TUCS

Ville Lukkarila

Sensitivity and topological mixing are undecidable for reversible one-dimensional cellular automata

Turku Centre for Computer Science

TUCS Technical Report
No 927, January 2009

Sensitivity and topological mixing are undecidable for reversible one-dimensional cellular automata

Ville Lukkarila
Turku Centre for Computer Science
FI-20520 Turku
Finland
Department of Mathematics
University of Turku
FI-20014 Turku
Finland
E-mail: ville.lukkarila@utu.fi
This research has been supported by the Fund of Yrjö, Vilho and Kalle Väisälä.

TUCS Technical Report
No 927, January 2009

Abstract

It is shown by a reduction from the reversible Turing machine halting problem that sensitivity is undecidable even for reversible one-dimensional cellular automata. With a few additional constructions, the undecidability of topological mixing and the undecidability of topological transitivity follow. Furthermore, sets of topologically mixing cellular automata and non-sensitive cellular automata are recursively inseparable. It follows that Devaney's chaos and Knudsen's chaos are undecidable dynamical properties. [This paper has been submitted to Journal of Cellular Automata in September 2008.]

Keywords: cellular automata, chaos, sensitivity, topological transitivity, topological mixing

TUCS Laboratory
Discrete Mathematics for Information Technology

1 Introduction

Cellular automata are a simple formal model for the study of phenomena caused by local interaction of finite objects. A cellular automaton consists of a regular lattice of cells. Each cell has a state which is updated on every time step according to some local rule which is the same for all the cells in the lattice. The locally used update rule is simply called a local rule. On every time step the next state of the cell is determined according to its own previous state and the previous states of a finite number of its neighbors. The state information of the entire lattice at any time step is called a configuration of the cellular automaton.

The cellular automata were introduced by von Neumann to study biologically motivated computation and self-replication [18]. The mathematical study of cellular automata in symbolic dynamics was initiated by Hedlund [9]. Although cellular automata may seem a simple model for computation, they can exhibit very complex behavior. A well-known example of such complex behavior is the Game-of-Life. Even though the rule according to which the lattice is updated is quite simple in the Game-of-Life, some state patterns interact in a somewhat complex manner. In fact, the Game-of-Life has been shown to be computationally universal. In particular, any Turing machine can be simulated with some cellular automaton in a natural way.

Cellular automata have been studied very extensively also as discrete time dynamical systems. Injectivity, surjectivity, nilpotency, equicontinuity, sensitivity to initial conditions, topological transitivity, topological mixing, chaos, and different variants of expansivity are widely studied properties of cellular automata. These properties have been studied also in the sense of algorithmic decidability and undecidability. Nilpotency is an undecidable property even for one-dimensional cellular automata [10]. Injectivity and surjectivity are known to be decidable for one-dimensional cellular automata but undecidable for two-dimensional cellular automata [11]. It was mentioned in [6] that sensitivity, equicontinuity, transitivity, and ergodicity are believed to be undecidable properties of cellular automata but no proof was given. It was shown in [8] that equicontinuity and sensitivity are undecidable for irreversible one-dimensional cellular automata. Recently it was shown by Kari and Ollinger that equicontinuity is undecidable even for reversible one-dimensional cellular automata [13].

In this paper it is shown that also sensitivity to initial conditions is an undecidable property even for reversible one-dimensional cellular automata. By modifying the construction even further, it is shown that topological mixing and topological transitivity are undecidable properties even for reversible cellular automata. Due to the close relation between transitivity and different definitions of chaotic behavior, it follows that chaotic behavior is an undecidable property both according to the definition of Devaney [7] and according to the definition of Knudsen [15].

2 Cellular automata

Cellular automata are dynamical systems which update the variables on an infinite d-dimensional lattice according to some function with a finite number of arguments. Formally, a *cellular automaton* is a 4-tuple $\mathcal{A} = (d, A, N, f)$, where d is the *dimension*, A is the *state set*, $N = (\overrightarrow{x_1}, \ldots, \overrightarrow{x_n})$ is the *neighborhood vector* consisting of vectors in $\mathbb{Z}^d$ and $f : A^n \to A$ is the *local rule*. A *configuration* $c \in A^{\mathbb{Z}^d}$ is a mapping which assigns a unique state for each cell location in $\mathbb{Z}^d$. The cells in locations $\overrightarrow{x} + \overrightarrow{x_i}$ are called *neighbors* of the cell in location $\overrightarrow{x}$.

