TARTU ÜLIKOOL

MATEMAATIKA-INFORMAATIKATEADUSKOND

MATEMAATILISE STATISTIKA INSTITUUT

Joosep Raudsik

Fellegi-Holt´i meetod ja deduktiivne imputeerimine andmestiku *Väliskülastajad Eestis* näitel

Magistritöö (30 EAP)

Juhendajad:

Maiki Ilves

Mare Vähi

TARTU

2015

Fellegi-Holt´i meetod ja deduktiivne imputeerimine andmestiku *Väliskülastajad Eestis* näitel

Kokkuvõte

Käesolev töö annab ülevaate Fellegi-Holt´i paradigmast ning erinevatest võimalustest selle lahendamiseks. Antud põhimõtet kasutatakse tihti automaatses andmete parandamises. Lähenemine seisneb vea tuvastamise probleemi lahendamises läbi järgneva põhimõtte: muuta võimalikult väike (kaalutud) arv väärtusi nii, et vaatlus vastaks defineeritud reeglitele. Neljandas peatükis demonstreeritakse paradigma võimalik kasu keerulisema kontrollreeglite struktuuri korral. Töö teises pooles tutvustatakse deduktiivset imputeerimist, mis samuti põhineb kontrollreeglitel – täpsemalt kasutades Eesti Statistikaameti andmestikku *Väliskülastajad Eestis.* Töö tulemusena saab olla veendunud, et kontrollreeglite hulga lihtsus vähendab paradigma efektiivsust ja algoritm teeb valikuid tulenevalt juhuslikkusest ning ei sobi antud andmestikule.

Võtmesõnad: Automaatne andmete parandamine; imputeerimine.

Fellegi-Holt paradigm and deductive imputation based on *Foreign Visitors in Estonia* dataset

Summary

This paper gives an overview of the Fellegi-Holt paradigm and the methods for solving it. The mentioned principle is broadly used for automated editing worldwide. It approaches the error localization problem by stating that the data of a record should be made to satisfy all edits by changing the fewest possible (weighted) number of fields. As seen in the fourth paragraph the paper the benefit of the paradigm is evident using a bit more complex sets of edit rules. The second part of the paper describes deductive imputation, what also is based on the edit rules defined for the dataset. In the last part of this paper we investigate the efficiency of the described methods under more plain sets of edit rules - more precisely using the Estonian Statistics Office´s dataset *Foreign Visitors in Estonia* as an example. As a conclusion it is certain

that the simplicity of the edit rules set undermines the efficiency of the Fellegi-Holt paradigm and forces the algorithms to use a high level of randomness in the process, what makes applying deductive imputation afterwards hardly a reasonable tool to use.

Key words: Automated editing; imputation.

Sisukord

1. Sissejuhatus 5
2. Vigade liigid 6
3. Kontrollreeglid 8
4. Fellegi-Holt´i paradigma 10
   4.1. Andmete automaatne parandamine 10
   4.2. Paradigma kirjeldus 11
   4.3. Matemaatiline esitus 11
   4.4. Algoritmid arvulistele andmetele 12
      4.4.1. Fellegi ja Holti väljapakutud lahendus ja Fourier-Motzkini eemaldus 12
      4.4.2. *Harude ja tõkete* algoritm 15
      4.4.3. *Lõigatud tasandite* põhimõte 19
      4.4.4. Tippude moodustamise lähenemine 21
   4.5. Algoritm segaandmetele 22
      4.5.1. *Harude ja tõkete* algoritm segaandmetele 23
5. Deduktiivne imputeerimine 26
   5.1. Bilansireeglite alusel 26
   5.2. Mitte-negatiivsus piirangute kasutamine 28
   5.3. Kasutades faktortunnuseid 29
6. Andmestiku *Väliskülastajad Eestis* automaatne parandamine 32
   6.1. Bilansireegleid sisaldavad struktuurid 33
   6.2. KUI-SIIS lauseid sisaldavad struktuurid 34

Kasutatud kirjandus 36

Lisa 1 – Küsitluse ankeet 37

Lisa 2 – Andmestiku *Väliskülastajad Eestis* kontrollreeglid 40

Lisa 3 – Rakendustarkvara R kood 43

1. Sissejuhatus

Käesoleva töö eesmärk on: esmalt Fellegi-Holt´i paradigma tutvustamine koos erinevate lahenduste tutvustamisega, teiseks ülevaate andmine deduktiivsest imputeerimisest ning kolmandaks tutvustatud meetodite rakendamine andmestikule *Väliskülastajad Eestis*.

Töö esimene osa annab ülevaade Fellegi-Holt´i paradigmast ja selle lahendamisest erinevate algoritmide põhjal. Tegemist on põhimõttega, mida kasutatakse automaatses andmete parandamises, et teha kindlaks millised tunnuse väärtused tuleb muuta selleks, et vaatlus vastaks andmestikule defineeritud kontrollreeglitele. Erinevate lahendustena tutvustatakse töös Fellegi ja Holt´i poolt välja pakutud meetodit, *harude ja tõkete* algoritmi, *lõigatud tasandite* põhimõtet ja tippude moodustamise lähenemist.

Teises pooles tutvustatakse deduktiivset imputeerimist. Tegemist on ainsa imputeerimise meetodiga, mille korral ei tehta eeldusi imputeeritava tunnuse jaotuse kohta, vaid väärtused on tuletatavad kontrollreeglite põhjal. Erinevate võimalustena tutvustatakse imputeerimist bilansireeglite alusel, kasutades mitte-negatiivsus piiranguid ning faktortunnuste korral.

Töö viimases osas rakendatakse ning uuritakse tutvustatud meetodite sobivust Eesti Statistikaameti poolt valitud andmestikule *Väliskülastajad Eestis.*

2. Vigade liigid

Sõltumata valdkonnast kaasneb andmete kogumisega tihtipeale vigu. Võimalike veaallikatena on arvestatav osatähtsus nii andmete kogumisel, sisestamisel kui mõõtevigadel. On oluline mõista, et erineva veaallikaga kaasnevad erinevad vead ning sellest tulenevalt on ka nende käsitlemine isesugune. Parim võimalus vigadega tegelemiseks on sooritada uus mõõtmine, kuid alati ei ole see võimalik või on vaadeldud väärtus juba muutunud.

Vigu on võimalik liigitada mitmel viisil (Memobust: Statistical Data Editing 2014). Esimene oluline erinevus on süstemaatiliste ja juhuslike vigade vahel. Teine võimalik liigitamise viis on arvesse võtta vaatluse mõju suurust ning kolmandaks, kas tegu on erindiga või mitte.

Süstemaatiline viga esineb vastajatel korduvalt, näiteks kui vastaja mõistab või loeb valesti ankeedis olnud küsimust või eksitakse ühikutega - küsitakse inimese pikkust sentimeetrites, aga vastaja annab vastuse meetrites. Süstemaatiline viga viib kergesti nihkeni ka tehtud järeldustes. Samas on antud liiki eksimust suhteliselt lihtne parandada, sest tekkiva vea mehhanism on teada.

Juhuslikud vead, nagu nimi ka ütleb, ei oma süstemaatilist allikat. Näiteks andmete sisestamisel tekkiv eksitus, kus vastaja sisestab kavatsetust ühe numbri rohkem. Andmetöötluses võetakse tihti antud liiki vea tekkimise tõenäosus võrdseks nulliga. Juhusliku veaga võib kaasneda tunnuse (või tunnuste hulga) erandlik väärtus, mis aitab meil sellist liiki vea tuvastada ja parandada.

Mõjukat liiki vigadel on oluline mõju tehtavatele järeldustele. Nende töötlemiseks kasutatakse enamasti valikulist töötlemist, mis jääb antud töö raamidest välja.

Erindi näol on tegemist vaatlusega, mis ei ole kooskõlas varasemate teadmistega väärtuste muutumispiirkonna kohta või ülejäänud vaatlustega. Vaatluse erindiks arvamine võib olla nii õigustatud, aga ka ekslik. Arvates erindi analüüsist välja, kui tegu oli lihtsalt ekstreemse vaatlusega kaotame informatsiooni üldkogumi kohta ning vastupidisel juhul kaasame analüüsi vigast väärtust (või vigaseid väärtuseid) sisaldava tunnuse. Mõlemal juhul on tehtavad

järeldused kallutatud. Sellest tulenevalt on vaja nende kindlaks tegemiseks usaldusväärseid meetodeid (Ghosh-Dastier & Schafer 2003). On oluline mõista, et ainult teadmine mittesobivusest koostatud mudeli või teiste andmetega ei tähenda, et tegu on veaga. Erindid on tihedalt seotud mõjukate vaatlustega ning vaatlus, mis on erind, on tihti ka mõjukas vaatlus ja vastupidi.

3. Kontrollreeglid

Automaatses andmete parandamises on oluline roll kontrollreeglitel (edaspidi ka lihtsalt reegel) ehk eeldefineeritud tingimustel, millele iga vaatlus või mingi tunnuste hulk peab vastama. Üks võimalus leida andmetest vigu on uurida täpsemalt reeglitele mittevastavaid vaatluseid. Kui vaatlus on kooskõlas kõigi reeglitega nimetame vaatlust järjepidevaks.

Tihti eristatakse rangeid ning leebeid reegleid, millest esimesed peavad kehtima igal juhul, näiteks on inimese pikkus alati positiivne suurus. Teisel juhul tehakse mõningased järeleandmised ning reeglile mittevastamine viitab ebatavalistele väärtustele, mille käsitlemist töö seatud narratiiv ei nõua.

Lineaarseid kontrollreegleid eksisteerib kolme tüüpi:

- väärtused piiratud vahemikus,
- bilansi reeglid,
- piiratud väärtuste suhe.

Olgu x_i ($i \in N$) tunnuste väärtused. Arvulisi tunnuseid sisaldavad tingimused on esitatavad järgneval kujul (Memobust: Automated editing 2014):

$$a_{j1}x_1 + \ldots + a_{jn}x_n + b_j \geq 0 \qquad (3.1)$$

ja

$$a_{j1}x_1 + \ldots + a_{jn}x_n + b_j = 0 \qquad (3.2)$$

kus j näitab reegli järjekorranumbrit, a_{ij} on koenfitsendid ja b_j konstandid. Märgime siinkohal, et reegel, mis sätestab suhte kahe muutuja vahel:

$x1/x2 \geq a$

on esitatav eelmainitud kujul:

$x_1 - ax_2 \geq 0$.

Faktortunnustele vastavaid reegleid kirjeldatakse KUI-SIIS (ing. IF-THEN) lausete abil, millest annab täpsema ülevaate peatükk 4.5.

4. Fellegi-Holt´i paradigma

Käesolevas peatükis käsitletakse 1976. aastal Fellegi ja Holti poolt välja pakutud paradigmat (edaspidi ka põhimõte) andmete automaatseks parandamiseks. Järgnevaga antakse ülevaade andmete automaatsest parandamisest ning antud meetodi rollist sellel maastikul, paradigma üldisest põhimõttest ja matemaatilisest kujust ning erinevatest algoritmidest selle lahendamiseks. Lahendamiseks kasutatavate algoritmide kirjeldused põhinevad raamatul Statistical Data Editing and Imputation (De Waal, Pannekoek & Scholtus 2011) ning näited on koostatud töö autori poolt, kui ei ole öeldud teisiti.

4.1. Andmete automaatne parandamine

Traditsiooniliselt on andmete parandamine olnud manuaalne protsess, mis tähendab ulatuslikke teadmisi andmete iseloomu kohta. Aja möödudes on muutunud arvuti poolt tehtavad arvutused „odavamaks“, millest tulenevalt nõuavad rohkem tähelepanu meetodid, mis ei eelda kasutajapoolset sisendit ning võivad olla arvutuslikult keerukamad. Andmete parandamine on tihti jagatud sammudeks, mille üks osa on automaatne, eesmärgiga leida juhusliku allikaga vead. Seda probleemi nimetatakse vea tuvastamise probleemiks. Meetodeid, mis ei nõua kasutajapoolset sisendit on kahte liiki. Käesolevas peatükis käsitletav põhineb optimiseerimisel, kus võetakse aluseks andmestikule defineeritud kontrollreeglid. Enamik sellistest praktikas rakendust leidvatest meetoditest põhinevad just Fellegi-Holti paradigmal (Scholtus 2014), mis töötati välja nimitegelaste poolt 1976. aastal. Teise lähenemise korral moodustatakse parandatavatest andmetest statistilisi mudeleid, mille erindeid käsitletakse kui võimalikke vigu andmetes. Meetod töötab paremini väiksema arvu tunnuste korral ning peale parandamist ei pruugi vastata sätestatud kontrollreeglitele (Memobust: Automated editing 2014), millest tulenevalt selle rakendamist täppisteadustes ei soosita. Viimane lähenemine jääb ka antud töö raamidest välja.

