

Algoritmi korrektsuse tõestamisest

(Pisut matemaatika ja informaatika vahekorrast)

Jüri Kiho

emeritprofessor

<http://kodu.ut.ee/~kiho/>

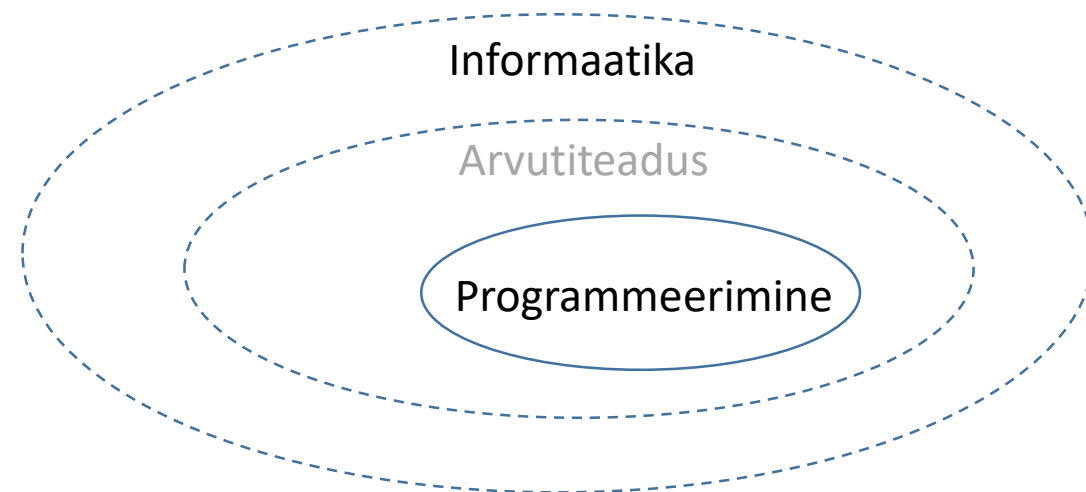
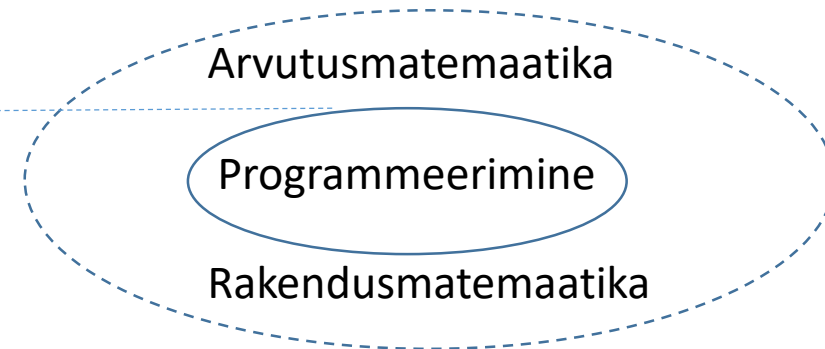
jyri.kiho@ut.ee

19. september 2018

Matemaatikaõpe teaduskonniti Tartu ülikoolis*

- Filosoofiateaduskond (al 1802.a.)
- Füüsika-matemaatikateaduskond (1850)
- Matemaatika-Loodusteaduskond (1918)
- Füüsika-matemaatikateaduskond (1961)
- Matemaatikateaduskond (1967)
- Matemaatika-informaatikateaduskond (2001)
- Loodus- ja täppisteaduste valdkond (2016)

Esimene
arvuti



Täpsustus:

Pisut matemaatika ja programmeerimise vahekorra

*

<https://www.math.ut.ee/et/Teaduskonna-ajalugu-ja-koosseis>

Kas

$$c^2 = a^2 + b^2$$

kehtib?

Kas

$$c^2 = a^2 + b^2$$

kehtib?

Kas kehtib (TEOREEM):

Kui a ja b on täisnurkse kolmnurga kaatetite pikkused,

siis

$$c^2 = a^2 + b^2$$

kus c on selle kolmnurga hüpotenuusi pikkus.

...

