

博士學位論文

セルオートマトンとその応用に関する研究

**Study of Cellular Automata and its
Applications**

山下 和也

平成 30 年 3 月

概要

セルオートマトン^{*1}(CA: cellular automaton) とは, セルとよばれる同一の有限オートマトンを格子状に並べた空間に対して単純な規則で状態遷移を繰り返す離散的計算モデルである. セルオートマトンを用いた計算モデルは, 経済現象, 物理現象, 社会現象などの幅広い分野に適用されている. これまでにセルオートマトンに関する問題が数多く提起されているが, 本論文は, 不完全な同期下のセルオートマトンの時空間パターンの分類, セルオートマトンを用いた雪の結晶の類似パターン生成, セルオートマトン上で定義される一斉射撃問題の拡張について述べる.

まず, 不完全な同期下のセルオートマトンの時空間パターンの分類について述べる. Wolfram は 2 状態 3 近傍 1 次元セルオートマトンの時空間パターンが 4 つのクラスに分類できることを示した. 通常のセルオートマトンは全てのセルが同期して遷移するが, Fatès 等は, セル列の一部が同期して遷移するような不完全な同期下のセルオートマトンを提起した. しかし, その時空間パターンの分類は明らかになっていない. 本論文では, Fatès 等が定義した不完全な同期下のセルオートマトンの時空間パターンと, このモデルの両極にある全てのセルが同期して遷移する通常のセルオートマトンと完全にランダムな非同期セルオートマトンの時空間パターンを比較することによって, 不完全な同期下の単純セルオートマトンを 5 つのグループに分類する. また, 不完全な同期下のセルオートマトンにおける遷移関数と時空間パターンの特徴との関係を明らかにするとともに, 非同期セルオートマトンとの時空間パターンの相違が, 不完全な同期下のセルオートマトンにおける局所的な同期によって引き起こされることを明らかにする.

^{*1} 正確な発音に近いセルラオートマトンとよばれることもある.

次に、セルオートマトンを用いた雪の結晶の類似パターン生成について述べる。Packard は、六角格子上の 2 状態 7 近傍 2 次元セルオートマトンを用いた簡単なモデルで、角板とよばれている結晶の類似パターンを生成した。その後、轡田等は Packard のモデルを拡張した 3 状態 7 近傍 2 次元セルオートマトンを用いて、Packard のモデルではできなかった樹枝状六花、広幅六花、扇状六花等の類似パターンを生成した。しかし、いずれのモデルでも、成長過程で環境が大きく変化してできる複雑な形をした結晶や、大気中の塵等が結晶に付着して空気抵抗が微妙に変わり結晶の各辺の成長速度が変化してできる六方非対称な結晶の類似パターンは生成できない。本論文では、轡田等のモデルを拡張したモデルを提案する。具体的には、シミュレーション途中で遷移関数を入れ替える点と遷移規則から六方対称性を放棄する点の 2 点を拡張したモデルを提案する。提案モデルを用いて、成長過程で環境が大きく変化してできる複雑な形をした結晶や、六方非対称な結晶等の類似パターンを生成する。

最後に、一斉射撃問題の拡張について述べる。一斉射撃問題は、セルオートマトンに関する有名な問題の 1 つで、1957 年頃に Myhill によって提起され、Moore によって広く知らされた問題である。一斉射撃問題を拡張したさまざまな問題が知られているが、本論文では、7 セグメントディスプレイ上の数字パターンの一斉射撃問題、16 セグメントディスプレイ上の英字パターンの一斉射撃問題、副将軍のいる一斉射撃問題を提起する。まず、7 セグメントディスプレイを 2 次元セルオートマトン上でモデル化し、7 セグメントディスプレイで表現される数字パターンの一斉射撃問題を提起する。7 セグメントディスプレイ上の数字パターンの一斉射撃問題のうちいくつかの問題がセグメント木とよばれる特殊な木の一斉射撃問題に一般化できることを示す。さらにセグメント木の一斉射撃問題が、セグメント列とよばれる等間隔に区切られた特殊な 1 次元セル列の一斉射撃問題へ帰着できることを示す。セグメント列の一斉射撃問題とセグメント列に帰着できない数字パターンの一斉射撃問題に対してそれぞれアルゴリズムを与える。これらの問題に対する最小時間を示し、全てのセル列に対して最小時間で射撃させるような最小時間解は存在しないことを示す。次に、16 セグメントディスプレイを 2 次元セルオートマトン上でモデル化し、16 セグメントディスプレイで表現される英字パターンの一斉射撃問題を提起する。16 セ

グメントディスプレイ上の英字パターンの一斉射撃問題のうちいくつかの問題が 1-2 セグメント木の一斉射撃問題に帰着できることを示し，一斉射撃アルゴリズムを与える．最後に，1次元セルオートマトン上で定義される従来の一斉射撃問題に副将軍とよばれる新しい状態を導入し，副将軍のいる一斉射撃問題を提起する．特に副将軍が1つある場合について，副将軍の位置に注目して解を与える．副将軍が1つある場合の最小時間を示し，全てのセル列について最小時間で射撃させるような最小時間解は存在しないことを示す．ただし，本論文で提案した解が最小時間解に限りなく近い解であることを示す．

目次

概要	i
第 1 章 序論	1
参考文献	3
第 2 章 セルオートマトン	6
2.1 1次元セルオートマトン	7
2.2 2次元セルオートマトン	9
参考文献	12
第 3 章 不完全な同期下の単純セルオートマトンの時空間パターンによる分類	14
3.1 不完全な同期下のセルオートマトン	15
3.2 不完全な同期下の単純セルオートマトンの分類	16
3.2.1 同期セルオートマトン, 非同期セルオートマトンと同様なパター ンに収束する	18
3.2.2 p の値によって徐々にパターンが変化する	20
3.2.3 非同期セルオートマトンと同様なパターンに収束する	21
3.2.4 同期セルオートマトンと同様なパターンに収束する	23
3.2.5 同期セルオートマトン, 非同期セルオートマトンとも異なるパ ターンに収束する	24

3.3	不完全な同期下のセルオートマトンにおける遷移関数と時空間パターンの特徴との関係	25
3.4	非同期セルオートマトンの時空間パターンと差異が生じた要因	29
3.5	不完全な同期下のセルオートマトンでクラスタが死滅するセルオートマトンの遷移関数に関する考察	31
3.6	Wolfram の分類との比較	34
3.7	まとめ	36
	参考文献	37
第 4 章	雪の結晶の類似パターン生成	39
4.1	雪の結晶成長	40
4.2	セルオートマトンを用いたモデル	41
4.3	環境の変化によって生じる複雑な形をした雪の結晶	43
4.4	環境の変化によって生じる複雑な形をした雪の結晶の類似パターンの生成	44
4.5	六方非対称な雪の結晶	47
4.6	六方非対称な雪の結晶の類似パターン生成	48
4.7	まとめ	52
	参考文献	52
第 5 章	一斉射撃問題の拡張	54
5.1	一斉射撃問題の定義	55
5.2	最小時間アルゴリズム	56
5.3	7 セグメントディスプレイ上の数字パターンとセグメント列の一斉射撃問題	58
5.3.1	7 セグメントディスプレイ上の数字パターン	58
5.3.2	7 セグメントディスプレイのモデル化	59
5.3.3	数字パターンの一斉射撃問題	59
5.3.4	数字パターンの一斉射撃アルゴリズム	61

5.3.5	数字パターン 2, 3, 4, 5, 6, 8, 9 の一斉射撃アルゴリズム	61
5.3.6	セグメント列の一斉射撃問題	63
5.3.7	セグメント列の一斉射撃アルゴリズム	64
5.3.8	セグメント列の一斉射撃問題の最小時間	66
5.3.9	セグメント列の一斉射撃問題の最小時間解	67
5.3.10	数字パターン 7 の一斉射撃アルゴリズム	68
5.3.11	数字パターン 7 の一斉射撃問題の最小時間	70
5.3.12	数字パターン 7 の一斉射撃問題の最小時間解	70
5.3.13	数字パターン 0 の一斉射撃アルゴリズム	70
5.3.14	数字パターン 0 の一斉射撃問題の最小時間	72
5.3.15	数字パターン 0 の一斉射撃問題の最小時間解	73
5.3.16	数字パターン 1 の最小時間アルゴリズム	73
5.4	16 セグメントディスプレイ上の英字パターンと 1-2 セグメント木の一斉射撃問題	73
5.4.1	16 セグメントディスプレイ上の英字パターン	73
5.4.2	16 セグメントディスプレイのモデル化	74
5.4.3	英字パターンの一斉射撃問題	75
5.4.4	英字パターンの一斉射撃アルゴリズム	76
5.4.5	セグメント木に帰着できる英字パターンの一斉射撃アルゴリズム	77
5.4.6	1-2 セグメント木の一斉射撃問題	80
5.4.7	1-2 セグメント木の一斉射撃アルゴリズム	82
5.4.8	英字パターン R の一斉射撃アルゴリズム	87
5.4.9	英字パターン I の一斉射撃アルゴリズム	88
5.5	副将軍のいる一斉射撃問題	88
5.5.1	一斉射撃アルゴリズム	89
	単純なアルゴリズム	89
	高速なアルゴリズム	90

5.5.2	最小時間	95
5.5.3	最小時間解	96
5.6	まとめ	97
	参考文献	99
第 6 章	結論	101
付録 A	単純セルオートマトンの時空間パターン	107
A.1	グループ 1: 同期 CA, 非同期 CA と同様なパターンに収束する	107
A.2	グループ 2: 遷移確率 p の値によって徐々にパターンが変化する	117
A.3	グループ 3: 非同期 CA と同様なパターンに収束する	129
A.4	グループ 4: 同期 CA と同様なパターンに収束する	136
A.5	グループ 5: 同期 CA, 非同期 CA と異なるパターンに収束する	137
謝辞		139

第 1 章

序論

計算機の発展に伴い，さまざまな自然現象や社会現象のメカニズムが計算機実験により解明されている。しかし，脳の機能，地球規模の気候変動，経済活動などのように非常に複雑な振る舞いをする現象が数多く存在し，そのメカニズムは未だ明らかにされていない。これらの現象のように，相互作用する多くの要素で構成され，局所的な相互作用によって大域的に複雑な振る舞いをする系を複雑系とよぶ。複雑系のモデルは，経済学，物理学，生物学，化学，計算機科学などの多岐にわたる分野で研究されている。

複雑系の計算モデルの一つにセルオートマトンがある。セルオートマトンとは，セルとよばれる同一の有限オートマトンを格子上に並べた空間に対して，隣接するセルの状態を入力とした単純な規則で状態遷移を繰り返す離散的計算モデルである。隣接するセルの局所的な相互作用によって，複雑系に分類される現象を再現することができる。セルオートマトン研究の初期は，細胞のように自己複製する系に関する研究が主な問題であったが，1970年に Conway が 2 状態 9 近傍 2 次元セルオートマトン上でライフゲームを提案すると注目をあびるようになった [1]。セルオートマトンの応用範囲は幅広く，暗号技術 [2]，免疫システム [3,4]，交通流シミュレーション [5–9]，海洋汚染シミュレーション [10]，人の移動シミュレーション [11]，避難流動のシミュレーション [12–14]，人口分布予測 [15]，都市開発 [16] など，さまざまな分野に適用されている。

セルオートマトンに関する問題は数多く存在するが，本論文では，不完全な同期下のセ

ルオートマトンの時空間パターンの分類，セルオートマトンを用いた雪の結晶の類似パターン生成，セルオートマトン上で定義される一斉射撃問題の拡張の 3 つの問題に対して一定の解を与えることを目的とする。

本論文は，本章を含めて 6 章から構成される。

第 2 章では，本論文のテーマであるセルオートマトンの歴史とその定義について述べる．特に本論文で用いる 1 次元セルオートマトンと 2 次元セルオートマトンについて述べ，セルオートマトンが局所的な相互作用によって複雑な振る舞いすることを示す。

第 3 章では，Fatès 等が定義した不完全な同期下のセルオートマトン [17] の時空間パターンの分類について述べる．不完全な同期下のセルオートマトンの時空間パターンの分類はこれまで研究されていなかった．本論文では，完全に同期した通常のセルオートマトン [18–20] と金田の定義した完全に非同期なセルオートマトン [21] の時空間パターンと，不完全な同期下のセルオートマトンの時空間パターンを比較することによって，不完全な同期下の単純セルオートマトンを 5 つのグループに分類する．また，不完全な同期下のセルオートマトンにおける遷移関数と時空間パターンの特徴との関係を明らかにするとともに，非同期セルオートマトンとの時空間パターンの相違が，不完全な同期下のセルオートマトンにおける局所的な同期によって引き起こされることを明らかにする。

第 4 章では，雪の結晶の類似パターン生成について述べる．轡田等が定義した六角格子上の 3 状態 7 近傍 2 次元セルオートマトン [22] を拡張したモデルを提案し，提案モデルを用いて類似パターンを生成する．まず，環境の大きな変化を遷移関数の入れ換えに対応させたモデルを提案し，成長過程で環境が大きく変化してできる複雑な形をした結晶の類似パターンを生成する．次に，遷移規則から六方対称性を放棄したモデルを提案し，大気中の塵等が結晶に付着して空気抵抗が微妙に変わり結晶の各辺の成長速度が変化してできる六方非対称な結晶のような複雑な形をした雪の結晶の類似パターンを生成する。

第 5 章では，セルオートマトンに関する有名な問題の 1 つである一斉射撃問題 [23–29] を拡張し，7 セグメントディスプレイ上の数字パターンの一斉射撃問題，16 セグメントディスプレイ上の英字パターンの一斉射撃問題，副将軍のいる一斉射撃問題を提起し，これらの問題に対し解を与え，その一斉射撃時間について述べる．まず，7 セグメントディ

スプレイ上の数字パターンの一斉射撃問題を提起し、これらの問題のいくつかがセグメント木とよばれる特殊な木の一斉射撃問題に一般化できることを示す。セグメント木の一斉射撃問題が、セグメント列とよばれる等間隔に区切られた特殊な 1 次元セル列の一斉射撃問題へ帰着できることを示し、セグメント列の一斉射撃問題とセグメント列に帰着できない数字パターンの一斉射撃問題に対してそれぞれアルゴリズムを与える。これらの問題に対する最小時間を示し、全てのセル列に対して最小時間で射撃させるような最小時間解は存在しないことを示す。次に、16 セグメントディスプレイ上の英字パターンの一斉射撃問題のうちいくつかの問題が 1-2 セグメント木の一斉射撃問題に帰着できることを示し、一斉射撃アルゴリズムを与える。さらに、従来の一斉射撃問題に副将軍とよばれる新しい状態を導入し、副将軍のいる一斉射撃問題を提起し、特に副将軍が 1 つある場合の一斉射撃問題の最小時間を示し、最小時間解に限りなく近い解を示す。全てのセル列について最小時間で射撃させるような最小時間解は存在しないことを示す。

第 6 章では、不完全な同期下のセルオートマトンの時空間パターンの分類、セルオートマトンを用いた雪の結晶の類似パターン生成、セルオートマトン上で定義される一斉射撃問題の拡張の各問題に対して得られた結果を総括する。

参考文献

- [1] Conway, J.: The game of life, *Scientific American*, Vol. 223, No. 4, p. 4 (1970).
- [2] Chaudhuri, P. P.: *Additive cellular automata: theory and applications*, Vol. 1, John Wiley & Sons (1997).
- [3] Chowdhury, D. and Stauffer, D.: Systematics of the models of immune response and autoimmune disease, *Journal of Statistical Physics*, Vol. 59, No. 3, pp. 1019–1042 (1990).
- [4] Pandey, R. B. and Stauffer, D.: Metastability with probabilistic cellular automata in an HIV infection, *Journal of statistical physics*, Vol. 61, No. 1, pp. 235–240 (1990).
- [5] Resnick, M.: Turtles, termites, and traffic jams: explorations in massively pow-

- erful microworlds (1995).
- [6] Nagel, K. and Barrett, C. L.: Using microsimulation feedback for trip adaptation for realistic traffic in Dallas, *International Journal of Modern Physics C*, Vol. 8, No. 03, pp. 505–525 (1997).
- [7] Howard, K. R.: Unjamming traffic with computers, *Scientific American*, Vol. 277, No. 4, pp. 86–92 (1997).
- [8] 山本直史, 森下信: セルラオートマトンによる交通流のモデル化とその制御, 日本機械学会論文集 C 編, Vol. 65, No. 637, pp. 3553–3558 (1999).
- [9] 玉城龍洋, 安江里佳, 北栄輔 (英輔): セル・オートマトンによる自動車専用道路の交通シミュレーション, 情報処理学会論文誌: 数理モデル化と応用, Vol. 46, No. SIG 10 (TOM 12), pp. 30–40 (2005).
- [10] 森下信, 長谷川純一, 中野孝昭, 築山洋: セルラオートマトン法による拡散現象の解析, 日本造船学会論文集, Vol. 1998, No. 183, pp. 181–187 (1998).
- [11] 大高善光: セルラオートマトン法による店舗内購買シミュレーション, 日本機械学会第 75 回通常総会講演会講演論文集, 1998. 3, pp. 411–412 (1998).
- [12] 松田泰治, 大塚久哲, 樗木武, 内田広明: セルオートマトン法を用いた地下街の避難行動シミュレーションに関する一考察, 地域安全学会論文集, No. 2, pp. 95–100 (2000).
- [13] 大鑄史男, 小野木基裕: セルオートマトン法による避難流動のシミュレーション, 日本オペレーションズ・リサーチ学会和文論文誌, Vol. 51, pp. 94–111 (2008).
- [14] 小久保聡, 山本和弘, 山下博史: セルオートマトンによる火災時の避難行動のシミュレーション, 日本機械学会論文集 B 編, Vol. 74, No. 748, pp. 2724–2730 (2008).
- [15] 五十嵐誠, 大貝彰: セル・オートマトンを用いた土地利用政策の影響を考慮した人口・従業者分布予測モデル, 日本建築学会計画系論文集, No. 567, pp. 67–74 (2003).
- [16] Li, X. and Yeh, A. G.-O.: Modelling sustainable urban development by the integration of constrained cellular automata and GIS, *International Journal of Geographical Information Science*, Vol. 14, No. 2, pp. 131–152 (2000).
- [17] Fates, N. A. and Morvan, M.: An experimental study of robustness to asyn-

- chronism for elementary cellular automata, *Complex Systems*, Vol. 16, pp. 1–27 (2005).
- [18] Wolfram, S.: Cellular automata as simple self-organizing systems, Technical report (1982).
- [19] Wolfram, S.: Cellular automata, *Los Alamos Science*, Vol. 9, pp. 2–27 (1983).
- [20] Wolfram, S.: Universality and complexity in cellular automata, *Physica D: Non-linear Phenomena*, Vol. 10, No. 1-2, pp. 1–35 (1984).
- [21] Kanada, Y.: Asynchronous 1D cellular automata and the effects of fluctuation and randomness, in *Proceedings of the fourth conference on artificial life (A-Life IV)*. MIT Press (1994).
- [22] 轡田康, 広瀬貞樹, 蛭川繁, 木村春彦: セルオートマトンを用いた雪の結晶の類似パターン生成, 電子情報通信学会論文誌 D, Vol. 83, No. 9, pp. 909–918 (2000).
- [23] Moore, E. F.: The Firing Squad Synchronization Problem, in Moore, E. F. ed., *Sequential Machines, Selected Papers*, pp. 213–214, Addison-Wesley, Reading MA. (1964).
- [24] Minsky, M. and McCarthy, J.: Computation : Finite and infinite machines, *Prentice Hall*, pp. 28–29 (1967).
- [25] Goto, E.: A minimal-time solution of the firing squad synchronization problem, *Course Notes for Applied Mathematics 298*, pp. 52–59 (1962).
- [26] Waksman, A.: An optimum solution to the firing squad synchronization problem, *Information and Control*, Vol. 9, pp. 66–78 (1966).
- [27] Balzer, R.: An 8-state minimal time solution to the firing squad synchronization problem, *Information and Control*, Vol. 10, pp. 22–42 (1967).
- [28] Gerken, H.-D.: Über Synchronisationsprobleme bei Zellularautomaten, *Diplomarbeit, Technische Universität Braunschweig*, pp. 1–50 (1987).
- [29] Mazoyer, J.: A six states minimal time solution to the firing squad synchronization problem, *Theoretical Computer Science*, Vol. 50, pp. 183–238 (1987).

第 2 章

セルオートマトン

セルオートマトン (Cellular Automata: CA) とは, セルとよばれる同一の有限オートマトンを格子上に並べた空間に対して, 単純な規則で状態遷移を繰り返す離散的計算モデルである. その歴史は古く, 1950 年代に Neumann と Ulam が有限オートマトン理論からセルオートマトン理論へ発展させた. 初期の興味は, コンピュータに自己修復機能を与えることを目的とした「抽象的な空間に配置された何個の有限状態のセルで自己と同じ配置を構成できるか」という問題であり, 細々と研究されていた. 1970 年に Conway が 2 状態 9 近傍 2 次元セルオートマトンであるライフゲームを提案した [1]. 今でいうコンピュータゲームのはしりであり, これを期にセルオートマトンが注目されるようになった. ライフゲームの規則は単純であるが, 非常に複雑な挙動を示し, ランダム性と規則性の間で変動する. また, ライフゲームはチューリング完全であり, チューリングマシンと同等の計算能力を持つことが示された [2]. 1983 年に Wolfram はこれまで 2 次元モデルが中心であったセルオートマトンに 1 次元モデルを提案し, 1984 年に 2 状態 3 近傍 1 次元セルオートマトンの時空間パターンが 4 つのクラスに分類できることを示した [3-5]. 現在, セルオートマトンは暗号技術 [6], 免疫システム [7,8], 交通流シミュレーション [9-13], 海洋汚染シミュレーション [14], 人の移動シミュレーション [15], 避難流動のシミュレーション [16-18], 人口分布予測 [19], 都市開発 [20] など幅広い分野に適用されている.

2.1 1次元セルオートマトン

セルオートマトンを構成する有限オートマトン \mathcal{A} は, $\mathcal{A} = (\mathcal{S}, \delta)$ の 2 項組で定義される. \mathcal{S} は状態の有限集合であり, δ は遷移関数である. 有限オートマトン \mathcal{A} を, 図 2.1 のように直線上に配置する. 図中の四角が 1 つの有限オートマトンの有限制御部を表し, セルを繋いでいる直線は読取りヘッドを表している. 1 つの有限オートマトンをセルとよび, セルを並べた空間をセル空間とよぶ. 通常の有限オートマトンであれば読取りヘッドが入力テープの記号を読み取って状態遷移を繰り返すが, セルオートマトンを構成する有限オートマトンには入力テープが無く, 読取りヘッドは接続しているセルの状態を入力として読み取る. 自身のセルと隣接するセルを合わせて近傍セルとよび, 遷移関数は近傍セルの状態を入力として定義される. 例えば, セルに左右の 2 個のセルが隣接しているような 3 近傍セルオートマトンの場合, 位置 i のセル (以下, 単にセル C_i とよぶ) の時刻 t における状態を s_i^t で表すと, セル C_i の時刻 $t+1$ における状態 s_i^{t+1} は遷移関数 δ によって式 2.1 のように定まる. ただし, $s_i^{t+1}, s_{i-1}^t, s_i^t, s_{i+1}^t \in \mathcal{S}$ である.

$$s_i^{t+1} = \delta(s_{i-1}^t, s_i^t, s_{i+1}^t) \quad (2.1)$$

状態遷移は全てのセルが同期して並行に行われる. 1 回状態遷移する時間を 1 ステップとよぶ. 全てのセルが適当な状態を持っているとき, このセル空間を様相とよぶ. 特に時刻 $t=0$ における様相を初期様相とよぶ. 初期様相から状態遷移を繰り返し, その様相を観測する. これがセルオートマトンの基本である.

一般的にセルオートマトンは無限のセル空間で定義されるが, 現実的には不可能のため, 実際には有限のセル空間が用いられる. 有限のセル空間を用いた際に問題になるのが境界にあるセルの近傍セルであるが, 両端のセルが接しているとみなす巡回境界条件また

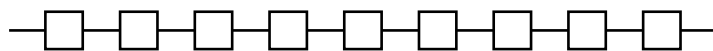


図 2.1: 1次元セルオートマトンのセル空間

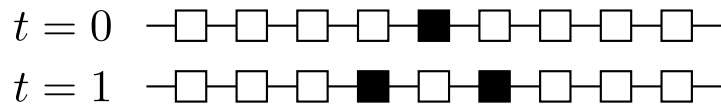


図 2.2: 1次元セルオートマトンの状態遷移

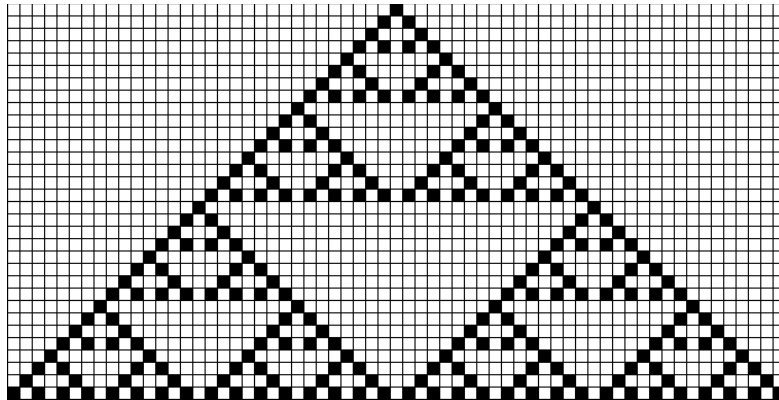


図 2.3: 1次元セルオートマトンの時空間パターン

は境界外のセルがステップごとにランダムな状態をとる散逸境界条件がよく用いられる。

例えば、2つの状態0, 1に対して式 2.2 を満たす遷移関数 δ を持つ1次元セルオートマトンを考える。

$$\begin{aligned}
 \delta(0, 0, 0) &= 0 & \delta(0, 0, 1) &= 1 \\
 \delta(0, 1, 0) &= 0 & \delta(0, 1, 1) &= 1 \\
 \delta(1, 0, 0) &= 1 & \delta(1, 0, 1) &= 0 \\
 \delta(1, 1, 0) &= 1 & \delta(1, 1, 1) &= 0
 \end{aligned} \tag{2.2}$$

図 2.2 上段のように、状態1のセルがひとつだけ存在し、残りは状態0のセルである初期様相に対して、遷移関数 δ を適用すると1ステップで図 2.2 下段のようになる。図中で状態1のセルは黒色の四角形、状態0のセルは白色の四角形で表している。

同様に状態遷移を繰り返し、図 2.3 のように様相を時刻ごとに上から下へと並べ、様相の変化の過程を2次的に表したパターンを時空間パターンとよぶ。図中では簡略化のため各セルと各時刻の様相を隣接させて表している。

単純な規則を繰り返し適用することで、この例ではフラクタル図形の1種であるシェルピンスキーギャスケットを生成することができる。

2.2 2次元セルオートマトン

セルを2次元平面上に配置すると2次元のセル空間を定義できる。2次元セルオートマトンでは、図 2.4 に示すような正方格子に配置したセル空間がよく用いられる。正方格子上の2次元セルオートマトンの近傍セルを図 2.5 に示す。自身（灰色のセル）と上下左右の4個のセルを含めた Neumann 近傍（図 2.5(a)）、あるいは斜めのセルも加えた Moore 近傍（図 2.5(b)）がよく用いられる。

2次元セルオートマトン上で定義されるよく知られた問題にライフゲームがある。正方格子に配置された2次元セルオートマトンを考える。各セルには生（状態1）と死（状

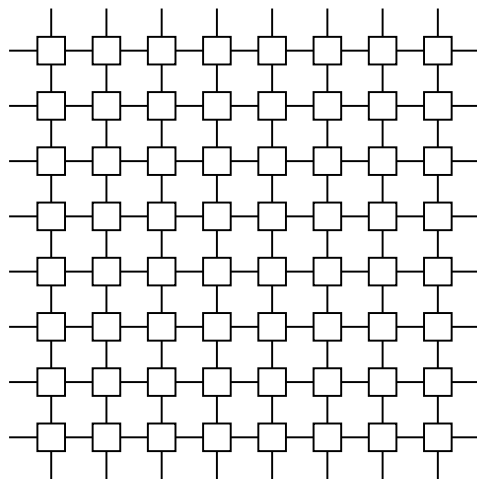
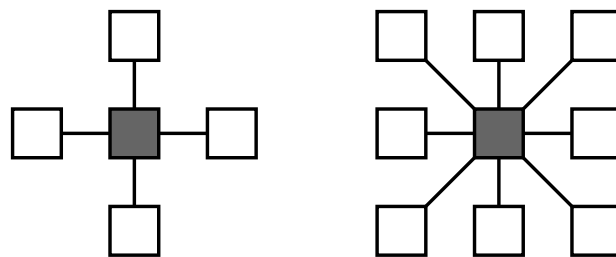


図 2.4: 正方格子上の2次元セルオートマトンのセル空間



(a) Neumann 近傍 (5 近傍) (b) Moore 近傍 (9 近傍)

図 2.5: 正方格子上の2次元セルオートマトンの近傍セル

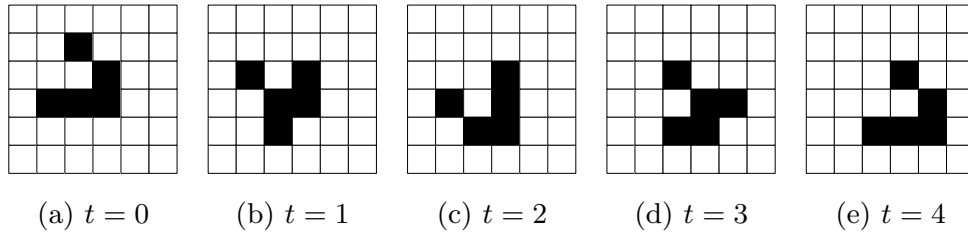


図 2.6: ライフゲームの状態遷移 (グライダー)

態 0) の 2 つの状態がある. 近傍には Moore 近傍を用いる. 位置 (x, y) のセル (以下, 単にセル $C_{x,y}$ とよぶ) の時刻 t における状態を $s_{x,y}^t$ と表すと, セル $C_{x,y}$ の時刻 $t+1$ における状態 $s_{x,y}^{t+1}$ は近傍セルの状態により式 2.3 のように定まる. ただし, k はセル $C_{x,y}$ 以外の近傍セルで状態 1 のセルの個数を表す. このように, 自身以外の近傍セルの総和によって遷移規則が定まるセルオートマトンを外総和型セルオートマトンとよぶ. 外総和型セルオートマトンでは, どのような状態が周囲に何個あるかということによって次の状態が決められ, それらが周囲のどの位置にあるのかは関係しない.

$$s_{x,y}^{t+1} = \begin{cases} 1 & \text{if } (s_{x,y}^t = 0 \text{ and } k = 3) \text{ or } (s_{x,y}^t = 1 \text{ and } 2 \leq k \leq 3) \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

簡単に言えば以下の 4 つの規則にしたがって状態遷移する.

誕生 死のセルに生のセルが 3 つ隣接していれば, 誕生する.

生存 生のセルに生のセルが 2 つか 3 つ隣接していれば, 生存する.

過疎 生のセルに生のセルが 1 つ以下隣接していれば, 過疎により死滅する.

過密 生のセルに生のセルが 4 つ以上隣接していれば, 過密により死滅する.

例えば, 図 2.6(a) のような初期様相 (黒が生セル, 白が死セル) を考える. 状態遷移を繰り返すと図 2.6(a) から 4 ステップで図 2.6(e) のように遷移する. 図中では簡略化のため各セルを隣接して表している. このパターンは, 4 ステップで斜め方向に 1 セルだけ移動をすることがわかる. このパターンはグライダーとよばれる.

ライフゲームには, 移動物体 (形を変えずに移動する物体), 固定物体 (形が変化しない

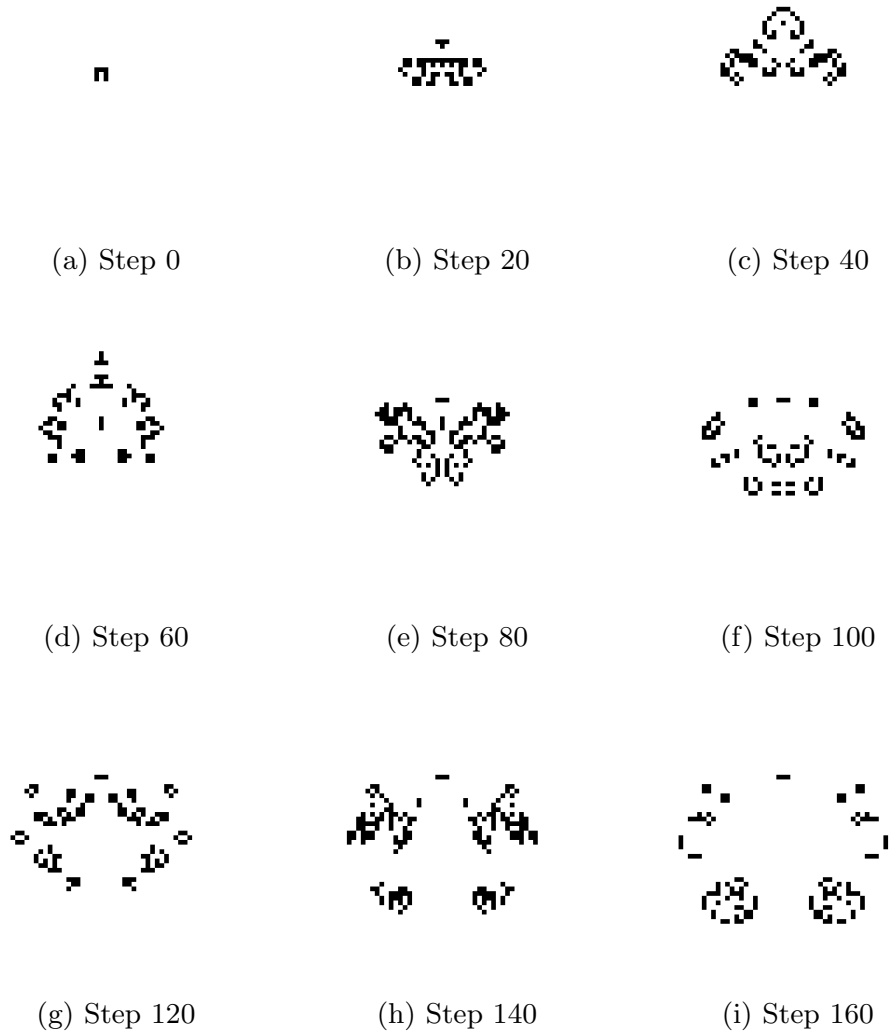


図 2.7: パイ・ヘプトミノの状態遷移

物体), 振動子 (数ステップ後に元の形に戻る物体), 繁殖型 (無限に増え続ける物体) の 4 種類の物体がよく知られている. 先程のグライダーは移動物体に分類される. 7 つの生のセルで構成されるパイ・ヘプトミノとよばれるパターン (図 2.7(a)) から状態遷移を繰り返すと, 20 ステップごとに図 2.7(b) から図 2.7(i) のような様相が得られる. ただし, セルの枠線は省略して表している. 初期様相は単純なパターンでありながら, 単純な規則を用いた状態遷移を繰り返すことで多様なパターンを生み出すことがわかる.

参考文献

- [1] Conway, J.: The game of life, *Scientific American*, Vol. 223, No. 4, p. 4 (1970).
- [2] Chapman, P.: Life universal computer (2002), <http://www.igblan.free-online.co.uk/igblan/ca>.
- [3] Wolfram, S.: Cellular automata, *Los Alamos Science*, Vol. 9, pp. 2–27 (1983).
- [4] Wolfram, S.: Statistical mechanics of cellular automata, *Reviews of modern physics*, Vol. 55, No. 3, p. 601 (1983).
- [5] Wolfram, S.: Universality and complexity in cellular automata, *Physica D: Non-linear Phenomena*, Vol. 10, No. 1-2, pp. 1–35 (1984).
- [6] Chaudhuri, P. P.: *Additive cellular automata: theory and applications*, Vol. 1, John Wiley & Sons (1997).
- [7] Chowdhury, D. and Stauffer, D.: Systematics of the models of immune response and autoimmune disease, *Journal of Statistical Physics*, Vol. 59, No. 3, pp. 1019–1042 (1990).
- [8] Pandey, R. B. and Stauffer, D.: Metastability with probabilistic cellular automata in an HIV infection, *Journal of statistical physics*, Vol. 61, No. 1, pp. 235–240 (1990).
- [9] Resnick, M.: Turtles, termites, and traffic jams: explorations in massively powerful microworlds (1995).
- [10] Nagel, K. and Barrett, C. L.: Using microsimulation feedback for trip adaptation for realistic traffic in Dallas, *International Journal of Modern Physics C*, Vol. 8, No. 03, pp. 505–525 (1997).
- [11] Howard, K. R.: Unjamming traffic with computers, *Scientific American*, Vol. 277, No. 4, pp. 86–92 (1997).
- [12] 山本直史, 森下信: セルラオートマトンによる交通流のモデル化とその制御, 日本機械学会論文集 C 編, Vol. 65, No. 637, pp. 3553–3558 (1999).

- [13] 玉城龍洋, 安江里佳, 北栄輔 (英輔): セル・オートマトンによる自動車専用道路の交通シミュレーション, 情報処理学会論文誌: 数理モデル化と応用, Vol. 46, No. SIG 10 (TOM 12), pp. 30–40 (2005).
- [14] 森下信, 長谷川純一, 中野孝昭, 築山洋: セルラオートマトン法による拡散現象の解析, 日本造船学会論文集, Vol. 1998, No. 183, pp. 181–187 (1998).
- [15] 大高善光: セルラオートマトン法による店舗内購買シミュレーション, 日本機械学会第 75 回通常総会講演会講演論文集, 1998. 3, pp. 411–412 (1998).
- [16] 松田泰治, 大塚久哲, 樗木武, 内田広明: セルオートマトン法を用いた地下街の避難行動シミュレーションに関する一考察, 地域安全学会論文集, No. 2, pp. 95–100 (2000).
- [17] 大鑄史男, 小野木基裕: セルオートマトン法による避難流動のシミュレーション, 日本オペレーションズ・リサーチ学会和文論文誌, Vol. 51, pp. 94–111 (2008).
- [18] 小久保聡, 山本和弘, 山下博史: セルオートマトンによる火災時の避難行動のシミュレーション, 日本機械学会論文集 B 編, Vol. 74, No. 748, pp. 2724–2730 (2008).
- [19] 五十嵐誠, 大貝彰: セル・オートマトンを用いた土地利用政策の影響を考慮した人口・従業者分布予測モデル, 日本建築学会計画系論文集, No. 567, pp. 67–74 (2003).
- [20] Li, X. and Yeh, A. G.-O.: Modelling sustainable urban development by the integration of constrained cellular automata and GIS, *International Journal of Geographical Information Science*, Vol. 14, No. 2, pp. 131–152 (2000).

第 3 章

不完全な同期下の単純セルオートマトンの時空間パターンによる分類

Wolfram は単純セルオートマトン（最も単純な 2 状態 3 近傍 1 次元のセルオートマトン）の時空間パターンを網羅的に調べ、単純セルオートマトンを以下の 4 つのクラスに分類した [1,2].

- クラス 1 全てのセルが同一の状態へ収束する
- クラス 2 規則的な変動を繰り返す
- クラス 3 不規則な変動をする
- クラス 4 規則性と不規則性が混在した変動をする

特にクラス 3 はカオス，クラス 4 はカオスの縁とよばれている．カオスの縁はクラス 1, 2 の規則的なものとクラス 3 の不規則なものの中に位置し，現実の複雑な現象はこのような規則性と不規則性の混在したものから生じると考えられている．また Wolfram は単純セルオートマトンの全ての遷移関数を調べ，それぞれのセルオートマトンの時空間パターンの構造を詳細に解析している [3].

Ingerson 等は，セルオートマトンにノイズが加わりセルオートマトンの同期性が失われた場合に，その時空間パターンにどのような変化が起こるかを研究するため，非同期セ

ルオートマトンを定義し、Wolfram と同様に非同期の単純セルオートマトンにおける時空間パターンを解析した [4].

非同期なセルオートマトンには、さまざまなモデルが提案されている [5–7]. 特に金田は、1 つのセルをランダムに選択して状態遷移する完全に非同期なセルオートマトンを定義し、その時空間パターンの特徴が遷移関数から完全に決定できることを明らかにした [7].

一方、Fatès 等はセルオートマトンにおけるノイズへの頑強性を調べるために、遷移確率を用いたセルオートマトンを定義した [8]. これは、状態遷移するセルを遷移確率によってランダムに複数選択し、状態遷移を行うセルオートマトンである. Fatès 等はこのセルオートマトンにおいて、同時に遷移するセルの割合の変化がどのような影響を与えるかを研究した. しかし、Fatès 等が定義したセルオートマトンに関しては、その時空間パターンについてまだ詳しい研究が行われていない.

本章では、完全に同期した通常のセルオートマトンと金田の定義した完全に非同期なセルオートマトンの時空間パターンと、Fatès 等の定義したセルオートマトンの時空間パターンを比較することによって、Fatès 等の定義した単純セルオートマトンの時空間パターンの分類を試みる.

3.1 不完全な同期下のセルオートマトン

Fatès 等の定義したセルオートマトンは、複数のセルをランダムに選択して状態遷移する. s_i^t を時刻 t における位置 i にあるセルの状態, δ を遷移関数, p を遷移確率とすると、時刻 $t+1$ における位置 i にあるセルの状態 s_i^{t+1} は、式 3.1 のように表される. ただし、 p の値は $0 < p < 1$ である.

$$s_i^{t+1} = \begin{cases} \delta(s_{i-1}^t, s_i^t, s_{i+1}^t) & (p) \\ s_i^t & (1-p) \end{cases} \quad (3.1)$$

各セルは確率 p で状態遷移し、確率 $1-p$ で状態遷移しない. その結果、確率的に 1 ステップ当り空間内の p の割合のセルが同期して状態遷移することになる.

