$$\Box A \supset A,$$
$$\Box(A \supset B) \supset (\Box A \supset \Box B),$$
$$\Diamond A \supset \Box\Diamond A$$

tõlgitakse predikaatarvutuse valemiteks

$$\forall w\, Aw \supset Aw,$$
$$\forall w\, (Aw \supset Bw) \supset (\forall w\, Aw \supset \forall w\, Bw),$$
$$\exists w\, Aw \supset \forall w\, \exists w\, Aw.$$

Seega tõlgitakse täielikult modaliseeritud laused kinnisteks predikaatarvutuse valemiteks, kusjuures korduvale modaliseerimisele vastab korduv kvantifitseerimine. Täielikult modaliseerimata lausete tõlkimisel saadakse aga vabade muutujatega predikaatarvutuse valemid.

Selgub, et esialgsed modaalse lauseloogika laused on tõlgitud valemitega semantiliselt samaväärsed (ülesanne 12.5.1). Samuti järeldub monaadilise predikaatarvutuse lahenduvusest modaalse lauseloogika lahenduvus.

Ülesanded

1. *** Olgu p modaalse lauseloogika lause ja p' tema kanooniline tõlge monaadilise predikaatloogika keelde. Näidake, et valemid p ja p' on semantiliselt samaväärsed, s.t. leiduvad interpretatsioonid I ja I', nii et $I \models p \Leftrightarrow I' \models p'$.

2. ***Tõestage, et monaadiline predikaatarvutus on lahenduv.

12.6. Erinevad tuletussüsteemid

Lewis tõi modaalloogika jaoks sisse mitu aksiomaatilist tuletussüsteemi. Süsteemi T aksioomideks on kõik lausearvutuse tautoloogiad, millele lisatakse aksioomiskeemid

(T) $\Box p \supset p$;

(K) $\Box(p \supset q) \supset (\Box p \supset \Box q)$,

s.t. iga paratamatu lause on tõene ning juhul, kui väitest p järeldub paratamatult väide q, järeldub p paratamatusest q paratamatus. Tuletusreegliteks on *modus ponens* ja *paratamatuse sissetoomine*

$$\frac{p}{\Box p}\ (N).$$

Jättes aksioomiskeemi (T) välja, saame tuletussüsteemi K, mida kasutatakse *deontilises loogikas*. Süsteemis K tõlgendatakse sümbolit $\Box$ mitte kui paratamatust, vaid kui *kohustuslikkust*. Sellisel juhul ei tarvitse aksioomiskeem (T) kehtida, sest meile võivad olla kohustuslikud asjad, mis ei ole tõesed. Näiteks võib olla kohustuslik, et ma teeksin kõik eksamid viitele, kuigi see tegelikult nii ei ole. *Teadmiste loogikas* tähistab sümbol $\Box$ seda, mis on teada. Kuna tavaliselt on meie teadmiste sisuks tõesed väited, jäetakse teadmiste loogikas aksioomiskeem (T) alles. Aga *uskumiste loogikas* visatakse (T) jälle välja, sest $\Box$ tähistab seal seda, millesse usutakse.

Lewise tuletussüsteemid $S4$ ja $S5$ on süsteemi T laiendused. Süsteemis $S4$ lisatakse aksioomiskeem

(4) $\Box p \supset \Box\Box p$,

mis ütleb, et kõik, mis on paratamatu, on seda paratamatult. Samuti on kõik, mille võimalikkus on võimalik, ka ise võimalik:

(4$\Diamond$) $\Diamond\Diamond p \supset \Diamond p$.

Tuletussüsteemi $S5$ saamiseks lisatakse süsteemile $S4$ aksioomiskeem

(5) $\Diamond p \supset \Box\Diamond p$.

Võimalike maailmade semantika abil saab selgitada ka nende süsteemide vahelisi erinevusi. Selleks toome lisaks maailmade süsteemile sisse ka maailmadevahelise *ligipääsetavuse* seose R ning defineerime paratamatuseoperaatori semantika järgmiselt:

- $w[\Box p] = t \Leftrightarrow \forall u\,(u \in \mathcal{W}\ \&\ w\ R\ u \supset u[p] = t)$,

s.t. mingi väite paratamatult tõesuseks maailmas w peab see väide olema tõene kõigis maailmades, kuhu sellest maailmast on võimalik ligi pääseda. Analoogiliselt defineeritakse ka võimalikkuseoperaatori semantika.

Sõltuvalt seose R algebralistest omadustest saame erinevate modaalloogika tuletussüsteemide interpretatsioonid. Süsteemis K ei seata seosele R mingeid tingimusi. Süsteemis T nõutakse, et seos R oleks *refleksiivne*[3]. $S4$-s peab R olema lisaks refleksiivsusele ka *transitiivne* ja süsteemis $S5$ on R *ekvivalentsiseos*, s.t. ta on üheaegselt nii refleksiivne, sümmeetriline kui ka transitiivne. Süsteemid K, T, $S4$ ja $S5$ on nimetatud interpretatsioonide suhtes täielikud, s.t. tõeste ja tuletatavate valemite hulgad langevad nendes kokku.

Ülesanded

1. Näidake, et aksioomiskeemid (4) ja (4$\Diamond$) on ekvivalentsed.
2. Tõestage, et aksioomiskeemidest (5) ja (T) järeldub (4).

Kuna süsteemidel K, T, $S4$ ja $S5$ on olemas *lõpliku mudeli omadus* (kui valemite hulgal leidub mudel, siis sellel hulgal leidub ka lõplik mudel), on kõik need süsteemid lahenduvad.

Süsteemi $S4$ interpretatsioone kasutatakse samuti intuitsionistlikus loogikas, sest me kujutame intuitsionistlikku loogikat ette kui ajas monotoonselt kasvavat tõestatud väidete hulka. Aja kulgemise seos on aga refleksiivne ja transitiivne.

12.7. Modaalne predikaatloogika

Modaalsest lausearvutusest on lihtne üle minna modaalsesse predikaatarvutusse. Selleks lisatakse predikaatarvutuse tähestikku modaalsete operaatorite sümbolid $\Diamond$ ja $\Box$. Lihtne on üldistada ka võimalike maailmade interpretatsioone, lisades maailmadesse indiviidid ning kujutades iga maailma kui ühte võimalikku predikaatloogika interpretatsiooni. Edasi tekivad aga filosoofilised probleemid, sest pole selge, mida sellised maailmad endast täpselt kujutavad.

[3] Binaarne seos R on *refleksiivne*, kui kehtib $\forall x\, x R x$, *sümmeetriline*, kui $\forall x \forall y\,(x R y \supset y R x)$, ja *transitiivne*, kui $\forall x \forall y \forall z\,(x R y\ \&\ y R z \supset x R z)$.

Näiteks ei tarvitse modaalses predikaatloogikas enam kehtida võrdusaksioomid. Quine toob näiteks *võrdsete elementide asendamise seaduse*

$$a = b \supset (Aa \supset Ab),$$

mille kehtimine on modaalses predikaatloogikas küsitav. Näiteks on päikesesüsteemis kokku 9 planeeti. On selge, et arv 9 on paratamatult suurem kui 7. Ent pole tõsi, et planeetide arv on paratamatult suurem kui 7.

Ülesanded

1. Näidake, et võrdsete elementide asendamise seadus ei kehti teadmiste ja uskumiste loogikates.

Modaalses predikaatloogikas pakuvad huvi ka *Barcani valem* ja selle pöördvalem, mis väidavad, et üldisuskvantor ja paratamatuseoperaator (või võimalikkuseoperaator ja eksistentsikvantor) kommuteeruvad omavahel:

$$\forall x \, \Box p = \Box \forall x \, p.$$

Võtame valemiks p lause Mx, mis ütleb, et "objekt x on materiaalne". Vasakpoolne valem võtab nüüd kokku *kontingentse materialisti* seisukoha: kõik tegelikus maailmas eksisteerivad asjad on igas maailmas materiaalsed. Parempoolne valem tähendab aga, et mittemateriaalseid asju ei saagi olla. Kontingentne materialist ei tarvitse sellega nõustuda, sest teistes maailmades võivad esineda ka mittemateriaalsed asjad: need on objektid, mis oleksid võinud eksisteerida, kuid tegelikult ei eksisteeri.

13. Intuitsionistlik loogika

On kahte tüüpi mõistusi … matemaatiline ja nn. intuitiivne. Esimene jõuab oma vaadeteni aeglaselt, kuid need on kindlad ja ranged; teist on õnnistatud suurema paindlikkusega ning ta tegeleb samaaegselt armastatud asjade erinevate armastusväärsete osadega.
— Blaise Pascal (1623–1662), *Discours sur les passions de l'amour*

13.1. Sissejuhatus

Intuitsionistliku loogika tekkimine ja aksiomatiseerimine on selle sajandi loogikas omamoodi paradoks. Intuitsionism tekkis matemaatika aluste uurimisel ja on teiste mitteklassikaliste loogikaharudega võrreldes rohkem seotud keeruliste matemaatikaprobleemidega, eriti aga matemaatiliste objektide olemasolu ja tõesuse mõistmisega. Rohkem kui teised mitteklassikalised loogikad on intuitsionistlik loogika matemaatiline selles mõttes, et uuritakse just matemaatika konstruktsioonides ja arvutustes kasutatavat-kehtivat loogikat. Intuitsionism on ka ainus mitteklassikaline süsteem, mida kasutades on läbi uuritud suured valdkonnad matemaatikas. Teiselt poolt on tõestamise ja arvutamise/konstrueerimise seoste uurimisel leitud, et intuitsionistlik loogika kirjeldab arvutatavust märksa paremini kui klassikaline, mis on esile kutsunud arvutiteaduse huvi intuitsionismi vastu.

Intuitsionistliku matemaatikafilosoofia tekkimise aluseks oli klassikalisega võrreldes erinev vastus küsimusele, mida uurib matemaatika. Ajalooliselt on domineerinud platonistlik e. realistlik seisukoht, mida väljendatakse sõnadega, et matemaatika uurib matemaatilisi objekte: arve, funktsioone, hulki jne. Matemaatik kujutab endale ette, et nende objektide maailm eksisteerib umbes samuti kui füüsiliste objektide maailm. Selles maailmas on iga väide kas tõene või väär, sõltumata sellest, kas me seda tõeväärtust teame või mitte. Näiteks Goldbachi hüpoteesil (iga paarisarvu, mis on suurem kui 2, saab esitada kahe algarvu summana) on matemaatiku arvates kindel tõeväärtus, kuigi keegi seda praegu ei tea. Matemaatikute tegevus seisneb realistide arvates väidete tõesuse tõestamises või vääruse kindlakstegemises — matemaatikud uurivad objektiivselt eksisteerivat maailma.

Tegelikult me teame, et matemaatiliste objektide maailm on inimeste välja mõeldud ja arve ega hulki ei eksisteeri kusagil väljaspool matemaatikat iseseisvana. Matemaatikute maailma on olemas just niipalju, kui seda seni on välja mõeldud. Sealhulgas on ka näiteks naturaalarvude hulgal ainult niipalju omadusi, kui on fikseerinud (aksioomidena, traditsiooniliste arusaamade ja ettekujutustena või mingil muul teel) senine matemaatika. See ei tähenda muidugi, et tõesed on ainult need väited, mida keegi juba on tõestanud. Näiteks on kindlasti selliseid naturaalarve x, y ja z, mille kohta keegi pole arvutanud, kas väide $x + y = z$ on tõene või väär. Matemaatikas on aga meetodid, millega saab suvaliste x, y ja z jaoks kindlaks teha, kas $x + y = z$ kehtib või mitte ja seepärast on sellisel väitel kindel tõeväärtus. Aga me teame, et saab formuleerida väiteid, mille tõesus ega väärus ei järeldu seni matemaatikas paikapandud tõdedest, ja matemaatika võib edaspidi areneda ühes või teises suunas, tehes mõne sellise väite tõeseks või vääraks. Kurt Gödel ja Paul Joseph Cohen (s. 1934) tõestasid, et meil pole alust pidada kontiinumhüpoteesi ei tõeseks ega vääraks. Me võime arendada matemaatikat, mille hulkade universumis kontiinumhüpotees on tõene, või matemaatikat, kus see on väär. Teades juba kolmkümmend aastat selle väite sõltumatust, pole matemaatikud kiirustanud talle tõeväärtust fikseerima, kuigi

Hilbert nimetas aastal 1900 tema kehtimise küsimust matemaatika lahtistest probleemidest esimesena. See kõik aga tähendab, et meil pole põhjust eeldada, et täna ja praegu on näiteks Goldbachi hüpoteesil või kontiinumhüpoteesil tõeväärtus. Seega pole niisugusest vaatepunktist alust ka $A \vee \neg A$ arvamisel üldkehtivaks.

Intuitsionismi loojaks peetakse hollandi loogikut L. E. J. Brouwerit (1881–1966), kes vastukaaluna D. Hilberti alustatud formalistlikule, aksiomatiseerimisel baseeruvale suunale matemaatika aluste põhjendamisel propageeris matemaatika ehitamist intuitsiooni abil. Brouweri järgi seisneb matemaatika intuitiivsetes konstruktsioonides ja intuitiivsetes arutlustes selliseid konstruktsioone käsitlevate väidete kohta, aga mitte formaalses deduktsioonis formaalselt kehtestatud aksioomidest. Brouwer oli ammu enne mittetäielikkuseteoreemi tõestamist Gödeli poolt (1931) veendunud, et matemaatiliste konstruktsioonide kogu mitmekesisus ei saa olla hõlmatud mingisuguse fikseeritud formaalse süsteemiga. Mis puutub loogikasse, siis Brouwer tahtis luua intuitsionistlikku matemaatikat ega seadnud mingeid loogikaprintsiipe väljapoole matemaatikat ega sellest kõrgemale. Loomulikult ei eeldanud tema ideed ka intuitsionistliku loogika aksiomatiseerimist.

Loogikud olid aga huvitatud intuitsionistlikus matemaatikas kasutatavate printsiipide ekstraheerimisest, mis saigi teoks intuitsionistliku predikaatarvutuse aksiomatiseerimisega A. Heytingi poolt 1930. aastal. Järgnenud arengu tulemusena mõistetakse tänapäeval intuitsionismi all enamasti intuitsionistlikku loogikat ja intuitsionistliku loogika all põhiliselt Heytingi predikaatarvutust. Sellele uurimistööle on stimuleerivalt mõjunud intuitsionistliku tuletatavuse ja arvutatavuse vaheliste seoste avastamine ning tõestustest algoritmide eraldamise võimalus.

Intuitsionistliku matemaatika tõestustes kasutavad aga erinevad matemaatikud peale intuitsionistlikus predikaatarvutuses formaliseeritud reeglite veel suhteliselt universaalseid, kuid kohati omavahel vastuolus olevaid printsiipe (Churchi tees, Markovi printsiip jm.). Nii ei saa me rääkida ühtsest intuitsionistlikust matemaatikast ega rangelt võttes ka loogikast. Küll aga moodustab

Heytingi arvutus ühisosa kõigist “mõistlikest” intuitsionismidest, s.t. ainult võrdlemisi radikaalsete vaadetega intuitsionistid ei tunnista selles arvutuses tuletatavate valemite üldkehtivust.

13.2. Kaks tõestuse näidet

Paljudele on teada, et intuitsionistlik matemaatika ei aktsepteeri nn. puhta olemasolu tõestusi ega tunnista üldise printsiibina välistatud kolmanda seadust. Toome kõigepealt ühe lihtsa näite, kuidas välistatud kolmanda seadust kasutades saab tõestada kindlate omadustega objekti olemasolu ilma objekti ennast esitamata.

Teoreem 13.1. *Leiduvad sellised irratsionaalsed arvud a ja b, et a^b on ratsionaalarv.*

Tõestus. Vaatleme arvu $\sqrt{2}^{\sqrt{2}}$. Kui $\sqrt{2}^{\sqrt{2}}$ on ratsionaalarv, siis on teoreemi väide tõene, sest sobivad $a_1 = \sqrt{2}$ ja $b_1 = \sqrt{2}$. Kui $\sqrt{2}^{\sqrt{2}}$ on irratsionaalarv, siis on teoreemi väide tõene, sest sobivad $a_2 = \sqrt{2}^{\sqrt{2}}$ ja $b_2 = \sqrt{2}$, kuna $a_2^{b_2} = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^2 = 2$. □

Oleme teoreemi klassikaliselt mõtleva matemaatiku jaoks tõestanud, aga tõestusest ei saa teada, millised a ja b väärtused rahuldavad teoreemi tingimusi. Tõestuses kasutasime välistatud kolmanda seadust, pidades tõeseks disjunktsiooni

$$\sqrt{2}^{\sqrt{2}} \text{ on ratsionaalne või } \sqrt{2}^{\sqrt{2}} \text{ on irratsionaalne.}$$

Intuitsionistlikus tõlgenduses nõutakse väite $\exists x\, A(x)$ tõestamisel ka sobiva x väärtuse esitamist (ja tema sobivuse tõestamist), disjunktsiooni $A \vee B$ tõestamisel aga tõese liikme näitamist koos tema kehtivuse tõestamisega. See tähendab, et valemeid kujul $A \vee \neg A$ ei saa suvalise valemi A jaoks iseenesest tõeseks arvata, vaid seda saab teha alles iga valemi A jaoks eraldi, tõestades

disjunktsiooni sobiva liikme. Näiteks on tegelikult teada, et $\sqrt{2}^{\sqrt{2}}$ on irratsionaalne ja seega on teoreem tõene arvude a_2 ja b_2 sobivuse tõttu. Niisugune tõestus on muidugi oluliselt keerulisem kui paarile reale mahtuv klassikaline puhta olemasolu tõestus, aga ta annab ka probleemi kohta rohkem informatsiooni.

Toome siin ka ühe intuitsionistliku tõestuse näite. Näitame, et kuigi $A \vee \neg A$ pole intuitsionisti jaoks üldkehtiv, saab tõestada selle valemi kahekordse eituse. Märgime, et valemi A eituse tõestamiseks peavad intuitsionistid piisavaks, et valemist A järeldatakse vastuolu.

Teoreem 13.2. *Valem $\neg\neg(A \vee \neg A)$ on tõene.*

Tõestus. Meil tuleb näidata, et $\neg(A \vee \neg A)$ annab vastuolu. Selleks oletame, et $\neg(A \vee \neg A)$ kehtib. Näitame kõigepealt, et siis kehtib $\neg A$. Tõepoolest. Kui kehtiks A, siis kehtiks esimese liikme tõesuse tõttu ka $A \vee \neg A$ ja see oleks vastuolus $\neg(A \vee \neg A)$ kehtimisega. Et kehtib $\neg A$, siis võime teise liikme tõesuse põhjal väita, et kehtib $A \vee \neg A$ ja oleme saanud vastuolu. Järelikult on $\neg(A \vee \neg A)$ väär ja kehtib $\neg\neg(A \vee \neg A)$. □

Juhime tähelepanu sellele, et siin me ei kasutanud välistatud kolmanda seadust $A \vee \neg A$, vaid järeldasime oletusest $\neg(A \vee \neg A)$ kõigepealt $\neg A$ ja siis näitasime, et oletus ise ei saa kehtida.

Näeme ka, et intuitsionistlikus loogikas pole põhjust lugeda valemit $\neg\neg A \supset A$ üldkehtivaks. Valemi $\neg\neg A$ tõestamiseks piisab $\neg A$ ümberlükkamisest, A tõestamine võib aga nõuda keeruliste objektide konstrueerimist. Kokkuvõttes on intuitsionistlikult seisukohalt loomulik eristada kolme liiki valemeid:
1) tõesed (s.t. tõestatud),
2) väärad (sellised, mille eitus on tõestatud),
3) praegu teadmata ja isegi olematu tõeväärtusega valemid.
Viimaste suhtes pole põhjust lugeda, et ka $A \vee \neg A$ kehtib.

Intuitsionistlik lähenemine teeb olukorra keerulisemaks kui klassikaline. Sellegipoolest on konstrueeritud vastuvõetavaid intuitsionistlikke teooriaid. Küll aga loobub tõsine intuitsionist kõigi

tõeste valemite hulga vaatlemisest. Mittetriviaalsetel juhtudel on see hulk üsna piiritlemata. Samuti on ühe ja sama keele jaoks tihti erisugused intuitsionistlikud semantikad, s.t. ka ühe ja sama valdkonna erinevad intuitsionistlikud teooriad.

13.3. Intuitsionistlik loogikaseoste mõistmine

Intuitsionisti seisukohalt pole aritmeetika, matemaatilise analüüsi, hulgateooria jt. matemaatiliste distsipliinide väidete semantika lõpuni selge, sest pole täpselt selged nende teooriate (lõpmatud) objektid. Ei saa olla ka protseduuri, mis kontrolliks nende teooriate valemite tõesust. Seetõttu tuleb piirduda ainult semantiliste kokkulepetega, mida valemite tõlgendus peaks kindlasti rahuldama. Nende kokkulepete põhjal on ka võimalik formuleerida aksiomaatilisi teooriaid, mis formaliseerivad teatud aspekte sisulisest matemaatilisest teooriast (teatud põhjusi valemite tõesuseks), mitte aga kogu semantikat.

Iga kinnine valem väljendab mingit (üldiselt mittetäielikku) teadet matemaatilisest konstruktsioonist. Valemi tõesust saab näidata seda tõendava konstruktsiooni esitamisega. See konstruktsioon peab vastama teatavatele valemi kujust tulenevatele tingimustele — semantilistele kokkulepetele.

- Kui valem A on kujul $A_1 \,\&\, A_2$, siis k *tõendab* A parajasti siis, kui temast võib saada konstruktsioonid k_1 ja k_2, mis tõendavad vastavalt A_1 ja A_2. (Võib arvata, et A tõenduseks on paar, mis koosneb A_1 ja A_2 tõendustest).

- Kui valem A on kujul $A_1 \vee A_2$, siis k tõendab A parajasti siis, kui temast võib saada informatsiooni selle kohta, kumb A_i on tõene, ja selle A_i tõenduse.

- Kui valem A on kujul $A_1 \supset A_2$, siis k tõendab A parajasti siis, kui k annab üldise meetodi, mis teisendab A_1 suvalise tõenduse A_2 tõenduseks.

- Kui valem A on kujul $\neg B$, siis k tõendab A parajasti siis, kui k tõendab $A \supset \bot$, kus $\bot$ tähistab standardset väära lauset.

- Kui valem A on kujul $\exists x\, B(x)$, siis k tõendab A parajasti siis, kui k esitab x muutumispiirkonna mingi elemendi c ja valemi $B(c)$ tõenduse.

- Kui valem A on kujul $\forall x\, A(x)$, siis k tõendab A parajasti siis, kui k annab üldise meetodi, millega saab x muutumispiirkonna iga elemendi d jaoks leida valemi $A(d)$ tõenduse.

Kommenteerime veel pisut neid kokkuleppeid.

1. Kõigepealt, antud tingimusi tuleb tõepoolest vaadelda kui tingimusi, s.t. nad ei ole tehete ja kvantorite matemaatilised definitsioonid. Esiteks pole neis fikseeritud, mis laadi objektid võivad olla valemite tõendused: tõestused, algoritmid, nende mingid formaalsed esitused vms. Intuitsionistid ei fikseeri ka, mida mõistetakse üldise meetodina. Siin oleks küllalt loomulik mõista meetoditena algoritme mingis formalisatsioonis. Formaalsete detailide fikseerimine võib anda juba mingi konkreetse intuitsionistliku semantika (näiteks Kleene defineeritud rekursiivse realiseeritavuse). Samuti ei tähenda semantilised kokkulepped keerulisemate mõistete taandamist lihtsamatele. Esitatakse ju tingimused üldisuskvantori ja implikatsiooni jaoks omakorda üldisuskvantorit kasutades.

2. Millises vahekorras on kokkulepped klassikalise arusaamisega tõestamisest? Kui mõista tõendavate objektidena tõestusi, siis on tingimused kõigil juhtudel klassiku jaoks kindlasti piisavad, ilmselt aga mitte tarvilikud.

3. Välja arvatud eituse juht, esitatakse valemi tõendusele nõuded antud valemi osavalemite kaudu, mis viitab tõestuse otsimiseks soodsa osavalemi omaduse kehtimisele.

4. Intuitsionistliku semantika järgi on erinevalt klassikalisest loogikast kõik loogikatehted peale eituse iseseisvad ja neid ei saa avaldada teiste kaudu. Näiteks ei kehti samaväärsused ühe kvantori avaldamiseks teise kvantori ja eituse kaudu.

13.4. Intuitsionistlik predikaatarvutus

Ajalooliselt tekkis intuitsionistlik predikaatarvutus klassikalisest arvutusest. Me võime uurida klassikalise arvutuse reegleid ja aksioome ning kontrollida, millised neist on kooskõlas meie semantiliste kokkulepetega ja millised tuleb asendada uutega (nõrgematega). Kui võtame seejuures aluseks paragrahvis 6.1 toodud Hilberti tüüpi lausearvutuse K, siis avastame, et ainsana ei vasta intuitsionistlikele semantilistele kokkulepetele aksioom (K10): $\neg\neg p \supset p$. Klassikalised aksioomid ja reeglid kvantorite kohta on aga kõik aktsepteeritavad. Asendades aksioomi (K10) intuitsionistlikele kokkulepetele vastava aksioomiga $p \supset (\neg p \supset q)$, saame intuitsionistliku predikaatarvutuse, kus aksioomid on

(K1)–(K9)

(I10) $p \supset (\neg p \supset q)$

(H4)–(H5)

ja tuletusreeglid

modus ponens

(Gen)

Et lisatud aksioom (I10) on klassikaliselt tuletatav (samaselt tõene), siis on iga intuitsionistlikus süsteemis tuletatav valem seda samuti.

Gentzen näitas, et ülaltooduga samaväärse sekventsiaalse intuitsionistliku predikaatarvutuse saame, kui lubame paragrahvis 6.3 formuleeritud süsteemis kasutada vaid selliseid sekventse, mille suktsedent on tühi või sisaldab ainult ühte valemit. Jättes suktsedentidest liigsed liikmed välja, võime loogilised reeglid formuleerida tabelis 13.1 toodud viisil, kus Θ on kõikjal tühi või koosneb ühest valemist ning muutuja a ei esine reeglites $(\Rightarrow\forall)$ ja $(\exists\Rightarrow)$ vabalt Γ ja Θ valemites.

$$\frac{A, \Gamma \Rightarrow \Theta}{A \,\&\, B, \Gamma \Rightarrow \Theta}, \frac{B, \Gamma \Rightarrow \Theta}{A \,\&\, B, \Gamma \Rightarrow \Theta}\ (\&\Rightarrow) \qquad \frac{\Gamma \Rightarrow A \quad \Gamma \Rightarrow B}{\Gamma \Rightarrow A \,\&\, B}\ (\Rightarrow\&)$$

$$\frac{A, \Gamma \Rightarrow \Theta \quad B, \Gamma \Rightarrow \Theta}{A \vee B, \Gamma \Rightarrow \Theta}\ (\vee\Rightarrow) \qquad \frac{\Gamma \Rightarrow A}{\Gamma \Rightarrow A \vee B}, \frac{\Gamma \Rightarrow B}{\Gamma \Rightarrow A \vee B}\ (\Rightarrow\vee)$$

$$\frac{\Gamma \Rightarrow A \quad B, \Delta \Rightarrow \Lambda}{A \supset B, \Gamma, \Delta \Rightarrow \Lambda}\ (\supset\Rightarrow) \qquad \frac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \supset B}\ (\Rightarrow\supset)$$

$$\frac{\Gamma \Rightarrow A}{\neg A, \Gamma \Rightarrow}\ (\neg\Rightarrow) \qquad \frac{A, \Gamma \Rightarrow}{\Gamma \Rightarrow \neg A}\ (\Rightarrow\neg)$$

$$\frac{A(x/t), \Gamma \Rightarrow \Theta}{\forall x\, A(x), \Gamma \Rightarrow \Theta}\ (\forall\Rightarrow) \qquad \frac{\Gamma \Rightarrow A(x/a)}{\Gamma \Rightarrow \forall x\, A(x)}\ (\Rightarrow\forall)$$

$$\frac{A(x/a), \Gamma \Rightarrow \Theta}{\exists x\, A(x), \Gamma \Rightarrow \Theta}\ (\exists\Rightarrow) \qquad \frac{\Gamma \Rightarrow A(x/t)}{\Gamma \Rightarrow \exists x\, A(x)}\ (\Rightarrow\exists)$$

Tabel 13.1. Intuitsionistlik tuletussüsteem.

Struktuursetest reeglitest kaotavad kahe valemiga suktsedenti sisaldavad reeglid oma mõtte. Et ka intuitsionistliku süsteemi jaoks saab tõestada lõikereegli elimineeritavuse, jäävad alles kolm reeglit antetsedendi jaoks ja üks suktsedendi jaoks:

$$\frac{\Gamma \Rightarrow \Theta}{C, \Gamma \Rightarrow \Theta} \qquad \frac{\Delta, C, D, \Gamma \Rightarrow \Theta}{\Delta, D, C, \Gamma \Rightarrow \Theta}$$

$$\frac{C, C, \Gamma \Rightarrow \Theta}{C, \Gamma \Rightarrow \Theta} \qquad \frac{\Gamma \Rightarrow}{\Gamma \Rightarrow C}$$

Paragrahvis 13.2 toodud valemi $\neg\neg(A \vee \neg A)$ tõestuse saame

formaliseerida järgmise tuletusena:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{A \Rightarrow A \vee \neg A}\,(\Rightarrow\vee)}{\neg(A \vee \neg A), A \Rightarrow}\,(\neg\Rightarrow)}{A, \neg(A \vee \neg A) \Rightarrow}}{\neg(A \vee \neg A) \Rightarrow \neg A}\,(\neg\Rightarrow)}{\neg(A \vee \neg A) \Rightarrow A \vee \neg A}\,(\Rightarrow\vee)}{\neg(A \vee \neg A), \neg(A \vee \neg A) \Rightarrow}\,(\neg\Rightarrow)}{\neg(A \vee \neg A) \Rightarrow}}{\Rightarrow \neg\neg(A \vee \neg A)}\,(\Rightarrow\neg)$$

Paragrahvis 6.3 toodud konjunktsiooni kommutatiivsuse seaduse tuletuses on suktsedentides ainult üks valem. Järelikult kuulub see tuletus ka intuitsionistlikku arvutusse. Sekvents $\Rightarrow \forall x\, P(x) \vee \exists x \neg P(x)$ tuletatakse seal aga disjunktsiooni teise eksemplari sissetoomisega suktsedenti. Seega on see tõestus klassikaline. Veelgi enam. Ainsad reeglid, mille rakendamise tulemusena võiks selle sekventsi saada, on $(\Rightarrow\vee)$ ja valemi lisamine suktsedenti. Aga siis peaks olema tuletatav üks sekventsidest $\Rightarrow \forall x\, P(x)$, $\Rightarrow \exists x \neg P(x)$ või $\Rightarrow$. Need sekventsid aga pole ilmselt isegi klassikaliselt tuletatavad (nad pole samaselt tõesed). Näeme jälle, et sekventsionaalse arvutuse jaoks on mõnikord lihtne tõestada ka mittetuletatavust.

13.5. Kripke mudelid

Intuitsionistlikku tõesust on püütud formaliseerida ka Kripke mudelitega, mis kirjeldavad teadmiste võimalikku arengut. Aluseks võetakse modaalloogika $S4$, mille mudelites saab maailmad antisümmeetrilisuse lisatingimuse abil osaliselt järjestada[1]. Intuitsionistliku loogika Kripke mudelites nõutakse, et see osaline järjestus

[1]Binaarne seos R on *antisümmeetriline*, kui $\forall x \forall y\, (xRy \,\&\, yRx \supset x = y)$. *Osaline järjestus* on refleksiivne, antisümmeetriline ja transitiivne binaarne seos.

oleks puukujuline[2], mille juureks (minimaalseks elemendiks) on tegelik maailm. Puuharud kirjeldavad väidete tõendamise protsessi võimalikke arenguid, kusjuures juba tõendatud väited jäävad tõeseks ka edaspidi.

Definitsioon 13.3. Intuitsionistliku predikaatloogika *Kripke mudel* mingi signatuuri σ jaoks on

$$\mathcal{K} = \langle \mathcal{W}, \leq, w^*, dom, I \rangle,$$

kus $\mathcal{W}$ on võimalike maailmade hulk, w^* on tegelik maailm, $\leq$ on maailmadevahelise (puukujulise) järjestuse seos, dom annab iga maailma universumi ja I määrab iga maailma interpretatsiooni.

Me eeldame, et maailmade universumid kasvavad tõendatud väidete hulga kasvamisel monotoonselt:

$$w \leq w' \Rightarrow dom(w) \subseteq dom(w').$$

Loogiliste tehete semantika anname järgmiselt:

1. $w \models p \Leftrightarrow I(w) \models p$ (p on atomaarne valem).

2. $w \models p \,\&\, q \Leftrightarrow w \models p$ ja $w \models q$.
   $w \models p \vee q \Leftrightarrow w \models p$ või $w \models q$.
   $w \models p \supset q \Leftrightarrow$ iga $w' \geq w$ korral $w' \models p \Rightarrow w' \models q$.
   $w \models \neg p \Leftrightarrow$ iga $w' \geq w$ korral $w' \not\models p$.

3. $w \models \forall x\, p \Leftrightarrow$ iga $w' \geq w$ ja iga $c \in dom(w')$ korral $w' \models p\,\{x/c\}$.
   $w \models \exists x\, p \Leftrightarrow$ leidub $c \in dom(w)$, nii et $w \models p\,\{x/c\}$.

4. $\mathcal{K} \models p \Leftrightarrow w^* \models p$,

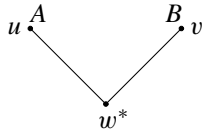
kus $w \in \mathcal{W}$ on suvaline võimalik maailm ja w^* on tegelik maailm. Tähistame intuitsionistliku järeldumist sümboliga $\models_I$.

[2]S.t. tsükliteta sidus orienteeritud juurega graaf.

Näide 13.4. Näitame, et

$$\neg(A \,\&\, B) \not\models_I \neg A \vee \neg B.$$

Valime kolmest maailmast koosneva Kripke mudeli



$$\mathcal{K} = \langle \{w^*, u, v\}, \leq, w^*, dom, I \rangle,$$

kus väide A tõendatakse ühes puuharus ja B teises puuharus.

$$\begin{aligned} &\leq = \{(w^*, u), (w^*, v)\}, \\ &dom(w^*) = dom(u) = dom(v) = \emptyset, \\ &I(w^*) = \emptyset, I(u) = \{A\}, I(v) = \{B\}. \end{aligned}$$

Selles mudelis kehtib $w^* \models \neg(A \,\&\, B)$, sest ükski tema maailm ei tõenda valemit $A \,\&\, B$: $A \notin I(w^*), I(v)$ ja $B \notin I(u)$. Kuid $w^* \not\models \neg A$, sest $w^* \leq u$ ja $u \models A$; samuti $w^* \not\models \neg B$, sest $w^* \leq v$ ja $v \models B$. Järelikult $w^* \not\models \neg A \vee \neg B$.

Näide 13.5. Tõestame, et

$$\forall x\, (Fx \vee A) \not\models_I \forall x\, Fx \vee A.$$

Tõestamiseks valime Kripke mudeli

$$\mathcal{K} = \langle \{w^*, u\}, \leq, w^*, dom, I \rangle,$$

kus

$$\begin{aligned} &\leq = \{(w^*, u)\}, \\ &dom(w^*) = \{a\} \subseteq dom(u) = \{a, b\}, \\ &I(w^*) = \{Fa\} \subseteq I(u) = \{Fa, A\}. \end{aligned}$$

$w^* \models Fa \vee A$, sest $w^* \models Fa$; $u \models Fa \vee A$ ja $u \models Fb \vee A$, sest $u \models A$. Seega $w^* \models \forall x\, (Fx \vee A)$.

Kuid $w^* \not\models A$, sest $A \notin I(w^*)$, ja $w^* \not\models \forall x\, Fx$, sest $u \geq w^*$, $b \in dom(u)$ ja $Fb \notin I(u)$. Seega $w^* \not\models \forall x\, Fx \vee A$.

Saab näidata, et Kripke semantika on intuitsionistlike tuletussüsteemide suhtes korrektne ja täielik.

Teoreem 13.6 (täielikkus). $\Gamma \vdash_I p \Leftrightarrow \Gamma \models_I p$.

Klassikaliselt tõesed laused saab kergesti teisendada intuitsionistlikult tõesteks lauseteks: kirjutame iga valemi ette kaks eituseseost.

Teoreem 13.7. $\Gamma \models p \Leftrightarrow \neg\neg\Gamma \models_I \neg\neg p$.

Järelikult võib intuitsionistlikku loogikat vaadata kui klassikalise loogika üldistust, millest saab valemite klassi piiramisega eraldada klassikalise fragmendi.

13.6. Rekursiivne realiseeritavus. Nelsoni teoreem

Kripke mudeleid võib konstrueerida suvalise signatuuri jaoks. Vaatleme veel ühte konkreetset aritmeetikavalemitele antud intuitsionistlikku semantikat, mis seob ka intuitsionistliku tuletatavuse ja algoritmilise arvutatavuse. Rekursiivse realiseeritavuse defineeris 1945. aastal Stephen Cole Kleene kui ühe võimaliku intuitsionistliku semantika.

Defineerime relatsiooni e **r** A, kus $e \in \mathbb{N}$ ja A on kinnine valem signatuuris $\langle 0; ', +, \cdot; = \rangle$: "Naturaalarv e realiseerib valemi A". Tähis φ_e tähendab definitsioonis funktsiooni, mida arvutab Turingi masin M_e, s.t. e on seda funktsiooni arvutava algoritmi kood.

1. Kui A on atomaarne aritmeetikavalem $s = t$, kus s ja t on (kinnised, s.t. muutujateta) termid, siis e **r** $s = t \Leftrightarrow e = 0$ ja $s = t$ on tõene.
2. e **r** $A \,\&\, B \Leftrightarrow e = 2^a 3^b$, kus a **r** A ja b **r** B.
3. e **r** $A \vee B \Leftrightarrow e = 2^0 3^a$, kus a **r** A, või $e = 2^1 3^b$, kus b **r** B.

4. $e\ \mathbf{r}\ A \supset B \Leftrightarrow \forall n\,[n\ \mathbf{r}\ A \Rightarrow [\varphi_e(n)\ \mathbf{r}\ B]]$.
5. $e\ \mathbf{r}\ \neg A \Leftrightarrow e\ \mathbf{r}\ (A \supset 0 = 1)$.
6. $e\ \mathbf{r}\ \exists x\ A(x) \Leftrightarrow e = 2^a 3^b$, kus $a\ \mathbf{r}\ A(b)$.
7. $e\ \mathbf{r}\ \forall x\ A(x) \Leftrightarrow \forall n\,[\varphi_e(n)\ \mathbf{r}\ A(n)]$.

See definitsioon vastab intuitsionistliku semantika üldisele skeemile paragrahvis 13.3, kusjuures tõendavateks konstruktsioonideks on valitud naturaalarvud (vaadelduna teatud algoritmi koodidena), üldiseks meetodiks rekursiivsed funktsioonid ja standardseks vääraks valemiks on võetud $0 = 1$. Oluline on siin, et induktsiooni baas — muutujateta termide võrdus — on intuitsionisti jaoks piisavalt üheselt mõistetav: selliste termide väärtused saab arvutada ja võrduse kehtimist kontrollida. Teiselt poolt võib rekursiivset realiseeritavust mõista kui klassikule arusaadavat "täpse definitsiooniga" antud intuitsionistlikku semantikat.

Definitsioon 13.8. Vabade muutujatega valemite jaoks defineeritakse:

$$e\ \mathbf{r}\ A(x_1, ..., x_n) \Leftrightarrow \forall k_1 \ldots \forall k_n\,[\varphi_e(k_1, \ldots, k_n)\ \mathbf{r}\ A(k_1, ..., k_n)],$$

s.t. tingimus on sama nagu $e\ \mathbf{r}\ \forall x_1 ... x_n\ A(x_1, \ldots, x_n)$ jaoks.

Definitsioon 13.9. Valemit A nimetame *realiseeritavaks* ja kirjutame $\mathbf{r}\ A$, kui leidub selline arv e, et $e\ \mathbf{r}\ A$.

Millist tõendavat informatsiooni sisaldab realiseeriv arv? Nagu näha, kodeerivad realisatsioonid huvitavamatel juhtudel teatud algoritme. Uurime, mis (milliste algoritmide olemasolu) on vajalik nii- või teistsuguse ehitusega valemi realiseeritavuseks?

1. Vaatleme valemeid kujul $\neg A$. Definitsiooni järgi tingimust lahti kirjutades saame:

$$e\ \mathbf{r}\ \neg A \Leftrightarrow$$
$$e\ \mathbf{r}\ (A \supset 0 = 1) \Leftrightarrow$$
$$\forall n\,[n\ \mathbf{r}\ A \Rightarrow [\varphi_e(n)\ \mathbf{r}\ 0 = 1]].$$

Et $0 = 1$ kui väär atomaarne valem ei ole definitsiooni järgi realiseeritav, siis saab $\neg A$ olla realiseeritav ainult juhul, kui ükski arv n ei realiseeri A. A mitterealiseeritavus on aga ka piisav selleks, et suvaline arv e realiseeriks $\neg A$, sest implikatsioon on siis vasaku poole vääruse tõttu tõene arvust $\varphi_e(n)$ sõltumata. Seega: $\neg A$ on realiseeritav parajasti siis, kui A pole realiseeritav, ja sellisel juhul realiseerib eitust iga naturaalarv. Arvutuslikust küljest tähendab see, et eituse realisatsioon ei sisalda mingit informatsiooni.

