väidete algoritmilise analüüsimise "projekti" eesmärgid

Olger Männik

March 16, 2022

Part I
Sissejuhatus

Selles failis on kirjeldatud ainult projekti eesmärgid. Mul on rohkem faile, kus on eesmärkide saavutamist alustatud ja sellega üsna kaugele jõutud.

sõna "väide" lahtiseletus: 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 elementaarseoseid 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 2). Seda saaks kasutada näiteks kõige üldisemate matemaatiliste probeemide (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 väitnus, 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 kindalsti 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 estitatud 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 kirjapandav 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 kirjapanemiseks.
4. lisada pogrammile funktsionaalsus, väitedete 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 standardsel 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 tegevat 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 ülesanded. 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 2.3.1 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 2.3.1 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 2.3.1 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 intuitiivselt 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(eksisteerib 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 intuitiivselt, ilma arvuti abita kontrollida, aga pikkade ja keerukate väidete vastuolulisust on niimoodi väga raske kontrollida. Selliste valemita 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 intuitiivselt 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:$\exists(A(x_1,x_1)\wedge \exists(A(x_2,x_2)\wedge A(x_2,x_1)\wedge A(x_1,x_2))\wedge\exists(\neg A(x_2,x_2)\wedge\neg A(x_1,x_2)\wedge A(x_2,x_1)\wedge \neg\exists(\neg A(x_3,x_3)\wedge\neg A(x_3,x_2)\wedge A(x_3,x_1)\wedge A(x_2,x_3)\wedge A(x_1,x_3))))\wedge\neg\exists(\exists(\exists(\neg A(x_1,x_1)\wedge \neg A(x_1,x_2)\wedge\neg A(x_2,x_1)\wedge\neg A(x_2,x_2)\wedge\neg A(x_3,x_1)\wedge\neg A(x_3,x_2)\wedge\neg A(x_3,x_3)\wedge \neg A(x_1,x_3)\wedge\neg A(x_2,x_3)))\vee A(x_1,x_1)\wedge\exists(A(x_2,x_2)\wedge\neg A(x_2,x_1)\wedge A(x_1,x_2)\neg\exists(\neg A(x_3,x_3)\wedge A(x_3,x_2)\neg\wedge A(x_3,x_1)\wedge A(x_2,x_3)\wedge A(x_1,x_3))))$

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:

- $STANDARD \wedge \exists(\exists({x_1}^3 - 4 * {x_1}^2 + 6 * x_1 - 24 - {x_2}^4 + 3 * x_2 = 0 \wedge x_2 > 1))$
- $STANDARD \wedge \exists(on_reaalarv(x_1) \wedge \neg\exists(x_1 < x_2 \wedge on_reaalarv(x_2)))$eksiseerib 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) järeldub 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 2.3.1 kirjeldatud syntaxsugarit kasutamata $\neg(STANDARD \rightarrow \neg\exists_{x_1}(on\_naturaalarv(x_1) \wedge \exists_{x_2}(on\_naturaalarv(x_2) \wedge \exists_{x_3}(on\_naturaalarv(x_3) \wedge \exists_{x_4}(x_1^{x_4} + x_2^{x_4} = x_3^{x_4} \wedge x_4 > 2 \wedge on\_naturaalarv(x_4)))))))$

2 süntax

2.1 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.

2.2 eraldi fail

2.2.1 predikaadid

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

operaatorisyntax eraldi lõigud, mille vahel on sulgude sees argumendid. 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 argumendi jaoks on tühjad sulud. juhul kui see predikaat on jutumärkides on tühjade sulgude asemel argumendikohtades 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 argumendi kohtade kõrval tähistada tagurpidi kaldkriipsuga(nagu pythoni syntaxis.).

Kõik mis pole loogikatehted, sulud ega kvantorid tõlgendada predikaatidena(2 argumendilise 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 järeldusmärkidlja ekvivalentsusmärkidel eraldi tähendus. {} optimeerimis tingimused.

Eraldi funktsooni sees saab sisestada käske.

2.3 syntaxsugar

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

2.3.1 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 kasutavasse notatsiooni ümber kirjutatada tuleb kvantoritele kvanteeritavate nimed (alaindeksitena) juurde kirjutada.

