Mida tähendab, et programm on korrektne?

- Mis on programm?
- Mida ta teeb?
- Kuidas tema tegevuse üle arutleda?

Olgu meil antud mingi *muutujate* hulk **Var**.

Aritmeetilist avaldist A defineeriv *abstraktne süntaks* on järgmine:

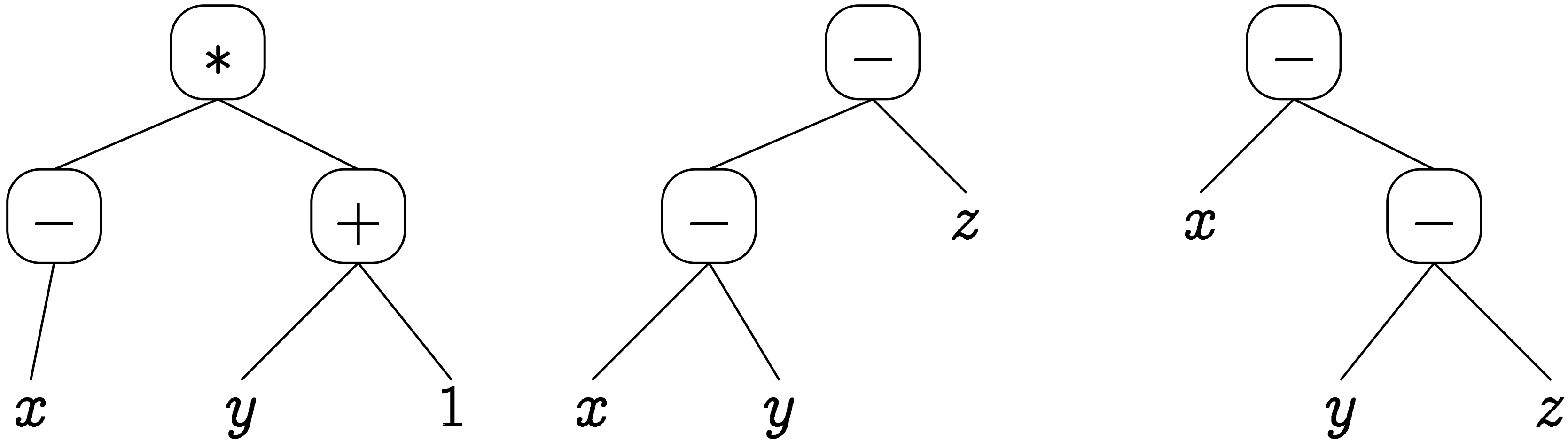
$$\begin{array}{lcl} A & ::= & n \\ & | & x \\ & | & -A_1 \\ & | & A_1 + A_2 \mid A_1 - A_2 \mid A_1 * A_2 \mid A_1/A_2 \end{array}$$

kus A_1, A_2 on aritmeetilised avaldised, $x \in \mathbf{Var}$ ja $n \in \mathbb{Z}$.

Olgu **Aexp** kõigi aritmeetiliste avaldiste hulk.

Eelmisel slaidil on toodud abstraktne süntaks.

Objekte, mida see süntaks defineerib, tuleks ette kujutada termidena / puudena.



$$(-x) * (y+1) \qquad\qquad (x-y)-z \qquad\qquad x-(y-z)$$

Konkreetne süntaks meid ei huvita. Kui vaja, siis lisame mingid grupeerijad (sulud).

Tõeväärtusavaldist B defineeriv abstraktne süntaks on järgmine:

$$\begin{array}{rcl} B & ::= & \mathsf{true} \mid \mathsf{false} \\ & \mid & A_1 = A_2 \mid A_1 \leqslant A_2 \mid \ldots \\ & \mid & \neg B_1 \\ & \mid & B_1 \wedge B_2 \mid B_1 \vee B_2 \mid \ldots \end{array}$$

kus A_1, A_2 on aritmeetilised avaldised, B_1 ja B_2 tõeväärtusavaldised.

Olgu **Bexp** kõigi tõeväärtusavaldiste hulk.

Olgu $\mathbb{B} = \{\mathsf{true}, \mathsf{false}\}$.

Programmi S defineeriv abstraktne süntaks on järgmine:

$$\begin{array}{rcl} S & ::= & x := A \\ & | & skip \\ & | & S_1 ; S_2 \\ & | & \mathit{if}\ B\ \mathit{then}\ S_1\ \mathit{else}\ S_2 \\ & | & \mathit{while}\ B\ \mathit{do}\ S_1 \end{array}$$

kus $x \in \mathbf{Var}$, A on aritmeetiline avaldis, B tõeväärtusavaldis ning S_1 ja S_2 programmid.

Teisi konstruktsioone vaatame hiljem.

Olgu **Prog** kõigi programmide hulk.

Programmiolek on kujutus, mis seab igale muutujale vastavusse tema väärtuse.

Võimalike väärtuste hulk on $\mathbf{Val} = \mathbb{Z}$. Võimalike programmiolekute hulk on

$$\mathbf{State} = \mathbf{Var} \to \mathbf{Val} \quad .$$

Kui $s \in \mathbf{State}$ ja $x \in \mathbf{Var}$, siis $s(x)$ on muutuja x väärtus programmiolekus s.

