

平成 2 1 年 度 ( 2 0 0 9 年 度 )

信州大学大学院工学系研究科修士学位論文

# セルペトリネット開発ツールの構築

専攻名 情報工学専攻

学籍番号 08TA564G

氏名 宮野 八州子

# 目次

1	はじめに	3
1-1	セルペトリネットとは	3
1-2	今回開発したツールの目標	3
2	システム概要	4
2-1	概念図	4
2-2	特徴	4
2-3	画面の流れ	5
3	プログラム概要	6
3-1	処理概要	6
3-2	I/O定義	7
3-2-1	Firing Rule File ( F R F )	7
3-2-2	Initial Status Definition File ( I N I )	8
3-2-3	Active Cell File ( A C F )	8
3-2-4	Marking Status File ( M S F )	9
3-2-5	出カイメッセージファイル	9
3-3	サブルーチンの一覧	10
4	システム管理層	11
4-1	システム開始・終了の制御、パラメータの受け渡し	11
4-2	実行セルの制御 ( A C F の役割)	11
4-3	各セルのマーク状況の更新・保持 ( M S F の役割)	12
4-4	画面表示	13
4-5	ペトリネット定義の読み込み ( F R F の役割)	14

4-6 ユーザーロジックに対応する変数・関数の提供 .....	15
5 近傍管理層 .....	19
5-1 出力Placeのあるセルを決定する方法 .....	19
5-2 セル間の制御の受け渡し .....	21
5-2-1 “特定セル指定による制御” の場合 .....	21
5-2-2 “法則指定による制御” の場合 .....	22
6 ユーザー管理層 .....	24
6-1 ユーザー F R F の作成 .....	24
6-2 初期状態定義ファイルの作成 .....	24
6-3 セルペトリネットの実行 .....	25
7 実行例 .....	26
7-1 基本的な動きの例 .....	26
7-2 複雑な動きの例 .....	28
7-3 鳥さんの散歩 .....	30
7-4 L I F E _ G A M E .....	34
7-5 トークンパッシングの方式の伝送状態のシミュレーション .....	37
7-6 道路の渋滞予測 .....	41
8 まとめと今後の課題 .....	49
謝辞 .....	50
参考文献 .....	51
付録 .....	52

# 1 はじめに

## 1-1 セルペトリネットとは

セルペトリネットとは、カラーペトリネット (CPTN) [6,7,8]の定義と近傍 (ある CPTN の transition から別の CPTN の place への出力 arc の集まり) の定義によって、同じ動きをする CPTN を複数動かすシステム構造を定義する概念であり、1996年に P. N. Kawamoto と中村八束による論文 “On cell Petri nets”で詳しく述べられている[1,2]。以降、「セル」を CPTN と空でない近傍で構成される単位と定義する。

セルペトリネットは世の中の自然現象や、ある事象のシステム (世界) 全体に与える影響を推察する手段として有効であると考えられる。たとえば、 $100 \times 100$  の  $10000$  個のセルで、ある任意のセルにボールが投げ込まれた時のそのボールの動きを時間ごとシミュレーションする場合を考える。さらに、一度に複数のボールが投げこまれた場合は、ボール同士がぶつかり、はじかれる場合もあるであろう。しかし、それぞれのセル内の動きのルールはすべてのセルで同じであり、自分のセルに制御が渡されたときにその時のセルのマーク状況に応じた動きを行い、次のセルに制御を渡せばよい。したがって、カラーペトリネットの定義はシステム全体で一つあればよく、セルごとにコピーして持つておく必要はない。また、あるセルから別のセルへの制御移動のルールを近傍として定義することにより、同じセルに制御が渡った場合でも、その時のマーク状態により次に動かすセルを制御することができる。このことによって、それぞれのセルでの動きを定義したカラーペトリネットの定義と近傍の定義の組み合わせにより、理論的には何千、何万というセルが同時に存在するセルペトリネットの世界を作ることができる。

## 1-2 今回開発したツールの目標

セルペトリネット開発ツールは今までも開発が試みられている[5]が、本論文では、このセルペトリネットを Web 上で稼働させるツールを紹介する。あくまでも、プロトタイプであるため、一度に動かすことのできるセルの数には限りがあるが、セル内の動きを定義する「カラーペトリネットの定義」、およびセル間の制御を定義する「近傍の定義」を実現できるツールの作成を目標とした。

Web アプリケーション開発言語としては、CAI で学習した PHP を用いることとした[9,10]。

このツールを構築した環境は、以下のとおりである。

Windows Vista

PHP V5.2.10,

Apache V2.2.1

## 2 システム概要

### 2-1 概念図

このツールを実現するシステムは、次の3つの層からなりたっている。

- ◆ システム全体をコントロールするシステム管理層
- ◆ セル間の制御をおこなう近傍管理層
- ◆ 主にセル内の動きを定義するユーザー管理層

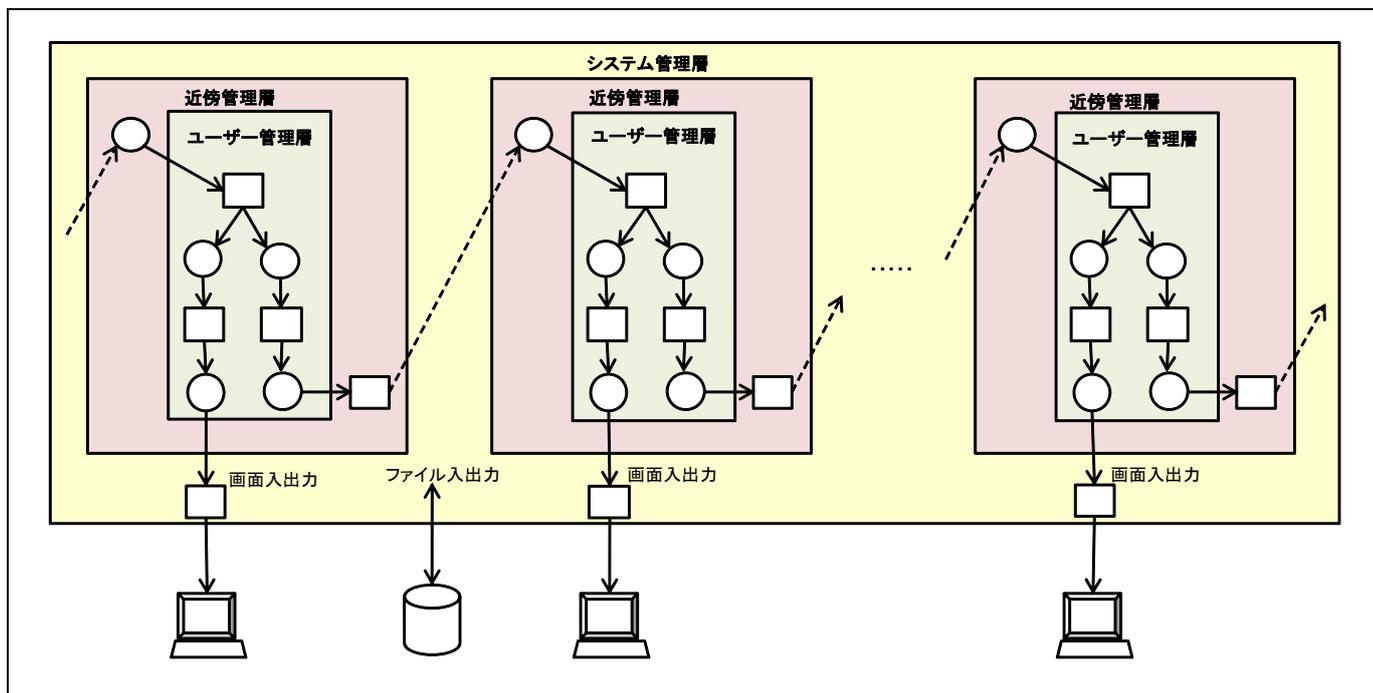


図 2-1 概念図

システム管理層では、画面の入出力や、ファイルの入出力、プログラム間のパラメータの受け渡しなどを行う。近傍管理層では、ユーザー管理層で決定されたセルのマーク情報に基づき次に制御が渡されるセルを決定し、そのセルの place に mark を送る。ユーザー管理層では、セルの定義（カラーペトリネットと近傍）および初期状態の定義を行う。また、開始画面で、動かすセルの個数や、開始セルの場所、次のセルの決定方法などを指定する。

### 2-2 特徴

システム管理層と近傍管理層の仕組みはプログラムに組み込まれているため、このツールを使用するユーザーはユーザー管理層の開発に専念することができる。ユーザーはプログラムを組むことな

く、セルの定義を Excel ファイルに記述し csv ファイルを作ることにより、ペトリネットを設計することができる。

ユーザーは開始画面でセル定義を記述したファイルとセルの初期状態を定義したファイル(任意)、実行したいセルの数、開始するセルの位置と次のセルの決定方法等を指定する。これにより、実行時にセルの個数などを簡単に変更してセルペトリネットを構築できる。

このツールでは、2次元までのセルペトリネットを構築することができる。

## 2-3 画面の流れ

次にこのツールを用いたセルペトリネットの実行例から、当システムの画面の動きをみる。この例では、セルを定義したファイルとして User\_FRF\_基本.csv を使用し、実行するセルの数は5 \* 5の25個とし、セル(0, 0)から始めてX軸方向に+1、Y軸方向に+1ずつしながら、そのセルの色を赤に変えていく。

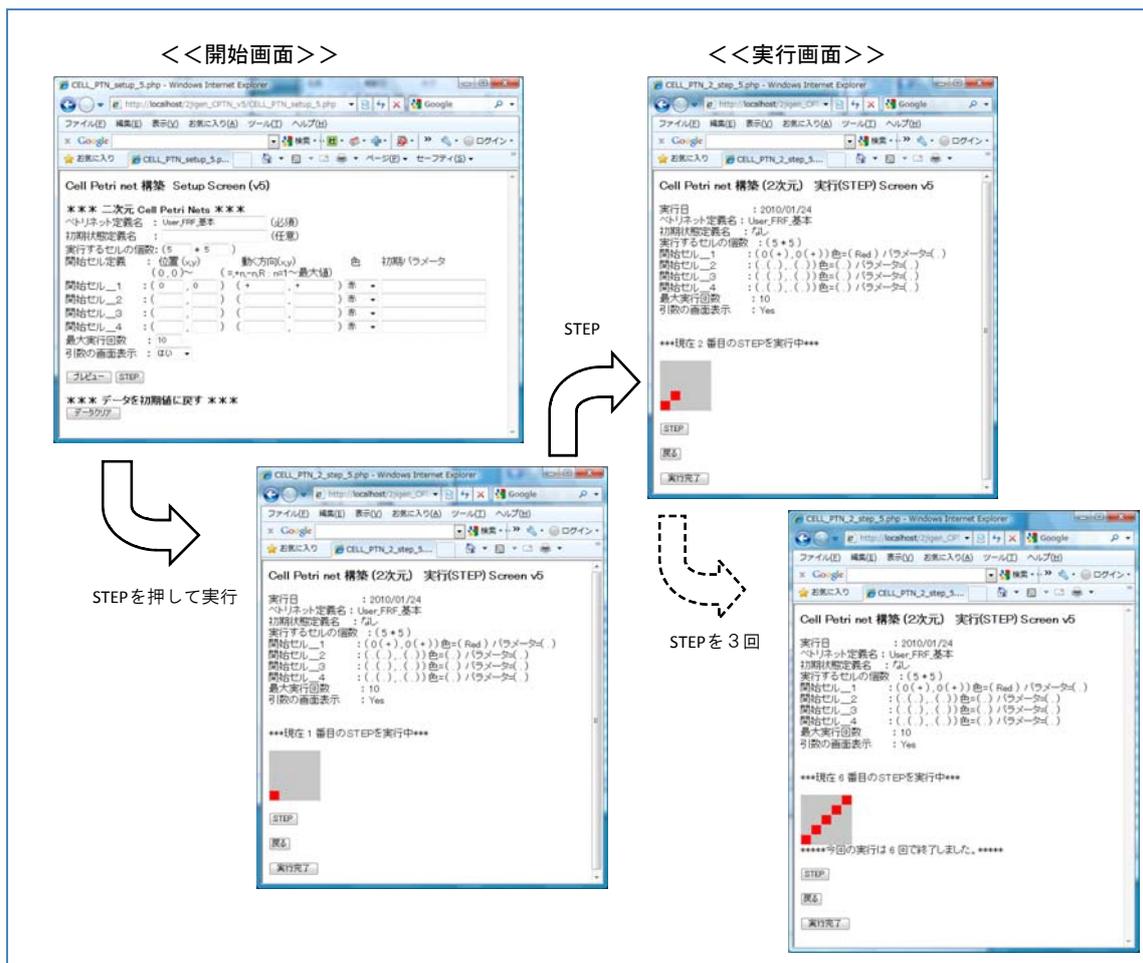


図 2-2 画面の流れ

## 3 プログラム概要

### 3-1 処理概要

このツールは2つの PHP プログラムから構成される。実行条件を入力する開始画面を出力する開始プログラムと、セルペトリネットを実行する実行プログラムである。開始プログラムは開始条件を実行プログラムにパラメータとして渡すだけでプログラムのロジックはほとんどないため、本論文では詳述しない。プログラムコードを付録に載せている。これより先は、セルペトリネット実行プログラムについて述べる。

セルペトリネット実行プログラムでは、次のような処理を行っている。ここで使われているファイル（FRF,INI,ACF,MSF,イメージファイル）については次の1/O定義の節で記述がある。

開始画面で

- ◆ Action として「STEP」が選ばれた場合、
  - ▶ 最初の実行の場合は、¥work フォルダのすべてのファイルを削除し、INI（初期状態定義ファイル）と開始画面の情報から必要なファイル（ACF,MSF）を初期化する
  - ▶ 実行するペトリネットの定義を記述しているユーザーFRF（Firing Rule File）およびシステムFRFを読む
  - ▶ 実行が予定されているセルの番号が記述されている ACF ファイル（Active Cell File）を読み、ACF に書かれているすべてのセルでペトリネットを実行する
  - ▶ 実行が終わったら、その時の各セルのマーク情報を MSF ファイル（Mark status File）に書き込み、次に実行するセル番号を ACF ファイルに書き込む
  - ▶ どこかのセルの色が変化していたら画面出力処理にすすむ。どのセルの色情報も変化していない場合はもう一度、ACF を読み、ペトリネットの実行を繰り返す
  - ▶ 画面出力になったら、すべてのセルの色情報を読み込んでイメージファイルを作成し、画面出力する。
  - ▶ あらかじめ開始画面で指定した最大実行回数を超えた場合、または次に実行するセルがなくなった場合は実行完了メッセージを出力し、最後のイメージファイルを画面出力する
- ◆ Action として「プレビュー」が選ばれた場合、
  - ▶ 開始プログラムで指定された開始位置と色情報、および初期状態定義ファイル（INI）に定義されたセルの色情報をもとに開始時の各セルの色を描画する
- ◆ Action として「戻る」が選ばれた場合、
  - ▶ 開始画面で指定されたパラメータを保持したまま開始画面に戻る

- ◆ Action として「実行完了」が選ばれた場合、
  - 開始画面で指定されたパラメータを保持せずに開始画面に戻る

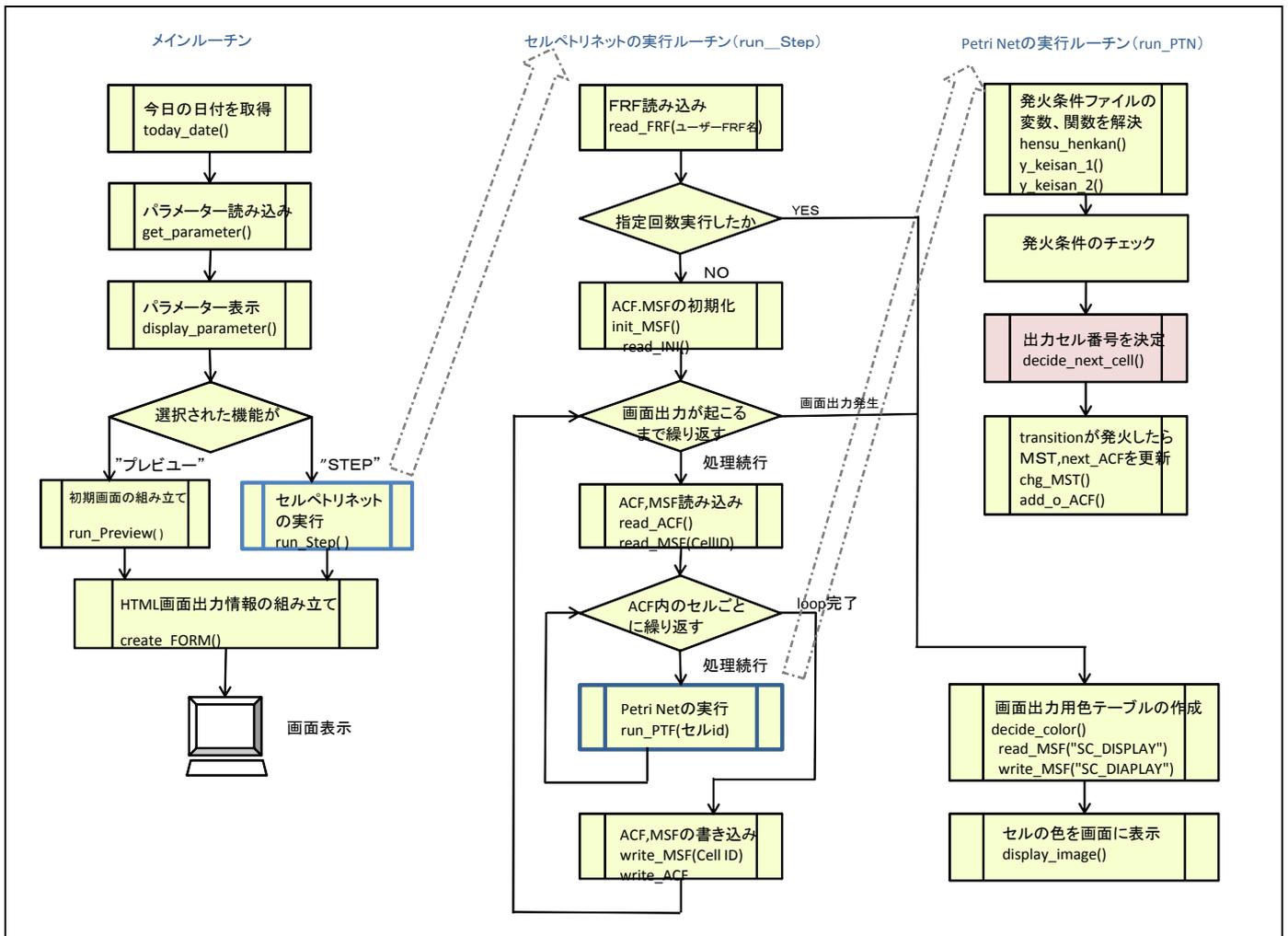


図 3-1 セルペトリネット実行プログラムのフローチャート

「STEP」処理については4章以降で詳述する。また、付録にプログラムコードが記載されている。

### 3-2 1 / 0 定義

このシステムで使われるファイルは次の5種類である。FRF,INI,ACF,MSF のサンプルが7章および付録に含まれている。

#### 3-2-1 FIRING RULE FILE ( F R F )

発火条件、transition が発火した時の出力情報を定義する csv ファイル

システムが管理する F R F では、セル間の制御の受け渡しに関する処理、画面出力の処理を定義している。ユーザーが管理する F R F では、そのセルの C P T N および近傍の定義を行う。

システム F R F の名前はプログラムの中に組み込まれているが、ユーザー F R F の名前は任意であり、システム起動時に開始画面で名前を指定する。ユーザー F R F はシステム F R F と同じフォルダーにあらかじめコピーしておく。

ファイルの構造は、以下の通り

- Comment Flag : コメント行の場合は “\*”
- Rule ID : 発火条件のまとまりを表す名前
- Transition ID : Transition の名前
- Arc Type : input Arc の定義なのか output Arc の定義なのかを識別する
- Output Cell ID : output Arc の場合の次のセルを決定するルールの記述
- Place ID : Place の名前
- Mark : input Arc の場合は input Place で発火条件と照合するマーク情報  
output Arc の場合は output Place に書かれるマーク情報

### 3-2-2 INITIAL STATUS DEFINITION FILE ( I N I )

セルペトリネット実行時に初期値として各セルが持つマーク状態を定義する csv ファイル

プログラム実行時に一度だけ読み込まれ、セルペトリネット実行前に初期状態を作り出すために使用される。I N I ファイルは F R F と同じフォルダーにあらかじめコピーしておく。

ファイルの構造は、以下の通り

- Comment Flag : コメント行の場合は “\*”
- Cell ID\_x : X 軸方向のセル番号
- Cell ID\_y : Y 軸方向のセル番号
- Place ID : Place の名前
- Mark : マーク情報

### 3-2-3 ACTIVE CELL FILE ( A C F )

現在実行可能なセルの番号を格納する csv ファイル

プログラムはこのファイルに格納されているすべてのセルで F R F の定義にしたがってペトリネットを実行する。A C F に入るセル番号がなくなったときは、実行の完了となる。

このファイルはシステムが自動的に更新していく。

ファイルは現在実行しているプログラムがあるフォルダーの中に作られる¥work フォルダー内に作られ、ファイル名は、ACF.csv

ファイル構造は以下の通り

- ◆ Cell ID : 実行する可能性のあるセルの Cell ID

### 3-2-4 MARKING STATUS FILE (MSF)

セルの現在のマーク情報を保持している csv ファイル。

各セルにおいて、ペトリネットを実行していくことにより place および mark の情報が変化していくが、現在の place、mark の情報をこのファイルに保持していく。

このファイルはシステムが自動的に作成・更新していく。

ファイルは¥work フォルダー内に作られ、ファイル名は MSF\_UC\_nnnnn\_mmmmm.csv ここで、nnnnn,mmmmm は 2次元セルを表す X 軸方向、Y 軸方向それぞれの 5 桁の値

ファイル構造は以下の通り

- ◆ Place ID : Place の名前
- ◆ Mark : その Place に書かれている Mark の値

また、MSF\_SC\_DISPLAY.csv は、画面出力セルのマーク情報保持ファイルであり、各セルの色情報が格納されている。

ファイル構造は以下の通り

- ◆ Place ID : Cell ID
- ◆ Mark : そのセルの色情報

### 3-2-5 出力イメージファイル

プログラムはどこかのセルで画面出力が出されるまで繰り返し実行される。画面出力が要求されると出力イメージファイルが作られ、画面に出力される。

ファイルは¥work フォルダー内に作られ、名前は IMG\_n\_tttttt.png ここで、n は実行ステップ番号、tttttt はファイルが作成された時間(hhmmss)である。ステップ番号は画面出力されるごとに 1 ずつ増えていく番号である。

### 3-3 サブルーチンの一覧

セルペトリネット実行プログラムはメインルーチンとサブルーチンから成り立っている。次の表はサブルーチンの一覧表である。

	サブルーチン名	内容	引数
ロジック	run_Step()	「STEP」が押された時の処理	-
	run_Preview()	「プレビュー」が押された時の処理	-
	today_date()	今日の日付および現在の時間を取得	-
	get_parameter()	パラメータ読み込み	-
	display_parameter()	パラメータ表示	-
	init_MSF()	ACF, MSFテーブルの初期化	-
	run_PTN()	Petri Netの実行	Cell ID
	hensu_henkan()	発火条件ファイルの変数、関数を解決	Markの値
	y_keisan_1()	一次方程式でY軸の値を計算	-
	y_keisan_2()	二次方程式でY軸の値を計算	-
	decide_next_cell()	出力セル番号を決定	-
	chg_MST()	Transitionが発火したらMSTを更新	-
	add_o_ACF()	次回実行するセルに加える	-
	decide_color()	画面出力用色テーブルの作成	-
	display_image()	セルの色を画面に表示	-
create_FORM()	HTML画面出力情報の組み立て	-	
FILE I/O	read_FRF()	FRF読み込み	FRF名
	read_INI()	INI読み込み	INI名
	read_MSF()	MSF読み込み	Cell ID
	write_MSF()	MSFへの書き込み	Cell ID
	read_ACF()	ACF読み込み	-
	write_ACF()	ACFへの書き込み	-

図 3-2 サブルーチン一覧

## 4 システム管理層

この章では、システム管理層に関するプログラムの詳細を述べる。

### 4-1 システム開始・終了の制御、パラメータの受け渡し

Web 画面で開始プログラム(CELL\_PTN\_setup\_6.PHP)を始動させることで開始画面が出力される。開始画面でユーザーが実行環境を指定し、“STEP” ボタンを押すことにより、セルペトリネット実行プログラム(CELL\_PTN\_2\_step\_6.PHP)が開始する。

サブルーチン `get_parameter()` では、開始画面から渡される実行条件を指定するパラメータを読み込み、必要な補正（省略時値の設定、パラメータの合成など）を行っている。`display_parameter()` では、読み込んだパラメータの内容を実行画面に表示する。ここで、表示不要の指定があればパラメータの表示は行われない。また、`today_date()` で今日の日付と現在の時刻を獲得している。この情報は、画面表示およびイメージファイルのファイル名に使われている。

Web プログラムは画面出力をすることでプログラムが終了する。したがって画面出力のときには次の実行に必要な情報は外部のファイルに書き込むか、パラメータとして引き渡す必要がある。今回は、次回実行すべきセルの番号と、画面出力時に各セルが持っているマーク情報は情報量が多いためそれぞれ ACF ファイル、MSF ファイルを使って受け渡し、開始画面から渡された開始条件情報と現在のステップ番号をパラメータとして受け渡すことにした。サブルーチン `create_HTML()` で、パラメータの引き渡しを行っている。

このツールでは、画面表示をする単位を 1 ステップとした。開始条件で実行する最大回数を指定する。これにより、今回は実装できなかったが、将来“連続実行”の機能を加えた場合に無限ループに陥ることを回避している。“STEP”での実行においても、指定された回数実行した場合と、もはや動く可能性のあるセルがなくなったときには実行の終了とみなし、画面にメッセージを出し、セルペトリネットの実行は終了するようにした。

### 4-2 実行セルの制御（ACF の役割）

セルペトリネットにおいてはある時点でどのセルが実行可能、または実行中かということは非常に重要になってくる。セルの数が多くなった場合、すべてのセルについて毎回ペトリネットを実行させては効率も悪いし、必要ないからである。しかし、その時点で実行可能なセルの場所は常に変化し、またその数も常に増減する。

このため、このツールでは、実行可能なセルの番号を A C F (Active Cell File) に保持することにした。実行が始まると A C F を読み、その中に列挙されているセル番号ごとにペトリネットを実行する。ペトリネットの実行中にマーク情報に変化があったセルは、次回ペトリネットを実行する必要がある可能性ありとしてプログラム内の配列 (next\_ACF) に登録しておき、すべてのセルの実行が終わった時点で A C F ファイルを置き換える。こうすることにより、常に次回実行すべきセルの一覧が保持される。

最初の実行の時は、開始画面から渡された開始セルの番号を A C F に登録する。この作業はサブルーチン init\_MSF() の中で行われている。

A C F への読み書きはサブルーチン read\_ACF()、write\_ACF() が行っている。

#### 4-3 各セルのマーク状況の更新・保持 (M S F の役割)

セルペトリネットでは、セルごとにマークの情報が異なる。しかも、ペトリネットの実行とともにマーク情報は次々に変わっていく。また、セルのマーク情報はプログラムをまたがって保持される必要がある。このマーク情報は大量になるので外部ファイル M S F (Mark Status File) に保持させることにした。ここで、ひとつのファイルに全セルのマーク情報を持たせるとファイルが膨大になってしまい、しかもある時点で必要になってくるセルはかなり限られた数であることも多いであろう。そこで、このシステムでは、セルごとに M S F を持たせることとした。セルごとに M S F をもつことにより、csv ファイルの OPEN/CLOSE の負荷は増えてしまうが、不要な情報のために行う read/write の負荷を減らすことができ、またプログラム開発も容易になるため、この方式を採用した。

セルペトリネットの実行が始まると A C F が読まれ、今回実行されるセルの番号がわかる。そこで A C F 内のすべてのセルの M S F ファイルを読み込み、プログラム内の配列 (MST: Mark Status Table) を作る。M S T の構造は次のとおりである。

- ◆ Cell ID : Cell ID
- ◆ Place ID : Place の名前
- ◆ Mark : その Place に書かれている Mark の値

この配列の値として M S F から読み込まれた時は 1 を与える。ペトリネットの実行で使われたら 0 にする。ペトリネットの実行で新たに加えられた mark には 1 を与える。すべての実行可能なセルについてのペトリネットの実行が完了した時点で値が 1 のもののみ M S F に書き戻される。こうして、それぞれのセルの最新のマーク情報が保持される。

MSTをプログラム内に持つことにより、ファイルのI/Oを減らすことができるが、実行する可能性のあるセル数が多くなったとき、配列が大きくなりすぎてエラーになる可能性が残ってしまった。これは今後の改良が望まれるところである。

