

2019年度 卒業論文

落ちものパズルゲームにおける文字列による
任意形状指定に関する研究

指導教員：渡辺 大地 准教授

メディア学部 ゲームサイエンス
学籍番号 M0116281
宮尾 悠平

2020年2月

2019年度 卒業論文概要

論文題目

落ちものパズルゲームにおける文字列による
任意形状指定に関する研究

メディア学部

学籍番号：M0116281

氏名

宮尾 悠平

指導
教員

渡辺 大地 准教授

キーワード

落ちものパズル、形状検索、文字列表現、GDL、正規表現

落ちものパズルゲームとは、画面上部から画面下部に向かって降下していくブロックをプレイヤーが操作し、ゲームごとに設定された条件を満たすようにブロックを積み重ねていき、ブロックを消去または変化させ得点を獲得していくアクションパズルゲームのことである。落ちものパズルゲームを構成するために必須なルールの一つに、ブロックの消去条件がある。ブロックの消去条件は非常に複雑な形状であったり、柔軟な形状が指定されることが多く、これらの形状は、条件に沿ったアルゴリズムを考え、実際にプログラミングすることで表現されることが一般的であり、ある決まった形式で表現することは困難である。落ちものパズルゲームの GDL において、消去条件のルールを規定する際、消去条件の指定が、ある決まった形式で表現できることが望まれており、消去条件の指定は、様々な形状や法則性を表現できる必要がある。

本研究では、法則性を文字列で記述できるという正規表現を参考に、落ちものパズルゲームにおいて、文字列を用いて消去形状の法則性を表現することが出来る手法の提案を目的とした。本手法では、法則性や機能を持たせたいくつかの文字列と数字によりブロックを表現し、消去形状を指定する。本提案手法を用い、既存の落ちものパズルゲームの消去形状の再現と、いくつかの形状について検証を行い、文字列を用いて消去形状の法則性を表現することが出来るかの検証を行った。その結果、「テトリス」「ルミネス」「ぶよぶよ」など、既存のゲームの消去形状の再現を行うことが出来た。また、いくつかの形状に対しても消去形状を指定することが出来た。

目次

第 1 章	はじめに	1
1.1	背景と目的	1
1.2	論文の構成	3
第 2 章	提案手法	4
2.1	提案手法の概要	4
2.2	提案手法の表現	5
第 3 章	検証と考察	10
3.1	既存のゲームの消去形状の再現	10
3.1.1	テトリスの消去形状の再現	10
3.1.2	ルミネスの消去形状の再現	12
3.1.3	ぷよぷよの消去条件の再現	13
3.2	いくつかの形状の探索	15
3.2.1	縦または横にブロックが並んでいる場合	15
3.2.2	T 字にブロックが並んでいる場合	17
3.2.3	十字型にブロックが並んでいる場合	18
3.2.4	ブロックが複雑な形状を形成している場合	21
3.3	考察	22
第 4 章	まとめ	23
	謝辞	24
	参考文献	25

目次

2.1	フィールド上に 2×3 のブロックが積まれている例	5
2.2	文字列”3R”で検索するフィールドの例	6
2.3	文字列を”3R”にしたときの探索結果	7
2.4	文字列を”3D”で検索するフィールドの例	7
2.5	文字列を”3D”にしたときの探索結果の例	8
2.6	L字を探索するフィールドの例	9
2.7	L字を探索したときの探索結果の例	9
3.1	文字列”8R”で探索するフィールド	11
3.2	文字列”8R”で探索した結果	11
3.3	ルミネスの再現で探索するフィールド	12
3.4	ルミネスの再現で探索した結果	13
3.5	3×3 での挙動の例	13
3.6	文字列”4C”で探索するフィールド	14
3.7	文字列”4C”で探索した結果	14
3.8	検証する形状の例	15
3.9	縦または横方向のブロックを探索するフィールド	16
3.10	縦または横方向のブロックを探索した結果	16
3.11	T字を形成するブロックを探索するフィールド	17
3.12	T字を形成するブロックを探索した結果	18
3.13	十字を形成するブロックを探索するフィールド	19
3.14	十字を形成するブロックを探索した結果	19
3.15	柔軟性を持つ十字を形成するブロックを探索するフィールド	20
3.16	柔軟性を持つ十字を形成するブロックを探索した結果	20
3.17	複雑な形状を探索するフィールドの例	21
3.18	複雑な形状を探索した結果のフィールドの例	21