Programmi täitmine kujutab endast programmioleku muutmist vastavalt eeskirjale (milleks on see programm).

Aritmeetilisele avaldisele A ja programmiolekule s seame vastavusse täisarvu $\mathcal{A}[\![A]\!]s$.

Funktsiooni $\mathcal{A}$ tüüp on

$$\mathbf{Aexp} \to (\mathbf{State} \to \mathbb{Z}),$$

$$\begin{aligned}
\mathcal{A}[\![n]\!]s &:= n \\
\mathcal{A}[\![x]\!]s &:= s(x) \\
\mathcal{A}[\![-A]\!]s &:= -\mathcal{A}[\![A]\!]s \\
\mathcal{A}[\![A_1 + A_2]\!]s &:= \mathcal{A}[\![A_1]\!]s + \mathcal{A}[\![A_2]\!]s \\
\mathcal{A}[\![A_1 - A_2]\!]s &:= \mathcal{A}[\![A_1]\!]s - \mathcal{A}[\![A_2]\!]s \\
&\dots
\end{aligned}$$

Tehtemärgid vasakul ja paremal pool on erinevat tüüpi...

Tõeväärtusavaldisele B ja programmiolekule s seame vastavusse tõeväärtuse $\mathcal{B}[\![B]\!]s$.

Funktsiooni $\mathcal{B}$ tüüp on

$$\mathbf{Bexp} \to (\mathbf{State} \to \mathbb{B})$$

$$\begin{aligned}
\mathcal{B}[\![\mathsf{true}]\!]s &:= \mathsf{true} \\
\mathcal{B}[\![A_1 = A_2]\!]s &:= \begin{cases} \mathsf{true}, & \text{kui } \mathcal{A}[\![A_1]\!]s = \mathcal{A}[\![A_2]\!]s \\ \mathsf{false}, & \text{muidu} \end{cases} \\
\mathcal{B}[\![\neg B]\!]s &:= \neg \mathcal{B}[\![B]\!]s \\
\mathcal{B}[\![B_1 \wedge B_2]\!]s &:= \mathcal{B}[\![B_1]\!]s \wedge \mathcal{B}[\![B_2]\!]s \\
&\cdots
\end{aligned}$$

Defineerimaks, mida teeb mingi programm, kasutame tema *struktuurset operatsioonilist semantikat*.

Kui S on programm ja s programmiolek, siis defineerime seose

$$\langle S, s \rangle \Longrightarrow \ldots$$

siin see $\ldots$ iseloomustab programmi ja tema olekut „ühe sammu pärast“.

Paari $\langle S, s \rangle$ nimetame *konfiguratsiooniks*.

- s — muutujate väärtused praegusel hetkel.
- S — programmiosa, mis veel läbi jooksutada tuleb.

„Üks samm“:

- Aritmeetilise avaldise väärtuse arvutamine ja selle omistamine muutujale.
- Tõeväärtusavaldise väärtuse arvutamine ja otsustamine, mida edasi täita.

$$\langle S, s \rangle \Longrightarrow \ldots$$

Siin ... on

- programmiolek, kui programmi S täitmine vajab üheainsat sammu;
- konfiguratsioon $\langle S', s' \rangle$, kus s' on programmiolek ühe sammu pärast ja S' peale ühte sammu veel täitaolev programm.

Semantika:

$$\langle x := A, s\rangle \Longrightarrow s'$$

kus

$$s'(y) := \begin{cases} s(y), & \text{kui } y \neq x \\ \mathcal{A}[\![A]\!]s, & \text{kui } y = x \end{cases} .$$

Seda s'-i tähistame $s\big[x \mapsto \mathcal{A}[\![A]\!]s\big]$.

$$\langle skip, s\rangle \Longrightarrow s$$

Programmi $S_1; S_2$ semantika defineerime rekursiivselt. Rekursioon on üle süntaksi.

- Kui $\langle S_1, s\rangle \Longrightarrow s'$, siis $\langle S_1; S_2, s\rangle \Longrightarrow \langle S_2, s'\rangle$.
- Kui $\langle S_1, s\rangle \Longrightarrow \langle S_1', s'\rangle$, siis $\langle S_1; S_2, s\rangle \Longrightarrow \langle S_1'; S_2, s'\rangle$.