4.2. Paradigma kirjeldus

Fellegi-Holt´i põhimõtte rakendamine on praktiseeritavaim meetod andmete automaatseks parandamiseks. Paradigma kohaselt tuleb andmetes muuta võimalikult väike (kaalutud) arv väärtuseid selleks, et andmed vastaks valitud reeglitele. Igale tunnusele omistatakse usaldusväärsust näitav kaal. Ebausaldusväärsetele tunnustele omistatakse väiksem kaal ning usaldusväärsematele suurem (Fellegi & Holt 1976).

Üldine Fellegi-Holti paradigma ütleb, et eesmärk on leida tunnuste alamhulk mille korral:

- tunnuste väärtused x_i ($i \in N$) on võimalik asendada nii, et seatud reeglid kehtiks,
- kõikide võimalike alamhulkade seast valida asendamiseks tunnuste alamhulk, mille korral usaldusväärsust näitavate kaalude summa on vähim.

Enne paradigma kirjeldamist matemaatilise probleemina toome välja mõned selle põhimõtte puudused. Paneme tähele, et Fellegi-Holt´i paradigma ei võta arvesse esialgse ja imputeeritud väärtuse erinevust. Teine puudus seisneb (Memobust: Automated Editing 2014) süstemaatiliste vigade tuvastamata jätmises. Näiteks mingi vastajate grupi poolt valesti tõlgendatud küsimuse vastused või sisestaja poolt numbriklahvi ekslik korduvvajutamine. Lisaks, saame paradigma abil vaid teada muutmist vajavad tunnused – nö. õigeid väärtusi käesolev paradigma välja selgitada ei aita.

Suurim eelis seisneb aga meetodi paindlikkuses – korrigeerimine sõltub iga vaatluse eripärast. Lisaks on praktikas suhteliselt lihtne grupeerida vaatlused, millele soovime rakendada erinevaid kaale ja kontrollreegleid.

4.3. Matemaatiline esitus

Selles alapeatükis kirjeldatakse vea tuvastamise probleemi kui matemaatilist optimiseerimise ülesannet, mida järgnevas alapeatükis erinevate meetoditega lahendama asume. Lahendamiseks käsitleme iga vaatlust eraldi. Lihtsustamaks tähistust jätame ära vaatluse

järjekorranumbrit näitava indeksi ehk tähistame pidevad tunnused x_j *(j=1,...,p)* ning tunnuste väärtused vektoriga *(*x_1*, ... ,* x_p*)*. (De Waal & Coutinho 2005)

Eesmärk on iga vaatluse korral leida tehisvaatlus *(*$\check{x}_1$*, ... ,* $\check{x}_p$*)*, mis rahuldab iga kontrollreeglit nii, et

$$\sum_{j=1}^{p} w_j y_j \qquad (4.1)$$

oleks minimaalne, kus y_j *(j=1,...p)* on defineeritud järgnevalt

$$y_j = \begin{cases} 1, \text{kui } \check{x}_j \neq x_j \text{ või } x_j \text{ on puuduv väärtus} \\ 0, \text{muul juhul} \end{cases}.$$

Valemis 4.1 märgib w_j tunnusele x_j (*j*=1,...,*p*) vastavat usaldusväärsust näitavat kaalu (mida suurem on antud kaal, seda kindlamad saame olla vastava tunnuse korrektsuses). Lahenduseks vea tuvastamise probleemile saame tunnuste hulga, mille väärtused antud vaatluse korral muutmist vajavad.

4.4. Algoritmid arvulistele andmetele

Võimalusi Fellegi-Holt´i paradigma lahendamiseks on mitmeid. Käesolevas alapeatükis antakse ülevaate neljast erinevast lähenemisest. Lisaks märgime ära, et kuna tegu on sama ülesande erisuguste lahendustega, siis lahendid on sõltumata kasutatavast meetodist samad (või sõltuvad juhusest mitme optimaalse lahendi korral). Esmalt kirjeldame Fellegi ja Holti poolt välja pakutud lahendust, teisena *harude ja tõkete* algoritmi, kolmandana *lõigatud tasandite* meetodit ning viimasena lähenemist hulknurga tippude moodustamise abil.

4.4.1. Fellegi ja Holti väljapakutud lahendus ja Fourier-Motzkini eemaldus

Fellegi ja Holt pakkusid 1976. aastal välja enda paradigma lahendamise läbi reeglite täiskomplekti (e. *vajalike reeglite hulk*) ehk kasutades teadaolevaid, aga ka (vajalikke) tuletatavaid reegleid. Viimased on tingimused, mis on järeldatavad algsetest reeglitest ning tekkemehhanismina kasutame Fourier-Motzkini eemaldust, mida enne meetodi sisu avamist ka kirjeldame.

Fourier-Motzkini eemaldus

Enne käsitletavale probleemile lahenduse pakkumist tutvustame Fourier-Motzkini eemaldust, mis mängib vea tuvastamise probleemi lahendamisel olulist rolli. Eemalduse abil saame kontrollreeglite komplektist eemaldada valitud tunnused.

Olgu eesmärgiks avaldada kontrollreeglid reeglite hulgana, mis ei sisalda muutujat x_r. Esimese sammuna kopeerime kõik võrdused, mis antud muutujat ei sisalda uude reeglite hulka ψ. Kui muutuja x_r reeglites esineb, siis valime ühe reeglitest ning avaldame muutuja teiste tunnuste kaudu.

Sisaldugu tunnus x_r reeglis kujul (3.2), saame kirjutada:

$$x_r = -\frac{1}{a_{sr}}(b_s + \sum_{j \neq r} a_{sj} x_j), \qquad (4.2)$$

Järgnevalt asendame valemis 4.2 toodud kuju kõikidesse teistesse reeglitesse, mis sisaldavad muutujat x_r. Oleme saanud uued reeglid, mille lisame hulka ψ.

Viime läbi analoogilise arutelu, kui valitud tunnus sisaldub ainult võrratustes. Meetod seisneb kõigi reeglipaaride läbi vaatamises, kus valitud tunnus kajastatud on. Sisaldugu x_r võrrandites *s* ja t. Eeskirja kohaselt tuleb kindlaks teha, kas x_r-ile vastavad koefitsientide märgid mõlemas reeglis on vastupidised ehk kas $a_{sr} * a_{tr} < 0$. Kui nii ei ole, siis seda reeglite paari edasi ei käsitleta, vastasel juhul saame seada tõkked:

$$x_r \leq -\frac{1}{a_{sr}}(b_s + \sum_{j \neq r} a_{sj} x_j) \qquad (4.3)$$

ja

$$x_r \geq -\frac{1}{a_{tr}}(b_t + \sum_{j \neq r} a_{tj} x_j). \qquad (4.4)$$

Kombineerides võrratusi 4.3 ja 4.4 saame:

$$-\frac{1}{a_{tr}}(b_t + \sum_{j \neq r} a_{tj} x_j) \leq x_r \leq -\frac{1}{a_{sr}}(b_s + \sum_{j \neq r} a_{sj} x_j),$$

millest järeldub vahetult võrrand, mis ei sisalda valitud muutujat:

$$-\frac{1}{a_{tr}}(b_t + \sum_{j \neq r} a_{tj} x_j) \leq -\frac{1}{a_{sr}}(b_s + \sum_{j \neq r} a_{sj} x_j).$$

Käitudes sarnaselt kõigi võimalike võrratuste paaridega, mis sisaldavad muutujat x_r ning lisades tekkinud võrrandid uude reeglite süsteemi ψ, saame uue reeglite hulga, mis ei sisalda valitud tunnust.

Meetodi kirjeldus

Vajalike reeglite hulk tekitatakse valides korduvalt muutuja, mille Fellegi ja Holt nimetasid genereerivaks väljaks. Kõiki reeglite paare käsitletakse (nii tuletatud, kui esialgsed) Fourier-Motzkini eemalduse abil, et teha kindlaks, kas selle reeglite paari abil on võimalik tekitada uus reegel eemaldades valitud muutuja ehk genereeriv väli. Protsessi jätkatakse iteratiivselt läbides nii esijalgsed kui tuletatud reeglid kuni tuletatavaid reegleid enam juurde ei teki. Nii saadakse vajalike reeglite hulk. Viimase sammuna leitakse vajalike reeglite hulgast minimaalne (kaalutud) tunnuste arv, mille muutmisel on võimalik saavutada andmete kooskõla reeglitega. Järeldatavate reeglite hulga tähtsust ning erinevust esialgselt defineeritud reeglitest aitab mõista järgnev näide.

Oletame, et reeglite kohaselt $x_1 < x_2$ ning $x_2 < x_3$ ning eksisteerib vaatlus (x_1, x_2, x_3)=(3, 2, 1). Tunnuse x_2 väärtus on vastuolus mõlema tingimusega, kuid ei eksisteeri ühtegi väärtust, mille korral mõlemad tingimused täidetud saaks. Järelikult on vaja korrigeerida tunnuste x_1 ja $x_3$väärtusi ning ainus mõlema reegliga vastuolus olnud väärtus võib jääda samaks.

Paneme tähele, et kui arvulistele andmetele on defineeritud mingi hulk kontrollreegleid, siis neist on võimalik tuletada lõpmatu hulk kontrollreegleid, näiteks olgu tingimus kujul $x_1 \geq x_2$, siis peab kehtima $\lambda x_1 \geq \lambda x_2$ iga λ korral. Kõikvõimalikud tunnuste hulgad, mille muutmine aitab andmed reeglitega vastavusse viia on sobiv lahendus vea tuvastamise probleemile, mille jaoks ei ole vaja kõiki võimalikke reegleid, vaid ainult vajalike reeglite hulka.

Meetodi paremaks mõistmiseks uurime näidet 4.1. Tegemist on nö. klassikalise näitega, mida kasutatakse tihti välja pakutud lahenduse kirjeldamiseks ning väljastati juba 1976. aastal Fellegi ja Holti poolt.

Näide 4.1

Olgu andmestikule seatud reeglid:

$$x_1 - x_2 + x_3 + x_4 \geq 0 \quad (4.5)$$

$$-x_1 + 2x_2 - 3x_3 \geq 0, \quad (4.6)$$

Millele Fourier-Motzkini eemaldust rakendades saame vajalikud järeldatud reeglid:

$$x_2 - 2x_3 + x_4 \geq 0,$$
$$x_1 - x_3 + 2x_4 \geq 0,$$
$$2x_1 - x_2 + 3x_4 \geq 0.$$

Viis eelnevat kontrollreeglit moodustavad vajalike reeglite hulga.

Lihtsa aritmeetika abil näeme, et esimene neist saadakse valides x_1 genereerivaks väljaks ning eemaldades see muutuja esialgsetest reeglitest. Analoogiliselt kahe järgneva puhul, kus genereerivaks väljaks on vastavalt x_2 ja x_3 oleme saanud vajalike reeglite hulga.

Näide järeldatud reeglist, mis ei kuulu vajalike reeglite hulka on:

$$-x_1 + 3x_2 - 5x_3 + x_4 \geq 0,$$

mis saadakse valemi 4.6 korrutamisel kahega ning liites tulemus valemile 4.5. Tegu ei ole reegliga, mis peaks kuuluma vajalike reeglite hulka, sest ükski muutuja ei taandu välja.

Olgu x_j=(3, 4, 6, 1) ning kõik usaldusväärtust näitavad kaalud võrdsed ühega. Näeme, et vaatlustulemus eksib esialgsetest reeglitest ainult valemi 4.6 vastu, kuid me ei saa teha järeldust, millise väärtuse muutmine vastaks Fellegi-Holt´i paradigma poolt defineeritud eeskirjale. Samas näeme, et vajalike reeglite hulgast eksib vaatlus kolme reegli vastu ning tunnus x_3 sisaldub kõigis kolmes reeglis, mille muutmine ongi ainus optimaalne lahendus vea tuvastamise probleemile.