第 1 章

はじめに

1.1 背景と目的

落ちものパズルゲームとは、ブロックを積み重ねていき、ブロックを消去または変化することにより得点を獲得していくアクションパズルゲームである。プレイヤーは、画面上部から画面下部に向かって降下していくブロックを操作し、ブロックを積み重ねていく。ブロックを動かせる範囲をフィールドと呼び、フィールドを四角形で分割し、その分割した四角形一つ一つをマスと呼ぶ。「テトリス」[1]を始め、「ぷよぷよ」[2]や「ルミネス」[3]など、多くの落ちものパズルゲームが存在する。落ちものパズルゲームは、ブロックを消去するための条件がゲームによって異なっている。テトリスでは、ブロックをフィールドの横 1 行のマス全てに配置することでブロックを消去する。ぷよぷよでは、ブロックの種類が赤色、青色、緑色など色で複数種類に分かれており、同色のブロックを 4 マス以上隣接させる事で、ブロックを消去する。ルミネスでは、同色のブロックで 2×2 の四角形を作り、その四角形がフィールドを横方向に流れているタイムラインと呼ばれる線に触れることでブロックを消去する。

近年、ゲーム内のコンテンツを自動生成する技術が利用されている。不思議のダンジョンシリーズ [4] や、ファンタースターオンライン 2 [5] では、ダンジョンの自動生成が行われている。

ティル・ナ・ノグ [6] というゲームでは、物語の自動生成が行われている。さらに、数独の様なパズルゲームの自動生成 [7] や、音楽ゲームにおいて、音楽に合わせたプレイヤーへの指示が記された譜面というものを自動生成する研究 [8] も行われている。また、ゲーム内のコンテンツではなく、ゲームのルール自体を自動生成する研究も行われている。その際、Game Description Language(GDL)[9] が用いられることがある。GDL とは、ゲームルールを体系化し、生成するためのゲームルール記述言語である。GDL は、General Game Playing[10] と呼ばれる、様々な未知のゲームをうまくプレイできる AI の実現を目指した、ゲーム AI 研究の分野の中で生み出された。Tom[11] は、単純なアクションゲームを GDL で再現した。Cameron ら [12] は GDL を用い、ボードゲームのルールを組み合わせ、新たなゲームルールを生成した。また、落ちものパズルゲームのルールの自動生成に関する研究もおこなわれている。栗原ら [13][14] は、落ちものパズルゲームのルールを体系化し、GDL として記述できるようにした。そして、既存の落ちものパズルゲームのルールの再現や、複数の落ちものパズルゲームのルールを混合した新たなゲームの生成に成功している。

本論文では、ブロックの消去が起こる時の、ある決まった形に積み重ねられたブロックの形状の事を消去形状と呼称する。落ちものパズルゲームを構成するために必須なルールの一つに、ブロックの消去条件がある。そのため、落ちものパズルゲームでは、ブロックの消去条件を指定する必要がある。ブロックの消去条件は、ゲームルールによって消去条件が異なるため、多種多様な形状が考えられる。また、例えば、ブロックが十字型を形成しているならば、ある程度形状が異なっても消滅させる、というような柔軟な形状が指定されることが多い。これらの形状は、条件に沿ったアルゴリズムを考え、実際にソースコードとして記述することで表現されることが一般的であり、ある決まった形式で表現するのは困難である。GDL において、アルゴリズムをソースコードとして実際に記述した状態では、ゲームルールの解析などに利用することが困難となっている。そのため、落ちものパズルゲームの GDL において、消去条件のルールを規定する

際、消去条件の指定が、ある決まった形式で表現できることが望まれる。また、消去条件の指定は、様々な形状や法則性を表現できる必要がある。

そこで本研究では、落ちものパズルゲームにおいて、文字列を用いて消去形状の法則性を表現することが出来る手法の提案を目的とする。提案手法では、正規表現という表現方法に着目した。正規表現とは、ある特定のパターンを持つ文字列を指定する表現方法の一つで、法則性を文字列として表すことが出来る。また、正規表現は、文字列の検索や置換、抽出などによく利用される。正規表現などで表される文字列の集合に合致する文字列を探すことをパターンマッチングという。渡部ら [15][16] は、数式の二次元的な位置構造を、文字列の入れ子構造になっているとみなし、正規表現を用いて、文字が二次元的に配置される数式のパターンマッチングを行う手法の提案を行った。また、全てのパターンにマッチする文字列のことをワイルドカードといい、数式検索にワイルドカードの機能を導入する事が試みられている [17][18]。

