A General Result Relating Totalistic Cellular Automata and Self-Referential Sentences

Shuwen Wu
Fudan University
Shanghai, 200433, P. R. China

Ming Hsiung
South China Normal University
Guangzhou, 510631, P. R. China

We investigate the relationship between totalistic cellular automata (TCAs) and self-referential statements. It is well known that cellular automata (CAs) give rise to many undecidability issues related to self-referential statements. The study of self-referential statements and CAs can be traced back to the work of M. Hsiung [1], who established the connection between elementary cellular automata (ECAs) and self-referential paradoxes in terms of their evolution processes. We further deepen the connection between CAs and self-referential sentences. Specifically, we demonstrate the relationship is not only evident in one-dimensional ECAs but also extends to more complex two-dimensional TCAs. We elaborate on commonly applied TCAs, including Moore and von Neumann types. By studying their connection with self-referential sentences, we propose an algorithm for determining the fixed points of these CAs. Then, we classify them based on the (in)stability characteristics observed in their evolutionary processes. Additionally, we discuss certain specific self-referential paradoxes induced by these automata. Finally, we present a general result between TCAs and self-referential sentences.

Keywords: totalistic cellular automata; self-referential sentences; fixed point, paradox

1. Introduction

Cellular automata (CAs) are dynamical systems that are both temporally and spatially discrete, characterized by local interactions and parallel evolution. The concept of CAs originated from von Neumann's proposal of a two-dimensional self-replicating automaton system in his well-known work [2] "The General and Logical Theory of Automata." Just from the title of the paper, it is apparent that there is a close relationship between automata and logic. In fact, the foundation for theoretically building von Neumann's automata is the

https://doi.org/10.25088/ComplexSystems.35.1.39

fixed-point theorem (or recursion theorem) in computability theory. From then on, people have extensively studied the universality and (un)decidability of automata (see, for instance, [3, 4]).

Since there are strong connections between dynamical systems, arithmetic and formal systems, as well as the undecidability of formal systems being closely related to the self-referential sentences, many researchers turn their attention to the correlation between the self-replication mechanism of CAs and the self-referential phenomenon in logic (such as [5]). (Broadly speaking, self-reference is used to denote a statement that refers to itself or its own referent. The self-referential sentence of the present paper refers more to the functional self-reference of the recursion theorem than to the formal linguistic self-reference of the kind used by Gödel. See [6] for more details.) In this respect, a link of CAs with the self-referential statements is established in [1]. The basic idea is that every cellular automaton (CA) can be associated with a set of self-referential statements such that the evolution process of a CA is essentially identical to the revision process of the corresponding self-referential statements. Here, the revision process of self-referential statements, proposed by [7, 8] (see also [9, 10]), belongs to the field of formal theories of truth. This paper further explores and demonstrates that these logical characteristics are not confined to elementary cellular automata (ECAs) but can be generalized to apply to all totalistic cellular automata (TCAs). Like ECAs, TCAs are a type of CA introduced by S. Wolfram (for more details, see [11, 12]). In TCAs, the state transition function depends only on the total sum of the states within a cell's neighborhood.

This paper focuses on two-dimensional TCAs (abbreviated as 2D-TCAs), including Moore-type (M-TCAs) and von Neumann-type (VN-TCAs), with the results further generalized to other two-dimensional CAs as well as higher-dimensional CAs. Additionally, a tree diagram method is introduced to determine the existence of fixed points, along with an algorithm to identify TCAs that possess fixed points. Based on the properties of these fixed points, we provide a discussion and classification and also explore the corresponding close connection between TCAs and self-referential paradoxes.

Standard notations are used in the paper. For example, we will use C, with certain subscripts, to denote a cell of a CA. Accordingly, it also is used to denote a statement (or a sentence). We use T to denote the truth predicate "be true," so that $\ulcorner C \urcorner$ denotes the sentence "C" is true. More notations will be introduced later.

The paper is structured as follows: Section 2 introduces the main subject, TCAs, together with their corresponding logical expressions. Section 3 shows how TCAs are connected with self-referential sentences through revision sequences and provides the definition and determination of fixed points for TCAs. Section 4 presents an effective tree-diagram method for finding fixed points in the evolutionary process of 2D-TCAs. We illustrate and prove this method using the VN-TCAs as a representative case and demonstrate its general applicability to M-TCAs as an example, further giving two classes of CAs that possess fixed points. Section 5 analyzes the paradoxical characteristics of TCAs without fixed points and, based on these paradoxical features, classifies the VN-TCAs and M-TCAs discussed in Section 5. Section 6 extends the analysis to higher-dimensional TCAs and establishes a general result: certain types of higher-dimensional TCAs exhibit the same global behavior at fixed points as 2D-TCAs. Finally, Section 7 concludes the paper.

This paper is an extended version of our conference paper "Totalistic Cellular Automata and Self-Referential Sentences" [13]. The original conference paper was limited to the exploration of the relationship between TCAs and self-referential sentences that we now refer to as "von Neumann-type TCAs" (the "TCA" in the conference paper referred only to the current VN-TCAs). However, we have since discovered that this type of exploration can be generalized to all TCAs, not just limited to the von Neumann type, making it more universal. Because of this, we reconstructed the whole paper: a new section (Section 3) has been added, and the other sections have been rewritten or extended accordingly. We also took this opportunity to correct some errors, such as in Figure 2.

2. Totalistic Cellular Automata and Algebraic Expressions

Since its proposal by von Neumann, CA theory, as a system model characterized by discrete time and space as well as complex behavioral patterns, has gradually developed into a mature theoretical tool. It provides highly useful idealized models for the dynamic behaviors of many real-world complex systems, including physical fluids, neural networks, molecular dynamics systems, natural ecosystems, military command and control networks and economics.

As Wolfram mentioned, CAs can be viewed as parallel-processing computers. Sufficiently complex CAs (e.g., Conway's Game of Life) are "universally computable," capable of calculating any Turing-computable function based on input [14]. Furthermore, according to Church's theoretical framework on the definition of computability, such CAs can simulate any possible system. As a powerful computational and modeling tool, CAs can exhibit complex behaviors emerging from simple rules, offering profound insights into the study of complex systems across various fields. Before exploring their connection with self-referential statements, we will provide a brief introduction to their fundamental principles and classifications.

CAs consist of countless cells arranged in a grid structure, each with its own state value. The state value of each cell evolves deterministically over time based on a set of explicit rules, which take into account the state values of adjacent cells. CAs are typically comprised of five components: cells, cell space, neighborhood, cell state set and evolution rules. Cells are the basic units of CAs, distributed within an n-dimensional grid in cell space. Cell space is a grid structure that can extend infinitely. The neighborhood refers to the collection of surrounding cells that influence the target cell. The cell state set is the set of values observed for the cells. Evolution rules are the mapping rules that determine the next state of a cell.

A standard CA is represented as a triplet $\langle S, r, f \rangle$, where S is a finite set of states, r is the radius of the neighborhood, and f is the iterative function (also called the local function). Based on cell space, CAs can be classified into one-dimensional CAs (ECAs), two-dimensional CAs and multi-dimensional CAs. Among two-dimensional CAs, depending on the neighborhood type, there are Moore, von Neumann and hexagonal types as shown in Figure 1. Additionally, CAs can be categorized by their rules into totalistic rules, non-totalistic rules, cycle rules and so on.

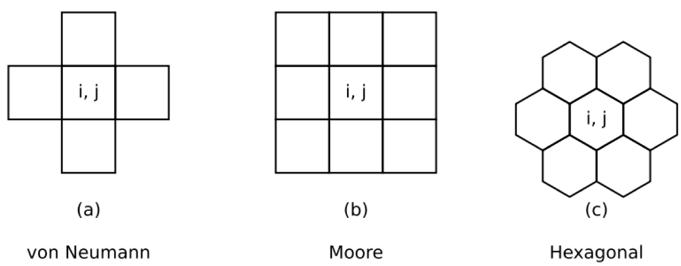

Figure 1. Different types of 2D-TCAs (image source: [15, p. 49: Figure 2.10]).

