迷路を解くゲルオートマトン

山下 達也[1,a)]　萩谷 昌己[1]

概要：DNA 合成技術の進歩により化学反応系の人為的な設計が可能となり，その技術を応用した計算手法である DNA コンピューティングが活発に研究されている．我々は化学反応系による実装に適したセルオートマトンであるゲルオートマトンによって迷路問題を解く手法を提案する．提案される手法はある種の自己安定性を持っていることが示され，このことから迷路途中で変更されるような場合に柔軟に解を変更できることが導かれる．

キーワード：DNA コンピューティング，ゲルオートマトン，セルオートマトン，分散計算，迷路，自己安定性

1. 序論

近年，DNA 分子を用いた計算手法やロボット工学が活発に研究されている [1], [3], [10]．そのような DNA コンピューティングの一つの可能性としてゲルオートマトンという手法が提案されている [6], [13]．ゲルオートマトンとは，DNA 分子等を用いて人工的に設計された化学反応拡散系による実装を想定したセルオートマトンのことである．セルオートマトンは単純な遷移規則であっても複雑な系となりうる [4], [11], [14] ため，実装可能な単純な化学反応系を用いた複雑な処理の数理モデルとして適していると考えられている．

ゲルオートマトンの実装方法として，多孔質のゲル状物質で仕切られた空間をセルとみなして必要なだけ配置し，その空間の中にうまく設計された化学溶液を注入する手法が想定されている．この化学反応系はセルを仕切る物質の中を拡散する分子群と，セルの中で留まり続ける分子群で構成される．前者はセルオートマトンにおけるセル同士の通信を実現するために利用されるものでここでは伝達子と呼ぶ．後者はセルオートマトンの内部状態に対応して安定状態をもっていて [12]，状態に応じた伝達子を増幅するように設計される．また，隣のセルから拡散した伝達子と反応して異なる安定状態へ遷移することでセルオートマトンの状態遷移を実現する．

セルオートマトンをこのようにして実装するために以下の性質を持つことが必要条件であると考えられる．

- 非同期 (asynchronous)
- 有無型 (Boolean-totalistic)
- 非迷彩型 (non-camouflage) *1

非同期セルオートマトンとは，セルの状態遷移が非同期に行われるセルオートマトンである．すなわち，非同期セルオートマトンで一ステップの

1　東京大学大学院情報理工学系研究科コンピュータ科学専攻
Department of Computer Science, Graduate School of Information Science and Technology, University of Tokyo
a)　t.yamashita@is.s.u-tokyo.ac.jp

*1　この概念はむしろ迷彩型と呼ぶべきであるという旨のご指摘を竹内先生より頂いた．その通りではあるのだが，先行研究で”non-camouflage” と呼んでしまった以上，ここでも「非迷彩型」と呼ぶことにする．

状態遷移が起きるとき，各々のセルは遷移規則にしたがって状態遷移を行うか，状態遷移をせずに元の状態のままでいるかのどちらかを行う．この性質は化学反応系で実装されたゲルオートマトンにおいては，全てのセルで一斉に状態遷移が起きるとは限らないことに対応している．

有無型セルオートマトンとは，総和型セルオートマトンの一種で，状態遷移先の決定に使うことのできる近傍の情報が一般よりも少ないセルオートマトンである．総和型セルオートマトンは近傍の情報から方向や相対的な位置関係を除き，どの状態のセルが幾つ存在するのかという情報を使って状態遷移を決定する．有無型セルオートマトンは使える情報がさらに少なく，近傍にどの状態のセルが存在してどの状態のセルは存在しないのかという情報のみを使う．ゲルオートマトンではセルにある伝達子が存在するかしないかで化学反応を変えることができても，その伝達子の発生源の方向や発生源の数で化学反応を変えることは困難であると考えられるためである．

非迷彩型セルオートマトンとは，状態 s のセルの状態遷移先が近傍に状態 s のセルがあるのかどうかに依存しないようなセルオートマトンである．ゲルオートマトンのセルが状態 s のときにはそのセルは s に対応した伝達子が増幅されているため，近傍から拡散した同じ伝達子の存在によって化学反応を変更できない．

