

| Imię i nazwisko | Data | Godzina |
|-----------------|-----------|---------|
| Jakub Radzik | 3.11.2021 | 7:30 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------------------------|---|-------|---|---|---|---|---|---|---|----|
| nr listy: 3 zrobione | V | OCaml | V | | | | | | | |
| nr listy: 4 zrobione | V | V | V | V | V | | | | | |

Lista 3:

1.

a) Typ $\text{to } (\text{int} \rightarrow \text{int} \rightarrow 'a) \rightarrow 'a$ ponieważ definiujemy funkcję f_1 przyjmującą funkcję x która operuje na INT-ach co widać po prawej stronie równania. Stąd wiemy że x przyjmuje INT-a i zwraca funkcję przyjmującą INT-a która zwraca typ generyczny i to wszystko pakujemy do f_1 który zwraca wynik funkcji x czyli $'a$.

b) Typ $\text{to } (\text{string} \rightarrow 'a) \rightarrow \text{string} \rightarrow \text{string} \rightarrow 'a$ ponieważ f_2 przyjmuje funkcję x która przyjmuje jako parametr $y \wedge z$, z użycia operatora \wedge (konkatenacja stringów) wiemy że y, z to stringi i x przyjmuje stringa jako parametr lecz nie wiemy co zwraca, stąd zapis $(\text{string} \rightarrow 'a)$. Dalsza część typu bierze się z tego że y, z to parametry przyjmowane po kolei przez f_2 i zwracające inne funkcje a w wyniku daje to co funkcja x .

2.

a) $(\text{let } \text{curry3 } f \ a \ b \ c = f(a,b,c);;)$ przyjmuje funkcję f które operuje na parametrach a,b,c , potem przyjmuje parametr a i zwraca funkcję przyjmującą b która zwraca funkcję przyjmującą parametr c zwracającą wynik f od parametrów a,b,c . Stąd typ:

$('a * 'b * 'c \rightarrow 'd) \rightarrow 'a \rightarrow 'b \rightarrow 'c \rightarrow 'd$

b) $(\text{let } \text{uncurry } f \ (a,b,c) = f \ a \ b \ c;;)$ funkcja jest przeciwieństwem funkcji curry stąd jej typ:

$('a \rightarrow 'b \rightarrow 'c \rightarrow 'd) \rightarrow 'a * 'b * 'c \rightarrow 'd$

Lista 4:

1.

a) Z zapisu funkcji wnioskuję że nie jesteśmy w stanie określić typów dla x, y, z a tym typu zwracanego przez x . Dlatego f_2 przyjmuje i zwraca $(a' \rightarrow b' \rightarrow 'c)$. Więc jej typ to: $(a' \rightarrow b' \rightarrow 'c) \rightarrow a' \rightarrow b' \rightarrow 'c$.

b) Z zapisu $x::y$ wiemy że dochodzi do przyłączenia elementu x do listy y , ale nie znamy typów. Więc x ma typ $'a$, a y typ $'a \text{ list}$. Funkcja f_2 bierze x zwraca funkcję przyjmującą y która zwraca funkcję przyjmującą z która zwraca $x::y$. Z tego wynika że typ to: $'a \rightarrow 'a \text{ list} \rightarrow b \rightarrow 'a \text{ list}$. Nie znamy typu parametru z ani nie jest powiedziane że to ten sam typ co x .