本研究の手法では、法則性を文字列で記述できるという正規表現を参考に、文字列を用いて消去形状を指定する。そして、提案手法を用い、既存の落ちものパズルゲームの消去形状の再現と、いくつかの形状について検証を行い、文字列を用いて消去形状の法則性を表現することが出来るかの検証を行った。検証した結果、テトリスやルミネス、ぷよぷよなど既存のゲームの消去形状の再現を行うことが出来た。また、いくつかの形状に対しても消去形状を指定することが出来、消去形状の指定に、ある程度の柔軟性を持たせることもできた。

1.2 論文の構成

本論文の構成は、2章では提案手法について述べる。3章では検証とその結果、考察について述べる。4章では本研究のまとめについて述べる。

第 2 章

提案手法

本章では、本研究で提案する法則性や柔軟性を持つ消去形状を、文字列で指定する手法について述べる。

2.1 提案手法の概要

本手法では、フィールドとして 2 次元配列を用意した。その 2 次元配列の中では、マスの状態を示す。ブロックは最大 4 種類を想定しており、それぞれ「1」「2」「3」「4」の数字を割り当てる。同じ数字のブロック同士で、上下または左右に隣り合っている消去形状をとった場合のみ、ブロックが消えることを想定している。そのため、斜め方向にブロックが並んでいる消去形状は想定しない。また、消去形状を指定する際に別の数字同士で消去形状を指定することは想定していない。ブロックがないマスは「0」とする。さらに、指定した消去形状と一致したフィールド上のブロックを消去待機状態のブロックとし、「5」とする。例えば、「1」がフィールド上に 2×3 で配置されていたら、フィールド上に 2×3 のブロックが積まれていることを表す。それを表したのが図 2.1 である。

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	1	1	1	0	0
0	1	1	1	0	0

図 2.1 フィールド上に 2×3 のブロックが積まれている例

2.2 提案手法の表現

本手法では、次に挙げる法則性や機能を持たせた文字列と数字によりブロックを表現し、消去形状を指定する。

- R : あるブロックを基準に右方向に同じ種類のブロックが隣り合っている形状を指定
- L : あるブロックを基準に左方向に同じ種類のブロックが隣り合っている形状を指定
- U : あるブロックを基準に上方向に同じ種類のブロックが隣り合っている形状を指定
- D : あるブロックを基準に下方向に同じ種類のブロックが隣り合っている形状を指定
- P : 基準となるブロックを指定
- [] : P と共に使用
- + : R,L,U,D と共に使用し、設定されているブロックの個数以上隣り合っている形状も同時に指定
- C : 上下方向または左右方向に、指定する個数以上のブロックが隣り合っている形状を指定
- | : いくつかの形状を同時に指定したい場合に使用

消去形状を文字列で指定し、指定した消去形状と一致するブロックを、フィールド上から探し出す。また、文字列によって指定した形状をフィールド上から探し出す際、一番左上のマスから右方向に順番にマスを探査する。そして、フィールドの右端のマスまで到達したら、下の行の左端のマスに移りまた右方向に向かってマスを探査する、という事をフィールドのすべてのマスに対して行う。その際に、ブロックを発見し次第、そのブロックを基準に文字列で指定した法則性を持つ形状を探し出す。そして、すべてのマスを探査後、消去待機状態のブロックに対して消去する処理を行う。

例として”3R”と記述すると、右方向に3個ブロックがつながっている、つまり同じ数字が右方向に3つ並んでいる部分をフィールド上から探し出す。文字列で表した消去形状に該当した部分は、フィールド上では消去待機状態である「5」に変わる。以下の図 2.2 はいくつかのブロックを配置したフィールドを図で示したものであり、図 2.3 は”3R”で探索した結果を図で示したものである。

