

TARTU ÜLIKOOL  
MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse instituut

Informaatika eriala

**Vahur Vaiksaar**

**Harilike ja kümnenndmurdudega  
seotud ülesannete lahendamine  
interaktiivses õpikeskkonnas**

**Magistritöö**

Juhendaja: dotsent R. Prank

Autor: ..... "....." mai 2010

Juhendaja: ..... "....." mai 2010

TARTU 2010



# Sisukord

Sissejuhatus	5
<b>1 Teised avaldiste teisendamise keskkonnad</b>	<b>7</b>
<b>2 T-algebra interaktiivne õpikeskkond</b>	<b>12</b>
2.1 Avaldised T-algebras	12
2.2 Ülesande lahendamise akna kirjeldus	13
2.3 Üldine dialoogiskeem T-algebras	15
2.3.1 Reegli valimine ja avaldise osade märkimine	16
2.3.2 Reegli rakendamise tulemuse sisestamine	17
2.4 T-algebra siseehitusest	22
2.4.1 Lahendamise mootor	24
2.4.2 Nõu andmine ja erinevad kontrollid	25
<b>3 Harilike murdude ülesandetüübid matemaatika õpikutes ja T-algebras</b>	<b>29</b>
3.1 Ülesandetüübid matemaatika õpikutes	29
3.1.1 Harilike murdude taandamine ja laiendamine	29
3.1.2 Harilike murdude võrdlemine	30
3.1.3 Ühenimeliste murdude liitmine ja lahutamine	30
3.1.4 Samanimelise murdosaga segaarvude liitmine ja lahutamine	31
3.1.5 Erinimeliste murdude ja segaarvude liitmine ja lahutamine	31
3.1.6 Murdude teisendused	31
3.1.7 Harilike murdude korrutamine	32
3.1.8 Harilike murdude jagamine	32
3.1.9 Ülesanded kõigile tehetele murdudega	33
3.2 Disainitud reeglid T-algebras	33
3.2.1 Reegel <i>Taandan</i>	35
3.2.2 Reegel <i>Laiendan</i>	37
3.2.3 Reegel <i>Võtan täisosast</i>	39
3.2.4 Reegel <i>Liigmurd segaarvuks</i>	40
3.2.5 Reegel <i>Segaarv liigmurruks</i>	41
3.2.6 Reegel <i>Kümnendmurd harilikuks murruks</i>	42
3.2.7 Reegel <i>Harilik murd kümnendmurruks</i>	43

3.2.8	Reegel <i>Leian pöördarvu</i> . . . . .	44
3.2.9	Reegel <i>Liidan/lahutan arve</i> . . . . .	45
3.2.10	Reegel <i>Korrutan/jagan arve</i> . . . . .	52
3.2.11	Reegel <i>Võrdlen murde</i> . . . . .	54
3.2.12	Reegel <i>Kaotan liigsed plussmärgid</i> . . . . .	55
3.2.13	Reegel <i>Viin miinused murru ette</i> . . . . .	55
3.2.14	Reegel <i>Liidan/lahutan nulli</i> . . . . .	56
3.2.15	Reegel <i>Korrutan/jagan nulli</i> . . . . .	57
3.2.16	Reegel <i>Kaotan murru lugejaga null</i> . . . . .	57
3.2.17	Reegel <i>Kaotan ühe murru nimetajast</i> . . . . .	58
3.2.18	Reegel <i>Jagan ühega</i> . . . . .	59
3.2.19	Reegel <i>Korrutan ühega</i> . . . . .	60
3.2.20	Reegel <i>Jagan piisava täpsuseni</i> . . . . .	60
3.2.21	Reegel <i>Ümardan</i> . . . . .	61
3.3	Disainitud ülesandetüübid T-algebras . . . . .	62
3.3.1	Ülesandetüüp <i>Taanda harilik murd (taandumatu murruni)</i> . . . . .	63
3.3.2	Ülesandetüüp <i>Liida/lahuta erinimelised murrud ja kui võimalik, siis taanda vastus</i> . . . . .	64
3.3.3	Ülesandetüüp <i>Leia hariliku murru või segaarvu kümnendlähend ümardatud täpsuseni ... kohta peale koma</i> . . . . .	66
	<b>Kokkuvõte</b>	<b>67</b>
	<b>Summary</b>	<b>68</b>
	<b>Viited</b>	<b>70</b>

## Sissejuhatus

Käesoleva töö autor võttis aastatel 2004–2008 osa õpiprogrammi loomise projektist T-algebra. T-algebra on interaktiivne õpikeskkond, mis võimaldab samm-sammulist algebra ülesannete lahendamist neljas erinevas koolimatemaatika valdkonnas:

- arvuliste avaldiste väärtuste arvutamine;
- tehted murdudega;
- lineaarvõrrandite ja võrratuste ning lineaarsete võrrandisüsteemide lahendamine;
- hulkliikmete lihtsustamine.

T-algebrat ei arendatud ainult koolidele, see on disainitud ka uuringumaterjali koostamise tööriistaks. Keskkond võimaldab uurida järgmisi peamisi aspekte: töö kiirus arvutil (sh võrreldes tööga paberil), õpilaste tehtavad vead avaldiste teisendamisel, õpilaste tehtavad vead interaktiivses õpikeskkonnas võrreldes paberil tehtavate vigadega.

Õpikeskkonda arendati alates 2004-ndast aastast Tartu Ülikooli Arvutiteaduste Instituudi magistri- ja doktoriõppe üliõpilaste (Vahur Vaiksaar, Marina Issakova, Dmitri Lepp) poolt juhendajate (Rein Prank – projektijuht, Eno Tõnisson) käe all. Programmi sisu konsultandid olid matemaatika õpetajad (suure panuse andsid Mart ja Maire Oja) ja kooli õpikute autorid (Tiit Lepmann, Anu Palu). See versioon programmist on välja arendatud Tiigrihüppe Sihtasutuse poolt finantseeritava projektina.

Magistritöö autor disainis ja programmeeris projekti raames arvuliste avaldiste väärtuste arvutamise ja murdudega tehete osa ning andis panuse ka õpikeskkonna infrastruktuuri arendamisse. Käesolev magistritöö kirjeldab murdude teisendamise ülesandetüüpide ja reeglite realiseerimist.

Harilikke ja kümnendmurde sisaldavate avaldiste teisendamine on põhikooli matemaatika õppekava oluline osa. Teema omandamise käigus lahendatakse hulganisti tehnilisi ülesandeid, mis valmistavad paljudele õpilastele raskusi. Traditsioonilise paberil lahendamise korral on probleemiks vähene vahetu õpetajapoolne tagasiside ja õpetajate suhteliselt suur koormus. Protsessi saaks parandada arvutiseeritud keskkonna kasutuselevõtuga, mis võimaldaks lahendada ülesandeid sarnaselt paberil lahendamisele ning annaks õpilasele

töö käigus vahetut tagasisidet. Olemasolevaid süsteeme on küll palju, kuid ükski neist ei vasta täpselt kirjeldatud nõutele. Käesoleva töö eesmärk on disainida ja programmeerida interaktiivne õpikeskkond, mis võimaldaks harjutada murdude teema tehnilisi ülesandeid ja vastaks kirjeldatud nõuetele.

Töö koosneb kolmest peatükist. Esimene peatükk annab ülevaate olemasolevatest murdude teisendamise keskkondadest. Teine peatükk tutvustab  $T$ -algebra interaktiivset õpikeskkonda tuginedes  $T$ -algebra töögrupi publikatsioonidele ja loob aluse kolmandas peatükis toodud töö põhiosa,  $T$ -algebra murdude teema lahenduse, paremaks mõistmiseks.

# 1 Teised avaldiste teisendamise keskkonnad

See peatükk annab ülevaate teistest avaldiste teisendamise keskkondadest, mis võimaldavad murdude teema tehniliste ülesannete lahendamist harjutada. Kirjeldatavate keskkondade juures on välja toodud, mida teeb õpilane ülesande lahendamisel ja mida teeb keskkond.

Leidub mitmeid arvutialgebra süsteeme (Maple [11], Mathematica [12], WIRIS [25]), mis on mõeldud algebra ülesannetega tegelemiseks. Neid süsteeme ei ole enamasti arendatud spetsiaalselt hariduslikke eesmärke silmas pidades. Arvutialgebra süsteemide kasutamine näeb tavaliselt välja selline, et kasutaja sisestab avaldise ja käsu, mispeale süsteem teostab sisemiselt arvutused ning kuvab tulemuse. Arvutialgebra süsteemid ei ole disainitud näitama käsukaupa lahenduskäike ega lahenduskäiguga seonduvaid selgitusi. Arvutialgebra süsteemid ei ole mõeldud situatsiooniks, kus õpilane lahendab samm-sammuliselt ülesandeid, teeb vigu, vajab tagasisidet, nõuandeid jne.

Väga palju leidub blanketipõhiseid õpikeskkondi (WIMS [24], MathScore [14], Wisc-Online [26], Young Einstein Mathematics [27], Maths Connections [13] jmt), mis võimaldavad harjutada lihtsaid, staatilise vaatenäitena esitatavaid lahenduskäike. Süsteem kuvab ülesande ja kõigi lahendussammude sisestuskastid (lahendussamme võib olla rohkem kui üks). Õpilane täidab sisestuskastid ning kinnitab sisestuse. Süsteem kontrollib sisestust ning annab õpilasele tagasiside, kas sisestatud lahendus on õige või vale. Joonisel 1 on näide blanketist, kus iga ülesande lahendus koosneb ühest sammust. Joonisel 2 toodud blanketil nõutakse seevastu kahesammulist lahendust. Näide ebatraditsioonilisest, valikvastustega blanketist on toodud joonisel 3.

Kuna iga ülesandetüübi jaoks on spetsiaalne blankett, siis oskavad blanketipõhised süsteemid erinevaid sisulisi vigu diagnoosida. Blanketipõhiste õpikeskkondade puuduseks on see, et lahendada saab ainult lihtsaid, vähe teisendussamme nõudvaid staatilise blanketina esitatavaid ülesandeid. Samuti ei saa õpilane teha ise otsuseid teostatavate teisendussammude ja nende järjekorra kohta, kuna need on blanketiga fikseeritud – süsteem kirjutab üheselt lahendustee ette.

Osad blanketipõhised keskkonnad (nt WIMS) toetavad ka mitmesammulist ülesannete lahendamist. Ülesande lahendamine toimub järgmiselt: süsteem kuvab õpilasele esimese sammu blanketi, õpilane täidab selle, süsteem kuvab järgmise sammu blanketi jne. Kui õpilane (vahe)tulemuse sisestamisel

Add each pair of fractions and simplify the result.

1.  $\frac{2}{3} + \frac{1}{2}$    $\frac{\text{ } \text{ }}{\text{ } \text{ }}$       2.  $\frac{1}{3} + \frac{2}{2}$    $\frac{\text{ } \text{ }}{\text{ } \text{ }}$       3.  $\frac{1}{2} + \frac{1}{3}$    $\frac{\text{ } \text{ }}{\text{ } \text{ }}$

4.  $\frac{1}{2} + \frac{2}{3}$    $\frac{\text{ } \text{ }}{\text{ } \text{ }}$

**Joonis 1.** Ühesammulise lahenduskäiguga harilike murdude liitmine õpikeskkonnas MathScore

Teisenda segaarv liigmurruks ja võimalusel taanda!

$$3 \frac{\text{ } \text{ }}{\text{ } \text{ }} = \frac{\text{ } \text{ }}{\text{ } \text{ }} = \frac{\text{ } \text{ }}{\text{ } \text{ }}$$

**Joonis 2.** Kahesammulise lahenduskäiguga blankett Limbergi loodud veebilehel [9]

Problem 1: (Drag the numbers to the correct box.)

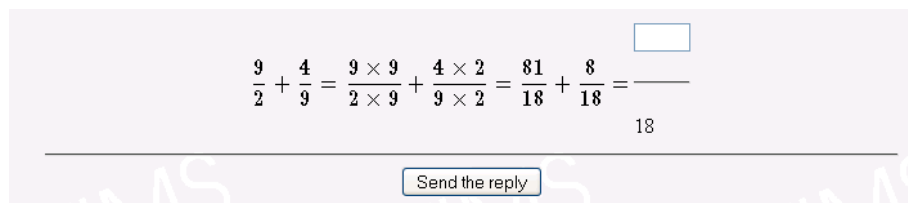
$$\begin{array}{r} \square \\ 8 \frac{5}{13} \square \\ - 4 \frac{7}{13} \\ \hline \square \frac{\square}{13} \end{array}$$

$$\begin{array}{r} 15 \\ 3 \\ 11 \\ 18 \\ 4 \\ 7 \end{array}$$

**Joonis 3.** Samanimeliste segaarvude lahutamise ülesanne Wisc-Online blanketil (kastidesse tulemuse sisestamise asemel on paremal ääres toodud arvud, mille hulgast tuleb teha valik ja need õigetesse kastidesse lohistada)



eksib, kuvab programm vastava veateate ja nõuab vea parandamist. Järgmise lahendussammuga saab edasi minna alles siis, kui eelmise sammu sisestus on korrektne. Lähteülesanne ja juba tehtud sammud on lahendamise ajal õpilasele nähtavad (vt joonis 4). Sellisel viisil ülesannete lahendamine sarnaneb tööle allpool kirjeldatavates sisendipõhistes keskkondades. Paraku on blanke tipõhised mitmesammulised ülesanded vähe paindlikud, sisaldades fikseeritud arvu fikseeritud järjekorras tehtavaid samme.



$$\frac{9}{2} + \frac{4}{9} = \frac{9 \times 9}{2 \times 9} + \frac{4 \times 2}{9 \times 2} = \frac{81}{18} + \frac{8}{18} = \frac{\quad}{18}$$

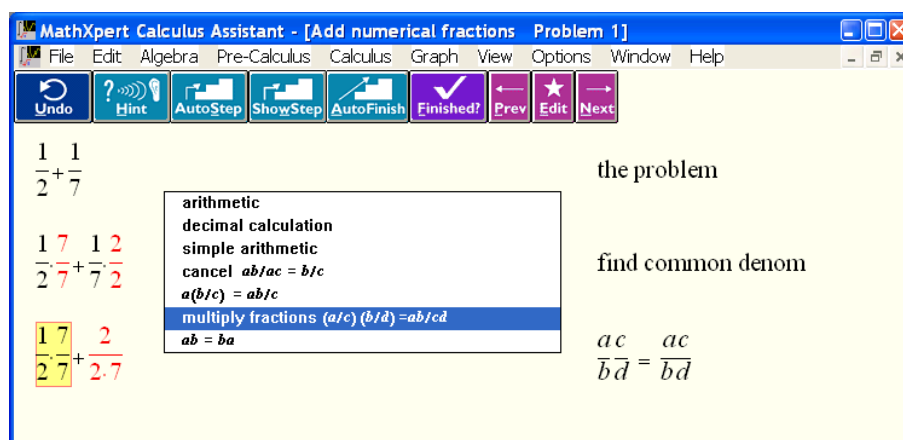
**Joonis 4.** Harilike murdude liitmine WIMSis, hetkel on teostamisel kolmas lahendussamm

Õpilase jaoks kõige huvitavamaid võimalusi sisaldavad interaktiivsed õpikeskkonnad, mis toetavad ülesannete samm-sammulist lahendamist. Dialoogi tüübi põhjal võib selliseid õpikeskkondi liigitada järgmiselt:

- dialoogita keskkonnad;
- reeglipõhised keskkonnad;
- sisendipõhised keskkonnad.

*Dialoogita keskkondades* (Bagatrix [3]) saab õpilane sisestada lähteavaldise, millele süsteem oskab samm-sammulise lahenduskäigu genereerida ning selgitada iga teostatud lahendussammu. Õpilane lahenduskäiku ei mõjuta ning on passiivse õppija rollis. Osa keskkondi, nt Algebrator [1], kombineerivad dialoogita ja sisendipõhist lähenemist, kuid nende põhirõhk on siiski dialoogita lahendussammude genereerimisel.

*Reeglipõhised keskkonnad* (MathXpert [4, 15], LiveMath [10]) toimivad põhimõttel, et õpilane valib igal lahendussammul teisendusreegli ja mõnel juhul ka alamavaldise ning süsteem teostab seejärel teisenduse. Erinevalt arvutialgebra süsteemidest lahendatakse ülesandeid koolis tehtavate sammudega. Teisendusreegli ja alamavaldise valimise järjekord sõltub konkreetsest

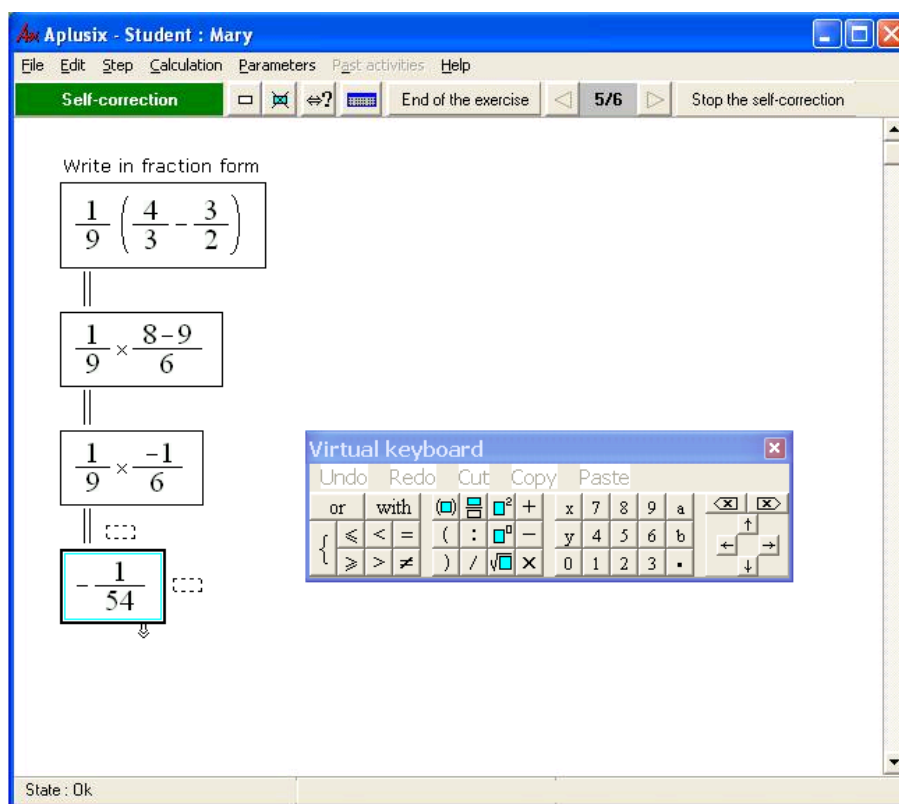


**Joonis 5.** Harilike murdude liitmine reeglipõhises õpikeskkonnas MathXpert (seni on teostatud kaks lahendussammu, kolmanda sammu teostamiseks on valitud alamavaldis, aga sobiv reegel on veel valimata)

süsteemist. Leidub süsteeme, kus kõigepealt tuleb valida alamavaldis ja süsteem pakub seejärel sobivaid teisendusreegleid, teistes süsteemides peab õpilane aga kõigepealt teisendusreegli valima. Reeglipõhistes keskkondades õpilane õpib ja harjutab lahendusalgoritmi, kuid algoritmi sammude teostamist õpib ta passiivselt, kuna teisendusreegli rakendamised teostab süsteem. Kuna paberil lahendades tekib enamik vigu just lahendussammu teostamisel, siis suurt osa tüüpvigude pole reeglipõhistes keskkondades lihtsalt võimalik teha. Õpilasel on võimalik eksida ainult alamavaldis ja teisendusreegli valimisel. Näide ülesande lahendamisel reeglipõhises keskkonnas on toodud joonisel 5.

