Abduction, Unpredictability and Garden of Eden

Chiaki Sakama and Katsumi Inoue

Abstract The notion of unpredictability has been a central theme in both natural and social sciences. In this paper, we first provide a formal account of unpredictability based on abduction. An abductive framework is defined as a pair $\langle B, \mathscr{H} \rangle$ where B is a background theory and $\mathscr{H}$ is a hypothesis space. Then, an event E is *predictable* under $\langle B, \mathscr{H} \rangle$ if there is a hypothesis $H \in \mathscr{H}$ such that $B \wedge H$ implies E. By contrast, an event E is *unpredictable* under $\langle B, \mathscr{H} \rangle$ if it is not predictable under $\langle B, \mathscr{H} \rangle$. We investigate formal properties of (un)predictability of events and argue its computational complexity. Next, we apply the notion of (un)predictability to the problem of identifying patterns in cellular automata (CAs). In CAs it is generally unforeseen whether a particular pattern is produced by a transition rule from the initial configuration. We represent CAs in abductive frameworks and relate the emergence of configurations to the predictability of events from the initial configuration. On the other hand, a configuration that cannot be reached by any initial configuration is called a *Garden of Eden* (GOE). We then characterize a GOE as an unpredictable event in an abductive framework. We show methods of computing CA configurations and checking GOE in logic programming.

1 Introduction

On March 11th in 2011, a most powerful earthquake hit the north-east coast of Japan and triggered a massive tsunami. The quake and the tsunami caused explosions and meltdown of nuclear plants at Fukushima, which has raised great concern about the

Chiaki Sakama
Department of Computer and Communication Sciences, Wakayama University, Sakaedani, Wakayama 640-8510, Japan. e-mail: sakama@sys.wakayama-u.ac.jp

Katsumi Inoue
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan. e-mail: inoue@nii.ac.jp

safety of nuclear power stations all over the world. Japanese experts said that such a powerful earthquake was "unthinkable" and the accident happened at the nuclear plant was "unexpected". The problem of predicting events in the world is one of the challenging issues in both natural and social sciences. In fact, there are many unforeseen and unpredictable events in nature and human lives. In his essay *Black Swan*, Nassim Nicholas Taleb says: "The inability to predict outliers implies the inability to predict the course of history" [37]. The "black swan" tells us that any event in the world is essentially *unpredictable*.

Scientific prediction is made using hypotheses together with knowledge about the world. According to Hempel's *deductive-nomological (D-N) model* [12, 13], given general laws $L_1,\ldots,L_m$ and an *explanandum* phenomenon E to be explained, particular facts $C_1,\ldots,C_n$ form *explanans* if $C_1,\ldots,C_n$, together with $L_1,\ldots,L_m$, deduce E. In this case, $C_1,\ldots,C_n$ are called a *D-N explanation* of E. On the other hand, explanans are also used for predicting an explanandum event under the laws. That is, if we have $C_1,\ldots,C_n$ before the occurrence of E, then $C_1,\ldots,C_n$ are used for predicting E under $L_1,\ldots,L_m$. "Thus, the logical structure of a scientific prediction is the same as that of a scientific explanation,... The customary distinction between explanation and prediction rests mainly on a pragmatical difference between the two" [12]. Explanation and prediction are also related to each other in Hempel's *hypothetico-deductive (H-D) model of confirmation* [15]. In H-D model, predicted events are tested against reality for estimating credibility of that explanation. In this way, scientific inquiry is constituted by three stages: *hypothesis generation (explanation)*, *prediction* and *confirmation*.

In first-order logic, explanations and predictions are formulated as follows. Let B be a consistent first-order formula which represents general laws and true facts (called a *background theory*). Given an observation O as a consistent formula, a formula H is an *explanation* of O if it satisfies

$$B \wedge H \models O$$

where $B \wedge H$ is consistent [24]. In this case, we say that a *hypothesis* H is *abduced* from O under B. By contrast, given a background theory B, a formula E is called a *prediction* by H if it satisfies

$$B \wedge H \models E$$

where $B \wedge H$ is consistent. Based on this schema, the notions of predictability and unpredictability of events are defined as follows.

- An event E is *predictable* under a background theory B if there is a hypothesis H such that $B \wedge H \models E$ and $B \wedge H$ is consistent;
- An event E is *unpredictable* under a background theory B if it is not predictable by any H under B.

Using this intuition, we first provide a formal account of (un)predictability. We introduce an *abductive framework* based on first-order logic and formulate the notion of (un)predictability based on this framework.

The notion of unpredictability is also referred to as a unique feature of *complex systems*. In complex systems their behaviors are generally unpredictable because of their sensitivity to initial conditions and the existence of unknown cause-effect dynamics. On the other hand, complex behaviors do not always imply the complexity of the system. In his book *A New Kind of Science* [39], Stephen Wolfram says: "For our everyday experience has led us to expect that an object that looks complicated must have been constructed in a complicated way. ... But ... at least sometimes such an assumption can be completely wrong." He observes this phenomenon in *cellular automata* (CAs) [38] in which complex behaviors of a system emerge from a simple initial condition and a simple transition rule. He then says that "it is this basic phenomenon that is ultimately responsible for most of the complexity that we see in nature". To understand (un)predictability in CAs, we apply our logical framework of (un)predictability to the problem of identifying patterns in cellular automata. We show that a CA configuration is (logically) predictable if it is deduced by an initial configuration using some transition rule. By contrast, a CA configuration is (logically) unpredictable if it is not deducible by any initial configuration. Such a CA configuration is called a *Garden of Eden (GOE)*, which is a configuration having no predecessors. Using the correspondence, we provide a method of computing CA configurations and verifying GOEs in *logic programming*.

The rest of this paper is organized as follows. Section 2 provides a logical framework of (un)predictability and investigates formal properties. Section 3 applies (un)predictability to cellular automata, and Section 4 shows its computation in logic programming. Section 5 discusses related issues and Section 6 summarizes the paper.

2 Formal Account of (Un)predictability

In this section, a background theory B, an observation O, a hypothesis H and an event E are all (consistent) first-order formulas. The symbol $\models$ represents the entailment relation in first-order logic. As usual, a set of formulas is identified with the conjunction of those formulas.

2.1 Abductive Framework

An *abductive framework* is a pair $\langle B, \mathscr{H} \rangle$ where B is a background theory and $\mathscr{H}$ is a set of formulas representing possible hypotheses.[1] Each element in $\mathscr{H}$ is often

[1] In this paper, we do not specify how $\mathscr{H}$ is constructed. For one case, each formula in $\mathscr{H}$ is extracted from B as a possible explanation of observations. For another case, $\mathscr{H}$ is specified as a hypothesis language and H is any formula constructed from $\mathscr{H}$. In any case, we simply denote as $H \in \mathscr{H}$ when H is a formula belonging to $\mathscr{H}$.

called an *abducible* [24]. Given an abductive framework $\langle B, \mathscr{H} \rangle$ and an observation O, a hypothesis $H \in \mathscr{H}$ is an *explanation* of O under $\langle B, \mathscr{H} \rangle$ if

1. $B \wedge H \models O$,
2. $B \wedge H$ is consistent.

In this case, O is *explained* by H under $\langle B, \mathscr{H} \rangle$. We also say that O is *explainable* under $\langle B, \mathscr{H} \rangle$ iff O is explained by some H under $\langle B, \mathscr{H} \rangle$.

