

ITT9030 Programmikeelte semantika Koduülesanded 1

Ülesannete tähtaeg on 12.4.2010. 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 $s x = 14$. Rehkenda välja olek, mis saadakse olekust s järgmise käsu täitmisel:

$$y := 1; z := 0; \text{while } y \leq x \text{ do } (y := 2 \times y; z := z + 1)$$

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 (b1 ∨ b2) then S1 else S2` ja `if b1 then S1 else (if b2 then S1 else S2)`,
- (b) `while b1 do (if b2 then S else skip)` ja `while (b1 ∧ b2) do S; while b1 do skip`
- (c) `x := a1; x := x + a2` ja `x := a1 + (a2[x ↦ a1])`

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`-käske 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. Vaatleme keele **While** laiendust käsuga `print a`, mis väljastab avaldise a (lisab ta väljundfaili lõppu).

Esita laiendatud keele loomulik semantika, mis käskude kohta tuletab nelikuid $\langle S, s \rangle \rightarrow \langle o, s' \rangle$, kus o (fail) on list arve, mis käsu S täitmise vältel väljastatakse.

Vaatleme edasi laiendust, kus on ka käsk `clear`, mis tühistab juba tehtud väljundi (puhastab faili).

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 õiglaselt, st niikaua kuni nii S_1 kui ka S_2 pole veel lõpetanud, teevad nad väikeseid samme kordamööda.

7. Vaatleme abstraktse masina keele laiendust instruksioonidega POP ja DUP, mis vastavalt kas eemaldavad pinu tipmise elemendi (sõltumata tema tüübist) või dubleerivad teda.

Täienda abstraktse masina semantikat nende instruksioonide jaoks.

Näita abstraktse masina semantikat ja käskude kompileerimisfunktsiooni kasutades, et järgmised koodilõigud on samatähenduslikud: $\mathcal{CS}[[y := a + x]] : \mathcal{CS}[[z := x]]$ ja $\text{FETCH } x : \text{DUP} : \mathcal{CA}[[a]] : \text{ADD} : \text{STORE } y : \text{STORE } z$ (eeldades, et y ja x on erinevad muutujad).