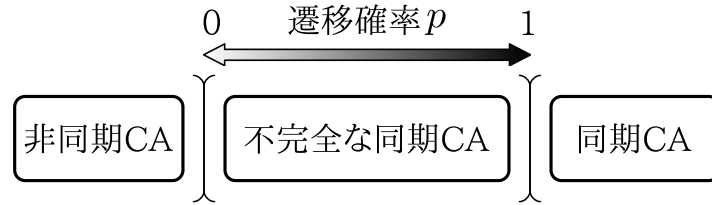


図 3.1: 非同期 CA, 不完全な同期下の CA, 同期 CA の関係

これは、全てのセルが同時に状態遷移する完全に同期した通常のセルオートマトン（同期セルオートマトン）と、金田が定義した 1 セルのみしか状態遷移しない完全に非同期なセルオートマトン（非同期セルオートマトン）の中間に位置するセルオートマトンと考えられる。図 3.1 に示すように、遷移確率 p が 1 に近ければ同期セルオートマトンに、0 に近ければ非同期セルオートマトンに近い。 p は $0 < p < 1$ で定義されているが、1 であればまさに同期セルオートマトンであり、 $\frac{1}{\text{セル数}}$ であればほぼ非同期セルオートマトンである。Fatès 等のモデルは、完全に同期しないという意味では非同期なセルオートマトンではあるが、選択された複数のセルは同期して状態遷移するが同期しないセルもあるという意味で、不完全な同期をするセルオートマトンとも考えられる。その意味で、本章では不完全な同期下のセルオートマトンとよぶことにする。

3.2 不完全な同期下の単純セルオートマトンの分類

単純セルオートマトンは 2 状態 3 近傍のセルオートマトンである。したがって、遷移関数は $2^3 = 8$ の近傍集合に対して次の状態 0, 1 を決めることになるので、単純セルオートマトンは $2^8 = 256$ 存在する。各単純セルオートマトンの遷移関数から式 3.2 によって 10 進数 D を得て、その単純セルオートマトンを $\#D$ とよぶ。

$$D = \delta(1, 1, 1) \times 2^7 + \delta(1, 1, 0) \times 2^6 + \delta(1, 0, 1) \times 2^5 + \delta(1, 0, 0) \times 2^4 + \delta(0, 1, 1) \times 2^3 + \delta(0, 1, 0) \times 2^2 + \delta(0, 0, 1) \times 2^1 + \delta(0, 0, 0) \times 2^0 \quad (3.2)$$

256 の単純セルオートマトンのうち、遷移関数を 0, 1 反転または左右反転して同一となるセルオートマトンを同値と見なすと、88 の同値類が存在する。各同値類の代表元を $\#D$ の最小のものとし、比較対象をこの 88 のセルオートマトンとする。

セル数を 200 とし、初期様相は 0, 1 が等確率^{*1}で現れるようにランダムに生成し、境界条件は両端のセルが接しているとみなす巡回境界条件を用いる。

遷移確率 p の値は 0.1 から 0.9 までとし、増分を 0.1 とする。

時空間パターンは、状態 0 を白、状態 1 を黒で表す。ただし、非同期セルオートマトンの時空間パターンでは、1 つ上の様相から空間内に配置したセルの数と同じ回数だけ状態遷移を行った後の様相をそれぞれ示すことにする。セル数を 200 としているので、非同期セルオートマトンの時空間パターンでは、ある時刻の様相から 200 回状態遷移が行われた時点の様相が次に描かれる。また、不完全な同期下のセルオートマトンでは、ある様相の 1 つ下の様相は、全体の p 割のセルが状態遷移を行った後の様相である。つまり、同期セルオートマトンや非同期セルオートマトンの時空間パターンではある様相から 1 つ下の様相へは 200 回セルの状態遷移が行われた後の様相であるのに対して、不完全な同期下のセルオートマトンではおおよそ $200 \times p$ 回、例えば $p = 0.3$ としたときはセルの状態遷移が約 60 回しか行われていない様相である。そのため、不完全な同期下のセルオートマトンの時空間パターンは、同期セルオートマトン、非同期セルオートマトンの時空間パターンより変化が遅く見えることに注意されたい。

不完全な同期下のセルオートマトンの時空間パターンを、その両極にある同期セルオートマトンと非同期セルオートマトンの時空間パターンと比較し、不完全な同期下の単純セルオートマトンを以下に示すような 5 つのグループに分類した^{*2}。

グループ 1 同期 CA, 非同期 CA と同様なパターンに収束する

グループ 2 p の値によって徐々にパターンが変化する

グループ 3 非同期 CA と同様なパターンに収束する

グループ 4 同期 CA と同様なパターンに収束する

グループ 5 同期 CA, 非同期 CA と異なるパターンに収束する

^{*1} セルオートマトンの中には初期様相に非常に鋭敏なものが存在する。本実験では遷移確率による定常的な収束形態を調べるのが目的であったため、試行ごとに同一の収束形態が得られた、0, 1 が等確率で現れる初期様相を用いた。

^{*2} 比較対象とした 88 のセルオートマトンの時空間パターンは付録 A を参照されたい。

3.2.1 同期セルオートマトン，非同期セルオートマトンと同様なパターンに収束する

グループ 1 は，同期セルオートマトン，非同期セルオートマトンと同様なパターンに収束するものである．このグループには以下の 27 のセルオートマトン*3が属している．

グループ 1

0	4	8	12	13	28	32	36	40	44
72	76	77	78	104	108	128	132	136	140
160	164	168	172	200	204	232			

グループ 1 は，基本的に図 3.2 のように全てのセルが同一の状態に収束するものと，図 3.3 のようにセルの状態の変化がなくなるものが属している．

#28 の時空間パターンを図 3.4 に示す．同期セルオートマトンの時空間パターンは，状態が変化しないセルと振動するセルの 2 つによって構成されている．不完全な同期下のセルオートマトンや非同期セルオートマトンでも，振動は不安定にはなるが，基本的にその 2 つの性質を持つセルによって構成されている．そのため，#28 をグループ 1 とした．

#108 の時空間パターンを図 3.5 に示す．細かい点がそれぞれのセルオートマトンで異



(a) 非同期 CA

(b) $p = 0.3$

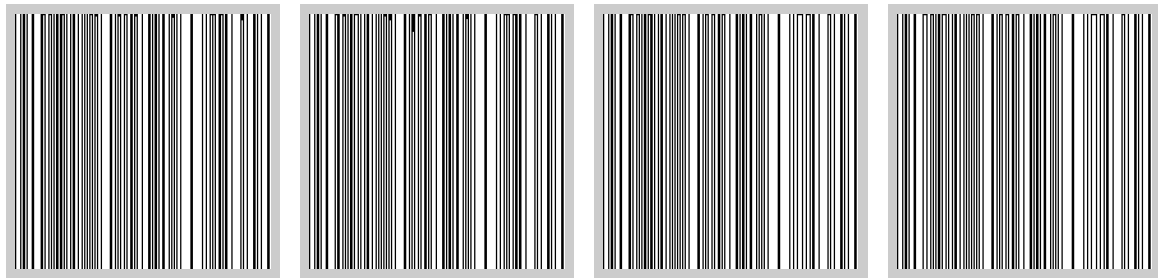
(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 3.2: 同一の状態に収束する時空間パターン (#40)

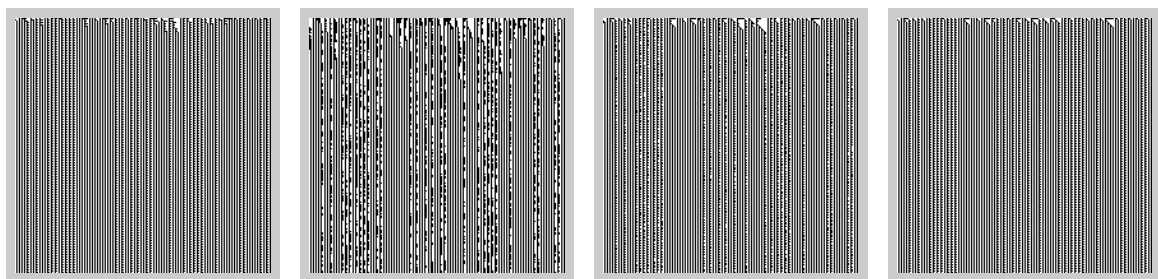
*3 記号#を省略して示す．



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

図 3.3: 静止状態に収束する時空間パターン (#76)

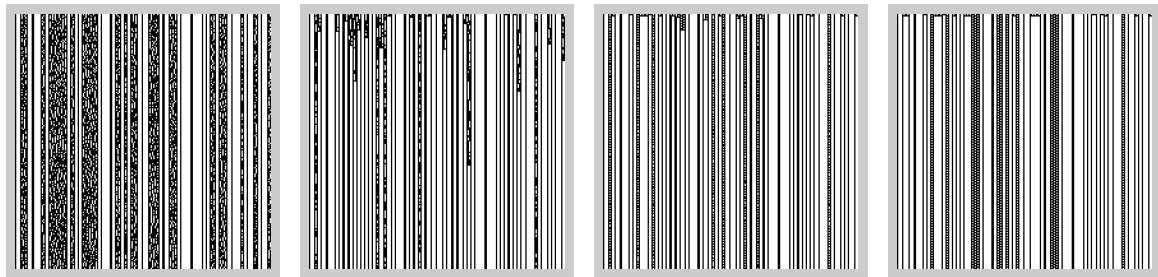


(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

図 3.4: #28 の時空間パターン

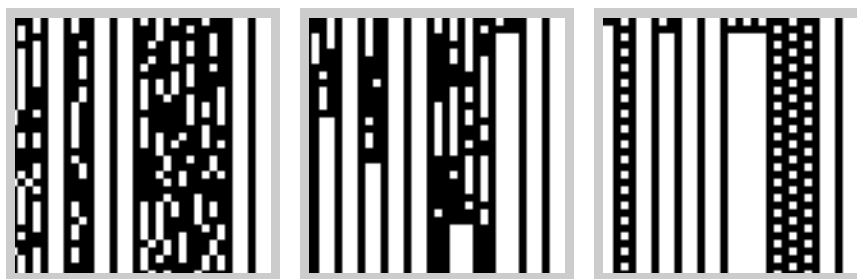
なって見える。図 3.5 の一部分を拡大した図を図 3.6 に示す。#28 に比べて各セルオートマトン間の違いが多少大きいように見えるが、これも #28 と同様に基本的に状態が変化しないセルと振動するセルの 2 つの性質を持つセルによって構成されている。そのため、#28 と同様に #108 もグループ 1 とした。



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

図 3.5: #108 の時空間パターン



(a) 非同期 CA (b) $p = 0.3$ (c) 同期 CA

不完全な同期下の CA

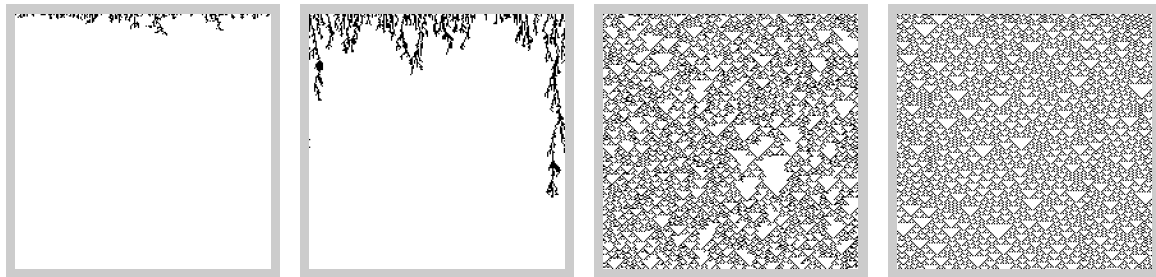
図 3.6: 拡大した #108 の時空間パターン

3.2.2 p の値によって徐々にパターンが変化する

グループ 2 は、遷移確率 p の値の変化によって徐々に時空間パターンが変化するものである。このグループには以下の 37 のセルオートマトンが属している。

グループ 2

1	3	7	9	11	18	19	22	23	25
26	27	30	33	35	37	41	43	45	46
50	51	54	57	58	60	62	90	105	106
110	122	126	142	146	150	178			

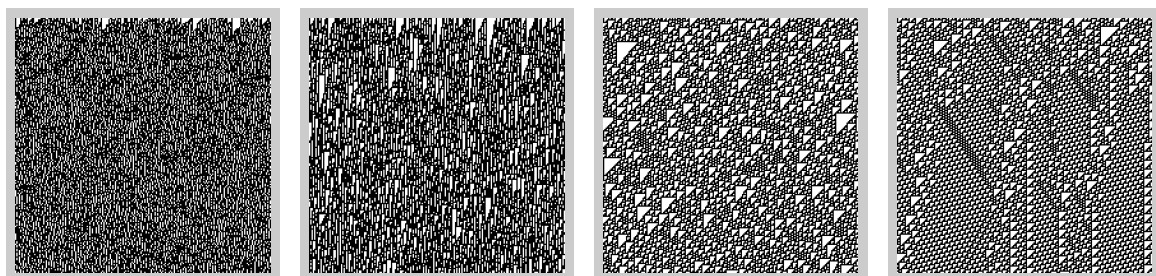


(a) 非同期 CA

(b) $p = 0.3$ (c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 3.7: 低 p 値で同一の状態へ収束する時空間パターン (#18)

(a) 非同期 CA

(b) $p = 0.3$ (c) $p = 0.9$

(d) 同期 CA

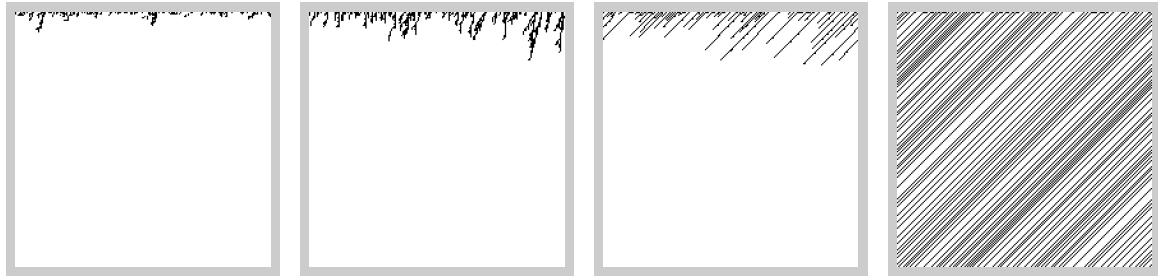
不完全な同期下の CA

図 3.8: 低 p 値で不規則な変動を行う時空間パターン (#110)

p の値が大きいときは同期セルオートマトンに近いパターンを示し、 p の値が小さいときは非同期セルオートマトンに近いパターンを示す。グループ 2 の中でも図 3.7 のように全てのセルが同一の状態、もしくは静止状態へと収束するものと、図 3.8 のように不規則な変動を行うものがある。

3.2.3 非同期セルオートマトンと同様なパターンに収束する

グループ 3 は、非同期セルオートマトンと同様な時空間パターンに収束するものである。このグループには以下の 19 のセルオートマトンが属している。



(a) 非同期 CA

(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

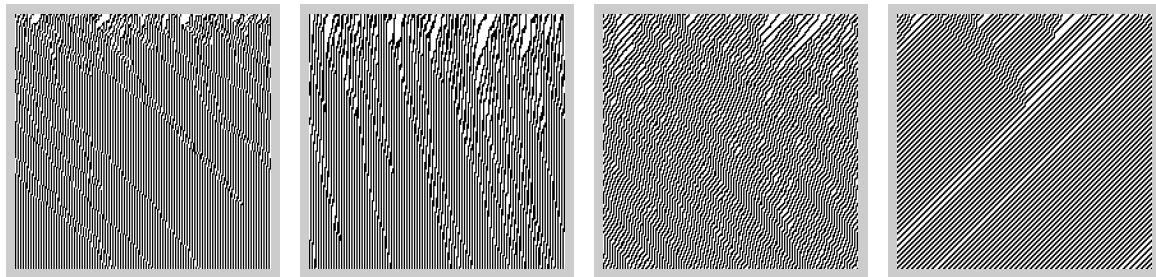
図 3.9: グループ 3 の同一の状態に収束する時空間パターン (#34)

グループ 3

2	5	10	14	15	24	29	34	42	56
73	74	94	130	138	152	162	170	184	

これは、全ての p の値において非同期セルオートマトンと同様な時空間パターンへ収束するものである。グループ 3 は基本的に、図 3.9 のように非同期セルオートマトンでは全てのセルが同一の状態に収束し、不完全な同期下のセルオートマトンも同様に全てのセルが同一の状態に収束するものと、図 3.10 のように非同期セルオートマトンと不完全な同期下のセルオートマトンが静止状態に収束するものがある。

図 3.10 に示した #14 の時空間パターンにおいて、同期セルオートマトンでは状態がシフト移動する。非同期セルオートマトンでは最終的に図 3.11 のように状態が静止する。不完全な同期下のセルオートマトンの低 p 値 ($p = 0.3$) での時空間パターンは、図 3.10 でも見られるように非同期セルオートマトンの時空間パターンとほぼ同様の変化を行い、やがて図 3.11 の形へ収束する。また高 p 値 ($p = 0.9$) での時空間パターンは、初期状態から暫くは変化が大きいですが、ある程度時間が経過すると図 3.12 のようになり、最終的には必ず図 3.11 のパターンに収束する。このように全ての p の値において、非同期セルオートマトンと同様な時空間パターンに収束する。



(a) 非同期 CA

(b) $p = 0.3$ (c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 3.10: グループ 3 の静止状態に収束する時空間パターン (#14)

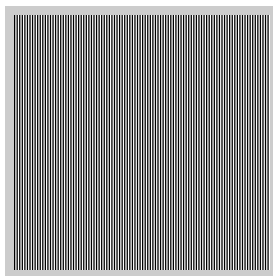
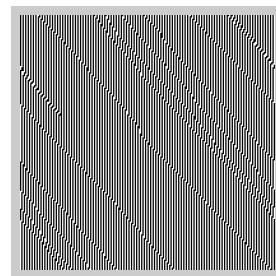


図 3.11: #14 の非同期 CA, 不完全な同期下の CA が収束するパターン

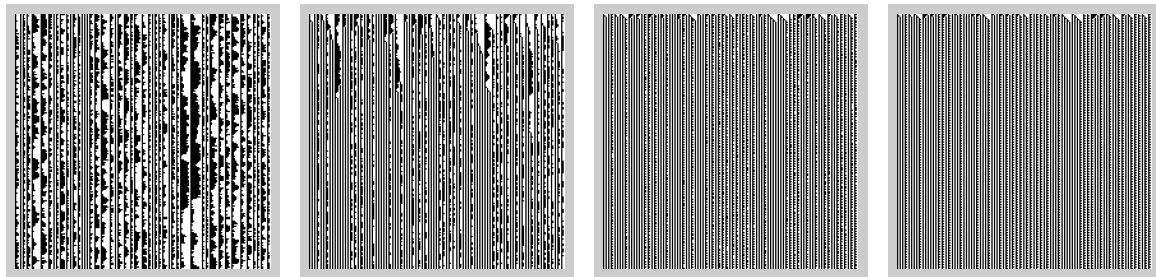
図 3.12: #14, $p = 0.9$ の約 800 ステップ程経過した後の時空間パターン

3.2.4 同期セルオートマトンと同様なパターンに収束する

グループ 4 は, 同期セルオートマトンと同様な時空間パターンに収束するものである。#156 のみがこのグループに属している。

グループ 4 は, 図 3.13 のように同期セルオートマトンでは状態が変化しないセルと振動するセルの 2 つによって構成されている。不完全な同期下のセルオートマトンでも同様に, 振動は不安定にはなるが基本的にその 2 つの性質を持つセルによって構成されている。しかし, 非同期セルオートマトンはこれと異なり, 状態の変化しないセルの間を状態 1-0 の境界がランダムウォーク的に変動するパターンになる。

この時空間パターンはグループ 1 の #28 とよく似ているが, #28 では同期セルオートマトン, 不完全な同期下のセルオートマトン, 非同期セルオートマトン間で同様な時空間



(a) 非同期 CA

(b) $p = 0.3$ (c) $p = 0.9$

(d) 同期 CA

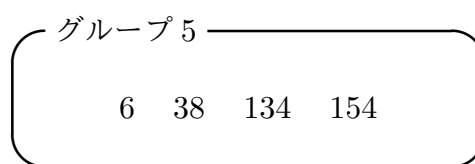
不完全な同期下の CA

図 3.13: グループ 4 の時空間パターン (#156)

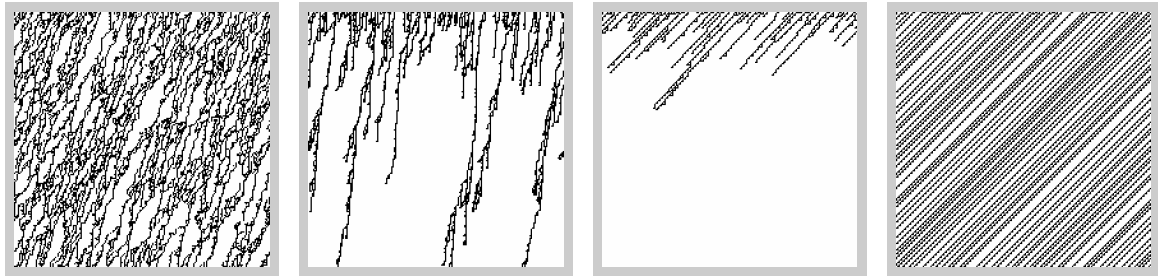
パターンであったのに対して、これは非同期セルオートマトンにおいて異なった時空間パターンが展開されている。

3.2.5 同期セルオートマトン，非同期セルオートマトンとも異なるパターンに収束する

グループ 5 は、同期セルオートマトン，非同期セルオートマトンのどちらとも異なる時空間パターンに収束するものである。このグループには以下の 4 のセルオートマトンが属している。



これは図 3.14 のように同期セルオートマトンでは状態がシフトする時空間パターンを示し、非同期セルオートマトンでは不規則な変動が続く時空間パターンを示している。これらの時空間パターンに対し、不完全な同期下のセルオートマトンでは、全てのセルが同一の状態へ収束する時空間パターンという全く異質な時空間パターンへ収束している。図 3.14 では $p = 0.3$ での時空間パターンが同一の状態へ収束していないが、この後さらに時間が経過すると全てのセルが同一の状態へ収束する。



(a) 非同期 CA

(b) $p = 0.3$ (c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 3.14: グループ 5 の時空間パターン (#38)

3.3 不完全な同期下のセルオートマトンにおける遷移関数と時空間パターンの特徴との関係

金田 [7] は, 非同期セルオートマトンにおける時空間パターンの特徴が遷移関数から完全に決定できることを明らかにした. その遷移関数と時空間パターンの特徴との関係は以下に示す式 3.3–3.10 のとおりである.

$$\delta(0,0,0) = \begin{cases} 1 \text{ のとき: 状態 } 0 \text{ のクラスタが } 2 \text{ つに分かれる} \\ 0 \text{ のとき: 状態 } 0 \text{ のクラスタが分かれることはない} \end{cases} \quad (3.3)$$

$$\delta(1,1,1) = \begin{cases} 0 \text{ のとき: 状態 } 1 \text{ のクラスタが } 2 \text{ つに分かれる} \\ 1 \text{ のとき: 状態 } 1 \text{ のクラスタが分かれることはない} \end{cases} \quad (3.4)$$

$$\delta(0,0,1) = \begin{cases} 1 \text{ のとき: 状態 } 0\text{-}1 \text{ の境界は左へ動く} \\ 0 \text{ のとき: 状態 } 0\text{-}1 \text{ の境界が左へ動くことはない} \end{cases} \quad (3.5)$$

$$\delta(1,0,0) = \begin{cases} 1 \text{ のとき: 状態 } 1\text{-}0 \text{ の境界は右へ動く} \\ 0 \text{ のとき: 状態 } 1\text{-}0 \text{ の境界が右へ動くことはない} \end{cases} \quad (3.6)$$

$$\delta(1,1,0) = \begin{cases} 0 \text{ のとき: 状態 } 1\text{-}0 \text{ の境界は左へ動く} \\ 1 \text{ のとき: 状態 } 1\text{-}0 \text{ の境界が左へ動くことはない} \end{cases} \quad (3.7)$$

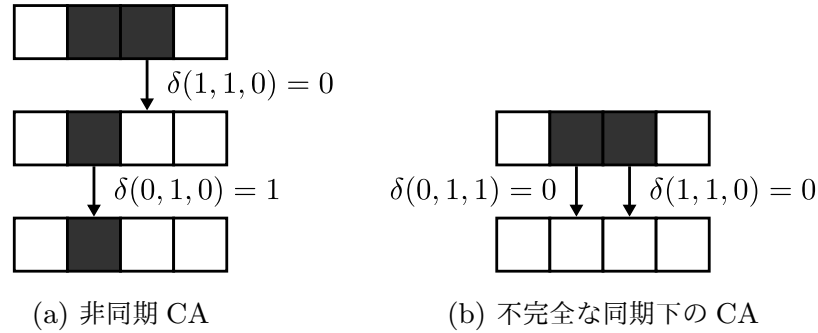


図 3.15: $\delta(0, 1, 0) = 1$ である場合の非同期 CA と不完全な同期下の CA におけるパターンの違い

$$\delta(0, 1, 1) = \begin{cases} 0 \text{ のとき} : \text{状態 } 0\text{-}1 \text{ の境界は右へ動く} \\ 1 \text{ のとき} : \text{状態 } 0\text{-}1 \text{ の境界が右へ動くことはない} \end{cases} \quad (3.8)$$

$$\delta(0, 1, 0) = \begin{cases} 0 \text{ のとき} : \text{状態 } 1 \text{ のクラスタは死滅する} \\ 1 \text{ のとき} : \text{状態 } 1 \text{ のクラスタは不死となる} \end{cases} \quad (3.9)$$

$$\delta(1, 0, 1) = \begin{cases} 1 \text{ のとき} : \text{状態 } 0 \text{ のクラスタは死滅する} \\ 0 \text{ のとき} : \text{状態 } 0 \text{ のクラスタは不死となる} \end{cases} \quad (3.10)$$

本節では、不完全な同期下のセルオートマトンにおける遷移関数と時空間パターンの特徴との関係を明らかにする。

$\delta(0, 1, 0) = 1$ である場合、非同期セルオートマトンでは図 3.15(a) のように、 $\delta(1, 1, 0) = 0$ (または $\delta(0, 1, 1) = 0$) によってクラスタ*4のサイズが小さくなり、サイズがちょうど 1 になったとき、 $\delta(0, 1, 0) = 1$ であるため必ず状態 1 のセルが 1 つ残る。したがって、 $\delta(0, 1, 0) = 1$ であればクラスタは不死となると決定できた。しかし、不完全な同期下のセルオートマトンでは図 3.15(b) のように、隣り合う 2 つの状態 1 のセルが $\delta(0, 1, 1) = 0$ と $\delta(1, 1, 0) = 0$ によってたまたま同期して状態遷移すると、例え $\delta(0, 1, 0) = 1$ であってもクラスタが死滅する可能性がある。

不完全な同期下のセルオートマトンの場合、クラスタが死滅するか、不死となるかの条件は以下のとおりである。

*4 これ以後、状態 1 のクラスタを単にクラスタとよぶことにする。

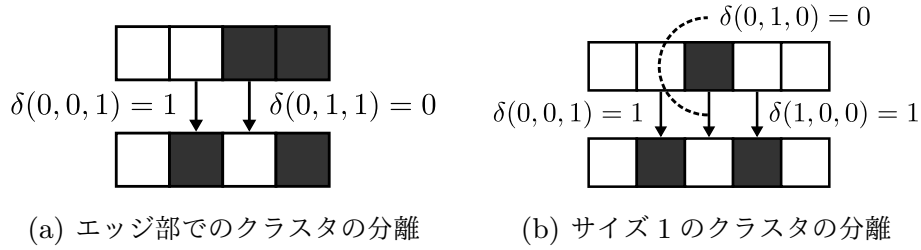


図 3.16: $\delta(1, 1, 1) = 1$ である場合のクラスタの分離

クラスタは死滅する可能性がある

$$\delta(0, 1, 0) = 0 \text{ or } (\delta(0, 1, 0) = 1 \text{ and } \delta(0, 1, 1) = 0 \text{ and } \delta(1, 1, 0) = 0)$$

クラスタは不死となる

$$\delta(0, 1, 0) = 1 \text{ and } (\delta(0, 1, 1) = 1 \text{ or } \delta(1, 1, 0) = 1)$$

$\delta(1, 1, 1) = 1$ である場合、非同期セルオートマトンではクラスタが分かれることはないで、その数が増えることはない。しかし、不完全な同期下のセルオートマトンでは、図 3.16 のようにクラスタのエッジ部またはサイズ 1 のクラスタの周辺で、局所的に同期して状態遷移が行われると、例え $\delta(1, 1, 1) = 1$ であっても、クラスタの数が増える場合がある。エッジ部でクラスタが分離する可能性のある条件は「 $\delta(0, 0, 1) = 1$ and $\delta(0, 1, 1) = 0$ 」またはその逆の「 $\delta(1, 0, 0) = 1$ and $\delta(1, 1, 0) = 0$ 」である。サイズ 1 のクラスタの周辺でクラスタの数が増える可能性のある条件は「 $\delta(0, 0, 1) = 1$ and $\delta(0, 1, 0) = 0$ and $\delta(1, 0, 0) = 1$ 」である。

不完全な同期下のセルオートマトンの場合、クラスタが分離する可能性があるか、ないかの条件は以下のとおりである。

クラスタは分離する可能性がある

$$\delta(1, 1, 1) = 0 \text{ or } \left(\begin{array}{l} \delta(1, 1, 1) = 1 \text{ and } (\delta(0, 0, 1) = 1 \text{ and } \delta(0, 1, 1) = 0) \text{ or} \\ \delta(1, 1, 1) = 1 \text{ and } (\delta(1, 0, 0) = 1 \text{ and } \delta(1, 1, 0) = 0) \text{ or} \\ \delta(1, 1, 1) = 1 \text{ and } (\delta(0, 0, 1) = 1 \text{ and } \delta(0, 1, 0) = 0 \text{ and } \delta(1, 0, 0) = 1) \end{array} \right)$$

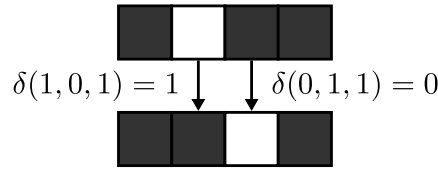


図 3.17: $\delta(1, 0, 0) = 0$ である場合のクラスタの右端の右方向への移動

クラスタが分離することはない

$$\delta(1, 1, 1) = 1 \text{ and } \left(\begin{array}{l} (\delta(0, 0, 1) = 0 \text{ or } \delta(0, 1, 1) = 1) \text{ and} \\ (\delta(1, 0, 0) = 0 \text{ or } \delta(1, 1, 0) = 1) \text{ and} \\ (\delta(0, 0, 1) = 0 \text{ or } \delta(0, 1, 0) = 1 \text{ or } \delta(1, 0, 0) = 0) \end{array} \right)$$

$\delta(1, 0, 0) = 0$ である場合，非同期セルオートマトンではクラスタの右端が右方向へ動くことはない．しかし，不完全な同期下のセルオートマトンでは，図 3.17 のようにクラスタの右端が右方向へ動く可能性がある．この条件は「 $\delta(1, 0, 1) = 1$ and $\delta(0, 1, 1) = 0$ 」である．

不完全な同期下のセルオートマトンの場合，状態 1-0 の境界が右へ動く可能性があるか，ないかの条件は以下のとおりである．

状態 1-0 の境界は右へ動く可能性がある

$$\delta(1, 0, 0) = 1 \text{ or } (\delta(1, 0, 0) = 0 \text{ and } \delta(1, 0, 1) = 1 \text{ and } \delta(0, 1, 1) = 0)$$

状態 1-0 の境界が右へ動くことはない

$$\delta(1, 0, 0) = 0 \text{ and } (\delta(1, 0, 1) = 0 \text{ or } \delta(0, 1, 1) = 1)$$

ほかの 5 つ $\delta(0, 0, 0)$, $\delta(0, 0, 1)$, $\delta(1, 1, 0)$, $\delta(0, 1, 1)$, $\delta(1, 0, 1)$ の値と時空間パターンの特徴との関係については，上で述べた 3 つのいずれかを 1, 0 反転，および左右反転させれば同様である．

3.4 非同期セルオートマトンの時空間パターンと差異が生じた要因

5つのグループのうち、グループ4及び5に属するセルオートマトンは、非同期セルオートマトンと異質な時空間パターンに収束するグループである。本節ではこの要因を明らかにする。

グループ4のセルオートマトン(#156)では、 $\delta(1, 1, 1) = 1$, $\delta(1, 0, 0) = 1$, $\delta(1, 1, 0) = 0$ である。 $\delta(1, 1, 1) = 1$ であるので、非同期セルオートマトンではクラスタが分かれることはない。しかし、 $\delta(1, 0, 0) = 1$, $\delta(1, 1, 0) = 0$ でもあるので、不完全な同期下のセルオートマトンでは図 3.18 のようにクラスタが分かれる場合がある。そのため、不完全な同期下のセルオートマトンと非同期セルオートマトンとで異なった時空間パターンに収束する。

グループ5のセルオートマトン(#6, #38, #134, #154^{*5})では、全て $\delta(0, 1, 0) = 1$, $\delta(0, 1, 1) = 0$, $\delta(1, 1, 0) = 0$ である。 $\delta(0, 1, 0) = 1$ であるので、非同期セルオートマトンではクラスタが死滅することはない。しかし、 $\delta(0, 1, 1) = 0$, $\delta(1, 1, 0) = 0$ でもあるので、不完全な同期下のセルオートマトンでは図 3.19 のようにクラスタが死滅する場合がある。さらに $\delta(0, 0, 0) = 0$ であるため、空間内の全てのクラスタが死に至ることで、同期セルオートマトン、非同期セルオートマトンとも異なる時空間パターンに収束する。

ただし、条件に合致したとしても、必ずしも最終的に異なる時空間パターンに収束するとは限らない。例えば図 3.20 に示した#152 の場合、 $\delta(1, 1, 1) = 1$, $\delta(1, 0, 0) = 1$, $\delta(1, 1, 0) = 0$ であるので、図 3.21 のようにクラスタが分離することがある。しかし、増えたクラスタがすぐに死滅してしまうため、結果的に非同期セルオートマトンと同様に全てのセルが同一の状態へと収束する。また、図 3.22 に示した#22 の場合、 $\delta(0, 1, 0) = 1$, $\delta(0, 1, 1) = 0$, $\delta(1, 1, 0) = 0$ であるので、図 3.23 のようにクラスタが死滅することがある。しかし、#22 ではクラスタが死滅する頻度よりクラスタが分離する頻度が高いため、

^{*5} 正確には#154の遷移関数を0, 1反転して得られる同値類の#166がこの条件を満たす。

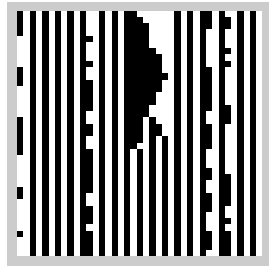
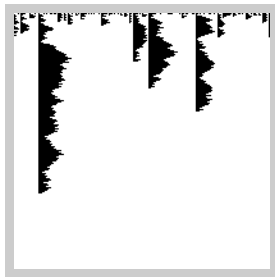


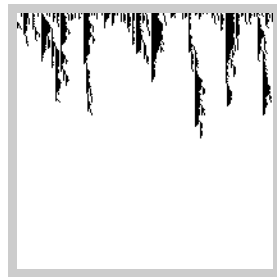
図 3.18: グループ 4 (#156) におけるクラスタの分離



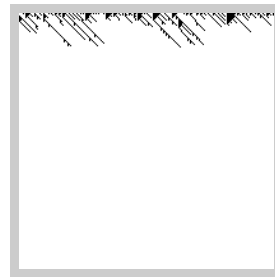
図 3.19: グループ 5 (#38) におけるクラスタの死滅



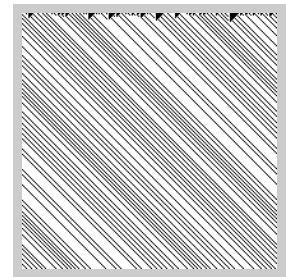
(a) 非同期 CA



(b) $p = 0.3$



(c) $p = 0.9$



(d) 同期 CA

不完全な同期下の CA

図 3.20: #152 の時空間パターン

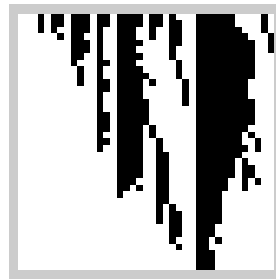
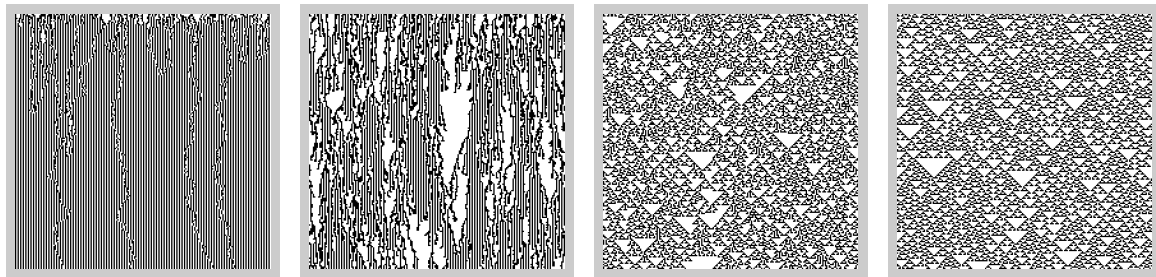


図 3.21: #152 で見られるクラスタの分離 ($p = 0.3$)

全てのクラスタが死に至ることなく，結果として遷移確率 p の値によって徐々に変化するパターンになる。



(a) 非同期 CA

(b) $p = 0.3$ (c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 3.22: #22 の時空間パターン

図 3.23: #22 で見られるクラスタの死滅 ($p = 0.3$)

3.5 不完全な同期下のセルオートマトンでクラスタが死滅するセルオートマトンの遷移関数に関する考察

グループ 5 のセルオートマトンは、非同期下ではクラスタが死滅しないが、不完全な同期下ではクラスタが死滅することがあるため、非同期下と不完全な同期下で異なる時空間パターンに収束していることがわかった。

しかし、グループ 5 以外に分類されたセルオートマトンでも、不完全な非同期下でクラスタが死滅する条件「 $\delta(0, 1, 0) = 1$ and $\delta(0, 1, 1) = 0$ and $\delta(1, 1, 0) = 0$ 」を満たすものがある。この条件を満たすもののうち、全てのクラスタが死に至るための規則 $\delta(0, 0, 0) = 0$ をもつようなセルオートマトンの $\delta(0, 1, 0)$, $\delta(0, 1, 1)$, $\delta(1, 1, 0)$, $\delta(0, 0, 0)$ 以外の遷移関

表 3.1: 不完全な同期下のセルオートマトンでクラスタが死滅する条件に合うセルオートマトンの遷移関数

CA	$\delta(1, 1, 1)$	$\delta(1, 0, 1)$	$\delta(1, 0, 0)$	$\delta(0, 0, 1)$	group
4	0	0	0	0	1
36	0	1	0	0	1
132	1	0	0	0	1
164	1	1	0	0	1
22	0	0	1	1	2
54	0	1	1	1	2
150	1	0	1	1	2
182	1	1	1	1	2
6	0	0	0	1	5
38	0	1	0	1	5
134	1	0	0	1	5
166	1	1	0	1	5

数の値を表 3.1 にまとめる*⁶.

表 3.1 を見ると $\delta(0, 0, 1)$ と $\delta(1, 0, 0)$ の 2 つの値が、クラスタが死滅するかどうか、すなわちセルオートマトンのグループ分けに影響を与えているように見える。

この $\delta(0, 0, 1)$ と $\delta(1, 0, 0)$ の値はクラスタの広がる方向に影響する。

「 $\delta(0, 0, 1) = 0$ and $\delta(1, 0, 0) = 0$ 」のときは、クラスタが外へ広がることがない。クラスタのサイズが大きくなることがないので、一度クラスタのサイズが 1 になると、このクラスタは死滅することがなくなる。

「 $\delta(0, 0, 1) = 1$ and $\delta(1, 0, 0) = 0$ 」または「 $\delta(0, 0, 1) = 0$ and $\delta(1, 0, 0) = 1$ 」のときは、クラスタが一方向（左または右）に広がる。また、「 $\delta(0, 0, 1) = 1$ and $\delta(1, 0, 0) = 1$ 」のときは、クラスタが両方向に広がる。どちらの場合も、一度クラスタのサイズが 1 となってもクラスタが広がってサイズが大きくなるため、クラスタが死滅する可能性がある。

*⁶ *5 で述べた理由により、ここでは #154 の代わりに #166 の遷移関数を載せた。同様に、#146 の遷移関数を 0, 1 反転して得られる #182 を載せた。

クラスタが広がるもののうち、一方向に広がるセルオートマトンのみがグループ 5 に分類され、クラスタが全て死滅する時空間パターンに収束している。この広がる方向と収束パターンの関係性について推測する。

単純に考えて、クラスタが全て死滅するのは、クラスタが分離して増加する頻度より、クラスタが死滅する頻度が高いためであると考えられる。様相が $\{0110\}$ *⁷ という死滅する可能性のある最小のクラスタが 1 つだけ存在する様相を仮定し、これがどのようなものかを考える。

クラスタが一方向に広がるものについては、右方向へ広がるものとする。すなわち「 $\delta(0,0,1) = 0$ and $\delta(1,0,0) = 1$ 」とする。このとき、様相 $\{0110\}$ の左端にある状態 0 は状態 0 のまま変わらないので、一方向については様相として $\{110\}$ を考える。各セルの状態遷移の有無の組み合わせは 8 通りとなる。またクラスタが両方向に広がる場合、様相 $\{0110\}$ において各セルの状態遷移の有無の組み合わせは 16 通りとなる。状態遷移の有無の組み合わせと、遷移後の様相を図 3.24 に示す。

クラスタが死滅する状態遷移の組み合わせは、一方向の場合は $[\bigcirc \bigcirc \times]$ *⁸、両方向性の場合は $[\times \bigcirc \bigcirc \times]$ で、両者とも 1 通りである。クラスタが分離する状態遷移の組み合わせは、一方向の場合は $[\times \bigcirc \bigcirc]$ の 1 通りで、両方向の場合は $[\times \times \bigcirc \bigcirc]$, $[\bigcirc \times \bigcirc \bigcirc]$, $[\bigcirc \bigcirc \times \times]$, $[\bigcirc \bigcirc \times \bigcirc]$, $[\bigcirc \bigcirc \bigcirc \bigcirc]$ の 5 通りである。遷移確率が p であるとき、一方向の場合、クラスタが死滅する確率は $p^2(1-p)$ となる。同様にクラスタが分離する確率は $p^2(1-p)$ となり、同確率で死滅と分離が発生する。両方向の場合、クラスタが死滅する確率は $p^2(1-p)^2$ となる。同様にクラスタが分離する確率は $p^2(1-p)^2 + p^3(1-p) + p^2(1-p)^2 + p^3(1-p) + p^4$ となり、クラスタが死滅する確率よりクラスタが分離し増加する確率が大幅に上回ることになる。このクラスタの死滅する状況下での死滅する確率と分離し増加する確率の差により、一方向にクラスタが拡張していくセルオートマトンでは全てのクラスタが死滅するが、両方向にクラスタが拡張するセルオートマトンでは全てのクラスタが死滅する時空間パターンには収束していないと考

*⁷ 様相の注目している一部分を取りだして、例えば $\{0110\}$ のように書く。様相の残りの部分は全て状態 0 とする。

*⁸ それぞれの対応する位置にあるセルについて、状態遷移の有無を \bigcirc と \times で表わしている。

遷移前の 様相	状態遷移の 有 (○) 無 (×)	遷移前の 様相 (一方向)	状態遷移の 有 (○) 無 (×)	遷移前の 様相 (両方向)
	×××		××××	
()	××○		×××○	
	×○×		××○×	
	×○○		××○○	
	○××		×○××	
	○×○		×○×○	
	○○×		×○○×	
	○○○		×○○○	
			○×××	
			○××○	
			○×○×	
			○×○○	
			○○××	
			○○×○	
			○○○×	
			○○○○	

図 3.24: 状態遷移の有無と遷移後の様相

えられる。このことが時空間パターンの違いとなって現れ、グループ 5 と 2 に分類される結果となったと推測できる。

3.6 Wolfram の分類との比較

本章における単純セルオートマトンの分類と Wolfram による分類にどのような違いがあるのか興味のあるところである。本節では、2 つの分類を比較する。本章における単純セルオートマトンの分類を表 3.2 にまとめ、表 3.3 に Wolfram による分類を示した。

グループ 3, 4, 5 は全てクラス 2 に含まれる。クラス 1 はグループ 1 に含まれる。また、クラス 3 は全てグループ 2 に含まれる。クラス 3 のセルオートマトンは元々無秩序な変動を繰り返しており、同期性にノイズが混入することによる影響は少ないので、グループ 2 に分類されたと考えられる。クラス 4 もグループ 2 に含まれる。グループ 1 はクラス 1 とクラス 2 のもののみからなる。その中でも、静止状態へ収束するクラス 2 のセルオートマトンは全てグループ 1 に含まれる。つまり、収束形態に変化がなくなる場合、同期性の変化とは無関係に同一形態へ収束しており、そのため、クラス 1 のセルオートマトンは全てグループ 1 に分類されたと考えられる。また、グループ 2 はクラス 2 とクラス 3, 4 のもののみからなる。クラス 2 には、全てのグループのものが含まれる。3.5 節で述べたような個々の遷移関数によって、形成される様相の特徴が大きく影響しているのではないかと考えられる。

3.7 まとめ

Fatès 等の定義した不完全な同期下のセルオートマトンの時空間パターンと、この両極に位置する同期セルオートマトンと非同期セルオートマトンの時空間パターンを比較することによって、不完全な同期下の単純セルオートマトンを 5 つのグループに分類した。

88 の単純セルオートマトンのうち、27, 37, 19, 1, 4 のセルオートマトンがそれぞれグループ 1, 2, 3, 4, 5 に分類された。

グループ 1 は、同期セルオートマトン、非同期セルオートマトンと同様な時空間パターンに収束するものである。同期性による影響を受けずに一定の時空間パターンを形成する。ノイズに一番強いグループといえる。

グループ 2 は、遷移確率 p の値の変化によって徐々に時空間パターンが変化するものである。同期性の変化が時空間パターンと密接に関係しているグループといえる。

グループ 3 は、非同期セルオートマトンと同様な時空間パターンに収束するものである。同期セルオートマトンは同期性に完全に依存した時空間パターンを形成している。わずかにノイズが混入しただけで、全く同期しない場合と同等となり、ノイズに最も脆弱なグループといえる。

グループ 4 は、同期セルオートマトンと同様な時空間パターンに収束するものである。グループ 3 と逆で、非同期セルオートマトンは完全な非同期性に依存した時空間パターンを形成している。局所的な同期によって発生するパターンが全体の時空間パターンにまで影響を与えているグループといえる。

グループ 5 は、同期セルオートマトンとも非同期セルオートマトンとも異なった時空間パターンに収束するものである。謂わば、グループ 3 の同期セルオートマトンと異なる性質とグループ 4 の非同期セルオートマトンと異なる性質を合わせたようなグループである。そのため、局所的な同期によって発生するパターンが全体の時空間パターンにまで影響を与えている。

また、金田が非同期セルオートマトンにおける時空間パターンの特徴が遷移関数から完全に決定できることを示しているが、本章では、不完全な同期下のセルオートマトンにおける遷移関数と時空間パターンの特徴との関係を明らかにした。

大多数のセルオートマトンはグループ 1, 2, 3 に属し、少数のセルオートマトンがグループ 4, 5 に属している。グループ 4, 5 の少数のセルオートマトンでは、非同期セルオートマトンと異なる時空間パターンに収束している。その要因を調べたところ、不完全な同期下で発生する局所的な同期によって、遷移関数と時空間パターンの特徴の関係が変化したためであった。

グループ 1, 2, 3 の中でも、局所的な同期によって非同期では起こりえないパターンが発生するセルオートマトンも多く含まれている。しかし、それらは全体の時空間パターンに決定的な影響を与えることなく、結果として非同期と同等の時空間パターンへと収束している。つまり、局所的な同期がミクロ的なパターンの相違を引き起こしても、その違いがマクロ的な時空間パターンを変えるまでに至ることは稀であると考えられる。

参考文献

- [1] Wolfram, S.: Cellular automata, *Los Alamos Science*, Vol. 9, pp. 2-27 (1983).
- [2] Wolfram, S.: Universality and complexity in cellular automata, *Physica D: Non-linear Phenomena*, Vol. 10, No. 1-2, pp. 1-35 (1984).