最初の実行の時、すべてのセルについて Place ID に “Cell ID”、Mark にセル番号を入れたマーク情報を MSF に書き込む。また、初期状態定義ファイル (INI) の情報と、開始画面の開始セル情報で各セルの初期情報を作り出す。この作業はサブルーチン `init_MSF()` の中で行っている。

MSF への読み書きはサブルーチン `read_MSF()`、`write_MSF()` が行っている。また、INI からの読み込みはサブルーチン `read_INI()` が行っている。

## 4-4 画面表示

セルペトリネットの実行の結果は、各セルの色の変化であらわされる。色情報はペトリネットを実行している中で “SC\_DISPLAY” というシステムセルにユーザーが色を書き込むことで、システムに渡される。具体的には、SC\_DISPLAY セルの MSF にセルごとの色情報が書き込まれる (Place ID = Cell ID、Mark = Red などの色情報)。セルペトリネットでは、どこかのセルで色の変化があった場合、その時の SC\_DISPLAY セルの MSF を読んでイメージファイルを作成しそれを描画する。

ペトリネットを実行していくと、同じセルに複数の色が書き込まれる可能性が出てくる。そこでこのツールでは、色の優先順位を赤>青>緑>白>灰色とし、複数の色が書かれた時はこの優先順位に従って優先順位の高い色を採用する。また、初期値としてすべてのセルに灰色を与えている。画面に表示するときには、そのセルに新たに色を与えられた時はその色を、色の変化がなかった時は前回表示した色をそのまま、一度も変化していない場合は省略時値の灰色にする。この作業はサブルーチン `decide_color()` で行っている。

表示する色が決定したら、`image/png` を使用してイメージファイルを作成し、そのファイルを表示する。セルの数によって各セルの描画の大きさを調整し、なるべく 1 画面に描画できるようにした。また、`image/png` では左上隅が原点(0,0)となり、Y 軸方向に 1 ずつ増やすと 1 行ずつ下に表示されてしまう。これは、通常感覚と異なり違和感を覚えるため、原点を左下隅になるよう補正した。また、ファイル名が同じになるとファイルとしては別のものができているにも関わらず画面上は古いものが表示されてしまうため、ファイル名に時刻を加えることで表示の切り替えを確実にした。この作業はサブルーチン `display_image()` で行っている。

原点が(0,0)のため、実行するセルの個数として 5x5 を指定した場合は、(0,0) から(4,4)が有効な範囲となる。開始画面で開始位置等を指定する場合に注意が必要である。このツールでは、この問題を

少しでも軽減するために、セルペトリネット開始時に“プレビュー”機能を設け、実行前にセルペトリネット実行開始時のイメージをあらかじめ確認できるようにした。この機能はサブルーチン run\_Preview()で行っている。

#### 4-5 ペトリネット定義の読み込み（FRFの役割）

ツール開発で最も関心があるのは、実行するペトリネットをどのように定義するかであろう。ペトリネットのツールを開発するときはPNML（Petri Net Markup Language）[3,4]を用いて定義していくことが多い。さらに、GUIで画面上にペトリネットのPlace、Transition、Arcを描画していくことにより自動的に定義できるものもいくつかある[5]。しかし、セルペトリネットを定義するために使えるツールが見つからなかったため、本論文のツールでは既存のMarkup Languageは使用していない。その代わりに、Excelで入力Arc、出力Arcを記述することによりペトリネットの定義ができるようにした。作成したExcelファイルをcsvファイルに変換したものがFRF(Firing Rule File)である。FRFは、システムが提供するシステムFRFと、ユーザーが作成するユーザーFRFの2種類がある。システムFRFには、セル間の制御の受け渡しに関する処理、画面出力の処理を定義している。

FRFの記述は、発火ルール単位に入力Arc、出力Arcを定義することで構成される。図4-1FRFの定義例で概要を説明する。最初のルール（Rule1）は、PlaceAというPlaceにredというmarkがあれば発火し、to\_displayというPlaceにredというmarkをかく。Output cell idが“=”になっているのは、自分自身（同じセル）のPlaceに書き出すことを指示している。2番目のルール（Rule2）はPlaceAにblueがあり、かつPlaceBにredがあれば発火し、to\_displayにgreenを書く。次のルール（SR41）では、output cell idが“SC\_DISPLAY”となっている。このため、出力placeは別のセル、この例ではSC\_DISPLAYとなる。また、Place id、Markに“%”のついた記述があるが、これは実行時に動的に値を決定するための変数である。%cell\_idは現在のセル番号を表している。また、%colorは入力Arcで与えられたデータをそのまま出力Arcに渡すことを意味している。変数に関する詳しい説明は次節で述べる。

* rule id	transition id	input/output	output cell_id	place id	mark
* 同じセル内の別のPlaceにmarkを書く					
Rule1	T1	i		PlaceA	red
Rule1	T1	o	=	to_display	red
* 入力条件が複数あるパターン					
Rule2	T2	i		PlaceA	blue
Rule2	T2	i		PlaceB	red
Rule2	T2	o	=	to_display	green
* "SC_DISPLAY"セルに現在のセルIDと色を書く(システムFRFでの定義例)					
SR41	ST4	i		to_display	%color
SR41	ST4	o	SC_DISPLAY	%cell_id	%color

図 4-1 FRF 定義例

図 4-2は、図 4-1で定義した動きをペトリネット図で表したものである。破線の矢印は別のセルの Placeへの書き出しを表している。Output cell idの指定については、次章で詳述する。

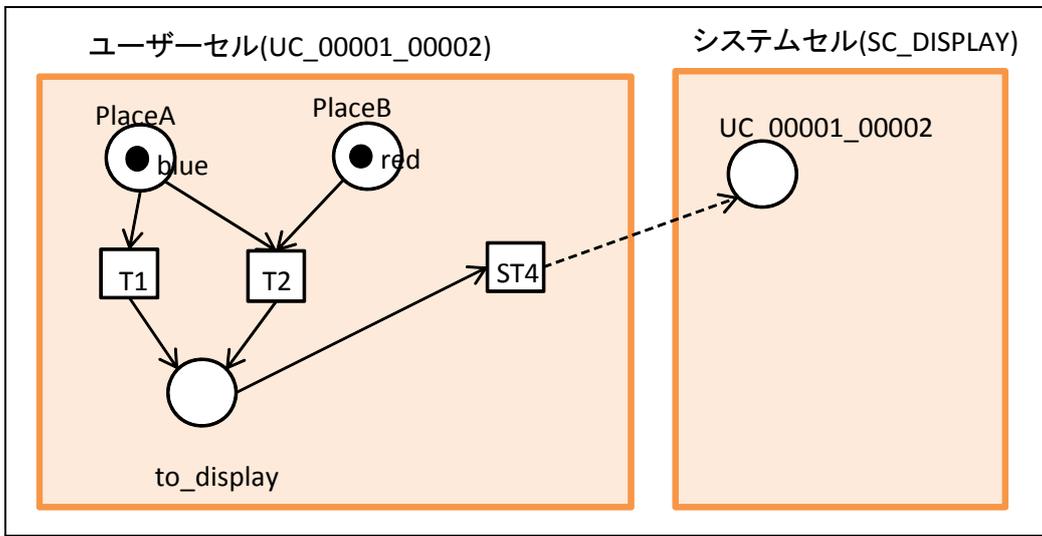


図 4-2 F R F 定義例のペトリネット図

#### 4-6 ユーザーロジックに対応する変数・関数の提供

セルペトリネットでは、同じ発火条件を異なるセルで動かすことになるため、マーク情報などを実行時に動的に変更したい場合がよくある。たとえば、図 4-1 F R F の例の最後の 2 行を解説する。

* Rule	Transition	I/O	output	place	mark
* ID	ID		cell id		
SR41	ST4	i		to_display	%color
SR41	ST4	o	SC_DISPLAY	%cell_id	%color

最後の行は、SR41 というルールで定義される transition ST4 の出力 Arc であり、出力セルは SC\_DISPLAY、出力 Place はその時実行しているセル ID、出力する色はその時入力 Arc が持っていた色であることを定義している。このとき実行している Cell ID はあらかじめ F R F に記述しておくことはできないので F R F では変数 %cell\_id として記述し、実行時にその時の Cell ID を与えるようにした。

このツールでは、F R F を記述するときの規則として、“%” で始めた名前は変数とみなす。%cell\_id はいつでも使えるシステム変数であり、実行中の Cell ID に置き換えられる。それ以外の名前はユーザーが自由に使うことができる変数で、そのルールの実行中有効である。変数は、まず入力 Arc の mark で定義した変数名(この例では%color)にその時の M S T (Mark status Table) の mark の値を与え、

その後出力 Arc の output cell id、place id、mark で使うことができる。変数の値はプログラム内の配列（\$w\_parm）に書き込まれ、次のルールに移ったタイミングで消去される。

また、mark で使える “%= ” で始まる関数をいくつか用意した。この関数により、FRF にペトリネットを表現していくときに、より複雑な動きを定義できるようになる。このツールでは14種類の関数を用意した。

関数名	形式	用法
concat	%=concat[%a:%b: ... :%x]	“.”をセパレーターとして元のmark値を分割し、それぞれ変数名を%a,%b...%xとする
concat_any	%=concat_any[(セパレーター):%a:%b: ... :%x]	ユーザーが指定したセパレーターでmark値の分割を行う
split	%=split[%a:%b: ... :%x]	“.”をセパレーターとして%a,%b...%xを結合する
split_any	%=split_any[(セパレーター):%a:%b: ... :%x]	ユーザーが指定したセパレーターで%a,%b...%xを結合する
加算	%=+[%a:%b]	%a + %b を計算する
減算	%=-[%a:%b]	%a - %b を計算する
掛け算	%=*[%a:%b]	%a * %b を計算する
割り算	%=/[%a:%b]	%a / %b の商を計算する (小数点以下第2位で四捨五入)
Mod	%=%[%a:%b]	%a / %b の余りを計算する
min	%=min[%a:%b: ... :%x]	%a,%b...%xの最小値を返す
max	%=max[%a:%b: ... :%x]	%a,%b...%xの最大値を返す
<	%=<[%a:%b]	比較演算子:%a<%bなら1を、それ以外なら0を返す
>	%=>[%a:%b]	比較演算子:%a>%bなら1を、それ以外なら0を返す
=	%==[%a:%b]	比較演算子:%a=%bなら1を、それ以外なら0を返す

図 4-3 関数一覧

これらの関数の使用例を図 4-4に示す。

ルール UR02 では、関数 %=split で place receive の mark を3つの変数の値にわせる。たとえば、このときMSTに実際入っている mark が “+1\_+1:red:0” だとすると、このマーク情報がプログラム内の変数用の配列（\$w\_parm）に次のように分割される。

```
$w_parm[%o_towards] = "+1_+1"
$w_parm[%o_color] = "red"
$w_parm[%gokei] = 0
```

この例では、これらの変数をその後のペトリネットの実行で使用できるように出力 arc でそれぞれ

```
Place=o_towards    mark=+1_+1
Place=o_color      mark=red
Place=gokei        mark=0
```

としてMSTに出力している。

ルール UR31 では、ur02 で出力された o\_towards という place の mark を関数 %=split\_any でさらに分割している。このとき、この変数の最初のパラメータは分割するセパレーターを表すため、分割のセパレーターとして “\_” が指示されている。この関数の実行結果は

```
$w_parm[%o_x]=+1
$w_parm[%o_y]=+1
```

である。次の出力 arc の定義で

```
Place=o_x      mark=+1
Place=o_y      mark=+1
```

がM S Tに書き出された。

ルール UR21 では、M S Tに入っている place esa の mark が3だとすると、変数\$w\_parm[%esa]=3 となる。Place gokei は UR02 で出力され、mark=0 だったため、変数\$w\_parm[%gokei]=0 となる。次の出力 arc の記述では足し算を行う関数 %=+が使われている。\$w\_parm[%gokei]に\$w\_parm[%esa]を加えるため、結果はM S Tに place=gokei mark=3 を出力する。

ルール UR33 では、place o\_y を入力 arc とし、出力する際に掛け算の関数%=\*で-1 を掛けている。このように、四則演算をおこなうときには、変数以外に定数を使うことも可能である。その場合は関数の中に直接数値を記述する。この関数の実行の結果、place=o\_y mark=-1 がM S Tに出力される。

ルール UR34 では、関数 %=concat\_any により、place o\_x と place o\_y をセパレーター “\_” で結合する。これにより、place=o\_towards mark=+1\_-1 が出力される。

ルール UR71 では、関数%=concat により、 o\_towards、 o\_color、 gokei の3つの place の mark 情報を結合して place=to\_next mark=+1\_-1:red:3 が出力される。

ここまでのM S T (Mark Status Table) の変化の様子を図 4-5で確認する。今回のF R Fの定義ではoutput cell idはすべて “=” だったので、すべて同じセルの中で動いている。ここでは、Cell IDをUC\_n\_mとあらわしている。実際にはn,mは5桁の数字が入る。図 4-5の配列の値は、それぞれの発火ルールの実行後のM S Tの各配列の値である。入力Arcで読み込まれると0になる（マークがなくなる）。

* rule id	transition id	input/output	output cell id	place id	mark
* 渡された引数を動く方向(o_towards)と色(o_color)と食べた餌の合計(gokei)を分離する					
UR02	UT02	i		receive	%=split[%o_towards:%o_color:%gokei]
UR02	UT02	o	=	o_towards	%o_towards
UR02	UT02	o	=	o_color	%o_color
UR02	UT02	o	=	gokei	%gokei
* 動く方向をX軸、Y軸にわけける					
UR31	UT31	i		o_towards	%=split_any[(.):%o_x:%o_y]
UR31	UT31	o	=	o_x	%o_x
UR31	UT31	o	=	o_y	%o_y
* ここで食べた餌の数を合計に加える					
UR21	UT2	i		esa	%esa
UR21	UT2	o	=	gokei	%gokei
UR21	UT2	o	=	gokei	%=[%gokei;%esa]
* Y軸の動きを反転させる計算					
UR33	UT33	i		o_y	%o_y
UR33	UT33	o	=	o_y	%=[-1:%o_y]
* X軸、Y軸の動きを合成する					
UR34	UT34	i		o_x	%o_x
UR34	UT34	i		o_y	%o_y
UR34	UT34	o	=	o_towards	%=concat_any[(.):%o_x:%o_y]
* 次のセルに移動するためにパラメータを合成する					
UR71	UT7	i		o_towards	%o_towards
UR71	UT7	i		o_color	%o_color
UR71	UT7	i		gokei	%gokei
UR71	UT7	o	=	to_next	%=concat[%o_towards:%o_color:%gokei]

図 4-4 関数使用の例

Cell Id	Place ID	Mark	配列の値
UC n.m	receive	+1 +1:red0	1
UC n.m	esa	3	1
UR02			
UC n.m	receive	+1 +1:red0	0
UC n.m	o_towards	+1 +1	1
UC n.m	o_color	red	1
UC n.m	gokei	0	1
UC n.m	esa	3	1
UR31			
UC n.m	receive	+1 +1:red0	0
UC n.m	o_towards	+1 +1	0
UC n.m	o_x	+1	1
UC n.m	o_y	+1	1
UC n.m	o_color	red	1
UC n.m	gokei	0	1
UC n.m	esa	3	1
UR21			
UC n.m	receive	+1 +1:red0	0
UC n.m	o_towards	+1 +1	0
UC n.m	o_x	+1	1
UC n.m	o_y	+1	1
UC n.m	o_color	red	1
UC n.m	gokei	3	1
UC n.m	esa	3	0
UR33			
UC n.m	receive	+1 +1:red0	0
UC n.m	o_towards	+1 +1	0
UC n.m	o_x	+1	1
UC n.m	o_y	-1	1
UC n.m	o_color	red	1
UC n.m	gokei	3	1
UC n.m	esa	3	0
UR34			
UC n.m	receive	+1 +1:red0	0
UC n.m	o_towards	+1 +1	0
UC n.m	o_x	+1	0
UC n.m	o_y	-1	0
UC n.m	o_towards	+1 -1	1
UC n.m	o_color	red	1
UC n.m	gokei	3	1
UC n.m	esa	3	0
UR71			
UC n.m	receive	+1 +1:red0	0
UC n.m	o_towards	+1 +1	0
UC n.m	o_x	+1	0
UC n.m	o_y	-1	0
UC n.m	o_towards	+1 -1	0
UC n.m	o_color	red	0
UC n.m	gokei	3	0
UC n.m	esa	3	0
UC n.m	to_next	+1 -1:red3	1

図 4-5 関数の使用例を実行したときのMSTの変化

## 5 近傍管理層

この章では、他のセルとの制御の受け渡しのロジックを解説する。

### 5-1 出力PLACEのあるセルを決定する方法

セルペトリネットにおいて、出力 Place のセルをどのように決定するかはツールの特性として重要になってくる。このツールでは、F R F で出力 Arc を定義するときに、出力するセルを指定する。この出力先を整理してみると、

- ① 自分自身
- ② ある特定のセル
- ③ すべてのセル
- ④ 定義されている範囲内のどこかのセル

と分類できるであろう。

①の自分自身は通常のペトリネットの動きであり、特別な処理は必要ない。このツールでは、F R F の output cell id に“=”を書くことで自分自身への出力を定義しており、プログラムは output Cell ID を今実行している Cell ID にしてM S Tに出力 Place を書き出す。

②の特定のセルは、F R F の output cell id に制御を渡したいセルを直接指定する。このツールでは、セルはシステムセルとユーザーセルに分けられる。システムセルの名前は SC\_xxxxx (xxxxx はあらかじめ定義されている名前、現在は SC\_DISPLAY のみ)、ユーザーセルの名前は UC\_nnnnn\_mmmmm (ここで nnnnn、mmmmm はそれぞれ5桁の数字でそれぞれX軸、Y軸のセル位置をあらわす)と定められているため、F R F の output cell id が“SC”または“UC”で始まる場合、プログラムは特定セルの指定と解釈し、指定された Cell ID を出力の Cell ID としてM S Tに書き込む。

③のすべてのセルへの出力の場合は、F R F の output cell id に“all”と記述する。これによりプログラムは範囲内のすべてのセルへの出力 Place をM S Tに書き出す。

②および③への出力は、あらかじめ近傍として定められたセルに制御を渡す場合に用いられる。また、初期値としてなんらかの Place、Mark をすべてのセルか特定のセルに与えるような場合にも使用される。それに対して、④の出力は、どのセルに制御を渡すかをあらかじめ定義しておくのではなく、制御を渡す法則を伝えることにより、制御を渡す時点で次に渡すべきセルを決定していくものである。同じ法則で近傍が定義でき、セル数が多く個々に定義していくのが大変な時有効である。

以後、②および③の方式を“特定セル指定による制御”、④の方式を“法則指定による制御”と分類する。

④の“法則指定による制御”をさらに整理すると

- A) X軸、Y軸方向にそれぞれ+1,-2のように規則的に動き方を指定できるもの
- B) 一次方程式、二次方程式など数式を使って動き方をあらわすことができるもの
- C) 近傍として定義されている中でランダムに動くもの

と分類できるであろう。このツールでは、開始画面で、またユーザーF R Fの中でそれぞれ次のように指定できる。

A)は、開始画面での“動く方向”の指定で、またはユーザーF R Fの中で place to\_next の mark で指定する方向パラメータ (%o\_towards) で、動きの規則を指定できるようにした。ここで、使われる規則はX軸方向、Y事項方向それぞれ

- +n : プラスの方向に n 動く。n=1 のときは 1 を省略できる
- n : マイナスの方向に n 動く。n=1 のときは 1 を省略できる
- = : 動かない。X軸方向が“=”ならY軸に平行に動き、Y軸方向が“=”ならX軸に平行に動き、両方とも“=”なら全く動かないことになる。
- 0 : =と同じ

このツールでは、セル番号を+nした時にその軸の最大値を超えた場合は、その時点で次のセルへの出力は停止する。また、-nした時に値がマイナスになった場合も次のセルへの出力は停止する。

B)は、このツールではサンプルとして一次方程式と二次方程式を用意した。今後、指数関数・三角関数などを加えていけば、さらに応用範囲が広がっていくであろう。使い方は、開始画面のX軸方向の動きに“C1”を指定すると一次方程式で計算され、“C2”を指定すると二次方程式での計算となる。それぞれY軸方向の動きの欄に係数をパラメータとして与える。

たとえば、開始画面での指定が

(C1,3&2) なら、X軸方向には+1、Y軸の値はその時の x の値で計算された  $y = 3x + 2$  の値

(C2,5&4&3) なら、X軸方向には+1、Y軸の値はその時の x の値で計算された  $y = 5x^2 + 4x + 3$  の値

というように、計算されて次のセルが決定する。

また、ユーザー F R F の中で place to\_next の mark で指定する方向パラメータ (%o\_towards) で、C1\_(3&2)、C2\_(5&4&3)と指定しても同じよう計算されて次のセルが決定する。ユーザー F R F で指定する場合は、定数はカッコ()でくくり、定数間のセパレーターとしては“&”を使用する。

C)は、開始画面の動く方向の定義または F R F 内の指定で、方向として“R”と指定することにした。プログラムではセルの範囲内でランダム関数を使って位置を決定する。X軸方向、Y軸方向それぞれにランダムを選択できる。

今回のツールでは、④の“定義されている範囲内のどこかのセル“を考えると、”定義されている範囲“とは、セル(0,0)から開始画面で“実行セルの個数”として指定した個数のセルまでの全手のセルを対象としている。したがって、実行セルの個数として(5,5)と指定したら、とり得る Cell ID は(0,0)から(4,4)までの25個のセルとなる。近傍として、この中の“いくつか”という場合もあるであろう。現在、この“いくつか”を表す方法として A)または B)の方法は用意されている。しかし、数式で表現できないセル番号の集まりを表現する方法は、今回は確立されていない。C)のランダムはその一助にはなるが、ランダムに動く範囲は全体であり、あらかじめ限られたセルのみを指定することはできない。今後、近傍を範囲内のセルの任意の部分集合で定義できるように改良していきたい。

## 5-2 セル間の制御の受け渡し

この節では、前節で分類した“特定セル指定による制御”と“法則指定による制御”それぞれの方式でのセル間の制御の受け渡しの方法をみていく。

### 5-2-1 “特定セル指定による制御”の場合

このツールでは、ユーザー F R F の output cell id に制御を渡すセル番号を直接指定することで特定のセルに制御を渡すことができる。すべてのセルで同じセルに制御を渡すのであれば、ユーザー F R F にそのセル ID を直接指定できるが、セルによって次の近傍が違うので必ずしも一意には決められない。そこで、初期状況定義ファイルに制御を渡すセル ID をあらかじめ定義しておき、ユーザー F R F では、変数を使ってそのセル用の近傍に制御を渡すようにできる。

たとえば、初期状況定義ファイル(INI)で図 5-1の様に定義する。これにより、セルUC\_00000\_00002 ではnext\_cellの値がUC\_00001\_00002 となり、セルUC\_00001\_00002 ではnext\_cellがUC\_00002\_00002 と定義された。次にユーザー F R F で図 5-2の様に定義すると、最後の行の%next\_cellは1行目のnext\_cellで取り込んだそのセルのnext cellの値が使われ、セルごとに異なるセルに制御を渡すことが可能になる。

*	cell_id_x	cell_id_y	place_id	mark
	0	2	next_cell	UC_00001_00002
	1	2	next_cell	UC_00002_00002
	2	2	next_cell	UC_00003_00002

図 5-1 初期状況定義ファイル(INI)での定義の例

*	rule_id	transition	input/ output	output_ cell_id	place_id	mark
*		id				
*						
	UR51	UT51	i		next_cell	%next_cell
	UR51	UT51	i		out_ct	%out_ct
	UR51	UT51	o	%next_cell	in_ct	%out_ct

図 5-2 ユーザーFRFでの定義の例

### 5-2-2 “法則指定による制御” の場合

このツールでは、ユーザーがユーザーFRFの中で to\_next という Place に定められた mark を書き込むことで次のセルに制御を渡す法則を伝達することができる。また、始動されたセルでは receive という Place に法則を伝える mark が入れられており、その後の動きに反映される。このふたつの Place はシステムとユーザーがセルの移動を制御するために共用する特別な Place である。to\_next という Place から receive という Place への transition の定義は、システムFRFの中で記述されている。

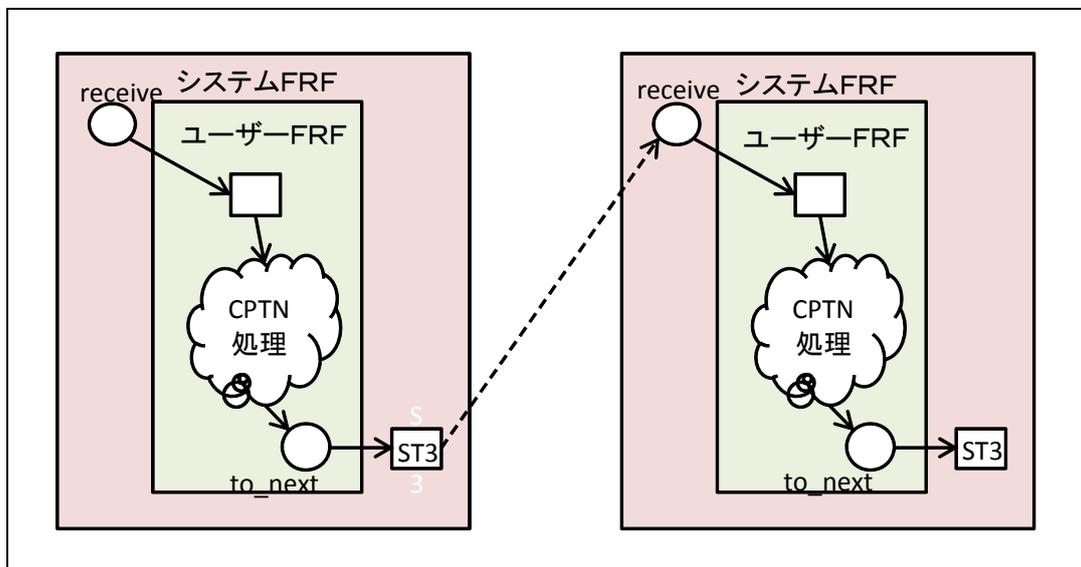


図 5-3 セル間の制御の概念図

システムFRFでのセル間の制御の定義を図 5-4に示す。

*	rule id	transition	input/ output	output_ cell id	place id	mark
*	SR30	ST3	i		to_next	<code>%=split[%o_towards:%o_color:%parm]</code>
*	SR30	ST3	o	<code>%o_towards</code>	receive	<code>%=concat[%o_towards:%o_color:%parm]</code>

図 5-4 to\_next に mark が入ったときの transition の定義 (システム F R F)

ユーザーが to\_next という Place に書く mark は、別のセルに制御を渡すときに引き渡すべき情報を結合したものである。システム F R F では、この mark を関数 split で次の 3 つの変数に分解している。

`%o_towards` 出力 Arc を書き出すセルを決定するための指定 (詳しくは前節を参照)

`%o_color` 色情報

`%parm` ユーザーが使うことのできるセル間で共用するパラメータ

ルール SR30 の出力 Arc の定義で output cell id が `%towards` となっている。これは place to\_next の mark の最初の変数である。プログラムは現在のセルの位置とここで指定された動きの規則に基づき、出力するセルを決定する。

開始画面で開始セルを指定すると、プログラムは初期処理で開始セルに receive place を書き込む。これにより、開始セルでペトリネットの実行が開始される。

## 6 ユーザー管理層

5章までで、このツールをほぼ説明してきた。この章では、ユーザーの視点に立って、どのようにセルペトリネットを定義・実行するかを見ていきたい。

### 6-1 ユーザー F R F の作成

F R F (Firing Rule File) は、発火条件となる入力 Arc とその transition の出力 Arc の定義をまとめて記述する。ただし、発火条件によっては、出力 Arc がない場合もある (ペトリネットを終了する場合など)。セルペトリネットを定義するためには、ユーザーはまず、ユーザー F R F を定義する。システム F R F と共用している Place として

- receive : この Place に mark がはいると、ペトリネットの実行が始まる
- to\_next : 別のセルの receive Place に mark を出力し、制御をわたす
- to\_display : 画面表示 (色を与える) を指示する

がある。そのほかの place、transition の定義は通常のペトリネットの定義と同じである。

receive、to\_next という Place の mark 情報は、決められた様式を満たす必要がある。

- %o\_towards : 出力 Arc を書き出すセルを決定するための指定 (5-1 節を参照)
- %o\_color : 色情報 (4-4 節を参照)
- %parm : ユーザーが使うことのできるセル間で共用するパラメータ

の3つのパラメータを “:” で結合したものである (5-2 節を参照)。

Place to\_next への出力が近傍の定義に相当する。この place を出力することで、システム transition が動き、別のセルの実行が開始される。

Place to\_display では mark として画面に出力したい色を指定する。この place を出力することで、システム transition が動き画面が出力される。

そのほかの place、transition はユーザーが自由に定義できる。これは、セルペトリネットの C P T N 定義に相当する部分である。より複雑なペトリネット定義を可能にするために変数や関数が用意されている (4-6 節を参照)。プログラムはユーザー F R F とその時の各セルのマーク状況を持つ M S T (Mark Status Table) の情報とで、ペトリネットを実行していく。

