

# 第 1 2 回課題

---

09430509 今田将也

## 3.2.3

---

関数宣言 `fun f(a:int, b, c, d, e) = ...`

### a) if a<b+c then d else e

aがintで指定されていて、また、そのaを用いた比較にbとcが利用されていることから、a:int,b:int,c:intである。また、if文の後のthenとelseはifと同じ型でないといけないが、ifの型は指定されていないため、d,cともに'a型であるだろう。したがって、`val f = fn : int * int * int * 'a * 'a -> 'a`

### b) if a<b then c else d

a)と同様に考えると、a,bはint型、c,dは'a型。そして、eは利用されておらず、特に型指定もないため新たな'b型である。したがって、`val f = fn : int * int * 'a * 'a * 'b -> 'a`

### c) if a<b then b+c else d+e

同様に、a,bはint型が決まる。そして、thenの後にそのbが使われていることから、cもint型であることが決まる。そして、thenとelseは同じ型であるというルールから、d,eも共にint型である。したがって、`val f = fn : int * int * int * int * int -> int`

### d) if a<b then b<c else d

同様に、a,b,cはint型。dはthenのところで、bとcを比較した真偽値を持つため、bool型であるだろう。eは'a型。したがって、`val f = fn : int * int * int * bool * 'a -> bool`

### e) if b<c then a else c+d

aがint型であるから、c,dもint型だということがわかる。cがint型であるから、ifでの比較に利用しているbもintだと推論でき、eは'aだと推論できる。したがって、`val f = fn : int * int * int * int * 'a -> int`

### f) if b<c then d else e

b,cは比較に用いられているが、特に指定がないため、int型だろう。aはintと指定されているため、自明である。d,eはthen,elseに使われているが、特に指定がないため共に、'a型である。したがって、`val f = fn : int * int * int * 'a * 'a -> 'a`

### g) if b<c then d+e else d\*e

aはintと指定されているため、自明である。b,cは比較に用いられているが、特に指定がないため、int型だろう。d,eについては計算が行われているため、bool型および'a型であることはない。よって標準のint型だと推論される。したがって、`val f = fn : int * int * int * int * int -> int`

## 資料MLOO.pdf 問3.3の2と3

### 2

まず, `fun cube`について推定すると, 計算をしているため`int`型で引数をとると考えられる. `fun twice f x`についてだが, 引数`f`と`x`をそれぞれ用いた再帰的な関数であるが, 計算はこの関数自体で見ると行っていないので, '`a`'型で引数`x`を受け取り, `f`はそれに対応する関数を受け取り, 最終的には再帰的に'`a`'型を返すものと推定できる. 次に`fun id`は受け取った引数`x`を単純に返しているだけなので, '`a`'型と推定できる.

では, `fn x => twice id x`はというと, `twice id x`の`id`は'`a`'型で, 引数を受け取るので, `x`も'`a`'型である必要がある. そして, `fun twice`は'`a`'で引数`x`と関数`id`を受け取るはずと推定できる.

したがって最終的な結果は'`a`->'`a`'と推定できる.

### 実際の結果

```
- fun cube x = x * x * x;
val cube = fn : int -> int
- fun twice f x = f (f x);
val twice = fn : ('a -> 'a) -> 'a -> 'a
- fun id x = x;
val id = fn : 'a -> 'a
- fn x => twice id x;
val it = fn : 'a -> 'a
```

### 3

関数 `thrice`について, `thrice : t1`, `f : t2`, `x : t3`, `f x : t4`, `f (f x) : t5`, `f f (f x) : t6`, と仮定する.

このとき求めたいのは`t1 = t2 -> t3 -> t6`である. `t3`を引数にすると, `t2`で`t4`を得られるため, `t2 = t3 -> t4`, また引数を変えて同様に, `t2 = t4 -> t5`, `t2 = t5 -> t6`

これより`t3 = t4 = t5 = t6`

したがって`t1`は

```
t1 = t2 -> t3 -> t3 = (t3 -> t3) -> t3 -> t3 = ('a -> 'a) -> 'a -> 'a
```

## 資料MLOO.pdf 問3.5 の 1 fun S x y z = (x z) (y z);

カリー化されたラムダ計算式であることがわかる. 関数`S`は1つ目の関数に3つ目の関数を適用し, その結果に, 2つ目の関数に3つ目の関数を適用した結果を適用している`S = λxyz.(xz)(yz)`である.

`x -> y(z), y(z) -> z, z -> (x z)(y z)`であることがわかる.

- `S : t1`

- $x : t2$
- $y : t3$
- $z : t4$
- $(x\ z) : t5$
- $(y\ z) : t6$
- $(x\ z)(y\ z) : t7$

求めたいのは、関数Sだから、 $t1 = t2 \rightarrow t3 \rightarrow t4 \rightarrow t7$ . 関数xは引数にzで、結果がt5の関数と同様だと捉えられるので、 $t2 = t4 \rightarrow t5$ といえる. 関数yも同様に、引数がzで結果がt6の関数と同様なので  $t3 = t4 \rightarrow t6$ である.

整理すると、

$$t1 = (t4 \rightarrow t5) \rightarrow (t4 \rightarrow t6) \rightarrow t4 \rightarrow t7$$

また、t5は引数がt6である. 結果はt4つまりt7であるから、 $t5 = t6 \rightarrow t7$

したがって、

$$t1 = (t4 \rightarrow t6 \rightarrow t7) \rightarrow (t4 \rightarrow t6) \rightarrow t4 \rightarrow t7$$

$t4 = 'a$ ,  $t6 = 'b$ ,  $t7 = 'c$ とすると、

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