Realisatsioon koos korrektsuse tõestusega

Transleerimine intuitiivselt

- **Transleerimisel** koostatakse lähtekeelsele programmile vastav sihtkeelne programm.
- Transleerimine on **korrektne**, kui transleerimisel programmi “tähendus säilib”.

Transleerimine täpsemalt

- Olgu antud lähtekeel L (nii süntaks kui semantika).

  Kuidas spetsifitseeritakse transleerimine?

  - Defineeritakse sihtkeel M (nii süntaks kui semantika).
  - Defineeritakse transleeriv funktsioon $\mathcal{T}$ lähtekeele programmide hulgast sihtkeele programmide hulka.
  - Defineeritakse vastavusrelatsioon $\approx$ lähtekeele ja sihtkeele semantiliste objektide vahel (mida tähendab "tähenduse säilimine").
    - See relatsioon peab vastama intuitsioonile tähenduse säilimisest. Lihtsalt "et asi välja tuleks" pole eriti veenev.
    - Võrdus üldiselt ei sobi, kuna transleerimisel võib programmeerimiskeele abstraktsustaseme langusest tulenevalt infot juurde tekkida.
  - Transleerimine on korrektne, kui

    lähtekeele iga programmi p korral $\mathcal{S}_L [\![p]\!] \approx \mathcal{S}_M [\![\mathcal{T} [\![p]\!]]\!]$.

Ülevaade järgnevast

- Oleme näidanud **motivatsiooni** programmeerimiskeelte formaalsete semantikate uurimiseks.
  - Reaalsete translaatorite, kompilaatorite, interpretaatorite korrektsuse käsitlemine.
- Järgnevas on meie **eesmärk** kirjutada translaator **While**-keelest masinkeelde ja tõestada tema korrektsus.
  - Teoreetilistes targutustes kasutame **abstraktset masinat**.

Abstraktne masin

Meie abstraktne masin

AM — meie käsitletav väärtustusmagasiniga assemblerilaadne näidismasinkeel.


Abstraktne süntaks

Süntaktilised kategooriad

$n \in \mathbf{Num}$	täisarvkonstandid
$t \in \mathbf{TVal}$	tõeväärtuskonstandid
$x \in \mathbf{Var}$	muutujad
$i \in \mathbf{Inst}$	instruktsioonid e. üksikkäsud
$c \in \mathbf{Code}$	kood

Abstraktne süntaks

- BNF-na:

$$
\begin{array}{rcl}
n & & \text{nagu } \textbf{While}\text{'is} \\
t & ::= & \texttt{true} \mid \texttt{false} \\
x & & \text{nagu } \textbf{While}\text{'is} \\
i & ::= & \text{PUSHN-}n \mid \text{PUSHT-}t \mid \text{FETCH-}x \mid \text{STORE-}x \\
 & \mid & \text{ADD} \mid \text{SUB} \mid \text{MULT} \mid \text{EQ} \mid \text{LE} \mid \text{AND} \mid \text{NEG} \\
 & \mid & \text{NOOP} \mid \text{BRANCH } c\ c \mid \text{LOOP } c \\
c & ::= & \varepsilon \mid i \mid c\,;c
\end{array}
$$

- Nõuame, et
  - $\forall c \in \mathbf{Code}.\ \varepsilon\,;c = c = c\,;\varepsilon$,
  - $\forall c_1, c_2, c_3 \in \mathbf{Code}.\ c_1\,;(c_2\,;c_3) = (c_1\,;c_2)\,;c_3$.
- Sisuliselt $c = i^*$, kus ε on tühisõne ja ; on konkatenatsioon.

Näited

- PUSHN-1 ; FETCH-x ; ADD ; STORE-x
- PUSHT-`true` ; LOOP (NOOP ; PUSHT-`true`)
- PUSHN-1;STORE-y;PUSHN-1;FETCH-x;EQ;NEG;LOOP (FETCH-x;FETCH-y;MULT;STORE-y;PUSHN-1;FETCH-x;SUB;STORE-x;PUSHN-1;FETCH-x;EQ;NEG)
- PUSHN-0;STORE-z;FETCH-x;STORE-r;FETCH-r;FETCH-y;LE;LOOP (FETCH-y;FETCH-r;SUB;STORE-r;PUSHN-1;FETCH-z;ADD;STORE-z;FETCH-r;FETCH-y;LE)

Mida need programmid teevad?

Vastus küsimusele

Õige vastus: **ei tea**, sest semantika pole veel defineeritud!

Jälitussemantika

Väärtustusmagasin

- $\mathbf{Stack} = (\mathbf{Z} \cup \mathbf{T})^*$ — **väärtustusmagasini** seisude semantiline kategooria.
  - Täisarvude ja tõeväärtuste järjendid, mida käsitleme listidena.
- Olgu operaator : pea-saba eraldaja (st mitte konkatenatsioon!).

Konfiguratsioonid

- **Konfiguratsioon** on kolmik $\langle c\ ,\ (e\,,s)\rangle$, kus
  - $c \in \mathbf{Code}$ on koodi jääk (st kood, mida hakatakse väärtustama),
  - $e \in \mathbf{Stack}$ on väärtustusmagasin,
  - $s \in \mathbf{State}$ on olek nagu **While**-keele semantikaski.
- Konfiguratsioon $\langle c\ ,\ (e\,,s)\rangle$ on **normaalne lõppkonfiguratsioon**, kui $c = \varepsilon$.
- Tähistame $\mathbf{Conf} = \mathbf{Code} \times (\mathbf{Stack} \times \mathbf{State})$.
  - St $\mathbf{Conf}$ on kõigi konfiguratsioonide hulk.

Üleminekurelatsioon 1

$$\triangleright : \mathbf{Conf} \rightsquigarrow \mathbf{Conf},$$

st $\triangleright$ on binaarne relatsioon hulgal $\mathbf{Conf}$.

Üleminekurelatsioon 2

- Salvestuskäsud:

$$\langle \text{PUSHN-}n\,;c\,,\,(e\,,s)\rangle \quad \rhd \quad \langle c\,,\,(\mathcal{N}\,[\![n]\!]:e\,,s)\rangle$$

$$\langle \text{PUSHT-}t\,;c\,,\,(e\,,s)\rangle \quad \rhd \quad \langle c\,,\,(\mathcal{B}\,[\![t]\!]:e\,,s)\rangle$$

$$\langle \text{FETCH-}x\,;c\,,\,(e\,,s)\rangle \quad \rhd \quad \langle c\,,\,(s\;x:e\,,s)\rangle$$

$$\langle \text{STORE-}x\,;c\,,\,(z:e\,,s)\rangle \quad \rhd \quad \langle c\,,\,(e\,,s[x\mapsto z])\rangle \quad \Leftarrow \; z\in\mathbf{Z}$$