- [3] Wolfram, S.: Cellular automata as simple self-organizing systems, Technical report (1982).
- [4] Ingerson, T. E. and Buvel, R. L.: Structure in asynchronous cellular automata, *Physica D: Nonlinear Phenomena*, Vol. 10, No. 1, pp. 59–68 (1984).
- [5] Schönfisch, B. and Roos, de A.: Synchronous and asynchronous updating in cellular automata, *BioSystems*, Vol. 51, No. 3, pp. 123–143 (1999).
- [6] Cornforth, D., Green, D. G. and Newth, D.: Ordered asynchronous processes in multi-agent systems, *Physica D: Nonlinear Phenomena*, Vol. 204, No. 1, pp. 70–82 (2005).
- [7] Kanada, Y.: Asynchronous 1D cellular automata and the effects of fluctuation and randomness, in *Proceedings of the fourth conference on artificial life (A-Life IV)*. MIT Press (1994).
- [8] Fates, N. A. and Morvan, M.: An experimental study of robustness to asynchronism for elementary cellular automata, *Complex Systems*, Vol. 16, pp. 1–27 (2005).

第4章

雪の結晶の類似パターン生成

雪の結晶はどれをとっても美しい形をしているが、その形は千差万別で、全く同じ形のものも存在しないといわれている。雪の結晶成長に気温や湿度等の気象条件が影響を与えるが、成長過程で気象条件が微妙に変化し、さまざまな形の結晶ができるといわれている。

雪の結晶の成長過程を計算機上でシミュレートする研究は古くから行われているが、その多くはラプラス方程式の数値計算によるもの [1-6] や DLA (Diffusion-limited aggregation) 法によるもの [7,8] 等、物理現象を表す方程式の数値計算にもとづいている。

Packard [9] は、六角格子上的 2 次元セルオートマトンを用いた簡単なモデルで、角板とよばれている結晶の類似パターンを生成した。しかし、Packard のモデルでは、六角形の角から樹枝状に成長していく樹枝状六花や広幅六花、扇状六花とよばれる複雑な形をした結晶の類似パターンを生成することができない。

轡田等 [10] は、Packard のモデルを拡張し、Packard のモデルではできなかった樹枝状六花、広幅六花、扇状六花等の類似パターンを生成した。

雪の結晶には、成長過程で環境が大きく変化してできるといわれているさらに複雑な形をした結晶がある。また、一般的に知られている雪の結晶は六方対称な形をしているが、大気中の塵等が結晶に付着して空気抵抗が微妙に変わり結晶の各辺の成長速度が変化して、六方非対称な結晶ができる場合がある。

本章では、轡田等が定義した六角格子上的 2 次元セルオートマトンを拡張して、このよ

うな成長過程で環境が大きく変化してできるといわれている結晶と、六方非対称な結晶の類似パターン生成を試みる。

4.1 雪の結晶成長

雪の結晶にはさまざまな形のものがあるが、図 4.1 にその代表的な例を示す*1。角板，扇状六花，広幅六花，樹枝状六花は平面的な結晶であるのに対し，角柱，針は立体的な結晶である。

雪の結晶成長には，粒子の拡散過程（雲粒（液体の水分子）が蒸発して水蒸気（気体の水分子）になり空中へ拡散する過程で，結晶界面へ運ばれる水分子の数に影響する），潜

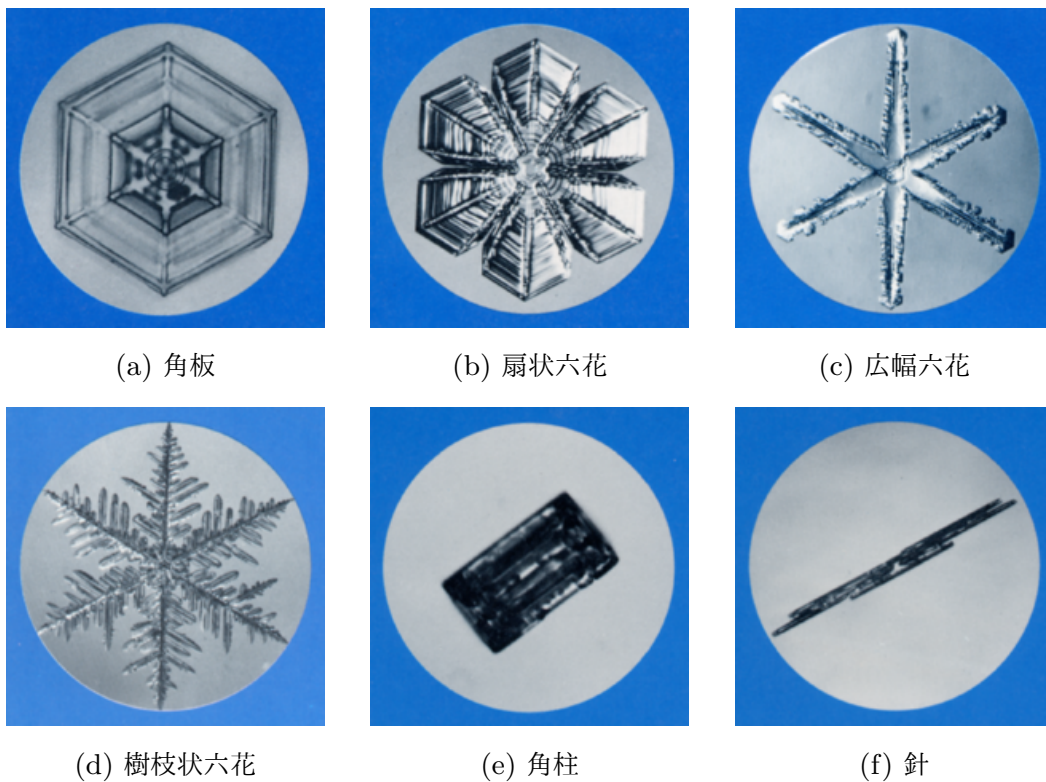


図 4.1: 雪の結晶の例

*1 本章で用いている雪の結晶の写真は，北海道大学図書刊行会のご好意により，文献 [11] から引用させて頂いたものである。

熱の拡散過程（潜熱，すなわち結晶化に伴う熱が結晶から空気中へ奪われていく過程で，結晶の付近の気温に影響する），表面拡散過程（結晶界面に運ばれてきた水分子が，結晶の表面を動いて規則正しい配列の中に組み込まれていく過程で，結晶の形に影響する）の 3 つの過程が関係している [12].

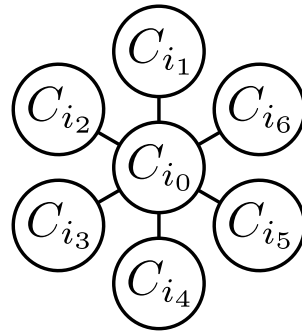
雲の中に何かのきっかけで水の粒ができると，その周りの雲粒（液体の水分子）は蒸発して水蒸気（気体の水分子）になる．水蒸気になって自由に飛び回ることができるようになった水分子が，氷の粒にぶつかって氷の粒に取り込まれ，結晶として成長していく．粒子の拡散過程は，雲粒が蒸発して水蒸気になり空中へ拡散する過程で，これによって結晶界面へ運ばれる水分子の数が影響を受ける．潜熱の拡散過程は，潜熱，すなわち結晶化に伴う熱が結晶から空気中へ奪われていく過程で，これによって結晶の付近の気温が影響を受ける．これらの 2 つの過程によって，結晶の成長速度が大きく左右される．表面拡散過程は，結晶界面に運ばれてきた水分子が，結晶の表面を動いて規則正しい配列の中に組み込まれていく過程で，これによって結晶の形が影響を受ける．

これらの過程は，気温や湿度等の気象条件と密接に関連するので，成長過程の気象条件の変化がこれらの過程に影響して，複雑な結晶が形造られる．

4.2 セルオートマトンを用いたモデル

通常，2次元セルオートマトンといえば正方格子上的セルオートマトンが一般的であるが，Packard [9] は，水の結晶（氷）が六方晶系に属すること，及び表面張力の異方性が六方対称性を持つことから，雪の結晶が六方対称性を持つようになることが知られている [12] ことを考慮して，六角格子上的 2次元セルオートマトンを新しく定義した．

六角格子上的 2次元セルオートマトンにおけるセルの近傍は 7 近傍である．例えば図 4.2 に示したセル C_{i_0} の近傍は，自分自身 C_{i_0} と自分の周りの 6 つのセル $C_{i_1}, C_{i_2}, \dots, C_{i_6}$ を含めた 7 つである．したがって，セル C_{i_0} の時刻 $t+1$ での状態 $s_{i_0}^{t+1}$ は，セル C_{i_0} の近傍の各セル $C_{i_0}, C_{i_1}, \dots, C_{i_6}$ の時刻 t での状態 $s_{i_0}^t, s_{i_1}^t, \dots, s_{i_6}^t$ から遷移関数 δ によって式 4.1 のように決められる．

図 4.2: セル C_{i_0} の近傍

$$s_{i_0}^{t+1} = \delta(s_{i_0}^t, s_{i_1}^t, \dots, s_{i_6}^t) \quad (4.1)$$

Packard は，水蒸気に対応する状態‘0’と氷に対応する状態‘1’の 2 状態で外総和型のセルオートマトンを用いて，角板とよばれる雪の結晶の類似パターンの生成に成功した。しかし，Packard のモデルでは 32 通りの遷移関数が考えられるが，角板以外の結晶の類似パターンは生成できなかった。

轡田等 [10] は，Packard のモデルを拡張し，Packard のモデルではできなかった樹枝状六花，広幅六花，扇状六花等の類似パターンを生成した。

轡田のモデルで拡張した点は次の 2 点*2である。1 つは，各セルの状態を，水蒸気に対応する状態‘0’，水蒸気と氷の間に対応する状態‘1’，氷に対応する状態‘2’の 3 状態に拡張した点である。もう 1 つは，外総和性を放棄した点である。すなわち，轡田のモデルでは，セル C_{i_0} の次の時刻での状態が，周囲のセルの状態の位置も考慮して決定される。

ただし，セル C_{i_0} の状態が‘2’のときは，次の時刻での状態も‘2’である。すなわち，一旦氷になったセルは氷であり続ける。状態‘2’が状態‘1’または‘0’に変化する，あるいは状態‘1’が状態‘0’に変化することは，物理的には融解や昇華に対応するものと考えられる。Packard のモデルでは融解や昇華は考慮していないので，轡田のモデルでも考慮していない。

*2 それぞれ各 1 点ずつ拡張してもうまく類似パターンを生成できなかったため，2 点同時に拡張している。

4.3 環境の変化によって生じる複雑な形をした雪の結晶

雪の結晶の形に主に影響を与えるのは気温と湿度で、気温と湿度がほぼ一定のときには雪の結晶はある程度決まった形に成長することが知られている。図 4.3 は気温と湿度（正確には過剰水蒸気密度）と雪の結晶の形との関係を示した図である [12]。図 4.1 に示した 6 つの結晶は、一定した環境で成長した代表的な結晶である。しかし、実際の雪の結晶が成長する過程では、成長途中で環境が大きく変化することがあり、そのようなときには結晶はより複雑な形に成長を遂げる。

図 4.4 に示した雪の結晶は、成長途中で環境が大きく変化して成長したと考えられている。図 4.4(a) と図 4.4(b) の結晶は、初めは角板に成長する環境で成長したが、途中で湿度が上がり、樹枝状六花あるいは広幅六花に成長した結晶であると考えられている。図 4.4(c) の結晶は、初めは角板に成長する環境で成長したが、途中で湿度が上がり樹枝状六花に成長した。その後、今度は湿度が下がり樹枝の先が角板に成長した結晶であると考えられている。

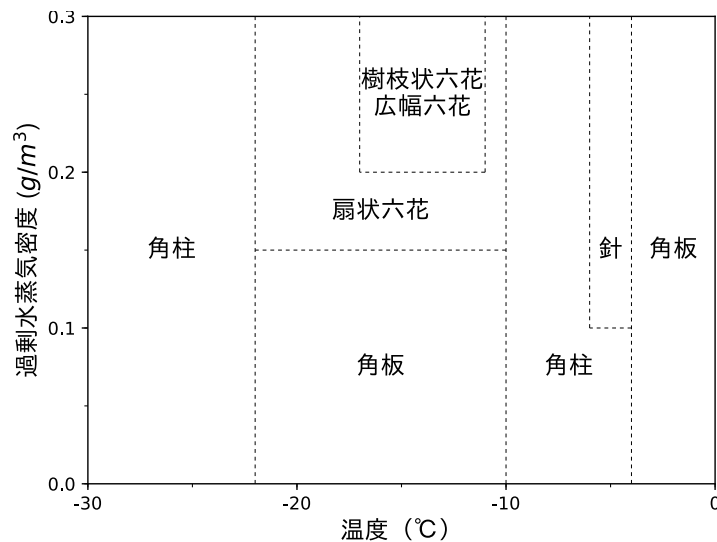


図 4.3: 雪の結晶の形と環境の関係

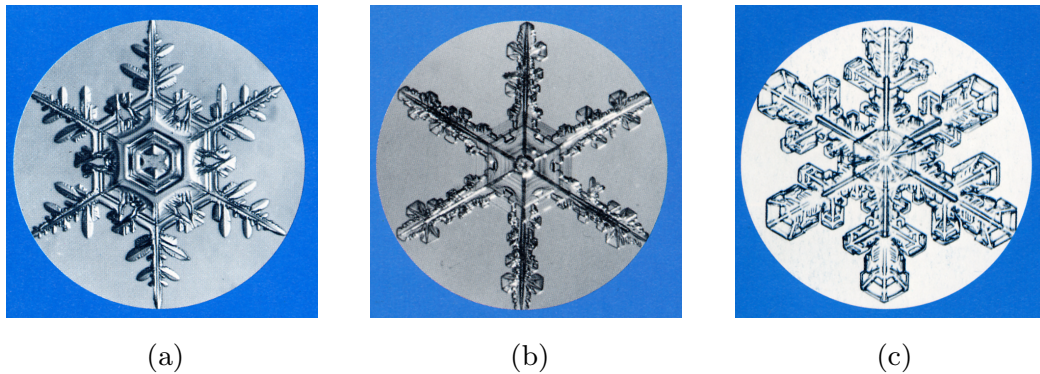


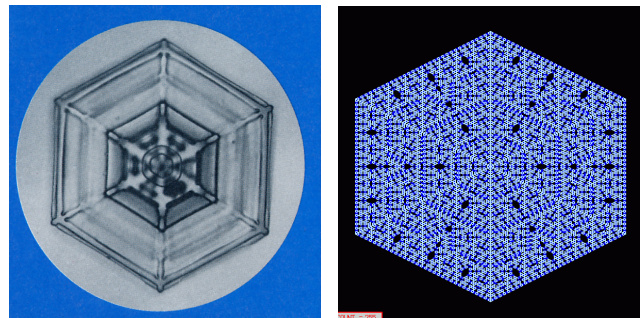
図 4.4: 環境の変化によって生じる複雑な形をした雪の結晶の例

4.4 環境の変化によって生じる複雑な形をした雪の結晶の類似パターンの生成

轡田のモデルではおおよそ $3^{3^7} \approx 2.9 \times 10^{1043}$ 通りの遷移関数が考えられるので、到底全てを試すことはできないが、多くの実験を行っても図 4.4 に示したような結晶の類似パターンを生成することはできなかった。

そこで、成長過程での環境の大きな変化をセルオートマトンの遷移関数の入れ換えに対応させることを考える。既に轡田のモデルで結晶の類似パターンがいくつか生成されているので、この遷移関数の 1 つを用いてパターンを生成し、途中で別のもう 1 つの遷移関数に入れ換えて、成長過程で環境が大きく変化してできる雪の結晶の類似パターンの生成を行う。

図 4.5, 図 4.6, 図 4.7 は轡田のモデルによって生成された類似パターンの例である。図 4.5 は角板, 図 4.6 は樹枝状六花, 図 4.7 は広幅六花に対応する。それぞれ、左に示したのは雪の結晶, 右に示したのは類似パターンである。また下に示したのは、遷移関数のうち状態 '0' が状態 '1' または状態 '2' に, 状態 '1' が状態 '2' に成長するルールである。明記されていない残りのルールは、全て状態が変化しないルールである。ここでは簡単のため、 $\delta(s_{i_0}^t, s_{i_1}^t, \dots, s_{i_6}^t) = s_{i_0}^{t+1}$ であることを、単に $s_{i_0}^t s_{i_2}^t \dots s_{i_6}^t, s_{i_0}^{t+1}$ と記述している。



rule					
0000002,1	0000112,1	0000121,2	0000211,1	0001001,1	0001002,1
0001011,1	0001012,2	0001101,1	0001221,1	0002101,2	0111111,1
0111112,1	0111122,1	0111212,1	0111222,1	0112112,1	0112122,1
0112212,1	0112222,1	0121212,1	0121222,1	0122122,1	0122222,1
0222222,1	1000002,2	1000012,2	1000021,2	1000121,2	1000122,2
1000202,2	1000221,2	1002122,2	1002212,2	1010122,2	1010221,2
1012221,2					

図 4.5: 轡田のモデルによる類似パターンの例 (角板)

図 4.8 に結果を示す. いずれも, 最初特定の 1 つのセルだけが状態 ‘2’ で, 残り全てのセルが状態 ‘0’ である初期様相 (水蒸気の中に氷の粒が 1 つだけ存在する様相) からスタートしている.

図 4.8(a) は, 最初図 4.5 の角板に分類される類似パターンを生成する遷移関数を 96 ステップ適用し, その後図 4.6 の樹枝状六花に分類される類似パターンを生成する遷移関数に入れ換えて 290 ステップまで適用した例である. 図 4.4(a) の結晶とよく似たパターンが得られている.

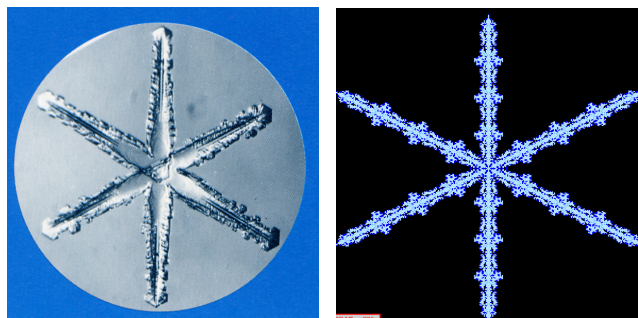
図 4.8(b) は, 最初図 4.5 の角板に分類される類似パターンを生成する遷移関数を 112 ステップ適用し, その後図 4.7 の広幅六花に分類される類似パターンを生成する遷移関数に入れ換えて 290 ステップまで適用した例である. 図 4.4(b) の結晶とよく似たパターンが得られている.



rule

0000002,1	0000102,2	0000111,1	0000112,2	0000121,1	0000122,2
0000201,2	0000211,2	0000221,2	0000222,1	0001001,1	0111111,1
0111112,2	0111122,1	0111212,1	0111222,1	0112112,1	0112122,1
0112212,1	0112222,1	0121212,1	0121222,1	0122122,1	0122222,1
0222222,1	1000002,2	1000121,2	1000222,2	1001221,2	1011202,2
1011221,2	1012211,2	1020211,2	1022122,2		

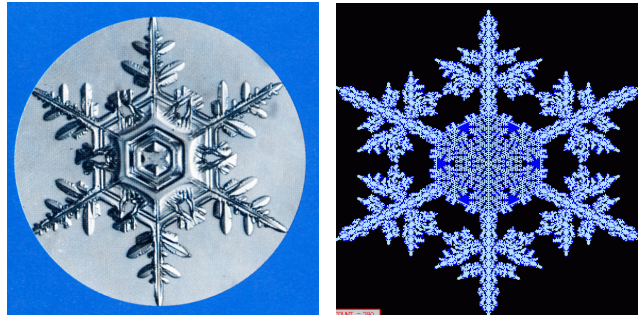
図 4.6: 轡田のモデルによる類似パターンの例 (樹枝状六花)



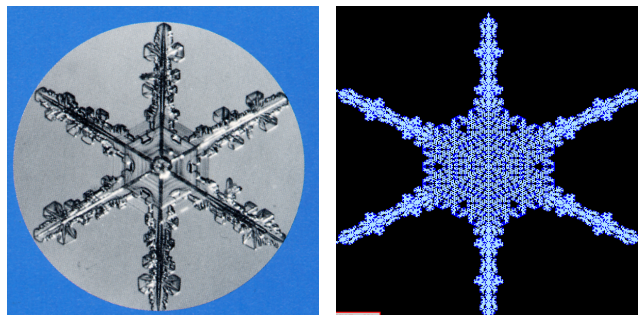
rule

0000002,1	0000102,2	0000111,1	0000112,2	0000121,1	0000122,2
0000201,2	0000211,2	0000221,2	0000222,1	0001001,1	0001121,1
0001211,1	0111111,1	0111112,1	0111122,1	0111212,1	0111222,1
0112112,1	0112122,1	0112212,1	0112222,1	0121212,1	0121222,1
0122122,1	0122222,1	0222222,1	1000002,2	1000121,2	1000222,2
1001221,2	1011202,2	1020211,2	1022122,2		

図 4.7: 轡田のモデルによる類似パターンの例 (広幅六花)



(a) 図 4.4(a) の結晶とその類似パターン



(b) 図 4.4(b) の結晶とその類似パターン

図 4.8: 提案モデルによる類似パターンの例

4.5 六方非対称な雪の結晶

雪の結晶は一般的に六方対称性を持つことが知られているが、六方非対称な結晶ができる場合がある。

図 4.9 に六方非対称な雪の結晶の例を示す^{*3}。図 4.9(a) の結晶は、角板に成長する環境で成長したが、途中で大気中の塵等が付着して空気抵抗が微妙に変わり結晶の各辺の成長速度が変化して成長したと考えられている。図 4.9(b) の結晶は、図 4.9(a) のように成長した結晶が、気温や湿度等が大きく変化せずにさらに成長を続けて三角形になったと考えられている。また、図 4.9(c) の結晶は、図 4.9(a) のように成長した結晶が、その後気温や湿度等が何度も大きく変化して成長したと考えられている [13]。

^{*3} 図 4.9 の 3 つの写真は、文献 [13] から引用したものである。

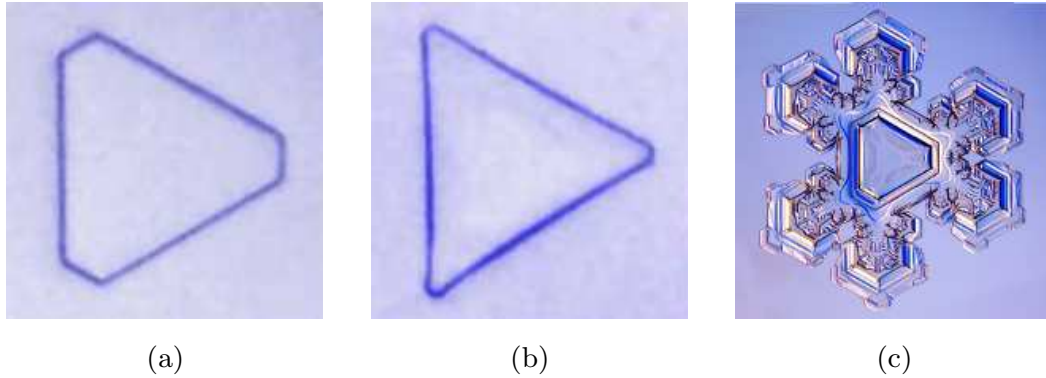


図 4.9: 六方非対称な雪の結晶

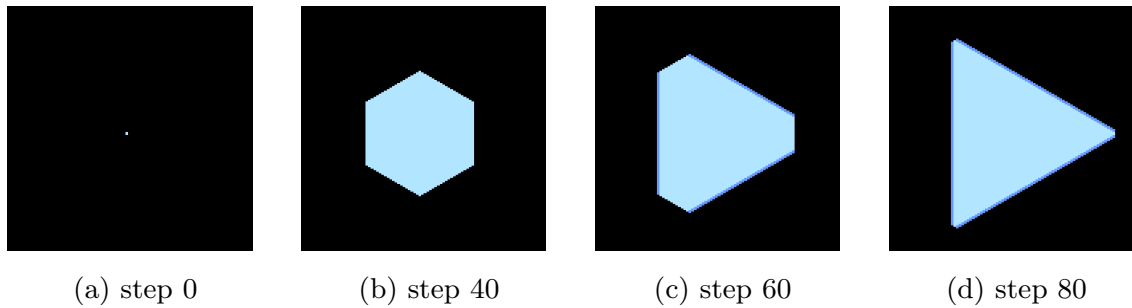
4.6 六方非対称な雪の結晶の類似パターン生成

従来の 2 次元セルオートマトンを用いたどのモデルも六方対称性を前提としているため、周囲のセルの状態が回転して同じである場合には、次の時刻で必ず同じ状態に遷移する。すなわち、全ての状態 $s_{i_0}^t, s_{i_1}^t, s_{i_2}^t, s_{i_3}^t, s_{i_4}^t, s_{i_5}^t, s_{i_6}^t$ について、以下の式 4.2 が成り立つことを前提としている。

$$\begin{aligned}
 \delta(s_{i_0}^t, s_{i_1}^t, s_{i_2}^t, s_{i_3}^t, s_{i_4}^t, s_{i_5}^t, s_{i_6}^t) &= \delta(s_{i_0}^t, s_{i_2}^t, s_{i_3}^t, s_{i_4}^t, s_{i_5}^t, s_{i_6}^t, s_{i_1}^t) \\
 &= \delta(s_{i_0}^t, s_{i_3}^t, s_{i_4}^t, s_{i_5}^t, s_{i_6}^t, s_{i_1}^t, s_{i_2}^t) \\
 &= \delta(s_{i_0}^t, s_{i_4}^t, s_{i_5}^t, s_{i_6}^t, s_{i_1}^t, s_{i_2}^t, s_{i_3}^t) \\
 &= \delta(s_{i_0}^t, s_{i_5}^t, s_{i_6}^t, s_{i_1}^t, s_{i_2}^t, s_{i_3}^t, s_{i_4}^t) \\
 &= \delta(s_{i_0}^t, s_{i_6}^t, s_{i_1}^t, s_{i_2}^t, s_{i_3}^t, s_{i_4}^t, s_{i_5}^t)
 \end{aligned} \tag{4.2}$$

そこで、式 4.2 の前提を外した上で、環境の変化を考慮したモデルを用いて、六方非対称な雪の結晶の類似パターン生成を試みた。

樽田のモデルにおいて、遷移関数の数は $3^{3^7} \approx 2.9 \times 10^{1043}$ 通りあり、環境の変化を考慮したモデルではさらに、環境の変化に対応してこの遷移関数を入れ替えるので、その組み合わせの数は無数にあり、全てを尽くすことは不可能である。そこで、角板、樹枝状六花、広幅六花、扇状六花に分類される類似パターンを生成することが知られている遷移関数を用いて、図 4.9(a)、図 4.9(b)、図 4.9(c)、の結晶について、それらが成長したとい



rule 1

0000002,1 0000022,1 1000121,2 1001221,2

rule 2

0000002,1 0000012,1 0000021,1 0000022,1

1012210,2 1100122,2 1221001,2

図 4.10: 図 4.9(a), (b) の類似パターンと遷移関数

われている環境に対応する遷移関数に入れ替えて類似パターンを生成できないか試した。図 4.10, 図 4.11 に生成結果を示す。

いずれも、中央にある特定の 1 つのセルだけが状態 ‘2’ で、残り全てのセルが状態 ‘0’ である初期様相（水蒸気の中に氷の粒が 1 つだけ存在する様相）からスタートしている。

また、図の下部に示したのは、遷移規則のうち状態 ‘0’ が状態 ‘1’ または状態 ‘2’ に、状態 ‘1’ が状態 ‘2’ に変化する遷移規則である。明記されていない残りの遷移規則は、全て状態が変化しない遷移規則である。下線を引いた遷移規則は六方非対称な遷移規則で、残りは六方対称な遷移規則である*4。

図 4.10 は、角板に分類される類似パターンを生成する遷移関数（図 4.10 rule 1）を 40 ステップまで適用し、その後に六方非対称な遷移規則を含む遷移関数（図 4.10 rule

*4 六方対称な遷移規則については、その中の 1 つのみ記述している。例えば、遷移規則 “0000002,1” は、6 つの遷移規則 “0000002,1”, “0000020,1”, “0000200,1”, “0002000,1”, “0020000,1”, “0200000,1” を表していることに注意されたい。

2)*⁵に入れ換えて 80 ステップまで適用した結果である。

図 4.10 rule 1 は、角板に分類される類似パターンを生成する遷移関数の中で最も基本的なものである。“0000002,1”, “1000121,2” は、初期様相から結晶が成長するために最低限必要な遷移規則である。また、“0000022,1”, “1001221,2” は、“0000002,1”, “1000121,2” と合わせて角板の 6 つの辺を成長させる遷移規則である。“0000002,1” と “0000022,1” とで、角板の周りを取り囲む状態 ‘0’ のセルが状態 ‘1’ に遷移し、次に “1000121,2” と “1001221,2” とで、それらが状態 ‘2’ に遷移して、角板が成長する。

図 4.10 rule 2 では、図 4.10 rule 1 と同様に、“0000002,1” と “0000022,1” とで、角板の周りを取り囲む状態 ‘0’ のセルが状態 ‘1’ に遷移するが、“1001221,2” に代わって六方非対称な遷移規則 “1012210,2”, “1221001,2”, “1100122,2” があるので左上, 左下, 右の辺上の状態 ‘1’ のセルのみが状態 ‘2’ に遷移する。さらに “0000012,1”, “0000021,1” と合わせて、その 3 方に成長する。

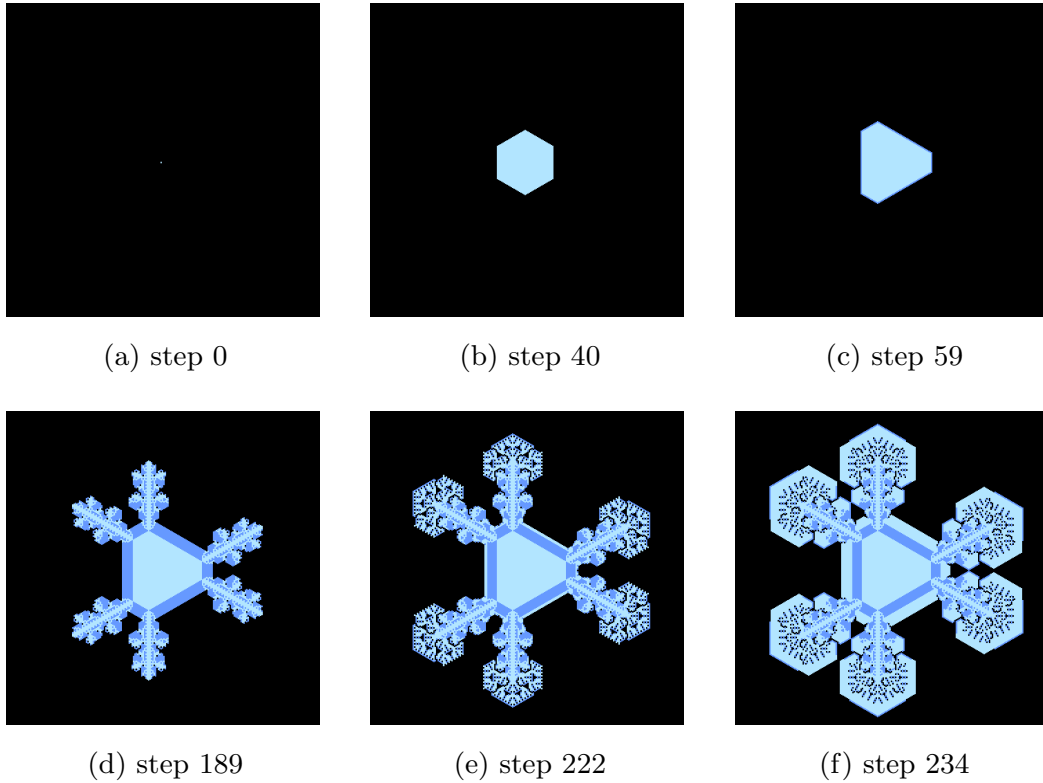
60 ステップで図 4.9(a), 80 ステップで図 4.9(b) によく似た六方非対称な雪の結晶の類似パターンが得られた。

図 4.11 は、図 4.10 と同様に、図 4.10 rule 1 の遷移関数を 40 ステップまで、図 4.10 rule 2 の遷移関数を 59 ステップまで適用した後、広幅六花の類似パターンを生成する遷移関数 (図 4.11 rule 3) を 189 ステップまで、扇状六花の類似パターンを生成する遷移関数 (図 4.11 rule 4) を 222 ステップまで、角板の類似パターンを生成する遷移関数 (図 4.11 rule 5) を 234 ステップまでそれぞれ順に適用した結果である。

図 4.9(c) の結晶は、図 4.9(a) のように成長した結晶が、その後気温や湿度等が何度も大きく変化して成長したと考えられているが、気温や湿度等がどのように何度大きく変化したのかわからないので、遷移関数をいろいろと入れ替えて試行錯誤を繰り返した。

図 4.9(c) に非常によく似た六方非対称な雪の結晶の類似パターンが得られた。

*⁵ 下線を引いた遷移規則 “1012210,2”, “1221001,2”, “1100122,2” は、ここには明記されていない遷移規則 “1122100,1”, “1210012,1”, “1001221,1” と合わせて考えてみると六方非対称な遷移規則であることが理解されよう。いずれも、状態 ‘1’ であるセルを、状態 ‘0’, ‘1’, ‘2’, ‘2’, ‘1’, ‘0’ であるセルが回りを囲った状況にある。3 つの場合には状態 ‘2’ となり結晶は成長するが、残り 3 つの場合には状態 ‘1’ のままで結晶が成長しないことに注意されたい。



rule 3					
0000002,1	0000101,1	0000111,1	0000112,2	0000121,1	0000122,1
0000211,2	0000221,1	0001111,1	0111122,2	1000121,2	1000122,2
1000221,2	1001122,2	1002211,2			
rule 4					
0000002,1	0000122,1	0000221,1	0000222,1	0001222,2	0002221,2
1000002,2	1000012,2	1000021,2	1000121,2	1001112,2	1001221,2
1001222,2	1002111,2	1002112,2	1002221,2		
rule 5					
0000002,1	0002201,2	1000021,2	1001221,2	0000022,1	0001221,1
1000121,2	1001222,2	0000122,1	0012221,2	1000122,2	1002221,2
0000221,1	0022222,2	1000221,2	1010122,2	0000222,2	0222222,2
1001021,2	1010221,2	0001022,2	1000012,2	1001201,2	1022222,2

図 4.11: 図 4.9(c) の類似パターンと遷移関数

4.7 まとめ

本章では、轡田のモデルを用いて、雪の結晶の成長過程において環境が大きく変化してできるといわれている複雑な形をした結晶の類似パターンの生成を試みた。環境の大きな変化をセルオートマトンの遷移関数の入れ換えに対応させ、2つの例ではあるがうまく類似パターンを生成することができた。

また、環境の変化を考慮したモデルを六方非対称な雪の結晶の類似パターン生成に適用した。本章で得られた結果は、ほんの一例であり、ほかにも六方非対称な結晶の類似パターンを生成させる規則が存在すると考えられる。

気象環境の変化や結晶が受ける微妙な空気抵抗の変化を、数値計算を用いた手法に導入することは難しいが、セルオートマトンを用いたモデルであれば容易である。しかしながら、セルオートマトンの遷移関数の物理的な意味が全く明確になっていないことが問題であり、本当の意味でのシミュレーションとはいえない。ただ、単なる類似パターンの生成ではあるが、いろいろな雪の結晶について、その結晶ができるといわれている環境の変化に対応して、そのそれぞれの環境に対応する遷移関数に入れ替えてシミュレーションするとその結晶によく似た類似パターンが実際に生成されることは非常に興味深い。

参考文献

- [1] 黒田登志雄, 横山悦郎: 結晶の形態形成 拡散と表面カインेटイクスの寄与, 日本結晶成長学会誌, Vol. 15, No. 1, pp. 76–79 (1988).
- [2] 黒田登志雄: 自然の美 雪結晶の成長と形態形成の秘密, メカライフ, No. 17, pp. 10–13 (1989).
- [3] 黒田登志雄, 横山悦郎: 雪の形態形成および氷の表面融解現象, 日本物理学会誌, Vol. 45, No. 8, pp. 541–548 (1990).
- [4] 覚井真吾: 雪の結晶成長のシミュレーション (ランダムなフラクタル・パターンの成長機構と統計, 研究会報告), 物性研究, Vol. 54, No. 1, pp. 17–23 (1990).

- [5] 横山悦郎：特集デンドライト雪の結晶成長表面カイネティクスと拡散場に支配された樹枝状パターンの形成機構, *バウンダリー*, Vol. 6, No. 2, pp. 12–16 (1990).
- [6] 古川義純：雪の結晶成長と表面構造, *表面科学*, Vol. 16, No. 10, pp. 651–654 (1995).
- [7] Matsushita, M. and Kondo, H.: Diffusion-limited aggregation with tunable lattice anisotropy, *Journal of the Physical Society of Japan*, Vol. 55, No. 8, pp. 2483–2486 (1986).
- [8] Miyazima, S. and Tanaka, T.: Simulation for growth of snowflake, *Journal of the Physical Society of Japan*, Vol. 56, No. 2, pp. 441–443 (1987).
- [9] Packard, N. H.: Lattice models for solidification and aggregation, *First International Symposium for Science on Form* (1986).
- [10] 轡田康, 広瀬貞樹, 蛭川繁, 木村春彦：セルオートマトンを用いた雪の結晶の類似パターン生成, *電子情報通信学会論文誌 D*, Vol. 83, No. 9, pp. 909–918 (2000).
- [11] 小林禎作：雪の結晶—冬のエフェメラル, 北海道大学図書刊行会 (1983).
- [12] 小林禎作：雪はなぜ六角か, 筑摩書房 (1984).
- [13] Libbrecht, K. and Arnold, H.: Aerodynamic stability and the growth of triangular snow crystals, *The Microscope*, Vol. 57, No. 4, pp. 157–163 (2009).

第 5 章

一斉射撃問題の拡張

一斉射撃問題は、セルオートマトンに関する有名な問題の 1 つであり、1957 年頃に Myhill によって提起され、Moore によって広く知らされた問題である [1]。最初にこの問題の解を示したのは Minsky と McCarthy で、 n 個のセルからなる 1 次元セル列に対して状態数 13, $3n + \mathcal{O}(\log n)$ ステップで射撃状態に遷移させるアルゴリズムを示した [2]。この問題の解としては少なくとも $2(n-1)$ ステップ必要であることは容易に示されるが、実際に Goto によって、 $2(n-1)$ ステップで射撃状態に遷移させる最小時間のアルゴリズムが示された [3]。Goto のアルゴリズムは状態数が数千にも及ぶ怪物的なものであったので、その後のより簡単な解を求める研究を促進させた。1966 年に Waksman が状態数 16, 1967 年に Balzer が状態数 8, 1987 年に Gerken が状態数 7, Mazoyer が状態数 6 のアルゴリズムをそれぞれ示している [4–7]。また、Balzer が一斉射撃問題の解として少なくとも 5 状態必要であることを示している [5]。

一斉射撃問題を拡張した研究も数多く行われている。セル列の両端を連結して環状にした系に対する問題 [8]、2 次元や 3 次元の空間にセルを配列した系に対する問題 [9, 10]、将軍が複数いる問題 [11–13]、Cayley グラフに対する問題 [14]、特殊な種類のグラフに対する問題 [15]、可逆型（時間を逆向きにしても決定的である）セルオートマトン上の問題 [16]、number-conserving な（状態を非負整数で表し、その総和が一定である）セルオートマトン上の問題 [17]、任意の隣接したセルの間の情報交換の量を制限した問題 [7, 18] な

どがある。

本章では、理論的興味から一斉射撃問題を拡張し、7 セグメントディスプレイ上の数字パターンの一斉射撃問題、16 セグメントディスプレイ上の英字パターンの一斉射撃問題、副将軍のいる一斉射撃問題を提起する。また、これらの問題に対してそれぞれ一斉射撃アルゴリズムを与える。

5.1 一斉射撃問題の定義

一斉射撃問題は、3 近傍 1 次元セルオートマトン上で定義される以下のような問題である。 n 個のセル C_0, C_1, \dots, C_{n-1} からなる 1 次元セル列を考える。セルの個数 n は任意の自然数であるが有限とする。左端のセル C_0 は将軍とよばれ、残りのセル C_1, C_2, \dots, C_{n-1} は兵士とよばれる。時刻 $t = 0$ で将軍だけは将軍状態とよばれる状態で、兵士は静止状態とよばれる状態である。各セルの動作は同期していて、時刻 $t + 1$ の状態は、時刻 t での自身の状態と隣接するセルの状態から、遷移関数に従って決められる。ただし、自身も隣接するセルも静止状態である場合には、次の状態も静止状態である。このとき、ある時刻に全てのセルを一斉に射撃状態とよばれる状態に遷移させることができるように、セルオートマトンの状態と遷移関数を定めよという問題である*1。

より詳しい定義は以下のとおりである。各セル C_i ($0 \leq i \leq n - 1$) の有限オートマトンを $\mathcal{A} = (\mathcal{S}, \delta)$ とする。 \mathcal{S} は状態の有限集合であり、将軍状態 G 、静止状態 Q 、射撃状態 F の相異なる 3 つの状態を少なくとも含んでいる。 δ は遷移関数であり、 $\delta: (\mathcal{S} \cup \{B\}) \times \mathcal{S} \times (\mathcal{S} \cup \{B\}) \rightarrow \mathcal{S}$ で定義される。 $B \notin \mathcal{S}$ はセル列の境界を示す信号である。静止状態 Q に対して、 $\delta(s_Q, Q, s_Q) = Q$ と定義する。ここで、 $s_Q \in \{Q, B\}$ である。時刻 t のセル C_i ($0 \leq i \leq n - 1$) の状態を s_i^t で示す。セル C_i ($0 \leq i \leq n - 1$) の初

*1 簡単にいえば『 C_0 を将軍、 C_1, C_2, \dots, C_{n-1} を兵士とし、情報はそれぞれ隣同士でしかやりとりできない状況とする。将軍が時刻 $t = 0$ に「一斉に射撃せよ！」という命令を出したとき、どのようにすれば一斉に射撃できるか。』という問題である。

期状態 s_i^0 は式 5.1 で定義される.

$$s_i^0 = \begin{cases} G & \text{if } i = 0 \\ Q & \text{otherwise} \end{cases} \quad (5.1)$$

時刻 $t + 1$ のセル C_i の状態 s_i^{t+1} は, 近傍セルの状態と遷移関数によって式 5.2 のように決まる. ただし, $s_{-1}^t = B$, $s_n^t = B$ である.

$$s_i^{t+1} = \delta(s_{i-1}^t, s_i^t, s_{i+1}^t) \quad (5.2)$$

時刻 t_F で全てのセルが初めて射撃状態 F に遷移したとき, 有限オートマトン \mathcal{A} が長さ n のセル列を一斉射撃したという. このとき, 各セル C_i ($0 \leq i \leq n - 1$) の状態は式 5.3 で定義される.

$$s_i^{t_F} = F \text{ and } s_i^t \neq F \text{ for all } 0 \leq t < t_F \quad (5.3)$$

有限オートマトン \mathcal{A} が長さ n のセル列を一斉射撃したとき, この射撃時刻 t_F を $t_F(n, \mathcal{A})$ と表す.