Leidub vähe reeglipõhiseid keskkondi, milles oleks murdudega manipuleerimise detailsusastmega reegleid. Enamik reeglipõhiseid keskkondi (sh nt LiveMath) on disainitud üliõpilaste või teadlaste töövahendiks.

*Sisendipõhistes keskkondades* (Aplusix [2, 16]) sarnaneb õpilase töö kõige enam traditsioonilisele, paberi ja pliiatsi abil tehtavale tööle. Iga teisendusamm koosneb peamiselt järgmise rea sisestamisest (vt joonis 6). Õpilane saab küll töötada vastavalt lahendusalgoritmile ja teostada igal lahendussammul ühe algoritmi sammu, kuid alati on võimalus kirjutada järgmise sammu tulemuseks süntaktilisi piiranguid arvestades ükskõik milline avaldis. Sellised süsteemid ei ole tavaliselt teadlikud erinevatest ülesandetüüpidest ning neile vastavatest lahendusalgoritmidest ja seega ei kontrolli, kas õpilane töötab vastavalt lahendusalgoritmile või mitte. Õpilane võib teha suvalisi vigu ning programm kontrollib ainult sisestatu süntaktilist korrektsust ja samaväärsust



**Joonis 6.** Harilike murde sisaldava avaldise lihtsustamine õpikeskkonnas Aplux

eelmise sammu avaldisega. Süsteem on õpetajaga samas situatsioonis, kuna tal ei ole ilmutatud teadmist lahendussammul teostatud operatsioonidest ega operandidest. Sellises situatsioonis on vea korral raske õpilasele veateatest “avaldised ei ole ekvivalentsed” täpsemat diagnoosi anda ning seda enamik sisendipõhiseid keskkondi ka ei tee.

T-algebras oleme püüdnud integreerida reegli- ja sisendipõhise dialoogi, võimaldades seeläbi õppida nii lahendusalgorithmi kui ka konkreetsete lahendussammude teostamist ning toetades seejuures, erinevalt sisendipõhistest keskkondadest, täpset veadiagnostikat tulemuse sisestamisel.

## 2 T-algebra interaktiivne õpikeskkond

T-algebra koosneb kahest programmist, millest üks on õpilastele ülesannete lahendamiseks ja teine õpetajatele ülesannete koostamiseks. See peatükk annab ülevaate õpilase programmi ehitusest ja üldisest dialoogiskeemist tuginedes T-algebra töögrupi publikatsioonidele [5, 6, 7, 19, 20, 21, 22]

### 2.1 Avaldised T-algebras

Peamine objekt, millega programm opereerib, on algebraline avaldis. See jaotis annab ülevaate, millised algebralised avaldised on programmis lubatud st millised avaldised loetakse korrektseks.

Algebraline avaldis defineeritakse T-algebras järgmiselt: elementaaravaldised on täisarvud, kümnendmurrud (kümnenderaldajana koma või punkt), harilikud murrud, segaarvud ja muutujad (väiksed tähed  $a..z$ ). Avaldised koostatakse rekursiivselt rakendades erinevad operatsioonid (unaarsed  $+$ ,  $-$ , binaarsed  $+$ ,  $-$ ,  $\cdot$ ,  $:$ , astendamine ( $\square^{\blacksquare}$  ja  $\blacksquare^{\square}$ ), murrujoonega jagamine ( $\frac{\blacksquare}{\square}$  ja  $\frac{\square}{\blacksquare}$ ), grupeerivad sümbolid (ümarsulud  $()$  ja kandilised sulud  $[]$ ). Lisaks toetab programm järgmisi keerulisemaid avaldisi: lineaarvõrrandid (märk  $=$ ), lineaarvõrratused (märgid  $<$   $\leq$   $\geq$   $>$ ) ja lineaarvõrrandite süsteemid (märk  $\{$ ).

Programmis peavad avaldised olema matemaatiliselt korrektsed ja sisaldama kombinatsioone ülalmainitud sümbolitest. Siin on mõned näited korrektsetest avaldistest:

- $\frac{1}{2} - 2\frac{2}{3}$ ;
- $x^2(1+x)^2$ ;
- $2x - 2 \leq 3 - x$ .

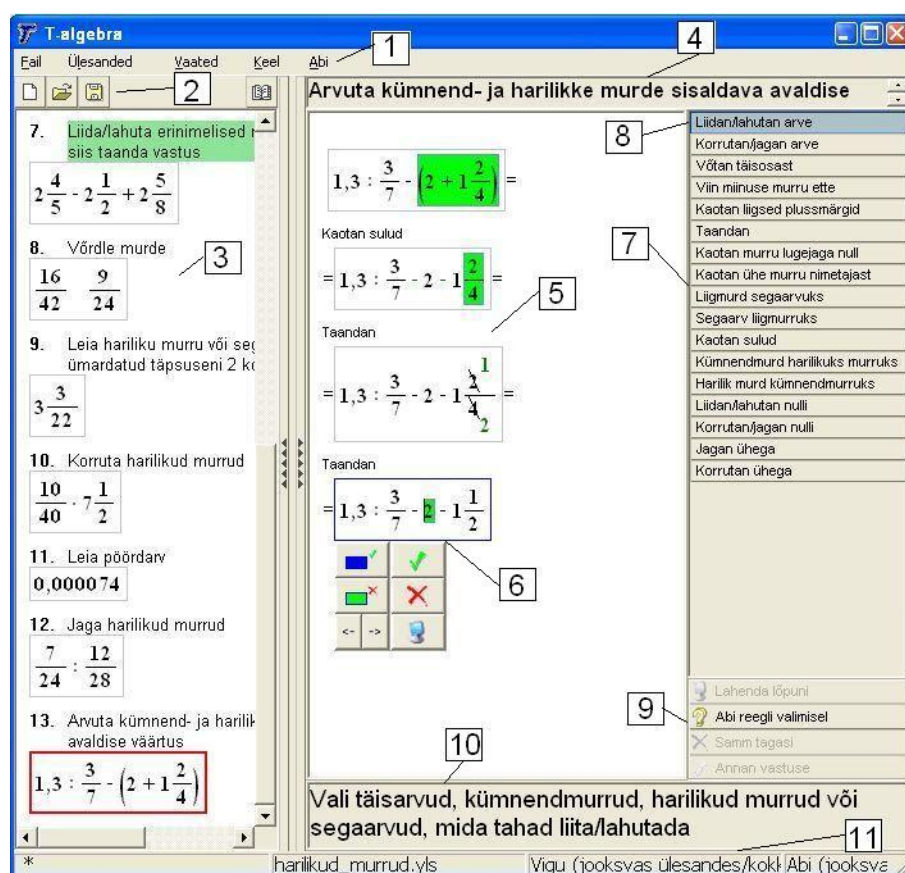
Järgmisi avaldisi loeb programm ebakorrekseteks:

- $3\frac{x}{3}$ , sest muutuja ei tohi esineda segaarvus;
- $a2b + 2bc3av$ , sest üksliikmete korrutamisel on korrutamise märk kohustuslik ja konstandid ei tohi üksliikmete korrutises muutujate vahel paikneda.

Programm ei sea avaldistele kvantitatiivseid piiranguid ning toetab mitmekorruselisi murde ja astendamisi. Ülesandetüübid võivad siiski avaldistele piiranguid seada. Näiteks harilike murdude korrutamisel peavad tegurid olema täisarvud, harilikud murrud või segaarvud.

## 2.2 Ülesande lahendamise akna kirjeldus

Joonisel 7 on toodud T-algebra ülesande lahendamise aken.



Joonis 7. T-algebra ülesande lahendamise aken

Ülesande lahendamise aken on jagatud kaheks loogiliseks osaks. Vasakpoolne osa sisaldab välja ülesannete kogu kuvamiseks. Aknas näidatakse avaldise ja ülesande tekste koos ülesande järjekorranumbriga selles failis. Lisaks kuvatakse ülesande lahenduse olek – lahendatud ülesanne on tähistatud rohelise taustaga. Hetkel lahendatav ülesanne kuvatakse punases kastis.

Akna parempoolne osa sisaldab hetkel lahendatava ülesande lahendussamme, virtuaalset klaviatuuri, menüüd võimalikeks tegevusteks ja juhendeid õpilasele selleks konkreetseks situatsiooniks.

Akna peamised komponendid on järgmised.

1. Programmi menüüriba, mis võimaldab tööd failidega (Uus, Ava, Salvesta, Salvesta kui, Sulge, Lõpeta) ja ülesannetega (Eelmine, Järgmine), avada lisaaknaid (näiteks vaadata vigade loendajaid – kategoriseeritud vea tüüpide järgi, või kõigi tehtud vigade nimekirja) või valida töötamise keelt.
2. Nupud, mis dubleerivad mõningaid menüüriba valikuid (Uus, Ava, Salvesta kui).
3. Lahendatavate ülesannete nimekiri, mis kajastab ühtlasi ülesannete lahendamise tulemust. Ülesannete nimekirja saab ära peita ja kasutada kogu akent lahenduse vaatamiseks.
4. Ülesande tekst.
5. Valitud ülesande lahenduse käik – lahendamisel teostatud sammud.
6. Viimane avaldis koos virtuaalse klaviatuuriga operandide märkimiseks. Joonis 7 seda hetkel ei kajasta, aga tulemuse sisestamise ajal on virtuaalsel klaviatuuril ka avaldise sisestamise nupud.
7. Reeglite menüü.
8. Valitud reegel (erineb teistest reeglitest taustavärvi poolest).
9. Nupud abi küsimiseks, lahenduse sammude tagasi võtmiseks ja vastuse andmiseks.
10. Juhendid ülesande lahenduse protsessi toetamiseks (osutavad, mida õpilane peaks järgmiseks tegema: valima järgmisena rakendatava reegli, märkima avaldise mõned osad, midagi sisestama jne).
11. Olekuriba, mis näitab kasutaja ja avatud ülesandekogu kohta informatsiooni.

## 2.3 Üldine dialoogiskeem T-algebras

Iga lahendussamm koosneb järgmistest etappidest:

1. teisendusreegli valimine (tegevus);
2. avaldise osade märkimine (objekt);
3. valitud reegli rakendamise tulemuse sisestamine (sisend).

Edaspidi nimetame seda skeemi kolme toodud etapi põhjal Tegevus-Objekt-Sisend skeemiks.

Joonisel 7 esitatud ülesande lahendamise akna osa (5) näitab kümnend- ja harilikke murde sisaldava avaldise  $1,3 : \frac{3}{7} - (2 + 1\frac{2}{4})$  väärtuse arvutamise käiku. Lahendus ei ole veel täielik, aga mõned sammud on juba tehtud. Esimesel sammul on valitud reegel *Kaotan sulud* ja reegli rakendamiseks on valitud lähteülesandes esinev sulgavaldis. Seejärel on sisestatud sulgavaldises sisalduva summa liikmetele õiged märgid, arvestades sulgavaldises ees olevat märki. Teise sammuna on samal viisil taandatud segaarvu murdosa. Kolmanda sammu tegemine on hetkel pooleli, reegel on valitud ja üks operand märgitud.

Kui lahendada seda ülesannet paberil edasi, siis õpilane uurib esmalt avaldist. Ta otsustab teostada lahutamise. Seejärel ta valib lahutamistehte operandid ning kirjutab lahutamise järel tekkinud avaldise järgmisele reale. Rakendades reeglit *Liidan/lahutan arve* järgib programm põhimõtteliselt sama tegevuste skeemi. Vastav lahendussamm koosneb järgmisest kolmest etapist (millest esimene on joonisel 7 juba teostatud).

1. Teisendusreegli valimine – õpilane valib nimekirjast reegli arvude lahutamiseks – programm lubab valida kõiki reegleid, kontrollimata, kas valitud teisendus on lahenduse hetkeseisu arvestades võimalik teostada.
2. Avaldise osade märkimine – õpilane märgib hiirt kasutades lahutamistehte operandid (esimene operand on juba märgitud) – programm kontrollib, et valitud avaldise osad oleks vahe liikmed ja lahutamist saaks teostada (st operandid oleks sama vahe liikmed). Õpilane ei pea kõiki sobivaid operande korraga märkima – minimaalselt tuleb märkida kaks vahe liiget. Esimene ja teine etapp moodustavad lahendussammu esimese, ühtselt kinnitatava osa ning nende teostamise järjekord ei ole oluline – võib alustada nii teisendusreegli valimisest kui ka avaldise osade märkimisest.

3. Valitud reegli rakendamise tulemuse sisestamine – programm kopeerib avaldise märkimata jäänud osad järgmisele reale ja palub õpilasel sisestada, sõltuvalt lahendamise režiimist, valitud vahe tulemuse või selle osad. Kuna õpilane peab märgitud osaavaldistele reegli rakendamise tulemuse sisestama, siis on sellel etapil eksimise tõenäosus kõige suurem. Parema veadiagnostika saavutamiseks kavandasime keerulise mate reeglite jaoks kolm sisestusrežiimi. Erinevad sisestusrežiimid on kirjeldatud detailsemalt jaotises 2.3.2.

See näide peaks andma idee, kuidas õpilase ja programmi tegevused omavahel seotud on – mida ja millal programmi poolt kontrollitakse. Kui lahendamise käigus kuvatakse mõnda etappi kontrollides veateade, siis peab õpilane vea kas ise ära parandama või laskma programmil seda teha ja alles siis saab järgmise etapiga edasi minna. Näiteks kui õpilane eksib vahe arvutamisel ning sisestab vale tulemuse, siis programm tuvastab selle standardse vea ning kuvab sobiva veateate. Programm kuvab õpilase iga tegevuse jaoks spetsiifilised juhised (“Vali tegevus”, “Vali täisarvud, kümnendmurrud, harilikud murrud või segaarvud, mida tahad liita/lahutada”, jne). Õpilane võib sammu igal ajahetkel katkestada. Igal etapil on võimalik programmilt abi küsida ja lasta programmil konkreetsed etapid automaatselt teostada. Õpilane võib tulemuse sisestamise etapil vajutada spetsiaalset arvuti pildiga nuppu, mispeale programm täidab kastid õigete vastustega. Sama nuppu saab kasutada ka operandide märkimisel – programm valib sobivad operandid ise. Vajutades nuppu *Abi reegli valimisel* saab õpilane programmilt enne igat sammu küsida, millist reeglit tuleks lahendusalgoritmi arvestades järgmisena rakendada. Sama nupp näitab, kas ülesanne on lahendatud.

Kuna T-algebras ülesande lahendamisel menüüs kuvatav reeglite komplekt koosneb koolis õpetatavatest lahendusalgoritmi sammudest, siis sarnaneb lahendus paberi ja pliiatsi abil tehtavale lahendusele. T-algebra säilitab ja kuvab ainult lõplikku ja korrektset lahendusteed. Näiteks kui õpilane teostab vale või ebavajaliku sammu ja võtab seejärel selle sammu tagasi, siis lahenduskäiku vaadates õpetaja tagasivõetud sammu ei näe.

### 2.3.1 Reegli valimine ja avaldise osade märkimine






Esimese kahe etapi järjekord ei ole programmi poolt fikseeritud. Õpilane võib operandid märkida enne, pärast või isegi nii enne kui pärast reegli valimist. Programm kontrollib alles pärast nende kahe etapi kinnitamist, kas märgitud

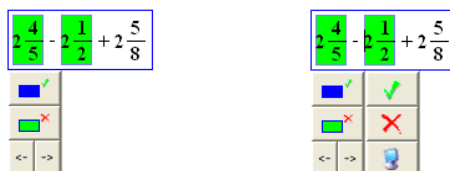


avaldised on valitud reegli jaoks sobilikud.

Menüüs kuvatav reeglite komplekt sõltub ülesandetüübist. Harilike murdudega seotud reeglitest on põhjalikum ülevaade peatükis 3.

Erinevalt paljudest teistest programmidest nõuab T-algebra diagnostilistel eesmärkidel operandide täpset märkimist. Näiteks reegli *Liidan/lahutan arve* rakendamiseks peab õpilane märkima need summa (vahe) liikmed, mida tegelikult liidetakse (lahutatakse). T-algebra avaldise redaktor on ehitatud selle arvestusega, et õpilane saaks märkida rohkem kui ühe osaavaldise (vt joonis 8). Programm lubab osaavaldise märkida nii sellele eelneva liitmise, lahutamise, korrutamise või jagamise märgiga kui ka ilma eelneva märgita, aga loeb märgi alati operandi osaks. Kui valitavad osaavaldised paiknevad kõrvuti, lubab programm need ühe suure tükina märkida, jagades edasiseks töötluks need ikkagi osadeks.

Avaldise osa märkimiseks tuleb esmalt valida avaldise osa (kas hiirega või klaviatuurilt) nagu tavalises tekstiredaktoris ja seejärel vajutada nuppu . Märgistuse eemaldamiseks on nupp . Nupud   on märgitud osade vahel liikumiseks. Kui õpilane on märkimise lõpetanud ja reegli valinud, peab ta esimesed kaks etappi nuppu  vajutades kinnitama.



**Joonis 8.** T-algebra programmi avaldise redaktor märkimise režiimis (vasakul – reegel pole veel valitud, paremal – reegel on juba valitud)

### 2.3.2 Reegli rakendamise tulemuse sisestamine

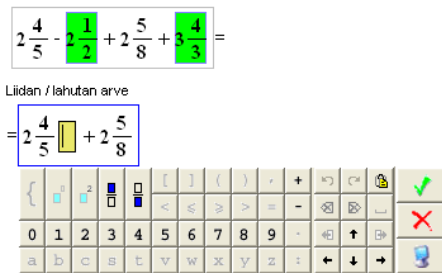
Proovisime T-algebras kavandada kõigi reeglite jaoks kolm üsna ühetaolist ja standardset sisestamise režiimi. Need kolm režiimi on *vaba* sisestamine, *struktuurne* sisestamine ja *osaline* sisestamine. Vaba sisestamise režiim on lihtsalt mõistetav (sarnane paberil töötamisele) ja toetatud iga reegli korral. Struktuurne ja osaline sisestamise režiim on spetsiifilisemad, programm annab ette vastuse struktuuri või täidab isegi osa vastusest ise.