### 6-2 初期状態定義ファイルの作成

セルペトリネットを実行する場合、それぞれのセルに初期値として何らかの place、mark を与える必要がある時がある。実行する C P T N 定義は同じだが、セルの状態は同じとは限らないからである。たとえば、各セルの近傍やそのセルの特性などを初期状況ファイルに定義しておくことができる。初期状態を作るために、このツールでは初期状態定義ファイル（I N I）を用意した。初期状態を与えるセルの番号と、書き出す place、mark 情報を定義しておく。このファイルは任意であり、なくても構わない。I N I ファイルがある場合は、セルペトリネットの最初の実行の時にファイルが読み込まれ、各セルの初期状態を作り出す。7-3 節、7-4 節の実行例において、I N I ファイルを使用している。

### 6-3 セルペトリネットの実行

Web でセルペトリネット開始プログラムを実行し、開始画面で実行に必要な要件を指定する。指定する項目は次の通りである。

- ▶ ペトリネット定義名        :   ユーザー F R F の名前 <sup>(注1)</sup>
- ▶ 初期状態定義名   :   初期状態定義ファイル（I N I）の名前（任意） <sup>(注1)</sup>
- ▶ 実行セルの個数   :   X 軸方向、Y 軸方向それぞれに実行可能な最大の個数 <sup>(注2)</sup>
- ▶ 開始セル定義   (以下の指定を最高 4 組まで可能)
  - ◇ 開始位置           :   X 軸、Y 軸それぞれの位置をあらわす番号 <sup>(注2)</sup>
  - ◇ 動く方向           :   X 軸、Y 軸それぞれの次のセルを決定する規則 <sup>(注3)</sup>
  - ◇ 色                   :   表示する色を選択（赤／青／緑／白／灰色）
- ▶ 最大実行回数       :   実行する最大回数 <sup>(注4)</sup>
- ▶ 引数の画面表示   :   実行画面で開始要件を表示するか否かを選択（はい／いいえ）

(注1) ユーザー F R F、I N I は csv ファイルとしてあらかじめ実行フォルダーに作成しておく。  
ここでは、拡張子を除いたファイル名のみを指定する

(注2) 個数で指定する。セル番号は(0,0)からふられるため、個数として(5,5)と指定したら、有効なセル番号の範囲は(0,0)から(4,4)までの 25 個である。

(注3) 規則として、+、-、=、R、C 1、C 2 がプログラムで用意されている（5-1 参照）。  
しかし、ユーザーがユーザー F R F の中で自由に定義を与えることもできる。  
（第 7 章のサンプル（応用）を参照）

(注4) プログラムはいずれかのセルで画面出力がされない限り、何度もセルペトリネットの実行を繰り返す。画面出力の単位を 1 step とする。

## 7 実行例

### 7-1 基本的な動きの例

この例では、このセルのペトリネットの実行が開始されたら、赤を表示するよう Place to\_display を書き出し、そのままの動き、色で Place to\_next に書き出し、次のセルへ制御を渡す。ここでは、ユーザーパラメータ (Place parm) は使用しておらず、ただパラメータとして受け渡すのみ。

*	rule_id	transition	input/	output_	place_id	mark
*		id	output	cell_id		
*	いろいろな動きに対応するために方向(o_towards)、色(o_color)、パラメータ(parm)に分離する					
	UR21	UT2	i	=	receive	%=split[%o_towards:%o_color:%parm]
	UR21	UT2	o	=	to_display	%o_color
	UR21	UT2	o	=	o_color	%o_color
	UR21	UT2	o	=	o_towards	%o_towards
	UR21	UT2	o	=	parm	%parm
*	通常の動き(渡された動きの継続)					
	UR71	UT7	i		o_towards	%o_towards
	UR71	UT7	i		o_color	%o_color
	UR71	UT7	i		parm	%parm
	UR71	UT7	o	=	to_next	%=concat[%o_towards:%o_color:%parm]

図 7-1 User\_FRF\_基本.csv

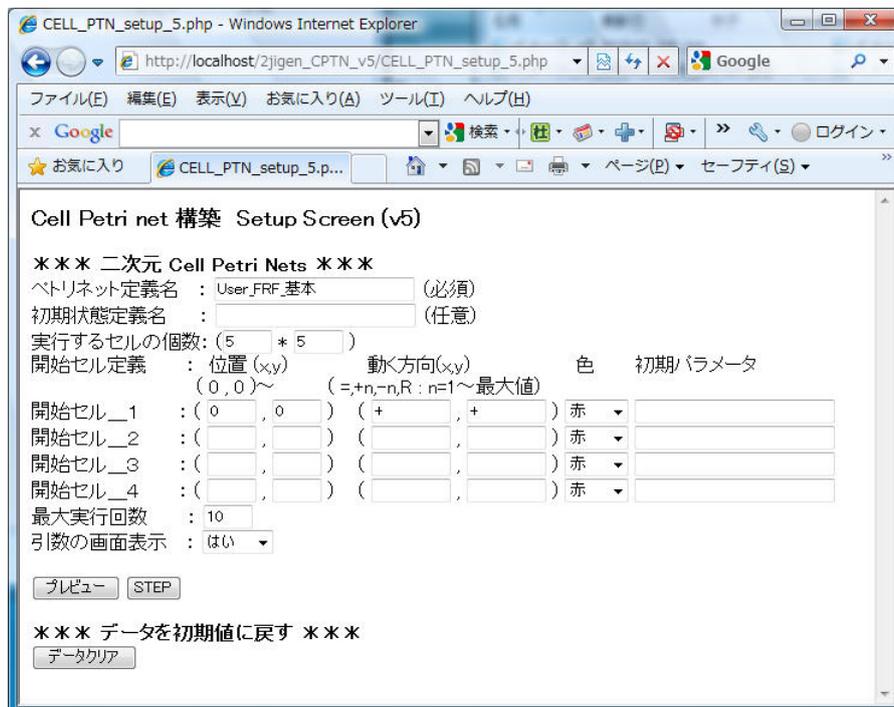


図 7-2 開始画面

開始画面で、開始セルは(0,0)、動く方向は(+,+)を指定しているため、実行画面では、“STEP”ボタンを押すたびに、左下隅から右斜め上に向かって1つずつセルの色が赤に変わっていく。6回実行するとセルペトリネットの実行が完了する。

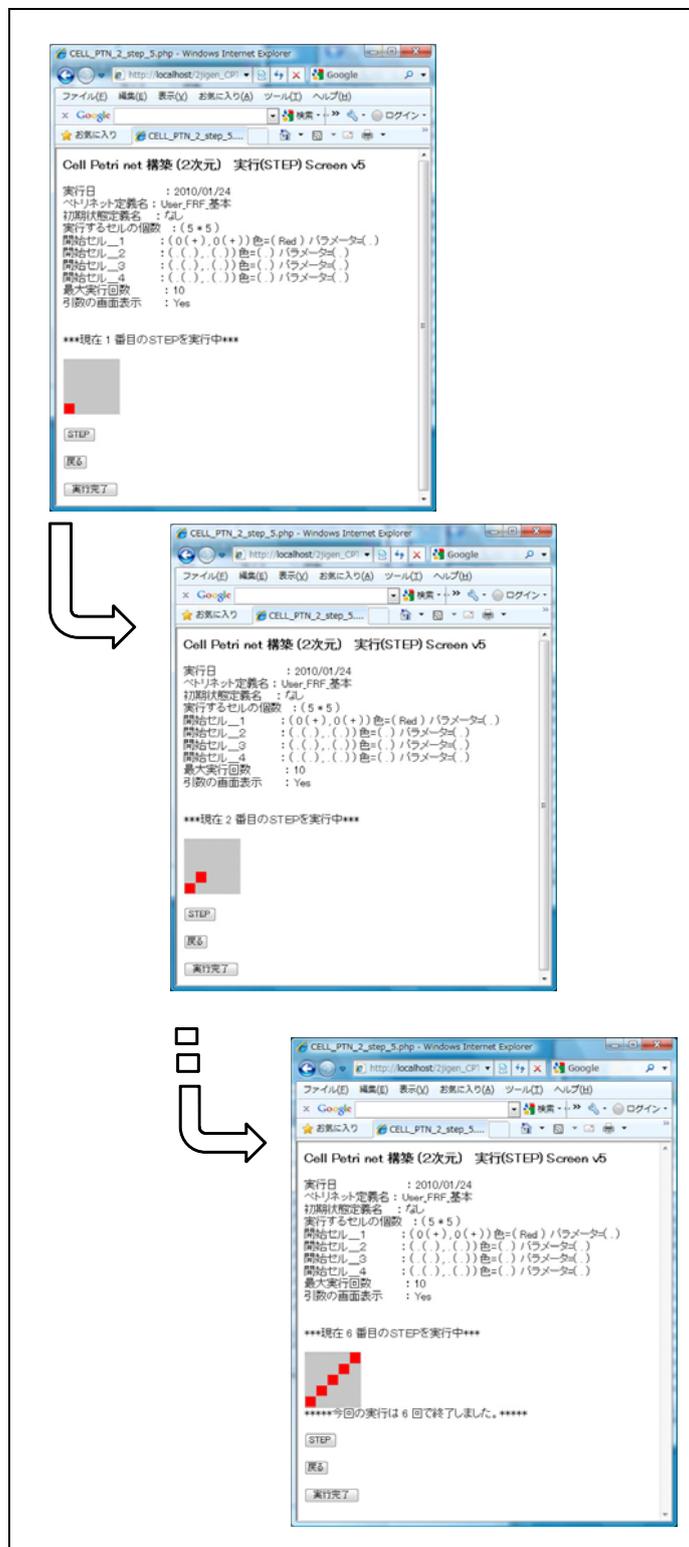


図 7-3 実行画面

## 7-2 複雑な動きの例

このサンプルでは、少し複雑な C P T N を定義している。また、近傍定義でも、あるセルからその周りのすべての方向に拡散する動きを加えている。

ルール UR41,UR42 では、開始画面で動く方向として U\_11,U\_21 と指定した動きをこのツールで提供されている一次方程式、二次方程式での計算に変換している。

ルール UR51 では、開始画面で動く方向として(R,R)つまり X 軸方向 Y 軸方向ともにランダムが指定された時には、ランダムの動きを継承しながら、そのセルから周囲のセルに拡散していく動き S\_S を加えている。拡散の動きは次のルール UR61 で定義している。このように、ユーザー F R F の中でセル移動の規則を加えたり、変更したりすることも可能である。

* rule_id	transition	input/	output_	place_id	mark
* id	output	cell_id			
* いろいろな動きに対応するために方向(o_towards)と色(o_color)とパラメーター(parm)に分離					
UR21	UT2	i		receive	%=split[%o_towards:%o_color:%parm]
UR21	UT2	o	=	to_display	%o_color
UR21	UT2	o	=	o_color	%o_color
UR21	UT2	o	=	o_towards	%o_towards
UR21	UT2	o	=	parm	%parm
* X軸方向の動きで指定した値をここで指定した数式で計算するように置き換える					
* 例1 $y=4x+3$ で計算					
UR41	UR4	i		o_towards	U_11
UR41	UR4	o	=	o_towards	C1_(4&3)
* 例2 $y=(-1)*x*x+2x+30$ で計算					
UR42	UR4	i		o_towards	U_21
UR42	UR4	o	=	o_towards	C2_(-1&2&30)
* ランダムの場合、ランダムに拡散の動きを加える					
UR51	UT5	i		o_towards	R_R
UR51	UT5	i		o_color	%o_color
UR51	UT5	i		parm	%parm
UR51	UT5	o	=	to_next	%=concat[R_R:%o_color:%parm]
UR51	UT5	o	=	o_towards	S_S
UR51	UT5	o	=	o_color	%o_color
UR51	UT5	o	=	parm	%parm
* 拡散(放射状にひろがる動き)					
UR61	UT6	i		o_towards	S_S
UR61	UT6	i		o_color	%o_color
UR61	UT6	i		parm	%parm
UR61	UT6	o	=	to_next	%=concat[+1_+1:%o_color:%parm]
UR61	UT6	o	=	to_next	%=concat[+1_=:%o_color:%parm]
UR61	UT6	o	=	to_next	%=concat[+1_-1:%o_color:%parm]
UR61	UT6	o	=	to_next	%=concat[=-+1:%o_color:%parm]
UR61	UT6	o	=	to_next	%=concat[=-1:%o_color:%parm]
UR61	UT6	o	=	to_next	%=concat[-1_+1:%o_color:%parm]
UR61	UT6	o	=	to_next	%=concat[-1_=:%o_color:%parm]
UR61	UT6	o	=	to_next	%=concat[-1_-1:%o_color:%parm]
* 通常の動き(渡された動きの継続)					
UR71	UT7	i		o_towards	%o_towards
UR71	UT7	i		o_color	%o_color
UR71	UT7	i		parm	%parm
UR71	UT7	o	=	to_next	%=concat[%o_towards:%o_color:%parm]

図 7-4 User\_FRF\_応用\_1.csv

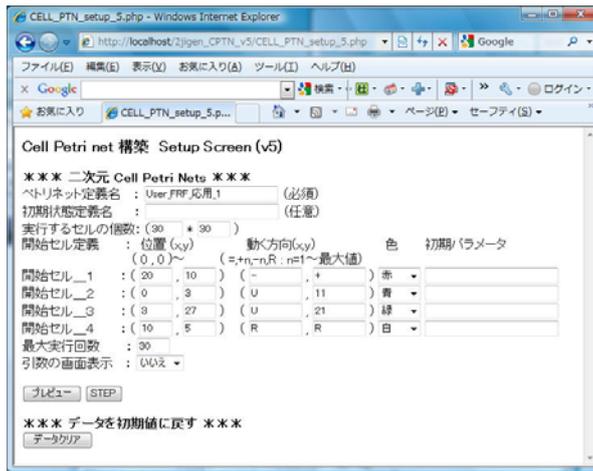


図 7-5 開始画面

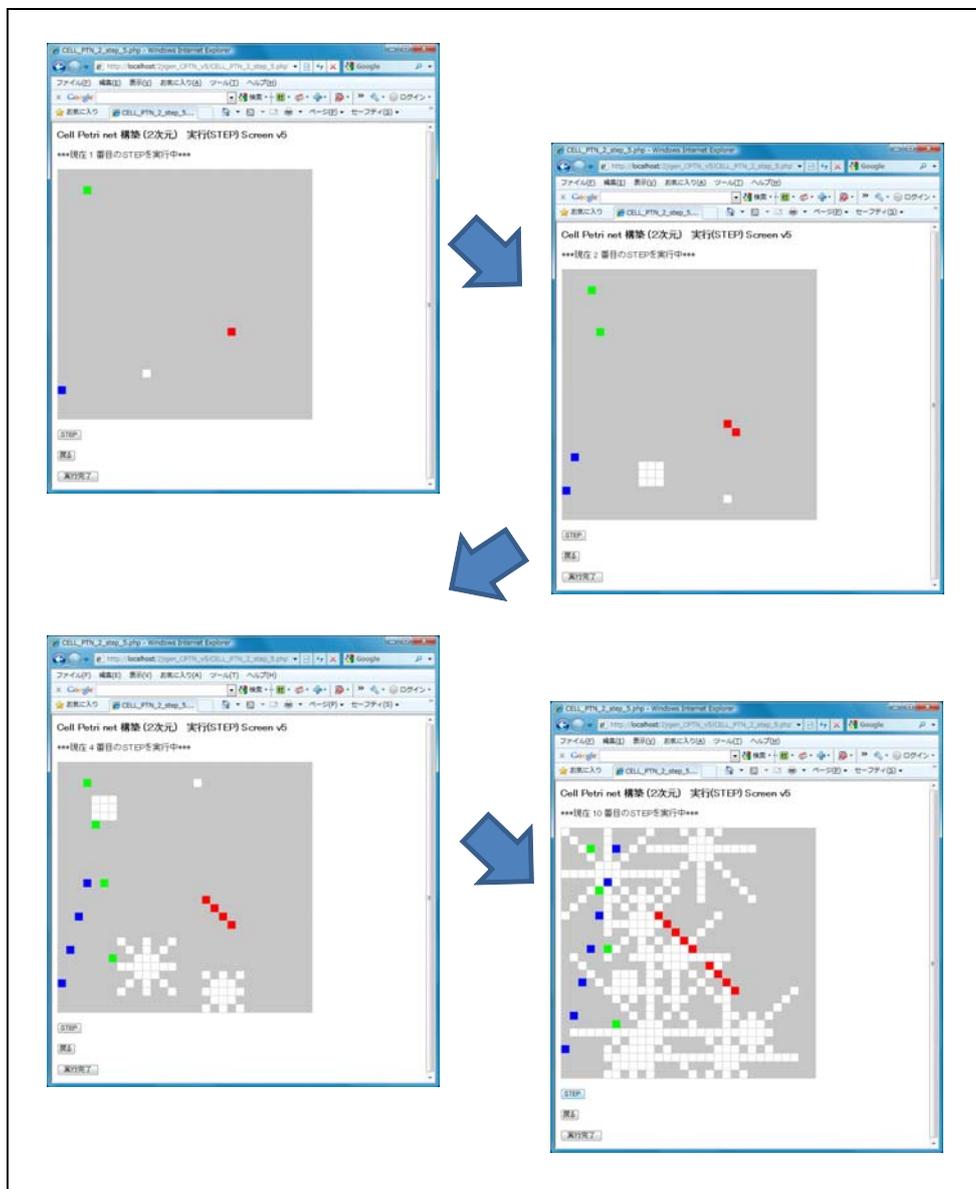


図 7-6 実行画面

### 7-3 鳥さんの散歩

このサンプルでは、セルの動き自体に少し意味（ストーリー）を持たせてみる。初期状態定義ファイル（INI\_torisan.csv）で、各セルに鳥の餌をいくつかずつ置いておく。開始画面で2羽の鳥を放つ。それぞれの鳥の足跡を青と緑であらわすことにする。鳥はセルの範囲でランダムに飛び回り、降りた地点で餌を食べる（UR31）。食べた餌の数が30個を超えたら、そこでおしまい（UR41,UR42,UR43）。セルの色を赤にして実行を完了する。セル間の共用パラメータ（%gokei）で食べた餌の合計を保持している。

```

* rule_id transition input/ output_ place_id mark
* id output cell_id
* 渡された引数を動く方向(o_towards)と色(o_color)と食べた餌の合計(gokei)に分離する
UR02 UT02 i receive %=split[%o_towards:%o_color:%gokei]
UR02 UT02 o = to_display %o_color
UR02 UT02 o = o_towards %o_towards
UR02 UT02 o = o_color %o_color
UR02 UT02 o = gokei %gokei

*
* ここで食べた餌の数を合計に加える
UR31 UT3 i esa %esa
UR31 UT3 i gokei %gokei
UR31 UT3 o = gokei %=+[%gokei:%esa]

* 餌を30個食べたら、色を赤にして動きを止める
UR41 UT4 i gokei 30
UR41 UT4 i o_towards %o_towards
UR41 UT4 i o_color %o_color
UR41 UT4 o = to_display Red

* 餌を31個食べたら、色を赤にして動きを止める
UR42 UT4 i gokei 31
UR42 UT4 i o_towards %o_towards
UR42 UT4 i o_color %o_color
UR42 UT4 o = to_display Red

* 餌を32個食べたら、色を赤にして動きを止める
UR43 UT4 i gokei 32
UR43 UT4 i o_towards %o_towards
UR43 UT4 i o_color %o_color
UR43 UT4 o = to_display Red

* 次のセルに移動
UR71 UT7 i o_towards %o_towards
UR71 UT7 i o_color %o_color
UR71 UT7 i gokei %gokei
UR71 UT7 o = to_next %=concat[%o_towards:%o_color:%gokei]

```

図 7-7 User\_FRF\_torisan.csv

このサンプルでは、初期状態定義ファイル（INI\_torisan.csv）で各セルに餌をいくつかつ置いている。

```

*   cell_id_x  cell_id_y  place_id mark
*   セルC_00000_00000～C_00009_00009までの各セルに餌をおく
      0         0 esa      3
      0         1 esa      2
      0         2 esa      3
      0         3 esa      1
      0         4 esa      3
      0         5 esa      2
      0         6 esa      3
      0         7 esa      3
      0         8 esa      1
      0         9 esa      3
      1         0 esa      2
      1         1 esa      3
      :
      :
      9         6 esa      3
      9         7 esa      3
      9         8 esa      1
      9         9 esa      3
*

```

図 7-8 INI\_torisan.csv

開始画面の開始セル\_2の初期パラメータで3を指定している。この値は開始時の place receive の mark の値として使われる。

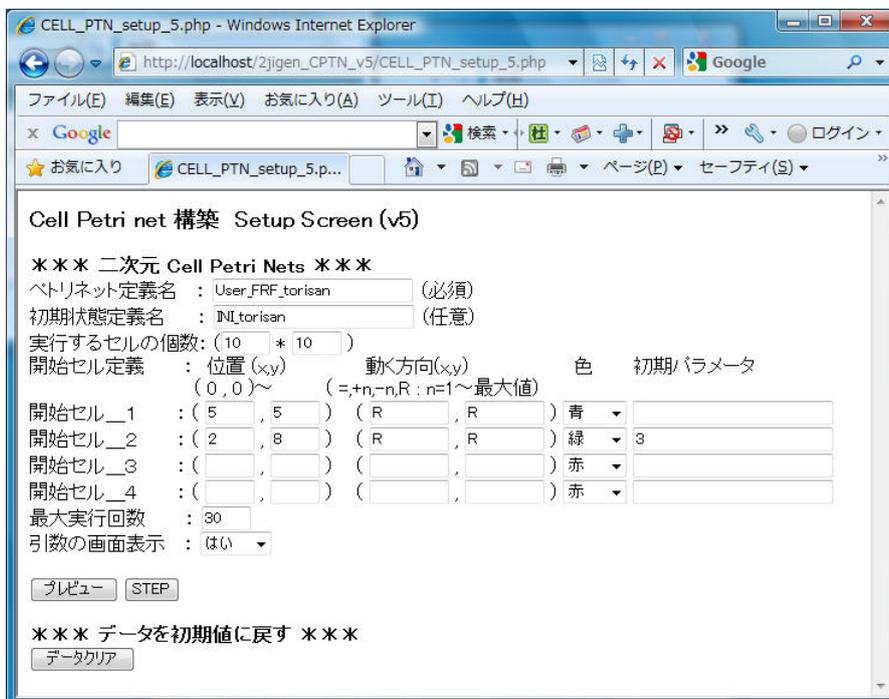


図 7-9 開始画面

Step1 では、開始画面で指定したセルに青い鳥、緑の鳥がそれぞれ止まっている。

Step2 では、それぞれランダムに場所を移動し、2個ずつ降り立った場所が表示されている。

Step10 では、緑の鳥が餌を30個以上食べたので、ここで色を赤にして終了

Step18 で青い鳥の動きも止まり、今回のセルパトリネットの実行は完了。

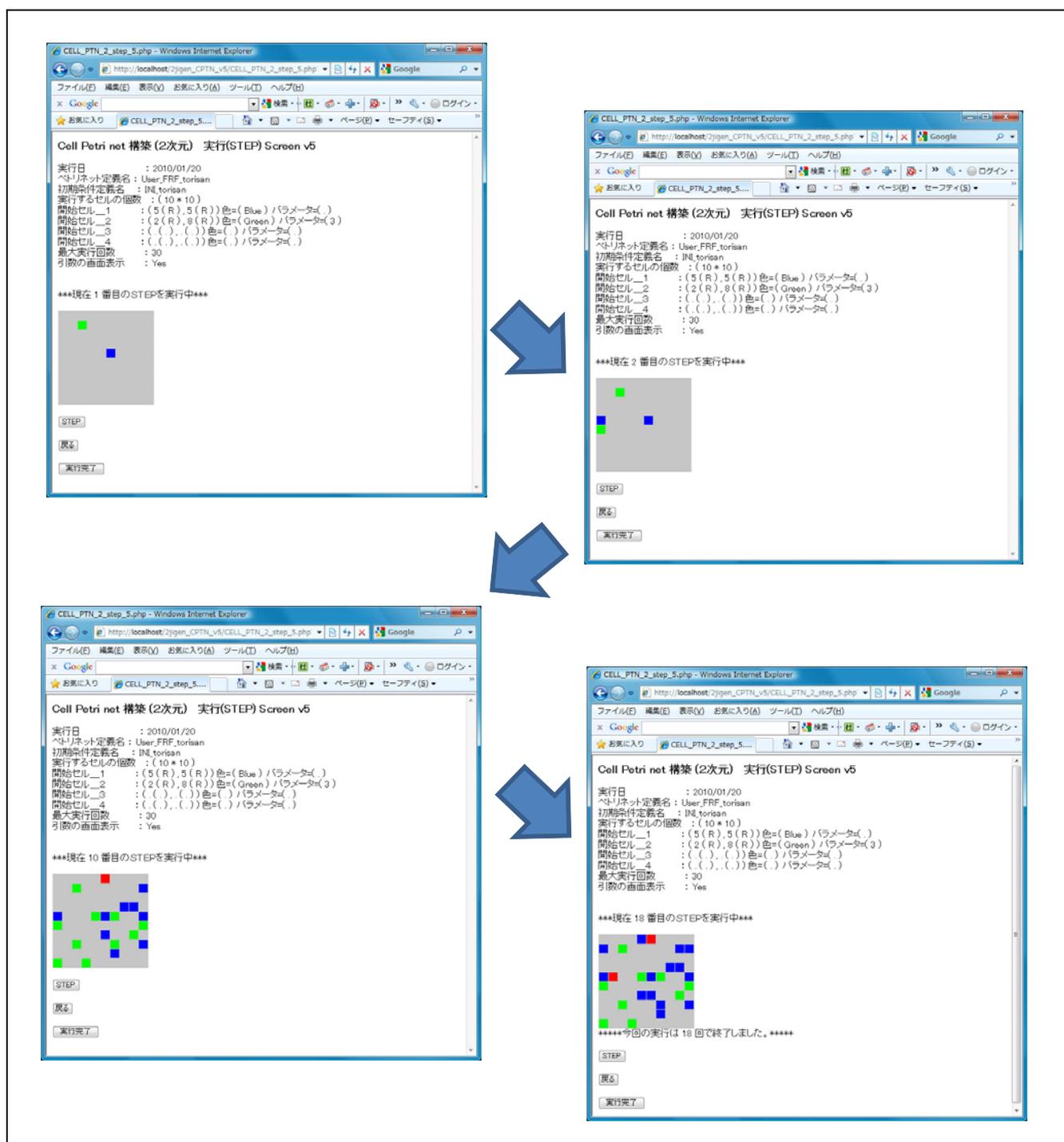


図 7-10 実行画面

次に、STEP2 実行後と STEP3 実行後の A C F の内容と、M S F の内容から変化の様子をみる。

Step2 実行後の UC\_00004\_00005 の M S F を見ると、receive R\_R:Green:7 という place があるので、緑の鳥がすでに餌を 7 個食べた状態でこのセルに飛んできたことがわかる。A C F にこのセルが含まれているので、Step3 でこのセルは実行される。このセルには esa 3 という place があるので、今までに食べた餌の合計値 7 に 3 を足して、次に渡す。Step3 実行後の M S F では UC\_00004\_00005 の place receive は実行後なのでなくなっている。また、餌 (place esa) も食べられてしまったのでなくなっている。同様に、青い鳥は Step2 実行後に UC\_00008\_00006 に飛んできていて、この時点で 4 個の餌を食べている。Step3 の実行で餌を 2 個食べ、次のセルに飛んでいく。

