

väidete algoritmiline analüüsimine

Olger Männik

March 31, 2022

Part I

Sissejuhatus

mõiste “väide” lahtiseletus: (võibolla on ka üldisem definitsioon kus kvantorid asendatakse, millegagi, mis tagastavad midagi muud kui booleane)

Sõna väide on selles failis formaalloogika tähenduses, aga see on väga sarnane selle sõna tavakeelsele tähendusele, ehk see on miski, mille tõele vastamisest järeldub mingi tõsiasi. Selles failis kasutan väidete kirja panemiseks kindlat syntaxit, kus väiteid kirjeldavad tekstilõigud võivad sisaldada boolean-operaatoreid, kvantoreid ja predikaate (või elementaarseid või algasju predikaatide asemel).

eesmärgid Eesmärk on koostada algoritm ja implementeerida see arvutiprogrammina, mille abil saaks väiteid minimaalsel vabadusastmel salvestada. Minimaalsel vabadusastmel salvestatud väiteid saaks kergesti analüüsida. Kasutaja sisestaks väited kergesti loetavas syntaxis (seda on kirjeldatud peatükis 19). Seda saaks kasutada näiteks kõige üldisemate matemaatiliste probleemide (mis võivad olla võrranditena esitatud) lihtsustamiseks, lahendamiseks ja lahendite kontrollimiseks.

Kuna iga väide on salvestatud minimaalsel vabadusastmel, siis saab lihtsasti kontrollida, et kas see on üheselt vale, üheselt tõene või mitte kumbagi, sest kõigile üheselt valedele ja üheselt tõestele väidetele vastab sama arv. Tõenäoliselt ongi praktilisel kasutamisel kõige olulisem küsimus, et kas väide vormis *aksioomid* \rightarrow *vaide* on üheselt vale, üheselt õige või mitte kumbagi (*programm*(*aksioomid* \rightarrow *vaide*)).

Kui väidete minimaalsel vabadusastmel salvestamine on võimatu, sellise algoritmi koostamine liiga raske, algoritm ajaliselt liiga keerukas et seda praktiliselt kasutada saaks või algoritm liiga palju mälu nõudev et seda praktiliselt kasutada saaks, siis on alternatiivseks eesmärgiks teha algoritmi, mis määrab, et kas sisendiks antud väide on vastuoluline või mitte ilma seda väidet minimaalsel vabadusastmel salvestamata ning implementeerida see algoritm arvutiprogrammina.

Kui ka sellise programmi tegemine pole võimalik (mõned inimesed on väitnud, et see pole kas Gödeli 2. ebatäielikkuse teoreemi või predikaatarvutuse mittelahenduvuse teoreemi kohaselt võimalik), siis on järgmiseks alternatiivseks eesmärgiks teha programm, mis piiramatult jätkab kontrollimist, et kas mingi väide on vastuoluline või mitte. Kuigi see programm ei pruugiks iga sisendi puhul tagastada, et kas sisendiks antud väide on kindlasti vastuoluline või see kindlasti ei ole vastuoluline (on kooskõlaline), tagastaks see, et peale kui mitut tsüklit ei leidnud see algoritm, et väide on vastuoluline.

Hea oleks käsitleda minimaalsel vabadusastmel kirjeldatud väiteid naturaalarvudena. Tähistan naturaalarvu, mis kirjeldab väidet S tähisega n_S . Selleks tuleb defineerida bijektiivne vastavus võimalike väidete (võimalik vist ainult kindlas notatsioonis olevate väidete korral) ja naturaalarvude vahele. Juhul kui selline vastavus defineerimine on võimalik, on võimalik defineerida lõpmatult erinevaid selliseid vastavusi. Tähistan funktsiooni, mis seab väite S vastavusse naturaalarvuga n_S F 'iga. Seega $F(S) = n_S$. Selle pöördfunktsiooni tähistan f 'iga. Seega $f(n_S) = S$. Järgnevalt loetlen omadusi, mis vastavusel (ja funktsioonil F) võiks olla, et sellisel kujul olevat väidet praktikas mugav kasutada oleks:

- Hea on kui sellisel kujul esitatud väidete vahelisi loogikatehteid (näiteks AND, NAND, OR, NOR ja XOR) saab kergesti arvutada rakendades väiteid kirjeldavatele naturaalarvudele (n_S) loogikatehele vastavaid lihtsasti arvutatavaid funktsioone.

- Hea on kui tavalises notatsioonis lühidalt kirjepandav väide on ka väikese või lihtsasti kirjeldatava naturaalarvuga vastavusse pandud.
- Hea on kui Falsele(kõigile üheselt valedele väidetele) vastab 0.
- Hea on kui Truele(kõigile üheselt tõestele väidetele) vastab 1.

lühemad eesmärgid prioriteetsuse järjekorras:

1. koostada algoritm, mis võtaks sisendiks väite ja tagastaks, et kas see on vastuoluline või mitte.
2. implementeerida see algoritm arvutiprogrammina.
3. koostada oma süntax , mida see programm sisendiks võtab, väidete mugavamaks kirjanemiseks.
4. lisada programmile funktsionaalsus, väitede salvestamiseks, kujul, mis on: Minimaalse vabadusastmete arvuga ehk mistahes 2 erinevat bittide kombinatsiooni (kui see on binaarkujul esitatud) tähistavad erinevaid loogikalisi lauseid ja universaalne ehk millega saab mingi bittide kombinatsiooniga (kui see on binaarkujul esitatud) iga loogikalise lause kirja panna. Teisisõnu funktsioon väite ja tähistuse vahel on bijektiivne.
5. lisada programmile funktsionaalsus esitada väiteid standardisel ja kergesti loetaval kujul.
6. lisada programmile funktsionaalsus hinnata väite keerukust mingi parameetriga.
7. lisada programmile funktsionaalsus määrata väite parameeter K, ehk ,et kui mitme kvantori sees kõige rohkemate kvantorite sees asuv kvantor on ,kui väide on esitatud vormis, kus kõige rohkemate kvantorite sees asuv kvantor on võimalikult väheste kvantorite sees.

Minu teada sama asja tegevad programmi ei ole veel tehtud. Sarnasemad programmid on Prolog ja Julog.jl , aga ned ei suuda nii üldiseid ja keerukaid ülesandeid lahendada kui programm, mida mina teha tahan suudaks, ning nende syntax on ebamugavam. Näiteks Wolfram ei oska lahendada lihtsamaidki sarnaseid ülesandeid. Sisend $\exists x(\exists y(A(x,y))) \wedge \neg \exists x(\exists y(A(y,x)))$, mille peale minu programm peaks tagastama, et see on vastuoluline on Wolframi syntaxis “Exists[x,Exists[y,A[x,y]]]&&!Exists[x,Exists[y,A[y,x]]]”, aga wolfram ei saa aru, et see on vastuoluline. Olen teinud isegi ühe foorumi postituse Wolframi foorumisee, kus inimesed kinnitasid, et Wolfram ei suuda selliseid ülesandeid lahendada. Selle postituse URL on <https://community.wolfram.com/groups/-/m/t/2416379> .

1 Näited programmi kasutamisest

1.1 lihtsamad näited

Järgnevates näidetes on kasutatud peatükis 19.3.1 kirjeldatud syntaxsugarit.

- Sellise väite korral programm tagastab, et väide on üheselt vale (lühendatult **ÜV**): $\exists(A(x_1)) \wedge \neg \exists(A(x_1))$ (sama väide ilma peatükis 19.3.1 kirjeldatud syntaxsugarit kasutamata $\exists_{x_1}(A(x_1)) \wedge \neg \exists_{x_1}(A(x_1))$), sest on väidetud, et miski, mis rahuldab predikaati A eksisteerib, ning samuti on väidetud, et seda ei eksisteeri.
- Sellise väite korral programm tagastab, et väide pole üheselt vale: $\exists(A(x_1)) \wedge \neg \exists(A(x_1) \wedge B(x_1))$ (sama väide ilma peatükis 19.3.1 kirjeldatud syntaxsugarit kasutamata $\exists_{x_1}(A(x_1)) \wedge \neg \exists_{x_1}(A(x_1) \wedge B(x_1))$), sest on väidetud, et eksisteerib miski, mis rahuldab predikaati A, aga ei eksisteeri midagi, mis rahuldaks nii predikaati A kui ka predikaati B ning vastuolu ei ole. Sellest on lihtsam aru saada kui panna predikaatidele mingid intuiitiivselt lihtsamini mõistetavad nimed. nt.: $\exists(on_loom(x_1)) \wedge \neg \exists(on_loom(x_1) \wedge on_auto(x_1))$. Ehk on väidetud, et eksisteerib mingi asi, mis on loom(exsisteerib mingi loom), aga ei eksisteeri midagi, mis oleks nii loom kui ka auto.
- Sellise väite korral programm tagastab, et väide on üheselt vale: $\exists(A(x_1) \wedge B(x_1)) \wedge \neg \exists(A(x_1))$, sest on väidetud, et eksisteerib miski, mis rahuldab nii predikaati A kui ka predikaati B, aga ei eksisteeri midagi, mis rahuldaks predikaati A.

- Sellise väite korral programm tagastab, et väide on üheselt vale: $\exists(\exists(A(x_1, x_2) \wedge \exists(A(x_2, x_3) \wedge A(x_2, x_2)))) \wedge \neg\exists(\exists(\exists(A(x_2, x_1) \wedge A(x_1, x_3) \wedge A(x_1, x_1))))$, sest on väidetud, et eksisteerib miski (x_2), mis rahuldab millegagi predikaati A olles ise 2. argument, rahuldab millegi muuga predikaati A, olles ise 1. argument ja rahuldab predikaati A olles selle mõlemaks argumendiks ning väite teises osas on väidetud, et selliste omadustega asja just ei eksisteeri.

Eelnevalt näiteks toodud väidete vastuolulisust on kerge intuiitiivselt, ilma arvuti abita kontrollida, aga pikkade ja keerukate väidete vastuolulisust on niimoodi väga raske kontrollida. Selliste valemite analüüsimiseks olekski see programm kasulik. Järgnevalt toon mõned näited keerukamatest väidetest, mille vastuolulisust on raske kontrollida:

- järgneva väite korral programm tagastab, et väide ei ole üheselt vale: $\exists(on_mari(x_1) \wedge \neg\exists(M(x_1, x_2) \wedge \exists(M(x_1, x_3) \wedge \neg M(x_2, x_3) \wedge M(x_3, x_1))) \wedge \exists(M(x_1, x_3) \wedge M(x_2, x_3)) \wedge \neg\exists(M(x_2, x_3) \wedge M(x_3, x_3)))) \wedge \exists(\exists(on_mari(x_2) \wedge M(x_2, x_1) \wedge \exists(M(x_1, x_3) \wedge M(x_2, x_3)) \wedge \exists(M(x_2, x_3) \wedge M(x_3, x_2) \wedge \neg M(x_1, x_3))) \wedge \neg\exists(M(x_2, x_1) \wedge M(x_2, x_2))) \vee \exists(M(x_1, x_1) \wedge \neg\exists(M(x_1, x_2) \wedge M(x_2, x_2) \wedge M(x_2, x_1))))$
Sellest, et vastuolu pole, on lihtsam aru saada kui panna predikaatidele intuiitiivselt lihtsamini mõistetavamad nimed. nt.: predikaat M on “ x_1 ’le meeldib x_2 ”, ehk kui $M(x_1, x_2)$, siis x_1 meeldib x_2 ’le. Siis on väidetud, et: [kõigil, kes Marile meeldivad, [ei ole kedagi kes Marile meeldiks, aga temale mitte ja kellele meeldiks Mari] või [ei ole kedagi, kes nii talle kui Marile meeldiks] või [on keegi, kes neile meeldib ja kes iseendale meeldib]] ja leidub keegi, kes meeldib marile, nii, et eksisteerib [keegi, kes meeldib nii marile kui talle] ja [keegi, kes meeldib marile, kellele mari meeldib ja kelle ei meeldi talle] ja ei leidu kedagi, kes nii iseendale kui ka talle ta meeldiks või [on keegi(x_1) kes endale meeldib, aga [kellel pole kedagi kes nii talle kui iseendale meeldiks ja kellele meeldiks tema(x_1)]].
- Sellise väite korral programm tagastab, et väide on üheselt vale: $\exists(A(x_1, x_1) \wedge \exists(A(x_2, x_2) \wedge A(x_2, x_1) \wedge A(x_1, x_2)) \wedge \exists(\neg A(x_2, x_2) \wedge \neg A(x_1, x_2) \wedge A(x_2, x_1) \wedge \neg\exists(\neg A(x_3, x_3) \wedge \neg A(x_3, x_2) \wedge A(x_3, x_1) \wedge A(x_2, x_3) \wedge A(x_1, x_3)))) \wedge \neg\exists(\exists(\exists(\neg A(x_1, x_1) \wedge \neg A(x_1, x_2) \wedge \neg A(x_2, x_1) \wedge \neg A(x_2, x_2) \wedge \neg A(x_3, x_1) \wedge \neg A(x_3, x_2) \wedge \neg A(x_3, x_3) \wedge \neg A(x_1, x_3) \wedge \neg A(x_2, x_3))) \vee A(x_1, x_1) \wedge \exists(A(x_2, x_2) \wedge \neg A(x_2, x_1) \wedge A(x_1, x_2) \wedge \neg\exists(\neg A(x_3, x_3) \wedge A(x_3, x_2) \wedge \neg A(x_3, x_1) \wedge A(x_2, x_3) \wedge A(x_1, x_3))))$

Siin on toodud mõned väited, mille üheselt valesust saaks minu programmiga kontrollida, aga mille vastuolulisuse kontrollimiseks on vaja kasutada matemaatiliste tehte märkide tähendusi kirjeldavaid väiteid ja arvude kohta käivaid aksioome. Programmi kasutaja ei peaks neid käsitsi sisestama vaid saaks need standard libarist võtta. Standard libariks olevat väidet tähistan siinkohal nimega STANDARD. Samuti on selleks vaja kasutada mõndasid syntax-sugareid.

mitte ÜV:

- $STANDARD \wedge \exists(x_1 + 3 = 7 \wedge on_reaalarv(x_1))$
- $STANDARD \wedge \neg\exists(0 * x_1 = 3 \wedge on_reaalarv(x_1))$
- $STANDARD \wedge \neg\exists((on_naturaalarv(x_1) \rightarrow x_1 > x_1) \vee x_1 = x_1 + 1)$
- $STANDARD \wedge \exists(on_reaalarv(x_1) \wedge \exists(x_1 + x_2 = 23 \wedge x_1 + 2 * x_2 = 37 \wedge x_1 < 53 \wedge x_1 * x_2 < 200 \wedge on_reaalarv(x_2)))$
- $STANDARD \wedge \neg\exists(\neg\exists(x_1 > x_2 \wedge on_reaalarv(x_1) \wedge on_reaalarv(x_2)))$ ei leidu arvu, millest suuremat ei leiduks.
- $STANDARD \wedge \neg\exists(\exists(x_1 > x_2 \wedge x_1 < x_2 \wedge on_reaalarv(x_1) \wedge on_reaalarv(x_2)))$ ei leidu kahte arvu nii, et mõlemad oleksid teisest suuremad.
- $STANDARD \wedge \exists(\neg\exists(x_1 * x_2 \neq 0 \wedge on_reaalarv(x_1) \wedge on_reaalarv(x_2)))$ Leidub arv(0), nii et ei leiduks arvu, millega seda korrutades ei saaks vastuseks nulli.
- $STANDARD \wedge \exists(\exists(x_1 + x_2 = 9 \wedge x_1 * x_2 = 20 \wedge x_1 > x_2 \wedge on_reaalarv(x_1) \wedge on_reaalarv(x_2)))$ Need arvud on 5 ja 4.

ÜV:

- $STANDARD \wedge \exists(\exists(x_1^3 - 4 * x_1^2 + 6 * x_1 - 24 - x_2^4 + 3 * x_2 = 0 \wedge x_2 > 1))$
- $STANDARD \wedge \exists(on_reaalarv(x_1) \wedge \neg\exists(x_1 < x_2 \wedge on_reaalarv(x_2)))$ eksisteerib selline reaalarv, nii et ei leidu ühtegi teist reaalarvu, mis sellest suurem oleks.

- $STANDARD \wedge \neg \exists (on_kompleksarv(x_1) \wedge \exists (x_1 < x_2 \wedge on_kompleksarv(x_2) \wedge x_1^2 = x_2^2))$ ei leidu kahte kompleksarvu, mille ruut oleks sama ja millest esimene oleks teisest suurem.
- $STANDARD \wedge \neg \exists (on_kompleksarv(x_1) \wedge \exists (on_kompleksarv(x_2) \wedge \exists (on_kompleksarv(x_3) \wedge x_1^5 = x_2^5 \wedge x_2^5 = x_3^5 \wedge x_1 \neq x_2 \wedge x_2 \neq x_3 \wedge x_1 \neq x_3)))$ ei leidu 3e erinevat kompleksarvu, mille 5 aste oleks sama.

Saaks kontrollida ka, et kas mingi hüpotees(teoreem) järelneb aksioomidest(ehk standard libarist). Kui standard libaris on piisavalt palju infot, siis järgneva seose korral programm tagastab, et see seos on ÜV, ehk et see hüpotees oli tõene.

$\neg (STANDARD \rightarrow \neg \exists (on_naturaalarv(x_1) \wedge \exists (on_naturaalarv(x_2) \wedge \exists (on_naturaalarv(x_3) \wedge \exists (x_1^{x_4} + x_2^{x_4} = x_3^{x_4} \wedge x_4 > 2 \wedge on_naturaalarv(x_4))))))$

- (sama väide ilma peatükis 19.3.1 kirjeldatud syntaxsugarit kasutamata $\neg (STANDARD \rightarrow \neg \exists_{x_1} (on_naturaalarv(x_1) \wedge \exists_{x_2} (on_naturaalarv(x_2) \wedge \exists_{x_3} (on_naturaalarv(x_3) \wedge \exists_{x_4} (x_1^{x_4} + x_2^{x_4} = x_3^{x_4} \wedge x_4 > 2 \wedge on_naturaalarv(x_4))))))$))

1.2 Matemaatiliste mõistete vahelise väite kirjeldamine.

1.3 algoritmi või arvutiprogrammi analüüsimiseks

igal pool asendada käsk $x:=y$ väitega $x(t+1) = y(t)$, kus t on programmi samm.

1.4 Füüsikalise mudeli kirjeldamiseks

Oluline on ka näidata, et milliste erinevate väärtuste korral on süsteemid füüsikaliselt identsed. Näiteks ei ole oluline potentsiaali 0-nivoo ja koordinaatide alguspunkt.

Oluliseks pointiks nii tõenäosuste kui geomeetrisel ruumi(intuitiivsetst vektoritest) on, et intuitiivse tähendusega aja korrutus arvuga on samuti intuitiivne. See on vist kuidagi taustsüsteemiga seotud. taust süsteemi, mille algus on kohas \vec{a} olev vektor \vec{b} asub kohas $\vec{a} + \vec{b}$.

Füüsikalise väite(mudeli) kasulikkus on seda suurem, mida täpsemalt see katseandmetega klapi ja lihtsam on(mida väiksem on selle keerukus). Lihtsust on ilmselt õige hinnata kui füüsikalised suurused on kirjapandud minimaalsel vabadusastmel ja võimalikult sümmeetriliselt(näiteks ruumiline ja ajaline sümmeetria), et selgelt eristada matemaatilisi algmõisted (funktsioon, võrdsus, hulk, implikatsioon jne(siin näitena toodud mõisted ei moodusta minimaalset funktsionaalselt täielikku hulka)) ja füüsilisi algmõisted (ruum, aeg, elektromagnetväli, elektrimagnetväli, gravitomagnetväli, gravitomagnetväli, aegruum). Võtta kasutusele ka füüsikalised mõisted, mis ei ole algmõisted (nt. konkreetseid ained, tahke, vedel, gaasiline, aatom).

Ükski tõestamatu väide ei tohi omada füüsikalist tähendust(näiteks ei oma koordinaatide alguspunkti valik füüsikalist tähendust.) ja ükski füüsikalise tähenduseta väide ei tohi olla tõestatav.

Füüsikalise mudeli, mis sisaldab suuruseid(nt. Aeg, kordinaat ja väljatugevus) ja iselommustakse reaalarvuga, kirjeldamiseks tuleb defineerida vastavad aritmeetilised tehked reaalarvudele. Kas alati tuleb defineerida ka jagamine, et saaks väita nt. ,et üks suurus on teisest 3 korda suurem?

1.4.1 mudeli lihtsus

Katseandmetega paremini kokku sobiv mudel on tõenäolisem. Lihtsam mudel tõenäolisem. ##tekstist on informatsioon puudu, vajab täiendamist

Mudelite lihtsuse võrdlemiseks peavad neil olema samad elementaarseosed. Kui ühel mudelil on rohkem predikaate või algasju kui teisel tuleb need eksistentsiaalsuskvantori abil eemaldada(näiteks ühes on ainult punktkehad ja teises ka väli, mille väärtus määrab, et kui suure tõenäosusega punktkeha igas punktis asub). Keerukus arvutatakse juba töödeldud väitest. võibolla on see keerukus Gödeli keerukus?

Lisaks tavalisele keerukusele defineerida teine mõiste keerukus2 nii , et $(\frac{sobivusandmetega(mudel1)}{keerukus2(mudel1)} > \frac{sobivusandmetega(mudel2)}{keerukus2(mudel2)}) \rightarrow (mudel1 \text{ "on parem mudel kui" } mudel2)$

idee1 keerukuse hindamiseks Üks võimalus mudeli lihtsuse kirjeldamiseks on hulga, mis sisaldab kõiki selle mudeliga sobivaid erinevaid füüsikalisi süsteeme võimsus. Kõik PML süsteemid saab kirjeldada ühe reaalarvuga (järjend, mis sisaldab kõiki PML üht reaalarvu)

idee2 keerukuse hindamiseks Veel 1 võimalus keerukuse hindamiseks: reaalarvude arv, mida on vaja mistahes mudeliga sobitava süsteemi kirjeldamiseks nii, et mistahes diferentsiaalse muutuse süsteemis saab teha mingi nendest reaalarvudest diferentsiaalselt väiksel määral muutet.

Part II

funktsiooni F struktuur

Kuna väide tohib koosneda ainult booleanalgebrast (taseme 1)kvantoritest ja argumentidest saab F'i kirjeldada väite abil, mis nende kvantorite vahelise booleanfunktsioonile naturaalarvu vastavusse paneb. Kuna erinevaid kvantoreid on lõpmatult

Seega saab vaadata iga funktsiooni F nendest osadest koosnevana $F(vide) = F_{boolean_funktsioon_naturaalarvaks}(F_{nummerda\ lplik\ arv\ kvantoreid} n_s)$. Jaotasin F'i osadeks, et selle defineerimise erinevaid võimalusi oleks lihtsam kirjeldada oleks. **kvantorite see võib ja k booleanfunktsioone olla**

väite argumente on alati lõplikult. Erinevaid väiteid on loenduv lõpmatus. Iga väidet saab kirjeldada booleanfunktsiooni võimalikest kvantoritest kaudu. Kuna võimalikke kvantoreid on lõpmatult tuleb võtta mingi lisamuutuja t , mis seab kvantoritele piirid nii, et iga t väärtuse korral on võimalikke kvantoreid lõplik arv ja ükskõik milliseid kvantoreid sisaldavad veerud saab mingi t väärtuse korral.

2 filtreerida ja nummerdada veerud

Iga väite kirjeldamiseks piisab veergude vahelist booleanfunktsiooni. Osad veergude booleankombinatsioonid on üheselt valed.

3 konverteerida booleanfunktsioon naturaalarvuks

Part III

definiitsioonid

Tasemefunktsioon h_i on bijektiivses seoses kõigi elementaarfunktsioonide väärtustega, kui elementaarseoste vähemalt üheks argumentiks on taseme i kvanteeritav ja ei ole kõrgema taseme kvanteeritavaid. Elemndi põhisest notatsioonist sellesse notatsiooni konverteerimisel võivad elementaarseoste argumentideks olla ka algasjad. Tasemefunktsiooni võimalikku väärtuste arvu tähistan $H(k)$.

Part IV

Võimalusi argumentide ja kvanteeritavate vahelise suhte noteerimiseks

Väite argumentid on reaalsed asjad(mitte fiktiivmuutujad), mille kohta väide midagi ütleb. Sõltuvalt notatsioonist võivad võivad väite argumentid olla mitmel erineval kujul. Selles töös on käsitletud 3e erinevat kuju argumentidele, milleks on **predikaat**, **elementaarseos** või **algasi**. Et väitest järeleduks midagi mingi asja kohta, ehk et väide mingi asja kohta saaks käia peab see seda asja argumenina sisaldama. Argumenti kuju sõltub notatsioonist, mille abil väite argumenti ja muu(kvanteeritavate) vahelist suhet kirjeldatakse. Et väitel oleks intuitiivselt mõistetav tähendus, peab väite argument sisaldama midagi intuitiivselt mõistetavat. Vastavate argumentide valimise abil saab defineerida kõik matemaatilised mõisted.

Väite argumendi ja muu vahelist suhet võib kirjeldada ükskõik mis notatsioonis, milles saab suvalise lõpliku arvu võimalusi kirjeldada. Kuna nii on kirjutatud lihtne mõista ja see on levinud notatsioon on siin peamiselt kasutatud hulga-põhist-notatsiooni.

Notatsioone saab üksteiseks ümber kirjutada ilma väidet muutmata. Väide on notatsiooni transformatsiooni suhtes invariantne.

4 argumendiks on kasutaja sisestatud predikaadid

Predikaadi notatsioonis võib väite argumendist mõelda kui funktsioonidest, mis tagastavad booleane (kas True või False). Algväited võivad olla näiteks “x on null”(x), “x on hulk”(x), “x on naturaalarv”(x) “x on funktsioon”(x) või “x_1 on suurem kui x_2”(x_1,x_2).

