

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Informaatika eriala

Dmitri Gabbasov

Sekventsiaalse predikaatarvutuse õpiprogramm

Bakalaureusetöö (6 EAP)

Juhendaja: Rein Prank

TARTU 2014

Sekventsiaalse predikaatarvutuse õpiprogramm

Lühikokkuvõte:

Sekventsiaalse lause- ja predikaatarvutuse tuletusülesannete lahendamiseks kasutatakse Tartu Ülikooli ainetes programmi, mille on loonud 1993. aastal Rein Prank. Paraku on see programm juba üsna vana ning selle kasutamine on mõneti ebamugav. Antud bakalaureusetöö raames loome tuletusülesannete lahendamise õpiprogrammi kaasaegse variandi, mis võimaldab sekventsiaalse predikaatarvutuse ülesannete lahendamist praktikumides ja kodutöodes ning kontrolltööde läbiviimist.

Võtmesõnad:

Õpiprogramm, sekventsiaalne lausearvutus, sekventsiaalne predikaatarvutus, võrdusega predikaatarvutus, tuletamine

A Proof Editor for Sequent Calculus

Abstract:

In the University of Tartu, sequent calculus tasks are solved using a program written in 1993 by Rein Prank. Unfortunately the program is fairly old by now, and using it can be quite cumbersome. We create a new web-based application, that allows constructing sequent calculus proofs in tutorial lessons as well as homeworks and also facilitates conducting tests.

Keywords:

Educational software, sequent calculus, propositional logic, predicate logic, proof editor

Sisukord

Sissejuhatus	4
1 Tuletamise ülesanded SML kursuses	5
2 Programm KOLDR93	7
2.1 Ülevaade	7
2.2 Lahendamist vajavad probleemid	10
3 Teistest tuletamiskeskondadest	11
3.1 Pandora	11
3.2 Stanfordini kursus <i>Introduction to Logic</i>	11
3.3 CMU kursus <i>Logic & Proofs</i>	12
4 Uue programmi kirjeldus kasutaja vaatepunktist	14
4.1 Ülesannete koostamine	14
4.2 Tuletamine	15
4.3 Tulemuste vaatamine	17
4.4 Tudengite haldus	18
5 Realisatsiooni kirjeldus	19
6 Installeerimisjuhend	20
6.1 Ehitamine	20
6.2 Käivitamine	20
7 Kokkuvõte	22
Lisad	24
Lähtekood	24
Näidis ülesandekogud	24

Sissejuhatus

Antud bakalaureusetöös loome õpiprogrammi, millega on võimalik lahendada sekventsiaalse lause- ja predikaatarvutuse tuletamise ülesandeid. Tartu Ülikoolis tehakse nimeetatud ülesandeid sellistes matemaatilise loogika ainetes nagu Sissejuhatus matemaatilisse loogikasse (SML) ja Matemaatiline loogika ja algoritmiteooria. Aastaid on nende ülesannete tegemiseks kasutatud programmi KOLDR93, ent paraku on see jäänud vanaks ning selle kasutamine on tänapäeval mõneti ebamugav. Eesmärgiks on luua kaasaegne programm, mis võimaldab ülesannete lahendamist praktikumides ja kodutöodes ning kontrolltööde läbiviimist.

Töö esimeses peatükis tutvustame SML aines käsitletavaid tuletamise ülesandeid, eelkõige anname ülevaate kasutatavast tuletamise süsteemist. Teises peatükis vaatame lähemalt praegu Tartu Ülikooli ainetes kasutatavat tuletamise programmi KOLDR93, tuues välja ka selle olulisemad puudused. Kolmandas peatükis vaatame mõningaid teiste autorite poolt loodud ning mujal ülikoolides kasutatavaid tuletamise keskkondi. Neljandas peatükis kirjeldame antud bakalaureusetöö raames loodud uut programmi. Viiendas peatükis kirjeldame loodud programmi realisatsiooni detaile. Lõpuks näitame kuuendas peatükis, kuidas valminud programmi kokku ehitada ja paigaldada.

Töoga on kaasas loodud programmi lähtekood ja näidis ülesandekogud.

1 Tuletamise ülesanded SML kursuses

Aine SML raames käsitletakse põhiliselt kolme aksiomaatilist teooriat: sekventsiaalne lausearvutus, sekventsiaalne predikaatarvutus ja võrdusega predikaatarvutus [1]. Kõigi kolme esitamiseks kasutatakse *Gentzeni-tüüpi aksiomaatilist süsteemi*. Tuletatavateks objektideks sellistes süsteemides on *sekventsid*, avaldised kujul

$$\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n \vdash \mathcal{G},$$

kus $\mathcal{F}_1, \dots, \mathcal{F}_n, \mathcal{G}$ on lause- või predikaatarvutuse valemid. Aksiomideks loetakse kõiki sekventse, mis omavad kuju

$$\Gamma, \mathcal{F}, \Delta \vdash \mathcal{F},$$

kus Γ ja Δ on suvalised valemite järjendid, mis võivad olla ka tühjad. Võrdusega predikaatarvutuse puhul lisanduvad veel järgmised aksiomid:

$$\begin{aligned} \Gamma \vdash t = t, \\ s_1 = t_1, \dots, s_n = t_n \vdash P(s_1, \dots, s_n) \rightarrow P(t_1, \dots, t_n), \end{aligned}$$

kus $t, s_1, \dots, s_n, t_1, \dots, t_n$ on termid ja P predikaatsümbol.

Tuletusreeglid on järgmised.

Paremale sissetoomise reegel	Vasakule sissetoomise või paremalt eemaldamise reegel
$\frac{\Gamma \vdash \mathcal{F} \quad \Gamma \vdash \mathcal{G}}{\Gamma \vdash \mathcal{F} \& \mathcal{G}} \text{ (}\&\text{)}$	$\frac{\Gamma, \mathcal{F}, \mathcal{G} \vdash \mathcal{H}}{\Gamma, \mathcal{F} \& \mathcal{G} \vdash \mathcal{H}} \text{ (}\&\text{)}\text{}$
$\frac{\Gamma \vdash \mathcal{F}}{\Gamma \vdash \mathcal{F} \vee \mathcal{G}} \text{ (}\vee\text{)} \quad \frac{\Gamma \vdash \mathcal{G}}{\Gamma \vdash \mathcal{F} \vee \mathcal{G}} \text{ (}\vee\text{)}$	$\frac{\Gamma, \mathcal{F} \vdash \mathcal{H} \quad \Gamma, \mathcal{G} \vdash \mathcal{H}}{\Gamma, \mathcal{F} \vee \mathcal{G} \vdash \mathcal{H}} \text{ (}\vee\text{)}\text{}$
$\frac{\Gamma, \mathcal{F} \vdash \mathcal{G}}{\Gamma \vdash \mathcal{F} \rightarrow \mathcal{G}} \text{ (}\rightarrow\text{)}$	$\frac{\Gamma \vdash \mathcal{F} \quad \Gamma \vdash \mathcal{F} \rightarrow \mathcal{G}}{\Gamma \vdash \mathcal{G}} \text{ (}\rightarrow\text{)}$
$\frac{\Gamma, A \vdash B \quad \Gamma, B \vdash A}{\Gamma \vdash A \sim B} \text{ (}\sim\text{)}$	$\frac{\Gamma \vdash A}{\Gamma, A \sim B \vdash B} \text{ (}\sim\text{)}$
$\frac{\Gamma, \mathcal{F} \vdash \mathcal{G} \quad \Gamma, \mathcal{F} \vdash \neg \mathcal{G}}{\Gamma \vdash \neg \mathcal{F}} \text{ (}\neg\text{)}$	$\frac{\Gamma \vdash \neg \neg \mathcal{F}}{\Gamma \vdash \mathcal{F}} \text{ (}\neg\text{)}$

Predikaatarvutuse puhul tulevad juurde ka kvantorreeglid.

$$\frac{\Gamma \vdash \mathcal{F}(x)}{\Gamma \vdash \forall x \mathcal{F}(x)} \quad (\forall \vdash)$$

$$\frac{\Gamma \vdash \mathcal{F}(t)}{\Gamma \vdash \exists x \mathcal{F}(x)} \quad (\exists \vdash)$$

$$\frac{\Gamma, \mathcal{F}(t) \vdash \mathcal{G}}{\Gamma, \forall x \mathcal{F}(x) \vdash \mathcal{G}} \quad (\forall \vdash)$$

$$\frac{\Gamma, \mathcal{F}(x) \vdash \mathcal{G}}{\Gamma, \exists x \mathcal{F}(x) \vdash \mathcal{G}} \quad (\exists \vdash)$$

Neis tähistab t suvalist termi. Reeglites $(\vdash \forall)$ ja $(\exists \vdash)$ ei tohi muutuja x esineda vabalt üheski alumise sekvenssi valemis.