有限オートマトン \mathcal{A} が全ての長さ n のセル列を一斉射撃するとき, この \mathcal{A} を一斉射撃問題の解とよぶ. 長さ n のセル列の解の最小時間 $t_{F \min}(n)$ を式 5.4 で示す.

$$t_{F \min}(n) = \min\{t_F(n, \mathcal{A}) \mid \mathcal{A} \text{ は一斉射撃問題の解である}\} \quad (5.4)$$

全ての n に対して $t_F(n, \mathcal{A}) = t_{F \min}(n)$ であるとき, この解 \mathcal{A} を最小時間解とよぶ.

5.2 最小時間アルゴリズム

一斉射撃問題に対しては多くのアルゴリズムが知られており, 最小時間アルゴリズムも既にいくつか知られている. ここでは, わかりやすい Waksman の最小時間アルゴリズム [4] の考え方を簡単に説明する.

時刻 $t = 0$ に将軍 C_0 は速度 $\frac{1}{1}, \frac{1}{3}, \frac{1}{7}, \dots, \frac{1}{2^k - 1}, \dots$ (k は $1 \leq k \leq \lfloor \log_2(2n - 2) \rfloor$ を満たす自然数) の信号を右方向に送出する. これら一連の信号を射撃信号とよぶ. 信号

とは、ある特定の状態（情報）がセル列を移動することを指している。分数で信号の速度を表し、 t ステップに 1 セル伝播する信号を速度 $\frac{1}{t}$ の信号という。時刻 $t = n - 1$ に速度 $\frac{1}{1}$ の信号がセル C_{n-1} に到達し、 C_{n-1} は将軍状態に遷移する。将軍状態になったセル C_{n-1} は C_0 と同様の射撃信号を左方向に送出する。時刻 $t = 0$ に C_0 が送出した速度 $\frac{1}{3}$ の信号と時刻 $t = n - 1$ に C_{n-1} が送出した速度 $\frac{1}{1}$ の信号が C_0 と C_{n-1} の中間で衝突する。信号が衝突した場所にあるセルは将軍状態に遷移する。将軍状態になったこのセルは射撃信号を左方向と右方向に送出する。このように信号が将軍状態の中間で衝突し、その場所のセルが将軍状態に遷移することを繰り返すことで、セル空間を再帰的に均等に分割することができる。均等に分割していくといずれ全てのセルが将軍状態に遷移する。各セルは、全てのセルが将軍状態になったかどうか、自身と隣接するセルが将軍状態になったことでわかるので、その次のステップに一斉に射撃状態に遷移することができる。

このような一連の処理にかかる時間は、速度 $\frac{1}{1}$ の信号が 1 次元セル列を往復する時間に等しく、 $2(n - 1)$ ステップであることがわかる。

図 5.1 は Waksman の最小時間アルゴリズムの時間空間図式で、セル列 C_0, C_1, \dots, C_{n-1} が状態遷移する過程を上から下へと時系列順に並べたものである。横軸はセル空間、縦軸は時間を表し、図中の分数は射撃信号の速度を表している。黒い点はそのセルが将軍状態になったことを示している。

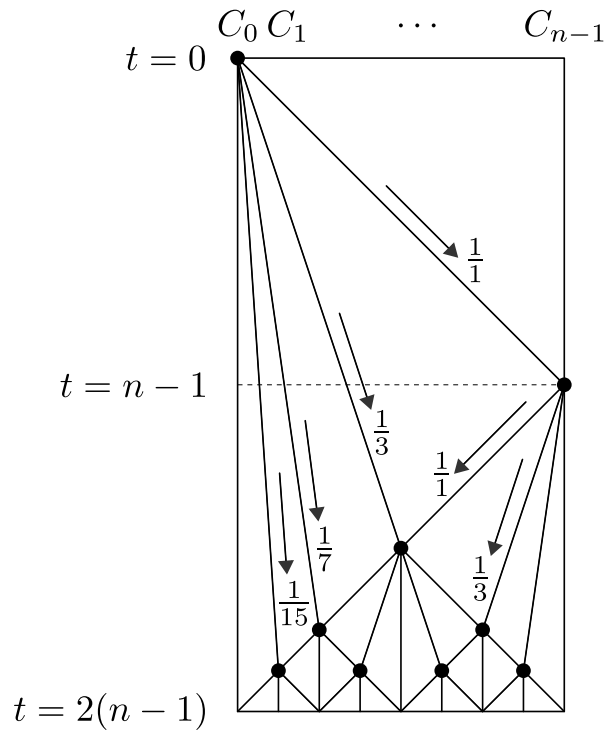


図 5.1: Waksman の最小時間アルゴリズムの時間空間図式

5.3 7セグメントディスプレイ上の数字パターンとセグメント列の一斉射撃問題

本節では、理論的興味から、従来の一斉射撃問題を拡張し7セグメントディスプレイ上の数字パターンの一斉射撃問題を提起する。

5.3.1 7セグメントディスプレイ上の数字パターン

図 5.2 に示すような7セグメントディスプレイは、電子機器でアラビア数字を表示するためによく用いられる表示装置である。7つのセグメントとよばれる素子で構成されており、各セグメントを個別に点灯・消灯することでアラビア数字を表示することができる。

7セグメントディスプレイ上で表される0から9の数字パターンを図 5.3 に示す*2。

*2 数字パターン 6, 7, 9 には別の表示方法もあるが、図 5.3 で示すパターンのみを考える。

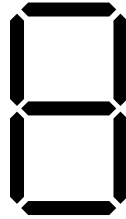


図 5.2: 7 セグメントディスプレイ

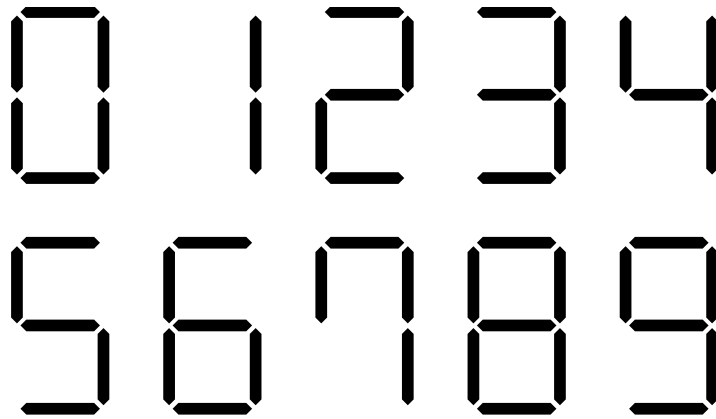


図 5.3: 7 セグメントディスプレイ上の数字パターン

5.3.2 7 セグメントディスプレイのモデル化

図 5.4 に示すように 7 セグメントディスプレイを 2 次元セルオートマトン上でモデル化する。7 つのセグメントはそれぞれ n 個のセルで構成される。これらセグメントを構成するセルをセグメントセルとよぶ。図では網掛けして示している 6 つのセルをセグメントの間に配置する。これら 6 つのセルをジョイントセルとよぶ。6 つのジョイントセルは、上段左から C_A, C_B 、中段左から C_C, C_D 、下段左から C_E, C_F とする。

5.3.3 数字パターンの一斉射撃問題

7 セグメントディスプレイ上の数字パターンの一斉射撃問題を 5 近傍 2 次元セルオートマトン上で定義する。数字パターンの初期様相を図 5.5 に示す。将軍は、各数字パターン

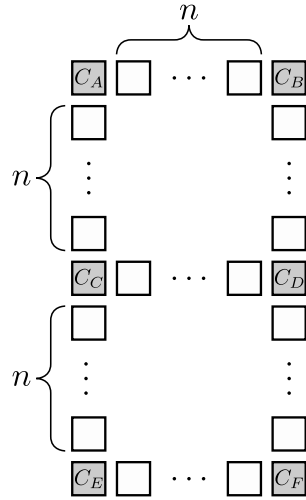


図 5.4: 7セグメントディスプレイのモデル化

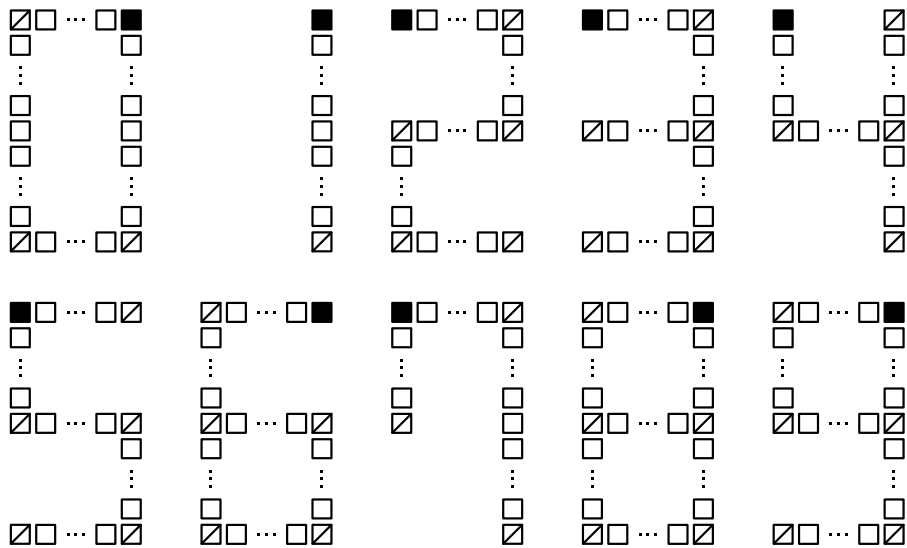


図 5.5: 数字パターンの初期様相

の筆順の最初の位置にあるジョイントセルとする*³。数字パターン 0, 1, 6, 8, 9 の将軍はジョイントセル C_B , 数字パターン 2, 3, 4, 5, 7 の将軍はジョイントセル C_A である。図では黒四角で示している。セグメントを構成するセルの個数 n は任意であるが有限とする。

*³ 日本の小学校で一般的に教えられている筆順を採用した。

5.3.4 数字パターンの一斉射撃アルゴリズム

7セグメントディスプレイ上の数字パターンの一斉射撃問題のいくつかは、セグメント木とよばれる特殊な木の一斉射撃問題に一般化できる。さらに、セグメント木の一斉射撃問題は、ジョイントセルで等分割された1次元セル列の一斉射撃問題に帰着できる。次節以降で、このようなセル列の一斉射撃問題とそのほかの数字パターンの一斉射撃問題を解くアルゴリズムを与える。

まず、図 5.5 に示す数字パターンの初期様相におけるジョイントセルに注目する。図 5.5 で近傍の状態から自身がジョイントセルとわかるものは \square で示している。数字パターン 2, 3, 4, 5, 6, 8, 9 において、ジョイントセル $C_A, C_B, C_C, C_D, C_E, C_F$ はセグメントを構成するセルとは異なる近傍をもつ。このため、これらの数字パターンのジョイントセルは近傍の状態から自身がジョイントセルであると認識することができる。しかし、数字パターン 0 のジョイントセル C_C, C_D と数字パターン 1, 7 のジョイントセル C_D は、セグメントを構成するセルと同じ近傍をもつセルが存在する。このため、これらのジョイントセルは近傍の状態から自身がジョイントセルであると認識することができない。この違いは、アルゴリズムを構成するうえで非常に重要である。

5.3.5 数字パターン 2, 3, 4, 5, 6, 8, 9 の一斉射撃アルゴリズム

数字パターン 2, 3, 4, 5, 6, 8, 9 のうち、数字パターン 2, 3, 4, 5 はいわゆる木である。数字パターン 2, 3, 4, 5 の将軍を手で持ってぶら下げると図 5.6 に示すような木になる。レベル i ($0 \leq i \leq 5$) の節点はジョイントセルであり、レベル 0 の節点が将軍である。節点を結ぶ全ての枝はセグメントで構成される。図では枝を直線で表しているが、実際には n 個のセルで構成される。

残りの数字パターン 6, 8, 9 は、木ではなく図 5.7 に示すようなグラフである。図中で破線によって半分に割ったように表している 4 つの節点に注目する。これら 4 つの節点は、それぞれ根（将軍）からの経路が複数存在する。将軍との信号（情報）のやりとりをする同じ長さの径路が複数存在するので、1 つでも十分であるという意味で分割する。す

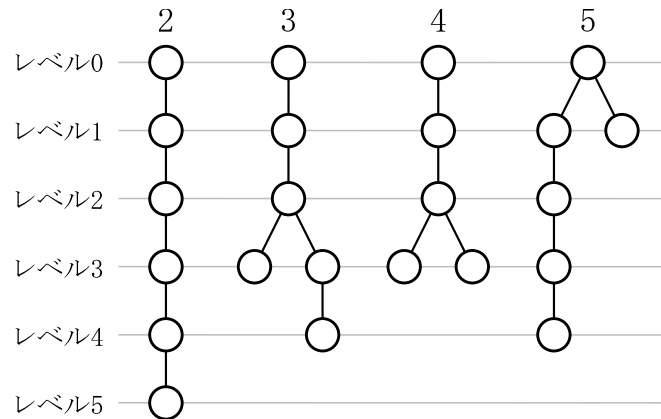


図 5.6: 数字パターン 2, 3, 4, 5 の木

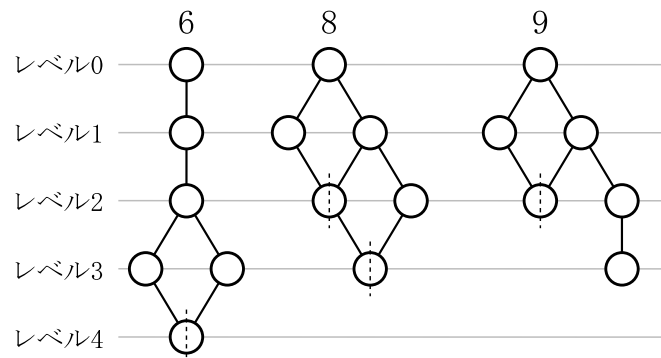


図 5.7: 数字パターン 6, 8, 9 のグラフ

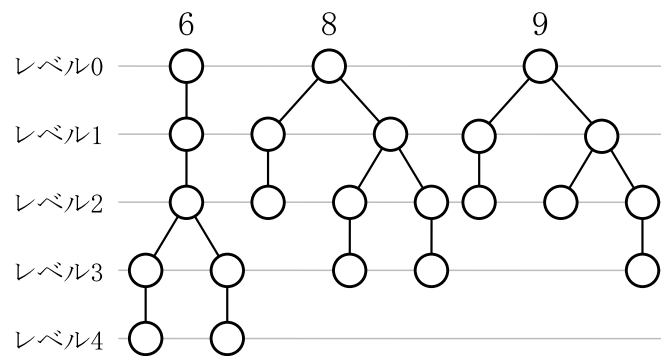


図 5.8: 数字パターン 6, 8, 9 の木

ると, 図 5.8 に示すような木になる.

以上により, 数字パターン 2, 3, 4, 5, 6, 8, 9 の一斉射撃問題は, 以下に示すようなセグメント木の一斉射撃問題に一般化できる.

セグメント木とは、以下の 2 つの条件を満たす木である。

- 枝は 1 つのセグメント (n 個のセル) からなる。
- 節点も 1 つのセルであるが、セグメントセルと自身を区別できる。

根から節点に至る径路にあるセグメント (枝) の数をその節点のレベルという。また、節点のレベルの最大値をセグメント木の高さという。セグメント木の根を将軍とし、全てのセルを一斉に射撃状態に遷移させる問題が、セグメント木の一斉射撃問題である*4。

5.3.6 セグメント列の一斉射撃問題

高さ h のセグメント木の一斉射撃問題は、 $h + 1$ 個のジョイントセルで等間隔に区切られた 1 次元セル列の一斉射撃問題に帰着できる。

このようなジョイントセルで等間隔に区切られた 1 次元セル列を定義するため、従来の一斉射撃問題に状態 Q_J を導入する。 Q_J をジョイント状態とよぶ。ジョイント状態 Q_J は、式 5.5–5.7 を満たす。ここで、 $s_Q \in \{Q, B\}$, $s \in \mathcal{S}$, $* \in \mathcal{S} \cup \{B\}$ である。

$$\delta(s_Q, Q_J, s_Q) = Q_J \quad (5.5)$$

$$\delta(*, s, Q_J) = \delta(*, s, Q) \quad (5.6)$$

$$\delta(Q_J, s, *) = \delta(Q, s, *) \quad (5.7)$$

ジョイント状態 Q_J は静止状態 Q に準ずる状態である。左右のセルが静止状態もしくは境界信号であるとき状態遷移しない (式 5.5)。また、ジョイント状態を近傍にもつセル (ジョイントセルの左右にあるセグメントセル) は、ジョイント状態と静止状態を区別できない (式 5.5, 5.6)。しかし、ジョイント状態のセルが一旦将軍からの信号を受け取ると、静止状態が遷移する状態とは別の状態に遷移することができる。

長さ $h(n + 1) + 1$ のセル列 $C_0, C_1, \dots, C_{h(n+1)}$ に対して、セル C_0 が将軍状態 G 、全てのセル $C_{m(n+1)}$ ($m \in \{x \in \mathbb{N} \mid 1 \leq x \leq h\}$) *5がジョイント状態 Q_J であり、残りの

*4 もはや 2 次元セルオートマトン上で定義される一斉射撃問題ではないことに注意されたい。

*5 \mathbb{N} は 0 を除く正の整数の集合である。

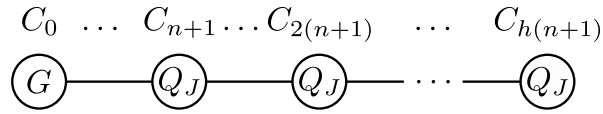


図 5.9: セグメント列

セルが静止状態 Q である。このようなセル列をセグメント列とよぶ。図 5.9 に示すように、セグメント列は将軍がジョイントセルによって h 個のパーツに分割されている。そして、セグメント列の全てのセルを一斉に射撃状態に遷移させる問題が、セグメント列の一斉射撃問題である。

5.3.7 セグメント列の一斉射撃アルゴリズム

セグメント列の一斉射撃問題を解くアルゴリズムを示す。ここで、ある $\alpha \in \mathbb{N}$ に対して、セグメントの長さ n が $\alpha l - 1$ であると仮定して一般性を失わない。

アルゴリズム 5.1

1. 時刻 $t = 0$ に、将軍 C_0 が速度 $\frac{i}{h\alpha}$ の信号 S_i ($0 \leq i \leq h\alpha$) を送出する。
2. 時刻 $t = n+1$ に、セル C_0 からの信号 $S_{h\alpha}$ をセル C_{n+1} が受け取り、速度 $\frac{i-\alpha}{h\alpha}$ *6 の信号 T_i ($0 \leq i \leq h\alpha$) を送出する。
3. 時刻 $t = h(n+1)$ に、セル $C_{\frac{i}{\alpha}(n+1)}$ ($0 \leq i < h\alpha$) が信号 S_i と T_i を同時に受け取って、将軍状態に遷移する。同時刻に、セル $C_{h(n+1)}$ が信号 $T_{h\alpha}$ を受け取って、将軍状態に遷移する。
4. セル $C_{\frac{i}{\alpha}(n+1)}$ ($0 \leq i \leq h\alpha$) は、つながっている全てのセル列に対して、最小時間アルゴリズムを適用する。

図 5.10 に、 $h = 3$, $\alpha = 2$ におけるアルゴリズム 5.1 の時間空間図式を示す。図で円と黒い点は異なる状態であることを表している。円はセル C_0 がセグメント列の将軍状態であることを表し、黒い点はそのセルが 5.2 節の最小時間アルゴリズムにおける将軍状態

*6 正の速度の信号は右方向に、負の速度の信号は左方向に進むことを表す。

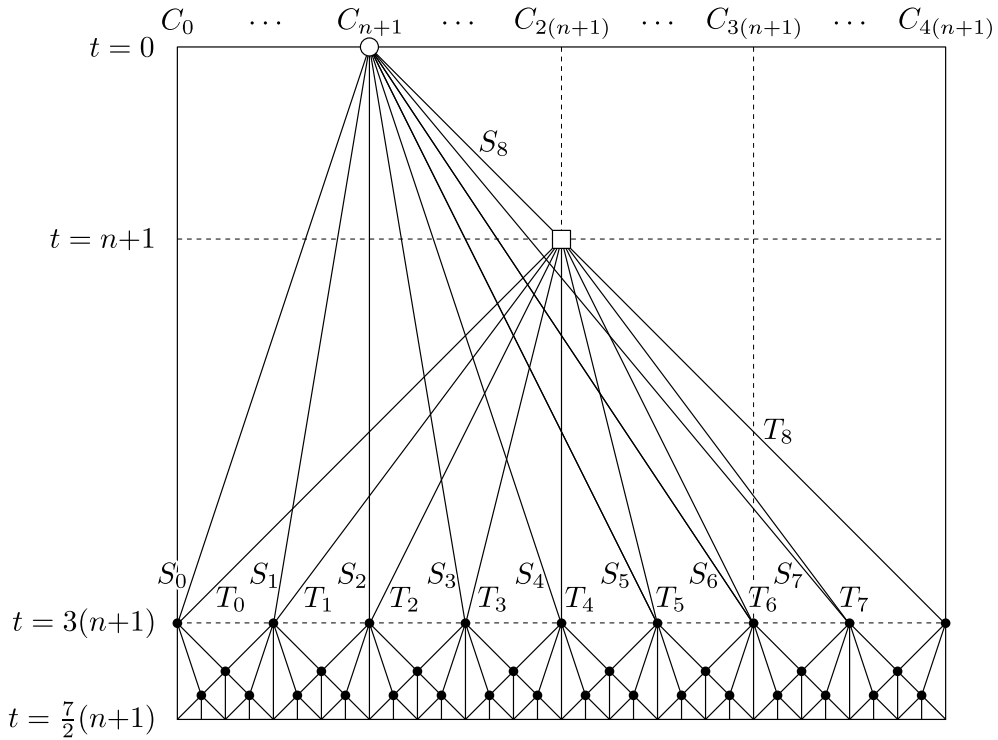


図 5.10: $h = 3, \alpha = 2$ におけるセグメント列の一斉射撃アルゴリズムの時間空間図式

であることを表している。四角はジョイントセル C_{n-1} が将軍 C_0 からの信号 S_6 を受け取って、信号 T_i ($0 \leq i \leq 6$) を送出することを表している。

セル $C_{\frac{i}{\alpha}(n+1)}$ ($0 \leq i \leq h\alpha$) が将軍状態に遷移するためには $h(n+1)$ ステップ必要である。さらに、将軍状態のセルを両端にもつ長さ $\frac{1}{\alpha}(n+1)+1$ のセル列を最小時間アルゴリズムで一斉射撃させるために $\frac{1}{\alpha}(n+1)$ ステップ必要である。よって、アルゴリズム 5.1 の射撃時間は $(h + \frac{1}{\alpha})(n+1)$ である。 α の値を大きくすることで、 $\frac{1}{\alpha}(n+1)$ を小さくすることができ、射撃時間を限りなく $h(n+1)$ に近づけることができる。

数字パターン 4, 8, 9 の一斉射撃問題は、長さ $3(n+1)$ のセグメント列の一斉射撃問題に帰着することができ、 $(3 + \frac{1}{\alpha})(n+1)$ ステップで解くことができる。数字パターン 3, 5, 6 の一斉射撃問題は、長さ $4(n+1)$ のセグメント列の一斉射撃問題に帰着することができ、 $(4 + \frac{1}{\alpha})(n+1)$ ステップで解くことができる。数字パターン 2 の一斉射撃問題は、長さ $5(n+1)$ のセグメント列の一斉射撃問題に帰着することができ、 $(5 + \frac{1}{\alpha})(n+1)$ ステップで解くことができる。

5.3.8 セグメント列の一斉射撃問題の最小時間

$n = \alpha l - 1$, $\alpha \in \mathbb{N}$ のとき, 長さ $h(n+1) + 1$ のセグメント列の一斉射撃問題を $(h + \frac{1}{\alpha})(n+1)$ ステップで解くアルゴリズムを示した.

本節では, 長さ $h(n+1) + 1$ のセグメント列の一斉射撃問題の最小時間を示す. セル $C_{h(n+1)}$ は時刻 $t = h(n+1) - 1$ までジョイント状態 Q_J であるため, 射撃時間は $h(n+1)$ ステップ未満になることはない. よって, 以下の補題が成り立つ.

補題 5.1 長さ $h(n+1) + 1$ のセグメント列の一斉射撃問題の任意の解 \mathcal{A} に対して, $t_F(n, \mathcal{A}) \geq h(n+1)$ が成り立つ.

ある長さ $h(n_0+1) + 1$ のセグメント列を $h(n_0+1)$ ステップで射撃させ, $n \neq n_0$ である長さ $h(n+1) + 1$ のセグメント列は射撃させない有限オートマトン \mathcal{A}_{n_0} を容易に構成できる. $\mathcal{A}_{n_0} = (S, \delta)$ は以下のように構成される. ただし, 記号 $*$ は $S \cup \{B\}$ である.

$$S = \{X, Q, Q_J, G = S_0, S_1, S_2, \dots, S_{h(n_0+1)} = F\} \quad (5.8)$$

$$\delta(S_{i-1}, Q, *) = S_i \quad \text{if } i \neq l(n_0+1), 1 \leq l \leq h \quad (5.9)$$

$$\delta(S_{i-1}, Q_J, *) = S_i \quad \text{if } i = l(n_0+1), 1 \leq l \leq h \quad (5.10)$$

$$\delta(*, S_{i-1}, s) = S_i \quad \text{if } s \neq X \quad (5.11)$$

$$\delta(S_{i-1}, Q_J, *) = X \quad \text{if } i \neq n_0 + 1 \quad (5.12)$$

$$\delta(*, S_i, X) = X \quad (5.13)$$

$$\delta(S_{n_0}, Q, *) = X \quad (5.14)$$

\mathcal{A}_{n_0} の動作を簡単に説明する. まず, 長さ $h(n_0+1) + 1$ のセグメント列を考える. 将軍 $G = S_0$ は, セルの数を数えながら $S_1, S_2, \dots, S_{h(n_0+1)}$ と順に遷移する. その状態は 1 つ 1 つ右のセルに伝播する. その結果, 時刻 $t = i$ に C_0 から C_i の全てのセルが状態 S_i に遷移する. そして, 時刻 $t = h(n_0+1)$ にセグメント列を構成する全てのセルが, 状態 $S_{h(n_0+1)} = F$ つまり射撃状態に遷移する.

$n < n_0$ のとき, 時刻 $t = n+1$ にセル C_{n+1} が状態 Q_J から状態 X に遷移する. 状態

X は左方向に伝播し、時刻 $t = 2(n+1) < h(n_0+1)$ に全てのセルが状態 X, Q, Q_J のいずれかになっている。

同様に $n > n_0$ のとき、時刻 $t = n_0 + 1$ にセル C_{n_0+1} が状態 Q から状態 X に遷移する。そして、時刻 $t = 2(n_0+1) \leq h(n_0+1)$ に全てのセルが状態 X, Q, Q_J のいずれかになっている。

\mathcal{A}_{n_0} と適当な解 \mathcal{A} を同時に模倣する有限オートマトン \mathcal{A}' はセグメント列の一斉射撃問題の解であり、 $t_F(n_0, \mathcal{A}') = h(n_0+1)$ である。合成した有限オートマトン \mathcal{A}' の状態の集合は、射撃状態 F と 2 つの有限オートマトン \mathcal{A}_{n_0} と \mathcal{A} の状態のペアで表される状態を含む。この有限オートマトン \mathcal{A}' は、2 つの有限オートマトン \mathcal{A}_{n_0} と \mathcal{A} を同時にシミュレートし、どちらかの有限オートマトンが射撃したとき射撃状態 F に遷移する。以上により、以下の定理が成り立つ。

定理 5.1 長さ $h(n+1)+1$ のセグメント列の一斉射撃問題に対して $t_{F \min}(n) = h(n+1)$ が成り立つ。

5.3.9 セグメント列の一斉射撃問題の最小時間解

長さ $h(n+1)+1$ のセグメント列の一斉射撃問題の最小時間が $h(n+1)$ であることを示したが、全ての長さ n に対して最小時間で射撃させるような最小時間解が存在するかを考える。以下の補題は最小時間解が存在しないことを示している。

補題 5.2 $\mathcal{A} = (\mathcal{S}, \delta)$ を長さ $h(n+1)+1$ のセグメント列の一斉射撃問題の解としたとき、もし $n+2 > |\mathcal{S}|^2$ であれば $t_F(n, \mathcal{A}) > h(n+1)$ が成り立つ。

証明 もし $n+2 > |\mathcal{S}|^2$ であれば $t_F(n, \mathcal{A}) = h(n+1)$ が成り立つと仮定し、背理法により補題 5.2 が成り立つことを証明する。

時刻 t のセル C_i の状態を s_i^t とする。

$n+2 > |\mathcal{S}|^2$ なので、 $s_{i-1}^i = s_{j-1}^j$, $s_i^i = s_j^j$ を満たすような 2 つの異なる整数 i, j ($(h-1)(n+1) \leq i < j \leq h(n+1)$) が存在する。つまり、セル $C_{(h-1)(n+1)}$ の右側

のセル列で、セルのペア $C_{i-1}C_i$ と $C_{j-1}C_j$ が存在し、時刻 i のときのセル $C_{i-1}C_i$ の状態 $s_{i-1}^i s_i^i$ と時刻 j のときのセル $C_{j-1}C_j$ の状態 $s_{j-1}^j s_j^j$ が等しいことを表している。

$(h-1)(n+1) \leq i < h(n+1) - 1$ のとき $s_{i+1}^{i+1} = \delta(s_i^i, Q, Q)$ であるので、 $0 \leq k < h(n+1) - 1 - j$ のとき $s_{i+k}^{i+k} = s_{j+k}^{j+k}$ である。さらに、 $(h-1)(n+1) \leq i \leq h(n+1) - 1$ のとき $s_i^{i+1} = \delta(s_{i-1}^i, s_i^i, Q)$ であるので、 $0 \leq k \leq h(n+1) - 1 - j$ のとき $s_{i+k}^{i+k+1} = s_{j+k}^{j+k+1}$ である。従って、 $s_{h(n+1)-1+i-j}^{h(n+1)+i-j} = s_{h(n+1)-1}^{h(n+1)}$ が成り立つ。

仮定から、一斉射撃時間は $h(n+1)$ であり、 $s_{h(n+1)-1}^{h(n+1)} = F$ が成り立つ。従って、 $s_{h(n+1)-1+i-j}^{h(n+1)+i-j} = s_{h(n+1)-1}^{h(n+1)} = F$ も成り立つ。

時刻 $t = h(n+1) + i - j$ にセル $C_{h(n+1)-1+i-j}$ が射撃状態に遷移することから、 \mathcal{A} は解ではない。これは仮定に矛盾するため、 $t_F(n, \mathcal{A}) = h(n+1)$ は成り立たない。□

定理 5.2 セグメント列の一斉射撃問題の最小時間解は存在しない。

5.3.10 数字パターン 7 の一斉射撃アルゴリズム

数字パターン 7 の一斉射撃問題はセグメント列の一斉射撃問題に帰着できない。しかし、同様に解くことができる。数字パターン 7 は長さ $4(n+1) + 1$ のセル列 $C_0, C_1, \dots, C_{4(n+1)}$ である。最初に、セル C_{n+1} が将軍状態、セル $C_{i(n+1)}$ ($i = 0, 2, 4$) がジョイント状態、残りのセルが静止状態である。

ある自然数 α に対して $n = \alpha l - 1$ だと仮定して一般性を失わない。数字パターン 7 の一斉射撃問題を解くアルゴリズムを以下に示す。

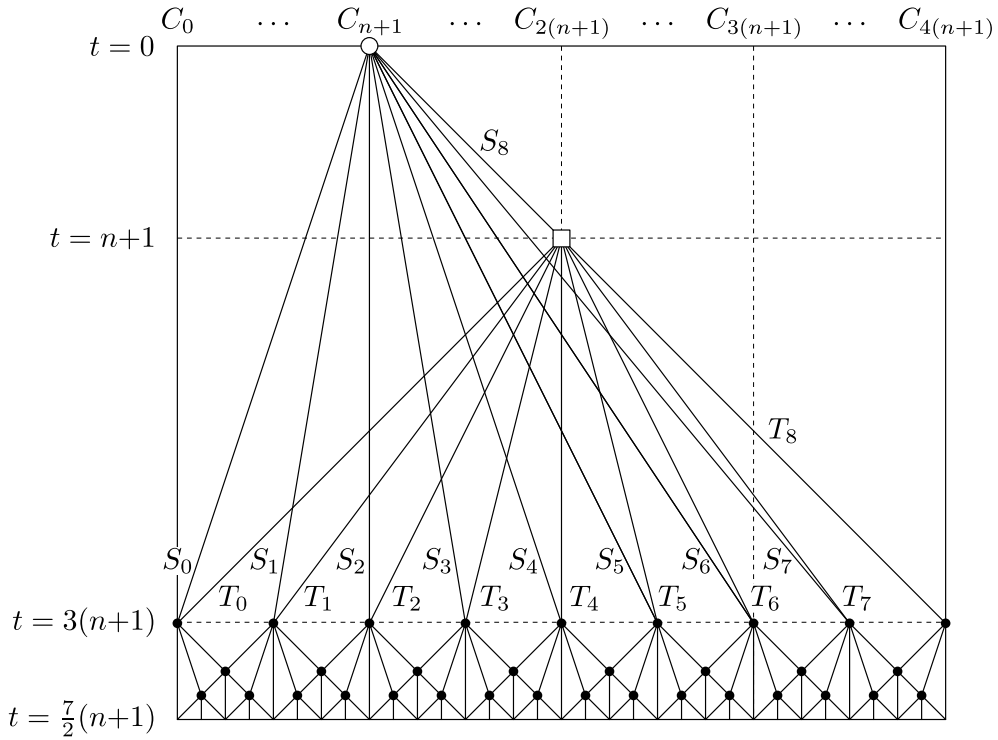


図 5.11: $\alpha = 2$ における数字パターン 7 の一斉射撃アルゴリズムの時間空間図式

アルゴリズム 5.2

1. 時刻 $t = 0$ に, 将軍 C_{n+1} は速度 $\frac{i-\alpha}{3\alpha}$ の信号 S_i ($0 \leq i \leq 4\alpha$) を送出する.
2. 時刻 $t = n + 1$ に, セル $C_{2(n+1)}$ は将軍 C_{n+1} からの信号 $S_{4\alpha}$ を受け取って, 速度 $\frac{i-2\alpha}{2\alpha}$ の信号 T_i ($0 \leq i \leq 4\alpha$) を送出する.
3. 時刻 $t = 3(n + 1)$ に, セル $C_{\frac{i}{\alpha}(n+1)}$ ($0 \leq i < 4\alpha$) は信号 S_i と T_i を同時に受け取って, 最小時間アルゴリズムの将軍状態に遷移する. 同時刻に, セル $C_{4(n+1)}$ は信号 $T_{4\alpha}$ を受け取って, 最小時間アルゴリズムの将軍状態に遷移する.
4. セル $C_{\frac{i}{\alpha}(n+1)}$ ($0 \leq i \leq 4\alpha$) はつながっている全てのセル列に最小時間アルゴリズムを適用する.

図 5.11 に, $\alpha = 2$ における数字パターン 7 の一斉射撃アルゴリズムの時間空間図式を示す.

セル $C_{\frac{i}{\alpha}(n+1)}$ ($0 \leq i \leq 4\alpha$) が最小時間アルゴリズムの将軍状態に遷移するには $3(n+1)$

ステップ必要である．このため，数字パターン 7 の一斉射撃問題を $(3 + \frac{1}{\alpha})(n + 1)$ ステップで解くことができる． α が大きくなれば，射撃時間は $3(n + 1)$ に限りなく近づく．

5.3.11 数字パターン 7 の一斉射撃問題の最小時間

セル $C_{4(n+1)}$ は時刻 $t = 3(n + 1) - 1$ までジョイント状態 Q_J なので，射撃時間は $3(n + 1)$ 以上であるとわかる．よって以下の補題が成り立つ．

補題 5.3 数字パターン 7 の一斉射撃問題のある解 \mathcal{A} に対して， $t_F(n, \mathcal{A}) \geq 3(n + 1)$ が成り立つ．

5.3.8 節と同様に以下の定理が成り立つ．

定理 5.3 数字パターン 7 の一斉射撃問題に対して $t_{F \min}(n) = 3(n + 1)$ が成り立つ．

5.3.12 数字パターン 7 の一斉射撃問題の最小時間解

数字パターン 7 の一斉射撃問題の最小時間は $3(n + 1)$ であるが，全ての長さ n を最小時間で解くような最小時間解は存在しないことを示す．

補題 5.4 有限オートマトン $\mathcal{A} = (S, \delta)$ が数字パターン 7 の一斉射撃問題の解であるとする．もし $2(n + 1) > |S|^2$ であれば， $t_F(n, \mathcal{A}) > 3(n + 1)$ が成り立つ．

補題 5.4 の証明は補題 5.2 の証明と同様である．これより，以下の定理が成り立つ．

定理 5.4 数字パターン 7 の一斉射撃問題の最小時間解は存在しない．

5.3.13 数字パターン 0 の一斉射撃アルゴリズム

数字パターン 0 の一斉射撃問題はセグメント列の一斉射撃問題に帰着できない．しかし，同様に解くことができる．数字パターン 0 のジョイントセル C_F を半分に割ると，長さ $6(n + 1) + 1$ のセル列 $C_0, C_1, \dots, C_{6(n+1)}$ である．セル C_0 と $C_{6(n+1)}$ が半分に割った

ジョイントセル C_F である．最初に，セル $C_{3(n+1)}$ が将軍状態，セル $C_{i(n+1)}$ ($i = 0, 2, 5, 6$) がジョイント状態，残りのセルが静止状態である．

ある自然数 α に対して $n = \alpha l - 1$ だと仮定して一般性を失わない．数字パターン 0 の一斉射撃問題を解くアルゴリズムを以下に示す．

アルゴリズム 5.3

1. 時刻 $t = 0$ に，将軍 $C_{3(n+1)}$ が速度 $\frac{i-3\alpha}{3\alpha}$ の信号 S_i ($0 \leq i \leq 6\alpha$) を送出する．
2. 時刻 $t = n + 1$ に，セル $C_{2(n+1)}$ が将軍 $C_{3(n+1)}$ からの信号 S_0 を受け取って，速度 $\frac{i-2\alpha}{2\alpha}$ の信号 T_i ($0 \leq i \leq 4\alpha$) を送出する．
3. 時刻 $t = 2(n + 1)$ に，セル $C_{5(n+1)}$ が将軍 $C_{3(n+1)}$ からの信号 $S_{6\alpha}$ を受け取って，速度 $\frac{i-2\alpha}{2\alpha}$ の信号 T_i ($4\alpha < i \leq 6\alpha$) を送出する．
4. 時刻 $t = 3(n + 1)$ に，セル $C_{\frac{i}{\alpha}(n+1)}$ ($0 < i < 6\alpha$) は信号 S_i と T_i を同時に受け取って，最小時間アルゴリズムの将軍状態に遷移する．同時刻に，セル C_0 が信号 T_0 を，セル $C_{6(n+1)}$ が信号 $T_{6\alpha}$ を受け取って，セル C_0 , $C_{6(n+1)}$ も最小時間アルゴリズムの将軍状態に遷移する．
5. セル $C_{\frac{i}{\alpha}(n+1)}$ ($0 \leq i \leq 6\alpha$) はつながっている全てのセル列に対して最小時間アルゴリズムを適用する．

図 5.12 に， $\alpha = 2$ における数字パターン 0 の一斉射撃アルゴリズムの時間空間図式を示す．

セル $C_{\frac{i}{\alpha}(n+1)}$ ($0 \leq i \leq 6\alpha$) が最小時間アルゴリズムの将軍状態に遷移するには $3(n+1)$ ステップ必要である．このため，数字パターン 0 の一斉射撃問題を $(3 + \frac{1}{\alpha})(n + 1)$ ステップで解くことができる． α が大きくなれば，射撃時間は $3(n + 1)$ に限りなく近づく．

n 個のセルで構成される環状のセル列の一斉射撃問題に対して， n ステップで一斉射撃させるアルゴリズムが示されている [8]．数字パターン 0 の一斉射撃問題を， $6(n + 1)$ 個のセルで構成される環状のセル列の一斉射撃問題として解くと， $6(n + 1)$ ステップ必要である．従って，環状のセル列の一斉射撃アルゴリズムよりも，アルゴリズム 5.3 の方が，少なくとも $2(n + 1)$ ステップ早いことがわかる．これは，アルゴリズム 5.3 がジョイン

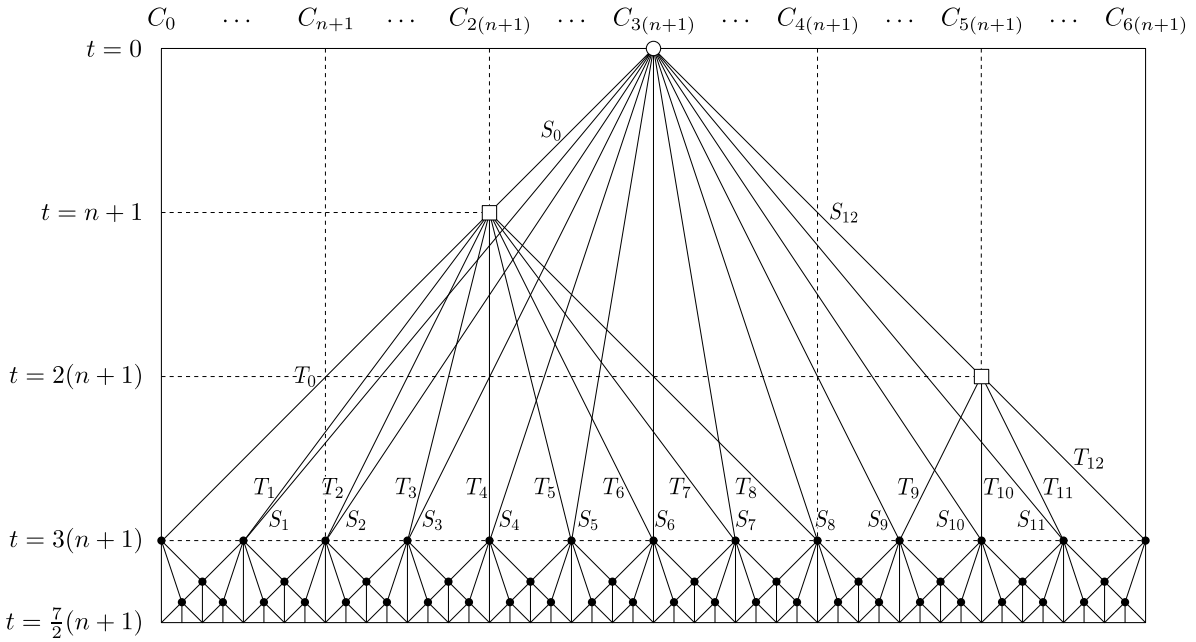


図 5.12: $\alpha = 2$ における数字パターン 0 の一斉射撃アルゴリズムの時間空間図式

トセル C_A, C_B, C_E, C_F の位置情報を利用しているためである。

5.3.14 数字パターン 0 の一斉射撃問題の最小時間

セル $C_{6(n+1)}$ は時刻 $t = 3(n + 1) - 1$ までジョイント状態 Q_J なので、射撃時間は $3(n + 1)$ 以上であるとわかる。よって以下の補題が成り立つ。