Iga lahendussammu kolmandal etapil (sisestamisel) peab õpilane sisestama selle avaldise osa, mis moodustab eelnevalt valitud operatsiooni tulemuse. Programm genereerib valitud reeglit ja märgitud osaavaldisi arvestades

järgmisele reale tulemusavaldise, jättes selles täitmata konkreetsed olulised kohad. Paberil töötades peavad õpilased ise kogu tulemusavaldise kirjutama. Sellest tulenevalt üritavad nad vähendada rutiinset ümberkirjutamist tehes mitu teisendust korraga. Programm lihtsustab õpilaste tööd, kopeerides muutumatud osad. Õpilased peavad sisestama ainult muutuvad osad. Igal sammul saab teha ainult ühe teisenduse. Selline lähenemine lihtsustab programmi jaoks lahenduse kontrollimist ja annab õpetajale õpilase lahendusest parema ülevaate. Tulemused saab sisestada klaviatuurilt või virtuaalselt klaviatuurilt.

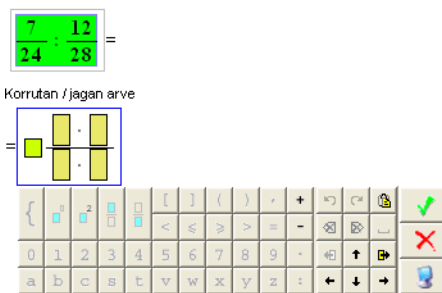
Avaldise osad, mida õpilane sisestama peab, on esile tõstetud kollaste kastidena. Kasutaja poolt defineeritavate osade kuju ja arv sõltub valitud reeglist, märgitud osadest ja režiimist. Programm kaitseb sisestamise ajal avaldise teisi osasid muutmise eest – ainult esile tõstetud asukohad avaldises on muudetavad. See lihtsustab programmil lahenduse kontrollimist, lisaks eelmise ja uue avaldise ekvivalentsuse kontrollile võimaldab see kontrollida eraldi sisestavate osade korrektsust, tõhustades sellega programmi üldist vigadele reageerimist. Sisestamise režiimi valib õpetaja ülesande koostamise ajal.

**Kolm sisestamise režiimi** *Vabas* sisestamise režiimis genereerib programm järgmisele reale kopeeritud avaldise sisse märgistatud osade asemel ühe (mõnede harilike murdude ja võrranditega seotud reeglite korral kaks) sisestuskasti (vt joonis 9). Õpilane peab sellesse kasti sisestama avaldise, mis asendab kogu eelmisel real märgitud osa. Kuigi režiimi nimi on “vaba sisestamine”, piirab programm ikkagi mõningal määral sisestatavate sümbolite hulka. Redaktor annab õpilasele sisestamisel vabaduse, aga pärast sisestust kontrollib programm lisaks avaldise süntaktilisele korrektsusele ja ekvivalentsusele eelmise avaldisega ka reegli rakendamise korrektsust. Näiteks kui õpilane üritab taandamise reegli abil murdu laiendada ja teeb seda korrektselt, siis kogu avaldis on küll ekvivalentne eelmisega, aga programm loeb selle ikkagi veaks ning kuvab vastava veateate.



**Joonis 9.** Vastuse sisestamine (vabas sisestamise režiimis)

*Struktuurses* sisestamise režiimis kasutab programm informatsiooni rakendatava reegli ja märgitud operandide kohta ning ennustab selle põhjal nõutava sisendi struktuuri, kasutades selleks erinevaid sisendkaste märke, koefitsientide, muutujate, astmenäitajate jne jaoks (vt joonis 10).



**Joonis 10.** Struktuurne sisestamine

Kastide suurus ja asukoht peaks koheselt kasutajale osutama, kuhu ja mida sisestada. Selles režiimis on kastidesse sisestatavate sümbolite hulk piiratud. Kui kursor on mõnes sisendkastis, siis mittekasutatavad nupud on virtuaalsel klaviatuuril passiivsed ning tavalise klaviatuuri vastavad nupud ei tööta. Näiteks joonisel 10, kus on valitud reegel *Korrutan/jagan arve* ja märgitud jagatis, pakub programm jagatise ühele murrujoonele kirjutamise struktuuri. Esimene kast on märgi jaoks, ülejäänud kastid on jagatava ja jagaja ühisele murrujoonele kirjutamiseks. Erijuhtudel on lubatud mõned kastid tühjaks jätta. Näiteks joonisel 10 toodud näite korral võib märgi sisestamata jätta. Kui lugejas või nimetaajas oleks mõni teguritest 1, siis selle võiks samuti kirjutamata jätta.

Kui kasutaja on sisestamise lõpetanud, kontrollib programm, kas uus avaldis on eelmisega ekvivalentne ja kas sisestatud osad on ekvivalentsed arvuti poolt arvutatud vastavate osadega.

Struktuurne sisestamise režiim on reeglispetsiifiline, igal reeglil on oma tulemusavaldise sisestamise muster. Osutus, et osade reeglite jaoks on see režiim kasutu. Näiteks oleks kasutu pakkuda struktuuri reegli *Kaotan sulud* rakendamisel, kuna tulemus muutuvad ainult märgid.

Kolmas režiim (*osaline* sisestamine) on teise režiimi lihtsustatud kuju, kus programm täidab mõned kastid ise. Erijuhul võib see režiim pakkuda välja struktuurse režiimiga võrreldes tulemusavaldise teistsuguse või täpsema struktuuri (vt reegli *Liidan/lahutan arve* kirjeldust peatükis 3).

Mõnede reeglite juures oleme piirdunud ühe või kahe sisestamise režiimiga – kolme erineva režiimi toetamine pole vajalikuks osutunud (vt reeglite kirjeldusi peatükis 3).

**Täiendav sisend** T-algebra reegleid kavandades leidsime, et mõnede reeglite korral on raske kõiki õpilase poolt tehtavaid otsuseid Tegevus-Objekt-Sisend dialoogi kaudu väljendada. Seetõttu laiendasime mõnede reeglite dialoogi kahe võimaliku etapiga:

- reeglispetsiifilise täiendava informatsiooni sisestamine (täiendav informatsioon tähendab siin seda, et reegli rakendamise tulemus pole operandidega üheselt määratud);
- vahetulemuse sisestamine.

Iga reegel võib sõltuvalt režiimist kasutada neist korraga ühte või mõlemat.

Vaatleme lisaetappe murdude liitmise näitel. Eesti õpikud soovivad kirjutada erinimeliste harilike murdude liitmise järgmiselt.

$$\frac{1}{2} + \frac{1}{3} + \frac{1}{6} = \frac{3 + 2 + 1}{6}$$

Õpilased leiavad siin esmalt liidetavate murdude ühise nimetaja ja kirjutavad selle võrdusmärgi taha tulemuseks olevasse murdu. Seejärel arvutavad nad laiendajad ja kirjutavad need iga liidetava kohale ning lõpuks rehken-davad tulemuseks oleva murru lugeja. Õpilased võivad eksida kõigil kolmel kirjeldatud sammul. Kuna me tahame T-algebra detailsemate töörežiimide korral õpilasele iga vigase tulemuse kohta kohe ja võimalikult täpse veateate anda, küsib programm toodud näite korral õpilaselt sisendit kolmes etapis.

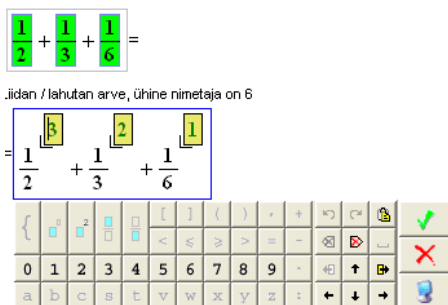
Täiendavat informatsiooni vajavate reeglite jaoks lisasime vastava etapi. Kui reegli operandid on märgitud, kontrollib programm, kas reegel on neile

rakendatav ja kuvab seejärel kasutajale täiendava informatsiooni küsimise akna. Kui kasutaja on sisestuse kinnitanud, kontrollib programm sisestatud informatsiooni korrektsust. Kui vigu ei leidu, võib järgneda teine lisaetapp või tavaline tulemuse sisestamine. Murdude liitmisel peab õpilane sisestama ühe numbri – valitud murdude ühise nimetaja (vt joonis 11).



**Joonis 11.** Reeglispetsiifilise täiendava informatsiooni sisestamine erinimeliste harilike murdude liitmisel

Murdude liitmisel üritasime programmi dialoogi laiendades järgida paberi ja pliiatsi lahedustes kasutatavat mustrit: esiteks sisestatakse eraldi aknas ühine nimetaja, seejärel sisestatakse murdudele laiendajad (vt joonis 12) ning viimase sammuna sisestatakse lõpptulemuse liikmed. Kuna tahtsime esialgse avaldise koos märgitud operandidega muutumatuna hoida, siis kopeerib programm peale ühise nimetaja sisestamist avaldise uuele reale ja paigutab murdude kohale laiendajate sisestamise kastid. Siin rakenduvad struktuurse ja osalise sisestamise režiimiga sarnased kitsendused – kastidesse saab sisestada ainult numbreid.



**Joonis 12.** Vahetulemuse sisestamine erinimeliste murdude liitmisel

Pärast vahetulemuse sisestamist kontrollib programm sisestatud osade korrektsust. Vea korral annab süsteem õpilasele sobiva veateate ja laseb enne jätkamist vea ära parandada. Kui vigu ei esinenud, laseb programm õpilasel sõltuvalt kehtivast režiimist vastuse sisestada.

## 2.4 T-algebra siseehitusest

Üks võimalus matemaatika õpisüsteemi ehitamiseks on tugineda olemasoleva arvutialgebra süsteemi poolt pakutavatele teenustele. Alternatiivne lähene mine on realiseerida kogu vajalik funktsionaalsus õpisüsteemi siseselt. Usume sarnaselt Beesoniga, et "... kui alustame haridusliku eesmärgiga ja sõnastame mõned lihtsad kavandamise põhimõtted, mis rohkem või vähem ilmselt tulenevad sellest eesmärgist, siis nendel põhimõtetel on mõju süsteemi arvutusliku tuumani, nii et mõne olemasoleva arvutamise süsteemi jäigale liidesele tuginedes on ideaalseid tulemusi raske saavutada" [16, lk 90]. Seepärast realiseerisime arvutusliku tuuma ise ning usume, et see on T-algebra eeliseks.

T-algebras on kirjeldatud ülesannete tüübid, lahendamise algoritmid ja algoritmi sammud (reeglid). Lisaks ülesannetele, mis baseeruvad mitmesammulisel algoritmil või eeskirjal (nt erinimeliste harilike murdude liitmine), sisaldab T-algebra ülesandeid algoritmi üksikute sammude rakendamise kohta (nt taandamine). T-algebra oskab ülesannetele samm-sammulisi, paberi ja pliiatsi abil tehtud lahendustele sarnaseid lahendusi genereerida. Programm lahendab ülesandeid kasutades kavandatud reeglipõhist dialoogi: valib ülesannet ja hetkeavaldist silmas pidades sobiva teisendusreegli, märgib operandid ning asendab operandid reegli rakendamise tulemusega.

Programm teab iga ülesandetüübi algoritmi. T-algebras on algoritmid realiseeritud järjestatud reeglite nimekirjana. Enamik ülesandeid sisaldab vajalikke baasreegleid avaldise lihtsustamiseks (*Kaotan liigsed plussmärgid, Liidan/lahutan nulli, Korrutan/jagan ühega* jne). Need ei vasta küll õpikutes kirjeldatud lahendussammudele, aga õpilane võib neid vajalikul hetkel parlahenduses kasutada. Teiseks sisaldavad algoritmid vajalikke murdudega seotud reegleid (*Taandan, Laiendan, Liidan/lahutan arve, Korrutan/jagan arve* jne). Murdude ülesanded sellega piirduvad. Kõrgema taseme ülesannete (lineaarvõrrandite ja võrratuste ning lineaarsete võrrandisüsteemide lahendamine, hulkliikmete lihtsustamine) lahendusalgoritmid sisaldavad omakorda murdude reegleid baasreeglitena ja lisanduvad uued spetsiifilised reeglid. Selline püramiidjas reeglite kasutamine tuleneb sellest, et uue teema ülesan-

ded baseeruvad peaaegu alati varem õpitul. Järgnevalt on esitatud erinimeliste murdude ja segaarvude liitmise ja lahutamise ülesande lahendusalgoritm (nimekiri reeglitest):

- reegel *Kaotan sulud*;
- reegel *Kaotan liigsed plussmärgid*;
- reegel *Laenan*;
- reegel *Liidan/lahutan arve*;
- reegel *Taandan*;
- reegel *Viin miinuse murru ette*;
- reegel *Liigmurd segaarvuks*;
- lihtsustamise reeglid (*Kaotan murru lugejaga null, Kaotan ühe murru nimetajast*).

T-algebra lahendab erinimeliste harilike murdude ja segaarvude liitmise ja lahutamise ülesande vastavalt õpikutes soovitatud algoritmile. See ei ole siiski ainus viis ülesannet lahendada. Programm aktsepteerib ka kõiki teisi õpilase tekitatud lahendusteid. T-algebra oskab ülesannet lahendada mitte ainult lähtuvalt antud tüübi jaoks lubatud esialgsest avaldisest, vaid ka lähtuvalt suvalisest pooleli olevast lahenduskäigust. Õpilane võib teha mõned lahendusammud ja programm oskab seejärel ikkagi lahenduse lõpetada. Joonisel 13 on toodud näide programmi poolt genereeritud lahenduskäigust vabas sisestamise režiimis.

$$2\frac{3}{8} - \left(\frac{5}{8} + \frac{1}{4}\right) =$$

Kaotan sulud

$$= 2\frac{3}{8} - \frac{5}{8} - \frac{1}{4} =$$

Liidan / lahutan arve

$$= 2\frac{3-5-2}{8} =$$

Võtan täisosast

$$= 1\frac{8+3-5-2}{8} =$$

Liidan / lahutan arve

$$= 1\frac{4}{8} =$$

Taandan

$$= 1\frac{1}{2} =$$

Taandan

$$= 1\frac{1}{2}$$

Vastus:

$$1\frac{1}{2}$$

**Joonis 13.** T-algebra poolt erinimeliste harilike murdude ja segaarvude liitmise ülesandele genereeritud lahenduskäik

### 2.4.1 Lahendamise mootor

Järjekord algoritmis on väga oluline, sest programm uurib reeglite nimekirja algusest lõpu poole, leiab esimese rakendatava reegli ja rakendab seda. Seejärel uurib nimekirja uuesti algusest, leiab ja rakendab uut (või sama) reeglit. Tsüklid kordub, kuni ühtegi reeglit enam rakendada ei saa või avaldis on antud tüüpi jaoks lahendatud kujul. Sellisel viisil koostab programm lahendustee ja saab ülesandele vastuse.

T-algebra kontrollib reegli rakendatavust üritades reeglile sobivaid operande leida. Kui operandid on leitud, saab reeglit leitud operandidele rakendada. Peale operandide leidmist arvutab programm sõltuvalt sisestamise režiimist vajalikud tulemuse osad, paneb need muutumatuks jäävate osadega kokku ja saab nii uue avaldise.

Programm kasutab lahendamise mootorit lisaks õpilase programmis abi andmisele (vt jaotis 2.4.2) ülesannete koostamise juures ebasobivate ülesannete välja praakimiseks. Ülesanne on sobiv ainult siis, kui T-algebra auto-

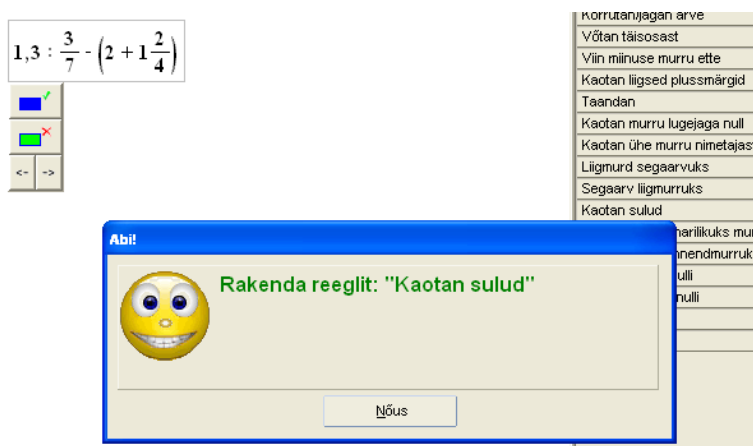


maatlahendaja oskab vastava ülesandetüübi algoritmi kasutades lähteavaldise lahendatud kujule teisendada. Enamike harilike murdude ülesandetüüpide korral on lahendatud kujuks taandumatu harilik murd või segaarv.

### 2.4.2 Nõu andmine ja erinevad kontrollid

Lisaks ülesannete lahendamisele oskab T-algebra kontrollida õpilase lahendussamme ja vastuseid, anda nõu jne. Käesolev jaotis annab programmi sel-listest oskustest ülevaate.

**Nõu andmine** Tahame, et õpilane osaleks alati lahendusprotsessis ning õpiks iga sammu kõiki etappe. Seepärast ei näita T-algebra abi andes kogu sammu korraga, vaid ainult sammu järgmist etappi. Sammu igal etapil on programmi käest võimalik abi küsida ning lasta programmil konkreet-sed etapid automaatselt lõpetada. Vajutades nuppu *Abi reegli valimisel* (vt joonis 7) saab õpilane enne igat sammu programmilt küsida, millist reeglit tuleks sel hetkel vastavalt algoritmile rakendada. T-algebra kontrollib üles-ande tüüpi, uurib aktiivset avaldist, leiab selle põhjal rakendatava reegli ning kuvab vastava abiteate (vt joonis 14). Sama nupp näitab, kas ülesanne on juba lahendatud.



