

セルペトリネットツール

信州大学大学院 情報工学専攻修士論文

大矢長門

概要

➡ 1. はじめに



➡ よく表計算ソフトの WorkSheet の Cell に入力して、各々の思うとおりの様々な処理を行うことができる。又はマクロを使えば、静的な挙動ならばほとんどのことは表計算ソフトでこと足りる。だが、動的な動きを目で確認したいときや、並列処理などのシミュレーションはプログラムを書かなければならないことが多い。



概要

- ➡ そのようなときに、表計算ソフトのように手軽に試すことの出来るものを作ることが出来ないかと考えた結果セルペトリネットツールというものを作ってみたら、どのようなことが出来るのかと考察し作った。
- ➡ 処理手順はファイルを読み込みN行M列のセルペトリネットを自動で生成し、token 1つが1 dot の点で画面に表示され、処理結果をファイルへ書き出す。
- ➡ 分散処理を行う上で必要な機能を盛り込んでみた。そして、このセルペトリネットでは様々なシミュレーションを目で見ながら、誰でも簡単に使うことが出来るようになるようにした。

Cell Petri Net の定義

P (Place), T (Transition), I (Input Place), O (Output Place), CS
(ColorSet)

PTN(Place/Transition Net)

CPN(CellPetriNet)

$$P \cap T = \emptyset, I \subseteq P \times T, O \subseteq T \times P$$

$$\text{アークの集合: } F \subseteq (P \times T) \cup (T \times P)$$

A Cell Petri Net

$$\text{CPN} = (\text{PTN}, \text{CS}, \rho)$$

$$\text{PTN} = (\text{P}, \text{T}, \text{I}, \text{O}),$$

$$\text{P} = \{p_0\}, \text{T} = \{t_0\}, \text{I} = \{[p_0, t_0]\}, \text{O} = \phi$$

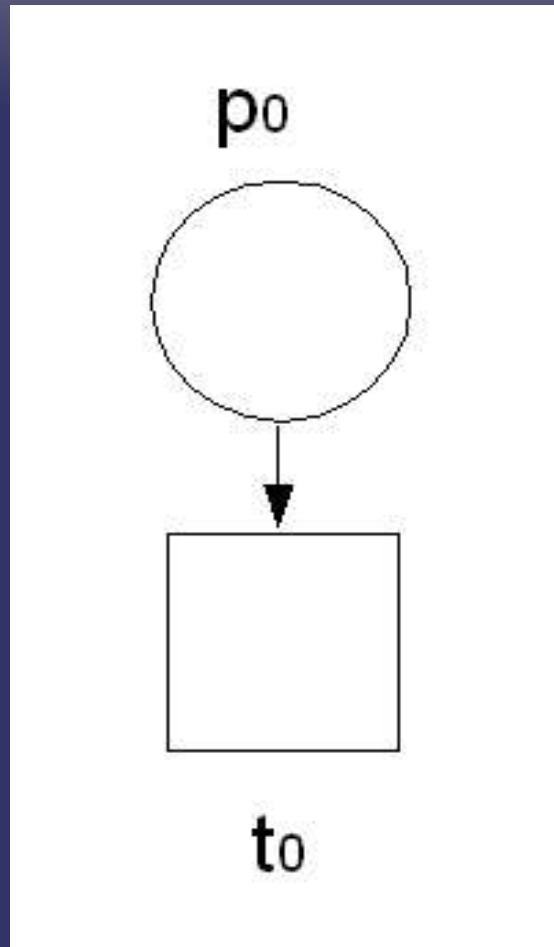


Figure 1: A Petri net cell.