Lisaks on olemas ka *struktuursed reeglid*.

$$\frac{\Gamma \vdash \mathcal{F}}{\Gamma, \mathcal{G} \vdash \mathcal{F}} \quad (S+)$$

$$\frac{\Gamma, \Delta, \mathcal{F} \vdash \mathcal{G}}{\Gamma, \mathcal{F}, \Delta \vdash \mathcal{G}} \quad (S \sim)$$

Tuletuseks nimetatakse sellises süsteemis sekvensside jada, kus iga sekvens, mis ei ole aksiom, viitab tuletusreeglile ja sobivatele eespool asetsevatele sekvenssidele. Tuletust otsida on samas mugavam mitte jada, vaid puu kujul. Etteantud sihtsekvensi jaoks selgitame välja, millise reegli abil see tekkida võib, seejärel selgitame välja reeglid, mille abil tekivad antud sekvensi eeldused jne. See tähendab, et puu konstrueerimine toimub alt üles liikudes. Puu lehtedeks on aksiomid. Sellise puu võib vajadusel alati kirja panna ka sekvensside jadana.

Näide. Tuletame sekvensi $A \& (B \vee C) \vdash A \& B \vee A \& C$.

$$\frac{\frac{\frac{\overline{A, B \vdash A} \quad \overline{A, B \vdash B}}{A, B \vdash A \& B} \quad (\& \vdash)}{A, B \vdash A \& B \vee A \& C} \quad (\vee \vdash)}{\frac{\frac{\overline{A, C \vdash A} \quad \overline{A, C \vdash C}}{A, C \vdash A \& C} \quad (\& \vdash)}{A, C \vdash A \& B \vee A \& C} \quad (\vee \vdash)}{A, B \vee C \vdash A \& B \vee A \& C} \quad (\vee \vdash)}{A \& (B \vee C) \vdash A \& B \vee A \& C} \quad (\& \vdash)$$

2 Programm KOLDR93

2.1 Ülevaade

KOLDR93 on Rein Pranki poolt 90-ndate alguses loodud programm, mida kasutatakse praegu matemaatilise loogika ainetes sekventsiaalse lause- ja predikaatarvutuse tuletamise ülesannete lahendamiseks.

Lahendatavad ülesanded on ülesandekogu failis, see pole muud, kui teatud viisil vormindatud tekstifail (joonis 1).

LAUSEARVUTUSE TULETUSÜLESANDED		
Ülesandeks on tuletada sekventsid (konstrueerida nende tuletuspuud). Tuletuspuu moodustamist alustatakse lõpust: lähtudes etteantud sekventsist, liigutakse samm-sammult ülespoole. Puu iga haru tipus peab olema aksiom. Sammu tegemiseks rakendatakse sobivat tuletusreeglit.		
A. Konjunktsioon ja disjunktsioon		
1.	$X \& Y \vdash Y \& X$	/4
2.	$X \vee Y \vdash Y \vee X$	/5
3.	$X \& (Y \& Z) \vdash (X \& Y) \& Z$	/7
4.	$(X \& Y) \& Z \vdash X \& (Y \& Z)$	/8
5.	$X \vee (Y \vee Z) \vdash (X \vee Y) \vee Z$	/10
6.	$(X \vee Y) \vee Z \vdash X \vee (Y \vee Z)$	/10

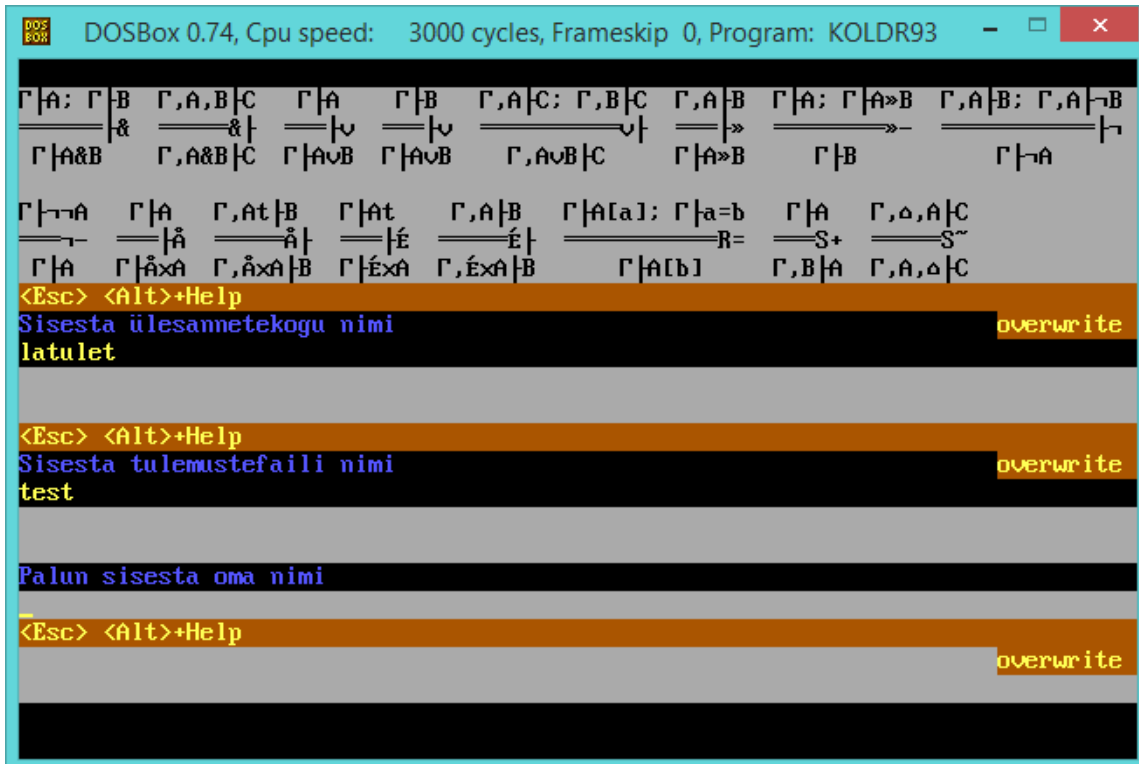
Joonis 1: Ülesandekogu faili algus.

Read, mis algavad numbriga, tähistavad tuletamiseks mõeldud sekventse. Selline vorming võimaldab ülesandekogule lisada ka seletavat teksti ja instruksioone. Iga sekventsi taga olev kaldkriipsuga number tähistab ettenähtud sammude arvu, millega antud sekventsi peaks olema võimalik tuletada. Selle numbri määrab õppejõud ise, kui koostab ülesandekogu.