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
1	1	0	0	0	0
0	0	0	1	1	1
1	1	1	1	0	0

図 2.2 文字列”3R”で検索するフィールドの例

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
1	1	0	0	0	0
0	0	0	5	5	5
5	5	5	5	0	0

図 2.3 文字列を”3R”にしたときの探索結果

”3D”と記述すると、下方方向に 3 個ブロックがつながっている、つまり同じ数字が下方方向に 3 個並んでいる部分をフィールド上から探し出す。フィールド上にいくつかのブロックを配置したものが図 2.4 であり、そのフィールドの中から”3D”で探索した結果が図 2.5 である。「1」が縦方向に 4 個並んでいる部分も「5」に変わっているが、これは 4 個の 1 のうち上 3 個の「1」を先に探索した後、下 3 個の「1」を探索したため、このような結果になっている。

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	1	0	0
1	1	0	1	0	1
1	0	0	1	1	1
1	1	0	1	0	1

図 2.4 文字列を”3D”で検索するフィールドの例

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	5	0	0
5	1	0	5	0	5
5	0	0	5	1	5
5	1	0	5	0	5

図 2.5 文字列を”3D”にしたときの探索結果の例

また、L字を形成しているブロックを探索したい場合は、[] と P を用いて、”3U[1P2R]”と記述する。3U という部分で、基準にしているブロック含めて上に 3 個のブロックが並んでいることを指定し、[1P2R] という部分で、上から 3 個目の位置のブロックから、そのブロックを含め右に 2 個ブロックが並んでいるという形状を指定している。基準になるブロックを含め、上に 3 つ、右に 2 つのブロックが並んでいる部分をフィールド上から探し出す。フィールド上にいくつかのブロックを配置したものが図 2.6 であり、そのフィールドの中から”3U[1P2R]”で探索した結果が図 2.7 である。

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
1	0	0	1	0	0
1	0	0	1	1	0
1	1	0	1	1	0

図 2.6 L 字を探索するフィールドの例

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
5	0	0	5	0	0
5	0	0	5	1	0
5	5	0	5	5	0

図 2.7 L 字を探索したときの探索結果の例

第 3 章

検証と考察

本章では、提案手法を用いて消去形状の法則性を表現することが出来るかを検証した。提案手法を用いて、実際に落ちものパズルゲームのブロックの消去条件として成立している、既存の落ちものパズルゲームの消去形状の再現を行った。また、落ちものパズルゲームのブロックの消去条件は、ルールによって様々な形状が想定されるため、いくつかの消去形状を想定し、様々な形状に対応できるかの検証を行った。あらかじめ用意した縦 10 マス、横 8 マスのフィールドのデータを用い、それに対して探索を行った。

3.1 既存のゲームの消去形状の再現

まずは既存の固定形型の落ちものパズルゲームである、テトリスとルミネス、ぷよぷよの消去形状の再現が出来るかを検証した。

3.1.1 テトリスの消去形状の再現

テトリスのブロックを消去するための条件は、ブロックをフィールドの横 1 行に隙間なく配置することである。今回用意したフィールドは、横 8 マスになっている。そのためテトリスの消去形状は、横 8 マスでブロックが並んでいる形状となる。本手法では、”8R”と記述することで消去

形状を再現することが出来る。実際に検証を行った結果、フィールドから横一行すべてが同じ数字で埋まっている部分を探索し、消去待機状態である「5」に置き換わっている。また、横一行すべてが埋まっていない行に関しては「1」のままになっている。よって、テトリスの消去形状を再現することが出来た。図 3.1 は探索する前のフィールドの状態であり、図 3.2 は文字列”8R”を元に探索し、該当箇所を消去待機状態である「5」に置き換えた後の探索結果となっている。

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

図 3.1 文字列”8R”で探索するフィールド

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1
5	5	5	5	5	5	5	5	5
0	0	0	0	1	1	1	1	1
0	1	1	1	1	1	1	1	1
5	5	5	5	5	5	5	5	5

