

Egzamin poprawkowy z programowania
część II

Zadanie 1. (15p) Funkcje `until` oraz `move` zdefiniowane są następująco:

```
fun until f p x = if (p x) then x else until f p (f x);
fun move (x::xs,ys) = (xs,x::ys);
fun len (x::xs) = 1 + len xs |
  len [] = 0;
fun second (x,y) = y;
```

- a) Podaj typy zdefiniowanych powyżej funkcji (2p).
- b) Napisz w SML-u funkcję `repeat n f x`, która oblicza $f^n(x)$, przy czym $f^0(x) = x$, a $f^{n+1}(x) = f(f^n(x))$ dla $n > 0$ (3p). Jaki typ ma ta funkcja? (1p).
- c) Napisz (nie używając jawnie rekurencji) funkcję `reverse` odwracającą listę za pomocą funkcji `repeat` (4p).
- d) Napisz (nie używając jawnie rekurencji) funkcję `reverse` za pomocą funkcji `until` (5p).

W punktach c) i d) możesz korzystać ze zdefiniowanych powyżej funkcji. W obu tych punktach możesz zdefiniować w sumie co najwyżej jedną funkcję pomocniczą.

Zadanie 2. (7p) Odpowiedz na następujące pytania:

- a) Jaki najogólniejszy typ ma w SML-u funkcja `f` zdefiniowana jako `fun f x y = (x y, (y,y))?`
- b) Jaki najogólniejszy typ ma w SML-u funkcja `f` zdefiniowana jako `fun f x y = [x=y]?`
- c) Jaki najogólniejszy typ ma w SML-u funkcja `f` zdefiniowana jako `fun f x y = x y::[y]?`
- d) Jaka (przykładowo) funkcja ma najogólniejszy typ równy `'a -> 'b -> 'b * 'a`?
- e) Jaka funkcja ma najogólniejszy typ równy `('a -> 'a -> 'b) -> 'a -> 'b`?
- f) Jaka funkcja ma najogólniejszy typ równy `('a -> 'a) -> 'a list -> 'a list`?
- g) Jaki jest najbardziej ogólny unifikator $h(X,Y,Z,V)$ oraz $h(f(Y),f(Z),f(V),f(a))$?

Zadanie 3. (9p) Zdefiniuj w Prologu następujące predykaty:

- a) (3p) `shuffle(L1,L2)` oznaczający, że lista `L2` powstała przez zamienienie elementów na pozycjach parzystych z elementami na pozycjach nieparzystych, przykładowo `shuffle([1,2, 3,4, 5,6, 7],[2,1, 4,3, 6,5, 7])`.
- b) (3p) `firsts(L1,L2)` prawdziwy, gdy lista `L2` jest listą pierwszych elementów z par składających się na listę `L1` przykładowo `firsts([(a,1), (b,5), (c,3)], [a,b,c])`.
- c) (3p) `subtree(Small,Big)` prawdziwy, gdy drzewo `Small` jest poddrzewem drzewa `Big`. Przykładowo `subtree(br(3,leaf,leaf), br(1,leaf,br(3,leaf,leaf)))`.

Zadanie 4. (10p) Zdefiniuj w SML-u funkcję `minus` przekształcającą listę liczb całkowitych na listę liczb przeciwnych (2p). Udowodnij (8p), że dla każdej listy liczb całkowitych `l`

```
rev (minus l) = minus (rev l)
```

Poniżej przypominam definicje potrzebnych predykatów:

```
fun [] @ xs = xs |
  (x::xs)@ys = x::(xs @ ys);
fun rev (x::xs) = (rev xs) @ [x] |
  rev [] = [];
```