As mentioned, we will study a special kind of two-dimensional CA. It is named after von Neumann because each cell is attached to its above, below, left and right cells, which are known as the von Neumann neighborhood. (Besides von Neumann neighborhoods, another typical neighborhood in two dimensions is the Moore neighborhood, which has been widely employed in various applications, such as the famous Conway's Game of Life. See [15] for more details.) Moreover, it is "totalistic" because the state (0 or 1) of any cell at a step (in an evolution process) is determined by the total of the states of the cells in a neighborhood at the previous step; we denote this type of CA rule by TCA. And so, the rule of any von Neumann two-dimensional TCA, according to Wolfram [14], can be given by a table like the one shown in Figure 2. (All such rule diagrams are drawn with the computer software Wolfram Mathematica 12 and will not be individually explained.)

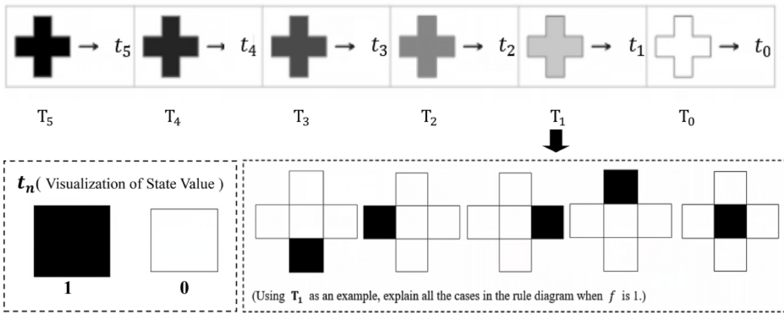

Figure 2. A rule for VN-TCA.

More specifically, we use $C_{i,j}$ to denote the cells where i and j are integers. And so, $C_{i,j+1}$, $C_{i,j-1}$, $C_{i-1,j}$ and $C_{i+1,j}$ are the above, below, left and right neighbors of $C_{i,j}$, respectively (as shown in Figure 3). So the rule in Figure 2 says that for $0 \le k \le 5$, whenever the sum of the values of $C_{i,j}$, $C_{i,j+1}$, $C_{i,j-1}$, $C_{i-1,j}$ and $C_{i+1,j}$ equals i at a given step, the value of $C_{i,j}$ at the next step is t_k. For instance, we show in the square T_1 of Figure 2 that when the sum of the present values of $C_{i,j}$, $C_{i,j+1}$, $C_{i,j-1}$, $C_{i-1,j}$ and $C_{i+1,j}$ is 1, then the value of $C_{i,j}$ at the next step is t_1.

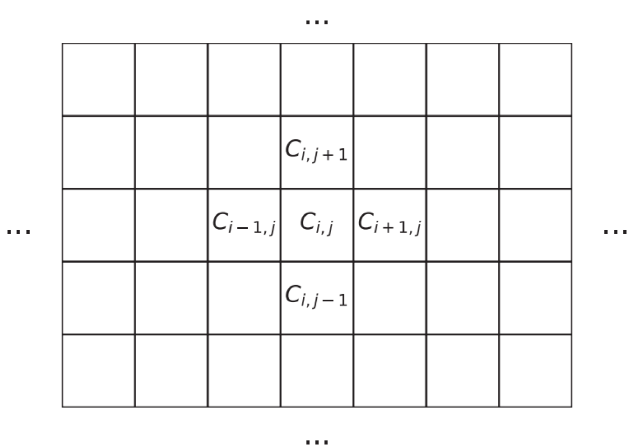

Figure 3. VN-TCA at step t.

Wolfram uses the number $b_5 \cdot 2^5 + b_4 \cdot 2^4 + \cdots + b_0 \cdot 2^0$ to code the rule (in Figure 2). It is called the Wolfram number of this rule. A rule of the Wolfram number n is denoted by R_n, which also denotes the VN-TCA with the Wolfram number n. It is clear that there are 2^6 VN-TCA rules.

Let $C_{i,j}(t)$ be the state of $C_{i,j}$ at step t. Let $n = b_5 \cdot 2^5 + b_4 \cdot 2^4 + \cdots + b_0 \cdot 2^0$. Then R_n can be represented as the algebraic expression:

$$\begin{aligned} &C_{i,j}(t+1) = \\ &\quad b(C_{i,j}(t) + C_{i,j+1}(t) + C_{i,j-1}(t) + C_{i-1,j}(t) + C_{i+1,j}(t)) \end{aligned} \tag{1}$$

where b is a function on $\{i \in \mathbb{N} \mid 0 \le i \le 5\}$ such that $b(i) = b_i$. Similarly, we can derive the corresponding algebraic expression for M-TCAs, as discussed in Section 4.

3. Self-Referential Sentences and Fixed Points

In order to give the self-referential sentences corresponding to 2D-TCAs, we must translate the algebraic expressions of 2D-TCAs into logical expressions. To this end, we first introduce some special normal Boolean formulas. In the following θ, with or without subscripts, is always a Boolean value; that is, θ is either 0 or 1. We stipulate that $\neg^{\theta}\, C$ is $\neg\, C$, if $\theta = 1$; it is C, if $\theta = 0$. Let $C_0, \ldots, C_m$ be the Boolean variables ($m \ge 0$). For $k \le m$, we define

$$\begin{aligned} \beta_{\vee}^{k}\big(C_0, \ldots, C_m\big) &= \bigvee\nolimits_{\theta_0+\cdots+\theta_m=k} \bigwedge\nolimits_{0\le i\le k} \neg^{1-\theta_i}\, C_i \\ \beta_{\wedge}^{k}(C_0, \ldots, C_m) &= \bigwedge\nolimits_{\theta_0+\cdots+\theta_m=k} \bigvee\nolimits_{0\le i\le k} \neg^{\theta_i}\, C_i. \end{aligned}$$

For any function b from $\{k \mid 0 \le k \le m\}$ to $\{0, 1\}$, we define the following two formulas:

$$\tau^{b}(C_0, \ldots, C_m) = b(C_0 + \cdots + C_m) \tag{2}$$

$$\beta_{\vee}^{b}(C_0, \ldots, C_m) = \bigvee\nolimits_{b(k)=1} \beta_{\vee}^{k}(C_0, \ldots, C_m) \tag{3}$$

$$\beta_{\wedge}^{b}(C_0, \ldots, C_m) = \bigwedge\nolimits_{b(k)=0} \beta_{\wedge}^{k}(C_0, \ldots, C_m). \tag{4}$$

Proposition 1. τ^b, $\beta_{\vee}^{b}$ and $\beta_{\wedge}^{b}$ are defined as given earlier. Then $\tau^b = \beta_{\vee}^{b} = \beta_{\wedge}^{b}$.

