

Kirjoita jokaiseen vastauspaperiin: T-93. 5500 Logiikka- ja rajoiteohjelmointi (5 op), nimesi, opiskelijanumerosi, koulutusohjelma ja vuosikurssi (tai jatko-op.)

1. Selitä **mahdollisimman lyhyesti** seuraavat käsitteet:

- | | |
|----------------|-----------------|
| a) CLP(X) | d) Hornin lause |
| b) Unifikaatio | e) Kvanttori |
| c) Propositio | f) Labeling |

2. Mitä etuja rajoite(logiikka)ohjelmoinnista on verrattuna logiikkaohjelmointiin?

3.

a) Predikaatti `d/3` on määritelty seuraavasti

```
d([],L,L).  
d([H|T],L,[H|R]):- d(T,[H|L],R).
```

Millä `Z`:n arvo(i)lla toteutuu? - `d([a,b,c],[],Z)`.

b) Predikaatti `minimum` toteutuu kun kolmas argumentti on minimi kahdesta ensimmäisestä. Ensimmäiset argumentit ovat numeroita, mutta kolmas voi olla numero tai vapaa muuttuja. Eliminoi toinen testi käyttäen `cut:ia`, ilman että predikaatin semantiikka muuttuu.

```
minimum(X,Y,X):- X <= Y.  
minimum(X,Y,Y):- X > Y.
```

4. Toteuta seuraavat Prolog-predikaatit:

- a) `repeat` Toteutuu äärettömän monta kertaa uudestaan. (1p)
- b) `for(I,A,B)` Toteutuu muuttujan `I` arvoilla `A`, `A+1`, ..., `B` missä `A` ja `B` ovat kokonaislukuja. (2p)
- c) `findall(X,G,L)` Kuten sisäärakennettu predikaatti: `L` on lista `X`:stä jotka toteuttavat `G`:n. Vihje: käytä `asserta` ja `retract`-predikaatteja. (3p)

5. Tietokannassa on kaupunkien välisten tieyhteyksien pituudet muodossa `dist(A,B,D)`, missä `A` ja `B` ovat kaupunkeja ja `D` niiden välinen etäisyys. Toteuta Prolog-predikaatti `shortest_path(A,B,Path,Dist)`, joka hakee kaupunkien `A` ja `B` välisen lyhimmän polun `Path` syvyyshaulla (depth-first search) ja palauttaa polun pituuden muuttujassa `Dist`. Haun pitää havaita syklit ja jättää syklittömät polut hakematta.