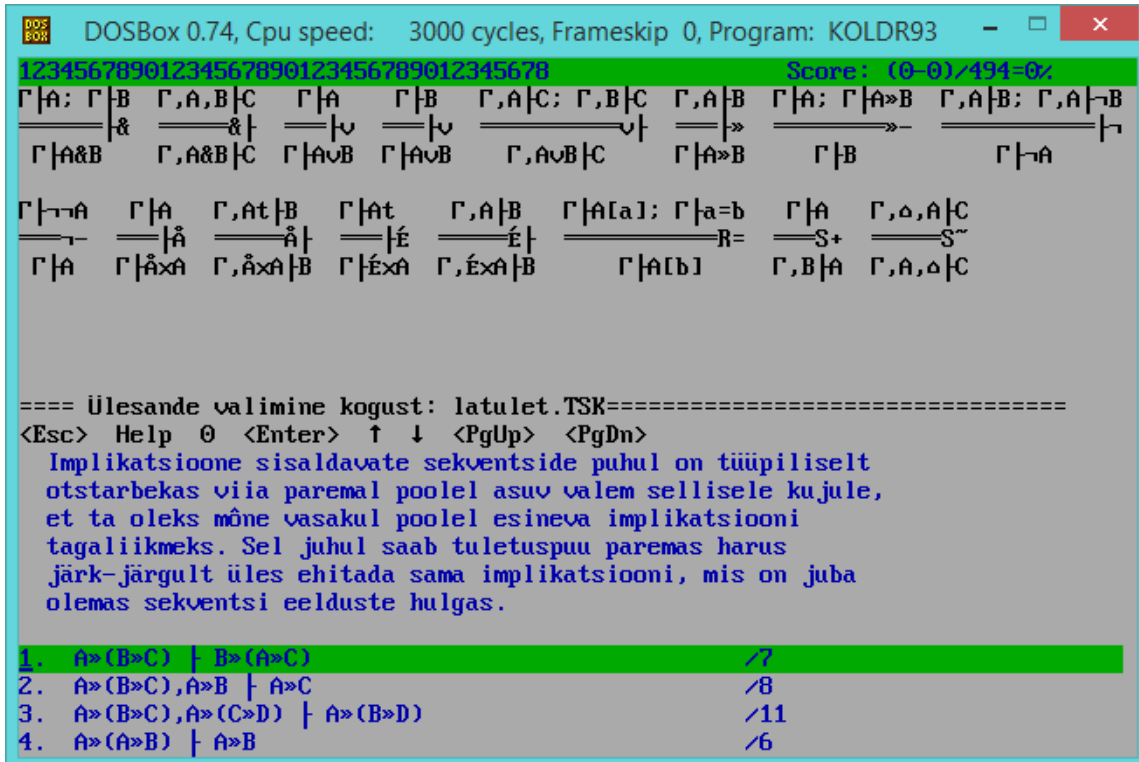
Programmi käivitades küsitakse kasutajalt esmalt tulemuste faili nime, nimelt salvestatakse tulemused eraldi faili ning alati on võimalik avada juba olemasolev fail. Juhul, kui sisestatud tulemustefaili ei eksisteeri, küsitake ka ülesandekogu faili nime ja lahendaja nime (joonis 2).

Tulemuste fail on sisuliselt ülesandekogu faili koopia, kuhu on lisatud lahendaja nimi, miinuspunktide arv, kontrollsumma – vältimaks faili käsitsi muutmist – ning iga lahendatud sekventsi tuletuspuu suurus. Tuletuspuid tulemuste failis ei sisaldu, neid saab soovi korral iga sekventsi puhul lasta salvestada eraldi faili.

Ülesandekogu avanedes saab kasutaja valida sekventsi (joonis 3) ning hakata seda tuletama (joonis 4). Tuletamisel liigub kasutaja nooleklahvidega mööda tuletuspuud ja saab



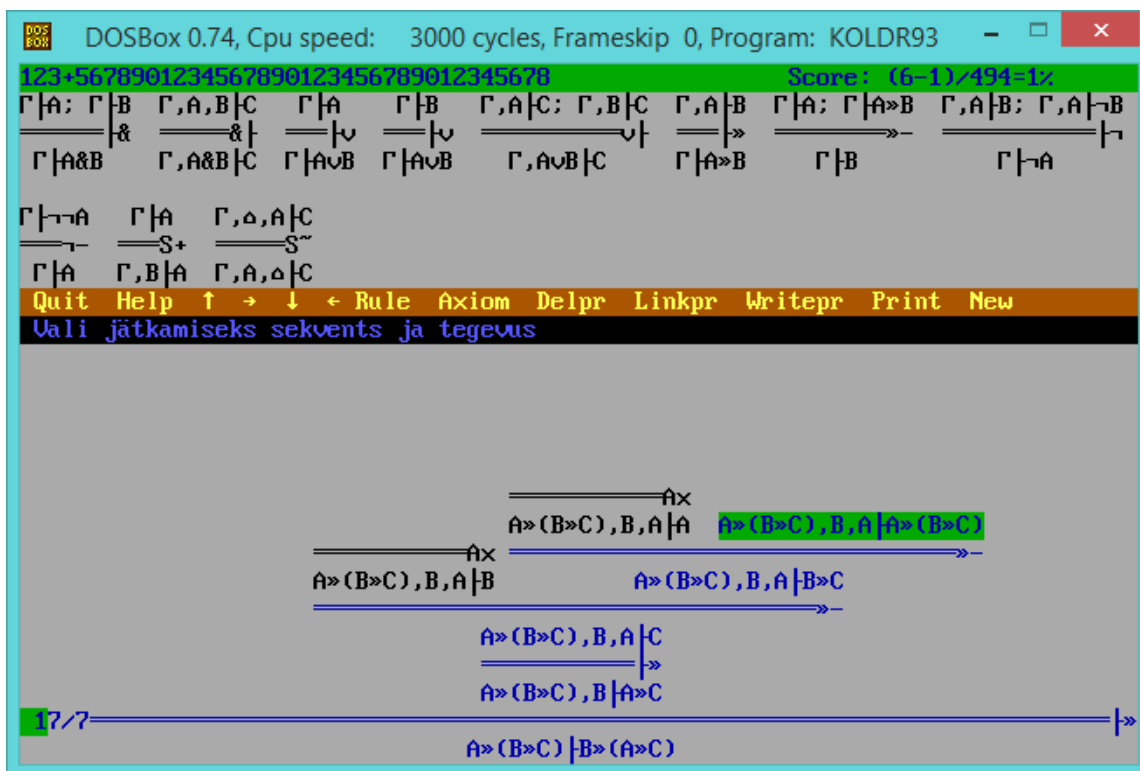
Joonis 2: KOLDR93 küsib käivitamisel failinimesid ja kasutaja nime.



Joonis 3: Ülesandekogu vaade programmis KOLDR93.

aktiivse sekventsiga juures sooritada ühe järgmistest toimingutest:

- 1) kontrollida kas sekvents on aksiom (ja see vastavalt märgistada),
- 2) rakendada sekventsile tuletusreeglit,
- 3) kustutada sekventsiga kohale jääva tuletuspuu osa,
- 4) laadida failist sekventsiga tuletus,
- 5) salvestada faili sekventsiga tuletus.



Joonis 4: Tuletamine programmis KOLDR93.

Reegli rakendamisel kontrollib programm, kas reegli rakendamine on võimalik, ja lisab reegli ülemised sekventsiga puusse. Kui reegli ülemised sekventsiga sisaldavad valemeid või terme, mida alumises sekventsiga veel ei ole, küsitakse need kasutajalt.

Teatud reeglitega (disjunktsiooni paremale sissetoomine, eituse paremale sissetoomine, implikatsiooni paremalt eemaldamine ja eelduse lisamine) on võimalik puusse tekitada ka mittetuletatavaid sekventse. Programm ei takista kasutajat selliseid sekventse loomast, küll aga saab kasutaja sellisele sekventsiga reeglit rakendada üritades veateate ja miinuspunkti.

2.2 Lahendamist vajavad probleemid

Programmi KOLDR93 on Tartu Ülikoolis tuletusülesannete lahendamiseks kasutatud juba veidi üle kahekümne aasta. Toome siiski siinkohal välja mõned selle programmi puudused, millest enamused tulenevad eelkõige programmi vanusest.

1. Programmi käivitamiseks on vajalik kas 32-bitine Microsoft Windows operatsioonisüsteem või DOS-emulaator. Tartu Ülikooli arvutiteaduse instituudi arvutiklassides, aga ka paljude tudengite isiklikes arvutites, on peal 64-bitised operatsioonisüsteemid, seega emulaatori kasutamine on pigem tavaline kui harv juhus. Mõistagi on emulaatorit vaja ka siis, kui soovitakse kasutada programmi Windowsist erineval operatsioonisüsteemil.
2. Programmi kasutajaliides on pärit ajast, mil programm oli loodud, ja jääb alla sellele, mida võiks oodata ühelt kaasaegselt arvutiprogrammilt. Võimatu on näiteks töötada hiirega, ka teatud märke (nt kvantoreid) ei suuda programm korralikult kuvada. Lisaks on programmi akna laius fikseeritud 80 tähemärgi peal, mis tähendab, et ühtegi sellest pikemat sekvenssi ei ole see võimeline kasutajale kuvama (ja ei võimalda neid ka tekitada). See piirang takistab muuhulgas näiteks mittetriviaalsete võrdusega predikaatloogika ülesannete loomist.
3. Töö programmiga on suuresti seotud failidega – ülesandekogu on fail, tulemused on failis, tuletuspuud on samuti eraldi failides. See tähendab, et toimub pidev õppejõu ja tudengi vaheline failide edastamine. Kontrolltööde puhul (kus soovitakse näha ka tuletuspuid) peab tudeng näiteks 4 ülesande korral õppejõule saatma kokku 5 faili.