4.4.2. *Harude ja tõkete* algoritm

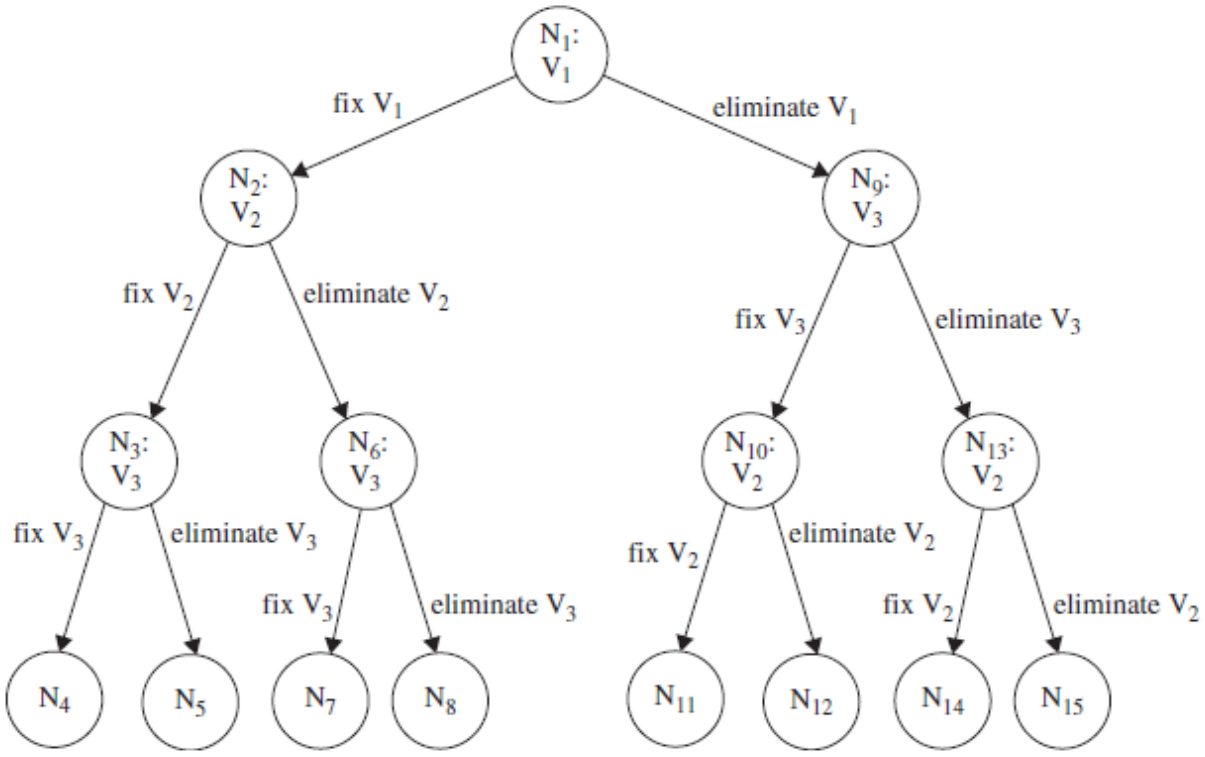
De Waal ja Quere pakkusid Fellegi-Holti´i paradigma lahendamiseks 2003. aastal välja *harude ja tõkete* algoritmi (ingl. *Branch-and-Bound*).

Käesolev ja alapeatükis 4.5.1 tutvustamisele tulevad meetodid põhinevad igale vaatlusele vastava kahendpuu moodustamisel, millest on toodud näide joonisel 4.1, kus ringi sees olev number tähistab tipu numeratsiooni ning *V* (ingl. sõnast *variable*) käsitletavat tunnust. Enne meetodi avamist läbime lühidalt vajaliku terminoloogia.

Kahendpuu on rekursiivselt defineeritud struktuur, mis sisaldab lõplikku hulka tippe. Tipu alluvateks (ingl. *child*) nimetatakse temast lähtuva alampuu harude juuri. Tipp *s* on tipu *t* ülemus (ingl. *parent*), kui tipp t on tipu *s* alluv. Igal kahendpuul on eristatav üks tipp – puu juur, millel puudub ülemus, kõigil teistel tippudel on täpselt üks ülemus. Kahendpuu igal tipul saab olla maksimaalselt kaks alluvat. Tippu, millel puuduvad alluvad nimetatakse leheks.

Tipp N_1 on joonisel 4.1 toodud puu juur, tema alluvad on tipud N_2 ning N_9. Kõigist joonisel olevatest tippudest on lehed N_4, N_5, N_7, N_8, N_{11}, N_{12}, N_{14} ja N_{15}.

Tõkete ja harude algoritmi kirjeldatakse kui kahendpuu läbimist, kus puu juur vastab algsele probleemile (reeglite hulgale) ning iga järgmine tipp selle probleemi alamprobleemile (Clausen 1999).



Joonis 4.1. Näide binaarpuust *harude ja tõkete* algoritmi rakendamisel

Antud algoritmi kohaselt koostatakse kahendpuu määrates üks vabalt valitud tunnus puu juureks. Järgnevalt moodustatakse tipule kaks alampuud, millest ühe korral eeldatakse, et vaadeldud väärtus on korrektne ning teisel mitte. Näiteks joonise 4.1 põhjal on tunnus V_1 valitud tippu N_1. Moodustades vasakpoolset alampuud eeldame, et vaadeldud väärtus on korrektne ning fikseerimise selle. Parempoolse alampuu moodustamisel eeldame, et tegu oli vigase väärtusega ning eemaldame tunnuse. Sellise konstruktsiooni abil saame läbi uurida kõik võimalused vigade leidmiseks, mille abil leiame parima lahenduse vea tuvastamise probleemile. Tunnust, mis on tipu poolt kas fikseeritud vaadeldud suurusega või kontrollreeglitest eemaldatud, antud alampuus edasi ei käsitleta. Iga tipp vastab mingile kontrollreeglite hulgale, mida ei ole veel käsitletud. Puu juurele vastab esialgne kontrollreeglite hulk.

Paneme tähele, et tunnuse eemaldamisega võivad kaasneda lisareeglid. Eeldame, et $x_1 \leq x_2$, ja $x_2 \leq x_3$, . Eemaldades x_2 antud reeglite hulgast peame lisama reegli $x_1 \leq x_3$, mis on järelduv kahest eelnevast. Tagamaks vastavust esialgsetele reeglitele rakendame tunnustele Fourier-

Motzkini eemaldust kombineerides võrratusi paarikaupa. Kui eemaldatav tunnus sisaldub bilansireeglis, avaldame antud tunnuse ning kasutame võrdust järgnevates kontrollreeglites.

Kontrollreeglite uuendamine igas harus on antud algoritmi kõige tähtsam samm, mis sõltub asjaolust, kas harus eemaldatakse või fikseeritakse väärtus. Tunnuse väärtuse fikseerimine esialgse vaatlustulemusena tähendab igas kontrollreeglis tunnuse väärtuse fikseerimist vaadeldud väärtusega (nii vastuolus, kui kehtivate kontrollreeglite puhul). Tekkinud kontrollreeglite hulka rakendatakse tekkivale alampuule iteratiivselt. Tulemusena võib mõni kontrollreegel olla tautoloogia ehk paratamatu tõde. Näiteks olgu kontrollreegel x_7>0 ning vaadeldud väärtus võrdne ühega. Saame, et 1>0, kui fikseerime x_7 esialgse väärtusega. Taolisi kontrollreegleid (,mis sisaldavad paratamatut tõde) ei ole vajalik kontrollreeglite hulgaga edasi kanda. Kui tulemuseks on vastuolu, siis antud alampuust ei ole võimalik leida lahendust vea tuvastamise probleemile.

Järgnevalt, kui kõiki tunnuste väärtusi on käsitletud oleme jõudnud lehtedeni. Järele jäänud kontrollreeglite hulk ei sisalda vastuolusid parajasti siis, kui ta rahuldab kõiki esialgseid reegleid. Viimase sammuna kontrollime iga lehte, mis ei sisalda vastuolu ning valime neist vea tuvastamise probleemi lahenduseks tunnuste hulga, mille usaldusväärsust märkivate kaalude summa on vähim.

Näide 4.2

Olgu vaatlusele defineeritud kontrollreeglid:

$$x_1 + x_2 = x_3$$
$$x_2 \geq 3$$
$$x_1 \geq 0$$

ning vaatluste vektor x_j =(1, 2, 5). Olgu kõik vastavad usaldusväärsust näitavad kaalud võrdsed ühega.

Alustame binaarpuu moodustamisega valides puu juureks tunnuse x_1. Esmalt vaatleme puu haru, kus eeldame, et vaatlustulemus on korrektne ning fikseerime selle. Saame järeldatud reeglid peale tekkinud tautoloogia eemaldamist:

$$1 + x_2 = x_3$$
$$x_2 \geq 3$$

Eeldades ka tunnuse x_2 õiget väärtust jõuame vastuoluni $2 \geq 3$ ning antud haru edasi ei käsitle. Samas eemaldades tunnus x_2 jääb alles võrrand $x_3 \geq 4$, mis sobib ka vaadeldud x_3 väärtusega. Oleme jõudnud binaarpuu leheni, kus usaldusväärsust näitavate kaalude summa $\sum w = 1$.

Järgmise sammuna uurime puu juure teist alampuud eemaldades tunnus x_1 ning jõuame süsteemini:

$$x_2 \geq 3$$
$$x_3 \geq x_2,$$

mis tekitab vaatlustulemuste põhjal vähemalt ühe tunnuse muutmise vajaduse juurde ehk $\sum w \geq 1$. Järelikult ei ole vajadust antud alampuud edasi uurida ning oleme lahendanud vea tuvastamise probleemi – muutmist vajav tunnus on x_2.

4.4.3. *Lõigatud tasandite* põhimõte

Garfinkel, Kunnathur ja Liepins pakkusid 1986. aastal Fellegi-Holt´i paradigma lahendamiseks välja *lõigatud tasandite* algoritmi. Antud meetodi korral lahendatakse vea tuvastamise probleem teisendades ta „hulga katmise“ probleemiks (ingl. *set-covering problem*).[1] Käesolevas peatükis kirjeldame antud meetodi täiendatud versiooni, mis töötati välja Ragsdale ja McKwowni poolt 1996. aastal. Hilisem meetod erineb esialgsest, kuna võtab arvesse ka väärtuse suuna muutumist algse ja imputeeritava väärtuse vahel vältimaks vastuolude tekkimist.

[1] Hulga katmise probleem on defineeritud, kui optimiseerimise ülesanne: milline on vähim alamhulkade hulk, mille ühend on lahendiks valitud hulk

Defineerime tunnuse y_j:

$$y_j = \begin{cases} 1\text{ , KUI } \check{x}_j \neq x_j \text{ või } x_j \text{ on puuduv väärtus} \\ 0\text{ , muul juhul} \end{cases}$$

Ragsdale´i ja McKeowni poolt välja pakutud lahenduse kohaselt asendame igas reeglis tunnused x_j $(j = 1, \dots, p)$ muutujatega $x_j^0 + x_j^+ + x_j^-$, kus x_j^0 tähistab vaadeldud väärtust, $x_j^+ \geq 0$ positiivset väärtuse muutust ning $x_j^- \geq 0$ negatiivset väärtuse muutust.

Lisaks defineerime muutujad y_j^+ ja y_j^- järgnevalt:

$$y_j^+ = \begin{cases} 1\text{ , KUI } x_j^+ > 0 \\ 0\text{ , KUI } x_j^+ = 0 \end{cases} \text{ ning}$$

$$y_j^- = \begin{cases} 1\text{ , KUI } x_j^- > 0 \\ 0\text{ , KUI } x_j^- = 0 \end{cases}.$$

Meetodi kohaselt leitakse lahendus vea tuvastamise probleemile minimiseerides sihifunktsioon:

$$\sum_{j=1}^{p} w_j (y_j^+ + y_j^-)$$

tingimustel, mis on tuletatavad kontrollreeglitest peale tunnuste x_j asendamist suurustega $x_j^0 + x_j^+ - x_j^-$. Neid tingimusi nimetataksegi nö. lõigatud tasanditeks, millest tuleneb meetodi nimi.

Eeskirja kirjeldab järgnev näide.

Näide 4.3

Olgu defineeritud kontrollreeglid:

$$x_1 + x_2 \leq 3$$
$$x_2 - x_3 \leq -2$$

ning vaatluste vektor $x_j = (1, 1, 2)$ vastavate usaldusväärsust näitavate kaaludega $w_j = (1,2,1)$.