Joonis 14. Abiteate kuvamine reegli valimisel

Operandide märkimisel võib õpilane vajutada spetsiaalset arvuti pildiga nuppu (vt joonis 7), mispeale programm valib sobivad operandid ise. Kui õpilane valib reegli, mida ei saa rakendada, ja palub abi operandide märki-miseks, siis T-algebra teatab, et valitud reeglit ei saa rakendada. Sama nupp

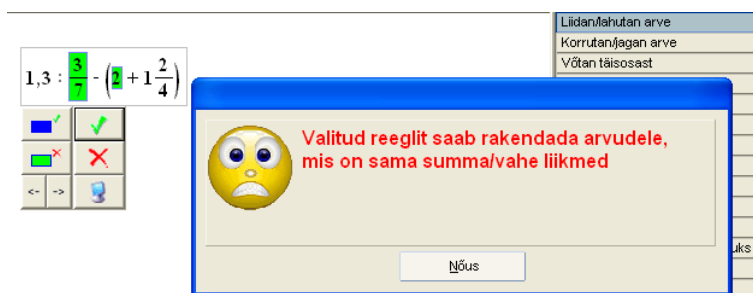
on kasutatav ka vastuse sisestamisel – programm paneb kastidesse õiged vastused.

T-algebra oskab hetkeavaldisest lähtuvalt genereerida ja kuvada ka kogu lahenduse (nupp *Lahenda lõpuni*, vt joonis 7), aga soovitame õpetajatel selle võimaluse ülesannete failis välja lülitada.

**Sammu etappide kontrollimine** T-algebras on õpilasele jäetud võimalus eksida sammu kõigil kolmel etapil ja lisaetappidel. Kuna viga on võimalik teha, on T-algebral võimalik ka sellele reageerida.

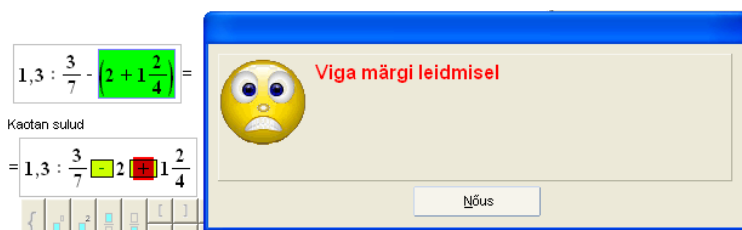
Esiteks on võimalik eksida reegli valimisel. Kui valitud reegli rakendamine on võimatu, ei informeerigi programm õpilast tehtud veast koheselt, sest õpilane ei leia reegli rakendamiseks sobivaid operande või teeb ebasobivaid operande valides vea. See annab õpilasele võimaluse parandada viga ilma täiendava abita. Kui kasutaja katkestab sammu enne märgitud osaavaldiste kinnitamist, siis võimatu reegli valimist veaks ei loeta. Mõnedel juhtudel, kui reegli rakendamine viiks lahendust täiesti valesse suunda, kuvab programm sobiva veateate ja ei jätka järgmise etapiga. Kui reegli rakendamine pole lihtsalt mõistlik – viib küll õige vastuseni, aga pikema lahendusteega, siis programm lubab jätkata ja jätab õpetajale võimaluse lahenduskäiku hinnata.

Teiseks võib õpilane teha vigu avaldise osade märkimisel. Esiteks kontrollib programm märgitud osade süntaktilist korrektsust. Teiseks kontrollib programm, kas märgitud osad on sobivad valitud operatsiooni rakendamiseks (näiteks reegli *Liidan/lahutan arve* operandid peavad olema täisarvud, kümnend- või harilikud murrud). Kolmandaks kontrollib programm, kas märgitud osad on valitud reegli jaoks sobivad (näiteks kümnend- ja hariliku murru liitmiseks tuleb esmalt kümnendmurd harilikuks murruks teisendada või vastupidi). Viimasena kontrollib programm märgitud osade asukohta (näiteks märgitud osad peavad olema sama summa või vahe liikmed (joonis 15)). Mõned reeglid on rakendatavad kogu avaldisele. Sellisel juhul ei pea midagi valima, aga märkimise võimalus on siiski alles jäetud.



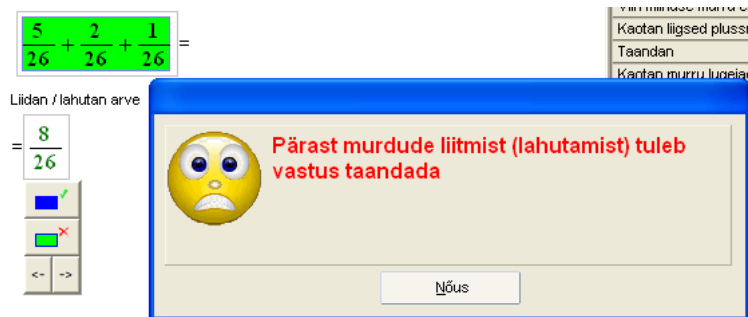
**Joonis 15.** Veateate kuvamine operandide märkimisel

Sisestamise etapil on kõige rohkem eksimise võimalusi, sest õpilane peab märgitud osadele reeglit rakendama ja tulemuse sisestama. Kui kasutaja sisestamise kinnitab, kontrollib programm esmalt sisestatud osade süntaktilist korrektsust ja reegli rakendamise korrektsust (kas vastus on sobival kujul). T-algebra kontrollib, kas sisestatud osad on ekvivalentsed programmi poolt arvutatud osadega. Lõpuks kontrollib programm kogu uue rea avaldise ekvivalentsust eelmise rea avaldisega. Kui õpilane vastuse sisestamisel eksib, kuvab programm sobiva veateate ning piisavalt täpse vea põhjuse tuvastamisel värvib vigase või puuduva sisendiga koha õpilase tähelepanu juhtimiseks punaseks (joonis 16).



**Joonis 16.** Veateate kuvamine vastuse sisestamisel

**Lahendatuse kontrollimine** Lahendusalgortim ja vastuse kuju sõltuvad ülesandetüübist. Kui õpilane arvab, et ta on ülesande lahendanud, peab ta vajutama nuppu *Annan vastuse* (joonis 7). Programm kontrollib, kas avaldis on sobival kujul. Kui ülesanne ei ole lahendatud, proovib programm tuvastada, millised algoritmi sammud teostamata on, ja kuvab vastava veateate. Näiteks samanimeliste murdude liitmisel tuleb tulemuseks olev murd enne vastuse andmist võimalusel taandada (joonis 17).



Joonis 17. Veateate kuvamine vastuse andmisel

Kui õpilane ei saa aru, et ülesanne on lõpuni lahendatud ja üritab ülesannet edasi lahendada, siis programm ei luba seda teha. Kui õpilane proovib uut reeglit valida, kuvab programm veateate “Ülesanne on juba lahendatud. Teata vastus. Reegli rakendamine ebaõnnestus!”.

**Esialgse avaldise kontrollimine** Ülesande koostamisel tuleb valida ülesandetüüp ja seejärel sisestada esialgne avaldis. Esialgne avaldis peab esiteks olema süntaktiliselt korrektne. Teiseks peab esialgne avaldis olema valitud ülesandetüübi jaoks lubatud kujul, näiteks taandamise ülesande esialgne avaldis peab olema murd, mille lugejas ja nimetajas on täisarv või täisarvude korrutis. Lisaks on enamike ülesandetüüpide korral spetsiifilised kitsendused. Näiteks taandamise ülesande korral peab murru lugejal ja nimetajal leiduma ühest suurem ühine tegur. Programm kontrollib esmalt, kas avaldis on ülesandetüübi jaoks sobival kujul, lahendab seejärel ülesande ja kuvab õpetajale soovi korral ülesande lahendustee ja vastuse.

**Kahe avaldise ekvivalentsuse kontrollimine** Programm kasutab kahe avaldise ekvivalentsuse kontrollimiseks ülalkirjeldatud lahendamise mootorit. Selleks eesmärgiks on koostatud spetsiaalne algoritm, mis sisaldab kokku 23 reeglit. Ekvivalentsuse kontrollimiseks koostab programm võrreldavate avaldiste vahe ja lihtsustab või lahendab selle vastavalt koostatud algoritmidele samamoodi nagu lahendamise mootor. Kui vahe on null, siis avaldised on ekvivalentsed.

Koostatud kontrollimise algoritm on sobilik T-algebras lubatud avaldistele. T-algebra ei käsitle trigonomeetriat ja absoluutväärtusi, seetõttu saime piirduda selle lihtsa algoritmi realiseerimisega.

## 3 Harilike murdude ülesandetüübid matemaatika õpikutes ja T-algebras

Töö autor valis T-algebra raames uurimiseks ja programmeerimiseks harilike murdude teema. T-algebra projekti ülesandepüstitus [18] pani paika esialgse nimekirja harilike murdudega seotud ülesandetüüpidest, mida programm kindlasti toetama peab. Autor alustas tööd Eesti matemaatika õpikute [8, 17] uurimisest. Kuna T-algebra raames oli eesmärk toetada ainult tehnilisi (avaldiste teisendamise) ülesandeid, siis jäid vaatluse alt välja kõik teised ülesandetüübid (tekstülesanded jm). Õpikute uurimise käigus ülesandetüüpide nimekiri mõnevõrra täienes. Käsikäes õpikute uurimisega tuvastas ja programmeeris autor ülesandetüüpide jaoks vajalikud reeglid ning viimase sammuna lisas programmi ülesandetüüpide toe.

### 3.1 Ülesandetüübid matemaatika õpikutes

Käesolev peatükk annab ülevaate harilike murdude teema käsitlesest ning vastavatest tehnilistest harjutusülesannetest matemaatika õpikutes ja T-algebras.

#### 3.1.1 Harilike murdude taandamine ja laiendamine

Mõlemas vaadeldud õpikus algab harilike murdude teema taandamisega. Antakse hariliku murru põhiomadus ning defineeritakse taandamine ja laiendamine:

- kui murru lugejat ja nimetajat korrutada või jagada ühe ja sama nullist erineva arvuga, siis saame selle murruga võrdse murru;
- murru jagamist ühe ja sama nullist erineva naturaalarvuga nimetatakse murru taandamiseks;
- murru korrutamist ühe ja sama nullist erineva arvuga nimetatakse murru laiendamiseks.

Peale selgitavat osa harjutab õpilane järgmisi ülesandeid: *Taanda harilik murd (taandumatu murruni)*, *Laienda harilikkude murdu etteantud nimetajani*, *Teisenda harilikkude murrud ühenimelisteks*.

### 3.1.2 Harilike murdude võrdlemine

Harilike murdude võrdlemiseks on matemaatika õpikutes esitatud järgmised reeglid:

- kahest ühenimelisest murrust on suurem see, mille lugeja on suurem;
- kui kahel murrul on võrdsed lugejad, siis on väiksem see murd, mille nimetaja on suurem;
- erinimeliste murdude võrdlemiseks tuleb need esmalt ühenimelisteks teisendada ja seejärel rakendada ühenimeliste murdude võrdlemise reeglit.

Ülaltoodud reeglite kinnistamiseks harjutatakse ülesannet *Võrdle harilikke murde*.

### 3.1.3 Ühenimeliste murdude liitmine ja lahutamine

Ühenimeliste murdude liitmine ja lahutamine tugineb kahel matemaatika õpikutes esitatud reeglil:

- ühenimeliste murdude liitmisel liidetakse nende murdude lugejad, nimetaja jääb endiseks;
- ühenimeliste murdude lahutamisel lahutatakse vähendatava lugejast vähendaja lugeja, nimetaja jääb endiseks.

Osad õpikud nõuavad pärast liitmist või lahutamist võimalusel vastuse taandamist, selline lähenemine on valitud ka T-algebras. Toodud reeglite kinnistamiseks harjutatakse ülesannet *Liida/lahuta ühenimelised murrud ja kui võimalik, siis taanda vastus*. Kuna negatiivseid arve õpitakse koolis hiljem kui tehteid murdudega, siis lahendatakse esialgu ainult selliseid ülesandeid, kus lõpptulemus ega ükski vahetulemus negatiivne pole. T-algebra ülesandetüübil selliseid piiranguid pole, sest hiljem, kui negatiivsed arvud selged on, harjutatakse ka keerukamaid, negatiivsete arvude tundmist nõudvaid ülesandeid. Siintoodu kehtib ka kõigi teiste harilike murdude ülesandetüüpide kohta.

### 3.1.4 Samanimelise murdosaga segaarvude liitmine ja lahutamine

Õpikutes tuuakse sisse segaarvu mõiste. Liigmurdu saab esitada naturaalarvu ja lihtmuru summana. Naturaalarvu ja lihtmuru summat, mis on kirjutatud ilma plussmärgita, nimetatakse segaarvuks. Summas olevat naturaalarvu nimetatakse segaarvu täisosaks, lihtmuru murdosaks. Segaarvu definitsiooni kinnistamiseks harjutatakse ülesandeid *Teisenda segaarv liigmurruks* ning *Eralda liigmuru täis- ja murdosa*. Järgnevalt esitatakse õpikutes samanimelise murdosaga segaarvude liitmise ja lahutamise reeglid:

- segaarvude liitmisel liidame täisosad eraldi ja murdosad eraldi;
- segaarvude lahutamisel lahutame täisosast täisosa ja murdosast murdosa ning liidame tulemused.

Toodud reeglite kinnistamiseks harjutatakse ülesannet *Liida/lahuta samanimelise murdosaga segaarvud ja kui võimalik, siis taanda vastus*.

### 3.1.5 Erinimeliste murdude ja segaarvude liitmine ja lahutamine

Õpikutes esitatakse erinimeliste murdude ja segaarvude liitmise reeglid järgmiselt:

- erinimeliste murdude liitmiseks ja lahutamiseks teisendatakse murrud esmalt ühenimelisteks ja seejärel toimitakse ühenimeliste murdude liitmise (või lahutamise) eeskirja järgi;
- kui summas või vahes leidub segaarve, siis teisendatakse nende murdosad esmalt ühenimelisteks.

Harjutatakse ülesannet *Liida/lahuta erinimelised murrud ja kui võimalik, siis taanda vastus*.

### 3.1.6 Murdude teisendused

Iga kümnendmurdu võib vaadata kui naturaalarvu ja järguühiku jagatist. Harjutatakse ülesandeid *Teisenda kümnendmurd harilikuks murruks ja kui võimalik, siis taanda vastus* ning *Teisenda harilik murd lõplikuks kümnendmurruks*. Viimase ülesandetüübi sisendiks on harilik murd, mis teisendub lõplikuks kümnendmurruks, või segaarv, mille murdosa teisendub lõplikuks kümnendmurruks. Perioodiliseks kümnendmurruks teisenduvate harilike murdude

kohta harjutatakse ülesannet *Leia hariliku murru või segaarvu kümnenlühend ümardatud täpsuseni ... kohta peale koma.*

### 3.1.7 Harilike murdude korrutamine

Õpikutes esitatakse harilike murdude korrutamise reeglid järgmiselt.

- Kahe hariliku murru korrutis võrdub murruga, mille lugejaks on antud murdude lugejate korrutis ja nimetajaks nimetajate korrutis.
- Kui tegurite hulgas on segaarve, siis teisendatakse need enne liigmurdudeks ja seejärel kasutatakse murdude korrutamise eeskirja. Kui tegurite seas on naturaalarve, siis võib need kirjutada murru kujul nimetajaga 1.

Reeglite kinnistamiseks harjutatakse ülesannet *Korruta harilikud murrud*, mille raames õpitakse harilike murdude, segaarvude ja naturaalarvude korrutamist.

### 3.1.8 Harilike murdude jagamine

Esmalt esitatakse õpikutes pöördarvu definitsioon:

- kaht arvu, mille korrutis on võrdne 1-ga, nimetatakse teineteise pöördarvuks.

Järgnevalt harjutab õpilane ülesannet *Leia pöördarv*, mille sisendiks on naturaalarv, harilik murd, segaarv või kümnenmurd. Peale pöördarvu teemat esitatakse jagamise eeskiri:

- selleks et jagada harilikku murdu hariliku murruga, tuleb jagatav korrutada jagaja pöördarvuga;
- kui andmete seas on sega- või naturaalarve, siis tuleb need esmalt (sarnaselt korrutamiselega) liigmurruks teisendada.

Reegli kinnistamiseks harjutatakse ülesannet *Jaga harilikud murrud*, mille raames õpitakse harilike murdude, segaarvude ja naturaalarvude jagamist.



### 3.1.9 Ülesanded kõigile tehetele murdudega

Õpikud sisaldavad harilike murdude osa viimase jaotisena keerukamaid ülesandeid eelnevalt õpitu kinnistamiseks. Harjutatakse ülesannet *Arvuta kümnenend- ja harilikke murde sisaldava avaldise väärtus*, kus avaldis võib sisaldada täisarvude, harilike murdude, segaarvude ja kümnenendmurdude liitmist, lahutamist, korrutamist ja jagamist.

T-algebras on realiseeritud kõik ülaltoodud harilike murdude ülesandetiübid.

## 3.2 Disainitud reeglid T-algebras

Käesolev peatükk annab põhjaliku ülevaate operatsioonidest murdudega T-algebras. Lühiülevaate teemast saab ka T-algebra töögrupi vastavast publikatsioonist [23].

T-algebra töötaades peab õpilane iga sammu esimesel etapil valima rakendatava reegli. Paberi ja pliiatsi abil lahendades teeb õpilane reegli valiku implitsiitselt, jättes selle paberile kirjutamata. Erinevalt T-algebra ülesandelahendamise dialoogist võib õpilane paberil ühe lahendussammu käigus rakendada mitut reeglit – nt korrutada harilikud murrud ja taandada tulemuse. Õpetaja peab lahendust kontrollides aru saama, milliseid reegleid õpilane rakendas. Erinevalt õpetajast on programmil olukorras, kus ühe lahendussammu käigus saab mitut reeglit rakendada, raske tuvastada, mida õpilane teha üritas. Kui programm teab, millist reeglit rakendati, on võimalik teostada erinevad kontrollid. Esiteks saab kontrollida, kas õpilane tunneb ülesande lahendusalgorithmi, st tuvastada kas ta oskab õige reegli valida. Teiseks saab programm palju tõhusamalt kontrollida, kas õpilase tegevused järgmistel etappidel on korrektsed: kas õpilase poolt märgitud avaldise osad on valitud reegli rakendamiseks sobivad ning kas tulemuse kuju vastab reeglile. Lisaks saab vea korral antavas tagasisides viidata reeglile, mille õpilane ise valinud on.

Reeglite kavandamisel on järgitud võimalikult täpselt õpikutes toodud algoritme ja eeskirju. Üks olulisemaid eesmärke oli teha ülesannete lahendamine programmi abil võimalikult sarnaseks paberil lahendamisele. Me loodame, et selle tulemusena kanduvad programmi abil lahendamise oskused üle ka paberil lahendamisele. Samuti lihtsustab sarnasus programmi kasutamist õppevahendina.