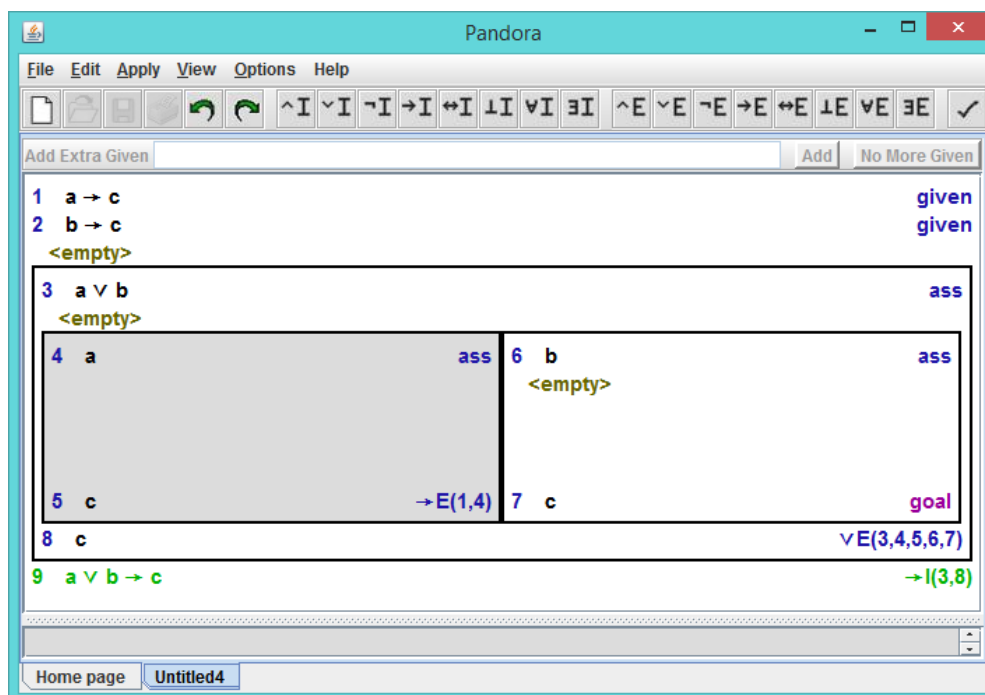
Antud bakalaureusetöö eesmärk on luua uus programm, mis oleks funktsionaalsuse poolest analoogne KOLDR93-le ja ei omaks ühtegi eelmainitud puudust. Teiste sõnadega, uus programm

- 1) peab olema operatsioonisüsteemist sõltumatu, st seda peaks olema võimalik kasutada suvalisel arvutil (ka näiteks tahvelarvutitel),
- 2) peaks omama moodsat ja lihtsasti kasutatavat kasutajaliidest,
- 3) peaks võimaldama lahendada ülesandeid ilma, et kumbki osapool peaks edastama teisele mingeid faile, meile vms.

3 Teistest tuletamiskeskondadest

3.1 Pandora

Pandora [2, 3] on Inglismaal Londoni Imperial College'is loodud tuletamise keskkond, kus tuletusi koostatakse nn. loomuliku tuletamise süsteemis [4]. Selle kirjutas ca 1998. aastal Dan Ehrlich ning ülikooli tudengid on seda hiljem mõnevõrra täiendanud [5].



Joonis 5: Tuletamine Pandoras. Üleval programmi aknas on näha tööriistariba tuletamisreeglite nuppudega. Valges alas on tuletus ise. Kaks ülemist valemit (1,2) on ülesandes antud eeldused. Viimane valem (9) on see, mida on vaja tuletada. Implikatsiooni tõestamiseks on tehtud eeldus, et kehtib selle vasak pool (3), misjärel on vaja tuletada selle parem pool (8). Et tuletada midagi disjunktsioonist (3), on vaja seda teha tema mõlemast poolest (4,6). Valem 5 on märgitud tõestatuks kasutades eeldust 1. Sama on vaja veel teha valemiga 7.

Üheks Pandora eripäraks on see, et peale tavapärase tuletusreeglite võimaldab ta kasutada ka mõningaseid muid tuletusvõtteid, mille eesmärk on valdavalt vähendada tuletuse pikkust ja konstrueerimiseks kuluvat aega. Näiteks on võimalik lisada tuletusse suvalisi eeldusi, märkides neid nn. *Trust Me* valemiteks. Peale selle on võimalik rakendada ka välistatud kolmanda “reeglit”, st kehtivaks saab kuulutada suvalise valemi kujul $A \vee \neg A$.

3.2 Stanfordini kursus *Introduction to Logic*

Stanfordi ülikooli Coursera kursuses *Introduction to Logic* [6] kasutatakse veebipõhist tuletamise süsteemi (joonis 7). Kursusel osalejatele antakse link ülesandele, mille ära

lahendades saavad nad nn. lahenduse koodi, mille peavad vastusena esitada. Kasutusel on loomuliku tuletamise süsteem.

Start from the given premises. Apply rules of inference by checking the lines you wish to use as premises and click the button for the desired rule of inference. Reiteration allows you to repeat an earlier item. To delete one or more lines from a proof, check the desired lines and click Delete. Whenever entering expressions, use Ascii characters only. Use ~ for \neg ; use & for \wedge ; use | for \vee ; use => for \Rightarrow ; and use <=> for \Leftrightarrow .

Proof Editor																							
<input type="checkbox"/> 1.	$p \Rightarrow \sim q$	Premise																					
<input type="checkbox"/> 2.	$\sim q \ \& \ p \Rightarrow r$	Premise																					
<input type="checkbox"/> 3.	p	Premise																					
<input type="checkbox"/> 4.	$\sim q$	Implication Elimination: 1, 3																					
<input type="checkbox"/> 5.	$p \ \& \ p$	And Introduction: 3, 3																					
<input type="checkbox"/> 6.	$p \ \& \ \sim q$	And Introduction: 3, 4																					
<input type="checkbox"/> 7.	$\sim q \ \& \ p$	And Introduction: 4, 3																					
<input type="checkbox"/> 8.	$\sim q \ \& \ \sim q$	And Introduction: 4, 4																					
<input type="checkbox"/> 9.	r	Implication Elimination: 2, 7																					
Goal	r	Complete Answer code: <input type="text" value="6578"/>																					
<table border="0"> <tr> <td><input type="button" value="Assumption"/></td> <td><input type="button" value="Negation Introduction"/></td> <td><input type="button" value="Implication Introduction"/></td> </tr> <tr> <td><input type="button" value="Reiteration"/></td> <td><input type="button" value="Negation Elimination"/></td> <td><input type="button" value="Implication Elimination"/></td> </tr> <tr> <td><input type="button" value="Delete"/></td> <td><input type="button" value="And Introduction"/></td> <td><input type="button" value="Biconditional Introductio"/></td> </tr> <tr> <td></td> <td><input type="button" value="And Elimination"/></td> <td><input type="button" value="Biconditional Eliminatio"/></td> </tr> <tr> <td></td> <td><input type="button" value="Or Introduction"/></td> <td></td> </tr> <tr> <td></td> <td><input type="button" value="Or Elimination"/></td> <td></td> </tr> <tr> <td colspan="2" style="text-align: center;"><input type="button" value="Reset"/></td> <td style="text-align: center;"><input type="button" value="Show XML"/></td> </tr> </table>			<input type="button" value="Assumption"/>	<input type="button" value="Negation Introduction"/>	<input type="button" value="Implication Introduction"/>	<input type="button" value="Reiteration"/>	<input type="button" value="Negation Elimination"/>	<input type="button" value="Implication Elimination"/>	<input type="button" value="Delete"/>	<input type="button" value="And Introduction"/>	<input type="button" value="Biconditional Introductio"/>		<input type="button" value="And Elimination"/>	<input type="button" value="Biconditional Eliminatio"/>		<input type="button" value="Or Introduction"/>			<input type="button" value="Or Elimination"/>		<input type="button" value="Reset"/>		<input type="button" value="Show XML"/>
<input type="button" value="Assumption"/>	<input type="button" value="Negation Introduction"/>	<input type="button" value="Implication Introduction"/>																					
<input type="button" value="Reiteration"/>	<input type="button" value="Negation Elimination"/>	<input type="button" value="Implication Elimination"/>																					
<input type="button" value="Delete"/>	<input type="button" value="And Introduction"/>	<input type="button" value="Biconditional Introductio"/>																					
	<input type="button" value="And Elimination"/>	<input type="button" value="Biconditional Eliminatio"/>																					
	<input type="button" value="Or Introduction"/>																						
	<input type="button" value="Or Elimination"/>																						
<input type="button" value="Reset"/>		<input type="button" value="Show XML"/>																					