There are some literature which assume additional conditions on explanations such that (i) $B \not\models O$ which states that O is a new event that is not deduced from B or (ii) $H \not\models O$ which states that O is not deduced by H alone [10, 13]. We do not employ the condition (i) because under the condition O is not explainable whenever $B \models O$. In this case, however, we could say that O is explained without introducing any hypothesis. We do not employ the condition (ii) because under the condition O is not explainable whenever $H \models O$. In this case, however, we could say that O is explained without the help of a background theory. For instance, given some cases O, *inductive generalization* [30] constructs a general rule H that governs those cases. With these reasons, we do not impose those additional conditions and call any H that satisfies the above two conditions an explanation.

Prediction and (un)predictability are defined as follows.

Definition 1 (prediction, (un)predictable). Given an abductive framework $\langle B, \mathscr{H} \rangle$, an event E is *predictable* by a hypothesis $H \in \mathscr{H}$ if

1. $B \wedge H \models E$,
2. $B \wedge H$ is consistent.

In this case, an event E is called a *prediction* by H. An event E is *predictable under* $\langle B, \mathscr{H} \rangle$ if there is $H \in \mathscr{H}$ such that E is predictable by H. An event E is *unpredictable under* $\langle B, \mathscr{H} \rangle$ if there is no $H \in \mathscr{H}$ such that E is predictable by H.

By the definition, every tautological sentence is predictable and every contradictory sentence is unpredictable. A prediction is often made by a hypothesis which is abduced by an observation. As addressed in the introduction, the logical structure of prediction is the same as that of explanation, so that predictability and explainability are logically equivalent.[2]

Proposition 1. *Let $\langle B, \mathscr{H} \rangle$ be an abductive framework. Then, an event E is predictable under $\langle B, \mathscr{H} \rangle$ iff E is explainable under $\langle B, \mathscr{H} \rangle$.*

Proof. The result holds by definition. □

Example 1. Let $\langle B, \mathscr{H} \rangle$ be the abductive framework such that

$$B = \{\, p \to r, \quad q \to \neg r, \quad s \to t \,\},$$
$$\mathscr{H} = \{\, p, \quad q \,\}.$$

[2] In this paper, we are concerned with prediction from the logical point of view and take a standpoint close to the D-N model. This equivalence does not hold under the *inductive-statistical (I-S) model* [14].

Then p, q, r and $\neg r$ are predictable under $\langle B, \mathscr{H} \rangle$, while s and t are unpredictable.

In Example 1, s and t become predictable if $\mathscr{H}$ is expanded to include s. Thus, unpredictability is due to the incompleteness of hypotheses. In other words, if $\mathscr{H}$ contains all formulas, every event E is predictable as far as it is consistent with a background theory.

Proposition 2. *Let $\langle B, \mathscr{F} \rangle$ be an abductive framework where $\mathscr{F}$ is the set of all formulas. Then, any event E is predictable under $\langle B, \mathscr{F} \rangle$ iff $B \wedge E$ is consistent.*[3]

Proof. If any event E is predictable under $\langle B, \mathscr{F} \rangle$, then $B \wedge H \models E$ and $B \wedge H$ is consistent. Hence, $B \wedge E$ is consistent. Conversely, if $B \wedge E$ is consistent, then $B \wedge E \models E$ implies that any E is predictable. □

Predictability in the case of Proposition 2 does not fully capture our understanding of predictability because we often fail to predict some events not because they are inconsistent with our background knowledge but because they are just unexpected. In fact, the above proposition characterizes the situation that an event E is predictable by E itself under B. This explains the reason for introducing an abductive framework $\langle B, \mathscr{H} \rangle$ and restricting possible hypotheses to the elements of $\mathscr{H}$. In practice, the hypothesis space $\mathscr{H}$ is expanded to explain new observations, and is reduced to eliminate falsified predictions.

There is a case in which an event E and its negation $\neg E$ are both predictable under $\langle B, \mathscr{H} \rangle$. In Example 1, both r and $\neg r$ are predictable by different hypotheses. The next proposition states a necessary condition on hypotheses which may predict conflicting events.

Proposition 3. *Let $\langle B, \mathscr{H} \rangle$ be an abductive framework. If an event E is predictable by H under $\langle B, \mathscr{H} \rangle$ and an event $\neg E$ is predictable by H' under $\langle B, \mathscr{H} \rangle$, then $B \wedge H \wedge H'$ is inconsistent.*

Proof. Suppose that E is predictable by H and $\neg E$ is predictable by H'. Then, $B \wedge H \models E$ where $B \wedge H$ is consistent, and $B \wedge H' \models \neg E$ where $B \wedge H'$ is consistent. If $B \wedge H \wedge H'$ is consistent, $B \wedge H \wedge H' \models E \wedge \neg E$, contradiction. □

A sufficient condition for the unpredictability of an event is given below.

Proposition 4. *Let $\langle B, \mathscr{H} \rangle$ be an abductive framework. If $B \models \neg E$, then an event E is unpredictable under $\langle B, \mathscr{H} \rangle$.*

Proof. If $B \models \neg E$, then $B \wedge H \models \neg E$ for any $H \in \mathscr{H}$. Hence, there is no H such that $B \wedge H \models E$ and $B \wedge H$ is consistent. □

When two different hypotheses individually predict two different events, those hypotheses jointly predict the two events as far as the joint hypothesis is consistent with a background theory.

[3] A similar result is shown in [21] in the context of explainability of observations.

Proposition 5. *Let $\langle B, \mathscr{H} \rangle$ be an abductive framework. Suppose that an event E is predictable by H under $\langle B, \mathscr{H} \rangle$ and an event E' is predictable by H' under $\langle B, \mathscr{H} \rangle$. Then, $E \wedge E'$ is predictable by $H \wedge H'$ under $\langle B, \mathscr{H} \rangle$ iff $B \wedge H \wedge H'$ is consistent.*

Proof. When E is predictable by H and E' is predictable by H' under $\langle B, \mathscr{H} \rangle$, it holds that $B \wedge H \models E$ where $B \wedge H$ is consistent, and $B \wedge H' \models E'$ where $B \wedge H'$ is consistent. On the other hand, if $E \wedge E'$ is predictable by $H \wedge H'$ under $\langle B, \mathscr{H} \rangle$, then $B \wedge H \wedge H' \models E \wedge E$ and $B \wedge H \wedge H'$ is consistent. Conversely, if $B \wedge H \wedge H'$ is consistent, then $B \wedge H \wedge H' \models E \wedge E$. □

When two different hypotheses predict the same event, the conjunction of those hypotheses does not always predict the event.

Example 2. Let $\langle B, \mathscr{H} \rangle$ be the abductive framework such that

$$B = \{ p \rightarrow r, \quad q \rightarrow \neg p, \quad q \rightarrow r \},$$
$$\mathscr{H} = \{ p, \quad q \}.$$

Then, r is predictable by both p and q under $\langle B, \mathscr{H} \rangle$, while r is unpredictable by $p \wedge q$ under $\langle B, \mathscr{H} \rangle$.

A *propositional* abductive framework $\langle B, \mathscr{H} \rangle$ is an abductive framework in which both B and $\mathscr{H}$ are (classical) propositional theories. In a propositional abductive framework, it is known that the computational complexity of deciding whether an observation has an explanation is Σ_P^2-complete [7]. By this fact, the following result holds.

Proposition 6. *In a propositional abductive framework, deciding whether an event is (un)predictable under $\langle B, \mathscr{H} \rangle$ is Σ_P^2-complete.*

Proof. An event E is predictable under $\langle B, \mathscr{H} \rangle$ iff E is explainable under $\langle B, \mathscr{H} \rangle$ (Proposition 1). Hence, the result holds by the complexity result in [7]. □

In a non-propositional abductive framework $\langle B, \mathscr{H} \rangle$, it is generally undecidable whether an event is predictable or not under $\langle B, \mathscr{H} \rangle$.