4.1 eriasjad

eriasjad on nagu predikaadid, aga võtavad kvanteeritavate asemel argumendiks kas booleane või booleanseoseid. Muidu võiks sulgusid ka eriasjaks pidada, aga need lihtsalt, mis mille argumendiks on. Kas siis kui kõigi predikaatide argumentide arv on teada pole sulge vaja? $A(B(3,4),C(1,2))$ vs $A(B(3,4),C(1,2))$

4.1.1 Võimalikke eriasjade valikuid:

- $\forall(B)$, NOR
- $\exists(B)$, NAND
- $\forall(NOR(B1,B2))$ (See on sama, mis $NOR(\forall(B1),\forall(B2))$)
- $\neg\forall(OR(B1,B2))!$
- $NOR(\forall(B1),B2)$
- $OR(\neg\forall(B1),\neg B2)$
- $\forall(NAND(B1,B2))$
- $\neg\forall(AND(B1,B2))$ (See on sama, mis $NAND(\forall(B1),\forall(B2))$)
- $NAND(\forall(B1),B2)$
- $AND(\neg\forall(B1),\neg B2)$

Mõistlikum on noest ilmselt See valida, mille korral algoritm kõige lihtsam tuleb. $NORK$ ehk $NORK = \forall(NOR(B1,B2)) = NOR(\forall(B1),\forall(B2)) = \neg\forall(B1) \wedge \neg\forall(B2) = \exists(\neg B1) \wedge \exists(\neg B2)$, tundub hea, sest saab kergesti DNF vormi kirjutada sellega

5 Argumendiks on elementaarseosed

Selles notatsioonis on väidete argumentideks elementaarseosed kvanteeritavate vahel. Miski muu peale kvanteeritavate ei tohi elementaarseoste argumendiks olla. Elementaarseosed on notatsiooni poolt ette antud ja kasutaja neid lisada ega eemaldada ei saa.

Võimalikud elementaarseose põhised notatsioonid erinevad selle poolest, et kui mitme asja kaupa kvanteeritavate vahelisis elementaarseoste väiteid kirjeldatakse ja et kui mitmel erineval viisil saavad samad asjad samas elementaarseoses olla. Tähistan arvu, mille kaupa asjade vahelist seost kirjeldatakse a'ga ja erinevate elementaarseoste arvu b'ga.

Kas iga väidet saab ümber kirjutada a ja b väärtustega kirjeldatuks ilma algmõisteid muutmata? a Kas iga väidet saab ümber kirjutada a ja b väärtustega kirjeldatuks ilma algmõisteid muutmata? a ja b väärtustega kirjeldatuks ilma algmõisteid muutmata?

Kas elementaarseostel peab ka intuitiivne tähendus olema, et väitel intuitiivne tähendus oleks?

Võimalikud elementaarseose põhised notatsioonid erinevad järgnevate omaduste poolest

- Kas elementaarseose väide sõltub argumentide järjekorrast. Ehk kas $A(x_1, x_2) = A(x_2, x_1)$ (kas saab mitte sümmeetrilisi seoseid kirjeldada.)
- Mitu argumenti on elementaarseostel. Ehk, et kui mitme asja kaupa kvanteeritavate vahelisis elementaarseoste väiteid kirjeldatakse. Tähistan arvu, mille kaupa asjade vahelist seost kirjeldatakse a' ga.
- Kui palju erinevaid võimalikke väärtusi kindlate argumentidega kindlal elementaarseose väite olla saab. Üldiselt eeldatakse, et see arv on 2 (True ja False).

Väite argumenti kohta vähem eeldusi tehes saab mõndasi väiteid mälu efektiivsemalt kirjeldada. Rohkem eeldusi tehes on argumente lihtsam tõlgendada.

5.1 väited elementaarseote kohta?

Osade väidete formuleerimise sõltub väites olevatest predikaatidest. Nii, et kui kasutaja sisestab uue predikaadi tuleb need väited alati ümber kirjutada. See teeks minu programmi kasutamise ebamugavaks.

Näiteks võrduse väitmiseks tuleb lisada eraldi võrduse elementaarseos " x_1 ja x_2 on võrdsed"(x_1, x_2). Et selle tähendus oleks kooskõlaline selle intuiitvise tähendusega, peab väitele lisama väite, et kui 2 asja on võrdsed, siis pole ükski elementaarfunktsioon neist asjadest erineva väärtusega (ükskõik mitu argumenti elementaarfunktsioonil on ja mis elementaarfunktsioonide ülejäänud argumentideks on) (See võrdusmärgi asi võib olla seotud Gödeli II teoreemiga). Kõigi argumentide puhul ei pruugi kehtida vastupidine, et kui kõik elementaarseosed 2 asjast on võrdsed, siis on need asjad ise võrdsed. Peale võrdusmärgi peab teisi väiteid, mis peavad kõigi predikaatide korral kehtima formuleerima kõigi predikaatide valikute puhul erinevalt. Kuna predikaadid ja nende argumentide arv on teada saab universaalsuskvantori, mis kehtib predikaatide kohta asendada jaatusega ja eksistentsiaalsuskvantori võitusega. Saab lisada ka mingi syntaxi, mis automaatselt lisab, et mingi väide kehtib kõigi predikaatide kohta. Osade argumentide puhul võib väitele lisada veel mingeid elementaarseoseid elementaarseosega " x_1 ja x_2 on võrdsed"(x_1, x_2) siduvaid väiteid nagu näiteks, et kui 2 hulka sisaldavad samu asju, siis on nad võrdsed.

Näiteks funktsioonide defineerimisel tuleb väide, et leidub funktsioone, mis $on_funktsioon$ lisada erinevate predikaatide valikute puhul erinevalt. Muidu ei saaks näiteks Peano aksioomi $\forall x_1 ("on_funktsioon"(x_1) \rightarrow \forall x_2 (x_2 \in N \rightarrow ("rakenda"(x_1, x_2) \rightarrow "rakenda"(x_1, x_2 + 1)) \wedge "rakenda"(x_1, 0) \rightarrow \forall x_2 (x_2 \in N \rightarrow rakenda(x_1, x_2))))))$ kasutada, sest väites ei ole kirjas, et leidub funktsioone, mis kõiki erinevaid naturaalarvusi asju eristavad. Kui väites on kirjas, et mingi predikaat on rahuldatud 0 puhul ja sellest, et see mingi arvu puhul rahuldatud on järeldub, et see on sellest arvust ühe võrra suurema arvu puhul rahuldatud ja, et see ei ole mingi naturaalarvu puhul rahuldatud, ei saa programm seda väidet Falseks lihtsustada, sest väites ei ole kirjas, et leidub funktsioon, mis on tõene ainult nende elementide puhul, mis seda predikaati rahuldavad.

Hulkade defineerimisel tekib analoogne probleem ehk, et ei ole kirjas, et leidub mis tahes sialduvusega hulkasid.

Üks võimalus on teha vaikiv eeldus, et kõik hulgad eksisteerivad ja anna kirja, et millised teised asjad eksisteerivad hulkade kaudu. Näiteks:

- võrduse probleem: $\forall (x_1, x_2) \leftrightarrow \forall (on_hulk(x_3) \rightarrow (x_1 \in x_3 \leftrightarrow x_2 \in x_3))$
- see, et kõik 2 argumentilised funktsioonid eksisteerivad: $\forall (\forall (\forall (\forall (\forall (on_hulk(x_1) \wedge on_hulk(x_2) \wedge on_hulk(x_3) \wedge x_4 \in x_1 \wedge x_5 \in x_2 \wedge x_6 \in x_3 \rightarrow \exists (on_funktsioon(x_7) \wedge rakenda(x_7, x_4, x_5, x_6)) \wedge \exists (on_funktsioon(x_7) \wedge \neg rakenda(x_7, x_1, x_2, x_3)))))))$

5.1.1 Võimalikud lahendused

kvanteerimine üle predikaatide igale universaalsuskvantorile lisada jaatusega, et sama väide kehtib ka predikaatide kohta ja konstantfunktsioon "rakenda" asendada predikaadi rakendamisega.

mingid konstantsed funktsioonid, mille abil saab kõik väite kõigi argumentide korral kirjeldada, mida interpreteeritakse eeldatava väitega, mis sisaldab jaatust üle kõigi kvantorite Äkki võrdusmärgist piisab? Näiteks hulkade puhul: $\forall(\forall("onhulk"(x_1) \wedge \neg(x_2 \in x_1) \rightarrow \exists("onhulk"(x_3) \wedge (x_2 \in x_3) \wedge \forall(x_4 \neq x_2 \rightarrow (x_4 \in x_1 \leftrightarrow x_4 \in x_3)))))) \wedge \exists("onhulk"(x_1) \wedge \forall(\neg(x_2 \in x_1)))$ või $\forall(\forall("onhulk"(x_1) \wedge \exists("onhulk"(x_3) \wedge \forall(x_4 = x_2 \leftrightarrow (x_4 \in x_1 \leftrightarrow x_4 \in x_3)))))) \wedge \exists("onhulk"(x_1)) \wedge \forall(\forall("onhulk"(x_1) \wedge "onhulk"(x_2) \rightarrow (x_1 = x_2 \leftrightarrow \forall(x_3 \in x_1 \leftrightarrow x_3 \in x_2))))$

Kas see loendumatute hulkade puhul kehtib? Vaja on põhjendust, et iga väite puhul piisab võrdusmärgist, et väite kuju ei sõltuks elementaarseoste valikust.

mingid konstantsed funktsioonid, mille abil saab kõik väite kõigi argumentide korral kirjeldada, mida programm predikaatide lisamisel automaatselt täiendab Äkki võrdusmärgist piisab?

Äkki piisab funktsioonist L , mapib kõigi predikaatide väärtused, kui nende argumentideks on kõik erinevad variatsioonid ülejäänud L 'i argumentidest, mis sisaldavad (vähemalt 1 korra) L 'i esimest argumenti, naturaalarvudele. Hulkade (ja ühe argumndiliste funktsioonide, mis tagastavad booleane) puhul näiteks: $\Pi_{b=0}^{2^{H(a)}} (\forall_{Ax} (\exists_Y (\sum_{i=0}^{H(A)} (L(Y, x, x..x) = i \wedge b[i])) \rightarrow \exists_{x_h} ("onhulk"(x_h) \wedge \forall_e (\sum_{i=0}^{H(A)} (L(e, x, x..x) = i \wedge b[i]) \leftrightarrow (e \in x_h))))))$

A on predikaadi, millel on kõige rohkem argumente, argumentide arv. \forall_{Ax} tähendab A kvantorit

või lausa:

$\Pi_{b=0}^{2^{H(a)}} (\forall_{Ax} (\exists_{x_h} ("onhulk"(x_h) \wedge \forall_e (\sum_{i=0}^{H(A)} (L(e, x, x..x) = i \wedge b[i]) \leftrightarrow (e \in x_h))))))$

võrdusmärgi korral: ...

kõik need väited saab teadaolevate predikaatide abil kirja panna nii, et jaatus üle b 'de, võitus üle i 'de ja funktsioon L jäävad ära.

5.2 Elementaarseosed eraldi

Sama argumentide arvuga elementaarseoseid võib olla mitu.

5.3 Iga argumentide arvu jaoks üks elementaarseos

Kirjeldada kõiki võrdsete argumentidega elementaarseoseid ühe elementaarseosega f_Q (kvanteeritavad). Tuleb otsustada, et kas seosefunktsiooni väärtus sõltub argumentide järjekorrast. Kirjeldatavast väitest sõltub, et kui paljude argumentidega seosefunktsioone on ja, et kui palju on elementaarseosetel võimalikke väärtusi.

Et konverteerida väide notatsiooni "Elementaarseosed eraldi" notatsiooni "Iga argumentide arvu jaoks üks elementaarseos" tuleb tuleb uued elementaarseosed viia bijektiivsesse sõltuvusse kõigi vanade võrdsete argumentide arvuga elementaarseoste väärtustega.

Kui selle funktsiooni tagastatavad väärtusi tõlgendada naturaalarvudena ei ole probleemiks, et väites ei ole veel naturaalarve defineeritud, sest f_Q tagastavatele elementidele ei ole vaja rakendada naturaalarvude kohta käivaid aksoome.

5.3.1 täpsemad stanardid

Seosefunktsiooni väärtus ei sõltu argumentide järjekorrast. Ei täpsusta et kui paljude argumentidega seosefunktsioone on ja, et kui palju on seosefunktsioonidel võimalikke väärtusi.

Kui selle funktsiooni tagastatavad väärtusi tõlgendada naturaalarvudena ei ole probleemiks, et väites ei ole veel naturaalarve defineeritud, sest f_Q tagastavatele elementidele ei ole vaja rakendada naturaalarvude kohta käivaid aksoome.

6 Elemendi põhine ehk argumendiks on algasjad

kõikide argumentide arvudega seosefunktsioonid saab lihtsalt panna kirja kahe argumendiga seosefunktsioonidega. lihtsalt võtta uus kvanteeritav ja siduda kõik algse seosefunktsiooni argumendid sellega uue 2 argumendise seosefunktsiooni abil. nt $f_{mingi}(a, b, c) = \exists_x (f(x, mingi), f(x, a), f(x, b), f(x, c))$. Mitmeargumendiliste seosefunktsioonide puhul see ei kehti.

(Kas tohib teha väiteid konstantsete elementaarseoste kohta)

Selles notatsioonis on väite argumentideks algasjad ehk algmõisted, mida lisaks kvanteeritavatele e

Selle notatsiooni halvaks küljeks on, et tekib mitmeid väiteid, mida on raske tõlgendada nt $g(+,+,*)$ või $g(>,>)$.

Näiteks elementaarsuhte “x on null” konverteerimiseks elementaarseopõhisest notatsioonist elemendipõhiseks tuleb elementaarfunktsiooni $\text{on_null}(x_1)$ asemel võtta algasi 0.

Notatsiooni heaks küljeks on, et saab eemaldada algmõisteid väitest. Näiteks kui algmõiste A4 on defineeritud algmõiste A3 kaudu, algmõiste 34 on defineeritud algmõiste A2 kaudu ja algmõiste A2 on defineeritud algmõiste A1 kaudu, siis saab eemaldada väitest algmõisted A3 ja A2 pannes väite ümber(ette) eksistentsiaalkvantorid \exists_{A3} ja \exists_{A4} .

Näiteks elementaarsuhte “a+b=c” konverteerimiseks elementaarseopõhisest notatsioonist elemendipõhiseks tuleb elementaarfunktsiooni “ $x_1+x_2=x_3$ ”(x_1,x_2,x_3) võtta algasi + nii, et konstantne elementaarfunktsioon $g(+,x_1,x_2,x_3)$ on kindla väärtusega siis kui $x_1+x_2=x_3$.

Tähistame elemente, mille vahelistes suhetes on a elementi ja on b erinevat võimalikku suhet järgnevalt: element(a,b). Selles tähistuses oleks hulgad tähistatud kui element(2,4). Näiteks erinevus hulkade ja elementide(2,2) vahel on, et kui kaks hulka võivad omavahel olla 4 erinevas suhtes, siis 2 elementi võivad omavahel olla 2' es suhtes. Hulkadeks võivad olla ka kvanteeritavad. Selle notatsiooni halvaks küljeks on, et see on vähem tuntud. Kui a ja b pole naturaalarvud, siis on elementide vahelised suhted üksteisest sõltuvad. Suhe ei tähenda siin jagatist.

Elemendi-põhises-notatsioonis võib ka muude asjade vahel olla kvantoritega väiteid, kuid muuks asjaks loetakse ikkagi K'd kvanteeritavat, mille kohta käib, nende omavahelisi suhteid ja nende suhteid algelementidega iseloomustav, kvantoreid sisaldav booleanfunktsioon. Kuna osadest asjadest(K'st kvanteeritavast, mille vahel on kvantoritega seosed) järeldub väiteid teiste muude asjade kohta, siis ei pruugi F_Q olla naturaalarv võimalikke väärtusi.

Sellises notatsioonis kirjutatud väidete iseloomustamiseks saab kasutada arv K ja N. K ja N on üheselt seotud asjade arvuga, mille eksisteerimise vahelist boolean funktsiooni väide kirjeldab.

- N on algelementide arv.
- K näitab, et kui mitme kvantori sees üks predikaatväide maksimaalselt on, kui väide on pandud kirja sellisel kujul, et predikaatväide, mis kõige rohkemate kvantorite sees olev väide on, oleks võimalikult väheste kvantorite sees.

Osade väidete konverteerimisel elemendipõhiseks ei pruugi K ja N olla naturaalarvud.

6.1 element(2,4)'e-põhine-notatsioon ehk hulga-põhine-notatsioon (Pole kindel kas toimib)

Hulga-põhises-notatsioonis on väite argument on noteeritud argument-hulkade(alghulkade) abil ja muid asju hulkade abil, mida iseloomustab nende omavahelised kuulumis-sisaldumis suhted ning kuulumis-sisaldumissuhted nende ja alghulkade vahel. Et hulk A kuulub hulka B tähitatakse $A \in B$. Selle hulga-põhise notatsiooni saab teisenda funktsioonipõhiseks teades, et ühe hulga kuulumisusuhteks n hulgaga on 2^{n*4} võimalust. Selleks tuleb viia iga hulka vahelise kuulumise booleankombinatsioon vastavusse ühe funktsiooni f_Q väärtusega saades funktsiooni, millel on 2^{n*4} erinevat väärtust.

Selle notatsiooni eeliseks hulga-põhise-notatsiooni ees on, et väidete kirjeldamisel DNF tabelina tekib vähem mõtetuid ridu.

Näiteks hulga-põhises-notatsioonis kirjeldatud väite $\forall_x(\forall_y(\neg\forall_z(z \in x \iff z \in y) \wedge x \in y \rightarrow y \in A))$ (A'sse kuuluvad kõik hulgad, mis sisaldavad mingit hulka peale iseenda) K=3 ja N=1.

6.1.1 konstantväide

Hulgad, mille elemendid on samad on ise samade asjade elemendid. ehk sama sisaldavusega hulgad on ka sama sisaldamisega ja, et hulgad, mis sisaldavad samu elemente on võrdsed.

6.2 element(2,2)'e-põhine-notatsioon (Pole kindel kas toimib)

Elementidest(2,2) või mõelda kui “asjadest”, mis ei saa midagi sisaldada, kuid saavad omavahel “ühendatud” olla.

Üks võimalus konvertida hulkade-notatsiooni olev väide nende element(2,2)ede vahelises notatsiooni olevak väiteks, on asendada iga hulk kahe element(2,2)' ga. Nimetame neid kahte element(2,2)'esid kuulumis ja sisaldumis elementideks. Kui hulga-põhises-notatsioonis sisaldab mingi hulk mingit teist hulka, siis elemendi-põhises-notatsiooni on

esimese hulga sisaldamiselement ühenduses teise hulga kuulumiselementiga. Kui hulgapõhises-notatsioonis sisaldab mingi hulk iseenneast, siis elemendipõhises notatsioonis on sellele hulgale vastav kuulumiselement ja sisaldumiselement omavahel ühenduses. Veel tuleb elemendi-põhisesse-notatsiooni lidada väide, et ükski kuulumus-element ei ole ühegi kuulumus-elementiga ühenduses ja ükski sisalduvus-element ei ole ühegi sisalduvus-elementiga ühenduses. Selline konvertimine ei ole kõige efektiivsem, sest saaks konvertida ka nii, et tekiks vähem elemente.

6.3 element(e,e)-põhine-notatsioon(Pole kindel kas toimib)

6.4 algmõisteks predikaadi

kasutaja saab algmõistetena ainult predikaate lisada. Selle eeliseks on, et ei pea ühtegi väidet kirjeldama erinevalt sõltuvalt väites olevate predikaatide valikust.

konstantseteks elementaarfunktsioonideks on:

- rakenda(predikaat,järjend) tagastab kas predikaat on True/False kui selle argumendiks on järjendis olevad asjad.
- järjekord(järjend,x1,x2) tagastab, et kas x1 on x2'est järjendis eespool või tagapool.

Eriti stiili lisab, et kvantor oleks justkui 1 argumendiline elementaarseos ja kui rakenda argumendid on väited, siis võib neid tõlgendada NAND'ina, mis annavad kvantorifunktsionaalselt täieliku hulga.

Kvanteeritavad peavad saama olla predikaadid, sest muidu sõltuks ikkagi osade väidete kuju predikaatide valikust. $p(x)$ asemele tuleb $p([x])$.

Lõppridade üheselt valesuse kontroll peaks samasugune tulema, sest kõigi platode predikaatide osa saab lihtsalt asendada rakenda(P,x)'iga.

Tuleb lisada konstantne väide.

6.4.1 konstantne väide

järjekord on transitiivne. kui x1 või x2 ei kuulu järjendisse on tagastab elementaarseos järjekord False. Ükski järjen ei kuulu ühegi järjendisse. iga järjendi plato eelaste järjekord ja kuulumine sellesse järjendisse on määratud selles platos.

kui rakena esimene argument ei ole predikaat, siis on rakenda alati False.

6.5 eeldused seosefunktsioonidele

2 võimalikku väärtust (kas True või False). Algmõisted pannakse ainult esimeseks argumendiks (See ei ole sama, mis eeldus, et kui algmõiste ei ole ei ole esimene argument, siis on elementaarseos üheselt vale, sest kuna kvantorid peavad kõigi asjade kohta kehtima peavad need ka algmõistete kohta kehtima. Ka kvanteeritavad võivad algmõistetega võrdsed olla.)

7 näiteid notatsiooni konverteerimisest.

7.1 elemendi põhiseest elementaarseose põhiseks

Niipidi saab alati konverteerida.

algasi A asemel võta elementaarseos "x_on_A" ja lisa väitele väide, et ainult 1 asi rahuldab seda (kui kaks kvanteeritavat seda rahuldavad, siis on nad võrdsed.).

Luuu uuued predikaadid, mis tähistavad predikaadi väärtust, kui nende argumentideks on algasjad. Näiteks: $\forall_{x_1}(\forall_{x_2}(F(a,x_1,x_2) = F_1(x_1,x_2) \wedge F(x_1,a,x_2) = F_2(x_1,x_2) \wedge F(x_1,x_2,a) = F_3(x_1,x_2)))$.

7.2 elementaarseose põhiseest elemendi põhiseks

Iga elementaarseoses põhises notatsioonis kirjeldatud väidet ei saa igattüüpi (tüüp näitab millised on konstantsed funktsioonid) elemendipõhiseks konverteerida. Näiteks ei saa 3 argumendilise predikaadiga väiteid konverteerida element(2,2) põhiseks konverteerida. Uurida, et millise elemendipõhiseks saab iga elementaarseosepõhist väidet konverteerida.

n argumendise elementaarseose f_1 asemele pane algasi, millega mingis elementaarseoses olevad asjad on ainult n 'i teise asjaga elementaarseoses (eeldab, et $\text{elemendi}(a,b) \text{ a} > n$).

Vist ei saa alati, sest konstantse elementaarsest argumentide arvust suurema argumentide arvuga predikaate konverteerida hulgapõhiseks.

Vist ikka saab: pannes argumendidjärjendisse ja kirjeldada järjendi elementide arv võrreldes elementide järjekorda järjendis elementide järjekorraga paarides. Selleks on vaja kahte 2 argumendilist seosefunktsiooni f_1 ja f_2 . näiteks:

$$F(a,b,c,d) \iff \exists_{x_1}(f_1(a,x_1) \wedge f_1(b,x_1) \wedge f_1(c,x_1) \wedge f_1(d,x_2) \wedge \exists_{x_2}(f_1(a,x_2) \wedge f_2(b,x_2) \wedge \text{sama jar jekord}(x_1,x_2)) \exists_{x_2}(f_1(b,x_2) \wedge f_2(c,x_2) \wedge \text{sama jar jekord}(x_1,x_2)) \exists_{x_2}(f_1(c,x_2) \wedge f_2(d,x_2) \wedge \text{sama jar jekord}(x_1,x_2)))$$

Part V

Ideid funktsiooni F defineerimiseks

8 Funktsioonide F tüüp a

definiitsioon: $F(S) = P(S, K(S)) \wedge \forall_s (\forall_k (P(s, k) = \sum_{i=0} (2^i * (False \neq (S \wedge P(i))))))$ pole valmis

pöördfunktsioon: $\forall_{n_S} (f(n_S) \leftrightarrow \exists_i (\text{plato}(i, 0, []) \wedge \text{lopprea_mark}(n_S, i))) \wedge \forall_n (\forall_k (\forall_{kv} (\text{plato}(n, k, kv) \leftrightarrow (h(n, kv) \wedge \forall_i ((i \in N \wedge (i < \text{veerge}(k))) \rightarrow (\text{saa_haru_mrk}(n, i) \leftrightarrow \exists_x (\text{plato}(i, k+1, kv+x)))))))$

Lühikirjeldus: Saa_h kirjeldab anud taseme kvanteeritava elementaarseoseid madalamate tasemete kvanteeritavatega.

Seda tüüpi funktsioonid on kõik funktsioonid, mis eeltoodud kujul on. Ükskõik millised saa_h ja saa_haru_märk täpselt on, on sellises vormis funktsioon a tüüpi funktsioon. (kas kõik võimalikud F funktsioonid, mille abil saab kõiki väiteid töödelda mitte a-tüüpi ei ole? vt section struktuur. Vist ei ole, kuid iga funktsioon, mis seab lõppreale arvu vastavusse on a- tüüpi funktsioon)

See, et kas läheb vaja tingimst $(i \in N \wedge (i < \text{veerge}(k)))$ sõltub funktsioonide h ja saahrumärk valikust.

8.1 K ja predikaadide nimetuste n_S 'i panemine

idee1 n_S binaarvormi alguses on nii palju 1esid järjest kui on K väärtus. peale neid ühtesid tuleb 0 ja siis ülejäänud osa n_S 'ist. See poleks enam min vabadusastme(F pole bijektiivne), sest täheduseta on need n_S id kus 0ile järnev arv on suurem kui antud K ja prdikaatie korral ülejäänud n_S osa olla saab. predikaadide kirjeldused peavad olema väitega koos mingi muus vorms(tõenäoliselt stringina).

Panna alati 1 n_S ette. Siis on järelolevate bittide arvu järgi K teada kui predikaatide argumentide arv on teada. Binaarsalvetuses piisab ees olevate nullide arvu teadmisest.

