

## IV

通常の方法で起動した状態の `ocaml` において、次の式の値は何になるか。ただし、何らかのエラーになったり評価が停止しないときは `x` と書け。

1. `let x =  
    let y = 1 + 2 in  
    y + 3 in  
x * y`

答: `x`

2. `let x = 7 in  
let f y = x + y in  
let x = 3 in  
f x`

答: `10`

3. `let f g x = g (g x) in  
f (fun y -> y * 2) 3`

答: `12`

以下では、あらかじめ

```
type 'a tree = Leaf of 'a | Node of 'a tree * 'a tree
```

と定義されていることにする。

4. `Node(Leaf (12 + 3), Leaf 4.5)`

答: `x`

5. `let rec f t =  
    match t with  
    | Leaf x -> x  
    | Node(l, r) -> f r ^ " " ^ f l in  
f (Node(Node(Leaf "this", Leaf "a"),  
        Node(Leaf "is", Leaf "pen")))`

答: `"pen is a this"`

授業で定義した純粋な 計算において、次の式の簡約結果を、1 ステップごとの簡約過程と共に示せ。let 式・自然数・論理値などは、授業で定義した  $\lambda$  式による表現を使用せよ。

例.  $1 + 1$

答:

$$\begin{aligned}
 1 + 1 &= \lambda s. \lambda z. 1s(1sz) \\
 &= \lambda s. \lambda z. 1s((\lambda s. \lambda z. sz)sz) \\
 &\rightarrow \lambda s. \lambda z. 1s((\lambda z. sz)z) \\
 &\rightarrow \lambda s. \lambda z. 1s(sz) \\
 &= \lambda s. \lambda z. (\lambda s. \lambda z. sz)s(sz) \\
 &\rightarrow \lambda s. \lambda z. (\lambda z. sz)(sz) \\
 &\rightarrow \lambda s. \lambda z. s(sz)
 \end{aligned}$$

1.  $(\lambda x. \lambda y. xy)y$

答の例:

$$\begin{aligned}
 (\lambda x. \lambda y. xy)y &= (\lambda x. \lambda y'. xy')y \\
 &\rightarrow \lambda y'. yy'
 \end{aligned}$$

2.  $(\lambda a. a)(\lambda b. b)((\lambda c. c)(\lambda d. d))$

答の例:

$$\begin{aligned}
 (\lambda a. a)(\lambda b. b)((\lambda c. c)(\lambda d. d)) &\rightarrow (\lambda b. b)((\lambda c. c)(\lambda d. d)) \\
 &\rightarrow (\lambda b. b)(\lambda d. d) \\
 &\rightarrow \lambda d. d
 \end{aligned}$$

3. `let is_zero =  $\lambda n. n(\lambda m. \text{false})\text{true}$  in is_zero 2`

答の例:

$$\begin{aligned}
 \text{let is\_zero} = \lambda n. n(\lambda m. \text{false})\text{true} \text{ in is\_zero } 2 &= (\lambda \text{is\_zero}. \text{is\_zero } 2)(\lambda n. n(\lambda m. \text{false})\text{true}) \\
 &\rightarrow (\lambda n. n(\lambda m. \text{false})\text{true})2 \\
 &\rightarrow 2(\lambda m. \text{false})\text{true} \\
 &= (\lambda s. \lambda z. s(sz))(\lambda m. \text{false})\text{true} \\
 &\rightarrow (\lambda z. (\lambda m. \text{false})((\lambda m. \text{false})z))\text{true} \\
 &\rightarrow (\lambda m. \text{false})((\lambda m. \text{false})\text{true}) \\
 &\rightarrow \text{false}
 \end{aligned}$$

## 講評・解説

- IV の 1. を 18 とした者がやや多かったが、 $x * y$  の  $y$  が `let y = 1 + 2 in ...` のスコープ外にあるので、エラーとなる。
- IV の 2. はよくできていた。
- IV の 3. を  $x$  とした者はレジюмеで「高階関数」の節をよくみること。
- IV の 4. はよくできていた。
- IV の 5. で二重引用符 `"` を忘れた者が極めて多かったが、ML であろうと C であろうと、ほとんどのプログラミング言語において  $x$  と `"x"` の意味はまったくことなるので、誤答とした。
- V の 1. はよくできていた。
- V の 2. もよくできていたが、

$$(\lambda a.a)(\lambda b.b)((\lambda c.c)(\lambda d.d)) \rightarrow (\lambda a.a)((\lambda c.c)(\lambda d.d))$$

などと簡約した答案がいくつかあった。しかし、関数適用は左結合的だから

$$(\lambda a.a)(\lambda b.b)((\lambda c.c)(\lambda d.d))$$

は

$$((\lambda a.a)(\lambda b.b))((\lambda c.c)(\lambda d.d))$$

であって

$$(\lambda a.a)((\lambda b.b)((\lambda c.c)(\lambda d.d)))$$

ではないので、 $(\lambda b.b$  を  $\lambda a.a$  に  $\alpha$  変換しない限り) そのような簡約はできない。よって、 $\alpha$  変換したことがわかるように書かれていなければ誤答とした。

- V の 3. は思ったよりできていたが、代入で括弧をつけ忘れた者や、 $e_1 e_2 e_3$  という形の式を  $e_1(e_2 e_3)$  と間違えた者が多かった。