

Programowanie deklaratywne

Kolokwium
15 stycznia 2016

1. (2+2)

- Proszę napisać funkcję (`member x l`), której wartością jest `#t`, jeżeli `x` jest elementem listy `l` a `#f` w przeciwnym przypadku.
- Proszę powtórzyć część a) używając funkcję `fold` znana z laboratorium.

2. (2+2)

- Proszę napisać funkcję (`numbers n`), której wartością jest lista liczb od 0 do `n`. Przykład:

```
> (number 5)
'(0 1 2 3 4 5)
```

- Proszę napisać funkcję (`flatten l`), której wartością jest wersja płaska listy `l`. Przykład:

```
> (flatten1 '(1 ((2) 3 (4 (5))) 6))
'(1 2 3 4 5 6)
```

3. (2+3)

- Niech będą podane następujące definicje.

```
(define (comb f g) (lambda (x) (f (g x))))
(define (double n) (+ n n))
```

Proszę pokazać jak się ewaluuje wyrażenie `((comb double double) 10)`.

- Niech będą podane następujące definicje.

```
(define m 3)
(define (p m)
  (define (pp x)
    (set! m (* 2 m))
    (+ x m))
  (pp 3))
```

Używając model środowiska proszę pokazać jak się ewaluuje wyrażenie `(p 2)`. Jaka teraz jest wartość zmiennej `m`?

4. (2+2)

- Proszę zdefiniować predykat `delete(X,L1,L2)`, który jest spełniony, jeżeli `L2` jest listą `L1` bez wszystkich wystąpień elementu `X`. Przykład:

```
?- delete(2,[1,2,3,2],L).
L = [1,3]
```

b) Proszę zdefiniować predykat `nth(N,L,X)`, który jest spełniony, jeżeli `X` jest `N`-tym elementem listy `L`. Przykład:

```
?- nth(2, [1,3,5], X).  
X = 3
```

5. (3+2)

a) Niech będzie podana następująca definicja.

```
member(X, [X|L]).  
member(X, [Y|L]) :- member(Y,L).
```

Używając drzewo odpowiedzi proszę pokazać, jak Prolog odpowiada na pytanie `member(2, [1,2,2])`.

b) Niech teraz będzie podana następująca definicja.

```
memberc(X, [X|L]) :- !.  
memberc(X, [Y|L]) :- memberc(Y,L).
```

Czy `!` zmieni semantykę definicji `member` z części a)?