8.2 Lihtne DNF

definiitsioon: $F(S) =$ pole valmis

pöördfunktsioon: $\forall_{n_S} (f(n_S) \leftrightarrow \exists_i (\text{plato}(i, 0, []) \wedge (\lfloor n_S * 2^{-i} \rfloor \% 2 = 1))) \wedge \forall_n (\forall_k (\forall_{kv} (\text{plato}(n, k, kv) \leftrightarrow (h(n, kv) \wedge \forall_i ((i \in N \wedge (i < \text{veerge}(k))) \rightarrow (\lfloor n * 2^{-i} \rfloor \% 2 = 1) \leftrightarrow \exists_x (\text{plato}(i, k+1, kv+x)))))) \wedge \forall_k (\text{veerge}(k) = H(k+1) * 2^{\text{veerge}(k+1)}) \wedge \text{veerge}(K) = 0 \wedge \text{veerge}(k) = H(k+1) * 2^{\text{veerge}(k+1)})$ pole valmis

omadused:

- osadele väärtustele n_s vastavad samad väited.
- Raske loetavasse vormi panna, sest üheselt samaväärsete lõppridade tuvastamine on raske.

Lühikirjeldus: a tüüpi, funktsioon, kus $harummrk(x, k)$ ja $saa_h(x, k)$ on sellised, et:

$$\begin{cases} loppreamark(n, i) = (\lfloor n * 2^{-i} \rfloor \% 2 = 1) \\ harummark(n, i) = (\lfloor n * 2^{-i} \rfloor \% 2 = 1) \\ saah(k) = \lfloor n_s * 2^{-veerge(k)} \rfloor \\ veerge(k) = H(k+1) * 2^{veerge(k+1)} \\ veerge(K) = 0 \end{cases}$$

$H(k)$ näitab, et kui mitmes erinevas seoses saab k 'nda taseme kvanteeritav x_k enda ja endast madalamate kvanteeritavatega olla. $H(0) = 0$.

$veerge(k)$ näitab, et mitu haru on k 'ndal tasemel.

funktsiooni $harummark$ saab kirja panna ka järgnevatel kujudel $harummark(n, i) = ((n - n \% 2^i) * 2^{-i} \% 2 = 1)$; $harummark(n, i) = ((n - n \% 2^i) \% 2^{i+1} \neq 0)$; $harummark(n, i) = \exists_{x_1} (x_1 \in N \wedge \exists_{x_2} (x_2 \in N \wedge x_2 < 2^i \wedge n = x_1 * 2^{i+1} + 2^i + x_2))$

Taseme K sees olevad harud võib tähelepanuta jätta, sest need (koos märgiga) on üheselt tõesed $\neg \exists_x (False)$, sest $harummark(x, K) = 0$, kui $x > 0$ ja $saaH(K+1) = lõpmatus$.

Idee põhineb tähelepanekul, et kui

1. kõik universaalsuskvantorid asendada eksistentsaalsuskvantoritega vastavalt reeglile $\forall_x (f(x)) = \neg \exists_x (\neg f(x))$,
2. viia kõik seosefunktsioonide väited kõige välimiste võimalike kvantorite sisse (Ehk võimalikest madalaimale tasemele) (ehk selle kõrgeima taseme argumenti tasemele) (sest muidu tekiks mitu samaväärt rida. nt.)
3. viia kõik kvantorite sisud ja kvantorite välised väited DNF-vormi (kus on alati jaatud kõikide erinevate argumentidega seosefunktsioonid (näiteks kui $K=2$ ja seosefunktsiooni argumentide järjekord pole oluline, kõigil elementaarseostel on 2 argumenti ja 3 võimalikku väärtust tuleb $f(x_1, x_1) = 0 \wedge f(x_1, x_2) = 0$ asendada $f(x_1, x_1) = 0 \wedge f(x_1, x_2) = 0 \wedge f(x_2, x_2) = 0 \vee f(x_1, x_1) = 0 \wedge f(x_1, x_2) = 0 \wedge f(x_2, x_2) = 1$) ja ... (täpsemat seletus vaja)
4. kõik võitused vastavalt reeglile $\exists (f_1(x) \vee f_2(x)) = \exists (f_1(x)) \vee \exists (f_2(x))$ kvantorite seest välja tuua.

saab mistahes väite vormi viia, mis sisaldab mingi sisuga omavahel võitunud üksteist välistavaid väiteid (lõppridu) (mille sees on ainult seosefunktsioonid, jaatud ja eitused), millele saab vastvusse panna naturaalarvu. Nimetame neid väiteid edaspidi lõppridadeks. Osad lõppread on üheselt valed ehk väite argumentist sõltumatult valed. Kõik lõppread, mis pole väite argumentist sõltumatult valed on omavahel erinevad.

Võib jääda ekslik arvamus, et kui defineerida kõik elementaarseosed olema K argumentiga ja jaatatada väitega väide, et osade elementaarseoste väärtus ei sõltu osadel positsioonidel olevatest argumentidest, siis vastaks ka järjestuses igale naturaalarvule vahemikus 0 kuni N_{max} erinev lõpprida, kuid see ei ole võimalik, sest väide et osade elementaarseoste väärtus ei sõltu osadel positsioonidel olevatest argumentidest sisaldab rohkem kui K 'd kvanteeritavat üksteise sees.

Millisele lõppreale milline naturaalarv vastab sõltub ainult K 'st, elementaarseoste argumentide arvudest ja seosefunktsiooni võimalike väärtuste arvust. Seega saab väite üheselt kirjeldada kirjeldades, et millistele naturaalarvudele vastavad lõppread on kirjeldatava väitega kooskõlas. Kirjeldamiseks, et millised lõppread on kirjeldatava väitega kooskõlas moodustan jada (ehk lõppveeru), mille liikmeteks on (boolean), et kas liikme indeksile (naturaalarvule) vastav lõpprida on väitega kooskõlas. Lõppveerg tõlgendatult kahendsüsteemis arvuna ongi arv n_s .

K ja elementaarseoste argumentide ja võimalike väärtuste arv ning muud omadused on eraldi kirjeldatud või bi-jektiivse funktsiooni abil lisatud arvule n_s .

Seletan siin mitmes lõigus sama ideed. Kõikides lõikudes on sama ideed erinevalt kirjeldatud.

8.2.1 Seletus 1

Iga taseme jaoks teha eraldi DNF-tabel, mille veergudeks on selle taseme veerge moodustava tabeli read. Taseme-veerge moodustava tabeli veergudeks on selle taseme kvanteeritava ja madalamata kvanteeritavate ning selle taseme kvanteeritava ja algasjade vahelisel seosefunktsiooni väärtused. Taseme-veerge moodustavat tabelit loetakse jaatades väiteid, et veeru seosefunktsioonil on vastav väärtus.

Iga taseme tabeli igas reas on iga veeru jaoks lahter arvuga, mis kirjeldab, et milliseid järgmise taseme ridu sisaldavad selle taseme kvantorid on omavahelises jaatuses eitatud ja millised mitte (so. DNF vorm). Iga erinevate arvudega täidetud ridu on täpselt üks. Kõrgeima taseme tabeli (mille number = K) ridades on iga veeru kohta ainult üks arv (boolean), mis kirjeldab ainult, et kas veerg on eitatud või mitte, sest pole järgmist taset, ega järgmise taseme ridu. Madalaima taseme tabeli (mille number = 0) veergudeks on algasjade vaheliste seosefunktsioonide väärtused. Madalaima taseme tabel on lõppDNF-tabel, mille read on lõppread, tabeli lõppveerg on töödeldud DNF vormis väite lõppveerg.

8.2.2 Seletus hargnemisega

Iga taseme jaoks teha eraldi DNF-tabel, mille veerutüvedeks on selle taseme-veerge moodustava tabeli read. Iga taseme tabeli iga veerutüvi hargneb veergudeks, mis erinevad selle poolest, et millist järgmise (ühe võrra suurema) taseme tabeli rida selle taseme kvantor sisaldab. Iga veeru (haru) ümber on eraldi selle taseme eksistentsiaalkvantor.

Taseme-veerge moodustava tabeli veergudeks on selle taseme kvanteeritava ja madalamata kvanteeritavate ning selle taseme kvanteeritava ja algasjade vahelise seosefunktsiooni väärtus. Kui palju erinevaid väärtuseid ühel seosefunktsioonil võib olla sõltub kirjeldatavast väitest. Seosefunktsiooni argumentideks võivad olla ainult kvanteeritavad ja algasjad. Seosefunktsioonil võib olla ükskõik kui palju argumente. Taseme-veerge moodustavat tabeli ridu loetakse jaatades väiteid, et veergude seosefunktsioonidel on vastavad väärtused. Kui seosefunktsiooni argumentide järjekord on oluline tuleb igas tasemes panna argumendid kõigis järjekordades.

Kõik veergude vahelised seosed võib arvesse võtta arvestades, et osade tasemete tabelite osad read on väite argumentide sõltumatult valed. DNF tabeli olemuse tõttu lõppridade vahelisi seoseid ei ole. Kas rida on väite argumentide sõltumatult vale kontrollimiseks tuleb kõrgema taseme kvanteeritava asemele asendada madalama taseme kvanteeritavad.

Kõrgeima taseme tabeli (mille number = K) veerud ei hargne, sest pole järgmist taset, ega järgmise taseme ridu. Madalaima taseme tabeli (mille number = 0) veergudeks on algasjade vaheliste seosefunktsioonide väärtused. Madalaima taseme tabel on lõppDNF-tabel, mille read on lõppread, tabeli lõppveerg on töödeldud DNF vormis väite lõppveerg.

Kuna kõik tabelid on DNF tabelid, siis: (VALE)

- Iga taseme tabeli igas reas on iga veeru kohta üks bitt, mis kirjeldab, et kas veerg (mille sees on selle taseme kvantor) on rea siseses jaatuses eitatud ja või mitte (so. DNF vorm).
- Iga erinevate bittidega täidetud ridu on täpselt üks.
- read on üksteist välistavad ehk omavahel vastuolus.

8.2.3 Seletus Ühe tabelina.

Tabelis on kõik esimese taseme veerud. Iga taseme kõik read lähevad iga ühe võrra madalama taseme iga veeru juurde alaveergudeks.

8.2.4 Seletus ühe tabelina (ei toimi)

Veergudeks on eksistentsiaalkvantorid. Iga veeru, mille kvanteeritava tase ei ole K , sees on ühe võrra kõrgema taseme eksistentsiaalkvantorid. Iga ühe võrra madalama taseme kvantori sees on $a(k)$ k 'nda taseme kvantorit. Esimese taseme kvantorid ei ole ühegi kvantori sees neid on $a(k)$ 'st tükki.

8.2.5 Seletus ühe tabelina pöördveergudega alustades seletamist 0-tasemest.

0 taseme seosefunktsioon (seosefunktsioonid, mille argumentideks on ainult algasjad) on tabeli veerg. Iga taseme, mille number pole K , veergudeks on seosefunktsioonid, mille argumentideks on selle ja madalama taseme kvanteeritavad ja

algasjad, ja kõrgema taseme read. Knda taseme veergudeks on ainult taseme K seosefunktsioon. Kuna iga taseme read ristuvad kõrgema taseme ridadega (on 90 kraadi pöördes) ja iga taseme veerud ristuvad kõrgema taseme veergudega (on 90 kraadi pöördes) nimetan seda seletust pöördveergudega seletuseks.

8.2.6 Seletus ühe tabelina pöördveergudega alustades seletamist K-tasemest.

K taseme seosefunktsioon(seosefunktsioonid, mille argumentideks on ainult algasjad) on Knda taseme tabeli veerg. Iga taseme, mille number pole 0, read on, koos ühe võrra madalama taseme seosefunktsiooniga, veergudeks ühe võrra madalamale tasemele. Otaseme seosefunktsioon on (lõpp)tabeli veerg. Kuna iga taseme read ristuvad kõrgema taseme ridadega (on 90 kraadi pöördes) ja iga taseme veerud ristuvad kõrgema taseme veergudega (on 90 kraadi pöördes) nimetan seda seletust pöördveergudega seletuseks.

8.2.7 seletus

iga plato iga bitt näitab, et et kas selle biti indeksiga haru on eitatud või jaatatud.

8.3 DNF, mille osad read on eemaldatud

definiitsioon: $F(S) = P(S, K(S)) \wedge \forall_s (\forall_k (P(s, k) = \sum_{i=0}^{2^i - \sum_{j=0}^i (neg \exists_s (False \neq (S \wedge P(j))))} * (False \neq (S \wedge P(i))))$ pole valmis

pöördfunktsioon: $f(n_S) =$ pole valmis

omadused:

- Raske käsitsi loetavast vormist arvutitoetavusse panna, sest üheselt valede lõppridade tuvastamine on raske.

Lühikirjeldus: Muidu sama nagu lihtne DNF tabel ainult, et üheselt valed lõppridadele vastavad arvud on vahele jäätud. (Kas jätab iga plato üheselt valed harud vahele või ainult lõpprea üheselt valed harud?) Ja väited, mida saab väiksema K'ga on ka vahele jäätud numeratsioonist. väidete, mida saab ka väiksema K'ga kirjeldada kõikide lubatud lõppridade taseme K-1 platode kõik harud on sama märgiga (ei ole kindel, et kõik väited, mida saab ka väiksema K'ga kirjeldada seda tingimust täidavad).

Numeratsioon jätab vahele need naturaalarvud, millelele vastav väide on sõltumatult kirjeldatavast väitest valed, kuid ei muuda seda, et millisele lõppreale vastab suurem arv ja millisele väiksem arv.

Lõpprida on üheselt vale kui see on kõigi väite argumentide puhul vale ehk $\neg \exists_Q (LR(Q))$.

8.3.1 väite argumentst sõltumatult valedele väidetele vastavate arvude leidmine ehk veergude vahelise seose arvestamine

1. eitatud harud asendatavad universaalsuskvantoritega. võttes harumärgi funktsiooni kujul $\forall_n (\forall_k (\forall_{kv} (plato(n, k, kv) \leftrightarrow$

$$(h(n, kv) \wedge \forall_i ((i \in N \wedge (i < veerge(k))) \rightarrow (|n * 2^{-i}| \% 2 = 1) \rightarrow \exists_x (plato(i, k + 1, kv + x)))) \wedge$$

$\forall_x (plato(i, k + 1, kx + x) \rightarrow \exists_i (i \in N \wedge (i < veerge(k)) \wedge |n * 2^{-i}| \% 2 = 1))))$. Sellel kujul on ilmselege, et tagumisest osast ei järeldu midagi, peal selle, et ei eksisteeri asju, mis ei rahuldaksid tingimusi, mida rahuldavad asjad mis esimese osa kohaselt peavad eksisteerima. Kui eeldada, et kvantorist saab välja tuua ainult selle info, mis on ülejäänuga jaatatud, siis saab seda eeldust põhjendada sellega, et see mida saab jaatuse ette tuua peab kõigi harude platofunktsioonidel ühine olema, aga see info on ka väite esimeses eksistentsiaalkvantoriosas olemas. Arusaamist lihtsustab kitsam näide seletusest kujult $\exists_{x_1} (p_1(x_1)) \wedge \exists_{x_1} (p_2(x_1)) \wedge \forall_{x_1} (p_1(x_1) \vee p_2(x_1))$.

2. piisab kui kontrollida, et kõik asjad, mis peavad eksisteerima täidaksid kõiki tingimusi, mida kõik asjad peavad täitma. Kõik asjad, mis eksisteerima peavad on eksistentsiaalsuskvantorites. Kõik tingimused, mida asjad täitma peavad on universaalsuskvantorites.

3. Iga plato universaalsus kvantori sees on võitatud need tingimused, mis on selle haru eksistentsaalsuskvantorite sees.

Plato kvanteeritava kirjeldamiseks tehtud väited kehtivad kõikjal, selle kvantori plato sees (sh selle kvantori sees olevate eitamata kvantorite sees). Seega kui mingi plato p haru teeb väite, mis on kvantorite, mille sees plato p on kvanteeritavatele seatud tingimusega vatuolus on see plato p väite argumendist sõltumatult vale. Kuna väide on sellises vormis, et kõikide võimalike tingimuste kohta on öeldud, et nad on kas tõesed või valed peab nende kohta olema jaatud haru.

4. piisab kui kontrollida, et iga plato kvanteeritav täidab samu tingimusi kui iga teise plato vähemalt 1 haru. VIGA PARANDADA KOPEERIDA JUPPE 27. JUUNI JA 30. JUULI VERSIOONIST.

5. **Lõpprida ei ole üheselt vale parajasti siis kui mistahes(kõigi) plato(mille ükski haru ei ole juba võrdsustatud ja), kuhu pääseb ilma eitusi läbimata, (vähemalt) ühe haru(p_2) (universaalsuskvantori kvanteeritava) saab võrdsustada, mistahes teise plato(p_1), kuhu pääseb ilma eitusi läbimata, kvanteeritavaga nii, et eelnevad võrdsustused jäävad kehtima.**

Efektivselt säilivad p_1 'ede vahelised ja p_2 'ede vahelised elementaarseosed ja elementaarseosed px 'i ja madalama eelastega. Tegelikult säilivad kõik elementaarseosed, aga kui universaalsuskvantor, mille kvanteeritavat (p_2) võrdsustatakse ise on teise universaalsuskvantori(px) sees, siis ei saa olla kirjeldatud elementaarseosed tema kvanteeritavate, mille tase on kõrgem kui kõrgeima tasemga eksistentsaalsuskvantor, mille sees ta on (ehk madalama tasemega universaalsuskvantor(px), mille sees ta on) vahel. Kahe argumendiliste predikaatide korral: $\forall i_1 (\forall i_2 (g(P1(i_1), P1(i_2)) = g(P2(i_1), P1(i_2))))$. sama indeksiga p_1 ja p_2 'e vahelised elementaarseosed ei pruugi olla samad, mis mitme selle indeksiga p_1 'e vaheline elementaarseos, sest isegi kui mõnel p_1 'el ja p_2 'el on tavalises puugraafi visualiseeringus ühiseid eelasi või järglasi, on p_2 'ed universaalsuskvantorite kvanteeritavad ja p_1 'ed eksistentsaalsuskvantorite kvanteeritavad ehk tegu ei ole siiski samade kvanteeritavatega. Kuna p_1 'esid valitakse ainult platodest kuhu pääseb ilma eitusi läbimata, siis ei saa universaalsuskvantorite seest p_1 'esid valida (see on selgem r 'i vormis kus universaalsuskvantori asemel on eitatud eksistentsaalsuskvantorid). Sama p_1 'e (eksistentsiaalkvantori kvanteeritavat) peab saama võrdsustada ka mitme plato vähemalt ühe haruga. Võrdsustada mingit plato p_2 kvanteeritavat haruga ei saa kui plato p_2 mingi haru kvanteeritava elementaarfunktsioonid on erinevad selle haru kvanteeritava elementaarfunktsioonidest (elementaarfunktsioonide argumentideks võivad olla ka madalama taseme kvanteeritavad, mis on mingi muu plato kvanteeritavaga võrdsustatud. Kas ka kõrgema taseme kvanteeritavad?).

NB: kui mingid P_1 'ed ja neile vastavad P_2 'ed o juba valitud ei tohi P_1 'ede lisamine P_2 'esid muuta, sest muidu ei saaks kontrollid, et on olemas haru, millel on mitu nõutud haru.

Teisisõnu lõpprida on üheselt vale parajasti siis kui mingite platode (korraga), kuhu pääseb ilma eitusi läbimata, ühtegi haru ei saa võrdsustada, mingite teiste platode, kuhu pääseb ilma eitusi läbimata, kvanteeritavatega.

Selles meetodis võrdsustakse universaalsuskvantorite kvanteeritad teiste platode eksistentsiaalkvantorite kvanteeritavatega.

6. Platode, millest kumbgi pole teisele eelaseks ega järglaseks kvanteeritavate vahelisi elementaarseoseid ei ole kirjeldatud, seega saab need korraga võrdsustada mingite teiste kvanteeritavatega parajasti siis kui neid saab eraldi võrdsustada nende teiste kvanteeritavatega. Et kontrollida kas mingi plato tasemega k vähemalt ühe haru kvanteeritav võib mitmete asjadega mitmes erinevas elementaarseoses olla (ehk, et kas sellel on mitu haru olemas) nii, et ka nende asjade vahelised elementaarseosed on määratud, saab kõigepealt $P_1[k+1:K]$ 'ks valida need asjad ja siis võtta eelmise sammu $P_2(k+1)$ selle sammu $P_1(k+1)$ 'ks (vastavalt postulaadile sobib selle sammu $P_2(k+1)$ 'ks siis ainult $P_1(k+1)$) $P_1[k+1:K]$ 'ks samuti need asjad, ga teistes järjekordades. See toimib, sest kõik esimese sammu P_2 'ede vahelised elementaarseosed peavad olema samad kui nende asjade vahelised. Seega võib ÜV'd kontrollida tehes asendused(p_2 'ed) ühte suutatud lihtahelasse korraga. Ühe suunatud lihtahela maksimaalne pikkus on K .

Seega lõpprida ei ole üheselt vale parajasti siis kui kõikide hulkade, kuhu kuuluvad platod millest igaüks on teisele kas eelaseks või järglaseks ja kuhu pääseb ilma eitusi läbimata, iga elemendi (vähemalt) ühe haru kvanteeritava saab võrdsustada, mistahes teise plato p_1 , kuhu pääseb ilma eitusi läbimata, kvanteeritavaga (nii, et platode, mille kvanteeritavatega need võrdsustati, kvanteeritavate vahelised elementaarseosed jäävad samaks.).

7. Kui mingi vormis on vähem kui K kvanteeritavate paari, mille elemendid on omavahel võrdsustatud, ja see on ÜV, siis on ÜV ka vorm kus samade kvanteeritavate paarid omavahel võrdsustatud, kuid kokku on K paari.

Seega piisab K asja vaja korraga “asendada”. Seega kontrollida saab järjendite kaupa, kus on K p1’e ja K p2’e. või kahe järjendite J_e ja J_u , millest mõlemas on K elementi järjetatud paaride kaupa.

Seega lõpprida ei ole üheselt vale parajasti siis kui kõikide hulkade, kuhu kuuluvad platod millest igaüks on teisele kas eelaseks või järglaseks, mis sisaldavad platot tasemega 0(lõpprida ise), ja kuhu pääseb ilma eitusi läbimata, iga elemendi (vähemalt) ühe haru kvanteeritava saab võrdsustada, mistahes teise plato p1, kuhu pääseb ilma eitusi läbimata, kvanteeritavaga(nii, et platode, mille kvanteeritavatega need võrdsustati, kvanteeritavate vahelised elementaarseosed jäävad samaks.).

Seega vaja kontrollida K kaupa kõigis järjekordades. VIGANE

8. __, sest nii saab mistahes suunatud ahelaid(P2’esid) kontrollida ja kui mingi plato(P2(i)), mille puhul ma kontrollin, et kas selle mingi haru kvanteeritava saab mingi muu kvanteeritavaga võrdsustada, eelase teine haru on millegagi võrdsustatud saab seda eraldi kontrollida(teise P1’ega), sest see pole samas suunatud ahelas. Seega lõpprida ei ole üheselt vale parajasti siis kui mistahes K plato kvanteeritavad saab mistahes järjekorras võrdsustada mingite teiste platodega, millest igaüks on teisele kas eelaseks või järglaseks kvanteeritavatega nii, et esimene neist on mistahes plato haru.

ehk

lõpprida ei ole üheselt vale parajasti siis kui mistahes platode vahelised elementaarseosed ja predikaatväited ühiste eelastega on samad kui mistahes plato (px) mingite järglaste, millest üks on selle plato otsene järglane, ühe tase on K ja millest igaüks on igale teisele ka eelaseks või järglaseks, vahelised elementaarseosed.

ehk $UV(LR) = \forall_{px} (\forall_{P1} (\exists_{P2} (\exists_i (k(P2(i)) = K) \wedge \text{”vastavate elementide vahelised seosed on samad või on uhel neist tapsustamata”} \wedge \forall_i (\text{”on haru”}(P2(i), P2(i+1)) \wedge \text{”paaseb ilmaeituslabimata”}(P1(i))))))$

kui px kasutamise asemel nõuda, et kui esimesed P1’ed on alates õppreast taseme järjekorras peavad vastavad P2 nende P1’edega võrduma, siis on tingimus järgnev $UV(LR) = \forall_{P1} (\forall_i (\text{”paaseb ilmaeituslabimata”}(P1(i)) \rightarrow \exists_{P2} ($

$\forall_{p0} (\forall_i (i < p0 \rightarrow k(P1(i)) = i) \rightarrow \forall_i (P1(i) = P2(i))) \wedge$

$len(P1) = K \wedge len(P2) = K \wedge \text{”vastavate elementide vahelised seosed on samad või on uhel neist tapsustamata”}(P1, P2) \wedge \forall_i (\text{”on haru”}(P2(i), P2(i+1))))$ VIGANE

9. Piisab kui valida ainult 3 tüüpi P1’esid:

valik1: mis algavad platost px eelastest, on tasemejärjekorras ja millest igaüks on teistele kas eelaseks või järglasteks

valik2: mis algavad plato px järglasest, on tasemejärjekorras ja millest igaüks on teistele kas eelaseks või järglasteks

valik3: Mis algavad platost px, on suvalises järjekorras ja millest igaüks on teistele kas eelaseks või järglasteks

Kui nende P1 vaikutega ei saa ÜV’d, ei saa ühegi teise valikua ÜV’d, sest:

olulised on ainult K-k(px) esimest P1’e elementi;

kuna valik2 ei anna ÜV saab iga p1’e, mi

sest kui mingi plato pa harusid ei saa taseme järjekorras teise plato pb harudega võrdsustada, siis saab ka nende asendustega üv kui valida platode ,mis jäävad pa ning pb ja pb kõrgeima ühise eelase vahele asendus 1 ja platode, mis jäävad plaode pb ning pb ja pb kõrgeima ühise eelase vahele asendus 2. Kui selline kontroll ei andnud ÜV’d, siis on platos px kõik võimalikud p2’ed tasemejärjekorras olemas. Seega tehes P1 valiku3 saab kontrollida, et kõik P1’ed saab kõigis järjekordades sinna asendada. Pole üöhjendust juhukohta kui P1’ed pole samas lihtahelas, aga kehtib ka siis.

Samuti on ÜV’d need platod, mille üks aatatud harudest on üheselt vale.

Ilma px’ita variandis on eed tingimused järgnevad:

valik1:

valik2: tasemejärjekooras ahel(pikkusega k), mille esimene element on tasemel 0 + taseme tasemejärjekooras ahel(pikkusega K-k), mille esimene element on tasemel eelmise lüli eelviimase elemendi aru.

valik3: p1’ed on suunatud lihtahel P1, aga ei pruugi olla tasemejärjekorras. Sama platod võib mitu korda olla P1’es.