Kavandatud reeglite hulk on täielik, kõik harilike murdudega seotud ülesanded on nende reeglite abil lahendatavad. Programmi reeglid vastavad koolis esitatud lahendusalgoritmi sammudele. Iga ülesandetüübi reeglikomplekt koosneb õpitava algoritmi või eeskirja reeglitest ning eelnevalt õpitud arvutamise reeglitest. Vaatleme näiteks erinimeliste harilike murdude liitmise ja lahutamise ülesandetüübi reegleid. Esiteks saab õpilane kasutada reeglit selle ülesandetüübi peamise operatsiooni teostamiseks: *Liidan/lahutan arve*. Enne liitmist või lahutamist operandide ja peale tehte teostamist vastuse sobivale kujule teisendamiseks on abiks järgmised (tugi)reeglid: *Võtan täisosast; Liigmurd segaarvuks; Segaarv summaks; Taandan*. Lisaks eeltoodule saab õpilane kasutada lihtsustusreegleid, milleks on *Viin miinuse murru ette; Kaotan liigsed plussmärgid; Kaotan murru lugeja null; Kaotan ühe murru nimetajast; Liidan/lahutan nulli*.

Kavandatud reeglite hulk on piisavalt väike, et kogu komplekti ülesande lahendamisel staatilise menüüna kuvada. See annab võimaluse diagnoosida, kas õpilane teab, millist algoritmi sammu hetkel teostada, milline reegel on antud sammul rakendatav.

Oleme üritanud reeglid selliselt kavandada, et need oleks rakendatavad võimalikult erinevates kontekstides. Näiteks reegel *Liidan/lahutan arve* on kasutatav lisaks harilike murdude teema ülesandetüüpidele ka teiste T-algebra ülesandetüüpide (mida käesolev töö ei kirjelda) juures tugireegli-na. See annab õpilasele võimaluse kord õpitud reeglit erinevate avaldiste ja ülesandetüüpide juures uuesti rakendada.

Pärast lahendussammu etapi kinnitamist teostab programm erinevaid kontrole. Vigade tegemise võimalusi ja kontrollimise põhimõtteid kirjeldab jaotis 2.4. Esitame veelkord lahendussammu teise ja kolmanda etapi kontrollimise põhimõtted ja reeglist sõltumatud üldised kontrollid. Pärast teist etappi kontrollib programm:

- kas avaldise mõned osad on märgitud (leidub reegleid, mille korral pole vaja midagi märkida);
- märgitud osade süntaktilist korrektsust;
- märgitud osade arvu (vaja ainult ühte osa, vaja vähemalt kahte osa, jne – kirjeldatud iga reegli jaoks eraldi);
- märgitud osa kuju (võib reegliti erineda, kirjeldatud iga reegli jaoks eraldi);

- mõnel juhul märgitud osa positsiooni (kirjeldatud iga reegli jaoks eraldi).

Pärast lahendussammu kolmandat etappi teostatavad kontrollid sõltuvad sisestamise režiimist ja reeglist. Järgnevalt on ära toodud üldised kontrollid, mida programm rakendab igas sisestamise režiimis iga reegli korral. Kontrollid teostatakse siin kirjeldatud järjekorras ja iga järgmine kontroll toimub ainult siis, kui eelmised kontrollid on edukalt läbitud. T-algebra kontrollib:

- tulemuse sisestatust (kastid ei ole tühjad), erijuhtudel, mis on iga reegli juures välja toodud, võivad osad kastid tühjaks jääda;
- sisestatud osade süntaktilist korrektsust;
- reeglist sõltuvaid aspekte (kirjeldatud iga reegli jaoks eraldi);
- sisestatud ja arvuti poolt arvutatud osade põhjal koostatud tervikavaldiste ekvivalentsust.

Järgnevalt vaatame lähemalt harilike murdude jaoks disainitud reegleid T-algebras. Iga reegli kohta on esmalt ära toodud kus ja millisel eesmärgil seda rakendada saab ning millised avaldise osad peab õpilane reegli valimiseks märkima. Samuti on antud ülevaade täiendava informatsiooni ja vahetulemuse sisestamisest, kui reegli rakendamine neid etappe ette näeb. Seejärel on kirjeldatud, kuidas näeb välja tulemuse sisestamine kolmes erinevas režiimis, käsitledes ainult reeglispetsiifilisi kontrole (üldised kontrollid on toodud ülalpool). Viimasena on iga lahenduse etapi kohta ära toodud juhend, mida õpilane programmi aknas näeb. Ühe reegli (*Liidan/lahutan arve*) jaoks on lisaks esitatud kõik veateated, mida programm veasituatsioonis õpilasele kuvab.

### 3.2.1 Reegel *Taandan*

#### Rakendused:

- operandide lihtsustamiseks enne harilike murdude liitmist, lahutamist, korrutamist ja jagamist;
- harilike murdude korrutamisel ja jagamisel viiakse operandid ühisele murrujoonele ning enne lugejas ja nimetajas korrutamist võimalusel taandatakse;

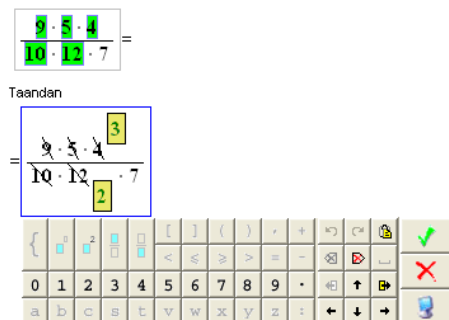
- harilike murdude liitmise, lahutamise, korrutamise ja jagamise tulemuse lihtsustamiseks.

**Juhend märkimisel:** Vali ühe murru lugejast ja nimetajast taandatavad täisarvud.

**Märkimine:** Märkida tuleb taandatava murdavaldise lugejast ja nimetajast kummastki vähemalt üks arv. Nendel peab olema ühest suurem ühine tegur. Kui murdavaldise lugeja või nimetaja on korrutis, siis võib sealt valida rohkem kui ühe teguri. T-algebra ei võimalda reegli paralleelset rakendamist, korraga saab taandada ainult ühe murdavaldise. Ebasobivate osade märkimisel kuvab programm vastava veateate.

**Tulemuse sisestamise juhend:** Sisesta taandamise tulemus.

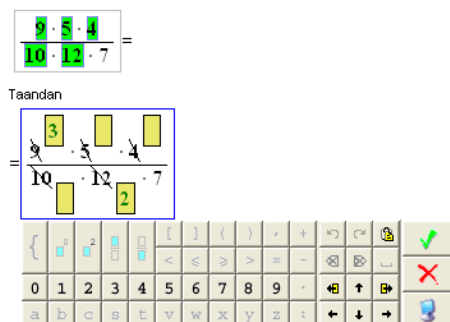
**Tulemuse sisestamine (vabas sisestamine režiimis):** Kui märgitud osad on korrektsed, kopeerib programm kogu avaldise järgmisele reale ning kuvab lugejas ja nimetajas märgitud arvud mahakriipsutatult. Mahakriipsutatud osadele asenduse sisestamiseks on nii lugejas kui ka nimetajas üks kast (ka siis kui lugejas või nimetajas on valitud rohkem kui üks täisarv) (joonis 18). Kastidesse on lubatud sisestada numbreid ja korrutamise märke. Peale sisestuse kinnitamist rakendab programm esmalt üldisi kontrole. Lisaks kontrollib T-algebra, et midagi oleks taandatud ja taandamise asemel ei laiendataks.



**Joonis 18.** Reegli *Taandan* tulemuse sisestamine vabas režiimis

**Tulemuse sisestamine (struktuurses ja osalises sisestamise režiimis):** Kui märgitud osad on korrektsed, kopeerib programm kogu avaldise järgmisele reale ning kuvab lugejas ja nimetajas märgitud arvud mahakriipsutatult. Iga mahakriipsutatud täisarvu kohale kuvab programm kasti asenduse sisestamiseks (joonis 19). Erinevalt vabast režiimist saab kastidesse sisestada

ainult numbreid. Peale sisestuse kinnitamist rakendab programm esmalt üldisi kontrolli erisusega, et kastidesse võib ühed kirjutamata jätta. Vähemalt üks kast peab siiski lugejas ja nimetajas täidetud olema. Lisaks kontrollib T-algebra, et taandamisel taandatava arvu absoluutväärtus ei suureneks (see välistab ka laiendamise) ja sisestatud arv koosneks algse arvu teguritest.



**Joonis 19.** Reegli *Taandan* tulemuse sisestamine struktuurses ja osalises režiimis

Kuigi mahakriipsutatud arve sisaldav avaldis pole väga ülevaatlik, hakkab õpilane paberil lahendades tavaliselt sellest seisust järgmist lahendusammu teostama – avaldise puhtal kujul ümber kirjutamine oleks liiga aeganõudev. Arvuti jaoks on ümber kirjutamine kiire toiming, peale tulemuse sisestamist kopeerib programm avaldise uuele reale ja asendab selles mahakriipsutatud osad kasutajalt saadud sisendiga. Sellega reegli rakendamine lõppeb.

### 3.2.2 Reegel *Laiendan*

**Rakendused:** Erinimeliste harilike murdude ühenimeliseks teisendamiseks võrdlemise, liitmise või lahutamise eesmärgil. Kuna murdude ülesannetes on laiendamine liitmise ja lahutamise reeglisse sisse integreeritud, siis laiendamise reegel on kasutusel ainult mõne ülesandetüübi (*Laienda etteantud nimetajani*, *Teisenda murrud ühenimelisteks*, *Võrdle murde*) korral.

**Juhend märkimisel:** Vali harilik murd või segaarvu murdosa, mida laiendada.

**Märkimine:** Märkida tuleb laiendatav harilik murd. Enamike ülesandetüüpide korral ei võimalda T-algebra reegli paralleelset rakendamist. Kahe ülesandetüübi (*Teisenda murrud ühenimelisteks*, *Võrdle murde*) korral on vaja tihti laiendada rohkem kui ühte murdu. Nende ülesannete kontekstis võib mitu murdu korraga märkida.

**Täiendava informatsiooni sisestamine:** Kui märgitud on rohkem kui üks murd, kuvab programm ühise nimetaja sisestamiseks eraldi akna (joonis 20).

**Täiendava informatsiooni sisestamise juhend:** Sisesta ühine nimetaja.

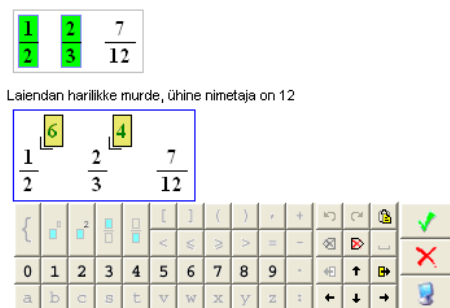


**Joonis 20.** Reegli *Laiendan* rakendamisel täiendava informatsiooni sisestamine

**Vahetulemuse sisestamise juhend:** Sisesta laiendaja.

**Vahetulemuse sisestamine:** Programm kuvab uuel real avaldise, milles on kastid laiendajate sisestamiseks (joonis 21). Pärast sisestuse kinnitamist kontrollib programm:

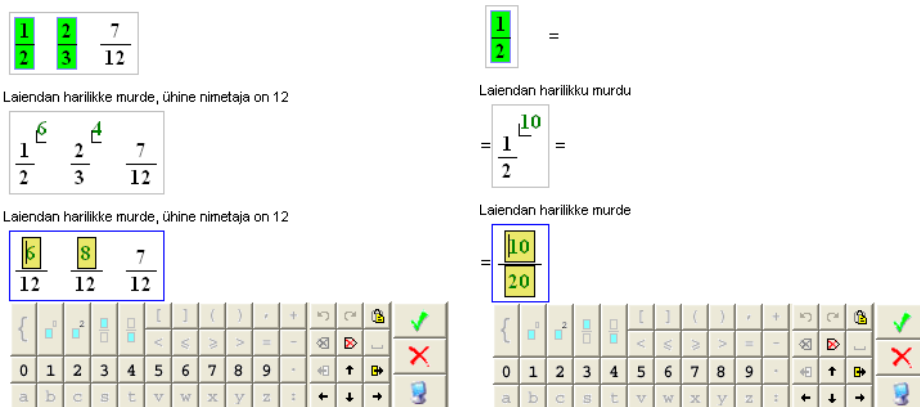
- kas laiendajad on positiivsed täisarvud (arv null pole lubatud);
- (rohkem kui ühe murru korral) kas laiendajad vastavad eelnevalt sisestatud ühisele nimetajale;
- (etteantud nimetajani laiendamise ülesande korral) kas sisestatud laiendaja vastab ülesandes antud nimetajale.



**Joonis 21.** Reegli *Laiendan* rakendamisel vahetulemuse sisestamine

**Tulemuse sisestamise juhend:** Sisesta uus lugeja ja nimetaja.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise. Mitme murru korraga laiendamisel on ühine nimetaja juba sisestatud, programm kuvab uuele reale kopeeritud avaldises kastid laiendatavate murdude lugejate sisestamiseks (joonis 22 vasakul). Ühe murru laiendamisel kuvab programm kastid lugeja ja nimetaja sisestamiseks (joonis 22 paremal). Peale sisestuse kinnitamist kontrollib T-algebra tulemuse vastavust vaheetapil sisestatud laiendajatele ja kuvab vajadusel vastava veateate.



Joonis 22. Reegli *Laiendan* tulemuse sisestamine

### 3.2.3 Reegel *Võtan täisosast*

**Rakendused:** Reeglit kasutatakse enne või pärast segaarvude vahe (erijuhul võib vähendatav olla täisarv ja vähendaja harilik murd) leidmist, kui vähendatava segaarvu murdosa on vähendaja murdosast väiksem.

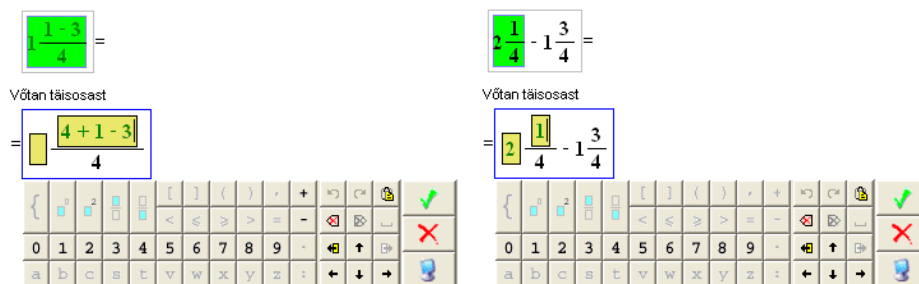
**Juhend märkimisel:** Vali segaarv, täisarv või erikujul segaarv (mille murdosa lugeja on summa või vahe), mille täisosast laenata.

**Märkimine:** märkida tuleb üks segaarv, täisarv või erikujul segaarv. T-algebra ei võimalda reegli paralleelset rakendamist, korraga saab laenata ühest täisarvust või segaarvu täisosast.

**Tulemuse sisestamise juhend:** Sisesta laenamise tulemus. Täisosa kasti võib jätta erijuhul tühjaks.

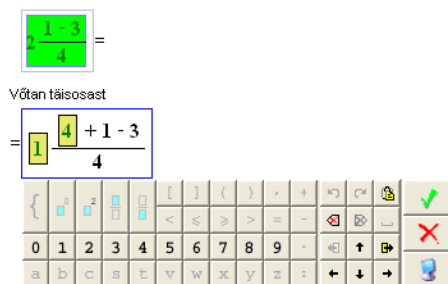
**Tulemuse sisestamine (vabas ja struktuurses režiimis):** Programm kuvab uuel real avaldise, milles on kastid märgitud segaarvu täisosa ja lugeja sisestamiseks. Erijuhul, kui märgitud oli täisarv, kuvab programm kasti ka nimetaja sisestamiseks. Erikujul segaarvu korral ei pea murdosa lugejat välja arvutama, programm lubab vastavasse kasti summa või vahe sisestada (joo-

nis 23 vasakul). Tavalise segaarvu täisosast võtmisel tohib murdosa lugeja kasti sisestada ainult numbreid (joonis 23 paremal). Peale sisestuse kinnitamist teostab programm üldised kontrollid erisusega, et täisosa null võib sisestamata jääda (joonis 23 vasakul). Lisaks kontrollib programm, et reegli rakendamise tulemusena segaarvu täisosa ei suureneks (see võib juhtuda, kui segaarvu murdosa on liigmurd).



**Joonis 23.** Reegli *Võtan täisosast* tulemuse sisestamine vabas ja struktuurses režiimis

**Tulemuse sisestamine (osalises sisestamise režiimis):** Vabast ja struktuursest režiimist erineb osaline sisestamise režiim ainult erikujul segaarvude korral. Programm kopeerib murdosa lugejas oleva summa või vahe muutumatuna uuele reale ja genereerib selle ette kasti täisosast laenatava osa sisestamiseks (joonis 24).



**Joonis 24.** Reegli *Võtan täisosast* tulemuse sisestamine osalises režiimis

### 3.2.4 Reegel *Liigmurd segaarvuks*

**Rakendused:** Reeglit kasutatakse murdudega seotud arvutusülesannetes liigmurruna saadud vastuse segaarvuks teisendamisel. Samuti on liigmurdude võrdlemisel kasulik need esmalt segaarvudeks teisendada.





**Joonis 25.** Reegli *Liigmurd segaarvuks* tulemuse sisestamine kõigis kolmes režiimis

**Juhend märkimisel:** Vali liigmurd või segaarv, mille murdosa on liigmurd.

**Märkimine:** Märkida tuleb liigmurd või segaarv, mille murdosa on liigmurd. T-algebra ei võimalda reegli paralleelset rakendamist, korraga saab segaarvuks teisendada ühe liigmuru.

**Tulemuse sisestamise juhend:** Sisesta tulemus. Murdosa kastid võib erijuhul tühjaks jätta.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, milles on kastid märgitud osale vastava segaarvu täisosa, lugeja ja nimetaja sisestamiseks (joonis 25). Täisosa sisestamine on kohustuslik. Erijuhul, kui reegli rakendamise käigus murdosa üldse ära kaob, tuleb lugeja ja nimetaja kastid tühjaks jätta (joonis 25 paremal). Peale sisestuse kinnitamist teostab programm üldised kontrollid arvestades, et lugeja ja nimetaja kastis võib sisestus puududa. Lisaks kontrollib programm, et täisosa suureneks ja murdosa sisestus oleks konsistentne (mõlemad kastid korraga tühjad või täidetud). T-algebra ei luba reegli rakendamise käigus taandada ega laiendada. Näiteks tuleb segaarvu  $2\frac{18}{15}$  teisendamiseks kujule  $3\frac{1}{5}$  rakendada lisaks kirjeldatavale reeglile ka taandamise reeglit.