Asendades kontrollreeglites tunnused x_j suurustega $x_j^0 + x_j^+ + x_j^-$ saame süsteemi:

$$-x_1^+ + x_1^- - x_2^+ + x_2^- \geq 1 \qquad (4.7)$$

$$-x_2^+ + x_2^- + x_3^+ - x_3^- \geq 1 \qquad (4.8)$$

ning sihifunktsiooni:

$$y_1^+ + y_1^- + 2y_2^+ + 2y_2^- + y_3^+ + y_3^-$$

tingimusel, et kehtib:

$$y_1^- + y_2^- \geq 1$$

$$y_2^- + y_3^+ \geq 1,$$

mis on vastavalt järeldatavad valemitest 4.7 ja 4.8. Antud sihifunktsiooni minimiseerimine annabki lahenduse vea tuvastamise probleemile. Antud juhul on ainsaks optimaalseks lahendiks tunnuse x_3 väärtuse suurendamine.

4.4.4. Tippude moodustamise lähenemine

Käesolev peatükk kirjeldab tihti rakendust leidvat lähenemist lahendamaks vea tuvastamise probleem läbi nö. hulktahuka tippude moodustamise. Antud meetodit rakendatakse erinevates statistikaprogrammides, näiteks GEIS ja SAS.

Kui tunnustele $(x_1, \dots, x_p)$ vastav esialgne vaatlustulemus $(x_1^0, \dots, x_p^0)$ on vastuolus kontrollreeglitega kujul 3.1 ja 3.2, siis üritatakse meetodi kohaselt leida vaatlustulemuse positiivne $x_j^+ > 0$ või negatiivne muutus $x_j^- > 0$ selliselt, et kõik kontrollreeglid oleksid rahuldatud. Selleks kasutatakse sihifunktsiooni 4.1 tingimusel, et uus tehisvaatlus

$$(x_1^0 + x_1^+ - x_1^-, \dots, x_p^0 + x_p^+ + x_p^-)$$

vastab kontrollreeglitele:

$$a_{k1}(x_1^0 + x_1^+ - x_1^-) + \cdots + a_{kp}(x_p^0 + x_p^+ + x_p^-) + b_k \geq 0 \qquad (4.8)$$

ja

$$a_{k1}(x_1^0 + x_1^+ - x_1^-) + \cdots + a_{kp}(x_p^0 + x_p^+ + x_p^-) + b_k = 0 \qquad (4.9)$$

iga kontrollreegli k korral vastavalt juhule 3.1 või 3.2.

Valemite 4.8 ja 4.9 abil saame reeglite hulga, mis moodustab nö. hulktahuka tundmatute x_j^+ ja x_j^- (j=1,…,p) jaoks, mis lahendatakse tippude moodustamise (ingl. *vertex generation*) lähenemise abil. Leides kõik võimalikud lahendid valime neist minimaalse sihifunktsiooni väärtusega tunnuste hulga. Tegu on levinud matemaatilise probleemiga, mille lahendamiseks on mitmeid meetodeid, mida käesolevas töös täpsemalt ei kirjeldata.

Näide 4.4.

Olgu defineeritud kontrollreeglid:

$$x_1 + x_2 - 2x_3 \geq 2$$
$$-x_2 - x_3 \geq 1$$

ning vaatlustulemus $x_j = (3, 2, 1)$ ning vastavad usaldusväärsust näitavad kaalud kõik võrdsed ühega.

Saame tingimused tundmatutele:

$$x_1^+ + x_2^+ - 2x_3^+ - x_1^- - x_2^- + 2x_3^- \geq -1$$

$$-x_2^+ - x_3^+ + x_2^- + x_3^- \geq 4$$

$x_j^+ \geq 0$ iga j $\in (1,2,3)$ ja

$x_j^- \geq 0$ iga j $\in (1,2,3)$,

mille alusel minimiseeritakse sihifunktsioon 4.1.

4.5. Algoritm segaandmetele

Selles peatükis uurime vea tuvastamise probleemi andmete korral, kus nii faktortunnused, kui arvulised andmed on kasutusel – nö. segaandmed (ingl. *mixed data*). Käesolevaga antakse ülevaade matemaatilisest probleemist ja selle lahendamisest juba tutvustatud *harude ja tõkete* algoritmi põhjal.

Tähistame faktortunnused v_j (j = 1, … , m) ja arvulised x_j (j = 1,… , p). Olgu reeglite arv K. Tähistame faktortunnuse v_j võimalike väärtuste hulka D_j-ga. Käesolevas alapeatükis (De Waal, Pannekoek & Scholtus 2011) käsitleme kontrollreegleid järgneval kujul:

KUI $v_j \in F$ iga j = 1, … m ,

SIIS $(x_1, \dots, x_p) \in \{x | a_{k1}x_1 + \dots + a_{k1}x_1 + b_k \geq 0\}$ (4.10)

või

KUI $v_j \in F$ iga j = 1, … m ,

SIIS $(x_1, \dots, x_p) \in \{x | a_{k1}x_1 + \dots + a_{k1}x_1 + b_k = 0\}$ (4.11)

,kus $F_j^k \subset D_j$.

Et vaatlus oleks järjepidev peab ta vastama kõigile kontrollreeglitele E_k $(k = 1, \dots, K)$, mis on toodud kujul 4.10 ja 4.11.

Fellegi-Holt´i paradigma kohaselt peame leidma igale vaatlusele vastava uue nö. tehisvaatluse ($\check{v}_1^0$,…, $\check{v}_m^0$, $\check{x}_1^0$,…, $\check{x}_m^0$) nii, et valemid 4.10 ja 4.11 kehtiksid ja vastaksid kontrollreeglitele ning

$$\sum_{j=1}^{m} w_j^c \delta\left(v_j^0, \check{v}_j\right) + \sum_{j=1}^{m} w_j^r \delta\left(x_j^0, \check{x}_j\right) \quad (4.12)$$

oleks minimaalne, kus w_j^c tähistab mittenegatiivset faktortunnuse usaldusväärsust näitavat kaalu ja w_j^r sama suurust pidevate tunnuste korral, kui x_j või on puuduv väärtus siis $\delta\left(x_j^0, \check{x}_j\right)$=1 ning $\delta\left(x_j^0, \check{x}_j\right)$=0, kui vaadeldud ja imputeeritav väärtus on võrdsed. Funktsioon δ on analoogiliselt defineeritud faktortunnuste korral.

4.5.1. *Harude ja tõkete* algoritm segaandmetele

Harude ja tõkete algoritmi rakendamine faktortunnustele ning segaandmetele on analoogiline võrreldes pidevate tunnustega. Peale tunnuse valimist moodustatakse sarnaselt alapeatükis 4.4.2. kirjeldatuga binaarpuu. Antud juhul reeglite komplekti uuendamise algoritmi kõige tähtsam samm (kas tunnus fikseeritakse või eemaldatakse) sõltub nüüd lisaks ka tunnuse tüübist ehk kas tegu on faktor- või arvulise tunnusega.

Tunnuse fikseerimine sõltumata iseloomust ei ole keeruline, lihtsalt asendatakse vaadeldud väärtus kõikidesse kontrollreeglitesse, kuid eemaldamine on suhteliselt keerukas protsess. See kätkeb uute reeglite genereerimist, mis antud tunnust enam ei sisalda. Genereerimise protsessis peame silmas pidama nii vastuollu minevaid, kui vastuolu mitte tekitavaid reegleid. Iga tekkiv reeglite hulk vastab ühele puu tipule. Pideva tunnuse eemaldamiseks rakendame Fourier-Motzkini eemaldust. Pideva tunnuse x_r eemaldamiseks käesolevast reeglite hulgast alustame kõigi reeglite kopeerimisega, mis mainitud tunnust ei sisalda uude (järgmisele tipule vastavasse) reeglite hulka. Järgnevalt uurime kõiki reegleid kujul 4.10 ja 4.11, mis sisaldavad tunnust x_r paari kaupa. Näiteks reeglite E_s ja E_t korral uurime ühisosa $F_j^s \cap F_j^t$. Kui vähemalt üks nendest hulkadest on tühi iga *j=1,…m* korral, siis käesolevat paari edasi ei uurita.

Järgmise sammuna moodustatakse tuletatud reegel. Kui reegli E_s KUI-SIIS lause järeldatav (*SIIS* lause osa) on võrdus, avaldame ta:

$$x_r = -\frac{1}{a_{sr}}\left(b_s + \sum_{j \neq r} a_{sj} x_j\right)$$

ning eemaldame reeglist E_t. Analoogiliselt kasutame võrdust, mis sisaldub reeglis E_t ehk kui reegli E_s SIIS tingimus on võrratus ja reegli E_t SIIS tingimus on võrdus. Kui mõlemad SIIS lause tingimused on võrratused kontrollime, kas x_r-i ees olev kordajad on vastupidise märgiga ehk kas $a_{sr} \times a_{tr} < 0$, tingimuse mitte kehtides antud paar edasi ei uurita. Kui kordajad omavad vastupidiseid märke saame uue tuletatud tingimuse:

$$\left(x_1, \ldots, x_p\right) \in \left\{x \middle| \tilde{a}_1 x_1 + \ldots + \tilde{a}_{r-1} x_{r-1} + \tilde{a}_{r+1} x_{r+1} + \ldots + \tilde{a}_p x_p + \tilde{b} \geq 0\right\},$$

kus

$$\tilde{a}_j = |a_{sr}| \times \mathrm{a_{tj}} + |a_{tr}| \times \mathrm{a_{sj}}$$

ja

$$\tilde{b} = |a_{sr}| \times \mathrm{b_t} + |a_{tr}| \times \mathrm{b_s},$$

kus × tähistab hulkade otsekorrutist. Oleme jõudnud tulemuseni, kus tunnus x_r ei sisaldu SIIS lauses.

Algoritmi lihtsuse huvides käsitletakse binaarpuus esmalt arvulisi tunnuseid ja alles siis faktortunnuseid. Peale arvuliste tunnuste läbimist on kõik käsitletavale tipule vastavad reeglid esitatavad kujul:

KUI $v_j \in F_j^k$ iga $j = 1, \dots, m$.

SIIS $(x_1, \dots, x_p) \in \emptyset$.

Faktortunnuse v_j eemaldamiseks kopeerime kõik antud tunnust mitte sisaldavad reeglid uude (järgmisele tipule vastavasse) reeglite hulka.

Järgmise sammuna rakendame Fellegi ja Holti poolt välja pakutud algoritmi KUI lausele, et genereerida tuletavad reeglid. Alustame indeksite hulga S kindlaks tegemisega, mille korral kehtib:

$$\bigcup_{k \in S} F_r^k = D_r. \qquad (4.14)$$

ja

$$\bigcap_{k \in S} F_j^k \neq \emptyset \text{ iga } j = 1, \dots, r-1, r+1, \dots, m \text{ korral.} \qquad (4.15)$$

Nendest indeksite hulkadest valitakse vähimad, mis ei lähe vastuollu valemitega 4.14 ja 4.15, kuid antud hulga alamhulk ei vasta valemile 4.14, mille abil moodustame järeldatud reegli:

KUI $v_r \in D_r, v_j \in \bigcap_{k \in S} F_j^k$ iga *j* = 1, …, *r*-1, *r*+1, …, *m* korral.

SIIS $(x_1, \dots, x_p) \in \emptyset$.

Vajab märkimist, et kirjeldatud meetodil tunnuse eemaldamise tulemusena saadakse uus tunnuste hulk, mis sisaldab vaid järele jäänud tunnuseid. Peale kõigi faktortunnuste käsitlemist saame reeglite hulga, mis ei sisalda tundmatuid (kõik tunnused on kas fikseeritud või eemaldatud).

Vastavalt Fellegi-Holt´i põhimõttele kontrollime iga puu lehte, et teha kindlaks, kas väärtuste imputeerimine on võimalik ilma ühtegi vastuolu tekitamata. Kui selliseid lehtesi on rohkem kui üks, siis valime hulga, mille muudetavatele tunnustele vastavate usaldusväärtust näitavate kaalude summa on vähim ehk minimiseeritakse sihifunktsioon 4.12.

5. Deduktiivne imputeerimine

Deduktiivne imputeerimine (ing. *deductive imputation)* on osa imputeerimisest tingimuste alusel (ing. *imputation under constraints*). Antud juhul ei sõltu imputeeritavad väärtused ei jaotustest, millele tunnus allub, ega kasutata imputeerimiseks teisi vaatlusi. Käesoleva meetodi korral sõltuvad imputeeritud väärtused ainult peatükis 3 kirjeldatud kontrollreeglitest ning on neist tuletatavad. Esimesena vaatleme imputeerimist arvuliste tunnuste korral, täpsemalt bilansireeglite alusel ning mitte-negatiivsus piiranguid kasutades ning teisena faktortunnuste põhjal.