補題 5.5 数字パターン 0 の一斉射撃問題のある解 \mathcal{A} に対して、 $t_F(n, \mathcal{A}) \geq 3(n + 1)$ が成り立つ。

5.3.8 節と同様に以下の定理が成り立つ。

定理 5.5 数字パターン 0 の一斉射撃問題に対して $t_{F \min}(n) = 3(n + 1)$ が成り立つ。

5.3.15 数字パターン 0 の一斉射撃問題の最小時間解

数字パターン 0 の一斉射撃問題の最小時間は $3(n+1)$ であるが、全ての長さ n を最小時間で解くような最小時間解は存在しないことを示す。

補題 5.6 有限オートマトン $\mathcal{A} = (\mathcal{S}, \delta)$ が数字パターン 0 の一斉射撃問題の解であるとする。もし $2(n+1) > |\mathcal{S}|^2$ であれば、 $t_F(n, \mathcal{A}) > 3(n+1)$ が成り立つ。

補題 5.6 の証明は補題 5.2 の証明と同様である。これより、以下の定理が成り立つ。

定理 5.6 数字パターン 0 の一斉射撃問題の最小時間解は存在しない。

5.3.16 数字パターン 1 の最小時間アルゴリズム

数字パターン 1 のジョイントセル C_D はセグメントセルと同じ近傍の状態をもつため、近傍からセグメントセルと区別することができない。つまり、長さ $2n+3$ の 1 次元セル列の一斉射撃問題である。従って、最小時間アルゴリズムを適用すると、 $4(n+1)$ ステップで解くことができ、当然ながら最小時間は $4(n+1)$ であるとわかる。

5.4 16 セグメントディスプレイ上の英字パターンと 1-2 セグメント木の一斉射撃問題

本節では、理論的興味から、7 セグメントディスプレイ上の数字パターンの一斉射撃問題を拡張し、16 セグメントディスプレイ上の英字パターンの一斉射撃問題を提起する。

5.4.1 16 セグメントディスプレイ上の英字パターン

16 セグメントディスプレイとは、図 5.13 に示すようなディスプレイである。名前が示すとおり 16 個のセグメントとよばれる素子で構成されていて、それぞれをオン、オフすることによって英字や数字などを表示することができる。

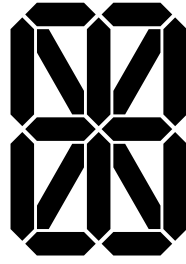


図 5.13: 16 セグメントディスプレイ

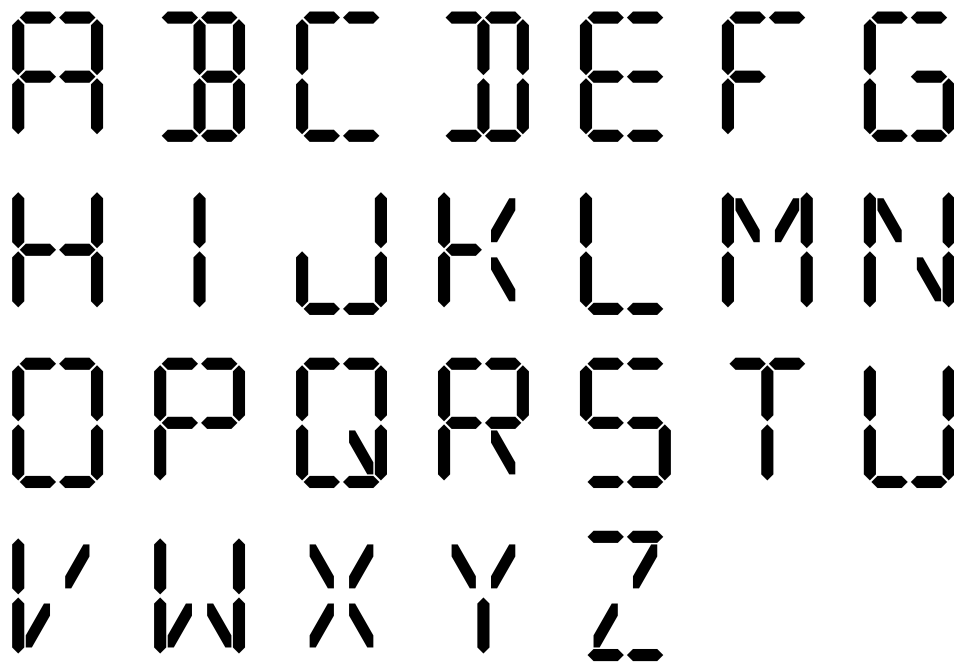


図 5.14: 16 セグメントディスプレイ上の英字パターン

16 セグメントディスプレイ上で，全ての英字はそれぞれ図 5.14 に示すように表示される．

5.4.2 16 セグメントディスプレイのモデル化

図 5.13 に示した 16 セグメントディスプレイを，図 5.15 に示すように 2 次元セル空間上でモデル化する．

16 個のセグメントはそれぞれ n 個のセルからなるものとする．ただし， n は任意の自然数であるが有限とする．これらセグメントを構成するセルをセグメントセルとよぶ．ま

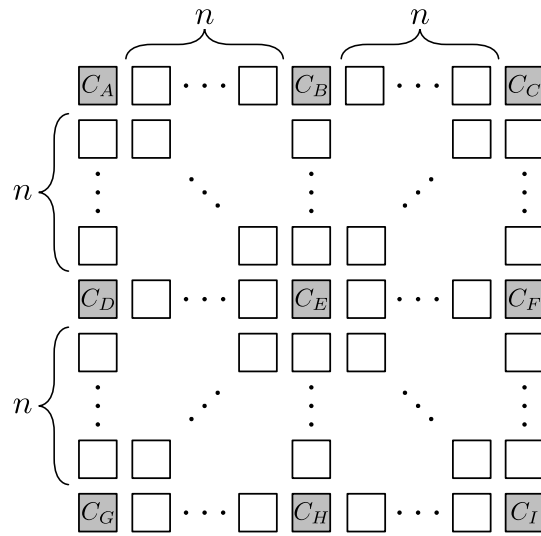


図 5.15: 16 セグメントディスプレイのモデル化

た、各セグメントをつなぐセルとして 9 個のセルを置く．図 5.15 では網かけして示してある．これら 9 個のセルをジョイントセルとよび、それぞれ上段左から C_A, C_B, C_C 、中段左から C_D, C_E, C_F 、下段左から C_G, C_H, C_I とする．

5.4.3 英字パターンの一斉射撃問題

図 5.14 に示した 16 セグメントディスプレイ上の英字パターンを、それぞれ図 5.15 のようにモデル化されたディスプレイ上に配置し、筆順^{*7}の 1 番初めの位置にあるジョイントセルを将軍として、16 セグメントディスプレイ上の英字パターンの一斉射撃問題を 9 近傍 2 次元セルオートマトン上の問題として定義する．

図 5.16 に各英字パターンの初期様相を示した．将軍セルが将軍状態で、残りのセグメントセルとジョイントセルは全て静止状態である．将軍は、英字パターン A, B, D, I, O, Q, T についてはジョイントセル C_B 、英字パターン C, G, J, S についてはジョイントセル C_C 、残りの英字パターンについてはジョイントセル C_A であり、図 5.16 では黒塗りで示してある．図には示していないが、2 次元セル空間上のセグメントセルとジョイントセル以外のセルからの入力境界信号である．

^{*7} 中学校で一般的に教えられている筆順を採用した．

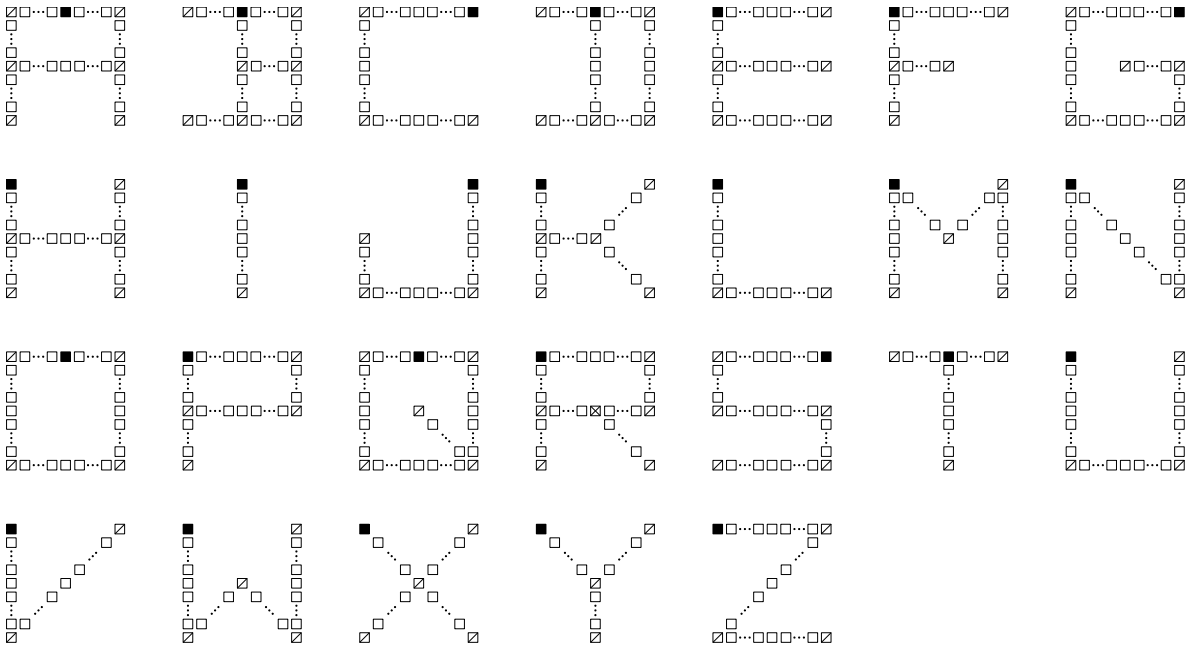


図 5.16: 英字パターンの初期様相

5.4.4 英字パターンの一斉射撃アルゴリズム

各英字パターンにおいて、近傍の状態から自身がジョイントセルであるとわかるものが多い。しかし、例えば英字パターン A のジョイントセル C_E は、自身とその左右のセルのみが静止状態であり、セグメントセルと全く同じ近傍の状態をもつため、近傍の状態のみでは自身がジョイントセルであることがわからない*8。図 5.16 で近傍の状態から自身がジョイントセルとわかるものは \boxtimes で示している。

ただし、英字パターン R のジョイントセル C_E (図中では \boxtimes で示している) はジョイントセル C_C の左隣のセグメントセルと同じ近傍の状態をもつため、自身がジョイントセルとわからない。

*8 ジョイントセル C_B も近傍の状態のみではわからないが、将軍であるため自身がジョイントセルであることがわかる。

5.4.5 セグメント木に帰着できる英字パターンの一斉射撃アルゴリズム

11 種の英字パターン A, B, C, K, L, N, U, V, X, Y, Z は 5.3.5 節で示したセグメント木の
一斉射撃問題に帰着することができる。

まず、3 種の英字パターン K, X, Y の将軍をそれぞれ手で持って、ぶらぶらと振ったと
きのイメージで表すと、図 5.17 に示すようなセグメント木になる。節点（図では円で描
いている）はジョイントセルで、特にレベル 0 の節点は将軍である。枝を直線で描いてい
るが、それぞれがセグメントに対応し、 n 個のセルからなる。

同様に、6 種の英字パターン C, L, N, U, V, Z は図 5.18 に示すようなセグメント木で
ある。近傍の状態から自身がジョイントセルであることがわからないものは枝に含まれる
ものとする。よって、全ての枝は $2n + 1$ 個のセルからなる。

一方、2 種の英字パターン A, B は木ではなく、図 5.19 に示すようなグラフである。

英字パターン A のジョイントセル C_E は、近傍の状態から自身がジョイントセルであ
ることがわからない。しかしジョイントセル C_E は、ジョイントセル C_A と C_D を通る径
路とジョイントセル C_C と C_F を通る径路の 2 つの径路から、将軍 C_B から送出された速
度 $\frac{1}{1}$ の信号を時刻 $t = 3(n + 1)$ に同時に受け取る。これによって、 C_E は自身がジョイ
ントセルであることがわかる。この時刻は、将軍から送出された速度 $\frac{1}{1}$ の信号をレベル 3
の節点が受け取る時刻と同じであるので、 C_E をレベル 3 の節点とする。

図 5.19 のジョイントセルのうち 3 つのセル（破線で半分にしたように描いてある節点）
を半分に割ったと考えて、それぞれ 2 つにすると、図 5.20 のようなセグメント木になる。

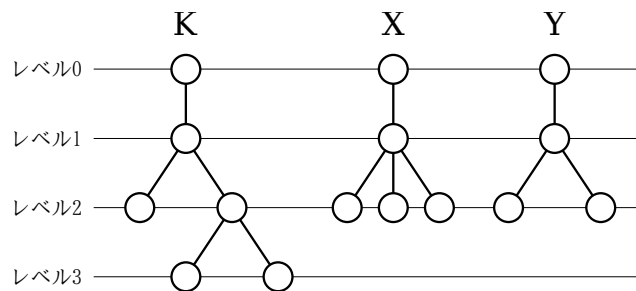


図 5.17: 英字パターン K, X, Y に対するセグメント木

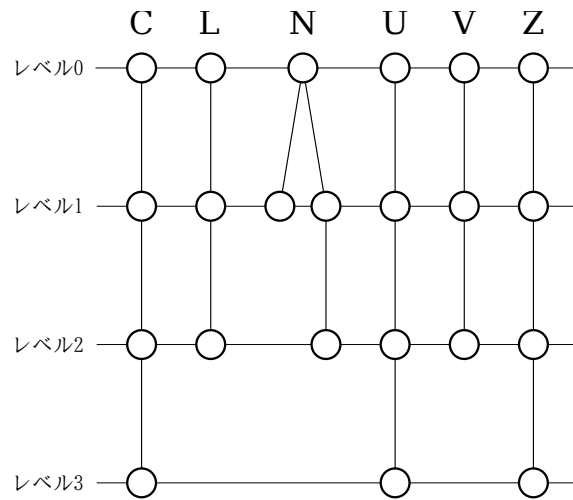


図 5.18: 英字パターン C, L, N, U, V, Z に対するセグメント木

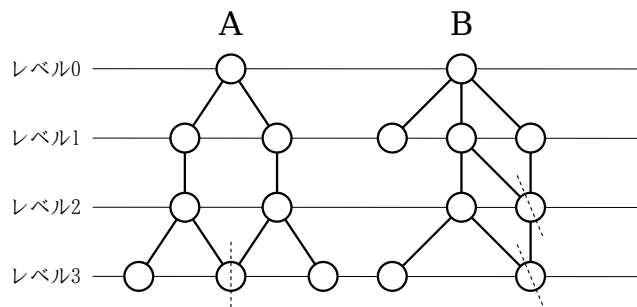


図 5.19: 英字パターン A, B に対するグラフ

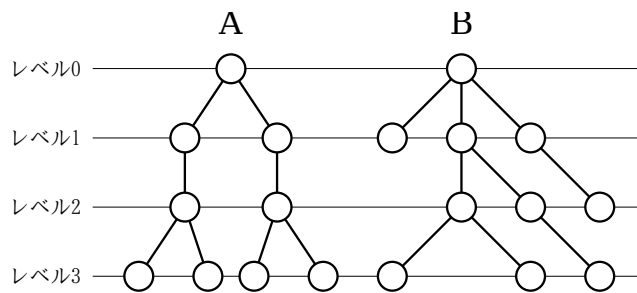


図 5.20: 英字パターン A, B に対するセグメント木

全ての枝は n 個のセルからなっている．半分に割ったジョイントセルには，将軍との信号（情報）のやりとりをする同じ長さの径路が複数存在するので，1つでも十分であるという意味で分割する．

表 5.1: セグメント木の一斉射撃アルゴリズムを用いて解ける英字パターンとその射撃時間

パターン	枝の長さ	h	射撃時間
A	n	3	$(3 + \frac{1}{\alpha})(n + 1)$
B	n	3	$(3 + \frac{1}{\alpha})(n + 1)$
C	$2n + 1$	3	$(6 + \frac{1}{\alpha})(n + 1)$
K	n	3	$(3 + \frac{1}{\alpha})(n + 1)$
L	$2n + 1$	2	$(4 + \frac{1}{\alpha})(n + 1)$
N	$2n + 1$	2	$(4 + \frac{1}{\alpha})(n + 1)$
U	$2n + 1$	3	$(6 + \frac{1}{\alpha})(n + 1)$
V	$2n + 1$	2	$(4 + \frac{1}{\alpha})(n + 1)$
X	n	2	$(2 + \frac{1}{\alpha})(n + 1)$
Y	n	2	$(2 + \frac{1}{\alpha})(n + 1)$
Z	$2n + 1$	3	$(6 + \frac{1}{\alpha})(n + 1)$

以上により, 11 種の英字パターンをセグメント木として表すことができることがわかった. また, セグメント木の一斉射撃問題はセグメント列の一斉射撃問題に帰着することができる. セグメント列の一斉射撃アルゴリズム (アルゴリズム 5.1) を用いると, $n = \alpha m - 1$ を満たす自然数 α, m に対して, 高さ h のセグメント木を $(h + \frac{1}{\alpha})(n + 1)$ ステップで一斉射撃させることができる. また, 最小時間は $h(n + 1)$ であるが, 全ての n に対して最小時間で射撃させるような最小時間アルゴリズムは存在しないことを示した. α は分割数で, 節点の位置情報を利用することによって, 全ての枝を α 個の枝と $\alpha - 1$ 個の節点に分割して一斉射撃させる. α を大きくすることで, 射撃時間を限りなく最小時間 $h(n + 1)$ に近づけることができる. つまり, 最小時間アルゴリズムに限りなく近いアルゴリズムであるといえる.

セグメント木の一斉射撃アルゴリズムを用いて解くことができる 11 種の英字パターンとその射撃時間を表 5.1 に示す.

5.4.6 1-2 セグメント木の一致射撃問題

13種の英字パターン D, E, F, G, H, J, M, O, P, Q, S, T, W は全ての枝が n 個もしくは $2n + 1$ 個のセルからなる木で表すことができる。まず, 9種の英字パターン E, F, G, H, J, M, S, T, W は図 5.21 に示すような木で表される。一方, 4種の英字パターン D, O, P, Q は木ではなく, 図 5.22 に示すようなグラフである。英字パターン A のジョイントセル C_E と同様に, 英字パターン O, Q のジョイントセル C_H をレベル 4 の節点とする。英字パターン A, B と同様に, 図 5.22 のジョイントセルのうちの 4 つのセル (破線で半分にしたように描いてある節点) を, 半分に割ったと考えてそれぞれ 2 つにすると, 図 5.23 に示すような木になる。全ての枝は n 個もしくは $2n + 1$ 個のセルからなっている。

図 5.21 と図 5.23 に示すような木で表される 13 種の英字パターン D, E, F, G, H, J, M, O, P, Q, S, T, W の一致射撃問題は, 以下に示す 1-2 セグメント木の一致射撃問題として一般化することができる。

1-2 セグメント木とは, 以下の 3 つの条件を満たす木である。

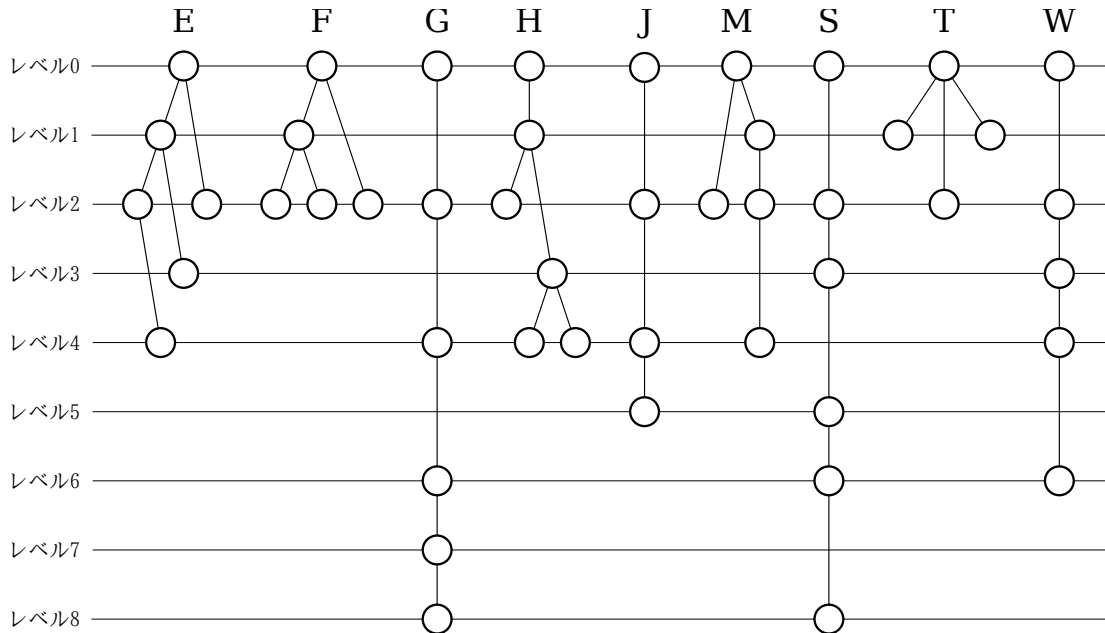


図 5.21: 英字パターン E, F, G, H, J, M, S, T, W に対する木

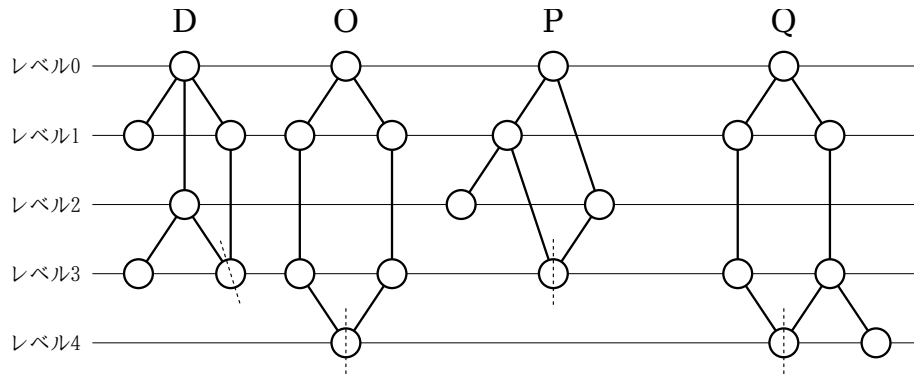


図 5.22: 英字パターン D, O, P, Q に対するグラフ

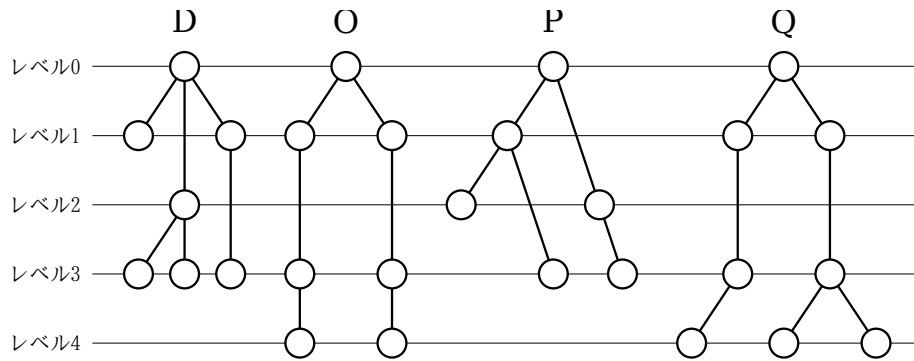


図 5.23: 英字パターン D, O, P, Q に対する木

- 枝は 1 つのセグメント (n 個のセル) または 2 つのセグメント ($2n + 1$ 個のセル) からなる.
- 1 つのセグメントからなる枝と 2 つのセグメントからなる枝がそれぞれ少なくとも 1 つは含まれる.
- 節点も 1 つのセルであるが, セグメントセルと自身を区別できる.

根から節点に至る径路にある枝の数をその節点のレベルというが, 2 つのセグメントからなる枝はレベル 2 つ分として数える. また, 節点のレベルの最大値を 1-2 セグメント木の高さという.

1-2 セグメント木の根を将軍とし, 全てのセルを一斉に射撃状態に遷移させる問題を, 1-2 セグメント木の 1-2 セグメント木の 1-2 セグメント木の一斉射撃問題とする.

5.4.7 1-2 セグメント木の一斉射撃アルゴリズム

高さ h の 1-2 セグメント木の一斉射撃問題に対する一斉射撃アルゴリズムを以下に示す。ここで、 α, m は $n = \alpha m - 1$ を満たす自然数である。

アルゴリズム 5.4

1. 時刻 $t = 0$ に、将軍はつながっている全ての枝に速度 $\frac{1}{1}$ の信号 s_d を送出する。
2. 信号 s_d を受け取った静止状態の節点は状態 Q_r になり、つながっている全ての枝に速度 $\frac{1}{1}$ の信号 s_d を送出する。
3. 状態 Q_r の節点は、全ての枝から同時に信号 s_d を受け取ったとき、つながっている全ての枝に速度 $\frac{1}{1}$ の信号 s_d を送出する。全てではないが少なくとも 1 つの枝から信号 s_d を受け取ったとき、親につながっている枝に速度 $\frac{1}{1}$ の信号 s_f を送出する。
4. 信号 s_f を受け取った節点は、親につながっている枝に速度 $\frac{1}{1}$ の信号 s_f を送出する。
5. 信号 s_d と信号 s_f を同時に受け取ったセルは状態 Q_r になり、親につながっている枝に速度 $\frac{1}{1}$ の信号 s_f を送出する。
6. 将軍は、全ての枝から同時に信号 s_d を受け取ったとき、つながっている全ての枝に速度 $\frac{1}{1}$ の信号 s_d を送出する。信号 s_f を受け取るか、全てではないが少なくとも 1 つの枝から信号 s_d を受け取ったとき、つながっている全ての枝に速度 $\frac{i}{h\alpha}$ の信号 s_i (i は $0 \leq i \leq h\alpha$ を満たす整数) を送出する。
7. 信号 $s_{h\alpha}$ を受け取ったレベル 1 にあるセルは、速度 $\frac{i-\alpha}{h\alpha}$ の信号 T_i (i は $0 \leq i \leq h\alpha$ を満たす整数) を送出する*⁹。
8. 将軍から $\frac{i}{\alpha}(n+1)$ (i は $0 \leq i < h\alpha$ を満たす整数) セルだけ離れた位置にあるセルは、信号 s_i と T_i を同時に受けとって将軍状態に遷移する。同時刻に信号 $T_{h\alpha}$ を受け取ったレベル h の節点も将軍状態に遷移する。これらの将軍状態になったセルはつながっている全ての枝を、両端に将軍がいる 1 次元の一斉射撃問題とみなし

*⁹ 正の速度の信号は葉方向へ、負の速度の信号は根方向へ移動することを示す。

て従来の一斉射撃問題の最小時間アルゴリズムを適用する。

このアルゴリズム 5.4 について以下のことがいえる。

補題 5.7 異なる長さのセグメントがつながっている節点のレベルの最小値を k とする。時刻 $t = (2k + 3)(n + 1)$ までに、2 つのセグメントからなる枝に含まれるレベル 1 にあるセルは自身がジョイントセルであることがわかる。

証明 レベル i ($0 \leq i \leq h$) の節点を N_i とする。

$k = 0$ の場合、時刻 $t = 2(n + 1)$ に、将軍は 1 つのセグメントからなる枝につながっている節点 N_1 から送出された速度 $\frac{1}{1}$ の信号 s_d を受け取って、つながっている全ての枝に速度 $\frac{i}{h\alpha}$ の信号 s_i (i は $0 \leq i \leq h\alpha$ を満たす整数) を送出する。同時刻に、2 つのセグメントからなる枝で将軍とつながっている節点 N_2 は、将軍から送出された速度 $\frac{1}{1}$ の信号 s_d を受け取って状態 Q_r になり、つながっている全ての枝に速度 $\frac{1}{1}$ の信号 s_d を送出する。

時刻 $t = 3(n + 1)$ に、2 つのセグメントからなる枝に含まれるレベル 1 にあるセルは将軍から送出された速度 $\frac{h\alpha}{h\alpha}$ の信号 $s_{h\alpha}$ と節点 N_2 から送出された速度 $\frac{1}{1}$ の信号 s_d を同時に受け取ることで、自身がジョイントセルであることがわかる。

$k = 1$ の場合と $k \geq 2$ でレベル 0 からレベル k にある枝が全て 1 つのセグメントからなる枝である場合、将軍につながっている枝は全て 1 つのセグメントからなる枝である。このため、時刻 $t = 0$ に全てのレベル 1 にあるセルは自身がジョイントセルであることがわかっている。

$k \geq 2$ でレベル 0 からレベル k にある枝が全て 2 つのセグメントからなる枝である場合、節点 N_j ($j = 2, 4, 6, \dots, k$) は、時刻 $t = j(n + 1)$ に信号 s_d を受け取って状態 Q_r になり、つながっている全ての枝に信号 s_d を送出する。このとき、節点 N_{j-4p} (p は $0 \leq p \leq \frac{j}{4}$ を満たす整数) も同時に信号 s_d を受け取って、つながっている全ての枝に信号 s_d を送出する。つまり、節点 N_i と節点 N_{i+2} ($i = 0, 4, 8, \dots$) が交互に信号の送信側と受信側となって信号のやり取りを行う。

時刻 $t = k(n + 1)$ に、節点 N_k は信号 s_d を受け取って状態 Q_r になり、つながってい

る全ての枝に信号 s_d を送出する.

時刻 $t = (k + 1)(n + 1)$ に, 節点 N_{k+1} は節点 N_k から送出された信号 s_d を受け取って状態 Q_r になり, つながっている全ての枝に信号 s_d を送出する.

時刻 $t = (k + 2)(n + 1)$ に, 節点 N_k は節点 N_{k+1} につながっている枝からのみ信号 s_d を受け取って, 親につながっている枝に信号 s_f を送出する. 同時刻に, 節点 N_{k-2} は信号 s_d を受け取って, つながっている全ての枝に信号 s_d を送出する. 節点 N_k は 1 つのセグメントからなる枝をその下に持っているので, 送信側になるのがこれまでより $2(n + 1)$ ステップだけ早くなり, 節点 N_k と N_{k-2} が同時に送信側になる.

時刻 $t = (k + 3)(n + 1)$ に, 節点 N_k と N_{k-2} を結ぶ枝の真ん中にあるセルは, 節点 N_k から送出された信号 s_f と節点 N_{k-2} から送出された信号 s_d を同時に受け取る. これによって, 自身がジョイントセルであることを認識し, 状態 Q_r になり, 親につながっている枝に信号 s_f を送出する.

節点 N_k から始まるタイミングのずれは根まで続く. 根に向かって節点 N_{k-2} , N_{k-4}, \dots, N_0 を結ぶ 2 つのセグメントからなる枝を, 1 つのセグメントからなる枝 2 つと節点に分割しながら信号を伝える.

時刻 $t = (2k + 2)(n + 1)$ に, 将軍 N_0 は信号 s_f を受け取り, つながっている全ての枝に速度 $\frac{i}{h\alpha}$ の信号 s_i (i は $0 \leq i \leq h\alpha$ を満たす整数) を送出する.

時刻 $t = (2k + 3)(n + 1)$ に, 将軍から送出された速度 $\frac{h\alpha}{h\alpha}$ の信号 $s_{h\alpha}$ と節点 N_2 から送出された速度 $\frac{1}{1}$ の信号 s_d を, レベル 1 にあるセルが同時に受け取る. これによって全ての 2 つのセグメントに含まれるレベル 1 にあるセルはジョイントセルであることがわかる. \square

定理 5.7 $n = \alpha m - 1$ を満たす自然数 α, m に対して, 時刻 $t = (h + 2k + 2 + \frac{1}{\alpha})(n + 1)$ に全てのセルが一斉に射撃状態に遷移する.

証明 時刻 $t = k(n + 1)$ に, レベル k の節点は信号 s_d を受け取って状態 Q_r になり, つながっている全ての枝に速度 $\frac{1}{1}$ の信号 s_d を送出する.

時刻 $t = (k + 2)(n + 1)$ に, レベル k の節点は全てではないが少なくとも 1 つの枝から

信号 s_d を受け取って、親につながっている枝に速度 $\frac{1}{\alpha}$ の信号 s_f を送る。

時刻 $t = (2k + 2)(n + 1)$ に、将軍は信号 s_f を受け取って、つながっている全ての枝に速度 $\frac{i}{h\alpha}$ の信号 s_i (i は $0 \leq i \leq h\alpha$ を満たす整数) を送る。ただし $k = 0$ の場合、時刻 $t = 2(n + 1)$ に将軍は全てではないが少なくとも 1 つの枝から信号 s_d を受け取って、信号 s_i を送る。

補題 5.7 より、時刻 $t = (2k + 3)(n + 1)$ に全てのレベル 1 にあるセルは自身がジョイントセルであることを認識することができる。よって、同時刻に全てのレベル 1 にあるセルは将軍から送られた速度 $\frac{h\alpha}{h\alpha}$ の信号 $s_{h\alpha}$ を受け取って、つながっている全ての枝に速度 $\frac{i-\alpha}{h\alpha}$ の信号 T_i (i は $0 \leq i \leq h\alpha$ を満たす整数) を送ることができる。

アルゴリズム 5.4 の (6) から (8) はセグメント木の一斉射撃アルゴリズムを利用して、将軍から送られる速度 $\frac{i}{h\alpha}$ の信号 s_i (i は $0 \leq i < h\alpha$ を満たす整数) と、レベル 1 にあるセルから送られる速度 $\frac{i-\alpha}{h\alpha}$ の信号 T_i (i は $0 \leq i < h\alpha$ を満たす整数) を衝突させることで、全ての枝を均等に分割して同期を取ることができる。これによって、 $(h + \frac{1}{\alpha})(n + 1)$ ステップで全てのセルが一斉に射撃状態に移ることが示される。

従って、時刻 $t = (2k + 2)(n + 1)$ にセグメント木の一斉射撃アルゴリズムを適用することで、時刻 $t = (h + 2k + 2 + \frac{1}{\alpha})(n + 1)$ に全てのセルが一斉に射撃状態に移る。□

このアルゴリズム 5.4 は、 $k = h - 1$ のときに最も時間がかかる。図 5.24 は、この最悪の場合の時間空間図式である。ただし、 $\alpha = 1$ である。横軸はセル空間で、 N_l ($0 \leq l \leq h$) はレベル l の全ての節点を表している。レベル 0 からレベル k にある枝は全て 2 つのセグメントからなる枝である。縦軸は時間を表している。黒い点はそのセルが将軍状態になったことを示している。丸印は枝に含まれるジョイントセルが自身がジョイントセルであることを認識し、節点と同じ動作をすることを示す。

アルゴリズム 5.4 を用いて解くことができる 13 種の英字パターンとその射撃時間を表 5.2 に示す。

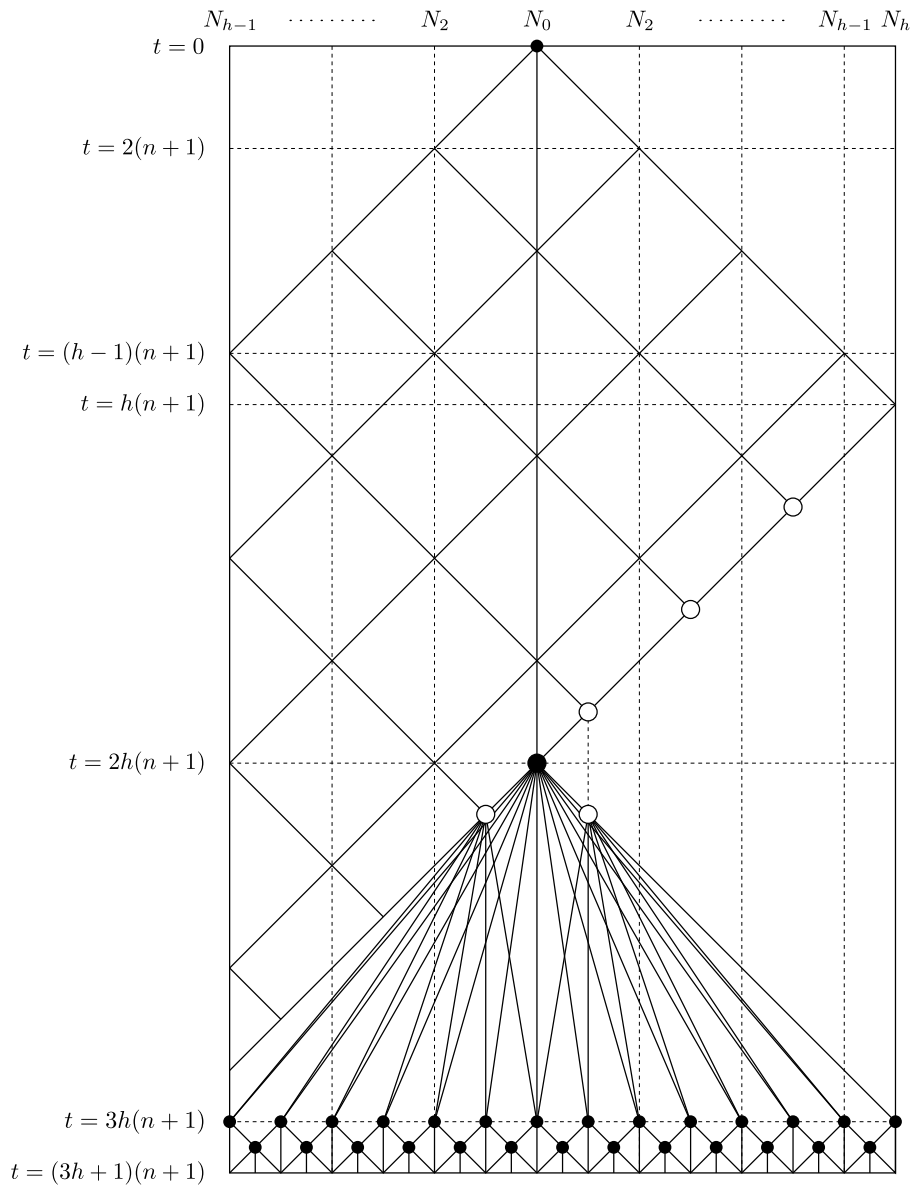


図 5.24: アルゴリズム 5.4 の時間空間図式 ($k = h - 1, \alpha = 1$)

表 5.2: アルゴリズム 5.4 を用いて解ける英字パターンとその射撃時間

パターン	h	k	射撃時間
D	3	0	$(5 + \frac{1}{\alpha})(n + 1)$
E	4	0	$(6 + \frac{1}{\alpha})(n + 1)$
F	2	0	$(4 + \frac{1}{\alpha})(n + 1)$
G	8	6	$(22 + \frac{1}{\alpha})(n + 1)$
H	4	1	$(8 + \frac{1}{\alpha})(n + 1)$
J	5	4	$(15 + \frac{1}{\alpha})(n + 1)$
M	4	0	$(6 + \frac{1}{\alpha})(n + 1)$
O	4	1	$(8 + \frac{1}{\alpha})(n + 1)$
P	3	0	$(5 + \frac{1}{\alpha})(n + 1)$
Q	4	1	$(8 + \frac{1}{\alpha})(n + 1)$
S	8	2	$(14 + \frac{1}{\alpha})(n + 1)$
T	2	0	$(4 + \frac{1}{\alpha})(n + 1)$
W	6	2	$(12 + \frac{1}{\alpha})(n + 1)$

5.4.8 英字パターン R の一斉射撃アルゴリズム

英字パターン R のジョイントセル C_C の左隣のセグメントセルは将軍から送出された信号を直接受け取る。これに対して、ジョイントセル C_E は、将軍から送出された信号を一度ジョイントセル C_D が受け取って送出した信号を受け取る。つまり、将軍が送出した信号とは別の信号を受け取ることができる。この信号の違いによりジョイントセル C_C の左隣のセグメントセルとジョイントセル C_E を区別できる。

英字パターン R の一斉射撃問題に対する一斉射撃アルゴリズムを以下に示す。

アルゴリズム 5.5

1. 時刻 $t = 0$ に、将軍はつながっている全てのセグメントに速度 $\frac{1}{1}$ の信号 s_1 を送出する。

2. 時刻 $t = n + 1$ に, ジョイントセル C_D は信号 s_1 を受け取って状態 Q_1 になり, つながっている全てのセグメントに速度 $\frac{1}{T}$ の信号 s_2 を送出する.
3. 時刻 $t = 2(n + 1)$ に, ジョイントセル C_A, C_E, C_G は信号 s_2 を, ジョイントセル C_C は信号 s_1 を受け取って将軍状態に遷移する.
4. 将軍状態になったジョイントセルは, つながっているセグメントをそれぞれ長さ $n + 2$ のセル列と考えて, 従来の一斉射撃問題の最小時間アルゴリズムを適用する.

時刻 $t = 2(n + 1)$ に, 英字パターン R を構成する全てのセグメントは, その片側に将軍状態のジョイントセルをもつ. これより, 同時刻に従来の一斉射撃問題の最小時間アルゴリズムを適用することで, アルゴリズム 5.5 は英字パターン R の一斉射撃問題を $4(n + 1)$ ステップで解くことができる.

5.4.9 英字パターン I の一斉射撃アルゴリズム

英字パターン I のジョイントセル C_E は, 近傍の状態のみでは自身がジョイントセルであることがわからない. しかし, 英字パターン I の一斉射撃問題は長さ $2n + 3$ の 1 次元セル列の一斉射撃問題であるので, 従来の一斉射撃問題の最小時間アルゴリズムを用いて $4(n + 1)$ ステップで解くことができる.

5.5 副将軍のいる一斉射撃問題

従来の一斉射撃問題に副将軍状態とよばれる新しい状態 G_S を導入し, 副将軍のいる一斉射撃問題を提起する. 副将軍状態 G_S は以下の条件を満たす. ここで, $s_Q \in \{Q, B\}$, $s \in S$, $* \in S \cup \{B\}$ である,

$$\delta(s_Q, G_S, s_Q) = G_S \quad (5.15)$$

$$\delta(*, s, G_S) = \delta(*, s, Q) \quad (5.16)$$

$$\delta(G_S, s, *) = \delta(Q, s, *) \quad (5.17)$$

副将軍状態は静止状態に似た状態である. 副将軍状態の左右のセルが静止状態または

境界信号であるとき、ほかの状態に遷移しない (式 5.15). 副将軍状態の左右にあるセルは、近傍の状態から静止状態と区別することができない (式 5.16, 5.17). しかし、一旦将軍からの信号 (命令) を受け取ると、静止状態とは別の状態に遷移することができる. 将軍からの命令により将軍のように振る舞うことができるため、この状態を副将軍状態とよぶ.

ここで、副将軍のいる一斉射撃問題を定義する. α は任意の自然数とし、 \mathcal{M} は $\{m \in \mathbb{N} \mid 0 < m < \alpha\}$ の部分集合とする. 自然数 l に対して、 $n = 1 + l\alpha$ と仮定する. セル C_i ($0 \leq i \leq n-1$) の初期状態 s_i^0 が以下の条件を満たすようなセル列を $A_{\alpha, \mathcal{M}, l}$ で表す.

$$s_i^0 = \begin{cases} G & \text{if } i = 0 \\ G_S & \text{if } i = lm, m \in \mathcal{M} \\ Q & \text{otherwise} \end{cases} \quad (5.18)$$

そして、ある値 α と集合 \mathcal{M} に対して、副将軍のいる一斉射撃問題は以下のように定義される初期様相の集合 $\text{FSSP}_{\text{SUB}}(\alpha, \mathcal{M})$ をもつ.

$$\text{FSSP}_{\text{SUB}}(\alpha, \mathcal{M}) = \{A_{\alpha, \mathcal{M}, l} \mid l \in \mathbb{N}\}. \quad (5.19)$$

値 α と集合 \mathcal{M} が変化すれば、異なる問題として定義される. 簡単に言えば、遷移関数は値 α と集合 \mathcal{M} を知っているが、値 n , l は知らない.