Selle „kui ..., siis ..." paneme kirja järgmiselt:

$$\frac{\langle S_1, s\rangle \Longrightarrow s'}{\langle S_1; S_2, s\rangle \Longrightarrow \langle S_2, s'\rangle} \qquad \frac{\langle S_1, s\rangle \Longrightarrow \langle S_1', s'\rangle}{\langle S_1; S_2, s\rangle \Longrightarrow \langle S_1'; S_2, s'\rangle}$$

Loeme, et $S_1; S_2; S_3$ tähendab $(S_1; S_2); S_3$.

$$\langle if\ \ B\ \ then\ \ S_1\ \ else\ \ S_2, s\rangle \Longrightarrow \langle S_1, s\rangle \quad \text{kui } \mathcal{B}[\![B]\!]s = \mathsf{true}$$

$$\langle if\ \ B\ \ then\ \ S_1\ \ else\ \ S_2, s\rangle \Longrightarrow \langle S_2, s\rangle \quad \text{kui } \mathcal{B}[\![B]\!]s = \mathsf{false}$$

$$\langle while\ \ B\ \ do\ \ S, s\rangle \Longrightarrow \langle S; while\ \ B\ \ do\ \ S, s\rangle \quad \text{kui } \mathcal{B}[\![B]\!]s = \mathsf{true}$$

$$\langle while\ \ B\ \ do\ \ S, s\rangle \Longrightarrow s \quad \text{kui } \mathcal{B}[\![B]\!]s = \mathsf{false}$$

Näide. Olgu $\mathbf{Var} = \{k, m, n\}$. Vaatame programmi S_F, mis on järgmine:

$$m := 1; k := 1; \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big)$$

Olgu programmi algolek $\{k \mapsto 3, m \mapsto 5, n \mapsto 4\}$.

$$\left\langle \begin{array}{c} m := 1; k := 1; \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 3, m \mapsto 5, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{c} k := 1; \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 3, m \mapsto 1, n \mapsto 4\} \end{array} \right\rangle$$

Tõepoolest, vastavalt $S_1; S_2$ semantika definitsioonile:

$$\frac{\dfrac{\langle m := 1, s\rangle \Longrightarrow s\big[m \mapsto 1\big]}{\langle m := 1; k := 1, s\rangle \Longrightarrow \langle k := 1, s\big[m \mapsto 1\big]\rangle}}{\langle m := 1; k := 1; while\ k \leqslant n\ do\ \big(m := m * k; k := k + 1\big), s\rangle}$$

$$\Longrightarrow$$

$$\langle k := 1; while\ k \leqslant n\ do\ \big(m := m * k; k := k + 1\big), s\big[m \mapsto 1\big]\rangle$$

$$\left\langle \begin{array}{c} k := 1; while\ k \leqslant n\ do\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 3, m \mapsto 1, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{c} while\ k \leqslant n\ do\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 1, m \mapsto 1, n \mapsto 4\} \end{array} \right\rangle$$

$$\left\langle \begin{array}{c} \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 1, m \mapsto 1, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{l} m := m * k; k := k + 1; \\ \qquad\qquad\qquad \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \qquad\qquad\qquad\qquad \{k \mapsto 1, m \mapsto 1, n \mapsto 4\} \end{array} \right\rangle$$

sest $\mathcal{B}[\![k \leqslant n]\!]\{k \mapsto 1, m \mapsto 1, n \mapsto 4\} = \mathsf{true}$.

$$\left\langle \begin{array}{r} m := m * k; k := k + 1; \qquad\qquad\qquad\qquad\qquad\qquad\qquad \\ \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 1, m \mapsto 1, n \mapsto 4\} \qquad\qquad\qquad\qquad \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{c} k := k + 1; \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 1, m \mapsto 1, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{c} \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 2, m \mapsto 1, n \mapsto 4\} \end{array} \right\rangle$$

$$\left\langle \begin{array}{c} \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 2, m \mapsto 1, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{l} m := m * k; k := k + 1; \\ \qquad\qquad\qquad \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \qquad\qquad\qquad\qquad \{k \mapsto 2, m \mapsto 1, n \mapsto 4\} \end{array} \right\rangle$$

sest $\mathcal{B}[\![k \leqslant n]\!]\{k \mapsto 2, m \mapsto 1, n \mapsto 4\} = \mathsf{true}$.

$$\left\langle \begin{array}{r} m := m * k; k := k + 1; \\ while\ k \leqslant n\ do\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 2, m \mapsto 1, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{c} k := k + 1; while\ k \leqslant n\ do\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 2, m \mapsto 2, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{c} while\ k \leqslant n\ do\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 3, m \mapsto 2, n \mapsto 4\} \end{array} \right\rangle$$

$$\left\langle \begin{array}{c} \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 3, m \mapsto 2, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{l} m := m * k; k := k + 1; \\ \qquad\qquad\qquad \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \qquad\qquad\qquad\qquad \{k \mapsto 3, m \mapsto 2, n \mapsto 4\} \end{array} \right\rangle$$

sest $\mathcal{B}[\![k \leqslant n]\!]\{k \mapsto 3, m \mapsto 2, n \mapsto 4\} = \text{true}$.

$$\left\langle \begin{array}{r} m := m * k; k := k + 1; \\ \textit{while } k \leqslant n \textit{ do } \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 3, m \mapsto 2, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{c} k := k + 1; \textit{while } k \leqslant n \textit{ do } \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 3, m \mapsto 6, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{c} \textit{while } k \leqslant n \textit{ do } \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 4, m \mapsto 6, n \mapsto 4\} \end{array} \right\rangle$$

$$\left\langle \begin{array}{c} \textit{while } k \leqslant n \textit{ do } \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 4, m \mapsto 6, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{l} m := m * k; k := k + 1; \\ \qquad\qquad\qquad \textit{while } k \leqslant n \textit{ do } \big(m := m * k; k := k + 1\big) \\ \qquad\qquad\qquad\qquad \{k \mapsto 4, m \mapsto 6, n \mapsto 4\} \end{array} \right\rangle$$

sest $\mathcal{B}[\![k \leqslant n]\!]\{k \mapsto 4, m \mapsto 6, n \mapsto 4\} = \mathsf{true}$.