EELDUS

VÄIDE

TÕESTUS

Kas algoritm

$$x := x^2 - y^2 - 1$$

on korrektne?

Kas

$$c^2 = a^2 + b^2$$

kehtib?

~

Selle omistamisdirektiivi kujusid mõnedes programmeerimiskeelsetes programmides:

```
x = x*x - y*y - 1;
```

```
x = Math.pow(x,2) - Math.pow(y,2) - 1;  
x = Math.pow(x,2.0) - Math.pow(y,2.0) - 1;
```

```
x = x**2 - y**2 - 1;
```

```
x = pow(x,2) - pow(y,2) - 1;
```

Kas algoritm

$x := x^2 - y^2 - 1$

on korrektne?

Kas algoritm

--- $x + y \leq 1$

$x := x^2 - y^2 - 1$

--- $x < 2y$

Algoritmi korrektsus

EELTINGIMUS (P)

TEGEVUS (ALGORITMI SISEMINE SPETSIFIKATSIOON Π)

JÄRELTINGIMUS (Q)

on korrektne?

Tingimuste paari P, Q nimetatakse algoritmi väliseks spetsifikatsiooniks.

ALGORITMI KORREKTSUSE **TEOREEM**:

Kui algoritmi tegevus (Π) algab olekus, mis rahuldab eeltingimust (P),
siis tegevus lõpeb olekus, mis rahuldab järeltingimust (Q).

EELDUS
VÄIDE

(Olek: muutujad ja nende jooksvad väärtused.)

...

TÕESTUS

Algoritm on korrektne, kui on tõestatud selle algoritmi korrektsuse teoreem.

Algoritmi korrektsuse näitamiseks tuleb tõestada selle algoritmi korrektsuse teoreem.

Testimisega korrektsust (üldiselt) tõestada ei saa. Küll aga saab sellega näidata, et algoritm ei ole korrektne.

Kas algoritm (programm)

--- $x + y \leq 1$	EELTINGIMUS (P)
$x := x^2 - y^2 - 1$	TEGEVUS (ALGORITMI SISEMINE SPETSIFIKATSIOON Π)
--- $x < 2y$	JÄRELTINGIMUS (Q)

on korrektne?

Vastus: see algoritm ei ole korrektne, sest testrakendusel eeltingimust rahuldavate arvudega $x = 3$ ja $y = -3$ saame oleku $x = -1$ ja $y = -3$, kus järeltingimus $x < 2y$ ei ole rahuldatud.

Algoritm $\mathcal{A}$

— — — $0.5 \leq a \leq 0.7 \wedge 1.8 \leq n \leq 2$

eeltingimus

$n := n + 1$

$\alpha := 1 - 2n$

$a := 1 - a$

$b := 2 + \cos^2 \alpha$

$b := b - \log_3(n^2 + \alpha) + \sin^2 \alpha - 3$

$x := \log_3(n^2 + \alpha)^2$

$y := ab$

$x := x + 2y$

$x := x/2$

— — — $0.5 \leq x \leq 1$

tegevus

järelingimus



See on **teoreem (oletus, väitlause)**:

Kui algoritmi $\mathcal{A}$ tegevus algab olekus, mis rahuldab eeltingimust $0.5 \leq a \leq 0.7 \wedge 1.8 \leq n \leq 2$,
siis tegevus lõpeb olekus, mis rahuldab järelingimust $0.5 \leq x \leq 1$.

EELDUS
VÄIDE

???

TÕESTUS

Programmi korrektsus

Programm $PR(\mathcal{A})$:

```
// 0.5 <= a <= 0.7 ja 1.8 <= n <= 2.0          (eeltingimus)
n = n + 1;
double alfa = 1 - 2*n;
a = 1 - a;
double b = 2 + Math.cos(alfa)*Math.cos(alfa);
b = b - log3(n*n + alfa) + Math.sin(alfa)*Math.sin(alfa) - 3 ;
double x = log3((n*n + alfa)*(n*n + alfa));
double y = a*b;
x = x + 2*y;
x = x/2;
// 0.5 <= x <= 1.0                               (järeltingimus)
```

Programmi $PR(\mathcal{A})$ testimise tulemusi:

n = 1,90;	a = 0,60;	x = 0,701
n = 1,80;	a = 0,50;	x = 0,535
n = 2,00;	a = 0,70;	x = 0,883
n = 1,93;	a = 0,59;	x = 0,706

Ei tõesta teoreemi.