10. Seda kas valik P1 valik1 või valik2 annavad vatuseks ÜV saab kontrollida:

Võitades kokku px järglase need harud, mis erinevad ainult selle poolest, et millised on predikaatide väärtused, kui üheks nende argumendiks on järglase kvanteeritav (x_{k+1}). Ehk eemaldab järglase harudest(rekusiivselt) predikaatide väärtused kui predikaatide argumendiks on ka x_{k+1} harud, mis sisaldavad x_k 'd, saab sama tulemuse kui võitades kokku kõik need x_k harud, mis erinevad ainult K-1 taseme haru märkide poolest(ehk eemaldades x_k 'st kõrgeima taseme harud.)

11. valiku3 kontrollimiseks üpiisab kontrollid, et igad px'i järglased(tasemejärjekorras) saab igas järjekorras px'i tagasi asendada. Sest kui seda kõigepealt ktrollida ja o1'ega kõik vastavad vahesamud kpntrollida saab kontrollitud ka, et vajalikud harud olemas oleks. _..._ POLE KINDEL ET ÕIGE

12. seda kas valik3 annab ÜV saab kontrollida kontrollides et kas kõigile jnumbritel, mis on px'ist alates tasemejärjekorras platodel eelneva harunumbriks, leiduvad ka numbrid , mis on samuti tasemejärjekorras harunumbriks px'ist alates, aga elementaarseoste järjekord on ära vahetatud.

13. mõlemad tingimused koos on: $\neg UV(LR, 0) \wedge \forall_n (\forall_k (\neg UV(n, k) \leftrightarrow (\forall_i (hm(n, i) \rightarrow (r_{meemaldasised}(n) \leftrightarrow r_{eemaldaviimanekvaneritav}(i) \wedge \neg UV(i, k+1)))) \wedge$

$\forall_{j1} (j1[0] = n \wedge \forall_i (harumark(j1[i], j1[i+1], k)) \rightarrow \forall_{w_{jarkord}} (w_{jarkord} < \Gamma(K - k - i) \rightarrow \exists_{j2} (j2[0] = n \wedge \forall_i (harumark(j2[i], j2[i+1], k) \wedge "h - osa nagu j1el aga teies jar jekorras"(j2[i], j1, i, w_{jarkord}))))))$
eeldab punkti 11!

14. Asendan funktsiooni harumärk($harumark(n_1, n_2) = ([n_1 * 2^{-n_2}] \% 2 = 1) = \exists_{x1} (x1 \in N \wedge \exists_{x2} (x2 \in N \wedge x2 < 2^{n_2} \wedge n_1 = x1 * 2^{n_2+1} + 2^{n_2} + x2))$) tingimusse

$\forall_n (\forall_k (\neg UV(n, k) \leftrightarrow (\forall_i (hm(n, i) \rightarrow (r_{eemalda sisemised}(n) \leftrightarrow r_{eemalda viimane kvaneritav}(i) \wedge \neg UV(i, k+1)))) \wedge$

$\forall_{j1} (j1[0] = n \wedge \forall_i (i \in N \wedge i < K - k \rightarrow j1[i] < veerge(k+i-1) \wedge \exists_{x1} (x1 \in N \wedge \exists_{x2} (x2 \in N \wedge x2 < 2^{j1[i+1]} \wedge j1[i] = x1 * 2^{j1[i+1]+1} + 2^{j1[i+1]} + x2))) \rightarrow$

$\forall_{w_{jarkord}} (w_{jarkord} < \Gamma(K - k - i) \rightarrow \exists_{j2} (j2[0] = n \wedge \forall_i (i \in N \wedge i < K - k \rightarrow j1[i] < veerge(k+i-1) \wedge \exists_{x1} (x1 \in N \wedge \exists_{x2} (x2 \in N \wedge x2 < 2^{j2[i+1]} \wedge j2[i] = x1 * 2^{j2[i+1]+1} + 2^{j2[i+1]} + x2))) \wedge "h - osa nagu j1el aga teies jar jekorras"(j2[i], j1, i, w_{jarkord}))))))$
eeldab punkti 11!

15. panen $r_{eemalda sisemised}(n) \leftrightarrow r_{eemaldaviimanekvaneritav}(i)$ asemele tingimuse mis seda kontrollib. Tingimus on siis:

$\forall_n (\forall_k (\neg UV(n, k) \leftrightarrow (\forall_i (hm(n, i) \rightarrow (samavaarseedharud(n, i, k) \wedge \neg UV(i, k+1)) \wedge \forall_{j1} (j1[0] = n \wedge \forall_i (i \in N \wedge i < K - k \rightarrow j1[i] < veerge(k+i-1) \wedge \exists_{x1} (x1 \in N \wedge \exists_{x2} (x2 \in N \wedge x2 < 2^{j1[i+1]} \wedge j1[i] = x1 * 2^{j1[i+1]+1} + 2^{j1[i+1]} + x2))) \rightarrow \forall_{w_{jarkord}} (w_{jarkord} < \Gamma(K - k - i) \rightarrow \exists_{j2} (j2[0] = n \wedge \forall_i (i \in N \wedge i < K - k \rightarrow j1[i] < veerge(k+i-1) \wedge \exists_{x1} (x1 \in N \wedge \exists_{x2} (x2 \in N \wedge x2 < 2^{j2[i+1]} \wedge j2[i] = x1 * 2^{j2[i+1]+1} + 2^{j2[i+1]} + x2))) \wedge "h - osa nagu j1el aga teies jar jekorras"(j2[i], j1, i, w_{jarkord}))))))$

$\wedge \forall_{n1} (\forall_{n2} (\forall_k (samavaarseedharud(n, i, k) \leftrightarrow \forall_{i1} (i1 \in N \wedge i1 < veerge(k) \wedge k < K - 1 \wedge hm(n1, i1) \rightarrow \exists_{i2} (hm(n2, i2) \wedge (f(h(i2)) = h(i1))) \wedge samavaarsed(i1, i2, k+1))) \wedge i1 \in N \wedge i1 < veerge(k+1) \wedge hm(n2, i1) \rightarrow \exists_{i2} (hm(n2, i2) \wedge (f(h(i1)) = h(i2))) \wedge samavaarsed(i2, i1, k+1))))))$
ehk $\neg UV(LR) = \forall_{p1} (\forall_{p1} (p1 \in P1 \rightarrow "paaseb ilma eitusi labimata"(p1)) \rightarrow \exists_{I2} (I2(0) = LR \wedge \forall_i (i \in N \wedge i \leq K - 1 \rightarrow (f(P1, I2(i+1)) \wedge \exists_{x1} (x1 \in N \wedge \exists_{x2} (x2 \in N \wedge (x2 \leq 2^{I2(i+1)} - 1) \wedge I2(i) = x1 * 2^{I2(i+1)+1} + 2^{I2(i+1)} + x2))))))$

VIGANE

16. sama mis järgmise punktis()järgmine kustutada. $\forall_n (\forall_k (\neg UV(n, k) \leftrightarrow \forall_{j1} (j1[0] = n \wedge \forall_i (i \in N \wedge i < K - k \rightarrow j1[i] < veerge(k+i-1) \wedge \exists_{x1} (x1 \in N \wedge \exists_{x2} (x2 \in N \wedge x2 < 2^{j1[i+1]} \wedge j1[i] =$

$x_1 * 2^{j1[i+1]+1} + 2^{j1[i+1]} + x_2))) \rightarrow$
 $\forall_{w_{\text{jarjekord}}} (w_{\text{jarjekord}} < \Gamma(K - k - i) \rightarrow$
 $\exists j_2 (j_2[0] = n \wedge \forall_i (i \in N \wedge i < K - k \rightarrow j_1[i] < \text{veerge}(k + i - 1) \wedge \exists_{x_h} ("h - \text{osa nagu } j_1 \text{ el aga teies jarjekorras"}(j_1, w, i, x_h) \wedge$
 $\exists_{x_1} (x_1 \in N \wedge x_1 < 2^{\text{veerge}(i) - j_2(i+1) - 1} \wedge \exists_{x_2} (x_2 \in N \wedge (x_2 \leq 2^{j_2(i+1)} - 1) \wedge$
 $j_2(i) = x_h * 2^{\text{veerge}(i)} + x_1 * 2^{j_2(i+1)+1} + 2^{j_2(i+1)} + x_2)))))) \wedge$
 $r_{\text{eemalda sisemised}}(n) \leftrightarrow r_{\text{eemaldaviimanekvaneritav}}(j_1[1]) \wedge \neg UV(j_1[1], k + 1)))$
VIGANE

17. Kuna on teada, et igal tasemel (k) on *veerge(k)* veergu ja selle taseme kvanteeritava predikaatväited eelmiste tasemete kvanteeritavatega on kirjeldatud bittidega, mis tulevad peale selle taseme haru märke (more significant), siis saan eelnevat tingimust lihtsustada pannes x_1 'e max väärtuseks $2^{\text{veerge}(i) - I_2(i+1) - 1}$ ja lisades uue kvanteeritava, mis kirjeldab vastava p_2 'e elementaarseoseid madalama taseme p_2 'edega. Seega:

$\neg UV(LR) = \forall_{p_1} (\forall_{p_1} (p_1 \in P_1 \rightarrow "paaseb ilma eitusi labimata"(p_1)) \rightarrow$
 $\exists I_2 (\text{postulaat} \wedge I_2(0) = LR \wedge \forall_i (i \in N \wedge i \leq K - 1 \rightarrow (\exists_{x_3} (f(I(i+1)(\text{voix}_3), P_1, i) \wedge \exists_{x_1} (x_1 \in N \wedge x_1 < 2^{\text{veerge}(i) - I_2(i+1) - 1} \wedge$
 $\exists_{x_2} (x_2 \in N \wedge (x_2 \leq 2^{I_2(i+1)} - 1) \wedge$
 $I_2(i) = x_3 * 2^{\text{veerge}(i)} + x_1 * 2^{I_2(i+1)+1} + 2^{I_2(i+1)} + x_2))))))$
 $x_3 < 2^{H(i)}$ kas f tingimus x_3 'el või $I(i+1)$ 'el?

18. Kasutades tähelepanekut, et kui otsitav plato peab jaatama ühte mitmetest harudest, siis topelt sobivuse vältimiseks saab otsida platosid, mis jaatab neist harudest ühte h_1 ja ei jaata ühtegi nendest harudest, mille indeks on selle indeksist väiksem.

19. Kuna P_1 määrab iga P_2 elemendi vaelised nõutud elementaarseosed selle P_2 elemendi eelastega, siis saab kasutada P_1 , ega bijektiivses seoses olevat järjendit I_1 , mille iga element määrab, et tingimused P_2 , et elementide h osale. Seega võib $f(P_1, i, I_2(i))$ asemel kasutada $f(I_1(i), I_2(i))$. Kui kõik P_1 elemendid on ülejäänudele kas eelaseks või järglaseks, siis on iga P_2 elemendi h osa kindlalt määratud. Selliste P_1 'ede puhul võib $f(P_1, i, I_2(i))$ asemele panna $[I_2(i) * 2^{-\text{veerge}(i)}] = I_1(i)$. (EI TOIMI, sest ka platode, mis on p_2 'e järglased elementaarseosed peavad samad olema kui p_1 'ega). Kuigi VÕIBOLLA saab valida alati kõrvalharu niie et p_1 'e ja p_2 'el pole ühiseid järglasi.

20. Algoritm, mis kontrollib selle tingimuse põhjal üheselt valesust võib alustada P_2 'e platode numbrite (i'de väärtuste) proovimist nii kõrgeima taseme harudest kui madalaima taseme platodest. tundub, et kasulik on alutada kõrgeimast tasemest.

Algoritm, mis kontrollib selle tingimuse põhjal üheselt valesust võib proovida järjest P_2 patode võimalike haru numbreid või salvestadajärjest iga taseme võimalikud harue numbrid.

21. Iga valitud "asendatav" plato määrab ise ühe kindla haru kuhu see sobib ja isegi kui sobib mitmesse, siis piisab ainult selle kontrollimisest, kuhu see sobib. ehk et p_2 'e saab kontrolli käigus kindlalt määrata. Kontrollitakse, et kas p_1 'ed võivad vähemalt ühe p_2 'e haruga võrdsed olla.

Ehk kui mingi on kontrollitud et mingi plato p_2 kvanteeritav mingi teise plato p_1 mingi haru p_h kvanteeritavaga võrdne olla ei saa isegi ilma p_h järglasi võrdustamata (sest p_h kvanteeritaval on enda eelastega nõutud teistsugused seosefunktsioonide väärtused kui p_2 'el nendega on), siis pole mõtet platot kontrollida, et plato p_h ega selle järglaste vähemalt üks haru mingi kvanteeritavaga võrdsustav on, sest $..._.$

ehk kuidas valida nii, et jääks p_1 'ed jääksid samasse harusid on piiratud ja võimaldab optimeerimist. (HOOPIS SIIN VIGA)

22. Osad harud saab välistada teades ainult p_1 'e numbrit, p_1 'e taset, p_2 'e numbrit ja p_2 'e taset, selle põhjal, et iga p_1 'e haru peab saama asendada p_2 'e ja iga p_2 'e haru peab saama asendada p_1 'e ja p_2 'el peavad eelatega samad elementaarseosed olema. Tähistan funktsiooni kas, mis kontrollib, et mingi haru sobib selle tingimusega o'ga. $o(n_1, n_2, k_1, k_2)$ on tõene siis kui plato numbriga n_2 ja tasemega k_2 harud saab korraga asendada taseme järjekorras plato numbriga n_1 tasemega k_1 mistahes harudega.

Ainult platode p_1 ja p_2 taset ja numbrit teades saab kontrollida, et kas p_2 'e saab võrdsustada p_1 'ega ja p_1 'e kõik järglased (samad järjekorras!) p_2 'e järglastega. Kuna võrdsustusi peab saama teha iga P_1 'e korral, siis juhul kui p_2 'e ei saa võrdsustada sellise haruga, kuhu saab ka p_1 'e kõik järglased asendada, siis on see LR ÜV. Seega võib nende harude kontrollimise ära jätta. See on tarvilik tingimus haru numbritele, mitte piisav tingimus.

23. Kui $P2(i-1)$ 'el leidub haru, mille puhul $o(n_2, n_1, k_2, k_1)$ pole rahuldatud, siis on LR ÜV, sest selle haru kvanteeritavat ei saaks võrdsustada $_{_}$. Siis järeldub o'st ka, et kui $P2(i)$ on $P1(i+1)$ eelane, siis $P2(i+1)=P1(i+1)$, mida on 8. punktis vaja.

8.3.2 väidete, mida saab madalama K'ga kirjeldada tuvastamine

Kui kõik võitused sisse viia, siis peab leiduma vähemalt 1 järjend, millesse kuuluvad platod, millest iga üks on eelemisele eelaseks ja milles kvanteeritavate seoste graaf on sidus. kvanteeritavate seoste graaf on graaf, kus kvanteeritavate vahel leidub serv kui vähemalt 1 nende vahelise predikaadi väärtus on teada.

8.3.3 väidete, mis osade predikaatide kohta midagi ei väida tuvastamine

funktsioonist o O funktsiooni on võimalik ka teisisiti seletada nii, et pisab kui ainult p_k otsestest järglastest eemaldatakse predikaatide väärtusi. Loob igast platost ja selle eelasest o-grupid ja kontrollib, et kas need on võrdsed.

platost p_k tasemega k o-grupide loomine Võitab kokku need harud, mis erinevad ainult selle poolest, et milised on predikaatide väärtused, kui üheks nende argumentiks on x_k . Ehk eemaldab harudest (rekursiivselt) predikaatide väärtused kui predikaatide argumentiks on ka x_k harud, mis sisaldavad x_k 'd.

selle plato eelset o-gruppide loomine võitab kokku kõik need harud, mis erinevad ainult K taseme haru märkide poolest. Ehk eemaldab harudest kõrgeima taseme harud.

funktsioonist o2 kõigis tasemtes on seosefunktsioonide väärtused kõrgemad kui madalamates (POLE NII LIHTNE)?

funktsioon o3 pole vaja, $P2$ 'ed on universaalsus kvantorites.

8.4 tüübi a F, mille ridade üheselt valesust on lihtne kontrollida

eemaldab, sosed, mis: saab madalama K'ga kirjeldada, on mitu sama tähendusega lõppveergu (sest osad lõppreadon üheselt valed), või osade predikaatide kohta pole midagi öeldud.

a tüüpi, funktsioon, kus $harumark(x, k) = [n_s * 2^t] \% (2^l - H_{max}(k) - t)$ ja $saah(x) = \lfloor x \rfloor$.

saa_h ja saa_haru_märk on valitud nii, et ridade üheselt valesust oleks kerge kontrollida.

harumärk peaks olema midagi foriee pöörde laadset, kus iga element määrab, et kas lisadaperioodiline järkide vaheldumine. Negatiivsete (või keskmisest väiksemate) indeksitega bitid määravad negatiivsetest märkidest alavad võnkumised. Võnkumiste periood on võrdne biti indeksi absoluutväärtusega. Haru märkide määramiseks lainete väärtused kas liidetakse ja siis määratakse märk või lainete äärtustega tehakse vastavas järjekorras võituseid ja jaatuseid.

Või mingi polünoomi laadne, kus polünoomi nullkohad tähistavad 0-bitti.

vahet pole, et kas i sagedus muudab iga i'indat või i kaupa.

lainevärgi valikud:

- viimane kirjutab oma märgiga üle (sama mis tavaline hm)
- või kõik flipivad bitte kui oma märk on 1
- või kõik kirjutavad üle 1'ega kui oma märk on 1.

ja

- sagedusbiti indeks määrab ka, et kas algab poolperioodist
- või mitte

8.4.1 harumärk nii, et kõigis platodes oleksid alati funktsioon o täidetud

hm näitab iga ja ainult selliste harude hulkade, mis sobivad mingite ühe võrra madalma taseme harude hulkadega kokku (kas neil hulkadel ühiseid elemente ei ole?), et milliste selle taseme kvanteeritava seosefunktsiooniväärtustega (vähemalt ühega) on nende hulkade elemendid (selle taseme harud) jaatatud. Kui on , siis peab see seda küigi tasemejärjekorras korraga olema.

hm ja h on nii valitud, et kõigi platode puhul, mis on jaatatud on täidetud tingimus $\forall_n (\forall_i (hm(n, i) \rightarrow (r_{meemaldasisemised}(n) \leftrightarrow r_{eemaldaviimanekvaneritav}(i)))) \wedge \forall_{j1} (j1[0] = px \wedge \forall_i (harumark(j1[i], j1[i+1])) \rightarrow \exists_{j2} (j2[0] = px \wedge \forall_i (harumark(j2[i], j2[i+1]) \wedge "h - osanagu j1 elagateies jar jekorras"(j2[i], j1, i))))).$ Oluline, et ükski plato ei jaataks harusid, mis seda tingimust ei täida.

8.4.2 harumärk nii, et kõigis platodes oleks funktsioon o2 täidetud

definiitsioon: $F(S) = \text{pole valmis}$

pöördfunktsioon: $f(n_S) = \text{pole valmis}$

omadused:

Lühikirjeldus: ilmselt mingi fourie pöörde sarnane asi. iga bait ei näita enam, et kas vastava indeksiga lõpprida on eitatud.

9 uus idee CNFiga

Muidu sama, aga eksistentsaalsus kvantorid tuleb asendada universaalsuskvantoritega; väide viia lõpuks CNF vormi ja võitused välja tuua kvantoritest.

10 grupeeritud veeergudega

Jaotada iga taseme harud gruppidesse. Lisaks DNFlõppveerule n_S ka seda, et millised veerud on IGAS tasemes kokku pandud ja, et kas grupp on kokku võitatud või jaatatud. Gruppi kirjeldab madalamas tasemes 1 bitt. Kui grupp on kokku võitatud, siis kirjeldab 1 bitt, et kas nad kõik on ei ole eitatud(1) või on(0). Kui grupp on kokku jaatatud, siis kirjeldab 1 bitt madalamas tasemes, et kas nad kõik on ei ole jaatatud(1) või on mitte(0). Võitusi saab viia kvatoritest välja, kui nii tekiks rohkem bitte(kombinatsioone) madalamasse tasemesse. See on ka a tüüpi funktsioon, kus lihtsalt saa_h ja $saa_haru_märk$ on grupidest sõltuvad(?! (võitused)).

Teise ideena võiks gruppide asemel olla funktsioon harude märkidest. Iga tasemel iseloomustavad $saa_haru_märk$ ja saa_h , et milline

Gruppide vahel DNF vorm.

- N_s pole bijektiivne, kuid kui nõuda, et kõik, mida saab kokku panna on kokku pandud, siis on n_S injektiivne.

11 Hea mõte

funktsioonil $c(h, harude_margid, k)$ on nii palju võimalikke väärtusi kui väite kirjeldamiseks vaja. Millisele harumärkide kombinatsioonile milline funktsiooni c väärtus vastab on lõppveerust eraldi n_S 'is kirjas. Igale harumärkide kombinatsioonile vastab mingi funktsiooni c väärtus. $saa_haru_märk(plato)$ kirjeldab, et milliste $c(h, harude_margid, k+1)$ väärtustega harud selles platos on eitatud ja milliste $c(h, harude_margid, k+1)$ väärtustega harud on eitatud.

Mitu taset allpool on c väärtustele vastavad bitid kirjeldatud võibki olla pakituse parameetrik.

12 ANF, mille osadeks parameetriteks on DNFi väärtus

13 DNF, mille veerud, mida omavahel ainult jaatatud on, on kokku pandud

Lisaks DNFlõppveerule sisaldab väide ka seda, et millised veerud on IGAS tasemes kokku pandud. Kui on kokku pandud, siis kirjeldab 1 bitt, et kas nad kõik on jaatatud või, et ükski neist ei ole jaatatud.

- N_s pole bijektiivne, kuid kui nõuda, et kõik, mida saab kokku panna on kokku pandud, siis on n_S injektiivne.

14 Tasemetega

Vaatab et eelmise taseme järgi, et millised veerud on korras “mitte väite argumenti sõltumatult vale olemise” reegli järgi.

eraldi biitidega (mitte harumärgi funktsiooniga) on kirjeldatud, et millisel platol on millised harud võimalikest. Biti indeks määrab haru omadused. iga plato p harude omadused on kirjeldatud näidates, et: milline on plato p variplatode haru on milliste selle taseme seosefunktsioonide puhul rahuldatud.

15 CNF mille need veerud, mida on väites ainult võitatud, on kokku pandud

16 NNF

17 prefixkuju

alguses olevad bitid näitavad, et milline on univeersaalsuskvantor ja milline on eksistentsiaalsuskvantor.

18 näitevorm

Part VI

väite vastuolulisust kontrolliva programmi kasutamine

19 süntax

19.1 pythoni moodulina

Predikaadid ja kvantorid on vastavate klasside objektid. Loogikatehted on ka objektid ja lisaks python ülelaaditavad operaatorid. On meetodid: `__bool__` ja `kas_min_nested`.

19.2 eraldi fail

19.2.1 predikaadid

Predikaadid on tähistatud kas operaatori-syntaxis või funktsiooni-syntaxis.

operaatorisyntax eraldi lõigud, mille vahel on sulgude sees argumentid. näiteks $(aaa(x_1)(x_2)bbb(x_3)ccc)$ tähendab $aaa()(())bbb()ccc(x_1, x_2, x_3)$. näiteks $(x_1 < x_2)$ tähendab $() < ()(x_1, x_2)$. Funktsioonisyntaxis ei ole vaja predikaatide lõpus olevaid argumente tühjade sulgudega vaja tähistada. Näiteks: $() < ()(x_1, x_2)$ asemel $() < (x_1, x_2)$.

funktsioonisyntax kohtadesse kus operaatori syntaxis oli vahekoht argumenti jaoks on tühjad sulud. juhul kui see predikaat on jutumärkides on tühjade sulgude asemel argumentikohtades kahekordsd jutumärid.

Sama predikaat võib samas väite lähtekoodis nii operaatori kui funktsioonisyntaxis tähistatud olla. Kui predikaadi nimi sisaldab sulge peab see predikaat olema jutumärkide sees. jutumärgid predikaadi nime alguses lõpus või argumenti kohtade kõrval tähistada tagurpidi kaldkriipsuga(nagu pythoni syntaxis.).

Kõik mis pole loogikatehted, sulud ega kvantorid tõlgendada predikaatidena(2 argumentilise puhul nagu operaatori syntax.).

Kõik 16(10) funktsiooni 2 booleanist ühte booleani sisseehitatud operaatoritena või olemas mingi funktsionaalselt täielik hulk neid ja kasutaja saab ise ülejäänud defineerida. Samuti oleks siis 1 kvantor a kasutaja ise saaaks teis defineerida.

peale iga võitust uus rida.

Peale iga kvantorit taane.

kui predikaat sisaldab mingit sümbolit, millel on ka muu tähendus, siis on see kas keelatud või tuleb tagurpidi kaldkriipuga tähistada.

Küsimärkidel eraldi tähendus. Kahekordse kriipsuga järelusmärgidja ekvivalentsusmärkidel eraldi tähendus. {} optimeerimis tingimused.

Eraldi funktsiooni sees saab sisestada käske.

19.3 syntaxsugar

Eraldi võimalik märkida alad kus kehtib mingi syntaxsugar nt. \sugar("boolean"){} ,

19.3.1 kvanteeritavate-nimede sugar

Kui kvantori alaindeksiks pole kvanteeritava nime kirjutatud, siis eeldatakse, et selle kvantori kvanteeritava nimi on x_{k+1} kus k näitab, et mitme kvantori sees antud kvantor on. Et seda süntax-sugarit kasutavas notatsioonis kirjeldatud seost seda süntax-sugarit mitte kasutavas notatsiooni ümber kirjutatada tuleb kvantoritele kvanteeritavate nimed (alaindeksitena) juurde kirjutada.