$$\left\langle \begin{array}{r} m := m * k; k := k + 1; \\ \mathit{while}\ k \leqslant n\ \mathit{do}\ \left(m := m * k; k := k + 1\right) \\ \{k \mapsto 4, m \mapsto 6, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{c} k := k + 1; \mathit{while}\ k \leqslant n\ \mathit{do}\ \left(m := m * k; k := k + 1\right) \\ \{k \mapsto 4, m \mapsto 24, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\left\langle \begin{array}{c} \mathit{while}\ k \leqslant n\ \mathit{do}\ \left(m := m * k; k := k + 1\right) \\ \{k \mapsto 5, m \mapsto 24, n \mapsto 4\} \end{array} \right\rangle$$

$$\left\langle \begin{array}{c} \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 5, m \mapsto 24, n \mapsto 4\} \end{array} \right\rangle$$

$$\Longrightarrow$$

$$\{k \mapsto 5, m \mapsto 24, n \mapsto 4\}$$

sest $\mathcal{B}[\![k \leqslant n]\!]\{k \mapsto 5, m \mapsto 24, n \mapsto 4\} = \mathsf{false}$.

Muuhulgas näitasime, et

$$\left\langle \begin{array}{c} m := 1; k := 1; \mathit{while}\ k \leqslant n\ \mathit{do}\ \big(m := m * k; k := k + 1\big) \\ \{k \mapsto 3, m \mapsto 5, n \mapsto 4\} \end{array} \right\rangle$$

$$\stackrel{*}{\Longrightarrow}$$

$$\{k \mapsto 5, m \mapsto 24, n \mapsto 4\}$$

Siin $\stackrel{*}{\Longrightarrow}$ tähistab seose $\Longrightarrow$ refleksiivset transitiivset sulundit.

$C \stackrel{*}{\Longrightarrow} C'$, kui leidub $n \geqslant 0$ ja $C_0, C_1, \ldots, C_n$ nii, et $C = C_0 \Longrightarrow C_1 \Longrightarrow \cdots \Longrightarrow C_n = C'$.

Programmi S *osalist korrektsust* väljendab üleskirjutus

$$\{P\}S\{Q\},$$

kus P ja Q on predikaadid programmiolekute hulgal. S.t.

$$P, Q : \textbf{State} \to \mathbb{B} \ .$$

See kirjutus tähendab: Iga $s, s' \in \textbf{State}$ jaoks, kus

- $P(s) = \mathsf{true}$,
- $\langle S, s\rangle \stackrel{*}{\Longrightarrow} s'$,

kehtib $Q(s') = \mathsf{true}$.

Näiteks kehtib $\{n \geqslant 0\}S_F\{m = n!\}$.

Samuti kehtib $\{\mathsf{true}\}$*while* true *do skip*$\{\mathsf{false}\}$.

Nimetusi:

- $\{P\}S\{Q\}$ — *Hoare'i kolmik*.
- P — *eeltingimus*.
- Q — *järeltingimus*.

Antud P, S ja Q. Kuidas tõestada, et $\{P\}S\{Q\}$?

Järgnevas tutvustame mõningaid reegleid, mis lubavad $\{P\}S\{Q\}$ tuletada teatavate predikaatarvutuse valemite samaselt tõesusest.

$$\frac{P \Rightarrow P' \quad \{P'\}S\{Q'\} \quad Q' \Rightarrow Q}{\{P\}S\{Q\}}$$

Tõepoolest, olgu $s, s' \in \mathbf{State}$ sellised, et $P(s)$ ja $\langle S, s\rangle \stackrel{*}{\Longrightarrow} s'$. Siit järeldub üksteise järel kohe, et $P'(s)$, $Q'(s')$ ja $Q(s')$.

$$\{P\}skip\{P\}$$

Tõepoolest, kui $\langle skip, s\rangle \overset{*}{\Longrightarrow} s'$, siis $s = s'$.

Olgu Q mingi predikaat programmiolekutel, olgu $x \in \mathbf{Var}$ ja $A \in \mathbf{Aexp}$. Olgu Q^x_A predikaat, mis väärtus leitakse järgmiselt:

1. Leides $Q^x_A(s)$ väärtust, leia kõigepealt $a := \mathcal{A}[\![A]\!]s$.
2. Leia $Q(s\left[x \mapsto a\right])$ väärtus ja tagasta see.

Kui Q on antud mingi valemiga, milles esinevad muutujanimed, konstandid, aritmeetiliste ja loogiliste tehete märgid, siis on Q^x_A antud valemiga, kus Q valemis on kõikjal x asendatud A-ga.

S.t. Q^x_A on Q-st lihtsalt leitav.

$$\{Q^x_A\}x := A\{Q\}$$

Tõepoolest, kui $\langle x := A, s\rangle \stackrel{*}{\Longrightarrow} s'$, siis Q^x_A rakendamine s-le on sama, mis Q rakendamine s'-le.

