

T-79.144

Syksy 2004

Logiikka tietotekniikassa: perusteet

Laskuharjoitus 12 (opetusmoniste, kappaleet 9.1 – 9.5)

30.11. – 3.12.2004

1. Osoita lauselogiikan avulla oheisten ehtolausekkeiden ekvivalenssi.

$$(a) \quad !(a == b \mid \mid a < b)$$

$$(b) \quad a != b \&\& !(b > a)$$

Opetusmoniste määritteli Boolean lauseet tiettyjen perustapausten avulla, muiden ollessa lyhennysmerkintöjä. Näin ollen:

$$a == b \equiv_{def} !(a > b) \&\& !(b > a)$$

ja

$$a < b \equiv_{def} b > a$$

$$a != b \equiv_{def} !(a == b)$$

Määriteltäköön atomilauseet  $A = "a > b"$  ja  $B = "b > a"$ . Näin a-kohdan lause saa muodon

$$\neg((\neg A \wedge \neg B) \vee B)$$

ja b-kohdan vastaavasti:

$$\neg(\neg A \wedge \neg B) \wedge \neg B$$

Havaitaan, että kun ensimmäiseen sovelletaan kerran De Morganin teoreemaa saadaan jälkimmäinen. Näin ollen ne luonnollisesti ovat ekvivalentit.

2. Osoita osittainen oikeellisuus seuraavissa tapauksissa:

Esitetyt ohjelmat koostuvat peräkkäisistä sijoituslausekkeista. Näin ollen ne voidaan ratkaista käyttämällä hyväksi opetusmonisteessa annettuja päättelysääntöjä:

$$\text{Sijoituslauseke: } \frac{[B\{x/E\}] \quad x = E \quad [B]}{[B]}$$

$$\text{Kompositio: } \frac{[B_0] \quad C_1 \quad [B_1] \quad [B_1] \quad C_2 \quad [B_2]}{[B_0] \quad C_1 ; C_2 \quad [B_2]}$$

$$(a) \models_p [x > 0] y = x + 1 [y > 1]$$

**Ratk.** Lähtemällä jälkiehdosta ja soveltamalla sijoituslausekkeen päättelysääntöä taaksepäin saadaan muoto  $[x + 1 > 1] y = x + 1 [y > 1]$

$x > 0$  on ekvivalentti lauseke  $x + 1 > 1$  kanssa, joten erityisesti siis sen vahvennus, jolloin väittämä pätee.

$$(b) \models_p [\text{true}] y = x ; y = x + x + y [y == 3 * x]$$

**Ratk.** Soveltamalla kahteen kertaan sijoituslausekkeen päättelysääntöä saadaan muoto:

$$\begin{aligned} [x + x + y == 3 * x] y = x + x + y [y == 3 * x] \\ [x + x + x == 3 * x] y = x [x + x + y == 3 * x] \end{aligned}$$

ja edelleen komposition sääntöä soveltamalla:

$$[x + x + x == 3 * x] y = x ; y = x + x + y [y == 3 * x]$$

Lauseke  $x + x + x == 3 * x$  on luonnollisesti aina totta kokonaislukujen joukossa, jolloin väittämä pätee.

$$(c) \models_p [x > 1] a = 1 ; y = x ; y = y - a [y > 0 \ \&\& \ x > y]$$

Samoin kuin b-kohdassa sijoituslausekkeen päättelysääntöä tulee soveltaa peräkkäin. Saadaan:

$$\begin{aligned} [y - a > 0 \ \&\& \ x > y - a] y = y - a [y > 0 \ \&\& \ x > y] \\ [x - a > 0 \ \&\& \ x > x - a] y = x [y - a > 0 \ \&\& \ x > y - a] \\ [x - 1 > 0 \ \&\& \ x > x - 1] a = 1 [x - a > 0 \ \&\& \ x > x - a] \end{aligned}$$

Kompositiosäännön suoraviivainen sovellus sivuutetaan. Nyt ensimmäisen ehdon  $x - 1 > 0 \ \&\& \ x > x - 1$  jälkimmäinen konjunktio on tutkittavissa kokonaislukustruktuureissa aina tosi. Edelleen ehto  $x - 1 > 0$  on yhtäpitävä ehdon  $x > 1$  kanssa, joka esiintyy väitteessä.