Samuti kehtib:

$$\mathbf{r}\ \neg\neg A \text{ parajasti siis, kui pole } \mathbf{r}\ \neg A,$$
$$\text{s.t. parajasti siis, kui } \mathbf{r}\ A.$$

Oleme sellega tõestanud, et realiseeritavusloogikas kehtib välimise kahekordse eituse ärajätmise seadus (see ei tähenda aga veel $\neg\neg A \supset A$ realiseeritavust!). Eespool veendusime, et alati niisugune seadus intuitsionistlikus loogikas ei kehti.

2. Vaatleme valemeid kujul $\forall x\,[A(x) \vee \neg A(x)]$. Definitsiooni lahti kirjutades saame:

$$e\ \mathbf{r}\ \forall x\,[A(x) \vee \neg A(x)] \Leftrightarrow$$
$$\forall n\,[\varphi_e(n)\ \mathbf{r}\ [A(n) \vee \neg A(n)]] \Leftrightarrow$$
$$\forall n\,[\varphi_e(n) = 2^0 3^a, \text{ kus } a\ \mathbf{r}\ A(n), \text{ või}$$
$$e = 2^1 3^b, \text{ kus } b\ \mathbf{r}\ \neg A(n)].$$

Niisiis võime suvalise n korral pärast $\varphi_e(n)$ arvutamist leida, kas see on paaritu või paarisarv ja nii teada saada, kas kehtib $A(n)$ või $\neg A(n)$. Seega kodeerib realiseeriv arv e muuhulgas omadust $A(x)$ lahendavat algoritmi ja järelikult on sellisel kujul oleva valemi realiseeritavuseks tarvilik omaduse $A(x)$ algoritmiline lahenduvus. Klassikaliselt on aga iga valem kujul $\forall x\,[A(x) \vee \neg A(x)]$ tuletatav. Järelikult leidub klassikaliselt tuletatav mitterealiseeritav valem.

3. Valemid kujul $\forall x \exists y\, A(x, y)$. Saame:

$$e \mathbf{r} \forall x \exists y\, A(x, y) \Leftrightarrow$$
$$\forall n\, [\varphi_e(n) \mathbf{r} \exists y\, A(n, y)] \Leftrightarrow$$
$$\forall n\, [\varphi_e(n) = 2^a 3^b, \text{ kus } a \mathbf{r} A(n, b)].$$

Seega saab realiseeriva algoritmi abil iga x jaoks arvutada sellise y, et $A(x, y)$ on realiseeritav, ja ka selle valemi realisatsiooni.

Matemaatikas tõestatakse tihti viimati toodud kujul olevaid teoreeme, mis väidavad, et parameetrite x suvaliste väärtuste korral leidub vaadeldaval võrrandil või võrrandisüsteemil lahend y. Teoreemi praktiliseks kasutamiseks on vaja ka algoritmi, mis võrrandi parameetrite järgi leiab lahendi. Järgmine teoreem näitab, et kui teoreem on tõestatud intuitsionistlikus süsteemis, siis tõestusest saab eraldada ka algoritmi.

Teoreem 13.10 (Nelson, 1947). *Iga intuitsionistlikus aritmeetikas tuletatav valem on rekursiivselt realiseeritav. Leidub algoritm, mis valemi tuletuse järgi leiab realisatsiooni.*

Teoreem tõestatakse induktsiooniga tuletuse ehituse järgi, näidates et

1) intuitsionistliku loogika ja aritmeetika aksioomid on realiseeritavad;

2) tuletusreeglite kohta kehtib: leidub algoritm, mis ülemise valemi suvalise realisatsiooni järgi leiab alumise valemi realisatsiooni.

13.7. Ülesanded

1. Näidake, et
   (a) $\neg\neg A \not\models_I A$;
   (b) $\not\models_I A \vee \neg\neg A$;
   (c) $\neg\forall x\, Fx \not\models_I \exists x\, \neg Fx$.

2. Tõestage, et $\neg A \vdash_I A \supset B$.

14. Hägusloogika

Et midagi saavutada ning selles kindel olla,
peab sihikule võtma suure.
— John Stuart Mill (1806–1873)

Vale, aga jäägitu.
Arg, aga vapper.
Vale, aga tõde.
— Linnar Priimägi

14.1. Soriitide paradoksid

Vaatleme *õunte rivi paradoksi*. Oletame, et me korjasime aiast sada õuna. Jaotame õunad punasteks, kollasteks ja rohelisteks. Järjestame igat liiki õunad värvi intensiivsuse järgi, nii et tulemuseks on pikk õunte rivi, mis algab päris roheliste õuntega; edasi tulevad kollakasrohelised toonid, mis lähevad järkjärgult üle kollasteks toonideks; kollastele õuntele järgnevad kollakaspunased õunad ning rivi lõpeb täiesti punaste õuntega.

Tähistame selle rivi õunad

$$\tilde{o}_1, \tilde{o}_2, \ldots, \tilde{o}_{100}.$$

Me saame sõnastada õunte rivi kohta kolm väidet:

1. Esimene õun $\tilde{o}_1$ on roheline.
2. Kui $\tilde{o}_i$ on roheline, siis $\tilde{o}_{i+1}$ on roheline $(1 \leq i < 100)$.
3. Viimane õun $\tilde{o}_{100}$ ei ole roheline.

Väited 1 ja 3 on ilmselt tõesed. Väide 2 tundub küll kahtlane, kuid ta on pigem tõene kui väär. Tema vääraks tunnistamiseks peaks rivis leiduma selline järjestikuste õunte paar, kus eelmine õun on roheline, järgmine õun ei ole aga roheline. Niisuguse paari leidmine on aga küsitav, sest järjestikused õunad on praktiliselt sama värvi.

Need kolm väidet moodustavad vasturääkiva lausete hulga. Selles veendumiseks piisab ainult lausearvutuse kasutamisest. Kirjutades teise väite iga õunte paari jaoks eraldi välja, saame väidete jada:

- $\tilde{o}_1$ *on roheline,*
- $\tilde{o}_1$ *on roheline* $\Rightarrow$ $\tilde{o}_2$ *on roheline,*
- $\tilde{o}_2$ *on roheline* $\Rightarrow$ $\tilde{o}_3$ *on roheline,*
- ...
- $\tilde{o}_{99}$ *on roheline* $\Rightarrow$ $\tilde{o}_{100}$ *on roheline,*

millest järeldub *modus ponens*'i korduva rakendamise teel väide

$$\tilde{o}_{100} \textit{ on roheline,}$$

mis on vastuolus 3. väitega.

Sellist tüüpi arutlusi nimetatakse *soriitide* (aheljärelduste) *paradoksideks*. Loetleme veel mõnda soriitide paradoksi.

- *Kuhjaparadoks*. 100 000 liivatera moodustavad kuhja. Ühe liivatera kuhjast eemaldamine ei likvideeri kuhja. Järelikult moodustab ka üks liivatera kuhja.

- *Inimeksistentsi paradoks*. Iga imik on inimene. Kui miski on inimene ajahetkel t, siis on ta seda ka ajahetkel $t - \delta$, kus δ on vabalt valitav ajavahemik. Seega on viljastunud munarakk ka inimene.

Kõik need arutlused põhinevad samal nähtusel, kus objektide omadused muutuvad suuremas vahemikus, kui predikaatidega on võimalik kirjeldada. Väikse muutuse korral jääb predikaat samaks, kuid suure muutuse korral predikaat muutub. Paradoksides annab väikeste muutuste jada kokkuvõttes suure muutuse. Antud juhul ei ole viga mitte eeldustes, vaid loogilises aparatuuris. Üks võimalik lahendus on üleminek kahevalentsest loogikast *mitmevalentsesse loogikasse*.

Ülesanded

1. Kiilaspea paradoks. Tõestage, et mitte keegi ei ole kiilaspea.
2. Hääletamisea paradoks. Tõestage, et ükski inimene pole hääleõiguse omamiseks kunagi piisavalt küps.

14.2. Hägusloogika semantika

Eelmises punktis toodud õunte rivi näite korral muutus predikaadi *roheline* rakendatavusaste pidevalt väiksemaks. Samas suurenes pidevalt predikaatide *kollane* ja *punane* rakendatavus. Klassikalises loogikas on iga väide kas tõene või väär. Seega kehtib iga predikaadi $p(x)$ korral

$$p(x) \in \{t, v\}.$$

Hägusloogikas võetakse predikaadi rakendatavusastme tähistamiseks kasutusele predikaadi *tõesusaste*

$$p(x) \in [0, 1],$$

kus tõeväärtuste hulk on asendatud reaalarvulise lõiguga $[0, 1]$, nii et 0 on *väär* ja 1 on *tõene*. See tähendab, et me loeme lauset $p(x)$ tõeseks niivõrd, kuivõrd predikaat p on objektile x rakendatav.

Eesti keeles ei kasutata mitte ainult kahte tõesusastet: tõene ja väär, vaid tõesus varieerub sageli kindlas vahemikus. Me võime

näiteks mingi lause kohta öelda, et see on *täiesti tõene*, *osaliselt tõene*, *peaaegu väär* vms.

Hägusloogika semantika kirjeldamiseks laiendame klassikalise interpretatsiooni mõistet. *Hägusinterpretatsioon* on loetelu

$$\mathcal{F} = \langle U_{\mathcal{F}}; Const_{\mathcal{F}}; Pred_{\mathcal{F}} \rangle,$$

kus $U_{\mathcal{F}}$ on interpretatsiooni universum, $Const_{\mathcal{F}}$ seab signatuuri igale konstantsümbolile c vastavusse universumi $U_{\mathcal{F}}$ mingi elemendi $c_{\mathcal{F}}$ ja $Pred_{\mathcal{F}}$ määrab kasutatavate predikaatide tõesusastmed kõigil universumi elementidel.

Klassikalist interpretatsiooni saab sellisel juhul vaadelda kui hägusinterpretatsiooni erijuhtu, kus kasutatakse ainult tõesusastmeid 0 ja 1. Samas peab hägusloogika semantika olema vastavuses loogikareeglitega ja meie intuitsiooniga ning lahendama soriitide paradoksid.

Me asendasime tõeväärtused *tõene* ja *väär* vastavalt reaalarvudega 1 ja 0. Sellisel juhul on mõistlik vaadelda tõesusastme x eitust kui tõesusastet $1 - x$. Tõesusastmete x ja y konjunktsiooni väärtuseks võib võtta neist arvudest väiksema. Nende disjunktsiooni väärtuseks sobib aga suurem arvudest x ja y. Nii annavad klassikalised loogikatehted ja hägustehted tõesusastmetel 1 ja 0 sama tulemuse. Miinimum- ja maksimumfunktsioonide kasutamine on antud juhul loomulik, sest nendel funktsioonidel on konjunktsiooniga ja disjunktsiooniga palju sarnaseid omadusi. Nad on näiteks kommutatiivsed ja assotsiatiivsed. Samuti on nad oma argumentide suhtes monotoonselt mittekahanevad.

Nii saame hägusloogika semantikareeglid:

1. Atomaarse valemi p tõesusastme $\mathcal{F}[p]$ määrab tema hägusinterpretatsioon $p_{\mathcal{F}}$:
   $\mathcal{F}[p] = p_{\mathcal{F}}$.

2. $\mathcal{F}[\neg p] = 1 - \mathcal{F}[p]$.

3. $\mathcal{F}[p \,\&\, q] = \min\{\mathcal{F}[p], \mathcal{F}[q]\}$,
   $\mathcal{F}[p \vee q] = \max\{\mathcal{F}[p], \mathcal{F}[q]\}$.

Võttes näiteks $\mathcal{F}[A] = 0{,}75$, saame

$$\mathcal{F}[\neg A] = 1 - \mathcal{F}[A] = 0{,}25\,.$$

Seega on *peaaegu tõese* lause eitus *praktiliselt väär*. Lihtne on veenduda, et hägusloogikas ei kehti vasturääkivusseadus ega välistatud kolmanda seadus. Meie näite korral on tõesusaste

$$\begin{aligned}\mathcal{F}[A \,\&\, \neg A] &= \min\{\mathcal{F}[A], \mathcal{F}[\neg A]\}\\ &= \min\{0{,}75\,,\ 0{,}25\}\\ &= 0{,}25\,,\end{aligned}$$

mis on tunduvalt suurem kui *täielik väärus* 0 ja

$$\begin{aligned}\mathcal{F}[A \vee \neg A] &= \max\{\mathcal{F}[A], \mathcal{F}[\neg A]\}\\ &= \max\{0{,}75\,,\ 0{,}25\}\\ &= 0{,}75\end{aligned}$$

pole ka *täiesti tõene*. Võtame *täieliku vääruse* tähistamiseks tarvitusele sümboli $\bot$ ning defineerime

$$\mathcal{F}[\bot] = 0.$$

Klassikalises loogikas arvatakse implikatsioon vääraks, kui tema eeldus on tõene ja järeldus on väär. Hägusloogikas tähendab see situatsiooni, kus eelduse tõesusaste on järelduse tõesusastmest suurem. Sellepärast on loomulik implikatsiooni tõesusastmeks pidada tõesuse kadu üleminekul tema eelduse tõesusastmelt järelduse tõesusastmele:

$$\mathcal{F}[p \supset q] = \begin{cases} 1 - (\mathcal{F}[p] - \mathcal{F}[q]), & \text{kui } \mathcal{F}[p] > \mathcal{F}[q];\\ 1, & \text{muidu.}\end{cases}$$

Iga hägusinterpretatsiooni $\mathcal{F}$ korral kehtivad siis näiteks võrdused

$$\mathcal{F}[1 \supset 0] = 1 - (1 - 0) = 0,$$
$$\mathcal{F}[0{,}75 \supset 0{,}5] = 0{,}25,$$
$$\mathcal{F}[0{,}5 \supset 0{,}75] = 1.$$

Üldisus- ja eksistentsikvantoreid võib klassikalisel juhul käsitleda vastavalt kui konjunktsiooni ja disjunktsiooni üle universumi

kõigi elementide. Hägusloogikas saame selliselt kvantorite semantikareeglid

$$\mathcal{F}[\forall x\ p(x)] = \min\{\mathcal{F}[p(c_\mathcal{F})] \mid c_\mathcal{F} \in U_\mathcal{F}\},$$
$$\mathcal{F}[\exists x\ p(x)] = \max\{\mathcal{F}[p(c_\mathcal{F})] \mid c_\mathcal{F} \in U_\mathcal{F}\}.$$

Hägusloogikas saab loogilist järeldumist defineerida mitmel viisil. Klassikalises mõttes ei tohi tuletada tõestest eeldustest väära järeldust:

$$\Gamma \models_F p \quad\Leftrightarrow\quad \text{ei leidu hägusinterpretatsiooni } \mathcal{F},\ \text{nii et } \mathcal{F}[\Gamma] = 1 \text{ ja } \mathcal{F}[p] = 0\,,$$

kus Γ on lõplik lausete hulk, mille tõesusastmeks on tema komponentlausete konjunktsiooni tõesusaste. Kui me ei luba teha tõeste eelduste põhjal mittetõest järeldust, siis saame *nõrga järeldumise* definitsiooni

$$\Gamma \models_{F2} p \quad\Leftrightarrow\quad \text{ei leidu hägusinterpretatsiooni } \mathcal{F},\ \text{nii et } \mathcal{F}[\Gamma] = 1 \text{ ja } \mathcal{F}[p] < 1.$$

Kui me aga keelame järeldamisel tõesusastme vähenemise, siis saame *tugeva järeldumise* definitsiooni

$$\Gamma \models_{F3} p \quad\Leftrightarrow\quad \text{ei leidu hägusinterpretatsiooni } \mathcal{F},\ \text{nii et } \mathcal{F}[p] < \mathcal{F}[\Gamma].$$

Saab näidata, et tugev järeldumine implitseerib nõrga järeldumise:

$$\Gamma \models_{F3} p \Rightarrow \Gamma \models_{F2} p.$$

Vastupidine väide aga ei kehti (ülesanne 14.2.3).

Nüüd on võimalik veenduda, et soriitide paradoksid ei kehti hägusloogikas. Näiteks õunte rivi paradoksi korral ei ole eeldused

Kui $\tilde{o}_i$ on roheline, siis $\tilde{o}_{i+1}$ on roheline,

enam täiesti tõesed, sest implikatsiooni järelduse tõesusaste on eelduse tõesusastmest väiksem. Mittetõestest eeldustest võib klassikalisel ja nõrgal järeldamisel teha suvalise järelduse. Soriitide paradoksid ei kehti ka tugevas mõttes, sest *modus ponens*'i reegel

ei ole tugeva järeldumise suhtes korrektne, s.t. tema järelduse tõesusaste võib olla eelduste tõesusastmetest väiksem. Selle näitamiseks valime laused A ja B, nii et $\mathcal{F}[A] = 0{,}5$ ja $\mathcal{F}[B] = 0{,}4$. Siis kehtib

$$A, A \supset B \not\models_{F3} B,$$

sest $\mathcal{F}[A \supset B] = 0{,}9$, $\min\{\mathcal{F}[A], \mathcal{F}[A \supset B]\} = 0{,}5$ ja $\mathcal{F}[B] < \mathcal{F}[A \,\&\, (A \supset B)]$. Nõrga järeldumise suhtes on aga *modus ponens* korrektne:

$$A, A \supset B \models_{F2} B.$$

Klassikaliste tuletusreeglite mittekorrektsus ongi peamiseks põhjuseks, miks seniajani on kirjeldamata hägusloogika korrektne ja täielik tuletusreeglite süsteem.

Ülesanded

1. Näidake, et
$$A \supset B \not\models_{F2,F3} \neg A \vee B.$$
2. *Tõestage, et *modus ponens*'i reegel on nõrga järeldumise suhtes korrektne.
3. Näidake, et $\Gamma \models_{F2} p \not\Rightarrow \Gamma \models_{F3} p$.
4. Näidake, et $\Gamma \models_{F2} p \Rightarrow \Gamma \models_{F} p$, kuid $\Gamma \models_{F} p \not\Rightarrow \Gamma \models_{F2} p$.

15. Mittemonotoonne loogika

Tõde tõuseb, vale vajub.
— Eesti vanasõna

Mittemonotoonne loogika tekkis 1980. aastatel. Siis hakkasid selle valdkonna vastu huvi tundma tehisintellekti esindajad, kes töötasid välja mitu selliste loogikate kirjeldamise meetodit.

15.1. McCarthy näide

Vaatleme lauseid:

- *Köögist saab külma vett.*
- *Homme tõuseb päike.*
- *Virge Naeris ei võida olümpiamängudel kuldmedalit.*
- *Selles kursuses ei räägita õllejoomise kultuurist.*

Nende lausete tõepärasus on küllalt suur. Kuid võib ette kujutada olukordi, kus majas on külm vesi kinni keeratud või Virge Naeris saab kolmikhüppes olümpiakulla.

Me ei vaatle lauseid kui vankumatuid tõdesid, vaid nad võivad osutuda vääradeks, kui me saame konkreetse olukorra kohta täpsustatud informatsiooni.

Vaatleme mittemonotoonse loogika standardnäidet lausete väärtustamise kohta mittemonotoonsetes arutlustes. Oletame, et ma ütlen:

- *Linnud lendavad.*
- *Tweety on lind.*

Kas sellest saab järeldada, et Tweety lendab? Selline järeldus on antud juhul eeldatav. Kuid Tweety võib ka mitte lennata — ta võib olla näiteks jaanalind või pingviin. Ta võib olla ka lihtsalt lennuvõimetu. Me peame kõigil nendel juhtudel oma esialgse järelduse tagasi võtma.

Järelikult ei ole vaadeldav reegel mitte

$$lind(x) \supset lendab(x),$$

vaid näiteks

$$lind(x) \,\&\, normaalne(x) \supset lendab(x),$$

s.t. kõik normaalsed linnud lendavad. See väide pannakse tavaliselt kirja *vaikimisireegli* abil:

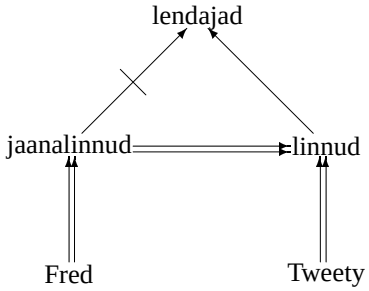
$$lind(x) \,\&\, \neg eban(x) \supset lendab(x),$$

s.t. kui lind ei ole ebanormaalne, siis ta lendab. Õige ei ole ka väide, et kõik ebanormaalsed linnud ei lenda, sest jaanalind võib näiteks lennata lennukiga. Siis on tegemist *erandi erandiga*.

Kuidas saab teha otsustuse $\neg eban(Tweety)$? Loomulikult ei saa me seda tuletada, vaid saame eeldada, et Tweety on normaalne lind, kuna puuduvad seda ümber lükkavad väited. Aga kui ilmub välja vastupidine informatsioon, siis me tühistame selle eelduse.

Me teame näiteks, et jaanalinnud ei lenda:

$$jaanalind(x) \supset lind(x) \,\&\, \neg lendab(x).$$



Joonis 15.1. Lendavate objektide skeem.

Sellest saab aga tuletada, et jaanalinnud on ebanormaalsed (ülesanne 15.1.1, lk. 276):

$$jaanalind(x) \supset eban(x).$$

Seega, kui Fred on jaanalind, siis me ei saa eeldada $\neg eban(Fred)$.

Tähistades vaikimisireegleid ühekordsete nooltega ja rangeid reegleid topeltnooltega, saame joonisel 15.1 toodud skeemi:

$$lind(x) \,\&\, \neg eban(x) \supset lendab(x),$$
$$jaanalind(x) \,\&\, \neg eban_j(x) \supset \neg lendab(x),$$
$$jaanalind(x) \supset lind(x),$$
$$Tweety(x) \supset lind(x),$$
$$Fred(x) \supset jaanalind(x).$$

Me kasutame siin erisuguseid ebanormaalsuse predikaate, sest ebanormaalsed linnud (kes ei lenda) erinevad ebanormaalsetest jaanalindudest (kes lendavad). See, et objekt on ühes mõttes ebanormaalne, ei tähenda sugugi, et ta on igas mõttes ebanormaalne.

Ülesanded

1. Näidake, et kehtib arutlus.

$$\frac{lind(x)\,\&\,\neg eban(x) \supset lendab(x)\qquad jaanalind(x) \supset lind(x)\,\&\,\neg lendab(x)}{jaanalind(x) \supset eban(x)}$$

15.2. Näite formaalne käsitlus

Miks nimetatakse niisugust arutluse tüüpi mittemonotoonseks? Klassikaline loogika on monotoonne, sest temas kehtib

$$\Delta \models p,\ \Delta \subseteq \Gamma \Rightarrow \Gamma \models p,$$

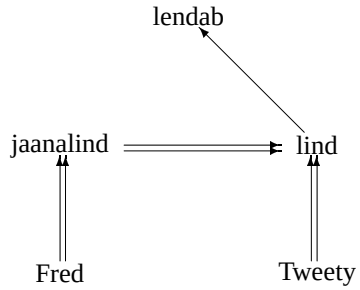
s.t. loogiliste järelduste hulk ei kahane eelduste hulga kasvamisel. Mittemonotoonsete arutluste korral see ei kehti. Kui selgub, et Tweety on jaanalind, tuleb tühistada eelnev otsustus Tweety lendamise kohta, mistõttu järelduste hulk ei kasva uute faktide teadasaamisel monotoonselt.

Ent kuidas me otsustasime, et Tweety on normaalne lind ning oskab lennata? Vastus on selline. et meil on hulk eeldusi, mida me soovime kasutada, ning me usume neist eeldustest nii paljusid kui võimalik.

Olgu T kõigi lausete hulk, mille kohta me teame, et nad on tõesed, ja A eelduste hulk, mida me soovime arutlustes vaikimisi kasutada (näiteks normaalsuse eeldused kõigi vaadeldavate objektide kohta). Me ütleme sellisel juhul, et paar (T, A) on *mittemonotoonne teooria* ja T on *baasteooria* (s.t. teooria see osa, mida me teame kindlalt).

Definitsioon 15.1. Mittemonotoonse teooria (T, A) *laiend* on A suvaline maksimaalne osahulk $E \subseteq A$, mis ei anna lausete hulgaga T vastuolu. S.t. teooria laiend on üks võimalik viis, kuidas asjad võiksid olla, eeldusel, et me oleme T faktide õigsuses veendunud.

Joonisel 15.2 on toodud lendavate objektide lihtsustatud skeem,



Joonis 15.2. Lendavate objektide lihtsustatud skeem.

kust on välja jäetud jaanalindude mittelendamist väljendav vaikimisireegel:

$$T = \begin{cases} lind(x)\,\&\,\neg eban(x) \supset lendab(x) \\ jaanalind(x) \supset lind(x)\,\&\,\neg lendab(x) \\ lind(Tweety) \\ jaanalind(Fred) \end{cases}$$

Me soovime eeldada, et Fred ja Tweety on normaalsed linnud:

$$A = \begin{cases} \neg eban(Tweety) \\ \neg eban(Fred) \end{cases}$$

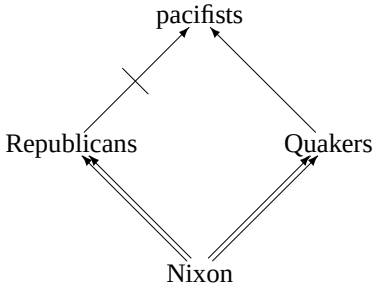
Kuid Fred on jaanalind ja on seepärast ebanormaalne. Seega kehtib $T \models eban(Fred)$ ning lause $\neg eban(Fred)$ ei saa esineda mitte üheski T mittevasturääkivas laiendis. Järelikult on mittemonotoonse teooria (T, A) laiend üheselt määratud:

$$E = \{\neg eban(Tweety)\}.$$

Kuna

$$T \cup E \models lendab(Tweety),$$

siis on $lendab(Tweety)$ antud teooria mittemonotoonne järeldus.



Joonis 15.3. Nixoni romb.

Ülesanded

1. Näidake, et
   (a) hulk E on vaadeldava mittemonotoonse teooria ainuke laiend;
   (b) $T \cup E \models lendab(Tweety)$.

Kuid laiend ei tarvitse olla ühene. Joonisel 15.3 on toodud Nixoni romb

$$T = \begin{cases} Quaker(x)\ \&\ \neg ab_r(x) \supset pacifist(x) \\ Republican(x)\ \&\ \neg ab_p(x) \supset \neg pacifist(x) \\ Quaker(Nixon) \\ Republican(Nixon)\,, \end{cases}$$

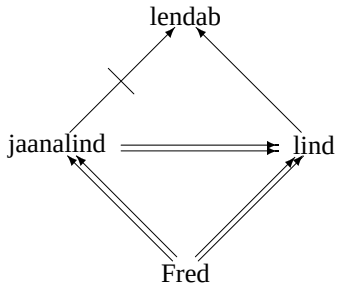
kus predikaat ab_r märgib religioosset ebanormaalsust ja ab_p poliitilist ebanormaalsust.

Me soovime eeldada, et Nixon on nii poliitiliselt kui ka religioosselt normaalne:

$$A = \{\neg ab_r(Nixon), \neg ab_p(Nixon)\}.$$

Ent

$$T \models ab_r(Nixon) \vee ab_p(Nixon),$$



Joonis 15.4. Fredi romb.

s.t. Nixon peab olema ühes mõttes ebanormaalne. Seega on teoorial (T, A) kaks mittemonotoonset laiendit:

$$E_1 = \{\neg ab_p(Nixon)\},$$
$$E_2 = \{\neg ab_r(Nixon)\},$$

kuid me ei saa kumbagi laiendit teisele eelistada, sest me ei tea, kas Nixon on patsifist või ei ole.

Ka lindude ja jaanalindude näite korral saab konstrueerida Nixoni rombi sarnase figuuri. Jätame Tweety vaatluse alt välja ja uurime, millise otsustuse saab teha Fredi kohta, eeldades, et ta on igati normaalne:

$$A = \{\neg eban(Fred), \neg eban_j(Fred)\}.$$

Fredi rombi korral annab lause

$$eban(Fred) \vee eban_j(Fred).$$

kehtivus (nagu Nixoni rombi korral) vastuolu. Antud juhul on aga võimalik eelistada ühte teooria laiendit teisele: kuna jaanalinnud moodustavad lindude alamklassi, siis tuleb eelistada jaanalindude reeglit

$$jaanalind(x)\ \&\ \neg eban_j(x) \supset \neg lendab(x).$$

Üldjuhul ei ole kasutatava reegli valik nii lihtne. Seepärast on

mitme võimaliku laiendi hulgast ühe valimise probleem praegu tähtis uurimisvaldkond.

15.3. Suletud maailm

Kogu *negatiivset informatsiooni* pole mõtet täielikult kirjeldada, sest selle hulk kasvab universumi ühtlasel suurenemisel geomeetriliselt.

Näiteks, kui me teame, et Fred on jaanalind, siis me peaksime ka toonitama, mida Fred ei ole:

$$\neg lendab(Fred),$$
$$\neg pingviin(Fred),$$
$$\neg kass(Fred),$$
$$\ldots$$

Kui me aga teame, et Jaan on Tartus, siis ta ei saa olla Tallinnas.

Eeldades, et kogu positiivne informatsioon on antud, saab *suletud maailma eelduse* (Reiter, 1978) abil teadmiste baasist TB negatiivsed faktid lihtsalt tuletada:

$$TB \nvdash p \Rightarrow TB \vdash \neg p,$$

s.t. kui p pole tõestatav, siis järeldame $\neg p$.

Kuna reeglid on harilikult implikatiivsed, siis ei saa positiivsetest faktidest negatiivset informatsiooni otse tuletada. Me ei saa näiteks tuletada väidetest

$$lind(Tweety),$$
$$pingviin(x) \supset lind(x)$$

lauset $\neg pingviin(Tweety)$, sest selle väidete hulga *Herbrandi baas* (mis koosneb antud signatuuri kõigist kinnistest elementaarvalemitest)

$$B = \{lind(Tweety), pingviin(Tweety)\},$$

on ka tema mudeliks. Lause

$$\neg pingviin(Tweety),$$

tuletamiseks peame kasutama *täielikkuse eeldusi* (Clark, 1978), mis muudavad implikatsioonid ekvivalentsideks.

Moodustame iga predikaadi jaoks *täielikkuseaksioomi*, mis väidab, et predikaat on tõene *ainult* teadmiste baasis kirjeldatud juhtudel. Täieliku teadmiste baasi saamiseks tuleb lisada veel võrduse aksioomid.

Lindude näite korral koosneb täielik teadmiste baas reeglitest

$$TB = \begin{cases} x = Tweety \supset lind(x) \\ pingviin(x) \supset lind(x) \\ lind(x) \supset x = Tweety \vee pingviin(x) \\ \neg pingviin(x)\,, \end{cases}$$

mille kaks viimast reeglit on täielikkuseaksioomid ühekohaliste predikaatide $lind$ ja $pingviin$ jaoks. Neist predikaatidest viimane on alati väär, sest $pingviin$ ei ole esialgses teadmiste baasis mitte ühegi reegli paremas pooles. Nüüd saab juba lihtsalt tuletada

$$TB \vdash \neg pingviin(Tweety).$$

Clark lisab ka *nimede unikaalsuse aksioomid*, mis tagavad, et erinevad indiviidkonstandid viitavad erinevatele objektidele. Lisades teadmiste baasi väite

$$pingviin(Opus),$$

tuleb täieliku teadmiste baasi saamiseks asendada predikaadi $pingviin$ kirjeldus väitega

$$pingviin(x) \equiv x = Opus$$

ning lisada unikaalsuse aksioom

$$Opus \neq Tweety.$$

Toome teadmiste baasi TB ka lendamise predikaadi

$$lind(x)\ \&\ \neg pingviin(x) \supset lendab(x)$$

koos täielikkuseaksioomiga

$$lendab(x) \supset lind(x) \,\&\, \neg pingviin(x).$$

Nüüd saab tuletada väited

$$TB \vdash lendab(Tweety),$$
$$TB \vdash \neg lendab(Opus).$$

Kuid ilma nimede unikaalsuse aksioomita ei oleks saanud mitte kumbagi neist lausetest tuletada.

Täielik teadmiste baas pole aga üheselt määratud, sest lendamise predikaadi täielikkuseaksioomi saab vaadelda ka kui predikaadi $pingviin$ täielikku kirjeldust:

$$lind(x) \,\&\, \neg lendab(x) \supset pingviin(x),$$
$$pingviin(x) \supset lind(x) \,\&\, \neg lendab(x).$$

Saadud täielikkuseaksioom ei ole aga eelmisega samaväärne (ülesanne 15.3.1). Täielikkuseaksioom võib olla ka vasturääkiv. Näiteks valemi

$$\begin{aligned} Pa &= Pa \vee Pa \\ &= \neg Pa \supset Pa \end{aligned}$$

täielikkuseaksioom

$$Px \equiv x = a \,\&\, \neg Pa$$

annab asenduse $\{x/a\}$ korral vastuolu $Pa \equiv \neg Pa$. Kahjuks pole täpselt teada, millal võib täielikustamist kasutada ja millal mitte.

Ülesanded

1. Näidake, et järgmised laused ei ole loogiliselt samaväärsed:

$$lendab(x) \supset lind(x) \,\&\, \neg pingviin(x),$$
$$pingviin(x) \supset lind(x) \,\&\, \neg lendab(x).$$

15.4. Piiramine

McCarthy on esitanud suletud maailma eelduse rakendamiseks terve hulga reegleid. Need reeglid on süntaktilised (nagu täielikustamine), lisades uusi aksioome, mille eesmärgiks on minimaalsete suletud maailma mudelite forsseerimine mittetäieliku teooria jaoks. Piiramistest on levinum predikaadi piiramine ja universumi piiramine.

Predikaadi piiramise näitena vaatleme maailma, kus kehtib valem

$$block(A) \vee block(B).$$

Aksioom

$$\forall x\,(block(x) \supset x = A) \vee \forall x\,(block(x) \supset x = B)$$

suleb plokkide maailma, kasutades välistavat disjunktsiooni ning minimeerides predikaadi $block$ ekstensiooni.

Universumi piiramise korral kinnitatakse, et eksisteerivad ainult antud kontekstis mainitavad indiviidid. Vaatleme näiteks lihtsat andmebaasi

Õpetaja	Õpilane	Juhendab	
Kask	*Kuusk*	*Kask*	*Mänd*
Tamm	*Mänd*	*Tamm*	*Kuusk*
Platon	*Aristoteles*	*Platon*	*Aristoteles*

Lõpliku indiviidide arvu korral saab *universumi suletuse eelduse* sõnastada indiviidide loeteluga kujul

$$\forall x\,(x = Kask \vee x = Tamm \vee \ldots \vee x = Aristoteles).$$

Nüüd saab vastata küsimusele “Kas kõigil õpilastel on õpetaja?” jaatavalt. Muidu oleksime pidanud vastama “Pole teada”.

Kui indiviidide arv pole fikseeritud, saab universumi piiramise aksioomi esitada ka väitena

$$\forall x\,(\tilde{o}petaja(x) \vee \tilde{o}pilane(x)).$$

15.5. Vaikimisi-loogika

Vaikimisi-loogikas (Reiter, 1978) püütakse minimaalsete mudelite asemel määrata meid huvitavad mudelid. Selleks lisatakse esimest järku loogikale uued tuletusreeglid — vaikimisireeglid. Erinevalt tavalistest tuletusreeglitest võivad vaikimisireeglite eeldused viidata nii tuntud faktidele kui ka kaheldavatele väidetele, andes võimaluse kasutada puuduvat negatiivset informatsiooni.

Definitsioon 15.2. *Vaikimisiteooria* on paar $\Delta = (D, W)$, kus D on vaikimisireeglite hulk ja W on suvaline esimest järku loogika valemite hulk.

Vaikimisireeglid on kujul

$$\frac{\alpha : \beta_1, \ldots, \beta_m}{\gamma},$$

kus α on eeldus, $\beta_i \quad (i = 1, \ldots, m)$ on põhjendused ja γ on järeldus.

Vaikimisireegel ütleb, et kui me teame tema eeldust ja meil pole midagi toodud põhjenduste vastu (s.t. nende eitused ei ole tuletatavad), siis võib tuletada tema järelduse.

Kui põhjendusi on üks ($m = 1$), vaadeldakse eraldi kahte juhtumit:

- Kui põhjendus on kujul $\beta = \gamma$, siis on tegemist *normaalse* vaikimisireegliga.
- Kui põhjendus on kujul $\beta = \gamma \,\&\, \omega$ (kus ω on suvaline valem), on tegemist *poolnormaalse* vaikimisireegliga.

Kirjanduses vaadeldaksegi põhiliselt normaalseid ja poolnormaalseid reegleid.

Normaalne vaikimisireegel on näiteks

$$\frac{lind(x) : lendab(x)}{lendab(x)},$$

mis ütleb, et iga lind, kelle kohta ei ole teada, et ta ei lenda, oskab lennata. Poolnormaalne vaikimisireegel on aga

$$\frac{lind(x) : lendab(x) \,\&\, \neg pingviin(x)}{lendab(x)},$$

mis väidab, et vaikimisi mittepingviinid lendavad.

Vaikimisi-loogikas saab näiteks esitada *suletud maailma reegli*:

$$\frac{: \neg p}{\neg p},$$

sest kui meil pole midagi $\neg p$ vastu (näiteks $\nvdash p$), siis võib $\neg p$ tuletada.

Vaatame jälle traditsioonilist vaikimisiteooriat

$$W = \begin{cases} pingviin(x) \supset lind(x) \\ pingviin(x) \supset \neg lendab(x) \\ jaanalind(x) \supset lind(x) \\ jaanalind(x) \supset \neg lendab(x) \\ lind(Tweety)\,, \end{cases}$$

mille vaikimisireeglite hulk on

$$D = \left\{ \frac{lind(x) : lendab(x)}{lendab(x)} \right.\!.$$

Selles teoorias saab tuletada väite

$$lendab(Tweety).$$

Hiljem võib see lause osutuda mittetuletatavaks. Näiteks kui selgub, et Tweety on pingviin. Me saame selles teoorias tuletada ka väite

$$pingviin(Opus) \vdash \neg lendab(Opus).$$

Kahjuks ei ole vaikimisi-loogika kõikide järelduste hulk rekursiivselt loetletav, s.t. ei leidu algoritmi, mis loetleks kõik antud vaikimisiteooriast järelduvad väited.

15.6. Autoepisteemiline loogika

Autoepisteemiline loogika (Moore, 1983) kasutab maailma kirjeldamiseks teadmise modaalset predikaati, mille ta omistab oma uskumiste üle arutlevale *ideaalselt mõtlevale agendile*.

Moore interpreteerib mittemonotoonses loogikas aksioomi

$$\forall x\ (lind(x)\ \&\ M(lendab(x)) \supset lendab(x))$$

mitte kui "Tüüpiline lind lendab", vaid kui "Ainsad linnud, kes ei lenda, on linnud, kelle kohta on *teada*, et nad ei lenda". Nii muutub arutluse all olev aksioom väiteks agendi (mitte aga tüüpilise indiviidi) teadmiste kohta.

Moore märgib, et vaikimisi-loogika ja autoepisteemiline loogika on erinevatel põhjustel mittemonotoonsed. Vaikimisi-loogika on katseline ja on seepärast ümberlükatav. Autoepisteemiline loogika teeb ainult *kehtivaid järeldusi* ja tema mittemonotoonsus tuleneb kontekstist, sest väidete tähendused sõltuvad agendi teadmistest: kui agent õpib midagi juurde, siis muutub ka väidete tõesus.

15.7. Ülesanded

1. *On teada, et kõik teed viivad Rooma. Tõestage, et Ülikooli tänav ei vii Rooma.

16. Lineaarloogika

Lineaarloogika on klassikalise loogika laiendus, milles püütakse formaliseerida *ressursside* liikumisega seotud probleeme. Tema autoriks on prantsuse loogik Jean-Yves Girard.

Lineaarloogika lahutab hariliku disjunktsiooni ja konjunktsiooni nö. algosadeks, kummagi kaheks eri komponendiks. Lineaarloogika on seega "eriti peeneteraline" loogika, mis annab täpsemad kirjeldusvõimalused kui harilik loogika.

16.1. Tegevuste ja situatsioonide kordumatus

Kuna klassikalise ja intuitsionistliku loogika aluseks on *samasusseadus*, siis tegeldakse nendes teooriates püsivate tõdedega. Näiteks kehtib neis arutlus

$$\frac{A,\ A \supset B}{B},$$

kus lausete A ja $A \supset B$ tõesusest saadakse lause B tõesus. Samal ajal säilib ka lause A tõesus.

Niisugune stabiilsus on hea matemaatikas, kuid ei kehti reaalsetes situatsioonides. Väitel, et A-st järeldub B, võib loomulikus keeles ja tegelikus elus olla mitu tähendust. Kõige avarama tähenduse formaliseerib klassikaline loogika:

$$A \supset B = \neg A \vee B.$$

Intuitsionistlik loogika formaliseerib järeldust pisut kitsamalt, nõudes, et A tõestusest saaks *konstrueerida* B tõestuse. Veel rangemalt käsitleb järeldust *relevantsusloogika*, mis formaliseerib järelduse *kausaalset* iseloomu: $A \supset B$ võib öelda ainult sel juhul, kui B tõestamise juures läheb tingimata vaja ka A-d. Näiteks ei ole väide

> *Kuul on lehmi, järelikult olen kolm meetrit pikk*

relevantselt tõsi isegi siis, kui Kuul lehmi pole. *Lineaarloogika* läheb relevantsusloogikast teatud mõttes sammu kaugemale: A-st B järeldamine *tarvitab* A *ära*, nii et pärast järeldusreegli rakendamist enam A-d alles pole. Füüsikas ja keemias nimetatakse sellist protsessi *reaktsiooniks*. Samas annab lineaarloogika võimaluse käsitleda järeldust soovi korral klassikaliselt: lineaarloogika annab vahendid järelduse tähendust soovi kohaselt määrata.