本論文では以上の三条件を満たすセルオートマトンをゲルオートマトンと呼ぶ．なお，セルオートマトンの定義される空間は二次元正方格子にフォンノイマン近傍を備えたものを仮定する．これはセルオートマトンを研究する上でよく用いられる設定である．

有無型セルオートマトンの遷移規則は次のような形式で書き表される．

$$r\,(s_0 \wedge \ldots \wedge \neg t_0 \wedge \ldots) \to u$$
$$r'\,(s'_0 \wedge \ldots \wedge \neg t'_0 \wedge \ldots) \to u'$$
$$\vdots$$

一行目の規則は状態 r のセルの近傍に状態 $s_0, \ldots$ のセルがそれぞれ存在し，かつ状態 $t_0, \ldots$ のセルはどれも存在しないとき，そのセルは状態 u に遷移できることを表す．我々は非同期セルオートマトンを考えているので，複数の規則が適用可能な状況や，どの規則も適用できない状況を許容する．すなわち，複数の規則が適用可能なセルが存在するときはそのいずれかが非決定的に適用でき，どの規則も適用できないセルは状態遷移をしないことにする．この形式で遷移規則を表現したとき，非迷彩型セルオートマトンの遷移規則はどの規則においても各 $s_0, \ldots, t_0, \ldots$ と r が異なるように表される．

先行研究ではゲルオートマトンは計算万能であることが示されている [7], [17]．さらに，分散計算をゲルオートマトンで行う枠組みとしてポピュレーションプロトコルをシミュレーションする手法が提案されている [16]．ポピュレーションプロトコルとは互いに通信をする多くのエージェントで計算を行う典型的な分散システムの数理モデルである [2]．分散システムの分野では任意の状態から目的の状態へと遷移できる性質を自己安定性といい，耐障害性を得るための設計指針として研究が進んでいる [5], [9], [15]．問題を自己安定的に解くゲルオートマトンの研究は大規模な分子ロボット群の制御や障害に強い新たな材料物質の開発に繋がると期待される．本論文では自己安定的に迷路を解くゲルオートマトンを紹介する．なお，化学反応拡散系の数理モデルによる迷路を解くセルオートマトンのシミュレーションも行われている [8] が，このセルオートマトンは総和型でない点で本研究と異なる．

2. 迷路を解くゲルオートマトン

どのようなゲルオートマトンであれば迷路を自己安定的に解くことができるか探っていく．なお，諏訪で開催された夏のプログラミングシンポジウム2018において迷路を解くゲルオートマトンを発表したのちに，基本的なアイディアは同じだがより単純で無駄の少ないものを発見した．本論文では後から発見されたものを紹介する．

2.1 迷路とその解の表現

まず，迷路をゲルオートマトン上で表現するために空白 (Blank)，壁 (Wall)，始点 (Start)，終点 (Terminal) に対応して四状態 B, W, S, T を導入する．状態 S, T のセルがそれぞれ丁度一つだけあるコンフィギュレーションを仮定する．それら二つのセルをそれぞれ始点，終点と呼び，初期コンフィギュレーションで空白状態 B のセルを通路と呼ぶ．以後，説明のためにコンフィギュレーションの一部を図示する際には空白状態のセルを白い四角で，壁状態のセルを黒い四角で，他の状態のセルは対応する文字のついた四角で表す．

二つの相異なるセル a, b が状態の集合 A について連結であるとは，セルの列 $a = c_0, c_1, \ldots, c_L = b\,(L \in \mathbb{N}^+)$ が存在して，各 $i \in \{0, \ldots, L-1\}$ について c_i と c_{i+1} が隣接していて，間のセル $c_1, \ldots, c_{L-1}$ の状態が全て A に属することをいう．特に三状態 W, S, T を除く状態について連結であるときは単に連結であるという．二つのセル a, b が連結であるとき，そのことを示すセルの列 $(c_i)_{i \in \{0, \ldots, L\}}$ を a と b を繋ぐ経路と呼ぶ．経路 (c_i) が単純であるとは，$i \neq j$ ならば $c_i \neq c_j$ であることをいう．迷路問題の目標は始点と終点を繋ぐ単純な経路を得ることである．

