

「アルゴリズムとデータ構造入門」

第2回課題

工学部情報学科
平成 25 年入学
学籍番号：1029-25-2723
森井 崇斗

October 22, 2013

1 階上のプログラムの作成と実行

1.1 プログラムの内容

```
1 (define (fact n)
2   (if (<= n 0 )
3       1
4       (* n (fact (- n 1)))))
5 )
6 )
```

1.2 出力結果

```
> (fact 103)
9902900716486180407546715254581773349090165822114492483005280554699876665841
6222832141441073883538492653516385977292093222882134415149891584000000000000
000000000000
```

1.3 説明

1. define で (if ~) をグローバル変数 fact に定義し、またその引数として受け取った値を変数 n に格納します。
2. 2 行目以降の条件構文では n が 0 以下ならば 1 を返し、それ以外なら n と “fact を n-1 を引数とした際の結果 “の乗を返します。

3. この部分は再帰的になっているので、 $n = 1$ になるまで 2. を繰り返します。

つまり、`(fact n)` は `(fact n-1)` の結果に n を掛けたものを結果とする。この時利用される `(fact n-1)` の結果は $n-1$ に `(fact n-2)` の結果を掛けたものになる。… というように繰り返し、`(fact 1)` の結果として 1 と `(fact 0)` の結果を必要とする処理に到達する。この時実行される `(fact 0)` は再帰せずに 1 を結果として返す。

ここから逆に辿って行くと $n(n-1)(n-2)\cdots 2\cdot 1$ という式と同じ計算が行われ、結果として階乗の値が導き出されます。

2 教科書 1-1-4 ~ 1-2-2 の想定質問とその回答

2.1 1-1-4 より

2.1.1 想定質問

`define` は合成手続きに名前を付けると説明されているが、名前は自由に命名することが可能であるのか？

2.1.2 想定質問への回答

Scheme には他のプログラミング言語でいうところの予約語 (*reserved keywords*) は存在しません。

つまり、`define` や `if` などを `define` で再定義することも許される。

例えば次のようなプログラムを作成実行することも可能です。

プログラム例

```
(define
  (define x y)
    (* x y)
  )
```

実行例

```
>(define 3 5)
15
```