At every time step the new configuration c' is determined by

$$c'(\overrightarrow{x}) = f(c(\overrightarrow{x} + \overrightarrow{x_1}), \ldots, c(\overrightarrow{x} + \overrightarrow{x_n})), \tag{1}$$

that is, the new state of the cell in location $\overrightarrow{x}$ is computed by applying the local rule to its neighbors. The *global rule* $F : A^{\mathbb{Z}^d} \to A^{\mathbb{Z}^d}$ is defined by setting $F(c) = c'$ in the sense of equation 1.

A cellular automaton is said to be *reversible* if the global rule F has an inverse mapping F^{-1}. It can be shown that if the inverse mapping F^{-1} exists, it is a global rule of a cellular automaton, that is, it is defined by a local rule. It is also known that F is reversible if, and only if, it is injective. Furthermore, in the case of cellular automata injectivity of the global rule implies surjectivity of the global rule [12].

The distance between two different configurations c and e can be defined to be

$$d(c, e) = \left(\frac{1}{2}\right)^{\min\{\|\overrightarrow{x}\|_\infty \mid c(\overrightarrow{x}) \neq e(\overrightarrow{x})\}},$$

where $\|\cdot\|_\infty$ is the max-norm. Function $d(\cdot, \cdot)$ is also a metric thus making the set of configurations a metric space. The balls in the metric are called *cylinders* and they form a basis for the topology. Radius r cylinder containing configuration c is the set

$$\mathrm{Cyl}(c, r) = \left\{ e \in A^{\mathbb{Z}^d} \mid c(\overrightarrow{x}) = e(\overrightarrow{x}) \text{ when } \|\overrightarrow{x}\|_\infty \leq -\log_2 r \right\}$$

For every radius r there are only finitely many cylinders and these cylinders are by definition disjoint. Therefore, radius r cylinders form a partition of the space of configurations. Hence, every cylinder is clopen because the complement of every cylinder is a union of other cylinders with the same radius.

In the one-dimensional case, one can define cylinders differently as sets

$$\mathrm{Cyl}(w, k) = \left\{ c \in A^{\mathbb{Z}} \mid c(i + k) = w(i) \text{ when } i \leq |w| - 1 \right\}$$

where $w \in A^*$ is a finite word and $w(i-1)$ denotes the ith letter of the word. The word consisting of states in locations i through j (when $i \leq j$) in a configuration c is denoted by $c[i, j]$.

Pair (X, F) is a *dynamical system* if X is a compact metric space and $F : X \to X$ is a continuous mapping. In particular, d-dimensional cellular automata are dynamical systems of the form $(A^{\mathbb{Z}^d}, F)$.

A dynamical system (X, F) is said to be *periodic* if there exists such a positive integer p that $F^p(x) = x$ for every $x \in X$. The dynamical system is said to be *ultimately periodic* if there exists such positive integers p_0 and p that $F^{p_0+p}(x) = F^{p_0}(x)$ for every $x \in X$.

A point $x \in X$ is an *equicontinuity point* of mapping F if for any $\varepsilon > 0$ there exists such $\delta > 0$ that for any point $y \in X$ and integer $n \in \mathbb{N}$,

$$d(x, y) < \delta \Longrightarrow d(F^n(x), F^n(y)) < \varepsilon.$$

A dynamical system (X, F) is *equicontinuous* if every point $x \in X$ is an equicontinuity point.

Theorem 2.1 ([2]). *A cellular automaton is equicontinuous if, and only if, it is ultimately periodic.*

Definition 2.2. *A word $w \in A^*$ is* blocking *if there exists such a sequence of words $(w_n)_{n=0}^{\infty}$ that $w_n \in A^r$ and there exists such an integer i that $F^n(\mathrm{Cyl}(w, i)) \subseteq \mathrm{Cyl}(w_n, 0)$ for any $n \in \mathbb{N}$.*

Theorem 2.3 ([2]). *Any equicontinuity point has an occurrence of a blocking word. Conversely, any point with infinitely many occurrences of blocking words to the left and right of the origin is an equicontinuity point.*

Definition 2.4. *A dynamical system (X, F) is* sensitive to initial conditions *(or* sensitive*) if there exists such $\varepsilon > 0$ that for any $x \in X$ and $\delta > 0$ there exists such a point $y \in X$ that*

$$0 < d(x, y) < \delta \text{ and } d(F^n(x), F^n(y)) \geq \varepsilon$$

for some integer $n \in \mathbb{N}$. If the constant ε exists, it is known as the sensitivity constant*.*

For one-dimensional cellular automata sensitivity is equivalent to the nonexistence of equicontinuity points.