経路をゲルオートマトンで表現する方法を考えよう．まず思いつく方法として，ある特別な状態 P のセルの連なりでパスを表現するものがある．このとき，始点から状態 P のセルだけを通って終点にたどり着けるようなコンフィギュレーションを迷路の解だと思うと，通路をすべて状態 P に遷移させるだけで解を得られることになってしまうが，これは単純でない経路も含んでしまい，迷路の解とは言い難い．

コンフィギュレーションが解を表す条件として，状態 P のセルの隣には S, T, P のいずれかの状態のセルが丁度二つだけあることを付け加えると，確かに迷路の解と言えるものになりそうである．しかし，ゲルオートマトンは有無型であるから，コンフィギュレーションが上記の条件を満たしているか判定するゲルオートマトンを作ることも難しくなってしまう．

そこで，n 個の状態 $P_i\,(i = 0, \ldots, n-1)$ を用いて $P_0, P_1, \ldots, P_{n-1}, P_0, \ldots$ のように添字の順に並んだセルの連なりで経路を表現することを考える．$n \geq 3$ を仮定すれば，経路の向きを表現可能となる．S から P_i を順番に辿ると必ず有限の長さで T へ到達することをもって迷路の解を定義すると，始点にいる人はローカルな情報だけを頼りに袋小路に迷い込むこともループした通路を回り続けることもなく終点へたどり着ける．また，状態 P_i のセルの近傍には P_i と異なる状態 P_{i-1}, P_{i+1} が存在することになるので非迷彩型のゲルオートマトンに適した表現と言える．以下では添字順の状態 P_i のセルの連なりを道と呼ぶ．

2.2 道の探索

始点の隣にある通路を状態 P_i に遷移させたいのだが，やみくもに遷移をさせると終点へ到達できない通路までも解に含めてしまいかねない．そこで，終点への到達可能性 (Reachability) を示す状態 R を導入する．状態 T か R に隣接している通路は状態 R に遷移をするわけである．この規則は次の二式で表される．

$$B\,(T) \to R \tag{1}$$

$$B\,(R) \to R \tag{2}$$

始点に隣接する通路が状態 R になったら，それはその通路を通って終点へ到達できることを示している．そこで，そのようなセルを状態 P_0 へと遷移させる規則を追加する．

$$R\,(S) \to P_0 \tag{3}$$

もちろん，状態 P_0 のセルの隣に状態 R のセルが存在すれば，それは状態 P_1 に遷移し，さらにその隣の状態 R のセルは状態 P_2 へと遷移していく．そのような規則は各 $i \in \mathbb{Z}/n\mathbb{Z}$ について

