

ライフゲームを作ろう

宇佐美雅紀

2013 年 8 月 6 日

Haskell の練習として、ライフゲームを作成します。

1 ライフゲームとは

ライフゲームは、生命の誕生、進化、淘汰などのプロセスを簡易的なモデルで再現したシミュレーションゲームであり、セル・オートマトンの一種です。詳しくは、Wikipedia でライフゲームを検索してみてください。

(<http://ja.wikipedia.org/wiki/%E3%83%A9%E3%82%A4%E3%83%95%E3%82%B2%E3%83%BC%E3%83%A0>)

2 ライフゲームのルール

誕生

死んでいるセルに隣接する生きたセルがちょうど 3 つあれば、次の世代が誕生する。

生存

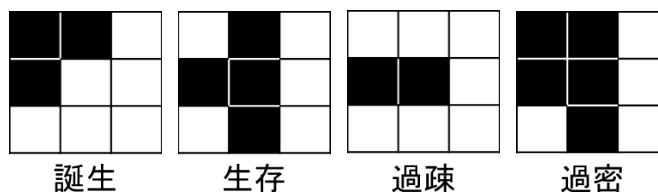
生きているセルに隣接する生きたセルが 2 つか 3 つ以下ならば、次の世代でも生存する。

過疎

生きているセルに隣接する生きたセルが 1 つ以下ならば、過疎により死滅する。

過密

生きているセルに隣接する生きたセルが 4 つ以上ならば、過密により死滅する。



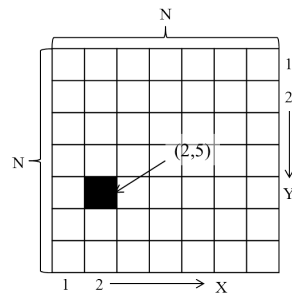
3 ライフゲームのデータ構造

ライフゲームは任意のサイズ ($N \times M$) の格子状のゲーム盤上で動作します。ここでは、簡単のために格子のサイズは縦横同じとします ($N \times N$)。格子の各マスは、生きているか死んでいるかのどちらかです。

レコード構文を使って、以下のように定義します。

```
data Lifegame = Lifegame { size :: Int
                           , life :: [(Int, Int)]
                           }
```

life はタプルのリストで、ゲーム盤上で生きているセルがいる場所を示します。タプル (ダブル) はゲーム盤上の位置 (x 座標、 y 座標) を示します。タプルが指示していない位置は、生きているセルがありません。



ここで図が欲しいな