Näiteks järgnevas valemis, kus pole kvanteeritavate nimesi kirjutatud:

$\exists(\exists(A(x_1, x_2, x_3, x_2))) \wedge \exists(\exists(\exists(A(x_1, x_2, x_3, x_4)) \rightarrow \exists(\neg A(x_1, x_1, x_3, x_4)))) \wedge \neg \forall(\neg A(x_1, x_1, x_1, x_1))$ eeldatakse need olema: $\exists_{x_1}(\exists_{x_2}(\exists_{x_3}(A(x_1, x_2, x_3, x_2)))) \wedge \exists_{x_1}(\exists_{x_2}(\forall_{x_3}(\exists_{x_4}(A(x_1, x_2, x_3, x_4)) \rightarrow \exists_{x_4}(\neg A(x_1, x_1, x_3, x_4)))) \wedge \neg \forall_{x_1}(\neg A(x_1, x_1, x_1, x_1))$

Võib küll tekkida mitmeid samanimelisi kvanteeritavaid(ehmkvantorit fiktiivmuutujaid), aga need ei saa mitte kunagi samas kohas valemis kasutusel olla. Juhul kui tahetakse vabaneda ka olukorrast, kus on mitu samanimelist kvanteeritavat võib kõigi kvanteeritavate nimedele lisada punkti ja peale seda arvu, mis näitab, et mitu samanimelise kvanteeritavaga kvantorit sellest kvantorist vasakul pool asub. Näiteks toodud väite puhul:

$\exists_{x_{1.0}}(\exists_{x_{2.0}}(\exists_{x_{3.0}}(A(x_{1.0}, x_{2.0}, x_{3.0}, x_{2.0})))) \wedge \exists_{x_{1.1}}(\exists_{x_{2.1}}(\forall_{x_{3.1}}(\exists_{x_{4.0}}(A(x_{1.1}, x_{2.1}, x_{3.1}, x_{4.0})) \rightarrow \exists_{x_{4.1}}(\neg A(x_{1.1}, x_{1.1}, x_{3.1}, x_{4.0})))) \wedge \neg \forall_{x_{1.2}}(\neg A(x_{1.2}, x_{1.2}, x_{1.2}, x_{1.2}))$

Sellises süntax-sugariga saab kirjeldada kõiki väiteid, mida ilma selle süntax-sugaritagi saab kirjeldada. Et seda süntax-sugarit mitte kasutavas notatsioonis kirjeldatud seost seda süntax-sugarit kasutavas notatsiooni ümber kirjutatada tuleb kõigepealt muuta kvanteeritavate nimed eespool kirjeldatudeks ja seejärel eemaldada kvanteeritavate nimed kvantorite juurest (alaindeksitena).

Näiteks järgnev valem:

$\exists_a(\neg A(a, a) \wedge \exists_b(A(a, b) \wedge \neg A(b, a) \wedge \neg A(b, b) \wedge \exists_c(\neg A(a, c) \wedge A(b, c) \wedge \neg A(c, a) \wedge \neg A(c, b) \wedge \neg A(c, c)) \wedge \exists_d(\neg A(a, d) \wedge A(b, d) \wedge \neg A(d, a) \wedge \neg A(d, b) \wedge A(d, d))) \wedge \neg \exists_e(\neg A(e, e) \wedge \exists_f(\neg A(e, f) \wedge A(f, e) \wedge \neg A(f, f) \wedge \exists_g(A(e, g) \wedge \neg A(f, g) \wedge \neg A(g, e) \wedge \neg A(g, f) \wedge \neg A(g, g)) \wedge \exists_h(A(e, h) \wedge \neg A(f, h) \wedge \neg A(h, e) \wedge \neg A(h, f) \wedge A(h, h))))$

selle syntax-sugariga kirjeldatuna on:

$\exists(\neg A(x_1, x_1) \wedge \exists(A(x_1, x_2) \wedge \neg A(x_2, x_1) \wedge \neg A(x_2, x_2) \wedge \exists(\neg A(x_1, x_3) \wedge A(x_2, x_3) \wedge \neg A(x_3, x_1) \wedge \neg A(x_3, x_2) \wedge \neg A(x_3, x_3)) \wedge \exists(\neg A(x_1, x_3) \wedge A(x_2, x_3) \wedge \neg A(x_3, x_1) \wedge \neg A(x_3, x_2) \wedge A(x_3, x_3))) \wedge \neg \exists(\neg A(x_1, x_1) \wedge \exists(\neg A(x_1, x_2) \wedge A(x_2, x_1) \wedge \neg A(x_2, x_2) \wedge \exists(A(x_1, x_3) \wedge \neg A(x_2, x_3) \wedge \neg A(x_3, x_1) \wedge \neg A(x_3, x_2) \wedge \neg A(x_3, x_3)) \wedge \exists(A(x_1, x_3) \wedge \neg A(x_2, x_3) \wedge \neg A(x_3, x_1) \wedge \neg A(x_3, x_2) \wedge A(x_3, x_3))))$

või alternatiivselt:

$$\begin{aligned} & \exists(\neg A(1,1) \wedge \exists(A(1,2) \wedge \neg A(2,1) \wedge \neg A(2,2) \wedge \exists(\neg A(1,3) \wedge A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge \neg A(3,3)) \wedge \exists(\neg A(1,3) \wedge \\ & A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge A(3,3))) \wedge \neg \exists(\neg A(1,1) \wedge \exists(\neg A(1,2) \wedge A(2,1) \wedge \neg A(2,2) \wedge \exists(A(1,3) \wedge \neg A(2,3) \wedge \\ & \neg A(3,1) \wedge \neg A(3,2) \wedge \neg A(3,3)) \wedge \exists(A(1,3) \wedge \neg A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge A(3,3)))) \end{aligned}$$

19.3.2 boolean sugar

väiteid saab panna predikaadi argumendiks. $A(\forall(\dots))$ asemel $boolean(x) \wedge (toene(x) \iff \forall(\dots)) \wedge A(x)$

19.3.3 operaatorite (järjekorruga) sugar

1. kuni valemis operaatoreid on:

2. : : paneb operaatoritega valemi ümber universaalsuskvantori nt. $x_{11} + x_{12} * x_{13} = x_{14}$ peale seda sammu: $\forall(x_{11} + x_{12} * x_{13} = x_{14})$

3. : : jaatab olemasolevale väite uus kvanteeritav on võrdne kõige esimesena tehtava operatsiooni tulemusega. näiteks $\forall(x_{11} + x_{12} * x_{13} = x_{14})$ peale seda sammu on: $\forall(* (x_{12}, x_{13}, x_{15}) \wedge x_{11} + x_{12} * x_{13} = x_{14})$

4. : : asendab kõige esimesena tehtava operatsiooni uue kvanteeritavaga, et selle tehte tulemus on võrdne uue kvanteeritavaga. nt. $\forall(* (x_{12}, x_{13}, x_{15}) \wedge x_{11} + x_{12} * x_{13} = x_{14})$ peale seda sammu: $\forall(* (x_{12}, x_{13}, x_{15}) \wedge x_{11} + x_{15} = x_{14})$

näited: $+(x_1, x_2, x_3)$ tähendab $x_1 + x_2 = x_3$

$*(x_1, x_2, x_3)$ tähendab $x_1 * x_2 = x_3$

$CALL(x_1, \dots, x_n)$ tähendab, et funktsioon x_1 argumentidega ... tagastab x_n 'i.

operaatorite järjekord: $**, [*, /, [+ , -]$

näide näiteks teisendab $\exists(\exists(x_1 * x_2 + x_3 = f(x_3, x_2, x_1, x_2)))$ järgenvaks:

$*(x_1, x_2, x_4) \wedge +(x_4, x_3, x_5) \wedge on_f(x_6) \wedge CALL(x_6, x_3, x_2, x_1, x_2, x_7)$

näide näiteks teisendab valemi:

$\exists(on_null(x_1)) \wedge$

$\forall(x_1 = x_1) \wedge$

$\forall(\forall(\forall(x_1 = x_2 \wedge x_2 = x_3 \rightarrow x_3 = x_1))) \wedge$

$\forall(\forall(on_naturaalarv(x_1) \wedge x_1 = x_2 \rightarrow on_naturaalarv(x_2))) \wedge$

$\forall(\forall(on_naturaalarv(x_1) \wedge S(x_1, x_2) \rightarrow on_naturaalarv(x_2))) \wedge$

$\forall(\forall(\forall(= (x_1, x_2) \wedge S(x_1, x_3) \wedge S(x_2, x_4) \leftrightarrow = (x_3, x_4))) \wedge$

$\forall(on_null(x_1) \rightarrow \neg \exists(S(x_2, x_1) \wedge on_naturaalarv(x_2))) \wedge$

$\exists(on_null(x_1) \wedge x_1 \in K) \wedge \forall(\forall(x_1 \in K \wedge S(x_1, x_2) \rightarrow x_2 \in K)) \rightarrow \forall(x_1 \in K) \forall(on_null(x_1) \wedge \forall(x_2 + x_1 = x_2))) \wedge$

$\forall(\forall(x_1 + S(x_2) = S(x_1 + x_2))) \forall(on_null(x_1) \wedge \forall(x_2 * x_1 = x_1)) \wedge$

$\forall(\forall(x_1 * S(x_2) = x_1 + x_1 * x_2))$

kujule:

$\exists(on_null(x_1)) \wedge$

$\forall(= (x_1, x_1)) \wedge$

$\forall(\forall(\forall(= (x_1, x_2) \wedge = (x_2, x_3) \rightarrow = (x_3, x_1))) \wedge$

$\forall(\forall(on_naturaalarv(x_1) \wedge = (x_1, x_2) \rightarrow on_naturaalarv(x_2))) \wedge$

$\forall(\forall(on_naturaalarv(x_1) \wedge S(x_1, x_2) \rightarrow on_naturaalarv(x_2))) \wedge$

$\forall(\forall(x_1 = x_2 \leftrightarrow S(x_1) = S(x_2))) \wedge$

$\forall(on_null(x_1) \rightarrow \neg \exists(S(x_2, x_1) \wedge on_naturaalarv(x_2))) \wedge$

$\exists(on_null(x_1) \wedge x_1 \in K) \wedge \forall(\forall(x_1 \in K \wedge S(x_1, x_2) \rightarrow x_2 \in K)) \rightarrow \forall(x_1 \in K) \forall(on_null(x_1) \wedge \forall(+ (x_2, x_1, x_2))) \wedge$

$\forall(\forall(\forall(\forall(\forall(S(x_2, x_3) \wedge + (x_1, x_2, x_4) \wedge S(x_4, x_5) \wedge + (x_1, x_3, x_6) \leftrightarrow = (x_6, x_5)))))) \forall(on_null(x_1) \wedge \forall(* (x_2, x_1, x_1))) \wedge$

$\forall(\forall(\forall(\forall(\forall(S(x_2, x_3) \wedge * (x_1, x_2, x_4) \wedge + (x_1, x_4, x_5) \wedge \leftrightarrow = (x_6, x_5))))))$

19.3.4 transitiivsete operaatorite sugar

$eta = b = c$ tähendab $a = b \wedge b = c$ nt. operaatorite rakendamise järjekord. `\syntax nt \enable_syntaxsugar(kommunitiivsed_võdusmärgid)`

19.3.5 kantsulgude sugar

Kantsulgudel sulgudel selline tähendus, et $[aOP1bOP1cOP1d]OP2e = (aOP2e)OP1(bOP2e)OP1(cOP2e)OP1(dOP2e)$.

19.3.6 predikaatide overloading

võib olla mitu sama nime, aga erineva argumentide arvuga, predikaati. kui argumentide arv on erinev, aga siis on tegemist erinevate predikaatidega - sellel et nimi on sama pole mingit tähtsust.

19.3.7 väited predikaatide argumentidena

Elementaarseoste argumentideks saavad olla väited kui eelnevalt on defineeritud True ja False. Et nende tähendus olks sama, mis intuiitiivne tähendus peab lisama ka nendega seotud väited $\forall(\forall("onTrue"(x_1) \wedge "onTrue"(x_2) \rightarrow " = "(x_1, x_2)) \wedge ("onTrue"(x_1) \wedge "onFalse"(x_2) \rightarrow \neg " = "(x_1, x_2))))$. Elementeersesos(väide) tähendab: $(vide \rightarrow \exists("onTrue(x_1) \wedge "elementaatseos"(x_1)) \wedge (\neg vide \rightarrow \exists("onFalse"(x_1) \wedge "elementaatseos"(x_1))))$

19.4 importimine

Kasutaja ei peaks iga kord uuesti kõiki (näiteks reaalarvude kohta käivaid) aksioome sisestama, vaid peaks saama neid moodulina "importida" ("includida").

pannes sisendisse `\include_from_URI(foo)` asendatakse sisendis see käsk URII foo olevaga.

`\include_from_standarlibari(bar)` asendatakse sisendis see käsk standarlibaris kohal bar olevaga.

importitud väidetel on võivad olla viimased sulud puudu, sest need ütleva, et midagi eksisteerib, aga lasevad kasutajal sama kvantori (mille lähtekood on teises failis) väiteid juurde kirjutada.

Moodulid on eelnevalt "kompileeritud". Importimisel nende väitenumbrit muudetakse nii, et see sobiks kokku ülejäänud predikaatide ja K'ga. Nii on efektiivsus suurem. Moodulite lähtekood ei pea kõigi moodulite puhul avalik olema.

19.5 näiteid selles syntaxis definitsioonides

19.5.1 piirväärtus

$$(\lim_{x \rightarrow a}(f(x)) = A) \leftrightarrow \forall \varepsilon (\varepsilon > 0 \rightarrow \exists \delta (\delta > 0 \wedge \forall x ((|x - a| < \delta) \wedge (|x - a| \neq 0)) \rightarrow |f(x) - A| < \varepsilon))$$

Iga kauguse ε funktsiooni piirväärtusest kohal a , kohta leidub kaugus δ kohast a nii, et iga koht x , mis on a 'le lähemal kui δ , aga mitte kohas a , korral on funktsioon kohal x lähemal A 'le kui ε .

Minu süntaksis on see definitsioon: $(\lim_{x \rightarrow a}(f(x)) = A) \leftrightarrow \forall x_1 (x_1 > 0 \rightarrow \exists x_2 (x_2 > 0 \wedge \forall x_3 (((|x_3 - a| < x_2) \wedge (|x_3 - a| \neq 0)) \rightarrow |f(x_3) - A| < x_1)))$

19.5.2 peano aksioomid

selles näites on kasutatud peatükis 19.3.3 kirjeldatud syntaxsugarit.

$$\begin{aligned} & \exists(on_null(x_1)) \wedge \\ & \forall(x_1 = x_1) \wedge \\ & \forall(\forall(x_1 = x_2 \wedge x_2 = x_3 \rightarrow x_3 = x_1))) \wedge \\ & \forall(\forall(on_naturaalarv(x_1) \wedge x_1 = x_2 \rightarrow on_naturaalarv(x_2))) \wedge \\ & \forall(\forall(on_naturaalarv(x_1) \wedge S(x_1, x_2) \rightarrow on_naturaalarv(x_2))) \wedge \\ & \forall(\forall(x_1 = x_2 \leftrightarrow S(x_1) = S(x_2))) \forall(on_null(x_1) \rightarrow \neg \exists(S(x_2, x_1) \wedge on_naturaalarv(x_2))) \wedge \\ & \exists(on_null(x_1) \wedge x_1 \in K) \wedge \forall(\forall(x_1 \in K \wedge S(x_1, x_2) \rightarrow x_2 \in K)) \rightarrow \forall(x_1 \in K) \end{aligned}$$

7. aksioom ilma väiteta, et kõik naturaalarvude alamhulgad eksisteerivad on kasutu.

19.5.3 liitmine

selles näites on kasutatud peatükis 19.3.3 kirjeldatud syntaxsugarit.

$$\forall(on_null(x_1) \wedge \forall(x_2 + x_1 = x_2))) \wedge \\ \forall(\forall(x_1 + S(x_2) = S(x_1 + x_2)))$$

19.5.4 korrutamine

selles näites on kasutatud peatükis 19.3.3 kirjeldatud syntaxsugarit.

$$\forall(on_null(x_1) \wedge \forall(x_2 * x_1 = x_1)) \wedge \\ \forall(\forall(x_1 * S(x_2) = x_1 + x_1 * x_2))$$

20 standard library

standard librarys on igasugu kasulikke väiteid, mida kasutaja saab enda omadega jaatada, et enda omade tõesust kontrollida. Praegu ma standarlibrary olulisel määral veel kirjutada ei ole jõundnud, aga siin on mõned näited asjadest, mis sreal olla võiks:

$$\text{robinsoni aritmeetika} \quad \forall(on_null(x_1) \neg \exists(S(x_1, x_2))) \wedge \\ \forall(\forall(\forall(x_1 = x_2 \wedge S(x_1, x_3) \wedge S(x_2, x_4) \rightarrow x_3 = x_4))) on_null(x_1) \vee \exists(S(x_2, x_1)) \forall(on_null(x_1) \wedge \forall(x_2 + x_1 = x_2))) \wedge \\ \forall(\forall(x_1 + S(x_2) = S(x_1 + x_2))) \forall(on_null(x_1) \wedge \forall(x_2 * x_1 = x_1)) \wedge \\ \forall(\forall(x_1 * S(x_2) = x_1 + x_1 * x_2))$$

$$\text{astendamine} \quad \forall(on_null(x_1) \wedge \forall(x_2^{x_1} = S(x_1))) \text{iga arv astmes 0 on 1.} \\ \forall(\forall(x_1^{S(x_2)} = x_1^{x_2} * x_1))$$

$$\text{lahutamine} \quad \forall(\forall(\forall(x_1 - x_2 = x_3 \leftrightarrow x_2 + x_3 = x_1)))$$

$$\text{jagamine} \quad \forall(\forall(\forall(x_1 / x_2 = x_3 \leftrightarrow x_2 * x_3 = x_1)))$$

$$\text{peano viimane naturaalarvude hulga võimsuse aksioom} \quad \exists(on_null(x_1) \wedge x_1 \in K) \wedge \forall(\forall(x_1 \in K \wedge S(x_1, x_2) \rightarrow x_2 \in K)) \rightarrow \forall(x_1 \in K)$$

$$\text{kõik naturaalarvud on reaalarvud} \quad \forall(on_reaalarv(x_1) \rightarrow on_naturaalarv(x_1))$$

$$\text{reaalarvude võrdlemine} \quad \forall(\forall(x_1 > x_2 \leftrightarrow \exists(on_naturaalarv(x_3) \wedge x_1 * x_3 > x_2 * x_3))) \\ \text{reaalarvude pid}$$

$$\text{täisarvu definitsioon} \quad \forall(on_tisarv(x_1) \leftrightarrow \exists(on_naturaalarv(x_2) \wedge on_naturaalarv(x_1 + x_2)))$$

$$\text{ratsionaalarvu definitsioon} \quad \forall(on_ratsionaalarv(x_1) \leftrightarrow \exists(on_naturaalarv(x_2) \wedge on_tisarv(x_1 * x_2)))$$

21 lisa funktsioonid

21.1 lähtekoodi värvimine

- sulud mustad ja vahelduvate suurustega või vahelduvate värvidega.
- loogikatehted tumesinised
- kvatorid helesinised
- predikaadid oranžid

Ebaolulised kohad, mis midagi ei mõjuta muudab lähtekoodis halliks. Kuvab igast predikaadist ainult nii mitu esimest sümbolit, et viimane sümbol pole sama mis mõnel teisel predikaadil.

21.2 tagastab ka lähtekoodi latexis ja lähtekoodi latexi lähtekoodi ja sama lihtsustatud väite kohta

21.3 kuvab puugraafina ka valitud plato

alguses ainult juure. Kui klõpsata, siis kuvab plato, millele klõpsiti harud.

22 Gödeli teoreemid

Antud süntaksis on Gödeli teoreemid ilmselged. Need on üsna mõtetud.

kui sain gödeli teoreemidest õieti aru, siis

esimene teoreem väidab, et: Iga väite S_1 jaoks leidub mingi teine väide S_2 , mille jaatamisel esimesega (S_1 'ga) ei saa ei üheselt vale ega üheselt õiget tulemust (ei ole lihtsustatav Trueks ega Falseks).

$$\forall_{x_1} (\exists_{x_2} (on_piisavalt_aritmeetikat(x_1) \rightarrow ()))$$

Gödeli lause on kõige väiksema K_g ja väikseima numbriga väide, mille tõesus ega valesus antud väitest ei järeldu.

See originalisõnastuses, et need samas keeles peavad olema, tähendab minu süntaksis, et need peavad sisaldama samu predikaate.

teine Gödeli teoreem

hüpotees1 originaal sõnastus on:

“ühelgi kooskõllalisest süsteemist, milles on piisavalt aritmeetikat, ei järeldu selle kooskõllalisus.”

antud süntaxis on sõnastus:

“ühegi väite, mille mitte-üheseltvalesus (kooskõllisust) ei saa näidata lõpliku arvu näidete abil, mitte-üheseltvalesus ei saa tõestada.”

Naturaalarvude aksioomide kooskõllalisust ei saa selle näiteväitega jaatamise abil näidata, sest iga igakord naturaalarvu kohta eitatud kvantorite mitterahudamiseks luua uus näide (ja tüübiunktsiooniväärtus) ja sellele omakorda uus. Piisavalt aritmeetikat sisaldab süsteem siis kui see nõuab lõpmatult paljude asjade, millest ühegi pole teiste asjadega täpselt ientsed kuulumised. piisavalt palju aritmeetikat sisaldab süsteem kui selle mitteüheseltvalesust ei saa tõetada selle jaatamisel näiteväitega.

Teine hüpotees on, et piisava aritmeetika peano aksioomiele toob hoopis f-aksioom.

Võimalik, et Gödeli II teoreem väidab, et osade väidete mitteüheseltvalesust ei saagi tõestada. Kui saab tõestada et mingi väite mitteüheseltvalesust ei saa tõestada, siis tõestab, see pole üheselt vale?

hüpotees2 Naturaalarvude aksioome ei saa selles süntaksis kirjeldada, sest ei saa väita, et mingi hulga võimsus võrdne või suurem naturaalarvude hulga alamhulkade võimsusest. Seda on vaja, et väites, et kui mingi 0 kuulub mingisse hulka ja sellest, et mingi naturaalarv mingisse hulka kuulub järeldub, et iga naturaalarv sellesse hulka kuulub kuuluvad kõik naturaalarvud sellesse hulka. Pole ju võimalik väita, et kõiki naturaalarvude alamhulgad üldse eksisteerivad.

Kuigi Gödeli teoreemi kohaselt pole võimalik kogu täisarvude matemaatikat aksiomaatiliselt kirjeldada(???), saab arvutite etteantavasse väiteesse alati uusi aksioome juurde lisada ja tõestada, et teatud arvust K_{max} väiksemate K parameetritega väidete abil kirjeldatavad aksioomid on lisatud.

23 sünonüümidest

Kuna erinevates allikates kasutatakse samade asjade kohta erinevaid nimetusi, toon siin välja, et kuidas on mujal nimetatud asju, mida siin failis kasutan.

väide ehk: formaalne süsteem, aksiomaatiline süsteem, valem, lause, seos.

jaatus ehk: konjutsioon.

võitus ehk: disjunksioon.

üheselt vale ehk: samaselt väär, vastuoluline, samaselt vale, mitte kehtestatav.

üheselt tõene ehk: samaselt tõene.

eksistentsaalsuskvantor ehk: olemasolukvantor.

universaalsuskvantor ehk: üldisuskvantor.

elementaarseos ehk: notatsiooni poolt ette antud predikaat.

element ehk: algasi, konstantsümbol.

kvanteeritav ehk: kvantori fiktiivmuutuja

haru ehk: veerg

DNF-vorm ehk: disjuktiivne normaalkuju.

CNF-vorm ehk: konjuktiivne normaalkuju.

Osades käsitlustes on ka kandjahulk, aga selle faili terminoloogias on selleks universaalne hulk (või seda pole) seega väited kehtivad kõigi asjade kohta. Kui tahad et väide kehtiks näiteks ainult naturaalarvude kohta, siis pane $\forall_x(f(x))$ asemele $\forall_x(\text{on_naturaalarv}(x) \rightarrow f(x))$

24 LVd, mida saab madalama K'ga kirjeldada

Kui kui kõik bitid on nullid või ühed on väide vastavalt üheselt vale või üheselt tõene.

K=1 lõpprida saab madalama K'ga kirjeldada ainult siis kui kõik selle bitid on 1'ed või 0'id.

väidet ei saa madalama K'ga kirjeldada ainult siis kui see sisaldab K'd kvanteeritavat, millest moodustub ring. Ühendatuks loetakse kvanteeritavaid, mille vaheline elementaarseos on kirjeldatud.

24.1 hüpotees: kui o1 lõppveerust on ÜV saab väidet kirjeldada väiksema K'ga.

Vale, sest _._._

24.2 hüpotees: kui o2 lõppveerust on ÜV saab väidet kirjeldada väiksema K'ga.

Vale, sest _._._

Part VII

Sorteerimata

vahest saab väite ÜVsust ÜTsust kontrollida ilma lõppveergu leidmata.

o1'e sarnast funktsiooni lõppreale rakendades saab võibolla kontrollida, et kas väide on minimaalse K ja predikaadidega kirjeldatud.

Need lõppveerud, mis ei tee osade predikaatide koha mingeid väiteid tuleb ka eemaldada nagu needgi, mida saab madalma K'ga esitada ja need, mis jaatavad ÜV LR'e.

väite parssimisel: alustades kõige sisemisemast platost iga kvantori puhul määrata, et mis numbrid saavad olla tema platonumbrid. Selle arvestamiseks teha vastavad loogikatehted tema elementaarseostega eelastega ja tema sees

olevate kvantorite võimalike platoniumbritega. See on võtuste välja viimeseks efektiivsem võimalus kui asja täis DNF vormi viimine. Et määrata kas väide on ÜV vaadata, et kas mõni lõppveerg rahuldab talle seatud tingimust.

Kui asendada kvantorid tõenäosus funktsiooniga($\forall_x(f(x))$ asendada $\forall_x(P(x))$ ja $\exists_x(f(x))$ asendada $1 - \forall_x(1 - P(x))$) ja võitamine liitmisega ja jaatamine korrutamisega(sest kõik harud sh. lõppread on omavahel sõltumatud.) siis saab tõenäosusi arvutada. Kui asendada need kompleksarvuliste lainefunktsioonidega saab kvantarvutusi teha?forall< Panna eelnevad 2 mõtet väitevormis kirja. Nii saabgi vist tõenäosuste kirjeldamiseks vajalikud aksioomid. vähem üldises mõttes on univrsaalus kvantor($\Pi(f(x))$) jaatus ja eksistentsiaalsuskvantor võitus.

lõplik: $\vee; \wedge$

loenduv: $\sum_{i=0}(); \Pi_{i=0}()$

üldine: $\exists_x(); \forall_x()$

:tõenäosus: $1 - P(\neg); P()$

kvant?:lainefunktsioon?

