セルオートマトンの計算プログラム

```
0001: /*
0002:   ca1.c
0003:   yukio sugawa
0004:   2012/2/18
0005:
0006:   セルオートマトン (1 次元) 計算プログラム
0007:   ルールと初期状態から、時間発展を計算します。
0008:   使い方  $./ca1 (ルール番号) < (初期状態ファイル名)
0009: */
0010:
0011: #include <stdio.h>
0012: #include <stdlib.h>
0013: #include <string.h>
0014:
0015: #define N       65
0016: #define R       8
0017: #define MAXT    50
0018: #define BUFSIZE 256
0019:
0020: void setrule(char *arg, int rule[]);
0021: void initca(int ca[]);
0022: int cvalue(char ch);
0023: void putca(int ca[]);
0024: void nextt(int ca[], int rule[]);
0025:
0026: int main(int argc, char *argv[])
0027: {
0028:   int t;
0029:   int ca[N] = {0};
0030:   int rule[R];
0031:
0032:   if(argc < 2){
0033:     fprintf(stderr, "使い方  ca1 (ルール番号) < (初期
状態ファイル名)\n");
0034:     exit(1);
0035:   }
0036:   setrule(argv[1], rule);
0037:
0038:   initca(ca);
0039:   putca(ca);
0040:
0041:   for(t = 0; t < MAXT; ++t){
0042:     nextt(ca,rule);
0043:     putca(ca);
0044:   }
0045:
0046:   return 0;
0047: }
```

```
0048:
0049: void nextt(int ca[], int rule[])
0050: {
0051:   int nextca[N] = {0};
0052:   int i;
0053:
0054:   for(i = 1; i < N - 1; ++i){
0055:     nextca[i] = rule[ca[i + 1] * 4 + ca[i] * 2 +ca[i - 1]];
0056:   }
0057:
0058:   for(i = 0; i < N; ++i){
0059:     ca[i] = nextca[i];
0060:   }
0061: }
0062:
0063: void putca(int ca[])
0064: {
0065:   int i;
0066:   for(i = N - 1; i >= 0; --i){
0067:     printf("%ld",ca[i]);
0068:   }
0069:   printf("\n");
0070: }
0071:
0072: void initca(int ca[])
0073: {
0074:   char linebuf[BUFSIZE];
0075:   int  i = 0;
0076:
0077:   if(fgets(linebuf,BUFSIZE, stdin) == NULL){
0078:     fprintf(stderr, "初期状態の読み取りに失敗しました
\n");
0079:     exit(1);
0080:   }
0081:
0082:   for(i = 0; linebuf[i] != '\0'; ++i){
0083:     ca[N - 1 - i] = cvalue(linebuf[i]);
0084:   }
0085: }
0086:
0087: int cvalue(char ch)
0088: {
0089:   if(ch == '1'){
0090:     return 1;
0091:   }else{
0092:     return 0;
0093:   }
0094: }
0095:
```

```
0096: void setrule(char *arg, int rule[])
0097: {
0098:   int ruleno;
0099:   int i;
0100:
0101:   ruleno = atoi(arg);
0102:   if((ruleno < 0) || (255 < ruleno)){
0103:     fprintf(stderr, "ルール番号が正しくありません (%d) \n", ruleno);
0104:     exit(1);
0105:   }
0106:
0107:   printf("ルール番号 %d :", ruleno);
0108:
0109:   for(i = 0; i < R; ++i){
0110:     rule[i] = ruleno % 2;
0111:     ruleno /= 2;
0112:   }
0113:
0114:   for(i = R - 1; i >= 0; --i){
0115:     printf("%ld", rule[i]);
0116:   }
0117:   printf("\n");
0118: }
```

行列形式のデータを座標形式に変換するプログラム

```
0001: /*
0002:   coord.c
0003:   yukio sugawa
0004:   2012/2/18
0005:   行列形式のデータを座標形式に変換します
0006: */
0007:
0008: #include <stdio.h>
0009:
0010: #define BUFSIZE 1024
0011:
0012: int main()
0013: {
0014:   int  lineno = 0;
0015:   char linebuf[BUFSIZE];
0016:   int  i;
0017:
0018:   while(fgets(linebuf, BUFSIZE, stdin)!= NULL){
0019:     if(linebuf[0] == '0'){
0020:       for(i = 0; linebuf[i] != '\0'; ++i){
0021:         if(linebuf[i] == '1'){
0022:           printf("%d %d\n", i, -lineno);
0023:         }
0024:       }
```

```
0025:       ++lineno;
0026:     }
0027:   }
0028:
0029:   return 0;
0030: }
```

初期状態ファイル　「init.txt」の内容

```
00000000000000000000000000000001
```

上記のプログラムでファイルを作り、「wgnuplot」でグラフを描画する例

「ca1」の操作方法

./ca1 18 < init.txt | ./coord > fig18out.txt 「Enter」

または、

/.ca1 18 < init.txt > ca18out.txt 「Enter」
/.coord < ca18out.txt > fig18out.txt 「Enter」

「wgnuplot」を起動する。

「File」-「Chenge Directory」で、「fig18out.txt」のディレクトリを指定する。

plot "fig18out.txt.txt"「Enter」