1次元配列の Cell について

$$P = \{p_1, \dots, p_n\}$$

$$T = \{t_1, \dots, t_n\}$$

$$I = \{[p_1, t_1], \dots, [p_n, t_n]\}$$

$$O = \{[t_1, p_1], \dots, [t_i, p_{i-1}], [t_i, p_{i+1}], \dots, [t_n, p_n]\}$$

$$C = \{c_1, \dots, c_n\}$$

Liniar Cell Petri Net

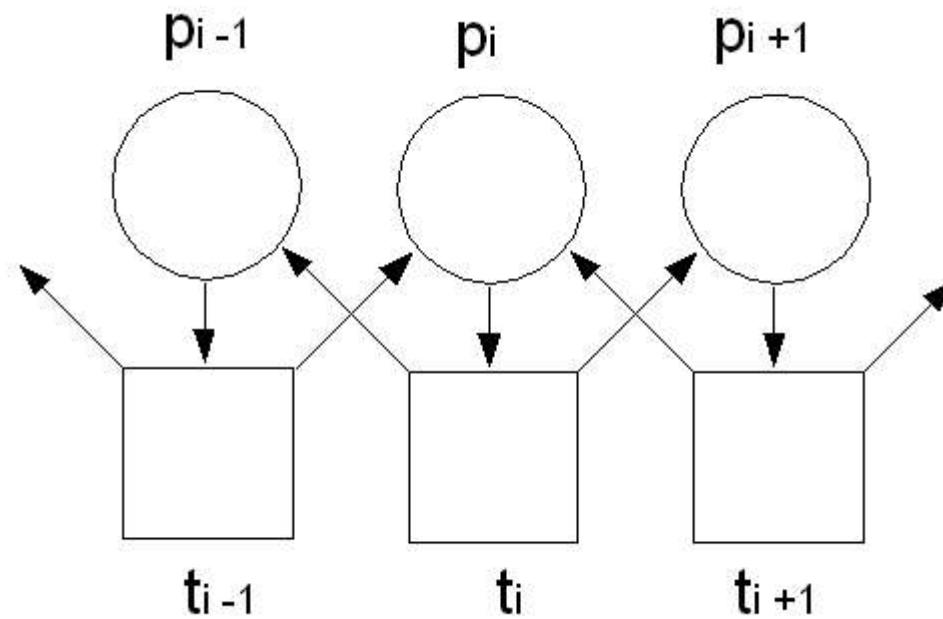


Figure 2: Liniar Petri net cell.

Liniar Cell Petri Net

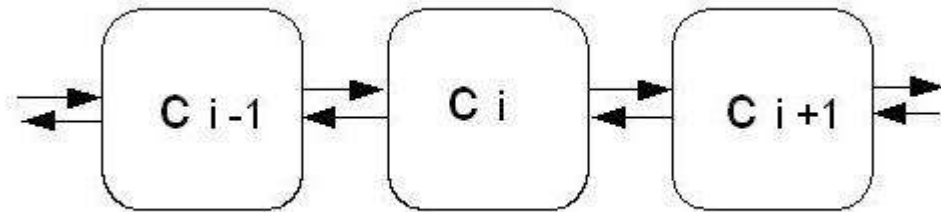


Figure 3: Liniar Petri net cell upside.

N 行 M 列 Cell Petri Net

$$P = \{p_{ij}\}, T = \{t_{ij}\}, I = \{[p_{ij}, t_{ij}]\},$$

$$O_{ij} = \{[t_{ij}, p_{i-1 j}], [t_{ij}, p_{i+1 j}], [t_{ij}, p_{i j-1}], [t_{ij}, p_{i j+1}]\} \quad \text{ただし } 0 \leq i \pm 1 \leq N, 0 \leq j \pm 1 \leq M$$