5.1. Bilansireeglite alusel

Bilansireeglid sätestavad andmestikule summana kirjeldatava seose (valem 3.1). Käesolevas peatükis kasutame tähistamiseks maatrikstähistust järgneval kujul (Pannekoek 2006). Olgu andmestikus *p* arvulist tunnust $x_1, \dots, x_p$, millele on sätestatud *r* kontrollreeglit. Tähistame vaatluste vektori $x = \left(x_1, \dots, x_p\right)'$ ning defineeritud kontrollreeglid esitame:

$$Rx = b, \tag{5.1}$$

kus, *R* on maatriks, mille abil esitatakse bilansireeglid. Maatriksi *R* veerud, mis ei sisalda nullist erinevaid elemente vastavad tunnustele, mis ei sisaldu üheski bilansireeglis. Järjestades tunnuseid ümber saame jagada vektori x vaadeldud ja puuduvateks tunnuste väärtuseks: $x = (x'_o, x'_{mis})'$. Järjestades sarnaselt ümber ning tähistades maatriksi R vastavad veerud saame kirjutada:

$$[R_0 \quad R_{mis}] \begin{bmatrix} x_o \\ x_{mis} \end{bmatrix} = b. \tag{5.2}$$

ning järelikult:

$$R_{mis}x_{mis} = b - R_o x_o$$

Tähistame $a = b - R_o x_o$

Süsteemi 5.1 lahend on järgnev (Memobust: Deductive Imputation 2014).

$$\check{x}_m = R_{mis}^{-}a + (R_{mis}^{-}R_{mis} - \mathrm{I})\mathrm{z}, \qquad (5.3)$$

kus R_{mis}^{-} märgib maatriksi *R* üldistatud pöördmaatriksit ja *z* on suvaline vektor. Mittelahenduva süsteemi korral ei eksisteeri reeglitele alluvaid sobivaid väärtusi.

Saadud lahendit rakendame imputeerimisel.

Näide 5.1.

Olgu defineeritud kontrollreeglid:

$$x_1 + x_3 = x_4$$
$$x_2 = x_5$$
$$x_2 + x_3 = x_6.$$

Kirjutame välja reeglite maatriksi:

$$R = \begin{pmatrix} 1 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 \end{pmatrix}.$$

Olgu vaadeldud väärtuste vektor $x_o = (x_1, x_4, x_5, x_6)' = (5,9,3,7)'$ ning puuduvate väärtuste vektor $x_m = (x_2, x_3)'$, mille põhjal saame välja kirjutada maatriksi *R* valemis 5.2 defineeritud alammaatriksid:

$$R_o = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \text{ ja}$$
$$R_{mis} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Lihtsa aritmeetika abil saame

$$a = b - R_o x_o = -R_o x_o = \begin{pmatrix} 4 \\ 3 \\ 7 \end{pmatrix}$$

ning kuna $(R_{mis}^{-}R_{mis} - \mathrm{I})$ on nullmaatriks, leiame lahenduse arvutades $R_{mis}^{-}a$:

$\check{x}_2 = 3$ ja $\check{x}_3 = 4$.

5.2. Mitte-negatiivsus piirangute kasutamine

Käesolev arutelu on sarnane eelmise peatükiga, kuid lisame mõnedele numbrilistele väärtustele piirangud nende mittenegatiivsuse kohta. Ka lahendus leitakse analoogiliselt (De Waal, Pannekoek & Scholtus 2011). Esmalt leiame süsteemi 5.1 lahendi. Olgu eelmises alapeatükis tutvustatud vektori *a* üks väärtus null ehk $a'_{mis,l}x_m = 0$, kus $a'_{m,l} = (a_{mis,l1}, \dots, a_{mis,lm})$ on maatriksi R_{mis} *l*-is rida.

Kui

i) kõik vektori $a_{mis,l}$ mittenullilised elemendid on sama märgiga
ii) iga väärtus $a_{mis,lj} \neq 0$ vastab tunnusele $x_{mis,j}$, millele on seatud piirang tema mittenegatiivsuse kohta

saame imputeerida $\check{x}_{mis,j} = 0$ iga *j* puhul, kus $a_{mis,lj} \neq 0$.

Näide 5.2

Olgu andmestikus 7 tunnust, mis kõik on mittenegatiivsete väärtustega ja defineeritud on lisaks järgmised kontrollreeglid:

$$x_1 + x_2 + x_3 + x_4 = x_5$$
$$x_4 + x_5 = x_6$$
$$x_1 + x_2 + x_3 + x_5 = x_7.$$

Reeglite maatriks avaldub (valem 5.1):

$$Rx_i = \begin{pmatrix} 1 & 1 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & -1 \end{pmatrix}$$

Olgu vaadeldud väärtuste vektor $x_o = (x_1, x_5, x_6, x_7)' = (2,6,10,8)'$ ning puuduvate väärtuste vektor $x_m = (x_2, x_3, x_4)'$, mille põhjal saame välja kirjutada maatriksi *R* valemis 5.2 defineeritud alammaatriksid:

$$R_o = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 1 & 0 & -1 \end{pmatrix}$$ ja

$$R_{mis} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

Milles tulenevalt saame:

$$a = -R_o x_o = \begin{pmatrix} 4 \\ 4 \\ 0 \end{pmatrix}.$$

Ning kirjutame välja seose:

$$a = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ 0 \end{pmatrix}. \qquad (5.4)$$

Arvutades analoogiliselt eelneva näite põhjal valemis 5.3 näidatu saame imputeerida $\check{x}_4 = 4$:

$$\check{x}_m = R_{mis}^{-} a + (R_{mis}^{-} R_{mis} - \mathrm{I})\mathrm{z} = \begin{pmatrix} 0 \\ 0 \\ 4 \end{pmatrix} + \begin{pmatrix} -0{,}5 & 0{,}5 & 0 \\ 0{,}5 & -0{,}5 & 0 \\ 0 & 0 & 0 \end{pmatrix} z,$$

kuid $\check{x}_2$ ja $\check{x}_3$ jäävad sõltuma suvalisest vektorist *z*.

Paneme tähele, et vektori *a* kolmas element on null ning mittenullilised elemendid on mõlemad positiivsed ja on tehtud eeldus tunnuste mittenegatiivsuse kohta. Tingimuste i ja ii põhjal saame imputeerida: $\check{x}_2 = 0$ ja $\check{x}_3 = 0$.

5.3. Kasutades faktortunnuseid

Käesolevas peatükis kirjeldame deduktiivset imputeerimist faktortunnuste korral (De Waal, Pannekoek & Scholtus 2011), kus rakendatakse meetodit faktortunnuste eemaldamiseks reeglitest, mida esmalt kirjeldame.

Igale faktortunnusele v_j vastab tema määratud võimalike väärtuste hulk D_j. Kõik reeglid esialgsest reeglite hulgast Ω on kirja pandavad järgnevalt:

$$F_1^k \times \ldots \times F_m^k,$$

kus $F_j^k \subseteq D_j$. See tähendab, et vaatlus rahuldab reeglit k siis ja ainult siis, kui $v_j \in F_j^k$ iga $j = 1, \ldots, m$. Kui $F_j^k = D_j$ iga $j = 1, \ldots, m$, siis reegel *k* ei sea piiranguid tunnusele v_j.

Deduktiivseks imputeerimiseks rakendame alapeatükis 4.5.1. tutvustatud meetodit tunnuse v_g eemaldamiseks esialgsest kontrollreeglite hulgast Ω arvestades kõiki vähimaid indeksite hulkasid S, mille korral:

$$\bigcup_{k \in S} F_g^k = D_g$$

ja

$$\bigcap_{k \in S} F_g^k \text{ iga } j = 1, \ldots, g-1, g+1, \ldots, m.$$

Millest saame tuletada reegli, mis ei sisalda tunnust v_g:

$$\bigcap_{k \in S} F_1^k \times \ldots \times \bigcap_{k \in S} F_{g-1}^k \times D_g \times \bigcap_{k \in S} F_{g+1}^k \times \ldots \times \bigcap_{k \in S} F_m^k. \qquad (5.5)$$

Asendades esialgse reeglite hulga Ω uue reeglite hulgaga Ω_1, mis sisaldab kõiki reegleid esialgsest reeglite hulgast Ω, mis ei sisalda tunnust v_g ning kõiki tuletatud reegleid valemist 5.5 saame reeglite hulga, mis ei sisalda tunnust v_g ning rahuldab kõiki tingimusi parajasti siis, kui seda teeb esialgne reeglite hulk Ω.

Kirjeldatud meetodit rakendatakse deduktiivseks imputeerimiseks faktortunnuste korral järgneva algoritmi kolmandas sammus.

Algoritmi kirjeldus

Olgu esialgne kontrollreeglite hulk Ω ning faktortunnuste vaatlusvektor $(v_1 .. v_m)$. Igale tunnusele v_j vastab tema võimalike väärtuste hulk D_j. Eeldame, et kõik vektori puuduvad väärtused on võimalik asendada nii, et nad vastaks kontrollreeglitele.

1. Olgu *M* indeksite hulk, mis vastab tunnuste puuduvatele väärtustele. Defineerime $T := \emptyset$. Olgu Ω reeglite hulk, mille kõik väärtused on teada ning Ω_0 reeglite hulk, mis sisaldab puuduvaid väärtusi.

2. Kui $M\backslash T \neq \emptyset$ valime $g \in M\backslash T$, vastasel juhul lõpetame algoritmi läbimise, sest kõiki puuduvaid väärtusi on käsitletud.
3. Eemaldame kõik tunnused, mis sisalduvad hulgas $M\backslash\{g\}$ hulgast Ω_0 alapeatüki alguses kirjeldatud meetodi põhjal. Tähistame tunnusele v_g vastava järele jäänud reeglite hulga Ω^*-ga.
4. Kui reeglite hulgale Ω^* vastab mitu väärtust hulgas D_g, siis defineerime uuesti hulga $T \coloneqq T \cup \{g\}$ ning naaseme algoritmi teise sammu juurde.
5. Kui reeglite hulgale Ω^* vastab täpselt üks väärtus hulgast D_g, siis imputeerime selle väärtuse tunnuse v_g väärtuseks. Järgnevalt uuendame hulka Ω_0 lisades sinna imputeeritud väärtus ning defineerime $M \coloneqq M\backslash\{g\}$ ning naaseme algoritmi teise sammu juurde.

6. Andmestiku *Väliskülastajad Eestis* automaatne parandamine

Käesolevas peatükis rakendame Fellegi-Holti paradigmat Eesti Statistikaameti andmestikule *Väliskülastajad Eestis* eesmärgiga uurida, kuidas mõjutab paradigma efektiivsust väiksem ning vähem kattuvaid tunnuseid omav kontrollreeglite hulk. Küsitluse ankeet, mille põhjal koostati andmestik on Lisas 1. Kontrollreeglite hulk on antud andmestiku puhul suhteliselt suur – kokku peavad kõik vaatlused vastama ligi 50-le reeglile, millest tulenevalt võib tunduda loogiline töös kirjeldatud meetodite rakendamine.

Varasemate näidetega on demonstreeritud kirjeldatud meetodite efektiivsust kontrollreeglite süsteemide puhul, kus reeglites on tunnuste ühisosa. Uurime, kuidas sobib Fellegi-Holt´i paradigma rakendamine lihtsama struktuuriga Lisas 2 toodud reeglite struktuuride näitel.

Käesolevas andmestikus saame jagada kontrollreeglite süsteemid blokkideks, mis ei oma tunnuste ühisosa. Iga sellist süsteemi (Lisa 2) käsitleme iga vaatluse korral eraldiseisva vea tuvastamise probleemina. Täpsemalt on neid kahte tüüpi. Esimene ning neljas reeglite hulk sisaldavad bilansireegleid ning ülejäänud on KUI-SIIS tüüpi kontrollreeglite süsteemid. Kõik tunnuste nimed algavad sõnaga *ALG*, mis tähendab, et tegemist on vaadeldud väärtusega, tunnuse teine osa *VALISK* (e. väliskülastaja) viitab andmestiku nimele ning lõpus on ära toodud number (ja täpsustus), millisest küsimusest (või küsimuse osast) vastus pärineb.

Esimene ohumärk meetodi rakendamiseks on tunnuste väike ühisosa reeglite hulkades, teiseks ei aita Fellegi-Holti paradigma efektiivsusele kaasa tõsiasi, et tunnuste usaldusväärsust näitavad kaalud on võrdsed ühega ehk meil ei ole põhjust arvata, et ühegi tunnuse väärtus on usaldusväärsem kui teistel. Sellest tulenevalt sõltub muudetava väärtuse valimine Fellegi-Holt´i paradigma põhjal suuresti juhusest. Järgnevas kahes alapeatükis käsitleme esmalt bilansireegleid ning teisena KUI-SIIS tüüpi kontrollreegleid.