2.2 Comparing (Un)predictability

Given two different abductive frameworks, the notion of *explainable equivalence* is introduced in [18]. Formally, two abductive frameworks $\langle B_1, \mathscr{H}_1 \rangle$ and $\langle B_2, \mathscr{H}_2 \rangle$ are *explainably equivalent* if, for any observation O, O is explainable under $\langle B_1, \mathscr{H}_1 \rangle$ iff O is explainable under $\langle B_2, \mathscr{H}_2 \rangle$. We apply the notion to comparing (un)predictability of events in two different abductive frameworks.

Definition 2. Two abductive frameworks $\langle B_1, \mathscr{H}_1 \rangle$ and $\langle B_2, \mathscr{H}_2 \rangle$ are *predictably equivalent* if, for any event E, E is predictable under $\langle B_1, \mathscr{H}_1 \rangle$ iff E is predictable under $\langle B_2, \mathscr{H}_2 \rangle$.

Clearly, if $\langle B_1, \mathscr{H}_1 \rangle$ and $\langle B_2, \mathscr{H}_2 \rangle$ are predictably equivalent, any event E is unpredictable under $\langle B_1, \mathscr{H}_1 \rangle$ iff E is unpredictable under $\langle B_2, \mathscr{H}_2 \rangle$.

Proposition 7. *Two abductive frameworks $\langle B_1, \mathscr{H}_1 \rangle$ and $\langle B_2, \mathscr{H}_2 \rangle$ are explainably equivalent iff they are predictably equivalent.*

Proof. The result holds by Proposition 1. □

When two abductive frameworks have the common hypotheses, logical equivalence of two background theories is a sufficient condition for explainable equivalence of the two frameworks [18]. The fact is rephrased in our context as follows.

Proposition 8. *If $B_1 \equiv B_2$, then two abductive frameworks $\langle B_1, \mathscr{H} \rangle$ and $\langle B_2, \mathscr{H} \rangle$ are predictably equivalent.*

Proof. The result holds by Proposition 7 and the result in [18]. □

The converse of Proposition 8 does not hold in general.

Example 3. Let $\langle B_1, \mathscr{H} \rangle$ and $\langle B_2, \mathscr{H} \rangle$ be two abductive frameworks such that

$$\begin{aligned} B_1 &= \{\, p \to r \,\}, \\ B_2 &= \{\, q \to r \,\}, \\ \mathscr{H} &= \{\, p, \ \ q \,\}. \end{aligned}$$

Then, $\langle B_1, \mathscr{H} \rangle$ and $\langle B_2, \mathscr{H} \rangle$ are predictably equivalent, but $B_1 \not\equiv B_2$.

The result implies that when two different agents predict exactly the same events, the fact does not necessarily imply that they have the common background knowledge. On the other hand, when two different agents have the common background knowledge, an agent who has more hypotheses can predict more events.

Proposition 9. *Let $\langle B, \mathscr{H}_1 \rangle$ and $\langle B, \mathscr{H}_2 \rangle$ be two abductive frameworks. If $\mathscr{H}_1 \subseteq \mathscr{H}_2$, then any event predictable under $\langle B, \mathscr{H}_1 \rangle$ is also predictable under $\langle B, \mathscr{H}_2 \rangle$.*

Proof. If $\mathscr{H}_1 \subseteq \mathscr{H}_2$, then any observation explainable under $\langle B, \mathscr{H}_1 \rangle$ is also explainable under $\langle B, \mathscr{H}_2 \rangle$ [19]. Hence, the result holds. □

On the other hand, given two abductive frameworks $\langle B_1, \mathscr{H} \rangle$ and $\langle B_2, \mathscr{H} \rangle$, the relation $B_1 \subseteq B_2$ does not imply that any event predictable under $\langle B_1, \mathscr{H} \rangle$ is also predictable under $\langle B_2, \mathscr{H} \rangle$.

Example 4. Let $\langle B_1, \mathscr{H} \rangle$ and $\langle B_2, \mathscr{H} \rangle$ be two abductive frameworks such that

$$\begin{aligned} B_1 &= \{\, p \wedge q \to r \,\}, \\ B_2 &= \{\, p \wedge q \to r, \quad \neg p \vee \neg q \,\}, \\ \mathscr{H} &= \{\, p, \ \ q \,\}. \end{aligned}$$

Then, r is predictable under $\langle B_1, \mathscr{H} \rangle$ but is unpredictable under $\langle B_2, \mathscr{H} \rangle$.

As such, predictability of events is *semi-monotonic* [19] with respect to the increase of information in an abductive framework.

3 Unpredictability and Garden of Eden

(Un)predictability is often argued in the context to *complex systems*. In those systems, large-scale complex and organized phenomena may emerge from local interactions among individuals. Dynamics of a system are complex in its nature, and the behavior is generally unpredictable. In this section, we consider *cellular automata* (CAs) as a simple model of complex systems, and argue possible connections between (un)predictability in CAs and the one in abductive frameworks.

3.1 Cellular Automata

Cellular automata [38] are mathematical idealizations of physical systems in which space and time are discrete, and physical quantities take on a finite set of discrete values. Because of their simple mathematical constructs and distinguished features, cellular automata have been used for modelling advanced computation such as massively parallel computers and evolutionary computation, and also used for simulating various complex systems in the world.

A cellular automaton (CA) consists of a regular grid of *cells*, each of which has a finite number of possible *states*. The state of each cell changes synchronously in discrete time steps (or *generations*) according to a local and identical *transition rule*.[4] The state of a cell in the next time step is determined by its current state and the states of its surrounding cells (called *neighbors*). The collection of all cellular states in the grid at some time step is called a *configuration*. An *elementary* CA consists of a one-dimensional array of (possibly infinite) cells, and each cell has one of two possible states 0 or 1. A cell and its two adjacent cells form a neighbor of three cells, so there are $2^3 = 8$ possible patterns for neighbors. A transition rule describes for each pattern of a neighbor, whether the central cell will be 0 or 1 at the next time step. Then, $2^8 = 256$ possible rules are considered and 256 elementary CAs are defined accordingly. Stephen Wolfram gave each rule a number 0 to 255 (called the *Wolfram code*), and analyzed their properties [38]. The evolution of an elementary CA is illustrated by starting with the initial configuration in the first row, the configuration at the next time step in the second row, and so on. Figure 1 shows the rule 30 and one of its evolution where the black cell represents the state 1 and the white cell represents the state 0. The figure shows the first 16 generations of the rule 30 starting with a single black cell. It is known that the rule 30 displays aperiodic and random patterns in a chaotic manner.

Another well-known rule 110 displays patterns between stability and chaos. The rule 110 is also known as one of the simplest CAs that can simulate a universal Turing machine. In elementary CAs, evolving patterns depend strongly on the initial configuration and a rule used. An important feature of elementary CAs is their *irreversibility*. A transition rule may transform several different configurations into

[4] There are asynchronous CAs [38] while we consider synchronous ones.

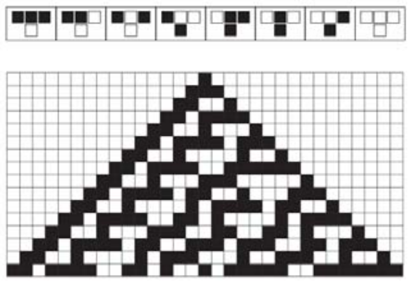

Fig. 1 Evolution of patterns by the rule 30

the same configuration. Thus, a particular configuration has the unique descendant, but does not have the unique ancestor in general. In other words, a transition rule is not a surjective function in general. This fact implies that there is a configuration which has no predecessor under a transition rule. Such a configuration is discussed in the next subsection.

