

Deklaratív programozás
Záróvizsga-feladat, 2002. május 28.

Tekintse át és hasonlítsa össze a Prolog eljárás-, ill. az SML függvénydefiníciók, valamint a két nyelv adatdefiníciók lehetőségeit!

Mutassa be, hogy egy adott hívás kiértékelése (egyszerűsítése, redukciója) hogyan zajlik le!

Hasonlítsa össze a két nyelvben a klózik kiválasztási algoritmusát valamint a paraméterátadást (egyesítést ill. mintaillesztést).

A csak az egyik nyelvre jellemző nyelvi elemek közül ismertesse a logikai változó, ill. a részlegesen alkalmazható függvény fogalmát és a használatukból fakadó előnyöket!

Mondanivalójának szemléltetésére írja meg azt a Prolog-eljárást, ill. SML-függvényt t2l néven, amely a

```
% type tree(T) ---> node(T, list(tree(T))).
```

ill.

```
datatype 'a tree = N of 'a * 'a tree list
```

deklarációval létrehozott fa csomópontjaiban tárolt, adott értéknél nagyobb egész számok egy (tetszőleges sorrendű) listáját előállítja.

SML specifikáció és példák:

```
t2l v t = a t fában tárolt és v-nél nagyobb egészek egy listája
t2l : int -> int tree -> int list

- t2l 3 (N(4, []));
> val it = [4] : int list
- t2l 2 (N(4, [N(1, []), N(2, []), N(3, [])]));
> val it = [3, 4] : int list
```

Prolog specifikáció és példák:

```
% t2l(V, T, L): L a T fában tárolt és V-nél nagyobb egészek egy listája.
% :- pred t2l(integer::in, tree(integer)::in, list(integer)::out).

| ?- t2l(2, node(4, []), L).
L = [4] ? ;
no

| ?- t2l(2, node(4, [node(1, []), node(3, [])]), L).
L = [4, 3] ? ;
no
```