Üleminekurelatsioon 3

- Arvutuskäsud:

$$
\begin{array}{lcll}
\langle \text{ADD} ; c\ ,\ (z_1 : z_2 : e\ , s)\rangle & \rhd & \langle c\ ,\ (z_1 + z_2 : e\ , s)\rangle & \Leftarrow\ z_1, z_2 \in \mathbf{Z} \\
\langle \text{SUB} ; c\ ,\ (z_1 : z_2 : e\ , s)\rangle & \rhd & \langle c\ ,\ (z_1 - z_2 : e\ , s)\rangle & \Leftarrow\ z_1, z_2 \in \mathbf{Z} \\
\langle \text{MULT} ; c\ ,\ (z_1 : z_2 : e\ , s)\rangle & \rhd & \langle c\ ,\ (z_1 z_2 : e\ , s)\rangle & \Leftarrow\ z_1, z_2 \in \mathbf{Z} \\
\langle \text{EQ} ; c\ ,\ (z_1 : z_2 : e\ , s)\rangle & \rhd & \langle c\ ,\ (z_1 = z_2 : e\ , s)\rangle & \Leftarrow\ z_1, z_2 \in \mathbf{Z} \\
\langle \text{LE} ; c\ ,\ (z_1 : z_2 : e\ , s)\rangle & \rhd & \langle c\ ,\ (z_1 \leqslant z_2 : e\ , s)\rangle & \Leftarrow\ z_1, z_2 \in \mathbf{Z} \\
\langle \text{AND} ; c\ ,\ (t_1 : t_2 : e\ , s)\rangle & \rhd & \langle c\ ,\ (t_1 \wedge t_2 : e\ , s)\rangle & \Leftarrow\ t_1, t_2 \in \mathbf{T} \\
\langle \text{NEG} ; c\ ,\ (t : e\ , s)\rangle & \rhd & \langle c\ ,\ (\neg t : e\ , s)\rangle & \Leftarrow\ t \in \mathbf{T}
\end{array}
$$

Üleminekurelatsioon 4

- Ülejäänud käsud:

$$\begin{aligned}
&\langle \text{NOOP} ; c\ ,\ (e\,,s)\rangle && \rhd \langle c\ ,\ (e\,,s)\rangle\\
&\langle \text{BRANCH}\ \ c_1\ \ c_2 ; c\ ,\ (\mathbf{tt} : e\,,s)\rangle && \rhd \langle c_1 : c\ ,\ (e\,,s)\rangle\\
&\langle \text{BRANCH}\ \ c_1\ \ c_2 ; c\ ,\ (\mathbf{ff} : e\,,s)\rangle && \rhd \langle c_2 : c\ ,\ (e\,,s)\rangle\\
&\langle \text{LOOP}\ \ c_1 ; c\ ,\ (\mathbf{tt} : e\,,s)\rangle && \rhd \langle c_1 ; \text{LOOP}\ \ c_1 ; c\ ,\ (e\,,s)\rangle\\
&\langle \text{LOOP}\ \ c_1 ; c\ ,\ (\mathbf{ff} : e\,,s)\rangle && \rhd \langle c\ ,\ (e\,,s)\rangle
\end{aligned}$$

Näited 1

$$\begin{aligned}
&\langle \text{PUSHN-}1 \text{ ; FETCH-}x \text{ ; ADD ; STORE-}x \text{ , } (\varepsilon \text{ , } [x \mapsto 3]) \rangle \\
\rhd\ &\langle \text{FETCH-}x \text{ ; ADD ; STORE-}x \text{ , } (1 : \varepsilon \text{ , } [x \mapsto 3]) \rangle \\
\rhd\ &\langle \text{ADD ; STORE-}x \text{ , } (3 : 1 : \varepsilon \text{ , } [x \mapsto 3]) \rangle \\
\rhd\ &\langle \text{STORE-}x \text{ , } (4 : \varepsilon \text{ , } [x \mapsto 3]) \rangle \\
\rhd\ &\langle \varepsilon \text{ , } (\varepsilon \text{ , } [x \mapsto 4]) \rangle
\end{aligned}$$

Näited 2

$\langle$PUSHT-`true` ; LOOP $($NOOP ; PUSHT-`true`$)$, $(e\,,s)\rangle$

$\triangleright$ $\langle$LOOP $($NOOP ; PUSHT-`true`$)$, $(\mathbf{tt} : e\,,s)\rangle$

$\triangleright$ $\langle$NOOP ; PUSHT-`true` ; LOOP $($NOOP ; PUSHT-`true`$)$, $(e\,,s)\rangle$

$\triangleright$ $\langle$PUSHT-`true` ; LOOP $($NOOP ; PUSHT-`true`$)$, $(e\,,s)\rangle$

$\triangleright$ ...

Näited 3

Mida teevad järgmised programmid?

- PUSHN-1;STORE-y;PUSHN-1;FETCH-x;EQ;NEG;LOOP (FETCH-x;FETCH-y; MULT ; STORE-y ; PUSHN-1 ; FETCH-x ; SUB ; STORE-x ; PUSHN-1 ; FETCH-x ; EQ ; NEG)
- PUSHN-0 ; STORE-z ; FETCH-x ; STORE-r ; FETCH-r ; FETCH-y ; LE ; LOOP (FETCH-y ; FETCH-r ; SUB ; STORE-r ; PUSHN-1 ; FETCH-z ; ADD ; STORE-z ; FETCH-r ; FETCH-y ; LE)

Märkusi

- **While** ja **AM** defineerimismetoodikates olid mõned erinevused:
  - **AM** abstraktses süntaksis tõime sisse universaalvõrdused,
  - see võimaldas **AM** semantika defineerimisel reegliteta hakkama saada.
- Tulemuses aga põhimõttelist vahet pole.
- Mõlemat metoodikat saab kasutada nii **AM** kui **While** korral.

Transleerimine keelest While keelde AM

Transleerivad funktsioonid

Defineeritakse transleerivad funktsioonid vastavalt süntaktilistele kategooriatele:

- $\mathcal{CN} : \mathbf{Num} \to \mathbf{Num}$,
- $\mathcal{CV} : \mathbf{Var} \to \mathbf{Var}$,
- $\mathcal{CA} : \mathbf{AExp} \to \mathbf{Code}$,
- $\mathcal{CB} : \mathbf{BExp} \to \mathbf{Code}$,
- $\mathcal{CS} : \mathbf{Stmt} \to \mathbf{Code}$.

Kategooriad Num ja Var

Kategooriad **Num** ja **Var** on ühised:

$$\mathcal{CN}[\![n]\!] = n$$
$$\mathcal{CV}[\![x]\!] = x$$

Kategooria AExp

$$
\begin{aligned}
\mathcal{CA}[\![n]\!] &= \text{PUSHN-}\mathcal{CN}[\![n]\!] \\
\mathcal{CA}[\![x]\!] &= \text{FETCH-}\mathcal{CV}[\![x]\!] \\
\mathcal{CA}[\![a_1 + a_2]\!] &= \mathcal{CA}[\![a_2]\!] ; \mathcal{CA}[\![a_1]\!] ; \text{ADD} \\
\mathcal{CA}[\![a_1 - a_2]\!] &= \mathcal{CA}[\![a_2]\!] ; \mathcal{CA}[\![a_1]\!] ; \text{SUB} \\
\mathcal{CA}[\![a_1 * a_2]\!] &= \mathcal{CA}[\![a_2]\!] ; \mathcal{CA}[\![a_1]\!] ; \text{MULT}
\end{aligned}
$$