For two-dimensional CAs, the most widely known CA is the *Game of Life* (or *Life*) by John H. Conway [6]. The game has the universe of an infinite two-dimensional grid of square cells, each of which has one of two possible states, 1 (live) or 0 (dead). Each cell interacts with its eight surrounding cells and changes its state according to the following transition rule:

- Any live cell with two or three live neighbors stays alive at the next time step. Otherwise, the cell dies of loneliness or overcrowding at the next time step.
- Any dead cell with exactly three live neighbors will become alive.

Depending on the initial configurations, different patterns evolve by applying the transition rule repeatedly at each time step. Figure 2 illustrates a pattern called a *glider*, where the pattern at $t = 0$ moves across the grid and appears at $t = 4$.

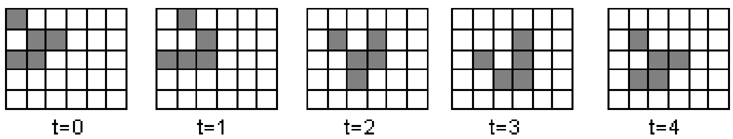

Fig. 2 Glider in *Life*

The *Life* has the power of a universal Turing machine and provides an example of emergence and self-organization. Note that the transition rule of *Life* is not surjective, so it is irreversible.

3.2 Garden of Eden

Both Wolfram's rule 30 and the *Life* have simple transition rules, yet produce unpredictable patterns. It is generally unforeseen whether a particular pattern is produced from the initial configuration unless doing a state by state computation. To understand unpredictability in CAs, we relate CAs to the abductive framework of Section 2. To this end, we represent a transition rule as logical formulas in a background theory B. Every CA configuration is represented as a formula, and all initial configurations are put in a hypotheses space $\mathcal{H}$. With this setting, a configuration E is produced from $H \in \mathcal{H}$ using B iff $B \wedge H \models E$ where $B \wedge H$ is consistent. Then, one can see whether a particular pattern will be produced from the initial configuration by checking if it is predictable under $\langle B, \mathcal{H} \rangle$.

Proposition 10. *Let B be a background theory specifying a transition rule of a CA, and $\mathcal{H}$ be the set of all initial configurations. Then, a configuration E is produced from an initial configuration H using a transition rule of B iff E is predictable by $H \in \mathcal{H}$ under $\langle B, \mathcal{H} \rangle$.*

Proof. A configuration E is produced from H using B iff $B \wedge H \models E$ where $B \wedge H$ is consistent. Then, the result holds by Definition 1. □

Example 5. In one-dimensional CAs, a configuration at a time step t is represented by a (possibly infinite) sequence of cells $\langle \cdots x^t_{i-1} x^t_i x^t_{i+1} \cdots \rangle$ where x^t_i is a cell at a time step t. Wolfram's rule 30 is then represented by $B = \{ x^{t+1}_i \leftrightarrow x^t_{i-1} \oplus (x^t_i \vee x^t_{i+1}) \}$ where $\oplus$ represents exclusive disjunction (Table 1).

Table 1 Wolfram's rule 30

x^t_{i-1}	x^t_i	x^t_{i+1}	x^{t+1}_i
1	1	1	0
1	1	0	0
1	0	1	0
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

Given H as the initial configuration $\langle \cdots x^0_{i-1} x^0_i x^0_{i+1} \cdots \rangle = \langle \cdots 010 \cdots \rangle$[5], the configuration E at the time step 4: $\langle \cdots x^4_{i-3} x^4_{i-2} x^4_{i-1} x^4_i x^4_{i+1} x^4_{i+2} x^4_{i+3} \cdots \rangle = \langle \cdots 1101111 \cdots \rangle$ is predictable by H under $\langle B, \mathcal{H} \rangle$ (cf. Figure 1).

Our next question is whether there is any configuration that is unpredictable under $\langle B, \mathcal{H} \rangle$. To answer this question, we introduce specific patterns in CAs. A configuration that cannot be reached by any initial configuration is called a *Garden of*

[5] This is an abbreviation of the formula $(\cdots \wedge x^0_{i-1} \equiv 0 \wedge x^0_i \equiv 1 \wedge x^0_{i+1} \equiv 0 \wedge \cdots)$.

Eden (GOE).[6] A CA contains a GOE if and only if its transition rule is not a surjective function. An *orphan* is a GOE that can be extended in any way to form other GOEs. One of the open questions is the existence of a GOE in CAs. In the elementary CAs, for instance, the rule 30 has no GOE while the rule 110 has some GOEs.[7] There is an algorithm for deciding whether an arbitrary one-dimensional CA has GOEs [2]. In two-dimensional CAs, on the other hand, the question is whether there is a GOE in an arbitrary grid size. An extensive search for a GOE by examining all possible initial configurations is expensive even in relatively small grid sizes. In the *Life*, the smallest currently known orphan (or GOE), discovered by researchers of the TU Delft in December 2011 [9], contains 56 living cells and fits within a 10×10 grid (Figure 3, black cells are alive, white ones are dead, and gray ones are either alive or dead). It is also known that there are no GOE that are 6×6 cells wide or smaller [9].

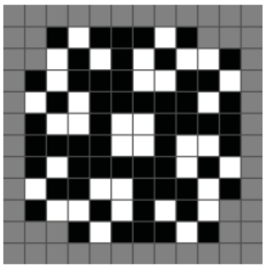

Fig. 3 A Garden of Eden within a 10×10 grid

From the viewpoint of abduction, a GOE is considered a configuration that is unpredictable by any predecessor under a transition rule. A GOE is related to unpredictability as follows.

Proposition 11. *Let B be a background theory specifying a transition rule of a CA, and $\mathscr{H}$ is the set of possible initial configurations. Then, a configuration E is a GOE iff E is unpredictable under $\langle B, \mathscr{H} \rangle$.*

Proof. The result holds by Proposition 10 □

Propositions 10 and 11 provide connections between (un)predictablity in formal logic and the corresponding notion in CAs. Moreover, they show the possibility of computing CA configurations and GOEs in terms of abduction. In the next section, we argue computational methods of CAs using logic programming.

[6] A GOE is created only if it is given as the initial configuration, hence it is named after the concept in the Bible.

[7] In fact, the rule 30 is a surjective function [27], while the rule 110 is not.

4 Computing CA in Logic Programming

In Section 3, we formulate (un)predictability of CA configurations using the abductive framework in first-order logic. In this section, we show a method of computing CAs and GOEs in logic programming.

4.1 Computing Elementary CA

A *(logic) program* P consists of *rules* of the form:

$$A \leftarrow B_1, \ldots, B_m, \mathit{not}\, B_{m+1}, \ldots, \mathit{not}\, B_n$$

where $A, B_1, \ldots, B_n$ are atoms and *not* is *negation as failure*. The left-hand side of the rule is the *head*, and the right-hand side is the *body*. Let $\mathscr{B}$ be the Herbrand base of a program P, and $ground(P)$ the set of ground instances of all rules in a program P. An (Herbrand) interpretation $I \subseteq \mathscr{B}$ *satisfies* a ground rule $A \leftarrow B_1, \ldots, B_m, \mathit{not}\, B_{m+1}, \ldots, \mathit{not}\, B_n$ if $\{B_1, \ldots, B_m\} \subseteq I$ and $\{B_{m+1}, \ldots, B_n\} \cap I = \emptyset$ imply $A \in I$. An interpretation I is an *(Herbrand) model* of a program P if I satisfies every rule in $ground(P)$. A model I is a *supported model* [3] of a program P if for any ground atom $A \in I$, there is a ground rule $A \leftarrow B_1, \ldots, B_m, \mathit{not}\, B_{m+1}, \ldots, \mathit{not}\, B_n$ in $ground(P)$ such that $\{B_1, \ldots, B_m\} \subseteq I$ and $\{B_{m+1}, \ldots, B_n\} \cap I = \emptyset$. A program having at least one supported model is *consistent*. When a ground atom A is included in every supported model of a consistent program P, it is written as $P \models A$.[8]

An *abductive program* is a pair $\langle P, \mathscr{A} \rangle$ where P is a program and $\mathscr{A}$ is a set of ground atoms. Every element in $\mathscr{A}$ is called an *abducible*. Given an observation O as a ground literal, O is *explained* under $\langle P, \mathscr{A} \rangle$ if there is $H \subseteq \mathscr{A}$ such that $P \cup H \models O$ and $P \cup H$ is consistent.

