

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) Unifikaatio
- b) Differenssilista
- c) Alfa-beta -haku
- d) Hornin lause
- e) Kvanttori
- f) Generate and test

2. Miten rajoite(logiikka)ohjelmointi eroaa logiikkaohjelmoinnista

3.

a) Predikaatti  $m/3$  on määritelty seuraavasti

$m([], L, L)$ .

$m([H|T], L, [H|R]) :- m(L, T, R)$ .

Millä  $Z$ :n arvo(i)lla toteutuu  $?- m([a,b,c], [1,2], Z)$ .

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) `even(L)` (1p) Listan  $L$  pituus on parillinen (0, 2, 4, ...).

b) `insert(X, T, R)` (2p) Lisää alkion  $X$  binäärisen hakupuuhun  $T$  palauttaen puun  $R$ . Ei-tyhjä puu on muotoa `t(Left, Y, Right)` ja tyhjä puu on `nil`. Oleta että predikaatti `gt(X, Y)` toteutuu jos alkio  $X$  on suurempi kuin  $Y$ .

c) `flatten(T, L)` (3p) Termi  $T$  on mahdollisesti monitasoinen lista josta `flatten` muodostaa yksitasoisen listan. Esimerkki:

`flatten([a, [b, []], c], d(zap)), [a, b, c, d(zap)]).`

5. Pinokone tuntee seuraavat viisi käskyä:

`push(X)` Työntää arvon  $X$  pinoon.

`add, sub, mul, div` Pinon kaksi päällimmäistä arvoa korvataan näiden summalla, erotuksella, tulolla tai osamäärällä.

Toteuta predikaatti `encode(Struct, Code)`, jossa

`Struct` on aritmeettinen lauseke (Prolog-termi), joka muodostuu funktoreista `'+' / 1`, `'-' / 1`, `'+' / 2`, `'-' / 2`, `'*' / 2`, `'/' / 2`, sekä numeerisista vakioista.

`Code` on sekvenssi (lista) käskyjä jolla pinokone laskee lausekkeen `Struct` arvon (tulos jää pinon päälle).

Esimerkki: `encode(-(3+4)/5, [push(3), push(4), add, push(-1), mul, push(5), div])`. Predikaattia voidaan siten käyttää sekä kääntämään lauseke pinokoneen kielelle, että kääntämään pinokoneen kieli takaisin lausekkeeksi (ei aina yksikäsitteistä) ja lisäksi tarkistamaan annetun lauseke-käännösparin oikeellisuus kuten esimerkissä. **HUOM!** `Struct` ei ole merkkijono eikä lista vaan valmis Prolog-termi.