

1 ライフゲーム

ライフゲームは、生物コロニーのシミュレーションと解釈することが可能な、興味深いセルオートマトンです。

ライフゲームは次のようなルールによって規定された2次元セルオートマトンです。ただし、セルの状態は0または1の2つの状態のいずれかをとりません。

ライフゲームのルール

1. 時刻 t_k において、セル c_{ij} の周囲8セルの状態1の総和 $s_{ij}^{t_k}$ が3ならば、次の時刻 t_{k+1} におけるセル c_{ij} の状態 $a_{ij}^{t_{k+1}}$ は1
2. 時刻 t_k において、セル c_{ij} の周囲8セルの状態1の総和 $s_{ij}^{t_k}$ が2ならば、次の時刻 t_{k+1} におけるセル c_{ij} の状態 $a_{ij}^{t_{k+1}}$ は変化なし ($a_{ij}^{t_k} = a_{ij}^{t_{k+1}}$)
3. 上記以外の場合、次の時刻 t_{k+1} におけるセル c_{ij} の状態 $a_{ij}^{t_{k+1}}$ は0

Cによる数値計算とシミュレーション 小高 知宏 オーム社 100頁より

2 ライフゲーム life.c

```
/*
   life.c
   ライフゲーム計算プログラム
   セルオートマトンの一種、ライフゲームのプログラムです
   使い方 ./life <(初期状態ファイル名)
   出力は行列形式です。
   初期状態ファイルに初期配置座標を2行一緒に記述します
   (x,y)の順に記述します
*/

#include <stdio.h>
#include <stdlib.h>

#define N 50
#define MAXT 50
#define BUFSIZE 256

void initworld(int world[][N]);
void putworld(int world[][N]);
void nextt(int world[][N]);
int calcnext(int world[][N], int i, int j);

int main(int argc, char *argv[])
{
    int t;
    int world[N][N] = {0};
```

```

    initworld(world);
    printf("t = 0\n");
    putworld(world);

    for(t = 1; t < MAXT; ++t){
        nextt(world);
        printf("t = %d\n", t);
        putworld(world);
    }

    return 0;
}

void nextt(int world[][N])
{
    int nextworld[N][N] = {0};
    int i,j;

    for(i = 1; i < N-1; ++i){
        for(j = 1; j < N-1; ++j){
            nextworld[i][j] = calcnext(world, i, j);
        }
    }

    for(i = 0; i < N; ++i){
        for(j = 0; j < N; ++j){
            world[i][j] = nextworld[i][j];
        }
    }
}

int calcnext(int world[][N], int i, int j)
{
    int no_of_one = 0;
    int x,y;

    for(x = i-1; x <= i+1; ++x){
        for(y = j-1; y <= j+1; ++y){
            no_of_one += world[x][y];
        }
    }

    no_of_one -= world[i][j];

    if(no_of_one == 3){
        return 1;
    }else{
        if(no_of_one == 2){

```

```

        return world[i][j];
    }else{
        return 0;
    }
}
}

void putworld(int world[][N])
{
    int i,j;

    for(j = N-1; j >= 0; --j){
        for(i = 0; i < N; ++i){
            printf("%ld", world[i][j]);
        }
        printf("\n");
    }
}

void initworld(int world[][N])
{
    char linebuf[BUFSIZE];
    int x,y;

    while(fgets(linebuf, BUFSIZE, stdin) != NULL){
        if(sscanf(linebuf, "%d", &x) < 1) break;
        if(fgets(linebuf, BUFSIZE, stdin) == NULL) break;
        if(sscanf(linebuf, "%d", &y) < 1) break;
        if((x >= 0) && (x < N) && (y < N)) world[x][y] = 1;
    }
}
}

```

3 初期値

2行で一組の座標を表す。この例の場合、(1,46)(2,46)(2,48),(3,46),(3,47)の5点に生物を配置することを意味する。

```

1
46
2
46
2
48
3
46
3
47

```



```

void fputworld(int world[] [N]);
void nextt(int world[] [N]);
int calcnext(int world[] [N], int i, int j);

int main(int argc, char *argv[])
{
    int t;
    int world[N] [N] = {0};
    FILE *pipe;

    initworld(world);
    printf("t = 0\n");
    fputworld(world);

    if((pipe = popen(GNUPLOT " -persist","w")) == NULL){
        fprintf(stderr, "パイプが開けません。");
        exit(1);
    }

    fprintf(pipe, "set xrange [0:%d]\n", N);
    fprintf(pipe, "set yrange [0:%d]\n", N);

    for(t = 1; t < MAXT; ++t){
        nextt(world);
        printf("t = %d\n", t);
        fputworld(world);
        fprintf(pipe, "plot \"\" TMPFILE \"\" with point pt 4\n");
        fflush(pipe);
        usleep(100000);
    }

    return 0;
}

void nextt(int world[] [N])
{
    int nextworld[N] [N] = {0};
    int i,j;

    for(i = 1; i < N-1; ++i){
        for(j = 1; j < N-1; ++j){
            nextworld[i] [j] = calcnext(world, i, j);
        }
    }

    for(i = 0; i < N; ++i){
        for(j = 0; j < N; ++j){
            world[i] [j] = nextworld[i] [j];
        }
    }
}

```

```

    }
}

int calcnext(int world[][N], int i, int j)
{
    int no_of_one = 0;
    int x,y;

    for(x = i-1; x <= i+1; ++x){
        for(y = j-1; y <= j+1; ++y){
            no_of_one += world[x][y];
        }
    }

    no_of_one -= world[i][j];

    if(no_of_one == 3){
        return 1;
    }else{
        if(no_of_one == 2){
            return world[i][j];
        }else{
            return 0;
        }
    }
}

void fputworld(int world[][N])
{
    int i,j;
    FILE *fp;

    if((fp = fopen(TMPFILE, "w")) == NULL){
        fprintf(stderr, "一時ファイルが開けません。 \n");
        exit(1);
    }

    for(i = 0; i < N; ++i){
        for(j = 0; j < N; ++j){
            if(world[i][j] == 1){
                fprintf(fp, "%d %d\n", i, j);
            }
        }
    }
    fclose(fp);
}

void initworld(int world[][N])
{

```

```
char linebuf[BUFSIZE];
int x,y;

while(fgets(linebuf, BUFSIZE, stdin) != NULL){
    if(sscanf(linebuf, "%d", &x) < 1) break;
    if(fgets(linebuf, BUFSIZE, stdin) == NULL) break;
    if(sscanf(linebuf, "%d", &y) < 1) break;
    if((x >= 0) && ( x < N) && (y < N)) world[x][y] = 1;
}
}
```

7 「glife」の操作方法

「glife」の操作方法

```
./glife < lifeinit1.txt
```

上記の「./」は、Linuxでのプログラムを実行する時の例です。
ウィンドウズでは、glife < lifeinit1.txt 「Enter」と入力してください。

8 「glife」の実行結果

別紙を参照してください。