Definition 2.5. *A dynamical system (X, F) is* topologically transitive *(or* transitive*) if for all nonempty open subsets U and V of X there exists such a positive integer n that $F^n(U) \cap V \neq \emptyset$.*

Definition 2.6. *A dynamical system (X, F) has a* dense orbit *if there exists such a point $x \in X$ that for every point $y \in X$ with any $\varepsilon > 0$ there exists such an integer $n \in \mathbb{N}$ that $d(F^n(x), y) < \varepsilon$.*

It can be proved with a topological argumentation that a cellular automaton is topologically transitive if, and only if, it has a dense orbit. A short introduction to transitivity in terms of symbolic dynamics can be found in [17]. It is known that transitivity implies sensitivity [5].

Definition 2.7. *A dynamical system* (X, F) *is* topologically mixing *(or* mixing*) if for all nonempty open subsets* U *and* V *of* X *there exists such a positive integer* n_0 *that* $F^n(U) \cap V \neq \emptyset$ *for all* $n \geq n_0$*.*

Topological mixing is an even stronger property than transitivity. Clearly, a mixing dynamical system is transitive also.

Definition 2.8. *Let* $P(F)$ *denote the set of periodic points of a dynamical system* (X, F)*, that is,*

$$P(F) = \{x \in X \mid \exists n \in \mathbb{N} \setminus \{0\} : F^n(x) = x\}.$$

A dynamical system (X, F) *is said to have* dense periodic points *if* $P(F)$ *is a dense subset of* X*, or equivalently, for any point* $x \in X$ *and any positive real number* $\varepsilon > 0$ *there exists such a periodic point* $y \in P(F)$ *that* $d(x, y) < \varepsilon$*.*

Sometimes the denseness of periodic points is called regularity or topological regularity [4]. On the other hand, cellular automata, whose column subshifts are sofic subshifts, are called regular [16] because the factor words of the elements of a column subshift form a regular language in the sense of formal languages. It is an open problem whether denseness of periodic points is equivalent to surjectivity or not.

Definition 2.9 (Devaney's chaos [7]). *A dynamical system* (X, F) *is said to be* chaotic *according to Devaney's definition if*

1. *it is topologically transitive,*
2. *it has dense periodic points and*
3. *it is sensitive to initial conditions.*

In [4] the authors reviewed some of the properties of discrete time dynamical systems in terms of cellular automata. The authors discussed also Knudsen's definition of chaotic behavior with respect to cellular automata.

Definition 2.10 (Knudsen's chaos [15]). *A dynamical system* (X, F) *is said to be* chaotic *according to Knudsen's definition if*

1. *it has a dense orbit and*
2. *it is sensitive to initial conditions.*

In the case of reversible cellular automata, Devaney's and Knudsen's definitions of chaos are equivalent because a reversible cellular automaton has always dense periodic points and a cellular automaton is transitive if, and only if, it has a dense orbit. As transitivity implies sensitivity, both definitions of chaos are equivalent to transitivity in the reversible case.

A set $A \subseteq \Sigma^*$ is said to be *recursive* if there exist such an algorithm that for a given word $x \in \Sigma^*$ it would return "1" if $x \in A$ and "0" if $x \notin A$. The set A is said to be *recursively enumerable* if there exists such a semi-algorithm that for a given word x it would return "1" if, and only if, $x \in A$. Recursive enumerability of set A means that the elements of A can be printed in some order, although it is not known in which order. The problem of determining whether a given word belongs to a given set A is said to be *decidable* if A is recursive and *undecidable* otherwise. Two sets A and B are said to be *recursively inseparable* if there does not exist such two disjoint recursive sets A' and B' that $A \subseteq A'$ and $B \subseteq B'$ [14]. In particular, if two sets are recursively inseparable, there does not exist an algorithm that would distinguish the elements of the first set from the elements of the second set.

3 Undecidability of sensitivity

In this section it is shown that sensitivity to initial conditions is an undecidable property for reversible one-dimensional cellular automata. The result follows by constructing a cellular automaton which is sensitive if, and only if, a given reversible Turing machine does not halt on an empty tape.