$$R\,(P_i) \to P_{i+1} \tag{4}$$

として表される．

さて，この規則で迷路を解けるだろうか．答えは否である．迷路を解く際に気をつけなけらばならないのは行き止まりとループであるが，今のま

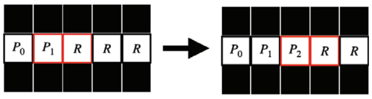

図 1　ヘッドの進展

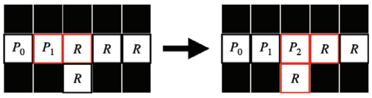

図 2　ヘッドの分岐

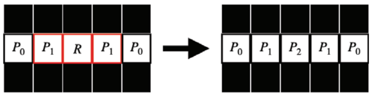

図 3　ヘッドの整合的な衝突

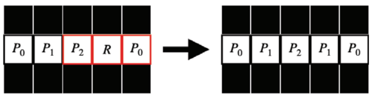

図 4　ヘッドの整合的な衝突の別例

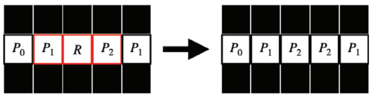

図 5　ヘッドの整合的でない衝突

まではどちらにも対処できない．

このことを議論するためにいくつかの用語を定める．まず，状態 R のセルと，そのセルに接する状態 P のセルの組をヘッドと呼ぶ．つまり，ヘッドとは規則 4 の適用ができることを保証するセルのペアである．図 1 のように，あるヘッドの R 状態のセルの近傍にまた R 状態のセルが存在するとき，規則 4 によって状態遷移するとヘッドが近傍に移動したように見える．このような状態遷移をヘッドの進展と呼ぶ．ヘッドが進展した軌跡に残る道をヘッドが生成した道と呼ぶ．図 2 のように，ヘッドが進展した先のセルが複数の状態 R のセルと隣接しているとき状態遷移によりヘッドは増加する．このような状態遷移をヘッドの分岐と呼ぶ．また，図 3，4，5 のように，複数のヘッドが状態 R のセルを共有しているとき，そのセルが状態遷移をするとヘッドは減少する．このような状態遷移をヘッドの衝突と呼ぶ．特に図 3 のように衝突後の道の添字が P_i, P_{i+1}, P_i のようになるときその衝突は整合的であるといい，特に状態 P_{i+1} のセル（図では状態 P_2 のセル）を衝突地点と呼ぶ．整合的な衝突は図 3 で表される状態遷移の他に，図 4 で表されるような状態遷移でも発生することに注意してほしい．ヘッドの衝突が起きた後のコンフィギュレーションで P_i, P_{i+1}, P_i の状態の三つのセルが並んでいるときのことを整合的と呼ぶのである．

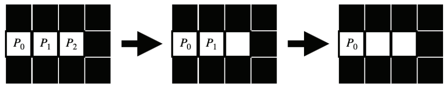

図 6　道の縮退

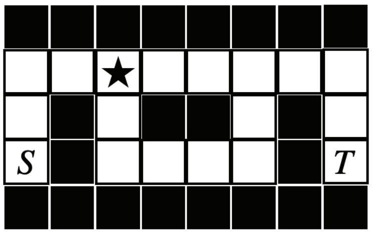

図 7　ループ状の通路の例

2.3　行き止りの解消

上記の規則 4 だけでは行き止まりであっても，始点と終点から到達可能であれば状態 P_i へと遷移して安定してしまう．幸い，状態 P_i のセルは自身が行き先のある通路の中にいるのか，それとも行き止まりにいるのかを，近傍に行き先 P_{i+1} または R または T が存在するか否かで判定できる．そこで次の規則を設けることで，行き止まりに到達した経路が消えるようにする．

$$P_i\,(\neg T \land \neg R \land \neg P_{i+1}) \rightarrow B \qquad (5)$$

図 6 で表されるように，この規則の適用は規則 4 で作られた道を縮退させる．

2.4　ループの解消

次にループについて考察する．図 7 のような始点にも終点にも到達可能なループ状の通路を考える．このようなループでは，ループ上の他のセルを通らずに始点（終点）に到達できるセルが存在する．そのセルをループの入口（出口）と呼ぶこと

にする．図の $\star$ で印づけられたセルは入口である．

このループを構成する全てのセルが規則 1，2 により状態 R へと遷移した後，入り口にヘッドが到達すると，ヘッドは分岐をする．二つのヘッドはそれぞれの経路を進展しやがて衝突する．衝突は整合的な場合とそうでない場合がありえる．

整合的でない場合には，規則 5 により二つのヘッドに生成された道は最も近い入口または出口まで縮退し，ループの入口と出口を繋ぐ道だけが残る．

入り口で分岐した二つのヘッドの衝突がたまたま整合的だったときには衝突地点が入口だった場合，衝突地点が出口だった場合，いずれでもない場合でさらに場合分けが必要になる．

衝突地点が入口でも出口でもなかった場合には，衝突が整合的であっても衝突地点で規則 5 が適用できて，整合的でない場合と同じように道が縮退して入り口と出口を繋ぐ道だけが残る．