Kategooria BExp

$$
\begin{aligned}
\mathcal{CB}[\![t]\!] &= \text{PUSHT-}t \\
\mathcal{CB}[\![a_1 = a_2]\!] &= \mathcal{CA}[\![a_2]\!]\,;\mathcal{CA}[\![a_1]\!]\,;\text{EQ} \\
\mathcal{CB}[\![a_1 \leq a_2]\!] &= \mathcal{CA}[\![a_2]\!]\,;\mathcal{CA}[\![a_1]\!]\,;\text{LE} \\
\mathcal{CB}[\![b_1 \wedge b_2]\!] &= \mathcal{CB}[\![b_2]\!]\,;\mathcal{CB}[\![b_1]\!]\,;\text{AND} \\
\mathcal{CB}[\![\neg b]\!] &= \mathcal{CB}[\![b]\!]\,;\text{NEG}
\end{aligned}
$$

Kategooria Stmt

$$
\begin{aligned}
\mathcal{CS}[\![x := a]\!] &= \mathcal{CA}[\![a]\!] \,;\, \text{STORE-}\mathcal{CV}[\![x]\!] \\
\mathcal{CS}[\![\texttt{skip}]\!] &= \text{NOOP} \\
\mathcal{CS}[\![S_1 \,;\, S_2]\!] &= \mathcal{CS}[\![S_1]\!] \,;\, \mathcal{CS}[\![S_2]\!] \\
\mathcal{CS}[\![\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2]\!] &= \mathcal{CB}[\![b]\!] \,;\, \text{BRANCH } (\mathcal{CS}[\![S_1]\!]) \ (\mathcal{CS}[\![S_2]\!]) \\
\mathcal{CS}[\![\texttt{while } b \texttt{ do } S]\!] &= \mathcal{CB}[\![b]\!] \,;\, \text{LOOP } (\mathcal{CS}[\![S]\!] \,;\, \mathcal{CB}[\![b]\!])
\end{aligned}
$$

Näited 1

$$\begin{aligned}\mathcal{CS}[\![x := x + 1]\!] &= \mathcal{CA}[\![x + 1]\!] \,;\, \text{STORE-}\mathcal{CV}[\![x]\!] \\ &= \mathcal{CA}[\![1]\!] \,;\, \mathcal{CA}[\![x]\!] \,;\, \text{ADD} \,;\, \text{STORE-}x \\ &= \text{PUSHN-}\mathcal{CN}[\![1]\!] \,;\, \text{FETCH-}\mathcal{CV}[\![x]\!] \,;\, \text{ADD} \,;\, \text{STORE-}x \\ &= \text{PUSHN-}1 \,;\, \text{FETCH-}x \,;\, \text{ADD} \,;\, \text{STORE-}x\end{aligned}$$

Näited 2

$$\begin{aligned} & \mathcal{CS}[\![\texttt{while true do skip}]\!] \\ = \; & \mathcal{CB}[\![\texttt{true}]\!] ; \text{LOOP} \;\; (\mathcal{CS}[\![\texttt{skip}]\!] ; \mathcal{CB}[\![\texttt{true}]\!]) \\ = \; & \text{PUSHT-}\texttt{true} ; \text{LOOP} \;\; (\text{NOOP} ; \text{PUSHT-}\texttt{true}) \end{aligned}$$

Näited 3

$$
\begin{aligned}
& \mathcal{CS}[\![y := 1; \texttt{while } \neg(x = 1) \texttt{ do } (y := y * x; x := x - 1)]\!] \\
= \; & \mathcal{CS}[\![y := 1]\!] ; \mathcal{CS}[\![\texttt{while } \neg(x = 1) \texttt{ do } (y := y * x; x := x - 1)]\!] \\
= \; & \mathcal{CA}[\![1]\!] ; \text{STORE-}\mathcal{CV}\,y ; \mathcal{CB}[\![\neg(x = 1)]\!] ; \\
& \text{LOOP } (\mathcal{CS}[\![y := y * x; x := x - 1]\!] ; \mathcal{CB}[\![\neg(x = 1)]\!]) \\
= \; & \text{PUSHN-}\mathcal{CN}[\![1]\!] ; \text{STORE-}y ; \mathcal{CB}[\![x = 1]\!] ; \text{NEG} ; \\
& \text{LOOP } (\mathcal{CS}[\![y := y * x]\!] ; \mathcal{CS}[\![x := x - 1]\!] ; \mathcal{CB}[\![x = 1]\!] ; \text{NEG}) \\
= \; & \text{PUSHN-}\mathcal{CN}[\![1]\!] ; \text{STORE-}y ; \mathcal{CA}[\![1]\!] ; \mathcal{CA}[\![x]\!] ; \text{EQ} ; \text{NEG} ; \\
& \text{LOOP } (\mathcal{CA}[\![y * x]\!] ; \text{STORE-}\mathcal{CV}[\![y]\!] ; \mathcal{CA}[\![x - 1]\!] ; \text{STORE-}\mathcal{CV}[\![x]\!] ; \\
& \mathcal{CA}[\![1]\!] ; \mathcal{CA}[\![x]\!] ; \text{EQ} ; \text{NEG}) \\
= \; & \text{PUSHN-}1 ; \text{STORE-}y ; \text{PUSHN-}\mathcal{CN}[\![1]\!] ; \text{FETCH-}\mathcal{CV}[\![x]\!] ; \text{EQ} ; \text{NEG} ; \\
& \text{LOOP } (\mathcal{CA}[\![x]\!] ; \mathcal{CA}[\![y]\!] ; \text{MULT} ; \text{STORE-}y ; \\
& \mathcal{CA}[\![1]\!] ; \mathcal{CA}[\![x]\!] ; \text{SUB} ; \text{STORE-}x ; \\
& \text{PUSHN-}\mathcal{CN}[\![1]\!] ; \text{FETCH-}\mathcal{CV}[\![x]\!] ; \text{EQ} ; \text{NEG}) \\
= \; & \text{PUSHN-}1 ; \text{STORE-}y ; \text{PUSHN-}1 ; \text{FETCH-}x ; \text{EQ} ; \text{NEG} ; \\
& \text{LOOP } (\text{FETCH-}\mathcal{CV}[\![x]\!] ; \text{FETCH-}\mathcal{CV}[\![y]\!] ; \text{MULT} ; \text{STORE-}y ; \\
& \text{PUSHN-}\mathcal{CN}[\![1]\!] ; \text{FETCH-}\mathcal{CV}[\![x]\!] ; \text{SUB} ; \text{STORE-}x ; \\
& \text{PUSHN-}1 ; \text{FETCH-}x ; \text{EQ} ; \text{NEG}) \\
= \; & \text{PUSHN-}1 ; \text{STORE-}y ; \text{PUSHN-}1 ; \text{FETCH-}x ; \text{EQ} ; \text{NEG} ; \\
& \text{LOOP } (\text{FETCH-}x ; \text{FETCH-}y ; \text{MULT} ; \text{STORE-}y ; \\
& \text{PUSHN-}1 ; \text{FETCH-}x ; \text{SUB} ; \text{STORE-}x ; \text{PUSHN-}1 ; \text{FETCH-}x ; \text{EQ} ; \text{NEG})
\end{aligned}
$$

Näited 4

$$\begin{aligned}
&\mathcal{CS}[\![z := 0; r := x; \texttt{while } y \leq r \texttt{ do } (r := r - y; z := z + 1)]\!] \\
= \;& \text{PUSHN-}0 \text{ ; STORE-}z \text{ ; FETCH-}x \text{ ; STORE-}r \text{ ; FETCH-}r \text{ ; FETCH-}y \text{ ; LE ;} \\
& \text{LOOP } (\text{FETCH-}y \text{ ; FETCH-}r \text{ ; SUB ; STORE-}r \text{ ;} \\
& \text{PUSHN-}1 \text{ ; FETCH-}z \text{ ; ADD ; STORE-}z \text{ ; FETCH-}r \text{ ; FETCH-}y \text{ ; LE})
\end{aligned}$$

Märkusi

- Transleerivad funktsioonid on kompositsioonilised!