Näitab vaid kehtivust neil neljal juhul.

Teoreem (oletus, väitlause):

Kui programmi $PR(\mathcal{A})$ tegevus algab olekus, mis rahuldab eeltingimust, siis tegevus lõpeb olekus, mis rahuldab järeltingimust.

EELDUS

VÄIDE

Omistamisdirektiivi aksioom

Selleks, et pärast omistamisdirektiivi

$x := e$

täitmist kehtiks (järelingimus) Q , peab enne täitmist kehtima (nõrgim eelingimus) Q_e^x . Viimane saadakse järelingimusest Q selles muutuja x kõigi esinemiste asendamisel avaldisega e .

Algoritm

--- Q_e^x
 $x := e$
 --- Q
 on korrektne

On korrektne:

--- $|x| < |y + 1|$ Q_e^x

$x := x^2 - y^2 - 1$

--- $x < 2y$

järelingimus (Q)

$x := x^2 - y^2 - 1$
 $x := e$

--- $x < 2y$

Algoritm

--- $x > 0 \wedge y > 0 \wedge x + y \leq 1$ P

$x := x^2 - y^2 - 1$

--- $x < 2y$ Q

on korrektne, sest eelingimusest P järeldub nõrgim eelingimus Q_e^x :

$x > 0 \wedge y > 0 \wedge x + y \leq 1 \Rightarrow x > 0 \wedge y > 0 \wedge x < y + 1$

$x \leq -y + 1$

$x < -y + 1 + 2y$

$x > 0 \wedge y > 0 \wedge x < y + 1 \Rightarrow |x| < |y + 1|$

$$Q_e^x \Leftrightarrow x^2 - y^2 - 1 < 2y \Leftrightarrow x^2 < y^2 + 2y + 1 \Leftrightarrow x^2 < (y + 1)^2 \Leftrightarrow |x| < |y + 1|$$

Lineaarse algoritmi korrektsuse tõestamine (näide 1)

Algoritmi $\mathcal{A}$

$0.5 \leq a \leq 0.7 \wedge 1.8 \leq n \leq 2$

$n := n + 1$

$\alpha := 1 - 2n$

$a := 1 - a$

$b := 2 + \cos^2 \alpha$

$b := b - \log_3(n^2 + \alpha) + \sin^2 \alpha - 3$

$x := \log_3(n^2 + \alpha)^2$

$y := ab$

$x := x + 2y$

$x := x/2$

$0.5 \leq x \leq 1$

Eeltingimus

Algoritmi sisemiseks spetsifikatsiooniks on omistamisdirektiivide järjend.

Järelingimus



I.

--- $0.5 \leq a \leq 0.7 \wedge 1.8 \leq n \leq 2$



$\log_3 1.8 = 0.535$
 $\log_3 2.0 = 0.631$

--- $1 \leq 4a \log_3 n \leq 2 \iff 0.25 \leq a \log_3 n \leq 0.5$

$n := n + 1$

--- $1 \leq 2a \log_3(n^2 + 1 - 2n) \leq 2 \iff 1 \leq 2a \log_3(n - 1)^2 \leq 2$

$\alpha := 1 - 2n$

--- $1 \leq 2a \log_3(n^2 + \alpha) \leq 2$

$a := 1 - a$

--- $1 \leq \log_3(n^2 + \alpha)^2 + 2a(2 + \cos^2 \alpha - \log_3(n^2 + \alpha) + \sin^2 \alpha - 3) \leq 2 \iff$