このように、A C F をみれば、次に実行されるセルの番号がわかり、M S F を調べれば、その時点でのセルの状態を見ることができる。

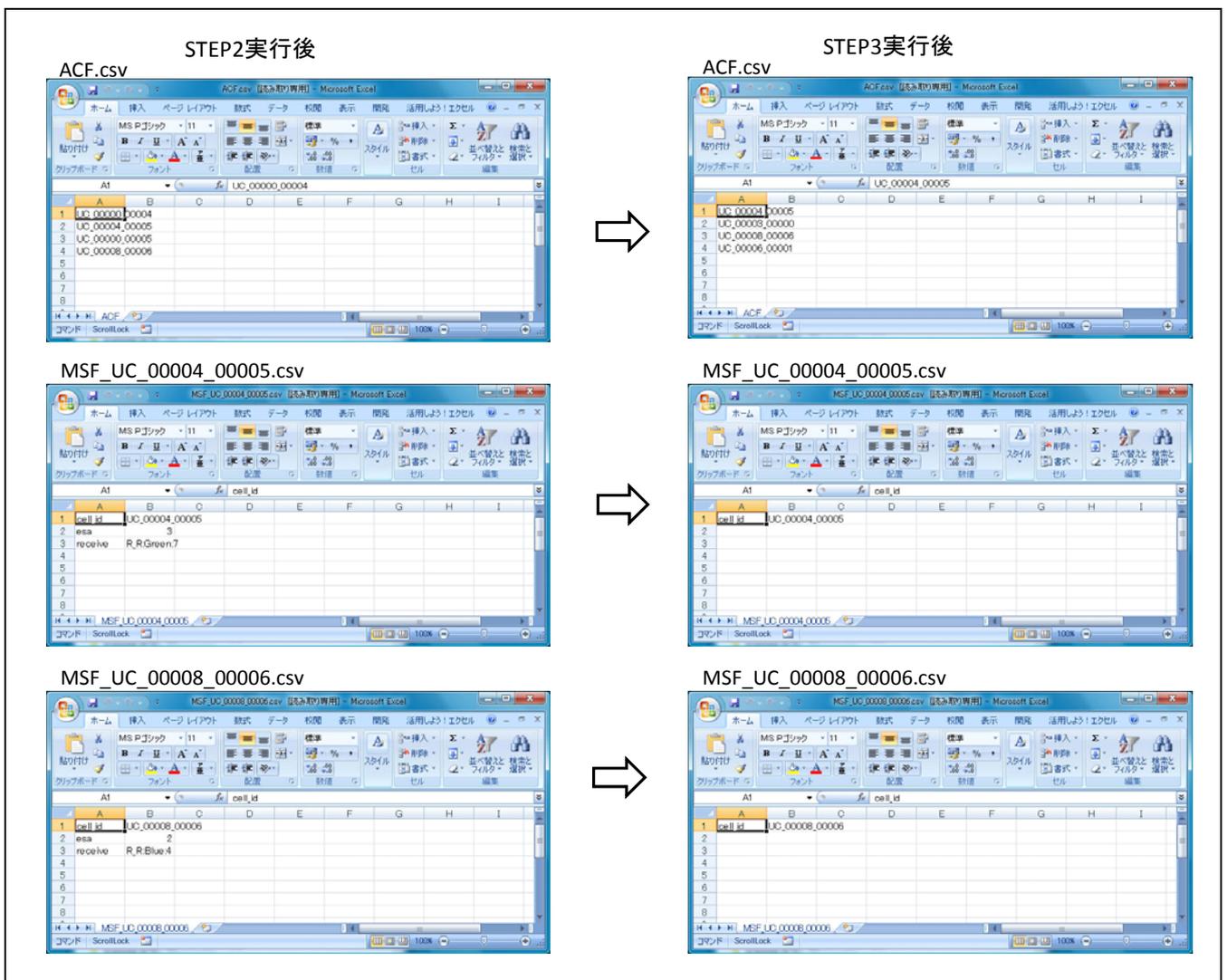


図 7-11 Step2,Step3 実行後の A C F, M S F の状態

## 7-4 LIFE\_GAME

このサンプルでは、条件によって今までの動きと異なる動きに変化させている。セルの範囲内に6個の岩がある。岩は青であらわされる (INI\_life\_game.csv)。開始セルから指定された方法で動きだした動物は、岩にぶつくと跳ね返る。跳ね返るとは、X軸方向には今まで通りの動き、Y軸方向は今までの動きに-1を掛けた動きであると定義する (UR31~UR34)。止まったセルに餌があれば食べ、10個以上食べたら止まる (UR41~UR43)。すべての動物が止まったか、セルの範囲内から消えた時、あるいは指定された最大実行回数に達した時にセルペトリネットの実行完了とする。

* cell_id_x	cell_id_y	place_id	mark
* 岩の設置			
5	5	iwa	yes
12	10	iwa	yes
2	18	iwa	yes
15	13	iwa	yes
8	0	iwa	yes
19	3	iwa	yes
* 岩の色を青にする			
5	5	to_display	Blue
12	10	to_display	Blue
2	18	to_display	Blue
15	13	to_display	Blue
8	0	to_display	Blue
19	3	to_display	Blue
* 餌の設置			
6	4	esa	1
14	14	esa	2
8	9	esa	3
10	2	esa	1
2	12	esa	3
3	5	esa	2
7	17	esa	3
1	1	esa	2
1	7	esa	1
13	3	esa	3
18	18	esa	1
6	18	esa	3
19	1	esa	2
8	2	esa	1
12	19	esa	3
4	4	esa	2
2	2	esa	1
6	2	esa	2
4	5	esa	2
19	0	esa	2
15	8	esa	3
16	3	esa	2
5	13	esa	2
9	15	esa	1
4	12	esa	1
17	9	esa	2
2	6	esa	1
11	7	esa	3
3	8	esa	1

図 7-12 INI\_life\_game.csv

```

* rule_id transition input/ output_ place_id mark
* id output cell_id
* 渡された引数を動く方向(o_towards)と色(o_color)と食べた餌の合計(gokei)に分離する
UR02 UT02 i receive %=split[%o_towards:%o_color:%gokei]
UR02 UT02 o = to_display %o_color
UR02 UT02 o = o_towards %o_towards
UR02 UT02 o = o_color %o_color
UR02 UT02 o = gokei %gokei
*
* ここで食べた餌の数を合計に加える
UR21 UT2 i esa %esa
UR21 UT2 i gokei %gokei
UR21 UT2 o = gokei %+[%gokei:%esa]
* 岩に当たったら反転する
* 動く方向をX軸、Y軸にわける
UR31 UT31 i iwa yes
UR31 UT31 i o_towards %=split_any[(.):%o_x:%o_y]
UR31 UT31 o = o_x %o_x
UR31 UT31 o = o_y %o_y
UR31 UT31 o = iwa yes
*
UR324 UT32 i o_y +
UR324 UT32 o = o_y 1
*
UR325 UT32 i o_y -
UR325 UT32 o = o_y -1
*
UR326 UT32 i o_y =
UR326 UT32 o = o_y 0
* Y軸の動きを反転させる計算
UR33 UT33 i o_y %o_y
UR33 UT33 o = o_y %=*[-1:%o_y]
* 岩にあたって反射させる
UR34 UT34 i o_x %o_x
UR34 UT34 i o_y %o_y
UR34 UT34 o = o_towards %=concat_any[(.):%o_x:%o_y]
* 岩の場所の色はいつでも青
UR35 UT35 i iwa yes
UR35 UT35 i to_display %o_color
UR35 UT35 o = to_display Blue
UR35 UT35 o = iwa yes
* 餌を10個以上食べたら終了
* 餌を10個食べたら動きを止める
UR41 UT4 i gokei 10
UR41 UT4 i o_towards %o_towards
UR41 UT4 i o_color %o_color
* 餌を11個食べたら動きを止める
UR42 UT4 i gokei 11
UR42 UT4 i o_towards %o_towards
UR42 UT4 i o_color %o_color
* 餌を12個食べたら動きを止める
UR43 UT4 i gokei 12
UR43 UT4 i o_towards %o_towards
UR43 UT4 i o_color %o_color
* 次のセルに移動
UR71 UT7 i o_towards %o_towards
UR71 UT7 i o_color %o_color
UR71 UT7 i gokei %gokei
UR71 UT7 o = to_next %=concat[%o_towards:%o_color:%gokei]

```

図 7-13 User\_FRF\_life\_game.csv

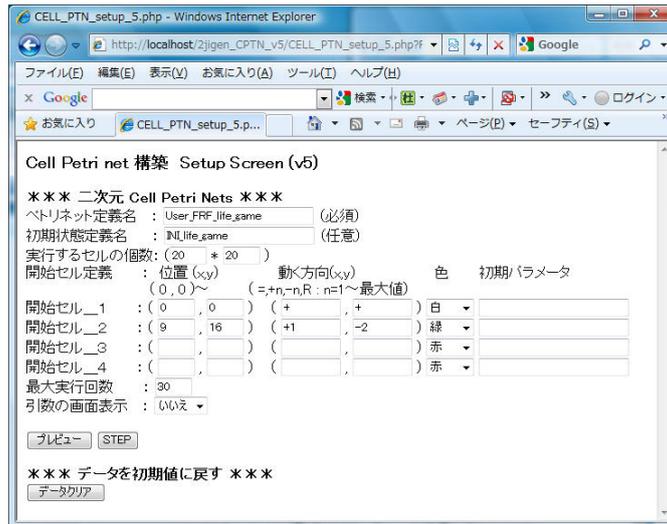


図 7-14 開始画面



図 7-15 実行画面

## 7-5 トークンパッシングの方式の伝送状態のシミュレーション

ネットワーク技術者養成プログラム(Nagano Solution)の教材“OS | モデルと伝送制御” [11]の6. 2. 4節トークンパッシング方式の伝送状態をシミュレーションしてみる。

今回のシミュレーションでは、端末を10個ネットワークにつなぐ。各端末をそれぞれセルとする。端末0（セル0）を開始セルとしてフリートークンを流し始め、フリートークンを受け取った端末に伝送データがあれば、データをトークンに入れて伝送路に戻す。端末9まで流れたら端末0にトークンをパスする。各セルが次にトークンを流す端末を近傍定義として初期情報定義ファイルに定義する。

また、

- 端末2から端末4へ
- 端末3から端末7へ
- 端末5から端末0へ
- 端末6から端末8へ
- 端末9から端末1へ

それぞれデータを伝送するよう、伝送データを初期情報定義ファイルに与えておく。

各端末ではパスされてきたトークン内のデータが自分宛てか判断し、自分宛てでなければ次の端末にパスしていく。自分宛のデータがきたらトークンからデータを取り出して空にして、フリートークンをまたネットワーク上に流す。データを受け取った端末では返信データを送り元に返すようデータを作成し、伝送を待つ。今回は、各端末1回ずつ送信を行ったら終了することとした。

シミュレーションの結果を各セルの色で表す。

このセルにトークンが回ってきて、フリートークンを伝送路に戻した場合は“白”、ビジートークンを戻した場合は“赤”とする。また、今回はトークンが回ってきていない端末については、伝送を待っているデータがある場合は“青”、伝送するデータがない場合は“緑”にすることとした。まだ一度もトークンが回ってきていない初期状態は“灰色”である。

このシミュレーションのペトリネット定義ファイル(FRF)、初期状況定義ファイル(INI)、開始画面は次のようになった。

* rule_id	transition	input/	output_	place_id	mark
* id	id	output	cell_id		
* テストのため、各端末からの送信は1回までとする					
UR01	UT0	i		send_kaisu	1
UR01	UT0	i		data	%data
UR01	UT0	o	=	send_kaisu	1
* ***** token を受け取った次の時の処理 *****					
* 前回tokenがきていて、現在転送するデータがあるときは、色を青にする					
UR11	UT1	i		token_color	%color
UR11	UT1	i		data	%data
UR11	UT1	o	=	data	%data
UR11	UT1	o	=	now_color	Blue
* 前回tokenがきていて、現在転送するデータがないときは、色を緑にする					
UR21	UT2	i		token_color	%color
UR21	UT2	o	=	now_color	Green
* 前回tokenがきたときの色(red/white)を伝送データ有無(blue/green)に変更する					
UR31	UT3	i		now_color	%color
UR31	UT3	o	=	to_display	%color
* ***** token を受け取った時の処理 *****					
* tokenを受け取ったら、色(busy=red,free=white)とtokenの内容に分ける					
UR51	UT5	i		receive	%=split[%o_towards:%o_color:%token]
UR51	UT5	o	=	o_color	%o_color
UR51	UT5	o	=	token	%token
* busy token が来たとき、					
UR61	UT61	i		o_color	Red
UR61	UT61	i		token	%=split_any[(&):%dest:%data]
UR61	UT61	o	=	o_color	Red
UR61	UT61	o	=	dest	%dest
UR61	UT61	o	=	token_data	%data
* 自分あてだったら、色を白にしてfree tokenを作る					
UR62	UT62	i		dest	%cell_id
UR62	UT62	i		token_data	%data
UR62	UT62	i		o_color	Red
UR62	UT62	o	=	o_color	White
UR62	UT62	o	=	token	.
UR62	UT62	o	=	next_dest	%=concat_any[(.):UC:%data:00000]
* 自分あてではなかったら、そのまま token をわたす準備をする					
UR63	UT63	i		dest	%dest
UR63	UT63	i		token_data	%data
UR63	UT63	o	=	token	%=concat_any[(&):%dest:%data]
* free token がきて、送りたいデータがあったら、色を赤にしてbusy tokenを作る					
UR71	UT7	i		o_color	White
UR71	UT7	i		token	%token
UR71	UT7	i		data	%data
UR71	UT7	i		send_kaisu	%kaisu
UR71	UT7	o	=	o_color	Red
UR71	UT7	o	=	token	%data
UR71	UT7	o	=	send_kaisu	%=+[%kaisu:1]
* tokenがあれば、次に送る					
UR81	UT8	i		token	%token
UR81	UT8	i		next_cell	%next_cell
UR81	UT8	i		o_color	%o_color
UR81	UT8	o	=	to_next	%=concat[%next_cell:%o_color:%token]
UR81	UT8	o	=	to_display	%o_color
UR81	UT8	o	=	next_cell	%next_cell
UR81	UT8	o	=	token_color	%o_color
* データが到着したら、次に送るデータの準備をする。					
UR92	UT92	i		next_dest	%dest
UR92	UT92	i		cell_id	%=split_any[(.):%a:%x:%y]
UR92	UT92	o	=	data	%=concat_any[(&):%dest:%x]
UR92	UT92	o	=	cell_id	%cell_id

図 7-16 User\_FRF-Token\_Passing.csv

*	cell_id_x	cell_id_y	place_id	mark
* 近傍定義(次のセル)				
	0	0	next_cell	UC_00001_00000
	1	0	next_cell	UC_00002_00000
	2	0	next_cell	UC_00003_00000
	3	0	next_cell	UC_00004_00000
	4	0	next_cell	UC_00005_00000
	5	0	next_cell	UC_00006_00000
	6	0	next_cell	UC_00007_00000
	7	0	next_cell	UC_00008_00000
	8	0	next_cell	UC_00009_00000
	9	0	next_cell	UC_00000_00000
* 伝送データ				
				転送先&データ
	2	0	data	UC_00004_00000&00002
	3	0	data	UC_00007_00000&00003
	5	0	data	UC_00000_00000&00005
	6	0	data	UC_00008_00000&00006
	9	0	data	UC_00001_00000&00009
* データ転送回数制御				
		place		
	0	0	send_kaisu	0
	1	0	send_kaisu	0
	2	0	send_kaisu	0
	3	0	send_kaisu	0
	4	0	send_kaisu	0
	5	0	send_kaisu	0
	6	0	send_kaisu	0
	7	0	send_kaisu	0
	8	0	send_kaisu	0
	9	0	send_kaisu	0
*				

図 7-17 INI\_token\_passing.csv

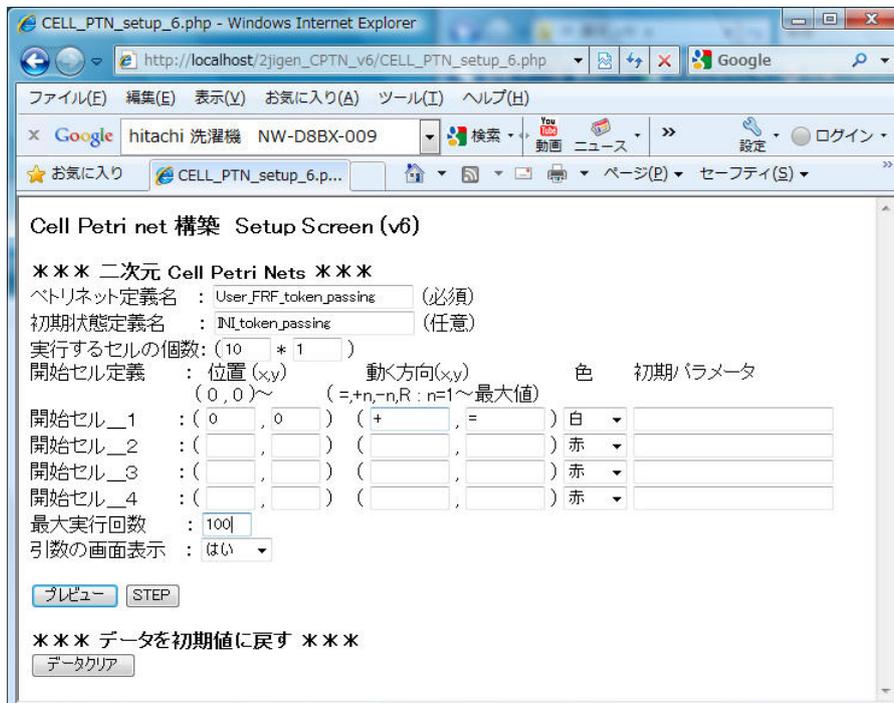


図 7-18 Token Passing 開始画面

また実行時の各セルの色の移り変わりは次のようになった。

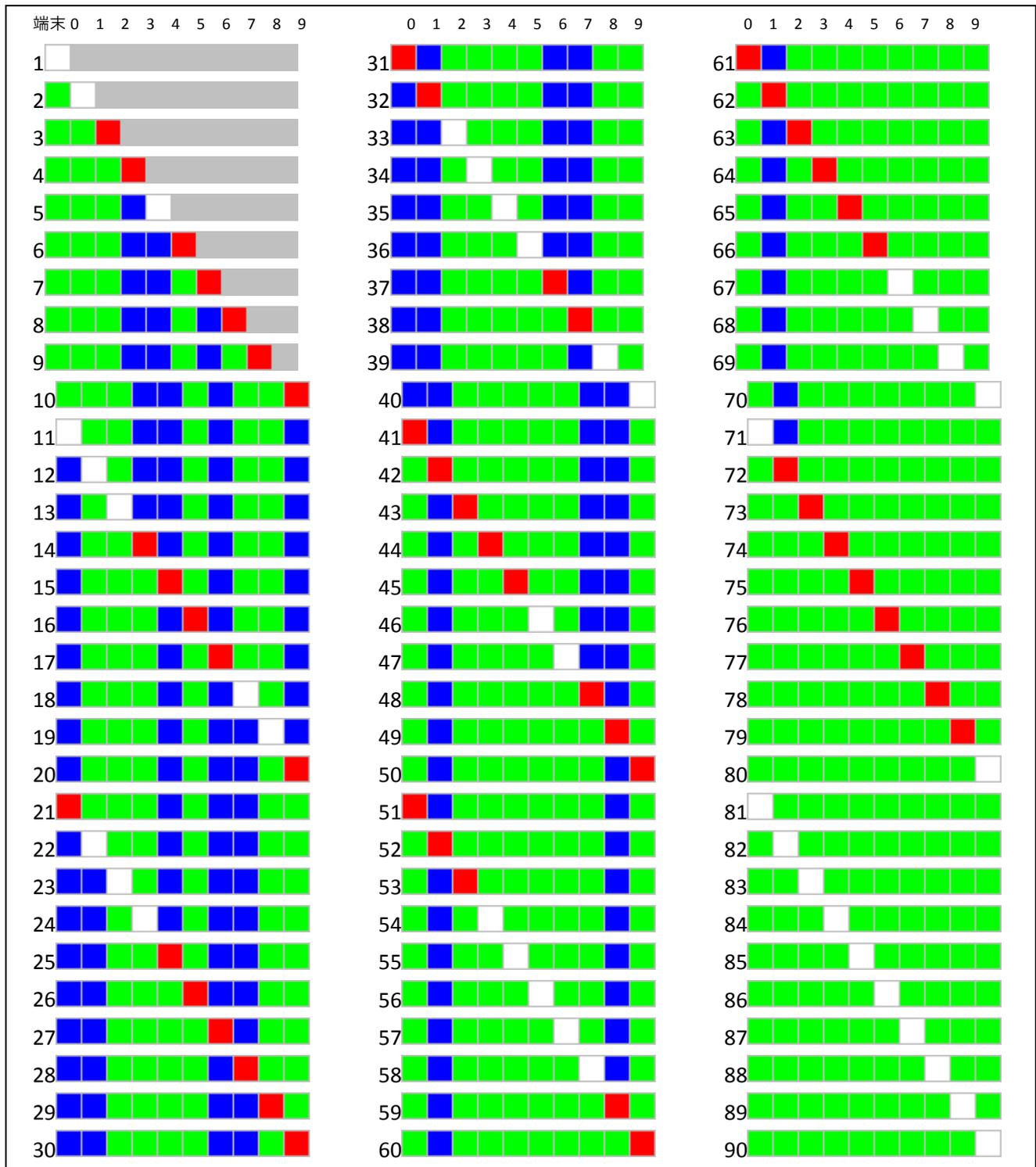


図 7-19 User\_FRF\_token\_passing を実行したときの結果 (step1 ~ step90)

白：free token が伝送路に戻された  
 赤：busy token が伝送路に戻された  
 灰色：初期値

青：伝送待ちデータあり  
 緑：伝送待ちデータなし

## 7-6 道路の渋滞予測

この例では、縦横それぞれ1本ずつ道があり、交差点では信号によって縦・横の動きを制御する。道路はそれぞれ区画にわけ、1区画を1セルとみなしてその時の渋滞状況を算出する。交差点も1区画であり、セルの範囲には建物など道路以外の区画もあるが、ここは処理の対象外となる。

道路および信号区画での基本的な処理の流れは、

- 前の区画から進入してくる車の量、現在区画に滞留している車の量、この区画を通行できる車の最大量から通行する車の量を計算する(UT41~UT44)、
- 次の区画から進入OK(gostatus walk)がきていれば次の区画に車を流す(UT51)
- この区画に滞留している車が指定された値を超えていなければ前の区画に進入OKを伝える(UT56)

それぞれの区画で毎回同じ処理が繰り返され、渋滞状況が予測される。

その区画が受け入れOKの場合は区画の色を緑で表示、滞留台数が規定値を超え、これ以上受け入れられない場合は“渋滞”と判断し、その区画の色を赤にする。信号区画では、横方向の通行が可能な時は青、縦方向の通行が可能な時は白にする。道路以外の場所は灰色で表示する。時間単位に各区画の渋滞状況を計算しながら画面表示していく。

この例のための初期状態定義(INI)、ペトリネット定義(FRF)、開始画面の指定は次のようになる。

*	cell_id_x	cell_id_y	place_id	mark		
* 道路の最後の区画にはいつでも通行可能を与える指示用place						
	4	2	control	walk		
	3	4	control	walk		
* 道情報(縦方向/横方向)						
	0	2	road_status_yoko			
	1	2	road_status_yoko			
	2	2	road_status_mae_y			
	3	2	road_status_singou			
	4	2	road_status_ato_y			
*						
	3	0	road_status_tate			
	3	1	road_status_mae_t			
	3	2	road_status_singou			
	3	3	road_status_ato_t			
	3	4	road_status_tate			
*						
* 信号情報						
	3	2	singou	yes		
	3	2	s_value	2		
	3	2	yoko_time	2		
	3	2	tate_time	2		
	3	2	total_time	4	yoko_time + tate_time	

(次ページに続く)

* 近傍定義				
* 通常の走行で次に進む区画(横方向)				
	0	2 next_cell	UC_00001_00002	
	1	2 next_cell	UC_00002_00002	
	2	2 next_cell	UC_00003_00002	
	3	2 next_cell_y	UC_00004_00002	
	4	2 next_cell	UC_00005_00002	(処理されないセル)
* 車の進入OK(gostatus walk)を伝える区画(横方向)				
	0	2 prev_cell	UC_00005_00002	(処理されないセル)
	1	2 prev_cell	UC_00000_00002	
	2	2 prev_cell	UC_00001_00002	
	3	2 prev_cell_y	UC_00002_00002	
	4	2 prev_cell	UC_00003_00002	
* 通常の走行で次に進む区画(縦方向)				
	3	0 next_cell	UC_00003_00001	
	3	1 next_cell	UC_00003_00002	
	3	2 next_cell_t	UC_00003_00003	
	3	3 next_cell	UC_00003_00004	
	3	4 next_cell	UC_00003_00005	(処理されないセル)
* 車の進入OK(gostatus walk)を伝える区画(縦方向)				
	3	0 prev_cell	UC_00003_00005	(処理されないセル)
	3	1 prev_cell	UC_00003_00000	
	3	2 prev_cell_t	UC_00003_00001	
	3	3 prev_cell	UC_00003_00002	
	3	4 prev_cell	UC_00003_00003	
* 車の流入台数				
	0	2 input_ct	2	横方向
	3	0 input_ct	3	縦方向
* 最大通過代数(next_cellに移すことのできる最大の台数)				
	0	2 max_out	5	
	1	2 max_out	5	
	2	2 max_out	5	
	3	2 max_out_y	5	
	4	2 max_out	5	
*				
	3	0 max_out	4	
	3	1 max_out	4	
	3	2 max_out_t	3	
	3	3 max_out	3	
	3	4 max_out	3	
* オーバーフローする台数(滞留している車がこの数を超えたらprev_cellに超えた分を戻す)				
	0	2 of_ct	7	
	1	2 of_ct	7	
	2	2 of_ct	6	
	3	2 of_ct_y	6	
	4	2 of_ct	6	
*				
	3	0 of_ct	5	
	3	1 of_ct	5	
	3	2 of_ct_t	4	
	3	3 of_ct	4	
	3	4 of_ct	4	
* 通過できなかった台数(初期値=0)				
	0	2 stuck_ct	0	
	1	2 stuck_ct	0	
	2	2 stuck_ct	0	
	3	2 stuck_ct_y	0	
	4	2 stuck_ct	0	
*				
	3	0 stuck_ct	0	
	3	1 stuck_ct	0	
	3	2 stuck_ct_t	0	
	3	3 stuck_ct	0	
	3	4 stuck_ct	0	
* 初期化処理				
	3	4 to_next	==:Green:0	
	3	3 to_next	==:Green:0	
	4	2 to_next	==:Green:0	
	3	2 to_next	==:Green:0	
	2	2 to_next	==:Green:0	
	1	2 to_next	==:Green:0	
	0	2 to_next	==:Green:0	
	3	1 to_next	==:Green:0	
	3	0 to_next	==:Green:0	

図 7-20 INI\_traffic\_jam\_20.csv

* rule_id	transition	input/ output	output cell_id	place_id	mark
* UR11	UT11	i		receive	%data
* UR11	UT11	o	=	to_next	%data
* 流入量 (input_ct) placeがあれば、指定台数ずつ毎回車を流入させる ただし、道路の最初の区画にかぎる					
UR12	UT12	i		input_ct	%input_ct
UR12	UT12	o	=	in_ct	%input_ct
UR12	UT12	o	=	input_ct	%input_ct
* 道路の最後の区画はいつでも通行可能にする					
UR13	UT13	i		control	walk
UR13	UT13	o	=	go_status	walk
UR13	UT13	o	=	control	walk
* 区画の色を赤にする(default)					
UR14	UT14	i		road_status	%status
UR14	UT14	o	=	o_color	Red
UR14	UT14	o	=	road_status	%status
* *** 信号区画の処理 ***					
* 信号の状態を計算(1)					
UR21	UT21	i		road_status	singou
UR21	UT21	i		s_value	%s_value
UR21	UT21	o	=	s_value	%=[%s_value:1]
UR21	UT21	o	=	road_status	singou
* 信号の状態を計算(2)					
UR22	UT22	i		road_status	singou
UR22	UT22	i		s_value	%s_value
UR22	UT22	i		total_time	%total_time
UR22	UT22	o	=	hantei_1	%=>[%s_value:%total_time]
UR22	UT22	o	=	s_value	%s_value
UR22	UT22	o	=	total_time	%total_time
UR22	UT22	o	=	road_status	singou
* 信号の状態を計算(3)					
UR23	UT23	i		road_status	singou
UR23	UT23	i		hantei_1	1
UR23	UT23	i		s_value	%s_value
UR23	UT23	o		s_value	1
UR23	UT23	o	=	road_status	singou
* 信号を判定する準備					
UR24	UT24	i		road_status	singou
UR24	UT24	i		yoko_time	%yoko_time
UR24	UT24	i		s_value	%s_value
UR24	UT24	o	=	hantei_2	%=>[%s_value:%yoko_time]
UR24	UT24	o	=	s_value	%s_value
UR24	UT24	o	=	yoko_time	%yoko_time
UR24	UT24	o	=	road_status	singou
* 信号が横方向が青だったら singou を yoko にする					
UR25	UT25	i		road_status	singou
UR25	UT25	i		hantei_2	0
UR25	UT25	i		singou	%status
UR25	UT25	o	=	singou	yoko
UR25	UT25	o	=	road_status	singou
* 信号が縦方向が青だったら singou を tate にする					
UR26	UT26	i		road_status	singou
UR26	UT26	i		hantei_2	1
UR26	UT26	i		singou	%status
UR26	UT26	o	=	singou	tate
UR26	UT26	o	=	road_status	singou
* *** 信号での処理					
* *** 横方向の通行に関する処理					
* 前(左)の区画から入ってきた車(input_y)を滞留量(stuck_y)に加える					
UR31y	UT31y	i		road_status	singou
UR31y	UT31y	i		in_ct_y	%input
UR31y	UT31y	i		stuck_ct_y	%stuck
UR31y	UT31y	o	=	stuck_ct_y	%=[%stuck:%input]
UR31y	UT31y	o	=	road_status	singou
* 前回の横区画(右)に流そうとした量(output_y)を破棄する					
UR32y	UT32y	i		out_ct_y	%output
* 信号が横方向で赤だったら、横方向は通行可能place(go_status)を破棄する					
UR33y	UT33y	i		road_status	singou
UR33y	UT33y	i		singou	tate
UR33y	UT33y	i		go_status_y	walk
UR33y	UT33y	o	=	singou	tate
UR33y	UT33y	o	=	road_status	singou