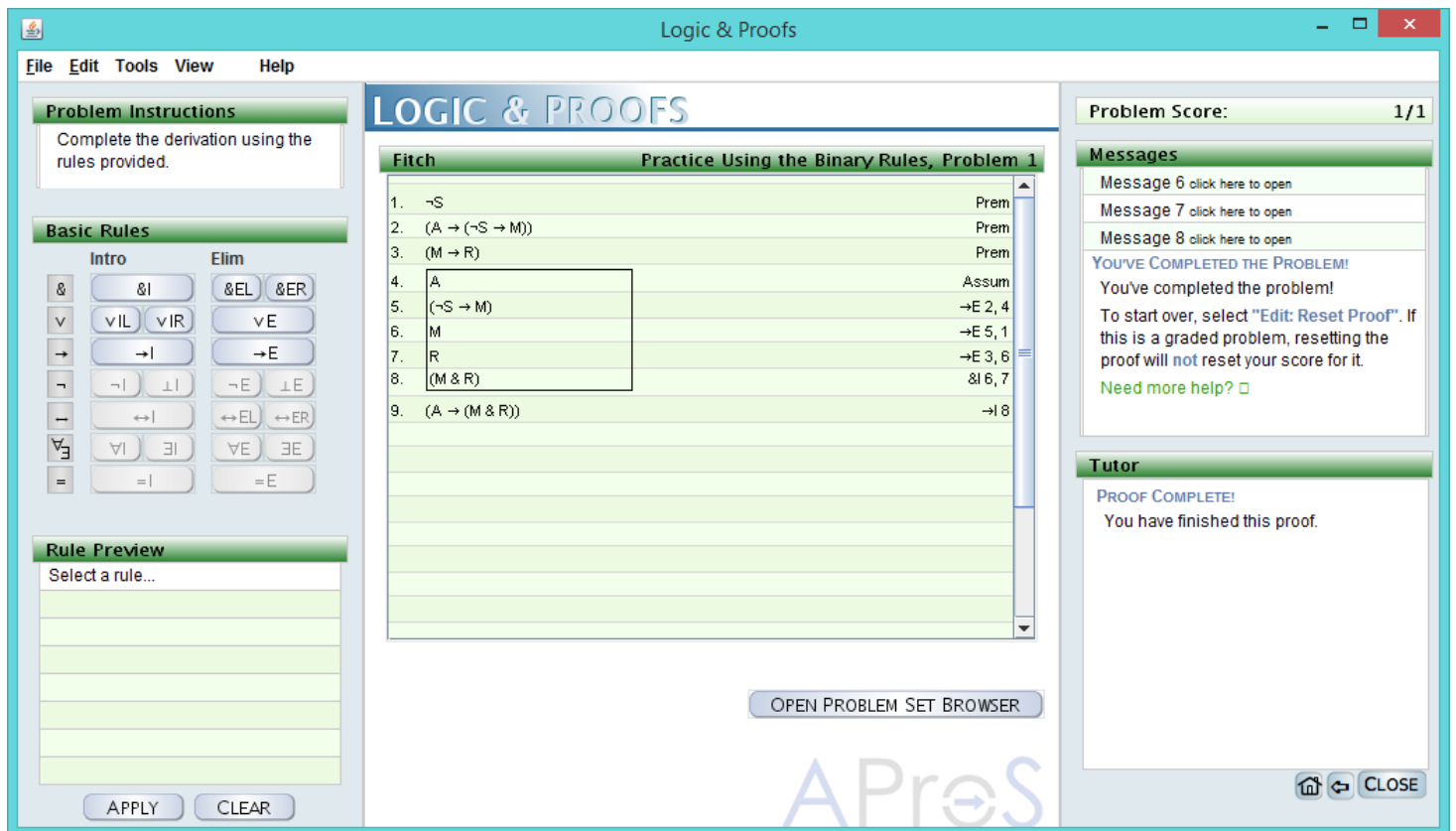
Joonis 6: Tuletamine Stanfordini *Coursera* kursuses. Keskel valges alas on tuletus. All asuvad tuletusreeglite nupud. Näha on ka lõpetatud tuletuse puhul näidatavat vastuse koodi.

Tuleb ära märkida, et kogu tuletamise rakendus on realiseeritud JavaScriptis ja jookseb täiesti veebilehitsejas. Vastuse kood on tegelikult lehe lähtekoodis näha, seega ei sobi see lahendus hinnatavate kontrolltööde läbiviimiseks.

3.3 CMU kursus *Logic & Proofs*

Carnegie Mellon'i ülikooli *online* kursuses *Logic & Proofs* [7] kasutatakse tuletamiseks *AProS Proof Lab* [8] nimelist Java rakendust. Rakendus võimaldab sirvida ja sooritada

kõiki kursuse harjutusi. Tulemused salvestatakse ja edastatakse automaatselt. Kasutusel on jällegi loomuliku tuletamise süsteem.



Joonis 7: Tuletamine CMU kursuses. Vasakul on näha nupud tuletusreeglite rakendamiseks. Keskkel on lõpule viidud tuletus.

4 Uue programmi kirjeldus kasutaja vaatepunktist

Anname nüüd ülevaate antud bakalaureusetöö raames loodud uuest tuletamise programmist. Tegu on veebipõhise rakendusega ning selle kasutamine toimub läbi veebilehitseja.

Rakendus on kasutajate põhine, st selle kasutamiseks tuleb kõigepealt sisse logida. Iga kasutaja kuulub ühte kindlasse rolli, nendeks on:

- 1) *lektor* – saab luua ülesandekogusid ja uusi kasutajaid,
- 2) *assistent* – saab vaadata tudengite lahendusi ja muuta nende rühma ja parooli,
- 3) *tudeng* – saab sirvida ülesandekogusid ja neid lahendada.

Peale selle saab assistent teha kõike seda, mida ka tudeng, ja lektor kõike seda, mida ka assistent.