3. Osoita, että  $\models_p [\text{true}] P [z == \min(x, y)]$ , missä P on seuraava ohjelma:

```

if (x > y) then {
    z = y
} else {
    z = x
}

```

**Ratk.** Tehtävän ratkaisuun tarvitaan edelläesitettyjen päättelysääntöjen lisäksi seuraava sääntö:

Ehtolauseke:  $\frac{[B_1 \ \&\& \ B] \ C_1 \ [B_2] \quad [B_1 \ \&\& \ !B] \ C_2 \ [B_2]}{[B_1] \ \text{if}(B) \ \text{then} \ \{C_1\} \ \text{else} \ \{C_2\} \ [B_2]}$

Pyritään osoittamaan  $z == \min(x, y)$ . Tämä on siis ylläolevan säännön ehto  $B_2$ . Tutkitaan komentoja  $C_1$  ja  $C_2$  erikseen.  $C_2$ :sta saadaan.

$$[x == \min(x, y)] \ z = x \ [z == \min(x, y)]$$

Ehto  $x == \min(x, y)$  on yhtäpitävää lausekkeen  $x \leq y$  tai lausekkeen  $!(x > y)$  kanssa. Luonnollisesti lauseke voidaan kirjoittaa myös muotoon  $\text{true} \ \&\& \ !(x > y)$ . Komennosta  $C_1$  saadaan vastaavasti:

$$[y == \min(x, y)] \ z = y \ [z == \min(x, y)]$$

Ja edelleen samalla päättelyllä esiehdoksi  $\text{true} \ \&\& \ (x > y)$ . Nyt voidaan soveltaa ehtolausekkeen päättelysääntöä (viivan yläpuolella olevat lausekkeet on johdettu) ja saadaan:

```

[true]
if (x > y) then {
    z = y
} else {
    z = x
} [z == min(x, y)]

```

#### 4. Osoita ohjelmasta Sum seuraavat ominaisuudet:

- (a)  $\models_p [\text{true}] \ \text{Sum} \ [z == x + y]$
- (b)  $\models_t [0 \leq y] \ \text{Sum} \ [z == x + y]$

Sum on seuraava ohjelma:

```

z = x ;
v = y ;
while(!(v == 0)) {
    z = z + 1 ;
    v = v - 1
}

```

Tehtävän a-kohdassa pitää todistaa muotoa  $[B_1] \text{ while}(B) \{C\} [B_2]$  oleva osittainen oikeellisuus. Tähän on annettu seuraava menettely:

- A1. Valitaan  $I$  siten, että  $\models_{\mathbb{Z}} (I \ \&\& \ !B) \rightarrow B_2$ .
- A2. Haetaan heikoin esiehto  $I'$  siten, että  $\models_p [I'] C [I]$ .
- A3. Osoitetaan  $\models_{\mathbb{Z}} I \ \&\& \ B \rightarrow I'$ , minkä nojalla  $\models_p [I \ \&\& \ B] C [I]$  ja edelleen  $\models_p [I] \text{ while}(B) \{C\} [I \ \&\& \ !B]$ .
- A4. Osoitetaan  $\models_{\mathbb{Z}} B_1 \rightarrow I$ .

Menettelyssä  $I$  on invariantti, jonka pitäisi olla aina voimassa, silmukan lauseiden suorituksen jälkeen. Nyt silmukkaa tutkimalla havaitaan, että muutujan  $z$  arvo kasvaa ja  $v$  pienenee, erityisesti siis niiden summa pysyy vakiona. Kun niille lisäksi annetaan alkuarvot  $x$  ja  $y$  on summa näiden vakioiden summan suuruinen. Olkoon siis  $I$  lauseke  $z + v == x + y$ . Nyt havaitaan, että menetelmän kohta A1 pätee, koska  $!B$  on lauseke  $v == 0$ , josta luonnollisesti seuraa, että  $z == x + y$ . Lähdetään siis hakemaan silmukan komentojen heikoimpia esiehtoja. Saadaan:

$$\begin{aligned}
& [z + v - 1 == x + y] \ v = v - 1 \ [z + v == x + y] \\
& [z + 1 + v - 1 == x + y] \ z = z + 1 \ [z + v - 1 == x + y]
\end{aligned}$$

Saadussa esiehdossa  $+1$  ja  $-1$  kumoutuvat, jolloin esiehto on sama kuin invariantti  $I$ . Menetelmän kohdassa A3 tulee osoittaa, että  $I \ \&\& \ B \rightarrow I'$ , so. lause:

$$z + v == x + y \ \&\& \ !(v == 0) \rightarrow z + v == x + y$$

Tämä luonnollisesti pätee. Näin ollen pätee myös:

```

[z + v == x + y]
while(!(v == 0)) {
    z = z + 1 ;
    v = v - 1
} [z + v == x + y && v == 0]