### 3.2.5 Reegel *Segaarv liigmurruks*

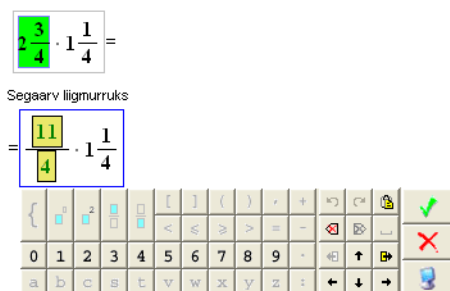
**Rakendused:** Reeglit kasutatakse enne korrutamist või jagamist segaarvude liigmurruks teisendamiseks.

**Juhend märkimisel:** Vali sega- või täisarv, mida liigmurruks teisendada.

**Märkimine:** Märkida tuleb üks sega- või täisarv. T-algebra ei võimalda reegli paralleelset rakendamist korraga mitmele sega- või täisarvule.

**Tulemuse sisestamise juhend:** Sisesta liigmurru lugeja ja nimetaja.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, milles on kastid märgitud sega- või täisarvule vastava liigmurru lugeja ja nimetaja sisestamiseks (joonis 26). Peale sisestuse kinnitamist teostab programm üldised kontrollid. T-algebra ei luba reegli rakendamise käigus segaarvu taandada ega laiendada, nende tegevuste jaoks on eraldi reeglid.



**Joonis 26.** Reegli *Segaarv liigmurruks* tulemuse sisestamine kõigis kolmes režiimis

### 3.2.6 Reegel *Kümnendmurd harilikuks murruks*

**Rakendused:** Kui summas, vahes, korrutises või jagatises leidub samaaegselt harilikke ja kümnendmurde, siis enne arvutamist teisendatakse kümnendmurrud esmalt harilikeks murdudeks või segaarvudeks. Samuti teisendatakse kümnendmurd harilikuks murruks, kui on vaja selle pöördarv leida.

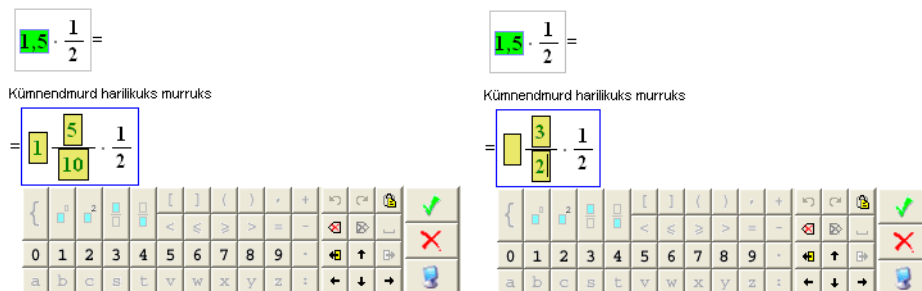
**Juhend märkimisel:** Vali kümnendmurd või täisarv, mida harilikuks murruks teisendada.

**Märkimine:** Märkida tuleb üks kümnendmurd või täisarv. T-algebra ei võimalda reegli paralleelset rakendamist korraga mitmele kümnendmurrule või täisarvule.

**Tulemuse sisestamise juhend:** Sisesta tulemus.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, kus märgitud täisarvu või kümnendmurru asemel on kolm segaarvu struktuurile vastavat kasti (joonis 27). Peale sisestuse kinnitamist teostab programm üldised kontrollid erisusega, et täidetud peavad olema kas kõik kolm kasti, ainult täisosa kast (T-algebra aktsepteerib täisarvu teisendamist iseendaks) või ainult murdosa kastid. Kui valitud kümnendmurd on ühest suurem, lubab programm esitada selle segaarvuna (joonis 27 vasakul) või liigmurruna (joonis 27 paremal). Erinevates kontekstides on otstarbekas

kümnendmurd erinevalt lahti kirjutada. Sisestatud murd või segaarvu murdosa võib olla taandamata (joonis 27 vasakul) või taandatud kujul (joonis 27 paremal).



**Joonis 27.** Reegli *Kümnendmurd harilikuks murruks* tulemuse sisestamine kõigis kolmes režiimis

### 3.2.7 Reegel *Harilik murd kümnendmurruks*

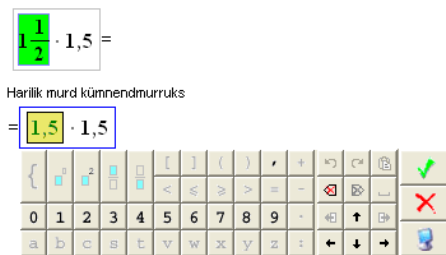
**Rakendused:** Kui summas, vahes, korrutises või jagatises leidub samaaegselt kümnendmurde ja lõplikke harilikke murde, siis on mõnikord kasulik harilikud murrud esmalt kümnendmurdudeks teisenda ja arvutused kümnendmurdudega teostada.

**Juhend märkimisel:** Vali harilik murd või segaarv, mida kümnendmurruks teisendada.

**Märkimine:** Märkida tuleb üks kümnendmurruks teisendatav harilik murd või segaarv. T-algebra ei võimalda reegli paralleelset rakendamist korraga mitmele harilikule murrule või segaarvule.

**Tulemuse sisestamise juhend:** Sisesta kümnendmurd või täisarv.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, kus on kast märgitud harilikule murrule või segaarvule vastava kümnendmuru sisestamiseks (joonis 28). Peale sisestuse kinnitamist teostab programm üldised kontrollid.



**Joonis 28.** Reegli *Harilik murd kümnendmurruks* tulemuse sisestamine kõigis kolmes režiimis

### 3.2.8 Reegel *Leian pöördarvu*

**Rakendused:** Hariliku murruga jagamiseks tuleb jagatavat jagaja pöördarvuga korrutada. Kuna pöördarvu leidmine on jagamise reeglisse integreeritud, siis pöördarvu leidmise reeglit kasutatakse ainult spetsiaalses pöördarvu leidmise ülesandes.

**Juhend märkimisel:** Vali arv, mille pöördarvu tahad leida.

**Märkimine:** Kuna reegel on kasutatav ainult pöördarvu leidmise ülesandes, mille algne avaldis on täisarv, kümnendmurd, harilik murd või segaarv, siis midagi märkima ei pea. Programm ei loe kogu avaldise märkimist siiski veaks.

**Tulemuse sisestamise juhend:** Täida kastid nii, et tulemus oleks esialgse avaldise pöördarv.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real kolm segaarvu struktuurile vastavat kasti (joonis 29). Peale sisestuse kinnitamist teostab programm üldised kontrollid kahe erisusega. Esiteks peavad täidetud olema kas kõik kolm kasti, ainult täisosa kast või ainult murdosa kastid. Teiseks, erinevalt enamikest reeglitest ei säilita pöördarvu leidmine ekvivalentsust (ekvivalentsust ei säilita ka reeglid *Jagan piisava täpsuseni* ja *Ümardan*), seega T-algebra kontrollib eelmise ja uue avaldise ekvivalentsuse asemel, et sisestatud täisarv, harilik murd või segaarv oleks teisendatava arvu pöördarv.



**Joonis 29.** Reegli *Leian pöördarvu* tulemuse sisestamine kõigis kolmes režiimis

### 3.2.9 Reegel *Liidan/lahutan arve*

Selle reegli kohta esitame ka kõik veateated, mida programm veasituatsioonis õpilasele kuvab.

**Rakendused:** Täisarve ning kümnend- ja harilikke murde sisaldava summa või vahe välja arvutamiseks.

**Juhend märkimisel:** Vali täisarvud, kümnendmurrud, harilikud murrud või segaarvud, mida tahad liita/lahutada.

**Märkimine:** Märkida tuleb sama summa (vahe) need liikmed, mida kokku liita (lahutada). Valitavate liikmete arv pole piiratud ja valitavad liikmed ei pea paiknema summas (vahes) kõrvuti. Programm ei luba samaaegselt valida kümnend- ja harilikke murde. Harilikud murrud ja segaarvud tuleb kõigepealt eraldi reegli abil kümnendmurdudeks teisendada või vastupidi. Summa (vahe) liikmed võivad sulgudes olla, sellisel juhul tuleb vastav liige koos sulgudega valida.

#### **Veateated peale märkimist:**

- Ühtegi avaldist pole valitud;
- Valitud reegli rakendamiseks tuleb valida vähemalt kaks avaldist;
- Kõik valitud avaldised pole summa/vahe liikmed;
- Valitud reeglit saab rakendada arvudele, mis on sama summa/vahe liikmed;
- Valitud reeglit saab rakendada ainult samas summas/vahes olevatele liikmetele;

- Teisenda kõigepealt harilikud murrud ja segaarvud kümnendmurdudeks või vastupidi.

**Täiendava informatsiooni sisestamine:** Kui valitud summa või vahe liikmete hulgas on vähemalt kaks erinimelist harilikku murdu või segaarvu ning sisestamise režiim struktuurne või osaline, kuvab programm ühise nimetaja sisestamiseks eraldi akna (joonis 30). Sisestada tuleb valitud harilike murdude ja segaarvu murdosade ühine nimetaja.

**Täiendava informatsiooni sisestamise juhend:** Sisesta ühine nimetaja.

**Veateated peale täiendava informatsiooni sisestamist:**

- Ühine nimetaja sisestamata;
- Viga ühise nimetaja sisestamisel;
- Viga ühise nimetaja arvutamisel.



**Joonis 30.** Reegli *Liidan/lahutan arve* täiendava informatsiooni sisestamine

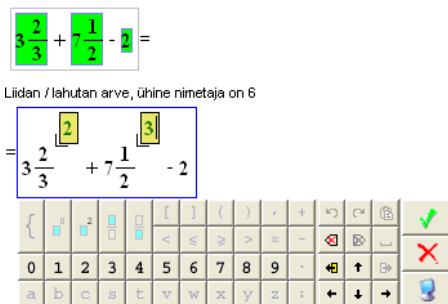
**Vahetulemuse sisestamise juhend:** Sisesta laiendajad.

**Vahetulemuse sisestamine:** Kui valitud summa või vahe liikmete hulgas on vähemalt kaks erinimelist harilikku murdu või segaarvu, kopeerib programm peale täiendava informatsiooni sisestuse kinnitamist avaldise uuele reale ja genereerib märgitud murdude ja segaarvude kohale kastid laiendajate sisestamiseks (joonis 31). Pärast sisestuse kinnitamist kontrollib programm:

- kas laiendajad on positiivsed täisarvud (arv null pole lubatud);
- kas laiendajad vastavad eelnevalt sisestatud ühisele nimetajale.

**Veateated peale vahetulemuse sisestamist:**

- Laiendaja sisestamata;



**Joonis 31.** Reegli *Liidan/lahutan arve* vahetulemuse sisestamine

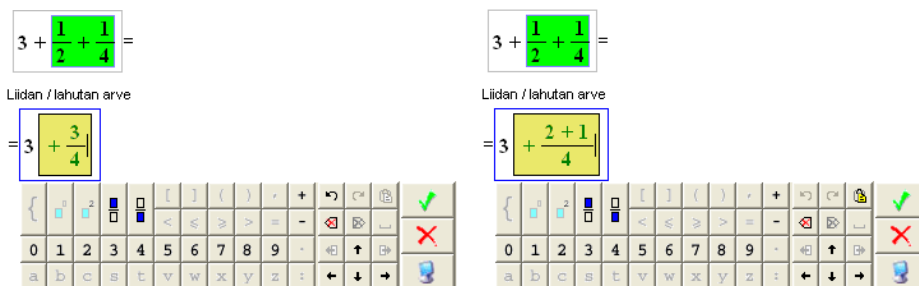
- Viga sisestamisel;
- Viga laiendaja arvutamisel.

**Tulemuse sisestamise juhend:** Sisesta liitmise/lahutamise tulemus.

**Tulemuse sisestamine (vabas sisestamise režiimis):** Programm kopeerib avaldise uuele reale ning kuvab märgitud summa või vahe liikmete asemele kasti tulemuse sisestamiseks (joonis 32). Kasti võib sisestada kohe lõpptulemuse (joonis 32 vasakul) või mõne olulise vahetulemuse (joonis 32 paremal). Peale sisestuse kinnitamist rakendab programm üldisi kontrole erisusega, et kui sisestuskast pole summa või vahe esimene liige ja märgitud liikmetele reegli rakendamise tulemus on null, siis võib kasti tühjaks jätta. Lisaks, kui sisestuskast on summa või vahe mitteesimene liige, kontrollib T-algebra kasti sisestatud avaldise alguses summa või vahe märgi olemasolu. Selle kontrolliga välistab programm võimaluse, et kasti sisestatud osa ja sellele eelnev mittemuudetavas avaldise osas paiknev täisarv segaarvu (või erikujul segaarvu) moodustaks. See ei pruugi alati viga olla (näiteks mõlemal joonisel 32 toodud pildil säilitaks teisendus plussmärgi sisestamata jätmisel ekvivalentsuse), aga selguse huvides nõuab programm siiski märgi sisestamist.

**Veateated peale tulemuse sisestamist (vabas sisestamise režiimis):**

- Tulemus sisestamata;
- Viga sisestamisel;
- Avaldise märk sisestamata;
- Viga arvutamisel.



**Joonis 32.** Reegli *Liidan/lahutan arve* tulemuse sisestamine vabas režiimis

**Tulemuse sisestamine (struktuurses sisestamise režiimis):** Programm kopeerib avaldise uuele reale ning kuvab märgitud summa või vahe liikmete asemele kastid tulemuse sisestamiseks. Uues avaldises leidub alati kast tulemuse märgi sisestamiseks. Ülejäänud kastide arv sõltub märgitud liikmetest, esinevad järgmised võimalused:

- kui märgitud liikmete hulgas on ainult täisarvud ja kümnendmurrud, leidub kast arvu sisestamiseks (joonis 33);
- kui märgitud liikmete hulgas on ainult samanimelised harilikud murrud (ühist nimetajat ei küsitud), leiduvad kastid murru lugeja ja nimetaja sisestamiseks (joonis 34);
- kui märgitud liikmete hulgas on ainult harilikud murrud, millest vähemalt kaks on omavahel erinev (ühine nimetaja on sisestatud), siis leidub kast murru lugeja sisestamiseks (murru nimetaja on mittemuudetaval kujul uues avaldises olemas, kuna kasutaja juba sisestas selle) (joonis 35);
- kui märgitud liikmete hulgas on vähemalt üks segaarv ja kõik märgitud harilikud murrud ja segaarvud on samanimelised (ühist nimetajat ei küsitud), siis leiduvad kastid täisosa, lugeja ja nimetaja sisestamiseks (joonis 36);
- kui märgitud liikmete hulgas on vähemalt üks segaarv ja vähemalt kaks omavahel erinev harilikku murdu või segaarvu, (ühine nimetaja on sisestatud), siis leiduvad kastid täisosa ja lugeja sisestamiseks (murru nimetaja on mittemuudetaval kujul uues avaldises olemas, kuna kasutaja juba sisestas selle) (joonis 37).

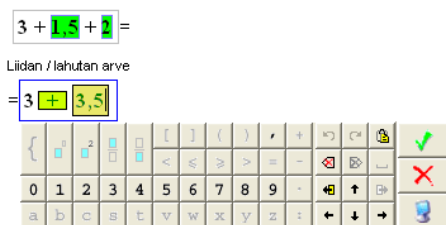


Täisosa ja nimetaja kastidesse saab sisestada ainult arve, lugeja kasti lõpliku tulemust ei pea kohe välja arvutama (näide kohe välja arvutamisest on joonisel 36), selle võib väljendada summa või vahena (nii on tehtud joonistel 34, 35 ja 37). Peale sisestuse kinnitamist teostab programm üldised kontrollid järgmiste erisustega:

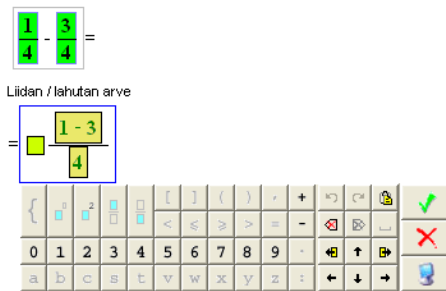
- kui sisestatava tulemuse kastid on summa või vahe esimeseks liikmeks, siis võib plussmärgi sisestamata jätta;
- kui sisestatava tulemuse kastid vastavad segaarvu struktuurile ja tulemuse täisosa on null, siis tuleb vastav kast tühjaks jätta.

**Veateated peale tulemuse sisestamist (struktuurses sisestamise režiimis):**

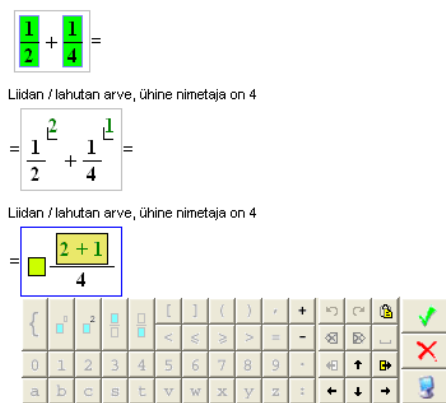
- Tulemus sisestamata;
- Lugeja sisestamata;
- Nimetaja sisestamata;
- Avaldise märk sisestamata;
- Viga sisestamisel;
- Viga arvutamisel;
- Viga lugeja arvutamisel;
- Viga märgi sisestamisel;
- Viga ühise nimetaja arvutamisel;
- Kui täisosa on null, siis seda ei kirjutata.