図 3.2 文字列”8R”で探索した結果

3.1.2 ルミネスの消去形状の再現

ルミネスのブロックの消去するための条件は、同色のブロックで 2×2 の四角形を作り、その四角形が、フィールドを横方向に流れているタイムラインと呼ばれる線に触れることである。今回は、ブロックで 2×2 の四角形を作るといふ部分の再現を行う。本手法では、“2R[1P2D][2P2D]”と記述することで消去形状を再現することが出来る。実際に検証を行った結果、フィールドから同じ数字のブロックで 2×2 を形成している部分を探し出し、消去待機状態を示す「5」に置き換わっている。以下の図 3.3 は探索する前のフィールドの状態であり、図 3.4 は文字列“2R[1P2D][2P2D]”を元に探索し、該当箇所を消去待機状態である「5」に置き換えた後の探索結果となっている。フィールド上で、同じ数字で 3×3 を形成している部分も探し出しているが、これは 2×2 の形状が4つ集まっていると判断できるため、正しい挙動である。図 3.5 は「1」で 3×3 を形成している部分を探索する際の挙動の例である。

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0
0	0	1	1	0	1	1	1	1
1	1	0	1	0	1	1	1	1
1	1	0	1	1	1	1	1	1

図 3.3 ルミネスの再現で探索するフィールド

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
5	5	0	0	0	0	0	0	0
5	5	5	5	0	0	0	0	0
0	0	5	5	0	5	5	5	5
5	5	0	1	0	5	5	5	5
5	5	0	1	1	5	5	5	5

図 3.4 ルミネスの再現で探索した結果

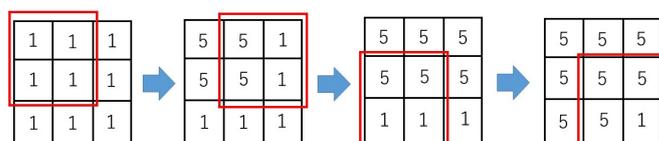


図 3.5 3×3での挙動の例

3.1.3 ぷよぷよの消去条件の再現

ぷよぷよのブロックを消去するための条件は、同色のブロックが上下方向または左右方向で4個以上並んでいることである。今回は、ぷよぷよでのブロックの色を、ブロックの数字に対応させ、同じ数字のブロックが上下方向または左右方向で4個以上並んでいる時に、消去判定を発生させることにする。本手法では、“4C”と記述することによって消去形状を再現することが出来る。実際に検証を行った結果、同じ数字のブロックが4個以上並んでいれば、ブロックの形状に関わ

らず、フィールド上から探し出し、消去待機状態である「5」に置き換えることが出来た。4種類のブロックを配置したフィールドの例を表したのが図 3.6 であり、そのフィールドに対して”4C”で探索を行ったものが図 3.7 である。

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	4
3	3	0	0	0	0	3	1
1	1	3	0	0	0	1	1
1	2	2	2	4	2	1	4
1	2	2	3	4	4	4	4

図 3.6 文字列”4C”で探索するフィールド

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	2	4
3	3	0	0	0	0	3	5
5	5	3	0	0	0	5	5
5	5	5	5	5	2	5	5
5	5	5	3	5	5	5	5

図 3.7 文字列”4C”で探索した結果

3.2 いくつかの形状の探索

固定形型の落ちものパズルゲームで表現できる消去形状として、以下の形状の検証を行った。

- 縦または横にブロックが並んでいる場合
- ブロックが T 字型に並んでいる場合
- ブロックが十字型に並んでいる場合
- ブロックが複雑な形に並んでいる場合

検証する形状の例を図で表したものが、図 3.8 である。

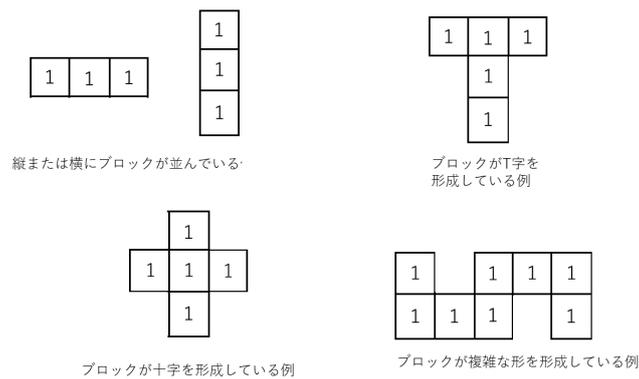


図 3.8 検証する形状の例

3.2.1 縦または横にブロックが並んでいる場合

ブロックを消去するための条件を、縦方向または横方向にブロックが 4 個以上並んでいる場合と設定する。本手法では、"4U|4R"と記述することで、縦方向または横方向にブロックが 4 個以上並んでいる形状を探し出すことが出来る。入力した文字列の 4U の部分で、縦方向に 4 個ブロックが並んでいるという消去形状を指定し、4R の部分で横方向に 4 個ブロックが並んでいるという消去形状を指定する。4U と 4R の間に | の文字を入れることによって、縦方向または横方向のど

