

# SAT ソルバを用いたライフゲームにおける物体の合成手法

西村 春輝 長谷部 浩二

セルオートマトンは、セルという小さな単位が互いに結合し相互作用を起こす数理モデルの一種である。その中でも Conway のライフゲームは、物体と呼ばれる特殊な性質を持ったパターンが出現することで知られており、新種の物体の生成手法についての研究が盛んに行われている。本研究では、充足可能性問題に依拠した近年の物体生成の手法を応用し、複数の物体を合成することによりキメラと称する新種の物体を生成する手法を提案する。その基本となるアイデアは、2つの物体の断片を並べ、それらの間に適切なパターンを補完することで接合するというものである。これにより、物体全体を生成する場合と比べ探索範囲を大幅に削ることや、キメラ同士のキメラを作ることで全く新しい物体を作ることが可能になる。本研究では SAT ソルバを用いた実装で接合用のパターンを自動生成し、固定物体、振動子、移動物体の 3つのカテゴリーを対象に実際に合成を行い提案手法の有効性を示す。

## 1 序論

セルオートマトンは、セルと呼ばれる小さな単位が連結し相互作用を起こすことで多様かつ複雑な振る舞いを見せる数理モデルである。中でも Conway が考案したライフゲーム [1] は、物体と呼ばれる特殊な性質を持つパターンが出現することで知られている。ライフゲーム内の物体は生物に似たその振る舞いから多くの研究者の注目を集めている。また、Buckingham らの研究 [2] が示すように計算機をライフゲーム内に構築する際に物体を回路の部品として利用可能であることから、今もなお新種の物体を発見するための試みが多くなされている。例えば、Harold により De Bruijn の表を用いた手法 [3] が提案され、一部は物体生成ツールとして実装されている [4]。近年では Knuth により物体の生成問題を充足可能性問題へ選

元し、SAT ソルバに解かせる手法 [5] が提案されている。しかし、これらの物体生成の手法にはいくつかの課題が存在する。De Bruijn の表や SAT ソルバを用いる手法は、特定の矩形範囲内に入る物体を全て網羅的に探索できるという利点がある一方、条件を満たすパターンを総当たりで探すのと本質的には同じことをしており、物体のサイズが大きくなればその分だけ探索に時間がかかる。この際に既存の物体の一部分を利用することができれば計算量を減らせる可能性があるが、利用した部分にどのような加工を施せば目的の物体が得られるかについては不明な点が多い。

本研究では、ライフゲームにおいて既存の物体が持つ形状の規則性を利用し比較的大きなサイズの物体を生成することを目的とする。より具体的には、既存の物体の断片に別のパターンや他の物体の断片を補完することで、新しい種類の物体を生成する（本論文ではこれをキメラと呼ぶ）。ある断片に対して適用できる補間パターンは限られているが、本研究では近年の SAT ソルバを用いた物体生成の手法を拡張し、補間パターンの導出に応用する。実装の際 SAT ソルバとして Z3py [6] を用い、固定物体、振動子、移動物体の 3種類のカテゴリーのうちいずれか 1つに分類される物体を中心に合成を試みたところ、全てのカタ

\*Object Synthesis Method in Game of Life Using SAT Solver

This is an unrefereed paper. Copyrights belong to the Author(s).

Haruki Nishimura, 筑波大学情報学群情報科学類, College of Information Science, University of Tsukuba.

Koji Hasebe, 筑波大学システム情報系, Faculty of Engineering, Information and Systems, University of Tsukuba.

ゴリーで合成に成功した。振動子を例にとると、周期 2 の振動子の断片と周期 3 の振動子の断片の間にパターンを補完することで周期 6 の振動子を得られた。

本研究の利点は主に 3 つ存在する。1 つ目にカメラの生成では物体の一部分がすでに完成しており、同じサイズの物体を何も無いところから生成する場合と比較して探索範囲が小さくなるため、より早く物体を発見することが可能となる。2 つ目にカメラどうしのカメラを作り続けることで、元の物体とは全く違う形状の物体を漸進的に作っていくことが理論上可能であり、全く新しい物体を生成する際にも利用可能な手段となり得る。3 つ目に、共通の部品を持つ物体を比較することで、物体の形状が持つ大局的な規則性を知ることができる。

本論文の構成は以下の通りである。第 2 節ではライフゲームの概要を示す。第 3 節では SAT ソルバを用いた物体の合成手法について説明する。第 4 節では SAT ソルバとして Z3py を用いた際の物体の合成例を示す。第 5 節では先行研究について述べる。第 6 節では結論と今後の課題を述べる。

## 2 ライフゲームの概要

### 2.1 ライフゲームの規則

セルオートマトンは、格子空間上に敷き詰められた複数のセルが周囲のセルの状態に応じて状態を変化させるオートマトンの一種である。その中でもライフゲームは、0 または 1 の状態をとるセルが 2 次元の格子空間に敷き詰められ、以下の 3 つの規則に従って時間とともに遷移していく。