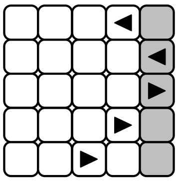
Theorem 3.1 ([1]). *It is undecidable whether a given reversible Turing machine halts when started on an empty tape and the initial state.*

3.1 Concept of signals

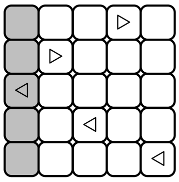
In what follows, the concept of a *signal* is used frequently. Although it is a very informal concept, formally a signal could be described as just a state or a component of a state travelling a (piecewise) linear path. A signal has speed, that is, a number of cells it travels per one time step and a direction to which it moves.

Let $|i - j| < |v|$. If a signal with speed $v > 0$ in location i is said to *collide* with or *bounce* off the cell in location $j > i$, its speed is changed from v to $-v$ and its new location is $(j + 1) - (|v| - |i - j|)$. Similarly, if a signal with speed $v < 0$ in location i bounces off the cell in location $j < i$, its new location is $(j-1)+(|v|-|i-j|)$. If the signal is located between two cells whose distance is less than the velocity of the signal and both of which it would bounce off, the same idea of changing the speed to the opposite and computing the new location is applied repeatedly. If a signal, which is located at cell i, does not have its path intersect any cells from which it would bounce off in locations j, where $|i - j| < |v|$, then its new location is simply $i + v$.

Geometrically speaking, if cell i occupies the unit interval $[i - \frac{1}{2}, i + \frac{1}{2}]$ and a signal travelling to the right is represented by a line intersecting a point (i, t) at time t with a slope of $1/v$, the signal is reflected as a line with respect to a vertical axis $x = j + \frac{1}{2}$. Similarly, a signal travelling to the left as a line through a point (i, t) at time t with a slope of $1/v$ is reflected with respect to a vertical axis $x = j - \frac{1}{2}$. The idea of signals is shown in Figure 1.



(a) A signal with speed 1 bounces back from a cell to the right.



(b) A signal with speed 2 bounces back from a cell to the left.

Figure 1: Examples of how signals are represented on a cellular automaton and how they bounce back from certain cells.

The concept of signals is used in the following constructions because that way the main ideas of the proofs can be expressed more efficiently. The constructions are sufficiently complex in the sense that simply giving complete description of the local rule is somewhat infeasible.

3.2 Outline of the construction

For each reversible Turing machine, the cellular automaton is constructed in four layers. The construction is presented in this way to make it more readable. The local rule on each of these layers maps states of the neighborhood into a new state component of the layer depending on the state components of the particular layer and the earlier layers. That is, each layer has a single purpose and its contents depend only on the contents of the layer itself and the contents of the underlying layers in the previous configuration.

Layer 1. The configuration is split into areas on each of which the Turing machine is simulated.

Layer 2. For each simulation area on layer 1, a signal is used to check the validity of the initial tape configuration.

Layer 3. If the contents of layers 1 and 2 do not match a valid simulation, signals are generated to forward this information.

Layer 4. If the signals of layer 3 collide with the simulation area border signals, the border signals can be modified to let signals pass from one side to another.

The construction is such that the cellular automaton consisting of only the first three layers always has equicontinuity points. However, the fourth layer is constructed in such a way that in the final construction blocking word sequences exist if, and only if, certain states can be avoided in the blocking word sequences of the cellular automaton consisting of the first three layers.

The idea of the construction is to use the first and the second layer to simulate periodically a computation by a reversible Turing machine starting on an empty tape. The Turing machine computation is simulated on the first layer on a finite area. The configuration of the cellular automaton is divided into areas on each of which the Turing machine computation is simulated (on arbitrary input). The construction on the second layer is used to detect whether the Turing machine simulation was started on a specific initial state and an empty tape. The construction for the first layer is practically the same as in [13] where equicontinuity of reversible cellular automata was proven to be an undecidable property.

The third layer is used to forward information about the validity of the computation. That is, if the first layer does not contain periodically a simulation of a Turing machine computation started on an empty tape, a set of signals is generated to spread this information. The signals symbolizing a failed attempt to find a positive solution to the problem of Theorem 3.1 are restricted inside the particular simulation area of the Turing machine. Therefore, each of the simulation areas is in fact a blocking word sequence when layer 4 is not considered, because the contents of one simulation area are unaffected by the contents of the other simulation areas. The rule which is used on the third layer to draw the signals is determined locally by the contents of the first and second layer.

