

T-106.6200 Ohjelmistotekniikan erikoiskurssi L (Scheme-kesäkurssi)

Tentti, 3. 9. 2009

Kirjoita jokaisen palauttamasi paperin ylälaitaan selvästi: "T-106.6200, 3. 9. 2009", koko nimesi, opiskelijanumerosi ja koulutusohjelmasi sekä palauttamiesi **paperien kokonaismäärä**.

Tentissä saa olla mukana kurssikirja Abelson, Sussman & Sussman: *Structure and Interpretation of Computer Programs*, tai sama teksti tai osa siitä tulostettuna – ei kuitenkaan muuta materiaalia.

1. Vastaa seuraaviin kysymyksiin lyhyesti.

- a) (1p) Millä tavalla sijoituslauseen (`set!`) käyttäminen laiskassa Schemessä voi ohjelmoijan kannalta olla ongelmallisempaa kuin standardi-Schemessä?
- b) (1p) Mitä hyötyjä saavutetaan analysoivalla tulkilla verrattuna tavalliseen metasirkulaariseen tulkkiin?
- c) (2p) Mitä ovat Common Lisp Object Systemin (CLOSin) geneeriset funktiot? Miten nämä eroavat perinteisten olio-ohjelmointikielten metodeista?
- d) (2p) Mitä tarkoittavat viite-eheys (referential transparency) ja hygieenisuus R5RS-makroissa?

2. a) (2p) Määrittele ääretön virta (stream) p , joka sisältää kaikki kahden potenssin ($2^0, 2^1, 2^2, 2^3, \dots$). Tee määritelmä *implisiittisesti* eli siten, että määrittelet muuttujan, jonka arvo on riippuvainen siitä itsestään.

Esimerkki p :n käyttämisestä: `(display-stream p)` \Rightarrow 1 2 4 8 16 32 ...

- b) (4p) [SICPin tehtävä 4.63 (s. 453)] Alla oleva tietokanta määrittelee Raamatun sukulaisuussuhteita Adan jälkeläisistä Aatamiin Kainin kautta.

```
(son Adam Cain)      ; Cain on Adamin poika
(son Cain Enoch)
(son Enoch Irad)
(son Irad Mehujael)
(son Mehujael Methushael)
(son Methushael Lamech)
(wife Lamech Ada)
(son Ada Jabal)
(son Ada Jubal)
```

Määrittele seuraavat tätä tietokantaa käyttävät säännöt logiikkatulkin kielellä:

- i. (2p) Jos S on F :n poika ja F on G :n poika, niin S on G :n pojanpoika.
- ii. (2p) Jos W on M :n vaimo ja S on W :n poika, niin S on M :n poika.

Tekemiesi sääntöjen avulla pitäisi pystyä laskemaan esimerkiksi kyselyt "kuka on Kainin pojanpoika", "ketkä ovat Lemekin (Lamech) pojat", "ketkä ovat Metusaelin pojanpojat". Näitä kyselyitä ei tarvitse kertoa vastauksessa.

Käännä!

3. (6p) Määritellään uusi toistorakenne `until`, jonka syntaksi on

```
(until test expr ...)
```

Aluksi evaluoidaan rungossa olevat `expr`-lausekkeet ja tämän jälkeen evaluoidaan testi. Jos testi palauttaa epätoden, toistetaan silmukkarakenne evaluoimalla runko uudestaan. Toistorakenteen evaluointi päättyy kun `test`-osio evaluoituu todeksi, jolloin koko rakenteen paluuarvona palautetaan arvo `#t`.

Toteuta tämä `until`-rakenne uutena erikoismuotona SICPin kohdan 4.1 metasirkulaariseen (ei analysoivaan) tulkkiin. Älä kuitenkaan tee sitä pelkkänä syntaksimuunnoksena, kuten `cond` on tehty kirjan tulkissa. Anna vastauksenasi kaikki koodi, mitä metasirkulaarisesta tulkista tulee muuttaa.

4. a) (3p) Toteuta edellisen tehtävän `until`-rakenne joko R5RS-makroilla tai `defmacro`-rakenteella (joko Scheme-tulkkien `defmacro`- tai `define-macro`-rakenteella tai Common Lispillä).
Tässä tehtävässä ei siis muokata SICPin tulkkeja vaan kirjoitetaan makromäärittelyjä. Makromäärittelyjen syntaksin ei tarvitse olla täsmälleen oikein.
- b) (3p) Mihin tarvitaan SICPin kohdan 5.5 kääntäjässä `linkage`-argumenttia? Mitä vaihtoehtoja sen arvolla on ja miksi niitä on niin monta?
5. a) (3p) Mitä seuraavat lausekkeet palauttavat, kun ne evaluoidaan peräkkäin, yksi kerrallaan, saman Scheme-tulkin päätasolla:

```
(define kk '())  
(define (fact n)  
  (if (= n 0)  
      (call-with-current-continuation  
        (lambda (k)  
          (set! kk k)  
          1))  
      (* n (fact (- n 1)))))  
(fact 4)  
(kk 2)  
(fact 3)  
(kk 2)
```

Kahden `define`-lauseen paluuarvoa ei tarvitse kertoa, vain muut neljä arvoa.

Vihje: (`kk 2`):n paluuarvo syntyy siten, että päätasolla aiemmin tehty `fact`-kutsu palaa uudelleen.

- b) (3p) Kerro lyhyesti, miten SICPin kohdan 4.3 epädeterministinen Scheme-tulkki (`ambeval`) toteuttaa epädeterministisen haun. Mitä useiden tulkin proseduurien saama `fail`-argumentti tekee, kun sitä kutsutaan? Miksi monet tulkin proseduurit käyttävät `succeed`-argumenttiaan eivätkä suoraan palautta laskemaansa paluuarvoa?