Proof. Suppose $\{k \mid b(k) = 1\} = \{k_1, \ldots, k_l\}$. It means that

$$\begin{aligned} &\tau^{b}(C_0, \ldots, C_m) = 1, \\ &\text{if and only if } C_0 + \cdots + C_m = k_1, \ldots, \text{or } k_l. \end{aligned} \tag{5}$$

Using Boolean logic, we can easily see that

$$\beta_{\vee}^{k}(C_0, \ldots, C_m) = 1, \text{ if and only if } C_0 + \cdots + C_m = k. \tag{6}$$

Hence, it follows immediately

$$\begin{aligned}&\beta_\vee^b(C_0, \ldots, C_m) = 1,\\ &\text{if and only if } C_0 + \cdots + C_m = k_1, \ldots, \text{ or } k_l.\end{aligned} \tag{7}$$

By equations (5) and (7), we obtain $\tau^b = \beta_\vee^b$. $\tau^b = \beta_\wedge^b$ can be proved dually. □

Equation (1) can be also expressed as:

$$\begin{aligned}&C_{i,j}(t+1) =\\ &\quad \tau^b(C_{i,j}(t), C_{i,j+1}(t), C_{i,j-1}(t), C_{i-1,j}(t), C_{i+1,j}(t)).\end{aligned} \tag{8}$$

We will say that τ^b is the update function, which can also be used to pin down a VN-TCA.

By Proposition 1, the algebraic expression (8) can be equivalently translated into one of the following logical expressions:

$$\begin{aligned}&C_{i,j}(t+1) =\\ &\quad \beta_\vee^b(C_{i,j}(t), C_{i,j+1}(t), C_{i,j-1}(t), C_{i-1,j}(t), C_{i+1,j}(t))\end{aligned} \tag{9}$$

$$\begin{aligned}&C_{i,j}(t+1) =\\ &\quad \beta_\wedge^b(C_{i,j}(t), C_{i,j+1}(t), C_{i,j-1}(t), C_{i-1,j}(t), C_{i+1,j}(t)).\end{aligned} \tag{10}$$

To sum up, the VN-TCA with the coding number $n = b_5 \cdot 2^5 + b_4 \cdot 2^4 + \cdots + b_0 \cdot 2^0$ is the one with the update function τ^b as given in equation (2). Corresponding to one of the 2D-TCAs, we can define a set of sentences, say $\{C_{i,j} \mid i, j \in \mathbb{Z}\}$, in which for any $i, j \in \mathbb{Z}$,

$$\begin{aligned}&C_{i,j} \equiv\\ &\quad \beta_\vee^b(T\ulcorner C_{i,j}\urcorner, T\ulcorner C_{i,j+1}\urcorner, T\ulcorner C_{i,j-1}\urcorner, T\ulcorner C_{i-1,j}\urcorner, T\ulcorner C_{i+1,j}\urcorner).\end{aligned} \tag{11}$$

Informally, $C_{i,j}$ is a sentence that declares the sentences, among $C_{i,j}$ itself, $C_{i,j+1}$, $C_{i,j-1}$, $C_{i-1,j}$ and $C_{i+1,j}$, are true or untrue in some combinatorial way. Note that $C_{i,j}$ is a self-referential sentence in the sense that what $C_{i,j}$ declares is relevant to itself. See [1] for more details.

A different but equivalent formulation of the given set of sentences is to use $\beta_\wedge^b$ instead of $\beta_\vee^b$. In this way, we have associated every 2D-TCA with a set of self-referential sentences.

At the end of this section, we introduce the evolution process for 2D-TCAs and the revision process for self-referential sentences. First, as usual, we define the evolution processes for VN-TCAs by their evolution sequences. We do this in terms of the algebraic expressions of VN-TCAs.

Definition 1. Let R_n be one of the VN-TCAs whose algebraic expression is given by equation (1) or (8), and $C(0)$ is an infinite matrix of Boolean values, whose i, j entry is denoted by $(C_{i,j}(0))_{i,j\geq 0}$. For any (discrete) $t \geq 1$, we define the infinite matrix $(C_{i,j}(t))_{i,j\geq 0}$ of Boolean values by equation (8). The evolution sequence starting from $C(0)$ for R_n is the infinite sequence $C(0), C(1), \ldots, C(t), \ldots$ $(t \geq 0)$.

The revision process we present is a logical tool developed by philosophers H. G. Herzberger and A. Gupta (see [7, 8, 10]) in response to the need to analyze self-referential sentences (and, more generally, circular definitions). To avoid a roundabout introduction, we follow the line set up in [1, pp. 750–751] and straightforwardly define the revision process for the self-referential sentences associated with VN-TCAs. For convenience, we say a function is a *valuation function* if it is a function whose values are the Boolean values.

Definition 2. Let $\{C_{i,j} \mid i, j \in \mathbb{Z}\}$ be a set of sentences given by equation (11) and h_0 be a valuation function on the set $\{C_{i,j} \mid i, j \in \mathbb{Z}\}$. For any $n \geq 1$, we can define a valuation function h_n recursively by:

$$h_{n+1}(C_{i,j}) = \beta_{\vee}^{b}(h_n(C_{i,j}), h_n(C_{i,j+1}), h_n(C_{i,j-1}), h_n(C_{i-1,j}), h_n(C_{i+1,j})). \tag{12}$$

As for $\{C_{i,j} \mid i, j \in \mathbb{Z}\}$, the *revision sequence* starting from h_0 is the sequence of functions $h_0, h_1, \ldots, h_n, \ldots (n \geq 0)$.

Then, we have established the connection between the revision sequence and VN-TCAs. Similarly, we can also construct the connection between M-TCAs and the revision sequence. The evolution sequence and the revision sequence are the main tools used to study the properties of 2D-TCAs and self-referential sentences. We will establish their connection by these two sequences.

This paper provides a classification of 2D-TCAs based on the properties of their fixed points (if any). To this end, we first give the following definition.

Definition 3. Let $\{C_{i,j} \mid i, j \in \mathbb{Z}\}$ be the set of sentences as given in Definition 2 and h_n $(n \geq 0)$ be its revision sequence starting from h_0. If there is a number m such that for any $n \geq m$, $h_m(C_{i,j}) = h_n(C_{i,j})$ holds for any $i, j \in \mathbb{N}$, we say h_n is a fixed point of the given revision sequence.

Under Definition 3, we are able to derive Theorem 1.

Theorem 1. For any 2D-TCA, its evolution sequence has no fixed point, if and only if the (fully developed) branches of its tree diagram are closed.

Proof. This proof is illustrated by taking the R_{31} CA of type VN-TCA as an example. As we proved before, we have the logical formulas that express R_{31} CA:

$$C_{i,j} = \neg T\ulcorner C_{i,j}\urcorner \vee \neg T\ulcorner C_{i,j+1}\urcorner \vee \neg T\ulcorner C_{i,j-1}\urcorner \vee \neg T\ulcorner C_{i-1,j}\urcorner \vee \neg T\ulcorner C_{i+1,j}\urcorner. \tag{13}$$

Suppose h_n is one of the fixed points of R_{31} CA, according to Definition 3; for any $n \geq m$, we have:

$$h_n(C_{i,j}) = h_m(C_{i,j}). \tag{14}$$

Now, we try to find all possible values of $C_{i,j}$ in the two-dimensional space. Through fixing the value of $C_{i,j}$ (0 or 1), we can use the tree diagram to search the states of all possible cells for fixed points according to the rules of R_{31}, as shown in Figure 4.

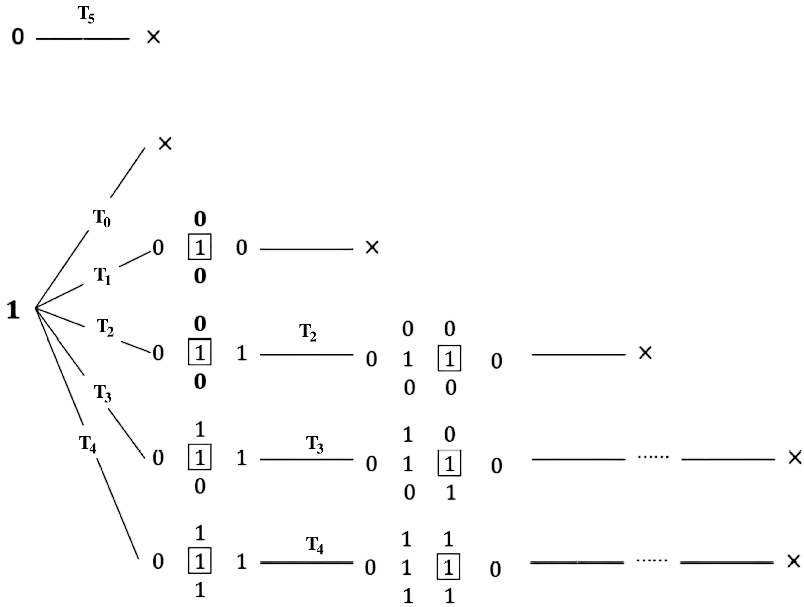

Figure 4. Seeking the fixed point for R_{31}.

