

# ソフトウェア基礎論 レポート課題 (2)

(締切: 12/6 の講義)

五十嵐 淳

京都大学 大学院情報学研究科知能情報学専攻

e-mail: igarashi@kuis.kyoto-u.ac.jp

平成 17 年 11 月 22 日

課題 1: 形無し  $\lambda$  計算 (ゲーム名 Untyped) に関する以下の問からふたつ以上選んで解答せよ .

1. 既に見たように , 以下で定義される  $\text{fixY}$  という  $\lambda$  項

$$\text{fixY} \equiv \lambda f. ((\lambda x. f (x x)) (\lambda x. f (x x)))$$

を (項構成子である)  $\text{fix}$  の代わりに使用することで , 再帰関数が表現できる . しかし , eager な評価戦略の元で  $\text{fixY}$  を  $\text{fix}$  の代わりに使用することはできない . 例えば乗算を行う関数

$$\text{fix } (\lambda \text{mult}. \lambda m. \lambda n. \text{if eq } m \ 0 \ \text{then } 0 \ \text{else } n - (0 - \text{mult } (m - 1) \ n))$$

と

$$\text{fixY } (\lambda \text{mult}. \lambda m. \lambda n. \text{if eq } m \ 0 \ \text{then } 0 \ \text{else } n - (0 - \text{mult } (m - 1) \ n))$$

を 3 に適用し , eager 戦略で簡約を行った時の様子を観察し ,  $\text{fixY}$  が何故うまく働かないかを答えよ . ( $\text{eq}$  の定義は

$$\text{eq} \equiv \lambda x. \lambda y. \text{if } x > y \ \text{then } \text{false} \ \text{else if } y > x \ \text{then } \text{false} \ \text{else } \text{true}$$

とする .)

2. eager 戦略でも働くように  $\text{fixY}$  の定義を改良せよ . (もちろん  $\text{fix}$  を使ってはならない .)

3. 以下の関数 `ifzero` は Church 数  $n$  とふたつの  $\lambda$  項  $t_1, t_2$  を引数として,  $n$  が (Church 数の) 0 ならば  $t_1$  を返し, さもなければ  $t_2$  を返す. この動作を説明せよ.

$$\text{ifzero} \equiv \lambda n. \lambda t_1. \lambda t_2. n (\lambda x. t_2) t_1$$

4. Church numeral 上の指数関数を表現する  $\lambda$  項を変数,  $\lambda$  抽象, 関数適用のみを用いて (つまり純粋な  $\lambda$  計算の項として) 定義し, 指数関数を表現していることを説明せよ.

## 課題 2: 単純型付 $\lambda$ 計算 (ゲーム名 Typed) に関する以下の問に答えよ.

1. 型付け判断  $\bullet \vdash \lambda f : \text{int} \rightarrow \text{int}. \lambda f : \text{int} \rightarrow \text{bool}. \lambda x : \text{int}. \#^0 f (\#^1 f x)$  の導出を書け.
2. 以下の型付け判断

$$\text{eq} : \text{int} \rightarrow \text{int} \rightarrow \text{bool}, \text{mult} : \text{int} \rightarrow \text{int} \rightarrow \text{int} \vdash \\ \text{fix } (\lambda f : \alpha. \lambda n : \beta. \text{if eq } n \ 0 \ \text{then } 1 \ \text{else } \text{mult } n \ * \ f \ (n - 1))$$

の中の  $\alpha, \beta$  に判断が成立するように, 適当な型をあてはめて, 導出を完成させよ.