* 信号が横方向で青だった時の変数の変換処理					
* 定数値 + stuck_ct_y					
	UR35y	UT35y	i		road_status singou
	UR35y	UT35y	i		singou yoko
	UR35y	UT35y	i		stuck_ct_y %stuck
	UR35y	UT35y	i		max_out_y %max_out
	UR35y	UT35y	i		of_ct_y %of_ct
	UR35y	UT35y	i		next_cell_y %next_cell
	UR35y	UT35y	i		prev_cell_y %prev_cell
	UR35y	UT35y	o	=	stuck_ct %stuck
	UR35y	UT35y	o	=	max_out %max_out
	UR35y	UT35y	o	=	of_ct %of_ct
	UR35y	UT35y	o	=	next_cell %next_cell
	UR35y	UT35y	o	=	prev_cell %prev_cell
	UR35y	UT35y	o	=	max_out_y %max_out
	UR35y	UT35y	o	=	of_ct_y %of_ct
	UR35y	UT35y	o	=	next_cell_y %next_cell
	UR35y	UT35y	o	=	prev_cell_y %prev_cell
	UR35y	UT35y	o	=	singou yoko
	UR35y	UT35y	o	=	road_status singou
* 通行可能place(go_status walk_y)					
	UR36y	UT36y	i		road_status singou
	UR36y	UT36y	i		singou yoko
	UR36y	UT36y	i		go_status_y %status
	UR36y	UT36y	o	=	go_status %status
	UR36y	UT36y	o	=	singou yoko
	UR36y	UT36y	o	=	road_status singou
*					
* *** 縦方向の通行に関する処理					
* 前(下)の区画から入ってきた車(input_t)を滞留量(stuck_t)に加える					
	UR31t	UT31t	i		road_status singou
	UR31t	UT31t	i		in_ct_t %input
	UR31t	UT31t	i		stuck_ct_t %stuck
	UR31t	UT31t	o	=	stuck_ct_t %=-+[%stuck:%input]
	UR31t	UT31t	o	=	road_status singou
* 前回の縦区画(上)に流そうとした量(output_t)を破棄する					
	UR32t	UT32t	i		out_ct_t %output
* 信号が縦方向で赤だったら、縦方向は通行可能place(go_status_t)を破棄する					
	UR33t	UT33t	i		road_status singou
	UR33t	UT33t	i		singou yoko
	UR33t	UT33t	i		go_status_t walk
	UR33t	UT33t	o	=	singou yoko
	UR33t	UT33t	o	=	road_status singou
* 信号が縦方向で青だった時の変数の変換処理					
* 定数値 + stuck_ct_t					
	UR35t	UT35t	i		road_status singou
	UR35t	UT35t	i		singou tate
	UR35t	UT35t	i		stuck_ct_t %stuck
	UR35t	UT35t	i		max_out_t %max_out
	UR35t	UT35t	i		of_ct_y %of_ct
	UR35t	UT35t	i		next_cell_t %next_cell
	UR35t	UT35t	i		prev_cell_t %prev_cell
	UR35t	UT35t	o	=	stuck_ct %stuck
	UR35t	UT35t	o	=	max_out %max_out
	UR35t	UT35t	o	=	of_ct %of_ct
	UR35t	UT35t	o	=	next_cell %next_cell
	UR35t	UT35t	o	=	prev_cell %prev_cell
	UR35t	UT35t	o	=	max_out_t %max_out
	UR35t	UT35t	o	=	of_ct_y %of_ct
	UR35t	UT35t	o	=	next_cell_t %next_cell
	UR35t	UT35t	o	=	prev_cell_t %prev_cell
	UR35t	UT35t	o	=	singou tate
	UR35t	UT35t	o	=	road_status singou
* 通行可能place(go_status walk)					
	UR36t	UT36t	i		road_status singou
	UR36t	UT36t	i		singou tate
	UR36t	UT36t	i		go_status_t %status
	UR36t	UT36t	o	=	go_status %status
	UR36t	UT36t	o	=	singou tate
	UR36t	UT36t	o	=	road_status singou

*									
* ***	今この区画にある車の台数を計算 ***								
*									
* 前の区画から流入した量(input_1)を現在の滞留量(stuck_ct)に加える									
	UR41	UT41	i		in_ct	%input			
	UR41	UT41	i		stuck_ct	%stuck_ct			
	UR41	UT41	o	=	stuck_ct	%=[%stuck_ct:%input]			
* 前回の区画に流そうとした量(output)を破棄する									
	UR42	UT42	i		out_ct	%output			
* 次の区画に流す車があるかを判定する準備									
	UR43	UT43	i		stuck_ct	%stuck_ct			
	UR43	UT43	o	=	hantei_3	%=>[%stuck_ct:0]			
	UR43	UT43	o	=	stuck_ct	%stuck_ct			
* 次の区画に流す車の量を計算する									
	UR44	UT44	i		hantei_3	1			
	UR44	UT44	i		stuck_ct	%stuck_ct			
	UR44	UT44	i		max_out	%max_out			
	UR44	UT44	o	=	out_ct	%=min[%stuck_ct:%max_out]			
	UR44	UT44	o	=	stuck_ct	%stuck_ct			
	UR44	UT44	o	=	max_out	%max_out			
*									
* 次の区画に車を流す									
* 信号の直前の区画(横方向)									
	UR51y	UT51y	i		go_status	walk			
	UR51y	UT51y	i		road_status	mae_y			
	UR51y	UT51y	i		out_ct	%out_ct			
	UR51y	UT51y	i		next_cell	%next_cell			
	UR51y	UT51y	i		stuck_ct	%stuck_ct			
	UR51y	UT51y	i		o_color	%color			
	UR51y	UT51y	o	%next_cell	in_ct_y	%out_ct			
	UR51y	UT51y	o	=	stuck_ct	%=[%stuck_ct:%out_ct]			
	UR51y	UT51y	o	=	o_color	Green			
	UR51y	UT51y	o	=	next_cell	%next_cell			
	UR51y	UT51y	o	=	road_status	mae_y			
* 信号の直前の区画(縦方向)									
	UR51t	UT51t	i		go_status	walk			
	UR51t	UT51t	i		road_status	mae_t			
	UR51t	UT51t	i		out_ct	%out_ct			
	UR51t	UT51t	i		next_cell	%next_cell			
	UR51t	UT51t	i		stuck_ct	%stuck_ct			
	UR51t	UT51t	i		o_color	%color			
	UR51t	UT51t	o	%next_cell	in_ct_t	%out_ct			
	UR51t	UT51t	o	=	stuck_ct	%=[%stuck_ct:%out_ct]			
	UR51t	UT51t	o	=	o_color	Green			
	UR51t	UT51t	o	=	next_cell	%next_cell			
	UR51t	UT51t	o	=	road_status	mae_y			
* 通常の区画および信号区画									
	UR51	UT51	i		go_status	walk			
	UR51	UT51	i		out_ct	%out_ct			
	UR51	UT51	i		next_cell	%next_cell			
	UR51	UT51	i		stuck_ct	%stuck_ct			
	UR51	UT51	i		o_color	%color			
	UR51	UT51	o	%next_cell	in_ct	%out_ct			
	UR51	UT51	o	=	stuck_ct	%=[%stuck_ct:%out_ct]			
	UR51	UT51	o	=	o_color	Green			
	UR51	UT51	o	=	next_cell	%next_cell			
*									
* 現在この区画に残っている車がoverflowポイントを超えていないかを判定する準備									
	UR55	UT55	i		stuck_ct	%stuck_ct			
	UR55	UT55	i		of_ct	%of_ct			
	UR55	UT55	o	=	hantei_4	%=>[%of_ct:%stuck_ct]			
	UR55	UT55	o	=	stuck_ct	%stuck_ct			
	UR55	UT55	o	=	of_ct	%of_ct			
* overflow pointを超えていなかったら、前の区画に通行可能(go_status walk)を書き込む									
* 信号の直後の区画(横方向)									
	UR56y	UT56y	i		hantei_4	1			
	UR56y	UT56y	i		road_status	ato_y			
	UR56y	UT56y	i		prev_cell	%prev_cell			
	UR56y	UT56y	i		o_color	%color			
	UR56y	UT56y	o	%prev_cell	go_status_y	walk			
	UR56y	UT56y	o	=	prev_cell	%prev_cell			
	UR56y	UT56y	o	=	o_color	Green			
	UR56y	UT56y	o	=	road_status	ato_y			

* 信号の直後の区画(縦方向)						
	UR56t	UT56t	i		hantei_4	1
	UR56t	UT56t	i		road_status	ato_t
	UR56t	UT56t	i		prev_cell	%prev_cell
	UR56t	UT56t	i		o_color	%color
	UR56t	UT56t	o	%prev_cell	go_status_t	walk
	UR56t	UT56t	o	=	prev_cell	%prev_cell
	UR56t	UT56t	o	=	o_color	Green
	UR56t	UT56t	o	=	road_status	ato_t
* 通常の区画および信号区画						
	UR56	UT56	i		hantei_4	1
	UR56	UT56	i		prev_cell	%prev_cell
	UR56	UT56	i		o_color	%color
	UR56	UT56	o	%prev_cell	go_status	walk
	UR56	UT56	o	=	prev_cell	%prev_cell
	UR56	UT56	o	=	o_color	Green
*						
* *** 区画の色を決定し、出力する ***						
*						
* 信号区画の色を決定する(信号が横方向青の場合)						
	UR61	UT61	i		road_status	singou
	UR61	UT61	i		singou	yoko
	UR61	UT61	i		o_color	%color
	UR61	UT61	o	=	o_color	Blue
	UR61	UT61	o	=	singou	yoko
	UR61	UT61	o	=	road_status	singou
* 信号区画の色を決定する(信号が縦方向青の場合)						
	UR62	UT62	i		road_status	singou
	UR62	UT62	i		singou	tate
	UR62	UT62	i		o_color	%color
	UR62	UT62	o	=	o_color	White
	UR62	UT62	o	=	singou	tate
	UR62	UT62	o	=	road_status	singou
* その区画の色を出す						
	UR65	UT65	i		o_color	%color
	UR65	UT65	o	=	to_display	%color
*						
* *** 信号の区画で、変数をもどす						
* *** 横方向の処理						
* 信号が横方向青のとき(stuck_ct_y)						
	UR81y	UT81y	i		road_status	singou
	UR81y	UT81y	i		singou	yoko
	UR81y	UT81y	i		stuck_ct	%stuck_ct
	UR81y	UT81y	o	=	stuck_ct_y	%stuck_ct
	UR81y	UT81y	o	=	singou	yoko
	UR81y	UT81y	o	=	road_status	singou
* 信号が横方向青のとき(go_status_y)						
	UR82y	UT82y	i		road_status	singou
	UR82y	UT82y	i		singou	yoko
	UR82y	UT82y	i		go_status	%status
	UR82y	UT82y	o	=	go_status_y	%status
	UR82y	UT82y	o	=	singou	yoko
	UR82y	UT82y	o	=	road_status	singou
* *** 縦方向の処理						
* 信号が横方向青のとき(stuck_ct_y)						
	UR81t	UT81t	i		road_status	singou
	UR81t	UT81t	i		singou	tate
	UR81t	UT81t	i		stuck_ct	%stuck_ct
	UR81t	UT81t	o	=	stuck_ct_t	%stuck_ct
	UR81t	UT81t	o	=	singou	tate
	UR81t	UT81t	o	=	road_status	singou
* 信号が横方向青のとき(go_status_y)						
	UR82t	UT82t	i		road_status	singou
	UR82t	UT82t	i		singou	tate
	UR82t	UT82t	i		go_status	%status
	UR82t	UT82t	o	=	go_status_t	%status
	UR82t	UT82t	o	=	singou	tate
	UR82t	UT82t	o	=	road_status	singou
* 信号区画での定数値の破棄						
	UR85	UT85	i		road_status	singou
	UR85	UT85	i		max_out	%max_out
	UR85	UT85	i		of_ct	%of_ct
	UR85	UT85	i		next_cell	%next_cell
	UR85	UT85	i		prev_cell	%prev_cell
	UR85	UT85	o	=	road_status	singou

* *** データクリア							
	UR91	UT91	i		hantei_1	%data	
	UR92	UT91	i		hantei_2	%data	
	UR93	UT91	i		hantei_3	%data	
	UR94	UT91	i		hantei_4	%data	
	UR95	UT91	i		o_color	%data	
*							

図 7-21 User\_FRF\_traffic\_jam.csv

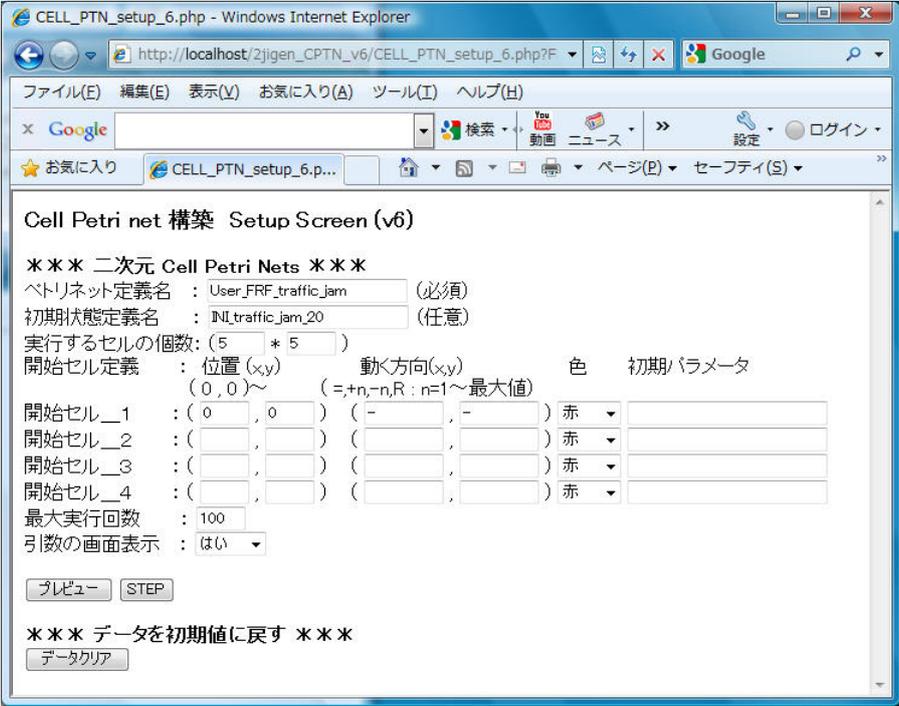


図 7-22 開始画面

実行画面は次のように表示される。

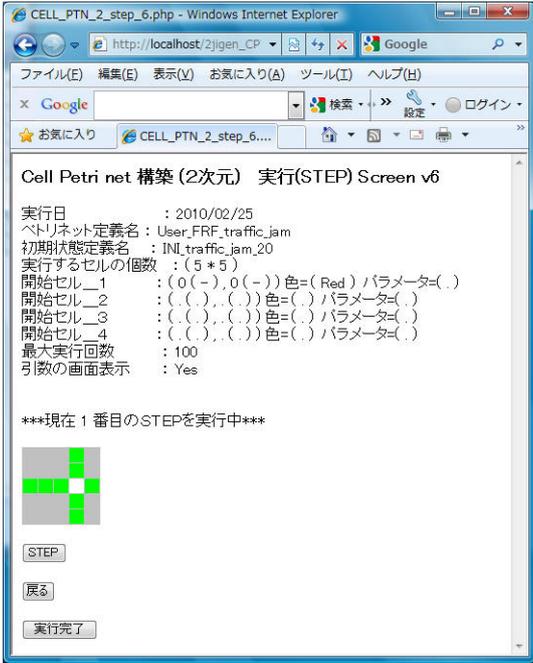
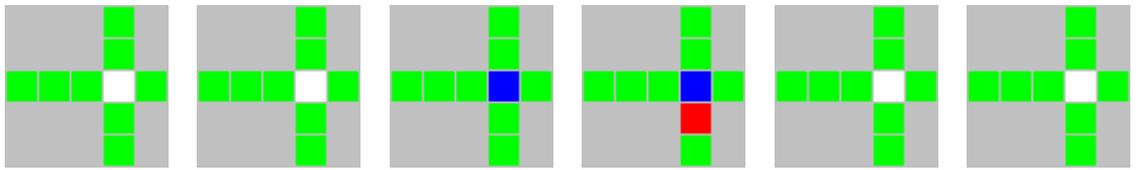


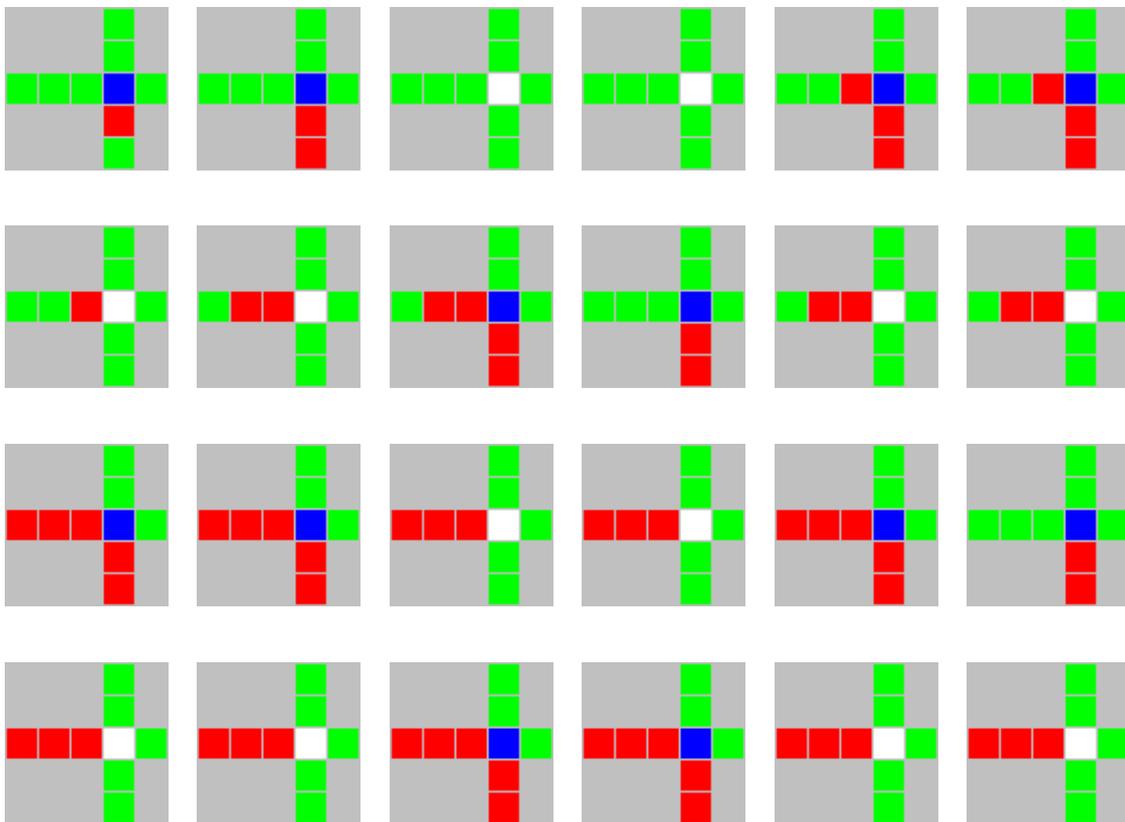
図 7-23 実行画面

ここで、セルペトリネットの実行結果の画像表示の部分だけをぬきだして Step1 から Step6 まで並べてみると、



step1	step2	step3	step4	step5	step6
信号は縦が青	縦が青	横が青	横が青	縦が青	縦が青
渋滞なし	渋滞なし	渋滞なし	縦で渋滞	渋滞なし	渋滞なし

のように、状況の推移をみる事ができる。以下、step30 まで実行を続けてみた。



初期状況定義ファイルで指定する信号の変わる時間、毎時進入する車の量、各区画で最大通過できる車の量、最大滞留できる車の量などの条件値を変えることにより、さまざまな条件でのシミュレーションが可能である。また、道路の長さ、本数、信号の場所や個数なども初期状況定義ファイルで変更可能である。

## 8 まとめと今後の課題

本研究では、Web 対応のセルペトリネットの開発ツールの構築を行った。カラーペトリネットの定義、別のセルへの制御の受け渡しなどを中心に設計および開発を行った。かなり自由にセルペトリネットを構築できており、プロトタイプとしての役割は果たせたと思う。

今後改良したい点は以下の通り。

- ◆ 現在の2次元での対応を3次元以上に拡げていければと思う。
- ◆ STEP ごとの実行ではなく、一度の実行でセルペトリネットを実行完了まで繰り返し行う機能を組み入れたい。(4-1 節)
- ◆ ペトリネット実行中の実行セルのすべての Place の Mark 情報を M S T (Mark Status Table) というプログラム内の配列として持っている。このため、一度に実行可能なセルが多くなる(たとえば全セルが実行対象となるような)場合、セルの個数によっては記憶容量を超えてしまう可能性がある。この点を改善できればと思う。(4-3 節)
- ◆ 近傍の範囲は開始画面で指定した最大個数( $n,m$ )により計算される  $n \times m$  個の連続したセル(すなわち  $(0,0)$  から  $(n-1,m-1)$  まで)の中で、数式等で規則付けできるもの、またはその全体である。“規則付けできない”部分集合を定義できるようにしたい。(5-1 節)
- ◆ 関数や開始画面で指定する計算式は、サンプル的に最低限のものを用意した。今後、さらに拡充していきたい。(4-6 節、5-1 節)

また、2009 年に ISO で国際標準データフォーマットに登録された PNML に対応していない。PNML フォーマットから今回開発したツールへの変換を行うツールが開発できれば、より広く活用されるであろう。

## 謝辞

本研究において、指導教員である信州大学大学院工学系研究科 Pauline Naomi Kawamoto 先生には、セルペトリネットについて、詳しくご教授いただきました。さらに、このツールの開発にあたっては、システムをデザインする段階から、セミナーを通して、いろいろ助言をいただきました。また、PHPでのイメージ出力の方法など、技術的なヒントもたくさんいただきました。

CAIでは、それぞれ担当の先生方から、さまざまご指導をいただきました。特に、CGI演習（応用）で学んだPHPの知識は、今回のツールの構築のベースとなりました。

修士学位審査にあたり、事務的なことでお世話になりました職員の皆様に感謝いたします。

## 参考文献

- [1] P. N. Kawamoto and Y. Nakamura. *On cell Petri nets*. Journal of Applied Functional Analysis, 1, 12-22, 1996.
- [2] 実川充. セルペトリネットの形式化. 修士論文資料(2007).  
<[http://sakura.cs.shinshu-u.ac.jp/pauline/work/SUGSI/papers/2007/thesis/jitsukawa\\_thesis.pdf](http://sakura.cs.shinshu-u.ac.jp/pauline/work/SUGSI/papers/2007/thesis/jitsukawa_thesis.pdf)>
- [3] *Petri Net Markup Language*(2006)  
< <http://www2.informatik.hu-berlin.de/top/pnml/about.html> >
- [4] J. Barros and L. Gomes. *Operational PNML: Towards a PNML support for model construction and modification*. In: Workshop on the Definition, Implementation and Application of a Standard Interchange Format for Petri Nets, Bologna, Italy, (2004)  
< <http://www2.uninova.pt/gres/opnml/BarrosGomesPNML2004.pdf>>
- [5] 大矢長門. セルペトリネットツール. 修士論文資料(2004).  
< [http://sakura.cs.shinshu-u.ac.jp/pauline/work/SUGSI/papers/2004/thesis/ooya\\_thesis.pdf](http://sakura.cs.shinshu-u.ac.jp/pauline/work/SUGSI/papers/2004/thesis/ooya_thesis.pdf) >.
- [6] 中村八束 『グラフ理論』.CAI教材  
<<http://cai.cs.shinshu-u.ac.jp/sugsi/Lecture/php/index.html>>
- [7] 太田淳 『ペトリネット入門』 (1995)  
<<http://www.aichi-pu.ac.jp/ist/~qua/introPN.pdf>>
- [8] 村田忠夫 『ペトリネットの解析と応用』 近代科学社(1992)
- [9] 新村正明 『CG I 演習(応用)』.CAI教材  
< <http://cai.cs.shinshu-u.ac.jp/sugsi/Lecture/php/index.html>>
- [10] *PHP 言語Reference*  
< <http://www.php.net/manual/ja/langref.php> >
- [11] 『OS I モデルと伝送制御』.Nagano Solution(ネットワーク技術者養成プログラム)教材  
< <http://www.wakasato.org/learn/nepc/course1/chapter01/index.html>>

## 付録の目次

I. セルペトリネット開始プログラム (CELL_PTN_setup_6.php) .....	53
II. セルペトリネット実行プログラム (CELL_PTN_2_step_6.php) .....	60
III. システムF R F (Firing_Rule_system_51.csv) .....	85
IV. A C F、M S Fの例 .....	86

## I. セルペトリネット開始プログラム (CELL\_PTIN\_setup\_6.php)

```
<HTML>
<HEAD>
  <TITLE>CELL_PTIN_setup_6.php</TITLE>
</HEAD>
<BODY>
<H3>Cell Petri net 構築 Setup Screen (v6)</H3>

<?php
  /**** CELL_START からのパラメーター引き渡し*****
  $frf_name=$_REQUEST['FRF_name'];
  $ini_name=$_REQUEST['INI_name'];
  $max_x=$_REQUEST['max_x'];
  $max_y=$_REQUEST['max_y'];
  $s_pos_1_x=$_REQUEST['start_id_1_x'];
  $s_pos_1_y=$_REQUEST['start_id_1_y'];
  $s_to_1_x=$_REQUEST['start_to_1_x'];
  $s_to_1_y=$_REQUEST['start_to_1_y'];
  $s_col_1=$_REQUEST['start_col_1'];
  $s_parm_1=$_REQUEST['start_parm_1'];
  $s_pos_2_x=$_REQUEST['start_id_2_x'];
  $s_pos_2_y=$_REQUEST['start_id_2_y'];
  $s_to_2_x=$_REQUEST['start_to_2_x'];
  $s_to_2_y=$_REQUEST['start_to_2_y'];
  $s_col_2=$_REQUEST['start_col_2'];
  $s_parm_2=$_REQUEST['start_parm_2'];
  $s_pos_3_x=$_REQUEST['start_id_3_x'];
  $s_pos_3_y=$_REQUEST['start_id_3_y'];
  $s_to_3_x=$_REQUEST['start_to_3_x'];
  $s_to_3_y=$_REQUEST['start_to_3_y'];
  $s_col_3=$_REQUEST['start_col_3'];
  $s_parm_3=$_REQUEST['start_parm_3'];
  $s_pos_4_x=$_REQUEST['start_id_4_x'];
  $s_pos_4_y=$_REQUEST['start_id_4_y'];
  $s_to_4_x=$_REQUEST['start_to_4_x'];
  $s_to_4_y=$_REQUEST['start_to_4_y'];
  $s_col_4=$_REQUEST['start_col_4'];
  $s_parm_4=$_REQUEST['start_parm_4'];
  $max_step=$_REQUEST['max_step'];
  $step_no=$_REQUEST['step_no'];
  $parm_disp=$_REQUEST['parm_disp'];
  $run_Reset=$_REQUEST['Reset'];

  if ( $run_Reset == "データクリア" ) {
    $frf_name = "";
    $ini_name = "";
    $max_x = "";
    $max_y = "";
    $s_pos_1_x = "";
    $s_pos_1_y = "";
    $s_to_1_x = "";
    $s_to_1_y = "";
    $s_col_1 = "";
    $s_parm_1 = "";
    $s_pos_2_x = "";
    $s_pos_2_y = "";
    $s_to_2_x = "";
    $s_to_2_y = "";
    $s_col_2 = "";
```

```

$s_parm_2 = "";
$s_pos_3_x = "";
$s_pos_3_y = "";
$s_to_3_x = "";
$s_to_3_y = "";
$s_col_3 = "";
$s_parm_3 = "";
$s_pos_4_x = "";
$s_pos_4_y = "";
$s_to_4_x = "";
$s_to_4_y = "";
$s_col_4 = "";
$s_parm_4 = "";
$max_step = "";
$parm_disp = "";
}

if ( $max_x == "" ) { $max_x=20; }
if ( $max_y == "" ) { $max_y=20; }
if ( $s_pos_1_x == "" ) { $s_pos_1_x=0; }
if ( $s_pos_1_y == "" ) { $s_pos_1_y=0; }
if ( $s_to_1_x == "" ) { $s_to_1_x="+"; }
if ( $s_to_1_y == "" ) { $s_to_1_y="+"; }
if ( $s_col_1 == "" ) { $s_col_1="Red"; }
if ( $s_col_2 == "" ) { $s_col_2="Red"; }
if ( $s_col_3 == "" ) { $s_col_3="Red"; }
if ( $s_col_4 == "" ) { $s_col_4="Red"; }
if ( $max_step == "" ) { $max_step=10; }
if ( $parm_disp == "" ) { $parm_disp="Yes"; }

if ( $ini_name == "なし" ) { $ini_name=""; }

?>

<FORM method="GET" action="CELL_PTN_2_step_6.php">
  <b> * * * 二次元 Cell Petri Nets * * * </b> <br>
  <?php
    if ( $frf_name == "" ) {
      print " ペトリネット定義名 : <input type="text" name="FRF_name" size=30> (必須)<br> ¥n";
    } else {
      print " ペトリネット定義名 : <input type="text" name="FRF_name" value=$frf_name size=30> (必須)
    }
  <br> ¥n";
    if ( $ini_name == "" ) {
      print " 初期状態定義名 : <input type="text" name="INI_name" size=30> (任意)<br> ¥n";
    } else {
      print " 初期状態定義名 : <input type="text" name="INI_name" value=$ini_name size=30> (任意)
    }
  <br> ¥n";
    print " 実行するセルの個数: (<input type="text" name="max_x" value=$max_x size=5> ¥n";
    print " * <input type="text" name="max_y" value=$max_y size=5> ) <br> ¥n";
    print " 開始セル定義 : 位置 (x,y) 動く方向(x,y) 色 初期パラメータ<br> ¥n";
    print " ( 0, 0 ) ~ ( =,+n,-n,R : n=1~最大値 ) <br> ¥n";
    print " 開始セル__1 : ( <input type="text" name="start_id_1_x" value=$s_pos_1_x size=5>
      , <input type="text" name="start_id_1_y" value=$s_pos_1_y size=5> )
      ( <input type="text" name="start_to_1_x" value=$s_to_1_x size=10>
      , <input type="text" name="start_to_1_y" value=$s_to_1_y size=10> ) ";
  switch ( $s_col_1 ) {
    case('Red'):
      print " <SELECT NAME='start_col_1'>
        <OPTION VALUE="Red" selected> 赤</OPTION>
        <OPTION VALUE="Blue" >青</OPTION>
        <OPTION VALUE="Green" >緑</OPTION>