簡略の為、N 行 M 列のセルペトリネットの場合には以下の様に記す。

$$P = \{[p_{nm}]\}$$

$$T = \{[t_{nm}]\}$$

$$I = \{[p_{nm}, t_{nm}]\}$$

$$O = \{[t_{nm}, p_{n \pm 1 m \pm 1}]\} \quad \text{ただし } 0 \leq n \pm 1 \leq N, 0 \leq m \pm 1 \leq M$$

$$C = \{c_{11}, \dots, c_{nm}\}$$

j 列目 Sheet Cell Petri Net

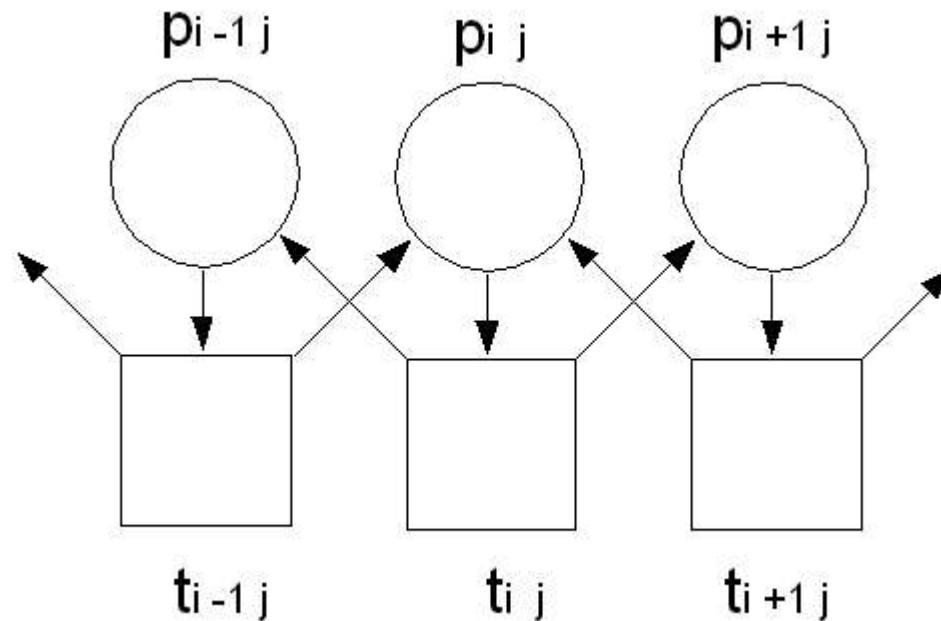


Figure 4: sheet Petri net cell colum.

Sheet Cell Petri Net(2次元)