4.1 Ülesannete koostamine

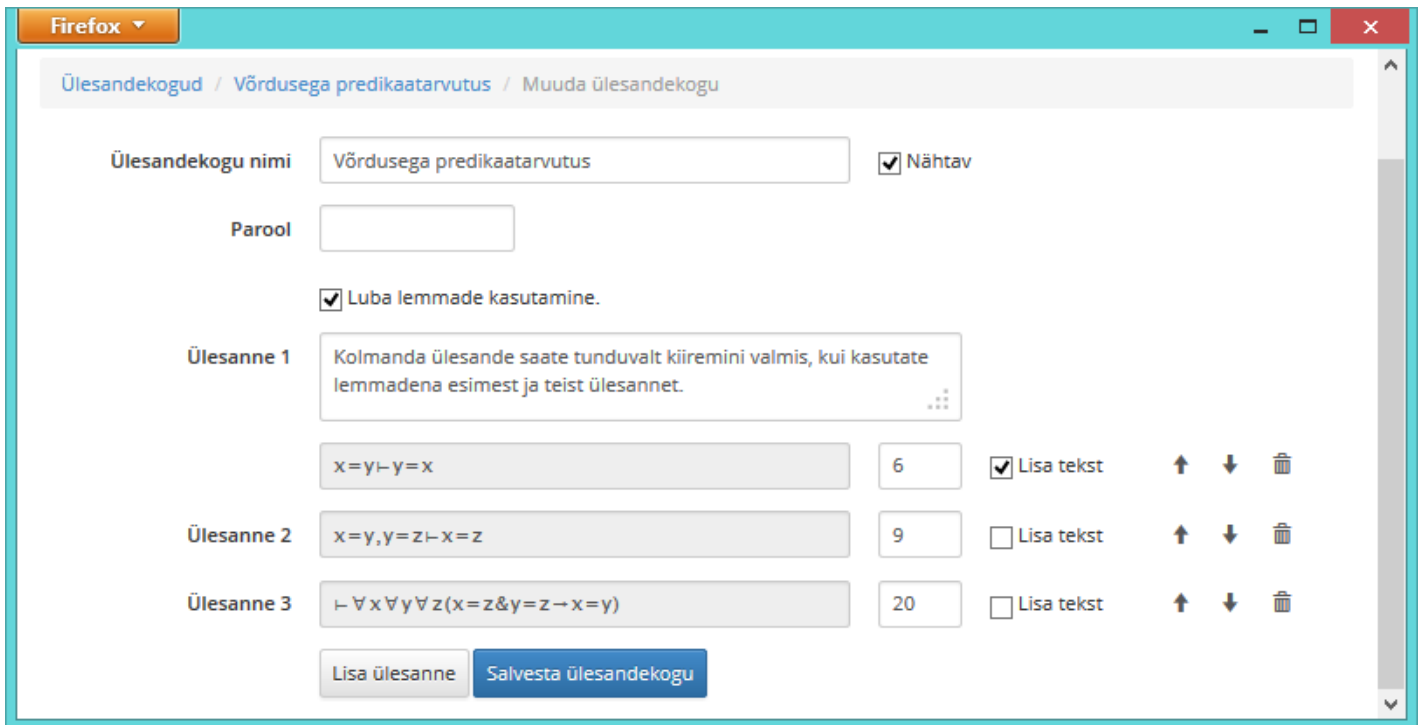
Ülesandekogude konseptsioon on uues rakenduses analoogne KOLDR93-e omaga. Ülesandeid saab koostada ainult lektori rollis olev kasutaja. Joonis 8 näitab, kuidas näeb välja ülesandekogu loomise ja ühtlasi ka muutmise vaade. Ülesandekogu juures saab määrata järgmised atribuudid:

- 1) ülesandekogu nimi,
- 2) kas ülesandekogu on lahendamiseks avatud (nähtav),
- 3) parool (ülesandekogu avamiseks peavad tudengid teadma parooli),
- 4) kas lemmade kasutamine on lubatud (lemmadest räägime hiljem).

Ülesandekogule saab sekventse lisada ja neid kustutada. Muuta olemasolevaid sekventse ei saa, muidu ei vastaks teiste poolt koostatud tuletuspuud enam sellele sekventsile.

Iga sekventsipuhul saab määrata ka ettenähtud (optimaalse) sammude arvu, millega seda peaks saama tuletada. Peale selle saab iga sekventsiga siduda ka jupikese teksti. Sekventsiga seotud teksti näidatakse ülesannete vaates vahetult enne seda sekventsit. Lisatud tekst toetab *Markdown* süntaksit, mis tähendab, et see võib sisaldada vormindusinstruktsioone, millega võib tekitada näiteks paksus kirjas teksti, vahepealkirju jm.

Ülesandekogusid on võimalik ka failidest importida ja failidesse eksportida, hõlbustamaks nende ümberkäimist juhul, kui programmi soovitakse näiteks kasutada mitmes erinevas serveris.



Joonis 8: Ülesandekogu muutmine.

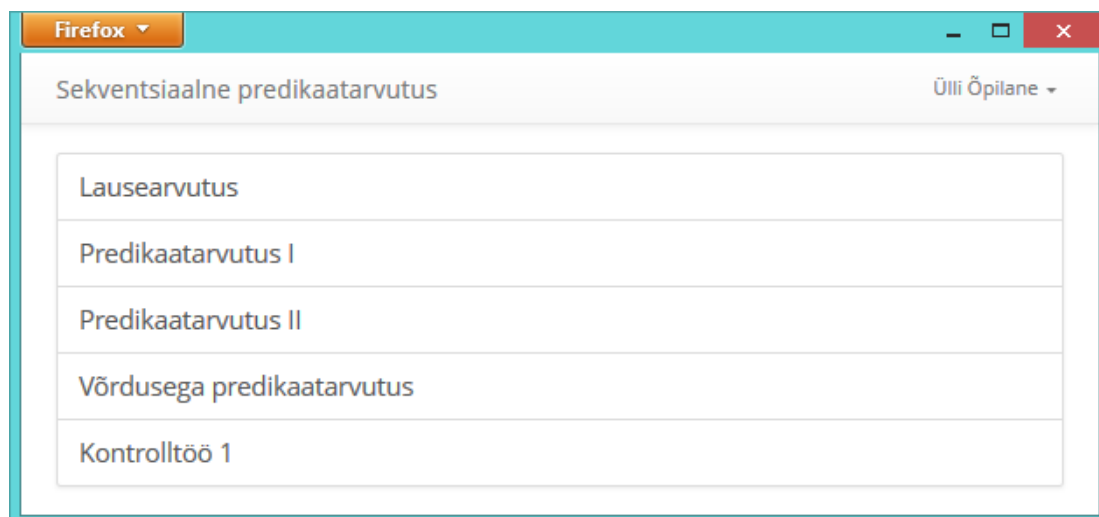
4.2 Tuletamine

Rakenduse avalehel on nimekiri kõigist ülesandekogudest (joonis 9). Valides neist mõne avaneb vaade ülesannetest s.o sekventsides (joonis 10), kus on muuseas näha ka mitme sammuga on iga sekvents tuletatav ning, juhul, kui kasutaja on sekvensi tuletamist juba alustanud, ka tehtud sammude arvu. Lõpuni tuletatud sekventsist annab märku rohelse taustaga sammude arv.

Nüüd, valides sekvensi, saab hakata seda tuletama (joonis 11). Analoogselt vanale programmile tuleb tuletamiseks valida soovitud sekvents ning seejärel kas valida mõni tuletusreegel või märkida sekvents aksioomiks. Seda tegevust tuleb korrata seni kuni kõik puu tippudes olevad sekventsid on märgitud aksioomideks. Juhul, kui kasutaja poolt valitud reegel ei ole aktiivsele sekventsile rakendatav, saab ta vastava veateate. Samuti näidatakse veateadet, kui, rakendades reeglit, mis nõuab kasutajalt valemi sisestamist, sisestab ta vigase valemi või kui tuletamise käigus püüab kasutaja rakendada reeglit mittetuletatavale sekventsile. Valitud sekvensi kohal oleva tuletuspuu osa saab ka ära kustutada.

Aksioomi kontrolli tehes märgitakse käsilolev haru kas lõpetatuks (kui sekvents tõesti on aksioom) või kuvatakse veateade (kui sekvents siiski pole aksioom). Rakendus toetab nii aksioomi $\Gamma, A, \Delta \vdash A$ kui ka võrdusega predikaatarvutuse aksioome.

Erinevalt KOLDR93-st toetab uus programm ka nn. lemmade rakendamist. See või-



Joonis 9: Ülesandekogude nimekiri.



Joonis 10: Avatud ülesandekogu.

maldab märkida sekventse, mis pole aksioomid, lõpetatuks, kasutades mõnda muud samas ülesandekogus eespool olevat ülesannet. Näiteks, kui mõne sekventsi tuletamise käigus jõudakse olukorda, kus on vaja tuletada võrduse sümmeetrilisus (sekvents kujul $\Gamma, x = y \vdash y = x$), siis, eeldusel, et samas ülesandekogus on juba olemas sümmeetrilisuse tõestamise ülesanne (st. eespool peab olema ülesanne sekventsiga $a = b \vdash b = a$), saab seda eelnevat ülesannet “rakendada” aktiivsele sekventsile, märkides see niimoodi tuletatuks.