Vastavus semantikate vahel

Mõned konventsioonid ja matemaatilised tähistused

- Siirdesüsteemi üleminekute käigus läbitavate konfiguratsioonide jadu käsitleme listidena.
- **While**-keele konfiguratsioonide puhul loeme

  $$\mathrm{state}\,\langle p\ ,\ s\rangle = s \quad \text{ja} \quad \mathrm{state}\,s = s;$$

  AM-keele konfiguratsioonide puhul loeme

  $$\mathrm{state}\,\langle p\ ,\ (e\ ,s)\rangle = (e, s).$$

- Olgu c **While**- või **AM**-keele struktuurse semantika konfiguratsioon ning $\to$ vastav üleminekurelatsioon. Siis $\mathrm{chain}\,c$ olgu list, mis rahuldab järgmisi tingimusi:

  $$\mathtt{head}\,(\mathrm{chain}\,c) = c,$$
  $$\mathtt{tail}\,(\mathrm{chain}\,c) = \begin{cases} \varepsilon, & \text{kui } \neg\exists d.\ c \to d, \\ \mathrm{chain}\,d, & \text{kui } c \to d. \end{cases}$$

  Teisi sõnu, $\mathrm{chain}\,c$ on kõigi konfiguratsioonist c alustades läbitud siirdesüsteemi konfiguratsioonide list.

Programmi jälitussemantika

- Olgu p **While**-programm. Ütleme, et p tähendus jälitussemantikas on

$$\mathcal{TS}_{\mathbf{While}}[\![p]\!] = \{\mathtt{map}\ \mathrm{state}\ (\mathrm{chain}\ \langle p\ ,\ s\rangle) \mid s \in \mathbf{State}\}\ .$$

- Olgu p **AM**-programm. Ütleme, et p tähendus jälitussemantikas on

$$\mathcal{TS}_{\mathbf{AM}}[\![p]\!] = \{\mathtt{map}\ \mathrm{state}\ (\mathrm{chain}\ \langle p\ ,\ (\varepsilon\ , s)\rangle) \mid s \in \mathbf{State}\}\ .$$

Definitsioon

- Loeme kooinduktiivselt

$$
\begin{aligned}
s \approx t \iff\ & \\
& s \neq \varepsilon \ \wedge\ t \neq \varepsilon \ \wedge \\
& \mathtt{head}\, t = (\varepsilon, \mathtt{head}\, s) \ \wedge \\
& (\mathtt{tail}\, s = \varepsilon \ \Rightarrow\ \mathtt{tail}\, t = \varepsilon) \ \wedge \\
& (\mathtt{tail}\, s \neq \varepsilon \ \Rightarrow\ \exists k > 0.\, \mathtt{tail}\, s \approx \mathtt{tail}^k\, t).
\end{aligned}
$$

- Laiendame relatsiooni $\approx$ loomulikul viisil hulkadele:

$$S \approx T \iff (\forall s \in S.\, \exists t \in T.\, s \approx t) \ \wedge\ (\forall t \in T.\, \exists s \in S.\, s \approx t).$$

- Järelikult transleerimine on korrektne parajasti siis, kui

$$\mathcal{TS}_{\mathbf{While}}[\![p]\!] \approx \mathcal{TS}_{\mathbf{AM}}[\![\mathcal{CS}[\![p]\!]]\!]$$

Translaatori korrektsuse tõestus

Aluslemmad AM semantika kohta

AM struktuurse semantika üleminekute ühtne kuju

Nummerdame üleminekuaksioomid arvudega $1, \ldots, 16$.

- Aksioomide kujust tulenevalt siis

$$\langle c\ ,\ (e\ ,s)\rangle \rhd \langle c'\ ,\ (e'\ ,s')\rangle \iff$$
$$\exists i \exists d.\, c = i\ ;d\ \wedge\ c' = f(i,e)\ ;d\ \wedge \bigvee_{k=1}^{16} P_k(i,e,e',s,s'),$$

  kus

  - $f : \mathbf{Inst} \times \mathbf{Stack} \dashrightarrow \mathbf{Code}$ on üleminekul toimuvat koodiasendust formaliseeriv osaline kujutus,
  - P_k on predikaat, mis näitab k-nda üleminekuaksioomi kasutuspiirkonda.

- Ülesanne: ekstraheerida aksioomidest välja f ning $P_1, \ldots, P_{16}$ otsesed definitsioonid.

Magasini saba ei mängi rolli

- Seejuures
  - iga $k \in \{1, \ldots, 16\}$, $i \in \mathbf{Inst}$, $e, e', e'' \in \mathbf{Stack}$, $s, s' \in \mathbf{State}$ korral
    $$P_k(i, e, e', s, s') \;\Rightarrow\; P_k(i, e +\!+\, e'', e' +\!+\, e'', s, s'),$$
  - iga $i \in \mathbf{Inst}$, $e, e'' \in \mathbf{Stack}$ korral
    $$f(i, e) \neq \bot \;\Rightarrow\; f(i, e +\!+\, e'') = f(i, e).$$

Lemma 1.1

***AM** struktuurne semantika on determineeritud.*

– Igale konfiguratsioonile on rakenduv ülimalt üks 16 aksioomist.
  Seega algkonfiguratsioon määrab üheselt kogu järgneva üleminekute jada.

Lemma 1.2

Olgu $i \in \mathbf{Inst}$, $c, d \in \mathbf{Code}$, $e, e' \in \mathbf{Stack}$, $s, s' \in \mathbf{State}$.

$$\langle i\,;d\ ,\ (e\,,s)\rangle \rhd \langle c\ ,\ (e'\,,s')\rangle \ \Rightarrow\ \exists c'.\, c = c'\,;d.$$

– Sobib $c' = f(i, e)$.

Lemma 1.3

Olgu $i \in \mathbf{Inst}$, $c \in \mathbf{Code}$, $e, e' \in \mathbf{Stack}$, $s, s' \in \mathbf{State}$.

$$\exists d.\ \langle i\,;d\ ,\ (e\,,s)\rangle \rhd \langle c\,;d\ ,\ (e'\,,s')\rangle\ \Rightarrow$$
$$\forall d.\ \langle i\,;d\ ,\ (e\,,s)\rangle \rhd \langle c\,;d\ ,\ (e'\,,s')\rangle\,.$$

– Ükski P_k ei sõltu koodijäägist d.