Kujutame näiteks ette olukorda, kus me saame sooritada kahte tegevust:

A: kulutame 5 krooni.

B: saame paki sigarette.

Tähistame tegevuse, kus me ostame 5 krooni eest paki sigarette, valemiga

$$A \multimap B.$$

Kuna me saame suitsude ostmisel oma kroonidest lahti, siis ei saa seda operatsiooni enam korrata. Loomulikult on tihti ka juhtumeid, kus reaktsiooni ei toimu või seda saab ignoreerida. Näiteks matemaatiliste abitulemuste tõesus säilib ka pärast tõestuse lõppu. Samuti on inimesi, kellel on piiramatul hulgal raha. Loetletud juhtudel on tegemist stabiilsete olukordadega, mille käsitlemisel kasutatakse spetsiaalseid seoseid ! ja ?, mida nimetatakse *eksponentideks*.

Eksponendid väljendavad tegevuse korratavust, s.t. reaktsiooni puudumist. Valem $!A$ ütleb, et me võime kulutada nii palju kroone, kui tahame. Sümbol $\multimap$ tähistab *lineaarset implikatsiooni*, s.t. implikatsiooni, mis oma eelduse ära tarvitab. Lineaarloogikas saab tavalise implikatsiooni avaldada kujul

$$A \supset B = (!A) \multimap B,$$

s.t. lausest A järeldub B parajasti siis, kui itereeritud A põhjustab B.

Kõik klassikalise ja intuitsionistliku loogika laused saab tõlkida lineaarloogika keelde. Järelikult ei kaota me lineaarloogikasse üleminekuga mitte midagi. Milline on aga saadud kasu?

16.2. Kaks konjunktsiooni ja disjunktsiooni

Lineaarloogikas kasutatakse kahte konjunktsiooni: $\otimes$ (*korda*) ja & (*koos*), mis esindavad sõna *ja* erinevaid keelelisi kasutusi:

- $\otimes$ korral sooritatakse mõlemad vaadeldavad tegevused.
- & korral sooritatakse ainult üks (meie valitud) tegevus.

Olgu võimalikud tegevused:

A: kulutame 5 krooni;

B: saame paki “Rumbat”;

C: saame paki “Marlborot”.

Tegevus $A \multimap B$ tähendab, et me kulutame viis krooni ja saame vastu ühe paki “Rumbat”. Me ei saa aga sooritada tegevust

$$A \multimap B \otimes C,$$

sest viie krooni eest ei saa osta kahte pakki sigarette, mis maksavad kokku kümme krooni. Me saame küll sooritada tegevuse

$$A \otimes A \multimap B \otimes C,$$

s.t. ostame kümne krooni eest kaks pakki sigarette. Võimalik on sooritada ka tegevus

$$A \multimap B \,\&\, C.$$

Selleks valime tegevuste B ja C hulgast ühe ning sooritame selle.

Kuigi seos & sarnaneb disjunktsiooniga, ei ole seda õige disjunktsioonina käsitleda. Näiteks valemid

$$A \,\&\, B \multimap A \qquad \text{ja} \qquad A \,\&\, B \multimap B$$

on lineaarloogikas tõesed (ülesanne 16.6.2), klassikalise disjunktsiooni korral need aga ei kehti.

Lineaarloogikas on samuti kaks disjunktsiooni:

- $\oplus$ (*pluss*) on duaalne seosega &, väljendades võimalust sooritada üks võimalikest tegevustest.
- $\parr$ (*võrdsus*) on duaalne seosega $\otimes$ ning väljendab tegevustevahelist sõltuvust.

Meie näite korral tähendab tegevus $A \multimap B \oplus C$ võimalust saada viie krooni eest pakk "Rumbat" või pakk "Marlborot"; me ei saa aga otsustada, kumma paki me saame.

Seose $A \parr B$ saab avaldada lineaarse eituse kaudu, kui

$$A^\perp \multimap B \qquad \text{või} \qquad B^\perp \multimap A.$$

16.3. Lineaarne eitus

Lineaarne eitus $\perp$ (*null*, mitte midagi) on tähtsaim lineaarne seos. Tema kaudu saab avaldada lineaarse implikatsiooni

$$A \multimap B = A^\perp \parr B$$

ning tal on ka transponeeriv toime:

$$A \multimap B = B^\perp \multimap A^\perp.$$

Lineaarloogikas (nagu klassikalises loogikas) kehtib eituse eitamise seadus:

$$A = A^{\perp\perp}.$$

Kuid erinevalt klassikalisest loogikast on lineaarloogikal ka väga lihtne konstruktiivne tähendus. Lineaarse eituse keerukas iseloom tagab De Morgani seaduste kehtivuse kõigi seoste ja kvantorite

korral:

$$(A \otimes B)^\perp = A^\perp \parr B^\perp \qquad (A \parr B)^\perp = A^\perp \otimes B^\perp$$
$$(A \,\&\, B)^\perp = A^\perp \oplus B^\perp \qquad (A \oplus B)^\perp = A^\perp \,\&\, B^\perp$$
$$(!A)^\perp = ?A^\perp \qquad (?A)^\perp = !A^\perp$$
$$(\forall x\, A)^\perp = \exists x\, A^\perp \qquad (\exists x\, A)^\perp = \forall x\, A^\perp$$

16.4. Sekventsiaalne tuletus

Toome näitena ka lineaarse lauseloogika sekventsiaalse tuletussüsteemi (Girard, 1987):

Aksioomid:

$$\vdash a, a^\perp$$

(Piisab, kui vaadelda ainult atomaarseid valemeid a.)

Korrutamisreeglid[1]*:*

$$\vdash 1 \ \text{(1-aksioom)} \qquad \frac{\vdash \Gamma}{\vdash \perp, \Gamma}\perp$$
$$\frac{\vdash A, \Gamma \quad \vdash B, \Delta}{\vdash A \otimes B, \Gamma, \Delta}\otimes \qquad \frac{\vdash A, B, \Gamma}{\vdash A \parr B, \Gamma}\parr$$

Liitmisreeglid:

$$\vdash \top, \Gamma \ \text{(aksioom, } \Gamma \text{ on suvaline)}$$
$$\frac{\vdash A, \Gamma \quad \vdash B, \Gamma}{\vdash A \,\&\, B, \Gamma}\& \qquad \frac{\vdash A, \Gamma}{\vdash A \oplus B, \Gamma}1\oplus \quad \frac{\vdash B, \Gamma}{\vdash A \oplus B, \Gamma}2\oplus$$

[1] 1 on $\otimes$ neutraalne element ja 0 on $\oplus$ neutraalne element.

Eksponentsiaalsed (modaalsed) reeglid:

$$\frac{\vdash A, \Gamma}{\vdash ?A, \Gamma}\ \text{(mahajätmine)}$$

$$\frac{\vdash \Gamma}{\vdash ?A, \Gamma}\ \text{(nõrgendamine)} \qquad \frac{\vdash A, ?\Gamma}{\vdash !A, ?\Gamma}\ \text{(!-salvestus)}$$

$$\frac{\vdash ?A, ?A, \Gamma}{\vdash ?A, \Gamma}\ \text{(koondamine)}$$

(?Γ tähistab sekventsi, mille iga elemendi ette on kirjutatud ?)

Selles süsteemis võib suvaliselt muuta sekventsi elementide järjestust. Tuleb tähele panna ka seda, et siin puudub reegel konstandi 0 jaoks.

Implikatsiooni defineerime seosega

$$A \multimap B = A^{\perp} \parr B$$

ja eituse asendusreeglitega

$$1^{\top} = \bot, \quad \bot^{\perp} = 1, \quad \top^{\perp} = 0, \quad 0^{\perp} = \top,$$
$$a^{\perp\perp} = a \ (a \text{ on atomaarne valem}),$$
$$(?A)^{\perp} = !A^{\perp}, \quad (!A)^{\perp} = ?A^{\perp},$$
$$(A \otimes B)^{\perp} = A^{\perp} \parr B^{\perp}, \quad (A \parr B)^{\perp} = A^{\perp} \otimes B^{\perp},$$
$$(A \oplus B)^{\perp} = A^{\perp} \,\&\, B^{\perp}, \quad (A \,\&\, B)^{\perp} = A^{\perp} \oplus B^{\perp}.$$

Ja lõpuks tuletusnäide:

$$\dfrac{\dfrac{\dfrac{\vdash A, A^{\perp} \quad \vdash B, B^{\perp}}{\vdash (A \otimes B), A^{\perp}, B^{\perp}}}{\vdash ((A \otimes B) \parr B^{\perp}), A^{\perp}}}{\vdash (((A \otimes B) \parr B^{\perp}) \parr A^{\perp})}$$

16.5. Lineaarloogika väljendusvõimsus

Lineaarloogikat võib vaadelda kui mingi süsteemi seisude ja üleminekute kirjeldamise vahendit. Näiteks malemängus on seisuks maleseis ja üleminekuks malekäik. Keemiliste reaktsioonide korral on seisuks molekulide loetelu ja üleminekuks on keemiline reaktsioon, mis annab esialgse molekulide loetelu põhjal uue molekulide hulga. Me saame tähistada keemilistes reaktsioonides esinevaid seise molekulide konjunktsioonidena ja seisudevahelisi üleminekuid implikatsioonidena. Vaatleme näiteks keemilist valemit

$$2H_2 + O_2 \to 2H_2O,$$

mis ütleb, et molekulid H_2, H_2 ja O_2 annavad keemilise reaktsiooni tulemusena molekulid H_2O ja H_2O. Lineaarloogikas saab selle valemi üles kirjutada kujul

$$H_2 \otimes H_2 \otimes O_2 \multimap H_2O \otimes H_2O,$$

sest keemilistes reaktsioonides tuleb kõik komponendid valida ja samas ei tohi ühtegi komponenti korduvalt kasutada.

Eksponendid muudavad lineaarloogika sama võimsaks kui klassikalise ja intuitsionistliku loogika. Lineaarloogika on neist tegelikult võimsam. Me peame olema selliste väidetega ettevaatlikud, sest kõike, mida saab üleüldse väljendada, saab väljendada ka klassikalises loogikas. Ka intuitsionistliku loogika korral saab näidata, et ta on klassikalisest loogikast võimsam. Me peame silmas seda, et lineaarloogikas saab mugavamalt kirjeldada üleminekuid, mille käsitlemine klassikalises loogikas on väga keeruline. Lineaarloogika on praegu populaarne just sellepärast, et palju abstraktseid masinaid ja süsteeme saab temas üllatavalt lihtsalt esitada.

Ent erinevalt klassikalisest ja intuitsionistlikust lausearvutusest on eksponente sisaldav lineaarne lausearvutus mittelahenduv.

16.6. Ülesanded

1. Kirjeldage lineaarloogikat kasutades, kuidas saab sõita Tartust Tallinnasse.

2. Näidake, et valemid $A \,\&\, B \multimap A$ ja $A \,\&\, B \multimap B$ on lineaarloogikas tõesed.

3. *Näidake, et klassikalised lihtsustamis- ja idempotentsusreeglid

$$A \,\&\, B \supset A,$$
$$A \supset A \,\&\, A$$

ei kehti seoste $\otimes$ ja $\multimap$ korral.

IV

Loogika rakendusi programmeerimises ja tehisintellektis

17. Automaatne teoreemitõestamine

Nagu nimigi ütleb, tegeldakse nimetatud valdkonnas meetodite otsimisega loogikateoreemide automaatseks tõestamiseks arvuti abil. Kuivõrd arvuti kasutamine on seejuures väga olulisel kohal, on automaatne teoreemitõestamine tihedalt seotud *teoreetilise arvutiteadusega*. Kuivõrd loogikateoreemidena saab formaliseerida suurt hulka mõtlemisülesandeid, on automaatne teoreemitõestamine ka *tehisintellektiteaduse* üks osa.

Ütlesime, et eesmärgiks on *loogikateoreemide* tõestamine. Nagu varasematest peatükkidest näha, saab iga konkreetse matemaatikateoreemi ja mitmesugused kombineerimist nõudvad praktilised probleemid esitada loogika keeles, s.o. loogikateoreemidena. Loogika kasutamine universaalse baaskeelena ei kitsenda teoreemitõestajate rakendussfääri, vaid muudab selle, vastupidi, nii laiaks kui vähegi võimalik.

Teatavasti on mitmesuguseid, eri eesmärkideks sobivaid loogikasüsteeme. Automaatse teoreemitõestamise jaoks pakuvad nad kõik huvi. Enamik pingutusi on siiamaani suunatud klassikalise predikaatarvutuse teoreemide lahendamisele — viimatinimetatud loogikasüsteem on kasutatavaim ja üldiselt peetakse seda kõige olulisemaks süsteemiks olemasolevatest.

17.1. Eesmärgid

Automaatse teoreemitõestamise otsene eesmärk on luua sellised arvutiprogrammid ja arvutid, mis suudaksid keerulisi, täpset mõtlemist nõudvaid probleeme mõistliku aja jooksul automaatselt lahendada. Inimesest kasutaja annab arvutile täpselt formaliseeritud probleemi (näiteks uue matemaatikateoreemi, programmeerimisülesande, elektroonikasüsteemi korrektsuse kontrolli, inseneriülesande või andmebaasi, millest järeldusi/lahendusi otsida) ning soovib, et arvuti selle probleemi mõistliku aja jooksul lahendaks.

Kriitilise tähtsuse omandab fraas *mõistliku aja jooksul*: kui mingil loogika keeles kirjeldatud probleemil on üldse lahendus, siis teoreetiliselt suudab ka üpris lihtne arvutiprogramm sellise probleemi alati lahendada. Paraku kulub taolisel lihtsal programmil peaaegu iga probleemi lahendamiseks lootusetult kaua aega: enne jõuab päike kustuda ja maailma lõpp kätte jõuda, kui arvuti vastuse leiab.

Praktiliselt kasutuskõlblik automaatse teoreemitõestamise süsteem peaks suutma lahendada huvipakkuvaid probleeme *mõistliku aja*, s.o. paari sekundi, tunni või isegi päeva ja kuu jooksul, olenevalt probleemi tähtsusest.

Paraku suudavad olemasolevad automaatse tõestamise süsteemid lahendada mõistliku aja jooksul ainult üksikuid tõeliselt keerulisi ja praktiliselt olulisi probleeme; rõhuva enamiku raskete probleemide puhul pole neist süsteemidest üksi suuremat kasu. Küll aga saab neid kasutada interaktiivselt: kasutaja annab ette suunised ning leiab ise tõestuse kõige keerulisemad osad, jättes lihtsamad, kuid töömahukad alamülesanded arvutile.

Sestap on automaatse teoreemitõestamise ala põhieesmärgiks suurendada automaatsete teoreemitõestajate kiirust. Tõestajate kiiruse suurendamise nimel tehakse tööd kolmes peasuunas:

- Otsingumeetodite väljatöötamine, mis garanteeriksid vanade meetoditega võrreldes suurema otsingukiiruse iga võimaliku ülesande jaoks.

  Siia kuulub paremate insenerilahenduste leidmine nii teoreemitõestajate tarkvara kirjutamisel kui ka riistvara, s.o. arvutite rakendamisel. Riistvara rakendamisel on aktuaalsemaks probleemiks tõestuseotsingu efektiivne jaotamine paljude erinevate, kuid samaaegselt töötavate protsessorite või arvutite vahel.

- Otsingustrateegiate väljatöötamine, mis annavad suurema kiiruse kas enamikul praktilist huvi pakkuvatel juhtudel või siis spetsiaalsete ülesannete puhul.

Peale tõestajate kiiruse suurendamise on teine oluline eesmärk hõlbustada automaatse tõestaja ja inimesest kasutaja vahelist suhtlemist. Soovitatavalt peaks tõestaja kasutajal olema võimalik anda tõestajale näpunäiteid ehk soovitusi, kuidas probleemi tõestust kergemini leida. Kasutaja peaks saama soovi korral osa tõestuse sammudest ise leida, suunates automaatse tõestaja seni lahendamata jäänud alamprobleemide juurde. Samas peaks kasutaja saama informatsiooni lõpetamata jäänud tõestuseotsingutest, suutmaks hinnata, kuidas tõestuseotsingu strateegiat järgmisel katsel muuta.

Mitmed olemasolevad süsteemid võimaldavadki taolist inimkasutaja ja automaatse tõestaja koostööd, kuid mugavusest, lihtsusest ja koostöö sujuvusest on asi veel küllalt kaugel.

17.2. Sissejuhatus resolutsioonimeetodisse

Automaatse teoreemitõestamise jaoks on välja töötatud hulk üpris erinevaid meetodeid. Nende hulgast on üks efektiivsemaid ja kindlasti kõige levinum *resolutsioonimeetod*, mille 1965. aastal esitas J. Robinson. Samaaegselt, kuid Robinsonist sõltumatult leiutas sarnaste põhiprintsiipidega *pöördmeetodi* S. Maslov. 1965. aastal oli resolutsioonimeetod tõeline murrang automaatses teoreemitõestamises, võimaldades suhteliselt väikese ajakuluga tõestada palju teoreeme, mis olid varasemate meetodite, näiteks Davisi-Putnami meetod (üks viimase leiutajatest, H. Putnam, sai hiljem kuul-

saks funktsionalistliku filosoofiasuuna rajajana), jaoks lootusetult väljaspool haardeulatust.

Üldise resolutsioonimeetodi jaoks on välja töötatud suur hulk üksteisest oluliselt erinevaid otsingustrateegiaid. Mitme üksteisest erineva modifikatsiooni tõttu on tegu väga paindliku meetodiga: enamikku praktikas kasutatavaid, näiliselt resolutsioonimeetodist erinevaid tõestuseotsingu-algoritme on võimalik esitada kui resolutsioonimeetodi spetsiaalseid strateegiaid. Tuntud *loogilise programmeerimise keel* Prolog baseerub ühel niisugusel resolutsioonimeetodi strateegial, mis sobib küll hästi loogilise programmeerimise, kuid halvasti automaatse teoreemitõestamise jaoks.

Resolutsioonimeetod on iseenesest väga lihtne tõestusmeetod: meetodi elementaarseima variandi juures kasutatakse ainult kahte tuletusreeglit, *resolutsioonireeglit* ja *faktoriseerimisreeglit*. Enne tõestuse alustamist teisendatakse tõestatav teoreem võimalikult lihtsale kujule, disjunktide hulgaks. *Disjunktiks* nimetatakse sellist disjunktsiooni, mis koosneb ainult positiivsetest ja negatiivsetest elementaarvalemitest. Taolisi eitusega või ilma selleta elementaarvalemeid nimetatakse *literaalideks*. Kõik disjunktis esinevad muutujad on vaikimisi seotud üldisuskvantoriga $\forall$. Järgnevas kasutame muutujatena tähti `x,y,z,u,v,w` — ülejäänud sümbolid on kas konstandid, funktsionaal- või predikaatsümbolid. Mõned disjunktide näited:

- Ühestainsast literaalist koosnev disjunkt, mis väljendab, et Jaan on Martini isa: `isa(Jaan,Martin)`.
- Ühestainsast literaalist koosnev disjunkt, mis väljendab, et Riivo **ei ole** Veiko isa: ¬ `isa(Riivo,Veiko)`.
- Kahest literaalist koosnev disjunkt, mis väljendab, et Veiko isa on kas Martin või Riivo: `isa(Martin,Veiko)` $\vee$ `isa(Riivo,Veiko)`.
- Teatavasti on järeldust esitav lause $A \Rightarrow B$ ekvivalentne lausega $\neg A \vee B$. Järgnev kahest literaalist koosnev disjunkt väljendab, et juhul kui Martin on Veiko isa, siis on Jaan Veiko vanaisa: ¬ `isa(Martin,Veiko)` $\vee$ `vanaisa(Jaan,Veiko)`.
- Järgnev muutujaid `x` ja `y` sisaldav disjunkt ütleb, et emad on alati naissoost: ¬ `ema(x,y)` $\vee$ `naissoost(x)`.

  Muutujaid sisaldavat disjunkti on kohane mõista järgmiselt: disjunkt kehtib, ükskõik mis objektid ka muutujate asemele panna. Viimasest disjunktist saame muutuja `x` konstandiga `Jaan` ja muutuja `y` konstandiga `Martin` asendades järgmise kehtiva disjunkti: ¬ `ema(Jaan,Martin)` $\vee$ `naissoost(Jaan)`.
- Järgnev muutujaid `x`, `y` ja `z` sisaldav disjunkt ütleb, et kui `x` on `y` isa ja `y` on `z` isa, siis `x` on `z` vanaisa: ¬ `isa(x,y)` $\vee$ ¬ `isa(y,z)` $\vee$ `vanaisa(x,z)`. Tõepoolest, lause $(A \,\&\, B) \Rightarrow C$ on ekvivalentne lausega $\neg A \vee \neg B \vee C$.
- Eelmine disjunkt ei olnud vanaisa-suhte defineerimiseks siiski piisav. Puudujäänud emapoolset liini pidi vanaisaks olemise määrab disjunkt ¬ `isa(x,y)` $\vee$ ¬ `ema(y,z)` $\vee$ `vanaisa(x,z)`. Reeglid, mis harilikus predikaatarvutuses on esitatavad üheainsa valemiga, ei pruugi olla esitatavad üheainsa disjunktiga: sageli läheb tarvis mitut disjunkti.

Resolutsioonimeetodi põhiprintsiip on fakte ja reegleid esitavate etteantud disjunktide hulgast uute disjunktide tuletamine. Disjunktide hulka käsitletakse kui disjunktide konjunktsiooni. Resolutsioonimeetod on *täielik* meetod: kui etteantud disjunktide hulgast järeldub loogiliselt mingi disjunkt D, siis resolutsioonimeetod tuletab alati kas sellesama disjunkti D või üldisema disjunkti D', nii et D on D'-i erijuht. Erijuhu täpse definitsiooni anname hiljem.

17.2.1. Lihtsaim juht: muutujateta disjunktid

Kui disjunktides muutujaid ei ole, s.t. tegu on sisuliselt lausearvutusega, siis on *resolutsioonireeglit* üpris lihtne esitada. Kahest

disjunktist $A_1 \vee A_2 \vee \ldots \vee A_n$ ja $\neg A_1 \vee B_2 \vee \ldots \vee B_m$, millest esimene sisaldab positiivset literaali A_1 ja teine selle eitusega varianti $\neg A_1$, saame tuletada uue disjunkti, pannes mõlemad kokku ja lõigates välja literaalid A_1 ning $\neg A_1$:

$$\frac{A_1 \vee A_2 \vee \ldots \vee A_n \quad \neg A_1 \vee B_2 \vee \ldots \vee B_m}{A_2 \vee \ldots \vee A_n \vee B_2 \vee \ldots \vee B_m}$$

Seejuures eeldame, et literaalide järjekord disjunktis ei ole oluline ning seda võib vabalt muuta. Äralõigatavad literaalid ei pruugi seega olla disjunktides esimesel positsioonil.

Resolutsioonireeglit võib mõista kui klassikalise *modus ponens*'i

$$\frac{A \quad A \Rightarrow B}{B}$$

üldistust. Tõepoolest, *modus ponens*'i saab esitada disjunktidena kui

$$\frac{A \quad \neg A \vee B}{B}$$

ning resolutsioonireegel lisab *modus ponens*'i kahele eeldusele nö. ballastina literaale, mis lähevad muutumatuna edasi *modus ponens*'i järeldusse.

Muutujateta disjunktide korral ütleb resolutsioonimeetodi teine reegel, *faktoriseerimisreegel*, et kui disjunktis on kaks ühesugust literaali, siis võib ühe neist kõrvaldada:

$$\frac{A_1 \vee A_1 \vee A_2 \vee \ldots \vee A_n}{A_1 \vee A_2 \vee \ldots \vee A_n}.$$

Erinevalt resolutsioonireeglist, mis tuletab uusi disjunkte, jättes eelduseks olnud alles, kasutame faktoriseerimisreeglit *lihtsustava* reeglina: kui faktoriseerimisreegli rakendamisel disjunktile C saame uue disjunkti C', siis kustutame eelduse C, s.t. ei kasuta teda enam kunagi uute disjunktide tuletamise juures.

Toome siinkohal kaks resolutsioonimeetodi rakendamise näidet. Olgu meil antud järgmised muutujateta disjunktid 1–4:

(1) `isa(Jaan,Martin)`,

(2) $\neg$ `isa(Riivo,Veiko)`,

(3) `isa(Martin,Veiko)` $\vee$ `isa(Riivo,Veiko)`,

(4) $\neg$ `isa(Martin,Veiko)` $\vee$ `vanaisa(Jaan,Veiko)`.

Resolutsioonimeetod võimaldab disjunktidest 1–4 tuletada disjunkti `vanaisa(Jaan,Veiko)`, seejuures ei kasuta me üldse disjunkti 1:

(2 ja 3 annavad disjunkti 5) `isa(Martin,Veiko)`,

(4 ja 5 annavad disjunkti 6) `vanaisa(Jaan,Veiko)`.

Nagu öeldud, on resolutsioonimeetodi sisuks algsetest ja juba tuletatud disjunktidest üha uute disjunktide tuletamine, kasutades seejuures resolutsiooni- ja faktoriseerimisreegleid. Tuletatud disjunktid saab seega jaotada tasemete vahel. Esimese taseme moodustavad algselt antud disjunktid. Teise taseme moodustavad algselt antud disjunktidest tuletatud disjunktid. Kolmanda taseme moodustavad esimese ja teise taseme disjunktidest tuletatud disjunktid jne.

Definitsioon 17.1. Olgu S disjunktide hulk. Tähistagu $Res\_Fakt(S)$ kõigi disjunktide hulka, mida saab tuletada hulgast S resolutsioonireegli ja faktoriseerimisreegli ühekordse rakendamisega. Defineerime nüüd:

$$T^1(S) = S,$$

$$T^{i+1}(S) = T^i(S) \cup Res\_Fakt(T^i(S)),$$

$$T^*(S) = \bigcup_i T^i(S).$$

Olgu meil antud järgmised disjunktid 1–4:

(1) A,

(2) ¬ A ∨ B,

(3) ¬ B ∨ C,

(4) ¬ C.

Vaatame, milliseid disjunkte viimastest resolutsioonimeetodiga üldse tuletada saab. Selleks esitame resolutsioonimeetodi töö sammhaaval. Iga tuletussamm koosneb ühest resolutsioonireegli ja kui võimalik, siis ka faktoriseerimisreegli rakendamisest.

Teine tase:

(1 ja 2 annavad disjunkti 5) B,

(2 ja 3 annavad disjunkti 6) ¬ A ∨ C,

(3 ja 4 annavad disjunkti 7) ¬ B.

Kolmas tase:

(5 ja 3 annavad disjunkti 8) C,

(5 ja 7 annavad disjunkti 9) tühi disjunkt ehk vastuolu,

(6 ja 1 annavad disjunkti 8 koopia) C,

(6 ja 4 annavad disjunkti 10) ¬ A,

(7 ja 2 annavad disjunkti 10 koopia) ¬ A,

(7 ja 5 annavad disjunkti 9 koopia) tühi disjunkt ehk vastuolu.

Neljas tase:

(8 ja 4 annavad disjunkti 9 koopia) tühi disjunkt ehk vastuolu.

(10 ja 1 annavad disjunkti 9 koopia) tühi disjunkt ehk vastuolu.

Kuna viimasel, neljandal tasemel on kõik disjunktid varasemate disjunktide koopiad, siis ei ole resolutsioonimeetodiga võimalik enam ühtegi disjunkti lisaks tuletada.

Vaadeldud näide on resolutsioonimeetodi kasutamise juures tüüpiline: enamasti rakendatakse resolutsioonimeetodit disjunktide hulga *vastuolulisuse* tõestamiseks. Kui vastuolu on leitud, siis peatatakse töö kohe, kuna mingit lisainformatsiooni edasine tuletamine ei anna. Meie näites tuleks töö peatada kohe pärast disjunkti 9 tuletamist.

Vastuolu otsimise motiiviks on asjaolu, et mistahes valemi F tuletatavus on ekvivalentne tema eituse $\neg F$ vastuolulisusega. Uurime näiteks, kas mingist disjunktide hulgast Δ järeldub mingi disjunkt D. Valem $\Delta \Rightarrow D$ kehtib parajasti siis, kui $\Delta \,\&\, \neg D$ on vastuoluline, s.t. disjunktide $\Delta \,\&\, \neg D$ hulgast saab tuletada tühja disjunkti. Meenutame, et disjunktide hulka $\{C_1, C_2, \ldots, C_n\}$ võib vaadelda disjunktide konjunktsioonina $C_1 \,\&\, C_2 \,\&\, \ldots \,\&\, C_n$.

Ühes äsjatoodud näites tuletasime disjunktidest 1–4 disjunkti `vanaisa(Jaan,Veiko)`. Oletame, et meid huvitas algul nimelt küsimus, kas `vanaisa(Jaan,Veiko)` on nimetatud disjunktidest 1–4 tuletatav. Seda küsimust saab formuleerida järgmiselt: kas disjunktide hulk 1–4 pluss ¬ `vanaisa(Jaan,Veiko)` on vastuoluline, s.t. kas neist saab tuletada tühja disjunkti? Tõepoolest:

(1) `isa(Jaan,Martin)`,

(2) ¬ `isa(Riivo,Veiko)`,

(3) `isa(Martin,Veiko)` ∨ `isa(Riivo,Veiko)`,

(4) ¬ `isa(Martin,Veiko)` ∨ `vanaisa(Jaan,Veiko)`,

(5) ¬ `vanaisa(Jaan,Veiko)`.

Tuletus:

(2 ja 3 annavad disjunkti 6) `isa(Martin,Veiko)`,

(4 ja 6 annavad disjunkti 7) `vanaisa(Jaan,Veiko)`,

(7 ja 5 annavad tühja disjunkti) vastuolu.

Tühja disjunkti saab samadest disjunktidest 1–5 tuletada ka teisel viisil, alustades otsitava disjunkti eitusest (5) ning liikudes nö. tagasisuunas:

(5 ja 4 annavad disjunkti 6) ¬ `isa(Martin,Veiko)`,

(6 ja 3 annavad disjunkti 7) `isa(Riivo,Veiko)`,

(7 ja 2 annavad tühja disjunkti) vastuolu.

Vaatame, milliseid disjunkte viimasest hulgast 1–5 resolutsioonimeetodiga üldse tuletada saab. Lisaks näites toodud disjunktidele saab tuletada veel ainult ühe:

(3 ja 4 annavad 8) `isa(Riivo,Veiko)` ∨
`vanaisa(Jaan,Veiko)`

Tõestame nüüd muutujateta resolutsioonimeetodi korrektsuse ja täielikkuse, ning näitame seejärel, et muutujateta disjunktide jaoks on resolutsioonimeetod *lahendav algoritm*.

Definitsioon 17.2. Kui $D \in T^*(S)$, siis ütleme, et D on disjunktide S hulgast resolutsioonimeetodiga *tuletatav*.

Teoreem 17.3 (muutujateta resolutsioonimeetodi korrektsus). *Kui disjunkt D on disjunktide hulgast S resolutsioonimeetodiga tuletatav, siis kehtib valem $S \Rightarrow D$.*

Tõestus. Lihtne induktsioon tuletuse sügavuse järgi.

Induktsiooni baasjuht on olukord, kus $D \in S$. Sel juhul kehtib $S \Rightarrow D$.

Induktsiooni sammu jaoks veendume esiteks, et kui mingi disjunkt C on tuletatud ühe resolutsioonireegli rakendusega disjunktidele C_1 ja C_2, siis kehtib lause $(C_1 \,\&\, C_2) \Rightarrow C$. Teiseks veendume, et kui disjunkt C on tuletatud ühe faktoriseerimisreegli rakendusega disjunktile C_1, siis kehtib lause $C_1 \Rightarrow C$. □

Täielikkuse tõestame tühja disjunkti ehk vastuolu jaoks. Arvestame seejuures, et disjunktide hulk on interpreteeritav disjunktide konjunktsioonina.

On mitu resolutsioonimeetodi täielikkuse tõestust. J. Robinsoni esimene tõestus [Robinson 65] kasutas semantilisi puid. Siin esitame R. Boyeri tõestuse [Boyer 71], mis kasutab induktsiooni mõõdul k.

Definitsioon 17.4. Olgu S disjunktide hulk. Defineerime $k(S)$ kui kõigi hulgas S esinevate literaalide arvu ja disjunktide arvu vahe, kusjuures ühe literaali mitmesugused esinemised hulgas S loetakse eri literaalideks.

Näide: $k(\{\ \mathtt{A} \vee \neg\ \mathtt{B},\ \neg\ \mathtt{A},\ \mathtt{A} \vee \mathtt{C}\}) = 5 - 3 = 2$.

Teoreem 17.5 (muutujateta resolutsioonimeetodi täielikkus). *Kui muutujateta disjunktide hulgal S ei ole ühtegi mudelit (s.t. S on vastuoluline ehk loogiliselt väär), siis on tühi disjunkt hulgast S resolutsioonimeetodiga tuletatav.*

Tõestus. Induktsioon $k(S)$ järgi.

Baasjuht: $k(S) = 0$. Sel juhul võrdub literaalide esinemiste arv hulgas S disjunktide esinemiste arvuga hulgas S, mistõttu S koosneb üheelemendilistest disjunktidest. Sellisel hulgal S ei ole mudelit ainult sel juhul, kui mingi literaali A jaoks kehtib nii $A \in S$ kui $\neg A \in S$. Sel juhul saab tuletada tühja disjunkti hulgast S ühekordse resolutsioonireegli rakendamisega.

Induktsiooni samm: eeldame, et $0 < k(S)$ ja teoreem kehtib kõigi selliste disjunktihulkade R korral, mille jaoks $k(R) < k(S)$. Eelduse $0 < k(S)$ tõttu peab hulgas S olema vähemalt üks mitte-üheelemendiline disjunkt. Olgu selleks disjunkt D: $L \vee A_1 \vee \ldots \vee A_n$, kus $0 < n$. Olgu Δ saadud disjunktist D literaali $\mathtt{L}$ eemaldamise teel: $A_1 \vee \ldots \vee A_n$. Olgu S_1 disjunktide hulk $(S - D) \cup \{\Delta\}$ ja S_2 disjunktide hulk $(S - D) \cup \{\mathtt{L}\}$. Iga hulga S_1 või S_2 mudel on ka hulga S mudel. Järelikult, kuna teoreemi eelduse kohaselt hulgal S mudelit ei ole, pole mudelit ka hulkadel

S_1 ja S_2. On lihtne näha, et $k(S_1) < k(S)$ ja $k(S_2) < k(S)$. Seega saab induktsiooni eelduse kohaselt tühja disjunkti tuletada nii hulgast S_1 kui S_2. Olgu T_2 tühja disjunkti tuletus hulgast S_2. Vaatleme kahte juhtu.

Esimene juht: kui tuletuses T_2 ei ole disjunkti L, siis on T_2 samas ka tühja disjunkti tuletuseks hulgast S, mida oligi vaja tõestada.

Teine juht: kui tuletuses T_2 on disjunkt L, teisendame tuletuse T_2 uueks tuletuseks T_2', asendades disjunkti L iga esinemise disjunktiga D, s.o. $L \vee \Delta$, ja teostades kõik muudetud resolutsioonisammud uuesti. T_2' on korrektne tuletus hulgast S, mis lõpeb tühja disjunkti asemel kas disjunktiga Δ või $\Delta_1 \vee \ldots \vee \Delta_m$. Viimasel juhul annab m-kordne faktoriseerimisreegli rakendamine tulemuseks Δ. Seega saab disjunktide hulgast S tuletada disjunkti Δ. Kuivõrd hulgast S_1, s.o. $(S - D) \cup \{\Delta\}$, saab eelduse kohaselt tuletada tühja disjunkti ning kehtib $(S - D) \subset S$, siis on tühi disjunkt tuletatav ka hulgast S, mida oligi vaja tõestada. □

Teoreem 17.6. *Muutujateta disjunktihulkade jaoks on resolutsioonimeetod* lahendav algoritm, *s.t. iga uuritava disjunktihulga jaoks peatub mingil hetkel meetodi töö, kusjuures vastuolu leitakse siis ja ainult siis, kui disjunktihulgal ei ole ühtegi mudelit.*

Tõestus. Muutujateta resolutsioonimeetodi täielikkuse teoreemi järgi tuletab resolutsioonimeetod tühja disjunkti igast disjunktihulgast, millel pole mudelit. Jääb üle vaadata olukorda, kus uuritav disjunktihulk S ei ole vastuoluline, s.t. mudel on olemas. Resolutsioonimeetodi korrektsuse teoreemist järeldub, et vastuolu sel juhul ei tuletata. Hulk S sisaldab N literaali. On lihtne veenduda, et tuletatud disjunktid koosnevad ainult hulgas S sisalduvatest literaalidest. N erinevast literaalist on võimalik konstrueerida 2^N erinevat disjunkti, seega on tuletatavate disjunktide hulk lõplik. □

Ülesanded

1. *(R. Kowalski) Esitage laused disjunktidena, kasutades ainult konstante *kass* ja *mina*:

   - Lindudele meeldivad ussikesed.
   - Kassidele meeldivad kalad.
   - Sõbrad meeldivad teineteisele.
   - Minu kass on minu sõber.
   - Minu kass sööb kõike, mis talle meeldib.

   Mida me saame nendest lausetest järeldada selle kohta, mida minu kass sööb?

2. *(R. Kowalski) Oletame, et ma usun järgmisi fakte:

   - Leidub draakon.
   - Draakon kas magab oma koopas või kütib metsas.
   - Kui draakon on näljane, siis ta ei saa magada.
   - Kui draakon on väsinud, siis ta ei saa küttida.

   Leidke resolutsiooni abil vastused järgmistele küsimustele. Mida teeb draakon siis, kui ta on näljane? Mida teeb draakon siis, kui ta on väsinud? Mida teeb draakon siis, kui ta on näljane ja väsinud? Küsimustele vastamiseks kasutage eeldust

   - *Kui x ei saa teha toimingut y, siis x ei tee y.*

17.2.2. Üldjuht: muutujaid sisaldavad disjunktid

Järgnevalt vaatame resolutsioonimeetodi rakendamist juhul, kui disjunktid sisaldavad muutujaid. Olgu meil antud andmebaas nimega D:

(1) `isa(Jaan,Peeter)`,

(2) `isa(Jaan,Martin)`,

(3) `isa(Martin,Veiko)`,

(4) `isa(Riivo,Leo)`,

(5) `ema(Leena,Leo)` $\vee$ `isa(Leena,Leo)`,

(6) $\neg$ `isa(Leena,Leo)`,

(7) ¬ isa(x,y) ∨ ¬ isa(y,z) ∨ vanaisa(x,z),

(8) ¬ isa(x,y) ∨ ¬ ema(y,z) ∨ vanaisa(x,z).

Püüame kõigepealt tõestada, et andmebaasist järeldub väide A: vanaisa(Jaan,Veiko). Nagu varem märgitud, kehtib $D \Rightarrow A$ parajasti siis, kui $D \,\&\, \neg A$ on vastuoluline. Moodustame uue disjunktide hulga D', lisades andmebaasile D väite A eituse

(9) ¬ vanaisa(Jaan,Veiko)

ja püüame saadud D'-st tuletada vastuolu ehk tühja disjunkti.

Kuidas kasutada muutujaid sisaldavaid vanaisa-reegleid? Nagu varem öeldud, ei tähista suvaline muutujaid sisaldav väide V midagi muud, kui *V kõigi konkretiseeritud väidete hulka*, kus me *konkretiseerimise* all mõistame V kõigi muutujate asendamist kas konstantide või keerukamate muutujaid mittesisaldavate objektidega. Selline kõigi konkretiseeritud väidete hulk on kahtlemata lõpmatu. Väite

¬ isa(x,y) ∨ ¬ isa(y,z) ∨ vanaisa(x,z)

konkretisatsioonide hulka kuuluvad näiteks

¬ isa(Leo,Leo) ∨ ¬ isa(Leo,Leo) ∨
vanaisa(Leo,Leo),

¬ isa(Jaan,Martin) ∨ ¬ isa(Martin,Leo) ∨
vanaisa(Jaan,Leo),

¬ isa(Martin,Jaan) ∨ ¬ isa(Jaan,Leo) ∨
vanaisa(Martin,Leo)

jne. jne., millest enamik ei aita meid kuidagi väite vanaisa(Jaan,Veiko) tuletamisel. Kuidas leida sobiv konkretiseering? Lihtsaim, kuid väga ebaefektiivne meetod on kõigi konkretiseeringute järjestikune proovimine, lootuses kunagi sobiva peale sattuda. Resolutsioonimeetodi peamine idee ongi sellise kõigi konkretiseeringute proovimise asendamine efektiivsema meetodiga.