$\iff 1 \leq \log_3(n^2 + \alpha)^2 + 2a(-\log_3(n^2 + \alpha)) \leq 2 \iff$

$\iff 1 \leq 2 \log_3(n^2 + \alpha) - 2a \log_3(n^2 + \alpha) \leq 2 \iff$

$\iff 1 \leq 2(1 - a) \log_3(n^2 + \alpha) \leq 2$

$b := 2 + \cos^2 \alpha$

--- $1 \leq \log_3(n^2 + \alpha)^2 + 2a(b - \log_3(n^2 + \alpha) + \sin^2 \alpha - 3) \leq 2$

$b := b - \log_3(n^2 + \alpha) + \sin^2 \alpha - 3$

--- $1 \leq \log_3(n^2 + \alpha)^2 + 2ab \leq 2$

$x := \log_3(n^2 + \alpha)^2$

--- $1 \leq x + 2ab \leq 2$

$y := ab$

--- $1 \leq x + 2y \leq 2$

$x := x + 2y$

--- $0.5 \leq x/2 \leq 1 \iff 1 \leq x \leq 2$

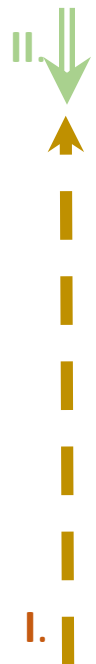
$x := x/2$

--- $0.5 \leq x \leq 1$

Lineaarse algoritmi korrektsuse tõestamine (näide 2)

Algoritm $\mathcal{C}$

$$--- k \neq n \wedge s = 1 - \frac{1}{k}$$
$$m := k + 1$$
$$s := s + \frac{1}{km}$$
$$k := m$$
$$--- s = 1 - \frac{1}{k}$$



$$--- k \neq n \wedge s = 1 - \frac{1}{k}$$
$$--- s + \frac{1}{k(k+1)} = 1 - \frac{1}{k+1}$$
$$m := k + 1$$
$$--- s + \frac{1}{km} = 1 - \frac{1}{m}$$
$$s := s + \frac{1}{km}$$
$$--- s = 1 - \frac{1}{m}$$
$$k := m$$
$$--- s = 1 - \frac{1}{k}$$

$$\Rightarrow s = 1 - \frac{1}{k+1} - \frac{1}{k(k+1)} \Rightarrow s = 1 - \frac{1}{k}$$

Nõrgim eeltingimus

Algoritm $\mathcal{C}$ on korrektne,
sest nõrgim eeltingimus $s = 1 - \frac{1}{k}$
järeljub eeltingimusest $k \neq n \wedge s = 1 - \frac{1}{k}$

NB!
Predikaat $s = 1 - \frac{1}{k}$
on invariant(ne)

Π Tsükkel
 $t?$ lõpetamistingimus
 $\leftarrow$ tsüklist väljumine (kui tingimus t on täidetud)
 Π_0 tsükli sisu (keha)

Korrektuse käsitlemiseks tuleb tsükkel varustada eeltingimusega (P) ja järeltingimusega (Q) ning tsükli invariandiga (R):

$---$ P
 Π Tsükkel
 $t?$
 $\leftarrow$
 $---$ R
 Π_0
 $---$ Q

Tsükli direktiivi aksioom

Π on (osaliselt) korrektne, kui

1) $P \Rightarrow R$ (tsükli sisenemisel kehtib R).

2) Algoritm

$---$ $\neg t \wedge R$

Π_0

$---$ R

on korrektne (st R on tsükli sisus invariantne).

3) $t \wedge R \Rightarrow Q$ (tsüklist väljumisel kehtivast tingimusest järelneb järeltingimus).

$--- P$
 $\left[\begin{array}{l} \text{Tsükkel } \Pi \\ t? \\ \leftarrow \\ --- R \\ \Pi_0 \end{array} \right]$
 $--- Q$

$--- -k \neq n \wedge s = 1 - \frac{1}{k} \quad P$
 $\left[\begin{array}{l} \text{Tsükkel } \Pi \\ k = n? \\ \leftarrow \\ --- -s = 1 - \frac{1}{k} \quad R \\ m := k + 1 \\ s := s + \frac{1}{km} \\ k := m \end{array} \right]$
 $--- -s = 1 - \frac{1}{n} \quad Q$

Tsükkel Π on (osaliselt) korrektne, sest

1) $P \Rightarrow R$ (tsükli sisenemisel kehtib R).