```

```

                <OPTION VALUE=%"White%" >白</OPTION>
                <OPTION VALUE=%"Grey%" >灰色</OPTION>
            </SELECT> ";
        break ;
    case('Blue'):
        print " <SELECT NAME='start_col_1'>
                <OPTION VALUE=%"Red%" > 赤</OPTION>
                <OPTION VALUE=%"Blue%" selected>青</OPTION>
                <OPTION VALUE=%"Green%" >緑</OPTION>
                <OPTION VALUE=%"White%" >白</OPTION>
                <OPTION VALUE=%"Grey%" >灰色</OPTION>
            </SELECT> ";
        break ;
    case('Green'):
        print " <SELECT NAME='start_col_1'>
                <OPTION VALUE=%"Red%" > 赤</OPTION>
                <OPTION VALUE=%"Blue%" >青</OPTION>
                <OPTION VALUE=%"Green%" selected>緑</OPTION>
                <OPTION VALUE=%"White%" >白</OPTION>
                <OPTION VALUE=%"Grey%" >灰色</OPTION>
            </SELECT> ";
        break ;
    case('White'):
        print " <SELECT NAME='start_col_1'>
                <OPTION VALUE=%"Red%" > 赤</OPTION>
                <OPTION VALUE=%"Blue%" >青</OPTION>
                <OPTION VALUE=%"Green%" >緑</OPTION>
                <OPTION VALUE=%"White%" selected>白</OPTION>
                <OPTION VALUE=%"Grey%" >灰色</OPTION>
            </SELECT> ";
        break ;
    case('Grey'):
        print " <SELECT NAME='start_col_1'>
                <OPTION VALUE=%"Red%" > 赤</OPTION>
                <OPTION VALUE=%"Blue%" >青</OPTION>
                <OPTION VALUE=%"Green%" >緑</OPTION>
                <OPTION VALUE=%"White%" >白</OPTION>
                <OPTION VALUE=%"Grey%" selected>灰色</OPTION>
            </SELECT> ";
        break ;
    default:
        print " <SELECT NAME='start_col_1'>
                <OPTION VALUE=%"Red%" selected> 赤</OPTION>
                <OPTION VALUE=%"Blue%" >青</OPTION>
                <OPTION VALUE=%"Green%" >緑</OPTION>
                <OPTION VALUE=%"White%" >白</OPTION>
                <OPTION VALUE=%"Grey%" >灰色</OPTION>
            </SELECT> ";
        break ;
}
if ( $s_parm_1 == "" or $s_parm_1 == "." ) {
    print " <input type=%"text%" name=%"start_parm_1%" size=30> <br> ¥n " ;
} else {
    print " <input type=%"text%" name=%"start_parm_1%" value=$s_parm_1 size=30> <br> ¥n " ;
}

if ( $s_pos_2_x == "" or $s_pos_2_x == "." ) {
    print " 開始セル_2          : ( <input type=%"text%" name=%"start_id_2_x%" size=5>
        , <input type=%"text%" name=%"start_id_2_y%" size=5> )
        ( <input type=%"text%" name=%"start_to_2_x%" size=10>
        , <input type=%"text%" name=%"start_to_2_y%" size=10> ) " ;
} else {
    print " 開始セル_2          : ( <input type=%"text%" name=%"start_id_2_x%" value=$s_pos_2_x size=5>

```

```

        , <input type="text" name="start_id_2_y" value=$s_pos_2_y size=5>
        ( <input type="text" name="start_to_2_x" value=$s_to_2_x size=10>
        , <input type="text" name="start_to_2_y" value=$s_to_2_y size=10> ) " ;
    }
switch ($s_col_2) {
    case("):
    case('Red'):
        print " <SELECT NAME='start_col_2'>
                <OPTION VALUE="Red" selected> 赤</OPTION>
                <OPTION VALUE="Blue" >青</OPTION>
                <OPTION VALUE="Green" >緑</OPTION>
                <OPTION VALUE="White" >白</OPTION>
                <OPTION VALUE="Grey" >灰色</OPTION>
            </SELECT> " ;

        break ;
    case('Blue'):
        print " <SELECT NAME='start_col_2'>
                <OPTION VALUE="Red" > 赤</OPTION>
                <OPTION VALUE="Blue" selected>青</OPTION>
                <OPTION VALUE="Green" >緑</OPTION>
                <OPTION VALUE="White" >白</OPTION>
                <OPTION VALUE="Grey" >灰色</OPTION>
            </SELECT> " ;

        break ;
    case('Green'):
        print " <SELECT NAME='start_col_2'>
                <OPTION VALUE="Red" > 赤</OPTION>
                <OPTION VALUE="Blue" >青</OPTION>
                <OPTION VALUE="Green" selected>緑</OPTION>
                <OPTION VALUE="White" >白</OPTION>
                <OPTION VALUE="Grey" >灰色</OPTION>
            </SELECT> " ;

        break ;
    case('White'):
        print " <SELECT NAME='start_col_2'>
                <OPTION VALUE="Red" > 赤</OPTION>
                <OPTION VALUE="Blue" >青</OPTION>
                <OPTION VALUE="Green" >緑</OPTION>
                <OPTION VALUE="White" selected>白</OPTION>
                <OPTION VALUE="Grey" >灰色</OPTION>
            </SELECT> " ;

        break ;
    case('Grey'):
        print " <SELECT NAME='start_col_2'>
                <OPTION VALUE="Red" > 赤</OPTION>
                <OPTION VALUE="Blue" >青</OPTION>
                <OPTION VALUE="Green" >緑</OPTION>
                <OPTION VALUE="White" >白</OPTION>
                <OPTION VALUE="Grey" selected>灰色</OPTION>
            </SELECT> " ;

        break ;
    default:
        print " <SELECT NAME='start_col_2'>
                <OPTION VALUE="Red" selected> 赤</OPTION>
                <OPTION VALUE="Blue" >青</OPTION>
                <OPTION VALUE="Green" >緑</OPTION>
                <OPTION VALUE="White" >白</OPTION>
                <OPTION VALUE="Grey" >灰色</OPTION>
            </SELECT> " ;

        break ;
}
if ( $s_parm_2 == "" or $s_parm_2 == ".") {
    print " <input type="text" name="start_parm_2" size=30> <br> ¥n " ;
}

```

```

} else {
    print " <input type='text' name='start_parm_2' value=$s_parm_2 size=30> <br> ¥n ";
}

if ( $s_pos_3_x == "" or $s_pos_3_x == "." ) {
    print " 開始セル_3      : ( <input type='text' name='start_id_3_x' size=5>
      , <input type='text' name='start_id_3_y' size=5> )
      ( <input type='text' name='start_to_3_x' size=10>
      , <input type='text' name='start_to_3_y' size=10> ) ";
} else {
    print " 開始セル_3      : ( <input type='text' name='start_id_3_x' value=$s_pos_3_x size=5>
      , <input type='text' name='start_id_3_y' value=$s_pos_3_y size=5> )
      ( <input type='text' name='start_to_3_x' value=$s_to_3_x size=10>
      , <input type='text' name='start_to_3_y' value=$s_to_3_y size=10> ) ";
}

switch ( $s_col_3 ) {
    case("):
    case('Red'):
        print " <SELECT NAME='start_col_3'>
            <OPTION VALUE='Red' selected> 赤</OPTION>
            <OPTION VALUE='Blue'>青</OPTION>
            <OPTION VALUE='Green'>緑</OPTION>
            <OPTION VALUE='White'>白</OPTION>
            <OPTION VALUE='Grey'>灰色</OPTION>
        </SELECT> ";
        break ;
    case('Blue'):
        print " <SELECT NAME='start_col_3'>
            <OPTION VALUE='Red'> 赤</OPTION>
            <OPTION VALUE='Blue' selected>青</OPTION>
            <OPTION VALUE='Green'>緑</OPTION>
            <OPTION VALUE='White'>白</OPTION>
            <OPTION VALUE='Grey'>灰色</OPTION>
        </SELECT> ";
        break ;
    case('Green'):
        print " <SELECT NAME='start_col_3'>
            <OPTION VALUE='Red'> 赤</OPTION>
            <OPTION VALUE='Blue'>青</OPTION>
            <OPTION VALUE='Green' selected>緑</OPTION>
            <OPTION VALUE='White'>白</OPTION>
            <OPTION VALUE='Grey'>灰色</OPTION>
        </SELECT> ";
        break ;
    case('White'):
        print " <SELECT NAME='start_col_3'>
            <OPTION VALUE='Red'> 赤</OPTION>
            <OPTION VALUE='Blue'>青</OPTION>
            <OPTION VALUE='Green'>緑</OPTION>
            <OPTION VALUE='White' selected>白</OPTION>
            <OPTION VALUE='Grey'>灰色</OPTION>
        </SELECT> ";
        break ;
    case('Grey'):
        print " <SELECT NAME='start_col_3'>
            <OPTION VALUE='Red'> 赤</OPTION>
            <OPTION VALUE='Blue'>青</OPTION>
            <OPTION VALUE='Green'>緑</OPTION>
            <OPTION VALUE='White'>白</OPTION>
            <OPTION VALUE='Grey' selected>灰色</OPTION>
        </SELECT> ";
        break ;
    default:

```

```

print " <SELECT NAME='start_col_3'>
      <OPTION VALUE=%"Red%" selected> 赤</OPTION>
      <OPTION VALUE=%"Blue%">青</OPTION>
      <OPTION VALUE=%"Green%">緑</OPTION>
      <OPTION VALUE=%"White%">白</OPTION>
      <OPTION VALUE=%"Grey%">灰色</OPTION>
</SELECT> ";
break ;
}
if ( $s_parm_3 == "" or $s_parm_3 == ".") {
print " <input type=%"text%" name=%"start_parm_3%" size=30> <br> ¥n ";
} else {
print " <input type=%"text%" name=%"start_parm_3%" value=$s_parm_3 size=30> <br> ¥n ";
}
}

if ( $s_pos_4_x == "" or $s_pos_4_x == ".") {
print " 開始セル__4      : ( <input type=%"text%" name=%"start_id_4_x%" size=5>
      , <input type=%"text%" name=%"start_id_4_y%" size=5> )
      ( <input type=%"text%" name=%"start_to_4_x%" size=10>
      , <input type=%"text%" name=%"start_to_4_y%" size=10> ) ";
} else {
print " 開始セル__4      : ( <input type=%"text%" name=%"start_id_4_x%" value=$s_pos_4_x size=5>
      , <input type=%"text%" name=%"start_id_4_y%" value=$s_pos_4_y size=5> )
      ( <input type=%"text%" name=%"start_to_4_x%" value=$s_to_4_x size=10>
      , <input type=%"text%" name=%"start_to_4_y%" value=$s_to_4_y size=10> ) ";
}
}

switch ($s_col_4) {
case('Blue'):
print " <SELECT NAME='start_col_4'>
      <OPTION VALUE=%"Red%"> 赤</OPTION>
      <OPTION VALUE=%"Blue%" selected>青</OPTION>
      <OPTION VALUE=%"Green%">緑</OPTION>
      <OPTION VALUE=%"White%">白</OPTION>
      <OPTION VALUE=%"Grey%">灰色</OPTION>
</SELECT> ";
break ;
case('Green'):
print " <SELECT NAME='start_col_4'>
      <OPTION VALUE=%"Red%"> 赤</OPTION>
      <OPTION VALUE=%"Blue%">青</OPTION>
      <OPTION VALUE=%"Green%" selected>緑</OPTION>
      <OPTION VALUE=%"White%">白</OPTION>
      <OPTION VALUE=%"Grey%">灰色</OPTION>
</SELECT> ";
break ;
case('White'):
print " <SELECT NAME='start_col_4'>
      <OPTION VALUE=%"Red%"> 赤</OPTION>
      <OPTION VALUE=%"Blue%">青</OPTION>
      <OPTION VALUE=%"Green%">緑</OPTION>
      <OPTION VALUE=%"White%" selected>白</OPTION>
      <OPTION VALUE=%"Grey%">灰色</OPTION>
</SELECT> ";
break ;
case('Grey'):
print " <SELECT NAME='start_col_4'>
      <OPTION VALUE=%"Red%"> 赤</OPTION>
      <OPTION VALUE=%"Blue%">青</OPTION>
      <OPTION VALUE=%"Green%">緑</OPTION>
      <OPTION VALUE=%"White%">白</OPTION>
      <OPTION VALUE=%"Grey%" selected>灰色</OPTION>
</SELECT> ";
break ;
}
}

```

```

default:
    print " <SELECT NAME='start_col_4'>
            <OPTION VALUE=%"Red%" selected> 赤</OPTION>
            <OPTION VALUE=%"Blue%">青</OPTION>
            <OPTION VALUE=%"Green%">緑</OPTION>
            <OPTION VALUE=%"White%">白</OPTION>
            <OPTION VALUE=%"Grey%">灰色</OPTION>
        </SELECT> ";
    break ;
}
if ( $s_parm_4 == "" or $s_parm_4 == "." ) {
    print " <input type=%"text%" name=%"start_parm_4%" size=30> <br> ¥n " ;
} else {
    print " <input type=%"text%" name=%"start_parm_4%" value=$s_parm_4 size=30> <br> ¥n " ;
}

print " 最大実行回数      : <input type=%"text%" name=%"max_step%" value=$max_step size=5> <br> ¥n " ;
switch ($parm_disp) {
    case ('Yes'):
        print " 引数の画面表示      : <SELECT NAME='parm_disp'>
                <OPTION VALUE=%"Yes%" selected> はい</OPTION>
                <OPTION VALUE=%"No%">いいえ</OPTION>
            </SELECT> <br> ¥n " ;

        break ;
    case ('No'):
        print " 引数の画面表示      : <SELECT NAME='parm_disp'>
                <OPTION VALUE=%"Yes%"> はい</OPTION>
                <OPTION VALUE=%"No%" selected>いいえ</OPTION>
            </SELECT> <br> ¥n " ;

        break ;
    default :
}
}
?>
<input type="hidden" name="step_no" value= 0 size=5> <br>
<input type="submit" name="Preview" value="プレビュー">
<input type="submit" name="Step" value="STEP">
</FORM>

<FORM method="GET" action="CELL_PTN_setup_6.php">
    <b>*** データを初期値に戻す *** </b> <br>
<input type="submit" name = "Reset" value="データクリア">
</FORM>

</BODY>
</HTML>

```

## II. セルペトリネット実行プログラム (CELL\_PTIN\_2\_step\_6.php)

```
<HTML>
<HEAD>
  <TITLE>CELL_PTIN_2_step_6.php</TITLE>
</HEAD>
<BODY>
<H3>Cell Petri net 構築 (2次元) 実行(STEP) Screen v6 </H3>

<?php

//=====
//=====--start main routine =====
//=====

/** 今日の日付を得る*****
today_date() ;

/**MSF, ACFファイル用のディレクトリ指定*****
$dir = "work/";

/** 前の画面からのパラメータ取得 *****
get_parameter ();

/** 引き渡されたパラメータの表示*****
display_parameter ();

/** プレビュー/STEP 処理の選択 *****
if ( $shori_Preview == "プレビュー" ) { run_Preview(); }
if ( $shori_Step == "STEP" ) { run_Step(); }

/** HTML画面出力情報の組み立て *****
create_FORM();

/** end of main shori *****

//=====
// ===** function定義 **=====
//=====

/*******
/** 通常処理 (Cell Petri Netsの実行) *****
/*******
function run_Step () {

    global $today, $now_time, $dir, $active_display_cell ;
    global $ct_FRF, $ct_ACF ;
    global $FRF, $ACF, $next_ACF, $MST, $w_MST, $w_COL ;
    global $frf_name, $max_step, $step_no ;

/** firing rule file の読み込み*****
read_FRF( $frf_name );
if ( $ct_FRF > 0 ) { //** FRFが読み込まれた場合のみ以下を実行

/** 指定された回数実行を行う*****
if ( $step_no >= $max_step ) {
    print "***指定された最大実行回数に達しています。実行を完了してください。*** <br><br> ¥n ";
    /** 最後に表示したイメージファイルを表示する
    if ( $handle = opendir($dir) {
```

```

while (false != ($file = readdir($handle))) {
    $w_value = split('[/]', $file);
    $w_ext = $w_value[1];
    if ($w_ext == "png") {
        $img_file = $dir . $file;
        print "<IMG src=$img_file ><br> ¥n ";
    }
}
closedir($handle);
}
else {
    $step_no++;
    print "***現在 $step_no 番目のSTEPを実行中*** <br><br> ¥n ";

    /*** 最初の実行時にはMSF,ACFを初期状態にする ****
    if ( $step_no == 1 ) {
        init_MSF();
    }

    /***画面処理が起こるまで繰り返す ****
    $active_display_cell = 0;
    while( $active_display_cell == 0 ){

        /***配列の初期化 ****
        $MST = ""; //** 現在のマーク状況テーブル
        $w_MST = ""; //** MSTからの変更差分テーブル
        $ACF = ""; //** 今回実行するセルリスト
        $next_ACF = ""; //** 次回実行するセルリスト
        $next_ACF[0] = "*";

        /*** ACF(Active Cell File)の読み込み ****
        read_ACF();

        /*** 実行中のセルのMSF(Marking status file)を読み込み ****
        for ( $i= 1; $i<$ct_ACF; $i++) {
            read_MSF($ACF[$i]);
        }

        /***PTN実行****
        for ( $i= 1; $i<$ct_ACF; $i++) { //**実行中のセルだけ処理
            run_PTN( $ACF[$i] );
        }

        /*** あたらしいマーク情報をMSFに書く ****
        $ct_n_ACF = count($next_ACF);
        for ( $i = 1; $i < $ct_n_ACF; $i ++ ) {
            write_MSF($next_ACF[$i]);
        }

        /***今回マークが変化したセル番号を保管(次回実行の候補) ****
        write_ACF($next_ACF);

        /*** loop終了判定 ****
        if ( count($next_ACF) == 1 ) { //**次に実行するセルがなければ終了
            break;
        }

    } // end of while(active_display_cell = 0)

    /*** SC_DISPLAY処理 ****
    /***各セルの表示する色を決定する ****
    decide_color();

```

```

/**imagepngを用いて結果を描画する *****
    display_image ( );                               /** 描画 *****

/** 終了処理 *****
    if ( count($next_ACF) == 1 ) {
        print "*****今回の実行は $step_no 回で終了しました。***** <br> ¥n";
    }

    if ($step_no == $max_step ) {
        print "<br> ***指定された最大実行回数に達しました。実行を完了してください。*** <br> ¥n ";
    }
} /** end of if (step_no < max_step)
} /** end of if (ct_FRF > 0)

}; /**end of function run_Step

*****
/** 出力画面イメージの確認 *****
*****
function run_Preview () {

    global $INI, $ct_INI, $ini_name ;
    global $MST ;
    global $dir, $max_x, $max_y, $w_start ;

    /**WORKディレクトリー内のすべてのファイルを削除
    if ( $dh = opendir($dir) ) {
        while ( ( $file = readdir($dh) ) != false ) {
            $w_f_name = $file ;
            unlink ( $dir . $w_f_name ) ;
        }
    } else { /**WORKディレクトリーがなければ作成する
        mkdir ($dir) ;
    }

    /** 初期条件定義ファイル の読み込み *****
    read_INI( $ini_name ) ;

    /** INIに書かれたセルの色情報を取り組む *****
    $k = 0 ;
    for ( $i=0 ; $i < $ct_INI ; $i ++ ) {
        if ( $INI[$i]["comment"] != "*"
            and $INI[$i]["p_id"] == "to display" ) {
            $w_color[$k]["cell_id"] = $INI[$i]["cell_id"] ;
            $w_color[$k]["color"] = $INI[$i]["p_mark"] ;
            $k++ ;
        }
    }

    /** 開始セルの色情報を取り組む *****
    for ( $i = 1 ; $i < count($w_start)+1 ; $i++ ) {
        if ( $w_start[$i]["pos"] <> "." ) {
            $w_color[$k]["cell_id"] = $w_start[$i]["pos"] ;
            $w_color[$k]["color"] = $w_start[$i]["col"] ;
            $k++ ;
        }
    }

    /** MSTを初期化する
    $MST = "" ;

    /** SC_DISPLAYのマーク状況をMSTに書き込む
    /** セルID情報を書く

```

```

$w_cell_id = "SC_DISPLAY" ;
$w_pid = "cell_id" ;
$w_pmk = $w_cell_id ;
$MST[$w_cell_id][$w_pid][$w_pmk] = 1 ;

/** defaultの色(灰色)情報をすべてのセルに書く
for ( $i = 0 ; $i < $max_x; $i ++ ) {
    for ( $j = 0 ; $j < $max_y; $j ++ ) {
        $w_pid = "UC_" . sprintf("%05d", $i) . "_" . sprintf("%05d", $j) ;
        $w_pmk = "Grey" ;
        $MST[$w_cell_id][$w_pid][$w_pmk] = 1 ;
    }
}

/** INI情報および開始セル情報から取り出したセルの色に置き換える
for ( $i = 0 ; $i < $k ; $i++ ) {
    $w_pid = $w_color[$i]["cell_id"] ;
    $w_pmk = $w_color[$i]["color"] ;
    $MST[$w_cell_id][$w_pid][$w_pmk] = 1 ;
}

/** SC_DISPLAYのマーク情報をMSFに書く
write_MSF( $w_cell_id ) ;

/** 各セルの表示する色を決定する *****
decide_color ( ) ;

/** imagepngを用いて結果を描画する ***** //*** 描画 *****
display_image ( ) ;

} //*** end of function run_Preview

/** *****
/** 今日の日付を得る(m added) *****
/** *****
function today_date () {
    global $today, $now_time ;
    $now = getdate(); // 現在時刻の情報を持った配列を作成
    $month = sprintf("%02d", $now['mon']) ; // 今月
    $day = sprintf("%02d", $now['mday']) ; // 今日
    $year = $now['year']; // 今年
    $today = $year . "/" . $month . "/" . $day ;

    $hour = sprintf("%02d", $now['hours']) ; // 時
    $minute = sprintf("%02d", $now['minutes']) ; // 分
    $second = sprintf("%02d", $now['seconds']) ; // 秒
    $now_time = $hour . $minute . $second ;
};

/** *****
/** 前の画面からのパラメーター取得 *****
/** *****
function get_parameter () {
    global $REQUEST ;
    global $frf_name, $ini_name, $max_x, $max_y, $max_step, $step_no, $parm_disp ;
    global $s_pos_1x, $s_pos_1y, $s_to_1x, $s_to_1y, $s_col_1, $s_parm_1 ;
    global $s_pos_2x, $s_pos_2y, $s_to_2x, $s_to_2y, $s_col_2, $s_parm_2 ;
    global $s_pos_3x, $s_pos_3y, $s_to_3x, $s_to_3y, $s_col_3, $s_parm_3 ;
    global $s_pos_4x, $s_pos_4y, $s_to_4x, $s_to_4y, $s_col_4, $s_parm_4 ;
    global $shori_Step, $shori_Preview ;
    global $w_start ;

```

```

$frf_name=$_REQUEST['FRF_name'];
$ini_name=$_REQUEST['INI_name'];
$max_x=$_REQUEST['max_x'];
$max_y=$_REQUEST['max_y'];
$s_pos_1x=$_REQUEST['start_id_1_x'];
$s_pos_1y=$_REQUEST['start_id_1_y'];
$s_to_1x=$_REQUEST['start_to_1_x'];
$s_to_1y=$_REQUEST['start_to_1_y'];
$s_col_1=$_REQUEST['start_col_1'];
$s_parm_1=$_REQUEST['start_parm_1'];
$s_pos_2x=$_REQUEST['start_id_2_x'];
$s_pos_2y=$_REQUEST['start_id_2_y'];
$s_to_2x=$_REQUEST['start_to_2_x'];
$s_to_2y=$_REQUEST['start_to_2_y'];
$s_col_2=$_REQUEST['start_col_2'];
$s_parm_2=$_REQUEST['start_parm_2'];
$s_pos_3x=$_REQUEST['start_id_3_x'];
$s_pos_3y=$_REQUEST['start_id_3_y'];
$s_to_3x=$_REQUEST['start_to_3_x'];
$s_to_3y=$_REQUEST['start_to_3_y'];
$s_col_3=$_REQUEST['start_col_3'];
$s_parm_3=$_REQUEST['start_parm_3'];
$s_pos_4x=$_REQUEST['start_id_4_x'];
$s_pos_4y=$_REQUEST['start_id_4_y'];
$s_to_4x=$_REQUEST['start_to_4_x'];
$s_to_4y=$_REQUEST['start_to_4_y'];
$s_col_4=$_REQUEST['start_col_4'];
$s_parm_4=$_REQUEST['start_parm_4'];
$max_step=$_REQUEST['max_step'];
$step_no=$_REQUEST['step_no'];
$parm_disp=$_REQUEST['parm_disp'];
$shori_Step=$_REQUEST['Step'];
$shori_Preview=$_REQUEST['Preview'];

```

///**入力データの補完**\*\*\*\*\*

```

if ( $ini_name == "" ) { $ini_name = "なし" ; }
if ( $s_pos_1x == "" ) { $s_pos_1x = 0 ; }
if ( $s_pos_1y == "" ) { $s_pos_1y = 0 ; }
if ( $s_to_1x == "" ) { $s_to_1x = "+1" ; }
if ( $s_to_1y == "" ) { $s_to_1y = "+1" ; }
if ( $s_to_1x == "r" ) { $s_to_1x = "R" ; }
if ( $s_to_1y == "r" ) { $s_to_1y = "R" ; }
if ( $s_parm_1 == "" ) { $s_parm_1 = "." ; }
$w_start[1]["pos"] = "UC_" . sprintf('%05d', $s_pos_1x) . "_" . sprintf('%05d', $s_pos_1y) ;
$w_start[1]["to"] = $s_to_1x . "_" . $s_to_1y ;
$w_start[1]["col"]=$s_col_1 ;
$w_start[1]["parm"]=$s_parm_1 ;

if ( $s_parm_2 == "" ) { $s_parm_2 = "." ; }
if ( $s_pos_2x == "" ) {
    $s_pos_2x = "." ;
    $s_to_2x = "." ;
    $s_pos_2y = "." ;
    $s_to_2y = "." ;
    $s_col_2 = "." ;
    $w_start[2]["pos"] = "." ;
} else {
    if ( $s_to_2x == "" ) { $s_to_2x = "+1" ; }
    if ( $s_to_2y == "" ) { $s_to_2y = "+1" ; }
    if ( $s_to_2x == "r" ) { $s_to_2x = "R" ; }
    if ( $s_to_2y == "r" ) { $s_to_2y = "R" ; }
    $w_start[2]["pos"] = "UC_" . sprintf('%05d', $s_pos_2x) . "_" . sprintf('%05d', $s_pos_2y) ;
}