衝突地点が出口だった場合は，二つのヘッドが生成した道はどちらも入り口と出口を繋ぐ道となっている．これらの道は縮退せずに残ったままで迷路の解の一部をなすため特に問題ではない．

衝突地点が入口だった場合というのは，入口で分岐した二つのヘッドのうち片方だけが進展しループを一周することで生じる図 8 で表される状況である．このとき，入口の状態を P_{i+2} とすると，入口の近傍には少なくとも状態 P_{i+1} のセルが二つと状態 P_{i+3} のセルが一つ存在することになる．出口で衝突した場合と異なり，この場合では道を辿ってループを回り続けることができてしまうので，ここからヘッドが進展して得られるコンフィギュレーションは解とはならない．しかも，ゲルオートマトンは有無型であったから，入口の近傍は正しい経路上のセルの近傍と区別されず，一度このようなコンフィギュレーションになるとそこから縮退によって正しい解を得るのは困難である．

そこで，入口での整合的な衝突を回避するための工夫が必要となる．入口で衝突する直前のコンフィギュレーションを考えると，図 8 の左側のようになっているはずである．状態 R のセルが状態 P_{i+1} に遷移すると入口で整合的に衝突してしまう．ところで，このセルは状態 P_{i+1} の他に状態 P_{i+3}

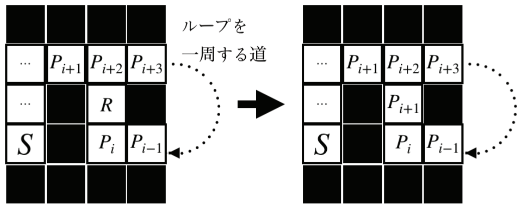

図 8 ループの入口での整合的な衝突

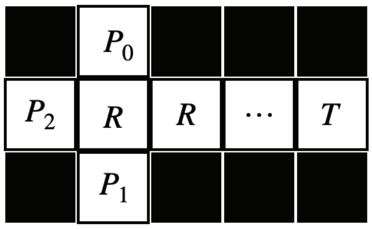

図 9 デッドロックを引き起こす状況（$n = 3$ の場合）

にも遷移することが可能である．こちらの遷移を引き起こせば衝突は整合的ではないので既に確かめたようにヘッドが縮退をして解となる経路だけが残る．図 8 左側で状態 R のセルが状態 P_{i+1} には遷移できないように，遷移規則 4 を以下の規則 6 に変更する．

$$R\,(P_i \wedge \neg P_{i+2}) \rightarrow P_{i+1} \tag{6}$$

この遷移規則によって印をつけたセルはいずれ状態 P_{i+3} へ遷移して整合的でない衝突を引き起こした後，必要なだけ縮退する．

規則 4 を規則 6 に変更したのちも以上の議論は概ね成り立つ．問題が起きるとすれば進展するべきヘッドの先に別のヘッドが複数あってデッドロックを引き起こす図 9 のような場合だけである．複数のヘッドが状態 R のセル c を共有していて，c が状態 R について終点と連結とする．仮に，ヘッドの一つが状態 P_0 であるとすると，このヘッドが 6 によって進展できなくなるのはセル c の近傍に状態 P_2 のセルが存在するときである．同様に，状態 P_2 のヘッドはセル c の近傍に状態 P_4 のセルが存在するときに限り進展できない．例えば図 9 のように $n = 3$ とすると，$P_{2+2} = P_1$，$P_{2+2+2} = P_0$ であるため，状態 P_0，P_1，P_2 の三つのヘッドが状態 R のセルを共有するとどのヘッドも進展できないデッドロックに陥る．一般に，状

態 $P_i, P_{i+2}, \ldots, P_{i+2(k-1)}$ の k 個のヘッドが状態 R のセルを共有してデッドロックになるのはこれらのヘッドのうちの一つが状態 P_{i+2k} であるときである．すなわち，$\mathbb{Z}/n\mathbb{Z}$ を加法についての群とみなしたときに，$2 \in \mathbb{Z}/n\mathbb{Z}$ の位数がセルの近傍の数より以下のときである．つまり，我々はゲルオートマトンをセルの近傍の数が 4 の二次元正方格子の上で定義していることから，たとえば $n = 5$ とすればこのようなデッドロックは回避することができる．