In one-dimensional CAs, a cell x_i at a time step t is represented by an atom $c(x_i, t)$. Two cells adjacent to x_i are then represented by $c(x_{i-1}, t)$ and $c(x_{i+1}, t)$. If the state of a cell x_i at a time step t is 1, it is represented by $c(x_i, t)$. Else if the state of a cell x_i at a time step t is 0, it is represented by $\mathit{not}\ c(x_i, t)$. A transition rule of an elementary CA is represented by a program P, in which the states of neighbors at a time step t are specified in the body of a rule and the state of the central cell at the next time step is specified in the head of a rule. This type of programs is called *acyclic programs* [4], that is, for every rule $A \leftarrow B_1, \ldots, B_m, \mathit{not}\, B_{m+1}, \ldots, \mathit{not}\, B_n$ in $ground(P)$, $|\, A \,| > |\, B_i \,|$ $(1 \leq i \leq n)$ holds where $|\ \ |\colon \mathscr{B} \rightarrow \mathbf{N}$ is a function (called a *level mapping*) from the Herbrand base to natural numbers. Every acyclic program has the unique supported model [4].

Example 6. Wolfram's rule 30 is written as the program P_{30}:

---

[8] We abuse the notation $\models$ for representing the entailment relation in both first-order logic and logic programs. The distinction is clear from the context, however.

$$c(x_i,t+1) \leftarrow c(x_{i-1},t), not\ c(x_i,t), not\ c(x_{i+1},t),$$
$$c(x_i,t+1) \leftarrow not\ c(x_{i-1},t), c(x_i,t), c(x_{i+1},t),$$
$$c(x_i,t+1) \leftarrow not\ c(x_{i-1},t), c(x_i,t), not\ c(x_{i+1},t),$$
$$c(x_i,t+1) \leftarrow not\ c(x_{i-1},t), not\ c(x_i,t), c(x_{i+1},t).$$

The four rules respectively represent transitions satisfying $x_i^{t+1} = 1$ in Table 1. Note that no rule is required for representing transitions satisfying $x_i^{t+1} = 0$. This is because if $c(x_i,t+1)$ is not derived from the program, it is assumed that the state of a cell x_i at a time step $t+1$ is 0 under the *closed world assumption* [32] of logic programs.

To compute CAs in logic programming, we assume a *finite* CA which consists of a finite number of cells. In a one-dimensional CA, we consider a sequence of n-cells $\langle x_1 \cdots x_n \rangle$. Two cells on the edges are handled in a toroidal manner, that is, x_1 and x_n are assumed to be adjacent. The situation is realized by introducing the extra rules for the edges to a program. For instance, in Example 6, the first rule of P_{30} is rewritten as:

$$c(x_i,t+1) \leftarrow c(x_{i-1},t), not\ c(x_i,t), not\ c(x_{i+1},t), i \neq 1, i \neq n,$$
$$c(x_1,t+1) \leftarrow c(x_n,t), not\ c(x_1,t), not\ c(x_2,t),$$
$$c(x_n,t+1) \leftarrow c(x_{n-1},t), not\ c(x_n,t), not\ c(x_1,t).$$

Other three rules in P_{30} are also rewritten in a similar way. In what follows, we assume that such extra rules are included in a program representing a transition rule of an elementary CA.

State transitions of CA configurations are computed by the *fixpoint operator* of logic programming. Given an interpretation $I \subseteq \mathscr{B}$, the mapping $T_P : 2^{\mathscr{B}} \rightarrow 2^{\mathscr{B}}$ is defined as follows [3]:

$$T_P(I) = \{A \mid (A \leftarrow B_1,\ldots,B_m, not\,B_{m+1},\ldots,not\,B_n) \text{ is in } ground(P),$$
$$\{B_1,\ldots,B_m\} \subseteq I \text{ and } \{B_{m+1},\ldots,B_n\} \cap I = \emptyset\}.$$

The *orbit* of I with respect to T_P is the sequence (often called *trajectory*) $\langle T_P{}^k(I)\rangle_{k\in\omega}$, where $T_P{}^0(I) = I$ and $T_P{}^{k+1}(I) = T_P(T_P{}^k(I))$ for $k = 0,1,\ldots$ It is known that I is a supported model of P iff I is a fixpoint of T_P, i.e., $T_P(I) = I$ [3].

The CA configuration at a time step $t\,(\geq 0)$ is represented by the set

$$I_t = \{\, c(x_i,t) \mid 1 \leq i \leq n \text{ and the state of a cell } x_i \text{ at a time step } t \text{ is } 1\,\}.$$

Let $\mathscr{I}_t = \{I_t \mid I_t$ is any configuration at a time step $t\}$. Then the following result holds.[9]

Proposition 12. *Let P be a program representing a transition rule of an elementary CA. Given an initial configuration by an interpretation I_0, the transition of config-*

[9] A similar result is shown in [20, Proposition 3.1] in the context of Boolean networks.

urations is provided by the orbit $\langle {T_P}^k(I_0)\rangle_{k\in\omega}$*. The configuration at a time step t is provided by* ${T_P}^t(I_0)\cap I_t$*.*

Proof. Let I_t be an interpretation which represents a configuration at a time step t. For each rule $A \leftarrow B_1,\ldots,B_m, not\, B_{m+1},\ldots,not\, B_n$ in $ground(P)$, if $\{B_1,\ldots,B_m\}\subseteq I_t$ and $\{B_{m+1},\ldots,B_n\}\cap I_t=\emptyset$, then $B_1,\ldots,B_m$ are neighbor cells with the state 1 and $B_{m+1},\ldots,B_n$ are neighbor cells with the state 0 at a time step t. Then, $A\in I_{t+1}$ is in $T_P(I_t)$ iff the state of the central cell A at the time step $t+1$ is 1. Thus, $I_t = T_P(I_{t-1})\cap I_t=\cdots={T_P}^t(I_0)\cap I_t$, and the orbit $\langle {T_P}^k(I_0)\rangle_{k\in\omega}$ represents the transition of configurations. □

Example 7. Suppose the program P_{30} of Example 6 with $n=7$, i.e., an elementary CA with seven cells. (We assume that extra rules are prepared for the edges.) Given the initial configuration as $I_0=\{c(x_4,0)\}$, the next generation of the CA is provided by $T_P(I_0)\cap I_1=\{c(x_3,1),c(x_4,1),c(x_5,1)\}$, which represents that $c(x_3,1)=c(x_4,1)=c(x_5,1)=1$ and the state of other cells are 0 (cf. Figure 1).