```

```

    $w_start[2]["to"] = $s_to_2x . "_" . $s_to_2y ;
    $w_start[2]["col"]=$s_col_2 ;
    $w_start[2]["parm"]=$s_parm_2 ;
}

if ( $s_parm_3 == "" ) { $s_parm_3 = "." ; }
if ( $s_pos_3x == "" ) {
    $s_pos_3x = "." ;
    $s_to_3x = "." ;
    $s_pos_3y = "." ;
    $s_to_3y = "." ;
    $s_col_3 = "." ;
    $w_start[3]["pos"] = "." ;
} else {
    if ( $s_to_3x == "" ) { $s_to_3x = "+1" ; }
    if ( $s_to_3y == "" ) { $s_to_3y = "+1" ; }
    if ( $s_to_3x == "r" ) { $s_to_3x = "R" ; }
    if ( $s_to_3y == "r" ) { $s_to_3y = "R" ; }
    $w_start[3]["pos"] = "UC_" . sprintf('%05d', $s_pos_3x) . "_" . sprintf('%05d', $s_pos_3y) ;
    $w_start[3]["to"] = $s_to_3x . "_" . $s_to_3y ;
    $w_start[3]["col"]=$s_col_3 ;
    $w_start[3]["parm"]=$s_parm_3 ;
}

if ( $s_parm_4 == "" ) { $s_parm_4 = "." ; }
if ( $s_pos_4x == "" ) {
    $s_pos_4x = "." ;
    $s_to_4x = "." ;
    $s_pos_4y = "." ;
    $s_to_4y = "." ;
    $s_col_4 = "." ;
    $w_start[4]["pos"] = "." ;
} else {
    if ( $s_to_4x == "" ) { $s_to_4x = "+1" ; }
    if ( $s_to_4y == "" ) { $s_to_4y = "+1" ; }
    if ( $s_to_4x == "r" ) { $s_to_4x = "R" ; }
    if ( $s_to_4y == "r" ) { $s_to_4y = "R" ; }
    $w_start[4]["pos"] = "UC_" . sprintf('%05d', $s_pos_4x) . "_" . sprintf('%05d', $s_pos_4y) ;
    $w_start[4]["to"] = $s_to_4x . "_" . $s_to_4y ;
    $w_start[4]["col"]=$s_col_4 ;
    $w_start[4]["parm"]=$s_parm_4 ;
}

} //*** end of function get_data

//*****
//*** 引き渡されたパラメータの表示 *****
//*****
function display_parameter () {
    global $today, $now_time ;
    global $frf_name, $ini_name, $max_x, $max_y, $max_step, $step_no, $parm_disp ;
    global $s_pos_1x, $s_pos_1y, $s_to_1x, $s_to_1y, $s_col_1, $s_parm_1 ;
    global $s_pos_2x, $s_pos_2y, $s_to_2x, $s_to_2y, $s_col_2, $s_parm_2 ;
    global $s_pos_3x, $s_pos_3y, $s_to_3x, $s_to_3y, $s_col_3, $s_parm_3 ;
    global $s_pos_4x, $s_pos_4y, $s_to_4x, $s_to_4y, $s_col_4, $s_parm_4 ;

//*** 引き渡されたパラメータの表示*****
if ($parm_disp == "Yes" ) {
    print "実行日          : $today <br> ¥n" ;
    print "ペトリネット定義名      : $frf_name <br> ¥n" ;
    print "初期状態定義名          : $ini_name <br> ¥n" ;
    print "実行するセルの個数      : ( $max_x * $max_y ) <br> ¥n" ;
    print "開始セル_1              : ( $s_pos_1x ( $s_to_1x ) , $s_pos_1y ( $s_to_1y ) ) 色=( $s_col_1 ) パラメータ

```

```

=( $s_parm_1 )<br> ¥n" ;
    print "開始セル__2          : ( $s_pos_2x ( $s_to_2x ) , $s_pos_2y ( $s_to_2y ) ) 色=( $s_col_2 ) パラメータ
=( $s_parm_2 )<br> ¥n" ;
    print "開始セル__3          : ( $s_pos_3x ( $s_to_3x ) , $s_pos_3y ( $s_to_3y ) ) 色=( $s_col_3 ) パラメータ
=( $s_parm_3 )<br> ¥n" ;
    print "開始セル__4          : ( $s_pos_4x ( $s_to_4x ) , $s_pos_4y ( $s_to_4y ) ) 色=( $s_col_4 ) パラメータ
=( $s_parm_4 )<br> ¥n" ;
    print "最大実行回数          : $max_step <br> ¥n" ;
    print "引数の画面表示        : $parm_disp <br> ¥n" ;
    print " <br> <br> ¥n" ;
}

} //*** end of function display_parameter

//*****
//*** HTML画面出力情報の組み立て *****
//*****
function create_FORM () {
    global $today, $now_time ;
    global $frf_name, $ini_name, $max_x, $max_y, $max_step, $step_no, $parm_disp ;
    global $s_pos_1x, $s_pos_1y, $s_to_1x, $s_to_1y, $s_col_1, $s_parm_1 ;
    global $s_pos_2x, $s_pos_2y, $s_to_2x, $s_to_2y, $s_col_2, $s_parm_2 ;
    global $s_pos_3x, $s_pos_3y, $s_to_3x, $s_to_3y, $s_col_3, $s_parm_3 ;
    global $s_pos_4x, $s_pos_4y, $s_to_4x, $s_to_4y, $s_col_4, $s_parm_4 ;

    print " <FORM method=¥"GET¥" action=¥"CELL_PTN_2_step_6.php¥"> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"FRF_name¥" value=$frf_name size=30> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"INI_name¥" value=$ini_name size=30> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"max_x¥" value=$max_x size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"max_y¥" value=$max_y size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_id_1_x¥" value=$s_pos_1x size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_id_1_y¥" value=$s_pos_1y size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_to_1_x¥" value=$s_to_1x size=10> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_to_1_y¥" value=$s_to_1y size=10> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_col_1¥" value=$s_col_1 size=6> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_parm_1¥" value=$s_parm_1 size=30> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_id_2_x¥" value=$s_pos_2x size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_id_2_y¥" value=$s_pos_2y size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_to_2_x¥" value=$s_to_2x size=10> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_to_2_y¥" value=$s_to_2y size=10> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_col_2¥" value=$s_col_2 size=6> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_parm_2¥" value=$s_parm_2 size=30> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_id_3_x¥" value=$s_pos_3x size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_id_3_y¥" value=$s_pos_3y size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_to_3_x¥" value=$s_to_3x size=10> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_to_3_y¥" value=$s_to_3y size=10> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_col_3¥" value=$s_col_3 size=6> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_parm_3¥" value=$s_parm_3 size=30> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_id_4_x¥" value=$s_pos_4x size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_id_4_y¥" value=$s_pos_4y size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_to_4_x¥" value=$s_to_4x size=10> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_to_4_y¥" value=$s_to_4y size=10> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_col_4¥" value=$s_col_4 size=6> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"start_parm_4¥" value=$s_parm_4 size=30> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"max_step¥" value=$max_step size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"step_no¥" value=$step_no size=5> ¥n" ;
    print " <input type=¥"hidden¥" name=¥"parm_disp¥" value=$parm_disp size=5> ¥n" ;

    print " <input type=¥"submit¥" name=¥"Step¥" value=¥"STEP¥"> ¥n" ;
    print " </FORM> ¥n" ;
} //*** end of function create_FORM

```

```

/*****
/**** INI(initial condition file)の読み込み ****
/*****
function read_INI ( $ini_name ) {

    global $INI, $ct_INI ;
    global $max_x, $max_y ;
    $INI = "" ;
    $count = 0 ;

/**** 初期条件定義ファイルが指定されていたら、このルーチンを実行する ****//
if ( $ini_name != "なし" ) {

    /**** 初期条件定義ファイルを読み込む
    $f_name = $ini_name . ".csv" ;
    $fp = fopen($f_name,"r") ;

    if ( $fp ) {
        while(!(feof($fp))) {
            $buf=fgets($fp,100);
            $buf=chop($buf);
            $data=explode(",",$buf) ;
            $w_comment = $data[0] ;
            $w_cell_id_x = $data[1] ;
            $w_cell_id_y = $data[2] ;
            $w_p_id = $data[3] ;
            $w_pmk= $data[4] ;
            /**** コメント行以外でセルが範囲以内のときは初期条件として有効 ****//
            if ( substr($w_comment,0,1) != "*" and $w_p_id != "" ) {
                if ( 0 <= $w_cell_id_x and $w_cell_id_x < $max_x and 0 <= $w_cell_id_y and $w_cell_id_y < $max_y ) {
                    $INI[$count][ "cell_id" ] = "UC_" . sprintf("%05d", $w_cell_id_x) . "_" . sprintf("%05d", $w_cell_id_y) ;
                    $INI[$count][ "p_id" ] = $w_p_id ;
                    $INI[$count][ "p_mark" ] = $w_pmk ;
                    $count++ ;
                }
            }
        }
        fclose($fp) ;
        $ct_INI = $count - 1 ;
    } else { // ファイルがない場合
        $ct_INI = 0 ;
        print " ****ERR**** 初期条件定義ファイルが存在しません。再入力してください <br> ¥n " ;
    }
}

}; // end of function read_INI

```

```

/*****
/**** FRF(firing rule file)の読み込み ****
/*****
function read_FRF ( $frf_name ) {

```

```

    global $FRF, $ct_FRF ;
    $FRF = "" ;
    $count = 0 ;

```

```

/****User firing rule fileを読み込む
    $f_name = $frf_name . ".csv" ;
    $fp = fopen($f_name,"r") ;

```

```

    if ( $fp ) {
        while(!(feof($fp))) {
            $buf=fgets($fp,100);

```

```

        $buf=chop($buf);
        $data=explode("",$buf);
        $FRF[$count]["comment"] = $data[0];
        $FRF[$count]["rule_id"] = $data[1];
        $FRF[$count]["t_id"] = $data[2];
        $FRF[$count]["p_type"] = $data[3];
        $FRF[$count]["o_cell_towards"] = $data[4];
        $FRF[$count]["p_id"] = $data[5];
        $FRF[$count]["p_mark"] = $data[6];
        $count++;
    };
    fclose($fp);
    $ct_FRF = $count - 1;
} else { // ファイルがない場合
    $ct_FRF = 0;
    print " ***ERR*** 発火条件ファイルが存在しません。再入力してください <br> ¥n ";
}

/**system firing rule file を読み込む
$f_name = "Firing_Rule_system_51.csv"; //**2次元cell petri net (v5)用のシステムFRF
$fp = fopen($f_name,"r");

if ( $fp ) {
    while(!(feof($fp))) {
        $buf=fgets($fp,100);
        $buf=chop($buf);
        $data=explode("",$buf);
        $FRF[$count]["comment"] = $data[0];
        $FRF[$count]["rule_id"] = $data[1];
        $FRF[$count]["t_id"] = $data[2];
        $FRF[$count]["p_type"] = $data[3];
        $FRF[$count]["o_cell_towards"] = $data[4];
        $FRF[$count]["p_id"] = $data[5];
        $FRF[$count]["p_mark"] = $data[6];
        $count++;
    };
    fclose($fp);
    $ct_FRF = $count - 1;
} else { //** ファイルがない場合
    $ct_FRF = 0;
    print " ***ERR*** 発火条件ファイル(system2)が存在しません。確認してください。 <br> ¥n ";
}

}; // end of function read_FRF

/*******
/*** 最初の実行時にはMSF,ACFを初期状態にする*****
/*******
function init_MSF () {

    global $max_x, $max_y, $dir;
    global $w_start;
    global $ini_name, $INI, $ct_INI;
    global $MST;

    /***WORKディレクトリー内のすべてのファイルを削除
    if ( $dh = opendir($dir) ) {
        while ( ( $file = readdir($dh) ) != false ) {
            $w_f_name = $file;
            unlink ( $dir . $w_f_name );
        }
    } else { //**WORKディレクトリーがなければ作成する
        mkdir ( $dir );

```

```

}

//*** MST初期化
$MST = "";

//*** 最初に実行するセルのリストの初期化 *****
$first_ACF = ""; //*** 最初に実行するセルのリスト
$first_ACF[0] = "*"; //***first_ACFの初期値設定

//***システムセルのマークを初期状態にする (MSF)
$w_cell_id = "SC_DISPLAY";
$w_pid = "cell_id";
$w_pmk = $w_cell_id;
$MST[$w_cell_id][$w_pid][$w_pmk] = 1;
for ( $i = 0 ; $i < $max_x; $i ++ ) {
    for ( $j = 0 ; $j < $max_y; $j ++ ) {
        $w_pid = "UC_" . sprintf("%05d", $i) . "_" . sprintf("%05d", $j);
        $w_pmk = "Grey";
        $MST[$w_cell_id][$w_pid][$w_pmk] = 1;
    }
}

//***全セルのマークを初期状態にする (MSF)
for ( $i = 0 ; $i < $max_x ; $i++ ) {
    for ( $j = 0 ; $j < $max_y ; $j++ ) {
        //*** セルID placeをおく
        $w_cell_id = "UC_" . sprintf("%05d", $i) . "_" . sprintf("%05d", $j);
        $w_pid = "cell_id";
        $w_pmk = $w_cell_id;
        $MST[$w_cell_id][$w_pid][$w_pmk] = 1;
    }
}

//*** 初期情報定義ファイルを読む
read_INI($ini_name);

//*** 初期情報データの取り込み
if ( $ct_INI > 0 ) {
    for ( $i = 0 ; $i < $ct_INI+1 ; $i++ ) {
        $w_cell_id = $INI[$i]["cell_id"];
        $w_pid = $INI[$i]["p_id"];
        $w_pmk = $INI[$i]["p_mark"];
        if ( $w_pid == "to_display" ) {
            $w_pid = $w_cell_id;
            $w_cell_id = "SC_DISPLAY";
        }
        $MST[$w_cell_id][$w_pid][$w_pmk] = 1;
        if ( $w_pid == "to_next" ) {
            $hit = array_search($w_cell_id, $first_ACF, true); //***出力セルがあたらしいセルかどうかを確認する
            if ( $hit == FALSE ) { //*** first_ACFに存在しない場合
                $first_ACF[] = $w_cell_id; //*** first_ACFに追加する
            }
        }
    }
} //*** end of if ( ct_INI > 0 )

//*** 開始セルならreceive placeをおく
for ( $k = 0 ; $k < 4 ; $k++ ) {
    $w_cell_id = $w_start[$k+1]["pos"];
    if ( substr($w_cell_id,0,2) == "UC" ) {
        $w_pid = "receive";
        $w_pmk = $w_start[$k+1]["to"] . ":" . $w_start[$k+1]["col"] . ":" . $w_start[$k+1]["parm"] ;
        $MST[$w_cell_id][$w_pid][$w_pmk] = 1;
    }
}

```

```

        $hit = array_search($w_cell_id, $first_ACF, true);    //***出力セルがあたらしいセルかどうかを確認する
        if ( $hit == FALSE ) {                               //*** first_ACFに存在しない場合
            $first_ACF[] = $w_cell_id;                       //*** first_ACFに追加する
        }
    }
}

//*** MSF書き込み

write_MSF("SC_DISPLAY");

for ($i = 0 ; $i < $max_x ; $i++ ) {
    for ( $j = 0 ; $j < $max_y ; $j++ ) {
        $w_cell_id = "UC_" . sprintf("%05d", $i) . "_" . sprintf("%05d", $j);
        write_MSF($w_cell_id);
    }
}

//***最初に実行するセル情報の書き込み(ACF)
write_ACF ( $first_ACF );

}; // end of function init_MSF

//*****
//*** ACF(Active Cell File)の読み込み *****
//*****
function read_ACF () {

    global $dir ;
    global $ACF, $ct_ACF ;

    //*** 実行するセルのリストの初期化 *****
    $ACF = "" ;
    $ACF[0] = "*" ;

    $acf_name = "ACF" ;
    $f_name = $dir . $acf_name . ".csv" ;

    $count = 1 ;
    $fp = fopen($f_name, "r") ;

    if ( $fp ) {
        while(!(feof($fp))) {
            $buf=fgets($fp,100);
            $buf=chop($buf);
            $ACF[$count] = $buf ;
            $count++ ;
        };
        fclose($fp) ;
        $ct_ACF = $count - 1 ;
    } else { // ファイルがない場合
        $ct_ACF = 0 ;
        print " ***ERR*** ACFファイルが存在しませんでした。確認して下さい。<br> ¥n " ;
    }
}

}; // end of function read_ACF

//*****
//*** ACF(Active Cell File)の書き込み *****
//*****
function write_ACF ($w_ACF) {

    global $dir ;

```

```

$acf_name = "ACF" ;
$f_name = $dir . $acf_name . ".csv" ;

$ct_ACF = count($w_ACF) ;

$fp = fopen($f_name,"w") ;
if ( flock($fp,LOCK_EX) ) { //*** file lock *****
    for ($i = 1 ; $i < $ct_ACF ; $i++) {
        if ( $w_ACF[$i] != "SC_DISPLAY" ) {
            $o_data = $w_ACF[$i] . "¥n" ;
            if ( fputs($fp,$o_data) == FALSE ) {
                print "***ERR**$count* write_ACFで書き込みに失敗しました <br> ¥n " ;
            }
        }
    }
}
flock($fp,LOCK_UN) ; //*** file unlock *****
} else {
    print "***ERR*** write_ACFでファイルロックに失敗しました <br> ¥n " ;
}
fclose ($fp) ;

```

```
}; // end of function write_ACF
```

```

//*****
//*** MSF(marking status file)の読み込み *****
//*****
function read_MSF ($cell_id) {

```

```

    global $dir ;
    global $MST ;

    $msf_name = "MSF_" . $cell_id ;
    $f_name = $dir . $msf_name . ".csv" ;

    $count = 0 ;
    $fp = fopen($f_name,"r") ;
    if ($fp) {
        while(!(feof($fp))) {
            $buf=fgets($fp,100);
            $buf=chop($buf);
            $data=explode("",$buf) ;
            $p_id = $data[0] ;
            $p_mark = $data[1] ;
            if ( $p_id != "" ) {
                $MST[$cell_id][$p_id][$p_mark] = 1 ;
            }
            $count++ ;
        }
        fclose($fp) ;
        $ct_MSF = $count - 1;
    }

```

```
}; // end of function read_MSF
```

```

//*****
//*** MSF(Marking Status File)の書き込み *****
//*****
function write_MSF ($cell_id) {

```

```

    global $dir, $count ;
    global $MST ;

```

```

$msf_name = "MSF_" . $cell_id ;
$f_name = $dir . $msf_name . ".csv" ;

/** 出力データを抽出 *****/
$w_ct = 0 ;
$ct_pid = count($MST[$cell_id]) ;
$w_k_pid = array_keys ($MST[$cell_id]) ;
for ($i = 0 ; $i < $ct_pid + 1 ; $i ++ ) {
    $ct_pmk = count($MST[$cell_id][$w_k_pid[$i]]) ;
    $w_k_pmk = array_keys ($MST[$cell_id][$w_k_pid[$i]]) ;
    for ($k = 0 ; $k < $ct_pmk + 1 ; $k ++ ) {
        if ( $MST[$cell_id][$w_k_pid[$i]][$w_k_pmk[$k]] == 1 ) {
            $o_data[$w_ct] = $w_k_pid[$i] . "," . $w_k_pmk[$k] . "¥n" ;
            $w_ct ++ ;
        }
    } // end of loop (k)
} // end of loop (i)

/** 変更情報があるときのみMSFファイルに出力する *****/
if ( $w_ct > 0 ) {
    $fp = fopen($f_name,"w") ;
    if ( flock($fp,LOCK_EX) ) { //*** file lock *****/
        for ($i = 0 ; $i < $w_ct ; $i++) {
            if ( fputs($fp, $o_data[$i]) == FALSE ) {
                print "***ERR**$i* write_MSFで書き込みに失敗しました <br> ¥n " ;
            }
        }
        flock($fp,LOCK_UN) ; //*** file unlock *****/
    } else {
        print "***ERR*** write_MSFでファイルロックに失敗しました <br> ¥n " ;
    }
    fclose ($fp) ;
} // end of if $w_ct > 0

}; // end of function write_MSF

/** *****
/** 指定されたセルで Petri Netを実行 *****
/** *****
function run_PTN( $cell_id ) {

    global $dir, $max_x, $max_y, $active_display_cell, $count ;
    global $FRF, $next_ACF, $MST, $w_MST ;
    global $ct_FRF ;
    global $w_parm ;

    $w_rule_id = "" ;
    $w_tid = "" ;

    for ($count=0; $count < $ct_FRF; $count++) {
        for ($z=0; $z<1; $z++) {
            if ( $FRF[$count]["comment"] == "*" ) { // コメント行は読み飛ばす
                break ;
            }
        }

    /** 発火条件の区切りをさがす**
        if ( $w_rule_id == $FRF[$count]["rule_id"] ) { //***同一の発火条件
            if ( $FRF[$count]["p_type"] == "o" //***出力定義になった または
                or $w_fire_flag == 0 ) { //***前の条件が発火しなかったら
                $w_check_end = 1 ; //***同一発火条件終了**
            }
        } else { //***新しい発火条件
            $w_parm = "" ; //*** 同一条件内で有効な変数配列を初期化

```

```

if ($w_fire_flag == 1) { //***前の条件が発火していたら
    chg_MST( $cell_id, $w_rule_id ); //****MST更新***
}
$w_check_end = 0;
$w_fire_flag = 0;
$w_MST = "" ;
}

$w_rule_id = $FRF[$count]["rule_id"]; //***FRFデータをワークに移す
$w_t_id = $FRF[$count]["t_id"];
$w_p_type = $FRF[$count]["p_type"];
$w_o_towards = $FRF[$count]["o_cell_towards"] ;
$w_p_id = $FRF[$count]["p_id"] ;
$w_p_mark = $FRF[$count]["p_mark"] ;

//*** cell_id変数のセット *****
if ( $w_p_id == "%cell_id" ) { $w_p_id = $cell_id ; }
if ( $w_p_mark == "%cell_id" ) { $w_p_mark = $cell_id ; }

//*** $w_p_markの変数処理 *****
if ( substr($w_p_mark,0,1) == "%" ) {
    $w_p_mark = hensu_henkan( $cell_id, $w_p_type, $w_p_id, $w_p_mark ) ;
}

//*** $w_o_towardsの変数処理 *****
if ( substr($w_o_towards, 0, 1) == "%"
and substr($w_o_towards,0, 2) != "%="
and $w_p_type == "o" ) {
    $w_value = $w_parm[$w_o_towards] ;
    $w_o_towards = $w_value ;
}

//*** $w_p_idの変数処理 *****
if ( substr($w_p_id, 0, 1) == "%"
and substr($w_p_id,0, 2) != "%=" ) {
    if ( $w_p_type == "o" ) {
        $w_value = $w_parm[$w_p_id] ;
        $w_p_id = $w_value ;
    } else {
        $w_parm[$w_p_id] = $w_p_mark ;
    }
}

//***発火条件を調べる****
if ( $w_p_type == "i" and $w_check_end != 1 ) {
    if ( $MST[$cell_id][$w_p_id][$w_p_mark] == 1 ) {
        $w_MST[$w_rule_id][$cell_id][$w_p_id][$w_p_mark] = 0 ; //***マーク削除***
        $w_fire_flag = 1 ;
    } else {
        if ( $w_MST[$w_rule_id] != "" ) { //***この発火条件にあう要素がすでにあったら
            unset($w_MST[$w_rule_id]); //***w_MSTからその要素を削除する***
        }
        $w_fire_flag = 0 ;
    }
} // end of if <<発火条件を調べる >>

//*** transition発火 *****
if ( $w_p_type == "o" and $w_check_end == 1 and $w_fire_flag == 1 ) {
    //*** 出力セル決定 *****
    if ( $w_o_towards == "all" ) { //*** 全セルに出力 *****
        for ( $i = 0 ; $i < $max_x ; $i++ ) {
            for ( $j = 0 ; $j < $max_y ; $j++ ) {
                $o_cell_id = "UC_" . sprintf('%05d', $i) . "_" . sprintf('%05d', $j) ;
            }
        }
    }
}

```

```

        $w_MST[$w_rule_id][$o_cell_id][$w_p_id][$w_p_mark] = 1;    /***マーク追加***/
    }
} else {
    if ( $w_o_towards == "=" ) {
        $o_cell_id = $cell_id;
    } else {
        if ( substr( $w_o_towards,0,2) == "UC" ) {
            $o_cell_id = $w_o_towards ;
        } else {
            $o_cell_id = decide_next_cell( $cell_id, $w_o_towards );
        }
    }
    if ( $o_cell_id != "" ) {
        $w_MST[$w_rule_id][$o_cell_id][$w_p_id][$w_p_mark] = 1;    /***マーク追加***/
    }
} // end of if <<出力セル決定>>

/***画面出力transitionの発火***/
if ( $o_cell_id == "SC_DISPLAY" ) {
    $active_display_cell = 1;
} // end of if << transition 発火 >>
} // end loop of << for ($z=0; $z<1; $z++) >>
} // end loop of << for ($count=0; $count < $ct_FRF; $count++) >>

/***このセルの最後の発火transitionの処理
if ( $w_fire_flag == 1 and $w_check_end == 1 ) {
    chg_MST( $cell_id, $w_rule_id );    /***MST更新***/
} // end of if $w_fire_flag ==1

}; // end of function run_PTN

/*** $w_p_markの変数および関数の処理
function hensu_henkan( $cell_id, $w_p_type, $w_p_id, $w_p_mark ) {

global $w_parm, $MST ;

/*** 関数処理
if ( substr($w_p_mark, 0, 2) == "%=" ) {
    $w_title = "";
    $w_shiki = substr($w_p_mark,2);
    $w_len = strlen($w_shiki);
    $w_pos = strpos($w_shiki,'[');
    $w_func = substr($w_shiki,0,$w_pos);
    $ww_title = substr( $w_shiki, $w_pos+1, $w_len-($w_pos+2) );
    $w_title = split(":",$ww_title);
    $ct_title = count($w_title);
    for ($i = 0 ; $i < $ct_title ; $i++) {
        if (substr($w_title[$i],0,1) == "%" ) {
            $w_hikisu[$i] = $w_parm[$w_title[$i]] ;
        } else {
            $w_hikisu[$i] = $w_title[$i];
        }
    }
}

switch($w_func) {
    case ('split') :
        case('split_any'):
        $w_value = "";
        $ww_pmk = "";

```

```

$ct_m = count($MST[$cell_id][$w_p_id]);
if ($ct_m > 0) {
    $w_k_pmk = array_keys($MST[$cell_id][$w_p_id]);
    for ($m = 0; $m < $ct_m; $m++) {
        if ($MST[$cell_id][$w_p_id][$w_k_pmk[$m]] == 1) {
            $w_p_mark = $w_k_pmk[$m];
            if ($w_func == "split_any") {
                $w_separator = substr($w_title[0],1,1);
                array_splice($w_title, 0,1);    /*** $w_titleの0番目の要素を削除
                $ct_title--;                    /*** $w_titleの要素数を1減らす
            } else {
                $w_separator = ":";
            }
            $w_value = split($w_separator, $w_p_mark);
        }
    }
}
if ($w_value != "") {
    $w_combi = array_combine($w_title, $w_value);
    if ($w_parm == "") {
        $w_parm = $w_combi;
    } else {
        $result = array_merge($w_parm, $w_combi);
        $w_parm = "";
        $w_parm = $result;
    }
}
break;

case('concat') :
case('concat_any') :
    /*** concat (変数の合成) ***/
    /*** 任意のセパレーターでのconcat(変数の合成) ***/
    if ($w_func == "concat_any") {
        $w_separator = substr($w_hikisu[0],1,1);
        array_splice($w_hikisu, 0,1);    /*** $w_hikisuの0番目の要素を削除する
        $ct_title--;                    /*** $w_titleの要素数を1減らす
    } else {
        $w_separator = ":";
    }
    $w_p_mark = $w_hikisu[0] . $w_separator . $w_hikisu[1];
    for ($i = 2; $i < $ct_title; $i++) {
        $w_p_mark = $w_p_mark . $w_separator . $w_hikisu[$i];
    }
    break;

case('min') :
    /*** 最小値(min) ***/
    $w_p_mark = min($w_hikisu);
    break;

case('max') :
    /*** 最大値(max) ***/
    $w_p_mark = max($w_hikisu);
    break;

case('+') :
    /*** 計算:加算(+) ***/
    $w_p_mark = $w_hikisu[0] + $w_hikisu[1];
    break;

case('-') :
    /*** 計算:減算(-) ***/
    $w_p_mark = $w_hikisu[0] - $w_hikisu[1];
    break;

case('*') :
    /*** 計算:乗算(*) ***/
    $w_p_mark = $w_hikisu[0] * $w_hikisu[1];
    break;

case('/') :
    /*** 計算:除算(/) ***/
    if ($w_hikisu[1] != 0) {
        $w_p_mark = sprintf("%01.2f", round($w_hikisu[0] / $w_hikisu[1], 2));
    }
    break;

case('%') :
    /*** 計算:余り(mod) ***/

```

```

        if ($w_hikisu[1] != 0) {
            $w_p_mark = $w_hikisu[0] % $w_hikisu[1];
        }
        break;
    case('>'):
        /*** 比較演算子(>) ***/
        if ( $w_hikisu[0] > $w_hikisu[1] ) {
            $w_p_mark = 1;
        } else {
            $w_p_mark = 0;
        }
        break;
    case('<'):
        /*** 比較演算子(<) ***/
        if ( $w_hikisu[0] < $w_hikisu[1] ) {
            $w_p_mark = 1;
        } else {
            $w_p_mark = 0;
        }
        break;
    case('='):
        /*** 比較演算子(=) ***/
        if ( $w_hikisu[0] == $w_hikisu[1] ) {
            $w_p_mark = 1;
        } else {
            $w_p_mark = 0;
        }
        break;
    default :
    } // end of switch case
} // end of 関数処理

/*** 変数処理_input ($w_parmにデータを書き込む) *****/
if ($w_p_type == "i") {
    if ( substr($w_p_mark, 0, 1) == "%"
        and substr($w_p_mark, 0, 2) != "%=" ) {
        $w_value = "";
        $ww_pmk = "";
        $ct_m = count($MST[$cell_id][$w_p_id]);
        if ($ct_m > 0) {
            $w_k_pmk = array_keys($MST[$cell_id][$w_p_id]);
            for ($m = 0; $m < $ct_m; $m++) {
                if ( $MST[$cell_id][$w_p_id][$w_k_pmk[$m]] == 1 ) {
                    $w_parm[$w_p_mark] = $w_k_pmk[$m];
                    $w_p_mark = $w_k_pmk[$m];
                }
            }
        }
    }
}

/*** 変数処理_output ($w_parmからデータを取り込む) *****/
if ( $w_p_type == "o" ) {
    if ( substr($w_p_mark, 0, 1) == "%"
        and substr($w_p_mark, 0, 2) != "%=" ) {
        $w_value = $w_parm[$w_p_mark];
        $w_p_mark = $w_value;
    }
}

return $w_p_mark ;

}; // end of function hensu_henkan

/***/
/*** 発火条件の変更に伴いMSTの更新 *****/

```

```

//*****
function chg_MST( $cell_id, $w_rule_id)  {

    global $count ;
    global $w_MST, $MST ;

    $ct_c = count($w_MST[$w_rule_id]) ;
    $w_k_cid = array_keys ($w_MST[$w_rule_id]) ;
    for ($i = 0; $i < $ct_c; $i++) {
        add_o_ACF( $w_k_cid[$i] ) ;                //***次回実行するセルリストに追加する
        $ct_p = count($w_MST[$w_rule_id][$w_k_cid[$i]]) ;
        $w_k_pid = array_keys ($w_MST[$w_rule_id][$w_k_cid[$i]]) ;
        for ($k = 0; $k < $ct_p; $k++) {
            $ct_m = count($w_MST[$w_rule_id][$w_k_cid[$i]][$w_k_pid[$k]]) ;
            $w_k_pmk = array_keys ($w_MST[$w_rule_id][$w_k_cid[$i]][$w_k_pid[$k]]) ;
            for ($m = 0; $m < $ct_m; $m++) {
                $ww_cid = $w_k_cid[$i] ;
                $ww_pid = $w_k_pid[$k] ;
                $ww_pmk = $w_k_pmk[$m] ;

                $MST[$ww_cid][$ww_pid][$ww_pmk] = $w_MST[$w_rule_id][$ww_cid][$ww_pid][$ww_pmk] ;
            } // end loop for (m)
        } // end loop for (k)
    } // end loop for (i)
}; // end of function chg_MST

```

```

//*****
//*** 出力セルリストに追加し、必要ならMSFを読む *****
//*****
function add_o_ACF( $cell_id)  {

    global $dir ;
    global $ACF, $next_ACF ;

    //***出力セルがあたらしいセルかどうかを確認する
    $hit = array_search($cell_id, $next_ACF, true) ;
    if ( $hit == FALSE ) {
        $next_ACF[] = $cell_id;                //****next_ACFに存在しない場合
        $hit1 = array_search($cell_id, $ACF, true) ;                //****next_ACFに追加する
        if ( $hit1 == FALSE ) {
            read_MSF ( $cell_id) ;                //****ACFに存在しない場合
            //****MSFを読む
        }
    }
}; // end of function add_o_ACF

```

```

//*****
//***次のセルの決定*****
//*****
function decide_next_cell( $cell_id, $w_cell_to )  {

    global $max_x, $max_y ;
    global $count ;

    $o_cell_id = $cell_id ;                //***基本的に自分自身***
    $w_cell_num_x = substr($cell_id, 3,5) ;                //***セル番号(x)は4桁目以降
    $w_cell_num_y = substr($cell_id, 9,5) ;                //***セル番号(y)は10桁目以降
    $w_next_to = split("_", $w_cell_to) ;
    $w_next_to_x = $w_next_to[0] ;                //***X軸方向の動き方
    $w_next_to_y = $w_next_to[1] ;

    if ( ereg("[0-9]", $w_next_to_x)                //***数字のみか?

