

# 計算機科学実験IV

1029-28-9483 勝田 峻太郎

2018 年 10 月 9 日

## 1 課題 1 (拡張構文: 任意)

階乗計算を行う MiniML プログラムを, 再帰を用いず loop 構文と組を使って書きなさい. 同じく, フィボナッチ数を求めるプログラムを, loop 構文と組を使って書きなさい.

### 1.1 階乗計算

```
(* calculate factorial *)  
let fact n =  
  loop v = (n, 1) in  
    if v.1 > 1 then  
      recur (v.1 - 1, v.2 * v.1)  
    else v.2
```

### 1.2 フィボナッチ数

```
(* calculate the fibonacci number *)  
let fib n =  
  loop v = (n, (1, 0)) in  
    if v.1 > 1 then  
      let tmp1 = v.2.1 in  
      let tmp2 = v.2.1 + v.2.2 in  
      recur (v.1 - 1, (tmp2, tmp1))  
    else v.2.1 + v.2.2
```

## 2 課題 2 (フロントエンド: 必須)

MiniML の文法規則に従う MiniML プログラムを入力とし, 以下の `syntax.ml` により定義される抽象構文木を返す字句解析器・構文解析器を作成しなさい.

## 3 課題 3 (インタプリタ・型推論: 任意)

実験 3 で作成した  $ML^4$  言語のインタプリタと型推論器を基に, MiniML 言語のインタプリタと型推論器を作成しなさい.

## 4 課題 4 (recur 式の検査: 必須)

`syntax.ml` 中の `recur_check` 関数を完成させることにより, `recur` 式の検査を実装しなさい. `parser.mly` 中の呼び出している箇所を見ると分かるとおり, `recur_check` 関数は `unit` 型の値を返す. 末尾位置ではないところに書かれた `recur` 式を発見したら, 即座に例外を投げコンパイル処理を中断すること.

## 5 課題 5 (正規形への変換: 必須)

言語 C への変換と, 正規形への変換を同時に行う, `normal.ml` 中の `norm_exp` 関数を完成させよ. 関数は次に示す形で実装すること. 引数 `f` を適切に用いれば各場合分けで数行書くだけで完成する.

## 6 課題 6 (クロージャ変換: 必須)

`closure.ml` の `convert` 関数を完成させることにより, クロージャ変換を実装しなさい.