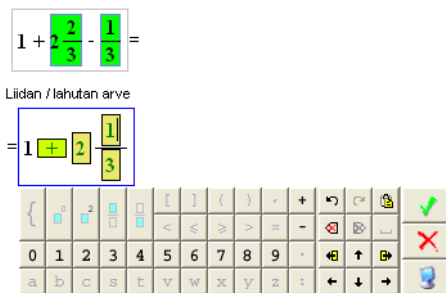
**Joonis 33.** Reegli *Liidan/lahutan arve* tulemuse sisestamine struktuurses režiimis, märgitud liikmete hulgas on ainult täisarvud ja kümnendmurrud



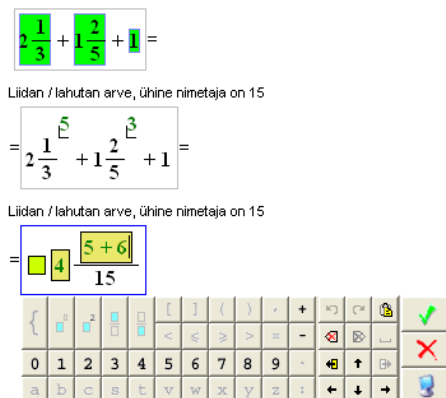
**Joonis 34.** Reegli *Liidan/lahutan arve* tulemuse sisestamine struktuurses režiimis, märgitud liikmete hulgas on ainult samanimelised harilikud murrud



**Joonis 35.** Reegli *Liidan/lahutan arve* tulemuse sisestamine struktuurses režiimis, märgitud liikmete hulgas on ainult harilikud murrud, millest vähemalt kaks on omavahel erinimelised

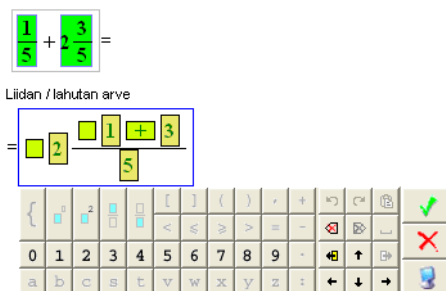


**Joonis 36.** Reegli *Liidan/lahutan arve* tulemuse sisestamine struktuurses režiimis, märgitud liikmete hulgas on vähemalt üks segaarv ja kõik märgitud harilikud murrud ja segaarvud on samanimelised



**Joonis 37.** Reegli *Liidan/lahutan arve* tulemuse sisestamine struktuurses režiimis, märgitud liikmete hulgas on vähemalt üks segaarv ja vähemalt kaks omavahel erinevat harilikku murdu või segaarvu

**Tulemuse sisestamine (osalises sisestamise režiimis):** Erineb struktuursest režiimist ainult selle poolest, et uuele reale kopeeritud avaldises on murru lugejasse tulemuse sisestamiseks summa või vahe iga liikme ja märgi jaoks eraldi kast (joonis 38). Peale sisestuse kinnitamist teostab programm üldised kontrollid struktuurse režiimi kirjelduses toodud erisustega ning lubab lisaks lugejas unaarse plussmärgi sisestamata jätta.



**Joonis 38.** Reegli *Liidan/lahutan arve* tulemuse sisestamine osalises režiimis

**Veateated peale tulemuse sisestamist (osalises sisestamise režiimis):**

- Tulemus sisestamata;
- Lugeja sisestamata;
- Lugejas arv sisestamata;

- Lugejas arvude vaheline märk sisestamata;
- Nimetaja sisestamata;
- Täisosa sisestamata;
- Avaldise märk sisestamata;
- Viga sisestamisel;
- Viga arvutamisel;
- Viga lugeja arvutamisel;
- Viga lugeja liikme arvutamisel;
- Viga märgi sisestamisel;
- Viga ühise nimetaja arvutamisel.
- Viga täisosa arvutamisel;
- Kui täisosa on null, siis seda ei kirjutata.

### 3.2.10 Reegel *Korrutan/jagan arve*

**Rakendused:** Kümnen- ja harilikke murde sisaldava korrutise või jagatise välja arvutamine.

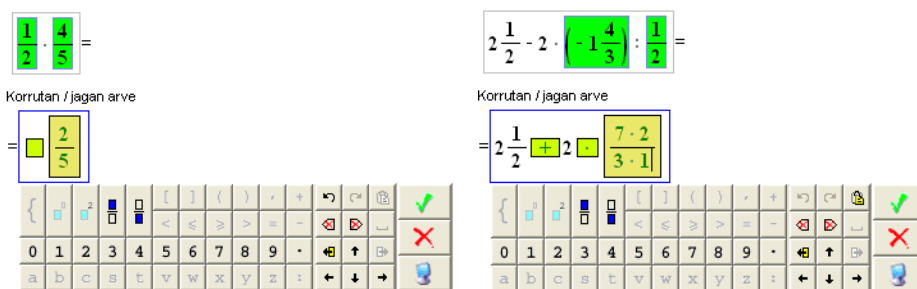
**Juhend märkimisel:** Vali täisarvud, kümnendmurrud, harilikud murrud või segaarvud, mida tahad korrutada/jagada.

**Märkimine:** Märkida tuleb sama korrutise (jagatise) need liikmed, mida korrutada (jagada). Valitavate liikmete arv pole piiratud ja valitavad liikmed ei pea paiknema korrutises (jagatises) kõrvuti. Programm ei luba samaaegselt valida kümnen- ja harilikke murde. Harilikud murrud ja segaarvud tuleb kõigepealt eraldi reegli abil kümnendmurdudeks teisendada või vastupidi. Korrutise (jagatise) liikmed võivad sulgudes olla, sellisel juhul tuleb vastav liige koos sulgudega valida.

**Tulemuse sisestamise juhend:** Sisesta korrutamise/jagamise tulemus.

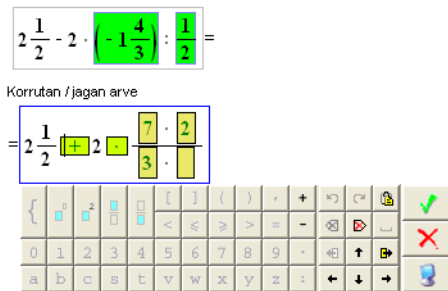
**Tulemuse sisestamine (vabas sisestamise režiimis):** Programm kopeerib avaldise uuele reale ning kuvab märgitud korrutise või jagatise liikmete asemele kastid tulemuse sisestamiseks (joonis 39). Uues avaldises leidub alati kast kogu korrutise või jagatise märgi (pluss või miinus) ja arvulise

tulemuse sisestamiseks. Kui arvuline tulemus pole korrutise või jagatise esimene liige, siis leidub lisaks kast arvulisele tulemusele eelneva korrutamise või jagamise märgi sisestamiseks (joonis 39 paremal). Arvulise tulemuse kasti sisestatavate sümbolite arv sõltub märgitud liikmetest. Kui märgitud liikmete hulgas on ainult täisarvud ja kümnendmurrud, siis saab sisestada numbreid ja murrujoont. Kui märgitud liikmete hulgas leidub vähemalt üks harilik murd või segaarv, siis saab sisestada ka korrutamise märki. Kasti võib sisestada kohe lõpptulemuse (joonis 39 vasakul) või mõne olulise vahetulemuse (joonis 39 paremal). Peale sisestuse kinnitamist rakendab programm üldiseid kontrole erisusega, et unaarse plussmärgi võib jätta sisestamata. Lisaks kontrollib programm, et kasutaja arvulise avaldise kasti kordseid murde ei sisestaks.



**Joonis 39.** Reegli *Korrutan/jagan arve* tulemuse sisestamine vabas režiimis

**Tulemuse sisestamine (struktuurses ja osalises sisestamise režiimis):** Erineb vabast sisestamise režiimist ainult juhul kui märgitud liikmete hulgas on vähemalt üks harilik murd või segaarv. Arvulise tulemuse sisestamiseks on murru lugejas ja nimetajas iga teguri jaoks eraldi kast. Murrujoon ja tegurite vahelised korrutamise märgid on mittemuudetaval kujul uuele reale kopeeritud avaldises juba olemas (joonis 40). Peale sisestuse kinnitamist rakendab programm üldiseid kontrole erisusega, et unaarse plussmärgi ning tegurid 1 murru lugejas ja nimetajas võib sisestamata jätta (vähemalt üks kast peab siiski lugejas ja nimetajas täidetud olema). Lisaks kontrollib T-algebra, et reegli rakendamisel ei laiendataks ega taandataks.



**Joonis 40.** Reegli *Korrutan/jagan arve* tulemuse sisestamine struktuurses ja osalises režiimis

### 3.2.11 Reegel *Võrdlen murde*

**Rakendused:** Spetsiaalne reegel, mille abil saab harjutada arvude võrdlemist.

**Juhend märkimisel:** Vali võrreldavad murrud, segaarvud või kümnendmurrud.

**Märkimine:** Kuna reegel on rakendatav ainult avaldisele, milles on kaks omavahel tühikuga eraldatud arvu, siis midagi märkima ei pea. Programm ei loe kogu avaldise märkimist siiski veaks.

**Tulemuse sisestamise juhend:** Pane õige võrdlusmärk.

**Tulemuse sisestamine (kõigis režiimides):** Programm kopeerib võrreldavad arvud uuele reale ning kuvab nende vahele kasti võrdlusmärgi sisestamiseks (joonis 41). Peale sisestuse kinnitamist kontrollib programm sisestatud märgi õigsust.



**Joonis 41.** Reegli *Võrdlen murde* tulemuse sisestamine vabas, struktuurses ja osalises režiimis

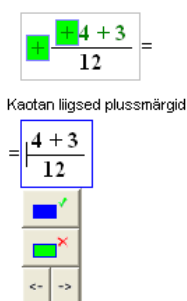
### 3.2.12 Reegel *Kaotan liigsed plussmärgid*

**Rakendused:** Arvutuste käigus võib teisendatavasse avaldisse tekkida unaarseid plussmärke. See reegel on mõeldud unaarsete plussmärkide kaotamiseks.

**Juhend märkimisel:** Vali liigsed plussmärgid, mida kaotada.

**Märkimine:** Märkima peab ühe või rohkem unaarset plussmärki.

**Tulemuse sisestamine (kõigis režiimides):** Tulemust ei ole vaja sisestada. Peale märkimise kinnitamist kopeerib programm avaldise uuele reale ning kustutab selles märgitud plussmärgid (joonis 42).



Joonis 42. Reegli *Kaotan liigsed plussmärgid* rakendamine

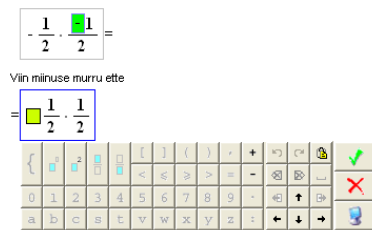
### 3.2.13 Reegel *Viin miinused murru ette*

**Rakendused:** Arvutuste käigus võib teisendatavasse avaldisse tekkida murde, mille lugejas või nimetajas oleva avaldise ees on unaarne miinusmärk. Kirjeldatav reegel võimaldab murru lugejast ja nimetajast sellised miinusmärgid murdu sisaldava korrutise ette viia. Kui murd pole korrutise liige, siis saab märgi vahetult murru ette viia.

**Juhend märkimisel:** Vali murru lugejast ja/või nimetajast miinusmärk, mida korrutise ette tuua.

**Märkimine:** Märkima peab ühe korrutise/jagatise piires murdude lugejate ja nimetajate unaarseid miinusmärke.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, milles on märgistusele vastava korrutise või jagatise (erijuhul märgitud murru) ees kast märgi sisestamiseks (joonis 43). Programm teostab üldised kontrollid erisusega, et unaarse plussmärgi võib sisestamata jätta.



Joonis 43. Reegli *Viin miinused murru ette* tulemuse sisestamine

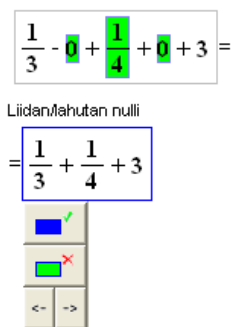
### 3.2.14 Reegel *Liidan/lahutan nulli*

**Rakendused:** Arvutuste käigus võib teisendatavasse avaldisse tekkida summasid või vahesid, mille üks või rohkem liiget on nullid. Ka ülesande läheteavaldis võib sellisel kujul olla. Summas või vahes esinevatest nullidest on võimalik vabaneda rakendades reeglit *Liidan/lahutan arve*. Kirjeldatav reegel pakub selleks mugavamat viisi.

**Juhend märkimisel:** Vali summa/vahe, kus null on liidetav või lahutatav.

**Märkimine:** Märkima peab summa või vahe vähemalt kaks liiget, millest vähemalt üks on null.

**Tulemuse sisestamine (kõigis režiimides):** Tulemust ei ole vaja sisestada. Peale märkimise kinnitamist kopeerib programm avaldisse uuele reale ning kustutab selles märgitud nullid (joonis 44). Kui kõik märgitud summa või vahe liikmed on nullid, jätab programm ühe nulli alles.



Joonis 44. Reegli *Liidan/lahutan nulli* rakendamine



### 3.2.15 Reegel *Korrutan/jagan nulli*

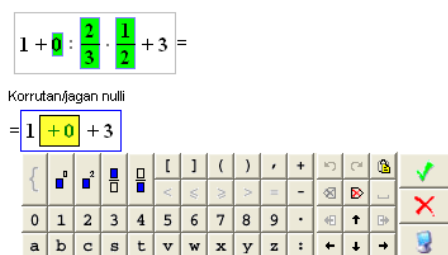
**Rakendused:** Arvutuste käigus võib teisendatavasse avaldisse tekkida korrutisi, kus mõni tegur on null või jagatisi, kus jagatav on null. Ka ülesande lähteavaldis võib sellisel kujul olla. Korrutises või jagatises esinevatel nullidest on võimalik vabaneda rakendades reeglit *Korrutan/jagan arve*. Kirjeldatav reegel pakub selleks mugavamalt viisi.

**Juhend märkimisel:** Vali korrutis, kus üheks teguriks on null, või jagatis, kus jagatav on null.

**Märkimine:** Märkima peab sellise korrutise või jagatise kõik liikmed, milles leidub tegur null või jagatav null. T-algebra ei võimalda reegli paralleelset rakendamist, korraga saab teisendada ainult ühte korrutist või jagatist.

**Tulemuse sisestamise juhend:** Sisesta tulemus.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, milles on kast märgitud osale vastava tulemuse sisestamiseks (joonis 45). Peale sisestuse kinnitamist teostab programm üldised kontrollid. Lisaks, kui sisestuskast on summa või vahe mitteesimene liige, kontrollib T-algebra kasti sisestatud avaldise alguses summa või vahe märgi olemasolu.



Joonis 45. Reegli *Korrutan/jagan nulli* tulemuse sisestamine

### 3.2.16 Reegel *Kaotan murru lugejaga null*

**Rakendused:** Arvutuste käigus võib teisendatavasse avaldisse tekkida murde, mille lugeja on null. Kirjeldatav reegel võimaldab sellistest murdudest vabaneda.

**Juhend märkimisel:** Vali murd, kus lugeja on null.

**Märkimine:** Märkima peab murru, mille lugeja on null, või segaarvu, mille murdosa lugeja on null. T-algebra ei võimalda reegli paralleelset rakendamist, korraga saab teisendada ainult ühe murru.

**Tulemuse sisestamise juhend:** Sisesta tulemus.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, milles on kast märgitud osale vastava tulemuse sisestamiseks (joonis 46). Peale sisestuse kinnitamist teostab programm üldised kontrollid. Seejärel kontrollib T-algebra, et kasti oleks sisestatud täpselt null ja loeb veaks muud nulliga ekvivalentsed avaldised. Lisaks, kui sisestuskast on summa või vahe mitteesimene liige, kontrollib T-algebra kasti sisestatud nulli eesplussi või miinuse olemasolu.



**Joonis 46.** Reegli *Kaotan murru lugejaga null* tulemuse sisestamine

### 3.2.17 Reegel *Kaotan ühe murru nimetajast*

**Rakendused:** Arvutuste käigus võib teisendatavasse avaldisse tekkida murde, mille nimetaja on üks. Kirjeldatav reegel võimaldab sellistest murdudest vabaneda.

**Juhend märkimisel:** Vali murd, kus nimetajaks on arv üks.

**Märkimine:** Märkima peab murru, mille nimetaja on üks. T-algebra ei võimalda reegli paralleelset rakendamist, korraga saab teisendada ainult ühe murru.

**Tulemuse sisestamise juhend:** Sisesta tulemus.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, milles on kast märgitud osale vastava tulemuse sisestamiseks (joonis 47). Peale sisestuse kinnitamist teostab programm üldised kontrollid. Seejärel kontrollib T-algebra, et kasti sisestatud avaldis oleks märgitud murru lugejaga täpselt samal kujul. Lisaks, kui sisestuskast on summa või vahe mitteesimene liige, kontrollib T-algebra kasti sisestatud avaldise alguses summa või vahe märgi olemasolu.



Joonis 47. Reegli *Kaotan ühe murru nimetajast* tulemuse sisestamine

### 3.2.18 Reegel *Jagan ühega*

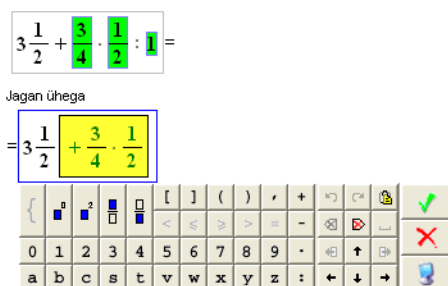
**Rakendused:** Reeglit kasutatakse ühega jagamistest vabanemiseks.

**Juhend märkimisel:** Vali jagatis, kus jagajaks on üks.

**Märkimine:** Märkima peab jagatise, kus jagajaks on üks. T-algebra ei võimalda reegli paralleelset rakendamist, korruga saab teisendada ainult ühte jagatist. Kui jagatises esineb jagajana rohkem kui üks üks, siis saab need korruga ära kaotada.

**Tulemuse sisestamise juhend:** Sisesta tulemus.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, milles on kast märgitud osale vastava tulemuse sisestamiseks (joonis 48). Peale sisestuse kinnitamist teostab programm üldised kontrollid. Seejärel kontrollib T-algebra, et kasti oleks sisestatud muutmata kujul kõik eelmisel real märgitud jagatise liikmed peale jagajateks olevate ühete. Lisaks, kui sisestuskast on summa või vahe mitteesimene liige, kontrollib T-algebra kasti sisestatud avaldise alguses summa või vahe märgi olemasolu.



Joonis 48. Reegli *Jagan ühega* tulemuse sisestamine

### 3.2.19 Reegel *Korrutan ühega*

**Rakendused:** Reeglit kasutatakse ühega korrutamistest vabanemiseks.

**Juhend märkimisel:** Vali korrutis, kus teguriks on üks.

**Märkimine:** Märkima peab korrutise, kus vähemalt üheks teguriks on üks. T-algebra ei võimalda reegli paralleelset rakendamist, korruga saab teisendada ainult ühte korrutist. Kui korrutises esineb tegurina rohkem kui üks üks, siis saab need korruga ära kaotada.

**Tulemuse sisestamise juhend:** Sisesta tulemus.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, milles on kast märgitud osale vastava tulemuse sisestamiseks (joonis 49). Peale sisestuse kinnitamist teostab programm üldised kontrollid. Seejärel kontrollib T-algebra, et kasti oleks sisestatud muutmata kujul kõik eelmisel real märgitud korrutise liikmed peale teguriteks olevate ühede. Lisaks, kui sisestuskast on summa või vahe mitteesimene liige, kontrollib T-algebra kasti sisestatud avaldise alguses summa või vahe märgi olemasolu.