Meenutame varem esitatud muutujateta resolutsioonireeglit:

$$\frac{A_1 \vee A_2 \vee \ldots \vee A_n \quad \neg A_1 \vee B_2 \vee \ldots \vee B_m}{A_2 \vee \ldots \vee A_n \vee B_2 \vee \ldots \vee B_m}$$

Järgnevas eeldame, et reegli eelduses esinevad disjunktid võivad sisaldada muutujaid. Modifitseerime resolutsioonireeglit niimoodi, et literaal A_1 vasakpoolses eelduses ja ja literaal A_1 parempoolses eelduses ei ole enam täpselt ühesugused: olgu parempoolses eelduses A_1 asemel literaal A_1'.

Et resolutsioonireegel oleks korrektne, peame nüüd nõudma, et (muutujaid sisaldavaid) literaale A_1 ja A_1' saab samasugusteks konkretiseerida ehk *unifitseerida*. S.t. asendades reegli vasakpoolses disjunktis osa muutujaid uute objektidega, saame literaali A_1 konkretiseeritud variandi, mille tähistame kui $A_1\sigma$. Oluline on, et saaksime ka parempoolse disjunkti muutujaid uute objektidega asendades algselt A_1-st erineva literaali A_1' konkretiseerida selliseks literaaliks $A_1'\tau$, et $A_1'\tau$ ja $A_1\sigma$ on täpselt ühesugused.

Muutujateta resolutsioonireegli modifitseerimise idee on järgmine: ei proovita läbi mitte kõiki võimalikke konkretiseeringuid, vaid ainult neid, mis võimaldavad rakendada resolutsioonireeglit. Viimases näites ei ole mingit mõtet katsetada disjunkti 7 konkretisatsiooni ¬ isa(Leo,Leo) ∨ ¬ isa(Leo,Leo) ∨ vanaisa(Leo,Leo), kuna mingeid uusi resolutsioonireegli rakendamisvõimalusi see ei anna.

Jätkame näite uurimist. Tahame rakendada resolutsioonireeglit disjunktidele 2 ja 7, s.o. isa(Jaan,Martin) ja ¬ isa(x,y) ∨ ¬ isa(y,z) ∨ vanaisa(x,z). Selleks tuleb nende esimesed literaalid unifitseerida. Esimest disjunkti enam kuidagi konkretiseerida ei saa, teise disjunkti saame aga konkretiseerida vajalikule kujule: ¬ isa(Jaan,Martin) ∨ ¬ isa(Martin,z) ∨ vanaisa(Jaan,z). Selleks asendasime muutuja x konstandiga Jaan ning muutuja y konstandiga Martin.

Taolisi muutujate asendusi nimetatakse *substitutsioonideks*, ning neid pannakse kirja järgmisel viisil: $\{x_1/t_1, \ldots, x_n/t_n\}$, kus

iga x_i on muutuja ning iga t_i on nimetatud x_i asemele substitueeritav objekt.

Oma näites rakendasime disjunktile substitutsiooni `{x/Jaan, y/Martin}`. Pärast substitutsiooni rakendamist saame kasutada muutujateta resolutsioonireeglit, mis annab uue disjunkti `¬ isa(Martin,z) ∨ vanaisa(Jaan,z)`. Unifitseerivat substitutsiooni (meie näites `{x/Jaan, y/Martin}`) nimetatakse *unifitseerijaks*.

Sooviksime järgnevalt rakendada resolutsioonireeglit, võttes üheks eelduseks viimati tuletatud disjunkti ja teiseks eelduseks disjunkti 3: `isa(Martin,Veiko)`. Selleks, et literaalid `¬ isa(Martin,z)` ja `isa(Martin,Veiko)` unifitseerida, rakendame esimesele eeldusele substitutsiooni `{z/Veiko}`, mis annab disjunkti `¬ isa(Martin,Veiko) ∨ vanaisa(Jaan,Veiko)`. Resolutsioonireegliga tuletame nüüd disjunkti `vanaisa(Jaan,Veiko)`, millest leiame disjunkti 9 abil järgmisel sammul vastuolu. Tuletuskäik oli kokkuvõtlikult järgmine:

(2 ja 7 annavad disjunkti 10) `¬ isa(Martin,z) ∨ vanaisa(Jaan,z)`,

(10 ja 3 annavad disjunkti 11) `vanaisa(Jaan,Veiko)`,

(11 ja 9 annavad tühja disjunkti) vastuolu.

Siiani jäi lahtiseks küsimus, kuidas leida vajalikku unifitseerivat substitutsiooni ja milline see peaks täpselt olema. Eriti keeruliseks muutub olukord siis, kui *mõlemad* unifitseeritavad literaalid sisaldavad muutujaid. Olgu meil tegu kahe niisuguse disjunktiga, kus `x, y, u` ja `v` on muutujad:

`P(a,x,y) ∨ S(x,y)` ja `¬ P(u,v,b) ∨ R(u,v)`.

Esimest kaht aatomit `P(a,x,y)` ja `P(u,v,b)` saab unifitseerida mitmel viisil, näiteks konkretiseerides mõlemad kas literaaliks `P(a,a,b)` või literaaliks `P(a,b,b)`. Resolutsioonimeetodi idee on rakendada *minimaalseid* substitutsioone, jättes alles kõik muutujad, millesse otseselt substitueerida vaja pole. Resolutsioonimeetod

rakendab seega substitutsiooni `{u/a, y/b, x/v}`, kus komponent `{x/v}` lihtsalt nimetab muutuja ümber. Mainitud substitutsiooni `{u/a, y/b, x/v}` rakendamine annab disjunktid

`P(a,v,b) ∨ S(v,b)` ja `¬ P(a,v,b) ∨ R(a,v)`,

millest resolutsioonireegel tuletab uue disjunkti `S(v,b) ∨ R(a,v)`.

Minimaalset unifitseerivat substitutsiooni nimetatakse *kõige üldisemaks unifitseerijaks*. Nimi on põhjendatud, kuna selline unifitseerija konkretiseerib literaale nii vähe kui võimalik. Saab tõestada, et mistahes kahel unifitseeritaval literaalil on üksainus kõige üldisem unifitseerija (muutujate ümbernimetamist arvestamata).

Minimaalse unifitseerija leidmise juures tekitab lisakeerukust asjaolu, et literaalis võivad peale muutujate ja konstantide olla ka keerukamad objektid, *funktsionaalsed termid*, mis on funktsiooni rakendused argumentidele. Näiteks võime arvu x ruutu võtmise tehet väljendada funktsionaalse termiga `ruut(x)`, kahe arvu x ja y korrutamise tehet termiga `*(x,y)`. Arvu x korrutist omaenda ruuduga saab siis väljendada funktsionaalse termiga `*(x,ruut(x))` ning literaali, mis ütleb, et $x * x = ruut(x)$, saame kirjutada `võrdub(*(x,x),ruut(x))`.

Defineerime mõiste *term*.

Definitsioon 17.7.

1. Iga muutuja ning konstant on *term*.
2. Iga konstruktsioon $f(t_1, \ldots, t_n)$, kus $0 \leq n$, f on funktsionaalsümbol ja iga t_i on *term*, konkreetsemalt, *funktsionaalterm*.
3. *Termi* saab konstrueerida ainult ülaltoodud reeglite 1 ja 2 abil.

Mainitud lisakeerukuseks on asjaolu, et ei saa olla substitutsiooni, mis muudaks samaseks mingi muutuja x ja sedasama muutujat x sisaldava funktsionaaltermi t. Püüame näiteks unifitseerida literaale `P(x,x)` ja `P(y,f(y))`. Esimeste argumentide `x`

ja y unifitseerimise järel saadud substitutsiooni {y/x} rakendades muutub term f(y) termiks f(x) ning meil tuleb järgmisena unifitseerida muutuja x funktsionaalse termiga f(x). Viimane on aga võimatu!

Defineerime *unifitseeritavuse* ja *kõige üldisema unifitseerija*.

Definitsioon 17.8. Ütleme, et termid t_1 ja t_2 on *unifitseeritavad*, kui leidub substitutsioon σ, nii et

$$t_1\sigma = t_2\sigma.$$

Termide t_1 ja t_2 *kõige üldisem unifitseerija* on substitutsioon ρ, mis rahuldab tingimusi:

1. ρ on termide t_1 ja t_2 unifitseerija.
2. t_1 ja t_2 iga unifitseerija σ korral leidub substitutsioon τ, nii et
$$\sigma = \tau \circ \rho,$$
s.t. iga termi t korral $\sigma(t) = \tau(\rho(t))$.

Literaalide unifitseeritavus ja kõige üldisem unifitseerija defineeritakse samamoodi.

Joonisel 17.1 toodud algoritm mgu arvutab literaalide või termide x ja y kõige üldisema unifitseerija. Edaspidi kasutamegi kahe literaali või termi x ja y kõige üldisema unifitseerija tähistamiseks kirjaviisi $mgu(x, y)$. Lühend *mgu* tuleb ingliskeelsest väljendist *most general unifier* ja on resolutsioonialases kirjanduses standardne. Algoritmi lahtikirjutamata osad tähendavad järgmist. term ja substitution tähistavad termi ja substitutsiooni tüüpi. Konstant EMPTYSUBST tähistab tühja substitutsiooni {}, konstant NOSUBST tähistab, et antud termidel ei saagi olla unifitseerivat substitutsiooni. Funktsioonid variable ja constant kontrollivad, kas argument on muutuja või konstant. Funktsioon length võtab argumendiks funktsionaalse termi ja annab selle argumentide arvu. Näiteks on termi ruut(x) argumentide arv üks, termi *(x,x) argumentide arv kaks, termi *(x,ruut(x)) argumentide arv samuti kaks. part(x,i) annab termi x i-nda argumendi,


```
substitution mgu(x,y)
term x,y;
{
 int i;
 substitution g;

 if x==y return EMPTYSUBST;
 else if variable(x) return mguvar(x,y);
 else if variable(y) return mguvar(y,x);
 else if (constant(x) || constant(y)) return NOSUBST;
 else if (length(x) != length(y)) return NOSUBST;
 i=0;
 g=EMPTYSUBST;
 while (i<=length(x)) {
   s=mgu(part(x,i),part(y,i));
   if s==NOSUBST return s;
   g=compose(g,s);
   x=substitute(x,g);
   y=substitute(y,g);
   i=i+1;
 }
 return g;
}

substitution mguvar(x,y)
term x,y;
{
 if occurs_in(x,y) return NOSUBST;
 else return makesubstitution(x,y);
}
```

Joonis 17.1. Unifitseerimisalgoritm.

kusjuures argumendiks 0 loetakse funktsionaalsümbolit. Näiteks on termi *(x,ruut(x)) argumendiks 0 funktsionaalsümbol *, argumendiks 1 muutuja x ja argumendiks 2 funktsionaalterm

`ruut(x)`. Funktsioon `compose` annab kahe substitutsiooni ühendamise resultaatsubstitutsiooni, `substitute(x,g)` rakendab `x`-le substitutsiooni `g` ja annab tulemuseks saadud resultaattermi. `make-substitution(x,y)` konstrueerib substitutsiooni $\{x/y\}$. Funktsioon `occurs_in` kontrollib, kas esimene argument esineb kusagil teise argumendi sees.

Mõned unifitseerimisnäited (meenutame, et muutujaid tähistavad meil sümbolid `x, y, z, u, v, w`):

- mgu(`P(a,x), P(y,b)`) = `{y/a,x/b}`.
- mgu(`P(x,f(x)), P(f(y),u)`) = `{x/f(y),u/f(f(y))}`.
- mgu(`L(g(x,x)), L(g(f(a),f(a)))`) = `{x/f(a)}`.
- mgu(`R(a,b), R(a,b)`) = `{}`.
- mgu(`P(a,x,x), P(y,b,y)`) ei eksisteeri.
- mgu(`P(f(x),f(x)), P(f(y),y)`) ei eksisteeri.
- mgu(`P(f(x),g(x)), P(f(y),f(y))`) ei eksisteeri.

Esitame nüüd resolutsioonireegli üldkuju. Viimane erineb muutujateta erijuhust selle poolest, et äralõigatavad literaalid ei pea olema identsed, vaid unifitseeritavad, ning allesjäänud literaalidele rakendatakse äralõigatud literaalide unifitseerijat.

$$\frac{A_1 \vee A_2 \vee \ldots \vee A_n \quad \neg B_1 \vee B_2 \vee \ldots \vee B_m}{(A_2 \vee \ldots \vee A_n \vee B_2 \vee \ldots \vee B_m)\sigma}\sigma = mgu(A_1, B_1)\,.$$

Enne resolutsioonireegli rakendamist tuleb alati kontrollida, kas disjunktid $A_1 \vee A_2 \vee \ldots \vee A_n$ ja $\neg B_1 \vee B_2 \vee \ldots \vee B_m$ sisaldavad ühiseid muutujaid. Kui jah, siis tuleb ühes disjunktis muutujad ümber nimetada, nii et ühiseid muutujaid ei jääks. Rakendame näiteks resolutsioonireeglit järgmistele disjunktidele:

`P(g(x),x,f(y)) ∨ R(x,y)` ja `¬ P(x,a,z) ∨ S(x,z).`

Kuna mõlemad sisaldavad muutujat `x`, nimetame teises disjunktis `x` ümber `u`-ks: `¬ P(u,a,z) ∨ S(u,z)`. Kommentaariks: kui muutujat ümber ei nimetata, pole mainitud

kahe disjunkti esimesed literaalid unifitseeritavad. Nüüd aga mgu(`P(g(x),x,f(y)), P(u,a,z)`) =`{u/g(a),x/a,z/f(y)}` ja tuletame disjunkti `R(a,y) ∨ S(a,f(y))`.

Järgnevalt meenutame muutujateta disjunktide faktoriseerimisreeglit, mis kõrvaldas disjunktist ülearused literaalide koopiad:

$$\frac{A_1 \vee A_1 \vee A_2 \vee \ldots \vee A_n}{A_1 \vee A_2 \vee \ldots \vee A_n}\,.$$

Üldisel, muutujatega juhul sisaldab ka faktoriseerimisreegel unifitseerimist:

$$\frac{A_1 \vee A_2 \vee A_3 \vee \ldots \vee A_n}{(A_2 \vee A_3 \vee \ldots \vee A_n)\sigma}\,\sigma = mgu(A_1, A_2)\,.$$

Faktoriseerimisreegli rakendamise näiteid:

- Disjunktist `P(x,y,y) ∨ P(a,y,z) ∨ S(x,y,z)` tuletame disjunkti `P(a,y,y) ∨ S(a,y,y)`.
- Disjunktist `R(x,y) ∨ R(y,x) ∨ R(a,a)` tuletame kolm disjunkti, faktoriseerides vastavalt literaale 1, 2 või 1, 3 või 2, 3: `R(x,x) ∨ R(a,a)` ja `R(a,a) ∨ R(a,a)` ja `R(a,a) ∨ R(a,a)`. Igaühele saadud disjunktidest saab omakorda rakendada faktoriseerimisreeglit, mis annab kõikidel juhtudel samasuguse ühest literaalist koosneva disjunkti: `R(a,a).`

Resolutsioonimeetodi täielikkuse üks alus on *Herbrandi teoreem*, mille esitame siin tõestuseta. Viimane on suhteliselt lihtne: jätame tõestuse leidmise harjutuseks lugejale.

Definitsioon 17.9. Disjunktide hulga S *Herbrandi universumiks* nimetatakse kõigi selliste muutujaid mittesisaldavate termide hulka, mida saab moodustada hulgas S sisalduvatest konstantidest ja funktsionaalsümbolitest. Kui hulgas S konstante pole, lubatakse kasutada suvalist konstanti a.

Teoreem 17.10 (Herbrandi teoreem). *Disjunktide hulk S on vastuoluline parajasti siis, kui eksisteerib lõplik vastuoluline muutujaid mittesisaldavate disjunktide hulk $S\sigma_1 \cup \ldots \cup S\sigma_n$, kus iga*

substitutsioon σ_i asendab kõik hulgas S esinevad muutujad termidega hulga S Herbrandi universumist.

Tõestame nüüd resolutsioonimeetodi täielikkuse üldjuhul. Tõestus kasutab muutujateta resolutsioonimeetodi täielikkuse tõestust koos järgneva *tõstmislemmaga*. Viimane ütleb sisuliselt, et kõige üldisemat unifitseerijat kasutav resolutsioonireegel katab kõikvõimalikud muutujaid mittesisaldavate konkretisatsioonidega tehtavad resolutsioonisammud.

Lemma 17.11 (tõstmislemma). *Olgu D_1 ja D_2 kaks ilma ühiste muutujateta disjunkti. Olgu $D_1\sigma$ ja $D_2\sigma$ antud disjunktide konkretisatsioonid, mis ei sisalda muutujaid. Olgu disjunkt R tuletatud disjunktidest $D_1\sigma$ ja $D_2\sigma$ resolutsioonireegli ühekordse rakendamisega. Siis saab disjunktidest D_1 ja D_2 resolutsioonireegli ühekordse rakendamisega tuletada sellise disjunkti R', et mingi substitutsiooni ρ jaoks kehtib $R = R'\rho$.*

Tõestus. Kuna R on saadud disjunktidele $D_1\sigma$ ja $D_2\sigma$ resolutsioonireeglit rakendades, siis sisaldab disjunkt $D_1\sigma$ literaali $L\sigma$ ja disjunkt $D_2\sigma$ literaali $\neg L'\sigma$, nii et $L\sigma = L'\sigma$ ja $R = (D_1\sigma - L\sigma) \vee (D_2\sigma - \neg L'\sigma)$, kus $(D_i\sigma - L\sigma)$ tähistab literaali $L\sigma$ eemaldamist disjunktist $D_i\sigma$. Olgu L literaalile $L\sigma$ vastav literaal disjunktis D_1 ja $\neg L'$ literaalile $\neg L'\sigma$ vastav literaal disjunktis D_2. Kuna literaalidel L ja $\neg L'$ on lemma eelduse kohaselt unifitseerija σ, siis on neil ka kõige üldisem unifitseerija $\mu = mgu(L, L')$. Järelikult saab disjunktidele D_1 ja D_2 rakendada resolutsioonireeglit, lõigates ära literaalid L ja $\neg L'$ ning andes tulemuseks disjunkti R': $(D_1\mu - L\mu) \vee (D_2\mu - \neg L'\mu)$. Kuna kõige üldisema unifitseerija definitsiooni järgi eksisteerib substitutsioon ρ, nii et $\sigma = \rho \circ \mu$, siis kehtib $R = R'\rho$, mida oligi vaja tõestada. □

Tõstmislemmat saab lihtsalt laiendada, võttes lisaks faktoriseerimisreegli. Järgnevas tõestuses eeldataksegi, et tõstmisreegel on faktoriseerimisele laiendatud.

Teoreem 17.12 (resolutsioonimeetodi täielikkus). *Kui disjunktide hulgal S ei ole ühtegi mudelit (s.t. S on vastuoluline ehk loogiliselt väär), siis on tühi disjunkt hulgast S resolutsioonimeetodiga tuletatav.*

Tõestus. Olgu disjunktide hulk S vastuoluline. Herbrandi teoreemi järgi eksisteerib siis lõplik vastuoluline muutujaid mittesisaldavate disjunktide hulk S': $S\sigma_1 \cup \ldots \cup S\sigma_n$, kus iga substitutsioon σ_i asendab kõik hulgas S esinevad muutujad termidega hulga S Herbrandi universumist. Muutujateta resolutsioonimeetodi täielikkuse teoreemist järeldub, et resolutsioonimeetodiga saab hulgast S' tuletada tühja disjunkti. Olgu T nimetatud tuletus. Asendame tuletuses T iga algdisjunkti $D\sigma_i$ vastava disjunktiga D hulgast S. Tõstmislemma järgi saab siis ühe muutujateta resolutsioonisammu resultaadid asendada unifitseerimist sisaldava resolutsioonisammu resultaatidega jne., kuni terve T on teisendatud tühja disjunkti tuletuseks hulgast S. □

Teatavasti tõestasid Turing ja Church 1930. aastatel predikaatarvutuse mittelahenduvuse. Resolutsioonimeetodi puhul avaldub mittelahenduvus asjaolus, et erinevalt muutujateta disjunktide juhust ei peatu suvalisele mittevastuolulisele disjunktihulgale rakendatud resolutsioonimeetod alati, vaid võib jääda lõpmatuseni tuletama uusi disjunkte. Selliselt käitub resolutsioonimeetod näiteks disjunktihulgale

```
{¬ P(x) ∨ P(f(x)),  P(y) ∨ ¬ P(f(y))}
```

rakendatuna.

Ülesanded

3. Kontrollige termide unifitseeritavust ning leidke nende kõige üldisemad unifitseerijad:
   (a) $p(f(y), w, g(z))$ ja $p(u, u, v)$;
   (b) $p(f(y), w, g(z))$ ja $p(v, u, v)$;
   (c) $p(a, x, f(g(y)))$ ja $p(z, h(z, w), f(w))$.

4. *Leidke unifitseerimisprobleemi

$$mgu(f(x_2, x_3, \ldots, x_n), f(g(x_1, x_1), g(x_2, x_2), \ldots, g(x_{n-1}, x_{n-1})))$$

lahend. Näidake, et see kasvab arvu n suurenedes eksponentsiaalselt.

17.2.3. Lahenduste otsimine

Siiani oleme resolutsioonimeetodit rakendanud ainult *ei/ja*-küsimustele vastamiseks: teoreemi saab kas tõestada või ei.

Resolutsioonimeetodit saab aga rakendada ka sellistele küsimustele vastamiseks: "Leia niisugune t, et teoreem kehtiks objekti t jaoks".

Toome näiteks järgmise andmebaasi:

(1) `isa(Jaan,Peeter)`,

(2) `isa(Jaan,Martin)`,

(3) `isa(Martin,Veiko)`,

(4) `isa(Riivo,Leo)`,

(5) `¬ isa(x,y) ∨ ¬ isa(y,z) ∨ vanaisa(x,z)`.

Küsime nüüd: "Kes on Veiko vanaisa?". Lahenduse leidmiseks lisame andmebaasile Veiko vanaisaks olemise väite eituse, jättes vanaisa positsioonile muutuja ning lisades väitele spetsiaalse *lahendusliteraali*:

(6) `¬ vanaisa(x,Veiko) ∨ lahendus(x)`.

Modifitseerime resolutsioonimeetodit pisut: laseme tööl peatuda, kui on tuletatud kas tühi disjunkt või ainult lahendusliteraale sisaldav disjunkt. Viimasel juhul on tuletatud lahendusliteraali argument otsitav lahendus.

(2 ja 5 annavad disjunkti 7) `¬ isa(Martin,z) ∨ vanaisa(Jaan,z)`,

(7 ja 3 annavad disjunkti 8) `vanaisa(Jaan,Veiko)`,

(8 ja 6 annavad disjunkti) `lahendus(Jaan)`.

Küsimustel võivad olla ka *mitteühesed lahendused*. Sellisel juhul tuletab meetod disjunkti, mis sisaldab mitu erinevat, mitteunifitseeritavat lahendusliteraali. Lihtne näide: disjunktide

(1) `p(a) ∨ p(b)`,

(2) `¬ p(x) ∨ lahendus(x)`

hulgast saab tuletada disjunkti `p(b) ∨ lahendus(a)` ning viimasest omakorda `lahendus(b) ∨ lahendus(a)`. Ühest lahendust ei ole antud juhul võimalik tuletada.

17.2.4. Võrduspredikaat

Üks olulisemaid nii loogikas kui rakendustes ettetulevaid predikaate on *võrdus*. Võrdusega on põhimõtteliselt võimalik formaliseerida kõike, mida loogikavahenditega formaliseerida saab. Võrduse kasulikkus matemaatikateoreemide, elektroonikaskeemide ja arvutiprogrammide formaliseerimisel motiveerib spetsiaalsete meetodite väljatöötamist võrduse kasutamisel tuletustes.

Kasutame võrduspredikaadiks edaspidi harilikku võrdussümbolit $=$. Literaal $=(t_1, t_2)$ tähistab siis termide t_1 ja t_2 võrdsust.

Kuna sümbolil "$=$" puuduvad maagilised omadused, mis ta iseenesest võrdust väljendama panevad, tuleb võrduspredikaat *defineerida* järgmiste disjunktidega:

$D_{\text{refleksiivsus}}$: `=(x,x)`,

$D_{\text{sümmeetria}}$: `¬ =(x,y) ∨ =(y,x)`,

$D_{\text{transitiivsus}}$: `¬ =(x,y) ∨ ¬ =(y,z) ∨ =(x,z)`,

asendus literaali: $\neg =(x_i, y_i) \vee \neg P(u_1, \ldots, x_i, \ldots, u_n) \vee P(u_1, \ldots, y_i, \ldots, u_n)$,

asendus termi: $\neg\ =(x_i, y_i)\ \vee\ f(u_1, \ldots, x_i, \ldots, u_n) = f(u_1, \ldots, y_i, \ldots, u_n)$,

kus viimased kaks disjunkti, asendus literaali ja asendus termi, on võrdsete termide asendatavust formaliseerivad disjunktiskeemid. Võrduse defineerimisel mingi konkreetse disjunktide hulga jaoks tuleb iga predikaatsümboli ja funktsionaalsümboli jaoks kirjutada eraldi välja n konkreetset asendust formaliseerivat disjunkti, kus n on antud sümboli argumentide arv.

Näide: koosnegu S kolmest üheliteraalilisest disjunktist:

(1) $\neg$ `p(f(x))`,

(2) `p(b)`,

(3) `=(f(a),b)`.

Tühja disjunkti tuletamiseks hulgast S on tarvis defineerida võrduspredikaat, s.o. lisada disjunktid $D_{\text{refleksiivsus}}$, $D_{\text{sümmeetria}}$, $D_{\text{transitiivsus}}$ ja järgmised kaks asendust defineerivat disjunkti:

(4) $\neg$ `=(x,y)` $\vee$ $\neg$ `p(x)` $\vee$ `p(y)`,

(5) $\neg$ `=(x,y)` $\vee$ `=(f(x),f(y))`.

Vastuolu tuletus:

(3 ja $D_{\text{sümmeetria}}$ annavad disjunkti 6) `=(b,f(a))`,

(6 ja 4 annavad disjunkti 7) $\neg$ `p(b)` $\vee$ `p(f(a))`,

(2 ja 7 annavad disjunkti 8) `p(f(a))`,

(1 ja 8 annavad tühja disjunkti) vastuolu.

Võrduspredikaati defineerivate disjunktide kasutamine on resolutsioonimeetodi jaoks kohmakas ja ebaefektiivne meetod. Näiteks võimaldab $D_{\text{transitiivsus}}$ üksi tuletada lõpmatu arvu uusi disjunkte, mida tegelikult vastuolu leidmiseks kunagi tarvis ei lähe.

Võrduspredikaadi aksiomatiseerimise asemel kasutatakse resolutsioonimeetodi juures enamasti spetsiaalset lisareeglit, nn. *paramodulatsioonireeglit*

$$\frac{A[t] \vee A_1 \vee \ldots \vee A_n \quad = (t_1, t_2) \vee B_1 \vee \ldots \vee B_m}{(A[t_2] \vee A_1 \ldots \vee A_n \vee B_1 \vee \ldots \vee B_m)\sigma}\sigma = mgu(t, t_1)$$

kus $A[t]$ tähistab positiivset või negatiivset literaali, mis sisaldab termi t, mis ei ole muutuja, ning $A[t_2]$ tähistab sedasama literaali, kus term t on asendatud termiga t_2. Lisaks reegli äsjatoodud kujule kasutame võrduspredikaadi sümmeetria tõttu veel teist kuju, kus literaal $= (t_1, t_2)$ on asendatud literaaliga $= (t_2, t_1)$.

Paramodulatsioonireegli abil saab sooritada termide asendusi otse, s.t. asendada ühe literaalis leiduva termi t temaga võrdse termiga t_2, sõltumatult termi t asukohast literaalis. Nõue, et t ei oleks muutuja, on kasutusel ainult efektiivsuse huvides — paramodulatsioonireegel on korrektne ka ilma sellise piiranguta.

Paramodulatsiooni kasutades saame eelmisest näitest tuletada vastuolu, kasutamata võrduspredikaati aksiomatiseerivaid disjunkte:

(3 ja 1 annavad paramodulatsiooni abil disjunkti 4) $\neg$ `p(b)`,

(2 ja 4 annavad tühja disjunkti) vastuolu.

Esitame järgmise teoreemi ilma tõestuseta.

Teoreem 17.13 (paramodulatsiooni täielikkus). *Tähistagu $S_{asendus}$ asendust defineerivate disjunktide hulka disjunktihulga S jaoks. Kui disjunktide hulk*

$$S \cup \{D_{refleksiivsus}, D_{sümmeetria}, D_{transitiivsus}\} \cup S_{asendus}$$

on vastuoluline, siis on tühi disjunkt hulgast $S \cup \{D_{refleksiivsus}\}$ paramodulatsioonireegliga täiendatud resolutsioonimeetodi abil tuletatav.

17.2.5. Loogikavalemi teisendamine disjunktide hulgaks

Kasutamaks resolutsioonimeetodit esimest järku predikaatarvutuse valemi tõestamisel, on tarvis uuritav valem kõigepealt disjunktide

hulgaks teisendada. Selle teisenduse jaoks eksisteerib mitmeid algoritme, kusjuures mõned algoritmid annavad teatud juhtudel väiksemaid ja resolutsioonimeetodi jaoks sobivamaid disjunktide hulki kui teised. On olemas ka selliseid algoritme, kus resolutsioonimeetod kombineeritakse disjunktideks teisendusega dünaamiliselt, s.t. resolutsioonimeetodit hakatakse rakendama juba enne, kui kogu valem on disjunktide hulgaks teisendatud.

Esitame siinkohal ühe suhteliselt lihtsa disjunktideks teisendamise algoritmi. Olgu meil antud esimest järku predikaatarvutuse valem F.

1. Teisendame valemi F ekvivalentseks valemiks F_1, asendades valemis F kõik seostega $\Rightarrow$ ja $\Leftrightarrow$ konstrueeritud alamvalemid ekvivalentsete alamvalemitega, mis sisaldavad ainult seoseid $\neg$, $\vee$ ja $\&$:
   - $A \Rightarrow B$ asendame valemiga $\neg A \vee B$,
   - $A \Leftrightarrow B$ asendame valemiga $(\neg A \vee B) \,\&\, (\neg B \vee A)$.
2. Teisendame valemi F_1 ekvivalentseks valemiks F_2, nihutades eitusseosed sissepoole, elementaarvalemite ette. Selleks kasutame järgmisi asendusi:
   - $\neg\neg A$ asendame valemiga A,
   - $\neg(A \,\&\, B)$ asendame valemiga $\neg A \vee \neg B$,
   - $\neg(A \vee B)$ asendame valemiga $\neg A \,\&\, \neg B$,
   - $\neg\forall x\, A$ asendame valemiga $\exists x \neg A$,
   - $\neg\exists x\, A$ asendame valemiga $\forall x \neg A$.
3. Teisendame valemi F_2 ekvivalentseks valemiks F_3, nimetades kõik muutujad selliselt ümber, et kahel eri muutujal oleksid alati erinevad nimed, s.t. ükski muutuja valemis ei oleks seotud rohkem kui ühe kvantoriga. Näide: teisendame valemi $\forall x\, P(x, x) \,\&\, \exists x\, Q(x)$ valemiks $\forall x\, P(x, x) \,\&\, \exists y\, Q(y)$.
4. Teisendame valemi F_3 valemiks F_4, kaotades kõik olemasolukvantorid $\exists$. See samm, *skolemiseerimine*, on teisendusalgoritmi juures kõige keerulisem ja selle korrektsus ei

ole elementaarne. Meetodi nimi tuleb leiutaja, Norra loogiku Thoralf Skolemi (1887–1963) nimest.

   - Kui olemasolukvantori abil ehitatud alamvalem $\exists x\, A$ ei ole ühegi üldisuskvantori $\forall$ mõjupiirkonnas, siis asendame alamvalemi $\exists x\, A$ valemiga A', kus A' on saadud valemist A, asendades seal muutuja x uue konstandiga a. *Uue konstandi* all mõtleme suvalist konstanti, mida *ei ole* teisendatavas valemis. Toome näite. Valemis $\forall x\, P(x, a) \,\&\, \exists y\, Q(y, y)$ ei asu $\exists y\, Q(y, y)$ ühegi üldisuskvantori mõjupiirkonnas, seega kaotame olemasolukvantori ja asendame muutuja y konstandiga b: $\forall x\, P(x, a) \& Q(b, b)$. NB! konstanti a ei või asenduseks kasutada, kuna ta juba on teisendatavas valemis.
   - Asugu nüüd alamvalem $\exists x\, A$ mingi hulga üldisuskvantorite $\forall y_1, \forall y_2, \ldots, \forall y_n$ mõjupiirkonnas. Asendame alamvalemi $\exists x\, A$ valemiga A', kus A' on saadud valemist A, asendades muutuja x uue funktsionaaltermiga $f(y_1, y_2, \ldots, y_n)$, kusjuures funktsionaalsümbol f on uus, s.t. f ei esine teisendatavas valemis. Näide: valemis $\forall x \forall y \exists z\, P(f(z), x, y, z)$ asendame muutuja z termiga $g(x, y)$, mis annab valemi $\forall x \forall y\, P(f(g(x, y)), x, y, g(x, y))$.
5. Teisendame valemi F_4 ekvivalentseks valemiks F_5, eemaldades kõik üldisuskvantorid $\forall$. Muutujad valemis F_5 on nüüd nn. *vabad muutujad*.
6. Teisendame valemi F_5 ekvivalentseks *konjunktiivsel normaalkujul* olevaks valemiks F_6. Selleks teisendame kõik alamvalemid kujul $A \vee (B \,\&\, C)$ ekvivalentsele kujule $(A \vee B) \,\&\, (A \vee C)$.
7. Valem F_6 on nüüd kujul $C_1 \,\&\, C_2 \,\&\, \ldots \,\&\, C_n$, kus iga C_i on disjunkt. Teisendame valemi F_6 disjunktide hulgaks $\{C_1, C_2, \ldots, C_n\}$.

Näitena toome valemi

$$\forall x((\forall y P(x, y)) \Rightarrow \neg\forall y(Q(x, y) \Rightarrow R(x, y)))$$

sammsammulise teisenduse disjunktide hulgaks:

1. $\forall x(\neg(\forall y P(x, y)) \vee \neg\forall y(\neg Q(x, y) \vee R(x, y)))$
2. $\forall x((\exists y\neg P(x, y)) \vee (\exists y Q(x, y) \,\&\, \neg R(x, y)))$
3. $\forall x((\exists y\neg P(x, y)) \vee (\exists z Q(x, z) \,\&\, \neg R(x, z)))$
4. $\forall x(\neg P(x, f(x)) \vee (Q(x, g(x)) \,\&\, \neg R(x, g(x))))$
5. $\neg P(x, f(x)) \vee (Q(x, g(x)) \,\&\, \neg R(x, g(x)))$
6. $(\neg P(x, f(x)) \vee Q(x, g(x))) \,\&\, (\neg P(x, f(x)) \vee \neg R(x, g(x)))$
7. $\{\neg P(x, f(x)) \vee Q(x, g(x)),\ \ \neg P(x, f(x)) \vee \neg R(x, g(x))\}$

Enamik ülaltoodud teisendussammudest annab tulemuseks lähtevalemiga ekvivalentse valemi. Ainsa erandina ei anna lähtevalemiga ekvivalentset tulemust *skolemiseerimine* ning seetõttu ei ole ka kogu teisenduse lõpptulemus lähtevalemiga loogiliselt ekvivalentne. Automaatse teoreemitõestamise tarvis piisab aga järgmisest Skolemi tõestatud teoreemist.

Teoreem 17.14. *Olgu F esimest järku predikaatarvutuse valem ja $Sk(F)$ selle valemi skolemiseeritud kuju. F on vastuoluline siis ja ainult siis, kui $Sk(F)$ on vastuoluline.*

Peamisi ebameeldivusi disjunktideks teisendamisel valmistab eelviimane samm, mis asendab kõik alamvalemid kujul $A \vee (B \,\&\, C)$ ekvivalentse, kuid suurema kujuga $(A \vee B) \& (A \vee C)$. Viimase asenduse tõttu võib alguses suhteliselt väike loogikavalem teiseneda väga suureks disjunktihulgaks.

17.3. Resolutsioonimeetodi strateegiad ja rakendused

Resolutsioonimeetodi üks ebameeldivamaid külgi on suure hulga mittevajalike disjunktide (näiteks korduvate disjunktide) tuletamine. Meetodi efektiivsuse suurendamise peamine vahend on selliste variantide ehk strateegiate väljatöötamine, mis tuletaksid võimalikult vähe disjunkte, kaotamata samas täielikkust.

Esitame siin resolutsioonimeetodi mõne kasutatavama strateegia.

17.3.1. Neelamisstrateegia ja tautoloogia kõrvaldamine

Neelamisstrateegia on pea kõige universaalsem resolutsioonimeetodi strateegiatest, seda saab kombineerida enamiku teiste strateegiatega ning ta on realiseeritud praktiliselt kõigis automaatse teoreemitõestamise programmides.

Definitsioon 17.15. Disjunkt D' *neelab* disjunkti D, kui mingi substitutsiooni σ ja disjunkti Γ jaoks kehtib $D = (D' \vee \Gamma)\sigma$.

Sisuliselt tähendab disjunkti D neelamine disjunkti D' poolt, et D' on üldisem või sama üldine (viimasel juhul $D' = D$) kui D: kõik, mis tuleneb loogiliselt D-st, tuleneb ka D'-st. Kui D' neelab D, siis kehtib alati $D' \Rightarrow D$.

Näide: disjunkt `p(x) ∨ r(y)` neelab disjunkti `p(a) ∨ r(v) ∨ s(z)`.

Neelamisstrateegia seisneb järgmises kahes piirangus:

1. Kui viimati tuletatud disjunkti D neelab kas mõni varemtuletatud või lähtedisjunkt, siis kõrvaldatakse D kohe, s.t. edasises tuletamises teda ei kasutata.

2. Kui viimati tuletatud disjunkt D ise neelab kas mõne varem tuletatud või lähtedisjunkti R, siis kõrvaldatakse R kohe.

Teoreem 17.16. *Neelamisstrateegia säilitab resolutsioonimeetodi täielikkuse.*

Tõestus. Induktsioon sammude arvu järgi vastuolu tuletuses. Induktsiooni baas on triviaalne. Induktsiooni samm: olgu disjunkt D tuletatud resolutsioonireegli rakendamisega disjunktidele D_1 ja D_2. Kui asendada disjunkt D_1 teda neelava disjunktiga D_1', siis kehtib üks kahest: muudetud tuletussammu resultaat D' neelab disjunkti D või D_1' neelab ise disjunkti D. Analoogiline väide kehtib faktoriseerimisreegli kohta. □

Tautoloogia kõrvaldamise strateegia on sarnane neelamisstrateegiaga.

Definitsioon 17.17. *Tautoloogiliseks* nimetame disjunkte $L \vee \neg L \vee \Gamma$, kus Γ on mingi disjunkt.

Antud strateegia seisneb tautoloogiliste disjunktide koheses kõrvaldamises. Strateegia säilitab resolutsioonimeetodi täielikkuse: tõestuseks piisab, kui märgata, et disjunktidest $L \vee \neg L$ ja D resolutsioonireegliga saadud disjunkt D' neelab eelduseks oleva disjunkti $L \vee \neg L$.

17.3.2. Toetushulga strateegia

Tõestuse otsimisel on tihti tegu olukorraga, kus on ette antud suur hulk suhteid ja omadusi defineerivaid valemeid, mingi matemaatikavaldkonna formalisatsioon või faktide/reeglite andmebaas. Sellise definitsioonide kogu või andmebaasi puhul on enamasti teada, et antud valemite hulk ei ole vastuoluline.

Taolisest valemite hulgast K mingi järelduse J tegemine tähendab valemi $K \Rightarrow J$ tõestamist, s.t. $K \,\&\, \neg J$ vastuolulisuse näitamist. Teisendame valemi K disjunktide hulgaks K' ja valemi $\neg J$ disjunktide hulgaks J'. Eesmärgiks on hulgast $K' \cup J'$ tuletada vastuolu.

Kui K' sisaldab palju disjunkte, siis tekib sageli olukord, kus resolutsioonimeetod juba esimestel tasemetel tuletab ainuüksi K'-s

leiduvatest disjunktidest tohutu hulga uusi disjunkte, millest enamikku vastuolu tuletamiseks kasutada ei saa. Küll aga põhjustab nende disjunktide tuletamine ajakulu ja muudab vastuolu leidmise äärmiselt raskeks. Resolutsioonimeetod nö. ei hooli asjaolust, et on tarvis näidata nimelt valemi J järeldatavust.

Toetushulga strateegia eesmärgiks on suunata resolutsioonimeetodit vastuolu otsimisel alustama nimelt valemist J, liikudes otsitavast järeldusest tagasisuunas faktide poole.

Olukorras, kus otsitakse $K' \cup J'$ vastuolu, ning on teada, et K' ei ole vastuoluline, nimetatakse hulka J' hulga $K' \cup J'$ *toetushulgaks*.

Toetushulga strateegia piirab resolutsioonireegli rakendamist, nõudes, et vähemalt üks reegli kahest eeldusest peab kas kuuluma toetushulka või olema toetushulga strateegiaga tuletatud. Pole raske tõestada, et toetushulga strateegia säilitab resolutsioonimeetodi täielikkuse.

Vaatame näitena järgmist disjunktide hulka, võttes disjunkti 4 toetushulga ainsaks elemendiks ja kasutades toetushulga strateegiat.