- 誕生：自身の状態が 0、周囲 8 セルのうちちょうど 3 セルの状態が 1 のとき、次の時刻でセルは 1 に状態遷移する。
- 生存：自身の状態が 1 であり、周囲 8 セルのうち 3 セルか 2 セルの状態が 1 のとき、次の時刻でセルは 1 に状態遷移する。
- 死滅：上記の 2 つの規則のいずれにもあてはまらないセルは、次の時刻で 0 に状態遷移する。

すなわち、ライフゲームは以下のように定義できる。

**定義 1.** (ライフゲーム) 自然数の集合  $\mathbb{N}$  に対して  $P \subseteq \mathbb{N} \times \mathbb{N}$  を領域といい、 $P$  の  $i$  行  $j$  列のセルを

$(i, j)$  と表す。また時刻  $t = 0, 1, 2, \dots (\in T = \mathbb{N})$  とする。ある時刻  $t$  におけるセル  $(i, j)$  の状態 (0 または 1) を関数  $\sigma : P \times T \rightarrow \{0, 1\}$  で定義し、その値を  $x_{i,j}^t$  と表す。このとき、 $P$  を領域とするライフゲームとは、以下の条件を満たす  $\sigma$  (ライフゲームの規則と呼ぶ) と、任意の  $i, j, t$  について  $x_{i,j}^{t+1} = \sigma(x_{i,j}^t)$  を満たす領域  $P_{Life}$  の 2 つ組のこととする。

$$\sigma(x_{i,j}^t) = \begin{cases} 1 & (\text{if } \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} x_{m,n}^t = 3) \\ 1 & (\text{if } x_{i,j}^t = 1 \text{ and } \\ & \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} x_{m,n}^t = 2) \\ 0 & \text{otherwise} \end{cases}$$

### 2.2 ライフゲームの規則と論理式

セルの状態を、1 を真として 0 を偽とするような真偽値とみなすことにより、ライフゲームの規則  $\sigma$  は 9 つのセルの状態を引数にとって 1 つのセルの状態を返す命題関数  $\delta$  に書き換えることが可能である。本論文では、 $\delta(V(x_{i,j}^t)) = \sigma(x_{i,j}^t)$  (where  $V(x_{i,j}^t) = (x_{i-1,j-1}^t, x_{i-1,j}^t, x_{i-1,j+1}^t, x_{i,j-1}^t, x_{i,j}^t, x_{i,j+1}^t, x_{i+1,j-1}^t, x_{i+1,j}^t, x_{i+1,j+1}^t)$ ) と定義する。

### 2.3 物体

ライフゲーム内のセルの状態のうち 1 を黒に、0 を白に対応させて毎時描画していくと、黒いパターンが拡大と縮小を繰り返す様子を観察できる。その一方で、何度遷移させても形状が変化しない固定物体や、 $p$  回の遷移で元の形状に戻る振動子、 $p$  回の遷移を通して移動しながら元の形状に戻る移動物体など、特殊な性質を持った物体と呼ばれるパターンが出現することがある。固定物体は、任意の時刻  $t$  における形状と位置が時刻  $t+1$  におけるそれと一致することから以下のように定義できる。

**定義 2.** (固定物体) 矩形範囲  $R (\subseteq P_{Life})$  において、任意の  $t \in T$  と任意のセル  $(i, j) \in R$  で  $x_{i,j}^{t+1} = x_{i,j}^t$  すなわち  $\sigma(x_{i,j}^t) = x_{i,j}^t$  が成立するとき、 $R$  を固定物体と呼ぶ。

また振動子は、任意の時刻  $t$  におけるパターンの形状と位置が時刻  $t+p$  のそれと一致することから、以下のように定義できる。

**定義 3.** (振動子) 矩形範囲  $R (\subseteq P_{Life})$  において、

任意の  $t \in T$  と任意のセル  $(i, j) \in R$  で  $x_{i,j}^{t+p} = x_{i,j}^t$  すなわち  $\sigma^p(x_{i,j}^t) = x_{i,j}^t$  が成立するとき、 $R$  を周期  $p$  の振動子と呼ぶ。

なお  $\sigma^p$  は、 $\sigma$  を  $p$  回適用することを意味する。一般に、 $\sigma^p$  を遷移規則に用いるセルオートマトンでは、1つのセルの状態を決定する際にその周囲  $p$  セルを参照することが分かっている。すなわち、 $\sigma^p$  は  $(2p+1)^2$  個のセルの状態を入力とする関数であり、 $\sigma^p : S^{(2p+1)^2} \rightarrow S$  (ただし、 $S = \{0, 1\}$ ) となる写像である。

移動物体は、任意の時刻  $t$  におけるパターンの形状が時刻  $t+p$  で一致し、かつ縦に  $a$  セル、横に  $b$  セル移動していることから、以下のように定義できる。