P1 ja P2 “sümmeetrilised”? Seeega mõlemad ahelas?

Asenada plato p1 platosse p2. Ehk võrdsustada plato p1 kvanteeritav plato p2 kvanteeritavaga. Ehk võrdsustada plato kvanteeritava p1 elementaarseosed plato p2 kvanteeritava elementaarseostega?

Iga plato kvanteeritava elementaarseosed(ka need mis on selle plato järglastes kirjeldatud) mistahes teise plato kvanteeritavaga peavad olema samad kui mistahes teise plato vähemalt ühe haru kvanteeritava elementaarseosed?

Iga kvanteeritava elementaarseosed(ka need mis on selle plato järglastes kirjeldatud) mistahes teise kvanteeritavaga peavad olema samad kui mistahes teise plato vähemalt ühe haru kvanteeritava elementaarseosed?

plato p_i kirjeldab ka eelasi, aga plato harunumber n_i ei kirjelda, et mis on selle haru eelased.

ks P2(i)’ga võrdsustatakse P1(i) või P2(i) haruga?

Kui ÜV jaoks vaja ainult jaatatud harusid kontrollid, siis on vaja 2^K korda vähem harusid kontrollida, sest keskmiselt on igal tasemel 2 korda vähem jaatatud harusid kui harusid kokku.

onhulk(x) aseme panna eestikeelne_nimetus_on(UTFencoding(“hulk”),x) kui seda teisendada tavakeelde, siis lihtsalt need abipredikaadid vahele jätta.

Kui võtta konstantsed predikaadid ja defineerida kasutaja predikatide asemel booleantagastusega funktsioonid, siis saab piirata predikaatide arvu ja predikaatide argumentide arvu(vähemalt 3 on iga väite kirjeldamiseks vajalik), aga mitte K’d.

Vist on paremini sõnastatud sama tähendusega lõik olemas, aga jätan ka selle lõigu siia: “piisab kui kontrollida et ükski vorm, kus igale platole p1, kuhu pääseb ilma eitusi läbimata, on kooskõlisuse kontrollimiseks pandud vastavusse(elementaarsetosed võrdsustatud), mistahes teine plato p2, kuhu pääseb ilma eitusi läbimata, ja kontrollitud, et see sobiks võiks täita selle haruga sama tingimust, ei oleks üheselt vale. Samale p1’e võib asendada mitmele p2’ele vastavusse. p1 ja p2 on kooskõlas kui need on kõigi asjadega samas elementaarseostes.”

panna minu AR(automated reasoning) algoritm kokku AI algoritmidega nii et AI leiab optimeerimiseks häid tingimusi ja hüpoteese mida AR saab kontrollida näiteks korrapärade tõttu kastaja sisendis.

Seoe mitte üheselt valesuse(kehtestavuse) tõestamiseks piisab kontrollimises, et üks lubatud lõpprida ei ole pheselt vale. väite mitte üheselt tõesuse tõestamiseks piisab kontrollimisest, et väite eituse vähemalt üks lubatud lõpprida ei ole pheselt vale.

Et kontrolida kas väide on kirjeldatav väiksema K’ga ei tule kontrolida mite lõppridu erldi vaid lõppveergu. Siis(kuid vist mitte ainult siis) kui väide on kirjeldatav väiksema K’ga kui kõrgeima taseme kvantori sisu ei sõltu osadest kvanteeritavatest. Vahest üheselt valede ja madlama K’ga esitatavate koos kontrollimine on matemaatiliselt ilus.

Vahest on kasulik jätta lõppridade asemel hoopis Lõppveerge vahele, mis jaatavad üheselt valesid ridu?

Gödeli teoreemide kohaselt ei saa tõestada, et peano aksioomid on kooskõlalised, aga ometi on selge, et need on kooskõlalised.

Huvitav on väide, et $\forall_x (False)$, ehk $\neg \exists_x (True)$ ehk ei eksisteeri midagi. Ma pole kindel, et kas see on üheselt vale või mitte. Sellise võimaluse lubatust kirjeldab DNF vormis kõige viimane lõpprida.

<https://community.wolfram.com/groups/-/m/t/2416379> : Why does Wolfram|Alpha not return True for following inputs:

trueQ[!(Exists[x,Exists[y,A[x,y]]] && !Exists[x,Exists[y,A[y,x]]]) Simplify[!(Exists[x,Exists[y,A[x,y]]] && !Exists[x,Exists[y,A[y,x]])]

Lineaarne esitusviis.

predikaadid min tasemel

ei tööta vist in nt järgneva LRi korral:

$$\neg E(A(x_1) \wedge \neg E(D(X_1, x_2))) \wedge \neg E(B(x_1) \wedge \neg E(D(x_2, x_1)))$$

predikaadid max tasemel

kujul $\neg E(\neg E(\neg E(\neg E(\neg E(\neg E(\neg E(BOOL(predikaatvited))))))))$

võibolla selliste väidete lihtsustamist(sh. üheselt valede ridade otsimist) ei saagi algoritmilselt teha ja see tekitab teadvust.

defineerida ka loogiliste tehete omavahelised väited nt $\forall_{x_1} (\forall_{x_2} (VOI(x, y) \iff \neg JA(\neg x, \neg y)))$

Rakendada mingit algoritmi, mis on mõeldud suvalise “madala entroopiaga” data kokku pakkimiseks.

	kõigi hulkade kohta ei või väiteid teha	kõigi hulkade kohta võib väiteid teha
kvatorid kehtivad ka alghulkade kohta		
kvatorid ei kehti ka alghulkade kohta	$2^{N^2} \cdot (2^{2^N} - 1)^{2^{N+1}}$	$2^{2^{2 \cdot N+1} + N^2}$

$$F_Q = \sum_{n_v=0}^{2^{2 \cdot N+1}-1} (\sum_{i_x=0}^{N-1} ((\lfloor \frac{Q}{2^{N^2-1-i_x \cdot N-i_x}} \rfloor \bmod_2 \nrightarrow \lfloor \frac{n_v}{2^{2 \cdot N}} \rfloor \bmod_2) \wedge (n_v \bmod_{2^N} = (Q \cdot 2^{N \cdot (1+i_x-N)}) \bmod_{2^N}) \wedge \prod_{j=0}^{N-1} (\lfloor \frac{Q}{2^{N^2-1-j \cdot N-i_x}} \rfloor \bmod_2 \nrightarrow \lfloor \frac{n_v}{2^{2 \cdot N-1-j}} \rfloor \bmod_2)))$$

	kõigi hulkade kohta ei või väiteid teha	kõigi hulkade kohta võib väiteid teha
kvatorid kehtivad ka alghulkade kohta		
kvatorid ei kehti ka alghulkade kohta		$\leq 2^{2^{K(K+N \cdot 2)} + N^2}$

KOGU_INFO on väide, mis sisaldab kogu tõest infot kige kohta. Loomulikult seda KOGU_INFOT ei ole mul kuhugi salvestatud. See on lihtsalt abstraktne mõiste.

vastuolulisuse valesususe erinevus

vastuolulisus(ehk üheselt valesus) Mingi väide V on ÜV kui väide, mis saadakse lisades väite ette eksistentsiaal-suskvantorid, nende sisse et need kvanteeritavad on booleanfinktsioonid ja asendades predikaadid V's nende kvantorite kvanteeritavatega.

näiteks väide $\exists(\exists(A(1,2) \wedge \exists(A(2,3)))) \wedge \neg \exists(A(1,1) \vee \exists(A(1,2) \wedge A(2,1)))$ ei ole vastuoluline, sest väide $\exists(on_booleanfunktsoon(x) (\exists(\exists(x_1(1,2) \wedge \exists(x_1(2,3)))) \wedge \neg \exists(x_1(1,1) \vee \exists(x_1(1,2) \wedge x_1(2,1))))$ ei ole vale.

näiteks väide $\exists(\exists(A(x_1, x_2) \wedge \exists(A(x_2, x_3) \wedge A(x_2, x_2)))) \wedge \neg\exists(\exists(A(x_2, x_1) \wedge A(x_1, x_3) \wedge A(x_1, x_1))))$ on vastuoluline, sest väide $\exists(on_booleanfunktsioon(x_1) \wedge \exists(\exists(x_1(x_2, x_3) \wedge \exists(x_1(x_3, x_4) \wedge x_1(x_3, x_3)))) \wedge \neg\exists(\exists(x_1(x_3, x_2) \wedge x_1(x_2, x_4) \wedge x_1(x_2, x_2))))$ on vale.

valesus Väide on vale kui selle jaatus *KOGU_INFO* ga on vastuoluline. $VALE(VIDE) \leftrightarrow V(VIDE \wedge KOGU_INFO)$

üheselt tõesus Mingi väide *V* on ÜT kui väide, mis saadakse lisades väite ette univrsaalsuskvantorid, nende sisse et need kvanteeritavad on booleanfunktsioonid ja asendades predikaadid *V*'s nende kvantorite kvanteeritavatega.

näiteks väide $\exists(\exists(A(1, 2) \wedge \exists(A(2, 3)))) \wedge \neg\exists(A(1, 1) \vee \exists(A(1, 2) \wedge A(2, 1)))$ ei ole vastuoluline, sest väide $\forall(on_booleanfunktsioon(x_1) \wedge \exists(\exists(x_1(1, 2) \wedge \exists(x_1(2, 3)))) \wedge \neg\exists(x_1(1, 1) \vee \exists(x_1(1, 2) \wedge x_1(2, 1))))$ ei ole vale.

näiteks väide $\exists(\exists(A(x_1, x_2) \wedge \exists(A(x_2, x_3) \wedge A(x_2, x_2)))) \wedge \neg\exists(\exists(A(x_2, x_1) \wedge A(x_1, x_3) \wedge A(x_1, x_1))))$ on vastuoluline, sest väide $\forall(on_booleanfunktsioon(x_1) \wedge \exists(\exists(x_1(x_2, x_3) \wedge \exists(x_1(x_3, x_4) \wedge x_1(x_3, x_3)))) \wedge \neg\exists(\exists(x_1(x_3, x_2) \wedge x_1(x_2, x_4) \wedge x_1(x_2, x_2))))$ on vale.

tõesus Väide on tõene kui selle jaatus *KOGU_INFO* ga ei ole vastuoluline. $TENE(VIDE) \leftrightarrow \neg V(VIDE \wedge KOGU_INFO)$

teisest failist

Uue praktilise töö eesmärgid

Sissejuhatus Käesoleva praktilise töö eesmärk on luua programm, mis suudab etteantud loogikalause põhjal kasutaja küsimustele vastata, loogikalauseid minimaalse vabadusastmega kirja panna, väite keerukust hinnata, loogikalauseid inimkeeles kirja panna, loogikalauseid, mis sisaldavad füüsikalisi algmõisteid illustreerida (nt. Punkt-massi koordinaatide sõltuvusest ajast graafiku joonistades) ja loogikalisi lauseid elemetaartehte (elementaartehteid omakorda LaTeXi või mõne programmeerimiskeele süntaksi) abil estada kui see võimalik on. Programmil on ka kasutaja liides tuntumate „asjade” (füüsikaliste mudelite) omadusi kirjeldavate lausete parameetrite abil moodustamiseks.

Eesmärgid:

- Füüsikaline seos on seos füüsikaliste mõistete vahel.
- Juhul kui ühest kujust teise teisendamine ei eelda ühegi füüsikalise suuruse vahelise seose teadmist on tegu tähistuse küsimusega.
- Pärismõiste on mõiste, mis tähistab midagi, mis on päriselt olemas ja omab füüsikalist tähendust ega sõltu mudelist ega tähistusest, milles seda kirjeldatakse.
- Lause keerukus on seda kirjeldava loogikalise lause eelmainitud viisil kirjapandud värgi suurus (juhul kui see on naturaalarvuna kirja pandud).
- Füüsikalise väite(mudeli) kasulikkus on seda suurem, mida täpsemalt see katseandmetega klappib ja mida lihtsam see on(mida väiksem on selle keerukus). Lihtsus on ilmselt õige hinnata kui füüsikalised suurused on kirjapandud minimaalsel vabadusastmel ja ja võimalikult sümmeetriliselt(ruumiline ja ajaline sümmeetria näiteks).
- Kasutajale esitatakse väited ja loogikalaused matemaatiliste mõistete abil (nt. Liitmine, korrutamine, univrsaalsuskvantor, loogikalised tehted(nt. XOR, AND, OR, NAND, NOR)).
- Ka matemaatilised mõisted tuleb igas väites ja loogikalauses matemaatiliste algmõistete abil defineerida.
- Luua kasutaja liides erinevaid spetsiifilisi olukordi kirjeldavate väidete (Nt. N-keha probleemi põhivõrrandite erinevates lähendustes) mugavaks kirja panemiseks.
- Luua programm, mis teisendab sisestatud väiteid erinevate algmõistete vahel.
- Luua programm, mis teisendab väite esitust nii, et sama väide oleks alati samal kujul kirja pandud.

- Eristada selgelt matemaatilised algmõisted (funktsioon, võrdsus, hulk, implikatsioon jne.(siin näitena toodud mõisted ei moodusta minimaalset funktsionaalselt täielikku hulka)) ja füüsikalised algmõisted (ruum, aeg, elektromagnetväli, elektrimagnetväli, gravitomagnetväli, gravitimagnetväli, aegruum). Võtta kasutusele ka füüsikalised mõisted, mis ei ole algmõisted (nt. konkreetsed ained, tahke, vedel, gaasiline, aatom).
- Defineerida kõik mõisted (sh. Arvud (sh. Naturaalarvud)) algmõistetes.
- Luua programm, mis teisendab väite esituse, kasutaja valitud algmõistetele. Kui valitud algmõisted ei moodusta funktsionaalselt täielikku rühma, siis annab programm veateate. Kui valitud algmõisted moodustavad funktsionaalselt täieliku, kuid mitte minimaalse, siis teisendab programm väite esitust niimoodi, et saadud esitus oleks võimalikult lihtne.
- Ükski tõestamatu väide ei tohi omada füüsikalist tähendust (näiteks ei oma koordinaatide alguspunkti valik füüsikalist tähendust.). Ükski füüsikalise tähenduseta väide ei tohi olla tõestatav. • Defineerida ainult vastava füüsikalise seose kirjapanemiseks vajalik algebra.
- Füüsikalise mudeli, mis sisaldab suuruseid(nt. Aeg, kordinaat ja väljatugevus) , mida iseelomustakse reaalarvuga, kirjeldamiseks tuleb defineerida vastavad aritmeetilised tehted reaalarvudele. Kas alati tuleb defineerida ka jagamine, et saaks väita nt. ,et üks suurus on teisest 3 korda suurem?
- Ainuke asi, mida me otseselt tajuda suudame on meie enese teadvus. Füüsikalist mõtet omavad mõisted (ehk on füüsikalised mõisted), mis kuidagi mõjutavad teadvust. Kuna ei ole olemas piisavalt head füüsika-teooriat teadvuse kirjeldamiseks(selle teiste füüsikaliste suurustega „sidumiseks”), siis tuleb füüsikaliste süsteemide kirjeldamiseks kasutada mõisteid, mis ilmselgelt kaudselt(meeleelundite kaudu jõuab info ajju, mis mõjutab teadvust) meie teadvust mõjutavad. Mõistlik on selliseid suurusi, millest ainult intuiitiivselt võib aru saada(füüsikalised algmõisted),võimalikult vähe(valida minaalne funktsionaalselt täielik algmõistete hulk) kasutada ja teised mõisted (füüsikalised mitte-alg mõisted) nende kaudu defineerida.
- Gödeli teoreem ütleb, et iga piisava aritmeetikaga seoses on vähemalt 1 väide, mida ei saa tõestada, ümberlükata ega tõestada, et seda ei saa tõestada. Kas siis kui teha mingi pikkusega üritus tõestada üritamiseks saab öelda, et mingi tõenäosusega seda tõestada ei saa? Kas kui pikkus on piisav, siis peaaegu kindlasti ei saa seda tõestada?
 - Kui füüsikalise suuruse(näiteks asukoht) väärtus võib olla null, siis on ka null algmõiste.(vist)
- Korrumise (ja tesite matemaatiliste mõistete) intuiitiivsed omadused tulevad välja alles arvude füüsikaliste algmõistetega(näiteks geomeetriliste(nt. Pikkuse või nurga)) seostamisel. Matemaatika aksioomid peavad looma ainult seoseid matemaatiliste mõistete vahel. Ei pea muretseda selle üle, et matemaatilistest aksioomidest ei järeldu, et arvteljed ei ole „välja venitatud”.
- Sulud on selle genereeritud „tavakeele” süntaksis sees. „(, hääldatakse SU ja ,,)” hääldatakse US.
- eelliide(prefix) näitab, et kas sõna on kasutatud ranges või vabas(analoogiana või naljana) tähenduses.
- Veergude omavahel jaatamisega(konjunkteerimisega) ja neile valikuliselt eituse ja jaatuse ette panemisega peab saama ühe viisil moodustada kõik read. • Read on väited, mille vabalt valitud elementide omavahel võitamisega(disjunkteerimisega) peab saama ühel viisil moodustada mistahes väite, mida antud N'i(alghulkade arvu) ja K(kvantorite arvu) puhul saab moodustada.
- Rida: [alghulkade vahelised seosed]JA KVANTOR([mis alghulkadesse kvanteeritav kuulub] TO [millistesse alghulkadesse kuulub])JA MITTE KVANTOR([mis alghulkadesse kvanteeritav kuulub] TO [millistesse alghulkadesse kuulub])
- Tuleb eeldada, et kõik hulgad, mida saab defineerida on olemas, sest muidu ei saa ühegi hulga puhul kindel olla, et kas see on olemas (kui mingis teoorias mingite omadustega hulki ei ole, siis teha eraldi hulk hulkadest, mis selles teoorias olemas on.). PF! Programm peaks selliseid väiteid siiski lugeda oskama, et kasutajale veateade väljastada. • Peaaegu kunagi ei ole K=1 korral veergude arv täisarv. K=0 puhul on alati veerge täisarv.
- Ridased on vist alati täisarv. Seega on kõik bittide kombinatsioonid „sisus” lubatud. Ehk kõi võimalused veergude bittidega täitmiseks lubatud.

- K näitab, et kui „sügaval” kvantorite „sees kõige „sisemine” seos on. O
- N näitab, et kui palju alghulki on.
- NP probleem panna kirjakvantoriga kujul. See seab kvanteeritavale tingimuse, et kas see kuulub (õigete) vastuste hulka. $\forall x(\text{TINGIMUS}(x) \rightarrow x \in \text{VASTUSED})$. Iga probleem on niimoodi kirjeldatav. Teha uus seos, mis kirjeldab pakutavat. Jaatada probleemi kirjeldavat seost ja pakkumist kirjeldavat seost. NB. See ainult kontrollib pakutud vasuse tõesust!
- Loenduva hulga puhul saab booleansumma asendada universaalsuskvantoriga ja korrutise eksistentsiaal kvantoriga. (symbolitega seletada)
- NP probleemi paremaks sõnastamiseks tuleks kirja panna, et millistele tingimustele peab (protsessor) ühe sammuga tehtavad operatsioonid vastama, alles siis uurida „et kas sammude arv on polünoomilises seoses sisendi pikkusega. Need tingimused võiks olla näiteks: 1. võimalikke operante kindel (lõplik) arv (seega on võimalik iga võimaliku operandi mingis kindlas vahemikus olevatele naturaalarvudele bijekteerida.). 2. operatsiooni operande kindel (lõplik) arv.
- Kas K'1 on piir? Kas näiteks mingi funktsioon (näiteks $\forall f(f \in F \leftrightarrow \forall x(f(x)))$) ei saa asendada kvantoreid?
- Ka arvutiprogramme saab sellisel viisil kirja panna, iga lähtekoodi real on lihtsalt ühe võrra suurem”käsu numbri argument”. See kirjeldaks programmi sisendi ja väljundi vahelist seost. • Kui kirja panna võimalikud „elementaarkäsud”(ja nende täitmiseks kuluva protsessori taktide arvu) saab kirjeldada seose, milles on otsitavaks programm kõige optimeeritumal kujul.
- Sisendi optimeerimismeetodiks võib kaaluda masinõppe algoritmide kasutamist.
- Kui tõestuses järeldeb mingist sammust ka eelmine samm, siis teha kahepoolse noole üks ots halliks. Halliks tagasi suund.

Näited:

Siin on mõned näited füüsikalistest süsteemidest ja tähistusviisidest.

- Füüsikaline mudel, mis ütleb, et kui on soe, siis ei ole külm; kui on külm siis ei ole soe; kui on suur, siis ei ole väike; kui on väike, siis ei ole suur. Matemaatilised algmõisted: NAND, sulud Füüsikalised algmõisted: soe; külm; suur; väike Seos: ((soe NAND külm) NAND (suur NAND väike)) NAND ((soe NAND külm) NAND (suur NAND väike))
- 1. Füüsikaline süsteem, milles on 2 punktkeha, mille omavaheline kaugus on 1 meeter: Matemaatilised algmõisted: Füüsikalised algmõisted: meeter, ruumipunkt, punktkeha

$$\begin{cases} \forall(a; \forall(b; \forall(c; f(a; b; c) = f(a; b-1; f(a; b; c-1)))) \\ \forall(a; \forall(b; f(a; b; 2) = f(a; b-1; a)) \\ \forall(a; \forall(c; f(a; 0; c) = (a = c) + c + 1)) \\ \begin{cases} \forall(o; N(o) = 0) = 0 \\ N(a) = N(S(a)) \\ N(a) \wedge N(b) \wedge (F(c) = F(S(c))) = F(b) \end{cases} \end{cases}$$

Seose standard vorm: Seose lahendamiseks tuleb leida „sobivuse” lokaalsed maksimumid. $B_1(n_1, n_2)$ on bijektiivne funktsioon, millel on 2 argumenti. Need argumendid on naturaalarvud. See võrdub mistahes argumentide korral naturaalarvuga. $B_2(q)$ on bijektiivne funktsioon, mille argumendiks on ratsionaalarvud. See võrdub mistahes argumendi korral naturaalarvuga. B_3 on bijektiivne funktsioon, mille argumendiks on füüsikaliste suuruste väärtused. See võrdub mistahes argumendi korral naturaalarvuga. R on seose standardformi funktsioon. See võrdub 1 kui $R(B_1(B_3(S_1); B_3(S_2)))$ S_1 sobib paremini mudeliga kui S_2 . See võrdub 0 kui $R(B_1(B_3(S_1); B_3(S_2)))$ S_2 sobib paremini mudeliga kui S_1 .

Alternatiivslt: R on seose standardformi funktsioon. See võrdub 1 kui $R(B_1(B_3(S_1);B_3(S_2)))$ S_1 sobib paremini mudeliga kui S_2 ja $B_3(S_1) > B_3(S_2)$. See võrdub 0 kui $R(B_1(B_3(S_1);B_3(S_2)))$ S_1 sobib paremini mudeliga kui S_2 ja $B_3(S_1) < B_3(S_2)$. See võrdub 0 kui $R(B_1(B_3(S_1);B_3(S_2)))$ S_2 sobib paremini mudeliga kui S_1 ja $B_3(S_1) > B_3(S_2)$. See võrdub 1 kui $R(B_1(B_3(S_1);B_3(S_2)))$ S_1 sobib paremini mudeliga kui S_2 ja $B_3(S_1) > B_3(S_2)$.

Variant 2 (SEOS AND X)=seos on (SEOS TO X) seosest järeldub X

ilm on ilus: ILM \in ILUSAD asukohas (0;0) on väljatugevus 0: $X \in \text{PUNKTID} \wedge X \in \text{NULISEXGAD} \wedge X \in \text{NULLISEYIKAD} \rightarrow \text{any_x}(0 \in X \rightarrow X \in X)$

Defineerida tehted naturaalarvudele robinsoni aritmeetika või millegi analoogsega ja aksiomeerida, et tehted on lõputult diferentseeritavad. Defineerida tehted naturaalarvudele robinsoni aritmeetika või millegi analoogsega ja panna tehted füüsikaliste suurustega kirja justkui need oleksid naturaalarvud (kuigi ei ole). Valida bijektiivne funktsioon ratsionaalarvudest naturaalarvudesse ja defineerida tehted ratsionaalarvudele neile selle funktsiooni abil vastavate naturaalarvude kaudu. Defineerida tehted selle kaudu, et milline füüsikaline olukord oleks sama kui füüsikaline olukord, mille üks parameetritest on saadud tehte tulemusena. Kahes pakkumises bijektiivse funktsiooni abil mappida kõik ratsionaalarvud (või muu pidev hulk) naturaalarvudeks ja anda „seosele” sisendiks. Mappitud ratsionaalarvude järjekord määrab, et milline neist millist füüsikalist suurust tähistab. Seos tagastab, et kumb pakkumine oli parem.

$q_1 * q_2 = q_3$ on sama mis $f(B(q_1);B(q_2))=B(q_3)$

Nii saab ainult reaalarvulisi suuruseid kirjeldavaid seoseid kasutada.

K1Seose kirjeldamine K1seos on seos hulkade vahel, mille, kirjeldamiseks (kvantorite, kuulumisväidete ja booleanide vaheliste tehete abil) ei ole kvantorite sees kvantoreid (näiteks $\forall x(x \in A) \leftrightarrow \forall x(x \in B)$ on K1seos, aga $\forall x(x \in A \rightarrow \forall y(y \in x \rightarrow y \in y))$ ei ole K1 seos). K1 on seos, mille nestedness on 1. K1seose üheseks kirjeldamiseks peab kirjeldama kõigi erineva sisuga kvantorite ja alghulkade omavaheliste kuulumiste vahelisi seoseid. Selle seose kirjeldamiseks läheb vaja ____ bitti. Kui neid bitte tõlgendada kahendsüsteemis arvuna saab seda seost üheselt kirjeldada naturaalarvuga (valem) ,mis on väiksem kui $(2^{**}(\text{valjalike bittide arv}))$. Kuna erinevate sisudega kvantorid on omavahel sõltuvad, siis piisab kui kirjeldada selliste sisudega kvantorite ja alghulkade kuulumiste vahelist seost, millest saab järeldada mistahes teiste sisudega kvantorite ja alghulkade kuulumiste mistahes booleankombinatsiooni, mis pole sõltumatult kirjeldatavast K1seosest vale. Ükskõik milliste sisudega kvantorid selleks valida, on osad booleankombinatsioonid neist sõltumatult kirjeldatavast K1seosest valed. See järeldub sellest, et nende booleankombinatsioonide arv, mis ei ole kindla väärtusega, ei ole esitatav kujul 2 astmes mingi naturaalarv. Osade sisudega kvantoritest (aga mitte nende eitustest) järeldub ka väiteid alghulkade omavaheliste kuulumiste kohta (näiteks $\forall x(x \in A) \Rightarrow A \in A$), aga alghulkade omavahelistest kuulumistest ei järeldu väiteid kvantorite kohta. Kui mingist kvantorite ja alghulkade vaheliste seoste booleankombinatsioonist järeldub, mingi väide alghulkade kohta ja on samas eitatud, siis on see booleankombinatsioon sõltumatult kirjeldatavast K1seosest vale. Seos ei tohi teha väiteid kõikide hulkade kohta (sest ____). Teisisõnu ei tohi sellest järelduda, et mingite elementidega hulki ei ole olemas. Kui mingite sisudega kvantorite booleankombinatsioon seda teeb, siis on see booleankombinatsiooni sõltumatult kirjeldatavast K1seosest vale. Näiteks väitest $\forall x(A \in x)$ järelduks, et sellist hulka, mis ei sisalda A'd (näiteks tühi hulk) ei ole olemas, seega on see väide vale.