(1) `p(x)` $\vee$ `q(x)`;

(2) $\neg$ `p(a)` $\vee$ `r(x)`;

(3) $\neg$ `q(x)` $\vee$ `r(a)`;

(4) $\neg$ `r(a)`.

Esimesel tasemel saab tuletada kaks disjunkti:

(4 ja 2 annavad disjunkti 5) $\neg$ `p(a)`, mis neelab disjunkti 2;

(4 ja 3 annavad disjunkti 6) $\neg$ `q(x)`, mis neelab disjunkti 3.

Teisel tasemel saab tuletada kaks disjunkti:

(5 ja 1 annavad disjunkti 7) `q(a)`;

(6 ja 1 annavad disjunkti 8) `p(x)`, mis neelab disjunkti 1.

Kolmandal tasemel annavad disjunktid 7 ja 6 vastuolu.

17.3.3. Järjestatud resolutsioon

Järjestatud resolutsioon on üks sellistest tõestuse otsimise strateegiatest, mille juures keelatakse osa resolutsioonireegli võimalikke rakendusi. Järjestatud resolutsioon ei ole üks konkreetne strateegia, vaid pigem terve strateegiate perekond, mille iga liige realiseerib üht kindlat literaalide järjestamise viisi disjunktis.

Järjestatud resolutsiooni strateegia on järgmine:

- Defineeritakse *literaalide järjestus* $\succ$ disjunktis, s.t. meetod, mille alusel saab iga kahe literaali A ja B jaoks kontrollida, kas $A \succ B$.
- Resolutsioonireeglil keelatakse disjunktis selliste literaalide äralõikamine, millest suuremaid (järjestuse $\succ$ mõttes) literaale disjunkt sisaldab. Niisiis, reeglit

  $$\frac{A_1 \vee A_2 \vee \ldots \vee A_n \quad \neg B_1 \vee B_2 \vee \ldots \vee B_m}{(A_2 \vee \ldots \vee A_n \vee B_2 \vee \ldots \vee B_m)\sigma}\sigma = mgu(A_1, B_1)$$

  võib rakendada ainult juhul, kui ühegi literaali A_i korral ei kehti $A_i \succ A_1$ ning ühegi literaali B_i korral ei kehti $B_i \succ \neg B_1$.

Näide. Defineerime järjestuse $\succ_1$, kasutades järjestamiseks literaalis esinevaid predikaatsümboleid:

$$p \succ_1 q \succ_1 \neg q \succ_1 \neg p \succ_1 \neg r \succ_1 \succ r.$$

Vaatame, milliseid disjunkte järgmisest $\succ_1$ abil järjestatud disjunktihulgast tuletada saab. Paneme tähele, et $\succ_1$ lubab nendest disjunktidest ära lõigata ainult kõige esimesi disjunkte:

(1) `p(x)` $\vee$ `q(x)`;

(2) $\neg$ `p(a)` $\vee$ `r(x)`;

(3) $\neg$ `q(x)` $\vee$ `r(a)`;

(4) $\neg$ `r(a)`.

Esimesel tasemel saab tuletada üheainsa disjunkti:

(1 ja 2 annavad disjunkti 5) `q(a)` $\vee$ `r(x)`.

Teisel tasemel saab tuletada kaks disjunkti:

(5 ja 3 annavad disjunkti 6) `r(a)` $\vee$ `r(x)`;

(6 annab faktoriseerimisel disjunkti 7) `r(a)`.

Disjunkt 7 neelab disjunktid 3 ja 6 ning annab kolmandal tasemel koos disjunktiga 4 vastuolu.

Võrdluseks: piiranguteta resolutsioonimeetodiga saame äsjasest disjunktihulgast tuletada enam kui kakskümmend uut disjunkti.

Järjestatud resolutsiooni täielikkuse tõestamiseks pole leitud universaalset meetodit ning tõestus sõltub sageli järjestuse definitsioonist. Üldjuhul ei saa järjestusstrateegiaid kombineerida toetusstrateegiaga ning mitut eriti tugevat järjestusstrateegiat ei saa kombineerida ka neelamisstrateegiaga. Küsimusele, kas *kõik* sellised järjestused, mis on sõltumatud muutujate ümbernimetamisest, säilitavad resolutsioonimeetodi täielikkuse, pole raamatu kirjutamise hetkeks vastust veel leitud.

Üks lihtsamaid, samas aga väga praktiline järjestamismeetod on järjestada literaale polaarsuse järgi, näiteks lugedes negatiivseid literaale suuremaks kui positiivseid.

Olgu meil jällegi tegemist reeglite/faktide andmebaasiga K, millest püüame tuletada disjunkti J. Teatavasti saab iga disjunkti esitada implikatsioonina: kehtib valem

$$(\neg N_1 \vee \ldots \vee \neg N_n \vee P_1 \vee \ldots \vee P_m) \Leftrightarrow (N_1 \& \ldots \& N_n \Rightarrow P_1 \vee \ldots \vee P_m).$$

Järgmises näites kuuluvad disjunktid 1–2 andmebaasi K, millest püütakse tuletada disjunkti 3 eitus. Illustratiivsuse huvides esitame disjunktid kõrvuti oma implikatsioonikujuga.

(1)	$\neg$ `M(x)` $\vee$ `P(x)`	$M(x) \Rightarrow P(x)$
(2)	`M(a)`	$M(a)$
(3)	$\neg$ `P(z)`	$P(z) \Rightarrow$

Kasutame kõigepealt järjestust, kus negatiivsed literaalid tulevad enne positiivseid. Sel juhul saab esimesel tasemel tuletada vaid ühe disjunkti: disjunktid 1 ja 2 annavad P(a), ning viimane annab teisel tasemel vastuolu disjunktiga 3. Tõestuse otsimine algas seega *faktidest* andmebaasis K, jõudes lõpuks soovitud järelduse tuletamiseni.

Järgmisena kasutame vastupidist järjestust: positiivsed literaalid tulevad enne negatiivseid. Esimesel tasemel annavad nüüd disjunktid 3 ja 1 disjunkti ¬ M(x), millest teisel tasemel tuletame koos disjunktiga 2 vastuolu. Tõestuse otsimine algas seega soovitud järeldusest ning jõudis lõpuks faktideni andmebaasis K.

Väga keeruline on vastata küsimusele, milline järjestus millisel juhul paremini sobib, s.t. põhjustab väiksema arvu disjunktide tuletamise. Vaatame kahte näidet, kus esimeses on eelistatav suund *faktidest järelduseni*, teises aga *järeldusest faktideni*. Esitame andmebaasis olevad disjunktid implikatsioonidena.

Esimene näide. Olgu meil järgmine andmebaas:

putukas(x) ⇒ elusolend(x)
imetaja(x) ⇒ elusolend(x)
sipelgas(x) ⇒ putukas(x)
mesilane(x) ⇒ putukas(x)
ämblik(x) ⇒ putukas(x)
lõvi(x) ⇒ imetaja(x)
tiiger(x) ⇒ imetaja(x)
sebra(x) ⇒ imetaja(x)

Oletame, et Zeke on sebra. Küsime: kas Zeke on elusolend? Kasutame järjestatud resolutsiooni, kus negatiivsed literaalid tulevad enne positiivseid, s.t. otsingu suund on faktidest järelduseni. Kirjutame välja kõik tuletatavad disjunktid, eraldades tasemed horisontaaljoonega. Disjunktid 1 ja 2 on meie küsimuse eitusele vastavad lähtedisjunktid.

(1) sebra(Zeke)
(2) ¬ elusolend(Zeke)

---

(3) imetaja(Zeke)

---

(4) elusolend(Zeke)

---

(5) vastuolu.

Järgmisena kasutame vastupidiselt järjestatud resolutsiooni: positiivsed literaalid tulevad enne negatiivseid, s.t. otsingu suund on *järeldusest faktideni*. Kirjutame jällegi välja kõik tuletatavad disjunktid.

(1) sebra(Zeke)
(2) ¬ elusolend(Zeke)

---

(3) ¬ putukas(Zeke)
(4) ¬ imetaja(Zeke)

---

(5) ¬ sipelgas(Zeke)
(6) ¬ mesilane(Zeke)
(7) ¬ ämblik(Zeke)
(8) ¬ lõvi(Zeke)
(9) ¬ tiiger(Zeke)
(10) ¬ sebra(Zeke)

---

(11) vastuolu.

Nagu näha, tuletatakse viimase järjestuse kasutamise korral palju rohkem disjunkte kui esimese järjestuse korral.

Teine näide. Olgu meil järgmine sebrade andmebaas:

sebra(x) ⇒ imetaja(x)
sebra(x) ⇒ triibuline(x)
sebra(x) ⇒ keskmiselt_suur(x)
imetaja(x) ⇒ elusolend(x)

```
elusolend(x) ⇒ soe(x)
triibuline(x) ⇒ mitteühevärviline(x)
triibuline(x) ⇒ mittetäpiline(x)
keskmiselt_suur(x) ⇒ mitteväike(x)
keskmiselt_suur(x) ⇒ mittesuur(x)
```

Uurime, kas asjaolust, et Zeke on sebra, järeldub, et Zeke on mittesuur. Kasutame kõigepealt otsingut suunaga faktidest järelduseni. Disjunktid 1 ja 2 on meie küsimuse eitusele vastavad lähtedisjunktid.

```
(1) sebra(Zeke)
(2) ¬ mittesuur(Zeke)
-----------------------
(3) imetaja(Zeke)
(4) triibuline(Zeke)
(5) keskmiselt_suur(Zeke)
-----------------------
(6) elusolend(Zeke)
(7) mitteühevärviline(Zeke)
(8) mittetäpiline(Zeke)
(9) mitteväike(Zeke)
(10) mittesuur(Zeke)
-----------------------
(11) vastuolu.
```

Järgmiseks kasutame otsingut suunaga *järeldusest faktideni*, mis vastupidi esimesele näitele on praeguse andmebaasi jaoks tunduvalt efektiivsem.

```
(1) sebra(Zeke)
(2) ¬ mittesuur(Zeke)
-----------------------
(3) ¬ keskmiselt_suur(Zeke)
-----------------------
(4) ¬ sebra(Zeke)
-----------------------
(5) vastuolu.
```


17.4. Näiteid ja rakendusi

Nüüdisaegsed resolutsioonimeetodil põhinevad automaatse teoreemitõestamise programmid tuletavad sekundis mitu tuhat disjunkti. Seejuures tehakse iga tuletatud disjunktiga tavaliselt veel hulk mitmesuguseid operatsioone, nagu neelamiskontroll, disjunkti lihtsustamine jne., mis võivad kulutada tunduvalt rohkem aega kui disjunkti tuletamine ise.

Kuitahes kiiresti tõestajad ka disjunkte ei tuletaks, ei pääse keegi mööda asjaolust, et tuletatavate disjunktide hulk kasvab tasemete lõikes kiiremini kui eksponentsiaalselt. Sügavate, paljusid samme sisaldavate tõestuste täisautomaatne leidmine ei õnnestu kaugeltki alati — pigem vastupidi. Keeruliste tõestuste automaatse leidmise nimel töötatakse pidevalt välja uusi strateegiaid ja heuristikaid — viimaste efekt on palju suurem kui arvutikiiruste paari- või kümnekordsel kasvamisel.

Vaatame nüüd mõningaid mittetriviaalseid, samas aga mitte ka ülemäära keerulisi näiteid mitmest sfäärist.

17.4.1. Algebra

Esimene näide on suhteliselt lihtne teoreem rühmateooriast. Olgu meil kahekohaline assotsiatiivne tehe * koos pöördelemendi võtmise funktsiooniga i. Tõestame, et kui sellisel tehtel on parempoolne ühikelement e, siis on seesama e ka vasakpoolne ühikelement.

Disjunktid 1–3 formaliseerivad pöörd- ja ühikelemendi ning tehte * assotsiatiivsuse. Disjunkt 4 on teoreemi $\forall x(*(e,x) = x)$ skolemiseeritud eitus.

```
(1) =(*(x,i(x)),e)
(2) =(*(x,e),x)
(3) =(*(x,*(y,z)),*(*(x,y),z))
(4) ¬ =(*(e,c),c)
(5) =(x,x)
```

Tõestuse otsimisel kasutame paramodulatsiooni. Automaatne teoreemitõestaja leiab vastuolu sekundi murdosa vältel, pärast ligi poolteise tuhande disjunkti tuletamist. Järgnevas tõestuses on kõik sammud peale viimase paramodulatsioonireegli rakendused.

(2 ja 3 annavad 8) `=(*(x,*(e,y)),*(x,y))`

(1 ja 3 annavad 10) `=(*(x,e),*(*(x,y),i(y)))`

(8 ja 1 annavad 46) `=(*(x,*(e,i(x))),e)`

(2 ja 10 annavad 145) `=(x,*(*(x,y),i(y)))`

(1 ja 145 annavad 175) `=(x,*(e,i(i(x))))`

(175 ja 46 annavad 206) `=(*(i(x),x),e)`

(206 ja 145 annavad 233) `=(i(x),*(e,i(x)))`

(233 ja 175 annavad 560) `=(x,i(i(x)))`

(560 ja 175 annavad 569) `=(x,*(e,x))`

(569 ja 4 annavad 1421) ¬ `=(c,c)`

(5 ja 1421 annavad tühja disjunkti) `vastuolu`.

17.4.2. Metaloogika

Teise näite võtame loogika vallast. Uurime selliseid klassikalise lausearvutuse valemeid, mis sisaldavad ainult lausemuutujaid ja implikatsiooni. Łukasiewicz tõestas, et taoliselt piiratud klassikalise lausearvutuse aksiomatiseerimiseks piisab lisaks *modus ponens*'i reeglile ühestainsast aksioomiskeemist:

$$((x \Rightarrow y) \Rightarrow z) \Rightarrow ((z \Rightarrow x) \Rightarrow (u \Rightarrow x)).$$

Täielikkuse tõestamiseks tuleb nimetatud aksioomist *modus ponens*'iga tuletada kõik harilikud implikatsiooni formaliseerivad aksioomiskeemid. Siinses näites tuletame neist ainult ühe, Peirce'i reegli:

$$(((x \Rightarrow y) \Rightarrow x) \Rightarrow x).$$

Implikatsiooni $x \Rightarrow y$ esitame termi kujul: `i(x,y)`. Literaal `p(`t`)` ütleb, et t on tuletatav lausearvutusvalem. Disjunkt 1 formaliseerib *modus ponens*'i reeglit, disjunkt 3 Łukasiewiczi aksioomi ja disjunkt 2 Peirce'i reegli üldkehtivuse eitust.

(1) ¬ `p(x)` ∨ ¬ `p(i(x,y))` ∨ `p(y)`

(2) ¬ `p(i(i(i(a,b),a),a))`

(3) `p(i(i(i(x,y),z),i(i(z,x),i(u,x))))`

Tuletusreeglina kasutame *hüperresolutsioonireeglit*, mis teeb korraga mitu harilikku resolutsioonisammu. Antud näites vastab üks hüperresolutsioonisamm täpselt ühele *modus ponens*'i reegli rakendusele ja koosneb seega kahest järjestikusest hariliku resolutsiooni sammust. Tõestuse leidmiseks kulub mitu sekundit, mille jooksul tõestaja tuletab *ca* 20 000 disjunkti, millest 90% neelatakse.

(3, 1, 3 annavad 4)
`p(i(i(i(i(x,y),i(z,y)),i(y,u)),i(v,i(y,u))))`

(4, 1, 4 annavad 5) `p(i(x,i(i(y,z),i(z,i(y,z)))))`

(4, 1, 3 annavad 6) `p(i(i(i(x,i(y,z)),i(i(u,y),i(v,y))),`
`i(w,i(i(u,y),i(v,y)))))`

(5, 1, 5 annavad 7) `p(i(i(x,y),i(y,i(x,y))))`

(7, 1, 3 annavad 12) `p(i(i(i(x,i(y,x)),y),i(z,y)))`

(6, 1, 12 annavad 15) `p(i(x,i(i(y,z),i(z,z))))`

(15, 1, 15 annavad 20) `p(i(i(x,y),i(y,y)))`

(20, 1, 12 annavad 23) `p(i(i(x,y),i(x,y)))`

(23, 1, 3 annavad 26) `p(i(i(i(x,y),x),i(z,x)))`

(26, 1, 3 annavad 50) `p(i(i(i(x,y),i(y,z)),i(u,i(y,z))))`

(50, 1, 3 annavad 94)
`p(i(i(i(x,i(y,z)),i(u,y)),i(v,i(u,y))))`

(94, 1, 3 annavad 490) `p(i(x,i(i(i(i(y,z),u),z),i(y,z))))`

(490, 1, 490 annavad 498) `p(i(i(i(i(x,y),z),y),i(x,y)))`

(498, 1, 3 annavad 507)
`p(i(i(i(x,y),i(i(x,y),z)),i(u,i(i(x,y),z))))`

(507, 1, 26 annavad 1143) `p(i(x,i(i(i(y,z),y),y)))`

(1143, 1, 1143 annavad 1162) `p(i(i(i(x,y),x),x))`

(1162, 2 annavad tühja disjunkti) `vastuolu.`

17.4.3. Programmide verifitseerimine

Kolmanda näitena vaatame arvutiprogrammi korrektsuse tõestamist.

Kirjutame funktsionaalse programmi `inter`, mis arvutab kahe loendi ühisosa.

Definitsioon 17.18. *Loend* on andmestruktuur, mis ehitatakse induktiivselt tühjast loendist `nil` ja suvalistest termidest — loendi elementidest — paarikonstruktori `c` abil. Loendile `x` elemendi `h` lisamisel saadud uus loend on `c(h,x)`.

Näiteid: arvuloend 1, 2, 5, 15 kirjutatakse `c(1,c(2,c(5,c(15,nil))))`, konstantide loend `e,e,d` kui `c(e,c(e,c(d,nil)))`.

Defineerime kõigepealt funktsionaalse programmi mem, millega saab kontrollida, kas loend sisaldab teatud elementi:

$$\begin{aligned} mem(x, nil) &= FALSE \\ mem(x, c(y,z)) &= \mathit{if}\ \ x = y\ \ \mathit{then}\ \ TRUE \\ &\quad\ \mathit{else}\ \ mem(x,z) \end{aligned}$$

Avaldis $mem(x, y)$ annab tõese resultaadi $TRUE$ siis ja ainult siis, kui loend y sisaldab elementi x; vastasel korral on resultaadiks $FALSE$.

Nüüd saame kirjutada kahe loendi ühisosa arvutava programmi $inter$, mis kasutab funktsiooni mem:

$$\begin{aligned} inter(nil, y) &= nil \\ inter(c(x,y), z) &= \mathit{if}\ \ mem(x,z)\ \ \mathit{then}\ \ c(x, inter(y,z)) \\ &\quad\ \mathit{else}\ \ inter(y,z) \end{aligned}$$

Järgnevas eeldame, et programmi mem korrektsus on juba tõestatud fakt.

Meie eesmärgiks on tõestada, et $inter$ on korrektne, s.t. $inter$ arvutab alati õige resultaadi. Enne kui saame hakata otsima tõestust, peab olema täpselt defineeritud, milline on ühisosa arvutamise *õige resultaat*: peame defineerima ehk *spetsifitseerima* ühisosa mõiste. On lihtne veenduda, et järgmine valem sobib ühisosa olemasolu spetsifikatsiooniks:

$$\forall x \forall y \exists z \forall u((mem(u,x)\ \&\ mem(u,y)) \Leftrightarrow mem(u,z)).$$

Valem ütleb, et suvalise kahe objekti x ja y jaoks on selline objekt z, et mingi objekt u on korraga nii x kui y element parajasti siis, kui ta on z element.

Ühisosa olemasolu valemit modifitseerides saame kergesti valemi, mis väidab, et $inter(x,y)$ annab alati loendite x ja y korrektse ühisosa:

$$\forall x \forall y \forall u((mem(u,x)\ \&\ mem(u,y)) \Leftrightarrow mem(u, inter(x,y))).$$

Viimati toodud valemi tõestamine ongi meie eesmärk.

Järgmise sammuna teisendame funktsionaalsed programmid mem ja $inter$ neile ekvivalentseteks loogikavalemiteks. Parema loetavuse mõttes kirjutame edaspidi literaali $=(t_1, t_2)$ kujul $t_1 = t_2$. Programm mem annab:

$$\forall x(\neg mem(x, nil))\ \&\ (\forall x, y, z(mem(x, c(y,z)) \Leftrightarrow (x = y \vee mem(x,z))).$$

Programmi $inter$ teisendame valemiteks

$$\forall y(inter(nil, y) = nil),$$
$$\forall x, y, z(mem(x,y) \Rightarrow inter(c(x,z), y) = c(x, inter(z,y)))$$

ja

$$\forall x, y, z(\neg mem(x, y) \Rightarrow inter(c(x, z), y) = inter(z, y)).$$

Arve, loendeid ja muid potentsiaalselt lõpmatuid objekte käsitlevate teoreemide tõestamiseks ei piisa enamasti puhtast esimest järku predikaatarvutusest. Põhjuseks asjaolu, et lõpmatu struktuur defineeritakse *induktsiooniprintsiibiga* ning tõestuste juures on seetõttu samuti vaja rakendada induktsiooni. Harilikuks induktsiooni kasutamise valdkonnaks on aritmeetika ning analoogiliselt viimasega kasutatakse induktsiooni teoreemide tõestamisel loendite kohta.

Võimalikke induktsiooniskeeme on lõpmatu arv, ning kui induktsiooni tõestamise juures tarvis läheb, siis läheb sageli tarvis ka mitmesuguseid eraldi tõestatavaid lisalemmasid. Metamatemaatika positsioonilt võib selle asjaolu kohta öelda lihtsalt, et aritmeetika ja loendite teooria *pole normaliseeritavad*. Praktikas tähendab see puhta esimest järku predikaatarvutusega võrreldes suuri lisaraskusi tõestuse otsimisel: tarvis on leida sobiv induktsiooniskeem ning vajalikud lisalemmad. On mitu automaatset teoreemitõestajat, mis püüavad vajalikke induktsiooniskeeme ja lisalemmasid leida, kuid keerukamatel juhtudel on see täisautomaatsüsteemi jaoks väga raske, s.t. otsimine võtab lootusetult kaua aega. Sel puhul kasutatakse interaktiivseid süsteeme, kus kasutaja suunab süsteemi sobiva induktsiooniskeemi ning lisalemmade juurde ja laseb süsteemil alles seejärel automaatselt tegutseda.

Käesoleva teoreemi tõestamiseks piisab siiski kõige lihtsamat tüüpi induktsiooniskeemi, *struktuurse induktsiooni* kasutamisest, samuti ei lähe vaja lisalemmasid, mida peaksime eraldi tõestama hakkama. Seesuguse juhuga saavad automaatsed induktsiooni kasutavad süsteemid kergesti hakkama.

Järgmine reegliskeem on selline struktuurse induktsiooni variant loendite jaoks, mis kasutab induktsiooni jaoks muutujat x_1.

$$\frac{\forall x_2 \ldots x_n.A\{x_1/nil\} \qquad \forall x((\forall x_2 \ldots x_n.A\{x_1/x\}) \Rightarrow (\forall h x_2 \ldots x_n.A\{x_1/c(h, x)\}))}{\forall x_1 \ldots x_n.A}$$

A kohale tuleb paigutada tõestatav valem. Eeldatakse, et muutuja

x_1 asemele võib substitueerida ainult loendeid, s.t. x_1 on loendi tüüpi. Esimene eeldus reegliskeemis on induktsiooni baas: valem A kehtib, kui muutuja x_1 asendada tühja loendiga nil. Teine eeldus reegliskeemis on induktsiooni samm: eeldame, et A kehtib suvalise loendi x jaoks ning järeldame sellest, et A kehtib, kui loendile x lisada suvaline element h.

17.4.4. Induktsiooni baas

Induktsiooni baasjuht on valem

$$\forall y \forall u((mem(u, nil) \,\&\, mem(u, y)) \Leftrightarrow mem(u, inter(nil, y))),$$

mis tuleb tõestada, kasutades programmide mem ja $inter$ teisendust loogikavalemiteks.

Esitame probleemi teisenduse disjunktihulgaks:

(1) `(x = x)`

(2) `(inter(nil,y) = nil)`

(3) `¬ mem(u,y) ∨ (inter(c(u,x),y) = c(u,inter(x,y)))`

(4) `mem(u,y) ∨ (inter(c(u,x),y) = inter(x,y))`

(5) `¬ mem(x,nil)`

(6) `¬ mem(x,c(y,z)) ∨ (x = y) ∨ mem(x,z)`

(7) `¬ (x = y) ∨ mem(x,c(y,z))`

(8) `¬ mem(x,z) ∨ mem(x,c(y,z))`

(9) `¬ mem(su,nil) ∨ ¬ mem(su,sy) ∨`
`¬ mem(su,inter(nil,sy))`

(10) `mem(su,nil) ∨ mem(su,inter(nil,sy))`

(11) `mem(su,sy) ∨ mem(su,inter(nil,sy))`

Disjunktid 2, 3 ja 4 saame programmi $inter$ teisendusest, disjunktid 5, 6, 7 ja 8 programmi mem teisendusest ning disjunktid 9, 10 ja 11 baasjuhu valemi teisendusest. Konstandid `su` ja `sy` saame muutujate u ja y skolemiseerimisel.

Vastuolu tuletame väga lihtsalt:

(2 ja 5, paramodulatsioon: 14) ¬ mem(x,inter(nil,y))

(10 ja 5, resolutsioon: 46) mem(su,inter(nil,sy))

(46 ja 14 annavad tühja disjunkti) vastuolu.

17.4.5. Induktsiooni samm

Induktsiooni sammuks on valem

$$\forall x((\forall y \forall u((mem(u,x)\ \&\ mem(u,y)) \Leftrightarrow mem(u, inter(x,y)))) \Rightarrow$$
$$(\forall h \forall y_1 \forall u_1((mem(u_1, c(h,x))\ \&\ mem(u_1, y_1)) \Leftrightarrow$$
$$mem(u_1, inter(c(h,x), y_1)))),$$

mis tuleb tõestada, kasutades programmide mem ja $inter$ teisendust loogikavalemiteks.

Probleemi teisendus disjunktihulgaks:

(11) (x = x)

(10) (inter(nil,y) = nil)

(1) ¬ mem(u,y) ∨ (inter(c(u,x),y) = c(u,inter(x,y)))

(14) mem(u,y) ∨ (inter(c(u,x),y) = inter(x,y))

(2) ¬ mem(x,nil)

(3) ¬ mem(x,c(y,z)) ∨ (x = y) ∨ mem(x,z)

(4) ¬ (x = y) ∨ mem(x,c(y,z))

(5) ¬ mem(x,z) ∨ mem(x,c(y,z))

(6) ¬ mem(u,sx) ∨ ¬ mem(u,y) ∨ mem(u,inter(sx,y))

(7) mem(u,sx) ∨ ¬ mem(u,inter(sx,y))

(8) mem(u,y) ∨ ¬ mem(u,inter(sx,y))

(15) mem(su1,c(sh,sx)) ∨ mem(su1,inter(c(sh,sx),sy1))

(16) mem(su1,sy1) ∨ mem(su1,inter(c(sh,sx),sy1))

(9) ¬ mem(su1,c(sh,sx)) ∨ ¬ mem(su1,sy1) ∨
¬ mem(su1,inter(c(sh,sx),sy1))

Esimesed disjunktid saame jällegi programmide $inter$ ja mem teisendusest. Viimased kuus disjunkti saame sammuvalemi teisendusest. Konstandid sx, sh, sy1 ja su1 saame muutujate x, h, y_1 ja u_1 skolemiseerimisel.

Saadud disjunktihulgast vastuolu tuletamine on juba küllalt raske ülesanne. Sobiva otsimisstrateegia kasutamisel leiavad nüüdisaegsed tõestajad vastuolu paari minuti jooksul, tuletades üle kümne tuhande disjunkti. Meie kasutatud tõestaja leitud tuletus sisaldab üle kolmekümne hüper- ja paramodulatsioonisammu, tõestuse sügavus on 20 taset. Sellise sügavusega tõestuste leidmiseks ei ole enam praktiliselt võimalik kasutada strateegiat, kus enne järgmise tasemeni liikumist tuletatakse kõik disjunktid antud tasemel. Selle asemel kasutatakse heuristikaid, mis kontsentreerivad tuletuse süsteemile perspektiivsemana näivate disjunktide suunas. Mitu järgneva tuletuse sammu sisaldab mitme reegli järjest rakendamist. Jätame reeglite leidmise lugeja hooleks.

(11, 4 annavad: 17) mem(x,c(x,y))

(14, 9 annavad: 50) ¬mem(su1,c(sh,sx)) ∨ ¬mem(su1,sy1) ∨
¬mem(su1,inter(sx,sy1)) ∨ mem(sh,sy1)

(16, 14 annavad: 67) mem(su1,sy1) ∨ mem(su1,inter(sx,sy1))
∨ mem(sh,sy1)

(16, 1 annavad: 69) mem(su1,sy1) ∨
mem(su1,c(sh,inter(sx,sy1))) ∨ ¬mem(sh,sy1)

(15, 3 annavad: 118) mem(su1,inter(c(sh,sx),sy1)) ∨ su1=sh
∨ mem(su1,sx)

(15, 14 annavad 139) mem(su1,c(sh,sx)) ∨
mem(su1,inter(sx,sy1)) ∨ mem(sh,sy1)

(67, 8 annavad: 160) mem(su1,sy1) ∨ mem(sh,sy1)

(160, 3 annavad: 175) mem(su1,sy1) ∨ mem(x,sy1) ∨
¬mem(sh,c(x,y)) ∨ mem(sh,y)

(175 annab: 181) mem(su1,sy1) ∨ ¬mem(sh,c(su1,x)) ∨
mem(sh,x)

(69, 160 annavad: 500) `mem(su1,sy1) ∨ mem(su1,c(sh,inter(sx,sy1)))`

(500, 3 annavad: 508) `mem(su1,sy1) ∨ su1=sh ∨ mem(su1,inter(sx,sy1))`

(508, 8 annavad: 534) `mem(su1,sy1) ∨ su1=sh`

(534, 181, 17 annavad: 560) `mem(su1,sy1) ∨ mem(sh,x)`

(560, 2 annavad: 568) `mem(su1,sy1)`

(568, 3 annavad: 576) `mem(x,sy1) ∨ ¬mem(su1,c(x,y)) ∨ mem(su1,y)`

(139, 576 annavad: 798) `mem(su1,inter(sx,sy1)) ∨ mem(sh,sy1) ∨ mem(su1,sx)`

(798, 7 annavad: 835) `mem(sh,sy1) ∨ mem(su1,sx)`

(798, 6, 568 annavad: 846) `mem(su1,inter(sx,sy1)) ∨ mem(sh,sy1)`

(835, 5 annavad: 864) `mem(sh,sy1) ∨ mem(su1,c(x,sx))`

(450, 864, 568, 846 annavad: 736) `mem(sh,sy1)`

(4736, 1 annavad: 4740, 4739) `inter(c(sh,x),sy1)= c(sh,inter(x,sy1))`

(118, 4740 annavad: 4748) `mem(su1,c(sh,inter(sx,sy1))) ∨ su1=sh ∨ mem(su1,sx)`

(9, 4740, 568 annavad: 4751) `¬mem(su1,c(sh,sx)) ∨ ¬mem(su1,c(sh,inter(sx,sy1)))`

(4748, 3 annavad: 4789) `su1=sh ∨ mem(su1,sx) ∨ mem(su1,inter(sx,sy1))`

(4789, 7 annavad: 4793) `su1=sh ∨ mem(su1,sx)`

(4793, 4751, 17 annavad: 4813) `¬mem(su1,c(sh,sx)) ∨ mem(su1,sx)`

(4813, 4793, 17 annavad: 4876) `mem(su1,sx)`

(4876, 6, 568 annavad: 4891) `mem(su1,inter(sx,sy1))`

(4876, 5 annavad: 4899) `mem(su1,c(x,sx))`

(4891, 5 annavad: 4922) `mem(su1,c(x,inter(sx,sy1)))`

(4922, 4751, 4899 annavad tühja disjunkti) `vastuolu`.

18. Loogiline programmeerimine ja Horni disjunktid

Programmeerimiskeeli ja -meetodeid saab klassifitseerida mitmel moel. Meie raamatu kontekstis sobib jaotada programmeerimiskeeled kõigepealt kahte gruppi.

Imperatiivsed keeled sobivad samm-sammult, kindlas järjekorras täidetavate algoritmide esitamiseks. Programmid kujutavad endast arvutile antavate käskude jada. Tuntumad imperatiivsed keeled on C, Basic, Pascal, Java, objektorienteeritud keeled ja assemblerkeeled. Teoorias kasutatav *Turingi masin* sobib puhtalt imperatiivse keele näiteks.

Imperatiivsete keelte peamiseks eeliseks on arvuti tegevuse täpse kontrollimise ja suunamise võimaldamine, mis enamasti tagab maksimaalse töökiiruse.

Miinusteks on programmeerimise suur töömahukus — lahenduskäigu kõik detailid tuleb süsteemile esitada — ning suured raskused programmide analüüsimisel, näiteks optimeerimise, verifitseerimise või paralleliseerimise tarvis.

Deklaratiivsed keeled sobivad algoritmi esitamiseks käskude jadast abstraktsemal viisil. Programmeerija ei pruugi alati kõiki algoritmi detaile kirja panna, vaid võib esitada otsitava lahenduse *kirjelduse* ning juba programmi täitmise käigus otsustab süsteem automaatselt, mis viisil täpselt lahendust otsida. Deklaratiivseteks keelteks võib arvata loogilise programmeerimise keeled (näiteks Prolog) ja mitu funktsionaalset keelt (näiteks Haskell). Teoorias kasutatav *lambda-arvutus* on puhtalt funktsionaalse deklaratiivse keele näide.

Deklaratiivsed keeled võimaldavad kirjutada enamikku programme kiiremini ja mugavamalt kui imperatiivsed keeled — programmeerija ei pea kõigi detailide eest hoolt kandma. Tunduvalt lihtsam on ka programmide analüüs, näiteks programmi automaatsel kohandamisel paralleelarvutile, kus programmi täitmise juures töötab samaaegselt hulk protsessoreid.

Peamiseks miinuseks on programmide väiksem töökiirus — deklaratiivne programm ei pruugi küll alati aeglasem olla kui imperatiivne, kuid on seda harilikult siiski. Põhjuseks on siin keele automaattranslaatori väiksem intelligentsus kogenud programmeerijaga võrreldes.

Mitte iga ülesannet ei ole deklaratiivselt mugavam esitada kui imperatiivselt — mõni ülesanne sobib oma iseloomu poolest imperatiivse paradigmaga, mõni deklaratiivsega.

Reaalsed programmeerimiskeeled ei ole peaaegu kunagi puhtalt imperatiivsed või deklaratiivsed. Imperatiivset ja deklaratiivset programmeerimisviisi tuleb käsitleda kui programmeerimise paradigmasid. Konkreetsed programmeerimiskeeled kalduvad rohkem kas ühte või teise paradigmasse, kuid puhtpragmaatilistel kaalutlustel püütakse konkreetsete keelte sisse ehitada võimalusi mitmest paradigmast.

Nüüdisaegsed imperatiivsed keeled, näiteks C, sisaldavad palju deklaratiivse keele elemente, ning vastupidi: deklaratiivne keel Prolog võimaldab soovi korral üpris imperatiivse programmeerimisstiili kasutamist. Traditsioonilised *funktsionaalsed keeled* Lisp,

Scheme ja ML sisaldavad nii imperatiivseid kui deklaratiivseid konstruktsioone.

Imperatiivsed keeled jaotatakse edasi *protseduraalseteks* (C, Pascal jms.) ja *objektorienteeritud* keelteks (SmallTalk, Java jms.). Pikemalt me imperatiivsetel keeltel ei peatu.

Deklaratiivsed keeled jaotatakse *loogilise programmeerimise keelteks* (näide: Prolog), kus otsitavat lahendust kirjeldatakse loogika keeles, ja *funktsionaalse programmeerimise keelteks* (näide: Haskell), kus lahendust kirjeldatakse funktsioonide koguga — ka viimast saab tegelikult käsitleda kui teatud tüüpi loogikasüsteemi.

Peatüki ülejäänud osas tutvustame põgusalt 1970. aastate alguses leiutatud loogilise programmeerimise keelt Prolog, mille nimi on lühend ingliskeelsest väljendist “Programming in logic”. Prologi aluseks on inglise arvutiteadlase Robert Kowalski tähelepanek, et Horni reeglit

$$p \leftarrow q_1, \ldots, q_n$$

saab lugeda nii *protseduraalselt*:

- *Programmi p arvutamiseks tuleb arvutada tema alamprogrammid $q_1, \ldots, q_n$,*

kui ka *deklaratiivselt*:

- *Kui $q_1, \ldots, q_n$ on tõesed, siis on ka p tõene.*

Siiski tuleb märkida, et ei ole mõtet samastada loogilist programmeerimist ja Prologi. Resolutsiooni asemel võib programmeerimiskeele aluseks olla ka mõni teine loogika tuletussüsteem. Küsimus on ainult programmide kirjutamise mugavuses ning täitmise efektiivsuses.

Järgneva materjali esituse juures eeldatakse varasemat tutvumist automaatse teoreemitõestamise peatükiga.

18.1. Prolog

Prolog on esimene ja kasutatavaim loogilise programmeerimise keel. Prologile lisaks on välja töötatud mitu uuemat loogilise programmeerimise keelt ja süsteemi, ning nende arendamine on ulatuslik ja levinud uurimisteema. Siin piirdume ainult standardse Prologi põhimõtete esitamisega. Tuleb arvestada, et Prologis programmeerimise õppimiseks käesoleva peatüki mahust ei piisa.

Prologi põhiidee on nõuda otsitava lahenduse kirjeldamist esimest järku predikaatarvutuse keeles, kusjuures Prologi süsteem sisaldab teatud tüüpi automaatset teoreemitõestajat, mis on võimeline automaatselt otsima ja tuletama lahendust.

Sellegipoolest ei ole Prolog siiski automaatse teoreemitõestamise süsteem: viimast realiseeriv mehhanism on Prologis väga piiratud, spetsiifiline ja loogiliselt mittetäielik, s.t. loogika mõttes eksisteerivad lahendused võivad vahel leidmata jääda.

Prolog on esmajoones programmeerimissüsteem ning sobib eriti hästi ekspertsüsteemide programmeerimiseks. Programmeerijale antakse võimalus suunata Prologi automaatset teoreemitõestajat — nimetame seda edaspidi Prologi otsingumootoriks — nii, nagu programmist soovib. Prologi mootor otsib lahendust lihtsa ning kindlalt määratud otsimismeetodiga — mingisugust programmeerijale ennustamatut omaloomingut Prologi mootorile ei lubata.

Konkreetsemalt on Prologi mootori aluseks resolutsioonimeetodi teatud mittetäielik strateegia.

18.1.1. Horni disjunktid ja otsingumootor

Prologi programm on *Horni disjunktide* kogu. Horni disjunkt on selline disjunkt, mis ei sisalda rohkem kui ühte positiivset literaali.

Prologi kirjaviisi järgi on kõik suure algustähega sõnad disjunktides *muutujad* ning väikese algustähega sõnad kas *konstandid, predikaadid* või *funktsioonisümbolid*. Prolog sisaldab konstantidena ka numbreid ja sõnesid.

Prologi kontekstis jaotatakse kõik programmi moodustavad disjunktid ehk *laused* järgmiselt.

Faktid on ühestainsast positiivsest literaalist koosnevad disjunktid.
Näiteks: `isa(jaan,peeter)`.

Reeglid on ühest positiivsest literaalist ja ühest või mitmest negatiivsest literaalist koosnevad disjunktid.

Traditsiooniliselt kirjutatakse positiivne literaal disjunkti algusesse, seejärel tuleb eraldussümbol `:-` ning lõpuks komadega eraldatud negatiivsed literaalid, mille eest eituse sümbol on ära jäetud. Nagu fakti nii ka reegli lõpetab alati punkt. Eraldussümbolit `:-` võib lugeda kui vasakule suunatud implikatsiooni: $\Leftarrow$. Näitena toome isapoolse vanaisareegli:

```
vanaisa(X,Z) :- isa(X,Y), isa(Y,Z).
```

Päringud on ühest või mitmest negatiivsest literaalist koosnevad disjunktid.

Päring ei ole niivõrd Prologi programmi osa kui otsingu käivitamise käsk. Prologi süsteemi töö sisuks on esitatud päringule vastuse leidmine, kasutades varem antud faktide ja reeglite andmebaasi. Korraga võib esitada vaid ühe päringu.

Kuna Prologis kirjutatud andmebaas ise puhtalt negatiivseid disjunkte ei sisalda, siis ei saa Prologi keeles esitada otse negatiivset informatsiooni, näiteks, et keegi Mihkel *ei ole* Peetri isa. Prologis programmeerimisel käsitletakse *mittetuletatavust* kui eitust, kuigi loogika seisukohalt on need kaks asjaolu täiesti erinevad. Prologi eituse-kontseptsiooni nimetatakse *suletud maailma eelduseks*, s.t. tõene on ainult see, mida antud Prologi programmist (kui suletud maailmast) tuletada saab.