2.5 安定性

ゲルオートマトンが自己安定的に迷路を解くことを目指していたのだった．分散システムが自己安定とはフェアな遷移を仮定すれば任意の初期状態から目的とする状態まで有限ステップで遷移して，目的の状態で安定することをいう．今回提案するゲルオートマトンを本当の意味で自己安定にすることはできていないが，許す初期状態を制限した弱い自己安定性を得ることができることを示す．また，その帰結としてある迷路を解いている最中のゲルオートマトンの中に任意に壁や通路を追加する操作を加えると操作後の迷路に対して正しい解を求められることを示す．

その前に，安定性を得るための準備として，次に示す新たな遷移規則をゲルオートマトンに追加する．

$$P_i\,(\neg S \land \neg P_{i-1}) \to B \tag{7}$$

始点から延びた道が，外部から壁を挿入されるなどの要因によって途中で切れたときに，始点と繋がらない道はこの規則 7 によって消滅する．道が消える遷移は規則 5 によって道が縮退する遷移とよく似ている．この規則は始点と繋がっている道を消すことはなく，また直接には新たな道が作られることを阻害も促進もしないので規則の追加によって迷路が解けなくなることもない．

ゲルオートマトンの初期コンフィギュレーションを次の条件 $(*)$ を満たすものに制限する．

- 始点と終点がちょうど一つずつ存在する．
- 有限個のセルをのぞいて全て壁状態である．
- 巡回する道が存在するコンフィギュレーションへの状態遷移列が存在しない．

一つ目の条件は考えている問題から当然に要求される．二つ目の条件は迷路が無限に大きいことはなく，有界であることを表す．三つ目の条件は迷路の要件ではなく，このゲルオートマトンに特有の条件となっている．構成したゲルオートマトンでは巡回した道のあるコンフィギュレーションからの状態遷移では巡回した道が保存されてしまうので，このような条件が必要となる．一方，前節 §2.4 の議論より，始点と終点と有限個の空白状態のセルの他はすべて壁状態となっているようなコンフィギュレーションは $(*)$ を満たすので，迷路を表現する能力は保たれる．

目的とするコンフィギュレーションは次の条件 $(\star)$ を満たすものである．

- 始点と終点が連結ならば始点から終点への道が存在する．
- 始点からの任意の道は有限の長さで終点に繋がる．
- 始点に連結なセルは状態 R ではない．

最初の二つの条件はすでに議論した迷路の解を表現している．三つ目の条件は，目的とするコンフィギュレーションに達したあとで新しい道が生成されることを防ぐ．始点に連結な状態 R のセルが存在しなければ新しい道は生成されないため，条件 $(\star)$ を満たしたコンフィギュレーションが状態遷移して得られるコンフィギュレーションはやはり $(\star)$ の二つ目の条件を満たす．一つ目のと三つ目の条件はこのゲルオートマトンの状態遷移で保存されるので，ゲルオートマトンが目的の状態で安定することがわかる．

以下では条件 $(*)$ を満たす任意のコンフィギュレーションは任意のフェアな遷移列によって条件 $(\star)$ を満たすコンフィギュレーションに到達することを示す．

まず，今回のゲルオートマトンの構成から導かれる補題として，セル c の状態が P_i のとき，フェアな状態遷移列で c が状態 P_i のままであるための条件は，状態遷移の中で c を含む始点から終点への道が生成されるか，c を含む巡回する道が生成さ

れることであることに注意する．

始点と終点が連結ならば始点から終点への道ができることを示す．仮定より始点と終点を繋ぐ経路が存在する．フェアな状態遷移でこの経路が道とならないのは，始点からこの経路を辿るヘッドの生成する道以外の道 p がこの経路を通っていて，しかも p が保たれているときのみである．このとき，道の保たれる条件とコンフィギュレーションの条件 $(*)$ より，p は始点から終点への道である．