Kontrollreeglid defineeritakse statistikapaketi R paketi *editrules* abil (Pakett „editrules“ 2015) ning vea tuvastamise probleem lahendatakse käsu *localizeErrors* abil (Lisa 3).

6.1. Bilansireegleid sisaldavad struktuurid

Esimene kontrollreeglite struktuur on defineeritud järgnevalt. Kolmandas küsimuses öeldud ööde summa peab kokku langema 15. küsimuses antud vastusega ning ükski tunnus ei saa olla negatiivne:

$$alg\_VALISK15\_1ARV \geq 0$$
$$alg\_VALISK15\_2ARV \geq 0$$
$$alg\_VALISK15\_3ARV \geq 0$$
$$alg\_VALISK15\_4ARV \geq 0$$
$$alg\_VALISK15\_5ARV \geq 0$$
$$alg\_VALISK03\_OOD \geq 0$$
$$alg\_VALISK15\_1ARV + alg\_VALISK15\_2ARV + alg\_VALISK15\_3ARV + alg\_VALISK15\_4ARV + alg\_VALISK15\_5ARV = alg\_VALISK03\_OOD.$$

Märgime ära, et antud andmestikus on kõigi mainitud tunnuste väärtused positiivsed, mis tähendab, et vigane vaatlus tekitab vastuolu täpselt ühe ehk viimase kontrollreegliga.

Olgu andmestikus vaatlus, kus tunnuste alg_VALISK15_2, alg_VALISK15_4 ja alg_VALISK03_OOD väärtused on vastavalt 2, 1 ja 4 ning ülejäänud käesolevas reeglite hulgas kasutusel olevad vaadeldud suurused on nullid. Antud vaatlus on reeglitega vastuolus.

Näeme, et iga tunnus, mis sisaldub antud kontrollreeglites sobib vea tuvastamise probleemi lahendamiseks ehk antud juhul on kuus lahendit, mis on ilmne paradigma algse sihifunktsiooni (valem 4.1) põhjal. Struktuuri lihtsusest tulenevalt oleme sunnitud parandamist vajava tunnuse juhuslikult valima. Järelikult ei ole Fellegi-Holt´i paradigma rakendamine sellise reeglite hulga puhul mõistlik ning ei ole ka põhjendatud jätkata deduktiivse imputeerimisega.

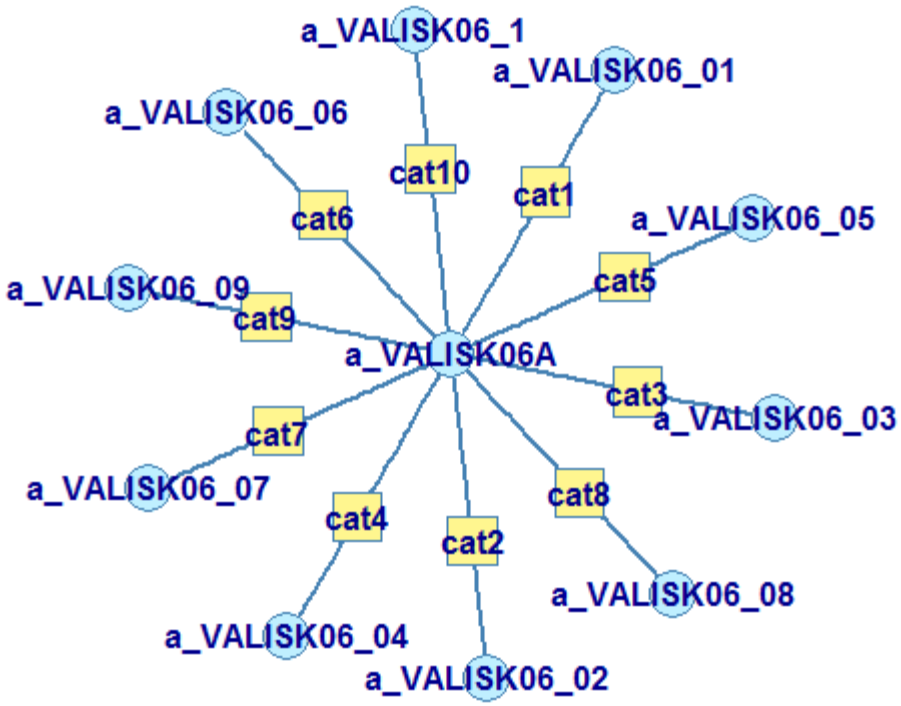
Analoogiline arutelu kehtib ka neljanda reeglite bloki (Lisa 2) korral.

6.2. KUI-SIIS lauseid sisaldavad struktuurid

Esimene KUI-SIIS tüüpi lauseid sisaldav reeglite struktuur (Lisa 2, teine blokk) sätestab asjaolu, et inimese märgitud reisi põhieesmärk (tunnus *alg_VALISK13A*) peab olema märgitud ka kõikide reisieesmärkide (tunnused *alg_VALISK13_1* kuni *alg_VALISK13_14*) seas:

$$KUI\ alg\_VALISK13A = 01\ SIIS\ alg\_VALISK13\_1 = 1$$
$$KUI\ alg\_VALISK13A = 02\ SIIS\ alg\_VALISK13\_2 = 1$$
…
$$KUI\ alg\_VALISK13A = 14\ SIIS\ alg\_VALISK13\_14 = 1$$

Struktuuri paremaks mõistmiseks vaatame seda kirjeldavat joonist 6.1, kus on joontega ühendatud tingimuste poolt seotud tunnused.



Joonis 6.1. Reeglite struktuuri visuaalne ülevaade.

Paneme tähele, et iga vaatlus saab rikkuda maksimaalselt ühte kontrollreeglit. Olgu andmestikus reeglitega vastuolus olev vaatlus, kus tunnuste *alg_VALISK13A* ja *alg_VALISK13_9*

väärtused on vastavalt 09 ja 0. Kuna tunnuste usaldusväärsust näitavad kaalud on võrdsed, siis vea tuvastamise probleemi lahendiks sobivad võrdväärselt mõlemad tunnused. Näeme, et lahendi valimine sõltub ainult juhusest ning järelikult ei ole andmete deduktiivne imputeerimine ka põhjendatud.

Analoogiline arutelu kehtib ka kolmanda, viienda, kuuenda, seitsmenda ja kaheksanda reeglite hulga (Lisa 2) korral.

Kasutatud kirjandus

Clausen, J 1999, *Branch and Bound Algorithms – Principles and Examples*, Department of Computer Science, University of Copenhagen, Copenhagen, Denmark.

De Waal, T & Coutinho, W 2005, *Automatic Editing for Business Surveys: An Assessment of Selected Algorithms*, Cambrian Printers, Wales, United Kingdom.

De Waal, T, Pannekoek, J & Scholtus, S 2011, *Handbook of Statistical Data Editing and Imputation*, John Wiley & Sons, New Jersey, USA.

Fellegi, P & Holt, D 1976, *„A Systematic Approach to Automatic Edit and Imputation“*. Journal of the American Statistical Association vol. 71.

Ghosh-Dastiar, B & Schafer, J 2003, Outlier Detection and Editing Procedures for Continuous Multivariate Data, Santa Monica, Princeton University.

Memobust Handbook: Automated Editing 2014. Kättesaadav: <http://www.cros-portal.eu/sites/default/files//Statistical%20Data%20Editing-04-M-Automatic%20Editing%20v1_4. pdf>

Memobust Handbook: Deductive Editing 2014. Kättesaadav: <http://www.cros-portal.eu/sites/default/files//Statistical%20Data%20Editing-04-M-Automatic%20Editing%20v1_4. pdf>

Memobust Handbook: Statistical Data Editing 2014. Kättesaadav: <http://www.cros-portal.eu/sites/default/files//Statistical%20Data%20Editing-04-M-Automatic%20Editing%20v1_4. pdf>

Pakett *„editrules“* (Statistikapaketi R dokumentatsioon) 2015, *Package „editrules“*. Kättesaadav: <http://cran.r-project.org/ web/packages/editrules/editrules.pdf> [kasutatud: 2. märts, 2015]

Pannekoek, J 2006, *Regression Imputation with Linear Equality Constraints On The Variables*, Conference of European Statisticians, Bonn, Germany.

Scholtus, S 2014, *A generalised Fellegi-Holt paradigm for automatic editing*, Conference of European Statisticians, Paris, France.

Lisa 1 – Küsitluse ankeet

K1	**Kas elate Eestis või olete siin viibinud üle aasta?** **1** – jah → **LÕPP** **2** – ei

K2	**Kas olete viibinud Eestis:** **1** alla 3 tunni → **K4** **2** ilma ööbimiseta reisil kestusega üle 3 tunni **3** ööbimisega reisil kestusega kuni 6 kuud **4** üle 6 kuu kuni 1 aasta
K3	**Kui kaua viibisite Eestis?** **1** ööbimisega reisi puhul ____ ööbimist **2** ühepäevakülastajana ____ tundi
K4	***Millises riigis Te alaliselt elate?*** **01** Soome **06** Prantsusmaa **11** Läti **02** Rootsi **07** Itaalia **12** Leedu **03** Taani **08** Venemaa **13** *MUU RIIK*…… **04** Saksamaa **09** Norra …… **05** Suurbritannia **10** USA
K5	**Vanuserühm:** **1** – (15–24) **2** – (25–34) **3** – (35–44) **4** – (45–54) **5** – (55–64) **6** – (65 ja vanem)
K5A	**Vastaja sugu:** **1** – mees **2** – naine
K6	**Mis oli Teie Eesti-reisi eesmärk?** *(MITME EESMÄRGI PUHUL MÄRKIGE KÕIK EESMÄRGID)* **01** puhkusereis **06** transiitreis **02** tööalane konverents või seminar **07** ravireis **03** töötamine Eestis (tasu Eestist) **08** sugulaste, tuttavate külastamine **04** muu tööalane reis (tasu välismaalt) **09** õppimine **05** ostureis **10** *MUU EESMÄRK* ……
K6A	**Mis oli Teie reisi <u>peaeesmärk</u>?** MÄRKIGE NR ..... **KUI KÜLASTAJA VIIBIS EESTIS ÜLE 6 KUU, SIIS SIIN KÜSITLUSE LÕPP**

K8	**Palun hinnake võimalikult täpselt Eestis <u>tehtud</u> kulutusi järgmiste liikide kaupa.**			
Kulutuse liik		**Kulutus**	Summa	*Rahaühik*
K8.1	Majutus	**Jah Ei**		
K8.2	Toitlustus (restoranid, baarid, kohvikud jm)	**Jah Ei**		EUR
K8.3	Transpordikulutused Eestis (bensiin, autorent jms)	**Jah Ei**		EUR
K8.4	**Meelelahutus ja vaba aeg (ekskursioon, kultuur, sport jm)**	**Jah Ei**		EUR
K8.5	Tervishoiuteenused (raviprotseduurid jm)	**Jah Ei**		EUR
K8.6	**Muud teenused (ilusalong, saun, sideteenused jm)**	**Jah Ei**		EUR
K8.7	Kaubad (toidu- ja tööstuskaubad, k.a alkohol kauplusest jm)	**Jah Ei**		EUR
K8A	KUI EI OSKA ÖELDA ERALDI KÕIGI LIIKIDE KOHTA **Kui suured olid Teie kulutused Eestis kokku?**			

K8B	KUI ISE EI TEINUD ÜLDSE KULUTUSI **Märkida null veergu „Summa"**			
K9	**Mitme inimese kulutused need olid?** ……inimeste arv **KUI KÜLASTAJA VIIBIS EESTIS ALLA 3 TUNNI, SIIS SIIN KÜSITLUSE LÕPP**			

K10	**Kuidas korraldasite oma reisi Eestisse?** *(AINULT ÜKS VASTUS)* **1** ostsite valmisreisi (k.a ristlus) **2** kasutasite reisifirma üksikteenuseid → **K11B** **3** reisisite reisifirma vahenduseta → **K11B**
K11	**Kui palju valmisreis maksis?** Summa_____ valuuta _____
K11A	**Mitme inimese kohta?** _____ inimest