Vaatleme esimese näitena jällegi sugulussidemete andmebaasi, konkreetselt niisugust Prologi programmi:

```
isa(jaan,peeter).
isa(jaan,martin).
isa(martin,veiko).
isa(riivo,leo).
ema(leena,leo).
```



```
vanaisa(X,Z) :- isa(X,Y), isa(Y,Z).
vanaisa(X,Z) :- isa(X,Y), ema(Y,Z).
```

Päringule, kas `isa(riivo,martin)` on antud andmebaasist tuletatav, vastab otsingumootor eitavalt (`?-` alustab päringuid, järgmisel real trükitu on Prologi vastus):

```
?- isa(riivo,martin).
no
```

Lahendust otsib Prologi mootor järgmiselt: kõik andmebaasis olevad laused vaadatakse järjest läbi (esimesega alustades ja viimasega lõpetades) ning iga lause esimest literaali püütakse päringuliteraaliga unifitseerida. Kui see ei õnnestu, vastatakse päringule eitavalt.

Päringule, kas `isa(riivo,leo)` on antud andmebaasist tuletatav, vastab otsingumootor jaatavalt:

```
?- isa(riivo,leo).
yes
```

Saime jaatava vastuse, kuna otsingumootor unifitseeris päringuliteraali andmebaasis oleva faktiga `isa(riivo,leo)`.

Päringud võivad sisaldada muutujaid: sellisel juhul leiab Prologi mootor *lahenduse*, s.t. muutuja asemele substitueeritava termi:

```
?- isa(jaan,X).
X=peeter
```

Kui me ei ole rahul esimese leitud lahendusega, võime instrueerida otsingumootorit uusi lahendusi otsima, s.t. püüdma unifitseerida päringut järgmiste andmebaasis olevate lausetega. Selleks trükime lahenduse järele semikooloni, ning Prolog esitab kas järgmise lahenduse või — kui lahendusi enam pole — vastab eitavalt.

```
?- isa(jaan,X).
X=peeter;
X=martin;
no
```

Mõistagi pakub programmeerimise juures peamist huvi *reeglite* rakendamine. Vaatame, kuidas reageerib otsingumootor päringule, kas Jaan on kellegi vanaisa:

```
?- vanaisa(jaan,X).
```

Esimene andmebaasi lause, millega päringuliteraal unifitseerub, on reegel `vanaisa(X,Z) :- isa(X,Y), isa(Y,Z).` Kuna enne unifitseerimist on tarvis muutujad ümber nimetada, nimetatakse päringuliteraalis muutuja `X` ümber muutujaks `X1`. Substitutsiooniks saame siis `{X=jaan, Z=X1}`. Reegel annab meile kaks *alamülesannet* ehk *alampäringut*

```
isa(jaan,Y), isa(Y,X1),
```

mis tuleb rahuldada. Otsingumootor asub järjekorras esimesele alampäringule lahendusi otsima, vaadates selleks andmebaasi algusest lõpuni läbi — samamoodi, kui oleks reageeritud kasutaja antud päringule.

Esimene alampäringuga `isa(jaan,Y)` unifitseeritav lause on fakt `isa(jaan,peeter)` ning substitutsiooniks saame `Y=peeter`. Kuna tegu oli faktiga, siis uusi alampäringuid ei teki ja jääb üle lahendada saadud substitutsiooni rakendamise järel konkretiseeritud teine alampäring `isa(peeter,X1)`. See alampäring aga ei unifitseeru ühegi lausega meie andmebaasis.

Niisuguses olukorras *pöördub otsingumootor tagasi* viimati leitud lahenduse juurde, heidab selle kõrvale ja otsib andmebaasist edasisi lahendusi. Viimase lahenduse saime esimese alampäringu `isa(jaan,Y)` unifitseerimisel faktiga `isa(jaan,peeter)`: niisiis heidab otsingumootor selle lahenduse kõrvale ning leiab järgmise lause, millega saab alampäringut unifitseerida. Selleks on `isa(jaan,martin)`, mis annab uueks substitutsiooniks `Y=martin`. Jällegi, uusi alampäringuid ei tekkinud ja jääb üle lahendada saadud substitutsiooni rakendamise järel konkretiseeritud teine alampäring `isa(martin,X1)`. See päringuliteraal unifitseerub andmebaasis oleva faktiga `isa(martin,veiko)`, andes substitutsiooniks `X1=veiko`. Kuna rohkem lahendamist vajavaid alampäringuid ei ole, siis on esialgsele päringule vastus leitud ja Prolog trükib:


```
X=veiko
```

Vaatame edasi, mis juhtub, kui vastame lahendusele semikooloni trükkimisega, s.t. uue lahenduse nõudmisega. Prolog reageerib viimase alampäringu viimati leitud lahenduse kõrvaleheitmisega. Kuna viimast päringuliteraali `isa(martin,X1)` ei saa unifitseerida enam ühegi fakti ega reegliga pärast fakti `isa(martin,veiko)`, heidab otsingumootor kõrvale ka eelmisele alampäringule `isa(jaan,Y)` leitud lahenduse ning hakkab otsima järgmist. Kuna ükski faktile `isa(jaan,martin)` järgnev lause päringuliteraaliga ei unifitseeru, heidetakse taas kõrvale järjekorras viimane leitud lahendus, s.o. meie esialgse päringuga `vanaisa(jaan,X)` unifitseeritud reegel `vanaisa(X,Z) :- isa(X,Y), isa(Y,Z).` Järgmisena saab päringuliteraaliga `vanaisa(jaan,X)` unifitseerida emapoolse vanaisareeglit `vanaisa(X,Z) :- isa(X,Y), ema(Y,Z)`, mispeale saame kaks alampäringut `isa(jaan,Y), ema(Y,X1)`. Kumbki esimese alampäringu lahendus (s.o. `Y=peeter` ja `Y=martin`) ei võimalda leida lahendusi teisele alampäringule. Kõigi võimaluste ammendumise järel trükib süsteem meile eitava vastuse: `no`.

18.1.2. Lahenduste ühesus

Resolutsioonimeetodit käsitlevas peatükis vaatlesime spetsiaalse *lahendusliteraaliga* lahenduste otsimise meetodit. Prologis kasutatav meetod on sisuliselt lahendusliteraali kasutamisega ekvivalentne, erinevus on vaid tehnilises teostuses. Lahendusi käsitledes juhtisime tähelepanu asjaolule, et klassikalise loogika kasutamisel on teatud päringutel ainult *mitteühesed lahendused*. Näiteks disjunktide hulgast

(1) $p(a) \vee p(b)$

(2) $\neg\, p(x) \vee lahendus(x)$

saab tuletada ainult mitteühese lahenduse: $lahendus(b) \vee lahendus(a)$.

Erinevalt klassikalise loogika üldjuhust on kõigil Prologis esitatud päringutel alati *ühene* lahendus, kuigi selliseid üheseid lahendusi võib päringul olla palju: põhjuseks on Horni disjunktide kasutamine. Prologis ei ole võimalik esitada näiteks disjunkti `p(a)` $\vee$ `p(b)`, kuna see ei ole Horni disjunkt.

Suhteliselt lihtsalt saab näidata, et Horni disjunktidega piirdumise korral langevad klassikaline ja intuitsionistlik loogika kokku. Niisiis on iga Prologi otsingumootori leitud tuletus nii klassikaliselt kui intuitsionistlikult korrektne.

Ülesanded

1. *Näidake, et lausearvutuse korral saab Horni disjunktide hulga mittevasturääkivust kontrollida polünomiaalse ajaga.

18.1.3. Loendid

Prolog lubab loendite ehitamiseks kasutada tühja loendit `[]` ja loendile elemendi lisamist ehk paarikonstruktorit, mida tähistab punkt. Loendile `x` elemendi `h` lisamisel saadud uus loend on term `.(h,x)`. Üheelemendiline, konstanti `a` sisaldav loend on `.(a,[])`. Arvudest 1, 2 ja 3 koosnev loend esitatakse termina `.(1,.(2,.(3,[])))`.

Mugavuse mõttes lubab Prolog kasutada loendite jaoks ka teistsugust kirjaviisi, mis otsingumootori käivitamise eel tõlgitakse programmeerija jaoks nähtamatult sisemiseks, paarikonstruktorit kasutavaks kirjaviisiks.

Mugavamas kirjaviisis kirjutatakse loendi elemendid nurksulgude vahele: `[a,b,c,c]` on neljast konstandist koosnev loend. `[X|Y]` tähistab loendit, mille esimene element on `X` ja esimese elemendi eemaldamise järel allesjääv loendi *saba* on `Y`.

Kirjutame nüüd programmi `member`, mis kontrollib, kas esimene argument kuulub teiseks argumendiks olevasse loendisse:

```
member(X,[X|Z]).
member(X,[Y|Z]) :- member(X,Z).
```

Predikaati `member` defineerivad üks fakt ja üks reegel. Fakt ütleb, et kui `X` on loendi esimene element, siis on `X` loendi element.

Reegel ütleb, et kui `X` on loendi saba element, siis on ta loendi element.

Vaatame, kuidas käsitleb Prolog päringut

```
?- member(10,[20,30,10,40]).
```

Päringuliteraal antud faktiga ei unifitseeru, küll aga unifitseerub ta antud reegliga, andes alampäringu `member(10,[30,10,40])`. Viimane unifitseerub jällegi ainult reegliga, andes alampäringu `member(10,[10,40])`, mis unifitseerub faktiga. Rohkem alampäringuid ei olnud ja süsteem trükib positiivse vastuse: `yes`.

Teise näitena kirjutame programmi `append`, mis liidab kaks loendit kokku. Predikaadi esimene ja teine argument on liidetavad loendid ning kolmas argument on kokkuliitmise resultaat. Näiteks kehtib: `append([a,b],[1,2,3],[a,b,1,2,3])`.

```
append([],Z,Z).
append([X|Y],Z,[X|R]) :- append(Y,Z,R).
```

Järgmine päring annab täpselt ühe lahenduse:

```
?- append([a,b],[c,d],L).
L=[a,b,c,d];
no
```

Predikaati `append` saab kasutada ka tagurpidisuunas: anname ette juba kokkuliidetud loendi ja otsingumootor genereerib kõik võimalikud lähteloendid:

```
?- append(X,Y,[a,b,c,d]).
X=[], Y=[a,b,c,d];
X=[a], Y=[b,c,d];
X=[a,b], Y=[c,d];
X=[a,b,c], Y=[d];
X=[a,b,c,d], Y=[];
no
```

Predikaadi `append` tagurpidist funktsioneerimist saab kasutada loendi kõiki *permutatsioone* genereeriva predikaadi defineerimiseks:

```
permutation(L,[H|T]) :-
   append(V,[H|U],L),
   append(V,U,W),
   permutation(W,T).
permutation([],[]).

?- permutation([a,b,c],X).
X=[a,b,c];
X=[a,c,b];
X=[b,a,c];
X=[b,c,a];
X=[c,a,b];
X=[c,b,a];
no
```

Viimase näitena defineerime efektiivset *sorteerimisalgoritmi*, *quicksort*'i realiseeriva predikaadi. Eeldame, et meil on antud sorteerimisel kasutatavat järjestust defineeriv predikaat `suuremvordne(X,Y)`.

```
qsort([H|T],S) :-
   split(H,T,A,B),
   qsort(A,A1),
   qsort(B,B1),
   append(A1,[H|B1],S).
qsort([],[]).

split(H,[A|X],[A|Y],Z) :-
   suuremvordne(A,H), split(H,X,Y,Z).
split(H,[A|X],Y,[A|Z]) :-
   suuremvordne(H,A), split(H,X,Y,Z).

?- qsort([5,6,1,7,5,3,4,9,0],X).
X=[9,7,6,5,5,4,3,1,0]
```


Ülesanded

2. Defineerige predikaat `slowsort`, mis genereerib loendi järjestamiseks selle kõikvõimalikud permutatsioonid. Kuidas saab seda programmi efektiivsemaks muuta?
3. Konstrueerige predikaat `reverse`, mis pöörab loendi elementide järjestuse ringi:

   ```
   ?- reverse([1,3,2],[2,3,1]).
   ```

18.1.4. Aritmeetika

Harilik täisarvude ja ujupunktarvudega aritmeetika on Prologi sisse ehitatud. Arvutuste tegemiseks kasutatakse spetsiaalset predikaati `is`, mis analoogiliselt harilike programmeerimiskeeltega leiab aritmeetikaavaldise väärtuse ja omistab selle muutujale:

```
?- X is 3+2.
X=5
?- X is 4*2, Y is X+X.
X=8, Y=16
```

Predikaadil `is` ei ole päris selget loogilist tähendust, tegu on pigem programmeerimise hõlbustamiseks kasutatava spetsiaalmeetodiga. Kui parempoolne argument on arvutatav aritmeetikaavaldis, on `is` loogika mõttes siiski korrektne: `is` realiseerib standardseid aritmeetikareegleid, mis töökiiruse huvides ei ole Prologis avalikult aksiomatiseeritud, vaid süsteemi sisse ehitatud.

Aritmeetikatehteid kasutame näiteks faktoriaali arvutamiseks:

```
fact(0,1).
fact(N,R) :-
   N > 1, N1 is N-1, fact(N1,R1), R is N*R1.
```

Selline programmeerimisstiil sarnaneb traditsioonilise imperatiivse programmeerimisega. Prologi aritmeetikaoperaatorite spetsiifilise iseloomu tõttu saab faktoriaalipredikaati kasutada ainult ühes suunas: esimene argument peab olema arv.

```
?- fact(4,X).
X=24.
```

Prolog *ei luba* kirjutada faktoriaaliprogrammi loogika mõttes loomulikumal viisil:

```
fact(0,1).
fact(N,N*R1) :- N > 1, fact(N-1,R1).
```

Viimane reegel on küll süntaktiliselt korrektne, kuid Prolog ei arvuta automaatselt aritmeetikaavaldiste väärtust: tarvis on kasutada spetsiaalpredikaati `is`.

Nagu näidetest paistab, lubab Prolog aritmeetikaavaldiste juures kasutada harilikku *infikskuju*, mis sisemiselt teisendatakse standardseteks *prefikskujul* termideks ja predikaatideks: `2+3` on sisemises esituses `+(2,3)` ning `X > 1` on seesama, mis `>(X,1)`.

Prolog sisaldab predikaadile `is` sarnast predikaati `=`, mis unifitseerib oma vasakpoolse argumendi parempoolsega, püüdmata seejuures arvutada parempoolse argumendi aritmeetilist väärtust. Tuleb tähele panna, et Prologi `=` ei ole tõeline võrduspredikaat. Sisuliselt on ta aksiomatiseeritud ainult faktiga `X=X` ning ei sümmeetria, transitiivsus ega asendusreeglid pole selle predikaadi jaoks aksiomatiseeritud.

18.1.5. Sisend/väljund

Peale aritmeetikaoperaatorite sisaldab Prolog veel muidki spetsiaalsete loogikaväliste omadustega operaatoreid ja predikaate. Sisend/väljund-predikaadid võimaldavad kasutada Prologi teksti trükkimisel, arvutigraafika, mängude jms. tüüpiliste programmeerimisülesannete juures. Näiteks trükib predikaat `write` välja talle argumendiks antud termi ja on seejärel kohe rahuldatud. Järgmine mittepeatuv programm trükib välja täisarvude lõputu rea:

```
arvurida(X) :-
   write(X), X1 is X+1, arvurida(X1).
?- arvurida(1).
```


18.1.6. Prologi mittetäielikkus ja piiratud korrektsus

Nagu öeldud, on Prologi otsimismootor üks resolutsioonimeetodi mittetäielik strateegia. Vaatame näidet, kus Prologi otsingumootor tsükeldub ja loogika mõttes eksisteerivat lahendit ei leia. Defineerime triviaalse predikaadi `sama`, mis kontrollib, kas esimene argument on seesama, mis teine. Lisame defineerivale faktile veel kommutatiivsusreegli:

```
sama(X,Y) :- sama(Y,X).
sama(X,X).
```

Kui nüüd esitada päring `?- sama(a,a)`, siis unifitseerib otsingumootor päringu kõigepealt reegliga, tekitades alampäringu `sama(a,a)`, mille lahendamiseks kasutatakse jällegi reeglit, mis tekitab taas sellesama alampäringu jne. lõpmatuseni. Erinevalt täielikest resolutsioonimeetodi strateegiatest ei jõua Prolog kunagi faktini `sama(X,X)`, mis annaks kohese positiivse lahenduse.

Lisaks mittetäielikkusele ei ole hulk Prologi realisatsioone ka loogika mõttes korrektsed. Põhjuseks on unifitseerimisalgoritmi lihtsustamine programmi töökiiruse huvides. Teatavasti ei ole võimalik unifitseerida muutujat sedasama muutujat sisaldava funktsioonaaltermiga, näiteks y ja $f(y)$. Kuna muutuja sisaldumise kontroll võtab aega, jätab mitu Prologi realisatsiooni selle lihtsalt ära ning käitub järgmiselt:

```
seesama(X,X).
katse :- seesama(Y,f(Y)).
?- katse.
yes
```

Programmeerijalt eeldatakse, et ta niisuguste olukordade tekkimist lihtsalt ei võimalda. Sageli sisaldavad Prologi realisatsioonid lülitit, millega saab valida, kas süsteem kontrollib muutuja sisaldumist või ei.

Praktilise programmeerimise seisukohalt pole ei Prologi mittetäielikkus ega piiratud korrektsus oluliseks puuduseks.

18.1.7. Otsingu blokeerimine: lõikepredikaat

Prolog sisaldab spetsiaalset loogikavälist *lõikepredikaati*, millega programmeerija saab otsingumootorit juhtida, blokeerides uute lahenduste otsimist. Lõikepredikaati (inglise keeles *cut*) tähistab hüüumärk `!` ning seda kasutatakse järgmistel juhtudel:

- Programmeerija teab, et päringul saab olla vaid üks lahendus ning uute lahenduste otsimine on mõttetu: sel juhul suurendab edasise otsingu blokeerimine programmi töökiirust, kuid ei muuda saadavat tulemust.
- Programmeerijale piisab ühestainsast lahendusest.
- Ainult esimene lahendus on korrektne, teised tuleb elimineerida: sellisel juhul muudab lõikepredikaat programmi sisu.

Esimese juhu näiteks kasutame predikaati `append`.

```
append([],Z,Z).
append([X|Y],Z,[X|R]) :- append(Y,Z,R).
```

Oletame, et soovime `append`'i kasutada ainult sel juhul, kui esimesed kaks kokkuliidetavat loendit on antud ja meid huvitab nimelt kokkuliitmise resultaadi leidmine. Päringule `append([],[a,b,c],[a,b,c])` vastamisel kasutab otsingumootor esialgu fakti. Kui aga hiljem nõutakse uute lahenduste otsimist, püüab otsingumootor kasutada reeglit, kuigi see alati ebaõnnestub. Lisame definitsiooni lõikepredikaadi:

```
append([],Z,Z) :- !.
append([X|Y],Z,[X|R]) :- append(Y,Z,R).
```

Esimese lahenduse leidmise juures lahendatakse nüüd lõikepredikaat, mis keelab antud päringule uute lahenduste otsimise, tekitades olukorra, nagu poleks järgmisi defineerivaid lauseid enam olemas.

Lõikepredikaadiga modifitseeritud `append` ei ole enam kahesuunalise toimega. Päringule `append(X,Y,[a,b,c])` leitakse ainult esimene lahendus, kõik edasised on blokeeritud:


```
?- append(X,Y,[a,b,c]).
X=[], Y=[a,b,c];
no
```

Järgmiseks vaatame programmi, mis arvutab kõigi arvude summa, alates ühest kuni etteantud esimese argumendini:

```
summa(1,1) :- !.
summa(X,R) :- X1 is X-1, summa(X1,R1), R is R1+X.
```

Siin takistab lõikepredikaat definitsiooni teise lause — üldjuhu reegli — ekslikku kasutamist baasjuhu jaoks. Tõepoolest, kui siin lõikepredikaati poleks, siis uute lahendite otsimisel päringule `summa(1,Y)` kasutataks üldreeglit ja programm tsükelduks!

Konkreetselt keelab lõikepredikaat uute lahenduste otsimist *talle eelnenud* lauseosa jaoks, esimene, s.o. defineeritav literaal kaasa arvatud. Näide:

```
p1(X,Y) :- p2(X), !, p3(Y).
p1(X,Y) :- p4(X).
p2(a).
p2(b).
p3(b).
p4(a).
p5(a).
p5(b).

?- p1(a,a).
no
?- p5(X), p1(X,X).
X=b;
no
?- p1(U,V).
U=a, V=b;
no
```

Esimese päringu puhul leitakse alampäringu `p2(X)` jaoks lahendus `p2(a)`, kuid `p3(a)` jaoks lahendust pole. Lõikepredikaat keelab uute lahenduste otsimise nii `p2(X)` kui ka `p1(a,a)` jaoks.

Teise päringu juures blokeeritakse samamoodi `p2(X)` ja `p1(a,a)` teise lahenduse otsimine, kuid `p5(X)` teine lahendus `X=b` annab uue alampäringu `p1(b,b)`, mille jaoks õnnestub lahendada esimese reegli alampäring `p3(b)`.

Kolmanda päringu juures blokeeritakse pärast esimese lahenduse leidmist edasiste lahenduste otsimine nii `p2(X)` kui ka `p1(U,V)` jaoks.

Lõikepredikaadiga sarnane spetsiaaloperaator on `not`. Päring `not(P)` on rahuldatud siis ja ainult siis, kui otsingumootor ei suuda rahuldada päringut `P`. Vaatamata näilisele sarnasusele loogilise eitusega on siin tegu hoopis millegi muuga, *suletud maailma eelduse* kasutamisega. Harilikku loogilist eitust Prologi programmid sisaldada ei saa. Operaatorit `not` saab Prologis defineerida selliste loogikaväliste spetsiaalvahenditega:

```
not(P) :- call(P), !, fail.
not(P).
```

Esimeses lauses põhjustab `call(P)` termi `P` esitamise päringuna, ning `fail` on päring, mida rahuldada ei saa.

Kokkuvõtteks. Lõikepredikaadi kasutamist saab soovi korral alati vältida, kuid enamasti muudab see programmeerimise tunduvalt töömahukamaks — tuleb lisada palju erijuhtude kontrolle — ja programmid vähem efektiivseks. Lõikereegli süstemaatilise rakendamisega on aga võimalik otsingumootor pea täielikult blokeerida ning kasutada Prologi sisuliselt funktsionaalse programmeerimiskeelena, milles lisaks on olemas ka mõned imperatiivse programmeerimise vahendid.

18.1.8. Sümbolaritmeetika

Pisut suurema näitena kirjutame programmi, mis leiab avaldise *tuletise* ja seejärel lihtsustab resultaati. Seejuures ei tegelda numbriliste avaldistega, vaid sümbolkujul antud matemaatikavalemitega. Tuletise leidmiseks kasutame järgmisi tuntud reegleid:

- $dc/dx \to 0$
- $dx/dx \to 1$
- $d(-U)/dx \to -(dU/dx)$
- $d(U+V)/dx \to dU/dx + dV/dx$
- $d(U-V)/dx \to dU/dx - dV/dx$
- $d(c*U)/dx \to c*dU/dx$
- $d(U*V)/dx \to U*dV/dx + V*dU/dx$
- $d(U^c)/dx \to c*U^{c-1}*dU/dx$
- $d(log_e U)/dx \to U^{-1}*dU/dx$

Defineerime predikaadi `d`; esimesest argumendist võetakse tuletis teise argumendi järgi ning kolmas argument on resultaadiks olev avaldis.

```
d(X,X,1) :- !.
d(C,X,0) :- atomic(C).
d(~(U),X,~(A)) :- d(U,X,A).
d(U+V,X,A+B) :- d(U,X,A), d(V,X,B).
d(U-V,X,A-B) :- d(U,X,A), d(V,X,B).
d(C*U,X,C*A) :-
   atomic(C), not(C = X), d(U,X,A), !.
d(U*V,X,B*U+A*V) :- d(U,X,A), d(V,X,B).
d(U/V,X,A) :- d(U* ^(V,-1),X,A).
d(^(U,V),X,V*W* ^(U,V-1)) :-
   atomic(V), not(C = X), d(U,X,W).
d(log(U),X,A* ^(U,-1)) :- d(U,X,A).
```

Kasutasime siin Prologi sisseehitatud spetsiaalpredikaati `atomic`, mis on rahuldatud siis ja ainult siis, kui tema argumendiks on konstant ehk atomaarne term, mitte aga funktsionaalterm. Nüüd näiteks:

```
?- d(x+1,x,X).
X=1+0
?- d(x*x-2,x,X).
X=x*1+1*x-0
```

Tuletise programmi resultaadid on lihtsustamata avaldised. Kirjutame predikaadi `simp` avaldiste lihtsustamiseks, mis kasutab predikaadiga `s` moodustatud lihtsustamisreeglite andmebaasi:

```
simp(E,E) :- atomic(E), !.
simp(E,F) :-
   E =.. [Op,La,Ra],
   simp(La,X), simp(Ra,Y), s(Op,X,Y,F), !.
simp(E,F) :-
   E =.. [Op,La,Ra],
   simp(La,X), simp(Ra,Y), F =.. [Op,X,Y], !.
simp(E,E).

s(+,X,0,X).
s(+,0,X,X).
s(+,X,Y,Z) :- integer(X), integer(Y), Z is X+Y.
s(+,X,Y,X+Y).

s(*,X,0,0).
s(*,0,X,0).
s(*,1,X,X).
s(*,X,1,X).
s(*,X,Y,Z) :- integer(X), integer(Y), Z is X*Y.
s(*,X,Y,X*Y).
```

Prologi sisseehitatud spetsiaalpredikaat `=..` lammutab suvalise termi koostisosadeks, alustades funktsionaalsümbolist ja jätkates argumentidega. Antud juhul lammutatakse ainult kahe argumendiga terme. Spetsiaalpredikaat `integer` on rahuldatud ainult juhul, kui argumendiks on arv. Defineerime lihtsustava tuletise-predikaadi ja esitame talle uuesti ülalesitatud päringud:

```
dsimp(X,Y,Z) :- d(X,Y,Z1), simp(Z1,Z).
```



```
?- dsimp(x+1,x,X).
X=1
?- dsimp(x*x-2,x,X).
X=x+x-0
```

Lihtsustamispredikaadi täiustamiseks (näiteks lahutamistehte jaoks) tuleb lihtsalt lisada lauseid predikaati `s` defineerivasse andmebaasi.

Ülesanded

4. Kirjutage programm, mis koostab ja kontrollib järgmise grammatika lauseid:

   <Sentence> ::= <Subject><Verbphrase>

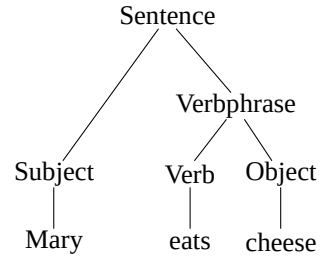
   <Subject> ::= John | Mary

   <Verbphrase> ::= <Verb><Object>

   <Verb> ::= eats | drinks | likes

   <Object> ::= wine | cheese | soup | chips | beer | fruits | fish | juice

   Näiteks lausele “Mary eats cheese” vastab süntaksipuu



5. Erinevad numbrid on asendatud erinevate tähtedega. Lahendage Prologi abil võrrand

$$\begin{array}{rr} & \text{SEND} \\ + & \text{MORE} \\ \hline & \text{MONEY} \end{array}$$

19. Funktsionaalne programmeerimine ja võrdussüsteemid

Funktsionaalses keeles kirjutatakse programme (matemaatiliste) funktsioonide defineerimise teel, määramata seejuures täpselt, mis strateegia järgi funktsiooni resultaati tuleb arvutada. Funktsioonile võib anda argumendiks teisi funktsioone ja funktsiooni arvutamise resultaadiks võib samuti olla funktsioon.

Defineerime näiteks faktoriaali funktsiooni `fakt` ja funktsiooni `map`, mis rakendab oma esimest, funktsionaalset argumenti teiseks argumendiks oleva loendi igale elemendile. Loend `[h|t]` esitatakse sisemiselt kui harilik esimest järku term $.(h, t)$.

```
     fakt(x)  =  if x=0 then 1 else x*fakt(x-1).
   map(f,[])  =  [].
map(f,[h|t])  =  [f(h) | map(f,t)].
```

Avaldise `map(fakt,[3,5,0])` väärtuseks on `[6,120,1]`.

Funktsionaalne programmeerimine on ajalooliselt esimene deklaratiivse programmeerimise viis ning enamik nüüdisaegseid programmeerimiskeeli — C, Pascal, Ada jne. — on funktsionaalse programmeerimise meetoditest mõjustatud.

Funktsionaalseid keeli saab jämedalt jagada kahte liiki: puhtad ja ebapuhtad. Puhtas keeles — Haskell, Hope, Miranda, FP — ei ole programmeerijal peale funktsioonide defineerimise ja sisseehitatud baasfunktsioonide (aritmeetika, loendid jms.) mingeid lisavahendeid — kõik kõrvalefektid on keelatud. Erinevalt imperatiivsetest keeltest, nagu C, Basic ja Pascal, ei luba puhas funktsionaalne keel muutujatele väärtusi omistada. Ainus efekt, mida funktsiooni rakendamine argumentidele annab, on resultaadi leidmine. Ebapuhtad funktsionaalsed keeled — ML, Lisp, Scheme — kombineerivad puhaste funktsionaalsete keelte mehhanisme imperatiivsete mehhanismidega.

Järgnevas ei tegele me konkreetsete funktsionaalsete keelte ega neis programmeerimise meetoditega. Vaatleme hoopis funktsionaalse programmeerimise teooria vundamendiks olevaid loogikasüsteeme, nende põhiprintsiipe ja -omadusi.

19.1. Deduktsioon ja reduktsioon: termiteisendussüsteemid

Funktsionaalseid programme saab käsitleda kui väga piiratud loogikakeeles esitatud valemeid. Konkreetsemalt, funktsionaalne programm on vaadeldav hulga üksikute võrdusliteraalide konjunktsioonina. Nimetame sellist valemiklassi edaspidi *võrdussüsteemide* klassiks.

Üks tüüpiline probleem, millest me oleme huvitatud, on küsimus, kas mingist ühikvõrduste hulgast E järeldub kahe termi t_1 ja t_2 võrdsus. Üldjuhul saame niisuguse probleemi lahendada ainult *deduktsiooni* teel, s.o. otsides termide võrduse tuletust.

Vaatame uuesti teoreemitõestamise peatükis antud näidet rühmateooriast. Olgu $R_=$ rühma defineerivate võrduste hulk:

(1) $1 * x = x$

(2) $x^{-1} * x = 1$

(3) $(x * y) * z = x * (y * z)$

Paremaks loetavuseks esitame võrdused harilikus matemaatilises kirjaviisis. Predikaatarvutuse standardses kirjaviisis näeb näiteks

võrdusreegel (2) välja kui literaal $=(*(inv(x),x),1)$. Tõestame taas, et $R_= \Rightarrow (x * x^{-1} = 1)$. Selleks *teisendame* vasakpoolse termi $x * x^{-1}$ parempoolseks termiks 1. Teisenduse juures tuleb võrdusreegleid 1–3 kasutada nii suunas vasakult paremale kui paremalt vasakule.

Kirjutame teisendussammude vahele, millist reeglit ja mis suunas rakendasime:

$$
\begin{aligned}
& x * x^{-1} \overset{1\leftarrow}{=} \\
& 1 * (x * x^{-1}) \overset{2\leftarrow}{=} \\
& ((x^{-1})^{-1} * x^{-1}) * (x * x^{-1}) \overset{3\rightarrow}{=} \\
& (x^{-1})^{-1} * (x^{-1} * (x * x^{-1})) \overset{3\leftarrow}{=} \\
& (x^{-1})^{-1} * ((x^{-1} * x) * x^{-1}) \overset{2\rightarrow}{=} \\
& (x^{-1})^{-1} * (1 * x^{-1}) \overset{1\rightarrow}{=} \\
& (x^{-1})^{-1} * x^{-1} \overset{2\rightarrow}{=} \\
& 1.
\end{aligned}
$$

Kuna võrdusreegleid tuleb rakendada mõlemas suunas ja ühelgi hetkel pole *a priori* selge, millist reeglit ja millisele alamtermile just rakendada tuleb, on teisendusahela leidmine küllalt vaevanõudev üritus. Kui teisendusahelat ei eksisteeri (s.t. termid ei ole võrdsed) siis ahela otsing üldjuhul ka ei peatu, s.t. meil ei ole algoritmi, millega saaks alati *lahendada* küsimuse kahe termi võrdsusest.

Ideaalis sooviksime vaevarikka deduktsiooni asemel süsteemis $R_=$ kasutada meetodit, mis oleks kiire, lihtne ja alati peatuks, s.o. lahendaks suvalise kahe termi võrdsuse küsimuse. Järgnevas näitame, et süsteemi $R_=$ jaoks taoline meetod tõepoolest eksisteerib. Veel enam, analoogiline efektiivne lahendusalgoritm eksisteerib mitte ainult süsteemi $R_=$, vaid mitme teisegi (kuid mitte kõigi) võrdussüsteemi jaoks.

Modifitseerime prooviks süsteemi $R_=$, lubades teha teisendusi ainult vasakult paremale. Saame ühesuunaliste teisendusreeglitega süsteemi $R_\rightarrow$:

(1) $1 * x \rightarrow x$

(2) $x^{-1} * x \rightarrow 1$

(3) $(x * y) * z \rightarrow x * (y * z)$

Definitsioon 19.1. *Termiteisendussüsteemiks*, lühendatult TTS, nimetame vasakult paremale orienteeritud võrdussüsteemi, kus iga teisendusreegli jaoks kehtib: reegli vasakpoolne term sisaldab iga muutujat reegli parempoolses termis.

Süsteem $R_\rightarrow$ on termiteisendussüsteem. Ka sel juhul saaksime termiteisendussüsteemi, kui reeglite 1 ja 3 vasakud pooled paremate pooltega ära vahetada — ainsana ei saa vahetada reegli 2 pooli (konstant 1 ei sisalda muutujat x).

Reegli $l \rightarrow r$ *rakendamiseks* termile t unifitseerime reegli vasakpoolse termi l teisendatava termi t mingi alamtermiga g, *tingimusel*, et g on termi l konkretisatsioon, s.t. unifikaatori σ jaoks $g = l\sigma$. Seejärel asendame termis t alamtermi g reegli parempoolse termiga substitutsiooni σ all, s.o. $r\sigma$. Reegli rakendamise samm on automaatse teoreemitõestamise peatükis vaadeldud *paramodulatsioonisammu* erijuht: lubatakse ainult *ühepoolset* unifitseerimist.

Näide. Reegliga (2) $x^{-1} * x \rightarrow 1$ saab teisendada termi $(1^{-1} * 1) * u$ kujule $1 * u$, kuid termi $(1^{-1} * u) * u$ teisendada ei saa: ei leidu substitutsiooni σ, nii et $(x^{-1} * x)\sigma$ annaks termi $(1^{-1} * u)$.

Definitsioon 19.2. TTS on *peatuv*, kui ta ei luba ühegi termi jaoks ehitada lõpmatut teisendusahelat.

On lihtne kontrollida, et süsteem $R_\rightarrow$ on peatuv. Reeglid 1 ja 2 vähendavad termi suurust, reegel 3 jätab suuruse samaks, kuid nihutab sulgusid alati paremale. Kontranäide: ükski reeglit $f(x) \rightarrow f(f(x))$ sisaldav TTS ei ole peatuv.

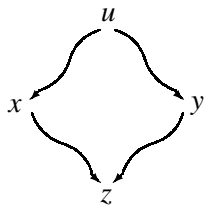
Definitsioon 19.3. Term on *normaalkujul*, kui talle ei saa enam rakendada ühtegi teisendust.

Näiteid süsteemi $R_{\rightarrow}$ jaoks: termid $x * x^{-1}$ ja 1 on normaalkujul, term $x * (y^{-1} * y)$ aga mitte, kuna reegliga 2 saab ta teisendada (normaal)kujule $x * 1$.

Intuitiivselt on selge, et otsitav termide t_1 ja t_2 võrdsuse lahendusmeetod võiks välja näha järgmine: teisendame termid normaalkujule $norm(t_1)$ ja $norm(t_2)$ ning kontrollime, kas $norm(t_1)$ ja $norm(t_2)$ on süntaktiliselt samased. Kui jah, siis on termid võrdsed, kui ei, siis nad ei ole võrdsed. Paraku ei võimalda süsteemi $R_{\rightarrow}$ poolt defineeritud normaalkuju sellist algoritmi kasutada. Nimelt võib süsteemis $R_{\rightarrow}$ ühel termil olla mitu erinevat normaalkuju: termi $(y^{-1} * y) * x$ saab reeglitega 2 ja 1 teisendada normaalkujule x, ning reegliga 3 teistsugusele normaalkujule $y^{-1} * (y * x)$.

Järgnevas tähistab $x \rightsquigarrow y$ asjaolu, et leidub teisendusahel termist x termini y.

Definitsioon 19.4. TTS on *kokkuvoolav*, kui iga termi u jaoks kehtib: kui eksisteerivad kaks termi x ja y, nii et $u \rightsquigarrow x$ ja $u \rightsquigarrow y$, siis eksisteerib term z, nii et $x \rightsquigarrow z$ ja $y \rightsquigarrow z$. Teisisõnu, kui termi u saab teisendada lahku mitmele eri kujule, siis saab kõik need erikujud ühekssamaks termiks kokku teisendada.



Definitsioon 19.5. Olgu E võrdussüsteem ja olgu S TTS. Ütleme, et S on E suhtes *korrektne*, kui iga süsteemi S jaoks võimaliku teisendussammu $t_1 \rightarrow t_2$ korral on võrdussüsteemist E tuletatav võrdus $t_1 = t_2$.

Definitsioon 19.6. Olgu E võrdussüsteem ja olgu S TTS. Ütleme, et S on E suhtes *täielik*, kui iga süsteemist E tuletatava võrduse $t_1 = t_2$ jaoks eksisteerib selline term c, et süsteemis S kehtivad $t_1 \rightsquigarrow c$ ja $t_2 \rightsquigarrow c$.

Järgmist kolme suhteliselt lihtsat, aga olulist lemmat peatuva ja kokkuvoolava TTS kohta soovitame lugejal harjutamiseks ise tõestada.

Lemma 19.7. *Peatuva ja kokkuvoolava TTS jaoks on igal termil a üks ja ainult üks normaalkuju, $norm(a)$.*

Lemma 19.8. *Termi t normaalkuju leidmiseks peatuva ja kokkuvoolava TTS jaoks võib teisendusreegleid rakendada mistahes järjekorras ja mistahes alamtermidele: lõpuks jõuame igal juhul sellesama normaalkujuni.*

Lemma 19.9. *Olgu $E_{\rightarrow}$ peatuv ja kokkuvoolav TTS. Olgu $E_{=}$ võrdussüsteem, mis on saadud $E_{\rightarrow}$ teisendusreeglite muutmisega harilikeks võrdusteks. Siis on TTS $E_{\rightarrow}$ võrdussüsteemi $E_{=}$ jaoks täielik.*

Täielik, peatuv ja kokkuvoolav TTS $E_{\rightarrow}$ annabki otsitava lahendusalgoritmi võrduste jaoks:

1. Leiame süsteemi $E_{\rightarrow}$ abil $norm(a)$ ja $norm(b)$.
2. Kui $norm(a)$ on süntaktiliselt samane termiga $norm(b)$, siis süsteemis $E_{=}$ tuletub $a = b$, vastasel korral ei tuletu $a = b$.

Sooviksime saada sellist süsteemi $R_{\rightarrow}$ modifikatsiooni $R+_{\rightarrow}$, mis oleks $R_{=}$ jaoks korrektne ja oleks nii peatuv (seda on ka $R_{\rightarrow}$) kui ka kokkuvoolav.

Esimene küsimus: kuidas tõestada, et mingi TTS R' on kokkuvoolav? Ei ole ju võimalik kontrollida lõpmatu hulga termide kõigi teisendatud kujude kokkuvoolavust. *Peatuvate* TTS-de jaoks annab vastuse järgmine Donald Knuthi ja Bendixi 1970. aastal tõestatud fundamentaalne teoreem, mille esitame siin tõestuseta.

Enne teoreemi esitamist defineerime *kriitilise paari* mõiste.

Definitsioon 19.10. Sisaldagu TTS R reegleid $\alpha_1 \to \beta_1$ ja $\alpha_2 \to \beta_2$. Olgu α_2 unifitseeritav termi α_1 mingi alamtermiga g. Olgu unifitseerimise tulemuseks substitutsioon σ. Termi $\alpha_1\sigma$ saab siis teise reegliga teisendada, olgu resultaadiks term α_1'. Selge, et termi $\alpha_1\sigma$ saab ka esimese reegliga teisendada, olgu resultaadiks term α_1''. Konstrueerime võrduse $\alpha_1'' = \alpha_1'$. Viimast võrdust nimetatakse *kriitiliseks paariks*.

Toome näite süsteemist $R_{\to}$. Kolmanda reegli $(x * y) * z \to x * (y * z)$ vasaku poole alamtermi $(x * y)$ saab unifitseerida teise reegli $v^{-1} * v \to 1$ vasakpoolse termiga. Saadud unifikaator σ on $\{x/v^{-1}, y/v\}$ ning $((x * y) * z)\sigma$ on $(v^{-1} * v) * z$. Konstrueeritud termile $(v^{-1} * v) * z$ saab rakendada teist reeglit, mis annab termi $1 * z$. Term $(v^{-1} * v) * z$ on ka kolmanda reegliga teisendatav, nimelt termiks $v^{-1} * (v * z)$. Kriitiline paar on seega $v^{-1} * (v * z) = 1 * z$.