2) Algoritm

$--- \neg t \wedge R$

Π_0

$--- R$

Kehtib, sest on eespool

tõestatud algoritm $\mathcal{C}$ (vt näide 2)

on korrektne (st R on tsükli sisus invariantne).

3) $t \wedge R \Rightarrow Q$ (tsüklist väljumisel kehtivast tingimusest järeldub järeldingimus).

Tsüklilise algoritmi korrektsus

--- Algoritmi eeltingimus PD

ALGATUSTEGEVUS

--- P

Tsükkel Π

$t?$

←

--- R

Π_0

--- Q (on ka algoritmi järeltingimuseks)

Algoritm $\mathcal{D}$

--- $n \geq 5$

$s := 0.5$

$k := 2$

--- $k \neq n \wedge s = 1 - \frac{1}{k}$

Tsükkel Π

$k = n?$

←

--- $s = 1 - \frac{1}{k}$

$m := k + 1$

$s := s + \frac{1}{km}$

$k := m$

--- $s = 1 - \frac{1}{n}$

--- $-2 \neq n \wedge 0.5 = 1 - \frac{1}{2}$

P

t

R

Q

PD
 PD_0

III. Näidata, et
 $PD \Rightarrow PD_0$

II. Leida algatustegevuse
nõrgim eeltingimus PD_0

I. Näidata, et tsükkel
on korrektne
(vt eelmine slaid)

Tsükkel Π

$i = n + 1?$

←

$$s := s + \frac{1}{(i-1)i}$$

$i := i + 1$

Algoritm $\mathcal{E}1$, saab tõestada.

$$--- s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k} \quad P$$

$i = n + 1?$

←

$$--- s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k}$$

$$s := s + \frac{1}{(i-1)i}$$

$i := i + 1$

t

R

$$--- s = \sum_{k=2}^n \frac{1}{(k-1)k} \quad Q$$

Algoritm $\mathcal{E}2$, saab tõestada. Ja tõestamegi.

$s := 0$

$i := 2$

$$--- s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{i-1} \Leftrightarrow P$$

$i = n + 1?$

$--- i = n + 1 \Leftrightarrow t$

←

$$--- s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{i-1} \Leftrightarrow R$$

$$s := s + \frac{1}{(i-1)i}$$

$i := i + 1$

$$--- s = \sum_{k=2}^n \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{n} \Leftrightarrow Q$$

(1)

(2)

(3)

Tsükkel on (osaliselt) korrektne, sest kehtivad 1), 2), 3):

1) $P \Leftrightarrow R$

3) $t \wedge R \Rightarrow Q$:

$$i = n + 1 \wedge s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{i-1} \Leftrightarrow \sum_{k=2}^n \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{n}$$

2) R on invariant:

$$--- \neg t \wedge R \Leftrightarrow i \neq n+1 \wedge s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{i-1} \quad (4)$$

$$--- s + \frac{1}{(i-1)i} = \sum_{k=2}^i \frac{1}{(k-1)k} \wedge s + \frac{1}{(i-1)i} = 1 - \frac{1}{i} \Leftrightarrow R \quad (5)$$



$$s := s + \frac{1}{(i-1)i}$$

$$--- s = \sum_{k=2}^i \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{i}$$

$$i := i + 1$$

$$--- R \Leftrightarrow s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{i-1}$$

$$(5) \Leftrightarrow$$

$$\Leftrightarrow s + \frac{1}{(i-1)i} = \sum_{k=2}^i \frac{1}{(k-1)k} \wedge s + \frac{1}{(i-1)i} = 1 - \frac{1}{i}$$

$$\Leftrightarrow s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k} \wedge s = 1 - \left(\frac{1}{i} + \frac{1}{(i-1)i} \right)$$

$$\Leftrightarrow s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k} \wedge s = 1 - \frac{i-1+1}{(i-1)i}$$

$$\Leftrightarrow s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{i-1} \Leftrightarrow R$$