5.5.1 一斉射撃アルゴリズム

1 つだけ副将軍状態のセルがある場合、セル列のどこに副将軍状態のセルがあれば射撃時間が短くなるか考える.

セル $C_{ml} = C_{\frac{m}{\alpha}(n-1)}$ (ここで m は $0 < m < \alpha$ を満たす自然数) を副将軍とする.

単純なアルゴリズム

直感的に、従来の一斉射撃問題の最小時間アルゴリズムを使うと、副将軍セルがセル列の左から $\frac{2}{3}(n-1)$ の位置にあれば、効率よく一斉射撃できる. $\alpha = 3$, $m = 2$ に対する

副将軍の一斉射撃問題を解く単純なアルゴリズムを示す.

アルゴリズム 5.6

1. 時刻 $t = 0$ に, 将軍 C_0 は射撃信号を右方向に送出する.
2. 時刻 $t = \frac{2}{3}(n-1)$ に, 副将軍 $C_{2l} = C_{\frac{2}{3}(n-1)}$ から送出された速度 $\frac{1}{1}$ の信号を受け取って将軍状態に遷移する. 副将軍は, 将軍のように振る舞い, 右側のセル列 $C_{\frac{2}{3}(n-1)}, C_{\frac{2}{3}(n-1)+1}, \dots, C_{n-1}$ に対して, 最小時間アルゴリズムを適用する. 同時刻に, セル列 $C_0, C_1, \dots, C_{\frac{2}{3}(n-1)}$ の右端のセルとして, 左方向に射撃信号を送出する.

副将軍の左側にある長さ $\frac{2}{3}(n-1) + 1$ のセル列を一斉射撃させるのに $\frac{4}{3}(n-1)$ ステップかかる. 将軍からの速度 $\frac{1}{1}$ の信号を副将軍が受け取るのに $\frac{2}{3}(n-1)$ ステップかかる. さらに, 副将軍の右側にある長さ $\frac{1}{3}(n-1) + 1$ のセル列を一斉射撃させるのに $\frac{2}{3}(n-1)$ ステップかかる. 従って, 長さ n のセル列においてセル $C_{\frac{2}{3}(n-1)}$ が副将軍セルであるとき, アルゴリズム 5.6 の一斉射撃時間は $\frac{4}{3}(n-1)$ ステップである.

図 5.25 にアルゴリズム 5.6 の時間空間図式を示す.

高速なアルゴリズム

副将軍セルがセル列の左から $\frac{2}{3}(n-1)$ の位置にあるとき, 最小時間アルゴリズムを利用することで $\frac{4}{3}(n-1)$ ステップで射撃するアルゴリズムを示した. しかし, セル C_{jl} ($0 \leq j \leq \alpha$) を同期させることで, より高速なアルゴリズムを構成できる.

まず, 副将軍セルがセル列の中央より右側, つまり $\frac{\alpha}{2} < m < \alpha$ の場合を考える. この場合, 以下のようにアルゴリズムを構成することができる.

アルゴリズム 5.7

1. 時刻 $t = 0$ に, 将軍*¹⁰ C_0 は射撃信号と速度 $\frac{m+j}{2m-1}$ ($0 \leq j \leq \alpha - m$) の信号 T_j を送出する.

*¹⁰ 将軍 C_0 は従来の最小時間アルゴリズムの将軍状態とは異なる状態であることに注意されたい.

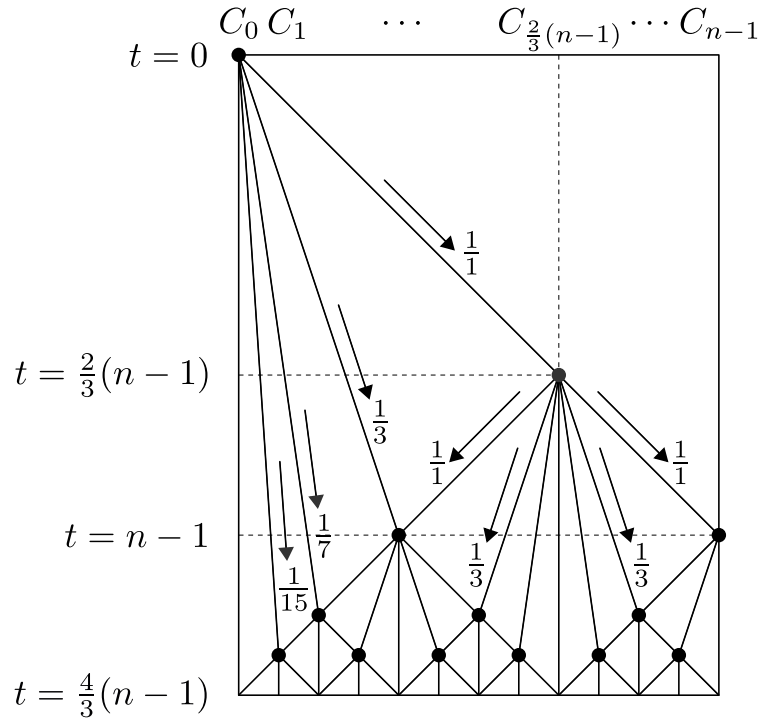


図 5.25: アルゴリズム 5.6 の時間空間図式

2. 時刻 $t = ml$ に，副将軍 C_{ml} は速度 $\frac{1}{1}$ の射撃信号を受け取って，左方向に射撃信号を送出し，右方向に速度 $\frac{j}{m-1}$ ($0 \leq j \leq \alpha - m$) の信号 S_j を送出する．信号 S_0 は移動しない信号である．
3. 時刻 $t = (2m - 1)l$ に，副将軍 C_{ml} は信号 T_0 を受け取る．同時刻に，セル $C_{(m+j)l}$ ($0 < j \leq \alpha - m$) が信号 T_j と S_j を同時に受け取って将軍状態^{*11}に遷移する．そして副将軍の右側において，副将軍 C_{ml} とセル $C_{(m+j)l}$ は副将軍か将軍状態のセルを両端にもつ長さ $l + 1$ の全てのセル列に対して最小時間アルゴリズムを適用する．

副将軍の左側のセル列を一斉射撃時間は，長さ $ml + 1$ のセル列の従来の一斉射撃問題の射撃時間に等しく $2ml$ ステップである．

副将軍の右側のセル列では時刻 $t = (2m - 1)l$ に，副将軍 C_{ml} が信号 T_0 を受け取り，

^{*11} このときの将軍状態は最小時間アルゴリズムの将軍状態と同じ状態である．

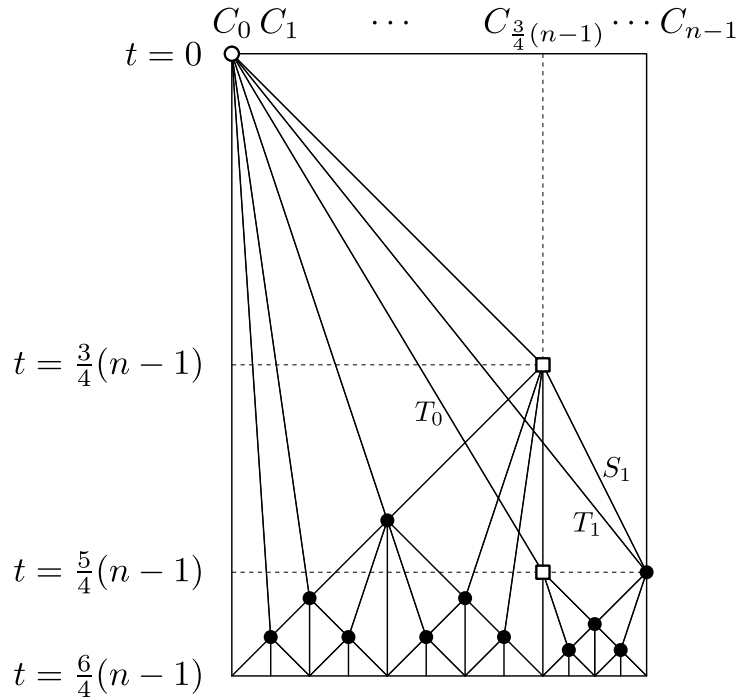


図 5.26: $\alpha = 4, m = 3$ のときのアルゴリズム 5.7 の時間空間図式

セル $C_{(m+j)l}$ ($0 < j \leq \alpha - m$) が信号 T_j と S_j を同時に受け取って将軍状態に遷移する。そして、副将軍が将軍状態のセルを両端にもつ長さ $l + 1$ のセル列に対して、両端から最小時間アルゴリズムを適用するので、この長さ $l + 1$ のセル列射撃時間は l ステップである。従って、副将軍の右側のセル列の一斉射撃時間は $(2m - 1)l + l = 2ml$ ステップである。

結果的に、長さ n のセル列に対してセル $C_{ml} = C_{\frac{m}{\alpha}(n-1)}$ ($\frac{\alpha}{2} < m < \alpha$) が副将軍であるとき、アルゴリズム 5.7 の一斉射撃時間は $2ml = \frac{2m}{\alpha}(n - 1)$ ステップである。この時間は速度 $\frac{1}{\alpha}$ の信号が将軍 C_0 と副将軍 C_{ml} を往復する時間に等しい。従って、このアルゴリズムの一斉射撃時間は m が $\frac{\alpha}{2}$ に近づくほど、つまり副将軍がセル列の中央に近づくほど短くなる。

図 5.26 に、 $\alpha = 4, m = 3$ のときのアルゴリズム 5.7 の時間空間図式を示す。図で、円 (○) は将軍状態、四角 (□) は副将軍状態、黒い点 (●) は最小時間アルゴリズムの将軍状態を表している。

次に、副将軍セルがセル列の中央より左側、つまり $0 < m \leq \frac{\alpha}{2}$ の場合を考える。この

場合、以下のようにアルゴリズムを構成することができる。

アルゴリズム 5.8

1. 時刻 $t = 0$ に、将軍 C_0 は速度 $\frac{j}{\alpha}$ ($0 \leq j \leq \alpha$) の信号 T_j を送出する。
2. 時刻 $t = ml$ に、副将軍 C_{ml} は信号 T_α を受け取って、速度 $\frac{j-m}{\alpha-m}$ ($0 \leq j \leq \alpha$)^{*12} の信号 S_j を左右のセル列に送出する。
3. 時刻 $t = \alpha l = n - 1$ に、右端のセル C_{n-1} は信号 S_α を受け取って将軍状態に遷移する。同時刻に、セル C_{jl} ($0 \leq j \leq \alpha - 1$) は信号 T_j と S_j を同時に受け取って将軍状態に遷移する。将軍状態のセル C_{jl} ($0 \leq j \leq \alpha$) は、つながっている長さ $l + 1$ の全ての将軍状態を両端にもつセル列に対して、最小時間アルゴリズムを適用する。

時刻 $t = n - 1$ に、セル C_{jl} ($0 \leq j \leq \alpha$) が一斉に将軍状態に遷移する。さらに、将軍状態を両端にもつ長さ $l + 1$ のセル列を両端から最小時間アルゴリズムを適用したとき、一斉射撃するのに l ステップかかる。従って、長さ n のセル列においてセル C_{ml} ($0 < m \leq \frac{\alpha}{2}$) が副将軍のとき、アルゴリズム 5.8 の射撃時間は $(n - 1) + l = \frac{\alpha + 1}{\alpha}(n - 1)$ ステップである。このアルゴリズムの射撃時間は副将軍の位置に依存しない。 α の値が大きくなれば、射撃時間は限りなく $n - 1$ に近づく。

例として、図 5.27 に $\alpha = 4$, $m = 1$ のときのアルゴリズム 5.8 の時間空間図式を示す。また、図 5.28 に $\alpha = 4$, $m = 2$ のときのアルゴリズム 5.8 の時間空間図式を示す。

*12 正の速度の信号は右方向に、負の速度の信号は左方向に進むことを表す。

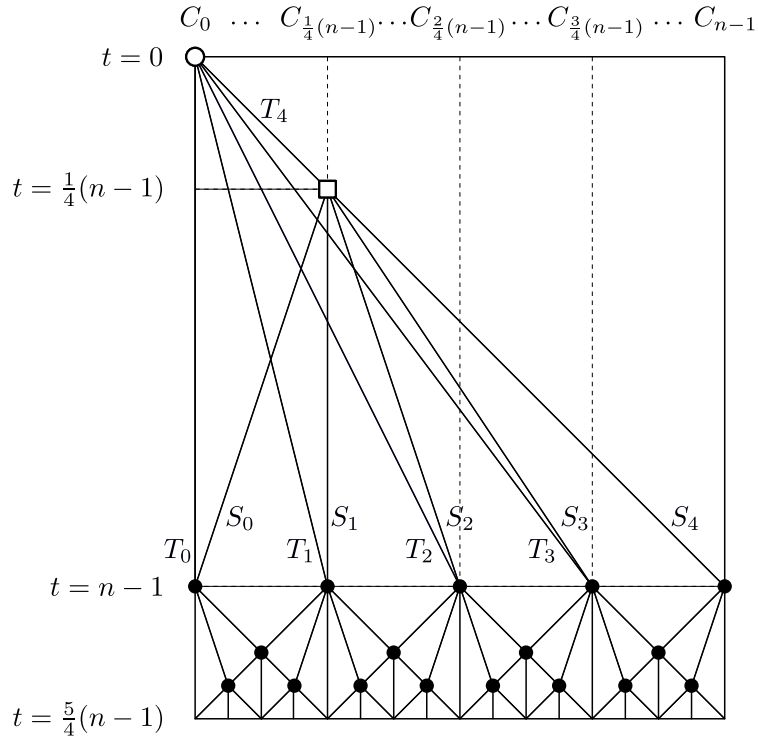


図 5.27: $\alpha = 4$, $m = 1$ のときのアルゴリズム 5.8 の時間空間図式

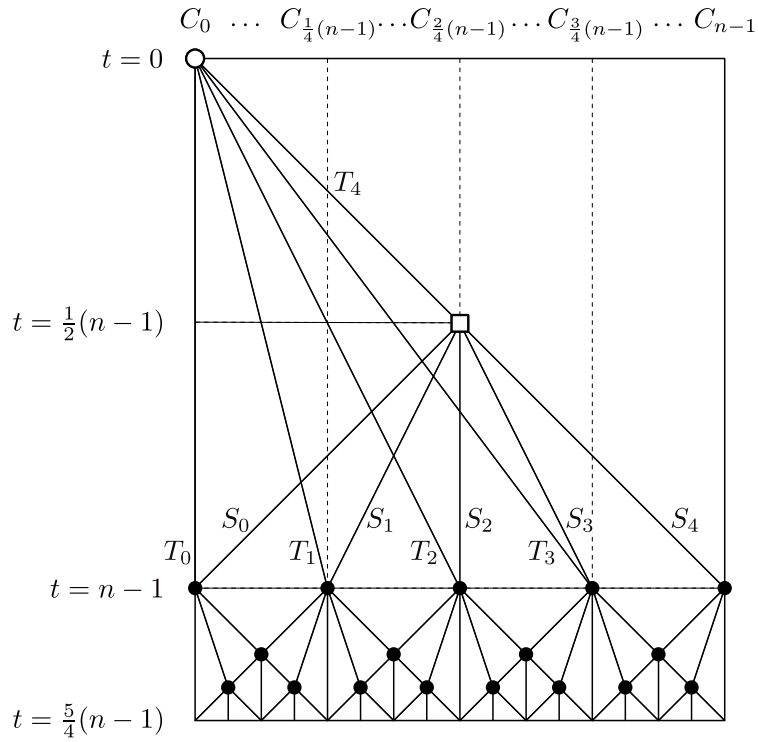


図 5.28: $\alpha = 4$, $m = 2$ のときのアルゴリズム 5.8 の時間空間図式

5.5.2 最小時間

本節では、長さ n のセル列に対する副将軍のいる一斉射撃問題の最小時間について考える。

セル C_{n-1} は時刻 $t = n - 2$ まで静止状態なので、射撃時間が $n - 2$ 以下になることはない。従って、以下の補題が成り立つ。

補題 5.8 長さ n のセル列に対する副将軍のいる一斉射撃問題の任意の解 \mathcal{A} について、 $t_F(n, \mathcal{A}) \geq n - 1$ が成り立つ。

ある長さ $n_0 = \alpha l_0 + 1$ のセル列においてセル $C_{ml_0} = C_{\frac{m}{\alpha}(n_0-1)}$ ($0 < m \leq \frac{\alpha}{2}$) が副将軍であるとき、長さ n_0 のセル列を $n_0 - 1$ ステップで一斉射撃させ、長さ $n \neq n_0$ のセル列を一斉射撃させないような有限オートマトン \mathcal{A}_{n_0} を容易に構成できる。有限オートマトン \mathcal{A}_{n_0} は、以下の状態の集合 \mathcal{S} と遷移関数 δ で構成される。ここで、 $s \in \mathcal{S} - \{X\}$, $* \in \mathcal{S} \cup \{B\}$ である。

$$\mathcal{S} = \{G = S_0, S_1, \dots, S_{ml_0}, \dots, S_{n_0-1} = F, Q, G_S, X\} \quad (5.20)$$

$$\delta(*, S_i, s) = S_{i+1} \quad (5.21)$$

$$\delta(S_{i-1}, Q, *) = S_i \quad \text{if } i \neq ml_0 \quad (5.22)$$

$$\delta(S_{ml_0-1}, G_S, *) = S_{ml_0} \quad (5.23)$$

$$\delta(S_{i-1}, G_S, *) = X \quad \text{if } i \neq ml_0 \quad (5.24)$$

$$\delta(S_{ml_0-1}, Q, *) = X \quad (5.25)$$

$$\delta(*, S_i, X) = X \quad (5.26)$$

$$\delta(X, Q, *) = Q \quad (5.27)$$

$$\delta(X, Q_S, *) = Q_S \quad (5.28)$$

有限オートマトン \mathcal{A}_{n_0} の動作を簡単に説明する。

まず、長さ $n = n_0$ の場合を考える。初期状態が $G = S_0$ である将軍 C_0 は、その状態を右方向に 1 つずつ状態を伝播しながら $S_1, S_2, \dots, S_{ml_0}, \dots, S_{n_0-1}$ と遷移する。その

結果, 時刻 $t = j$ ($0 < j \leq n_0 - 1$) にセル C_0 から C_j は全て状態 S_j になる. そして, 時刻 $t = n_0 - 1$ に全てのセルが状態 $S_{n_0-1} = F$ に遷移する.

次に, 長さ $n < n_0$ の場合を考える. 時刻 $t = ml$ に, 副将軍 C_{ml} が状態 G_S から状態 X に遷移する. 状態 X は左方向に 1 つずつ伝播する. 従って, 時刻 $t = 2ml \leq n_0 - 1$ に全てのセルが状態 X もしくは静止状態 Q である.

同様に, 長さ $n > n_0$ の場合を考える. 時刻 $t = ml_0$ に, セル C_{ml_0} が状態 Q から状態 X に遷移する. 状態 X が左方向に伝播し, 時刻 $t = 2ml_0 \leq n_0 - 1$ に全てのセルが状態 X , 静止状態 Q , 副将軍状態 G_S のいずれかである.

A_{n_0} と適当な解 \mathcal{A} を同時に模倣する有限オートマトン \mathcal{A}' はセグメント列の一斉射撃問題の解であり, $t_F(n_0, \mathcal{A}') = h(n_0 + 1)$ である. 以上により, 以下の定理が成り立つ.

定理 5.8 副将軍セルがセル列の中央もしくは中央よりも左側にある場合, $t_{F \min}(n) = n - 1$ が成り立つ.

5.5.3 最小時間解

副将軍セルがセル列の中央もしくは中央よりも左側にある場合, 長さ n のセル列における副将軍のいる一斉射撃問題の最小時間が $n - 1$ ステップであることを示したが, 全ての長さ n に対して最小時間で射撃させるような最小時間解が存在するか考える.

以下の補題はそのような最小時間解が存在しないことを示している.

補題 5.9 有限オートマトン $\mathcal{A} = (\mathcal{S}, \delta)$ が長さ n のセル列における副将軍のいる一斉射撃問題の解であるとき, もし $n - ml > |\mathcal{S}|^2$ であれば, $t_F(n, \mathcal{A}) > n - 1$ が成り立つ.

証明 $t_F(n, \mathcal{A}) = n - 1$ であると仮定し背理法を用いて証明する.

時刻 t のセル C_i の状態を s_i^t とする. もし $n - ml > |\mathcal{S}|^2$ であれば, $s_{i-1}^i = s_{j-1}^j$, $s_i^i = s_j^j$ を満たす異なる 2 つの整数 i, j ($ml \leq i < j \leq n - 2$) が存在する. これは, 副将軍 C_{ml} よりも右側のセル列で 2 つの隣り合ったセル $C_{i-1}C_i$ と $C_{j-1}C_j$ において, 時刻 i の隣り合ったセル $C_{i-1}C_i$ の状態 $s_{i-1}^i s_i^i$ と時刻 j の隣り合ったセル $C_{j-1}C_j$ の状態

$s_{j-1}^j s_j^j$ が等しくことを示している.

$ml \leq i < n-2$ のとき $s_{i+1}^{i+1} = \delta(s_i^i, Q, Q)$ なので, $0 \leq k \leq n-2-j$ のとき $s_{i+k}^{i+k} = s_{j+k}^{j+k}$ である. さらに, $ml < i \leq n-2$ のとき $s_i^{i+1} = \delta(s_{i-1}^i, s_i^i, Q)$ なので, $0 \leq k \leq n-2-j$ のとき $s_{i+k}^{i+k+1} = s_{j+k}^{j+k+1}$ である. 従って, $s_{n-2+i-j}^{n-1+i-j} = s_{n-2}^{n-1}$ である.

射撃時間が $n-1$ ステップであるという仮定から $s_{n-2}^{n-1} = F$ が成り立つ. 従って, $s_{n-2+i-j}^{n-1+i-j} = s_{n-2}^{n-1} = F$ が成り立つ. セル $C_{n-2+i-j}$ が時刻 $t = n-1+i-j < n-1$ に射撃状態に遷移し, 仮定に矛盾することから \mathcal{A} は解ではない. \square

以上より, 以下の定理が成り立つ.

定理 5.9 副将軍セルがセル列の中央もしくは中央よりも左側にある場合, 副将軍のいる一斉射撃問題の最小時間解は存在しない.

5.6 まとめ

本章では, 理論的興味から, 7セグメントディスプレイ上の数字パターンの一斉射撃問題, 16セグメントディスプレイ上の英字パターンの一斉射撃問題, 副将軍のいる一斉射撃問題を提起した.

7セグメントディスプレイ上の数字パターンの一斉射撃問題では, いくつかの数字パターンの一斉射撃問題を一般化することで, セグメント木を定義しセグメント木の一斉射撃問題を提起した. セグメント木は1次元セル列をジョイントセルによって等間隔に分割したセル列に帰着することができ, このようなセル列をセグメント列として定義し, セグメント列の一斉射撃問題を提起した. さらに, セグメント列の一斉射撃問題と残りの数字パターンの一斉射撃問題を解くアルゴリズムをそれぞれ示した.

長さ $h(n+1)+1$ のセグメント列に対して, $(h + \frac{1}{\alpha})(n+1)$ (ただし, $\alpha \in \mathbb{N}$) ステップで一斉射撃するアルゴリズムを示した. 従って, 開発したアルゴリズムは, 数字パターン 4, 8, 9 の一斉射撃問題を $(3 + \frac{1}{\alpha})(n+1)$ ステップで, 数字パターン 3, 5, 6 の一斉射撃問題を $(4 + \frac{1}{\alpha})(n+1)$ ステップで, 数字パターン 2 の一斉射撃問題を $(5 + \frac{1}{\alpha})(n+1)$ ステップで解くことができる. さらに, セグメント列の一斉射撃問題の最小時間は $h(n+1)$

であるが、全ての長さのセグメント列を最小時間で射撃させるような最小時間解は存在しないことを示した。

数字パターン 0, 7 の一斉射撃問題を $(3 + \frac{1}{\alpha})(n+1)$ ステップで解くアルゴリズムを示した。数字パターン 0, 7 の一斉射撃問題の最小時間は $3(n+1)$ であるが、全ての長さのセグメント列を最小時間で射撃させるような最小時間解は存在しないことを示した。

数字パターン 1 の一斉射撃問題は、長さ $2n+3$ の従来の一斉射撃問題であり、最小時間アルゴリズムを用いて $4(n+1)$ で解けることを示した。

16 セグメントディスプレイ上の英字パターンの一斉射撃問題では、まず、11 種の英字パターン A, B, C, K, L, N, U, V, X, Y, Z の一斉射撃問題が、セグメント木の一斉射撃問題に帰着できることを示した。次に、13 種の英字パターン D, E, F, G, H, J, M, O, P, Q, S, T, W の一斉射撃問題が 1-2 セグメント木の一斉射撃問題に帰着できることを示し、高さ h の 1-2 セグメント木の一斉射撃問題に対して、射撃時間 $(h + 2k + 2 + \frac{1}{\alpha})(n+1)$ の一斉射撃アルゴリズムを示した。ここで、 k は異なる長さの枝がつながっている節点のレベルの最小値であり、 α, m は $n = \alpha m - 1$ を満たす自然数である。さらに、英字パターン I の一斉射撃問題は、長さ $2n+3$ の従来の一斉射撃問題であり、最小時間アルゴリズムを用いて $4(n+1)$ で解けることを示した。最後に、英字パターン R の一斉射撃問題を $4(n+1)$ ステップで解くことができるアルゴリズムを示した。

副将軍のいる一斉射撃問題では、副将軍状態を追加した問題を提起し、副将軍セルが 1 つだけある場合について考えた。

副将軍セルが、セル列の中央より右側にある場合、長さ n のセル列を $2ml$ ステップで一斉射撃させるアルゴリズムを示した。このアルゴリズムの射撃時間は、副将軍セルが中央に近づくほど短くなる。

副将軍セルが、セル列の中央もしくは中央より左側にある場合、長さ n のセル列を $\frac{\alpha+1}{\alpha}(n-1)$ (ただし、 $\alpha \in \mathbb{N}$) ステップで一斉射撃させるアルゴリズムを示した。このアルゴリズムの射撃時間は、副将軍セルの位置に依存せず、 α が大きくなることで $n-1$ ステップに近づくことを示した。

さらに、副将軍のいる一斉射撃問題の最小時間が $n-1$ ステップであることを示した。

しかし、全ての長さ n に対して最小時間で射撃させるような最小時間解は存在しないことを示した。つまり、本章で示したアルゴリズムが限りなく最小時間に近い解である。

本章で扱ったそれぞれの問題に対して状態数がいくつ必要か、1-2 セグメント木の一斉射撃問題の最小時間と最小時間解の解明が今後の課題である。

参考文献

- [1] Moore, E. F.: The Firing Squad Synchronization Problem, in Moore, E. F. ed., *Sequential Machines, Selected Papers*, pp. 213–214, Addison-Wesley, Reading MA. (1964).
- [2] Minsky, M. and McCarthy, J.: *Computation : Finite and infinite machines*, Prentice Hall, pp. 28–29 (1967).
- [3] Goto, E.: A minimal-time solution of the firing squad synchronization problem, *Course Notes for Applied Mathematics 298*, pp. 52–59 (1962).
- [4] Waksman, A.: An optimum solution to the firing squad synchronization problem, *Information and Control*, Vol. 9, pp. 66–78 (1966).
- [5] Balzer, R.: An 8-state minimal time solution to the firing squad synchronization problem, *Information and Control*, Vol. 10, pp. 22–42 (1967).
- [6] Gerken, H.-D.: Über Synchronisationsprobleme bei Zellularautomaten, *Diplomarbeit, Technische Universität Braunschweig*, pp. 1–50 (1987).
- [7] Mazoyer, J.: A six states minimal time solution to the firing squad synchronization problem, *Theoretical Computer Science*, Vol. 50, pp. 183–238 (1987).
- [8] Culik, K.: Variations of the firing squad problem and applications, *Information Processing Letters*, Vol. 30, pp. 153–157 (1989).
- [9] Shinahr, I.: Two and three-dimensional firing squad synchronization problems, *Information and Control*, Vol. 24, pp. 163–180 (1974).
- [10] Kobayashi, K.: The firing squad synchronization problem for two dimensional arrays, *Information and Control*, Vol. 34, pp. 153–157 (1977).

-
- [11] Hisaoka, M., Yamada, H., Maeda, M., Worsch, T. and Umeo, H.: A design of firing squad synchronization algorithm for multi-general problems and their implementations, Technical Report COMP2002-133, IEICE (2003).
 - [12] Hisaoka, M., Yamada, H., Maeda, M., Worsch, T. and Umeo, H.: An optimum-time firing squad synchronization algorithm for multi-general problem and its implementation, Technical Report COMP2003-18, IEICE (2003).
 - [13] Schmid, H. and Worsch, T.: The firing squad synchronization problem with many generals for one-dimensional CA, *IFIP TCS 2004*, pp. 111–124 (2004).
 - [14] Róka, Z.: The firing squad synchronization problem on Cayley graphs, *Proc. of MFCS'95, Prague (Czech Republic), 1995. Lecture Notes in Computer Science*, Vol. 969, pp. 402–411 (1995).
 - [15] Nishitani, Y. and Honda, N.: The firing squad synchronization problem for graphs, *Theoretical Computer Science*, Vol. 14, pp. 39–61 (1981).
 - [16] Imai, K. and Morita, K.: Firing squad synchronization problem in reversible cellular automata, *Theoretical Computer Science*, Vol. 165, pp. 475–482 (1996).
 - [17] Imai, K., Morita, K. and Sako, K.: *Firing squad synchronization problem in number-conserving cellular automata*, Proc. of the IFIP Workshop on Cellular Automata, Santiago (Chile) (1998).
 - [18] Umeo, H.: A design of cellular algorithms for 1-bit inter-cell communications and related cellular algorithms, *Proc. of MCU'98*, Vol. 1, pp. 210–227 (1998).

第 6 章

結論

本論文では，セルオートマトンに関する問題のうち，不完全な同期下のセルオートマトンの時空間パターンの分類，雪の結晶の類似パターン生成，セルオートマトン上で定義される一斉射撃問題の拡張の 3 つの問題に対してそれぞれ一定の解を与えた．

第一に，不完全な同期下のセルオートマトンの時空間パターンと，この両極に位置する同期セルオートマトンと非同期セルオートマトンの時空間パターンを比較することによって 5 つのグループに分類した．表 6.1 に示すように，88 の単純セルオートマトンのうち，27, 37, 19, 1, 4 のセルオートマトンがそれぞれグループ 1, 2, 3, 4, 5 に分類された．

グループ 1 同期 CA, 非同期 CA と同様なパターンに収束する

グループ 2 遷移確率 p の値によって徐々にパターンが変化する

グループ 3 非同期 CA と同様なパターンに収束する

グループ 4 同期 CA と同様なパターンに収束する

グループ 5 同期 CA, 非同期 CA と異なるパターンに収束する

また，不完全な同期下のセルオートマトンにおける遷移関数と時空間パターンの特徴との関係を明らかにした．

大多数のセルオートマトンはグループ 1, 2, 3 に属し，少数のセルオートマトンがグループ 4, 5 に属している．グループ 4, 5 の少数のセルオートマトンでは，非同期セル

表 6.1: 不完全な同期下にある単純セルオートマトンの分類

	0	4	8	12	13	28	32	36	40	44
グループ 1	72	76	77	78	104	108	128	132	136	140
	160	164	168	172	200	204	232			
	1	3	7	9	11	18	19	22	23	25
グループ 2	26	27	30	33	35	37	41	43	45	46
	50	51	54	57	58	60	62	90	105	106
	110	122	126	142	146	150	178			
	2	5	10	14	15	24	29	34	42	56
グループ 3	73	74	94	130	138	152	162	170	184	
グループ 4	156									
グループ 5	6	38	134	154						

オートマトンと異なる時空間パターンに収束している。その要因を調べたところ、不完全な同期下で発生する局所的な同期によって、遷移関数と時空間パターンの特徴の関係が変化したためであることを明らかにした。

グループ 1, 2, 3 の中でも、局所的な同期によって非同期では起こりえないパターンが発生するセルオートマトンも多く含まれている。しかし、それらは全体の時空間パターンに決定的な影響を与えることなく、結果として非同期と同等の時空間パターンへと収束している。つまり、局所的な同期がミクロ的なパターンの相違を引き起こしても、その違いがマクロ的な時空間パターンを変えるまでに至ることは稀であると考えられる。

Wolfram は単純セルオートマトンの時空間パターンを 4 つのクラスに分類したが、定量的な分類手法ではないため、さまざまな分類手法が提案されている。不完全な同期下のセルオートマトンの時空間パターンの分類についても、同様の議論があると考えられる。このため、定量的な分類手法を構築することが今後の課題である。

第二に、雪の結晶の類似パターン生成問題について、轡田のモデルを用いて類似パターンを生成した。環境の大きな変化をセルオートマトンの遷移関数の入れ換えに対応させる

ことで、雪の結晶の成長過程において環境が大きく変化してできるといわれている複雑な形をした結晶の類似パターンを生成した。

また、環境の変化を考慮したモデルの遷移規則から、六方対称性を放棄することで、大気中の塵等が結晶に付着して空気抵抗が微妙に変わり結晶の各辺の成長速度が変化してできる六方非対称な結晶の類似パターンを生成した。

気象環境の変化や結晶が受ける微妙な空気抵抗の変化を、数値計算を用いた手法に導入することは難しいが、セルオートマトンを用いたモデルであれば容易である。セルオートマトンの遷移関数の物理的な意味が明確になっていないことが問題であるが、遷移規則とその組み合わせは膨大にあるため、これらの遷移規則と生成パターンをディープラーニング等の手法を用いて解析することで、遷移規則が持つ物理的な意味を解明することが今後の課題である。

第三に、一斉射撃問題の拡張について、7セグメントディスプレイ上の数字パターンの一斉射撃問題、16セグメントディスプレイ上の英字パターンの一斉射撃問題、副将軍のいる一斉射撃問題を提起した。

7セグメントディスプレイ上の数字パターンの一斉射撃問題では、いくつかの数字パターンの一斉射撃問題を一般化することで、セグメント木を定義しセグメント木の一斉射撃問題を提起した。セグメント木は1次元セル列をジョイントセルによって等間隔に分割したセル列に帰着することができ、このようなセル列をセグメント列として定義し、セグメント列の一斉射撃問題を提起した。さらに、セグメント列の一斉射撃問題と残りの数字パターンの一斉射撃問題を解くアルゴリズムを示した。

長さ $h(n+1)+1$ のセグメント列に対して、 $h(n+1)+\frac{1}{\alpha}(n+1)$ (ただし、 $\alpha \in \mathbb{N}$) ステップで一斉射撃するアルゴリズムを示した。さらに、セグメント列の一斉射撃問題の最小時間は $h(n+1)$ であるが、全ての長さのセグメント列を最小時間で射撃させるような最小時間解は存在しないことを示した。

数字パターン 0, 7 の一斉射撃問題を $3(n+1)+\frac{1}{\alpha}(n+1)$ ステップで解くアルゴリズムを示した。数字パターン 0, 7 の一斉射撃問題の最小時間は $3(n+1)$ であるが、全ての長さのセグメント列を最小時間で射撃させるような最小時間解は存在しないことを示した。

表 6.2: 7 セグメントディスプレイ上の数字パターンの一斉射撃問題の射撃時間

パターン	射撃時間
0	$3(n+1) + \frac{1}{\alpha}(n+1)$
1	$4(n+1)$
2	$5(n+1) + \frac{1}{\alpha}(n+1)$
3	$4(n+1) + \frac{1}{\alpha}(n+1)$
4	$3(n+1) + \frac{1}{\alpha}(n+1)$
5	$4(n+1) + \frac{1}{\alpha}(n+1)$
6	$4(n+1) + \frac{1}{\alpha}(n+1)$
7	$3(n+1) + \frac{1}{\alpha}(n+1)$
8	$3(n+1) + \frac{1}{\alpha}(n+1)$
9	$3(n+1) + \frac{1}{\alpha}(n+1)$

また、数字パターン 1 の一斉射撃問題は、長さ $2n+3$ の従来の一斉射撃問題であり、最小時間アルゴリズムを用いて $4(n+1)$ で解けることを示した。

本論文で示した 7 セグメントディスプレイ上の数字パターンの一斉射撃問題の射撃時間を表 6.2 に示す。

16 セグメントディスプレイ上の英字パターンの一斉射撃問題では、まず、11 種の英字パターン A, B, C, K, L, N, U, V, X, Y, Z の一斉射撃問題が、セグメント木の一斉射撃問題に帰着できることを示した。次に、13 種の英字パターン D, E, F, G, H, J, M, O, P, Q, S, T, W の一斉射撃問題が 1-2 セグメント木の一斉射撃問題に帰着できることを示し、高さ h の 1-2 セグメント木の一斉射撃問題に対して、射撃時間 $(h+2k+2+\frac{1}{\alpha})(n+1)$ の一斉射撃アルゴリズムを示した。ここで、 k は異なる長さの枝がつながっている節点のレベルの最小値であり、 α, m は $n = \alpha m - 1$ を満たす自然数である。さらに、英字パターン I の一斉射撃問題は、長さ $2n+3$ の従来の一斉射撃問題であり、最小時間アルゴリズムを用いて $4(n+1)$ で解けることを示した。最後に、英字パターン R の一斉射撃問題を $4(n+1)$ ステップで解くアルゴリズムを示した。

本論文で示した 16 セグメントディスプレイ上の英字パターンの一斉射撃問題の射撃時間を表 6.3 に示す.

副将軍状態を追加し副将軍のいる一斉射撃問題を提起した. 副将軍セルが 1 つだけある場合の一斉射撃アルゴリズムとその射撃時間を明らかにした.

副将軍セルが, セル列の中央より右側にある場合, 長さ n のセル列を $2ml$ ステップで一斉射撃させるアルゴリズムを示した. このアルゴリズムの射撃時間は, 副将軍セルが中央に近づくほど短くなる.

副将軍セルが, セル列の中央もしくは中央より左側にある場合, 長さ n のセル列を $\frac{\alpha+1}{\alpha}(n-1)$ (ただし, $\alpha \in \mathbb{N}$) ステップで一斉射撃させるアルゴリズムを示した. このアルゴリズムの射撃時間は, 副将軍セルの位置に依存せず, α が大きくなることで $n-1$ ステップに近づくことを示した.

さらに, 副将軍のいる一斉射撃問題の最小時間が $n-1$ ステップであることを示した. しかし, 全ての長さ n に対して最小時間で射撃させるような最小時間解は存在しないことを示した. つまり, 本論文で示したアルゴリズムが限りなく最小時間に近い解である.

本論文で扱ったそれぞれの一斉射撃問題に対して状態数がいくつ必要か, 1-2 セグメント木の一斉射撃問題の最小時間と最小時間解の解明が今後の課題である.

表 6.3: 16 セグメントディスプレイ上の英字パターンの一斉射撃問題の射撃時間

パターン	射撃時間
A	$(3 + \frac{1}{\alpha})(n + 1)$
B	$(3 + \frac{1}{\alpha})(n + 1)$
C	$(6 + \frac{1}{\alpha})(n + 1)$
D	$(5 + \frac{1}{\alpha})(n + 1)$
E	$(6 + \frac{1}{\alpha})(n + 1)$
F	$(4 + \frac{1}{\alpha})(n + 1)$
G	$(22 + \frac{1}{\alpha})(n + 1)$
H	$(8 + \frac{1}{\alpha})(n + 1)$
I	$4(n + 1)$
J	$(15 + \frac{1}{\alpha})(n + 1)$
K	$(3 + \frac{1}{\alpha})(n + 1)$
L	$(4 + \frac{1}{\alpha})(n + 1)$
M	$(6 + \frac{1}{\alpha})(n + 1)$
N	$(4 + \frac{1}{\alpha})(n + 1)$
O	$(8 + \frac{1}{\alpha})(n + 1)$
P	$(5 + \frac{1}{\alpha})(n + 1)$
Q	$(8 + \frac{1}{\alpha})(n + 1)$
R	$4(n + 1)$
S	$(14 + \frac{1}{\alpha})(n + 1)$
T	$(4 + \frac{1}{\alpha})(n + 1)$
U	$(6 + \frac{1}{\alpha})(n + 1)$
V	$(4 + \frac{1}{\alpha})(n + 1)$
W	$(12 + \frac{1}{\alpha})(n + 1)$
X	$(2 + \frac{1}{\alpha})(n + 1)$
Y	$(2 + \frac{1}{\alpha})(n + 1)$
Z	$(6 + \frac{1}{\alpha})(n + 1)$

付録 A

単純セルオートマトンの時空間パターン

第 3 章では, Fatès 等の定義した不完全な同期下のセルオートマトンの時空間パターンと, この両極に位置する同期セルオートマトンと非同期セルオートマトンの時空間パターンを比較することによって, 不完全な同期下の単純セルオートマトンを 5 つのグループに分類した. 本付録では, 分類対象とした 88 のセルオートマトンの時空間パターンを全て示す.

A.1 グループ 1: 同期 CA, 非同期 CA と同様なパターンに収束する

グループ 1 は, 同期セルオートマトン, 非同期セルオートマトンと同様なパターンに収束するものである. このグループに属している 27 のセルオートマトンの時空間パターンを図 A.1–A.27 に示す.