Näide:

$$\{m * k = n!\}m := m * k\{m = n!\}$$

$$\frac{\{P\}S_1\{R\} \quad \{R\}S_2\{Q\}}{\{P\}S_1;S_2\{Q\}}$$

S.t. järjestikuse täitmise jaoks osalise korrektsuse tõestamisel tuleb meil leida mingi „vahepealne“ predikaat R.

Seda võib püüda leida S_2 struktuurist lähtudes.

$$\frac{\{P(s) \wedge (\mathcal{B}[\![B]\!]s = \mathsf{true})\}S_1\{Q\} \qquad \{P(s) \wedge (\mathcal{B}[\![B]\!]s = \mathsf{false})\}S_2\{Q\}}{\{P\}\mathit{if}\ B\ \mathit{then}\ S_1\ \mathit{else}\ S_2\{Q\}}$$

Tõestus. Olgu s selline programmiolek, et $P(s)$, ja olgu s' selline, et $\langle \mathit{if}\ B\ \mathit{then}\ S_1\ \mathit{else}\ S_2, s\rangle \overset{*}{\Longrightarrow} s'$. Vaatame väärtust $\mathcal{B}[\![B]\!]s$.

Kui $\mathcal{B}[\![B]\!]s = \mathsf{true}$, siis

$$\langle \mathit{if}\ B\ \mathit{then}\ S_1\ \mathit{else}\ S_2, s\rangle \Longrightarrow \langle S_1, s\rangle \overset{*}{\Longrightarrow} s' \quad .$$

Reegli esimese eelduse järgi siis $Q(s')$.

Variant $\mathcal{B}[\![B]\!]s = \mathsf{false}$ on analoogiline.

$$\frac{\{R(s) \wedge (\mathcal{B}[\![B]\!]s = \mathsf{true})\}S\{R\}}{\{R\}\mathit{while}\ B\ \mathit{do}\ S\{R(s) \wedge (\mathcal{B}[\![B]\!]s = \mathsf{false})\}}$$

Predikaat R on *tsükliinvariant*. Ta kehtib tsükli keha alguses.

Väite $\{P\}\mathit{while}\ B\ \mathit{do}\ S\{Q\}$ tõestamiseks on meil lisaks tarvis

$$P \Rightarrow R$$

$$R(s) \wedge (\mathcal{B}[\![B]\!]s = \mathsf{false}) \Rightarrow Q$$

(esimesena toodud reegel)

Sobiva tsükliinvariandi leidmine P-st, Q-st, B-st ja S-st on üldiselt mittelahenduv.

Reegli tõestus. Olgu s selline, et $R(s)$, ning olgu s' selline, et $\langle while\ B\ do\ S, s\rangle \overset{k}{\Longrightarrow} s'$ mingi $k \in \mathbb{N}$ jaoks. Teeme induktsiooni k järgi.

Baas. Kui $k = 1$, siis

$$\langle while\ B\ do\ S, s\rangle \Longrightarrow s'$$

ja seega siis $\mathcal{B}[\![B]\!]s = \mathsf{false}$ ja $s' = s$. Seega olemegi näidanud, et $R(s') \wedge (\mathcal{B}[\![B]\!]s' = \mathsf{false})$.

Samm. Olgu $k > 1$. Siis $\mathcal{B}[\![B]\!]s = \mathsf{true}$ ja

$$\langle \mathit{while}\ B\ \mathit{do}\ S, s\rangle \Longrightarrow \langle S; \mathit{while}\ B\ \mathit{do}\ S, s\rangle \overset{k-1}{\Longrightarrow} s' \ .$$

Peavad leiduma selline s'' ning sellised i, j, et

$$\begin{aligned} \langle S, s\rangle &\overset{i}{\Longrightarrow} s'' \\ \langle \mathit{while}\ B\ \mathit{do}\ S, s''\rangle &\overset{j}{\Longrightarrow} s' \\ i + j &= k - 1 \ . \end{aligned}$$

Reegli eeldusest siis $R(s'')$ ning induktsiooni eeldusest $R(s') \wedge (\mathcal{B}[\![B]\!]s' = \mathsf{false})$.

Tähistagu üleskirjutus

$$\{P\}S \downarrow$$

seda, et iga $s \in \mathbf{State}$ jaoks, kus $P(s)$, leidub selline $s' \in \mathbf{State}$, et $\langle S, s\rangle \overset{*}{\Longrightarrow} s'$.

S.t. kui s rahuldab tingimust P, siis S, rakendatuna olekule s, termineerub.