Lemma rakendamiseks peab aktiivne sekvents ühilduma lemma sekventsiga. Sekventsid ühilduvad parajasti siis, kui lemma sekventsist on võimalik saada aktiivne sekvents kas indiviidimuutujate ümbernimetamise või lausemuutujate valemitega asendamise teel.

The screenshot shows a web browser window with the address bar displaying 'Ülesandekogud / Lausearvutus / 42'. Below the address bar is a list of logical inference rules, each with a checkmark, a refresh icon, or a close icon. The rules include:

- $\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \& B} (I\&)$
- $\frac{\Gamma, A, B \vdash C}{\Gamma, A \& B \vdash C} (\&E)$
- $\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} (I\vee_1)$
- $\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} (I\vee_2)$
- $\frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C} (\vee E)$
- $\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} (I\rightarrow)$
- $\frac{\Gamma \vdash A \quad \Gamma \vdash A \rightarrow B}{\Gamma \vdash B} (I\rightarrow)$
- $\frac{\Gamma, A \vdash B \quad \Gamma, B \vdash A}{\Gamma \vdash A \sim B} (I\sim)$
- $\frac{\Gamma \vdash A}{\Gamma, A \sim B \vdash B} (\sim E)$
- $\frac{\Gamma, A \vdash B \quad \Gamma, A \vdash \neg B}{\Gamma \vdash \neg A} (I\neg)$
- $\frac{\Gamma \vdash \neg \neg A}{\Gamma \vdash A} (I\neg)$
- $\frac{\Gamma \vdash A}{\Gamma, B \vdash A} (S+)$
- $\frac{\Gamma, \Delta, A \vdash B}{\Gamma, A, \Delta \vdash B} (S-)$

Below the list is a detailed proof tree for the rule $(I\vee_1)$. The root node is $\frac{-X \rightarrow Y, -(X \vee Y), X \vdash X}{-X \rightarrow Y, -(X \vee Y), X \vdash X \vee Y} (I\vee_1)$. The tree branches into two paths:

- The left path starts with $\frac{-X \rightarrow Y, -(X \vee Y), X \vdash X}{-X \rightarrow Y, -(X \vee Y) \vdash X} (I\vee_1)$, which then leads to $\frac{-X \rightarrow Y, -(X \vee Y) \vdash X}{-X \rightarrow Y, -(X \vee Y) \vdash X \vee Y} (I\vee_2)$.
- The right path starts with $\frac{-X \rightarrow Y, -(X \vee Y), X \vdash \neg(X \vee Y)}{-X \rightarrow Y, -(X \vee Y) \vdash \neg(X \vee Y)} (I\rightarrow)$, which then leads to $\frac{-X \rightarrow Y, -(X \vee Y) \vdash \neg(X \vee Y)}{-X \rightarrow Y \vdash \neg \neg(X \vee Y)} (I\neg)$.

The final result of the proof is $\frac{-X \rightarrow Y \vdash \neg \neg(X \vee Y)}{-X \rightarrow Y \vdash X \vee Y} (I\rightarrow)$.

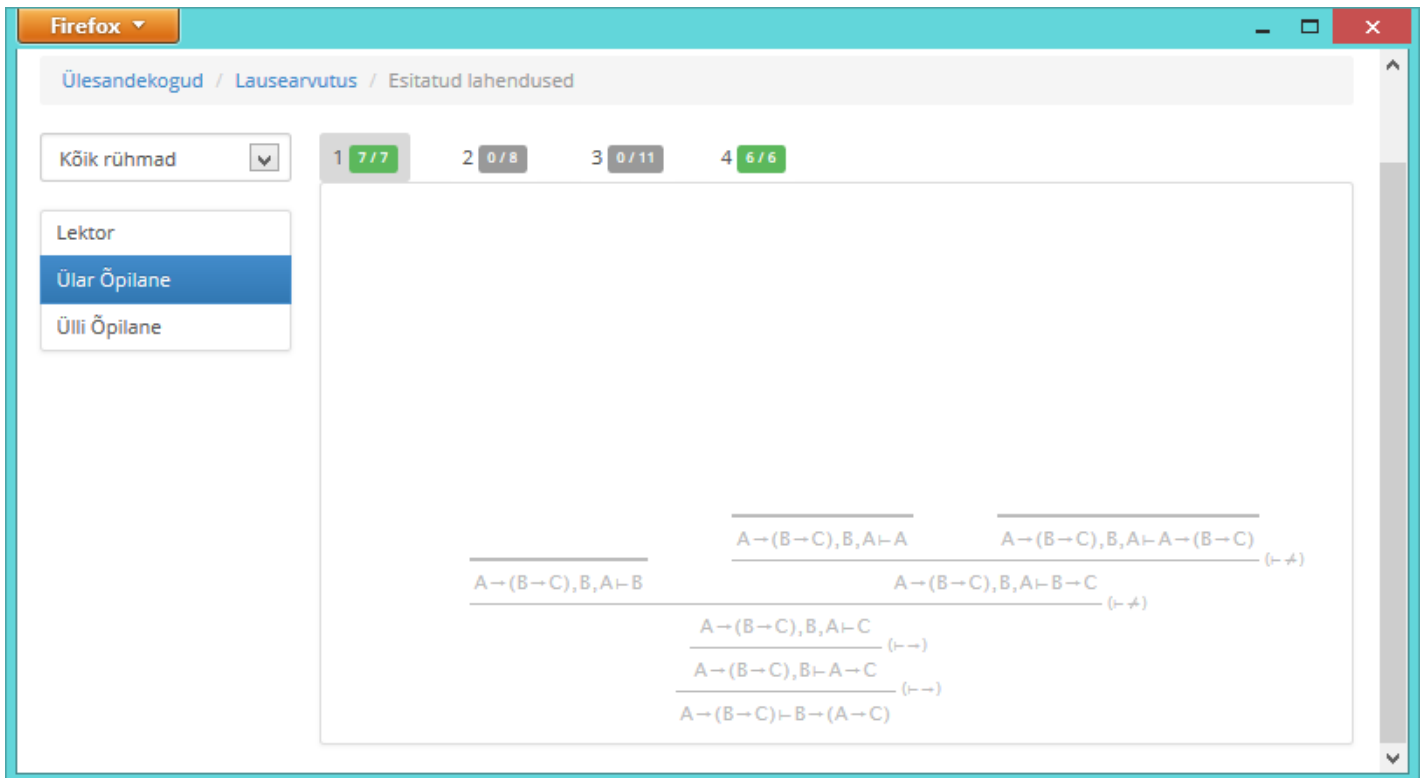
Joonis 11: Tuletamine. Üleval on tuletusreeglid, nende kõrval on kolm nuppu – aksioomi kontroll, lemma valimine ja aktiivse sekventsi kohalt tuletuspuu kustutamine. Alumises osas on tuletuspuu, mille nurgas on tehtud ja ettenähtud sammude arvud. Lõpetatud puu harud on halli värvi. Aktiivne sekvents on oranži fooniga.

Tuletused salvestatakse automaatselt serverisse ning õppejõud saab iga tudengi lahendust suvalisel hetkel vaadata (ka tuletamise ajal).

4.3 Tulemuste vaatamine

Lectori ja assistendi rollis olevad kasutajad saavad vaadata teiste kasutajate lahendusi. Joonis 12 näitab lahenduste vaadet. Vasakus ääres on kasutajate nimed, mida saab ka rühma järgi filtreerida. Valides mõne nime tekib keskele üles ülesannete nimekiri, kus iga ülesande juures on näha ka vastava tuletuspuu suurust. Roheline taust annab märku sel-

lest, et antud ülesanne on lõpuni lahendatud. Ülesandele vajutades avaneb ka tuletuspuu ise.



Joonis 12: Lahenduste vaade.

4.4 Tudengite haldus

Lektori õigustes kasutaja saab lisada rakendusse uusi kasutajaid ja määrata nende rolle. Assistenti õigustes kasutaja saab muuta vaid olemasolevate kasutajate rühma ja parooli. Kasutajate vaadet kujutab joonis 13.