In Figure 4, we see that there are no subrules that can maintain the state 0 for $C_{i,j}$, according to the fixed-point property. And if $C_{i,j} = 1$, by the rule of R_{31}, there must be one neighborhood cell that is 0; then there are also no fixed points. Hence, according to Figure 4, whether the value of $C_{i,j}$ is 0 or 1, no branch can obtain a fixed point. That is, Figure 4 illustrates the fact that there are no open branches in the tree diagram of R_{31}. Meanwhile, this means that the R_{31} CA does not have any fixed points. More specifically, R_{31} eventually evolves into a cyclic state in which state 1 turns into state 0, and state 0 turns back into state 1, as shown in Figure 5. Such a cyclic state is precisely characterized by the feature of a paradoxical sentence, which we discuss further in Section 5.

Moreover, we can find that the fixed point of other totalistic rule CAs can also be determined by this method. In this way, we find that only R_{31} and R_1 do not have any fixed points. In what follows, R_1 will be further discussed. □

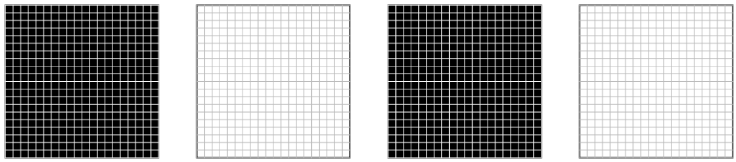

Figure 5. Cyclic state of the R_{31}.

4. Fixed Point of Totalistic Cellular Automata

Definition 3 shows it is not difficult to determine which CAs do not have any fixed points. Now, we look at the complex situation, that is, for those CAs with fixed points, how the fixed points are constructed. To obtain a fixed point according to the properties of totalistic rules, we first define a global condition and then construct the fixed point using the global condition that corresponds to each 2D-TCA:

Definition 4. Define the global condition. If the tree diagram of one 2D-TCA can construct a loop state from 1 to 0, we say the 2D-TCA satisfies the global condition of loop states; if the tree diagram constructs a nested state with 1 and 0, it satisfies the global condition of nested states.

Next, we show how to construct the fixed point according to Definition 4. And then, we find that the fixed point of a 2D-TCA found by this method is not unique.

Proposition 2. If the tree diagram of one 2D-TCA satisfies the global condition, then we can construct the fixed points corresponding to the 2D-TCA.

Proof. We start with a proof for the global condition of loop states. And we give an R_3 as an example to prove. According to the rules of R_3, we have the tree diagram Figure 6.

We can see that Figure 6 is different from Figure 4, in which some branch still continues. And the fixed-point values of 1 (e.g., $C_{i,j} = 1$) that we want can only be obtained by applying it to subrules T_1. Then, the center cell, according to T_1, can obtain the states of cells in its neighborhoods, and they all have a value of 0. We now suppose $C_{i,j} = 0$. Figure 6 shows that the state value 0 can be obtained through the subrules T_4, T_3 and T_2. However, under the subrules T_3 and T_2, the neighborhood cells with a state value 0 cannot be obtained by satisfying any subrules. According to Figure 6, if $C_{i,j} = 1$, the state values of its neighborhood cells must be 0. Hence, this contradicts the situation that holds under T_3 and T_2.

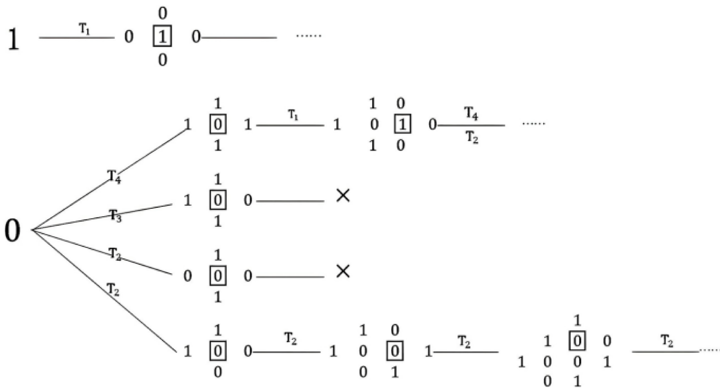

Figure 6. Seeking the fixed point for R_3.

Therefore, if we want to obtain $C_{i,j} = 0$, we have to follow the subrule T_4 or T_2. And under the subrule T_4, the states of the neighboring cells of the objective cell must all be 1. Then, we find that the neighborhood cells get into the situation of $C_{i,j} = 1$. Hence, we say that the CAs satisfy the global condition of loop states.

We now apply the loop states under subrules T_1 and T_4 to obtain the fixed point where state 0 alternates with state 1. That is, if the tree diagram has the loop states, then we at least can obtain a fixed point that satisfies the condition of global states. Finally, we obtain the fixed point of R_3 by applying the global condition of loop states of T_1 through T_4, as shown in Figure 7. Since the objective cell is not fixed, we can construct multiple fixed points of R_3 in this way. □

```
                                                  ......
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1             0 1 0 0 1 0 1 0 1 0 0 1 0 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0             0 1 0 0 1 0 1 0 1 0 0 1 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1             0 1 0 0 1 0 1 0 1 0 0 1 0 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0             0 1 0 0 1 0 1 0 1 0 0 1 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1             0 1 0 0 1 0 1 0 1 0 0 1 0 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0             0 1 0 0 1 0 1 0 1 0 0 1 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1             0 1 0 0 1 0 1 0 1 0 0 1 0 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0             0 1 0 0 1 0 1 0 1 0 0 1 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1             0 1 0 0 1 0 1 0 1 0 0 1 0 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0      ...... 0 1 0 0 1 0 1 0 1 0 0 1 0 0 ......
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1             0 1 0 0 1 0 1 0 1 0 0 1 0 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0             0 1 0 0 1 0 1 0 1 0 0 1 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1             0 1 0 0 1 0 1 0 1 0 0 1 0 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0             0 1 0 0 1 0 1 0 1 0 0 1 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1             0 1 0 0 1 0 1 0 1 0 0 1 0 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0             0 1 0 0 1 0 1 0 1 0 0 1 0 0
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1             0 1 0 0 1 0 1 0 1 0 0 1 0 0
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0             0 1 0 0 1 0 1 0 1 0 0 1 0 0
                                                  ......
```

Figure 7. The fixed points for R_3 (left) and R_8 (right), respectively.

Similarly, we give R_8 CA as an example to prove the global condition of nested states. First, we need to construct a tree diagram of R_8 CA, as shown in Figure 8.