Hea viis seose K1esitamiseks on luua üksühene vastavus võimalike seoste ja naturaalarvude vahele. Hea on kui sellisel kujul esitatud seoste vahel saab rakendada operaatoreid (AND OR NOR) rakendades neid vastavatele arvule bit kaup. Hea on kui tavalises notatsioonis lühidalt kirjapandav seos on kaväikese või lihtsasti kirjeldatava naturaalarvuga vastavusse pandud. Konkreetsem näide Üks võimalus K1seose kirjeldamiseks on teha DNF tabel (see pole suurema algmõistete arvuga seoste puhul mõistlik, sest võtab palju mälu, aga selle eeliseks on see, et siis saab sama algmõistete arvuga K1seoste vahelisi operatsioone teha lõppveeru vastavaid bittide (booleanide) vahel sama operandi rakendades), mille veergudeks on erineva sisuga kvantorid. Aga kuna erineva sisuga kvantorid on omavahel sõltuvad, siis piisab kui tabeli veergudeks võtta (ükskõik kas eitatud või eitamata) kvantorid sisuga, et $P(x)$ pole mingi kindla väärtusega. Ja väited alghulkade omavahelise kuulumise kohta. (täpsemat kirjeldust vaja sh. Veergude järjekorra kohta.) • $\forall x(\neg P(x)=n)$ ehk $\neg \exists (P(x)=n)$ või • $\neg \forall x(\neg P(x)=n)$ ehk $\exists (P(x)=n)$ Isegi sellise veergude valiku korral on need DNF tabeli read, mis väidavad et kõigil hulkadel on mingi omadus, (näiteks $\forall x(x \in x)$, $\forall x(A \in x)$, või $\forall x(x \in x \vee A \in x)$) kindlalt False väärtusega! P on funktsioon, mis loob bijektiivse seose, argumendiks oleva hulga alghulkade sisaldavuse ja alghulkades sisaldumise, ning 2^{N*2+1} väiksemaid naturaalarve sisaldava hulga vahele. Üks võimalus P defineerimiseks on teha DNF tabel, mille veergudeks on (ükskõik kas eitatud või eitamata) väited (veergude järjekord on oluline), et kvanteeritav kuulub mingisse alghulka, et kvanteeritav on iseenda element ja, et mingi alghulk kuulub kvanteeritavasse. (täpsemat kirjeldust vaja sh. veergude järjekorra kohta.) • $P = \sum_{i=0}^{N^2} (2^{**i} * (H_i \in))$ või • $P = \sum_{i=0}^{N^2} (2^{**i} * \neg())$

(veerust tehtav järeldus (vihikus olemas)) (lõppreast tehtav järeldus (vihikus olemas)) (seosest, mis on on esitatav

arvuna tehtav järeldus) Selliselgi juhul on osad lõppread sõltumata kirjeldatavast K1seosest kindlalt valed(vastav boolean peab olema False). Üks võimalus saada minimaalsel vabadusastmel kirjeldus on teha uus numeratsioon lõppridadele, mis jätab need read, mis on sõltumata kirjeldatavast K1seosest valed, vahele. Üheselt valed on need read, mis teevad väiteid kõigi hulkade kohta või on vastuolulised, sest kvantoritega veergudest järeldub alghulkade kohta midagi, mis on vastuolus alghulkade omavahelisi kuulumise kirjeldavate veergudega. Lõppreast tehtav järeldus alghulkade kohta on järmine: (lõppreast tehtav järeldus alghulkade kohta(vihikus olemas)) Kui see on vastuoluline, siis tuleb tuleb vastav lõpprida uues numeratsioonis vahel jätta, sest kvantoritega veergudest järeldub alghulkade kohta midagi, mis on vastuolus alghulkade omavahelisi kuulumise kirjeldavate veergudega.. Järgnev seos näitab, et millised read on vastuolulised: (funktsioon ,mis võtab argumendiks täisarvu (lõpprea numbri) ja tagastab booleani(et kas see on vastuoluline, sest kvantoritega veergudest järeldub alghulkade kohta midagi, mis on vastuolus alghulkade omavahelisi kuulumise kirjeldavate veergudega.))

(funktsioon ,mis võtab argumendiks täisarvu (lõpprea numbri) ja tagastab booleani(et kas see on vastuoluline, sest teeb kõigi hulkade kohta väite.)(vihikus olemas))

(funktsioon ,mis võtab argumendiks täisarvu (lõpprea numbri) ja tagastab booleani(et kas see on vastuoluline))
(seosest, mis on on esitatud naturaalarvuna tehtav järeldus, kus on eeldatud uut lõppveergude numeratsiooni.)

Sisemisest funktsioonist

mõte1

mõte: **Tundub edukas JÄTKATA** 1. kirjeldada meetod kuidas reabooleanid sisendist määratakse
2. Millised read on sellest järelduvalt väite argumendst sõltumatult valed.

teostus: 1. kvantorite sisud alates kõige sisemisest seosfunktsioonide väärtuste vahelisteks booleanfunktsioonideks.
2.rekusiivselt (alates kõige sisemisest kvantorist) asendab kõik kvantoriid veergude vahelise booleanfunktsiooniga. Selleks asendatakse kvantori sisusse veerule vastavad kvanteeritavate vaheliste väidete väärtused, et määrata tingimus, mil antud veerg on jaatud/eitatud. iga veeru jaoks asendatakse kvantori sisusse veerus olevad väited kõigis erinevates järjekordades.(st. kõik võimalused kus sisu kvanteeritavate asemele pannakse veerus olevad kvanteeritavad. Ühte veeru kvanteeritavat võib ka mitme asja asemele panna.).

3. Kui mingi välimiste kvanteeritavate väärtuste puhul teeb kvantor madalama taseme väite, siis...

mõte2

mõte: Millised täpselt on veergude vahelised seosed tasemetga variandis?

teostus: veerg on väite argumendst sõltumatult vale kui see sisaldab seos-funktsioonide väärtuste booleankombinatsiooni, mis on eelmises tasemes keelatud.

25 algoritm

algoritmi pseudokoodis.

25.1 versioon2:

1. leiab millisei predikaate on väites kasutatud ja nummerdab need.
2. leiab kui mitme kvantori see on kõige rohkemate kvantorite sees olev väide (maksimaalne K väärtus).
3. rekusiivselt: asendab iga eksistentsiaalkvantori funktsiooniga f, mis iga arvu puhul tagastab, et kas see arv võib olla plato numbriks. f(n) parajasti siis kui n võib olla plato number. Funktsiooni f loob programm kasutades kvantori kvanteeritava elementaarseoste madalama taseme kvanteeritavatega(tingimus plato numbri h osale) ja kvantori sees olevate platode(tingimus harumärkidele) vahelist booleanfunktsiooni. Kui kasutada on ainult NAND ja eksistentsiaalkvantor (universaalsuskvantori ja teised loogikatehted on predikaatidena esitatud) võib

funktsiooni f leidmise algoritm üsna lihtne olla.

Osad lihtsustused (ÜV plato numbrite leidmine, mitte loogika lihtsustused) võib juba funktsiooni f leidmisel teha.

4. lõpuks saadud funktsioon f lõpprea kohta võimaldabki lõppveeru määrata.
5. Muutes funktsioone h ja harumärk konvertida väide kujule, milles ühegi numbriga lõpprida ei jaata ühelt valesid lõppridu, ükski väide ei ole esitatav malama K 'ga ja pole ühtki predikaati, mille kohta ei ole mingeid väiteid tehtud.

25.2 versioon5:

1. asendab **sisend**is kõik universaalsuskvantorid eksistentsiaalsuskvantoriga kasutades seost: $\forall_x(f(x)) = \neg \exists_x(\neg A(x))$.
2. loob puugraafi lõppveerg-tüüpi tipu **LV**.
3. parsib **sisend**'it rekursiivselt funktsiooniga **moodusta_puu**. Selle tagastatud puugraafi tippude järjendi iga elemendi tipu **LV** haruks. ehk **LV.harud:=moodusta_puu(sisend)**.
4. iga **LV** haru **LR** korral:
5. : iga **LR**ile rekursiivselt (ei pruugi otseselt alluda) alluva tipu **tipp** korral:
6. : : iga funktsioonikutse **laiendatud_EKd(tipp)** poolt tagastatud järjendis oleva **tipp**'u laiendatud alluva **laiendatud_EK** korral:
7. kui laiendatud_EKsse tuleb eitatud kvantor poolikult(kõrgemad oksad puudu), siis see tugvendab eitatud_EKd, ja muudab algoritmi valeks.
8. : : : iga **tipp**'u eitatud haru **eitatud_EK** korral:
9. : : : :kui funktsioonikutse **sisu_täpsem_kui(laiendatud_EK, eitatud_EK)** tagastab, et **laiendatud_EK** sisu on täpsem kui **eitatud_EK** sisu (ja need on kooskõlas) ehk, et **laiendatud_EK** sisust järeldeb **eitatud_EK** sisu ehk kui:
10. : : : :kustutab **LVst LRi**.
11. : : : :läheb punkti 4 ja võtab uue **LRi**.
12. : : : :kui funktsioonikutse **sisud_kooskõlas(laiendatud_EK, eitatud_EK)** tagastab, et **laiendatud_EK** sisu ja **eitatud_EK** sisu on kooskõlas: (seda punkti pole vaja kui hiljem nagunii hiljem nagunii need **LRid**, millest täpsem variant olemas on, kustutatakse. Jääb ainult alles ainult see uus **LR**, millesse lisati see **eitatud_EK_omadus**, mis seal juba olemas oli)
: : : : (sel juhul on kindlasti **eitatud_EK** sisu täpsem kui **laiendatud_EK** sisu või **eitatud_EK** sisu on täpsuselt võrreldamatu **laiendatud_EK** sisuga)
13. : : : :lisab **tippu laiendatud_EK**
14. : : : :iga **eitatud_EK** haru ja predikaatväite **eitatud_EK_omadus** kohta:
15. : : : : :teeb **LRist** koopia **uus_LR**.
16. : : : : :lisab **uus_LR**is **laiendatud_EK**'sse **eitatud_EK_omadus**'e, aga vastasmärgiga kui see **eitatud_EK**'s oli. Peale seda enam **laiendatud_EK** ja **eitatud_EK** sisud kooskõlas ei ole.
17. : : : : :kui **tippus** pole **laiendatud_EK**st täpsema sisuga jaatatud haru:
18. : : : : : :lisab **uus_LRi LV**sse.
19. : kui **LRi** ühegi **tippu** ühegi **laiendatud_EK** sisu ei olnud ühegi **eitatud_EK** sisuga kooskõlas:
20. : : siis tagastab, et **sisend** ei ole ÜV.

21. : kustutab **LV**st **LR**i.
22. kui **LV**s enam ühtegi haru ei ole, siis tagastab, et **sisend on** ÜV.

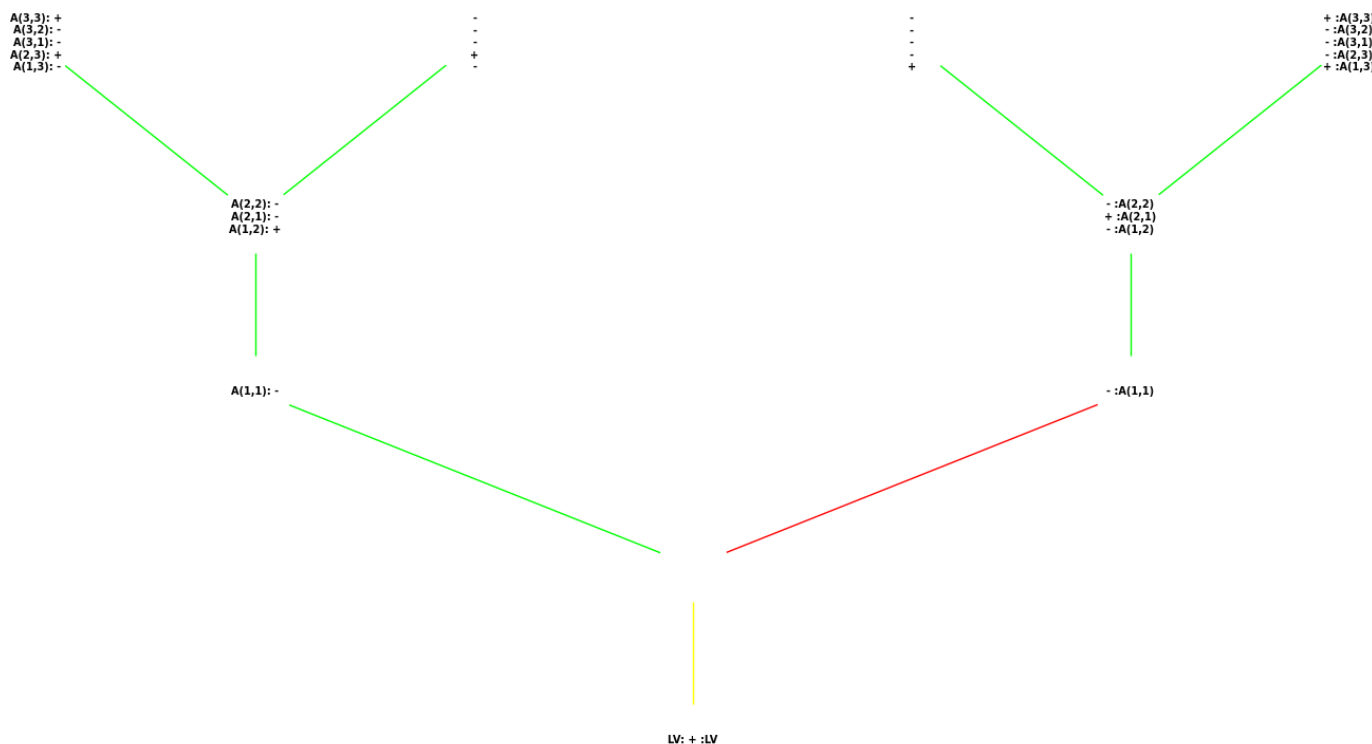
25.2.1 funktsioon moodusta_puu:

1. kui kutsutakse välja funktsioon **moodusta_puu** argumentidega **sisend**, ehk **moodusta_puu(sisend)**, siis:
Loob puugraafi tipu **tipp**.
2. viib argumentideks oleva väite **sisend** mitte-täielikku DNF-vormi **DNF_vorm**. Näiteks **sisend** $sisend = ((\neg R_1 \vee R_2) \text{NOR} (R_1 \wedge R_3)) \text{XOR} (R_2 \vee R_4)$ puhul peale seda sammu $sisend = R_2 \vee (R_3 \wedge R_4) \vee (R_4 \wedge \neg R_1) \vee (R_1 \wedge \neg R_3 \wedge \neg R_4)$
3. muudab **DNF_vorm**'i järjendiks eraldades selles omavahel võitatud olevad asjad. Näiteks **DNF_vorm**'i $DNF_vorm = E(A(x_4, x_1) \wedge E(A(x_5, x_1))) \vee \neg E(A(x_2, x_5)) \wedge A(x_4, x_3) \vee \neg E(B(x_1, x_2, x_5)) \wedge \neg E(A(x_4, x_5)) \wedge \neg A(x_4, x_2))$ puhul peale seda sammu $DNF_vorm = [A(x_4, x_1) \wedge E(A(x_5, x_1)), \neg E(A(x_2, x_5)) \wedge A(x_4, x_3), \neg E(B(x_1, x_2, x_5)) \wedge \neg E(A(x_4, x_5)) \wedge \neg A(x_4, x_2))]$
4. Iga järjendis **DNF_vorm** oleva elemendi **võitatud_asi** korral:
5. : muudab **võitatud_asi** järjendiks eraldades selles omavahel jaatatud olevad asjad. Näiteks **võitatud_asi**'i $voitatud_asi = A(x_4, x_1) \wedge E(A(x_5, x_1)) \wedge \neg E(B(x_1, x_2, x_5)) \wedge \neg E(A(x_4, x_5)) \wedge \neg A(x_4, x_2))$ puhul peale seda sammu $voitatud_asi = [A(x_4, x_1), E(A(x_5, x_1)), \neg E(B(x_1, x_2, x_5)), \neg E(A(x_4, x_5)), \neg A(x_4, x_2)]$
6. : iga **võitatud_asi**'s oleva asja **jaatatud_asi** korral:
7. : : kui **jaatatud_asi** on eitamata eksistentsiaalsuskvantor:
8. : : : muudab **jaatatud_asi**'i selleks järjendiks, mille **moodusta_puu** tagastab, kui selle argumentideks anda **jaatatud_asi** sisu (eksistentsiaalsuskvantori sees olev väide), ehk **jaatatud_asi:=moodusta_puu(jaatatud_asi.sisu)**.
9. : : : iga **jaatatud_asi** sees oleva **jupp**'i korral:
10. : : : : paneb **juppi** ümber eksistentsiaalsuskvantori.
11. : : kui **jaatatud_asi** on eitatud eksistentsiaalsuskvantor (eksistentsiaalsuskvantor, mille ees on eitus):
12. : : : muudab **jaatatud_asi2**'i selleks järjendiks, mille **moodusta_puu** tagastab, kui selle argumentideks anda **jaatatud_asi** sisu (eksistentsiaalsuskvantori sees olev väide), ehk **jaatatud_asi:=moodusta_puu(jaatatud_asi.sisu)**.
13. : : : muudab **jaatatud_asi** üheselt tõseks väiteks.
14. : : : iga **jaatatud_asi2** sees oleva **jupp**'i korral:
15. : : : : paneb **juppi** ümber eksistentsiaalsuskvantori ehk **jupp:=E(jupp)**
16. : : : : muudab **jaatatud_asi**'i jaatades sellega **juppi** ehk **jaatatud_asi&=~jupp**
17. : : : muudab **jaatatud_asi**'i järjendiks, mis sisaldab ainult seda, mis ta selle sammu alguses on.
18. : lisab **tippule** aluuvateks kõikvõimalikud erinevad väited, mille saab moodustada kui jaatada kokku üks suvaline element igast **võitatud_asi** elemendi elemendist. Näiteks **võitatud_asi**'i $voitatud_asi = [[[R_1, R_2], [R_3], [R_4, R_5, R_6]]$ puhul peale seda sammu on **tipp**'ule lisatud järgnevad alluvad: $lisatudalluvad = [R_1 \wedge R_3 \wedge R_4, R_1 \wedge R_3 \wedge R_5, R_1 \wedge R_3 \wedge R_6, R_2 \wedge R_3 \wedge R_4, R_2 \wedge R_3 \wedge R_5, R_2 \wedge R_3 \wedge R_6]$
19. tagastatakse **tipp**.

25.2.2 seletus2 funktsioon moodusta_puu:

1. viib kõige sisemiste kvantorite sisud DNF vormi (igas jaatuses on kõik $p(k)$ elemendid kas võiatud või jaatatud.).
2. viib kõige sisemiste kvantorite sisust seosefunktsioonide väited kõige välimiste võimalike kvantorite sisse. Ehk võimalikest madalamiale tasemele.
3. nüüd sisaldavad tasemel k olevad kvantorid ainult $p(k)$ elemente.
4. toob kõige sisemiste kvantorite sisesed võitused kvantorite seest välja. Kasutades reeglit $\exists(f_1(x) \vee f_2(x)) = \exists(f_1(x)) \vee \exists(f_2(x))$.
5. läheb tagasi punkti 1.

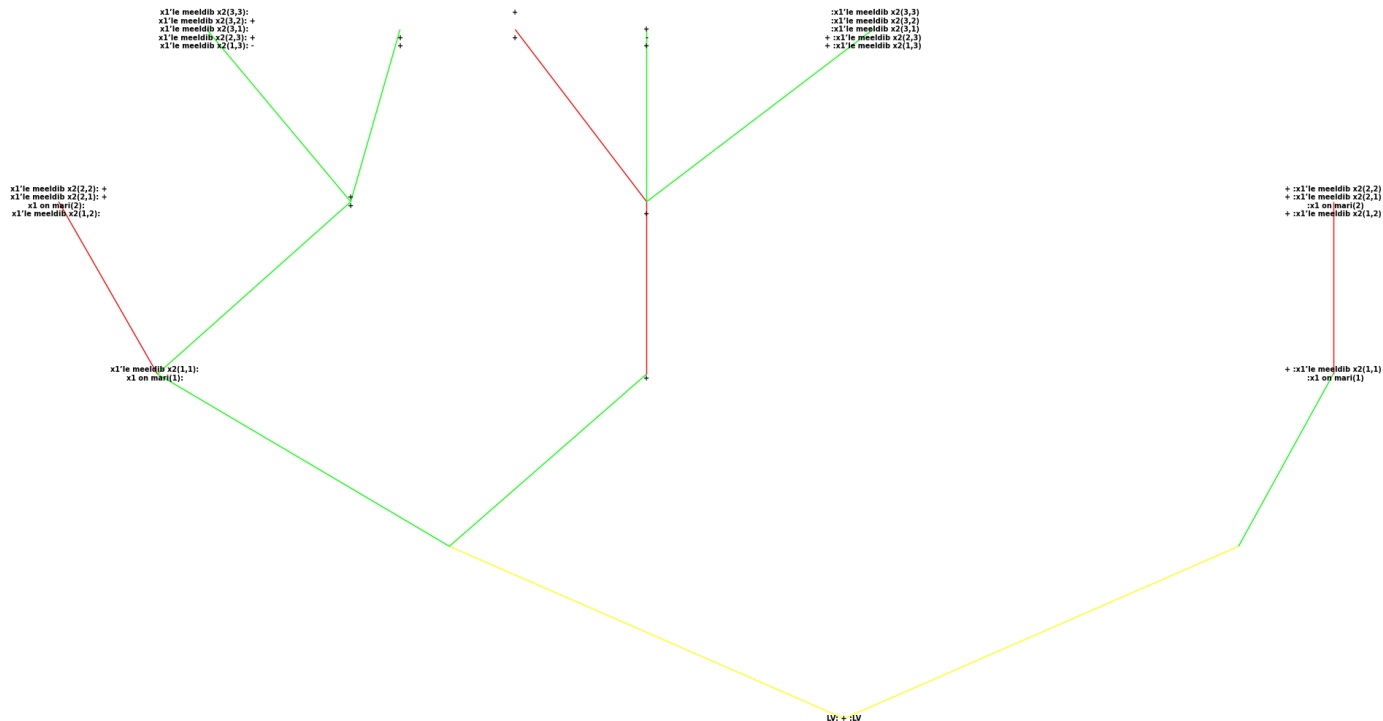
Pilt väite $\exists(\neg A(1,1) \wedge \exists(A(1,2) \wedge \neg A(2,1) \wedge \neg A(2,2) \wedge \exists(\neg A(1,3) \wedge A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge \neg A(3,3))) \wedge \exists(\neg A(1,3) \wedge A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge A(3,3))) \wedge \neg \exists(\neg A(1,1) \wedge \exists(\neg A(1,2) \wedge A(2,1) \wedge \neg A(2,2) \wedge \exists(A(1,3) \wedge \neg A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge \neg A(3,3))) \wedge \exists(A(1,3) \wedge \neg A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge A(3,3)))$ põhjal koostatud puust:



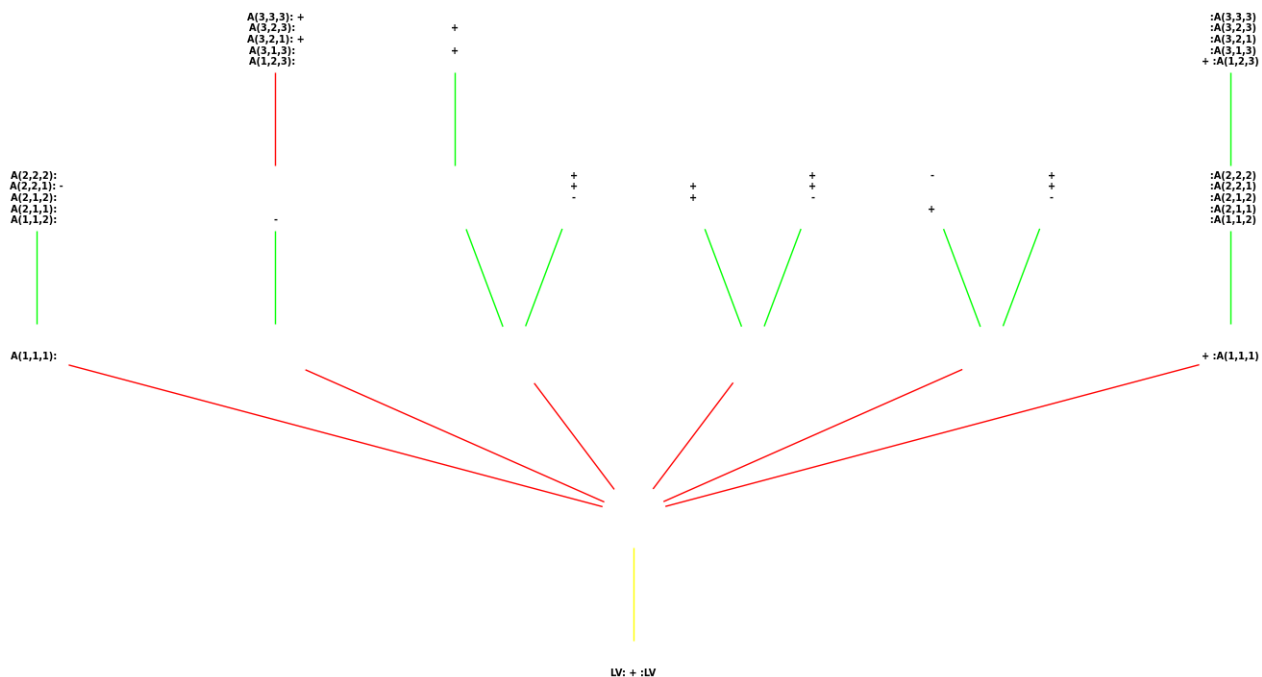
pilt väite [kõigil, kes Marile meeldivad, [ei ole kedagi kes Marile meeldiks, aga temale mitte ja kellele meeldiks Mari] või [ei ole kedagi, kes nii talle kui Marile meeldiks] või [on keegi, kes neile meeldib ja kes iseendale meeldib]] ja ja leidub keegi, kes meeldib marile, nii, et eksisteerib [keegi, kes meeldib nii marile kui talle] ja [keegi, kes meeldib marile, kellele mari meeldib ja kelle ei meeldi talle] ja ei leidu kedagi, kes nii iseendale kui ka talle ta meeldiks või [on keegi(x_1) kes endale meeldib, aga [kellel pole kedagi kes nii talle kui iseendale meeldiks ja kellele meeldiks tema(x_1)] ehk (operaatori süntaksis):