Proposition 13. *Let P be a program representing a transition rule of an elementary CA. Given a configuration at a time step t by an interpretation* I_t*,* I_t *is a GOE iff there is no* I_{t-1} *such that* $I_t=T_P(I_{t-1})\cap I_t$*.*

Proof. I_t is a GOE iff I_t has no predecessor under a transition rule in P iff there is no I_{t-1} such that $I_t=T_P(I_{t-1})\cap I_t$. □

Predictability and unpredictability of CA configurations are characterized by abductive logic programs as follows.

Proposition 14. *Let* $\langle P,\mathscr{I}_0\rangle$ *be an abductive program, where P represents a transition rule of an elementary CA and* $\mathscr{I}_0$ *is the set of all configurations at the time step* 0*.*

1. *A configuration* $E_t\in\mathscr{I}_t$ $(t>0)$ *is predictable if there is* $H\in\mathscr{I}_0$ *such that* E_t *is included in the supported model of a consistent program* $P\cup H$*.*
2. *A configuration* $E_t\in\mathscr{I}_t$ $(t>0)$ *is unpredictable if there is no* $H\in\mathscr{I}_0$ *such that* E_t *is included in the supported model of a consistent program* $P\cup H$*.*

Proof. (1) If there is $H\in\mathscr{I}_0$ such that E_t is included in the supported model of a consistent program $P\cup H$, then $P\cup H\models E_t$. Then, E_t is predictable under $\langle P,\mathscr{I}_0\rangle$. (2) If there is no $H\in\mathscr{I}_0$ such that E_t is included in the supported model of a consistent program $P\cup H$, then $P\cup H\not\models E_t$ for any $H\in\mathscr{I}_0$. Then, E_t is unpredictable under $\langle P,\mathscr{I}_0\rangle$. □

For checking whether E_t is included in the supported model of $P\cup H$, we can use a solver [11] which performs incremental computation of *answer sets* (which are equivalent to supported models in acyclic programs) at each time step.

4.2 Computing Life

We next consider two-dimensional CAs. Consider a two-dimensional CA with an $m \times n$ grid. Then, an atom $c(x_i, y_j, t)$ $(1 \le i \le m,\ 1 \le j \le n)$ represents that the state of a cell (x_i, y_j) at a time step t is 1; and a negated atom *not* $c(x_i, y_j, t)$ represents that the state of a cell (x_i, y_j) at a time step t is 0. A cell $c(x_i, y_j, t)$ has eight surrounding cells: $c(x_{i-1}, y_{j-1}, t)$, $c(x_i, y_{j-1}, t)$, $c(x_{i+1}, y_{j-1}, t)$, $c(x_{i-1}, y_j, t)$, $c(x_{i+1}, y_j, t)$, $c(x_{i-1}, y_{j+1}, t)$, $c(x_i, y_{j+1}, t)$, and $c(x_{i+1}, y_{j+1}, t)$. Like one-dimensional CAs, a transition rule of a two-dimensional CA is represented by a program P, in which the states of neighbors at a time step t are specified in the body of a rule and the state of the central cell at the next time step is specified in the head of a rule. The program is again an acyclic program.

In what follows, we use the notion of a *cardinality constraint* [28] of the form:

$$L\{A_1, \ldots, A_k\}U$$

where $A_1, \ldots, A_k$ are atoms, and L and U are two integers such that $L \le U$. The constraint is satisfied by any interpretation I such that $L \le |\, I \cap \{A_1, \ldots, A_k\}\,| \le U$. A rule may contain a cardinality constraint in its body. Any rule containing cardinality constraints is transformed into semantically equivalent rules having no constraint. For instance, $\{\, p \leftarrow 1\{q, r\}2 \,\}$ is equivalent to $\{\, p \leftarrow q,\ \ p \leftarrow r,\ \ p \leftarrow q, r \}$.

Example 8. A logic program for computing *Life* is provided by the program P_{Life}:[10]

$$\begin{aligned}
c(x_i, y_j, t+1) \leftarrow\ & c(x_i, y_j, t), 2\{c(x_{i-1}, y_{j-1}, t), c(x_i, y_{j-1}, t),\\
& c(x_{i+1}, y_{j-1}, t), c(x_{i-1}, y_j, t), c(x_{i+1}, y_j, t),\\
& c(x_{i-1}, y_{j+1}, t), c(x_i, y_{j+1}, t), c(x_{i+1}, y_{j+1}, t)\}3,\\
c(x_i, y_j, t+1) \leftarrow\ & not\ c(x_i, y_j, t), 3\{c(x_{i-1}, y_{j-1}, t), c(x_i, y_{j-1}, t),\\
& c(x_{i+1}, y_{j-1}, t), c(x_{i-1}, y_j, t), c(x_{i+1}, y_j, t),\\
& c(x_{i-1}, y_{j+1}, t), c(x_i, y_{j+1}, t), c(x_{i+1}, y_{j+1}, t)\}3.
\end{aligned}$$

The first rule says that any live cell with two or three live neighbors at a time step t stays alive at the next time step $t+1$. The second rule says that any dead cell with exactly three live neighbors at a time step t becomes alive at the next time step $t+1$. Rules for edges are considered for the cells $c(x_1, y_j, t)$, $c(x_m, y_j, t)$, $c(x_i, y_1, t)$, and $c(x_i, y_n, t)$. As before, we assume that the rules in *Life* are rewritten in an appropriate manner for those cells.

In two-dimensional CAs, a CA configuration at a time step t is represented by the set:

$$I_t = \{\, c(x_i, y_j, t) \mid 1 \le i \le m, 1 \le j \le n \text{ and a state of a cell } (x_i, y_j) \text{ at a time step } t \text{ is } 1 \,\}.$$

[10] A similar encoding is provided in [36].

Let $\mathscr{I}_t = \{ I_t \mid I_t$ is any configuration at a time step $t\}$. Then, the following result holds. (Proofs are similar to Propositions 12–14.)

Proposition 15. *Let P be a program representing the transition rule of Life. Given an initial configuration by an interpretation I_0, the transition of configurations is provided by the orbit $\langle {T_P}^k(I_0)\rangle_{k\in\omega}$. The configuration at a time step t is provided by ${T_P}^t(I_0)\cap I_t$.*

Proposition 16. *Let P be a program representing the transition rule of Life. Given a configuration by an interpretation I_t, I_t is a GOE iff there is no I_{t-1} such that $I_t = T_P(I_{t-1})\cap I_t$.*

Proposition 17. *Let $\langle P, \mathscr{I}_0\rangle$ be an abductive program, where P represents the transition rule of Life and $\mathscr{I}_0$ is the set of all configurations at the time step* 0.