**定義 4.** (移動物体) 矩形範囲  $R(\subseteq P_{Life})$  において、任意の  $t \in T$  と任意のセル  $(i, j) \in R$  で  $x_{i,j}^{t+p} = x_{i-a,j-b}^t$  すなわち  $\sigma^p(x_{i,j}^t) = x_{i-a,j-b}^t$  が成立するとき、 $R$  を周期  $p$ 、 $(a, b)$  方向への移動物体と呼ぶ。

以上の定義から分かるように、固定物体は周期 1 の特別な振動子であり、なおかつ周期 1、 $(0, 0)$  方向への特別な移動物体である。また、振動子は周期  $p$ 、 $(0, 0)$  方向への特別な移動物体であるといえる。

### 3 物体の合成

#### 3.1 補完パターン

本論文で述べる物体の合成とは、同じ周期  $p$  を持つ 2 つの断片の間にバッファとなる適切なパターンを埋め込んでつなげ、新しい物体を作ることを指す。異なる周期を持つ 2 つの断片の合成は、両方の周期について公倍数をとり同一の周期にそろえることで可能になる。接合によって生まれた新しいパターンが周期  $p$  の物体として成立するためには、パターン内部の全てのセル及びその周囲  $p$  セルの領域において、 $\sigma^p$  で遷移させた際に状態が変わらないことが求められる。なぜなら、 $\sigma^p$  下ではあるセルの状態がその周囲  $p$  セルの状態に影響を与えるからである。

すなわち物体の合成は以下のように定義される。ここでは 2 つの物体を左右に並べて合成する場合のみを定義するが、一般性を失うことなく、縦に 2 つに並べた場合についても定義をすることができる。

**定義 5.** (物体の合成)  $m_{max}, m_{min}, n_{max}, n_{min}, mid_u, mid_v, a, b, p \in \mathbb{N}$ ,  $mid_u < mid_v$  とする。また 3 つの矩形範囲  $R_0, R_1, R_2 \subseteq P_{Life}$  が存在し、 $R_2 = \{(i, j) \mid m_{min} \leq i \leq m_{max}, n_{min} \leq j \leq n_{max}\}$  とし、矩形範囲  $R_l, R_r, R_b$  を以下のように定める。

- $R_l = \{(i, j) \mid m_{min} \leq i \leq m_{max}, n_{min} \leq j \leq mid_u\}$  について、 $R_l \subseteq R_0$
- $R_r = \{(i, j) \mid m_{min} \leq i \leq m_{max}, mid_v \leq j \leq n_{max}\}$  について、 $R_r \subseteq R_1$
- $R_b = \{(i, j) \mid m_{min} \leq i \leq m_{max}, mid_u + 1 \leq j \leq mid_v - 1\}$  について、 $R_b \not\subseteq R_0$  かつ  $R_b \not\subseteq R_1$

このとき、

- $R_l$  を  $R_0$  の左半分の断片といい、セルの列  $C_l = \{(i, mid_u) \mid m_{min} \leq i \leq m_{max}\}$  を、 $R_l$  の切り口と呼ぶ。
- $R_r$  を  $R_1$  の右半分の断片といい、セルの列  $C_r = \{(i, mid_v) \mid m_{min} \leq i \leq m_{max}\}$  を、 $R_r$  の切り口と呼ぶ。
- $R_b$  を  $R_l$  と  $R_r$  の補完パターンといい、2 つのセルの列  $C_{bl} = \{(i, mid_u + 1) \mid m_{min} \leq i \leq m_{max}\}$  と  $C_{br} = \{(i, mid_v - 1) \mid m_{min} \leq i \leq m_{max}\}$  を  $R_b$  の切り口と呼ぶ。

また、矩形範囲  $R_s = \{(i, j) \mid m_{min} - p \leq i \leq m_{max} + p, n_{min} - p \leq j \leq n_{max} + p\}$  において、全ての  $(i, j) \in R$  で論理式

$$\sigma^p(x_{i,j}) \leftrightarrow x_{(i-a),(j-b)} \quad (1)$$

が真になるとき、 $R_2$  を断片  $R_l$  と断片  $R_r$  の合成と定義する。

定義 5 において矩形範囲  $R_2$  の内部およびその周囲  $p$  セルで式 (1) が成立していなくてはならないことになっているが、実際には  $R_b$  の内部およびその周囲  $p$  セルで式 (1) が成立するように  $R_b$  のパターンを選ぶだけで物体の合成が成功する。なぜなら、本研究では断片  $R_l$  と  $R_r$  が既存の物体を半分に割ったものである状況を想定しており、 $R_l$  や  $R_r$  に含まれるセルのうち矩形範囲  $R_b$  の周囲  $p$  セルより外に位置するものでは既に式 (1) が成立しているからである。つまり、 $R_b$  とその周囲  $p$  セルの矩形範囲をまとめて  $R_m$  とするならば、以下の論理式が真になるよう  $R_b$  内のセルの

状態を選んだとき合成が成功する。

$$\bigwedge \sigma^p(x_{i,j}) \leftrightarrow x_{i-a,j-b} \quad (\text{where } (i,j) \in R_m) \quad (2)$$

