

## ITT9031 Programmikeelte semantika Koduülesanded 1

Ülesannete tähtaeg on 16.4.2012. Hilinemise korral punktiarv väheneb. Ülesannete lahendamine on eksamihinde saamiseks nõutav.

Ülesannete ühine arutamine on aktsepteeritav, aga lahenduste kopeerimine ei ole lubatav. Küsimused on teretulnud meiliaadressil `tarmo@cs.ioc.ee`.

1. Olgu  $s$  olek nii, et  $s x = 30, s y = 42$ . Rehkenda välja olek, mis saadakse olekust  $s$  järgmise käsu täitmisel:

`while  $x \neq y$  do (if  $x \leq y$  then  $y := y - x$  else  $x := x - y$ )`

Tee arvutus läbi nii loomulikus kui ka struktuurses operatsioonsemantikas.

2. Kasutades loomulikku semantikat, näita (matemaatilise aruteluga), et järgmised **While**'i käsud on paarikaupa samatähenduslikud:

(a) `if ( $b_1 \wedge b_2$ ) then  $S_1$  else  $S_2$`  ja `if  $b_1$  then (if  $b_2$  then  $S_1$  else  $S_2$ ) else  $S_2$` ,

(b) `while  $b_1$  do (while ( $b_1 \wedge b_2$ ) do  $S$ )` ja `while  $b_1$  do (if  $b_2$  then  $S$  else (while true do skip))`,

(c)  `$x := a_1; y := a_2$`  ja  `$y := a_2; x := a_1$`  (eelda, et  $x \notin \text{FV}(a_2), y \notin \text{FV}(a_1)$ )

3. Rmt. harj. 2.11 on näidatud, kuidas keele **While** aritmeetikaavaldiste tähendus määratleda loomulikus semantikas, tuletades kolmikuid  $\langle a, s \rangle \rightarrow_{\text{Aexp}} z$ .

Täienda seda semantikat aritmeetikaavaldise  $b?a_0:a_1$  jaoks, mis testib tingimust  $b$  ja vastavalt tulemusele tagastab kas  $a_0$  või  $a_1$  väärtuse.

Täienda seda ka aritmeetikaavaldise  $S \triangleright a$  jaoks, mis täidab käsu  $S$  etteantud olekust ning tagastab  $a$  lõppväärtuse. Käsu  $S$  toime olekule on kohalik selles mõttes, et kogu avaldis  $S \triangleright a$  olekut ei muuda. Nt olekust  $s$ , kus  $s x = 4, s y = 7$ , annab avaldis  $x := x + 1 \triangleright y + x$  väärtuseks 12, aga olekut ei muuda.

(Oma lahenduse testimiseks võid selle ka realiseerida Haskellis.)

4. Keelel **While** on omadus, et avaldise väärtustamine ei muuda olekut. Vaatleme laiendust, kus see omadus on rikutud. Lubame omistust kasutada nii käsu kui ka aritmeetika-avaldisena, kusjuures aritmeetika-avaldisena omistus  $x := a$  nii tagastab  $a$  väärtuse kui ka omistab selle  $x$ -le.

Esita laiendatud keele loomulik semantika (modifitseerida tuleb aritmeetikaavaldisi, tõeväärtusavaldisi, omistuskäsku, if ja while-käskude puudutavad reeglid). Kolmikute  $\langle a, s \rangle \rightarrow_{\text{Aexp}} z$  asemel tuletatakse muudetud semantikas nelikuid  $\langle a, s \rangle \rightarrow_{\text{Aexp}} \langle z, s' \rangle$ , kus  $a$  on aritmeetikaavaldis,  $z$  on täisarv ning  $s, s'$  on olekud.

(Oma lahenduse testimiseks võid selle ka realiseerida Haskellis.)

5. Esita keele **While** laiendatud keele loomuliku semantika variant, mis loeb, mitu väikest sammu mingi käsu täitmine võtab. Käskude kohta peaks see semantika tuletama nelikuid  $\langle S, s \rangle \rightarrow \langle n, s' \rangle$ , kus  $n$  on väikeste sammude arv, mis kulub  $S$  täitmiseks.

Vaatleme edasi laiendust, kus on ka käsk `reset`, mis töötab muidu nagu `skip`, aga viib sammulugeja algseisu.

Esita ka selle laienduse loomulik semantika. (Nelikutest nagu ülal nüüd ei piisa: tuletatavad väärtused peavad olema pisut teistsugused.)

(Oma lahenduste testimiseks võid need ka realiseerida Haskellis.)

6. Muuda rmt. jaot. 2.4 toodud paralleelse **While**'i struktuurset operatsioonsemantikat nii, et käsk  $S_1 \text{ par } S_2$  on deterministlik, aga vaheldab  $S_1$  ja  $S_2$  väikeseid samme omalaadse õiglusmalli järgi, kus iga kahe  $S_1$  väikese sammu järel tehakse üks  $S_2$  väikene samm (kuni üks käskudest on lõpetanud, misjuhul jätkab ainult teine). kordamööda.

7. Vaatleme abstraktse masina keele laiendust instruksiooniga SWAP ja DUP, mis vahetavad pinu kaks tipmist elementi (sõltumata nende tüüpidest) ja dubleerivad pinu tipmise elementi (sõltumata tema tüübist).

Täienda abstraktse masina semantikat nende instruksioonide jaoks.

Näita abstraktse masina semantikat ja käskude kompileerimisfunktsiooni kasutades, et järgmised koodilõigud on samatähenduslikud:  $\mathcal{CS}[[x := z - y]]$  :  $\mathcal{CS}[[w := x * z]]$  ja  $\text{FETCH } z : \text{DUP} : \text{FETCH } y : \text{SWAP} : \text{SUB} : \text{DUP} : \text{STORE } x : \text{MUL} : \text{STORE } w$ .