

非手続き型言語 第13回課題

氏名:

学生番号:

2022年5月27日

1 型推論に関する問題集 問3.5

3. `fun A x y z = z y x`

`x` の型を `'a` と仮定する. `y` を `z` に適用し, その関数に `x` が適用されているから, `y` の型は `'b`, `z` の型は `'b -> 'a -> 'c`.

したがって, 全体の型は `'a -> 'b -> ('b -> 'a -> 'c) -> c`.

4. `fun B f g = f g g`

`g` の型を `'a` と仮定する. `g` を `f` に適用し, その関数に `g` が適用されているから, `f` の型は `'a -> 'a -> 'b`.

したがって, 全体の型は `('a -> 'a -> 'b) -> 'a -> 'b`.

2 型推論に関する問題集 問3.7

2. `fun f x y z = x (y z) : int`

`x (y z) : int` は `(x (y z)) : int` に等しい.

ここで, `z` の型を `'b` と仮定する. `z` を `y` に適用しているから, `y` の型は `'b -> 'a`. `(y z)` を `x` に適用しているから, `x` の型は `'a -> int`.

したがって, 全体の型は `('a -> int) -> ('b -> 'a) -> 'b -> int`.

3. `fun f x y z = (x y z) : int`

`z` の型を `'b` と仮定する. `y` を `x` に適用し, その関数に `z` が適用されているから, `y` の型は `'a`, `x` の型は `'a -> 'b -> int`.

したがって, 全体の型は `('a -> 'b -> int) -> 'a -> 'b -> int`.

4. `fun f x y z = x y (z : int)`

明らかに `z` の型は `int` である.

`y` を `x` に適用し, その関数に `z` が適用されているから, `y` の型は `'a`, `x` の型は `'a -> int -> 'b`.

したがって, 全体の型は `('a -> int -> 'b) -> 'a -> int -> 'b`.

5. `fun f x y z = x (y z : int)`

`z` の型を `'b` と仮定する. `z` を `y` に適用しているから, `y` の型は `'b -> int`. `y z` を `x` に適用しているから, `x` の型は `int -> 'a`.

したがって, 全体の型は `(int -> 'a) -> ('b -> int) -> 'b -> 'a`.

3 型推論に関する問題集 問3.11

1. `fn x => x > 1`

`x > 1` が含まれるから, `x` の型は `int`.

全体の型は `int -> bool`.

2. `fn x => fn y => fn z => (x y, x "Ada", y > z)`

"Ada" を `x` に適用しているから, `x` の型は `str -> 'a`. `y` を `x` に適用しているから, `y` の型は `str`. `y > z` が含まれるから, `z` の型は `str`.

全体の型は `(str -> 'a) -> str -> str -> 'a * 'a * bool`.

4 資料演習問題 5.6.4

```
fun comp F G =  
  let  
    fun C x = G(F(x))  
  in  
    C  
  end;
```

```
fun add1 x = x + 1;
```

このとき, `comp` の型は `('a -> 'b) -> ('b -> 'c) -> 'a -> 'c`, `add1` の型は `int -> int`.

a) `val compA1 = comp add1;`

`comp` の型は `('a -> 'b) -> ('b -> 'c) -> 'a -> 'c` であるから, `compA1` の型は `(int -> 'a) -> int -> 'a`.

b) `val compCompA1 = comp compA1;`

`comp` の型は `('a -> 'b) -> ('b -> 'c) -> 'a -> 'c` であるから, `compCompA1` の型は `((int -> 'a) -> 'b) -> (int -> 'a) -> 'b`.

c) `val f = compA1 add1;`

`compA1` の型は `(int -> 'a) -> int -> 'a` であるから, `f` の型は `int -> int`.

d) `f(2);`

```
f(2) = compA1 add1 2  
      = comp add1 add1 2  
      = add1 (add1 2)  
      = add1 3  
      = 4.
```

e) `val g = compCompA1 compA1;`

`compCompA1` の型は `((int -> 'a) -> 'b) -> (int -> 'a) -> 'b`,
`compA1` の型は `(int -> 'a) -> int -> 'a` であるから, `g` の型は `(int -> 'a) -> int -> 'a`.

f) `val h = g add1;`

`g` の型は `(int -> 'a) -> int -> 'a` であるから, `h` の型は `int -> int`.

g) `h(2);`

```
h(2) = g add1 2  
      = compCompA1 compA1 add1 2  
      = comp compA1 compA1 add1 2  
      = comp (comp add1) (comp add1) add1 2  
      = comp add1 (comp add1 add1) 2  
      = comp add1 add1 (add1 2)
```

```
= add1 (add1 (add1 2))  
= add1 (add1 3)  
= add1 4  
= 5.
```