A *simulation error* is said to be present in a cell, if the contents of the first and second layer in the neighborhood of the cell are such that the Turing machine simulation should be considered failed in the search for a positive solution to the problem of Theorem 3.1. Then, on the third layer, a different local rule is used depending on whether the cell has a simulation error present or not.

The fourth layer contains three different kinds of signals. One of the signal types is used to prevent the other two kinds of signals from crossing a simulation area. However, if the simulation area contains a negative instance to the problem of Theorem 3.1, the signals used as border signals can be erased and the other two signal types can pass through the simulation area. Therefore, the cellular automaton is sensitive if, and only if, the problem of Theorem 3.1 has a negative answer. Eventually, the state set of the cellular automaton is $A_1 \times A_2 \times A_3 \times A_4$, where A_i denotes the state components of layer i.

3.3 Layer 1: representing Turing machine computation

Let Q be the state set of the Turing machine and let the initial state of the Turing machine be denoted by q_0. Let Γ denote the tape alphabet of the Turing machine. Then the cell states representing the read-write head reading a single tape letter are $\{\blacktriangle, \blacktriangledown\} \times Q \times \Gamma$. The cell states representing a single tape letter are $\{\triangleright, \triangleleft\} \times \Gamma$.

Symbols $\blacktriangle$ and $\blacktriangledown$ are used together with the state set Q to distinguish which rule, the original Turing machine or its inverse, is used in the local rule of the cellular automaton. Therefore, the state set representing the Turing machine on the first layer is

$$A_1 = (\{\blacktriangle, \blacktriangledown\} \times Q \times \Gamma) \cup (\{\triangleright, \triangleleft\} \times \Gamma).$$

Let sets $\{\triangleright\} \times \Gamma$ and $\{\triangleleft\} \times \Gamma$ be denoted by T_L and T_R, respectively. The elements of T_L are used to represent the tape contents to the left from the read-write head and the elements of T_R are used to represent the tape contents to the right from the read-write head. The elements of T_L and T_R will be called *left tape states* and *right tape states*, respectively. Let ε denote the empty tape symbol. Let H denote the set of states containing the read-write head.

By associating each tape letter with either expression $\triangleright$ or $\triangleleft$ the tape is divided into areas where each read-write head occurrence is followed by a certain number of elements of T_L to the left and a certain number of elements of T_R to the right from the read-write head itself. Therefore, the configuration is always partitioned into disjoint areas in each of which the Turing machine is simulated on some configuration. Formally, a *simulation area* in configuration c is a sequence of cell locations $(n)_{n=i}^{j}$, where $i \leq j$ and $c(n) \in T_L \Rightarrow c(n+1) \in T_L \cup H \cup T_R$, $c(n) \in H \Rightarrow c(n+1) \in T_R$ and $c(n) \in T_R \Rightarrow c(n+1) \in T_R$ for every n with $i \leq n \leq j-1$ and $c(i-1) \notin T_L$ and $c(j+1) \notin T_R$. With this definition, a domain which does not contain a state representing the read-write head is a simulation area. The rightmost cell of a simulation area is called the *right border* of the simulation area. Likewise, the leftmost cell of the simulation area is called the *left border* of the simulation area. If the simulation area contains only one cell (in which case it is in a state from H), the left border and the right border are the same cell.

By labelling the cell states representing the different sides of the Turing machine tape, the read-write heads are forced not to enter other simulation areas. If the read-write head moves to the left, it labels the previous cell to belong to the right side of the tape. Similarly, if the read-write head moves to the right, it labels the previous cell to belong to the left side of the tape. This way the simulation area maintains its constant width.

Expressions $\blacktriangle$ and $\blacktriangledown$ are used to denote the application of the Turing machine transition function as the local rule and the corresponding inverse operation. By using values $\blacktriangle$ and $\blacktriangledown$ together with the original states of the Turing machine, the cellular automaton becomes reversible even if the transition function of the Turing machine was only a partial function. Let $c \in A_1^{\mathbb{Z}}$ and $c(i) = (\blacktriangle, q, a)$. If the Turing machine move defined by pair (q, a) cannot be executed, state $(\blacktriangle, q, a)$ is replaced with state $(\blacktriangledown, q, a)$. That is, state component $\blacktriangle$ is replaced with $\blacktriangledown$ if $\delta(q, a)$ is undefined, $\delta(q, a)$ defines a left move but $c(i-1) \notin T_L$ or $\delta(q, a)$ defines a right move but $c(i+1) \notin T_R$. Similarly, if the inverse move cannot be executed, state $(\blacktriangledown, q, a)$ is replaced with state $(\blacktriangle, q, a)$. With this construction a halting computation with the Turing machine always leads to a periodic computation with the cellular automaton. In the peculiar case of the read-write head being located