Kokkuvõtteks: kriitilise paari saame juhul, kui ühe reegli vasak pool on unifitseeritav teise reegli vasakus pooles sisalduva alamtermiga. Niimoodi saame termi, mida on võimalik teisendada kahe erineva reegliga, s.o. kahel viisil. Kui TTS on kokkuvoolav, siis peab olema võimalik neid kahte teisendustulemust uuesti üheks termiks kokku teisendada. Selge, et igal lõpliku arvu reeglitega TTS-l on lõplik arv kriitilisi paare. Knuthi-Bendixi teoreem ütleb, et peatuva TTS-i kokkuvoolavuse tõestamiseks pole vaja kontrollida kõiki võimalikke terme, piisab vaid kriitiliste paaride kokkuvoolavuse kontrollimisest.

Teoreem 19.11 (Knuth-Bendix). *Olgu S peatuv TTS. Kui süsteemi S iga kriitilise paari $t'' = t'$ jaoks eksisteerib term c, nii et süsteemis S kehtib nii $t'' \rightsquigarrow c$ kui $t' \rightsquigarrow c$, siis on süsteem S kokkuvoolav.*

Kuna iga peatuva TTS-i kõigi kriitiliste paaride kokkuvoolavust saab algoritmiliselt kontrollida, annab Knuthi-Bendixi teoreem efektiivse vahendi kokkuvoolavuse probleemi lahendamiseks.

Varasemas kontrollisime, et meie poolt näitena vaadeldav süsteem $R_{\to}$ ei ole kokkuvoolav, ning püstitasime ülesande leida

kokkuvoolav $R+_{\to}$, mis oleks $R_{=}$ jaoks korrektne, seega täielik. Knuthi-Bendixi teoreem annab *Knuthi-Bendixi täielikustamisalgoritmi*, millega on esialgu mittetäielikku TTS-i sageli võimalik täiendada uute reeglitega, nii et kokkuvõttes saadakse täielik TTS.

Knuthi-Bendixi algoritm:

1. Olgu S peatuv TTS. Olgu S' uus TTS, ning olgu ta esialgu samane TTS-ga S.

2. Genereerime järjest kõik süsteemi S' kriitilised paarid $t'' = t'$. Iga paari jaoks leiame normaalkujud $norm(t'')$ ja $norm(t')$. Kui süsteemi S' jaoks ei ole $norm(t'')$ ja $norm(t')$ süntaktiliselt samased, siis pole paar kokkuvoolav, mispuhul teeme järgmist: kui $t'' \to t'$ on termiteisendusreegel ning $S' \cup \{t'' \to t'\}$ on peatuv, siis lisame reegli $t'' \to t'$ süsteemi S' ja läheme tagasi sammu 2 algusse. Vastasel korral kontrollime, kas $t' \to t''$ on termiteisendusreegel ning kas $S' \cup \{t' \to t''\}$ on peatuv. Kui jah, siis lisame reegli $t' \to t''$ süsteemi S' ja läheme tagasi sammu 2 algusse. Kui kumbki juhtum ei kehti, peatub algoritm ebaõnnestumisega.

3. Kui kõik kriitilised paarid uues süsteemis S' on kokkuvoolavad, siis on S' täielik ja algoritm lõpetab töö, andes resultaadiks süsteemi S'.

Peamine raskus Knuthi-Bendixi täielikustamisalgoritmi juures on uue reegli lisamisel saadava süsteemi peatuvuse kontrollimine. Oluline on ka kriitilisest paarist saadava reegli sobival viisil orienteerimine (kui mõlemad suunad on lubatavad). On võimalik, et ebasobiva valiku korral peatub algoritm ebaõnnestumisega, samas, kui sobiva valiku korral oleks võimalik leida täielik süsteem.

Täielikustamisalgoritmi saab mitmel moel optimeerida. Näiteks on lihtne veenduda, et iga teisendusreegli $l \to r$ parempoolse termi r saab asendada normaliseeritud termiga $norm(r)$, kaotamata seejuures korrektsust või täielikkust. Samuti võib kasutada resolutsioonimeetodi juurest tuttavat *neelamisstrateegiat*: kui TTS

sisaldab reegleid $l \to r$ ning $l' \to r'$, nii et mingi substitutsiooni σ jaoks reegel $l' \to r'$ on samane reegliga $(l \to r)\sigma$, võib reegli $l' \to r'$ midagi kaotamata kõrvaldada.

Vaatame kõigepealt triviaalset näidet, kahest reeglist koosnevat TTS-i:

$$\{g(f(x)) \to x, \quad f(a) \to b\}.$$

On lihtne näha, et antud TTS on peatuv ning tal on üksainus kriitiline paar, $g(b) = a$, mis on saadud termi $g(f(a))$ kahel viisil teisendades. Leitud kriitilise paari mõlemad pooled, $g(b)$ ja a, on normaalkujul ja üksteisest erinevad, seega ei ole kriitiline paar kokkuvoolav. Lisame süsteemile reegli $g(b) \to a$. Uus kolmest reeglist koosnev süsteem on samuti peatuv. Kuna kolmanda reegli lisamine uusi kriitilisi paare juurde ei andnud, algoritm peatub ja annab resultaadiks täieliku süsteemi

$$\{g(f(x)) \to x, \quad f(a) \to b, \quad g(b) \to a\}.$$

Järgmiseks anname otsitud süsteemi $R+_{\to}$, mis on peatuv, kokkuvoolav ja rühmateooria aksiomaatika $R_=$ jaoks täielik. Esimesed kolm reeglit pärinevad süsteemist $R_{\to}$, ülejäänud on lisatud Knuthi-Bendixi täielikustamisalgoritmiga.

(1) $1 * x \to x$

(2) $x^{-1} * x \to 1$

(3) $(x * y) * z \to x * (y * z)$

(4) $x^{-1} * (x * z) \to z$

(5) $y * 1 \to y$

(6) $(y^{-1})^{-1} \to y$

(7) $1^{-1} \to 1$

(8) $y * y^{-1} \to 1$

(9) $y * (y^{-1} * x) \to x$

(10) $(x * y)^{-1} \to y^{-1} * x^{-1}$

Niisiis on rühmateooria aksiomaatikast tuletatavate võrduste jaoks deduktsioon asendatud reduktsiooniga ning kahe termi võrdsuse küsimuse saab alati lahendada.

Täielikud TTS-d ei eksisteeri kaugeltki mitte kõigi võrdussüsteemide jaoks. Piisab näiteks võrduse $x * x = 1$ lisamisest rühmateooria aksiomaatikale $R_=$ ning lõplikku ja täielikku TTS-i enam ehitada ei saa. Ka ühestainsast kommutatiivsest võrdusest $x * y = y * x$ koosneva süsteemi jaoks ei leidu täielikku TTS-i (ei leidu ka peatuvat TTS-i). Spetsiifilistel juhtudel, nagu kommutatiivsuse ja assotsiatiivsuse esinemisel aksiomaatikas, saavutatakse TTS-i peatuvus erivahenditega: aksioomidena esitamise asemel saab kommutatiivsuse ja assotsiatiivsuse ehitada otse unifitseerimisalgoritmi sisse.

Siiani uurisime *peatuvate* termiteisendussüsteemide omadusi. Samas on mitu programmeerimises ja arvutamise teoorias kasutatavat TTS-i *mittepeatuvad*. Ka mittepeatuv TTS võib olla kokkuvoolav, näitena toome ühest reeglist koosneva süsteemi $a \to a$ ning kahest reeglist süsteemi $x \to a, \quad a \to f(a)$.

Mittepeatuva TTS-i jaoks Knuthi-Bendixi teoreem ei kehti. Kontranäiteks toome konstante teisendava mittepeatuva süsteemi:

$$\{a \to b, \quad b \to a, \quad a \to c, \quad b \to d\}.$$

Mõlemad kriitilised paarid $b = c$ ja $a = d$ on kokkuvoolavad, vastavalt termideks c ja d. Süsteem tervikuna ei ole aga kokkuvoolav: $a \rightsquigarrow c$ ja $a \rightsquigarrow d$, kuid c ja d on mõlemad normaalkujul.

Lõpetuseks tasub meenutada loogilise programmeerimise peatükki, kus mainisime, et Horni lausetena antud valemite jaoks langevad klassikaline ja intuitsionistlik loogika kokku. Kuivõrd harilik võrduse aksiomaatika koosneb Horni lausetest, siis langevad ka võrdussüsteemide jaoks klassikaline ja intuitsionistlik loogika kokku.

19.2. Lambda-arvutus

Lambda-arvutuse keel on 1930. aastatel Alonzo Churchi leiutatud lihtne ja universaalne meetod funktsioonide kirjapanekuks. Lambda-arvutuse teooria tegeleb arvutatavuse ja arvutatavate funktsioonide uurimisega, kasutades selleks lambda-arvutuse keelt kui universaalset programmeerimiskeelt. *Churchi tees* väidab, et iga algoritmi saab lambda-arvutuse keeles kirja panna. On võimalik näidata, et lambda-arvutus, nagu ka Prolog, C ja Basic on üks paljudest universaalsetest programmeerimiskeeltest. Konkreetselt on lambda-arvutuse keel ja teooria *funktsionaalsete programmeerimiskeelte* aluseks.

Esitame kõigepealt informaalse sissejuhatuse.

Üks harilikumaid praktikas kasutatavaid funktsioonide kirjapaneku viise on selline:

$$f(x) = x * x + 1.$$

Funktsioon esitatakse, andes talle samas *nime*, konkreetses näites f. Lambda-arvutuses esitatakse funktsioone, vastupidi, kui anonüümseid, nimeta terme. Äsjatoodud näide on lambdakirjaviisis

$$\lambda x.x * x + 1.$$

Lambdasümboli λ järele kirjutatakse funktsiooni formaalseks parameetriks olev muutuja, seejärel punkt ja funktsiooni keha. Parameetrit nimetatakse *seotud muutujaks*, analoogiliselt muutuja sidumisele kvantoriga. Mitme formaalse parameetriga funktsioone esitatakse mitme üksteise sees oleva üheparameetrilise funktsioonina: $\lambda x.\lambda y.x * x + y * y$.

Funktsiooni rakendamiseks kirjutatakse traditsioonilise $f(3)$ asemel $(\lambda x.x * x + 1)3$ — viimase väärtuseks on 10. Analoogiliselt annab $((\lambda x.\lambda y.x * x + y * y)2)3$ väärtuseks 13. Lambdatermi rakendamisel asendatakse seotud muutuja termi kehas termile antud argumendiga. Seotud muutujate asendamine ongi lambda-arvutuse põhiidee. Pisut keerukam termi rakendamise ehk redutseerimise näide: $((\lambda f.f(f\ 2))(\lambda x.x * x + 1))$ annab $(\lambda x.x * x + 1)((\lambda x.x * x + 1)2)$ annab $(\lambda x.x * x + 1)(2 * 2 + 1)$ annab $(\lambda x.x * x + 1)5$ annab $5 * 5 + 1$ annab 26. Resultaat seejuures asenduste tegemise järjekorrast ei sõltu.

Siirdume nüüd lambda-arvutuse formaalse esituse juurde.

Definitsioon 19.12. *Lambdaterm* on kas muutuja, konstant, rakendusterm $\circ(F, A)$ või abstraktsiooniterm $\lambda x.M$, kus F, A ja M on lambdatermid.

Edaspidises kirjutame rakendustermi $\circ(F, A)$ lihtsamalt: FA.

Definitsioon 19.13. *Lambda-arvutuse* annab üksainus fundamentaalne aksioomiskeem, *β-reduktsioon* ehk lihtsalt *reduktsioon*:

$$(\lambda x.M)N = M[x := N],$$

kus $M[x := N]$ tähistab termi M pärast muutuja x asendamist termiga N.

Muutuja x olemasolu termis M ei ole kohustuslik. Muutuja x võib aksioomiskeemis asendada mistahes muutujaga. Mingis termis sisalduv lambdasümboliga seotud muutuja on vabalt ümbernimetatav. Muutuja ümbernimetamine formuleeritakse vahel eraldi α-reduktsiooni reeglina.

Tuleb tähele panna, et β-reduktsioon ei ole otseselt esitatav esimest järku võrdusena: tegu on nimelt aksioomiskeemiga. Reduktsioonile lisaks kasutatakse standardset aksiomaatikat võrduspredikaadi aksiomatiseerimiseks (vaata näiteks automaatse teoreemitõestamise peatükki). Kui orienteerida reduktsioonireeglis võrdus, saab lambda-arvutust vaadelda erilist tüüpi termiteisendussüsteemina.

Vahel lisatakse lambda-arvutusele teisigi reduktsioonireegleid, kuid nende olemasolu või puudumine ei muuda arvutuse põhiomadusi.

Reduktsiooni rakendamisel tuleb pöörata tähelepanu *seotud* ja *vabadele* muutujatele: asendatakse ainult vabu muutujaid. Näiteks $yx(\lambda x.x)[x := N] = yN(\lambda x.x)$, kuna alamtermis $(\lambda x.x)$ on muutuja x seotud. Üldjuhul tuleb muutujanimede sassimineku

vältimiseks seotud muutujad enne asendussammu tegemist ümber nimetada.

Vaadeldes lambda-arvutust termiteisendussüsteemina, märkame kõigepealt, et lambda-arvutus ei ole peatuv. Tähistame rakendustermi $(\lambda x.xx)(\lambda x.xx)$ tähega ω. Term ω redutseerub ühe sammuga uuesti iseendaks, võimaldades seega ehitada lõputu reduktsiooniahela. Vaatame nüüd rakendustermi $((\lambda x.\lambda y.x)(\lambda x.x))\omega$ ja püüame seda redutseerida normaalkujule. Alustades redutseerimist argumentidest (seestpoolt väljapoole) nagu harilikes programmeerimiskeeltes tavaks, peame kõigepealt normaalkujule viima termi ω, kuid sel termil normaalkuju puudub, seega satume lõpmatule reduktsiooniahelale. Alustades redutseerimist, vastupidi, *tervest* termist, s.o. väljastpoolt sissepoole ehk vasakult paremale, saame termi $((\lambda y.(\lambda x.x))\omega$ ning järgmisena $(\lambda x.x)$, mis on normaalkujul. Meie näites ei viinud seestpoolt väljapoole redutseerimine normaalkujule, küll aga leidsime normaalkuju väljastpoolt sissepoole redutseerides. Näide illustreeris järgmist Haskell Curry tõestatud teoreemi.

Definitsioon 19.14. *Redeks* on rakendusterm, mille vasakpoolne argument on abstraktsiooniterm.

Redeks on näiteks $((\lambda x.x)y)$, aga mitte (xx) või $(((\lambda x.x)y)y)$.

Teoreem 19.15 (normaliseerimine). *Kui lambdatermil M on normaalkuju, siis alati kõige vasakpoolsemat redeksit eelistav redutseerimisstrateegia jõuab normaalkujuni.*

Esitame ühe suhteliselt lihtsama teoreemi koos tõestusega:

Teoreem 19.16 (püsipunktiteoreem).

1. *Iga lambdatermi F jaoks eksisteerib lambdaterm X, nii et $FX = X$.*
2. *Püsipunktifunktsioonil $\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ nimega Y on järgmine omadus: iga lambdatermi F jaoks kehtib $F(YF) = YF$, s.t. Y arvutab termi püsipunkti.*

Tõestus.

1. Olgu W term $\lambda x.F(xx)$. Termiks X võtame WW. Tõepoolest, $WW = (\lambda x.F(xx))W = F(WW)$.
2. Kasutame eelmist punkti, pannes tähele, et $YF = (\lambda x.F(xx))(\lambda x.F(xx))$, mis on samane termiga X.

□

Lambda-arvutuse kõige fundamentaalsem teoreem on Alonzo Churchi ning Barkley Rosseri tõestatud kokkuvoolavuse teoreem:

Teoreem 19.17 (Church-Rosser). *Lambda-arvutus on kokkuvoolav TTS, s.t. iga lambdatermi M jaoks kehtib: kui eksisteerivad kaks termi A ja B, nii et $M \rightsquigarrow A$ ja $M \rightsquigarrow B$, siis eksisteerib term C, nii et $A \rightsquigarrow C$ ja $B \rightsquigarrow C$.*

Kokkuvoolavuse teoreemist järeldub muu hulgas, et igal normaalkujuga lambdatermil on üksainus normaalkuju. Kuna lambda-arvutus ei ole peatuv, siis ei järeldu kokkuvoolavusest termide võrdsuse probleemi lahenduvus. Saab näidata, et ei eksisteeri universaalset lahendavat algoritmi küsimuse jaoks, kas mingi võrdus $t = t'$ on lambda-arvutuses tuletatav.

Järgmiseks näitame, kuidas esitada lambda-arvutuses mittetriviaalseid programme. Kõigepealt tuleb rõhutada, et *puhta*, lisanditeta lambda-arvutuse selline kasutamine ei ole praktilise programmeerimise seisukohalt ratsionaalne. Lambda-arvutusel baseeruvad funktsionaalse programmeerimise keeled sisaldavad peale puhta lambda-arvutuse veel muid konstruktsioone ja vahendeid, näiteks arve, aritmeetikat, loendeid jms., mis muudavad programmid efektiivseteks ja mugavalt kirjapandavateks. Meie eesmärk on demonstreerida, et selliseid praktilisi lisakonstruktsioone on teoreetiliselt võimalik esitada puhta lambda-arvutuse vahenditega. Terviklikku formaalset täielikkuse tõestust me siinkohal ei esita.

Täisarvude esitamise skeem Churchi järgi:

0: $\lambda f.\lambda x.x$

1: $\lambda f.\lambda x.fx$

2: $\lambda f.\lambda x.f(fx)$

3: $\lambda f.\lambda x.f(f(fx))$

üldiselt n: $\lambda f.\lambda x.f^n x$

Rosseri esitatud peamised aritmeetikafunktsioonid, mõeldud Churchi järgi kodeeritud arvudele rakendamiseks:

+: $\lambda x.\lambda y.\lambda p.\lambda q.(xp)((yp)q)$

*: $\lambda x.\lambda y.\lambda z.x(yz)$

exp: $\lambda x.\lambda y.yx$

Toodud aritmeetikafunktsioonide esimesed kaks parameetrit on sisulised: neid kasutatakse arvutamise juures redutseerimisel. Ülejäänud parameetrid jäävad resultaadi komponentideks.

Näiteks: $(exp\ 2)2 = ((\lambda x.\lambda y.yx)2)2 = (\lambda y.y2)2 = 2\ 2 = (\lambda f'.\lambda x'.f'(f'x'))2 = (\lambda x'.2(2\ x')) = (\lambda x'.\lambda x.(2\ x')((2\ x')x)) = (\lambda x'.\lambda x.x'(x'((2\ x')x)) = (\lambda x'.\lambda x.x'(x'(x'(x'(x'x)))) = 4$.

Loogilised tõeväärtused *tõsi* ja *väär*, harilik programmeerimiskonstruktsioon "*if B then P else Q*" ning predikaat *null* tähendusega "x võrdub arvuga 0" on suhteliselt lihtsalt esitatavad:

tõsi: $\lambda x.\lambda y.x$

väär: $\lambda x.\lambda y.y$

if B then P else Q: $(B\,P)Q$

null: $\lambda x.((x\ (\lambda y.\lambda z.$*väär*$))\ (\lambda x.x))$ *tõsi*

Harilikes programmeerimiskeeltes kasutatakse rekursiivsete funktsioonide defineerimise juures *funktsiooni nime*. Näiteks faktoriaali definitsiooni

```
fakt(x) = if x=0 then 1 else x*fakt(x-1)
```

kehas kasutatakse defineeritavat nime `fakt`. Lambda-arvutuses aga funktsioonidel nime ei ole. Kuidas sellises olukorras rekursiivseid funktsioone defineerida?

Lahenduseks on püsipunktiteoreemi tõestuses toodud püsipunktifunktsiooni Y kasutamine. Meenutame, et funktsioonil Y on järgmine omadus: iga termi F jaoks kehtib $F(YF) = YF$. Püsipunktifunktsiooni kasutamise skeem on järgmine. Olgu rekursiivne funktsioon f defineeritud kui $f(x) = M$, kus M võib sisaldada nime f. Asendame definitsiooni lambdatermiga $(Y\ \lambda f.\lambda x.M)$, muutes funktsiooni nime seotud muutujaks ja rakendades termile püsipunktifunktsiooni.

Tähistame tähega F faktoriaali definitsioonist saadud termi

$$\lambda\texttt{fakt.}\lambda\texttt{x}.\mathit{if}\ x = 0\ \mathit{then}\ 1\ \mathit{else}\ x * \mathit{fakt}(x-1).$$

Faktoriaali funktsiooni defineerime kui rakendustermi (YF).

Saadud faktoriaalifunktsiooni (YF) näiteks arvule 4 rakendades saame seestpoolt väljapoole redutseerides ehitada mittepeatuva teisendusahela

$$(YF)4 = (F(YF))4 = (F(F(YF)))4 = \ldots$$

Normaalkujuga termide jaoks garanteeritult peatuvat väljastsissepoole-, vasakult-paremale-reduktsioonistrateegiat kasutades aga teisendusahel peatub, andes normaalkuju, mis ongi faktoriaali arvutamise resultaadiks:

$$\begin{aligned}
&(YF)4 = \\
&F(YF)4 = \\
&(\lambda \mathit{fakt}.\lambda x.\mathit{if}\ x = 0\ \mathit{then}\ 1\ \mathit{else}\ x * \mathit{fakt}(x-1))(YF)4 = \\
&(\lambda x.\mathit{if}\ x = 0\ \mathit{then}\ 1\ \mathit{else}\ x * (YF)(x-1))4 = \\
&\mathit{if}\ 4 = 0\ \mathit{then}\ 1\ \mathit{else}\ 4 * (YF)(4-1) = \\
&4 * (YF)3 = \\
&4 * F(YF)3 = \\
&\ldots \\
&4 * 3 * (YF)2 = \\
&\ldots \\
&4 * 3 * 2 * (YF)1 = \\
&\ldots
\end{aligned}$$

$4 * 3 * 2 * (YF)0 =$
$4 * 3 * 2 * 1 =$
24.

19.3. Kombinatoorne loogika

Lambda-arvutuse reduktsioonimehhanismil on puudusi:

- β-reduktsiooni aksioom ei ole esimest järku võrdus.
- β-reduktsiooni sammu alustades peab seotud muutujad ümber nimetama, vältimaks muutujate omavahel sassiminekut.

Mainitud puudustest on vaba lambda-arvutusega analoogiline, peamiselt teoorias kasutatav programmeerimiskeel, *kombinatoorne loogika*, mille 1920. ja 1930. aastatel leiutasid Curry ja Schönfinkel.

Kombinatoorne loogika on kahe või kolme äärmiselt lihtsa võrdusega esitatav esimest järku aksiomaatika, mis sisaldab konstante K, S ja I ning kahekohalist rakendusfunktsiooni $\circ(F, M)$, mida edaspidi kirjutame parema loetavuse huvides kujul $F \circ M$.

(1) $((S \circ x) \circ y) \circ z = (x \circ z) \circ (y \circ z)$

(2) $(K \circ x) \circ y = x$

(3) $I \circ x = x$

Konstant I on defineeritav kui term $(S \circ K) \circ K$ ning kolmas võrdus on seepeale tuletatav kahest esimesest. Kombinatoorse loogika esitamiseks piisab seega kahest esimesest võrdusest, kolmas võetakse aksiomaatikasse mugavuse mõttes.

Kui orienteerida kombinatoorse loogika aksiomaatika võrdused 1–3 termiteisendusreegliteks, saame mittepeatuva, kuid kokkuvoolava ja täieliku TTS-i. Analoogiliselt lambda-arvutusega ei eksisteeri universaalset lahendavat algoritmi küsimuse jaoks, kas võrdus $t = t'$ on kombinatoorse loogika aksioomidest tuletatav.

Kombinatoorne loogika ei piirdu ainult äsja toodud kolme konstandi defineerimisega. Sageli on kasulik tuua lisaks teisigi kombinaatoreid.

Definitsioon 19.18. *Kombinaatoriks* nimetame konstanti c, mille jaoks on antud termiteisendusreegel kujul

$$(\ldots((c \circ x_1) \circ x_2)\ldots) \circ x_n \to M,$$

kus M on term, mis võib sisaldada ainult konstante, muutujaid $x_1, \ldots, x_n$ ja rakendusfunktsiooni $\circ$.

Iga lambda-arvutuse termi saab teisendada kombinatoorse loogika termiks. Esitame teisendusalgoritmi, defineerides selleks teisendusfunktsioonid C ja A. Teisendusfunktsioonide argumendid on antud nurksulgudes, e tähistab suvalist lambdatermi, x ja v tähistavad muutujaid ning c tähistab konstante. A esitab nn. abstraktsioonialgoritmi, mis on otseselt esitatud substitutsioonimehhanismiga analoog lambda-arvutuse β-reduktsioonile.

- $C[(e_1 e_2)] = C[e_1] \circ C[e_2]$
- $C[(\lambda x.e)] = A[x, C[e]]$
- $C[v] = v$
- $C[c] = c$
- $A[x, (e_1 e_2)] = (S \circ A[x, e_1]) \circ A[x, e_2]$
- $A[x, x] = I$
- $A[x, v] = (K \circ v)$, kui $x \neq v$
- $A[x, c] = (K \circ c)$

Järgmise fundamentaalse teoreemi soovitame lugejal ise tõestada:

Teoreem 19.19 (substitutsioon). *Kombinatoorse loogika TTS-i jaoks kehtib*

$$(A[x, M] \circ N) \rightsquigarrow M[x := N],$$

kus $A[x, M]$ *on abstraktsioonialgoritmi rakendamise resultaat muutujale* x *ja termile* M. N *ja* M *on esimest järku termid ja* $M[x := N]$ *tähistab termi* M *pärast muutuja* x *asendamist termiga* N.

Seega saab pärast lambdatermi teisendust kombinatoorsesse loogikasse kasutada β-reduktsiooni asemel kombinatoorse loogika esimest järku termiteisendusreegleid. Viimaste rakendamise juures pole vaja pidevalt muutujaid ümber nimetada.

Näide. Teisendame lambdatermi $((\lambda x.(+\ x)x)\ 3)$ kombinatoorsesse loogikasse:
$C[((\lambda x.(+\ x)x)3)]$ annab
$C[(\lambda x.(+\ x)x)] \circ C[3]$ annab
$C[(\lambda x.(+\ x)x)] \circ 3$ annab
$A[x, C[(+\ x)x]] \circ 3$ annab
$A[x, C[(+\ x)] \circ C[x]] \circ 3$ annab
$A[x, (C[+] \circ C[x]) \circ C[x]] \circ 3$ annab kolme sammuga
$A[x, (+ \circ x) \circ x] \circ 3$ annab
$((S \circ A[x, (+ \circ x)]) \circ A[x, x]) \circ 3$ annab
$((S \circ A[x, (+ \circ x)]) \circ I) \circ 3$ annab
$((S \circ ((S \circ A[x, +]) \circ A[x, x])) \circ I) \circ 3$ annab
$((S \circ ((S \circ A[x, +]) \circ I)) \circ I) \circ 3$ annab
$((S \circ ((S \circ (K \circ +)) \circ I)) \circ I) \circ 3$.

Saadud termi $((S \circ ((S \circ (K \circ +)) \circ I)) \circ I) \circ 3$ teisendame kombinatoorse loogika reeglitega normaalkujule:
$((S \circ ((S \circ (K \circ +)) \circ I)) \circ I) \circ 3$ annab
$(((S \circ (K \circ +)) \circ I) \circ 3) \circ (I \circ 3)$ annab
$(((S \circ (K \circ +)) \circ I) \circ 3) \circ 3$ annab
$(((K \circ +) \circ 3) \circ (I \circ 3)) \circ 3$ annab
$(((K \circ +) \circ 3) \circ 3) \circ 3$ annab
$(+ \circ 3) \circ 3$.

Lõpuks saime sama termi, mille oleksime saanud esialgset lambdatermi normaalkujule viies. Harjutuseks soovitame otsida termi, millel normaalkuju puudub ning defineerida lambda-arvutuse püsipunktifunktsiooni Y vaste kombinatoorses loogikas.

Kombinatoorse loogika jaoks kehtivad kõik varem esitatud teoreemid lambda-arvutuse kohta: püsipunktiteoreem, normaliseerimisteoreem ja Churchi-Rosseri teoreem.

Põnevatele teoreetilistele võimalustele vaatamata võib kombinatoorne loogika näida praktiliseks kasutamiseks liiga kohmakas ja ebaefektiivne. See ei ole päriselt nii: näiteks on praktikas kasutatava funktsionaalse programmeerimiskeele Miranda jaoks realiseeritud optimeeriv kompilaator, mis teisendab programmid laiendatud kombinatoorsesse loogikasse, s.o. ülaltoodud konstandid ja reeglid pluss *ca* kümme lisakombinaatorit — viimaste ainus eesmärk on arvutamiskiiruse suurendamine. Uuemad puhtalt funktsionaalsete keelte — näiteks Haskell — kompilaatorid teisendavad programmi *superkombinaatoriteks*, s.o. mitte etteantud, fikseeritud hulgaks kombinaatoriteks, vaid uute, antud programmi jaoks sobivate kombinaatorite hulgaks. Kompileeritud programmi täitmine tähendab termi normaalkujule teisendamist.

Ülesannete vastused

2.4.1. Teise kehtivuse definitsiooni tõesusest järeldub esimese definitsiooni tõesus (sest tõene järeldus ei saa olla väär). Seepärast võib pidada esimest kehtivuse definitsiooni teisest üldisemaks. (Kahjuks muudab võimalikkuse modaalsuse kasutamine sellest arusaamise raskeks.)

3.2.5. Väidet $p \Leftrightarrow q$ loetakse "p siis ja ainult siis, kui q". Me saame selle lause jagada kaheks osaks: "p siis, kui q" ja "p ainult siis, kui q", ning avaldada nad implikatsiooni ja eituse kaudu:

$q \Rightarrow p$	p siis, kui q	$q \supset p$
	kui q, siis p	
	q on p jaoks *piisav*	
$p \Rightarrow q$	p ainult siis, kui q	$\neg q \supset \neg p$
	q on p jaoks *tarvilik*	$\parallel$
	kui p kehtib, siis peab kehtima ka q	$p \supset q$

3.3.1. Eesti keele lause on kas lihtlause või liitlause. Lihtlaused jagunevad laiendamata lihtlauseteks ja laiendatud lihtlauseteks. Laiendamata lihtlause koosneb alusest ja öeldisest. Laiendatud lihtlause koosneb näiteks alusest, öeldisest ja sihitisest. Liitlaused on põimlaused ja kiillaused jne. jne.

3.3.3. c) Väärtustustel $A = B = C = v$ ja $A = C = v$, $B = t$ on valemite $(A \supset B) \supset C$ ja $A \supset (B \supset C)$ tõeväärtused erinevad.

3.6.4. Asendades kehtivas arutluses lausemuutujad metamuutujatega,

saame kehtiva arutlusvormi, kusjuures esialgne arutlus on saadud arutlusvormi erikuju.

3.8.2. b) $(B \supset G) \supset (G \supset B)$; c) $((E_1 \vee E_2 \vee E_3) \supset B \,\&\, C) \supset (D \supset A)$.

4.2.1. a) $\forall x\,(Vx \vee Mx)$; d) $\forall x \exists y\, Axy$; g) $\exists x \exists y\, Axy \supset Ajj$.

4.4.1. Vihje: kasutage võrdusepredikaati.

4.7.2. a) $\forall x \forall y\,(Ix \,\&\, Ky \supset Vxy)$; e) $\exists x\,(Ix \,\&\, \neg \exists y\,(Ky \,\&\, Vxy))$.

4.11.3. a) $\forall x \forall y\,(x+y = y+x)$; b) $\forall x \forall y \forall z\,(x+y = z \supset (z-x = y) \,\&\, (z-y = x))$; d) $\forall x \forall y \forall z\,(xy = z \supset (x = 1 \supset z = y) \,\&\, (y = 1 \supset z = x))$.

5.1.1. $X \,\&\, Y \,\&\, \neg Z \vee \neg X \,\&\, \neg Y \,\&\, \neg Z$.

6.4.1. Valemite hulga

$$\{\forall x \exists y\, Pxy, \neg \exists x\, Pxx, \forall x \forall y\,(Pxy \supset \neg Pyx)\}$$

tõesuspuu on avatud, kuid selles ei ole lõplikku avatud lõpetatud haru. Selle lausete hulga mudeliks on näiteks interpretatsioon $\langle \mathbb{N}; < \rangle$.

7.2.1. Väide kehtib atomaarsete valemite korral, sest nad on iseendaga identsed. Ta kehtib ka induktsiooni sammu korral, sest konjunktsioon ja disjunktsioon on tõesed, kui nende argumendid on tõesed. Järelikult kehtib väide selle baasi kõigi valemite korral.

9.2.3. Vihje: kasutage sümbolite ASCII-tabelit.

11.3.1. Vihje: teooria, millel pole mudelit, on vasturääkiv; vasturääkivas teoorias saab aga tuletada vastuolu.

11.5.1. Näiteks $Px \stackrel{\text{def}}{=}$ "x on termin" ja $Qx \stackrel{\text{def}}{=}$ "x on magus".

12.2.1. Vihje: võtke väiteks p suvaline kontingentne lause.

12.3.2. $\models_{S5} \Box(p \vee \neg p)$, sest p peab olema paratamatult kas tõene või väär; $\models_{S5} \Box p \vee \Box \neg p$ aga ei kehti, sest p ei tarvitse olla kas paratamatult tõene või paratamatult väär. Ta võib näiteks olla kontingentne.

12.5.1. Vihje: defineerige interpretatsioonid I ja I' nii, et $U_I = \mathcal{W}_{I'} = \{w_1, \ldots, w_n\}$, $w_i[A] = t \Leftrightarrow w_i \in [A]$ ja vabad muutujad asendatakse tegeliku maailmaga $w^*_{I'} \in \mathcal{W}_{I'}$.

12.5.2. Vihje: jagage predikaatide ekstensioonid ekvivalentsiklassidesse.

14.2.2. Olgu $\mathcal{F}[A] = \mathcal{F}[A \supset B] = 1$. Implikatsiooni semantika põhjal saame võrduse $1 - (1 - \mathcal{F}[B]) = 1$. Järelikult $\mathcal{F}[B] = 1$.

15.7.1. Kirjeldame vaikimisiteooria

$$W = \begin{cases} \mathit{Tartu\_tänav}(x) \supset \mathit{tee}(x) \\ \mathit{Tartu\_tänav}(x) \supset \neg \mathit{viib\_Rooma}(x) \\ \mathit{Tartu\_tänav}(\mathit{Ülikooli\_tänav}) \end{cases}$$

$$D = \left\{ \frac{\mathit{tee}(x) : \mathit{viib\_Rooma}(x)}{\mathit{viib\_Rooma}(x)} \right.$$

Selles teoorias saab *modus ponens*'iga tuletada väite

$$\neg \mathit{viib\_Rooma}(\mathit{Ülikooli\_tänav}).$$

16.6.3. Lihtsustamis- ja idempotentsusreeglid

$$A \otimes B \multimap A,$$
$$A \multimap A \otimes A$$

ei kehti keemiliste reaktsioonide korral, sest aatomite hulgad peavad enne ja pärast reaktsiooni olema samad.

17.2.1. Minu kass sööb mind.

17.2.2. Kui draakon on näljane, siis ta kütib. Kui draakon on väsinud, siis ta magab. Draakon ei saa olla korraga näljane ja väsinud.

17.2.4.

$$x_2 = g(x_1, x_1),$$
$$x_3 = g(g(x_1, x_1), g(x_1, x_1)),$$
$$x_4 = g(g(g(x_1, x_1), g(x_1, x_1)), g(g(x_1, x_1), g(x_1, x_1))),$$
$$\ldots$$
$$x_n = g(g(\ldots g(x_1, x_1) \ldots), g(\ldots g(x_1, x_1) \ldots)).$$

18.1.1. Vihje: püüdke igal järgneval hulga läbimisel muuta tõeseks vähemalt üks vaadeldava hulga disjunktidest.

Kirjandus

Sissejuhatus

Loogika ajalugu:

J. Bochenski. *A history of formal logic.* University of Notre Dame Press, 1951.

J. van Heijenoort. *From Frege to Gödel.* Harvard University Press, 1967.

W. Kneale, M. Kneale. *The development of logic.* Clarendon Press, 1962.

Loogikafilosoofia:

M. A. Boden (toim.). *The philosophy of artificial intelligence.* Oxford University Press, 1990.

D. Gabbay, F. Guenthner (toim.). *Handbook of philosophical logic.* Kluwer, 1983–1989.

S. Haack. *Philosophy of logics.* Cambridge University Press, 1978.

G. Polya. *Kuidas lahendada ülesannet.* Tallinn, 1967.

W. V. O. Quine. *Philosophy of logic.* Prentice-Hall, 1970.

S. Read. *Thinking about logic: an introduction to the philosophy of logic.* Oxford University Press, 1995.

B. Russell. *Uurimus tähendusest ja tõest.* Tallinn, 1995.

G. H. von Wright. *Minerva öökull.* Tallinn, 1996.

I Klassikaline loogika

Eestikeelsetes formaalloogika õpikutes käsitletakse põhiliselt traditsioonilist süllogistlikku loogikat:

E. Grauberg. *Loogika, keel ja mõtlemine.* Tallinn, 1996.

I. Meos. *Loogika. Argumentatsioon. Mõtlemiskultuur.* Tallinn, 1996.

A. Pärl. *Otsustusõpetus ühes sissejuhatusega väiteloogikasse ja formaalloogilised mõtlemisseadused.* Tartu, 1970.

G. Vuks. *Formaalse loogika ehk õige mõtlemise alused.* Tartu, 1992.

Ingliskeelsed õpikud:

J. Allwood, L.-G. Andersson, Ö. Dahl. *Logic in linguistics.* Cambridge University Press, 1977.

M. Bergmann, J. Moor, J. Nelson. *The logic book.* McGraw-Hill, 1990.

G. Forbes. *Modern Logic: a text in elementary symbolic logic.* Oxford University Press, 1994.

W. Hodges. *Logic.* Pelican, 1977 (Penguin, 1991).

J. Lyons. *Linguistic semantics: an introduction.* Cambridge University Press, 1996.

T. J. McKay. *Modern formal logic.* Macmillan, 1989.

B. H. Partee, A. T. Meulen, R. E. Wall. *Mathematical methods in linguistics.* Kluwer, 1990.

II Matemaatiline loogika

Eestikeelsed õpikud:

I. Kull. *Matemaatiline loogika.* Tallinn, 1964.

J. Penjam. *Loogika arvutiteaduses.* Tallinn, 1995, [greta.cs.ioc.ee/pub/CompLog/].

J. Penjam. *Teoreetiline informaatika I–II.* Tallinn, 1993–1995, [greta.cs.ioc.ee/pub/TeorInf/].

R. Prank. *Matemaatiline loogika ja diskreetne matemaatika I–III.* Tartu, 1978–1983.

Ingliskeelsed õpikud:

M. Ben-Ari. *Mathematical logic for computer science.* Prentice-Hall, 1993.

A. G. Hamilton. *Logic for mathematicians.* Cambridge University Press, 1988.

S. C. Kleene. *Introduction to metamathematics.* Van Nostrand, 1952.

S. C. Kleene. *Mathematical logic.* John Wiley & Sons, 1967.

E. Mendelson. *Introduction to mathematical logic.* Van Nostrand, 1964.

A. Nerode, R. A. Shore. *Logic for applications.* Springer-Verlag, 1993.

C. H. Papadimitriou. *Computational complexity.* Addison Wesley, 1994.

W. V. Quine. *Mathematical logic.* Cambridge University Press, 1951.

J. R. Shoenfield. *Mathematical logic.* Addison Wesley, 1967.

Algoritmiteooria ja rekursiooniteooria õpikud:

G. S. Boolos, R. C. Jeffrey. *Computability and logic.* Cambridge University Press, 1989.

N. Cutland. *Computability: an introduction to recursive function theory.* Cambridge University Press, 1980.

A. J. Kfoury, R. N. Moll, M. A. Arbib. *A programming approach to computability.* Springer-Verlag, 1982.

Tänapäeval saab loogikast parema ülevaate, lugedes teatmeteoseid:

J. Barwise (toim.). *Handbook of mathematical logic.* North-Holland, 1977.

D. V. Gabbay, C. J. Hogger, J. A. Robinson (toim.). *Handbook of logic in artificial intelligence and logic programming.* 4 kd., Clarendon Press, 1993–1995.

J. van Leeuwen (toim.). *Handbook of theoretical computer science. Volume B. Formal models and semantics.* Elsevier, 1990.

S. C. Shapiro (toim.). *Encyclopedia of artificial intelligence.* 2 kd., John Wiley & Sons, 1992.

III Mitteklassikaline loogika

Universaalsed õpikud lisaks eespool mainitud Susan Haacki ja Graeme Forbese raamatutele:

L. Bolc, P. Borowik. *Many-valued logics: 1 Theoretical foundations.* Springer-Verlag, 1992.

M. Ginsberg. *Essentials of artificial intelligence.* Morgan Kaufmann, 1993.

S. Reeves, M. Clarke. *Logic for computer science.* Addison Wesley, 1990.

R. Turner. *Logics for artificial intelligence.* Ellis Horwood Limited, 1984.

Modaalloogika:

B. F. Chellas. *Modal logic.* Cambridge University Press, 1975.

G. E. Hughes, M. J. Cresswell. *A new introduction to modal logic.* Routledge, 1996.

Intuitsionistlik loogika:

M. Dummett. *Elements of intuitionism.* Clarendon Press, 1977.

Mittemonotoonne loogika:

D. W. Etherington. *Reasoning with incomplete information.* Pitman, 1988.

M. Ginsberg (toim.). *Readings in nonmonotonic reasoning.* Morgan Kaufmann, 1987.

Lineaarloogika:

J.-Y. Girard. *Linear logic: its syntax and semantics.* Advances in Linear Logic, Cambridge University Press, 1995, 1–42. [lmd.univ-mrs.fr/pub/girard/Synsem.ps.Z].