なお、ある論理式を真にする変数の割り当ての有無、及びその割り当ての具体的な内容を導出する問題は充足可能性問題 [7][8] と呼ばれている。この問題には割り当てを総当たりしなくてはならないケースがあり、NP 完全問題であることが知られている。しかし複数の節が連言でつながられた CNF と呼ばれる論理式においては、DPLL や矛盾からの節学習など、変数の割り当てを高速で網羅したり探索範囲を削ったりするためのアルゴリズムが発見されており、それらを実装した SAT ソルバと呼ばれるツールが実用化されている。上の論理式を CNF に変換し SAT ソルバに解かせることで、補完パターンが導出できるのである。

### 3.2 固定物体用の補完パターン

固定物体の場合補完パターンはその周囲 1 セルの内容に影響を与えるため、定義 2 を用いて固定物体の補完パターンを定義すると以下ようになる。

**定義 6.** (固定物体の補完パターン)  $e_x, e_y \in \mathbb{N}$  とし、 $R_e \subseteq P_{Life}$  かつ  $R_e = \{(i,j) \mid e_y - 1 \leq i \leq e_y + m, e_x - 1 \leq j \leq e_x + n, m > 0, n > 0\}$  となる大きさ  $(m+2)(n+2)$  の矩形範囲  $R_e$  において、 $\sigma(x_{i,j}^t) = x_{i,j}^t$  が全ての  $(i,j) \in R_e$  で成立するとき、 $R_c \subset R_e$  かつ  $R_c = \{(i,j) \mid e_y \leq i \leq e_y + m - 1, e_x \leq j \leq e_x + n - 1\}$  となる大きさ  $m \times n$  の矩形範囲  $R_c$  を固定物体の補完パターンと呼ぶ。

論理式  $\delta$  が  $\sigma$  と等価であることを利用すると、定義 6 より補完パターン  $R_c$  は以下の論理式を真にする。

$$\bigwedge_{i=e_y-1}^{e_y+m} \bigwedge_{j=e_x-1}^{e_x+n} (\delta(V(x_{i,j}^t)) \leftrightarrow x_{i,j}^t) \\ (\text{where } V(x_{i,j}^t) = \{x_{i+u,j+v}^t \mid -1 \leq u, v \leq 1\}) \quad (3)$$

これにより固定物体を合成する問題は、充足可能性問題に変換される。なお、(3) の論理式を固定物体における補完パターンの制約と呼び、その中の論理式  $\delta(V(x_{i,j}^t)) \leftrightarrow x_{i,j}^t$  を固定物体におけるセルの制約と呼ぶ。

例えば、図 1 の  $6 \times 2$  セルの水色枠内で補完パターンを作る場合、オレンジ枠内で示された  $8 \times 4$  セルに

ついてセルの制約を作り論理積でつなげることで、補完パターンの制約を表現できる。1 つのセルについて制約を作るにはその周囲 1 セルの状態が必要であるから、ピンク枠内の  $10 \times 6$  セルが参照される。水色枠内のセルの状態は変数、その外にあるセルの状態は定数となる。ピンク色の領域の左上の角を  $(1,1)$  とするとき、補完パターンの制約は以下のように表される。

$$\bigwedge_{i=2}^9 \bigwedge_{j=2}^5 (\delta(V(x_{i,j}^t)) \leftrightarrow x_{i,j}^t) \quad (4)$$

このとき、 $(i,j) = (2,5)$  のときのセルの制約は以下の論理式で表される。

$$0 \leftrightarrow \delta(0, 0, 0, 0, 0, 0, x_{3,4}^t, 1, 1) \quad (5)$$

また、 $(i,j) = (6,4)$  のときのセルの制約は、以下の論理式で表される。

$$x_{6,4}^t \leftrightarrow \delta(x_{5,3}^t, x_{5,4}^t, 1, x_{6,3}^t, x_{6,4}^t, 0, x_{7,3}^t, x_{7,4}^t, 1) \quad (6)$$

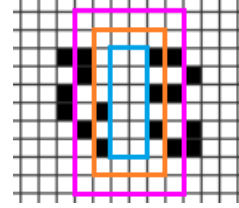


図 1 固定物体の接合例

### 3.3 振動子用の補完パターン

振動子の補完パターンは、定義 3 より  $\sigma^p : S \rightarrow S$  の上ではセルの状態が変化しない。補完パターンが周囲  $p$  セルの状態遷移に影響を与えることを考慮すると、振動子の補完パターンは以下のように定義される。

**定義 7.** (振動子の補完パターン)  $e_x, e_y, p \in \mathbb{N}$  とし、 $R_e \subseteq P_{Life}$  かつ  $R_e = \{(i,j) \mid e_y - p \leq i \leq e_y + m + p - 1, e_x - p \leq j \leq e_x + n + p - 1, m > 0, n > 0\}$  となる大きさ  $(m+2p)(n+2p)$  の矩形範囲  $R_e$  において、 $\sigma^p(x_{i,j}^t) = x_{i,j}^t$  が全ての  $(i,j) \in R_e$  で成立するとき、 $R_c \subset R_e$  かつ  $R_c = \{(i,j) \mid e_y \leq i \leq e_y + m - 1, e_x \leq j \leq e_x + n - 1\}$  となる大きさ  $m \times n$  の矩形範囲  $R_c$  を振動子の補完パターンと呼ぶ。