Võimalik on ka importida korraga mitu kasutajat. Vastavale lingile vajutades ilmub nähtavale suur teksti sisestuslahter. Sisestatava teksti iga rida peab sisaldama ühe tudengi andmeid. Ühel real peavad olema tudengi matriklinumber, eesnimi, perekonnanimi ja rühm. Eraldusmärgiks peab olema tabulaator. Selline formaat võimaldab näiteks Microsoft Excelist andmeid kopeerida ja neid otse lahtrisse kleepida.

Nimi	Kasutajanimi	Rühm	Roll	Valikud
Lektor	lektor		Lektor	
<input type="checkbox"/> Assistent	assistent		Assistent	
<input type="checkbox"/> Ülar Õpilane A54321	tud1	1	Tudeng	
<input type="checkbox"/> Ülli Õpilane A12345	tudeng	1	Tudeng	

Joonis 13: Kasutajate haldamine.

5 Realisatsiooni kirjeldus

Antud bakalaureusetöö raames loodud programmi näol on tegu veebipõhise serverrakendusega. Käivitamisel luuakse HTTP server, millele veebilehitsejad peavad siis päringuid esitama. Programm on jagatud kaheks mooduliks – ees- ja tagakomponendiks. Eeskomponent sisaldab ainult HTML faile ja JavaScript koodi ning loob kogu veebilehitsejas näha oleva kasutajaliidese. Tagakomponent see-eest sisaldab kogu info talletamise ja töötlemisega seotud loogikat, s.o andmebaasi haldamine, matemaatilise loogika teostus jm, tagakomponent ei genereeri HTML-i vaid tagastab kogu päritava info JSON formaadis.

Rakenduse tagakomponent on realiseeritud *Javas* ning kasutab *Jersey* raamistikku. Ära tuleb märkida, et kasutatud on märtsis 2014 välja tulnud *Java* versiooni 8, mis erineb varasemast versioonist muuhulgas selle poolest, et võimaldab kasutada nn. *lambda* funktsioone [9]. Andmebaasi haldamiseks kasutatakse H2 andmebaasi mootorit ja JPA programmiliidest.

Eeskomponent koosneb erinevaid vaateid realiseerivatest HTML failidest ja TypeScript keeles kirjutatud ning JavaScriptiks kompileeritavast koodist. Kasutajaliides on rajatud AngularJS raamistikule.

Rakendus toetab ka erinevaid keeli, kaasas on eesti ja inglise keele failid ning nende näitel on lihtne luua uusi.

6 Installeerimisjuhend

6.1 Ehitamine

Kogu vajalik lähtekood asub kaustas `axiom`, selles on omakorda kolm kausta:

- 1) `axiom-angular`, siin asub rakenduse eeskomponent,
- 2) `axiom-backend`, siin asub rakenduse tagakomponent,
- 3) `langs`, siin asuvad keele failid.

Rakenduse kokku ehitamiseks peab arvutis olema paigaldatud järgmine tarkvara:

- Java SE Development Kit (versioon 8u5)
- Apache Maven (versioon 3.2.1)
- Node.js (versioon 0.10.26)
- Node.js moodul `grunt-cli` (versioon 0.1.13)

Lisaks peab ehitamise ajal arvutil olema ligipääs internetile.

Tagakomponendi ehitamiseks tuleb kaustas `axiom-backend` jooksutada järgmine käsk:

```
mvn package
```

Selle tulemusena tekib kausta `axiom-backend/target` fail nimega `axiom.jar`. Eeskomponendi ehitamiseks tuleb kaustas `axiom-angular` jooksutada järgmised kaks käsku

```
npm install
```

```
grunt
```

Pärast seda tekib kausta `axiom-angular/dist` fail `static.zip`.

6.2 Käivitamine

Rakenduse käivitamiseks tuleb ühte kausta tõsta järgmised asjad:

- 1) fail `axiom.jar`,
- 2) kaust nimega `static`, kuhu sisse tuleb lahti pakkida arhiiv `static.zip`,
- 3) kaust `langs`

Seejärel tuleb käivitada tagakomponent järgmise käsuga:

```
java -jar axiom.jar
```

Selliselt käivitatusena hakkab server kuulama TCP pordil 8000 ja vastab ainult samast arvutist algatatud päringutele (st server seob ennast aadressi 127.0.0.1 külge). Vajadusel saab aadressi ja pordi ka ette anda:

```
java -Dee.ut.axiom.host=0.0.0.0 -Dee.ut.axiom.port=80 -jar axiom.jar
```

Sellisel viisil käivitatuna kuulab rakendus HTTP vaikepordil ja vastab ükskõik kust tulnud päringutele.

Esmasel käivitamisel luuakse automaatselt andmebaas ja esimene lektori õigustes kasutaja nimega `admin`, parooliks on samuti `admin`. Jääb üle vaid minna veebilehitsejaga aadressile `http://127.0.0.1/` ja rakendus ongi kasutusvalmis.

7 Kokkuvõte

Antud töö eesmärgiks oli luua sekventsiaalse lause- ja predikaatarvutuse tuletusülesannete lahendamise õpiprogrammi kaasaegne variant, mis võimaldaks ülesannete lahendamist praktikumides ja kodutöodes ning kontrolltööde läbiviimist. Töö raames loodi veebipõhine tuletamise keskkond, mis mainitud eesmärgid ka saavutas. Peale selle omab uus programm mitmeid eeliseid vana programmi ees, sealhulgas kaasaegne kasutajaliides, kahekümne aasta tagustest tehnoloogilistest piirangutest lahti saamine, mugav kasutamine, mis ei nõua failide edastamist, lemmade kasutamise võimalus ja tudengite lahendustest ülevaate omamine.

On veel mitmeid aspekte, mida annab loodud programmi juures täiendada. Näiteks võiks muuta tuletamise ajal kuvatavaid veateateid, et need oleks rohkem suunavad ning aitaks tudengil mõista, miks viga tekkis. Lisaks võiks programmile lisada automaatlahendaja, mis aitaks tudengitel tuletamise ajal õiget sammu teha. Peale selle võiks täiendada programmi poolt toetatavat signatuuri, võimaldamaks näiteks aritmeetika, rühmateooria vm aksiomaatiliste teooriate kasutamist.

Viited

- [1] R. Palm, R. Prank. *Sissejuhatus matemaatilise loogikasse*. 2004.
- [2] Pandora koduleht <http://www.doc.ic.ac.uk/pandora/>
- [3] K. Broda, J. Ma, G. Sinnadurai, A. Summers. *Pandora: A Reasoning Toolbox using Natural Deduction Style*. Logic journal of the IGPL vol. 15 no. 4, 2007.
- [4] Gerhard Gentzen. *Untersuchungen über das logische Schließen*. Mathematische Zeitschrift 39, lk 176–210, 1935.
- [5] Ian Hodkinson. Londoni Imperial College'i kursuse *140 Logic* slaidid.
http://www.doc.ic.ac.uk/~imh/teaching/140_logic/140.pdf
- [6] Stanford University. *Introduction to Logic*.
<https://class.coursera.org/intrologic-004>
- [7] Carnegie Mellon University. *Logic & Proofs*.
<https://oli.cmu.edu/courses/free-open/logic-proofs-course-details/>
- [8] AProS. <http://www.phil.cmu.edu/projects/apros/>
- [9] Oracle. *Enhancements in Java SE 8*.
<http://docs.oracle.com/javase/8/docs/technotes/guides/language/enhancements.html>

Lisad

Lähtekood

Tööga kaasas olevas arhiivis `axiom.zip` on kaust `axiom`, mis sisaldab programmi lähtekoodi.

Näidis ülesandekogud

Tööga kaasas olevas arhiivis `axiom.zip` on kaust `kogud`, milles on kolm näidis ülesandekogu – üks lausearvutuse, üks predikaatarvutuse ja üks võrdusega predikaatarvutuse teemal.

Litsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina **Dmitri Gabbasov** (06.04.1991)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose **Sekventsiaalse predikaatarvutuse õpiprogramm**, mille juhendaja on Rein Prank,
 - 1) reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 2) üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu autoriõiguse kahtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **18.05.2014**