次に始点からの任意の道は有限の長さで終点に繋がることを見る．これはコンフィギュレーションの制約によりループができないことと，フェアな状態遷移により道が保たれる条件からわかる．

さらに，始点から連結なセルで状態 R のものが存在すると，始点から連結であるためにやがて道を表す状態 P のいずれかに状態遷移してその後また状態 R に戻ることはないので，三つ目の条件も満たされる．

目的のコンフィギュレーションは安定であることもゲルオートマトンの構成から従う．

以上より，このゲルオートマトンがある種の自己安定性を持つことがわかる．特にこの系として，迷路を解いている途中や解き終わった後に外部から強制的に壁を追加または削除することで迷路の変更をしたときに，変更後の迷路の解を自動的に求め直すことがわかる．上述の制限を満たすコンフィギュレーションはこの操作を施したあとも制限を満たすためである．

3. 結論

本論文では迷路を解くゲルオートマトンの安定性について議論し．外的な要因による迷路の変更に強いシステムとなっていることを確認した．この成果は化学反応を利用して環境の変化に対応するシステムを作る目的に近づいたと言える．

一方，今回の研究では自己安定性そのものは得られず，似た形式の弱い性質を示すに止まった．すなわち，任意のコンフィギュレーションを許すのではなく，一定の制約を満たしたコンフィギュレーションの中での安定性を示した．これはゲルオートマトンの内部の障害には対応できていないために本来の意味での自己安定性を得られないためである．例えば，規則 6 は図 8 で整合的な衝突が起きることがないように規則 4 を改良して得られた規則だったが，この規則が障害によりうまく働かずに図の ◎ 印のセルが状態 P_{i+1} に遷移してしまうと，巡回する道ができてしまい，ゲルオートマトンがこの道を消す手段はなくなる．この種の障害にも対応するためには始点から終点まで道沿いにトークンを流すなどして，正常に解を求められているか確認する必要があるが，どのようにすればゲルオートマトンで解の確認をできるのかは未解決である．

また，自己安定的に解けることが知られている典型的な分散システムの問題としてリーダー選挙問題や全域木探索問題がある．これらの問題がゲルオートマトンにおいても自己安定的に解けるのかという問題も興味深い．特にリーダー選挙問題が解けると，今回提案した迷路を解くゲルオートマトンにおいて連結な道の中でリーダーを適当に決め，リーダーの出したシグナルが道を伝ってリーダーのところまで戻るか調べることで巡回する道の検出ができると考えられる．

4. 質疑

夏のプログラミングシンポジウムでの発表では多くの質問があった．当日の発表では迷路を解くゲルオートマトンの他にもいくつかのゲルオートマトンを紹介したため，そちらについての質問もあったがここでは省略する．

4.1 道を表現する状態の数 n が 5 である理由

道を表現する状態は $n = 3$ で良さそうに見えるがなぜ $n = 5$ なのかという質問が複数の方からあったが，当日うまく答えることができなかった．本論文中の §2.4 最終段落を参照してもらいたい．関連して，セルの空間として正方格子ではなく六角格子を用いた場合にはどうなるかという質問に対してほぼ同じゲルオートマトンで迷路を解けるが，n が異なるという趣旨の返答をした．この場合は §2.4 最終段落の議論でセルの近傍の数が 4 ではなく 6 となるので，例えば $n = 7$ とすれば良い．

また，そもそも道を表現する為に向きをつける必要はないのではないかという質問もあったが，向きをつけない道の表現では通路を全部道とみなしてしまうことを防げないという趣旨の返答をした．本論文の §2.1 で議論したように，ゲルオートマトンでの道の表現を単一の状態で行うのは困難である．

4.2 シャーレの中で動くゲルオートマトンについて

シャーレの中で実際に動くものではないことを残念がる声もあった．このことについては，今後のゲルオートマトン研究の発展に期待してもらいたい．

4.3 セル空間の拡張について

二次元ではなく，三次元のセル空間をもつゲルオートマトンや，さらに拡張して一般のグラフ状のセル空間を持つゲルオートマトンについての質問があった．理論上は三次元に拡張しても議論でき，また平面グラフの範疇に止まらない迷路問題などを考えることができるようになると返答した．一般のグラフのセル空間についてはゲルのボールによる実装が近いのではないかと返答した．