(a) 非同期 CA

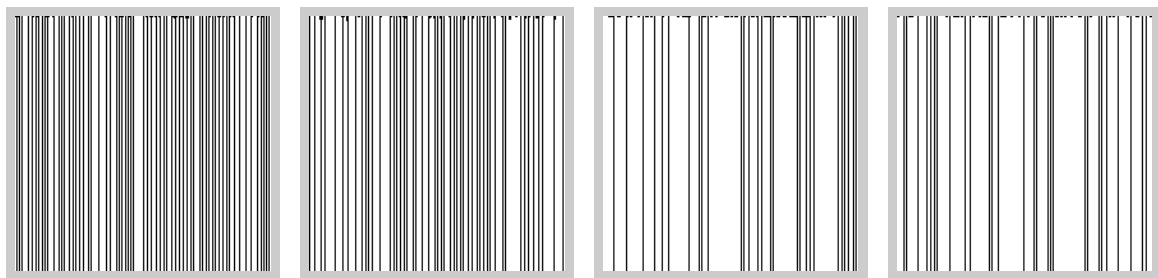
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.1: #0 の時空間パターン (グループ 1)



(a) 非同期 CA

(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.2: #4 の時空間パターン (グループ 1)



(a) 非同期 CA

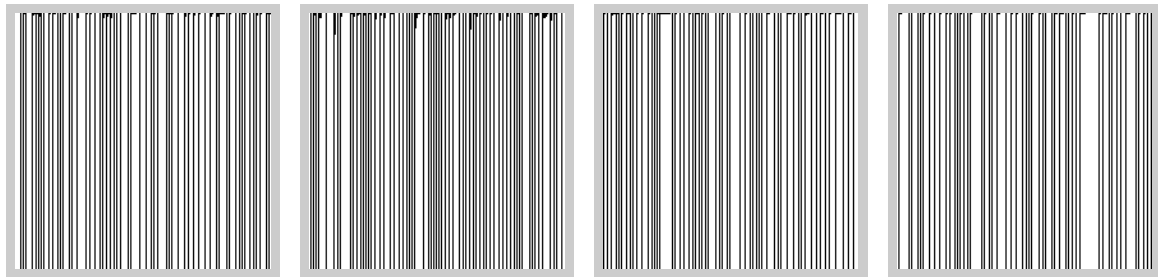
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

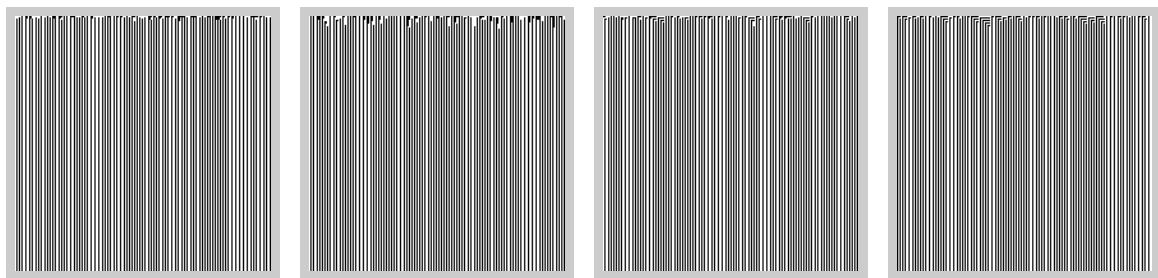
図 A.3: #8 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

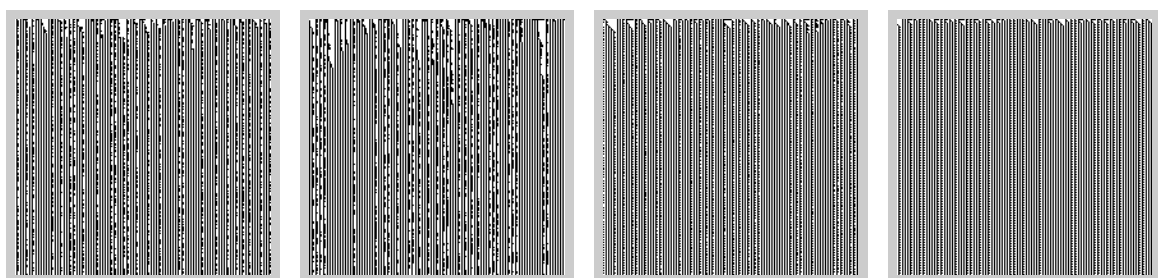
図 A.4: #12 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

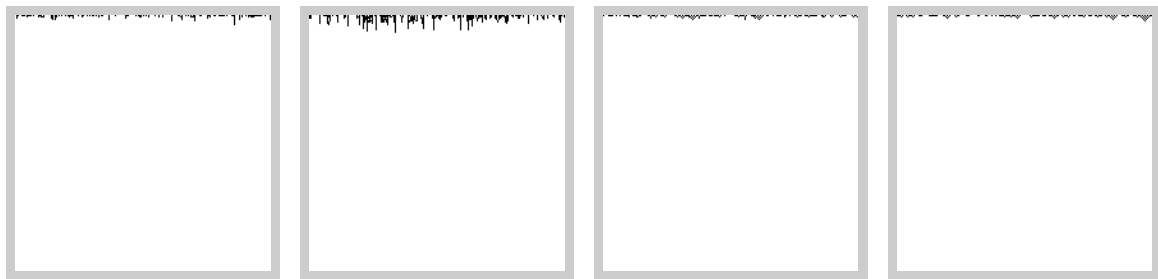
図 A.5: #13 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

図 A.6: #28 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

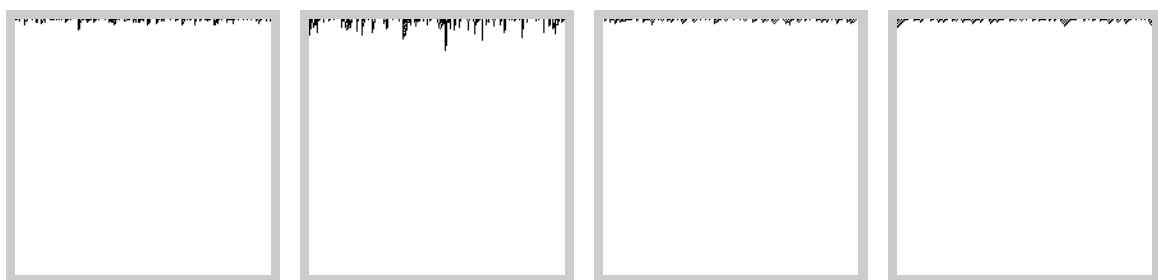
図 A.7: #32 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

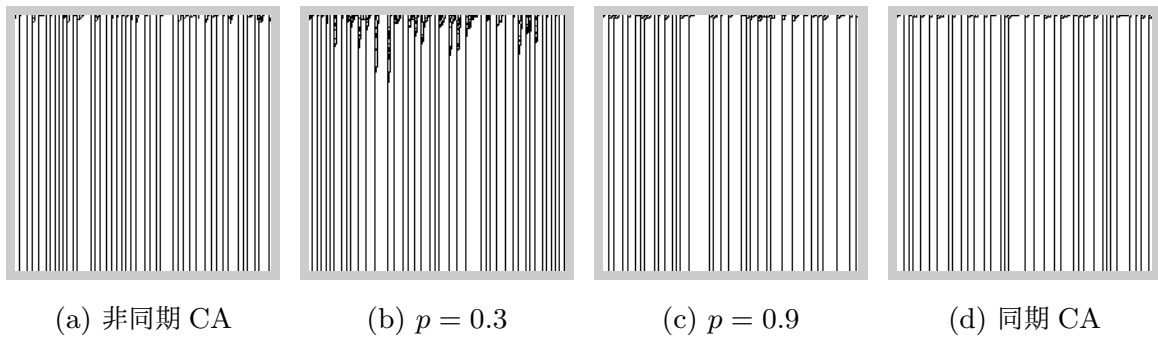
図 A.8: #36 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

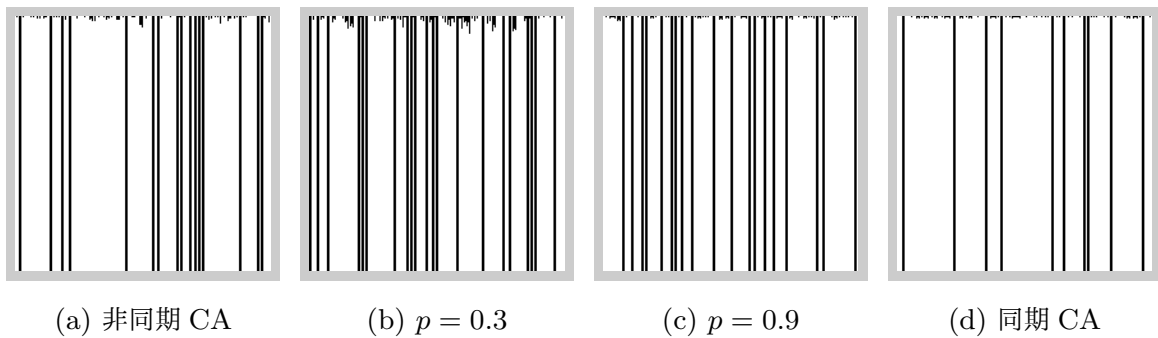
図 A.9: #40 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

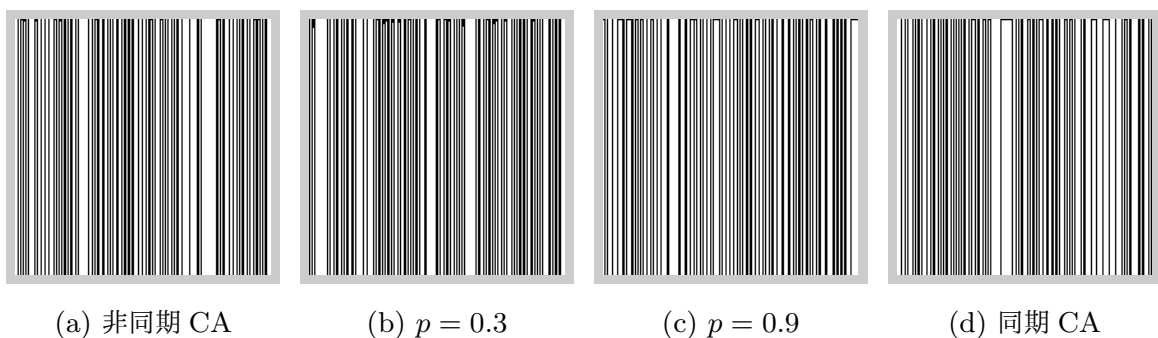
図 A.10: #44 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

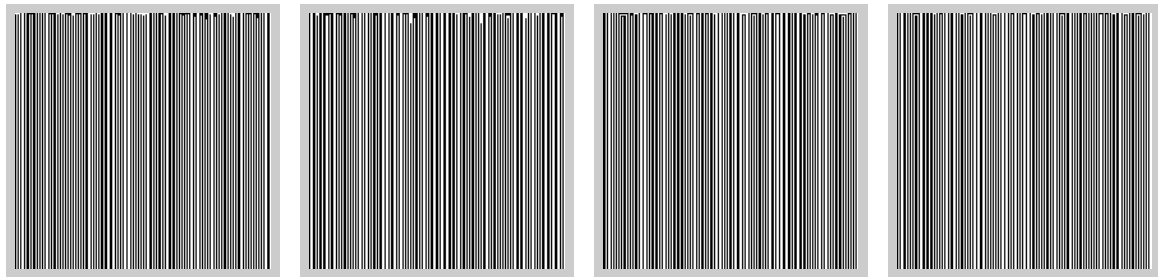
図 A.11: #72 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

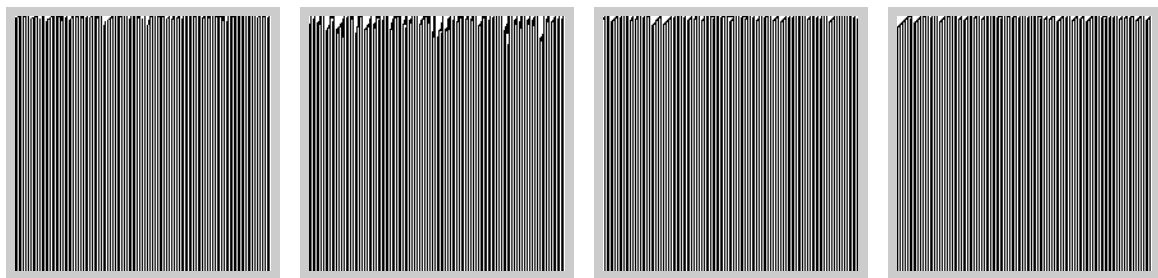
図 A.12: #76 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

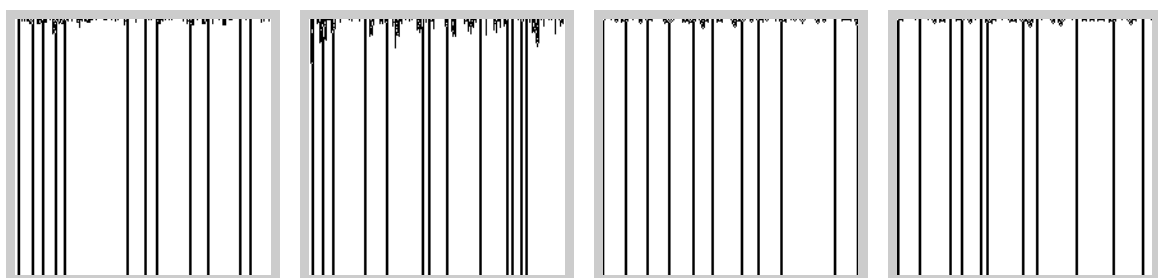
図 A.13: #77 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

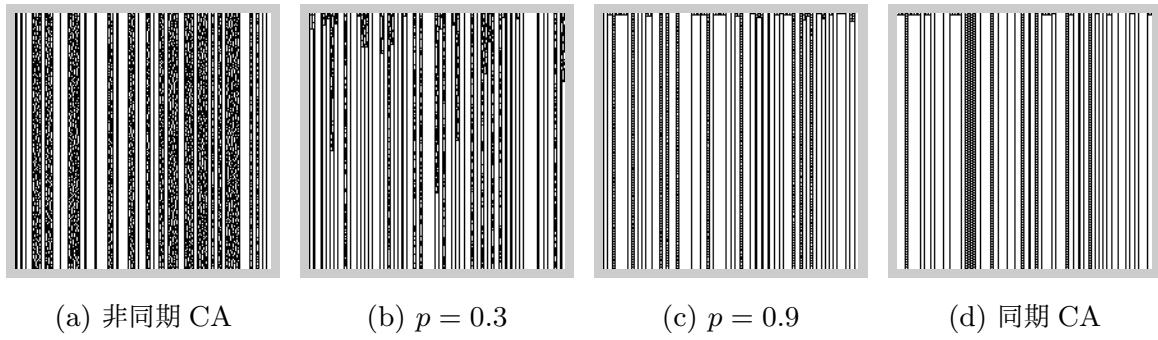
図 A.14: #78 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

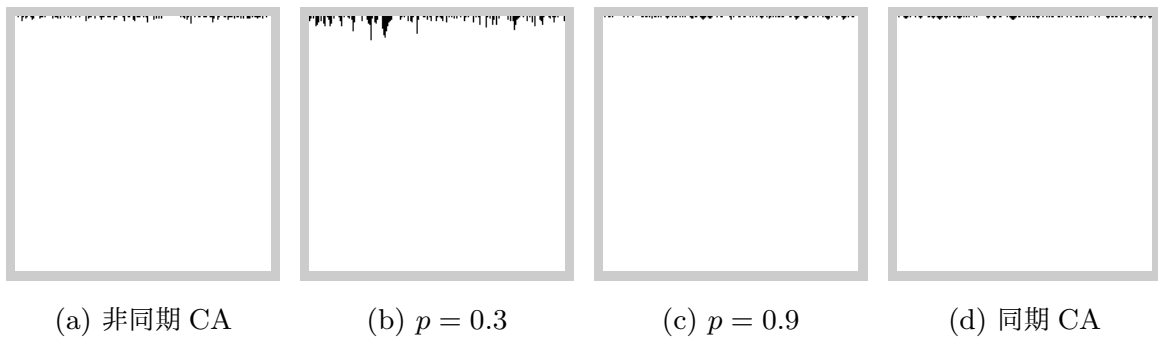
不完全な同期下の CA

図 A.15: #104 の時空間パターン (グループ 1)



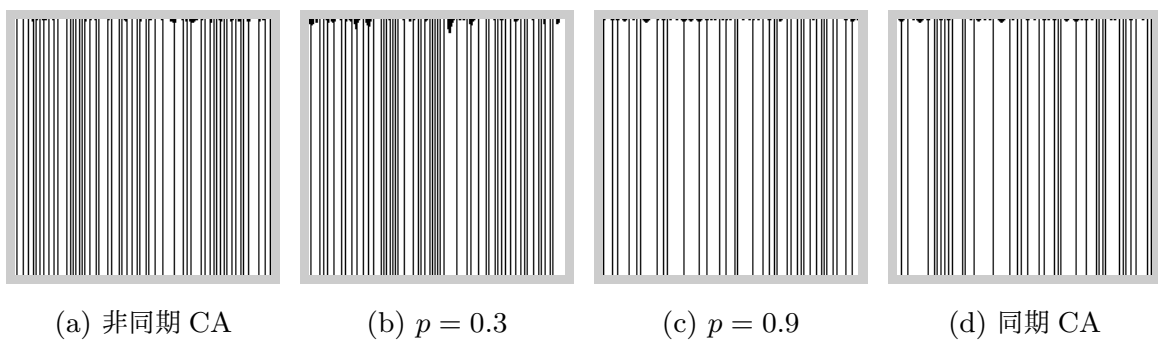
不完全な同期下の CA

図 A.16: #108 の時空間パターン (グループ 1)



不完全な同期下の CA

図 A.17: #128 の時空間パターン (グループ 1)



不完全な同期下の CA

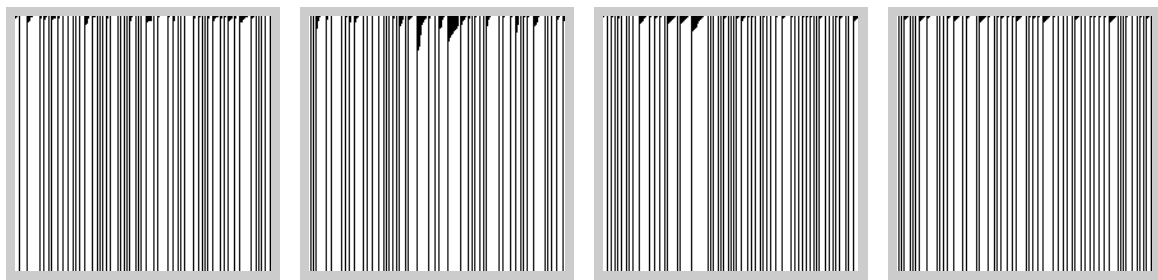
図 A.18: #132 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

図 A.19: #136 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

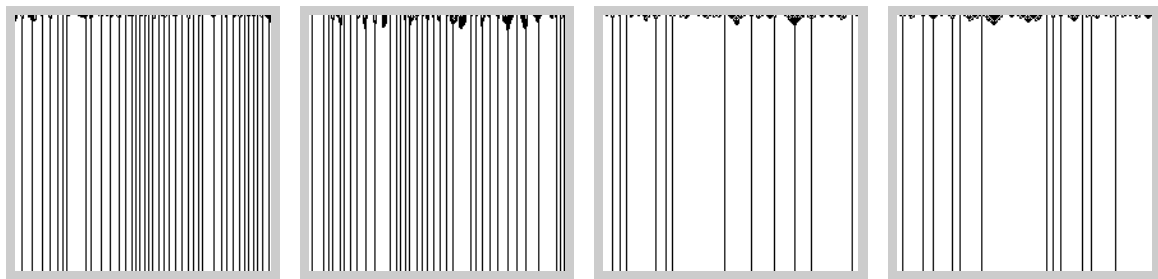
図 A.20: #140 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

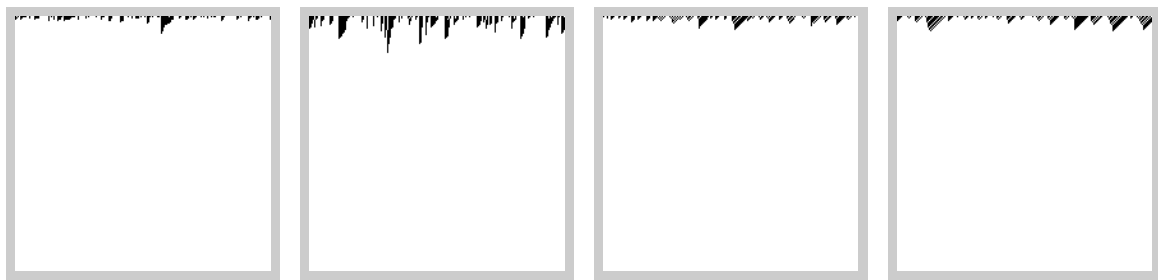
図 A.21: #160 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

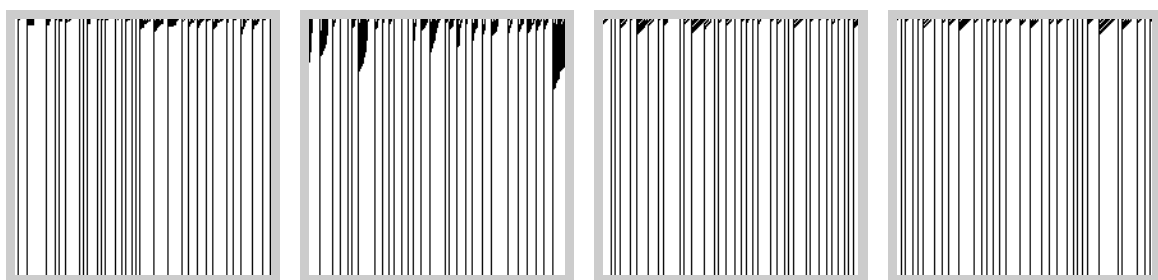
図 A.22: #164 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

図 A.23: #168 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

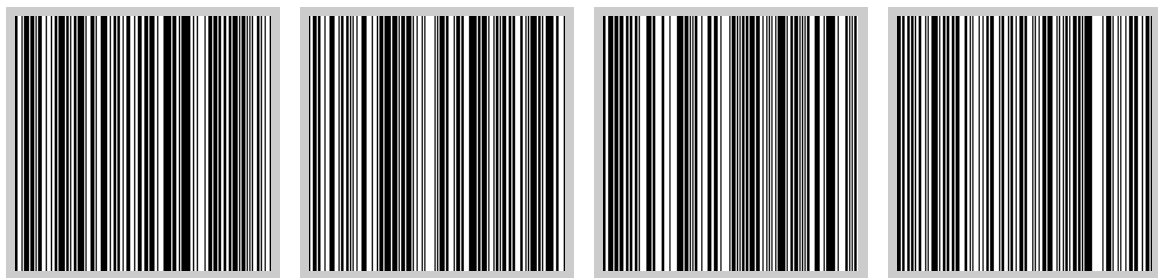
図 A.24: #172 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

図 A.25: #200 の時空間パターン (グループ 1)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

図 A.26: #204 の時空間パターン (グループ 1)



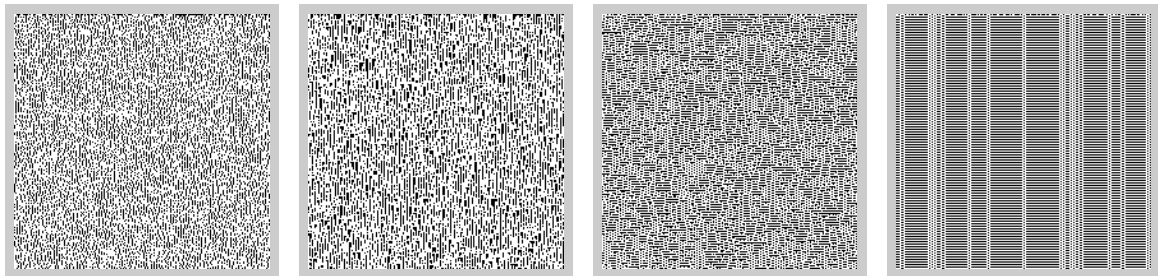
(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

図 A.27: #232 の時空間パターン (グループ 1)

A.2 グループ 2: 遷移確率 p の値によって徐々にパターンが変化する

グループ 2 は、遷移確率 p の値の変化によって徐々に時空間パターンが変化するものである。このグループに属している 37 のセルオートマトンの時空間パターンを図 A.28–A.64 に示す。



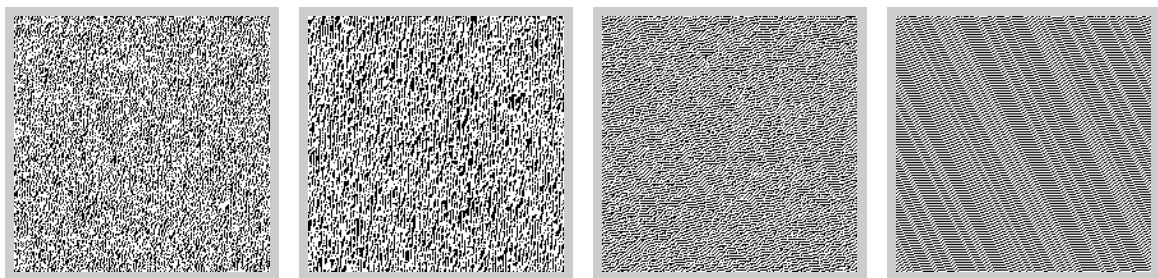
(a) 非同期 CA

(b) $p = 0.3$ (c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.28: #1 の時空間パターン (グループ 2)



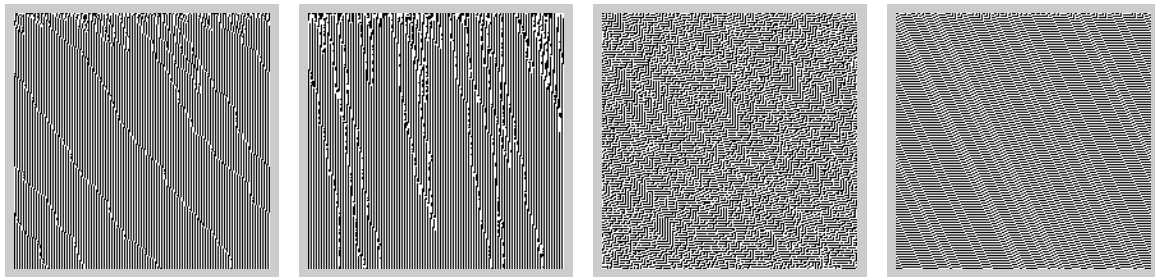
(a) 非同期 CA

(b) $p = 0.3$ (c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.29: #3 の時空間パターン (グループ 2)



(a) 非同期 CA

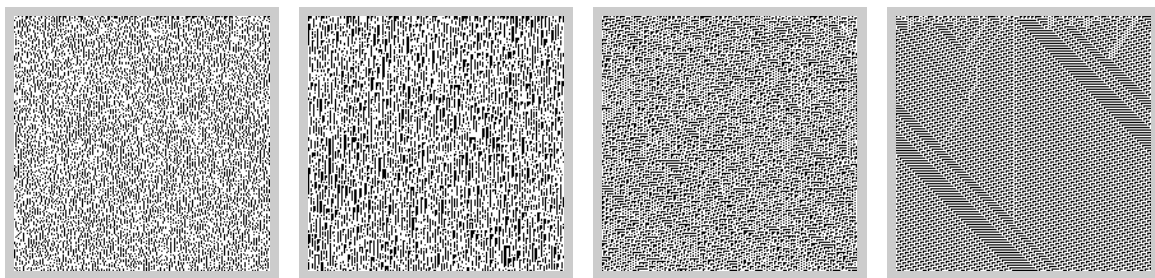
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.30: #7 の時空間パターン (グループ 2)



(a) 非同期 CA

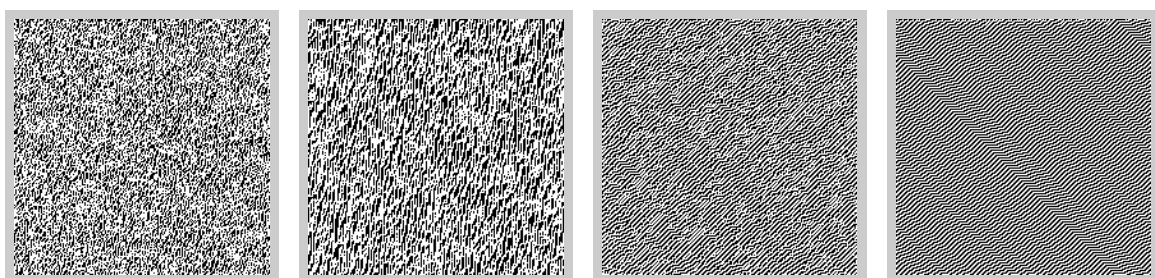
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.31: #9 の時空間パターン (グループ 2)



(a) 非同期 CA

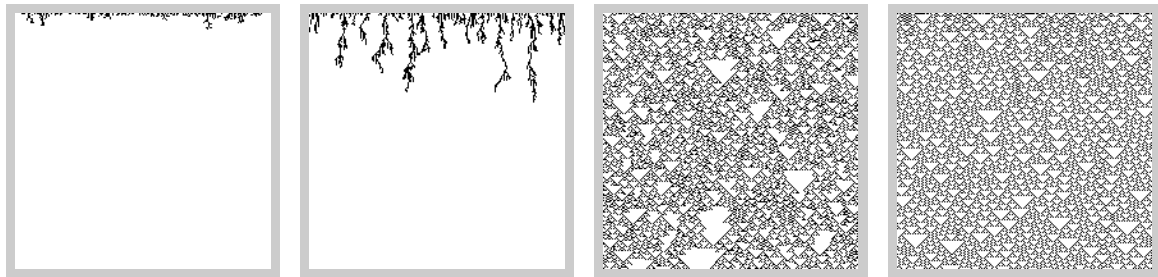
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.32: #11 の時空間パターン (グループ 2)



(a) 非同期 CA

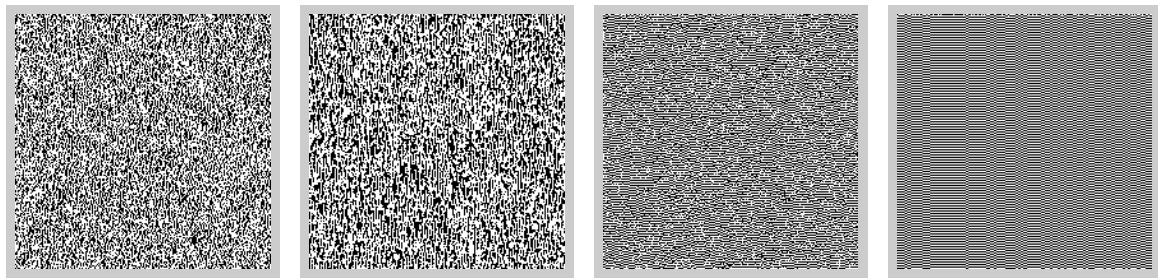
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.33: #18 の時空間パターン (グループ 2)



(a) 非同期 CA

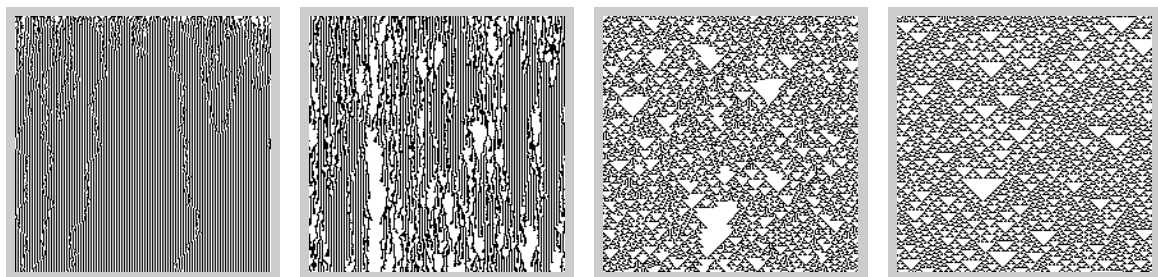
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.34: #19 の時空間パターン (グループ 2)



(a) 非同期 CA

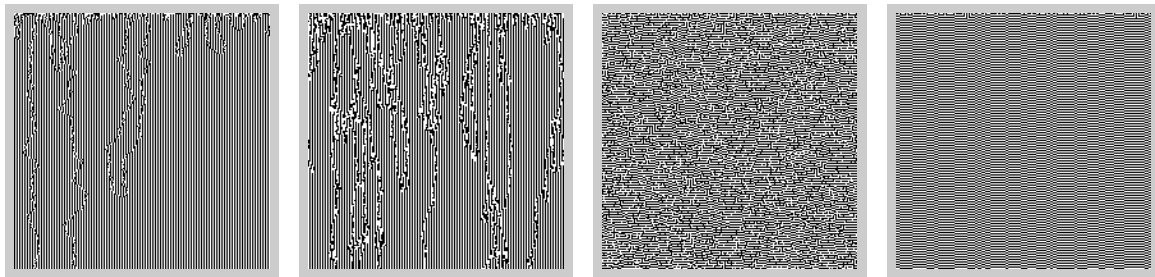
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.35: #22 の時空間パターン (グループ 2)



(a) 非同期 CA

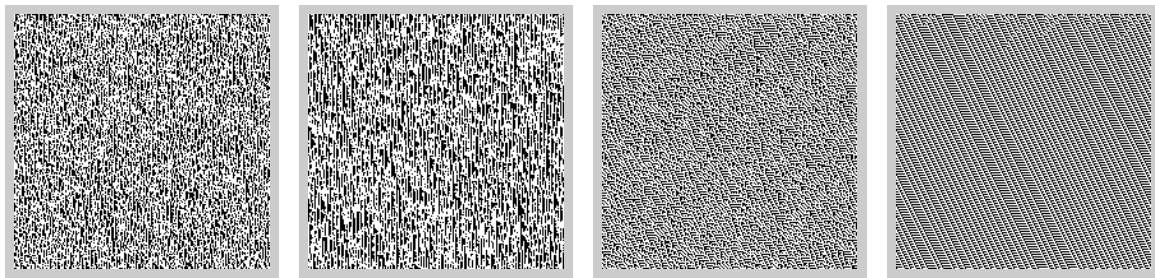
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.36: #23 の時空間パターン (グループ 2)



(a) 非同期 CA

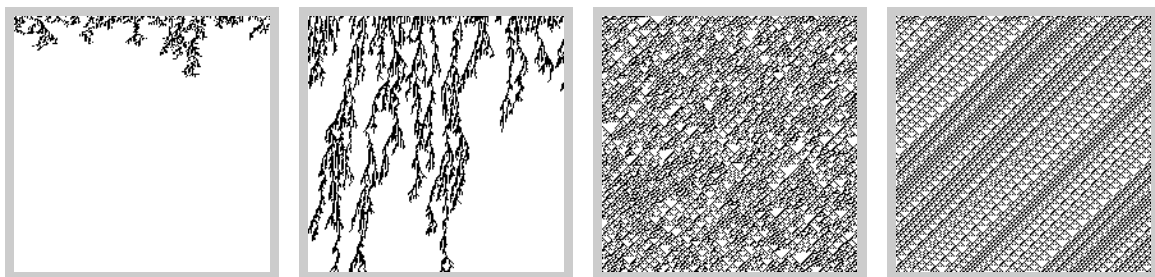
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.37: #25 の時空間パターン (グループ 2)



(a) 非同期 CA

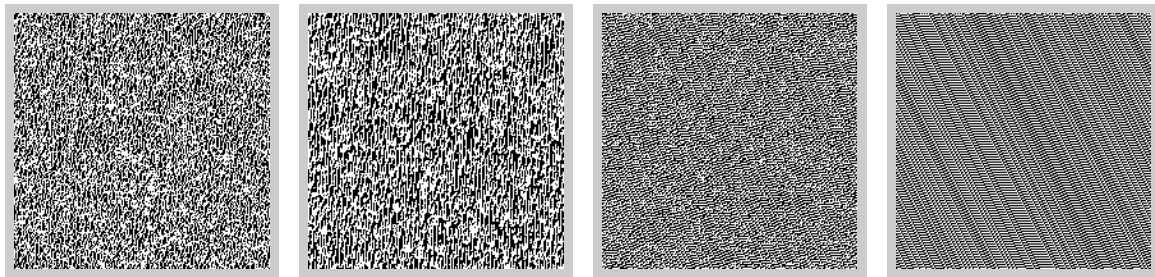
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.38: #26 の時空間パターン (グループ 2)



(a) 非同期 CA

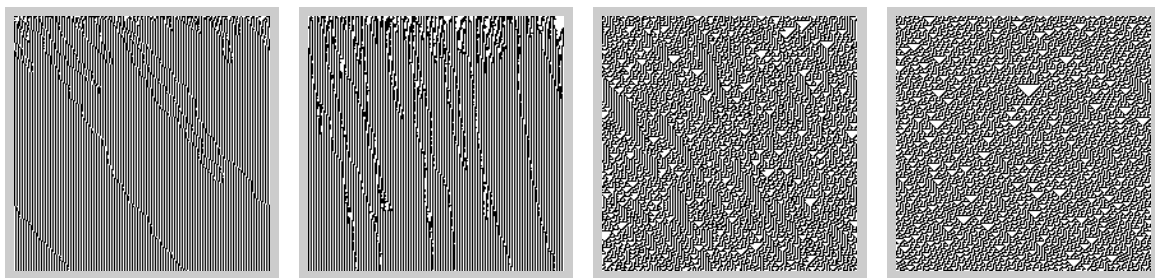
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.39: #27 の時空間パターン (グループ 2)



(a) 非同期 CA

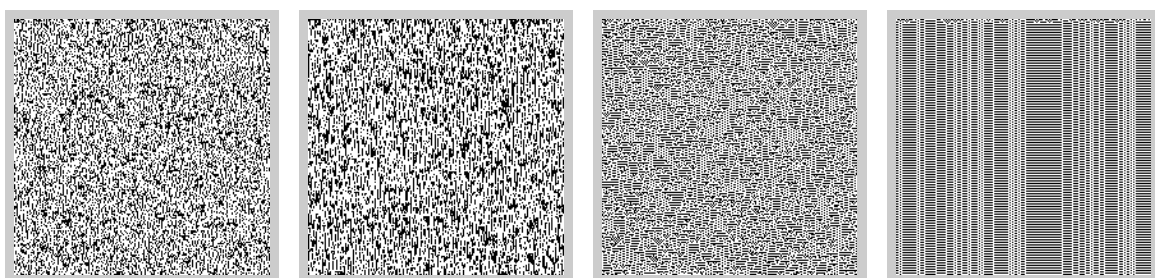
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.40: #30 の時空間パターン (グループ 2)



(a) 非同期 CA

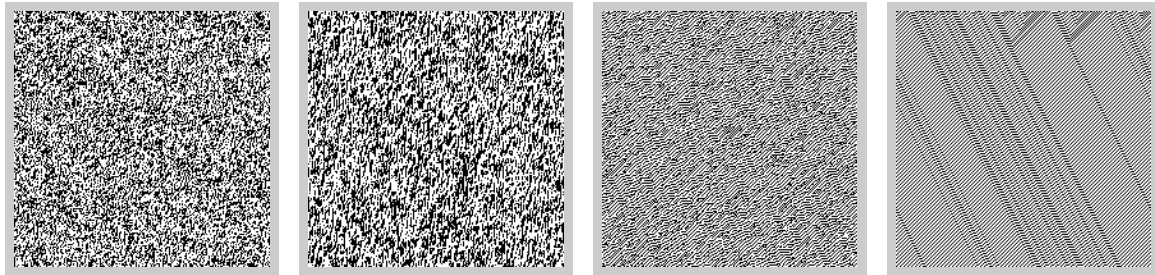
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.41: #33 の時空間パターン (グループ 2)



(a) 非同期 CA

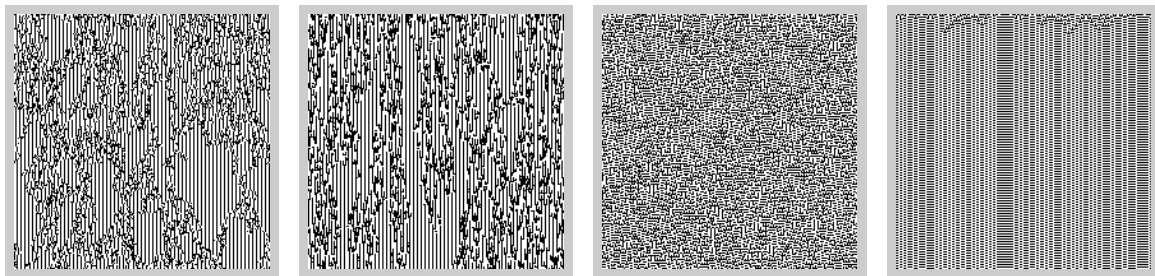
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.42: #35 の時空間パターン (グループ 2)



(a) 非同期 CA

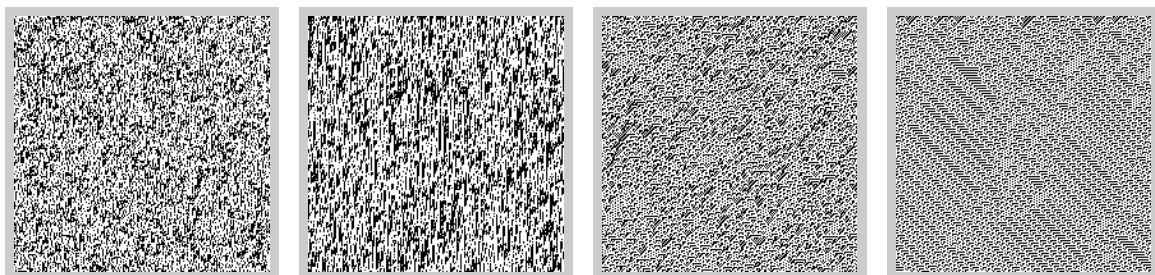
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.43: #37 の時空間パターン (グループ 2)



(a) 非同期 CA

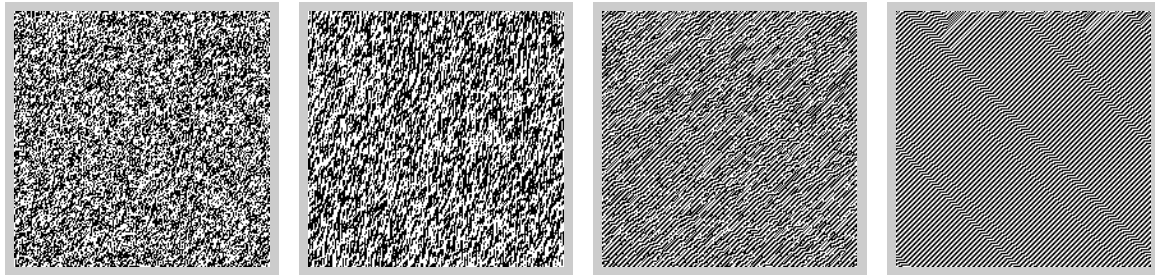
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.44: #41 の時空間パターン (グループ 2)



(a) 非同期 CA

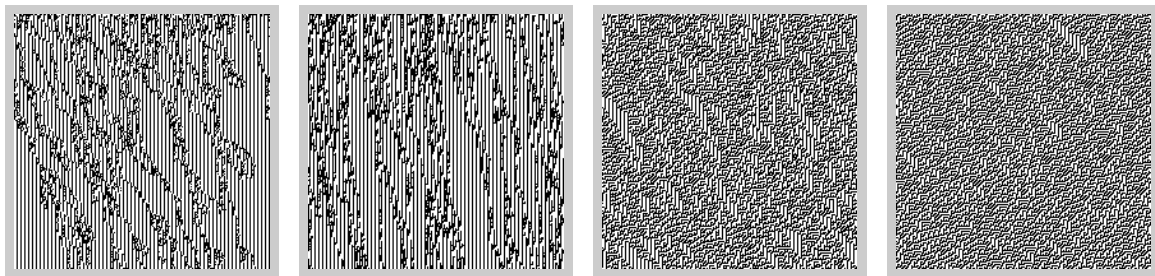
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.45: #43 の時空間パターン (グループ 2)



(a) 非同期 CA

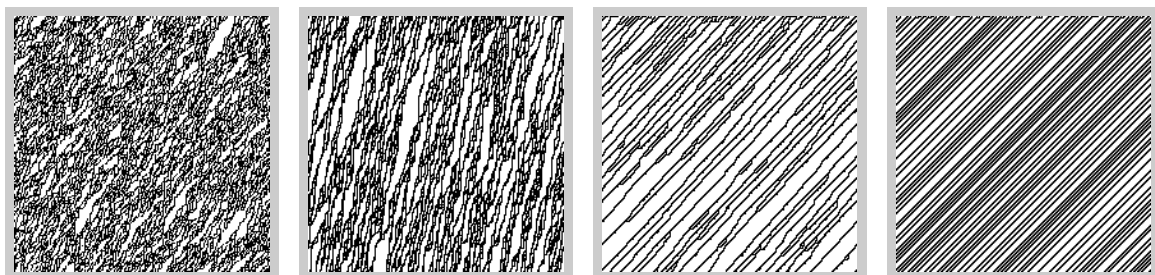
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.46: #45 の時空間パターン (グループ 2)



(a) 非同期 CA

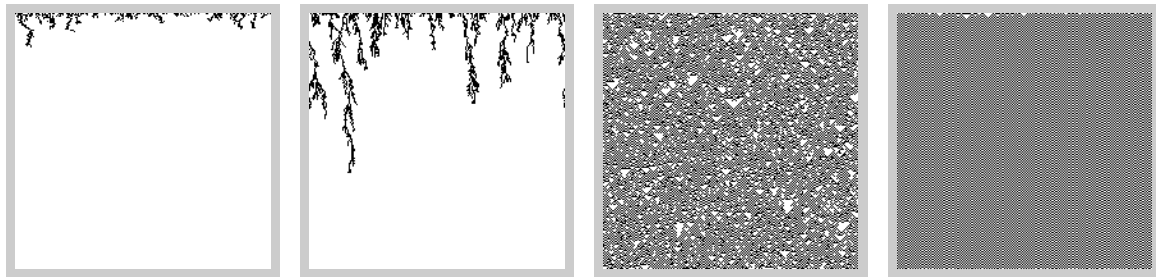
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.47: #46 の時空間パターン (グループ 2)



