

ITT9031 Programmikeelte semantika, kevad 2014

Koduülesanded 1

Ülesannete tähtaeg on 31.3.2014. 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 $s x = 60, s y = 18$. Rehkenda välja olek, mis saadakse olekust s järgmise käsu täitmisel:

$$\text{while } \neg(x == y) \text{ do (if } x \leq y \text{ then } y := y - x \text{ else } x := x - y)$$

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 (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 skip else S)` ja `while (b1 ∧ ¬b2) do S; while b1 do skip`

(c) `x := a1; x := a2` ja `x := a2[a1/x]`

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

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

Täiend 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. Loomulikku semantikat pole keeruline anda ka **While**-keele eksootilisematele laiendustele. Lisame keelele käsukujud `undo S` ja `do S ifafter b`. Esimese mõte on täita käsk S "tagurpidi". Teise puhul täidetakse käsk S ning siis kontrollitakse, kas lõppolekus kehtib b . Kui jah, siis nii lõpetataksegi. Kui ei, siis minnakse tagasi algolekusse. Defineeri nende käskude loomulik semantika (muude käskude semantikat muutmata).

Näita, et kui $\langle \text{do } S \text{ ifafter } b, s \rangle \rightarrow s'$, siis $\langle S; \text{if } b \text{ then skip else (undo } S), s \rangle \rightarrow s'$. Näita samuti (konkreetsel näitel), et vastupidine üldjuhul ei kehti.

5. Kirjuta välja **While** loomuliku semantika instrumenteering, mis käsu jaoks rehkendab välja maksimaalset täisarvu, mida on selle täitmise vältel omistatud.

See semantika peaks käskude kohta tuletama viisikuid $\langle S, s, m \rangle \rightarrow \langle s', m' \rangle$, kus m ja m' on maksimaalne omistatud täisarv enne jooksva käsu täitmise alustamist ja pärast lõpetamist.

(Oma lahenduste testimiseks võid need ka realiseerida Haskellis.)

6. Muuda rmt. jaot. 2.4 toodud paralleelse **While**'i struktuurset operatsioonsemantikat nii, et niipea, kui üks kahest paralleelselt täidetavast käsust lõpetab, katkestatakse ka teine.

7. Vaatleme abstraktse masina keele laiendust instruksioonidega **SWAP** ja **DUP**, mis vastavalt kas vahetavad pinu kaks tipmist elementi omavahel (sõltumata nende tüüpidest) või dubleerivad pinu tipmist elementi.

Täiend abstraktse masina semantikat nende instruksioonide jaoks.

Näita abstraktse masina semantikat ja käskude kompileerimisfunktsiooni kasutades, et järgmised koodilõigud on samatähenduslikud: $\mathcal{CS} \llbracket y := x - 3 * (z * x) \rrbracket$ ja `FETCH x : DUP : FETCH z : MUL : LOAD 3 : MUL : SWAP : SUB : STORE` (eeldades, et y ja x on erinevad muutujad).