582305 Symbolinen ohjelmointi
6. harjoitus, 7.11.2002

Tehtävä 6.1: Esitetään suunnattu verkko vieruslistoina käyttäen Scheme-kielen assosiaatiolistoja. Toisin sanoen, verkossa G on kaari solmusta p solmuun q täsmälleen silloin, kun lausekkeen `(member` q `(cdr (assoc` p G`)))` arvo on tosi. Ohjelmoi tässä esityksessä Scheme-funktio `(has-path-from` G p q`)` joka palauttaa toden arvon täsmälleen silloin kun verkossa G pääsee kulkemaan sen kaaria pitkin solmusta p solmuun q. Onko funktiosi tehokas? Miten se muuttuu, jos lisäksi vaaditaan, että myös tapauksessa $p = q$ on kuljettava vähintään yksi kaari?

Tehtävä 6.2: Kirjoita Scheme-lauseke, joka saa arvokseen seuraavan aivopähkinän kaikki ratkaisut:

> Baker, Cooper, Fletcher, Miller ja Smith asuvat samassa viisikerroksisessa talossa, kukin omassa kerroksessaan. Baker ei asu ylimmässä kerroksessa. Cooper ei asu alimmassa kerroksessa. Fletcher ei asu ylimmässä eikä alimmassa kerroksessa. Miller asuu jossakin ylemmässä kerroksessa kuin Cooper. Smith ja Fletcher eivät asu naapurikerroksissa. Fletcher ja Cooper eivät asu naapurikerroksissa. Missä kerroksessa kukin asuu?

Tehtävä 6.3: Luennoissa IV.2 määriteltiin alkeellinen Prolog-kielinen luonnollisten lukujen yhteenlaskupredikaatti `p/3` käyttäen nollana vakiota `o/0` ja seuraajafunktiona (eli $n \mapsto n + 1$) funktoria `s/1`. Tee samanlainen predikaatti `m/3` kertolaskulle.

Voiko predikaattiasi käyttää lukujen jakamiseen? Miksi?

Tehtävä 6.4: Luennoissa IV.3 annettiin esimerkkinä Englannin kuningatar Viktorian jälkeläisten (osittainen) sukupuu Prolog-faktoina sekä säännöt sukulaisuussuhteille "on äiti" (`mother/2`), "on isoäiti" (`grandmother/2`) ja "on esiäiti" (`ancestress/2`). Hae tämä lähdekooditiedosto itsellesi kurssin kotisivulta. Lisää siihen säännöt seuraaville sukulaisuussuhteille:

(a) "Ovat sisarukset". (Päätä itse, otatko mukaan myös sisar- ja velipuolet.)

(b) "Ovat serkukset".

Pystytkö kirjoittamaan sääntösi siten, ettei samaa henkilöä pidetä oman itsensä sisaruksena, jne.?

Laske sääntöjesi määrittelemät parit Prolog-tulkilla. (Kurssin kotisivun kautta saat toteutuksen SWI Prolog; muitakin ilmaisia toteutuksia on. Laitoksen Linux-järjestelmissä SWI Prolog käynnistyy komennolla `/opt/pl/bin/pl`.)

Tehtävä 6.5: Verenvuototauti eli hemofilia on sukukromosomeissa periytyvä sairaus, jossa veren hyytyminen estyy. Silloin esimerkiksi haavojen paraneminen vaikeutuu huomattavasti.

- Naisen sukukoromosomit ovat X ja X.
  Toisessa niistä saattaa olla taudin aiheuttava geenivirhe. Nainen ei kuitenkaan itse sairastu näkyvästi, koska hänellä on toisessa X-kromosomissaan virheetön geeni. (Periaatteessa molemmat X-kromosomit voisivat olla virheellisiä, mutta käytännössä tämä tapaus voidaan jättää huomiotta: verenvuototautuinen nainen kuolee viimeistään murrosiässä.)
- Miehen sukukromosomit ovat X ja Y.

Jos X-kromosomissa on geenivirhe, niin mies sairastuu näkyvästi, koska Y-kromosomi ei pysty virhettä peittämään.

- Lapsi saa kummaltakin vanhemmaltaan toisen tämän sukukromosomeista.

Anna "**jos... niin**" -säännöt geenivirhettä kantavien naisten tunnistamiseen, kun käytössäsi on tehtävän 6.4 Prolog-tiedosto. Tässä tiedostossa on jokaisesta miehestä m joko fakta `ill(`m`)` ("m sairastaa verenvuototautia") tai fakta `well(`m`)` ("m ei sairasta verenvuototautia").

Kirjoita nämä sääntösi Prolog-predikaattina `carrier/1`.

Tehtävä 6.6: Käytä Prolog-tulkkia etsimään kaikki geenivirhettä kantavat naiset kuningatar Viktorian sukupuusta tehtävässä 6.5.

Miksi sama nainen saattaa löytyä monta eri kertaa? Ja miksi kahden eri kerran välissä saattaa löytyä muita naisia?

(Tehtäviä yhteensä 6 kpl.)