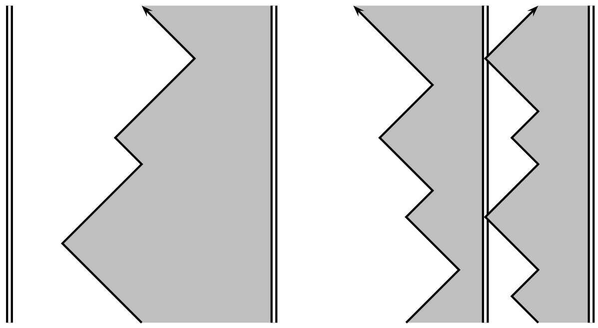


Figure 2: Labelling cells to represent either the left or the right side of the Turing machine tape. The zig-zag arrows represent read-write heads. The vertical double lines represent the borders between simulation areas.

between a right tape cell to the left and a left tape cell to the right, the read-write head state $(\blacktriangle, q, a)$ is constantly swapped with $(\blacktriangledown, q, a)$.

A read-write head is forced to be always present within a simulation area by defining the alternative cases to be simulation errors. That is, if a cell represents a left tape cell and the cell to its right represents a right tape cell, the cell is defined to have a simulation error. Similarly, if the cell represents a right tape cell and the cell to its left represents a left tape cell, the cell has a simulation error. With these constraints, a missing read-write head is always dealt with on the third layer.

3.4 Layer 2: verifying initial configuration

On the second layer, the validity of the Turing machine initial configuration is verified. If the Turing machine does not erase the tape periodically and re-enter the initial state, simulation errors are found present in the computation and they will affect the computation on the third layer.

The existence of the empty initial configuration is determined by defining signals which travel with twice the speed of the read-write head, that is, two cells per time step. These signals will be called *verification* signals. They simply bounce between the left border and the right border of the simulation area without ever passing from one simulation area to another. The greater speed of the signal is used to ensure that a verification signal intersects the path of the read-write head only once during a pass from the left side border to the right side border.

Let expressions $\blacktriangleright$ and $\blacktriangleleft$ denote a verification signal moving to the right and a verification signal moving to the left, respectively. The existence of the verification signal is forced by using states $\triangleright$ or $\triangleleft$. States $\triangleright$ and $\triangleleft$ tell whether the verification signal is located to the right or to the left from the cell, respectively. The idea is the same as on the first layer with the read-write head. A verification signal moving to the right can move only as far as there are states $\triangleleft$ to its right. The verification signal at location i

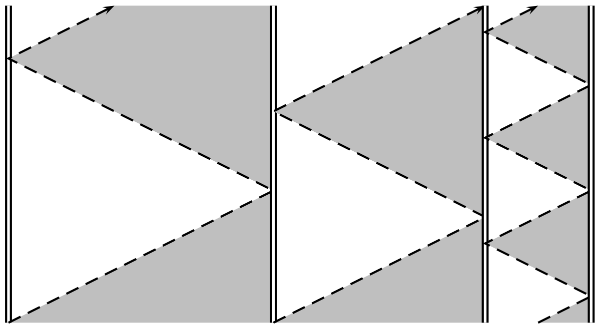


Figure 3: Drawing a verification signal to verify the existence of an empty tape configuration. The signal bounces between the borders of simulation areas.