```

Menetelmän kohta A4 redusoituu lauseen  $I \rightarrow I$  todistamiseen. Todistuksen loppuunsaattamiseksi pitää vielä tutkia alun sijoituslauseet. Saadaan:

$$\begin{aligned} [z + y == x + y] \quad v = y \quad [z + v == x + y] \\ [x + y == x + y] \quad z = x \quad [z + y == x + y] \end{aligned}$$

Ensimmäinen ehto  $x + y == x + y$  on luonnollisesti aina totta  $\square$ .

$$(b) \models_t [0 <= y] \text{ Sum } [z == x + y]$$

Tehtävän b-kohdassa pyydetään todistamaan, että ohjelma myös päättyy ja että tällöin tarvitaan vahvempi esiehto. Täydellisen oikeellisuuden todistamisessa tarvitaan seuraavaa vahvempaa päättelysääntöä:

$$\frac{[B_1 \ \&\& \ B_2 \ \&\& \ (E == n)] \ C \ [B_1 \ \&\& \ (E < n)]}{[B_1] \ \text{while}(B_2) \ \{C\} \ [B_1 \ \&\& \ !B_2]}$$

Säännössä  $B_1$  sisältää aiemman invariantin lisäksi vaatimuksen siitä, että kokonaislukulauseke  $E \geq 0$ . Sääntö sanoo, että aina kun toistolausekkeen komentoja  $C$  suoritetaan invariantin ja toistolausekkeen ehdon ollessa voimassa niin valitaan kokonaislukumuuttujan  $n$  arvo miten tahansa, niin aina, kun esiehto on voimassa niin jälkiehto on myös. Jälkiehto puolestaan sanoo, että invariantti säilyy ja kokonaislukulauseke pienenee aidosti (säilyen kuitenkin ei-negatiivisena). Näin ollen komentojen  $C$  toistaminen johtaa väistämättä siihen, että toistolausekkeen ehto menee lopulta epätodeksi ja ohjelman suoritus täten päättyy.

Esimerkkiohjelmassa lausekkeeksi  $E$  voidaan valita  $v$ . Tällöin päättelykomennolle  $C$  saa muodon:

$$\begin{aligned} [z + v - 1 == x + y \ \&\& \ 0 <= v - 1 \ \&\& \ v - 1 < n] \quad v = v - 1 \quad [z + v == x + y \ \&\& \ 0 <= v \ \&\& \ v < n] \\ [z + v == x + y \ \&\& \ 0 <= v - 1 \ \&\& \ v - 1 < n] \quad z = z + 1 \quad [z + v - 1 == x + y \ \&\& \ 0 <= v - 1 \ \&\& \ v - 1 < n] \end{aligned}$$

Jotta päättelysääntöä voisi käyttää täytyy todistaa, että saatu esiehto on päättelysäännössä tarvittavan esiehdon  $B_1 \ \&\& \ B_2 \ \&\& \ (E == n)$  looginen seuraus. Kyseinen esiehto saa muodon:

$$z + v == x + y \ \&\& \ 0 <= v \ \&\& \ !(v == 0) \ \&\& \ v == n$$

Looginen seuraavuus pätee, koska muuttujalle  $n$  voidaan valita vain positiivisia arvoja, jotta vasen puoli saadaan todeksi. Näin ollen voidaan päätellä:

```

[z + v == x + y && 0 <= v]
while(!(v == 0)) {
    z = z + 1 ;
    v = v - 1
} [z + v == x + y && 0 <= v && v == 0]

```

Nyt  $0 \leq v$  on jälkiehdon totuusarvon kannalta luonnollisesti redundantti. Päättelyn loppuunsaattamiseksi pitää vielä tutkia alun sijoituslauseet:

$$[z + y == x + y \ \&\& \ 0 \leq y] \ v = y \ [z + v == x + y \ \&\& \ 0 \leq v]$$

$$[x + y == x + y \ \&\& \ 0 \leq y] \ z = x \ [z + y == x + y \ \&\& \ 0 \leq y]$$

Saatu esiehto redusoituu siis muotoon  $0 \leq y$  eli ohjelma päättyy vain jos muuttujan  $y$  arvo on ei-negatiivinen, kun ohjelman suoritus aloitetaan.  $\square$