(a) 非同期 CA

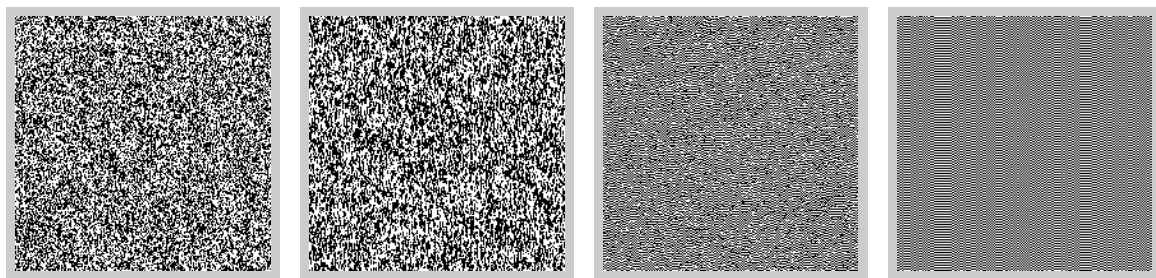
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.48: #50 の時空間パターン (グループ 2)



(a) 非同期 CA

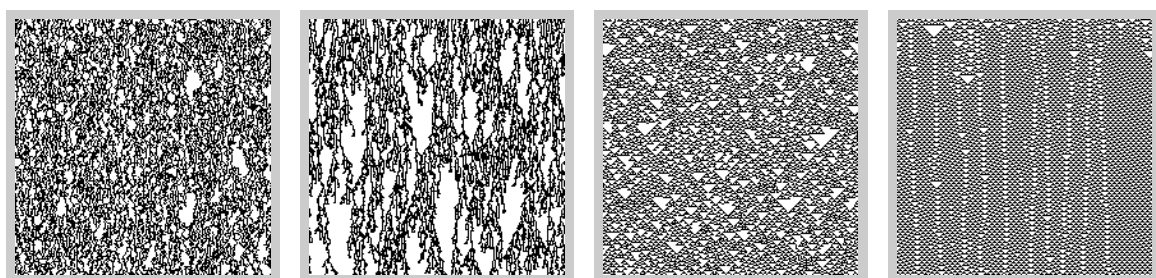
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.49: #51 の時空間パターン (グループ 2)



(a) 非同期 CA

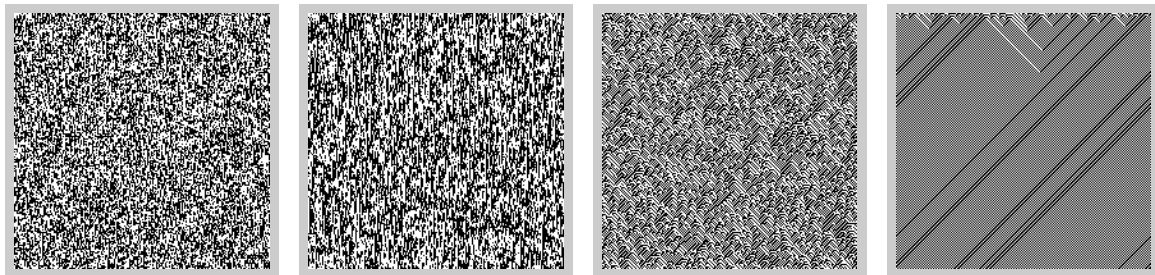
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.50: #54 の時空間パターン (グループ 2)



(a) 非同期 CA

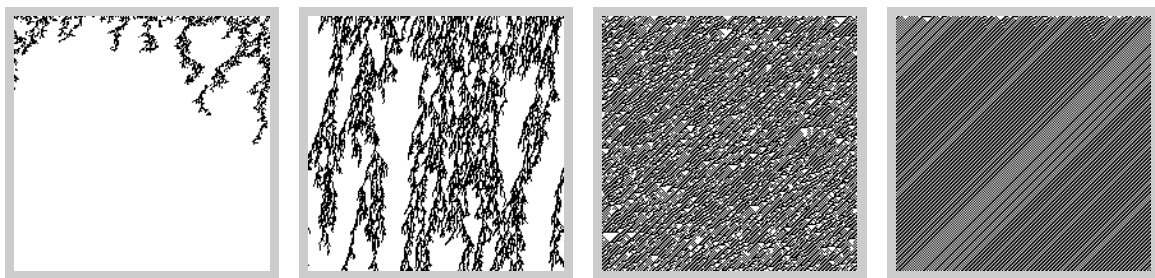
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.51: #57 の時空間パターン (グループ 2)



(a) 非同期 CA

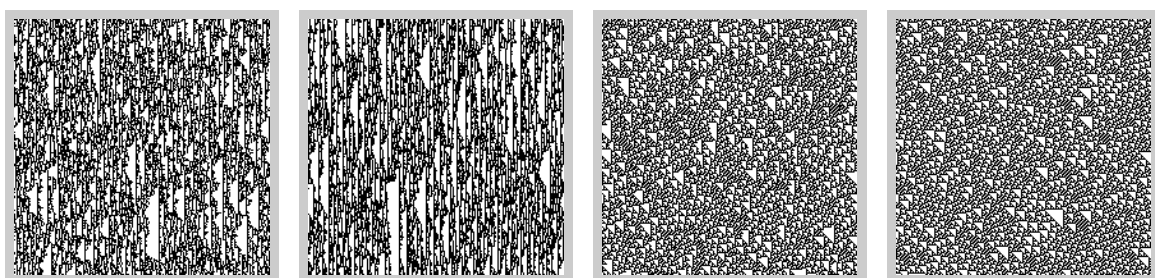
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.52: #58 の時空間パターン (グループ 2)



(a) 非同期 CA

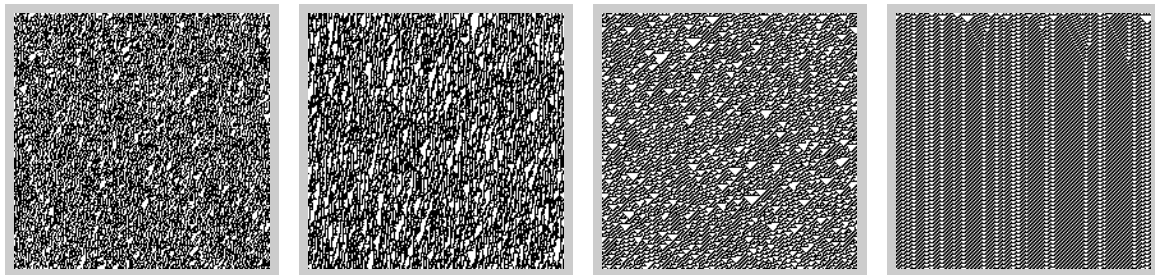
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.53: #60 の時空間パターン (グループ 2)



(a) 非同期 CA

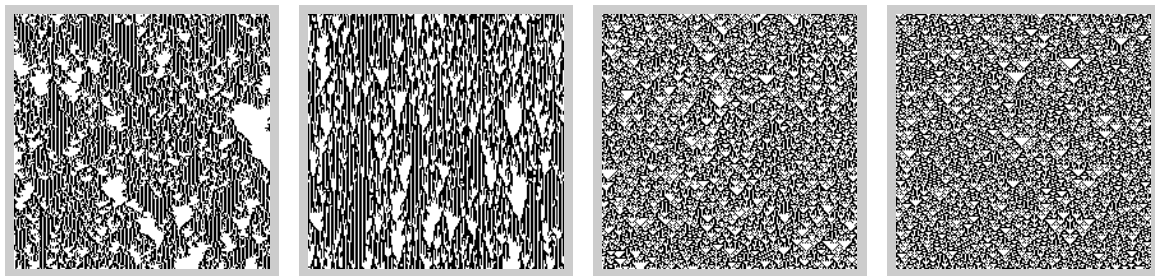
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.54: #62 の時空間パターン (グループ 2)



(a) 非同期 CA

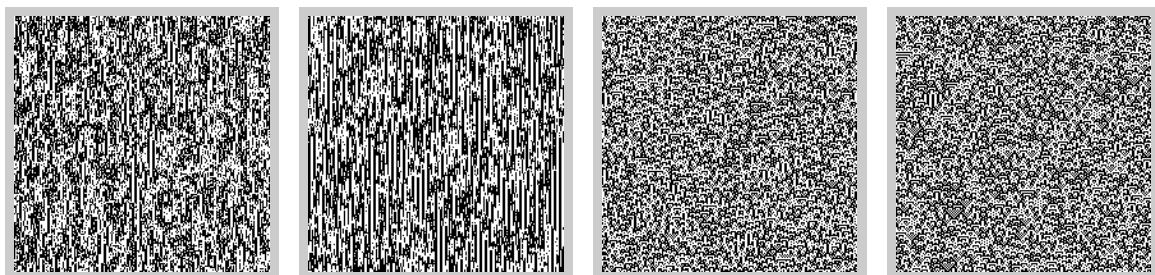
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.55: #90 の時空間パターン (グループ 2)



(a) 非同期 CA

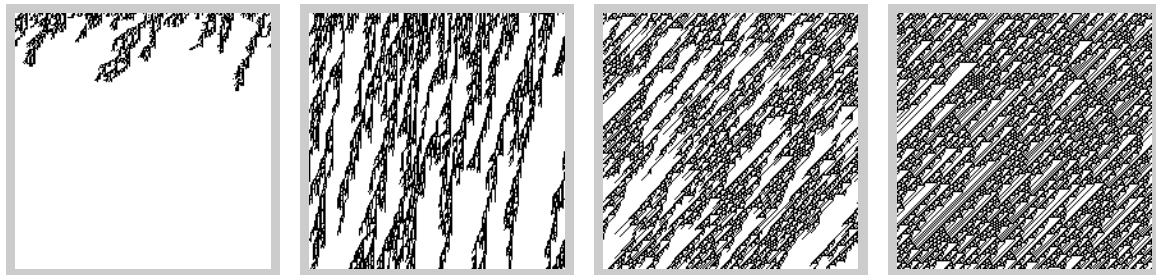
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.56: #105 の時空間パターン (グループ 2)



(a) 非同期 CA

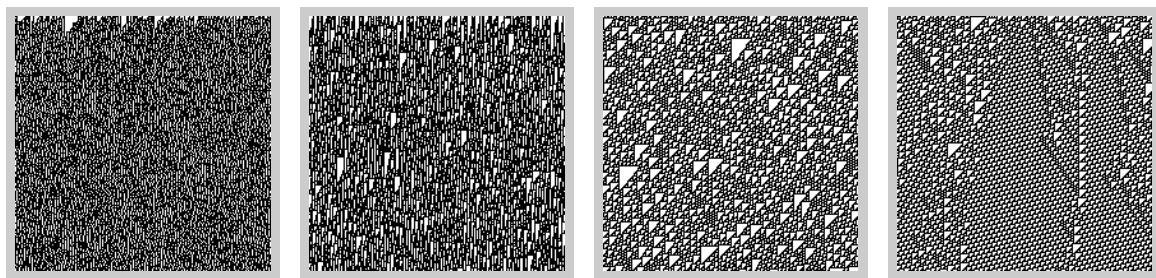
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.57: #106 の時空間パターン (グループ 2)



(a) 非同期 CA

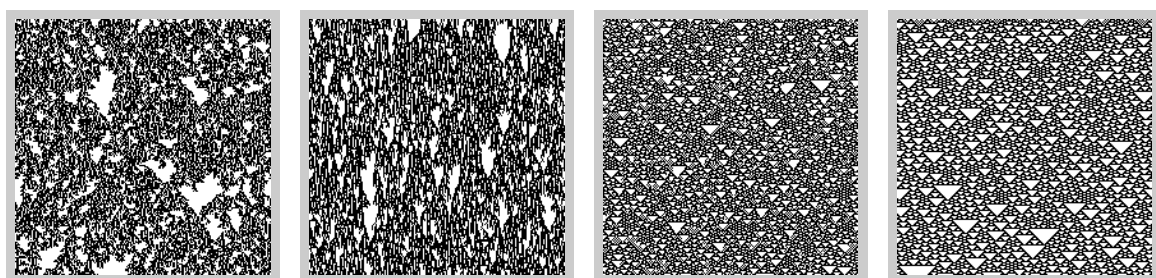
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.58: #110 の時空間パターン (グループ 2)



(a) 非同期 CA

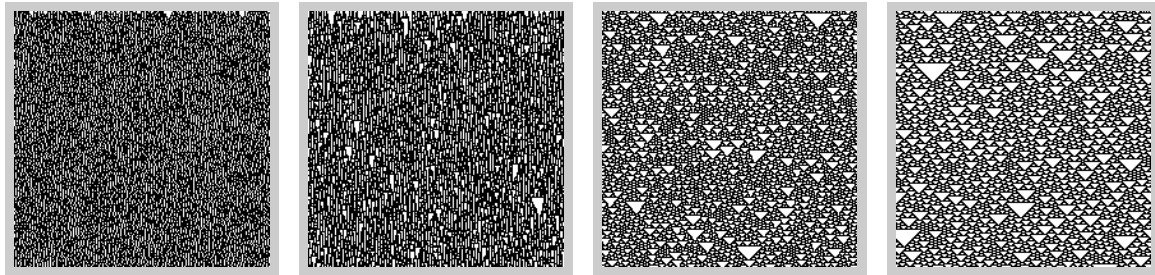
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.59: #122 の時空間パターン (グループ 2)



(a) 非同期 CA

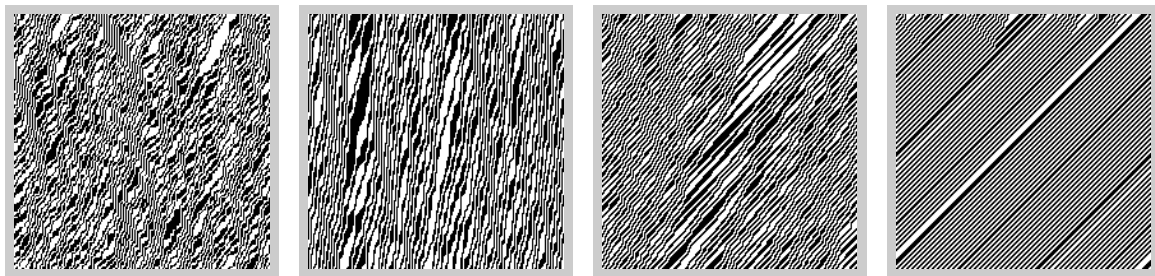
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.60: #126 の時空間パターン (グループ 2)



(a) 非同期 CA

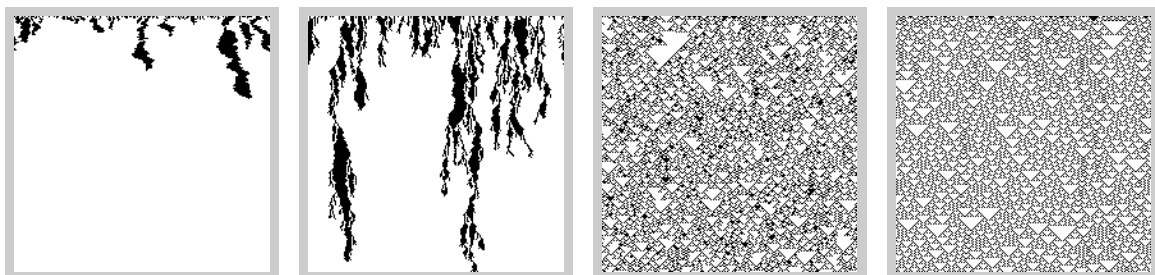
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.61: #142 の時空間パターン (グループ 2)



(a) 非同期 CA

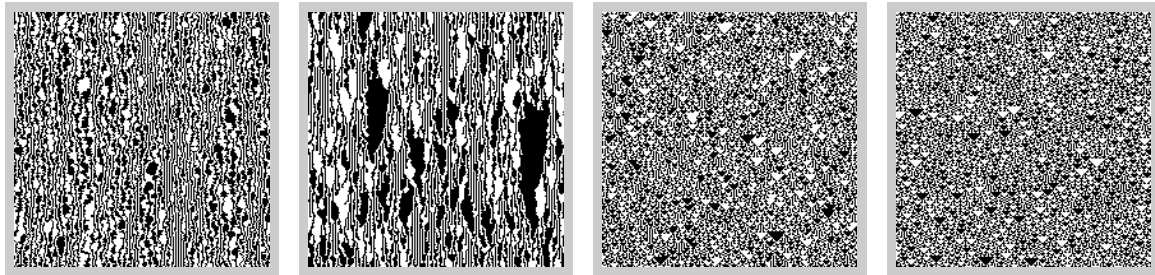
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.62: #146 の時空間パターン (グループ 2)



(a) 非同期 CA

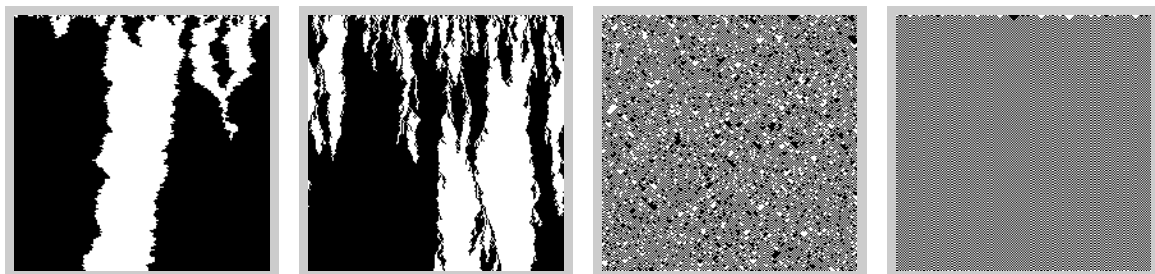
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.63: #150 の時空間パターン (グループ 2)



(a) 非同期 CA

(b) $p = 0.3$

(c) $p = 0.9$

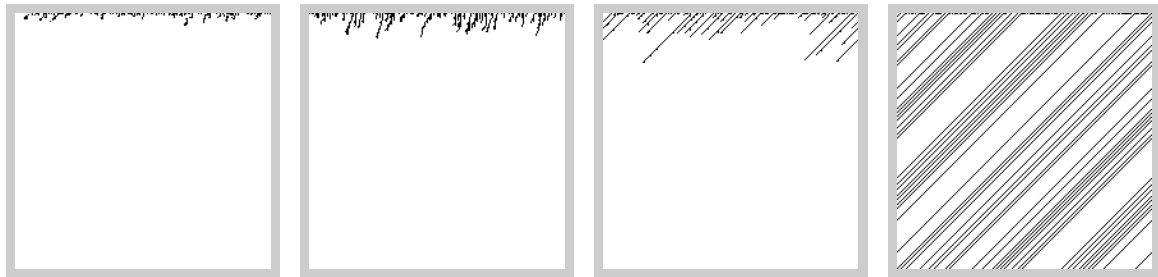
(d) 同期 CA

不完全な同期下の CA

図 A.64: #178 の時空間パターン (グループ 2)

A.3 グループ 3: 非同期 CA と同様なパターンに収束する

グループ 3 は、非同期セルオートマトンと同様な時空間パターンに収束するものである。このグループに属している 19 のセルオートマトンの時空間パターンを図 A.65–A.83 に示す。



(a) 非同期 CA

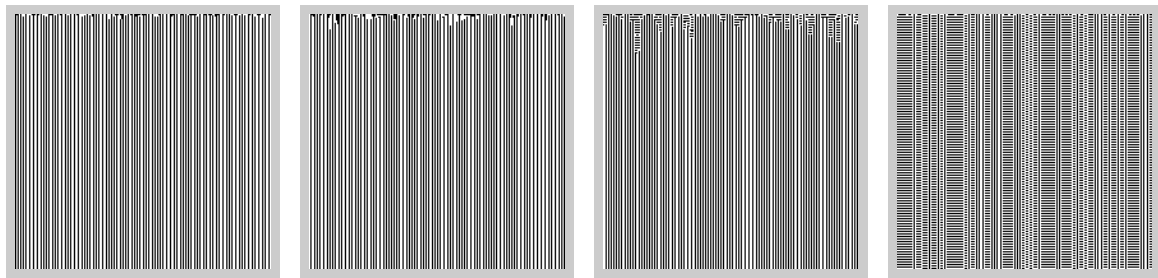
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.65: #2 の時空間パターン (グループ 3)



(a) 非同期 CA

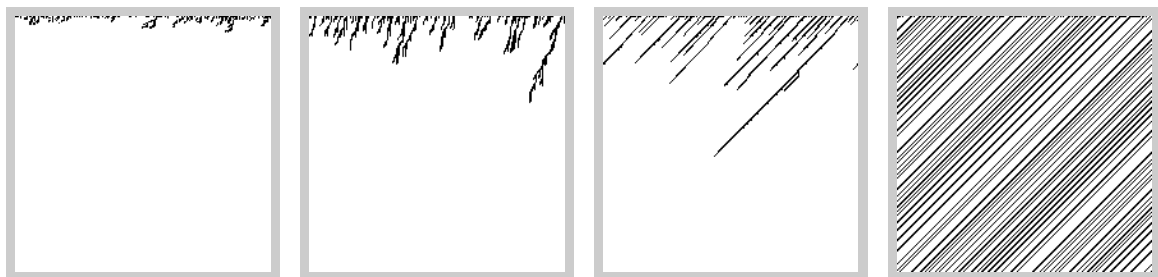
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.66: #5 の時空間パターン (グループ 3)



(a) 非同期 CA

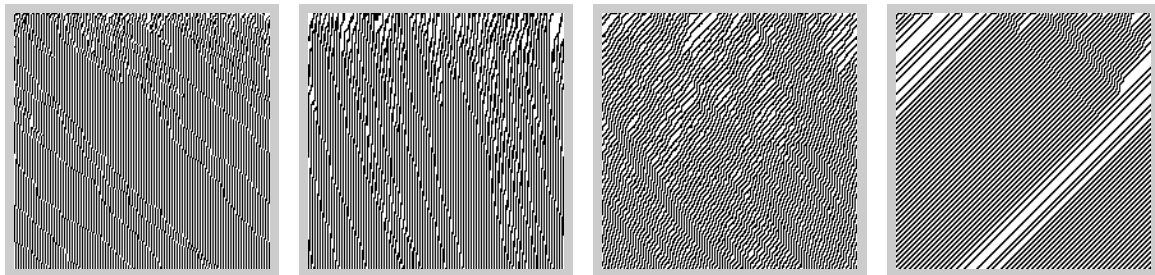
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.67: #10 の時空間パターン (グループ 3)



(a) 非同期 CA

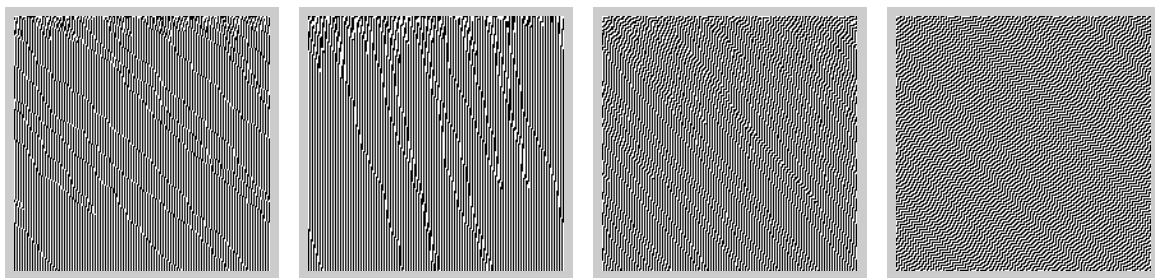
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.68: #14 の時空間パターン (グループ 3)



(a) 非同期 CA

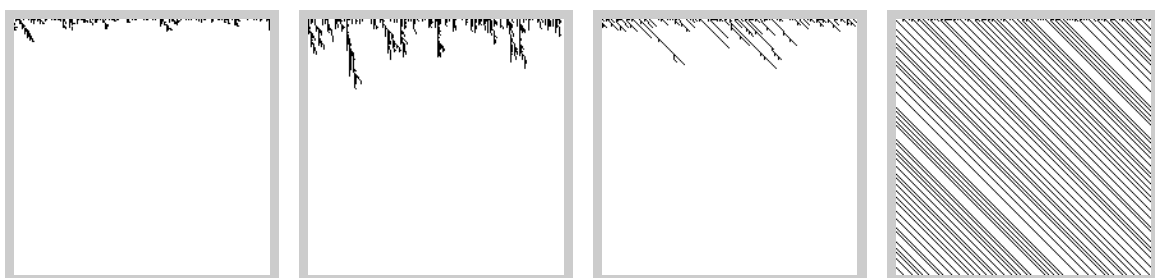
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.69: #15 の時空間パターン (グループ 3)



(a) 非同期 CA

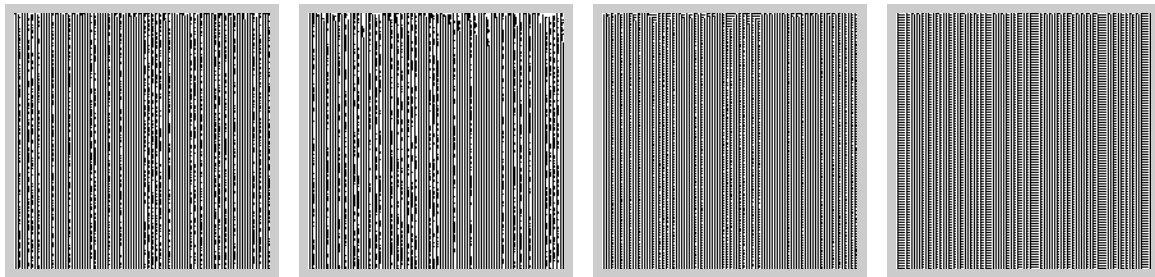
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

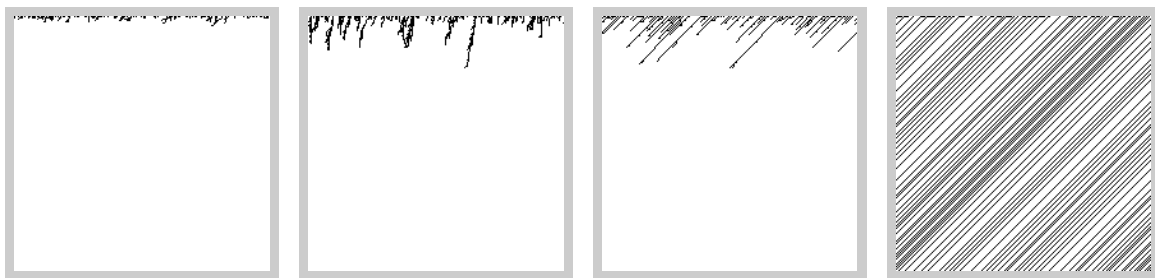
図 A.70: #24 の時空間パターン (グループ 3)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

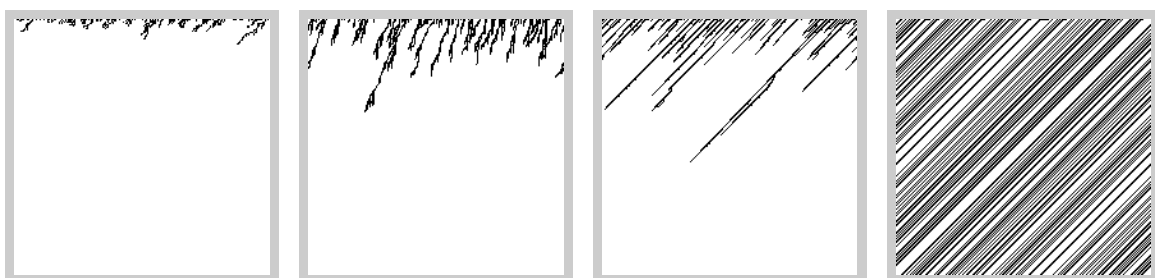
図 A.71: #29 の時空間パターン (グループ 3)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

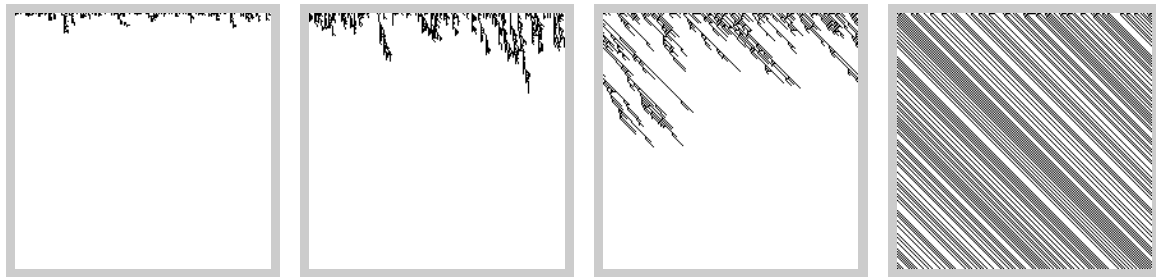
図 A.72: #34 の時空間パターン (グループ 3)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

図 A.73: #42 の時空間パターン (グループ 3)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

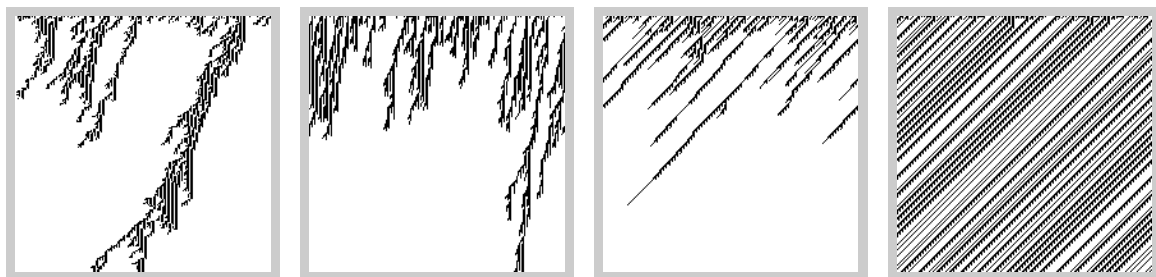
図 A.74: #56 の時空間パターン (グループ 3)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

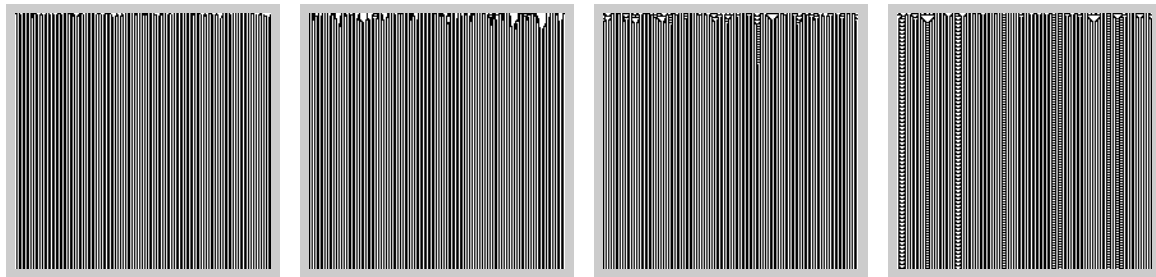
図 A.75: #73 の時空間パターン (グループ 3)



(a) 非同期 CA (b) $p = 0.3$ (c) $p = 0.9$ (d) 同期 CA

不完全な同期下の CA

図 A.76: #74 の時空間パターン (グループ 3)



(a) 非同期 CA

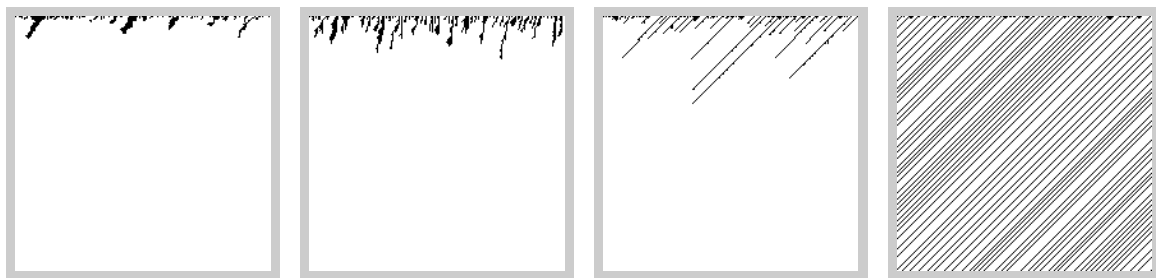
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.77: #94 の時空間パターン (グループ 3)



(a) 非同期 CA

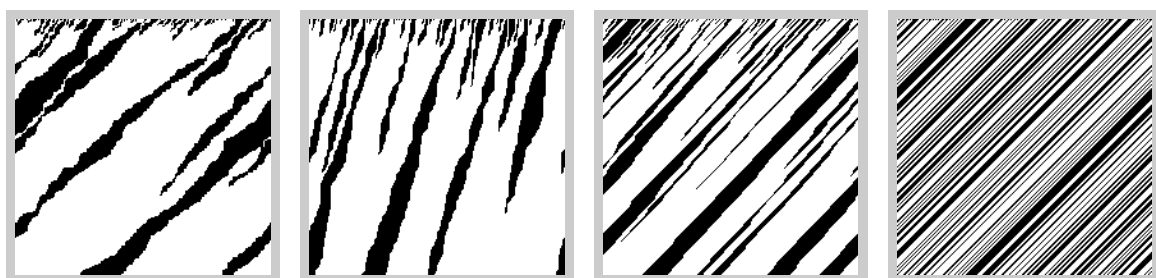
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.78: #130 の時空間パターン (グループ 3)



(a) 非同期 CA

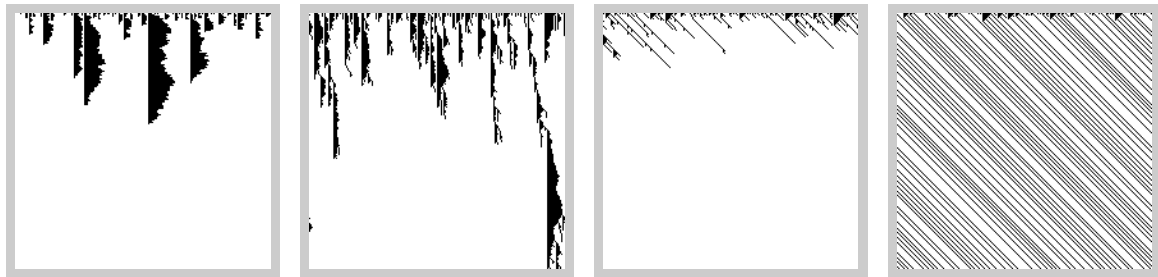
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.79: #138 の時空間パターン (グループ 3)



(a) 非同期 CA

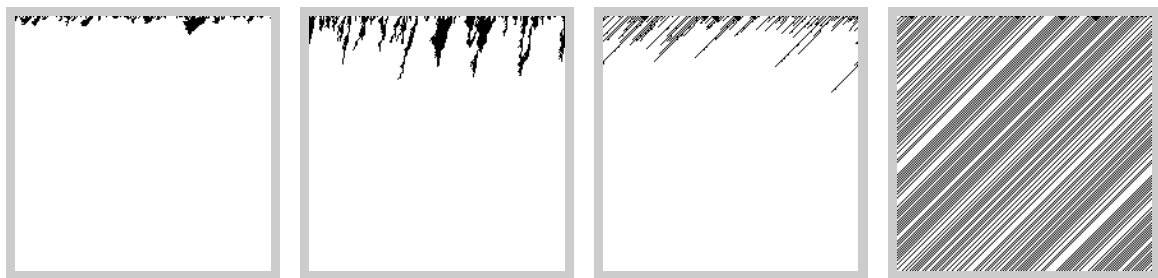
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.80: #152 の時空間パターン (グループ 3)



(a) 非同期 CA

(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.81: #162 の時空間パターン (グループ 3)



(a) 非同期 CA

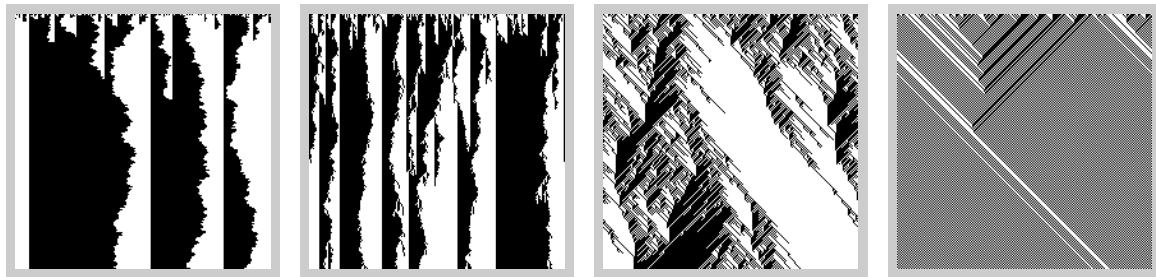
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.82: #170 の時空間パターン (グループ 3)



(a) 非同期 CA

(b) $p = 0.3$

(c) $p = 0.9$

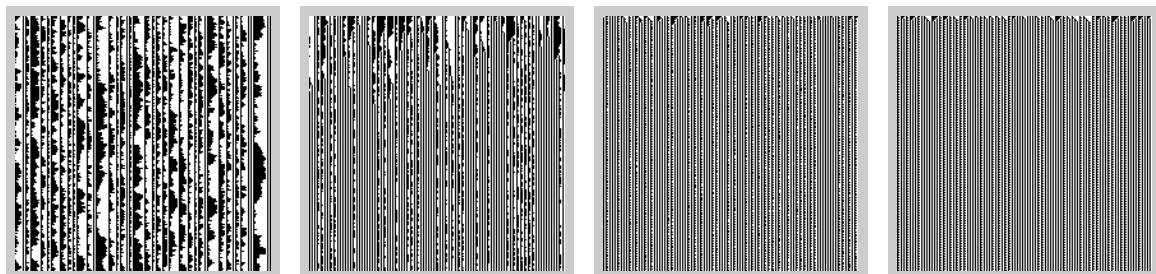
(d) 同期 CA

不完全な同期下の CA

図 A.83: #184 の時空間パターン (グループ 3)

A.4 グループ 4: 同期 CA と同様なパターンに収束する

グループ 4 は、同期セルオートマトンと同様な時空間パターンに収束するものである。このグループに属している #156 の時空間パターンを図 A.84 に示す。



(a) 非同期 CA

(b) $p = 0.3$

(c) $p = 0.9$

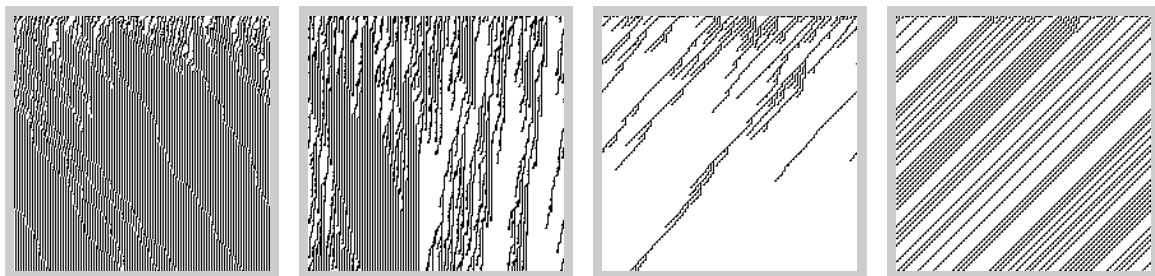
(d) 同期 CA

不完全な同期下の CA

図 A.84: #156 の時空間パターン (グループ 4)

A.5 グループ 5: 同期 CA, 非同期 CA とも異なるパターンに収束する

グループ 5 は, 同期セルオートマトン, 非同期セルオートマトンのどちらとも異なる時空間パターンに収束するものである. このグループに属している 4 のセルオートマトンの時空間パターンを図 A.85–A.88 に示す.



(a) 非同期 CA

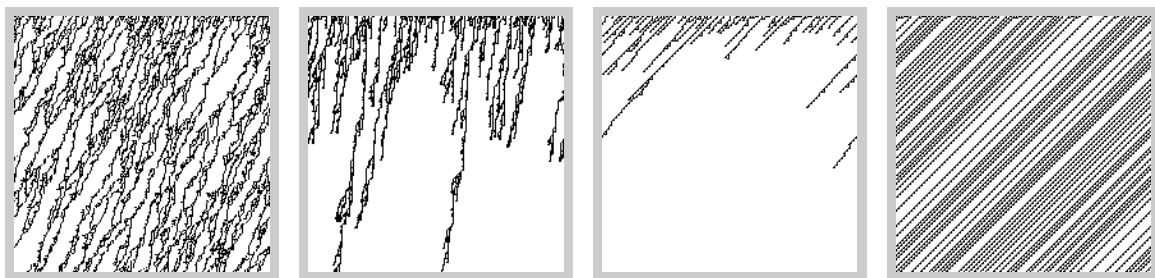
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.85: #6 の時空間パターン (グループ 5)



(a) 非同期 CA

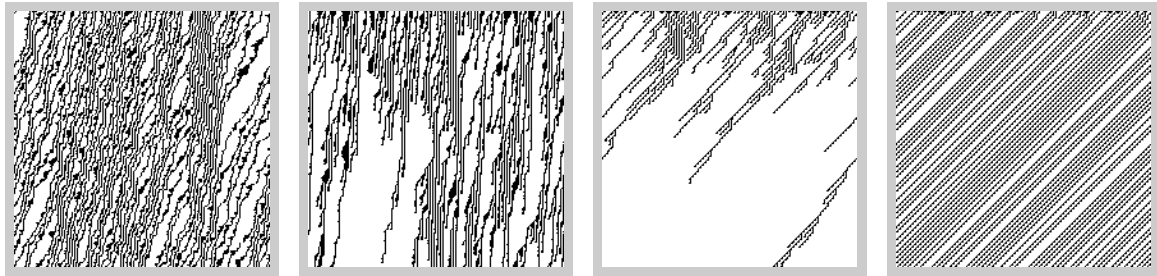
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.86: #38 の時空間パターン (グループ 5)



(a) 非同期 CA

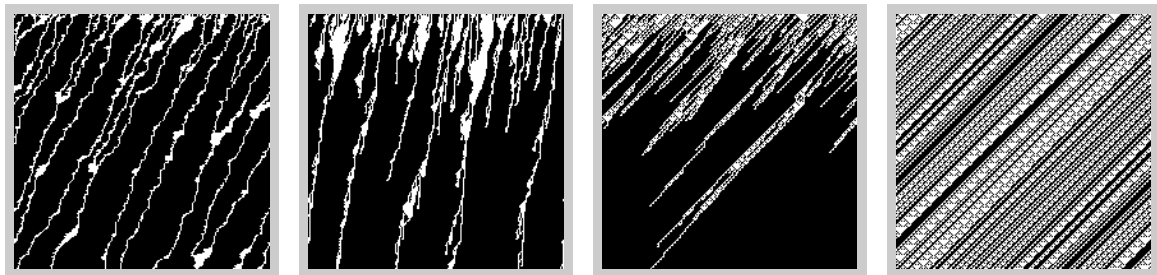
(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.87: #134 の時空間パターン (グループ 5)



(a) 非同期 CA

(b) $p = 0.3$

(c) $p = 0.9$

(d) 同期 CA

不完全な同期下の CA

図 A.88: #154 の時空間パターン (グループ 5)

謝辞

本論文は多くの方々のご指導や支えがあり完成することができました。ここに記して、深く感謝申し上げます。

富山大学理事・副学長の廣瀬貞樹先生には、卒業論文をはじめ、修士学位論文、博士学位論文まで多大なるご指導を賜りました。また、在学中から就職後の現在に至るまで、公私に渡り多大なるご助言を賜りました。深く感謝し厚く御礼申し上げます。

同大学教授の長谷川英之先生には、お忙しいなか査読委員長を務めていただきご助言を賜りました。心より感謝申し上げます。

同大学教授の中茂樹先生、同大学教授の唐政先生には、お忙しいなか査読委員を務めていただきご助言を賜りました。心より感謝申し上げます。

同大学准教授の参沢匡将先生には、お忙しいなか世話教員を努めていただきご助言を賜りました。心より感謝申し上げます。

岩手大学名誉教授の西谷泰昭先生には、一斉射撃問題の拡張およびアルゴリズムについて多大なご指導を賜りました。心より感謝申し上げます。

会津大学名誉教授の大川知先生と大阪工業大学元教授の大里延康先生には、研究を進めるにあたりさまざまなお助言を賜りました。心より感謝申し上げます。

最後に、これまで私をあたたく見守ってくれた両親と亡き祖父母、私を明るく励まし続けてくれた妻の香織、息子の新太、娘の琴子に心から感謝いたします。