Lemma 1.4

Olgu $c, c', c'' \in \mathbf{Code}$, $e, e' \in \mathbf{Stack}$, $s, s' \in \mathbf{State}$.

$$\langle c\ ,\ (e\ ,s)\rangle \rhd \langle c'\ ,\ (e'\ ,s')\rangle\ \Rightarrow\ \langle c\,;c''\ ,\ (e\ ,s)\rangle \rhd \langle c'\,;c''\ ,\ (e'\ ,s')\rangle\,.$$

– $c = i\,;d$,

lemmast 1.2

$$c' = c'''\,;d, \text{ niisiis } \langle i\,;d\ ,\ (e\ ,s)\rangle \rhd \langle c'''\,;d\ ,\ (e'\ ,s')\rangle\,,$$

lemmast 1.3

$$\begin{aligned}\langle c\,;c''\ ,\ (e\ ,s)\rangle\ &=\ \langle i\,;d\,;c''\ ,\ (e\ ,s)\rangle \\ &\rhd\ \langle c'''\,;d\,;c''\ ,\ (e'\ ,s')\rangle = \langle c'\,;c''\ ,\ (e'\ ,s')\rangle\,.\end{aligned}$$

Lemma 1.5

Olgu $c, c' \in \mathbf{Code}$, $e, e', e'' \in \mathbf{Stack}$, $s, s' \in \mathbf{State}$.

$$\langle c\ ,\ (e\ , s)\rangle \rhd \langle c'\ ,\ (e'\ , s')\rangle\ \Rightarrow\ \langle c\ ,\ (e +\!+\ e''\ , s)\rangle \rhd \langle c'\ ,\ (e' +\!+\ e''\ , s')\rangle\ .$$

– $c = i\ ;d$,

$$P_k(i, e, e', s, s')\ \Rightarrow\ P_k(i, e +\!+\ e'', e' +\!+\ e'', s, s').$$

Lemma 1.6

Olgu $c, c', c'' \in \mathbf{Code}$, $e, e', e'' \in \mathbf{Stack}$, $s, s' \in \mathbf{State}$.

$$\langle c\ ,\ (e\ ,s)\rangle \rhd \langle c'\ ,\ (e'\ ,s')\rangle\ \Rightarrow$$
$$\langle c\,;c''\ ,\ (e +\!\!+\ e''\ ,s)\rangle \rhd \langle c'\,;c''\ ,\ (e' +\!\!+\ e''\ ,s')\rangle\ .$$

– Lemmast 1.4 $\langle c\,;c''\ ,\ (e\ ,s)\rangle \rhd \langle c'\,;c''\ ,\ (e'\ ,s')\rangle$,
lemmast 1.5 $\langle c\,;c''\ ,\ (e +\!\!+\ e''\ ,s)\rangle \rhd \langle c'\,;c''\ ,\ (e' +\!\!+\ e''\ ,s')\rangle$.

Lemma 1.7

Olgu $c, c', c'' \in \mathbf{Code}$, $e, e', e'' \in \mathbf{Stack}$, $s, s' \in \mathbf{State}$.

$$\langle c\ ,\ (e\ ,s)\rangle \rhd^k \langle c'\ ,\ (e'\ ,s')\rangle\ \Rightarrow$$
$$\langle c\,;c''\ ,\ (e +\!\!+\ e''\ ,s)\rangle \rhd^k \langle c'\,;c''\ ,\ (e' +\!\!+\ e''\ ,s')\rangle\ .$$

– Induktsioon k järgi, kasutades lemmat 1.6.

Lemma 1.8

Olgu $c, c' \in \mathbf{Code}$, $e, e' \in \mathbf{Stack}$, $s, s' \in \mathbf{State}$.

$$\langle c\,;c'\ ,\ (e\,,s)\rangle \rhd^{k} \langle \varepsilon\ ,\ (e'\,,s')\rangle \ \Rightarrow$$
$$\exists e'', s'', k',\ k''.\ \langle c\ ,\ (e\,,s)\rangle \rhd^{k'} \langle \varepsilon\ ,\ (e''\,,s'')\rangle \ \wedge$$
$$\langle c'\ ,\ (e''\,,s'')\rangle \rhd^{k''} \langle \varepsilon\ ,\ (e'\,,s')\rangle \ \wedge$$
$$k' + k'' = k.$$

– Induktsioon k järgi,
vaadates eraldi juhte $c = \varepsilon$ ja $c \neq \varepsilon$.

Lemma 1.8 tõestus: $c = \varepsilon$

$c = \varepsilon$,

võtame $e'' = e$, $s'' = s$,

$$\langle c\ ,\ (e\,,s)\rangle = \langle \varepsilon\ ,\ (e\,,s)\rangle \rhd^0 \langle \varepsilon\ ,\ (e''\,,s'')\rangle\,,$$
$$\langle c'\ ,\ (e''\,,s'')\rangle = \langle c\,;c'\ ,\ (e\,,s)\rangle \rhd^k \langle \varepsilon\ ,\ (e'\,,s')\rangle\,,$$

$0 + k = k$.

Lemma 1.8 tõestus: $c \neq \varepsilon$

$c = i\,;d, i \in \mathbf{Inst}$,

$\langle i\,;d\,;c'\,,\,(e\,,s)\rangle \rhd \langle c''\,;d\,;c'\,,\,(e'''\,,s''')\rangle \rhd^{k-1} \langle \varepsilon\,,\,(e'\,,s')\rangle$,

$\langle i\,;d\,,\,(e\,,s)\rangle \rhd \langle c''\,;d\,,\,(e'''\,,s''')\rangle$ (lemma 1.3),

ind. eeld.:

$$\langle c''\,;d\,,\,(e'''\,,s''')\rangle \rhd^{k'''} \langle \varepsilon\,,\,(e''\,,s'')\rangle\,,$$
$$\langle c'\,,\,(e''\,,s'')\rangle \rhd^{k''} \langle \varepsilon\,,\,(e'\,,s')\rangle\,,$$
$$k''' + k'' = k - 1,$$

kokku

$$\langle c\,,\,(e\,,s)\rangle = \langle i\,;d\,,\,(e\,,s)\rangle \rhd^{k'''+1} \langle \varepsilon\,,\,(e''\,,s'')\rangle\,,$$
$$\langle c'\,,\,(e''\,,s'')\rangle \rhd^{k''} \langle \varepsilon\,,\,(e'\,,s')\rangle\,,$$
$$k''' + 1 + k'' = k.$$

Avaldiste transleerimine

Lemma 2.1

Olgu $a \in \mathbf{AExp}$*,* $c \in \mathbf{Code}$*,* $e \in \mathbf{Stack}$*,* $s \in \mathbf{State}$*.*

$$\langle \mathcal{CA}[\![a]\!] ; c\ ,\ (e\ ,s)\rangle \rhd^{+} \langle c\ ,\ (\mathcal{A}[\![a]\!]s : e\ ,s)\rangle\ .$$

– Induktsioon a ehituse järgi.