Reeglid, mis peaksid olema ilmsed:

$$\frac{P \Rightarrow P' \quad \{P'\}S\downarrow}{\{P\}S\downarrow} \qquad \{\mathsf{true}\}x := A\downarrow \qquad \{\mathsf{true}\}\mathit{skip}\downarrow$$

$$\frac{\begin{array}{c}\{P\}S_1\downarrow \\ \{P\}S_1\{R\} \\ \{R\}S_2\downarrow\end{array}}{\{P\}S_1;S_2\downarrow} \qquad \frac{\begin{array}{c}\{P(s)\wedge(\mathcal{B}[\![B]\!]s = \mathsf{true})\}S_1\downarrow \\ \{P(s)\wedge(\mathcal{B}[\![B]\!]s = \mathsf{false})\}S_2\downarrow\end{array}}{\{P\}\mathit{if}\ \ B\ \ \mathit{then}\ \ S_1\ \ \mathit{else}\ \ S_2\downarrow}$$

Vaatame programmi $while\ B\ do\ S$. Olgu s programmi algolek. On kolm võimalust.

- Programm termineerub.
- Programm ei termineeru, sest iteratsioone on lõpmata palju.
- Programm ei termineeru, sest mingil iteratsioonil S ei termineeru.

while B *do* S termineerumise uurimisel võtame S-i termineerumise eelduseks.

Iteratsioonide arvu lõplikkuse näitamiseks, on meil tarvis mingit funktsiooni E, mis

- seab igale programmiolekule s vastavusse naturaalarvu $E(s)$;
  - S.t. $E(s)$-l leidub vähim võimalik väärtus.
- väheneb igal iteratsioonil.
  - ... ja vähemalt 1 võrra, sest on naturaalarv.

$$\frac{\{R(s) \wedge (\mathcal{B}[\![B]\!]s = \mathsf{true})\}S \downarrow \qquad \{R(s) \wedge (\mathcal{B}[\![B]\!]s = \mathsf{true}) \wedge (e = E(s))\}S\{R(s) \wedge (E(s) < e)\}}{\{R\}\mathit{while}\ B\ \mathit{do}\ S \downarrow}$$

Siin R on jälle tsükliinvariant.

Näide:

$$a \geqslant 1 \wedge b \geqslant 0 \wedge a_0 = a \wedge b_0 = b$$

$\mathit{if}\ \ a < b\ \ \mathit{then}$

$\quad t := a$

$\quad a := b$

$\quad b := t$

$\mathit{while}\ b > 0\ \ \mathit{do}$

$\quad t := a$

$\quad a := b$

$\quad b := t \bmod a$

$r := a$

$$r = \gcd(a_0, b_0)$$

Vajame tsükliinvarianti:

$a \geqslant 1 \wedge b \geqslant 0 \wedge a_0 = a \wedge b_0 = b$

$\mathit{if}\ a < b\ \mathit{then}$

$\quad t := a$

$\quad a := b$

$\quad b := t$

$\mathrm{gcd}(a, b) = \mathrm{gcd}(a_0, b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b$

$\mathit{while}\ b > 0\ \mathit{do}$

$\quad t := a$

$\quad a := b$

$\quad b := t \bmod a$

$r := a$

$r = \mathrm{gcd}(a_0, b_0)$

9. rea nõrgim eeltingimus:

$$a \geqslant 1 \wedge b \geqslant 0 \wedge a_0 = a \wedge b_0 = b$$

1 $\quad$ *if* $a < b$ *then*

2 $\qquad t := a$

3 $\qquad a := b$

4 $\qquad b := t$

$$\gcd(a, b) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b$$

5 $\quad$ *while* $b > 0$ *do*

6 $\qquad t := a$

7 $\qquad a := b$

8 $\qquad b := t \bmod a$

$$a = \gcd(a_0, b_0)$$

9 $\quad r := a$

$$r = \gcd(a_0, b_0)$$

$$\frac{\{R(s) \land (\mathcal{B}[\![B]\!]s = \mathsf{true})\}S\{R\}}{\{R\}\mathit{while}\ B\ \mathit{do}\ S\{R(s) \land (\mathcal{B}[\![B]\!]s = \mathsf{false})\}}$$

Tsükli korrektsus:

Invariant R oli $\gcd(a, b) = \gcd(a_0, b_0) \land a \geqslant 1 \land b \geqslant 0 \land a \geqslant b$.

Me peame näitama, et

$R \land b > 0$

6 $t := a$

7 $a := b$

8 $b := t \bmod a$

R

ning $R \land b \leqslant 0 \Rightarrow a = \gcd(a_0, b_0)$.

$$\gcd(a,b) = \gcd(a_0,b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge b > 0 \wedge a \geqslant b$$

$t := a$

$a := b$

$b := t \bmod a$ $\quad$ $\gcd(a,b) = \gcd(a_0,b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b$

$$\gcd(a,b) = \gcd(a_0,b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge b > 0 \wedge a \geqslant b$$

$t := a$

$a := b$

$$\gcd(a, t \bmod a) = \gcd(a_0,b_0) \wedge a \geqslant 1 \wedge t \bmod a \geqslant 0 \wedge a \geqslant t \bmod a$$

$b := t \bmod a$

$$\gcd(a,b) = \gcd(a_0,b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b$$

- Kui $b \geqslant 0$ ja $b > 0$, siis $b \geqslant 1$. Kui $a \geqslant b$ ja $b \geqslant 1$, siis $a \geqslant 1$.
- Kui $a \geqslant 1$, siis $t \bmod a \geqslant 0$ ja $a > t \bmod a$.