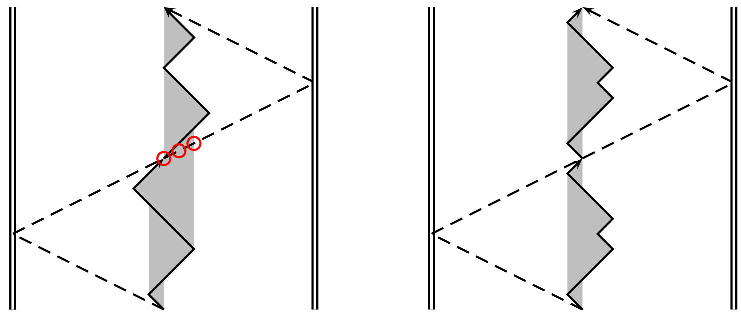
moving to the right (i.e. state $\blacktriangleright$) bounces back from cell i if cell $i+1$ has value $\triangleright$ or from cell $i+1$ if cell $i+1$ has value $\triangleleft$ and cell $i+2$ has value $\triangleright$. Similarly, a verification signal moving to the left can move only as far as there are states $\triangleright$ to its left. The verification signal at location i moving to the left (i.e. state $\blacktriangleleft$) bounces back from cell i if cell $i-1$ has value $\triangleleft$ or from cell $i-1$ if cell $i-1$ has value $\triangleright$ and cell $i-2$ has value $\triangleleft$. This way the configuration is divided into disjoint, consecutive and continuous areas on each of which a single verification signal bounces back and forth.

The state of a cell is changed from $\triangleleft$ to $\triangleright$ or from $\triangleright$ to $\triangleleft$ when the path of the verification signal crosses with the cell. Assume that a verification signal is contained in cell location i. Then the verification signal is replaced with state $\triangleright$ or $\triangleleft$, if the verification signal can move to the right or left, respectively. If the verification signal moves from cell location i to $i+2$, the state components of both cells i and $i+1$ are replaced with value $\triangleright$. Similarly, if the verification signal moves from cell location i to $i-2$, the state components of both cells i and $i-1$ are replaced with value $\triangleleft$.

Use of the verification signal is essentially the same method as dividing the configuration into simulation areas on the first layer according to left tape cells and right tape cells. Now the verification signal acts like the read-write head on the first layer changing the state component which points towards its location. The final state component set of the second layer is

$$A_2 = \{\triangleright, \triangleleft, \blacktriangleright, \blacktriangleleft\}.$$

To ensure that the areas where the verification signals bounce back and forth are located exactly the same way as the simulation areas, the alternative case is defined to be a source of simulation errors. This can be done by defining appearances of the state components pair $\triangleleft$ and $\triangleright$ to be simulation errors unless value $\triangleleft$ is located on the right border of a simulation area and value $\triangleright$ is located on the left border of another simulation area. If a right border and a left border to its right both contain either value $\triangleleft$ or $\triangleright$, it is considered a simulation error. It is considered a simulation error, if



(a) A simulation is executed unsuccessfully: The verification signal intersects the non-empty tape cells (denoted by circles) before the read-write head is able to erase them.

(b) A halting simulation is executed successfully: The Turing machine starts and halts periodically on an empty tape and the verification signal intersects the read-write head path only at the initial configuration.

Figure 4: Pairing the Turing machine simulation and the verification signal for the initial configuration. The solid arrow denotes the read-write head, the gray area denotes non-empty tape cells and the dashed arrow denotes the verification signal. The simulation area borders are denoted by vertical double lines.

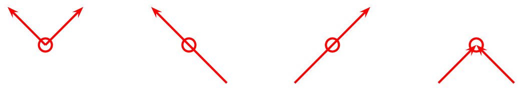
there are two verification signals side by side on top of the same simulation are. Hence, unless the domain on which a verification signal moves back and forth does not match a domain of a simulation area, simulation errors occur.

It is also considered a simulation error, if the path of a verification signal intersects with the path of a read-write head when the read-write head is not in the initial state. Likewise, it is considered a simulation error, if the verification signal intersects a cell which contains a different Turing machine tape symbol than the empty tape symbol ε. Therefore, simulation errors occur if the Turing machine simulation is not started on an empty tape with the read-write head being in the initial state.

3.5 Layer 3: detecting incorrect cell state combinations

The third layer is used to react on the simulation errors detected on the first two layers. This is done by introducing a new set of signals called *error signals*. An error signal travels one cell either to the left or right per one time step and it bounces back from a left or right simulation area border, respectively. An error signal always travels a straight line unless it collides with a simulation area border or there is a simulation error present in the cell. In a cell where there is a simulation error present, the propagation of error signals is determined according to a different rule. If there is no error signal entering a cell which contains a simulation error, then two error signals

(a) At a location where no simulation error is present, the error signals can intersect and propagate freely.



(b) At a location where simulation error is present, a single error signal can propagate freely but two error signals are either created or erased together.