Näiteks järgnevas valemis, kus pole kvanteeritavate nimesi kirjutatud:
$\exists(\exists(\exists(A(x_1,x_2,x_3,x_2))))\wedge\exists(\exists(\forall(\exists(A(x_1,x_2,x_3,x_4))\rightarrow\exists(\neg A(x_1,x_1,x_3,x_4)))))\wedge$
$\neg\forall(\neg A(x_1,x_1,x_1,x_1))$ 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(ehkmkvantori 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 süntaks-sugariga saab kirjeldada kõiki väiteid, mida ilma selle sütax-sugaritagi saab kirjeldada. Et seda süntax-sugarit mitte kasutavas notatsioonis kirjeldatud seost seda süntax-sugarit kasutavasse 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:
$\exists(\neg A(1,1)\wedge\exists(A(1,2)\wedge\neg A(2,1)\wedge\neg A(2,2)\wedge\exists(\neg A(1,3)\wedge A(2,3)\wedge\neg A(3,1)\wedge$
$\neg A(3,2)\wedge\neg A(3,3))\wedge\exists(\neg A(1,3)\wedge A(2,3)\wedge\neg A(3,1)\wedge\neg A(3,2)\wedge A(3,3))))\wedge$
$\neg\exists(\neg A(1,1)\wedge\exists(\neg A(1,2)\wedge A(2,1)\wedge\neg A(2,2)\wedge\exists(A(1,3)\wedge\neg A(2,3)\wedge\neg A(3,1)\wedge$
$\neg A(3,2)\wedge\neg A(3,3))\wedge\exists(A(1,3)\wedge\neg A(2,3)\wedge\neg A(3,1)\wedge\neg A(3,2)\wedge A(3,3))))$

2.3.2 boolean sugar

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

2.3.3 operaatorite (järjekorraga) 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. näteks $\forall(x_{11}+x_{12}*x_{13}=x_{14})$ 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 $x_1*x_2=x_3$
$CALL(x_1,...,x_n)$ tähendab, et funktsioon x_1 argumentidega ... tagastab x_n'i.
operaatorite järjekord: **,[*,/],[+,-]

näide näiteks teisendab $\exists(\exists(\exists(x_1*x_2+x_3=f(x_3,x_2,x_1,x_2))))$ 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:
$\exists(on\_null(x_1))\wedge$
$\forall(x_1=x_1)\wedge$
$\forall(\forall(\forall(x_1=x_2\wedge x_2=x_3\rightarrow x_3=x_1)))\wedge$
$\forall(\forall(on\_naturaalarv(x_1)\wedge x_1=x_2\rightarrow on\_naturaalarv(x_2)))\wedge$
$\forall(\forall(on\_naturaalarv(x_1)\wedge S(x_1,x_2)\rightarrow on\_naturaalarv(x_2)))\wedge$
$\forall(\forall(\forall(\forall(=(x_1,x_2)\wedge S(x_1,x_3)\wedge S(x_2,x_4)\leftrightarrow=(x_3,x_4)))\wedge$
$\forall(on\_null(x_1)\rightarrow\neg\exists(S(x_2,x_1)\wedge on\_naturaalarv(x_2)))\wedge$
$\exists(on\_null(x_1)\wedge x_1\in K)\wedge\forall(\forall(x_1\in K\wedge S(x_1,x_2)\rightarrow x_2\in K))\rightarrow\forall(x_1\in K)\forall(on\_null(x_1)\wedge\forall(x_2+x_1=x_2)))\wedge$
$\forall(\forall(x_1+S(x_2)=S(x_1+x_2)))\forall(on\_null(x_1)\wedge\forall(x_2*x_1=x_1))\wedge$
$\forall(\forall(x_1*S(x_2)=x_1+x_1*x_2))$
kujule:
$\exists(on\_null(x_1))\wedge$
$\forall(=(x_1,x_1))\wedge$
$\forall(\forall(\forall(=(x_1,x_2)\wedge=(x_2,x_3)\rightarrow=(x_3,x_1))))\wedge$
$\forall(\forall(on\_naturaalarv(x_1)\wedge=(x_1,x_2)\rightarrow on\_naturaalarv(x_2)))\wedge$
$\forall(\forall(on\_naturaalarv(x_1)\wedge S(x_1,x_2)\rightarrow on\_naturaalarv(x_2)))\wedge$
$\forall(\forall(x_1=x_2\leftrightarrow S(x_1)=S(x_2)))\wedge$
$\forall(on\_null(x_1)\rightarrow\neg\exists(S(x_2,x_1)\wedge on\_naturaalarv(x_2)))\wedge$
$\exists(on\_null(x_1)\wedge x_1\in K)\wedge\forall(\forall(x_1\in K\wedge S(x_1,x_2)\rightarrow x_2\in K))\rightarrow\forall(x_1\in K)\forall(on\_null(x_1)\wedge\forall(+(x_2,x_1,x_2)))\wedge$

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

2.3.4 transistiivsete operaatorite sugar

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

.

2.3.5 kantsulgude sugar

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

2.3.6 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.

2.3.7 väited predikaatide argumentidena

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

2.4 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 URIl foo olevaga.

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

imporditud 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". Imortimisel 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.