論理式  $\delta^p$  が  $\sigma^p$  と等価であることを利用すると、

定義 7 より補完パターン  $R_c$  は以下の論理式を真にする。

$$\bigwedge_{i=e_y-p}^{e_y+m+p-1} \bigwedge_{j=e_x-p}^{e_x+n+p-1} (\delta^p(V(x_{i,j}^t)) \leftrightarrow x_{i,j}^t)$$

(where  $V(x_{i,j}^t) = \{x_{i+u,j+v}^t \mid -p \leq u, v \leq p\}$ ) (7)

これにより振動子を合成する問題は、充足可能性問題に変換される。なお、(7) の論理式を周期  $p$  の振動子における補完パターンの制約と呼び、その中の論理式  $\delta^p(V(x_{i,j}^t)) \leftrightarrow x_{i,j}^t$  を周期  $p$  の振動子におけるセルの制約と呼ぶ。

例えば、図 2 の  $7 \times 3$  セルの水色枠内で補完パターンを作る場合、オレンジ枠内で示された  $11 \times 7$  セルについてセルの制約を作る必要がある。1 つのセルの制約を作る際に周囲  $p$  セルの状態が  $\delta^p$  に代入されるため、ピンク枠内の  $15 \times 11$  セルが参照される。水色枠内のセルの状態は変数、その外にあるセルの状態は定数となる。ピンク色の領域の左上の角を  $(1, 1)$  とするとき、補完パターンの制約は以下のように表される。

$$\bigwedge_{i=3}^{13} \bigwedge_{j=3}^9 (\delta^2(V(x_{i,j}^t)) \leftrightarrow x_{i,j}^t)$$

(where  $V(x_{i,j}^t) = \{x_{i+u,j+v}^t \mid -2 \leq u, v \leq 2\}$ )

このとき、 $(i, j) = (9, 4)$  のときのセルの制約は以下の論理式で表される。

$$0 \leftrightarrow \delta(1, 0, 0, x_{7,5}^t, x_{7,6}^t, 0, 0, 0, x_{8,5}^t, x_{8,6}^t, \dots, 0, 0, 1, x_{11,5}^t, x_{11,6}^t) \quad (8)$$

また、 $(i, j) = (6, 7)$  のときのセルの制約は、以下の論理式で表される。

$$x_{6,7}^t \leftrightarrow \delta(0, 0, 0, 0, 0, x_{5,5}^t, x_{5,6}^t, x_{5,7}^t, 0, 0, \dots, x_{8,5}^t, x_{8,6}^t, x_{8,7}^t, 0, 0) \quad (9)$$

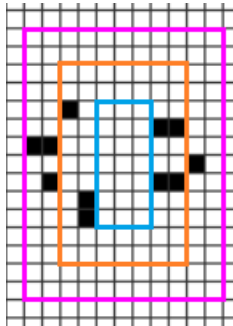


図 2 周期 2 の振動子の接合例

### 3.4 移動物体用の補完パターン

移動物体の補完パターンは、定義 4 より  $\sigma^p : S \rightarrow S$  を適用すると縦に  $a$  セル、横に  $b$  セル移動する。補完パターンが周囲  $p$  セルの状態遷移に影響を与えることを考慮すると、移動物体の補完パターンは以下のように定義される。

**定義 8.** (移動物体の補完パターン)  $e_x, e_y, p, a, b \in \mathbb{N}$  とし、 $R_e \subseteq P_{Life}$  かつ  $R_e = \{(i, j) \mid e_y - p \leq i \leq e_y + m + p - 1, e_x - p \leq j \leq e_x + n + p - 1, m > 0, n > 0\}$  となる大きさ  $(m + 2p)(n + 2p)$  の矩形範囲  $R_e$  において、 $\sigma^p(x_{i,j}^t) = x_{i-a,j-b}^t (-p \leq a, b \leq p)$  が全ての  $(i, j) \in R_e$  で成立するとき、 $R_c \subset R_e$  かつ  $R_c = \{(i, j) \mid e_y \leq i \leq e_y + m - 1, e_x \leq j \leq e_x + n - 1, m > 0, n > 0\}$  となる大きさ  $m \times n$  の矩形範囲  $R_c$  を周期  $p$ 、 $(a, b)$  方向への移動物体の補完パターンと呼ぶ。

論理式  $\delta^p$  が  $\sigma$  と等価であることを利用すると、定義 8 を用いて補完パターン  $R_c$  は以下の論理式を真にする。

$$\bigwedge_{i=e_y-p}^{e_y+m+p-1} \bigwedge_{j=e_x-p}^{e_x+n+p-1} (\delta^p(V(x_{i,j}^t)) \leftrightarrow x_{i-a,j-b}^t)$$