Lihtsustame:

$$\gcd(a, b) = \gcd(a_0, b_0) \land a \geqslant b \land b \geqslant 1$$

$$t := a$$

$$a := b$$

$$\gcd(a, t \bmod a) = \gcd(a_0, b_0) \land a \geqslant 1$$

$$b := t \bmod a$$

$$\gcd(a, b) = \gcd(a_0, b_0) \land a \geqslant 1 \land b \geqslant 0 \land a \geqslant b$$

$$\gcd(a, b) = \gcd(a_0, b_0) \wedge a \geqslant b \wedge b \geqslant 1$$

$$t := a$$

$$\gcd(b, t \bmod b) = \gcd(a_0, b_0) \wedge b \geqslant 1$$

$$a := b$$

$$\gcd(a, t \bmod a) = \gcd(a_0, b_0) \wedge a \geqslant 1$$

$$b := t \bmod a \qquad \gcd(a, b) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b$$

$$\gcd(a,b) = \gcd(a_0,b_0) \wedge a \geqslant b \wedge b \geqslant 1$$

$$\gcd(b, a \bmod b) = \gcd(a_0,b_0) \wedge b \geqslant 1$$

$t := a$

$$\gcd(b, t \bmod b) = \gcd(a_0,b_0) \wedge b \geqslant 1$$

$a := b$

$$\gcd(a, t \bmod a) = \gcd(a_0,b_0) \wedge a \geqslant 1$$

$b := t \bmod a$

$$\gcd(a,b) = \gcd(a_0,b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b$$

Meil tuleb näidata, et punasest järeldub sinine.

Kui $b \geqslant 1$, siis $\gcd(b, a \bmod b) = \gcd(a, b)$. Tõepoolest:

- Kui $x \mid b$ ja $x \mid a \bmod b$, siis ka

$$x \mid (a \bmod b + \lfloor a/b \rfloor b) = a \quad .$$

- Kui $x \mid a$ ja $x \mid b$, siis ka

$$x \mid (a - \lfloor a/b \rfloor b) = a \bmod b \quad .$$

S.t. a ja b ühistegurite hulk on võrdne b ja $a \bmod b$ ühistegurite hulgaga.

Meil tuli veel näidata, et $R \wedge b < 0 \Rightarrow a = \mathrm{gcd}(a_0, b_0)$.

S.t. väitest

$$\mathrm{gcd}(a, b) = \mathrm{gcd}(a_0, b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b \wedge b \leqslant 0$$

järeldub $a = \mathrm{gcd}(a_0, b_0)$.

Kuna $b \geqslant 0$ ja $b \leqslant 0$, siis $b = 0$. $\mathrm{gcd}(a, 0) = a$, kui $a \geqslant 1$.

$a \geqslant 1 \land b \geqslant 0 \land a_0 = a \land b_0 = b$

if $a < b$ *then*
    $t := a$
    $a := b$
    $b := t$

$\gcd(a, b) = \gcd(a_0, b_0) \land a \geqslant 1 \land b \geqslant 0 \land a \geqslant b$

while $b > 0$ *do*
    $t := a$
    $a := b$
    $b := t \bmod a$

$a = \gcd(a_0, b_0)$

$r := a$

$r = \gcd(a_0, b_0)$

Vaatame *if*-lauset ridades 1–4. Vaatame kõigepealt tema *else*-haru, selleks on *skip*.

Vaja on

$$\begin{array}{ll} & a \geqslant 1 \wedge b \geqslant 0 \wedge a_0 = a \wedge b_0 = b \wedge a \geqslant b \\ \textit{skip} & \\ & \gcd(a, b) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b \end{array}$$

See kehtib.

Vaatame *then*-haru.

$$a \geqslant 1 \land b \geqslant 0 \land a_0 = a \land b_0 = b \land a < b$$

2 $t := a$

3 $a := b$

4 $b := t$

$$\gcd(a, b) = \gcd(a_0, b_0) \land a \geqslant 1 \land b \geqslant 0 \land a \geqslant b$$

Vaatame *then*-haru.

$$a \geqslant 1 \wedge b \geqslant 0 \wedge a_0 = a \wedge b_0 = b \wedge a < b$$

2 $t := a$

3 $a := b$

4 $b := t$

$$\gcd(a, t) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge t \geqslant 0 \wedge a \geqslant t$$

$$\gcd(a, b) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b$$

Vaatame *then*-haru.

$$
\begin{array}{ll}
 & a \geqslant 1 \wedge b \geqslant 0 \wedge a_0 = a \wedge b_0 = b \wedge a < b \\
t := a & \\
 & \gcd(b, t) = \gcd(a_0, b_0) \wedge b \geqslant 1 \wedge t \geqslant 0 \wedge b \geqslant t \\
a := b & \\
 & \gcd(a, t) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge t \geqslant 0 \wedge a \geqslant t \\
b := t & \\
 & \gcd(a, b) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b
\end{array}
$$

Vaatame *then*-haru.

$$a \geqslant 1 \wedge b \geqslant 0 \wedge a_0 = a \wedge b_0 = b \wedge a < b$$

$$\mathrm{gcd}(b, a) = \mathrm{gcd}(a_0, b_0) \wedge b \geqslant 1 \wedge a \geqslant 0 \wedge b \geqslant a$$

2 $t := a$

$$\mathrm{gcd}(b, t) = \mathrm{gcd}(a_0, b_0) \wedge b \geqslant 1 \wedge t \geqslant 0 \wedge b \geqslant t$$

3 $a := b$

$$\mathrm{gcd}(a, t) = \mathrm{gcd}(a_0, b_0) \wedge a \geqslant 1 \wedge t \geqslant 0 \wedge a \geqslant t$$

4 $b := t$

$$\mathrm{gcd}(a, b) = \mathrm{gcd}(a_0, b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b$$

On ilmne, et sinine järeldub punasest.

