Tik-79.144 **Syksy 1999**
Logiikka tietotekniikassa: perusteet
Laskuharjoitus 12
Ratkaisut

1. a) Seuraava ohjelma toteuttaa binääripuun peilikuvauksen:

```
mirror(leaf(V), leaf(V)).

mirror(tree(V, Left, Right), tree(V, NewLeft, NewRight) :-
   mirror(Left, NewRight),
   mirror(Right, NewLeft).
```

Ohjelman ensimmäinen sääntö kertoo, että yksittäinen lehtisolmu ei muutu peilauksessa mitenkään. Toinen sääntö kertoo, että puun saa peilattua vaihtamalla vasen alipuu oikean alipuun peilikuvaan ja vastaavasti oikea alipuu vasemman peilikuvaan.

b) Tehtävänannossa annettu puu voidaan esittää `tree`- ja `leaf`-predikaattien avulla seuraavasti:

```
tree(a, tree(b, leaf(c), leaf(d)), leaf(e))
```

Tarkastellaan, miten `Prolog`-tulkki käsittelee ohjelmaa:

```
Welcome to SWI-Prolog (Version 3.1.0)
Copyright (c) 1993-1998 University of Amsterdam.  All rights reserved.

For help, use ?- help(Topic). or ?- apropos(Word).

%%% ladataan ohjelma 'mirror.pl'
1 ?- [mirror].
mirror compiled, 0.00 sec, 1,376 bytes.

Yes
%%% esitetään kysymys, 'trace' predikaatti tulostaa päättelyketjun.
2 ?- trace, mirror(tree(a, tree(b, leaf(c), leaf(d)), leaf(e)), X).
T Call:  (  7) mirror(tree(a, tree(b, leaf(c), leaf(d)), leaf(e)), _G463)
T Call:  (  8) mirror(tree(b, leaf(c), leaf(d)), _G587)
T Call:  (  9) mirror(leaf(c), _G591)
T Exit:  (  9) mirror(leaf(c), leaf(c))
T Call:  (  9) mirror(leaf(d), _G590)
T Exit:  (  9) mirror(leaf(d), leaf(d))
T Exit:  (  8) mirror(tree(b, leaf(c), leaf(d)), tree(b, leaf(d), leaf(c)))
T Call:  (  8) mirror(leaf(e), _G586)
T Exit:  (  8) mirror(leaf(e), leaf(e))
T Exit:  (  7) mirror(tree(a, tree(b, leaf(c), leaf(d)), leaf(e)),
                      tree(a, leaf(e), tree(b, leaf(d), leaf(c))))

X = tree(a, leaf(e), tree(b, leaf(d), leaf(c)))

Yes
```

```
3 ?-
```

Vastauksesi saatiin tieto siitä, että kyseiselle puulle on olemassa peilikuva ("Yes") sekä muttujasidonta, jolla se löytyi:

```
X = tree(a, leaf(e), tree(b, leaf(d), leaf(c)))
```

c) Vastauksessa predikaattien järjestyksellä on väliä. Muuttujanarvo "_" kertoo, että emme ole kiinnostuneet kyseisellä paikalla olevasta argumentista.

```
member_tree(leaf(V), V).

member_tree(tree(V, _, _), V).

member_tree(tree(_, Left, _), V) :-
   member_tree(Left, V).

member_tree(tree(_, _, Right), V) :-
   member_tree(Right, V).
```

Koko puu saadaan käytyä läpi kyselyllä:

```
member_tree(tree(a, tree(b, leaf(c), leaf(d)), leaf(e)),
                           X)
```

Esimerkkiajo:

```
Welcome to SWI-Prolog (Version 3.1.0)
Copyright (c) 1993-1998 University of Amsterdam.  All rights reserved.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- [tree].
tree compiled, 0.00 sec, 1,812 bytes.

Yes
2 ?- member_tree(tree(a, tree(b, leaf(c), leaf(d)), leaf(e)), X).

X = a ;

X = b ;

X = c ;

X = d ;

X = e ;

No
3 ?-
```