∗ $a = n \in \mathbf{Num}$,

$$\begin{aligned}\langle \mathcal{CA}[\![n]\!] ; c\ ,\ (e\ ,s)\rangle &= \langle \text{PUSHN-}n\ ; c\ ,\ (e\ ,s)\rangle \\ &\rhd \langle c\ ,\ (\mathcal{N}[\![n]\!] : e\ ,s)\rangle = \langle c\ ,\ (\mathcal{A}[\![n]\!]s : e\ ,s)\rangle\ .\end{aligned}$$

∗ $a = x \in \mathbf{Var}$,

$$\begin{aligned}\langle \mathcal{CA}[\![x]\!] ; c\ ,\ (e\ ,s)\rangle &= \langle \text{FETCH-}x\ ; c\ ,\ (e\ ,s)\rangle \\ &\rhd \langle c\ ,\ (s\ x : e\ ,s)\rangle = \langle c\ ,\ (\mathcal{A}[\![x]\!]s : e\ ,s)\rangle\ .\end{aligned}$$

∗ $a = a' \oplus a''$, kus $\oplus$ on $+$, $-$, $*$; võttes vastavalt w väärtuseks ADD, SUB, MULT, saame induktsiooni eeldust kasutades

$$\begin{aligned}\langle \mathcal{CA}[\![a' \oplus a'']\!] ; c\ ,\ (e\ ,s)\rangle &= \langle \mathcal{CA}[\![a'']\!] ; \mathcal{CA}[\![a']\!] ; w\ ; c\ ,\ (e\ ,s)\rangle \\ &\rhd^{+} \langle \mathcal{CA}[\![a']\!] ; w\ ; c\ ,\ (\mathcal{A}[\![a'']\!]s : e\ ,s)\rangle \\ &\rhd^{+} \langle w\ ; c\ ,\ (\mathcal{A}[\![a']\!]s : \mathcal{A}[\![a'']\!]s : e\ ,s)\rangle \\ &\rhd \langle c\ ,\ (\mathcal{A}[\![a']\!]s \oplus \mathcal{A}[\![a'']\!]s : e\ ,s)\rangle \\ &= \langle c\ ,\ (\mathcal{A}[\![a' \oplus a'']\!]s : e\ ,s)\rangle\ .\end{aligned}$$

Lemma 2.2

Olgu $b \in \mathbf{BExp}$, $c \in \mathbf{Code}$, $e \in \mathbf{Stack}$, $s \in \mathbf{State}$.

$$\langle \mathcal{CB}[\![b]\!]\,;c\ ,\ (e\ ,s)\rangle \rhd^{+} \langle c\ ,\ (\mathcal{B}[\![b]\!]s : e\ ,s)\rangle\ .$$

– Induktsioon b ehituse järgi, kasutades lemmat 2.1.

Lausete transleerimine

Definitsioonid

- Olgu funktsioon χ keele **While** struktuurse semantika konfiguratsioonide hulgast keele **AM** struktuurse semantika tühja magasiniga konfiguratsioonide hulka:

$$\chi\,\langle S\ ,\ s\rangle\ =\ \langle \mathcal{CS}[\![S]\!]\ ,\ (\varepsilon\ ,s)\rangle\ ,$$
$$\chi\ s\ =\ \langle\varepsilon\ ,\ (\varepsilon\ ,s)\rangle\ .$$

- Olgu $\blacktriangleright$ binaarne relatsioon keele **AM** struktuurse semantika tühja magasiniga konfiguratsioonide hulgal:

$$c \blacktriangleright c' \iff$$
leidub üleminekuahel $c = c_0 \rhd c_1 \rhd \ldots \rhd c_n = c'$, kus $n > 0$.

- Paneme tähele, et

$$\chi\ c \rhd^+ \chi\ c' \ \Rightarrow\ \chi\ c \blacktriangleright \chi\ c'.$$

Lemma 3.1

While *struktuurse semantika suvaliste konfiguratsioonide* c, d *korral*

$$(c \Rightarrow d) \;\Rightarrow\; \chi\, c \blacktriangleright \chi\, d.$$

– $c = \langle S\ ,\ s \rangle$,
induktsioon S ehituse järgi.
Vaatame kõik juhud eraldi läbi.

Lemma 3.1 tõestus: omistamine

$S = x := a$,

$$\begin{aligned}
\chi\langle S\ ,\ s\rangle &= \langle \mathcal{CS}[\![x := a]\!]\ ,\ (\varepsilon\ ,\ s)\rangle \\
&= \langle \mathcal{CA}[\![a]\!]\ ;\ \text{STORE-}x\ ,\ (\varepsilon\ ,\ s)\rangle \\
&\rhd^{+}\langle \text{STORE-}x\ ,\ (\mathcal{A}[\![a]\!]s : \varepsilon\ ,\ s)\rangle \qquad \text{(lemma 2.1)} \\
&\rhd\ \ \langle \varepsilon\ ,\ (\varepsilon\ ,\ s[x \mapsto \mathcal{A}[\![a]\!]s])\rangle = \chi(s[x \mapsto \mathcal{A}[\![a]\!]s]).
\end{aligned}$$

Lemma 3.1 tõestus: `skip`

$S =$ `skip`,

$$\chi\,\langle S\ ,\ s\rangle = \langle \mathcal{CS}[\![\texttt{skip}]\!]\ ,\ (\varepsilon\ ,s)\rangle = \langle \text{NOOP}\ ,\ (\varepsilon\ ,s)\rangle \rhd \langle \varepsilon\ ,\ (\varepsilon\ ,s)\rangle = \chi\ s.$$

Lemma 3.1 tõestus: kompositsioon

$S = S'; S''$,

$\chi\,\langle S'; S''\ ,\ s\rangle = \langle \mathcal{CS}[\![S'; S'']\!]\ ,\ (\varepsilon\ ,\ s)\rangle = \langle \mathcal{CS}[\![S']\!]; \mathcal{CS}[\![S'']\!]\ ,\ (\varepsilon\ ,\ s)\rangle$,

– $\langle S'\ ,\ s\rangle \Rightarrow s'$,
$\langle S'; S''\ ,\ s\rangle \Rightarrow \langle S''\ ,\ s'\rangle$,
ind. eeld.: $\langle \mathcal{CS}[\![S']\!]\ ,\ (\varepsilon\ ,\ s)\rangle = \chi\,\langle S'\ ,\ s\rangle \blacktriangleright \chi\, s' = \langle \varepsilon\ ,\ (\varepsilon\ ,\ s')\rangle$,

$$\begin{aligned}\chi\,\langle S\ ,\ s\rangle &= \langle \mathcal{CS}[\![S']\!]; \mathcal{CS}[\![S'']\!]\ ,\ (\varepsilon\ ,\ s)\rangle \\ &\rhd^{+} \langle \mathcal{CS}[\![S'']\!]\ ,\ (\varepsilon\ ,\ s')\rangle \\ &= \chi\,\langle S''\ ,\ s'\rangle\ .\end{aligned}$$

(ind. eeld., lemma 1.4)