明らかにブロックが4個並んでいる場合に、そのブロックを消去待機状態に置き換えることが出来る。結果として、フィールドから横に4個以上ブロックが並んでいる部分を探し出すことに成功した。探索するフィールドを示したものが図 3.9 であり、フィールドから横にブロックが4個以上並んでいる部分を探索した結果を示すのが図 3.10 である。

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0
1	0	1	1	1	1	1	0	0
1	0	1	0	1	1	1	1	1
1	0	1	0	0	1	1	1	1
1	0	1	1	1	1	1	0	0

図 3.9 縦または横方向のブロックを探索するフィールド

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	1	5	0	0	0	0	0	0
5	0	5	5	5	5	0	0	0
5	0	5	0	5	5	5	5	5
5	0	5	0	0	5	1	1	1
5	0	5	5	5	5	0	0	0

図 3.10 縦または横方向のブロックを探索した結果

3.2.2 T字にブロックが並んでいる場合

ブロックを消去するための条件を、横方向に3個ブロックが並んでおり、その内の中心のブロックから下に中心のブロックを含めて3個ブロックが並んでいる場合と設定する。本手法では、"3R[2P3D]"と記述することで、T字を形成しているブロックの消去形状を指定することが出来る。入力した文字列の3Rの部分で、T字の横方向に3個ブロックが並んでいる部分を表し、[2P3D]の部分で、横方向に3個並んでいるブロックのうち、左から2個目のブロックから、そのブロックを含め下方向に3個ブロックが並んでいるという形状を表している。結果として、フィールドからT字にブロックが並んでいる部分を探し出すことに成功した。探索するフィールドを示したものが図3.11であり、フィールドから、T字にブロックが並んでいる部分を探索した結果を示すのが図3.12である。

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0
0	0	0	0	0	1	0	0	0
1	1	1	0	1	1	1	1	0
0	1	0	1	1	0	1	0	0
0	1	0	1	1	0	1	0	0

図 3.11 T字を形成するブロックを探索するフィールド

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	5	5	5	0	0
0	0	0	0	0	5	0	0	0
5	5	5	0	1	5	5	5	0
0	5	0	1	1	0	5	0	0
0	5	0	1	1	0	5	0	0

図 3.12 T 字を形成するブロックを探索した結果

3.2.3 十字型にブロックが並んでいる場合

ブロックを消去するための条件を、縦方向に 5 個ブロックが並び、そのうちの上から 3 番目に位置しているブロックから、そのブロックを中心に左右方向に 2 個ずつブロックが並んでいる、十字型を形成している場合と設定する。本手法では、”3R3L3U3D”と記述することで十字型を形成しているブロックを探し出すことが出来る。結果として、十字型を形成しているブロックの探索に成功した。探索するフィールドを示したものが図 3.13 であり、フィールドから、ブロックが十字型を形成している部分を探索した結果を示すのが図 3.14 である。

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0
0	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0
0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	0	1	1

図 3.13 十字を形成するブロックを探索するフィールド

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0
0	1	5	1	1	1	1	1	1
0	0	5	0	0	0	0	0	0
5	5	5	5	5	0	0	0	0
0	0	5	0	0	0	0	1	1
0	0	5	0	0	0	0	1	1

図 3.14 十字を形成するブロックを探索した結果

本手法では、+の文字を用いると、数字で指定したブロックの個数以上のブロックが並んでいる場合にも、探索することが出来るようになる。+の文字を用いることによって、十字型を形成していれば、上下左右方向に並んでいるブロックの数に関係なく、フィールド上から探し出すような指定をすることが出来る。本手法では、“2R+2L+2U+2D+”と記述することによって、十字型を形成していれば、上下左右方向に並んでいるブロックの数に関係なくフィールドから探し出せる。結果として、フィールドから十字型を形成しているブロックを、それぞれの上下左右のブ