Algoritm $\mathcal{E2}$

$s := 0$

$i := 2$

$$--- s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{i-1} \Leftrightarrow P$$

$$i = n + 1? \quad --- i = n + 1 \Leftrightarrow t$$

$\leftarrow$

$$--- s = \sum_{k=2}^{i-1} \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{i-1} \Leftrightarrow R$$

$$s := s + \frac{1}{(i-1)i}$$

$$i := i + 1$$

$$--- s = \sum_{k=2}^n \frac{1}{(k-1)k} \wedge s = 1 - \frac{1}{n} \Leftrightarrow Q$$

Nõrgim eeltingimus, et pärast alustustegevusi

$s := 0$

$i := 2$

kehtiks

P , on:

tõene.

(1)

(2)

Seega, alustades

algoritmi $\mathcal{E2}$ ükskõik millisest olekust (ükskõik millise n väärtuse korral), siis töö lõppemise korral

muutuja s väärtuseks on summa

(3)

$$\sum_{k=2}^n \frac{1}{(k-1)k}, \text{ mis võrdub arvuga } 1 - \frac{1}{n}$$

Kokkuvõtteks (programmide aspektist)

- Vähegi suurema programmi korrektsuse käsitsi tõestamine ei ole praktiliselt võimalik.
 - Arvutusliku/matemaatilise iseloomuga programmide automaatne tõestamine on teostatav.
 - Reaaleluliste/andmetöötluse programmide korrektsuse tõestamine ei ole praktiliselt võimalik.
- Peamiseks takistuseks on välise spetsifikatsiooni (st eel- ja järeltingimuse) formaalsel kujul esitamise ülisuur keerukus.
- Näiteks kas või:

On koostatud programm järgmise pakkimisprobleemi lahendamiseks.

Antud on rida pealt lahtisi risttahukakujulisi kaste. Kasti suurus on määratud kolme parameetriga, milleks on alumise tahu lähiskülgede pikkused ja kasti kõrgus. Kasti tahu paksus on kõikidel kastidel sama.

Antud kastidest tuleb koostada võimalikult lühike objektide järjend, milles objektiks on kas üks tühi kast, või kast, milles on üks teine (tühi) kast.

Kast mahutatakse teise nii, et selle lahtine tahk on ikka pealpool; mahutatud kasti ülaserv ei tohi olla mahutava kasti ülaservast kõrgemal.

Praktilises tarkvaraarenduses koostatavate programmide korrektsus jääb reeglina tõestamata. See tähendab, et programmeerija põhitegevuseks on tõestamata väidete (teoreemide) kirjutamine:

Koostasin ja testisin programmi ***PR***. Väidan, et see programm on korrektne (et kehtib teoreem, oletus, väitlause):

Kui minu kirjutatud programm ***PR*** alustab tööd eeldustekohaste (eeltingimust rahuldavate) andmetega, siis selle (tegevuste osa) täitmine lõpeb lõpliku arvu sammude järel ja saadud tulemus rahuldab püstitatud nõudeid (järeltingimust).

Aga selle tõestust paraku ei ole ...

Kirjandus

Elementaarseid momente [1], ptk 8 ja lihtsaid ülesandeid [2], ptk 18-III. Põhjalik käsitus [3] E. W. Dijkstra (1930 – 2002).

[1] J. Kiho. *Algoritmid ja andmestruktuurid*. Kolmas, parandatud ja täiendatud trükk. TÜ, 2003, 147 lk.
<http://dspace.ut.ee/bitstream/handle/10062/16872/9985567676.pdf?sequence=1> (12.09.2018)

[2] A. Peder, J. Kiho, H. Nester. *Algoritmid ja andmestruktuurid. Ülesannete kogu*. TÜ, 2017, 131 lk.
https://moodle.ut.ee/pluginfile.php/570809/mod_resource/content/2/YlesanneteKogu_ver_4_0.pdf (12.09.2018)

[3] Edsger W. Dijkstra. *A Discipline of Programming*. Prentice-Hall Inc. Englewood Cliffs, N. J. 1976