– $\langle S'\ ,\ s\rangle \Rightarrow \langle S'''\ ,\ s'\rangle$,
$\langle S'; S''\ ,\ s\rangle \Rightarrow \langle S'''; S''\ ,\ s'\rangle$,
ind.eeld: $\langle \mathcal{CS}[\![S']\!]\ ,\ (\varepsilon\ ,\ s)\rangle = \chi\langle S'\ ,\ s\rangle \blacktriangleright \chi\langle S'''\ ,\ s'\rangle = \langle \mathcal{CS}[\![S''']\!]\ ,\ (\varepsilon\ ,\ s')\rangle$,

$$\begin{aligned}\chi\,\langle S\ ,\ s\rangle &= \langle \mathcal{CS}[\![S']\!]; \mathcal{CS}[\![S'']\!]\ ,\ (\varepsilon\ ,\ s)\rangle \\ &\rhd^{+} \langle \mathcal{CS}[\![S''']\!]; \mathcal{CS}[\![S'']\!]\ ,\ (\varepsilon\ ,\ s')\rangle \\ &= \langle \mathcal{CS}[\![S''']; S'']\!]\ ,\ (\varepsilon\ ,\ s')\rangle \\ &= \chi\,\langle S'''; S''\ ,\ s'\rangle\ .\end{aligned}$$

(ind. eeld., lemma 1.4)

Lemma 3.1 tõestus: tingimuslause

$S = \texttt{if}\ b\ \texttt{then}\ S'\ \texttt{else}\ S''$,

$$\begin{aligned}
&\chi\ \langle \texttt{if}\ b\ \texttt{then}\ S'\ \texttt{else}\ S''\ ,\ s\rangle \\
&= \langle \mathcal{CS}[\![\texttt{if}\ b\ \texttt{then}\ S'\ \texttt{else}\ S''\,]\!]\ ,\ (\varepsilon\ ,s)\rangle \\
&= \langle \mathcal{CB}[\![b]\!]\,;\textsc{branch}\ (\mathcal{CS}[\![S']\!])\ (\mathcal{CS}[\![S'']\!])\ ,\ (\varepsilon\ ,s)\rangle \\
&\rhd^{+}\langle \textsc{branch}\ (\mathcal{CS}[\![S']\!])\ (\mathcal{CS}[\![S'']\!])\ ,\ (\mathcal{B}[\![b]\!]s : \varepsilon\ ,s)\rangle\ ,
\end{aligned}$$

(lemma 2.2)

– $\mathcal{B}[\![b]\!]s = \mathbf{tt}$,

$\langle \texttt{if}\ b\ \texttt{then}\ S'\ \texttt{else}\ S''\ ,\ s\rangle \Rightarrow \langle S'\ ,\ s\rangle$,

$$\begin{aligned}
\chi\,\langle S\ ,\ s\rangle\ \rhd^{+}\ &\langle \textsc{branch}\ (\mathcal{CS}[\![S']\!])\ (\mathcal{CS}[\![S'']\!])\ ,\ (\mathcal{CB}[\![b]\!]s : \varepsilon\ ,s)\rangle \\
\rhd\ &\langle \mathcal{CS}[\![S']\!]\ ,\ (\varepsilon\ ,s)\rangle = \chi\,\langle S'\ ,\ s\rangle\ .
\end{aligned}$$

– $\mathcal{B}[\![b]\!]s = \mathbf{ff}$,

arutlus analoogiline.

Lemma 3.1 tõestus: tsükkel

$S = \texttt{while}\ b\ \texttt{do}\ S'$,

$$\begin{aligned}
&\chi\ \langle \texttt{while}\ b\ \texttt{do}\ S'\ ,\ s\rangle \\
&= \langle \mathcal{CS}[\![\texttt{while}\ b\ \texttt{do}\ S']\!]\ ,\ (\varepsilon\ ,s)\rangle \\
&= \langle \mathcal{CB}[\![b]\!]\,;\mathrm{LOOP}\ \ (\mathcal{CS}[\![S']\!]\,;\mathcal{CB}[\![b]\!])\ ,\ (\varepsilon\ ,s)\rangle \\
&\rhd^{+}\langle \mathrm{LOOP}\ \ (\mathcal{CS}[\![S']\!]\,;\mathcal{CB}[\![b]\!])\ ,\ (\mathcal{B}[\![b]\!]s : \varepsilon\ ,s)\rangle
\end{aligned} \qquad \text{(lemma 2.2)}$$

– $\mathcal{B}[\![b]\!]s = \mathbf{tt}$,

$\langle \texttt{while}\ b\ \texttt{do}\ S'\ ,\ s\rangle \Rightarrow \langle S'; \texttt{while}\ b\ \texttt{do}\ S'\ ,\ s\rangle$,

$$\begin{aligned}
\chi\,\langle S\ ,\ s\rangle \rhd^{+}\ &\langle \mathrm{LOOP}\ \ (\mathcal{CS}[\![S']\!]\,;\mathcal{CB}[\![b]\!])\ ,\ (\mathcal{B}[\![b]\!]s : \varepsilon\ ,s)\rangle \\
\rhd\ &\langle \mathcal{CS}[\![S']\!]\,;\mathcal{CB}[\![b]\!]\,;\mathrm{LOOP}\ \ (\mathcal{CS}[\![S']\!]\,;\mathcal{CB}[\![b]\!])\ ,\ (\varepsilon\ ,s)\rangle \\
=\ &\langle \mathcal{CS}[\![S']\!]\,;\mathcal{CS}[\![\texttt{while}\ b\ \texttt{do}\ S']\!]\ ,\ (\varepsilon\ ,s)\rangle \\
=\ &\langle \mathcal{CS}[\![S'; \texttt{while}\ b\ \texttt{do}\ S']\!]\ ,\ (\varepsilon\ ,s)\rangle \\
=\ &\chi\,\langle S'; \texttt{while}\ b\ \texttt{do}\ S'\ ,\ s\rangle\,.
\end{aligned}$$

– $\mathcal{B}[\![b]\!]s = \mathbf{ff}$,

$\langle \texttt{while}\ b\ \texttt{do}\ S'\ ,\ s\rangle \Rightarrow s$,

$$\begin{aligned}
\chi\,\langle \texttt{while}\ b\ \texttt{do}\ S'\ ,\ s\rangle \rhd^{+}\ &\langle \mathrm{LOOP}\ \ (\mathcal{CS}[\![S']\!]\,;\mathcal{CB}[\![b]\!])\ ,\ (\mathcal{B}[\![b]\!]s : \varepsilon\ ,s)\rangle \\
\rhd\ &\langle \varepsilon\ ,\ (\varepsilon\ ,s)\rangle = \chi\ s.
\end{aligned}$$

Lemma 3.2

***While**'i struktuurse semantika suvaliste konfiguratsioonide c, d korral*

$$(c \Rightarrow^k d) \;\Rightarrow\; \chi\, c \blacktriangleright^k \chi\, d.$$

– Induktsiooniga k järgi, kasutades lemmat 3.1.

Lemma 3.3

***While**'i struktuurse semantika suvaliste konfiguratsioonide c, d korral*

$$(c \Rightarrow^+ d) \;\Rightarrow\; \chi\, c \rhd^+ \chi\, d.$$

– Lemmast 3.2.

Lemma 3.4

Suvaliste $S \in \mathbf{Stmt}$, $s, s' \in \mathbf{State}$ *korral*

$$(\langle S\ ,\ s\rangle \Rightarrow^* s') \ \Rightarrow\ \langle \mathcal{CS}[\![S]\!]\ ,\ (\varepsilon\ ,s)\rangle \blacktriangleright^* \langle \varepsilon\ ,\ (\varepsilon\ ,s')\rangle\ .$$

– $\langle S\ ,\ s\rangle \Rightarrow^k s'$,
lemma 3.2.