ロックの数に関係なく探し出すことに成功している。いくつかのブロックを配置した、探索するフィールドを示したものが図 3.15 であり、フィールドから、十字型を形成したブロックを探索した結果が図 3.16 である。

0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	1	1	1	1	1	1	0
0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	0	0	1	0	0
1	1	1	0	1	1	1	1
0	1	0	0	0	1	0	0

図 3.15 柔軟性を持つ十字を形成するブロックを探索するフィールド

0	0	0	0	0	0	0	0
0	0	0	5	0	0	0	0
0	0	0	5	0	0	0	0
0	0	0	5	0	0	0	0
0	5	5	5	5	5	5	0
0	0	0	5	0	0	0	0
0	0	0	5	0	0	0	0
0	5	0	0	0	5	0	0
5	5	5	0	5	5	5	5
0	5	0	0	0	5	0	0

図 3.16 柔軟性を持つ十字を形成するブロックを探索した結果

3.2.4 ブロックが複雑な形状を形成している場合

ブロックを消去するための条件を、矩形波の様な形状にブロックが並んでいる場合と設定する。本手法では、“2D [2P3R [3P2U [2P3R [3P2D]]]]”と記述することで、矩形波の様な形状の探索を行うことができる。結果として、フィールド上から指定したような形状を探し出すことに成功している。探索するフィールドを示したものが図 3.17 であり、フィールドから指定した形状を探索した結果を示すのが図 3.18 である。

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0
1	1	1	1	1	0	0	0	0
0	0	0	1	0	1	1	1	1
1	0	1	1	1	1	0	1	1
1	1	1	0	1	1	1	1	1

図 3.17 複雑な形状を探索するフィールドの例

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
5	1	5	5	5	0	0	0	0
5	5	5	1	5	0	0	0	0
0	0	0	5	0	5	5	5	5
5	0	5	5	5	5	0	5	5
5	5	5	0	5	1	1	1	1

図 3.18 複雑な形状を探索した結果のフィールドの例

3.3 考察

本手法によって、テトリスやルミノス、ぷよぷよといった既存の落ちものパズルゲームの消去形状の再現を行うことが出来た。また、落ちものパズルゲームで表現できる形状として、縦方向や横方向にブロックが並んでいる場合や、T字型、十字型にブロックが並んでいる場合、ブロックがある程度複雑な形状をとっている場合の指定も行うことが出来た。さらに、十字型を形成しているブロックに関しては、長さに関してある程度の柔軟性を持たせることも出来た。これは、縦方向や横方向にブロックが並んでいる場合や、ブロックがT字型の形状を形成している場合など、様々な形状に対しても柔軟性を持たせることが可能になっている。しかし本手法では、指定した形状のブロックの並んでいる数の比率を基に拡大されているような形状の指定ができない。例えば、ブロックが十字型に並んでいる場合で考える。ブロックを中心に、上下左右のブロックの個数の比率が1:1:1:1になっていたら消去するといったような指定が出来ず、上下左右の方向に1個以上ブロックが並んでいたら消去するというような指定しかできない。法則性を持つ形状により柔軟に対応させるには、さらなる機能を追加する必要があると考察する。

第 4 章

まとめ

本研究では、落ちものパズルゲームにおいて、文字列を用いて消去形状の法則性を表現することが出来る手法の提案を行った。提案手法を用いて、テトリスやルミネス、ぷよぷよの様な既存の落ちものパズルゲームの消去形状の再現を行うことが出来た。また、縦方向や横方向にブロックが並んでいる場合や、T字型、十字型にブロックが並んでいる場合、ブロックが複雑な形状を形成している場合の消去形状の指定をすることが出来た。

今後の展望として、法則性を持つ形状を法則性を持つ形状により柔軟に対応させるには、さらなる機能を追加する必要がある。「コラムス」[19]という落ちものパズルゲームでは、縦方向や横方向の他に、斜め方向にブロックが並んでいる場合にも消去が起こる。さらに、スーパーパズルファイター IIX[20]という落ちものパズルゲームでは、同色のブロックをつなげただけでは消去されず、破壊用の特殊なブロックをつなげなければブロックが消去されない。これらの消去条件を満たすために、斜め方向にブロックが並んでいる場合や、特殊なブロックに対応する必要がある。また、指定した部分に、ブロックがあっても無くてもよい、というワイルドカードのような機能を追加することでより多くの形状を指定することが出来るようになると思われる。