2.5 näiteid selles syntaxis definitsioonides

2.5.1 piirväärtus

$(lim_{x\to a}(f(x)) = A) \leftrightarrow \forall_\epsilon(\epsilon > 0 \to \exists_\delta(\delta > 0 \wedge \forall_x(((|x-a| < \delta) \wedge (|x-a| \neq 0)) \to |f(x) - A| < \epsilon)))$

Iga kauguse ϵ funktsiooni piiväärtusest 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: $(lim_{x\to a}(f(x)) = A) \leftrightarrow \forall_{x_1}(x_1 > 0 \to \exists_{x_2}(x_2 > 0 \wedge \forall_{x_3}(((|x_3 - a| < x_2) \wedge (|x_3 - a| \neq 0)) \to |f(x_3) - A| < x_1)))$

2.5.2 peano aksioomid

selles näites on kasutatud peatükis 2.3.3 kirjeldatud syntaxsugarit.

$\exists(on\_null(x_1)) \wedge$
$\forall(x_1 = x_1) \wedge$
$\forall(\forall(\forall(x_1 = x_2 \wedge x_2 = x_3 \to x_3 = x_1))) \wedge$
$\forall(\forall(on\_naturaalarv(x_1) \wedge x_1 = x_2 \to on\_naturaalarv(x_2))) \wedge$
$\forall(\forall(on\_naturaalarv(x_1) \wedge S(x_1, x_2) \to on\_naturaalarv(x_2))) \wedge$
$\forall(\forall(x_1 = x_2 \leftrightarrow S(x_1) = S(x_2)))\forall(on\_null(x_1) \to \neg\exists(S(x_2, x_1) \wedge on\_naturaalarv(x_2))) \wedge$

$\exists(on\_null(x_1) \wedge x_1 \in K) \wedge \forall(\forall(x_1 \in K \wedge S(x_1, x_2) \to x_2 \in K)) \to \forall(x_1 \in K)$

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

2.5.3 liitmine

selles näites on kasutatud peatükis 2.3.3 kirjeldatud syntaxsugarit.

$\forall(on\_null(x_1) \wedge \forall(x_2 + x_1 = x_2))) \wedge$
$\forall(\forall(x_1 + S(x_2) = S(x_1 + x_2)))$

2.5.4 korrutamine

selles näites on kasutatud peatükis 2.3.3 kirjeldatud syntaxsugarit.

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

Part II
standard libary

seal on igasugu kasulikke väiteid, mida kasutaja saab enda omadega jaatada, et enda omade tõesust kontrollida.

robinsoni aritmeetika $\forall(on\_null(x_1)\neg\exists(S(x_1,x_2)))\wedge$
$\forall(\forall(\forall(\forall(x_1=x_2\wedge S(x_1,x_3)\wedge S(x_2,x_4)\rightarrow x_3=x_4))))on\_null(x_1)\vee\exists(S(x_2,x_1))\forall(on\_null(x_1)\wedge$
$\forall(x_2+x_1=x_2)))\wedge$
$\forall(\forall(x_1+S(x_2)=S(x_1+x_2)))\forall(on\_null(x_1)\wedge\forall(x_2*x_1=x_1))\wedge$
$\forall(\forall(x_1*S(x_2)=x_1+x_1*x_2))$

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

lahutamine $\forall(\forall(\forall(x_1-x_2=x_3\leftrightarrow x_2+x_3=x_1)))$

jagamine $\forall(\forall(\forall(x_1/x_2=x_3\leftrightarrow x_2*x_3=x_1)))$

peano viimane naturaalarvude hulga võimsuse aksioom $\exists(on\_null(x_1)\wedge$
$x_1\in K)\wedge\forall(\forall(x_1\in K\wedge S(x_1,x_2)\rightarrow x_2\in K))\rightarrow\forall(x_1\in K)$

kõik naturaalarvud on reaalarvud $\forall(on\_reaalarv(x_1)\rightarrow on\_naturaalarv(x_1))$

reaalarvude võrdlemine $\forall(\forall(x_1>x_2\leftrightarrow\exists(on\_naturaalarv(x_3)\wedge x_1*x_3>$
$x_2*x_3)))$
reaalarvude pid

täisarvu definitsioon $\forall(on\_tisarv(x_1)\leftrightarrow\exists(on\_naturaalarv(x_2)\wedge on\_naturaalarv(x_1+$
$x_2)))$

ratsionaalarvu definitsioon $\forall(on\_ratsionaalarv(x_1)\leftrightarrow\exists(on\_naturaalarv(x_2)\wedge$
$on\_tisarv(x_1*x_2)))$

3 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: disjuktsioon.

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

üheselt tõene ehk: samaselt tõene.

eksistentsaalsuskvantor ehk: olemasolukantor.

universaalsuskvantor ehk: üldisuskvantor.

elementaarseos ehk: notatsiooni poolt ette antud predikaat.

element ehk: algasi, konstantsümbol.

kvanteeritav ehk: kvantori fiktiivmuutuja

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 $\forall_x(on\_naturaalarv(x) \rightarrow f(x))$