(where  $V(x_{i,j}^t) = \{x_{i+u,j+v}^t \mid -p \leq u, v \leq p\}$ ) (10)

これにより移動物体を合成する問題は、充足可能性問題に変換される。なお、(10) の論理式を周期  $p$ 、 $(a, b)$  方向への移動物体における補完パターンの制約と呼び、その中の論理式  $\delta^p(V(x_{i,j}^t)) \leftrightarrow x_{i-a,j-b}^t$  を周期  $p$  の移動物体におけるセルの制約と呼ぶ。

### 3.5 補完パターンと物体の成立

周期  $p$ 、 $(a, b)$  方向への移動物体から取り出された 2 つの断片を補完パターンで接合することにより、新しいパターンが生成される。これを本論文ではキメラと呼ぶ。キメラが周期  $p$ 、 $(a, b)$  方向への移動物体であることは以下の定理を用いて示すことができる。

**定理 1.** (合成による物体の成立) 矩形範囲  $R (\subseteq P_{Life})$  を 3 つの矩形範囲  $R_0, R_1, R_2$  を含む矩形範囲とし、全ての  $(i, j) \in R_0 \cup R_1$  について  $\sigma^p(x_{i,j}^t) = x_{i-a,j-b}^t (-p \leq a, b \leq p)$  が成り立つとき、全ての  $(i, j) \in R_2$  においても  $\sigma^p(x_{i,j}^t) = x_{i-a,j-b}^t$  が成り立つならば、矩形範囲  $R$  は周期  $p$ 、 $(a, b)$  方向への移動

物体となる.

*Proof.*  $R_0, R_1, R_2$  の定義より  $R_0 \cup R_1 \cup R_2 = R$  である. したがって,  $x_{i,j}^t \in (R_0 \cup R_1 \cup R_2)$  となる全ての  $(i, j)$  で  $\sigma^p(x_{i,j}^t) = x_{i-a,j-b}^t (-p \leq a, b \leq p)$  が成立しているとき, 全ての  $(i, j) \in R$  について  $\sigma^p(x_{i,j}^t) = x_{i-a,j-b}^t$  が成立する. 定義 4 より,  $R$  は周期  $p, (a, b)$  方向への移動物体である.  $\square$

2つの物体の断片と補完パターンは, それぞれ先の定理における  $R_0, R_1, R_2$  に相当する. 補完パターンは定義 8 より,  $R_0$  と  $R_2$  の境界のパターンと  $R_1$  と  $R_2$  のそれを変更しない. そのため全ての  $(i, j) \in R_0 \cup R_1$  について  $\sigma^p(x_{i,j}^t) = x_{i-a,j-b}^t (-p \leq a, b \leq p)$  が成立する. また定義 8 より, 補完パターン内のセル  $(i, j)$  は全て  $\sigma^p(x_{i,j}^t) = x_{i-a,j-b}^t (-p \leq a, b \leq p)$  を満たす. したがって全ての  $(i, j) \in R_2$  において  $\sigma^p(x_{i,j}^t) = x_{i-a,j-b}^t (-p \leq a, b \leq p)$  が成立している. よって定理 1 より, キメラは周期  $p, (a, b)$  方向への移動物体である.

2つの振動子の断片と補完パターンから成る複合体も, 定理 1 より振動子であることが証明される. なぜなら, 定理 1 において  $(a, b) = (0, 0)$  を代入することで証明できるからである. 同様に, この定理に  $p = 1, (a, b) = (0, 0)$  を代入することで固定物体の断片と補完パターンから成るキメラが固定物体であることを証明できる.

## 4 合成結果と考察

### 4.1 実行環境

本研究では, 命題関数  $\delta^p$  とセルの制約の生成に python ライブラリ Z3py を用い, また同ライブラリを用いて補完パターンの制約を充足可能にする変数の割り当てを導出した.

### 4.2 振動子の合成

以下では振動子の合成例を示す. 図 3 の物体 (a),(b),(c) は, それぞれ周期 2 の振動子, 固定物体, 周期 3 の振動子である. 初めに (b) の断片の左端を周期 2 の振動子の一部とみなして (a) のピンク枠内の部分と接合した. その際に, 青枠で囲まれたより

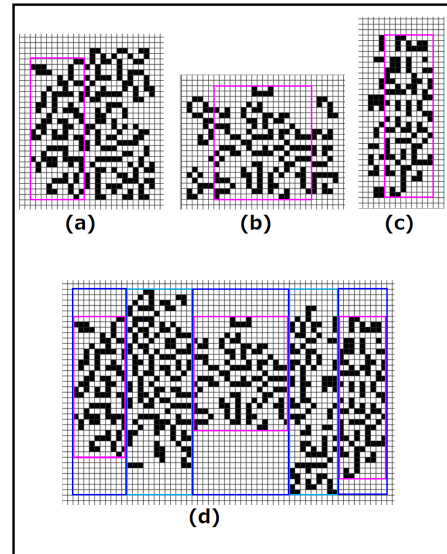


図 3 振動子の接合例

広い領域を定義 5 における断片とみなし部分的な合成を行った. 次に (b) の断片の右端を周期 3 の振動子の一部とみなし, (c) のピンク枠内の部分と接合した. この際も, 青枠で囲まれたより広い領域を定義 5 における断片とみなし部分的な合成を行った. 周期の違う振動子を固定物体を介して合成することに成功し, (d) に示す周期 6 の振動子を得た. 1 回目の接合におよそ 1 分, 2 回目の接合におよそ 4 分 50 秒を要した.