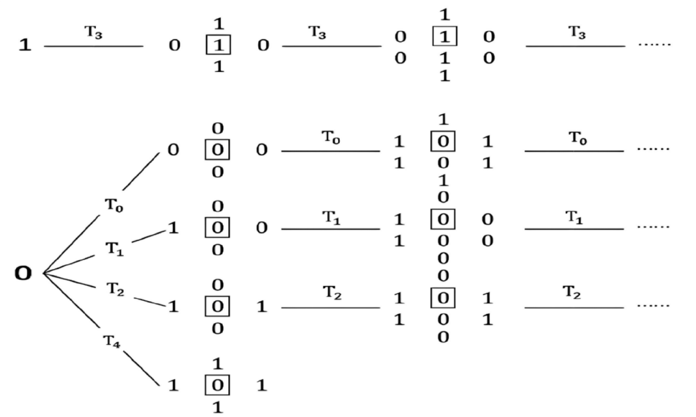

Figure 8. Seeking the fixed point for R_8.

Figure 8 shows that the state values of the tree diagram have an overlapping part. We can use the corresponding nested portion for the overlap part, which is the column of state 0 and the column of state 1 in the tree diagram. We can construct the global states of the fixed point by nesting the part in the overlap states of the tree diagram of R_8, that is, some columns of state 1 and two columns of state 0 arranged freely. Similarly, there is a combination condition $T_3 \to 1$ and $T_2 \to 0$ such that the fixed point can be constructed by some columns of state 1 and one column of state 0 arranged freely. Then, we can use the ideas given to construct a fixed point of R_8, as shown in Figure 7. Note also that according to the global condition of nested states, there are many fixed points in the R_8 CA.

Regarding the applicability of the global condition of loop states, we find that the R_3 CA is not the only one to satisfy the global condition of the loop states of $T_1 \to 1$ and $T_4 \to 0$. Specific automata are summarized in Table 1. Meanwhile, by the properties of the global condition, we can find other loop states with $T_2 \to 1$ and $T_3 \to 0$, as shown in Figure 9. Similarly, for our global condition of nested states, we can find more nested states like $T_4 \to 1$ and $T_1 \to 0$, $T_4 \to 1$ and $T_2 \to 0$. The specific automata are summarized in Table 1.

Similarly, this result can be extended to all kinds of 2D-TCAs, taking the M-TCA as an example. Each exponent corresponds to a cell position, as shown in Figure 10.

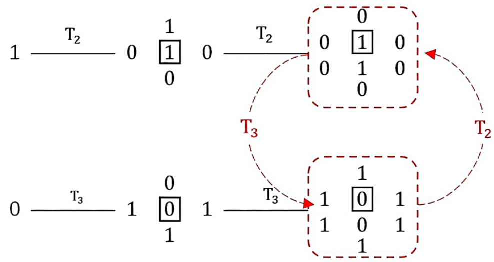

Figure 9. Satisfy the global conditions of loop states.

R_n $T_n = 1$ $T_n = 0$	T_1	T_2	T_3	T_4
T_1	—		8 9 12 13 24 25 28 29 40 41 44 45 56 57 60 61	16 17 20 21 24 25 28 29 48 49 52 53 56 57 60 61
T_2			8 9 10 11 24 25 26 27 40 41 42 43 56 57 58 59	16 17 18 19 24 25 26 27 48 49 50 51 56 57 58 59
T_3		4 5 6 7 20 21 22 23 36 37 38 39 52 53 54 55	—	
T_4	2 3 6 7 10 11 14 15 34 35 38 39 42 43 46 47			—

Table 1. Combination conditions for obtaining fixed points.

According to Section 3, the expression for the M-TCA can be derived as

$$C_{i,j}(t+1) = \tau^b(C_{i,j}(t),\ C_{i,j+1}(t),\ C_{i,j-1}(t),\ C_{i-1,j}(t),\\ C_{i+1,j}(t),\ C_{i-1,j+1}(t),\ C_{i-1,j-1}(t),\ C_{i+1,j+1}(t),\ C_{i+1,j-1}(t)), \tag{15}$$

where b and i correspond to $T_i = b_i$, according to the Wolfram number encoding. An M-TCA rule of the Wolfram number n is denoted by M_n. An M-TCA rule is shown in Figure 11.

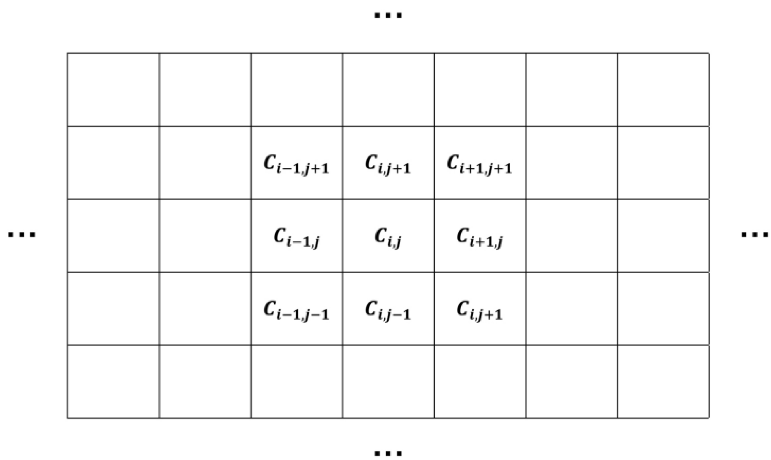

Figure 10. M-TCA at step t.

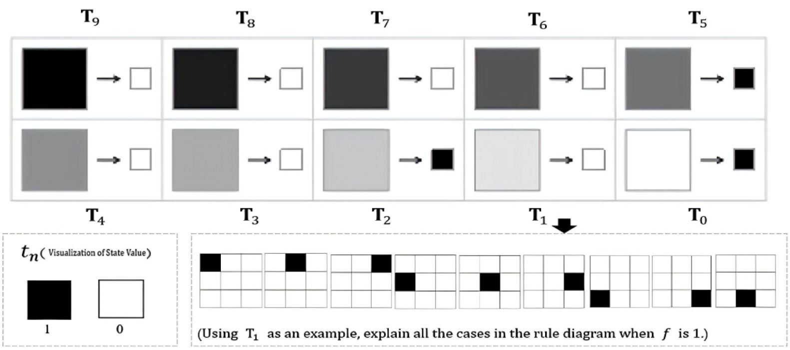

Figure 11. A rule for M-TCA.

According to Definition 4, to find the corresponding fixed points, we need to find the global conditions that are satisfied. Then, we can identify that the loop states of the global conditions can be formed by $T_1 \to 1$ and $T_4 \to 0$, $T_2 \to 0$; similarly, $T_8 \to 0$ and $T_5 \to 1$, $T_7 \to 1$ can also form the loop states of the global conditions. The global conditions that form the nested states are $T_3 \to 1$ and $T_6 \to 0$, $T_6 \to 1$ and $T_3 \to 0$. Therefore, we can also find all M-TCAs with fixed points.

We can observe that there are 2^{10} M-TCAs under the totalistic rule, that is, 1024, which means the situation becomes more diverse and complex. We fixed $T_1 \to 1$ and $T_4 \to 0$, $T_2 \to 0$ and randomly set the remaining condition values, which results in 2^7 M-TCAs. Let these M-TCAs form the set A_1. Similarly, under the loop-state condition $T_8 \to 0$ and $T_5 \to 1$, $T_7 \to 1$, we obtain 2^8 M-TCAs, which form the set A_2. The M-TCAs formed by the global conditions that constitute the nested states, $T_3 \to 1$ and $T_6 \to 0$, $T_6 \to 1$ and $T_3 \to 0$, are denoted as sets A_3 and A_4. At this point, we can obtain M-TCAs with fixed points under four different conditions, with some overlaps. We introduce the principle of tolerance in Theorem 2 and use it to calculate how many M-TCAs there are in the sets A_1 to A_4.

