582305 Symbolinen ohjelmointi
5. harjoitus, 31.10.2002

Huom: 24.10.2002 *ei* siis ole laskuharjoituksia! Näiden tehtävien ratkaisemiseen on siis *kaksi* viikkoa aikaa.

Tehtävä 5.1: *Avioliitto-ongelman* (Marriage Problem) suomalainen versio on seuraavanlainen.

Lavatansseissa on M miestä ja N naista. Jokaisella heistä on lista niistä vastakkaisen sukupuolen edustajista, joiden kanssa haluaisi tanssia, mieluisimmasta alkaen. Tanssiparit muodostuvat seuraavasti.

Joku vapaa mies lähestyy oman listansa mieluisinta sellaista naista, jota ei ole vielä pyytänyt tanssiin, ja esittää pyyntönsä. Jos naisella ei vielä ole kavaljeeria, ja mies on hänen listallaan, niin hän suostuu. Jos naisella taas on jo kavaljeeri, niin hän vertaa nykyistä kavaljeeria pyytäjään, ja valitsee heistä seuraavaksi kavaljeerikseen oman listansa mukaan mieluisamman. Rukkaset saanut mies palaa vapaiden miesten joukkoon jatkamaan parin etsintää. Näin jatketaan, kunnes kaikki mahdolliset parit on muodostettu.

Ohjelmoi tämä menetelmä Scheme-kielellä. Esitä jokaisen ihmisen tiedot listana, jonka ensimmäinen alkio on hänen nimensä, ja loput alkiot hänelle kelpaavien tanssipartnereiden nimet järjestyksessä. Kukin nimi voi olla mikä tahansa muu Scheme-tietoalkio kuin `#f`, ja listoilla voi esiintyä myös sellaisia nimiä, joita ei lavatansseissa ole paikalla.

Menetelmä vaatii $O(MN)$ askelta; valitse tietorakenteesi siten, ettei toteutuksesi ole hitaampi (ainakaan merkittävästi).

Anna lopuksi esimerkki sellaisesta tilanteesta, jossa jokin nainen ja mies jäävät ilman paria, vaikka jokainen ihminen esiintyy ainakin kahden muun ihmisen listalla.

Tehtävä 5.2: Tehtävän 5.1 ratkaisu voi muuttua naisten haussa, eli silloin kun naiset pyytävät ja miehet valitsevat. Anna esimerkki sellaisesta tilanteesta.

Kirjoita Scheme-funktio, joka laskee annetuista miehistä ja naisista ne parit, jotka muodostuvat, olipa hakuvuoro kumpien tahansa.

Tehtävä 5.3: Tarkastellaan kalvojen III.2 samastusalgoritmia `unify`, jonka Scheme-lähdekooditiedostot saat kurssin kotisivulta. Lisää siihen seuraavanlainen jäljitysominaisuus.

Päätasolla on muuttuja `unify-trace-flag`, jonka alkuarvo on `#t`. Silloin `unify` toimii kuten ennenkin. Mutta jos se asetetaan johonkin toteen arvoon, niin silloin `unify` alkaakin toimia siten, että joka rekursiokutsun aluksi tulostetaan sen parametrien sisältämä informaatio jossakin sopivassa muodossa kalvojen II.9.7 oletustulosporttiin. Joka rekursiokutsun lopuksi tulostetaan vastaavalla tavalla sen tuloksen sisältämä informaatio. Pidä huolta (esimerkiksi sopivin sisennyksin) että tulostuksesta pystyy näkemään mitkä alku- ja lopputulosteet kuuluvat samaan kutsuun.

Tehtävä 5.4: Tämäkin tehtävä laajentaa tehtävän 5.3 samastusalgoritmia `unify`. Lisää siihen tällä kertaa yksinkertainen käyttöliittymä, joka lukee käyttäjältä (eli kalvojen II.9.8 oletussyöteportista) kaksi termiä, yrittää samastaa ne, ja tulostaa vastauksen. Tätä toistetaan, kunnes käyttäjä kyllästyy ja antaa ensimmäisenä terminä tiedoston loppumerkin (joka saadaan DrScheme-käyttöliittymässä keltaisella painikkeella `eof`).

Tehtävä 5.5: Kirjoita Scheme-funktio `cart-prod`, joka saa parametreinaan mielivaltaisen

määrän listoja, ja palauttaa niiden *karteesisen tulon*.

Toisin sanoen, lausekkeen `(cart-prod` L_1 L_2 L_3 `...)` arvo koostuu kaikista niistä pareista `(`a `.` L`)` joissa a on listan L_1 alkio ja L on kutsun `(cart-prod` L_2 L_3 L_4 `...)` palauttama lista. Näiden parien keskinäisen järjestyksen voi ohjelmoija valita siten kuin parhaaksi näkee.

Esimerkiksi lausekkeen `(cart-prod '(1 2) '(a b c) '(#t #f))` (eräs mahdollinen) arvo on

```
((1 a #t) (1 a #f) (1 b #t) (1 b #f) (1 c #t) (1 c #f)
 (2 a #t) (2 a #f) (2 b #t) (2 b #f) (2 c #t) (2 c #f))
```

(missä ohjelmoija on valinnut säilyttävänsä syötelistojen järjestykset).

Tehtävä 5.6: Laajenna luentokalvojen II.9.3 (eli Scheme-oppikirjan luvun 3.2.3) pankkitiliesimerkkiä seuraavasti:

- Toteuta Scheme-proseduuri joka *perustaa uuden pankin*. Pankkia perustettaessa annetaan sen alkupääoma.
- Näin perustettuun pankkiin voi *avata tilejä*. Tiliä avattaessa annetaan sen alkusaldo.
- Avatulta tililtä voi *nostaa* rahaa. Jos tilin oma saldo ei riitä, voi puuttuvan osan lainata pankin kaikille tileille yhteisestä pääomasta. Pääoma ei kuitenkaan saa painua negatiiviseksi.
- Avatulle tilille voi *tallettaa* rahaa. Jos tilille on lainattu rahaa pääomasta, niin talletuksella maksetaan tätä lainaa pois.

(Eräs mahdollinen lähestymistapa on, että pankin perustamisfunktio luo sellaisen funktion, jolla perustettuun pankkiin voi avata tilejä. Tämä tilinavausfunktio taas luo sellaisen funktion, jolla avattua tiliä voi käsitellä.)

> "Mitä onkaan pankin ryöstäminen sen perustamiseen verrattuna?"
>
> *Bertolt Brecht*

(Tehtäviä yhteensä 6 kpl.)