$a \geqslant 1 \wedge b \geqslant 0 \wedge a_0 = a \wedge b_0 = b$

if $a < b$ *then*

  $t := a$

  $a := b$

  $b := t$

$\gcd(a, b) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b$

while $b > 0$ *do*

  $t := a$

  $a := b$

  $b := t \bmod a$

$a = \gcd(a_0, b_0)$

$r := a$

$r = \gcd(a_0, b_0)$

Oleme näidanud algoritmi osalise korrektsuse.

Täieliku korrektsuse näitamiseks tuleks näidata, et tsükkel termineerub.

$$\frac{\{R(s) \wedge (\mathcal{B}[\![B]\!]s = \mathsf{true})\}S \downarrow \qquad \{R(s) \wedge (\mathcal{B}[\![B]\!]s = \mathsf{true}) \wedge (e = E(s))\}S\{R(s) \wedge (E(s) < e)\}}{\{R\} while \; B \; do \; S \downarrow}$$

Olgu $E(s) = \max(s(b), 0)$, s.t. E-ks võtame muutuja b väärtuse.

On ilmne, et reegli esimene eeldus on täidetud. Näitame, et ka teine on täidetud.

$$\gcd(a,b)=\gcd(a_0,b_0)\land a\geqslant 1\land b\geqslant 0\land b>0\land a\geqslant b\land e=\max(b,0)$$

$t := a$

$a := b$

$b := t \bmod a$ $\quad \gcd(a,b)=\gcd(a_0,b_0)\land a\geqslant 1\land b\geqslant 0\land a\geqslant b\land \max(b,0)<e$

	$\gcd(a, b) = \gcd(a_0, b_0) \wedge a \geqslant b \wedge b \geqslant 1 \wedge e = \max(b, 0)$
$t := a$	
$a := b$	$\gcd(a, t \bmod a) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge t \bmod a \geqslant 0 \wedge a \geqslant t \bmod a \wedge \max(t \bmod a, 0) < e$
$b := t \bmod a$	$\gcd(a, b) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b \wedge \max(b, 0) < e$

Lihtsustame...

$$\mathrm{gcd}(a,b) = \mathrm{gcd}(a_0,b_0) \wedge a \geqslant b \wedge b \geqslant 1 \wedge e = b$$

6 $t := a$

7 $a := b$ $\quad \mathrm{gcd}(a, t \bmod a) = \mathrm{gcd}(a_0,b_0) \wedge a \geqslant 1 \wedge t \bmod a < e$

8 $b := t \bmod a$ $\quad \mathrm{gcd}(a,b) = \mathrm{gcd}(a_0,b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b \wedge b < e$

	$\gcd(a,b)=\gcd(a_0,b_0)\wedge a\geqslant b\wedge b\geqslant 1\wedge e=b$
$t := a$	$\gcd(b,t \bmod b)=\gcd(a_0,b_0)\wedge b\geqslant 1\wedge t \bmod b < e$
$a := b$	$\gcd(a,t \bmod a)=\gcd(a_0,b_0)\wedge a\geqslant 1\wedge t \bmod a < e$
$b := t \bmod a$	$\gcd(a,b)=\gcd(a_0,b_0)\wedge a\geqslant 1\wedge b\geqslant 0\wedge a\geqslant b\wedge b < e$

	$\gcd(a, b) = \gcd(a_0, b_0) \wedge a \geqslant b \wedge b \geqslant 1 \wedge e = b$
	$\gcd(b, a \bmod b) = \gcd(a_0, b_0) \wedge b \geqslant 1 \wedge a \bmod b < e$
$t := a$	$\gcd(b, t \bmod b) = \gcd(a_0, b_0) \wedge b \geqslant 1 \wedge t \bmod b < e$
$a := b$	$\gcd(a, t \bmod a) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge t \bmod a < e$
$b := t \bmod a$	$\gcd(a, b) = \gcd(a_0, b_0) \wedge a \geqslant 1 \wedge b \geqslant 0 \wedge a \geqslant b \wedge b < e$

Punasest järeldub sinine:

- $\gcd(b, a \bmod b) = \gcd(a_0, b_0)$ ja $b \geqslant 1$ me juba näitasime.
- $a \bmod b < b$, sest jagamise jääk on alati väiksem kui jagaja.

Oleme näidanud, et tsükkel termineerub. Programmi ülejäänud osade termineeruvuses veendumine on triviaalne.