K11B	***Kas Te külastasite või külastate selle reisi jooksul teisi riike peale Eesti?*** **1** – jah **2** – ei → **K12**
K11C	**Milliseid riike?** *(NIMETAGE KÕIK)* **1** Soome **2** Rootsi **3** Läti **4** Leedu **5** Venemaa **6** MUU RIIK……..
K12	***Mitu korda olete enne seda reisi Eestit külastanud?*** **1** mitte ühtegi **3** 2 korda **5** 6–10 korda **7** varem Eestis elanud **2** 1 kord **4** 3–5 korda **6** üle 10 korra
K13	**Milliseid kohti Eestis selle reisi ajal külastasite?** (*NIMETAGE KÕIK KÜLASTATUD KOHAD)* **1** Tallinn **4** Pärnu **7** Tartu **10** Lahemaa / Palmse **13** Valga **2** Narva **5** Haapsalu **8** Viljandi **11** Hiiumaa 1 **14** MUUD KOHAD **3** Rakvere **6** Saaremaa **9** Otepää **12** Võru ……
K13A	**Missugune neist oli Teie peamine sihtkoht Eestis?** …… *(KOHANIMI)*
K14	**Milliste liiklusvahenditega Eestis sõitsite?** (*NIMETAGE KUNI **KAKS** PEAMIST LIIKLUSVAHENDIT)* **1** rendiauto/-mikrobuss **3** liinibuss, rong vm **5** jalgratas **2** mitterenditud auto/mikrobuss **4** ekskursioonibuss **6** muu (nt veoauto, mootorratas)

K15	**Kus Te Eestis ööbisite?** *NB! Loetleda kõik ööbimiskohad. Mitte arvestada ööbimisi liinibussis, laevas või rongis*. **Majutuskoha liik:** **1** – hotell, motell, külalistemaja, hostel, turismitalu, kämping, telkimine tasulisel territooriumil, sanatoorium jmt; **2** – üüritud tuba, korter või suvila; **3** – tasuta majutus sugulaste või tuttavate juures, tasuta telkimine jmt; **4** – isiklik korter, suvila või muu isiklik elamispind; tööandja elamispind.		
	Liik	**Asukoht (maakond, linn)**	**Ööbimiste arv**
K15.1			

K15.2			
K15.3			
K17	***Kas teiega koos reisis alla 15-aastasi lapsi?*** ***1*** *– jah* ***2*** *– ei*		

K18	***Kas võtsite Eestis osa järgmistest tegevustest?***		
K18.1	organiseeritud ekskursioon (giidiga)	**Jah**	**Ei**
K18.2	iseseisvalt vaatamisväärsustega tutvumine	**Jah**	**Ei**
K18.3	kultuuriüritused	**Jah**	**Ei**
K18.4	muuseumi, näituse külastamine	**Jah**	**Ei**
K18.5	matkamine või looduses viibimine	**Jah**	**Ei**
K18.6	muu aktiivne harrastus või sport, spordivõistluse jälgimine	**Jah**	**Ei**
K18.7	sisseostude tegemine	**Jah**	**Ei**
K18.8	restoranis / pubis / kohvikus käimine	**Jah**	**Ei**
K18.9	ööelu	**Jah**	**Ei**
K18.10	ilu- või raviteenuste kasutamine	**Jah**	**Ei**

K19	**Palun hinnake järgmisi valdkondi Eestis skaalal 1 – väga halb, 5 – väga hea:** *(ÜKS VASTUS)*						
K19.1	*mulje Eesti inimestega suhtlemisest*	***1***	***2***	***3***	***4***	***5***	***0***
K19.2	*info kättesaadavus enne Eestisse saabumist*	***1***	***2***	***3***	***4***	***5***	***0***
K19.3	vajaliku info kättesaadavus Eestis viibides	***1***	***2***	***3***	***4***	***5***	***0***
K19.4	turvalisus	***1***	***2***	***3***	***4***	***5***	***0***
K20	***Milline oli Eesti-reis võrreldes ootusega?*** **5** palju parem ootusest **4** parem ootusest **3** vastas ootusele **2** halvem ootusest **1** palju halvem ootusest						
K21	**Kas Teie vanemad või Teie ise olete sündinud Eestis?**	**1** – jah	**2** – ei				
K22	**Kas Teil on Eestis sugulasi või tuttavaid?**	**1** – jah	**2** – ei				

K24	***TÄIDAB KÜSITLEJA*** *(AINULT MAANTEEL)* ***Liiklusvahend****:* ***1*** *liinibuss* ***2*** *ekskursioonibuss* ***3*** *sõiduauto* ***4*** *veoauto* ***5*** *jalgratas* ***6*** *jalgsi*
SUUR TÄNU TEILE ABI EEST!	kellaaeg: |_|_|_|_| kuupäev: |_|_|_|_| **2014**

Lisa 2 – Andmestiku *Väliskülastajad Eestis* kontrollreeglid

Esimene reeglite struktuur:

$alg\_VALISK15\_1ARV \geq 0$

$alg\_VALISK15\_2ARV \geq 0$

$alg\_VALISK15\_3ARV \geq 0$

$alg\_VALISK15\_4ARV \geq 0$

$alg\_VALISK15\_5ARV \geq 0$

$alg\_VALISK03\_OOD \geq 0$

$alg\_VALISK15\_1ARV + alg\_VALISK15\_2ARV + alg\_VALISK15\_3ARV +$
$alg\_VALISK15\_4ARV + alg\_VALISK15\_5ARV = alg\_VALISK03\_OOD$

Teine reeglite struktuur:

KUI $alg\_VALISK06A = 01$ SIIS $alg\_VALISK06\_01 = 1$

KUI $alg\_VALISK06A = 02$ SIIS $alg\_VALISK06\_02 = 1$

KUI $alg\_VALISK06A = 03$ SIIS $alg\_VALISK06\_03 = 1$

KUI $alg\_VALISK06A = 04$ SIIS $alg\_VALISK06\_04 = 1$

KUI $alg\_VALISK06A = 05$ SIIS $alg\_VALISK06\_05 = 1$

KUI $alg\_VALISK06A = 06$ SIIS $alg\_VALISK06\_06 = 1$

KUI $alg\_VALISK06A = 07$ SIIS $alg\_VALISK06\_07 = 1$

KUI $alg\_VALISK06A = 08$ SIIS $alg\_VALISK06\_08 = 1$

KUI $alg\_VALISK06A = 09$ SIIS $alg\_VALISK06\_09 = 1$

KUI $alg\_VALISK06A = 10$ SIIS $alg\_VALISK06\_10 = 1$

Kolmas reeglite struktuur:

KUI $alg\_VALISK08\_1 = 1$ SIIS $alg\_VALISK02 \in (3,4)$

Neljas reeglite struktuur:

$alg\_VALISK08\_1\_SUMMA \geq 0$

$alg\_VALISK08\_2\_SUMMA \geq 0$

$alg\_VALISK08\_3\_SUMMA \geq 0$

$alg\_VALISK08\_4\_SUMMA \geq 0$

$alg\_VALISK08\_5\_SUMMA \geq 0$

$alg\_VALISK08\_6\_SUMMA \geq 0$

$alg\_VALISK08\_7\_SUMMA \geq 0$

$alg\_VALISK08A \geq 0$

$alg\_VALISK08\_1\_SUMMA + alg\_VALISK08\_2\_SUMMA + alg\_VALISK08\_3\_SUMMA + alg\_VALISK08\_4\_SUMMA + alg\_VALISK08\_5\_SUMMA + alg\_VALISK08\_6\_SUMMA + alg\_VALISK08\_7\_SUMMA = alg\_VALISK08A$

Viies reeglite struktuur:

KUI $alg\_VALISK13A = 01$ SIIS $alg\_VALISK13\_1 = 1$

KUI $alg\_VALISK13A = 02$ SIIS $alg\_VALISK13\_2 = 1$

KUI $alg\_VALISK13A = 03$ SIIS $alg\_VALISK13\_3 = 1$

KUI $alg\_VALISK13A = 04$ SIIS $alg\_VALISK13\_4 = 1$

KUI $alg\_VALISK13A = 05$ SIIS $alg\_VALISK13\_5 = 1$

KUI $alg\_VALISK13A = 06$ SIIS $alg\_VALISK13\_6 = 1$

KUI $alg\_VALISK13A = 07$ SIIS $alg\_VALISK13\_7 = 1$

KUI $alg\_VALISK13A = 08$ SIIS $alg\_VALISK13\_8 = 1$

KUI $alg\_VALISK13A = 09$ SIIS $alg\_VALISK13\_9 = 1$

KUI $alg\_VALISK13A = 10$ SIIS $alg\_VALISK13\_10 = 1$

KUI $alg\_VALISK13A = 11$ SIIS $alg\_VALISK13\_11 = 1$

KUI $alg\_VALISK13A = 12$ SIIS $alg\_VALISK13\_12 = 1$

KUI $alg\_VALISK13A = 13$ SIIS $alg\_VALISK13\_13 = 1$

KUI $alg\_VALISK13A = 14$ SIIS $alg\_VALISK13\_14 = 1$

Kuues reeglite struktuur:

KUI $VALISK15\_1ASUK\_mk = 37$ SIIS $VALISK13\_1\_mk = 37$

KUI $VALISK15\_1ASUK\_mk = 44$ SIIS $VALISK13\_2\_mk = 44$

KUI $VALISK15\_1ASUK\_mk = 59$ SIIS $VALISK13\_3\_mk = 59$

KUI $VALISK15\_1ASUK\_mk = 67$ SIIS $VALISK13\_4\_mk = 67$

KUI $VALISK15\_1ASUK\_mk = 57$ SIIS $VALISK13\_5\_mk = 57$

KUI $VALISK15\_1ASUK\_mk = 74$ SIIS $VALISK13\_6\_mk = 74$

KUI $VALISK15\_1ASUK\_mk = 78$ SIIS $VALISK13\_7\_mk = 78$

KUI $VALISK15\_1ASUK\_mk = 84$ SIIS $VALISK13\_8\_mk = 84$

KUI $VALISK15\_1ASUK\_mk = 82$ SIIS $VALISK13\_9\_mk = 82$

KUI $VALISK15\_1ASUK\_mk = 59$ SIIS $VALISK13\_10\_mk = 59$

KUI $VALISK15\_1ASUK\_mk = 39$ SIIS $VALISK13\_11\_mk = 39$

KUI $VALISK15\_1ASUK\_mk = 86$ SIIS $VALISK13\_12\_mk = 86$

KUI $VALISK15\_1ASUK\_mk = 82$ SIIS $VALISK13\_13\_mk = 82$

Seitsmes reeglite struktuur:

KUI $alg\_VALISK18\_7 = 1$ SIIS $alg\_VALISK08\_7 = 1$

Kaheksas reeglite struktuur:

KUI $alg\_VALISK18\_8 = 1$ SIIS $alg\_VALISK08\_2 = 1$

Lisa 3 – Rakendustarkvara R kood

```
## Osa 1 - Pakettide installeerimine ning andmete sisselugemine ja korrastamine


#Vajalikud paketid

#load("editrules")

#Loeme andmed sisse

alg_andmed=read.csv("//failid/SA/Kasutajad/Joosep.Raudsik/Desktop/valis2.csv",sep=";")


andmed=subset(alg_andmed, select=c(

alg_VALISK15_1ARV,alg_VALISK15_2ARV,alg_VALISK15_3ARV,

alg_VALISK15_4ARV , alg_VALISK15_5ARV, alg_VALISK03_OOD,alg_VALISK06A, alg_VALISK06_01,

alg_VALISK06_02, alg_VALISK06_03, alg_VALISK06_04, alg_VALISK06_05, alg_VALISK06_06,

alg_VALISK06_07, alg_VALISK06_08, alg_VALISK06_09, alg_VALISK06_10, alg_VALISK08_1,

alg_VALISK02, alg_VALISK08_1_SUMMA, alg_VALISK08_2_SUMMA , alg_VALISK08_3_SUMMA,

alg_VALISK08_4_SUMMA , alg_VALISK08_5_SUMMA , alg_VALISK08_6_SUMMA ,alg_VALISK08_7_SUMMA,

alg_VALISK08A, alg_VALISK13A, alg_VALISK13_1, alg_VALISK13_2, alg_VALISK13_3, alg_VALISK13_4,

alg_VALISK13_5, alg_VALISK13_6, alg_VALISK13_7, alg_VALISK13_8, alg_VALISK13_9, alg_VALISK13_10,

alg_VALISK13_11, alg_VALISK13_12, alg_VALISK13_13, alg_VALI SK15_1LIIK, alg_VALISK15_2LIIK,

alg_VALISK15_3LIIK, alg_VALISK15_4LIIK, alg_VALISK15_5LIIK, alg_VALISK08_1_SUMMA,
```

alg_VALISK18_7, alg_VALISK18_8, alg_VALISK08_7, alg_VALISK08_2,alg_VALISK13_14,

VALISK15_1ASUK_mk, VALISK13_1_mk, VALISK13_2_mk, VALISK13_3_mk, VALISK13_4_mk, VALISK13_5_mk,