### 4.3 移動物体の合成

以下では移動物体の合成例を示す. 図 4 の (a),(b) は, どちらも周期 2,  $(0, -1)$  方向への移動物体である. これらからピンク枠内の一部分を取り出し青枠内で示した断片を作り, (c) に示すように水色枠内の補完パターンを間に挟んで周期 2,  $(0, -1)$  方向への移動物体を得た. 補完パターンの導出にはおよそ 1 分 50 秒かかった.

### 4.4 考察

補完パターンの制約に基づき適切な接合が行えることを確認した. 特に振動子の合成において, 固定物体を振動子として利用することで異なる周期を持つ

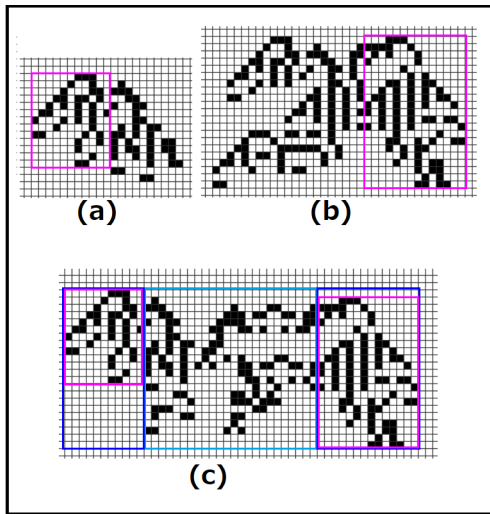


図 4 移動物体の接合例

振動子の合成に成功しており、これを拡張すれば限りなく大きな周期の振動子を作ることができる。具体的には、別々の周期を持つ振動子を大量に用意して、一部分を切り取り固定物体を用いて列車のように連結することにより、それらの周期が掛け合わさった周期の長い振動子を得られる。合成結果の図を見ると補完パターンのサイズが断片と比較して同じくらいかそれより大きくなっており、接合するためには断片同士の間隔が十分大きい必要があるとわかる。補完パターンの最小のサイズについてより詳しく調査する必要があるといえる。また、各セルの制約を CNF (連言標準形) に変換するまでに時間がかかっているようであり、Tseitin 変換や先行研究を参考にセルの制約を効率よく CNF に変換する手段を探す必要がある。

## 5 関連研究

### 5.1 De Bruijn の表を用いた物体生成

De Bruijn の表 [3] は、ライフゲームの物体のうち固定物体と呼ばれるものを生成するために使われる表である。ライフゲームの固定物体に含まれる  $3 \times 3$  セルの部分的なパターンは全て、ある 284 種類のパターンの集合  $D$  の要素になっていることが確定している。したがって、 $D$  からある要素を取出してそれらを重ね合わせることで固定物体を組み立てること

が可能になる。初めに  $D$  に属するパターンを横に 1 セルずつずらしながら重ねていき、横に細長いパターンを可能な限り多く作っておく。この際に De Bruijn の表が利用される。この表では  $D$  に含まれる  $3 \times 3$  のパターンが大きさ  $3 \times 2$  の小さなセグメント 2 つの接続規則に変換される。この接続規則にしたがってセグメントをつなげていくことで、 $3 \times n$  の細長いパターンが生成される。次にこの  $3 \times n$  のパターンを縦に 1 セルずつずらしながら重ねていき、固定物体を完成させる。この際にも De Bruijn の表が利用される。この表では  $3 \times n$  のパターンが  $2 \times n$  の小さなセグメントの接続規則に変換される。この接続規則に従ってセグメントを  $m - 1$  回つなげていくことで、大きさ  $m \times n$  の固定物体が完成する。

De Bruijn の表を用いて物体を生成するツールとして D. Eppstein による gfind [4] が知られており、移動物体の探索の際に De Bruijn の表が応用されている。移動物体も振動子も、ある 2 次元セルオートマトン下では固定物体としてふるまうため、De Bruijn の表を用いた物体の自動生成は他の種類の物体の生成に応用可能である。

