

Programowanie deklaratywne

Kolokwium
15 stycznia 2020

1. (2+2)

- a) Proszę napisać funkcję (`delete x l`), która skasuje z listy `l` pierwsze wystąpienie elementu `x`.
- b) Proszę napisać funkcję (`count x l`), której wartością jest ilość występowania elementu `x` w liście `l`.

2. (2+2)

- a) Proszę napisać funkcję (`filter pred l`) dla jednoargumentowego predykatu `pred`, której wartością jest lista zawierająca wszystkie elementy z listy `l` spełniające predykat `pred`. Przykłady:

```
> (filter even '(4 3 5 6))  
'(4 6)  
> (filter list? '((4 3) 7 (1 2 3) 10))  
'((4 3) (1 2 3))
```

- b) Binarne drzewo `d` można reprezentować jako listę

`(element lewe_poddrzewo prawe_poddrzewo)`.

Drzewo puste zostaje reprezentowane przez pustą listę `()`. Dla takiej reprezentacji proszę napisać funkcję (`inorder d`), której wartością jest lista elementów drzewa `d` w kolejności infiksowej. Przykład:

```
> (inorder '(4  
             (3 () ())  
             (2 (9 () ()) (7 () ())))  
'(3 4 9 2 7))
```

3. (3) Niech będą podane następujące definicje.

```
(define a 2)  
(define (p a)  
  (define (pp b)  
    (set! a (* 2 b))  
    (+ a b))  
  (pp 2))
```

Używając model środowiska proszę pokazać jak się ewaluuje wyrażenie `(p 3)`.

4. (2+2)

- a) Proszę zdefiniować predykat `member2(X,L)`, który jest spełniony, jeżeli `X` występuje conajmniej dwa razy w liście `L`.

b) Proszę zdefiniować predykat `down(N,L)`, który jest spełniony, jeżeli L jest listą `[N, ..., 0]`.

5. (3) Niech będzie dana następująca definicja predykatu `append`.

```
append([],L,L).  
append([X|L1],L2,[X|L3]) :- append(L1,L2,L3).
```

Proszę podać drzewo odpowiedzi na pytanie

```
?- append([1,2],[Z],[1,2,3]).
```