$\exists((1 \text{ on mari}) \wedge \neg \exists((1 \text{ le meeldib } 2) \wedge \exists((1 \text{ le meeldib } 3) \wedge \neg(2 \text{ le meeldib } 3) \wedge (3 \text{ le meeldib } 1)) \wedge \exists((1 \text{ le meeldib } 3) \wedge (2 \text{ le meeldib } 3)) \wedge \neg \exists((2 \text{ le meeldib } 3) \wedge (3 \text{ le meeldib } 3))) \wedge \exists(\exists((2 \text{ on mari}) \wedge (2 \text{ le meeldib } 1) \wedge \exists((1 \text{ le meeldib } 2) \wedge (2 \text{ le meeldib } 3)) \wedge \exists((2 \text{ le meeldib } 3) \wedge (3 \text{ le meeldib } 2) \wedge \neg(1 \text{ le meeldib } 3))) \wedge \neg \exists((2 \text{ le meeldib } 1) \wedge (2 \text{ le meeldib } 2))) | \exists((1 \text{ le meeldib } 1) \wedge \neg \exists((1 \text{ le meeldib } 2) \wedge (2 \text{ le meeldib } 2) \wedge (2 \text{ le meeldib } 1)))$

põhjal koostatud puust:



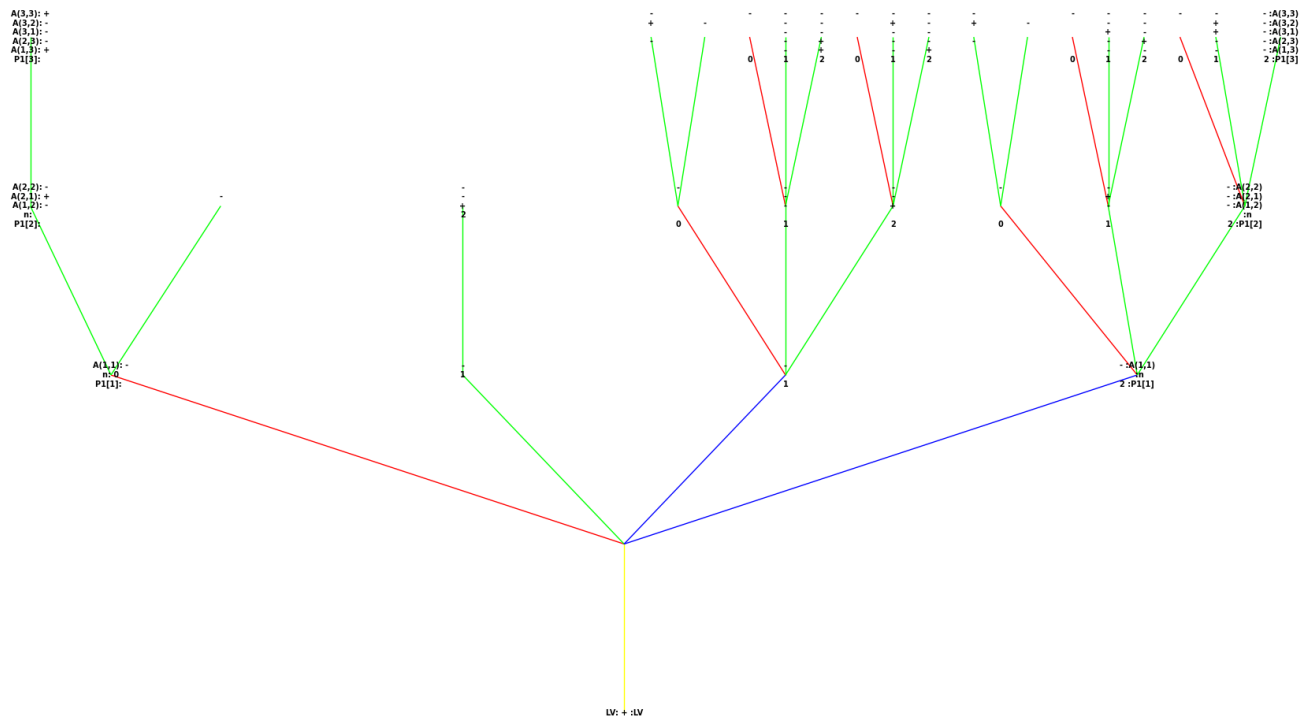
pilt väite $\neg \exists(\exists(A(1,2,3))) \wedge A(1,1,1) \vee \exists(\neg A(2,2,1) \vee \neg A(1,1,2) \wedge \neg \exists(A(3,3,3) \wedge A(3,2,1))) \vee \exists(A(2,2,2) \wedge \neg A(2,1,2) \wedge A(2,2,1)) \wedge \exists(A(2,1,2) \wedge A(2,2,1) \vee \neg A(2,2,2) \wedge A(2,1,1) \vee \exists(A(3,2,3) \wedge A(3,1,3)))$ põhjal koostatud puust:



25.2.3 funktsioon laiendatud_EKd:

1. kui kutsutakse välja funktsioon **laiendatud_EKd** argumentidega **tipp** ja **kõrgus**, ehk **laiendatud_EKd(tipp,kõrgus)**, siis loob tipud **pseudo_LR1**, **pseudo_LR2**
2. iga võimaliku ainult **tipp**'u alluvaid sisaldava järjendi, milles on **kõrgus** elementi, **P1** korral:
3. : lisab **P1** algusesse elemendi **pseudo_LR1**
4. : eemaldab **P1**'est kõik tippudest, mis puus on oma ülemuse sees olnud eitatud, suuremal indeksitel olevad tipud.
5. : loob uue järjendi nimega **laiendatud**
6. : märgib iga järjendi **laiendatud** elementide vahelistks predikaatväideteks olema samad predikaatväited, mis on samade indeksitega **P1** elementide vahel. Predikaatväidete indeksid võivad teistsugused olla kui **P1**'es olevates tippudes, sest nagu tippude kõrgusedki on **P1**'es tippude kvanteeritavate nimed teistsugused kui sisendiks antud **P1**'es. Kui **P1**'es ei ole mingid 2 **tipp**'ust kumbki teisele alluvaks, siis puus ei ole nende vahelisi predikaatväiteid ja neid ei saa olla ka **P1**'es ega järjendis **laiendatud**.
7. : mergeb **laiendatud**'e esimese elemendi ja **pseudo_LR2**'e kokku moodustades neist puu. ehk **pseudo_LR2:=merge(P1, laiendatud)**
8. iga võimaliku ainult **tipp**'u alluvaid sisaldava järjendi mille viimane element on puus eitatud ja, milles on vähem kui **kõrgus** elementi, **P1** korral:**LR1**

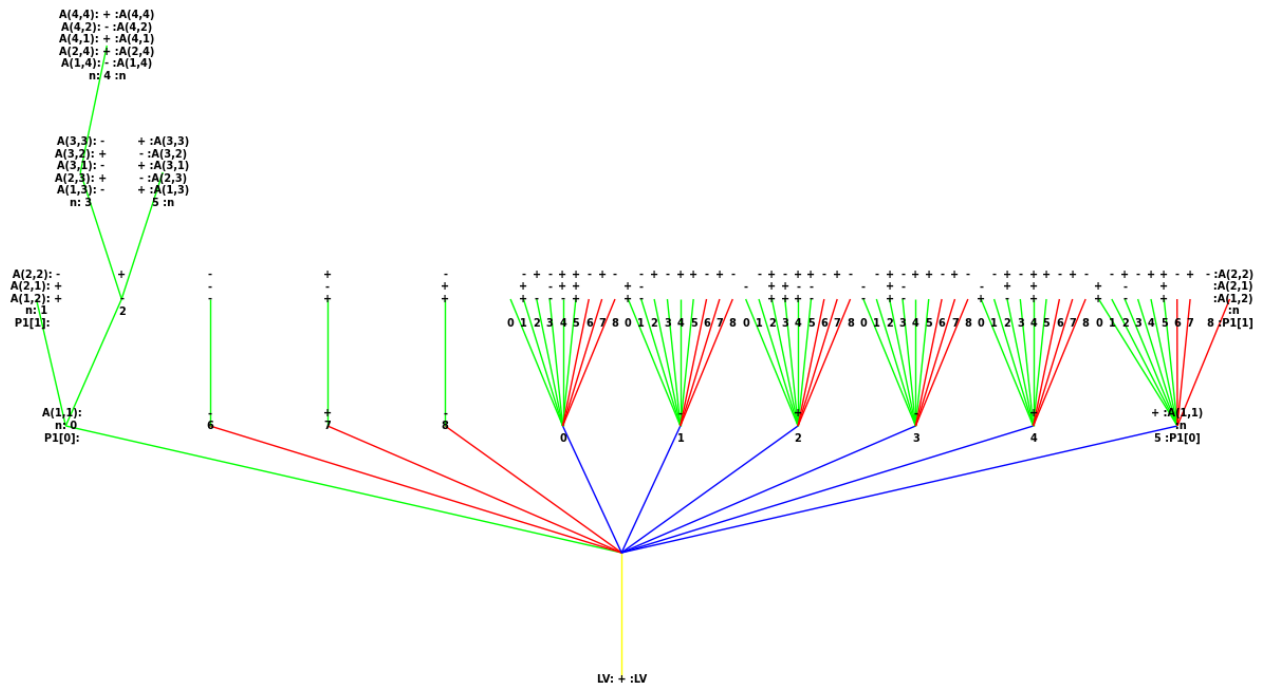
pilt väite $E(\neg A(1,1) \wedge E(A(1,2) \wedge \neg A(2,1) \wedge \neg A(2,2))) \wedge \neg E(E(\neg A(2,1)) \wedge \neg A(1,1) \wedge E(\neg A(1,2) \wedge A(2,1) \wedge \neg A(2,2) \wedge E(A(1,3) \wedge \neg A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge A(3,3))))$ põhjal koostatud puus, kuhu on lisatud laiendatud harud(sinist värvi):



pilt väite, mis on ÜV ($P1=[1,3]$), $\neg \exists (\neg A(1,1) \wedge \exists (\neg A(2,2) \wedge A(1,2) \wedge A(2,1))) \wedge \neg \exists (A(1,1) \wedge \exists (A(2,2) \wedge A(1,2) \wedge \neg A(2,1))) \wedge \neg \exists (\neg A(1,1) \wedge \exists (\neg A(2,2) \wedge \neg A(1,2) \wedge \neg A(2,1))) \wedge \exists (\exists (\neg A(2,2) \wedge A(1,2) \wedge A(2,1)) \wedge \exists (A(2,2) \wedge \neg A(1,2) \wedge \neg A(2,1)))$

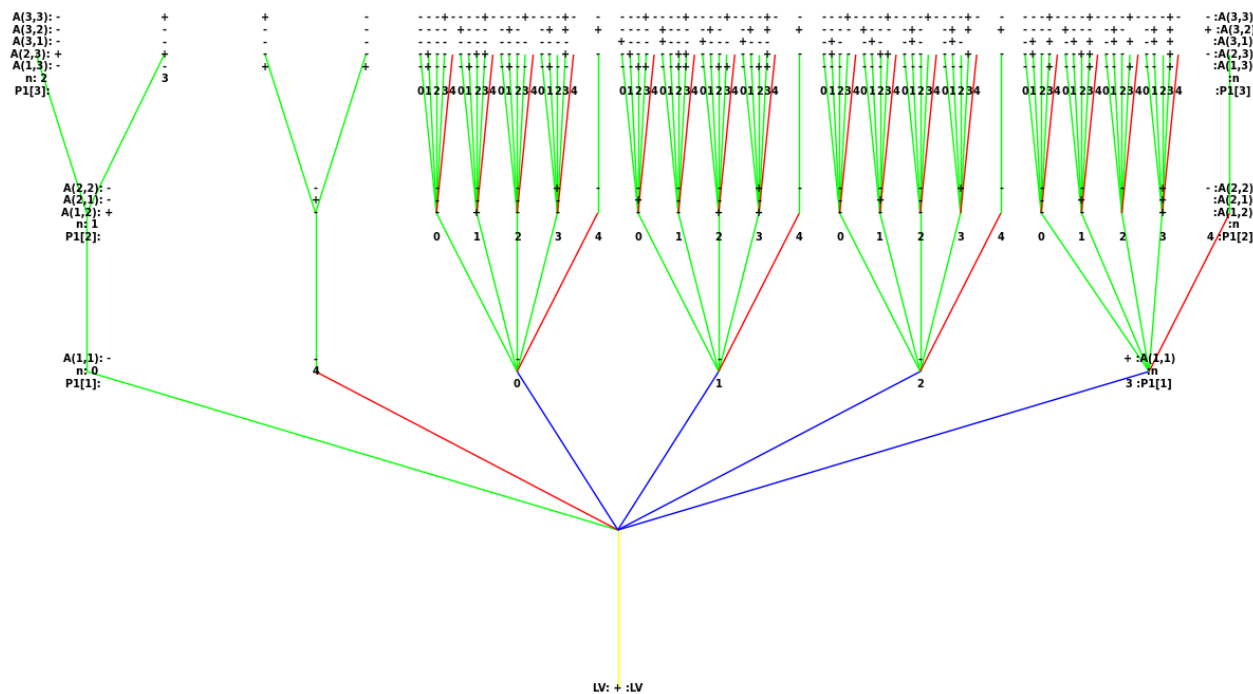
$\exists(A(3,3) \wedge \neg A(3,2) \wedge \neg A(2,3) \wedge A(1,3) \wedge A(3,1)) \wedge \exists(\neg A(3,3) \wedge A(3,2) \wedge A(2,3) \wedge \neg A(1,3) \wedge \neg A(3,1) \wedge \exists(A(4,4) \wedge A(2,4) \wedge \neg A(4,2) \wedge \neg A(1,4) \wedge A(4,1))))$

põhjal koostatud puus, kuhu on lisatud laiendatud harud(sinist värvi):

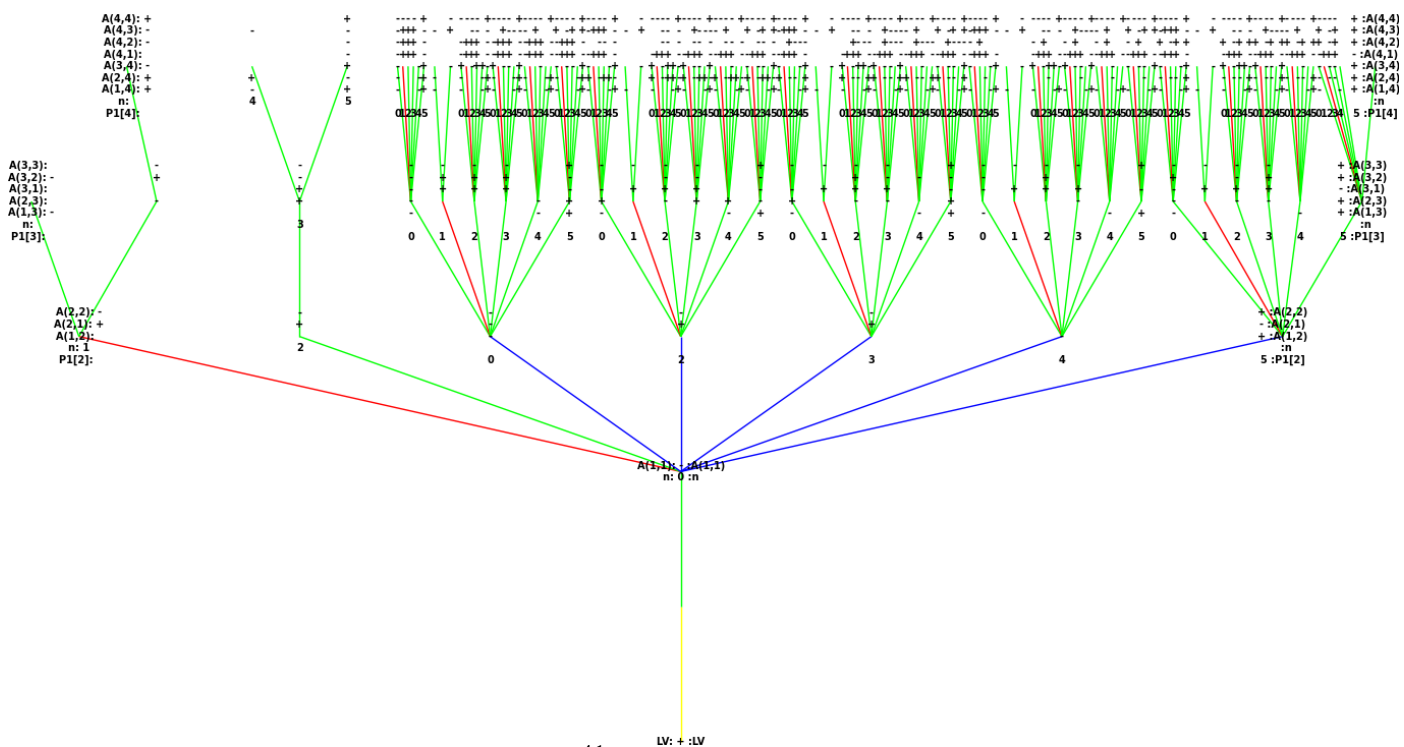


pilt väite $E(\neg A(1,1) \wedge \exists(A(1,2) \wedge \neg A(2,1) \wedge \neg A(2,2) \wedge \exists(\neg A(1,3) \wedge A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge \neg A(3,3)) \wedge \exists(\neg A(1,3) \wedge A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge A(3,3)))) \wedge \neg \exists(\neg A(1,1) \wedge \exists(\neg A(1,2) \wedge A(2,1) \wedge \neg A(2,2) \wedge \exists(A(1,3) \wedge \neg A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge \neg A(3,3)) \wedge \exists(A(1,3) \wedge \neg A(2,3) \wedge \neg A(3,1) \wedge \neg A(3,2) \wedge A(3,3))))$

põhjal koostatud puus, kuhu on lisatud laiendatud harud(sinist värvi):



pilt väite(mis on ÜV) $\exists(\neg A(1,1) \wedge \neg \exists(\exists(\neg A(3,2) \wedge \neg A(1,3)) \wedge \neg A(2,2) \wedge A(2,1) \wedge \exists(\neg A(2,3) \wedge A(3,2) \wedge \neg A(3,3) \wedge \exists(A(1,4) \wedge A(2,4) \wedge \neg A(3,4) \wedge \neg A(4,2) \wedge \neg A(4,3) \wedge A(4,4)))) \wedge \exists(\neg A(2,2) \wedge A(2,1) \wedge \exists(\neg A(3,3) \wedge A(3,1) \wedge A(2,3) \wedge \neg A(3,2) \wedge \exists(\neg A(4,3) \wedge \neg A(1,4) \wedge A(2,4)) \wedge \exists(A(4,4) \wedge \neg A(4,2) \wedge \neg A(2,4) \wedge \neg A(4,3) \wedge A(3,4) \wedge \neg A(2,4) \wedge A(1,4) \wedge \neg A(4,1))))$ põhjal koostatud puus, kuhu on lisatud laiendatud harud(sinist värvi):



25.2.4 funktsioon `sisu_täpsem_kui`

Tagastab, et kas 1. argumentiks oleva tipu sisust järelneb 2. argumentiks oleva tipu sisu. $tipp1.sisu \rightarrow tipp2.sisu$

kui kutsutakse välja funktsioon `sisu_täpsem_kui` argumentidega `tipp1` ja `tipp2`, ehk `sisu_täpsem_kui(tipp1,tipp2)`, siis:

1. iga **tipp2** sees oleva predikaatväite **pv2** kohta:
2. : iga **tipp1** sees oleva predikaatväite **pv1** kohta:
3. :: kui ja **pv1==pv2**:
4. :: :läheb punkti 1 ja võtab uue **pv2**'e
5. : tagastab False ehk, et 1. argumentiks oleva tipu **tipp1** sisust ei järelne 2. argumentiks oleva tipu **tipp2** sisu (ükski **pv1** ei olnud sama mis **pv2**)
6. iga **tipp2** sees oleva eitatud eksistentsiaalsuskvantori **eitatud_EK2** kohta:
7. : kui **eitatud_EK2**'el on harusid:
8. :: tagastab "võrreldamatu"
9. : iga **tipp1** sees oleva eitatud eksistentsiaalsuskvantori **eitatud_EK1** kohta:
10. :: kui **eitatud_EK1**'el on harusid:
11. :: : läheb punkti 9 võtab uue **eitatud_EK1**'e
12. :: kui `sisu_täpsem_kui(eitatud_EK2,eitatud_EK1)`:
13. :: : läheb punkti 6 ja võtab uue **eitatud_EK2**'e
14. :: tagastab False ehk, et 1. argumentiks oleva tipu **tipp1** sisust ei järelne 2. argumentiks oleva tipu **tipp2** sisu
15. iga **tipp2** sees oleva jaatatud eksistentsiaalsuskvantori **EK2** kohta:
16. : iga **laiendatud_EKd(tipp1)** sees oleva eksistentsiaalsuskvantori **EK1** kohta:
17. :: kui `sisu_täpsem_kui(EK1,EK2)`:
18. :: :läheb punkti 11 ja võtab uue **EK2**'e
19. :: tagastab False ehk, et 1. argumentiks oleva tipu **tipp1** sisust ei järelne 2. argumentiks oleva tipu **tipp2** sisu
20. tagastab True ehk, et 1. argumentiks oleva tipu **tipp1** sisust järelneb 2. argumentiks oleva tipu **tipp2** sisu

25.2.5 funktsioon `sisud_kooskõlas`

tagastab, et kas 1. argumentiks oleva tipu sisust järelneb 2. argumentiks oleva tipu sisu. $tipp1.sisu \rightarrow tipp2.sisu$

kui kutsutakse välja funktsioon `sisud_kooskõlas` argumentidega `tipp1` ja `tipp2`, ehk `sisud_kooskõlas(tipp1,tipp2)`, siis:

1. iga **tipp2** sees oleva predikaatväite **pv2** kohta:
2. : iga **tipp1** sees oleva predikaatväite **pv1** kohta:
3. :: kui **pv1==¬ pv2**:
4. :: : tagastab False ehk, et 1. argumentiks oleva tipu **tipp1** sisu ja 2. argumentiks oleva tipu **tipp2** sisud ei ole kooskõlas.
5. iga **tipp2** sees oleva eitatud eksistentsiaalsuskvantori **eitatud_EK2** kohta:

6. : iga **tipp1** sees oleva jaatatud eksistentsiaalsuskvantori **EK1** kohta:
7. : : kui **sisu_täpsem_kui(EK1,eitatud_EK2)**:
8. : : : tagastab False ehk, et 1. argumentiks oleva tipu **tipp1** sisu ja 2. argumentiks oleva tipu **tipp2** sisud ei ole kooskõlas.
9. iga **tipp2** sees oleva jaatatud eksistentsiaalsuskvantori **EK2** kohta:
10. : iga **tipp1** sees oleva eitatud eksistentsiaalsuskvantori **eitatud_EK1** kohta:
11. : : kui **sisu_täpsem_kui(EK2,eitatud_EK1)**:
12. : : : tagastab False ehk, et 1. argumentiks oleva tipu **tipp1** sisu ja 2. argumentiks oleva tipu **tipp2** sisud ei ole kooskõlas.
13. tagastab True ehk, et 1. argumentiks oleva tipu **tipp1** sisu ja 2. argumentiks oleva tipu **tipp2** sisud on kooskõlas.

25.2.6 funktsioon **lihtsusta_puud**

1. kui kutsutakse välja funktsioon **lihtsusta_puud** argumentiga **LV**, ehk **lihtsusta_puud(LV)**, siis:
lisab kõigile **LV**ga kaudselt ühenduses olevatele tippude, mille kvanteeritavate vahel on mingi predikaatväide kirjas, suunata servad(eelnavad servad olid suunatud juurtipust **LV** eemale.).
2. tagastab **lihtsusta_tipp(LV)**

25.2.7 funktsioon **lihtsusta_tipp**

1. kui kutsutakse välja funktsioon **lihtsusta_tipp** argumentiga **tipp1**, ehk **lihtsusta_tipp(tipp1)**, siis:
iga **tipp1**e haru **tipp2** puhul:
2. : koostab **tipp1**e põhjal **abi_tipp1**e, kus ,mis on muidu nagu **tipp1**, aga on lisatud servad kõikide tippude vahele, mille kvanteeritavate vahel on mingi predikaatväide kirjas.
3. : võtab uue tipu **uus_tipp1**
4. : lisab **uus_tipp1** harudeks kõik **abi_LR** sidususkomponendid
5. : leiab funktsiooni kutse **lühimapikimateegatipp(abi_LR)** abil tipu **tipp**
6. : teeb **tippu** uue puugraafi **uusLR** juurtipuks.
7. : eemaldab tippu **abi_LR**ist