Theorem 2. (Inclusion-exclusion principle. The following result is from [16, p. 145].) For any finite number of sets $A_1, A_2, \ldots, A_n$, the size of their union is given by:

$$\left| \bigcup_{i=1}^{n} A_i \right| = \sum_{1 \le i \le n} |A_i| - \sum_{1 \le i < j \le n} |A_i \cap A_j| + \sum_{1 \le i < j < k \le n} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n+1} |A_1 \cap A_2 \cap \cdots \cap A_n| .$$

Thus, we can obtain $A_1 + A_2 + A_3 + A_4 = 632$. That is, we obtain 632 M-TCAs with fixed points from the global conditions.

Meanwhile, we can observe that when $T_0 \to 0$, there must be fixed points, so we define these sets as A_5. Similarly, under the condition $T_9 \to 1$, the corresponding M-TCA set is defined as A_6. In the same way, based on the tolerance principle, we can derive all M-TCAs with fixed points: $A_1 + A_2 + A_3 + A_4 + A_5 + A_6 = 926$.

As noted earlier, the sets A_1 through A_6 correspond to different subrule conditions. Therefore, we can equivalently present the complementary set of TCAs that do not satisfy these subrule conditions. To sum up, there are 926 M-TCAs with fixed points. Because the number of M-TCAs with fixed points is relatively large, we directly output the complementary set, namely the M-TCAs without fixed points, obtained by using our code. The results are shown in Table 2. The source code is available at github.com/Wen-110/skills-copilot-codespaces-vscode/blob/main/M-TCA.

M-TCA (98)
1, 5, 7, 17, 19, 21, 23, 33, 37, 39, 49, 51, 53, 55, 73, 77, 79, 89, 91, 93, 95, 105, 109, 111, 121, 123, 125, 127, 129, 133, 135, 145, 147, 149, 151, 201, 205, 207,217, 219, 221, 223, 257, 261, 263, 273, 275, 277, 279,289, 293, 295, 305, 307, 309, 311, 329, 333, 335, 345, 347, 349, 351, 361, 365, 367, 377, 379, 381, 383, 385, 389, 391, 401, 403, 405, 407, 417, 421, 423, 433, 435, 437, 439, 457, 461, 463, 473, 475, 477, 479, 489, 493, 495, 505, 507, 509, 511.

Table 2. M-TCAs without fixed points.

5. Paradoxes Associated with Two-Dimensional Totalistic Cellular Automata

In this section, we turn to the self-referential sentences associated with 2D-TCAs. In particular, we pay special attention to those that are paradoxical. Definition 5 is due to Herzberger [8, pp. 483–489] and Gupta [7, pp. 6–14].

Definition 5. Let $\{C_{i,j} \mid i, j \in \mathbb{Z}\}$ be the set of sentences as given in Definition 2. We say it is paradoxical if any of its revision sequences have no fixed point.

Theorem 3 establishes a basic relation between 2D-TCAs and the corresponding self-referential sentences. Its proof is similar to the proof that Hsiung [1] gives for the ECAs. We refer the reader to this literature for details.

Theorem 3. Let $\{C_{i,j} \mid i, j \in \mathbb{Z}\}$ be the set of sentences associated with the 2D-TCAs with Wolfram number n. $\{C_{i,j} \mid i, j \in \mathbb{Z}\}$ is paradoxical, if and only if any evolution sequence for this 2D-TCA has no fixed point.

We know that the evolution sequences for R_1 and R_{31} have no fixed point. So by Theorem 3, the sets of sentences induced from R_1 and R_{31} are paradoxical. Similar results are observed in the 98 M-TCAs, all of which are paradoxical. We can also see that M_1 and M_{511} share the same characteristics as R_1 and R_{31}. We now turn to the definition of paradoxical 2D-TCAs.

First, here is the set of sentences associated with R_1 that is given by:

$$C_{i,j} = \neg T\ulcorner C_{i,j}\urcorner \wedge \neg T\ulcorner C_{i,j+1}\urcorner \wedge \neg T\ulcorner C_{i,j-1}\urcorner \wedge \neg T\ulcorner C_{i-1,j}\urcorner \wedge \neg T\ulcorner C_{i+1,j}\urcorner.$$

Consider $C_{i,j}$. We assume that $C_{i,j}$ is true, which can also be expressed by Boolean number 1. Then, we find $f(C_{i,j}) = 0$. If $f(C_{i,j}) = 0$, we find $\ulcorner C_{i,j}\urcorner = 1$. Contradiction. Meanwhile, if all cell states of R_1 are 0, then in the next state they become 1. Hence the evolution process cycles between 0 and 1. Thus, for a 2D-TCA without a fixed point, if there exists a corresponding paradox with a similar logical structure and periodic behavior, we call such a 2D-TCA a paradoxical CA. That is, according to the properties of automata, we could call R_1 a paradoxical TCA with the liar property. Similarly, we could call the M_1 a paradoxical TCA with the liar paradox property, its equation structure being the same as that of R_1:

$$\begin{aligned} C_{i,j} = {} & \neg T\ulcorner C_{i,j}\urcorner \wedge \neg T\ulcorner C_{i,j+1}\urcorner \wedge \\ & \neg T\ulcorner C_{i,j-1}\urcorner \wedge \neg T\ulcorner C_{i-1,j}\urcorner \wedge \neg T\ulcorner C_{i+1,j}\urcorner \wedge \neg T\ulcorner C_{i-1,j+1}\urcorner \wedge \\ & \neg T\ulcorner C_{i-1,j-1}\urcorner \wedge \neg T\ulcorner C_{i+11,j-1}\urcorner \wedge \neg T\ulcorner C_{i+1,j+1}\urcorner. \end{aligned}$$

Based on the preceding construction, we now discuss what it means for a 2D-TCA to have a correspondence with a specific paradox. We give R_1 as an example. We know that R_{31} also has no fixed points in VN-TCA. That is, there is a paradox property in R_{31}. Similarly, this property is also exhibited in M_{511}. We will find that it is the Curry paradox.

For R_{31}, we note that the corresponding set of self-referential sentences is given by

$$C_{i,j} = \neg\, T\ulcorner C_{i,j}\urcorner \vee \neg\, T\ulcorner C_{i,j+1}\urcorner \vee \neg\, T\ulcorner C_{i,j-1}\urcorner \vee \neg\, T\ulcorner C_{i-1,j}\urcorner \vee \neg\, T\ulcorner C_{i+1,j}\urcorner.$$

So, in some sense, the set of sentences associated with R_{31} is the dual of that with R_1. At the same time, we must point out that the previous equation can be reformulated equivalently as

$$C_{i,j} = T\ulcorner C_{i,j}\urcorner \to \neg\, T\ulcorner C_{i,j+1}\urcorner \vee \neg\, T\ulcorner C_{i,j-1}\urcorner \vee \neg\, T\ulcorner C_{i-1,j}\urcorner \vee \neg\, T\ulcorner C_{i+1,j}\urcorner.$$

We thus can see that the set of sentences associated with R_{31} is something like the Curry paradox. The Curry paradox was proposed by H. B. Curry [17, 18]. A popular version of the Curry paradox is:

$$\text{If the sentence (16) is true, then } C. \tag{16}$$

Similarly, we consider the value of $C_{i,j}$. Assume that $C_{i,j}$ is not true; we find $f(C_{i,j}) = 1$. If $f(C_{i,j}) = 1$, we have $\ulcorner C_{i,j}\urcorner = 0$. Contradiction. Then, we also can find a correspondence paradox—Curry paradox—such that they have similar logical expression and process periodicity. Therefore, R_{31} is a paradoxical CA. The same applies to M_{511}.

So far, we have found different numbers of fixed points from the 2D-TCAs. First, in Table 3 we summarize some results on the VN-TCAs. That is, we give a new classification of the 2^6 VN-TCAs according to the number of fixed points.