### 5.2 SAT ソルバを用いた生成

有界モデル検査 [9] とは、ある状態遷移系の遷移のパスが特定の性質を満たすかどうかを判定する手法の 1 つである。現実のシステムは際限なく状態遷移のパスが分岐して増えるため、それらのパスを 1 つずつ調べ性質を検査することは困難である。そこで有界モデル検査では、初期状態からの遷移を有限回数に抑え有限な長さを持つパスについて性質を検査する。この検査においてパスが満たすべき性質は論理式で表現され、この式の充足可能性を判定することで、当該の性質を持ったパスが存在するかどうかを確認できる。

Knuth は有界モデル検査で利用される論理式を応用し、ライフゲームにおける物体の生成問題を充足可能性問題に変換する手法を提案した [5]。  $T(X, X')$  を、ライフゲームにおいて平面中のパターン  $X$  が  $X'$  に遷移するときに真を返す論理式であるとする。このとき、論理式  $\bigwedge_{k=1}^n T(X_{k-1}, X_k)$  (5.2) は、パターン  $X_0$  に  $n$  回ライフゲームの規則を適用すると  $X_n$  に

なる際に真を返す。もし (5.2) において ( $X_0 = X_n$ ) であるなら, この式は  $X_0$  が周期  $n$  の振動子であるときに真を返す論理式となり, この式を CNF に変換して SAT ソルバで解けば振動子の具体的なパターンが求まる。ほかにも 5.2 に対して様々な条件を追加したり変数の内容を変更したりすることで固定物体や移動物体, エデンの園などの導出に利用可能な論理式を得ることができる。なお,  $T(X, X')$  を CNF に変換する手法が Knuth により提案されており [5], 5.2 を CNF に変換して SAT ソルバで利用することが容易になっている。また, 5.2 の論理式と SAT ソルバを用いて物体を探索するツールが実装されており, 例えば Cunningham による Logic Life Search [10] や Goucher による ikpx [11] などが知られている。

## 6 まとめと今後の課題

本論文では, ライフゲームにおける物体の合成手法を提案した。その基本的なアイデアは, 既存の物体の断片を並べ, 隙間に適切なパターンを追加することで接合するというものである。断片の間に補完されるパターンが満たさなくてはならない制約は, パターンを構成する各セルの状態を変数にとる論理式として記述可能で, 制約を満たすパターンが入力された際にこの論理式は真を返し, その逆で制約を満たしていないパターンに対して偽を返す。これにより, 物体の合成問題は充足可能性問題に還元され, SAT ソルバを用いて解くことが可能になる。また本研究では SAT ソルバとして Z3py を用いて簡易的な実装を行い, 振動子の合成と移動物体の合成に成功した。

今後の研究としては, 合成により大量に自動生成された物体から形状の規則性を抽出することを試みたいと考えている。同じ周期と移動方向を持つ物体を 2 つ用意したとき, 同一の構造が部分的に共有されつつそれ以外の部分では異なった構造をとっていることがある。そのため, もし物体の形状について規則

性が発見されるなら, それは 1 つの初期値に対して複数の分岐が存在するような規則であると思われる。なぜなら, ある同一の初期値を持つ 2 つの系に対してそのような規則を適用した場合, 序盤の挙動は共通であるが途中から差異が生じ, 最終的には別々の値になるからである。例えばこの規則性は確率的 1 次元セルオートマトンのような形態をとっていて, 乱数で決めた初期値を代入して遷移させるとプロットした結果から移動物体を得られる可能性がある。このような規則性が発見された場合, ただ単に物体の自動生成手法として利用可能なだけでなく, ライフゲームの性質を解明するための手段となることが期待される。

## 参考文献

- [1] Schiff, J. L. *Cellular Automata A Discrete View of the World*. John Wiley & Sons, Inc., 2008.
- [2] Rendell, P. *Turing machine universality of the game of life*. Springer, 2016.
- [3] McIntosh, Harold V. Life's still lifes. In *Game of Life Cellular Automata*, pp. 35–50. Springer, 2010.
- [4] Eppstein, D. gfind 4.9 – search for gliders in semitotalistic cellular automata. URL: <https://www.ics.uci.edu/~epstein/ca/gfind.c>.
- [5] Knuth, D.E. *The Art of Computer Programming, Volume 4, Fascicle 6: Satisfiability*. Addison-Wesley Professional, 2015.
- [6] z3py-tutorial. URL: <https://github.com/ericpony/z3py-tutorial>.
- [7] Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., and Malik, S. Chaff: Engineering an efficient sat solver. In *Proceedings of the 38th annual Design Automation Conference*, pp. 530–535, 2001.
- [8] 井上克巳, 田村直之ほか. Sat ソルバーの基礎 (<特集>最近の sat 技術の発展). 人工知能学会誌, Vol. 25, No. 1, pp. 57–67, 2010.
- [9] 土屋達弘. 充足可能性判定を利用したモデル検査. コンピュータ ソフトウェア, Vol. 29, No. 1, pp. 19–29, 2012.
- [10] Cunningham, O. logic-life-search. URL: <https://github.com/OscarCunningham/logic-life-search>.
- [11] Goucher, A.P. metasat. URL: <https://gitlab.com/apgoucher/metosat>.