VALISK13_6_mk, VALISK13_7_mk, VALISK13_8_mk, VALISK13_9_mk, VALISK13_10_mk, VALISK13_11_mk, VALISK13_12_mk,

VALISK13_13_mk

))

andmed$alg_VALISK13A = as.factor(andmed$alg_VALISK13A)

andmed$alg_VALISK13_1 = as.factor(andmed$alg_VALISK13_1)

andmed$alg_VALISK15_1ARV[is.na(andmed$alg_VALISK15_1ARV)] <- 0

andmed$alg_VALISK15_2ARV[is.na(andmed$alg_VALISK15_2ARV)] <- 0

andmed$alg_VALISK15_3ARV[is.na(andmed$alg_VALISK15_3ARV)] <- 0

andmed$alg_VALISK15_4ARV[is.na(andmed$alg_VALISK15_4ARV)] <- 0

andmed$alg_VALISK15_5ARV[is.na(andmed$alg_VALISK15_5ARV)] <- 0

andmed$alg_VALISK03_OOD[is.na(andmed$alg_VALISK03_OOD)] <- 0

andmed$alg_VALISK08_1_SUMMA[is.na(andmed$alg_VALISK08_1_SUMMA)] <- 0

andmed$alg_VALISK08_2_SUMMA[is.na(andmed$alg_VALISK08_2_SUMMA)] <- 0

andmed$alg_VALISK08_3_SUMMA[is.na(andmed$alg_VALISK08_3_SUMMA)] <- 0

andmed$alg_VALISK08_4_SUMMA[is.na(andmed$alg_VALISK08_4_SUMMA)] <- 0

andmed$alg_VALISK08_5_SUMMA[is.na(andmed$alg_VALISK08_5_SUMMA)] <- 0

andmed$alg_VALISK08_6_SUMMA[is.na(andmed$alg_VALISK08_6_SUMMA)] <- 0

andmed$alg_VALISK08_7_SUMMA[is.na(andmed$alg_VALISK08_7_SUMMA)] <- 0

```
andmed$alg_VALISK08A[is.na(andmed$alg_VALISK08A)] <- 0

levels(andmed$alg_VALISK13A) <- c(levels(andmed$alg_VALISK13A), "-1")
levels(andmed$alg_VALISK13_1) <- c(levels(andmed$alg_VALISK13_1), "-1")
levels(andmed$alg_VALISK15_1LIIK) <- c(levels(andmed$alg_VALISK15_1LIIK), "-1")
levels(andmed$alg_VALISK15_2LIIK) <- c(levels(andmed$alg_VALISK15_2LIIK), "-1")
levels(andmed$alg_VALISK15_3LIIK) <- c(levels(andmed$alg_VALISK15_3LIIK), "-1")
levels(andmed$alg_VALISK15_4LIIK) <- c(levels(andmed$alg_VALISK15_4LIIK), "-1")
levels(andmed$alg_VALISK15_5LIIK) <- c(levels(andmed$alg_VALISK15_5LIIK), "-1")

levels(andmed$alg_VALISK06A) <- c(levels(andmed$alg_VALISK06A), "-1")
levels(andmed$alg_VALISK06_05) <- c(levels(andmed$alg_VALISK06_05), "-1")
levels(andmed$alg_VALISK08_1) <- c(levels(andmed$alg_VALISK08_1), "-1")
levels(andmed$alg_VALISK02) <- c(levels(andmed$alg_VALISK02), "-1")

levels(andmed$alg_VALISK18_7) <- c(levels(andmed$alg_VALISK18_7), "-1")
levels(andmed$alg_VALISK08_7) <- c(levels(andmed$ alg_VALISK08_7), "-1")
levels(andmed$alg_VALISK18_8) <- c(levels(andmed$alg_VALISK18_8), "-1")
levels(andmed$alg_VALISK08_2) <- c(levels(andmed$alg_VALISK08_2), "-1")

andmed[andmed==""]<-NA
andmed[is.na(andmed)]<-'-1'

#Muudame sobivale kujule
```

```
attach(andmed)
head(andmed)

        ## OSA 2 – Kontrollreeglite defineerimine ja ülevaade
reeglid <- editset(expression(
        #Esimene
        alg_VALISK15_1ARV>=0,
        alg_VALISK15_2ARV>=0,
        alg_VALISK15_3ARV>=0,
        alg_VALISK15_4ARV>=0,
        alg_VALISK15_5ARV>=0,
        alg_VALISK03_OOD>=0,
        alg_VALISK15_1ARV + alg_VALISK15_2ARV + alg_VALISK15_3ARV +
        alg_VALISK15_4ARV + alg_VALISK15_5ARV == alg_VALISK03_OOD,

        #Teine reegel
        alg_VALISK06A %in% c('01','02','03','04','05','06','07','08','09','10','-1'),
        alg_VALISK06_01 %in% c(1,2,'-1'),
        alg_VALISK06_02 %in% c(1,2,'-1'),
        alg_VALISK06_03 %in% c(1,2,'-1'),
        alg_VALISK06_04 %in% c(1,2,'-1'),
        alg_VALISK06_05 %in% c(1,2,'-1'),
        alg_VALISK06_06 %in% c(1,2,'-1'),
        alg_VALISK06_07 %in% c(1,2,'-1'),
```

```
alg_VALISK06_08 %in% c(1,2,'-1'),
alg_VALISK06_09 %in% c(1,2,'-1'),
alg_VALISK06_10 %in% c(1,2,'-1'),

if (alg_VALISK06A == "01") alg_VALISK06_01 == "1",
if (alg_VALISK06A == "02") alg_VALISK06_02 == "1",
if (alg_VALISK06A == "03") alg_VALISK06_03 == "1",
if (alg_VALISK06A == "04") alg_VALISK06_04 == "1",
if (alg_VALISK06A == "05") alg_VALISK06_05 == "1",
if (alg_VALISK06A == "06") alg_VALISK06_06 == "1",
if (alg_VALISK06A == "07") alg_VALISK06_07 == "1",
if (alg_VALISK06A == "08") alg_VALISK06_08 == "1",
if (alg_VALISK06A == "09") alg_VALISK06_09 == "1",
if (alg_VALISK06A == "10") alg_VALISK06_10 == "1",

#Kolmas
alg_VALISK02 %in% c('1','2','3','4','-1'),
alg_VALISK08_1 %in% c('1','2','-1'),
if (alg_VALISK08_1 == "1") alg_VALISK02 %in% c("3","4"),

#Neljas
alg_VALISK08_1_SUMMA>=0,
alg_VALISK08_2_SUMMA>=0,
alg_VALISK08_3_SUMMA>=0,
```

```
alg_VALISK08_4_SUMMA>=0,

alg_VALISK08_5_SUMMA>=0,

alg_VALISK08_6_SUMMA>=0,

alg_VALISK08_7_SUMMA>=0,

alg_VALISK08A>=0,

alg_VALISK08_1_SUMMA + alg_VALISK08_2_SUMMA + alg_VALISK08_3_SUMMA +

alg_VALISK08_4_SUMMA + alg_VALISK08_5_SUMMA + alg_VALISK08_6_SUMMA +

alg_VALISK08_7_SUMMA == alg_VALISK08A,


#Viies

alg_VALISK13A %in% c('01','02','03','04','05','06','07','08','09','10','11','12','13','14','-1'),

alg_VALISK13_1 %in% c('0','1','2',"-1"),

alg_VALISK13_2 %in% c('0','1','2',"-1"),

alg_VALISK13_3 %in% c('0','1','2',"-1"),

alg_VALISK13_4 %in% c('0','1','2',"-1"),

alg_VALISK13_5 %in% c('0','1','2',"-1"),

alg_VALISK13_6 %in% c('0','1','2',"-1"),

alg_VALISK13_7 %in% c('0','1','2',"-1"),

alg_VALISK13_8 %in% c('0','1','2',"-1"),

alg_VALISK13_9 %in% c('0','1','2',"-1"),

alg_VALISK13_10 %in% c('0','1','2',"-1"),

alg_VALISK13_11 %in% c('0','1','2',"-1"),

alg_VALISK13_12 %in% c('0','1','2',"-1"),

alg_VALISK13_13 %in% c('0','1','2',"-1"),
```

alg_VALISK13_14 %in% c('0','1','2',"-1"),

if (alg_VALISK13A=="01") alg_VALISK13_1=="1",

if (alg_VALISK13A=="02") alg_VALISK13_2=="1",

if (alg_VALISK13A=="03") alg_VALISK13_3=="1",

if (alg_VALISK13A=="04") alg_VALISK13_4=="1",

if (alg_VALISK13A=="05") alg_VALISK13_5=="1",

if (alg_VALISK13A=="06") alg_VALISK13_6=="1",

if (alg_VALISK13A=="07") alg_VALISK13_7=="1",

if (alg_VALISK13A=="08") alg_VALISK13_8=="1",

if (alg_VALISK13A=="09") alg_VALISK13_9=="1",

if (alg_VALISK13A=="10") alg_VALISK13_10=="1",

if (alg_VALISK13A=="11") alg_VALISK13_11=="1",

if (alg_VALISK13A=="12") alg_VALISK13_12=="1",

if (alg_VALISK13A=="13") alg_VALISK13_13=="1",

if (alg_VALISK13A=="14") alg_VALISK13_14=="1",

###Kuues

VALISK15_1ASUK_mk %in% c('37','44','59','67','57','74','78','84','82','59','39','86','81', -1),

VALISK13_1_mk %in% c(-1, 37),

VALISK13_2_mk %in% c(-1, '44'),

VALISK13_3_mk %in% c(-1, '59'),

VALISK13_4_mk %in% c(-1, '67'),

VALISK13_5_mk %in% c(-1, '57'),

```
VALISK13_6_mk %in% c(-1, '74'),

VALISK13_7_mk %in% c(-1, '78'),

VALISK13_8_mk %in% c(-1, '84'),

VALISK13_9_mk %in% c(-1, '82'),

VALISK13_10_mk %in% c(-1, '59'),

VALISK13_11_mk %in% c(-1, '39'),

VALISK13_12_mk %in% c(-1, '86'),

VALISK13_13_mk %in% c(-1, '82'),


if (VALISK15_1ASUK_mk=='37') VALISK13_1_mk=='37',

if (VALISK15_1ASUK_mk=='44') VALISK13_2_mk=='44',

if (VALISK15_1ASUK_mk=='59') VALISK13_3_mk=='59',

if (VALISK15_1ASUK_mk=='67') VALISK13_4_mk=='67',

if (VALISK15_1ASUK_mk=='57') VALISK13_5_mk=='57',

if (VALISK15_1ASUK_mk=='74') VALISK13_6_mk=='74',

if (VALISK15_1ASUK_mk=='78') VALISK13_7_mk=='78',

if (VALISK15_1ASUK_mk=='84') VALISK13_8_mk=='84',

if (VALISK15_1ASUK_mk=='82') VALISK13_9_mk=='82',

if (VALISK15_1ASUK_mk=='59') VALISK13_10_mk=='59',

if (VALISK15_1ASUK_mk=='39') VALISK13_11_mk=='39',

if (VALISK15_1ASUK_mk=='86') VALISK13_12_mk=='86',

if (VALISK15_1ASUK_mk=='82') VALISK13_13_mk=='82',


#Seitsmes
```

```
        alg_VALISK08_7 %in% c('1','2','-1'),
        alg_VALISK18_7 %in% c('1','2','-1'),
        if (alg_VALISK18_7=='1') alg_VALISK08_7=='1',

        #Kaheksas
        alg_VALISK18_8 %in% c('1','2','-1'),
        alg_VALISK08_2 %in% c('1','2','-1'),
        if (alg_VALISK18_8=='1') alg_VALISK08_2=='1'
))

        ## OSA 3 - Meetodite rakendamine
#Leiame Fellegi-Holti meetodil vead
el <- localizeErrors(reeglid,andmed)
```

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Joosep Raudsik (sünnikuupäev: 22.01.1991)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
   Fellegi-Holt´i meetod ja deduktiivne imputeerimine andmestiku *Väliskülastajad Eestis* näitel, mille juhendajad on Mare Vähi ja Maiki Ilves,

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 13.05.2014