Lemma 3.5

*Kui **While**-keele struktuurses semantikas annab konfiguratsioon $\langle S\ ,\ s\rangle$ lõpmatu üleminekute ahela, siis **AM**-keele struktuurses semantikas annab konfiguratsioon $\langle \mathcal{CS}[\![S]\!]\ ,\ (\varepsilon\ , s)\rangle$ lõpmatu üleminekute ahela.*

– $\forall k.\, \exists d.\, \langle S\ ,\ s\rangle \Rightarrow^k d$,

lemma 3.2.

Lemma 3.6

*Olgu c **While**'i struktuurse semantika konfiguratsioon.*

$$(\exists d.\, \chi\ c \rhd d) \;\Rightarrow\; (\exists d.\, c \Rightarrow d).$$

– $c = s$ või $c = \langle S\ ,\ s\rangle$.

  * $c = s$,
    $\chi\ c = \langle \varepsilon\ ,\ (\varepsilon\ , s)\rangle$,
    $\neg \exists d.\, \chi\ c \rhd d$, vastuolu.
  * $c = \langle S\ ,\ s\rangle$,
    While'i omapära: $\exists d.\, c \Rightarrow d$.

Korrektsus

Lemma 4.1

*Olgu c, d **AM** struktuurse semantika konfi guratsioonid.*

$$c \rhd^k d \;\Rightarrow\; \mathrm{chain}\, d = \mathtt{tail}^k\, (\mathrm{chain}\, c).$$

– Induktsioon k järgi.

Lemma 4.2

***While**'i struktuurse semantika iga konfiguratsiooni c korral*

$$\texttt{map}\ \text{state}\ (\text{chain}\ c) \approx \texttt{map}\ \text{state}\ (\text{chain}(\chi\ c)).$$

– Mistahes **While**'i või **AM** struktuurse semantika konfiguratsiooni c korral

$$\texttt{head}\ (\texttt{map}\ \text{state}\ (\text{chain}\ c)) = \text{state}(\texttt{head}\ (\text{chain}\ c)) = \text{state}\ c,$$
$$\texttt{tail}\ (\texttt{map}\ \text{state}\ (\text{chain}\ c)) = \texttt{map}\ \text{state}\ (\texttt{tail}\ (\text{chain}\ c)).$$

Def. ϱ:

$$\texttt{map}\ \text{state}\ (\text{chain}\ c)\ \varrho\ \texttt{map}\ \text{state}\ (\text{chain}(\chi\ c)).$$

Tõestame kooinduktsiooniga, et $\varrho \subseteq \approx$.

Fikseerime **While**'i struktuurse semantika konfiguratsiooni c:

$$c = \langle p\ ,\ s\rangle\ , \quad \chi\ c = \langle \mathcal{CS}[\![p]\!]\ ,\ (\varepsilon\ , s)\rangle\ .$$

Näitame 4 tingimust $\approx$ definitsioonis.

Lemma 4.2 tõestus: esimesed 3 tingimust

- $\text{chain}\, c \neq \varepsilon$, $\texttt{map}\ \text{state}\ (\text{chain}\, c) \neq \varepsilon$,
  $\text{chain}(\chi\ c) \neq \varepsilon$, $\texttt{map}\ \text{state}\ (\text{chain}(\chi\ c)) \neq \varepsilon$.
- $\texttt{head}\ (\texttt{map}\ \text{state}\ (\text{chain}\, c)) = \text{state}\, c = s$,
  $\texttt{head}\ (\texttt{map}\ \text{state}\ (\text{chain}(\chi\ c))) = \text{state}(\chi\ c) = (\varepsilon, s)$.
- $\texttt{tail}\ (\texttt{map}\ \text{state}\ (\text{chain}\, c)) = \varepsilon$,
  $\texttt{tail}\ (\text{chain}\, c) = \varepsilon$,
  $\neg\exists d.\ c \Rightarrow d$,
  $\neg\exists d.\ \chi\ c \rhd d$, (lemma 3.6)
  $\texttt{tail}\ (\text{chain}(\chi\ c)) = \varepsilon$,
  $\texttt{tail}\ (\texttt{map}\ \text{state}\ (\text{chain}(\chi\ c))) = \varepsilon$.

Lemma 4.2 tõestus: tingimus 4

– $\texttt{tail}\,(\texttt{map}\,\text{state}\,(\text{chain}\,c)) \neq \varepsilon$,
$\texttt{tail}\,(\text{chain}\,c) \neq \varepsilon$,
$c \Rightarrow d$, $\text{chain}\,d = \texttt{tail}\,(\text{chain}\,c)$,
$\chi\,c \rhd^{+} \chi\,d$, (lemma 3.3)
$\chi\,c \rhd^{k} \chi\,d$, $k > 0$,
$\text{chain}(\chi\,d) = \texttt{tail}^{k}\,(\text{chain}(\chi\,c))$, (lemma 4.1)

$$\begin{aligned}
\texttt{tail}(\texttt{map}\,\text{state}\,(\text{chain}\,c)) &= \texttt{map}\,\text{state}\,(\texttt{tail}\,(\text{chain}\,c)) \\
&= \texttt{map}\,\text{state}\,(\text{chain}\,d) \\
\varrho\quad &\texttt{map}\,\text{state}\,(\text{chain}(\chi\,d)) \\
&= \texttt{map}\,\text{state}\,(\texttt{tail}^{k}\,(\text{chain}(\chi\,c))) \\
&= \texttt{tail}^{k}\,(\texttt{map}\,\text{state}\,(\text{chain}(\chi\,c))).
\end{aligned}$$

Korrektsus

– Olgu p **While**-programm.

  * Olgu $q \in \mathcal{TS}_{\mathbf{While}}[\![p]\!]$,
    $q = \mathtt{map}\ \mathrm{state}\ (\mathrm{chain}\ \langle p\ ,\ s\rangle)$,
    $q' = \mathtt{map}\ \mathrm{state}\ (\mathrm{chain}\ \langle \mathcal{CS}[\![p]\!]\ ,\ (\varepsilon\ , s)\rangle)$,
    $q' \in \mathcal{TS}_{\mathbf{AM}}[\![\mathcal{CS}[\![p]\!]]\!]$,
    $q \approx q'$ lemmast 4.2.
  * Olgu $q' \in \mathcal{TS}_{\mathbf{AM}}[\![\mathcal{CS}[\![p]\!]]\!]$,
    $q' = \mathtt{map}\ \mathrm{state}\ (\mathrm{chain}\ \langle \mathcal{CS}[\![p]\!]\ ,\ (\varepsilon\ , s)\rangle)$,
    $q = \mathtt{map}\ \mathrm{state}\ (\mathrm{chain}\ \langle p\ ,\ s\rangle)$,
    $q \in \mathcal{TS}_{\mathbf{While}}[\![p]\!]$,
    $q \approx q'$ lemmast 4.2.

  Kokku $\mathcal{TS}_{\mathbf{While}}[\![p]\!] \approx \mathcal{TS}_{\mathbf{AM}}[\![\mathcal{CS}[\![p]\!]]\!]$.