1. *A configuration $E_t \in \mathscr{I}_t$ $(t > 0)$ is predictable if there is $H \in \mathscr{I}_0$ such that E_t is included in the supported model of a consistent program $P\cup H$.*
2. *A configuration $E_t \in \mathscr{I}_t$ $(t > 0)$ is unpredictable if there is no $H \in \mathscr{I}_0$ such that E_t is included in the supported model of a consistent program $P\cup H$.*

We have implemented the *Life* in logic programming and verified that there is no GOE in grids of the 6×6 wide or smaller, and the configuration of Figure 3 is not generated by one step of deduction from any configuration of a 12×12 grid.[11]

5 Discussion

5.1 Extended Abduction

In Section 2, we considered an abductive framework based on first-order logic, and characterized the emergence of CA configurations in terms of (un)predictability of events under an abductive framework. In the abductive framework, however, hypotheses are only added to a background theory to explain an observation. As a result, predictability or unpredictability of events is restricted to the case in which they are deduced by expanding a theory. In real life, however, we may *revise* our background knowledge to explain a given observation, and would predict a new event not only by adding hypotheses to a background theory but also by removing information in the theory. For instance, suppose an abductive framework $\langle B, \mathscr{H}\rangle$ such that $B = \{\, p \rightarrow q\wedge r,\;\; \neg p\,\}$ and $\mathscr{H} = \{\, p, \neg p\,\}$. Given the observation $O = q$, it cannot be explained under $\langle B, \mathscr{H}\rangle$ because p is the only candidate hypothesis to explain q but $B\cup\{p\}$ is inconsistent. In this case, q is explained by removing $\neg p$ from B and

[11] In seeking/checking GOEs of a particular size, we need no rule for the edges in the grid. For instance, checking the GOE of Figure 3, we prepare the rules of Example 8 for the range of $1 \leq i, j \leq 12$. We then verified the 10×10 configuration is not produced by one step computation from any configuration of the 12×12 grid.

introducing p to B. Then r is predicted in the revised theory $(B \setminus \{\neg p\}) \cup \{p\}$. In this example, formulas in $B \setminus \mathscr{H}$ are considered *persistent knowledge* that is unchanged, while formulas in $B \cap \mathscr{H}$ are considered *temporary knowledge* that is subject to change.

To formulate the situation, we need to extend the abductive framework. Inoue and Sakama [16] introduce the framework of *extended abduction*, in which hypotheses can not only be added to a background theory but also be removed from it to explain (or unexplain) an observation. Let $\langle B, \mathscr{H} \rangle$ be an abductive framework and P a formula representing a *positive observation*. Then, $(I,J)\ (\in 2^{\mathscr{H}} \times 2^{\mathscr{H}})$ is an *explanation* of P under $\langle B, \mathscr{H} \rangle$ if

1. $(B \setminus J) \cup I \models P$,
2. $(B \setminus J) \cup I$ is consistent.

On the other hand, given a formula N representing a *negative observation*, a pair (I,J) is an *anti-explanation* of N under $\langle B, \mathscr{H} \rangle$ if

1. $(B \setminus J) \cup I \not\models N$,
2. $(B \setminus J) \cup I$ is consistent.

With this setting, we can compute explanations (resp. anti-explanations) to explain a positive observation (resp. unexplain a negative observation) by removing old temporary knowledge and introducing new hypothesis. The obtained (anti-)explanations are then used for predicting unseen phenomena. Note that in extended abduction, a background theory can be represented by *nonmonotonic logics* in general. In nonmonotonic logics, introducing a formula to a background theory does not necessarily increase proven formulas, and removing a formula from a background theory may increase proven formulas. In [16] extended abduction is introduced using *autoepistemic logic*, and extended abduction is also formulated in logic programming [17, 34]. An abductive framework based on extended abduction is called an *extended abductive framework*. By contrast, the abductive framework provided in Section 2 is called a *normal abductive framework*.[12]

Using this extended abductive framework, we can define that an event E is predictable (resp. unpredictable) under $\langle B, \mathscr{H} \rangle$ if E has an (resp. no) explanation (I,J) under $\langle B, \mathscr{H} \rangle$. Technically, the problem of (un)predictability of events under the extended abductive framework is transformed into the problem under the normal abductive framework. Put $B' = B \setminus \mathscr{H}$. Then, $(B \setminus J) \cup I \models E$ iff $B' \cup (B \setminus (B' \cup J)) \cup I \models E$ where $(B \setminus (B' \cup J)) \cup I \subseteq \mathscr{H}$ [34]. With this setting, an event E is predictable (resp. unpredictable) under an extended abductive framework $\langle B, \mathscr{H} \rangle$ iff E is predictable (resp. unpredictable) under a normal abductive framework $\langle B', \mathscr{H} \rangle$. By this fact, the problem of (un)predictability of events under the extended abductive framework is also computed in logic programming as in Section 4.

---

[12] Aliseda [1] introduces the framework of *abductive revision* which consists of expansion and contraction of a background theory. In this framework, contraction is done to remove anomaly in a background theory while it does not consider a nonmonotonic theory as a background theory.

5.2 Related Work

Poole [31] introduces a logical framework for abduction and prediction. In his *Theorist* framework, defaults are possible hypotheses used for prediction. He then considers three different cases of prediction. In the first case, an event E is predictable if it is explainable (i.e., E is in some extension of a default theory). In the second case, an event E is predictable if E is explainable and $\neg E$ is not explainable. The third case is that an event E is predictable if E is included in every extension. Comparing three definitions, the first one is weaker than the second one, which in turn is weaker than the third one. That is, if an event E is predictable under the third definition, it is also predictable under the first definition, but not vice versa. Poole considers that the third one is the most reasonable definition of prediction. In the abductive framework $\langle B, \mathscr{H} \rangle$ considered in this paper, if we view a hypothesis $h \in \mathscr{H}$ as a normal default rule $\frac{:h}{h}$ in the sense of [33] (h is assumed as far as it is consistent with B), our definition of predictability is close to the first definition of Poole. In the first definition of Poole, one may predict E and $\neg E$ using different hypotheses (cf. Proposition 3). Poole remarks that "it seems wrong to both predict α and $\neg\alpha$, ... It corresponds more to "may be true" than to prediction" [31]. In real life, however, we often predict contradictory results depending on different hypotheses. For instance, scientists predict a global temperature increase of between 2 and 6 degrees centigrade by 2100 depending on different scenarios [26].

Shanahan [35] characterizes abduction and deduction in terms of the *event calculus*. In his framework, events are temporally ordered and an event at some time point predicts a future event. On the other hand, prediction is not always made for future events. If a physicist proposes a new law, the law predicts some events which would be examined by past experimental data as well as future experiments. Evans and Kakas [8] introduce a logical framework for hypothetico-deductive reasoning. They introduce the 4-tuple $\langle T, O, A, S \rangle$ where T is a background theory (definite Horn clauses), and O, A and S are sets of ground atoms respectively representing the *observation set*, *abducibles* and *observables*. Given an observation G, a *corroborated explanation* Δ for G is defined as

1. $T \cup \Delta \models G$ where $\Delta \subseteq A$,
2. If $T \cup \Delta \models \Pi$ and $\Pi \subseteq S$, then $\Pi \subseteq O$.

In this framework, the pre-specified sets O and S are used for testing the validity of abductive explanation Δ. On the other hand, providing O and S as well as A means that a reasoner should know not only possible hypotheses but also predictable events in advance. Pereira and Pinto [29] introduce a mechanism of inspecting side-effects of abduction in logic programs. An abductive explanation may deduce new facts as predictions (or *side-effects*). To know whether facts of particular interests become true (or false) as a consequence of abduction, they introduce *inspection points* to a program. The inspection mechanism is useful not only for detecting a particular prediction, but for selecting plausible abductive solutions. Josephson [23] captures prediction as a part of an *inductive projection*. An inductive projection consists of two parts of inferences: *inductive generalization* from observations to the hypothesis "all A's are B's", and prediction from the hypothesis to the conclusion "the next A will be a B". He then considers that prediction is not typically deduction but belongs to the same family as *statistical syllogism* such that "m/n of the A's are B's (where $m/n > 1/2$). Therefore, the next A will be a B." Hempel also argues the statistical aspect of prediction as well as the statistical explanation [14]. The notion of (un)predictability of events is considered under statistical prediction, and would be used for characterizing the behavior of *probabilistic CAs* [39]. Formal analyses on (un)predictability of events in probabilistic models is an interesting research topic.