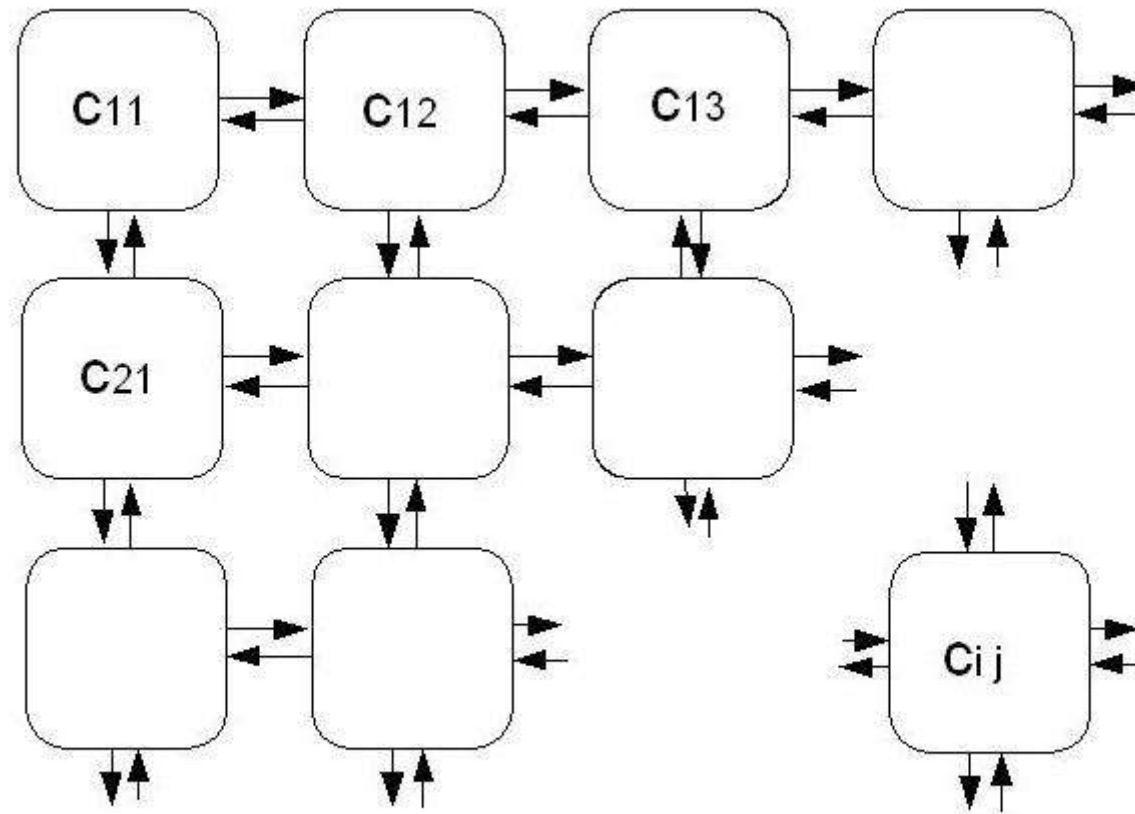


Figure 5 :Sheet Petri net cell.

i 行の Sheet Petri net cell 発火条件

firing rule は

$$t_{ij} \in T_{ij}$$

$$O_{ij kj}(t_{ij}) = \begin{cases} \{p_{i+1 j}\} & \cdots \text{ if } k=i+1 \\ \{p_{i-1 j}\} & \cdots \text{ if } k=i-1 \\ \phi & \cdots \text{ otherwise} \end{cases}$$

j列の Sheet Petri net cell 発火条件

$$O_{ij ik}(t_{ij}) = \begin{cases} \{p_{i j+1}\} & \cdots \text{ if } k=j+1 \\ \{p_{i j-1}\} & \cdots \text{ if } k=j-1 \\ \phi & \cdots \text{ otherwise} \end{cases}$$

絶対角度だとすると、

$$\rho_{ij ik}(t_{ij}, (p_{ij}, 0^\circ)) = \begin{cases} \{(p_{i+1 j}, 0^\circ)\} & \cdots \text{ if } k=i+1 \\ \phi & \cdots \text{ otherwise} \end{cases}$$

$$\rho_{ij ik}(t_{ij}, (p_{ij}, 180^\circ)) = \begin{cases} \{(p_{i-1 j}, 180^\circ)\} & \cdots \text{ if } k=i-1 \\ \phi & \cdots \text{ otherwise} \end{cases}$$

$$\rho_{ij ik}(t_{ij}, \phi) = \phi$$

$$\rho_{ij kj}(t_{ij}, (p_{ij}, 270^\circ)) = \begin{cases} \{(p_{i j+1}, 270^\circ)\} & \cdots \text{ if } k=j+1 \\ \phi & \cdots \text{ otherwise} \end{cases}$$

$$\rho_{ij kj}(t_{ij}, (p_{ij}, 90^\circ)) = \begin{cases} \{(p_{i j-1}, 90^\circ)\} & \cdots \text{ if } k=j-1 \\ \phi & \cdots \text{ otherwise} \end{cases}$$

$$\rho_{ij kj}(t_{ij}, \phi) = \phi$$

Token(marking)と発火条件

4.1 marking(tokenの配置)

通常 token は Place 数次元の非負整数ベクトル R

CS は RGB (赤緑青) に対応させ、 R^3 すなわち $0 \sim 256 \times 256 \times 256$ までの値とする。

token に遅延速度を与えることが出来る

4.2 place

待ち: (通常は待ちなし)

