

Imię i nazwisko	Data	Godzina
Paweł Kluska	3.11.2021	11:15

	1	2	3	4	5	6	7	8	9	10
Lista 3	V	Ocaml	V	V	V					
Lista 4	V	V	V	V	V					

Lista 3

1) a) `let f1 x = x 2 2;;`

Jest to funkcja jednoargumentowa zwracająca typ polimorficzny

Z kontekstu mogę odczytać że `x` jest funkcją przyjmującą 2 argumenty `Int` i zwracającą typ polimorficzny

`x: Int -> Int -> 'a`

Zatem całość ma typ

`f1: x -> 'a`

`f1 : (Int -> Int -> 'a) -> 'a`

b) `let f2 x y z = x (y ^ z);;`

Z kontekstu mogę odczytać że `y` i `z` są stringami (została użyta konkatencja stringów)

`x` jest funkcją przyjmującą `String` i zwracającą typ polimorficzny

`x: String -> 'a`

Zatem całość ma typ

`f2: x -> y -> z -> 'a`

`f2: (String -> 'a) -> String -> String -> 'a`

2) a) `let curry3 f x y z = f(x,y,z);;`

Z kontekstu mogę odczytać że `f` jest funkcją przyjmującą 3 elementową krotkę i zwracającą typ polimorficzny. Zatem

`f: x*y*z -> 'd`

`f: 'a*'b*'c -> 'd`

Całość ma typ

`Curry3: f -> x -> y -> z -> 'd`

`Curry3: ('a*'b*'c -> 'd) -> 'a->'b->'c -> 'd`

b) `let uncurry3 f(x,y,z) = f x y z;;`

Z kontekstu mogę odczytać że `f` jest funkcją przyjmującą 3 argumenty w postaci rozwiniętej zwracającą typ polimorficzny

`f: x->y->z->'d`

`f: 'a->'b->'c->'d`

Zatem całość ma postać

`f -> x*y*z -> 'd`

`('a->'b->'c->'d) -> 'a*'b*'c->d`

4) Jeżeli nastąpi przypadek, dla którego głowa listy będzie najmniejszym elementem listy, przestanie się ona dzielić i będą wywoływane kolejne wywołania rekurencyjne tej metody dla niezmiennionych list aż nastąpi przepełnienie stosu.

b) Jeżeli lista ma kilka takich samych elementów zostaną one pominięte i w posortowanej tablicy będzie tylko jeden element z jednego rodzaju.

Lista 4

1) a) let f1 x y z = x y z;;

Z kontekstu możemy odczytać że x jest funkcją dwu argumentową, z i y są wartościami polimorficznymi. A zatem

x: y -> z -> 'c

x: a' -> 'b -> 'c

Zatem całość ma postać

f1: x -> y -> z -> 'c

f1: (a' -> 'b -> 'c) -> 'a -> 'b -> 'c

b) let f2 x y = function z -> x::y;;

Z kontekstu można odczytać że x jest typem polimorficznym 'a, y jest listą z typami polimorficznymi 'a, wynikiem jest funkcja przyjmująca inny typ polimorficzny 'b i zwracająca listę typu polimorficznego 'b. A zatem

fw: 'b -> 'a list (fw – funkcja wynikowa)

f2: x -> y -> fw

f2: 'a -> 'a list -> 'b -> 'a list