We realized CA computations in logic programs, while a class of logic programs is viewed as CAs. A logic program is *covered* if every variable occurring in the body also occurs in the head. The programs we introduced in Section 4 are in this class. Blair *et al.* [5] show that every covered logic program is in fact a CA. It is known that a two-state CA is an instance of *Boolean networks* (BNs) [25]. Inoue [20] observes that any interpretation of a logic program P can reach either a fixpoint or a periodically oscillating cycle in an orbit, which corresponds to an *attractor* of the corresponding Boolean network. Inoue and Sakama [22] then semantically characterize those attractors as *supported classes* of the logic program P. These results indicate that orbits and attractors of CAs and BNs can be analyzed in terms of logic programs, yet GOEs cannot be analyzed in such works. In fact, it is necessary to invert transitions to find GOEs, which can only be analyzed by abduction in this paper.

6 Conclusion

This paper provided a logical account of (un)predictability and applied the notion to the problem of identifying configurations in CAs. In a CA future patterns are generally unforeseen while any configuration is (logically) predictable as far as it can be reached by a step by step computation. In this sense, GOEs are configurations that are really unpredictable in CAs. We realize CAs in logic programming and verified some of the existing GOEs. To find a new GOE, however, we need further optimization by eliminating symmetric patterns. The issue is left for future research.

Acknowledgements The authors thank Torsten Schaub for useful discussion for implementing *Life* in logic programming.

References

1. A. Aliseda. *Abductive reasoning: logical investigations into discovery and explanation.* Synthese Library 330, Springer (2006).
2. S. Amoroso and J. N. Patt. Decision procedure for surjectivity and injectivity of parallel maps for tessellation structures. *J. Computer and System Sciences* 6:448–464 (1972).

3. K. R. Apt, H. A. Blair and A. Walker. Towards a theory of declarative knowledge. In: J. Minker (ed.), *Foundations of Deductive Databases and Logic Programming*, pp. 89–148, Morgan Kaufmann (1988).
4. K. R. Apt and M. Bezem. Acyclic Programs. *New Generation Computing* 9:335–363 (1991).
5. H. A. Blair, J. Chidella, F. Dushin, A. Ferry and P. Humenn. A continuum of discrete systems. *Annals of Mathematics and Artificial Intelligence* 21:153–186 (1997).
6. J. H. Conway. The game of life. *Scientific American* 223:120–123 (1970).
7. T. Eiter and G. Gottlob. The complexity of logic-based abduction. *J. ACM* 42:3–42 (1995).
8. C. Evens and A. C. Kakas. Hypothetico-deductive reasoning. *Proceedings of the International Conference on Fifth Generation Computer Systems*, pp. 546–554, ICOT (1992).
9. A. Flammenkamp. Garden of Eden / Orphan. `http://wwwhomes.uni-bielefeld.de/achim/orphan_7th.html` (2011).
10. D. M. Gabbay and J. Woods. *The Reach of Abduction: Insight and Trial*. A Practical Logic of Cognitive Systems, vol. 2 (2005).
11. M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub and S. Thiele. Engineering an incremental ASP solver. In: *Proceedings of the 24th International Conference on Logic Programming*, Lecture Notes in Computer Science, 5366, pp. 190–205, Springer (2008).
12. C. G. Hempel. The function of general laws in history. *Journal of Philosophy*, 39:35–48 (1942).
13. C. G. Hempel and P. Oppenheim. Studies in the logic of explanation. *Philosophy of Science*, 15:135–175 (1948).
14. C. G. Hempel. *Aspects of Scientific Explanation*. In: C. Hempel. *Aspects of Scientific Explanation and other Essays in the Philosophy of Science*, The Free Press, New York (1965).
15. C. G. Hempel. *Philosophy of Natural Science*. Prentice Hall, NJ. (1966).
16. K. Inoue and C. Sakama. Abductive framework for nonmonotonic theory change. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 204–210, Morgan Kaufmann (1995).
17. K. Inoue. A simple characterization of extended abduction. In: *Proceedings of the 1st International Conference on Computational Logic*, Lecture Notes in Computer Science, 1861, pp. 718–732, Springer (2000).
18. K. Inoue and C. Sakama. On abductive equivalence. L. Magnani (ed.), *Model-Based Reasoning in Science and Engineering. Cognitive Science, Epistemology, Logic, Studies in Logic*, pp. 333–352, College Publications, London (2006).
19. K. Inoue and C. Sakama. Comparing abductive theories. In: *Proceedings of the 18th European Conference on Artificial Intelligence*, IOS Press, pp. 35–39 (2008).
20. K. Inoue. Logic programming for Boolean networks. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 924–930 (2011).
21. K. Inoue. DNF hypotheses in explanatory induction. In: *Proceedings of the 21nd International Conference on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence, 7207, pp. 173–188, Springer (2012).
22. K. Inoue and C. Sakama. Oscillating behavior of logic programs. In: *Correct Reasoning: Essays on Logic-Based AI in Honour of Vladimir Lifschitz*, Lecture Notes in Computer Science, 7265, pp. 345–362, Springer (2012).
23. J. R. Josephson. Conceptual analysis of abduction. In: J. R. Josephson and S. G. Josephson (eds.) *Abductive Inference, Computation, Philosophy, Technology*. Cambridge University Press (1996).
24. A. C. Kakas, R. A. Kowalski and F. Toni. The role of abduction in logic programming. In: D. M. Gabbay, C. J. Hogger and J. A. Robinson (eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 5, pp. 235–324, Oxford University Press (1998).
25. S. A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press (1993).
26. B. Metz, O. R. Davidson, P. R. Bosch, R. Dave, L. A. Meyer (eds.) *Climate Change 2007: Mitigation of Climate Change*. Contribution of working group III to the fourth assessment report of the intergovernmental panel on climate change. Cambridge University Press (2008).

27. G. J. Martínez, A. Adamatzky, R. Alonso-Sanz and J. C. Seck-Touh-Mora. Complex dynamics emerging in Rule 30 with majority memory. *Complex Systems* 18:345–365 (2009).
28. I. Niemelä, P. Simons and T. Soininen. Stable model semantics of weighted constraint rules. In: *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning*, Lecture Notes in Artificial Intelligence, 1730, pp. 317–331, Springer (1999).
29. L. M. Pereira and A. M. Pinto. Inspecting side-effects of abduction in logic programs. In: *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning*, Lecture Notes in Artificial Intelligence, 6565, pp. 148–163, Springer (2011).
30. G. D. Plotkin. A note on inductive generalization. In: B. Meltzer and D. Michie (eds.), *Machine Intelligence* 5, Edinburgh University Press, pp. 153–63 (1970).
31. D. Poole. Explanation and prediction: an architecture for default and abductive reasoning. *Computational Intelligence* 5:97–110 (1989).
32. R. Reiter. On closed world databases. In: H. Gallaire and J. Minker (eds.), *Logic and Data Bases*, Plenum, New York, pp. 55–76 (1978).
33. R. Reiter. A logic for default reasoning. *Artificial Intelligence* 13:81–132 (1980).
34. C. Sakama and K. Inoue. An abductive framework for computing knowledge base updates. *Theory and Practice of Logic Programming* 3:671–713 (2003).
35. M. Shanahan. Prediction is deduction but explanation is abduction. In: *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pp. 1055–1060 (1989).
36. T. Syrjanen. Modelling the Game of Life using logic programs. In: Leksa Notes in Computer Science: Festschrift in Honour of Professor Leo Ojala. Helsinki University of Technology (2000).
37. N. N. Taleb. *The Black Swan: The Impact of the Highly Improbable*. Random House (2007).
38. S. Wolfram. *Cellular Automata and Complexity*. Westview Press (1994).
39. S. Wolfram. *A New Kind of Science*. Wolfram Media Inc. (2002).