We looked at VN-TCAs that do not have any fixed points and found R_1 and R_{31}. Now, there are 2^6 TCAs. We know already that there are VN-TCAs with fixed points under these combining conditions: $T_1 \to 1$ and $T_4 \to 0$, $T_2 \to 1$ and $T_3 \to 0$, $T_3 \to 1$ and $T_1 \to 0$, $T_3 \to 1$ and $T_2 \to 0$, $T_4 \to 1$ and $T_1 \to 0$, $T_4 \to 1$ and $T_2 \to 0$. Meanwhile, there are many fixed points of TCAs under those conditions. Excluding those VN-TCAs, only six VN-TCAs remain, that is, R_0, R_{30}, R_{32}, R_{33}, R_{62} and R_{63}. Obviously, there is a unique fixed point in R_0 and R_{63}, which is 0 and 1, respectively. If VN-TCAs have $T_5 \to 1$ and $T_0 \to 0$, there is at least one fixed point in the VN-TCA, obviously. Hence, R_{30} and R_{33} have at least one fixed point; R_{32} and R_{62} have at least two fixed points. By searching through the tree diagrams of those VN-TCAs, we can prove that R_{30} and R_{33} only have one unique fixed point; R_{32} and R_{62} only have two fixed points, respectively. A list is given in Table 3.

Classification Standard of VN-TCAs	VN-TCA
Without fixed points	R_1, R_{31}
Only one unique fixed point with global state 0	R_0, R_{30}
Only one unique fixed point with global state 1	R_{33}, R_{63}
Only two fixed points with global states 0 and 1	R_{32}, R_{62}
Infinite fixed points	R_2 to R_{29}, R_{34} to R_{62}

Table 3. A classification of VN-TVAs.

In Table 3, the first row shows the VN-TCAs without fixed points, also called paradoxical VN-TCAs. Specifically, R_1 possesses the property of the liar paradox, and R_{31} possesses the Curry paradox. The second row shows those with a unique fixed point; that is, all cell values are 0, such as R_0. The third row shows those with a unique fixed point; that is, all cell values are 1, such as R_{33}. The fourth row shows those with two fixed points; that is, all cell values are 0 or 1, such as R_{32}. The last row shows those with many fixed points, such as R_2.

Due to their large number, a detailed classification of M-TCAs, similar to that of VN-TCAs, is not provided. However, as shown in Table 2, there exist M-TCAs without fixed points, including the liar-paradoxical M1 and the Curry-paradoxical M511. The paradoxical properties of the remaining M-TCAs without fixed points require further investigation. Meanwhile, M-TCAs with fixed points can also be further subdivided into categories, such as those containing only one fixed point or those containing multiple or infinitely many fixed points, but these cases are not listed individually here due to their sheer number. Summarizing these observations, the overall classification is presented in Table 4.

Type	Liar-Paradoxical	Curry-Paradoxical	Other Cases
VN-TCA	R_1	R_{31}	See Table 3 for a detailed classification of the remaining VN-TCA with fixed points.
M-TCA	M_1	M_{511}	See Table 2 for the M-TCA without fixed points. The M-TCAs with fixed points can in principle be subdivided into unique, multiple or infinitely many fixed points, but are not further classified here due to their large number.

Table 4. A classification of 2D-TCAs.

6. Other-Dimensional Totalistic Cellular Automata

In Section 5, we already have a sort of 2D-TCA about the fixed points. It is natural to consider whether one-dimensional or higher-dimensional CAs have the same properties. Hence, we have Proposition 3.

Proposition 3. If a one-dimensional CA is a triple $\langle S, r, f\rangle$ with $S = \{0, 1\}$, $r = 2$ and $f : S^{(2\cdot 2+1)} \to S$, then the classification of its fixed points is the same as that in Table 3.

Proof. A one-dimensional CA is a triple $\langle S, r, f\rangle$ with $S = \{0, 1\}$, $r = 2$ and $f : S^{(2\cdot 2+1)} \to S$. As [1] was mentioned earlier, we can take an automaton as a two-infinite tape, in which the cells are evenly aligned and are naturally indexed by the integers: C_i, $i \in \mathbb{Z}$. And C_i is determined by itself and its four neighbors, including C_{i-1}, C_{i-2} (left neighbor) and C_{i+1}, C_{i+2} (right neighbor). Let f be the update function of the automaton in question. Then we also can compute the state of the cell C_i at step $t+1$ by

$$C_i(t+1) = f(C_{i-1}, C_{i-2}, C_i, C_{i+1}, C_{i+2}),$$

as shown in Figure 12.

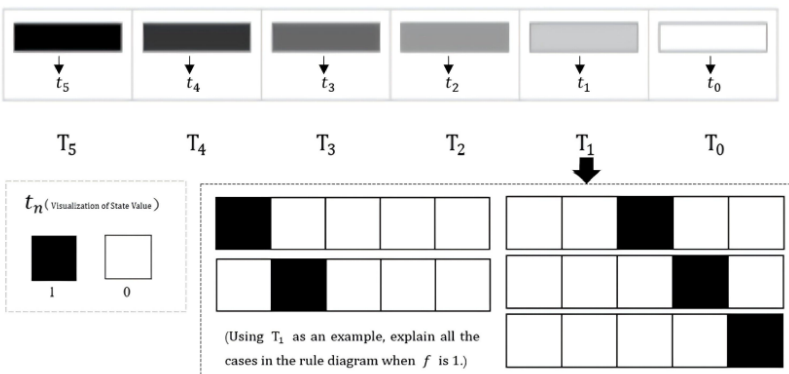

Figure 12. A rule for 1D-TCAs.

Similarly, we use its Wolfram number to code the 1D-TCA and denote it by D_n. Let $n = b_0 \cdot 2^0 + b_1 \cdot 2^1 + \cdots + b_5 \cdot 2^5$. Then D_n can be represented as the algebraic expression:

$$C_i(t+1) = f(C_{i-1} + C_{i-2} + C_i + C_{i+1} + C_{i+2}), \tag{17}$$

where f is a function on $\{i \in \mathbb{N} \mid 0 \leq k \leq 5\}$ such that $f(i) = b_i$.

According to Proposition 1 and Hsiung in [1], we can obtain the logical expression of the 1D-TCA:

$$C_i \equiv \beta_\vee^b(T\lceil C_{i-2}\rceil, T\lceil C_{i-1}\rceil, T\lceil C_i\rceil, T\lceil C_{i+1}\rceil, T\lceil C_{i+2}\rceil). \tag{18}$$

And we can establish the connection between the evolution process of a 1D-TCA with the revision process of the corresponding self-referential sentences and find all of the fixed points in the 1D-TCA.

According to Theorem 3, we can also prove that Proposition 3 is true in 1D-TCAs. Similarly, we use the method by [1] to determine the fixed points in 1D-TCAs. The proof is similar to [1]; we can obtain these results: the evolution sequences with no fixed point are D1 and D31; only one fixed point with the cells' status of 0 is D0 and D30; only one fixed point with the cells' status of 1 is D33 and D63; there are two fixed points with 0 and 1 states, respectively, D32 and D62; others have finite fixed points. See [1] for the details. Therefore, we can see that the classification of the fixed points coincides with that of the 2D-TCAs, as desired. □

Here we see that the properties of self-referential sentences and the fixed point are the same in TCAs whether they are one- or two-dimensional. Hence, we can also use Table 3 to classify 1D-TCAs. Similarly, we can extend the corresponding result to the nine-cell Moore type 2D-TCAs and observe the same fixed-point characteristics in the nine-cell 1D-TCAs. Then, it is natural to have Corollary 1.

Corollary 1. Higher-dimensional TCAs that have five cells—including the objective cell and its four neighbors—show the same global behavior in fixed points.