4.4 量子計算との関係について

分子を用いた計算手法であることから，量子計算と関係があるのかとの質問があった．この質問には特に関係はないと返答した．現在のゲルオートマトンの研究は量子力学的な効果が観測できないようなマクロで古典的な系による実装が想定されている．

参考文献

[1] Adleman, L. M.: Molecular computation of solutions to combinatorial problems, *Nature*, Vol. 369, p. 40 (1994).

[2] Aspnes, J. and Ruppert, E.: An introduction to population protocols, *Middleware for Network Eccentric and Mobile Applications*, Springer, pp. 97–120 (2009).

[3] Chen, H.-L., Doty, D. and Soloveichik, D.: Deterministic function computation with chemical reaction networks (2014).

[4] Cook, M.: Universality in elementary cellular automata, *Complex systems*, Vol. 15, No. 1, pp. 1–40 (2004).

[5] Cooper, C., Lamani, A., Viglietta, G., Yamashita, M. and Yamauchi, Y.: Constructing Self-stabilizing Oscillators in Population Protocols, pp. 187–200 (2015).

[6] Hagiya, M., Wang, S., Kawamata, I., Murata, S., Isokawa, T., Peper, F. and Imai, K.: On DNA-based gellular automata, *International Conference on Unconventional Computation and Natural Computation*, Springer, pp. 177–189 (2014).

[7] Isokawa, T., Peper, F., Kawamata, I., Matsui, N., Murata, S. and Hagiya, M.: Universal Totalistic Asynchonous Cellular Automaton and Its Possible Implementation by DNA, *15th International Conference on Unconventional Computation and Natural Computation: (UCNC 2016)*, pp. 182–195 (2016).

[8] Kawamata, I., Yoshizawa, S., Takabatake, F., Sugawara, K. and Murata, S.: Discrete DNA reaction-diffusion model for implementing simple cellular automaton, *International Conference on Unconventional Computation and Natural Computation*, Springer, pp. 168–181 (2016).

[9] Lamani, A. and Yamashita, M.: Realization of Periodic Functions by Self-stabilizing Population Protocols with Synchronous Handshakes, pp. 21–33 (2016).

[10] Murata, S., Konagaya, A., Kobayashi, S., Saito, H. and Hagiya, M.: Molecular robotics: A new paradigm for artifacts, *New Generation Computing*, Vol. 31, No. 1, pp. 27–45 (2013).

[11] Neumann, J. V.: *The Theory of Self-Reproducting Automata*, University of Illinois Press, Urbana (1966).

[12] Padirac, A., Fujii, T. and Rondelez, Y.: Bottom-up construction of in vitro switchable memories, *Proceedings of the National Academy of Sciences*, Vol. 109, No. 47, pp. E3212–E3220 (2012).

[13] Wang, S., Imai, K. and Hagiya, M.: On the composition of signals in gellular automata, *Computing and Networking (CANDAR), 2014 Second International Symposium on*, IEEE, pp. 499–502 (2014).

[14] Wolfram, S.: *A new kind of science*, Wolfram media Champaign (2002).

[15] Xu, X., Yamauchi, Y., Kijima, S. and Yamashita, M.: Space Complexity of Self-Stabilizing Leader Election in Population Protocol Based on k-interaction (2013).

[16] Yamashita, T. and Hagiya, M.: Simulating

Population Protocols by Gellular Automata, *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, IEEE, pp. 1579–1585 (2018).

[17] Yamashita, T., Isokawa, T., Peper, F., Kawamata, I. and Hagiya, M.: Turing-Completeness of Asynchronous Non-camouflage Cellular Automata, *International Workshop on Cellular Automata and Discrete Complex Systems*, Springer, pp. 187–199 (2017).