$3 \frac{1}{2} + \frac{3}{4} \cdot \frac{1}{2} \cdot 1 \cdot 1 =$

Korrutan ühega

$= 3 \frac{1}{2} + \frac{3}{4} \cdot \frac{1}{2}$

{	√	√	√	√			(	)	,	+	←	↵	✓
0	1	2	3	4	5	6	7	8	9	·	↵	↑	✗
a	b	c	s	t	v	w	x	y	z	:	←	↓	↵

Joonis 49. Reegli *Korrutan ühega* tulemuse sisestamine

### 3.2.20 Reegel *Jagan piisava täpsuseni*

**Rakendused:** Spetsiaalne reegel, mida kasutatakse ainult kümnendlähendi leidmise ülesandes.

**Juhend märkimisel:** Vali jagatis, murd või segaarv, mida piisava täpsuseni jagada.

**Märkimine:** Märkima peab kahe täisarvu jagatise, murru või segaarvu.

**Tulemuse sisestamise juhend:** Sisesta jagamise tulemus vähemalt ühe liigse kohaga.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, milles on kast märgitud osale vastava tulemuse sisestamiseks

(joonis 50). Kümnenlähendi leidmise ülesannetel on kohustuslik sisendparameter, mis ütleb mitu kohta peale koma ümardatult tuleb kümnenlähend leida. Programm kontrollib, et sisestatud tulemus oleks sellest vähemalt üks koht rohkem, see võimaldab jagamise tulemust hiljem nõutud täpsuseni ümardada.



**Joonis 50.** Reegli *Jagan piisava täpsuseni* tulemuse sisestamine, ülesandes nõutakse kümnenlähendi leidmist ümardatult kolm kohta peale koma

### 3.2.21 Reegel *Ümardan*

**Rakendused:** Spetsiaalne reegel, mida kasutatakse ainult kümnenlähendi leidmise ülesandes.

**Juhend märkimisel:** Vali kümnenmurd, mida ümardada.

**Märkimine:** Märkima peab ühe kümnenmuru.

**Tulemuse sisestamise juhend:** Sisesta ümardamise tulemus.

**Tulemuse sisestamine (kõigis režiimides):** Programm kuvab uuel real avaldise, milles on kast märgitud kümnenmurrule vastava ümardamise tulemuse sisestamiseks (joonis 51). Programm kontrollib, et ümardatud oleks ülesandes nõutud kohani.



**Joonis 51.** Reegli *Ümardan* tulemuse sisestamine, kui ülesandes nõutakse kümnenlähendi leidmist ümardatult kolm kohta peale koma

### 3.3 Disainitud ülesandetüübid T-algebras

T-algebra võimaldab õpilasel lahendada peaaegu kõiki Eesti õpikutes esinevaid harilike murdudega seotud tehnilisi ülesandeid. T-algebras on realiseeritud järgmised harilike murdude teema ülesandetüübid:

- *Taanda harilik murd (taandumatu murruni);*
- *Laienda etteantud nimetajani;*
- *Teisenda murrud ühenimelisteks;*
- *Võrdle murde;*
- *Liida/lahuta ühenimelised murrud ja kui võimalik, siis taanda vastus;*
- *Teisenda segaarv liigmurruks;*
- *Eralda liigmurru täis- ja murdosa;*
- *Liida/lahuta samanimelise murdosaga segaarvud ja kui võimalik, siis taanda vastus;*
- *Liida/lahuta erinimelised murrud ja kui võimalik, siis taanda vastus;*
- *Teisenda kümnendmurd harilikuks murruks ja kui võimalik, siis taanda vastus;*
- *Teisenda harilik murd lõplikuks kümnendmurruks;*
- *Leia hariliku murru või segaarvu kümnendlähend ümardatud täpsuseni ... kohta peale koma;*
- *Korruta harilikud murrud;*
- *Leia pöördarv;*
- *Jaga harilikud murrud;*
- *Arvuta kümnend- ja harilikke murde sisaldava avaldise väärtus.*

Iga ülesandetüübi jaoks on programmis kirjeldatud spetsiifiline lahendamise algoritm. T-algebra oskab ise ülesandeid lahendada ja aidata vastavalt sellele algoritmile ka õpilast. Lihtsamate ülesandetüüpide (nt *Laienda etteantud nimetajani*, *Taanda harilik murd (taandumatu murruni)* jm) reeglikomplekt koosneb ainult mõnest üksikust reeglist. Vastavate ülesannete lahendus koosneb enamasti vaid ühest või kahest sammust. Need ülesandetüübid on mõeldud selleks, et õpilane saaks harjutada hiljem lahendatavate arvutusülesannete üksikuid samme. Keerukamad ülesandetüübid (nt *Arvuta kümnen- ja harilikke murde sisaldava avaldise väärtus*) sisaldavad seevastu peaaegu kõiki murdude teema reegleid.

Selles jaotises esitan mõnede disainitud ülesandetüüpide kirjeldused, tuues iga näite korral välja järgmised omadused:

- lähteavaldis;
- parameetrid, mida ülesande koostaja sisestama peab;
- reeglid, mida õpilane ülesandeid lahendades kasutada saab;
- programmi poolt kasutatav lahendusalgoritm, so järjestatud nimekirjareeglitest (algoritm ei pruugi sisaldada kõiki reegleid, mida õpilane lahendamisel kasutada saab);
- programmi poolt genereeritud näidislahendus;
- programmi poolt aktsepteeritav lahendatud kuju.

### 3.3.1 Ülesandetüüp *Taanda harilik murd (taandumatu murruni)*

**Lähteavaldis:** Murd, mille lugejas ja nimetajas on täisarv või täisarvude korrutis ning lugejal ja nimetajal on ühest suurem ühine tegur.

**Parameetrid:** Puuduvad.

**Reeglid:**

- *Taandan;*
- *Korrutan/jagan arve;*
- *Kaotan liigsed plussmärgid;*
- *Kaotan ühe murru nimetajast.*

**Lahendusalgorithm:** Lahendusalgorithm sisaldab samas järjestuses kõiki eelmises punktis toodud reegleid.

**Näidislahendus (struktuurses sisestamise režiimis):** vt joonis 52.

The image shows a sequence of calculator screens illustrating the division of fractions:

- Initial expression:  $\frac{12}{5} \div \frac{3}{5}$
- Conversion to decimal:  $\frac{12}{5} = 2.4$ ,  $\frac{3}{5} = 0.6$ , resulting in  $2.4 \div 0.6$
- Conversion to integers:  $24 \div 6$
- Division:  $24 \div 6 = 4$
- Final result:  $4$

**Joonis 52.** Ülesandetüübi *Taanda harilik murd (taandumatu murruni)* näidislahendus struktuurses sisestamise režiimis

**Lahendatud kuju:** Taandumatu harilik murd, erijuhul kui murru nimetaja on üks, siis täisarv. Ülesandetüüp ei nõua ega võimalda vastuse esitamist segaarvuna, sest koolis tuuakse segaarvu mõiste sisse alles pärast taandamis-ülesannete harjutamist.

### 3.3.2 Ülesandetüüp *Liida/lahuta erinimelised murrud ja kui võimalik, siis taanda vastus*

**Lähteavaldis:** Summa või vahe murrudest, segaarvudest ja täisarvudest. Avaldises võivad esineda ka sulud.

**Parameetrid:** Puuduvad.

**Reeglid:**

- *Kaotan sulud;*
- *Kaotan liigsed plussmärgid;*
- *Laenan;*



- Liidan/lahutan arve;
- Taandan;
- Viin miinuse murru ette;
- Liigmurd segaarvuks;
- Kaotan murru lugejaga null;
- Kaotan ühe murru nimetajast;
- Segaarv liigmurruks;
- Liidan/lahutan nulli.

**Lahendusalgorithm:** Lahendusalgorithm sisaldab samas järjekorras eelmises punktis toodud kõiki reegleid peale kahe viimase.

**Näidislahendus (osalises sisestamise režiimis):** vt joonis 53.

$$\frac{1}{3} - \frac{3}{4} + \frac{2}{3} =$$

Kaotan sulud

$$= \frac{1}{3} - \frac{3}{4} + \frac{2}{3} =$$

Liidan / lahutan arve, ühine nimetaja on 12

$$= \frac{1}{3} - \frac{3}{4} + \frac{2}{3} =$$

Liidan / lahutan arve, ühine nimetaja on 12

$$= 1 \frac{1}{12}$$

Liidan / lahutan arve

$$= 1 \frac{1}{12} =$$

Taandan

$$= 1 \frac{1}{12} =$$

Taandan

$$= 1 \frac{1}{4}$$

Vastus:

$$1 \frac{1}{4}$$

**Joonis 53.** Ülesandetüübi *Liida/lahuta erinimelised murrud ja kui võimalik, siis taanda vastus* osalises sisestamise režiimis

**Lahendatud kuju:** Taandumatu harilik murd või täisarv. Liigmurd tuleb vastuse andmiseks segaarvu kujule teisendada.

### 3.3.3 Ülesandetüüp *Leia hariliku murru või segaarvu kümnendlähend ümardatud täpsuseni ... kohta peale koma*

**Lähteavaldis:** Harilik murd või segaarv.

**Parameetrid:** Arv, mitu kohta peale koma peab kümnendlähend sisaldama.

**Reeglid:**

- *Jagan piisava täpsuseni;*
- *Ümardan.*

**Lahendusalgortm:** Lahendusalgortm sisaldab samas järjestuses kõiki eelmises punktis toodud reegleid.

**Näidislahendus (parameetri väärtus on 4, vabas sisestamise režiimis):** vt joonis 54.



**Joonis 54.** Ülesandetüübi *Leia hariliku murru või segaarvu kümnendlähend ümardatud täpsuseni ... kohta peale koma* näidislahendus vabas sisestamise režiimis, parameetri väärtus on 4

**Lahendatud kuju:** Esialgse hariliku murru või segaarvu kümnendlähend, mis on ümardatud etteantud arv kohti peale koma.

## Kokkuvõte

Käesoleva töö eesmärk oli realiseerida interaktiivse õpikeskkonna T-algebra jaoks harilike ja kümnendmurdudega seotud operatsioonid ja ülesannete tüübid. Autor realiseeris 21 operatsiooni ja 16 ülesandetüüpi.

Projekti käigus õnnestus meil luua dialoogiskeem, mis võimaldab samm-sammulist ülesannete lahendamist sarnaselt traditsioonilisele paberil lahendamisele. Loodud Tegevus-Objekt-Sisend dialoogiskeem jätab õpilasele võimaluse eksida kõigil kolmel lahendussammu etapil: reegli valimisel, avaldise osade märkimisel ja tulemuse sisestamisel. Seega õpilane saab lahendamise käigus vigade kohta vahetut tagasisidet. Sammu igal etapil on võimalik programmi käest abi küsida ning lasta konkreetsed etapid automaatselt lõpetada. Lisaks suudab T-algebra tuvastada, kas ülesanne on lõpuni lahendatud, ja oskab õpilase soovil poolelioleva lahenduse ise ära lõpetada.

Harilike ja kümnendmurdude teema ülesannete kontekstis on oluline ära märkida ka T-algebra segaarvude tugi. Kuna Eestis on matemaatika õpetamisel harilike murdude käsitlemisel segaarvud olulisel kohal, siis on T-algebra murdudega seotud reeglid selliselt disainitud, et kõik vajalikud operatsioonid segaarvudega oleks toetatud. Enamik teisi autori poolt uuritud õpikeskkondi segaarvude tuge ei oma.

Magistritöö suure mahu tõttu ei ole selles tehtud harilike ja kümnendmurdude teemalisi eksperimente õpilastega. Eksperimentide tulemuste analüüs annaks tõenäoliselt uusi ideid T-algebra reeglite, ülesandetüüpide ja üldise dialoogiskeemi täiendamiseks.

# **Solving Problems on Common and Decimal Fractions in Interactive Learning Environment**

**Master Thesis**

**Vahur Vaiksaar**

## **Summary**

The aim of this thesis is to design and implement an interactive learning environment for practicing expression manipulation exercises on common and decimal fractions.

This thesis is part of a larger project, T-algebra, the result of which is an interactive learning environment for step-by-step solving of algebra problems in four areas of school mathematics: calculation of the values of numerical expressions, operations with fractions, solving of linear equations, inequalities and linear equation systems and simplification of polynomials.

Chapter 1 gives an overview of other existing expression manipulation systems. The studied systems are categorized and the actions of the student and of the system are noted for each category. Chapter 2 describes the design and general dialogue scheme of T-algebra. The first part presents the expressions that are allowed in the program, followed by a depiction of the problem solution window of the student program. The third part of Chapter 2 introduces the design of the step dialogue. The last part describes the core of T-algebra. Chapter 3 describes the domain of common and decimal fractions. The first part presents definitions, algorithms and problem types found from exploration of mathematics textbooks. The second part of this chapter describes the rules designed for solving exercises involving common and decimal fractions. The last part introduces composed problem types in the chosen domain and provides a thorough description of selected types.

The project team succeeded in creating a rule dialogue in T-algebra that allows the pupil to learn both the solution algorithms and their steps in a manner, which is similar to solving problems on paper. Each solution step consists of three stages: selecting a transformation rule, marking parts of the expression, and entering the result of the application of the selected rule. The designed three-stage dialogue enables the pupil to make the same mistakes in T-algebra as on paper, while it enables the program to provide help and

immediate feedback about mistakes. The design of the created environment is described in several publications of the T-algebra workgroup [19, 21].

An important achievement, in addition to the general dialogue design, is the support of mixed numbers. Because Estonian schools use mixed numbers extensively in common fraction arithmetic, T-algebra has been built to accommodate this. Most other systems studied by the author have not dealt with this rather inconvenient subject.

## Viited

- [1] Algebrator, Softmath. <http://www.softmath.com/>. Külastuse aeg 10.04.2010.
- [2] Aplusix, IMAG-Leibniz laboratory. <http://aplustix.imag.fr/en/index.html>. Külastuse aeg 10.04.2010.
- [3] Bagatrix, Bagatrix Inc. <http://www.bagatrix.com/>. Külastuse aeg 10.04.2010.
- [4] Beeson, M. (1998). Design Principles of Mathpert: Software to support education in algebra and calculus. *Kogumikus: Computer-Human Interaction in Symbolic Computation. (Toim.)Kajler, N.* Berlin: Springer-Verlag, 1998, 89-115.
- [5] Issakova, M.; Lepp, D. (2004). Rule dialogue in problem solving environment T-algebra. *Kogumikus Proceedings TIME-2004: Montreal International Symposium on Technology and its Integration into Mathematics Education*, 16 lk, 2004.
- [6] Issakova, M.; Lepp, D.; Prank, R. (2005). Input Design in Interactive Learning Environment T-Algebra. *5th IEEE International Conference on Advanced Learning Technologies*; Kaohsiung, Taiwan; 05.-08.07.2005. Los Alamitos, CA, USA: IEEE Computer Soc, 2005, 489 - 491.
- [7] Issakova, M.; Lepp, D.; Prank R. (2006). T-algebra: Adding Input Stage to Rule-Based Interface for Expression Manipulation. *The International Journal for Technology in Mathematics Education*, 13(2):89 - 96.
- [8] Kaasik, K. (2003). Matemaatika 5. klassile. Avita.
- [9] Limberg, K.. <http://www.sluha.pri.ee/stuff/matemaatika/oppematerjal/index.html>. Külastuse aeg 13.04.2010.
- [10] LiveMath, MathMonkeys. <http://www.livemath.com/>. Külastuse aeg 10.04.2010.
- [11] Maple, Waterloo Maple Inc. <http://www.maplesoft.com/>. Külastuse aeg 10.04.2010.

- [12] Mathematica, Wolfram Research. <http://www.wolfram.com/>.  
Külastuse aeg 10.04.2010.
- [13] Maths Connections, Transfinite Research. <http://www.transfinite.f9.co.uk/mathconn/mathcon0.htm>. Külastuse aeg 10.04.2010.
- [14] MathScore, Accurate Learning Systems Corporation. <http://www.mathscore.com/>. Külastuse aeg 10.04.2010.
- [15] MathXpert, Help With Math. <http://www.helpwithmath.com/>.  
Külastuse aeg 10.04.2010.
- [16] Nicaud, J. F.; Bouhineau, D.; Chaachoua, H. (2004). Mixing microworld and CAS features in building computer systems that help students learn algebra. *International Journal of Computers for Mathematical Learning*, 9(2):169-211.
- [17] Nurk, E.; Telgmaa, A. (2004). Matemaatika V klassile. Koolibri.
- [18] Prank, R. (2003). T-algebra projekti ülesandepüstitus, 19 lk.
- [19] Prank, R.; Issakova, M.; Lepp, D.; Vaiksaar, V. (2006). Designing Next-Generation Training and Testing Environment for Expression Manipulation. *Kogumikus: Lecture Notes in Computer Science: ICCS 2006; Reading; 28.-31 May, 2006. (Toim.)Alexandrov, V.N.*. Berlin: Springer-Verlag, 2006, 928 - 931.
- [20] Prank, R.; Issakova, M.; Lepp, D.; Vaiksaar, V.; Tõnisson, E. (2006). Problem Solving Environment T-algebra. *Kogumik: Teaching Mathematics: Retrospective and Perspectives: 7th international conference; Tartu, Estonia; May 12-13 2006. (Toim.)Abel, E.; Kudzma, R.; Lepik, M.; Lepmann, T.; Mencis, J.*. Tartu : TÜ Kirjastus, 2006, 190 - 197.
- [21] Prank, R.; Issakova, M.; Lepp, D.; Tõnisson, E.; Vaiksaar, V. (2007). Integrating Rule-based and Input-based Approaches for Better Error Diagnosis in Expression Manipulation Tasks. Li, S.; Wang, D.; Zhang, J. Z. (Toim.). *Symbolic Computation and Education* (174 - 191). Singapore: World Scientific.

- [22] Prank, R.; Issakova, M.; Lepp, D.; Tõnisson, E.; Vaiksaar, V. (2008). T-algebra - Intelligent Environment for Expression Manipulation Exercises. *Kogumikus: Topic Study Group 22: 11th International Congress on Mathematical Education. (Toim.)*Laborde, C.; Kynigos, C.. Monterrey, Mexico: ICMI - International Commission on Mathematical Instruction, 2008, 1 - 7.
- [23] Prank, R.; Vaiksaar, V. (2009). Operations with fractions in the learning environment T-algebra. Lepik, M. (Toim.). Teaching Mathematics: Retrospective and Perspectives. Proceedings of the 10th International Conference (289 - 296). Tallinn: Tallinn University.
- [24] WIMS, Gang, X.. [http://wims.unice.fr/wims/en\\_home.html](http://wims.unice.fr/wims/en_home.html). Külastuse aeg 10.04.2010.
- [25] WIRIS, Maths for More. <http://www.wiris.ee/>. Külastuse aeg 10.04.2010.
- [26] Wisc-Online, Wisconsin Technical College System. <http://www.wisc-online.com/>. Külastuse aeg 10.04.2010.
- [27] Young Einstein Mathematics, Maths Practice. <http://www.mathspractice.com.au/>. Külastuse aeg 10.04.2010.