Since higher-dimensional TCAs have at least an initial state in space, their evolution is more complex. But the evolution process for higher-dimensional CAs is also determined by the same rules, so through their evolution rules, we can simplify them to 2D-TCAs or even 1D-TCAs for corresponding research. As shown in Corollary 1, 1D-TCAs and 2D-TCAs exhibit the same behavior with respect to fixed points, so we can make the same extension. If we set a fixed direction in 3D-TCAs and give an initial state to an objective cell (shown as Figure 13(a)), we can simplify the 3D-TCA to a 2D-TCA, and then we can obtain the same classification as the 2D-TCA. Similarly, if we set a fixed direction in a 3D-TCA and give an initial state to an objective cell (shown as Figure 13(b)), we can simplify the 3D-TCA to 1D-TCA, and then we can obtain the same classification as the 1D-TCA. Therefore, we can see that in the case of 3D-TCAs mentioned earlier, the fixed-point results show the same global behavior with 2D-TCAs and 1D-TCAs. Moreover, Corollary 1 is consistent with the observations of Chate and Mannevile [19], who explored a wide variety of CAs of dimensions four, five and higher. Higher-dimensional situations will be more complex, but we can use the same ideas given to investigate it.

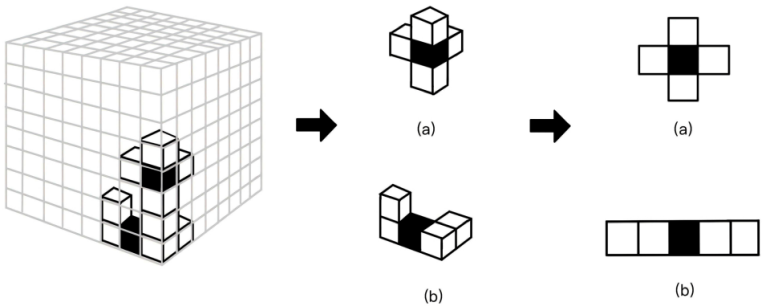

Figure 13. Simplified method for 3D-TCAs.

7. Conclusion

This paper presents a novel approach to connecting totalistic cellular automata (TCAs) with self-referential sentences and classifying TCAs based on their fixed-point properties. As mentioned, cellular automata (CAs) are a popular research topic across various disciplines, and their systematic classification remains a key focus in this field. Taking the von Neumann and Moore types as examples of typical two-dimensional TCAs (2D-TCAs), this paper establishes a connection between these automata and self-referential sentences. By applying a fixed-point determination algorithm, we uncover the fixed-point patterns of these automata and further associate them with self-referential paradoxes.

Moreover, we show one-dimensional TCAs with corresponding cells and two-dimensional TCAs exhibit similar behavioral patterns. We propose a conjecture regarding higher-dimensional TCAs: the complexity of CAs in terms of fixed-point behavior does not increase with dimensionality but rather manifests in different forms. This conclusion aligns with the viewpoint emphasized by Wolfram in *A New Kind of Science*: "Even from very simple programs, behavior of great complexity could emerge" [14, p. 19].

Above all, we find a close connection between CAs and self-referential sentences, which can yield universally applicable results. We utilize the fixed-point properties of CAs for classification and simultaneously reveal potential links between paradoxes and CAs without fixed points. However, several issues remain to be investigated. For example, how does the paradoxical nature of Moore-type CAs without fixed points manifest? Conversely, can paradoxes be leveraged to design new cellular automaton (CA) models? All of these questions need to be further studied. In general, we hope to establish a broad connection and identify structural similarities between CAs and self-reference, thereby promoting further research in both fields.

Acknowledgments

An earlier version of this paper was presented at the 15th National Conference on Modern Logic (Zhuhai, China). We appreciate two anonymous reviewers of that version, and we would also like to thank the anonymous reviewer of the present submission for helpful comments on this paper.

References

[1] M. Hsiung, "Elementary Cellular Automata and Self-Referential Paradoxes," *Journal of Logic and Computation*, **30**(3), 2020 pp. 745–763. doi:10.1093/logcom/exaa022.

[2] J. von Neumann, "The General and Logical Theory of Automata," *Cerebral Mechanisms in Behavior: The Hixon Symposium* (L. A. Jeffress, ed.), New York: Wiley, 1951 pp. 1–41.

[3] M. Cook, "Universality in Elementary Cellular Automata," *Complex Systems*, **15**(1), 2004 pp. 1–40. doi:10.25088/ComplexSystems.15.1.1.

[4] J. Kari, "Decidability and Undecidability in Cellular Automata," *International Journal of General Systems*, **41**(6), 2012 pp. 539–554. doi:10.1080/03081079.2012.695895.

[5] M. Prokopenko, M. Harré, J. Lizier, F. Boschetti, P. Peppas and S. Kauffman, "Self-Referential Basis of Undecidable Dynamics: From the Liar Paradox and the Halting Problem to the Edge of Chaos," *Physics of Life Reviews*, **31**(9), 2019 pp. 134–156. doi:10.1016/j.plrev.2018.12.003.

[6] C. Smorynski, "The Development of Self-Reference: Löb's Theorem," *Perspectives on the History of Mathematical Logic* (T. Drucker, ed.), Boston: Birkhhäuser, 1991 pp. 110–133. doi:10.1007/978-0-8176-4769-8_9.

[7] A. Gupta, "Truth and Paradox," *Journal of Philosophical Logic*, **11**(1), 1982 pp. 1–60. doi:10.1007/BF00302338.

[8] H. G. Herzberger, "Notes on Naive Semantics," *Journal of Philosophical Logic*, **11**(1), 1982 pp. 61–102. doi:10.1007/BF00302339.

[9] N. D. Belnap Jr., "Gupta's Rule of Revision Theory of Truth," *Journal of Philosophical Logic*, **11**(1), 1982 pp. 103–116. doi:10.1007/BF00302340.

[10] A. Gupta and N. D. Belnap, *The Revision Theory of Truth*, Cambridge, MA: MIT Press, 1993.

[11] S. Wolfram, "Statistical Mechanics of Cellular Automata," *Reviews of Modern Physics*, **55**(3), 1983 pp. 601–644. doi:10.1103/RevModPhys.55.601.

[12] S. Wolfram, "Universality and Complexity in Cellular Automata," *Physica D: Nonlinear Phenomena*, **10**(1), 1984 pp. 1–35. doi:10.1016/0167-2789(84)90245-8.

[13] S. Wu and M. Hsiung, "Totalistic Cellular Automata and Self-Referential Sentences," in *Logics for AI and Law: Joint Proceedings of the Third International Workshop on Logics for New-Generation Artificial Intelligence (LNGAI23)* Hangzhou, China (B. Bentzen, B. Liao, D. Liga, R. Markovich, B. Wei, M. Ziong and T. Xu, eds.), London: College Publications, 2023 pp. 21–36.

[14] S. Wolfram, *A New Kind of Science*, Champaign, IL: Wolfram Media, Inc., 2002.

[15] A. Ilachinski, *Cellular Automata: A Discrete Universe*, River Edge, NJ: World Scientific, 2001.

[16] R. A. Brualdi, *Introductory Combinatorics*, 5th ed., Upper Saddle River, NJ: Pearson, 2010.

[17] H. B. Curry, "The Combinatory Foundations of Mathematical Logic," *The Journal of Symbolic Logic*, **7**(2), 1942 pp. 49–64. doi:10.2307/2266302.

[18] H. B. Curry, "The Inconsistency of Certain Formal Logics," *The Journal of Symbolic Logic*, **7**(3), 1942 pp. 115–117. doi:10.2307/2269292.

[19] H. Chaté and P. Manneville, "Collective Behaviors in Spatially Extended Systems with Local Interactions and Synchronous Updating," *Progress of Theoretical Physics*, **87**(1), 1992 pp. 1–60. doi:10.1143/ptp/87.1.1.