4.3 transition

firing change rule(発火変更条件)を満たしていないと token の進行方向は変わらない。
条件変更はセル(,)において token の値{,,,}と m の比較によってなされる。

4.4 token(mark)

振る舞い

token の動きは

$$m = [m(p_1), m(p_2), \dots, m(p_x)]$$

$m(p_i)$ は place p_i の mark(token)の数

4.5 Arc

place から transition へは arc の本数を token が超えたら、arc の数だけ token を送り出せる。
arc の本数を arc の重み $W:F \rightarrow \{1, 2, \dots\}$ で表す。

5. セルペトリネットを使いやすくする為の概念

5.1 セルペトリネットを使った発火条件 ρ 成立の指定

5.2 タイムペトリネット(timed Petri net)の導入

発火遅延時間モデル
無反応(待ち)条件
遅延(slow)0~

5.3 オブジェクトペトリネット(object Petri net) token がデータなどを操作できる

5.4 進行条件(-1 は停止) 方向(argument)は 0,90,180,270 の4方向となる。

モジュール化

CellPetriNet はタイムペトリネットを使えるので、一つの Place から n 個の Transition、もしくは一つの Transition から n 個の place への置き換えをすることができる。それは、 n 個の Cell を使ってある働きを持った Cell 群へと変えることが出来る。そして、発火条件の範囲指定によって、token の流れを制御することができる。従って、このことにより、ペトリネットの処理がシミュレーション可能となる。

ファイルへの記入方法

ファイル名は Petri.Dat に記載しておく自動的に読み込むように作ってあります。
又は、別ファイルを読み込みたいときには、

C:¥ Petri ファイル名

と入力しても良いです。

6.1 画面表示

Cell[X 座標,Y 座標]

Cell[200,300] …200×300 のセルを作る

Cell[200,300]0 …200×300 のセルを作り、動作経路を消さない。

6.2 token の値

m(X 座標,Y 座標){角度,値,遅さ,待ち}

m(50,20){90,16777215,1,0} …token の属性は位置が x50y20,角度 90,値は最大,遅さ,待ち}

位置:x,y

進行角度:0,90,180,360,(停止は-1)

値:0~16777215=(256×256×256)-1

遅さ:0~

待ち:0~

6.3 Transition の判定基準

cell(){,,}

cell(5,10){+1,,}=6 Cs が6ならば、Cell(5,10)でCSに1をたす

cell(5,10){270,,}>90 角度が90より大きければ、Cell(5,10)で絶対角度 270° 方向へ曲がる

cell(5,10){*3,,} CSを3倍する

7. 結び

CellPetriNet は動的な動きを画面上に描き出せ、どの様な処理をしているかが一目で分かる。
セル各々が独立処理であるので、一部のバグではHangしない。
セルペトリネットを使いペトリネットの処理を行う。
シミュレーション(数値計算)を行うことが出来る。
などの利点があります。
まだまだ、CellPetri net の今後の動向が注目されると考えられます。

References

1. Pauline N. Kawamoto and Yatsuka Nakamura, On Cell Petri Nets, Shinshu University.
2. 白田昭司, 井上祥史, 伊藤敏 Lotus1-2-3による理工系シミュレーション入門 1993
3. 平林雅英 C++版 Windows95プログラムを10倍強力に作る 共立出版 1996
4. 柏原正三 C++効率的な最速学習徹底入門 技術評論社 2002
5. 川端一生 真田良蔵 C/C++言語ハンドブック ナツメ社 1999.9.18
6. C/C++300の技 技術評論社 平成13.11.1
7. 森下信 セルオートマトン 養賢堂 2003.3.25
8. 加藤恭義・光成友孝・築山洋 セルオートマトン法 森北出版 1998.10.30
9. 白田昭司・東野勝治・井上祥史・伊藤敏・葭谷安正 カオスとフラクタル オーム社 1999.2.26