J.-Y. Girard. *Linear logic.* Theoretical Computer Science, 50 (1987), 1–102.

IV Loogika rakendusi programmeerimises ja tehisintellektis

Automaatne teoreemitõestamine:

M. Koit. *Resolutsioonimeetod.* Tartu, 1989.

C.-L. Chang, R. C.-T. Lee. *Symbolic logic and mechanical theorem proving.* Academic Press, 1973.

M. Fitting. *First-order logic and automated theorem proving.* Springer-Verlag, 1996.

D. Loveland. *Automated theorem proving: a logical basis.* North-Holland, 1978.

U. Schöning. *Logic for computer scientists.* Birkhäuser, 1989.

Loogiline programmeerimine:

R. Kowalski. *Logic for problem solving.* North-Holland, 1979.

J. W. Lloyd. *Foundations of logic programming.* Springer-Verlag, 1987.

L. Sterling, E. Shapiro. *The art of Prolog.* MIT Press, 1994.

Funktsionaalne programmeerimine:

H. P. Barendregt. *The lambda calculus: its syntax and semantics.* North-Holland, 1984.

R. Bird, P. Wadler. *Introduction to functional programming.* Prentice-Hall, 1988.

L. C. Paulson. *ML for the working programmer.* Cambridge University Press, 1991.

Indeks

′ 115, 214
+ 214
:- 350
; 353
? 288
?- 351
* 60
** 60
0 214, 291
1 291
! 288, 360
:= 377
= 63, 154, 321, 358
=.. 364
[] 354
w^* 239
& 68, 289
. 68, 354
⊥ 145, 172, 201, 226, 255, 270, 290
∘ 377, 382
× 181
$\stackrel{\text{def}}{=}$ 168
≡ 74
∃ 99
$\exists_1$ 216
∀ 98, 221
↓ 205
↑ 206
⊃ 72
∈ 155
L_N 213
λ 376, 377
⇝ 370
↔ 74
⊸ 288
⅋ 290
⊕ 290
⊗ 289
√ 179
⊢ 132
$\vdash_H$ 169
⊨ 59, 62, 104, 106
$\models_I$ 259
M_n 205
⇔ 57, 63, 73, 74, 76
⇒ 72, 76
⇐ 76
f_n 205
□ 235
¬ 69
∨ 70
$\overline{p}$ 69
$\overline{m}$ 215
$p(n)$ 201
◇ 235
$R_=$ 367
$R_\to$ 368
e **r** A 261
→ 72
∼ 69, 74
⊆ 155
≻ 330
φ_e 261
∧ 68
$\{x/t\}$ 81, 104, 311

1-aksioom (*1-axiom*) 291

!-salvestus (*!-storage*) 292

4 246
4◇ 246

5 246

aatom (*atom*) 76, 97, 100
Abelard, P. 26
abstraktsioonialgoritm 383
abstraktsiooniterm 377
absurd 3, 24
Ackermann, W. 37
Add 134
aeg 10
agent 286
aju 2
aksiomaatika 35–37, 120
aksiomatiseerima 18
aksiomatiseeritav
  lõplikult 222
aksioom (*axiom*) 11, 16, 40, 157, 159, 161
alamtuletus 141
alamvalem (*subformula*) 77
alamvalemi omadus (*subformula property*) 178, 255
al-Farabi, A. N. 25
α-reegel 179
algebra 35, 335
algebraline süsteem 162
al-Ghazali 25
algoritm (*algorithm*) 10, 12, 39, 44–46, 118, 255
  disjunktideks teisendamise 324
algoritmiteooria (*computability theory*) 46, 165
alguspunkt 26
al-Kindi 25
α-reduktsioon 377
Anaximenes 64
and 68
andmebaas (*data base*) 12, 227
  sebrade 333
  sugulussidemete 12
Anselm 26
antetsedent 174
antiikloogika 27
antisümmeetriline 258
apooria 20
`append` 355, 360
Aquino, T. 26
araabia 25
argument 58, 96
Aristoteles 6, 20–27, 56, 95, 235
aritmeetika (*arithmetic*) 15, 18, 34, 37, 133, 161, 178, 213, 218, 221, 225, 340
  arutlemise 28
  esimest järku 222, 224
  formaalne (*formal*) 45, 115
  keel 213
  mittestandardne mudel (*nonstandard model*) 214, 222, 224
  Presburgeri 18
  standardinterpretatsioon 214, 222
  sümbol- 362
  teist järku 222, 225
aritmeetikas väljendatav
  funktsioon 216
  predikaat 215
aritmeetiline predikaat 215
Arnauld 27
arutlus (*argument*) 19, 58, 65
  kehtiv (*valid*) 59, 60, 82, 85, 86, 88, 240, 387
  korrektne (*sound*) 60
arutluse standardkuju 59
arutlusvorm (*argument form*) 82, 85
  kehtiv (*valid*) 86, 88
arvutama (*compute*) 198
arvutatav funktsioon 197
arvutatavus (*computability*) 197, 249, 251
arvuti 198, 298
  abstraktne 44
  programmeeritav 44
  universaalne 44

arvutigraafika 358
arvutiteadus (*computer science*) 34, 46, 249
  teoreetiline 34, 37, 297
arvutus (*calculus*) 12, 133
arvutusmasin 44
asendus (*substitution*) 104
  literaali 321
  termi 322
asendusreegel (*replacement rule, equivalence rule*) 133, 135, 191
Assoc 135
assotsiatiivsus 79, 135
  parempoolne 79
  vasakpoolne 79
atomaarne valem 100
`atomic` 363
atomism 47
automaatne sünteesimine 39
automaatne teoreemitõestamine 297–299
Averroes 25
Avicenna 25

baas 189
Babbage, C. 44
Barcani valem (*Barcan formula*) 248
β-funktsioon 217
β-reduktsioon 377
β-reegel 179
Bendix 371
Bernays, P. 37
Bernoulli, J. 29
Bethi tabelimeetod (*Beth tableaux*) 179
Bic 135
binaarne (*binary*) 96
Blake, W. 168
Bochenski, J. 50
Boethius 25, 26
Bolyai, J. 94, 218
Boole, G. 28–32, 65
Boole'i
  algebra (*Boolean algebra*) 30, 32, 65, 120
  funktsioon (*Boolean function*) 66, 67, 71
  interpretatsioon 117
  konstant 227
  loogika 65
Boyer, R. 307
Brouwer, L. E. J. 38–39, 251
Buridan, J. 27
Burley, W. 27

`call` 362
Cantor, G. 29, 32, 34, 206, 218
Capella, M. 25
Carnap, R. 34, 40, 47, 48
CD 134
CE 135
Chomsky, A. N. 51
Chrysippus 24
Church, A. 33, 44–46, 131, 197, 208, 319, 376, 379
Churchi tees (*Church's thesis*) 45, 46, 197, 207, 251, 376
Cicero 25
Clark, K. L. 281
Cohen, P. J. 250
Comm 135
Conj 134
Contra 135
CP 140
Curry, H. 378, 382
Cut 176, 177

Davisi-Putnami meetod 299
Dedekind, R. 32, 213
deduktsioon 20, 375
definitsioon 51, 73
  induktiivne 76, 187
  matemaatiline 73
  nominaalne 28
  reaalne 28
deklaratiivne
  keel 346, 347
  programmeerimine 366
DeM 135

De Morgan, A. 29, 31
De Morgani seadused 135, 290
diagonaliseerimismeetod 204
Diodorus Cronus 24
disjunkt (*clause*) 300
  tautoloogiline 328
  tühi (*empty*) 305
disjunktide hulk 325
disjunktiivne süllogism 82, 132, 134
disjunktsioon 38, 69
  hägus- 269
  intuitsionistlik 38
  välistav (*exclusive or*) 70
Dist 135
distributiivsus 135
DN 135
DNK 137
dom 259
DS 132, 134
duaalne 290
  valem 189
duaalsusreeglid 108, 112

eeldus (*premise*) 22, 58
efektiivne 197
EG 151, 152
EI 151
Einstein, A. 41, 87, 218
eitus (*negation*) 69
  hägus- 269
  intuitsionistlik 38
eituse eitamise seadus 290
eksijäreldus (*fallacy*) 240
eksistentsikvantori (*existential quantifier*) 99
  eemaldamine (*existential instantiation*) 152, 172
  sissetoomine (*existential generalization*) 152
eksperiment 158
ekspertsüsteem 349
eksponent (*exponent*) 288
eksponentsiaalne keerukus 130
ekstensionaalne 84
ekstensioon 103
ekvivalentne (*equivalent*) 197
ekvivalents (*equivalence*) 73, 74
ekvivalentsireegel (*rule of biconditional*) 135
ekvivalentsiseos (*equivalence relation*) 247
elementaardisjunktsioon 137
elementaarkonjunktsioon 137
elementaarselt ekvivalentne 118
elementaarteooria 118, 119, 219, 225
empirism 47
enamus (*most*) 129
episteemiline (*epistemic*) 236
erand 4
erijuht (*instance*) 81, 150
erikuju (*instance*) 14, 81, 86
esteetika 48
Eudemus 23
Eukleides 24, 133, 157, 213
Eukleidese geomeetria 217
Euler, L. 29
ex falso quodlibet 173
Exp 135

`fact` 357
`fail` 362
`fakt` 366, 380
fakt (*fact*) 301, 349
faktoriseerimisreegel 300, 302, 317
Fermat, P. de 16, 196
Fermat' suur teoreem 16
figuur 22
filosoofia 47
  analüütiline 34, 47, 50
finiitne (*finite*) 37
finitism 37
formaalne
  aksiomaatika 43
  aksiomaatiline
    süsteem 133
    teooria 161
  süsteem (*formal system*) 40, 42
formaliseeritud aksiomaatiline teooria 158

formalism 33, 36, 38, 43
Frege, G. v, 29–35, 41, 94
funktsionaalmuutuja 222
funktsionaalne
  keel 347, 366
  programm 367
  programmeerimine 229, 366
funktsionaalsümbol 115
funktsionaalterm 313
funktsionalism 50
funktsioon 29, 66, 230
  injektiivne (*injective, one-to-one*) 224
  intuitiivselt arvutatav 197
  karakteristlik 205
  püsipunkti- 378
  sürjektiivne (*surjective, onto*) 224
  Turingi mõttes arvutatav (*Turing computable*) 204
füüsika 26, 51

Gen 171
Gentzen, G. 41, 172, 174, 178, 256
geomeetria 157, 213
Girard, J.-Y. 287
Goldbachi hüpotees (*Goldbach's conjecture*) 250
Gorgias 20
grammatika (*grammar*) 76
Gödel, K. 18, 19, 32–34, 37, 41–43, 48, 50, 133, 214, 217, 250, 251
Gödeli
  mittetäielikkuse teoreemid (*incompleteness theorems*) 229
  numeratsioon (*Gödel numbering*) 204, 219
  teoreemid mittetäielikkusest (*incompleteness theorems*) 236

hajuv 231
haru (*branch*) 179
  avatud lõpetatud (*completed open*) 181
  suletud (*closed*) 181
Hegel, G. W. F. 29, 47
Henkin, L. 224
Herbrand, J. 37, 170
Herbrandi
  baas 280
  universum 317
heuristika 335, 343
Heyting, A. 38, 39, 251
Heytingi arvutus 252
Hilbert, D. 32–38, 41, 133, 187, 218, 251
Hilberti
  probleemid 33, 36
  programm 37, 43, 218, 229
  tüüpi tuletussüsteem 168
Hippias 20
Horni
  disjunkt (*Horn clause*) 349, 354
  reegel 348
HS 134
hulga element 154
hulgateooria 32, 33, 35, 39, 41, 156, 218
  Cantori 32
  intuitsionistlik 39
  paradoksid 36, 38
hulk (*set*) 229
  kõigi hulkade 36
  lõplik 224
  lõpmatu (*infinite*) 32, 39, 41, 187, 224
Hume, D. 47
hüperresolutsioonireegel 337
hüpoteetiline süllogism 134

Ibn Rushd 25
Ibn Sina 25
iff 74
if-then 72
imperatiivne keel 346, 347, 367
implikatsioon 38, 71, 238

hägus- 270
intuitsionistlik 38
lineaarne 288
range (*strict*) 238
implikatsiooni
eeldus (*antecedent*) 72
järeldus (*consequent*) 72
implikatsiooni paradoksid (*paradoxes of (material) implication*) 238
implikatsioonireegel (*conditional exchange*) 135
implikatsiooni sissetoomisreegel (*conditional proof*) 140
indiviid 21
indiviidmuutuja 95
indiviidterm 100
induktiivne
struktuur 227
induktsioon 4, 5, 188, 340
matemaatiline 5
struktuurne 340
induktsiooni
baas 187, 188
eeldus 190
samm 187, 188
induktsiooniaksioom 214, 221
induktsiooniprintsiip 340
induktsioonireegel 178
induktsiooniskeem 340
infikskuju 97, 358
informatsioon 51
inimene 43, 198
injektiivne 214
`integer` 364
intensionaalne 51, 84
$inter$ 339
interaktiivne süsteem 340
interpretatsioon (*interpretation, structure*) 103, 117, 239
hägus- 269
lõplik 223
interpretatsiooni kandja 103, 117
introspektsioon 51
intuitsionism 33, 37–39, 249, 251
intuitsionistlik
funktsiooniteooria 39
hulgateooria 39
loogika 39
matemaatika 39
mõõduteooria 39
IP 144
`is` 357
islami fundamentalism 26
isomorfism 118
isomorfne 214

Jevons, W. S. 30
Johannes XXI 26
jumalatõestus 26
järeldama (*conclude, imply*) 5
järeldub loogiliselt 106, 162
järelduma (*imply*) 90
järeldus (*consequence, conclusion*) 22, 58
nõrk (*weak*) 271
tugev (*strong*) 271
järeldusreegel 7, 9
järeldustehe 31

K 246
kahekordse eituse reegel 135
kanooniline tõlge 244
Kant, I. 29, 39
kaudse tõestuse reegel 144
kausaalne (*causal*) 288
keel (*language*) 40, 49
eesti 76, 81
formaalne 8, 10, 12, 40
lausearvutuse 9
loomulik (*natural*) 10, 49, 50, 76
lõplik 41
predikaatarvutuse 10, 11
programmeerimis- 10
keelefilosoofia 48, 50
kehtestatav (*satisfiable*) 88, 106
kiirkontrolli meetod 89

Kleene, S. C. 170, 255, 261
KNK 137
Knuth, D. 371
Knuthi-Bendixi täielikustamisalgoritm 373
kognitiivsusteadus (*cognitive science*) 51
kohtupraktika 57
kohustuslikkus (*obligation*) 246
Kolmogorov, A. 167
kombinaator 383
super- 385
kommutatiivsus 135
kompaktne 228
kompaktsus 133
kompilaator
optimeeriv 385
konditsionaalse tõestuse reegel 140
konfiguratsioon 199, 211
konjunktsioon (*conjunction*) 68, 72, 79
hägus- 269
konjunktsioonireegel 134
konkretiseerimine 310
konstantmuutuja 151
konstantsümbol 96
konstrueerima 38, 288
konstruktiivne (*constructive*) 229
dilemma 134
konstruktivism 37
kontiinum (*continuum*) 126, 221
kontiinumhüpotees (*continuum hypothesis*) 250
kontingentne (*contingent*) 61, 63, 88, 89, 106
kontingentne materialist 248
kontradiktsioon 88, 110
kontranäide 90, 113
koolkond
Bagdadi 25
funktsionalistlik 50
intuitsionistlik 39
koondamine (*contraction*) 292
koonduv 231
koos (*with*) 289
kooskõlaline (*consistent*) 57
korda (*times*) 289
korrektne (*sound*) 160, 164
korrektsus (*soundness*) 15, 132, 192
Kowalski, R. 348
kriitiline paar 372
Kripke, S. A. 34, 50, 239
Kripke
mudel 259
semantika 261
kumer 230
kvantor (*quantifier*) 29, 31
olemasolu- (*existential*) 38, 99
üldisus- (*universal*) 98
kvantorite
distributiivsusreeglid (*quantifier distribution*) 146
eemaldamisreeglid 151
eitamisreeglid (*quantifier negation*) 146
ettetoomisreeglid 148
järjestusreeglid (*quantifier (in)dependence*) 147
liigutamisreeglid (*quantifier movement*) 147
sissetoomisreeglid 151
küllaldase aluse seadus 56

Laar, M. 84
lahendamatuse astmed (*degrees on unsolvability*) 46
lahendus (*solution*) 351
mitteühene 353
ühene 354
lahendusliteraal (*goal*) 320, 353
lahenduv (*decidable, solvable*) 12, 23, 45, 120, 159, 162, 164
raskesti (*hard, intractable*) 130
lahenduvus (*solvability*) 46
lambda-arvutus (*lambda calculus*) 45, 46, 347, 376

lambdaterm 377
Lambert, J. H. 29
lause (*sentence*) 65, 76, 101
  atomaarne 97
  osaeitav 109
  osajaatav 109
  üldeitav 109
  üldjaatav 108
lausearvutus (*propositional calculus*) 12, 23, 25, 29, 30, 65, 77
  klassikaline 45
lausearvutuse
  tehe (*connective*) 66
  valem (*formula, wff*) 76
lauseloogika (*propositional (sentential) logic*) 65
lausemuutuja 6, 76
lausevorm (*propositional (sentential) form*) 81, 90
Leibniz, G. W. 28–30, 34, 44, 56, 222, 236
Leibnizi võrduse printsiip (*Leibniz principle of equality*) 222
Lewis, C. I. 235, 238, 239, 245
ligipääsetavuse seos (*accessibility relation*) 246
lihtlause 387
  laiendamata 387
  laiendatud 387
lihtsustamisreegel 132, 134
liiasusreegel (*redundancy*) 135
liik (*sort*) 228
liikumine 20
liitlause 76, 387
Lindemann, C. L. F. von 197
linditähestik 198
lineaarne implikatsioon 288
lineaarsus 124
lingvistika (*linguistics*) 51
lingvistiline pööre vi
lisalemma 340
lisamisreegel 134
literaal 137, 179, 300
  negatiivne 300
  positiivne 300
literaalide järjestus 330
Lobatševski, N. 218
loend (*list*) 338, 354, 366
  tühi (nil) 338
loendi
  esimene element (*head*) 354
  saba (*tail*) 354
loenduv (*enumerable*) 126
logitsism 32–34, 36, 43, 47, 48, 50
logitsistika 50
loodusteadused 51
loogika 1, 6, 12, 19, 39–41, 51, 228, 236
  algebra 30
  Aristotelese 22, 23, 31, 95, 131
  autoepisteemiline (*autoepistemic*) 286
  deontiline (*deontic*) 235, 246
  dialektiline 1
  episteemiline (*epistemic*) 50, 235
  esimest järku (*first-order*) 222, 228
  filosoofiline 32
  formaalne 1, 10
  hägus- (*fuzzy*) 268
  intuitsionistlik (*intuitionistic*) 38, 39, 50, 175, 226, 249, 261, 288, 293, 354, 375
  kaasaegne 30
  kahevalentne 56
  keskaegne 27
  klassikaline (*classical*) 9, 38, 39, 261, 268, 287, 293, 354, 375
  kombinatoorne 382
  konstruktiivne 229
  kõrgemat järku (*higher-order*) 222

  lineaar- (*linear*) 50, 287, 288, 293
  lõplikku järku (*finite order*) 227
  matemaatiline 1, 10, 28
  mitmeliigiline (*many-sorted*) 228
  mitmevalentne (*many-valued*) 268
  mitteklassikaline (*non-classical*) 9, 34, 50, 65
  mittemonotoonne (*non-monotonic*) 50, 273
  mittepredikatiivne (*impredicative*) 227
  modaal- (*modal*) 50, 235
  monaadiline esimest järku (*monadic first-order*) 229
  monaadiline teist järku (*monadic second-order*) 229
  naiste 167
  Port-Royali 27
  predikaat- 94
  predikatiivne (*predicative*) 227
  relevantne (*relevance*) 50
  relevantsus- (*relevance*) 288
  skolastiline 27
  sümbol- (*symbolic*) 27–29
  teadmiste (*of knowledge*) 246
  teist järku (*second-order*) 222, 224, 226, 228
  temporaal- (*temporal*) 235
  terminite 21
  tänapäeva 29
  uskumiste (*of belief*) 246
  vaikimisi- (*default*) 284
loogikareegel 32
loogikaseadus 55
loogikateadus 19, 34
loogiline
  järeldumine 60, 62, 63, 238
  konstant 77
  positivism 40, 47, 48, 50
  programmeerimine (*logic programming*) 300, 348
  ruut (*square of opposition*) 111
  seos (*connective, relation*) 66
  sümbol 102
  tehe 76
loogiliselt (*logically*) 61
  ekvivalentne (*equivalent*) 62, 63, 74, 80
  mittevõimalik (*impossible*) 236
  samaväärne (*equivalent*) 118
  tõene (*true*) 61–63, 88, 91, 106, 240
  võimalik (*possible*) 236
  väär (*false*) 61, 63, 88, 106
loomuliku tuletuse süsteem 131, 172
Łukasiewicz, J. 34, 336
Łukasiewiczi aksioom 337
Lull, R. 28
Luther, M. 27
lõikepredikaat (*cut*) 360
lõikereegel (*cut*) 176, 257, 362
lõikereegli elimineerimine (*cut elimination*) 178
lõikevaba (*cut-free*) 178
lõplik (*finite*) 17
lõpliku mudeli omadus (*finite-model property*) 247
lõpmatus 32, 36, 37
  potentsiaalne 340
Löwenheim, L. 41, 46

MacColl, H. 238
mahajätmine (*dereliction*) 292

male 293
`map` 366
Markovi printsiip 251
masin
  Babbage'i 44
  Leibnizi 44
  Pascali 44
  Turingi (*Turing machine*) 44, 46
Maslov, S. 299
matemaatika 34, 43, 48, 157, 196, 218, 250
  alused 33, 36, 37, 249
  intuitsionistlik 39
  kaasaegne 30
matemaatiline
  analüüs 35, 158
  induktsioon 5, 187
matemaatilise induktsiooni
  meetod 189
  printsiip 35, 41, 189, 214
  reegel 17
McCarthy, J. 283
Melissus 64
mem 338
`member` 354
metakeel 40, 80
metamuutuja 81
metateoreem 32
`mgu` 314
Mill, J. S. 213, 266
Minsky, M. L. 235
mitteekvivalents 70
mitteeukleidiline geomeetria 218
mittelahenduv (*undecidable*) 12, 45, 131
mittelahenduvus (*unsolvability*) 46
mitteloogiline sümbol 102
mittepredikatiivne (*impredicative*) 229
mittetäielik (*incomplete*) 133, 218, 349
mittetäielikkus (*incompleteness*) 19
mittevastuoluline (*consistent*) 36, 37
mittevasturääkiv (*consistent*) 57, 163
modaalne (*modal*) 23
modaalsus (*modality*) 23
modus ponens 7, 132, 134, 169, 271, 302
modus tollens 134
monaadiline predikaatarvutus (*monadic*) 244
monotoonne (*monotonic*) 276
Montague, R. 50
Moore, G. E. 47
Moore, R. 286
moraal 20, 48
MP 132, 134
MT 134
mudel (*model*) 40, 41, 50, 104, 161, 223
muutuja (*variable*) 14, 21, 101
  asendamine 147
  seotud (*bound*) 101, 376
  seotud esinemine 101
  sisaldumise kontroll (*occur check*) 359
  vaba esinemine 101
  vaba (*free*) 101
muutujate ümbernimetamine (*standardising apart*) 313, 316, 324, 352
mõiste 21, 222
  defineerimine 73
mõistekiri v
mõisteloogika 21, 23, 25
mõte 3, 65
mõtlemine 2–6, 19, 47, 49
  selge 47
mäng (*game*) 49, 160, 358

$\mathbb{N}$ 116, 187, 213, 246
$\mathcal{N}$ 214
Naeris, V. 273
naturaalarv (*natural number*) 187
naturaalarvud (*natural numbers*) 206, 213

naturaalarvude hulk 187
neelama 327
neelamisstrateegia 327, 373
negatiivne
  informatsioon 280, 350
  tulemus 196
Neumann, J. von 37, 196
Newton, I. 87, 218
Nicole, P. 27
`nil` 338
nimede unikaalsuse aksioomid 281
nimi 21
Nixoni romb (*Nixon diamond*) 278
Nobeli auhind 34
$norm(a)$ 371
normaalkuju
  disjunktiivne 137
  konjunktiivne 137, 325
  preeneks- (*prenex*) 146
  Skolemi 150
  täielik
    disjunktiivne 137, 138
    konjunktiivne 137, 138
`not` 362
not 69
null (*nil*) 290
nullfunktsioon 216
nõrgendamine (*weakening*) 292

objektkeel 40, 80
objektorienteeritud keel 348
Ockham, W. 27
Ockhami habemenuga 27
olemasolukvantor 324
omaaksioom 162
omadus (*property*) 35, 46, 222
or 70
osahulk 155
osaline järjestus (*partial order*) 258
osavalem (*subformula*) 77
otsingu suund
  faktidest järelduseni (*bottom-up*) 332, 334
  järeldusest faktideni (*top-down*) 332–334
otsustus 65

P 201
paarikonstruktor 354
palju (*many*) 129
paradoks (*paradox*) 24, 33, 36
  Cantori 33, 35, 206, 218
  habemeajaja 35
  hääletamisea (*voting age*) 268
  inimeksistentsi (*of human existence*) 268
  kiilaspea (*of the bald man*) 268
  kuhja- (*of the heap*) 267
  Russelli 35, 218, 227
  soriitide (*sorites, of vagueness*) 267
  Zenoni 20
  valetaja (*liar*) 24, 35, 41, 42, 214
  õunte rivi 266
parajasti siis, kui 63
paralleelide aksioom (*parallel postulate*) 128, 217
paramodulatsioon 323, 336, 369
paramodulatsiooni
  täielikkus 323
paramodulatsioonireegel 323
paratamatu (*necessary*) 2, 6, 16
paratamatult tõene 236
paratamatus (*necessity*) 10, 50, 235, 238, 246
paratamatuse sissetoomine (*necessitation*) 238, 246
Parmenides 19
Pascal, B. 28, 44, 94, 213, 249
patsifism (*pacifism*) 34
Peano, G. 34, 35
Peano
  aksiomaatika (*Peano's axioms, Peano's postulates*) 162, 164, 213

aritmeetika 42
postulaadid 35, 37
peatehe (*main connective*) 77
peatumisprobleem (*halting problem*) 204, 205, 208
Peirce, C. S. 30–32
Peirce'i reegel 38, 336
`permutation` 355
permutatsioon 137
Petrus Hispanus 26
Philon 24
pidevusaksioom 221
piiramine (*circumscription*) 283
predikaadi 283
universumi (*domain*) 283
piisav (*sufficient*) 76, 387
Platon 20
platonism 43, 250
Ploucquet, G. 29
pluss (*plus*) 290
Porphyrios 25, 26
positivism 40
predikaadi aarsus (*arity*) 96
predikaat (*predicate*) 46, 95
predikaatarvutus (*predicate calculus*) 12, 23, 30, 42, 45, 46, 65, 94
esimest järku (*first-order*) 31, 41
tüüpidega 226
predikaatarvutuse
mittelahenduvus 319
valem 100
predikaatmuutuja 222
predikatiivne (*predicative*) 229
prefikskuju 97, 146, 149, 358
Priimägi, L. 55, 266
prioriteet 78
probleem (*problem*) 196
endal peatumise (*self-halting*) 205
keeruline (*hard*) 298
kurjuse (*of evil*) 166
mittelahenduv 201
peatumis- (*halting*) 205
ringi kvadratuuri (*squaring the circle*) 197
termide võrdsuse 368, 379, 382
tõesuse 208
vaba tahte (*of free will*) 237
virga kopra (*busy beaver*) 201, 203, 205
üldine peatumis- (*general halting*) 205, 220
Prodicus 20
produktiivsusfunktsioon 201
programmeerimine 10, 44
deklaratiivne 366
funktsionaalne 45, 366
programmeerimise paradigma 347
programmeerimiskeel
Ada 229, 366
Algol 229
assembler 346
Basic 346, 367, 376
C 346–348, 366, 367, 376
deklaratiivne 346
FP 366
funktsionaalne 376
funktsionaalse programmeerimise 227, 348
Haskell 347, 348, 366, 385
Hope 366
imperatiivne 346
Java 346, 348
Lisp 347, 367
loogilise programmeerimise 348
Miranda 366, 385
ML 348, 367
objektorienteeritud 229, 346, 348
Pascal 346, 348, 366, 367
Prolog 300, 347, 348, 350, 354, 376
protseduraalne 348
Scheme 348, 367
SmallTalk 348

programmide automaatne süntees 229
programmi verifitseerimine 338
Prolog 348
standardne 349
Prologi
= 358
aritmeetika 357
eitav vastus (*fail*, *no*) 351
eitus 350, 362
mittetäielikkus 359
otsingumootor (*engine*) 349, 354
programm 349
vastus (*answer*) 351
propositsioon (*proposition*) 65
Protagoras 20
protseduraalne keel 348
puhas mõistus 29
Putnam, H. 50, 299
puu 259
avatud (*open*) 181
suletud (*closed*) 181
põhjendama 19, 58
põhjus 49
päring (*goal*, *query*) 350
Päts, K. 83
pöördmeetod 299
püsipunkt (*fixpoint*) 378
püsipunktifunktsioon 378
Pythagoras 20

QD 147, 148
QI 148
QM 148
QN 148
`qsort` 356
quicksort 356
Quine, W. V. O. 50, 228, 248

rakendusfunktsioon 382
rakendusterm 377
Ramus, P. 27
ratsionaalarvud (*rational numbers*) 116
ratsionaalarvude järjestus 125
Raudam, T. 65, 130
reaalarvud (*real numbers*) 116, 123, 206
reaalarvude järjestus 123
reaktsioon 288
realiseeritav (*realizable*) 262
realism 250
Red 135
redeks 378
reductio ad absurdum 20, 22
reduktsioon 375, 377
redutseerima 376
reegel (*rule*) 4, 5, 16, 49, 301, 350
refleksiivne 247
refleksiivsus 321
Reiter, R. 280, 284
rekursiivne
programm 227
realiseeritavus 255, 261
rekursiivselt loetletav (*recursively enumerable*) 165, 220, 285
rekursiooniteooria (*recursion theory*) 46
religioon 48, 50
renessanss 27
resolutsioon
järjestatud 330
resolutsioonimeetod 131, 299–303, 349, 353
resolutsioonimeetodi
korrektsus 306
lahendav algoritm 308
täielikkus 307, 319
resolutsioonireegel 300–302, 311, 316, 330
ressurss 287
retoorikud 20
`reverse` 357
Riemann, G. F. B. 218
ring (nõiaring) 36
Robinson, J. 299, 307
Rosser, B. 380
Rummo, L. 94

Russell, B. 1, 32–36, 39, 41, 43, 47, 48, 50, 227
ruum 20
rühm (*group*) 158, 162
  Abeli 162
rühmateooria 119, 161, 335, 367

samaselt (*logically*) 61
  tõene (*true*) 61, 118
  väär (*false*) 61
samastama 222
samasus (*identity*) 10
samasusseadus (*law of identity*) 55, 58, 61, 89, 90, 287
Savisaar, E. 84
Schlick, M. 47
Schröder, E. 31, 32
Schönfinkel, M. 382
Scotus, J. 26
see (*the*) 107
seis (*state*) 293
seisund (*state*) 198
sekvents (*sequence*) 174
semantika (*semantics*) 40, 41, 48, 50, 79, 80, 160, 222
  võimalike maailmade (*possible world*) 50
semantiline puu 307
seos (*relation*) 83
  avaldatav 73
  tõeväärtusfunktsiooniline (*truth-functional*) 83
  väljendatav 73
signatuur (*vocabulary, signature*) 102, 103, 115
Siimann, M. 84
`simp` 364
Simp 132, 134
simuleerima 44, 45
sisu 29, 160
skolastiline 22
Skolem, T. 41, 46, 214, 225, 325
Skolemi
  funktsioon (*Skolem function*) 150
  konstant 150
skolemiseerimine (*Skolemization*) 324, 326
`slowsort` 357
sofistid 20
Sokrates 20, 21
sorteerimisalgoritm 356
spetsifitseerima 339
`split` 356
standardkonfiguratsioon 200
statistiline 4
stoikud 24, 25
strateegia
  järjestatud resolutsiooni 330
  järjestus- 331
  neelamis- 327, 331
  tautoloogia kõrvaldamise 328
  toetus- 331
  toetushulga 329
struktuur 116, 118
  lõplik 121
struktuuride klass 119
substitutsioon 81, 82, 311
  minimaalne 312
successor-funktsioon 217
sugulussidemete
  andmebaas 350
  ülesanne 12
suhe (*relation*) 46
suktsedent 174
suletud maailma
  eeldus (*closed-world assumption*) 280, 350, 362
  reegel 285
sõna (*word*) 49
süllogism (*syllogism*) 21–23, 112, 133
  kategooriline 27
süllogismifiguur 112
süllogismi moodused 133
süllogistika 23, 95
sümbolaritmeetika 362

sümbolarv (*numeral*) 215
sümmeetria 321
sümmeetriline 247
süntaks 40, 76, 80, 160
süntaksipuu (*parse tree*) 78, 365

$\mathbb{Z}$ 209
Zenon 20
Zermelo-Fraenkeli hulgateooria 157

T 246
t (*T*) 67, 79
tagasipöördumine (*backtracking*) 352
taju 48
Tarski, A. 40, 41, 102
tarvilik (*necessary*) 387
tarvilikkus (*necessity*) 76
tase (*level*) 227
tautoloogia (*tautology*) 88–91
tautoloogiliselt õige 6
teadmine (*knowledge*) 10, 49, 51, 235, 286
teadmiste baas (*knowledge base*) 280
teaduslik 47
teadvus 2
tegelik maailm (*real world*) 239
tegevus (*action*) 198
tehisintellektiteadus 34, 51, 297
teist järku predikaatloogika 221
teoloogia 26
  katoliiklik 26
teooria (*theory*) 12, 223
  aksiomatiseeritav 120
  esimest järku (*first-order*) 118, 162
  lahenduv 120
  loendite 340
  matemaatiline 115
  mittelahenduv 120
  mittemonotoonne (*nonmonotonic*) 276
  mittenormaliseeritav 340
  mudel 162
  N 213
  struktuuri 120
  struktuuride klassi 120
  vaikimisi- (*default*) 284
teoreem (*theorem*) 12, 143, 157
  asendus- (*replacement*) 134, 136
  Cantori 33
  Churchi 208
  Churchi-Rosseri 379, 385
  deduktsiooni- 170
  duaalsus- (*duality*) 190
  Fermat' 196
  Gödeli esimene 219
  Gödeli teine 220
  Henkini 224
  Herbrandi 317
  Knuthi-Bendixi 372
  kompaktsus- (*compactness*) 223, 224
  korrektsuse- (*soundness*) 162, 193
  loetlemis- (*enumeration*) 207
  matemaatika- 175
  mitteolemasolu 196
  mittetäielikkuse (*incompleteness*) 18, 19, 32, 41, 42
  mudeli- 163
  Nelsoni 264
  normaliseerimis- 378, 385
  püsipunkti- (*fixpoint*) 378, 385
  Skolemi 225
  Skolemi-Löwenheimi 41, 223
  Skolemi-Löwenheimi laskumis- (*downward*) 223, 228
  Skolemi-Löwenheimi tõusmis- (*upward*) 223, 228
  substitutsiooni- 384
  täielikkuse- (*completeness*) 41, 163, 195

ümbertõstmis- 137
term 116, 313
muutuja asendamiseks vaba 171
termi
normaalkuju 369
rakendamine 376
termiteisendussüsteem (*re-write system*) 369
kokkuvoolav (*confluent*) 370
korrektne 370
mittepeatuv 375
peatuv 369
täielik (*complete*) 370
tertium non datur 172
Theophrastus 23
tihedus 124
toetushulk 329
transitiivne 247
transitiivsus 321
TTS 369
tuletama 5
tuletamine (*deduction, inference, derivation*) 160
tuletatav (*derivable*) 305, 306
vahetult 159
tuletis 362
tuletus (*derivation, deduction*) 8, 10, 132
loomulik (*natural*) 172
tuletusreegel (*inference rule*) 11, 40, 82, 90, 131, 132, 134, 158, 159
tuletussüsteem
H 168, 171, 192
K 170, 246
ND 172
*S*4 246, 258
*S*5 239, 246
T 238, 245
Turing, A. M. 44, 46, 197, 319
Turingi masin 44, 45, 197, 198, 346
arvutab funktsiooni 201, 204
universaalne 207
Turingi masina
graaf (*graph*) 199
produktiivsus 201
Turingi tees (*Turing's thesis*) 197
tõde (*truth*) 2, 19, 38
absoluutne 19
matemaatiline 19
paratamatu 19
platooniline 38
tõehulk 103
tõendama 254
tõene (*true*) 56, 67, 79, 269
interpretatsioonis 105
tõestama 5, 157
tõestus (*proof*) 8, 10, 58, 158
finiitne (*finitary*) 218
induktiivne 187
intuitsionistlik 39
kaudne (*indirect*) 144
klassikaline 39
konditsionaalne (*conditional*) 140
mitteformaalne (*informal*) 156, 207
otsene 144
vastuväiteline (*indirect*, reductio ad absurdum) 56, 57, 144
tõestuste teooria (*proof theory*) 178
tõesus (*true*) 3, 41
analüütiline 50
loogiline 50
tõesusaste (*degree of truth*) 268
tõesuspuu (*truth-tree*) 179
tõesuspuude meetod 131, 179
tõeväärtus (*truth value*) 56, 66, 79
tõeväärtustabel (*truth table*) 68, 85–87, 130
osaline 86
tõstmislemma 318
tähendus (*meaning*) 40, 51
tähestik (*lexicon*) 76, 100, 159

täielik (*complete*) 17, 41, 160, 164, 301
täielikkus 133, 192
täielikkuse (*completeness*) 281
aksioom 281
eeldus (*assumption*) 281
täielikult modaliseeritud (*fully modalized*) 241
täielikustamine (*completion*) 281
täisarvud (*integers*) 17, 35, 39–41, 43, 116, 379
tühikusümbol 198
tüübiteooria 39, 226
lihtne 36
tüüp (*type*) 36, 227, 229
funktsiooni- 227
tüüpide hierarhia 36

UG 150, 151
UI 151
unaarne 96
unifitseerija (*unifier*) 312
kõige üldisem (*most general, mgu*) 313, 314
unifitseerima (*unify*) 311, 314
ühepoolselt 369
unifitseerimisalgoritm 315
universaalne
funktsioon 207
Turingi masin 207
universaalsus 36
universum (*universe*) 30, 31, 95, 103
arutluse (*of discourse*) 31
universumi suletuse eeldus (*domain-closure assumption*) 283
uskuma 19
uskumine (*belief*) 235

v (F) 67, 79
vaatlus 158
vaikimisireegel (*default rule*) 10, 274, 284
normaalne (*normal*) 284
poolnormaalne (*semi-normal*) 284
valem (*formula, wff*) 100, 116
atomaarne (*atomic*) 76
kinnine (*closed*) 101
lahtine (*open*) 101
lausearvutuse 76
vastandid (*contraries*) 110, 111
vastuolu (*contradiction*) 36, 38, 145
vastuoluline (*inconsistent*) 37, 57, 305
vasturääkiv (*inconsistent*) 57, 62, 63, 110, 111, 193
vasturääkivus (*contradiction*) 24
vasturääkivusseadus (*law of non-contradiction*) 55, 61, 90, 270
verifitseerima 48
Victorinus, M. 25
Viini ring 47
vorm 29
võimalike maailmade semantika 239
võimalikkus (*possibility*) 10, 50, 235, 387
võimalik maailm (*possible world*) 236, 239
võimsus 33
võrdsete asendamise seadus (*substituivity of equals*) 248, 322
võrdsus (*par*) 290
võrdus (*equality*) 321
hulkade 154
võrduse aksioomid 162
võrdussüsteem 367, 375
võtmesõna (*keyword*) 59
vähe (*few*) 129
väide 6, 35, 65
analüütiline 50
empiiriline 50, 51
kategooriline 21
tinglik (*conditional*) 31
väiteskeem 24
välistatud kolmanda seadus (*law of excluded middle*) 56, 57, 90, 252, 270

väljaviimisreegel (*exportation*) 135
väljendatavus 49
väär (*false*) 56, 67, 79, 269
väärtustus (*valuation*) 85, 90

Walden, A. 130
Whitehead, A. N. 32, 34, 36, 39, 41, 47
Wiles, A. 196
Wittgenstein, L. 1, 31, 32, 34, 47–49
`write` 358

õppima 4

ühisosa 339
üldistama 4
üldistamisreegel (*generalization*) 171
üldisuskvantor (*universal quantifier*) 98
üldisuskvantori
  eemaldamine (*universal instantiation*) 150
  sissetoomine (*universal generalization*) 150, 172
üleminek (*transition*) 293
ümberpööramisreegel (*contraposition*) 135

xor 70