```

```

and ereg("[0-9]",substr($w_next_to_x,0,1)) {
    $w_next_to_x = "+" . $w_next_to_x ;
}
switch ( substr($w_next_to_x,0,1) ) {
    case '=' :
    case '0' : $o_cell_id_x = sprintf('%05d',$w_cell_num_x) ;
                break ;
    case '+' : $no = substr($w_next_to_x,1) ;
                if ( ctype_digit( $no ) == FALSE ) { $no = 1 ; }
                $o_cell_id_x = sprintf('%05d',$w_cell_num_x + $no) ;
                break ;
    case '-' : $no = substr($w_next_to_x,1) ;
                if ( ctype_digit( $no ) == FALSE ) { $no = 1 ; }
                $o_cell_id_x = sprintf('%05d',$w_cell_num_x - $no) ;
                break ;
    case 'C' : $o_cell_id_x = sprintf('%05d', $w_cell_num_x + 1) ;
                if ( substr($w_next_to_x,0,2) == "C1" ) {
                    $o_cell_id_y = y_keisan_1( $o_cell_id_x, $w_next_to_y ) ;
                }
                if ( substr($w_next_to_x,0,2) == "C2" ) {
                    $o_cell_id_y = y_keisan_2( $o_cell_id_x, $w_next_to_y ) ;
                }
                break ;
    case 'R' : $o_cell_id_x = sprintf('%05d', rand(0,$max_x-1)) ; //***乱数発生
                break ;
    case 'D' : $o_cell_id = "SC_DISPLAY" ;
                break ;
    default   :
}

if ( ereg("[0-9]", $w_next_to_y)
and ereg("[0-9]",substr($w_next_to_y,0,1)) ) {
    $w_next_to_y = "+" . $w_next_to_y ;
}
switch ( substr($w_next_to_y,0,1) ) {
    case '=' :
    case '0' : $o_cell_id_y = sprintf('%05d',$w_cell_num_y) ;
                break ;
    case '+' : $no = substr($w_next_to_y,1) ;
                if ( ctype_digit( $no ) == FALSE ) { $no = 1 ; }
                $o_cell_id_y = sprintf('%05d',$w_cell_num_y + $no) ;
                break ;
    case '-' : $no = substr($w_next_to_y,1) ;
                if ( ctype_digit( $no ) == FALSE ) { $no = 1 ; }
                $o_cell_id_y = sprintf('%05d',$w_cell_num_y - $no) ;
                break ;
    case 'R' : $o_cell_id_y = sprintf('%05d', rand(0,$max_y-1)) ; //***乱数発生
                break ;
    default   :
}

if ( $o_cell_id_x < 0 ) { $o_cell_id_x = "" ; }
if ( $o_cell_id_x >= $max_x ) { $o_cell_id_x = "" ; }
if ( $o_cell_id_y < 0 ) { $o_cell_id_y = "" ; }
if ( $o_cell_id_y >= $max_y ) { $o_cell_id_y = "" ; }

if ( substr( $o_cell_id ,0,2) == "UC" ) {
    if ( $o_cell_id_x != "" and $o_cell_id_y != "" ) {
        $o_cell_id = "UC_" . $o_cell_id_x . "_" . $o_cell_id_y ;
    } else {
        $o_cell_id = "" ; //***最大値、最小値にきたらストップ
    }
}
}

```

```

return $o_cell_id ;

};    //***end of function decide_next_cell

//*****
//*** Y軸の値の計算(1次方程式: y=ax+b) *****
//*****
function y_keisan_1($x, $a_b)    {

$w_len = strlen($a_b) ;
$ww = substr($a_b,1,$w_len-2) ; //*** 前後の()を取り除く
$w_value = split("&", $ww) ;    //*** "&"で二つの変数にわけ
$a = $w_value[0] ;
$b = $w_value[1] ;
$y = sprintf("%05d", $a * $x + $b ) ;

return $y ;

};    //***end of function y_keisan_1

//*****
//*** Y軸の値の計算(2次方程式: y=a*x*x+b*x + c) *****
//*****
function y_keisan_2($x, $a_b_c)    {

$w_len = strlen($a_b_c) ;
$ww = substr($a_b_c,1,$w_len-2) ;
$w_value = split("&", $ww) ;
$a = $w_value[0] ;
$b = $w_value[1] ;
$c = $w_value[2] ;
$y = sprintf("%05d", $a * $x * $x + $b * $x + $c ) ;

return $y ;

};    //***end of function y_keisan_2

//*****
//*** 各セルの表示する色を決定する *****
//*****
function decide_color( )    {

global $dir, $max_x, $max_y ;
global $MST, $w_COL, $next_ACF ;

//***MST["SC_DISPLAY"]がなかったらMSFを読む
$hit1 = array_search("SC_DISPLAY", $next_ACF, true) ;
if ( $hit1 == FALSE ) {    //****ACFに存在しない場合
    read_MSF ("SC_DISPLAY") ;    //****MSFを読む
}

//***MSTをみて、各セルの色を決定する
$w_chg_SPR = 0 ;
$w_cell_id = "SC_DISPLAY" ;

for ( $i = 0; $i < $max_x; $i++ ) {
    for ( $j = 0; $j < $max_y; $j++ ) {
        $w_pid = "UC_" . sprintf("%05d", $i) . "_" . sprintf("%05d", $j) ;
        //***SPR11***
        if ( $MST[$w_cell_id][$w_pid]["Red"] == 1 ) {
            $MST[$w_cell_id][$w_pid]["Blue"] = 0 ;
            $MST[$w_cell_id][$w_pid]["Green"] = 0 ;

```

```

    $MST[$w_cell_id][$w_pid]["White"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Grey"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Red_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Blue_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Green_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["White_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Grey_x"] = 0 ;
}
/**SPR12**
if ( $MST[$w_cell_id][$w_pid]["Blue"] == 1 ) {
    $MST[$w_cell_id][$w_pid]["Green"] = 0 ;
    $MST[$w_cell_id][$w_pid]["White"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Grey"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Red_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Blue_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Green_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["White_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Grey_x"] = 0 ;
}
/**SPR13**
if ( $MST[$w_cell_id][$w_pid]["Green"] == 1 ) {
    $MST[$w_cell_id][$w_pid]["White"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Grey"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Red_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Blue_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Green_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["White_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Grey_x"] = 0 ;
}
/**SPR14**
if ( $MST[$w_cell_id][$w_pid]["White"] == 1 ) {
    $MST[$w_cell_id][$w_pid]["Grey"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Red_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Blue_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Green_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["White_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Grey_x"] = 0 ;
}
/**SPR15**
if ( $MST[$w_cell_id][$w_pid]["Grey"] == 1 ) {
    $MST[$w_cell_id][$w_pid]["Red_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Blue_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Green_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["White_x"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Grey_x"] = 0 ;
}
/**SPR31**
if ( $MST[$w_cell_id][$w_pid]["Red"] == 1 ) {
    $w_chg_SPR = 1 ;
    $MST[$w_cell_id][$w_pid]["Red"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Red_x"] = 1 ;
}
/**SPR32**
if ( $MST[$w_cell_id][$w_pid]["Blue"] == 1 ) {
    $w_chg_SPR = 1 ;
    $MST[$w_cell_id][$w_pid]["Blue"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Blue_x"] = 1 ;
}
/**SPR33**
if ( $MST[$w_cell_id][$w_pid]["Green"] == 1 ) {
    $w_chg_SPR = 1 ;
    $MST[$w_cell_id][$w_pid]["Green"] = 0 ;
    $MST[$w_cell_id][$w_pid]["Green_x"] = 1 ;
}

```

```

    }
    /****SPR34***/
    if ( $MST[$w_cell_id][$w_pid]["White"] == 1 ) {
        $w_chg_SPR = 1 ;
        $MST[$w_cell_id][$w_pid]["White"] = 0 ;
        $MST[$w_cell_id][$w_pid]["White_x"] = 1 ;
    }
    /****SPR35***/
    if ( $MST[$w_cell_id][$w_pid]["Grey"] == 1 ) {
        $w_chg_SPR = 1 ;
        $MST[$w_cell_id][$w_pid]["Grey"] = 0 ;
        $MST[$w_cell_id][$w_pid]["Grey_x"] = 1 ;
    }
}
}

/****MSFにMSTの変更部分を書き出す
if ( $w_chg_SPR == 1 ) {
    write_MSF("SC_DISPLAY");
}

/****画面表示のための配列$w_COLに色をセット
$Grey = "G";
$Red = "R";
$Blue = "B";
$Yellow = "Y";
$Green = "M";
$White = "W";

/****各セルの色をセット
for ( $i = 0; $i < $max_x; $i++ ) {
    for ( $j = 0; $j < $max_y; $j++ ) {
        $w_pid = "UC_" . sprintf("%05d", $i) . "_" . sprintf("%05d", $j);
        /****Red
        if ( $MST[$w_cell_id][$w_pid]["Red_x"] == 1 ) {
            $w_COL[$i][$j] = $Red ;
        }
        /****Blue
        if ( $MST[$w_cell_id][$w_pid]["Blue_x"] == 1 ) {
            $w_COL[$i][$j] = $Blue ;
        }
        /****Green
        if ( $MST[$w_cell_id][$w_pid]["Green_x"] == 1 ) {
            $w_COL[$i][$j] = $Green ;
        }
        /****White
        if ( $MST[$w_cell_id][$w_pid]["White_x"] == 1 ) {
            $w_COL[$i][$j] = $White ;
        }
        /****Grey
        if ( $MST[$w_cell_id][$w_pid]["Grey_x"] == 1 ) {
            $w_COL[$i][$j] = $Grey ;
        }
    }
}

}; // end of function decide_color

/****
/**** imagepngを用いて結果を描画する
/****
function display_image ( ) {

```

```

global $now_time, $step_no, $max_x, $max_y, $dir;
global $w_COL ;

/**前回は描画したファイルを削除***/
if ($handle = opendir($dir)) {
    while (false !== ($file = readdir($handle))) {
        $w_value = split('[./]', $file);
        $w_ext = $w_value[1];
        if ($w_ext == "png") {
            unlink ($dir.$file);
        }
    }
    closedir($handle);
}

$img_file = $dir . "IMG_" . sprintf('%03d', $step_no) . "_" . $now_time . ".png";

/**X軸、Y軸それぞれのセルの大きさと全体の大きさを決定***/
if ($max_x > 99) {
    $w_size_x = 4;
} else {
    if ($max_x > 39) {
        $w_size_x = 8;
    } else {
        $w_size_x = 16;
    }
}
$w_width = $w_size_x * $max_x;

if ($max_y > 99) {
    $w_size_y = 4;
} else {
    if ($max_y > 39) {
        $w_size_y = 8;
    } else {
        $w_size_y = 16;
    }
}
$w_height = $w_size_y * $max_y;

/**描画開始***/
header("Content-type: image/png");

$imgWidth = $w_width + 1;
$imgHeight = $w_height + 1;

$image = imagecreate($imgWidth, $imgHeight);

$colorWhite = imagecolorallocate($image, 255, 255, 255);
$colorGrey = imagecolorallocate($image, 192, 192, 192);
$colorBlue = imagecolorallocate($image, 0, 0, 255);
$colorRed = imagecolorallocate($image, 255, 0, 0);
$colorGreen = imagecolorallocate($image, 0, 255, 0);

/**外枠描画***/
imageline($image, 0, 0, 0, $w_height, $colorGrey); //***左***
imageline($image, 0, 0, $w_width, 0, $colorGrey); //***上***
imageline($image, $w_width, 0, $w_width, $w_height, $colorGrey); //***右***
imageline($image, 0, $w_height, $w_width, $w_height, $colorGrey); //***下***

/**仕切り線描画***/
for ($i=1; $i<$max_x; $i++) { //***縦罫***
    imageline($image, $i * $w_size_x, 0, $i * $w_size_x, $w_height, $colorGrey);
}

```

```

}
for ($j=1; $j<$max_y; $j++) {
    imageline($image, 0, $j * $w_size_y, $w_width, $j * $w_size_y, $colorGrey);
}

// ***各セルの色決定 & 描画*****
for ($i=0; $i<$max_x; $i++) {
    for ($j=0; $j<$max_y; $j++) {
        $color = $colorGrey;
        switch ($w_COL[$i][$max_y - $j-1]) {
            case 'G': $color = $colorGrey; break;
            case 'R': $color = $colorRed; break;
            case 'B': $color = $colorBlue; break;
            case 'M': $color = $colorGreen; break;
            case 'W': $color = $colorWhite; break;
            default :
        }
        $from_x = $i * $w_size_x + 1;
        $to_x = ($i+1) * $w_size_x - 1;
        $from_y = $j * $w_size_y + 1;
        $to_y = ($j+1) * $w_size_y - 1;
        imagefilledrectangle($image, $from_x, $from_y, $to_x, $to_y, $color);
    }
}

imagepng($image, $img_file );
imagedestroy($image);

header("Content-type: text/html");
print " <IMG src=$img_file><br> ¥n ";

}; //***end of function display_image

?>

```

```

<FORM method="GET" action="CELL_PTN_setup_6.php">
<?php
print "<input type='hidden' name='FRF_name' value=$frf_name size=30> ¥n ";
print "<input type='hidden' name='INI_name' value=$ini_name size=30> ¥n ";
print "<input type='hidden' name='max_x' value=$max_x size=5> ¥n ";
print "<input type='hidden' name='max_y' value=$max_y size=5> ¥n ";
print "<input type='hidden' name='start_id_1_x' value=$s_pos_1x size=5> ¥n ";
print "<input type='hidden' name='start_id_1_y' value=$s_pos_1y size=5> ¥n ";
print "<input type='hidden' name='start_to_1_x' value=$s_to_1x size=10> ¥n ";
print "<input type='hidden' name='start_to_1_y' value=$s_to_1y size=10> ¥n ";
print "<input type='hidden' name='start_col_1' value=$s_col_1 size=6> ¥n ";
print "<input type='hidden' name='start_parm_1' value=$s_parm_1 size=6> ¥n ";
print "<input type='hidden' name='start_id_2_x' value=$s_pos_2x size=5> ¥n ";
print "<input type='hidden' name='start_id_2_y' value=$s_pos_2y size=5> ¥n ";
print "<input type='hidden' name='start_to_2_x' value=$s_to_2x size=10> ¥n ";
print "<input type='hidden' name='start_to_2_y' value=$s_to_2y size=10> ¥n ";
print "<input type='hidden' name='start_col_2' value=$s_col_2 size=6> ¥n ";
print "<input type='hidden' name='start_parm_2' value=$s_parm_2 size=6> ¥n ";
print "<input type='hidden' name='start_id_3_x' value=$s_pos_3x size=5> ¥n ";
print "<input type='hidden' name='start_id_3_y' value=$s_pos_3y size=5> ¥n ";
print "<input type='hidden' name='start_to_3_x' value=$s_to_3x size=10> ¥n ";
print "<input type='hidden' name='start_to_3_y' value=$s_to_3y size=10> ¥n ";

```

```

print " <input type='hidden' name='start_col_3' value='$s_col_3' size=6> \n" ;
print " <input type='hidden' name='start_parm_3' value='$s_parm_3' size=6> \n" ;
print " <input type='hidden' name='start_id_4_x' value='$s_pos_4x' size=5> \n" ;
print " <input type='hidden' name='start_id_4_y' value='$s_pos_4y' size=5> \n" ;
print " <input type='hidden' name='start_to_4_x' value='$s_to_4x' size=10> \n" ;
print " <input type='hidden' name='start_to_4_y' value='$s_to_4y' size=10> \n" ;
print " <input type='hidden' name='start_col_4' value='$s_col_4' size=6> \n" ;
print " <input type='hidden' name='start_parm_4' value='$s_parm_4' size=6> \n" ;
print " <input type='hidden' name='max_step' value='$max_step' size=5> \n" ;
print " <input type='hidden' name='step_no' value='$step_no' size=5> \n" ;
print " <input type='hidden' name='parm_disp' value='$parm_disp' size=5> \n" ;
?>
<input type="submit" value="戻る">
</FORM>

<FORM method="GET" action="CELL_PTN_setup_6.php">
<input type="submit" value="実行完了">
</FORM>

</BODY>
</HTML>

```

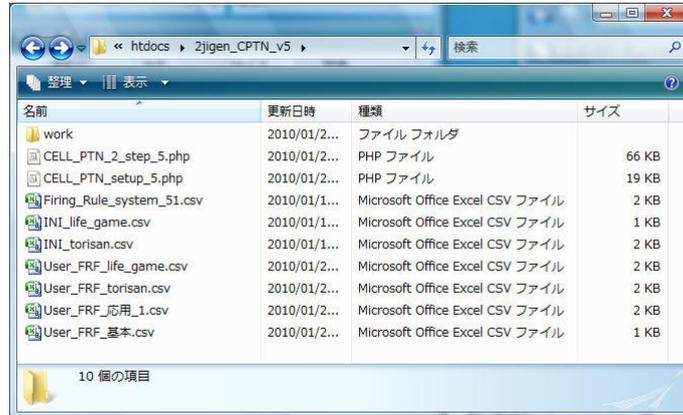
### III. システム F R F (Firing\_Rule\_system\_51.csv)

*	rule#	transition id	input/ output	o_cell id	place id	mark
*	SR30	ST3	i		to_next	%=split[%o_towards:%o_color:%pparm]
*	SR30	ST3	o	%o_toward:	receive	%=concat[%o_towards:%o_color:%pparm]
*	SR31	ST3	i		to_next	%=split[%o_towards:%o_color:%pparm]
*	SR31	ST3	o	%o_toward:	receive	%=concat[%o_towards:%o_color:%pparm]
*	SR32	ST3	i		to_next	%=split[%o_towards:%o_color:%pparm]
*	SR32	ST3	o	%o_toward:	receive	%=concat[%o_towards:%o_color:%pparm]
*	SR33	ST3	i		to_next	%=split[%o_towards:%o_color:%pparm]
*	SR33	ST3	o	%o_toward:	receive	%=concat[%o_towards:%o_color:%pparm]
*	SR34	ST3	i		to_next	%=split[%o_towards:%o_color:%pparm]
*	SR34	ST3	o	%o_toward:	receive	%=concat[%o_towards:%o_color:%pparm]
*	SR35	ST3	i		to_next	%=split[%o_towards:%o_color:%pparm]
*	SR35	ST3	o	%o_toward:	receive	%=concat[%o_towards:%o_color:%pparm]
*	SR36	ST3	i		to_next	%=split[%o_towards:%o_color:%pparm]
*	SR36	ST3	o	%o_toward:	receive	%=concat[%o_towards:%o_color:%pparm]
*	SR37	ST3	i		to_next	%=split[%o_towards:%o_color:%pparm]
*	SR37	ST3	o	%o_toward:	receive	%=concat[%o_towards:%o_color:%pparm]
*	SR38	ST3	i		to_next	%=split[%o_towards:%o_color:%pparm]
*	SR38	ST3	o	%o_toward:	receive	%=concat[%o_towards:%o_color:%pparm]
*	SR39	ST3	i		to_next	%=split[%o_towards:%o_color:%pparm]
*	SR39	ST3	o	%o_toward:	receive	%=concat[%o_towards:%o_color:%pparm]
*	SR391	ST3	i		to_next	%=split[%o_towards:%o_color:%pparm]
*	SR391	ST3	o	%o_toward:	receive	%=concat[%o_towards:%o_color:%pparm]
*	SR41	ST4	i		to_display	Grey
*	SR41	ST4	o	Display	%cell_id	Grey
*	SR42	ST4	i		to_display	Green
*	SR42	ST4	o	Display	%cell_id	Green
*	SR43	ST4	i		to_display	Blue
*	SR43	ST4	o	Display	%cell_id	Blue
*	SR44	ST4	i		to_display	Red
*	SR44	ST4	o	Display	%cell_id	Red
*	SR45	ST4	i		to_display	White
*	SR45	ST4	o	Display	%cell_id	White

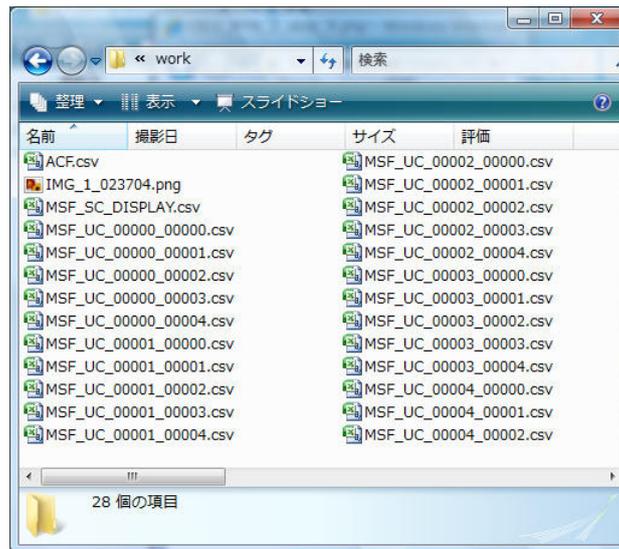
## IV. ACF、MSFの例

第7章の“基本的な動きの例”を実行したときの step1 実行直後の状態

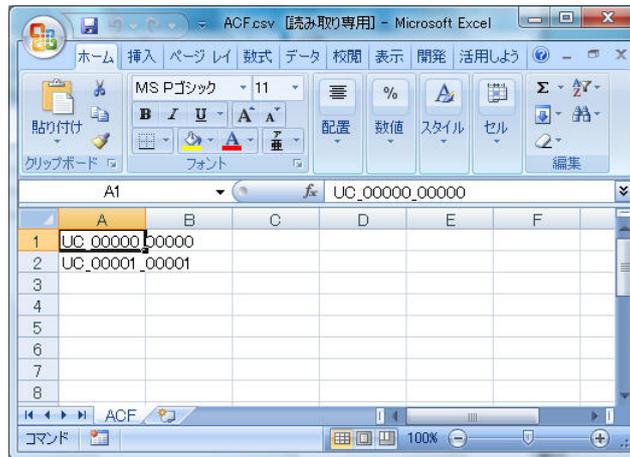
### ◆ 実行プログラムがあるフォルダーのファイル一覧



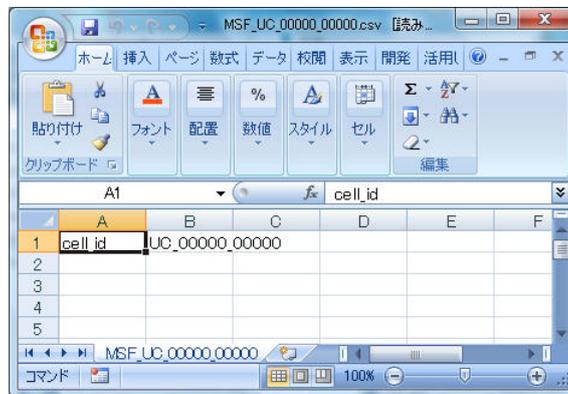
### ◆ 実行プログラムがあるフォルダー内のファイル一覧



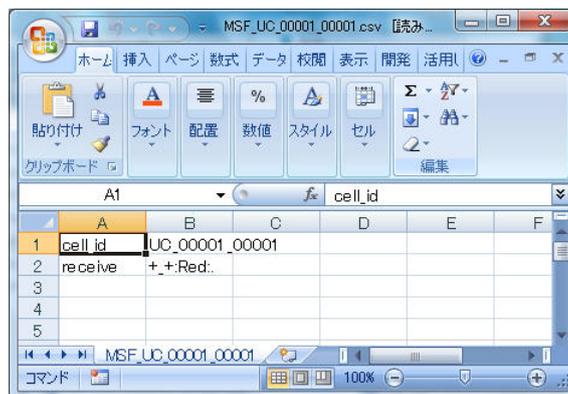
◆ ACF.csv



◆ MSF\_UC\_00000\_00000.csv



◆ MSF\_UC\_00001\_00001.csv



◆ MSF\_SC\_DISPLAY.csv

	A	B	C	D	E	F
1	cell_id	SC_DISPLAY				
2	UC_00000_Red_x					
3	UC_00000_Grey_x					
4	UC_00000_Grey_x					
5	UC_00000_Grey_x					
6	UC_00000_Grey_x					
7	UC_00001_Grey_x					
8	UC_00001_Grey_x					
9	UC_00001_Grey_x					
10	UC_00001_Grey_x					
11	UC_00001_Grey_x					
12	UC_00002_Grey_x					
13	UC_00002_Grey_x					
14	UC_00002_Grey_x					
15	UC_00002_Grey_x					
16	UC_00002_Grey_x					
17	UC_00003_Grey_x					
18	UC_00003_Grey_x					
19	UC_00003_Grey_x					
20	UC_00003_Grey_x					
21	UC_00003_Grey_x					
22	UC_00004_Grey_x					
23	UC_00004_Grey_x					
24	UC_00004_Grey_x					
25	UC_00004_Grey_x					
26	UC_00004_Grey_x					
27						

Step2 実行直後の状態

◆ ACF.csv

	A	B	C	D	E	F
1	UC_00001_00001					
2	UC_00002_00002					
3						
4						
5						

◆ MSF\_UC\_00001\_00001.csv

	A	B	C	D	E	F
1	cell_id	UC_00001_00001				
2						
3						
4						
5						

◆ MSF\_UC\_00002\_00002.csv

	A	B	C	D	E	F
1	cell_id	UC_00002_00002				
2	receive	+.+.Red.				
3						
4						
5						