

# プログラミング基礎演習第二回レポート

長谷川 禎彦

## 1 注意点

レポートは以下の2つを両方提出すること

- プログラムのソースコード (C 言語)
- レポート本体 (doc, docx, pdf, odt)

注意点は第一回と同じなため、第一回のPDFを参照せよ。なお、英語で書いてもよい (We accept documents written in English)。

## 2 課題

課題は 1,2 である。1 が出来ればレポート提出扱いになる。

### 1

文字列で表された関数を木構造にするプログラムを書け。つまり

`Plus[Times[Sin[13.4], 3], 2]`

のような定数に関する関数に対して、図 1(a) のような木構造を作る (第 11 回の課題を参照)。入力はコマンドライン引数や標準入力など外部から入力する (要はソースに書き込まない)。さらにこの木構造を評価することで、関数の値を計算せよ。なお、考える関数は Plus, Times, Subtract, Divide, Sin, Cos とせよ (関数の意味は文字通りである)。第 11 回の小課題のソースを流用して良い。なお、下の例の関数は WolframAlpha でそのまま計算できるので簡単に検算可能。

例

```
$ ./kadai01 Plus[Sin[3.4],Times[Cos[4.1],8]]
```

```
Plus[Sin[3.4],Times[Cos[4.1],8]] = -4.85413
```

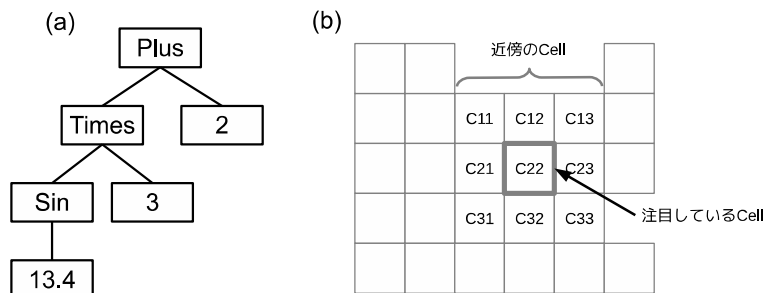


図 1: (a) 木構造. (b) ライフゲームの変数の位置関係.

```

If[C22, If[Or[Eq[Plus[C11, Plus[C12,
    Plus[C13, Plus[C21, Plus[C23, Plus[C31, Plus[C32, C33]]]]]], 3], Eq[Plus[
    C11, Plus[C12, Plus[C13, Plus[C21, Plus[C23, Plus[C31, Plus[C32, C33]]]]]],
2]], 1, 0], If[Eq[Plus[C11, Plus[C12,
    Plus[C13, Plus[C21, Plus[C23, Plus[C31, Plus[C32, C33]]]]]], 3], 1, 0]]

```

図 2: 23/3 の遷移ルールを表す関数。本来は一行の関数だが、長いので改行を入れてある。

関数	意味
Plus[ $x, y$ ]	$x + y$
Times[ $x, y$ ]	$x \times y$
Subtract[ $x, y$ ]	$x - y$
Divide[ $x, y$ ]	$x / y$
Eq[ $x, y$ ]	$x = y$ ならば 1, それ以外は 0 を返す
If[ $x, y, z$ ]	$x > 0$ なら $y$ の値を返す, それ以外は $z$ の値を返す
Or[ $x, y$ ]	$x > 0$ または $y > 0$ なら 1, それ以外は 0 を返す
And[ $x, y$ ]	$x > 0$ かつ $y > 0$ なら 1, それ以外は 0 を返す
Not[ $x$ ]	$x > 0$ なら 0, それ以外は 1 を返す

表 1: ライフゲームの遷移ルールを表す関数に用いる関数一覧。

## 2

プログラマブルなライフゲームを作成せよ（ライフゲームについては Wikipedia やソフトウェア 2 のスライドを参照）。通常のライフゲームでは遷移ルールはソースコードに書き込むため、一度コンパイルすると遷移ルールを書き換えることは出来ない。そこで、rule.txt に遷移ルールを書き、プログラムはそれを読み込むことでプログラムを再コンパイルすることなく遷移ルールを変更できるようにする。

セル  $(i, j)$  の状態遷移（alive と dead の変化）は近傍の 8 個のセルと、自分自身のセルの状態によって決まる（図 1(b)）。そこでこのルールを関数で表す。遷移ルールを記述するために用いる関数一覧を表 1 に示す。また用いる変数は 9 個であり、これは近傍の 8 個と自分自身の 1 個でその位置関係は図 1(b) に示す通りである（この変数は相対位置である。C22 は注目している  $(i, j)$  のセルの値を表す）。この変数は alive では 1, dead では 0 の値をとるようにする。例えば、良く用いられる

- 対象のセルが alive の時、周りに alive のセルが 2 か 3 個であれば次の時間でも alive。それ以外は dead となる。
- 対象のセルが dead の時、周りに alive のセルが 3 個であれば次の時間でも alive。それ以外は dead となる。

のルールは図 2 の関数によって表すことが出来る（連なる Plus は周りの alive なセルを数えている）。このルールは 23/3 と呼ばれる（最初の数字が、alive の時に次の時間で alive になるのに必要な alive 近傍の数、後の数字が dead の時、次の時間で alive になるのに必要な alive 近傍の数を表す）。23/36 など知られている。つまり、作成するプログラムは rule.txt に書かれた遷移ルールの関数を読み込み、この関数に相当する木構造を作り、この木構造を評価することでセルの状態遷移を計算する。

出力は画面への表示でも良いし、テキストに出力してもよい。初期配置は乱数による初期化でも、ファイルからの読み込みでも良い。境界条件やエリアのサイズは適当に設定せよ。ソフトウェア 2 のソースを流用して良い。また、変数や関数もよりよい表現方法があれば改変して良い（新しい関数の導入など。例えば、もっと一般的なルールを記述できるように改良できる）。

なお、全ての問題において大枠として題意を満たしていれば、入出力方法・内容とも自由に変更して良い。

### 3 提出方法

作成したソースファイル（ファイル名は自由，一つのプログラムを複数ファイルに分割しても良い）とレポート本体（doc, docx, pdf, odt 等）を「学籍番号.zip」としてまとめる．作成した zip ファイルをホームページの提出フォームから提出する（「ソースコード」と書かれた所から，ソースとレポート本体の zip ファイルを添付する）．その際，課題の選択を「レポート」とすること（「小課題」ではない）．

**締切：2015 年 02 月 17 日（火曜日）23:59**