謝辞

本研究を行うにあたり、ご指導いただいた渡辺先生と阿部先生に心より感謝いたします。また、研究室の皆様、特に栗原先輩に深く感謝いたします。

参考文献

- [1] Tetris | The addictive puzzle game that started it all! <https://tetris.com/>. 参照:2019.12.23.
- [2] ぷよぷよポータルサイト. <http://puyo.sega.jp/portal/index.html>. 参照:2019.12.23.
- [3] ルミネスリマスター Lumines Remastered. <https://luminesremastered.com/ja/>. 参照:2019.11.25.
- [4] 不思議のダンジョン風来のシレン | スパイク・チュンソフト. https://www.spike-chunsoft.co.jp/shiren_sp/. 参照:2020.1.23.
- [5] 『ファンタシースターオンライン 2』公式サイト | SEGA. <http://pso2.jp/>. 参照:2020.1.23.
- [6] (PS2、PSP) ティル・ナ・ノグ～悠久の仁～ オフィシャル WEB ページ. https://www.ss-alpha.co.jp/products/tirnanog_consumer.html. 参照:2020.1.23.
- [7] 一貴前田, 博奥乃. 数独の問題作成支援システムの設計と開発. 全国大会講演論文集, Vol. 70, pp. 799–800, 2008.
- [8] 俊宗香川, 宏史手塚, 真理稲葉. 音楽の重要な構成要素 の抽出の提案-音楽ゲーム用譜面自動生成のために-. エンタテインメントコンピューティングシンポジウム 2015 論文集, Vol. 2015, pp. 326–333, 2015.

- [9] Nathaniel Love, Timothy Hinrichs, David Haley, Eric Schkufza, and Michael Genesereth. General game playing: Game description language specification. *Stanford University, Tech.Rep*, Vol. LG-2006, p. 01, 2006.
- [10] Michael Genesereth, Nathaniel Love, and Barney Pell. General game playing: Overview of the aaai competition. *AI magazine*, Vol. 2, p. 62, 2005.
- [11] Tom Schaul. A video game description language for model-based or interactive learning. *IEEE Conference on Computational Intelligence in Games (CIG)*, pp. 1–8, 2013.
- [12] Cameron Browne and Frederic Maire. Evolutionary game design. *IEEE Transaction on Computational Intelligence and AI in Games*, Vol. 2, pp. 1–16, 2010.
- [13] 栗原一浩, 阿部雅樹, 渡辺大地. 落ちものパズルゲームルール記述言語の提案. ゲームプログラミングワークショップ2019 論文集, Vol. 2019, pp. 177–180, 2019.
- [14] 栗原一浩, 阿部雅樹, 渡辺大地. 落ちものパズルゲーム共通ルール記述言語を用いたゲームルールの自動生成. 情報処理学会研究報告.
- [15] 渡部孝幸, 宮崎佳典. 二次元の位置構造に着目した数式のパターンマッチング手法. 情報知識学会誌, Vol. 22, pp. 253–271, 2012.
- [16] 渡部孝幸, 宮崎佳典. 正規表現を用いた数式検索手法の提案. 情報処理学会論文誌, Vol. 56, pp. 1417–1427, 2015.
- [17] Bruce R. Miller and Abdou Youssef. Technical aspects of the digital library of mathematical functions. *Annals of Mathematics and Artificial Intelligence*, Vol. 38, pp. 121–136, 2003.
- [18] Moody E. Altamimi and Abdou S. Youssef. Wildcards in math search, implementation issues. *CAINE/ISCA*, pp. 90–96, 2007.
- [19] コラムス | メガドライブミニ | セガ. <https://sega.jp/mdmini/soft/columns.html>. 参

照:2020.1.23.

[20] スーパーパズルファイター IIX : CAPCOM. <http://www.capcom.co.jp/product/detail.php?id=32>. 参照:2020.1.23.