Figure 5: The error signals are allowed to propagate freely if no simulation error is encountered. If a simulation error is encountered, a single signal can propagate freely but two signals are either created or erased.

are created, one of which travels to the left and another to the right. A single error signal passes through a cell with a simulation error unchanged. If two error signals enter a cell with a simulation error, both of them are erased. An outline of these different cases is shown in Figure 5. It is shown in Figure 5(a) how error signals propagate when no simulation error is present and in Figure 5(b) it is shown how error signals are modified when a simulation error is present.

As explained already in the sections describing layers 1 and 2, occurrences of the following cell state combinations are defined to be simulation errors:

1. A simulation area does not contain a state representing the read-write head, that is, at two adjacent cells belonging to the same simulation area the leftmost cell has binary value $\triangleright$ (representing a left tape state) and the rightmost cell has binary value $\triangleleft$ (representing a right tape state) on layer 1.

2. A read-write head collides with a simulation area border on layer 1.

3. A verification signal intersects the path of the read-write head when the tape is not empty or the read-write head is not in the initial state on layer 1.

4. Of two adjacent cells belonging to the same simulation area one has value $\triangleright$ and the other has value $\triangleleft$ for the binary component of layer 2. That is, either the verification signal is missing or the areas on the first two layers do not match.

5. Two verification signals are located in two neighboring cells on top of the same simulation area, that is, the areas on the two layers do not match.

6. The binary component of layer 2 has value $\lhd$ on the left border, that is, the areas on the two layers do not match.

7. The binary component of layer 2 has value $\rhd$ on the right border, that is, the areas on the two layers do not match.

At these locations the error signals are modified according to Figure 5(b).

The cellular automaton consisting only of layers 1, 2 and 3 is equicontinuous on all configurations containing only finitely long simulation areas. This follows from the fact that the contents of one simulation area do not affect another simulation area.

Using the error signals it would be possible to allow a new set of diagonally advancing signals to cross simulation area borders. That is, at the location where an error signal bounces back from a simulation area border, a signal of another type would be allowed to cross the border. However, this is not enough to remove the blocking property of multiple simulation areas next to each other. Namely, a signal crossing one simulation area border might always "strangely" bounce back from the border of the next simulation area where an error signal might not be present. This might happen, for example, if adjacent simulation areas contain the same Turing machine computation but in a different stage. For this reason, the blocking property of the simulation area border should be more controlled and it should not depend only on the error signals.

On the other hand, if reversibility was not a requirement, then the simulation area border could be permanently changed to a state which allows information pass through it. Therefore, in the irreversible case, layer 4 would not be needed in its current complexity.

Formally, the state component of the third layer can be presented by elements of

$$A_3 = \{\lozenge, \blacktriangleright, \blacktriangleleft, \blacklozenge\},$$

where different elements represent different combinations of left and right moving error signals.

Let $(A^{\mathbb{Z}}, F)$ be the cellular automaton consisting of layers 1, 2 and 3. That is, the state set is

$$A = A_1 \times A_2 \times A_3,$$

and the global rule is defined as described is sections 3.3, 3.4 and 3.5. Now the following theorem follows directly:

Theorem 3.2. *Given a non-sensitive reversible one-dimensional cellular automaton $(A^{\mathbb{Z}}, F)$ and a subset $E \subseteq A$ of its state set, it is undecidable whether or not there exists such an equicontinuity point $x \in A^{\mathbb{Z}}$ that $F^n(x)(0) \in A \setminus E$ for every $n \in \mathbb{N}$.*

Proof. Let E be the set of states that contain error signals. By the construction of layers 1, 2 and 3 it is undecidable whether all finite simulation areas will generate error signals. If error signals appear in a simulation area, then they necessarily visit every cell of the simulation area according to the rules in Figure 5. Therefore, if the reversible Turing machine does not halt, every cell in every finite simulation area will at some point contain error signals.

If the reversible Turing machine does halt, then a finite simulation area exists on which no error signals appear and this area can be chosen to overlap the origin.

An infinite simulation area might never contain an error signal if it consists of, say, tape states of one side only. However, a configuration which contains a one-way or two-way infinite simulation area cannot be an equicontinuity point because the contents of the infinite simulation area can be chosen to contain or not an error signal which travels through every cell of the simulation area. □

3.6 Layer 4: possible sensitivity

Now a new cellular automaton $(B^{\mathbb{Z}}, G)$ is constructed by adding a new layer to the cellular automaton $(A^{\mathbb{Z}}, F)$ of Theorem 3.2. The construction could be done explicitly by using the construction $(A^{\mathbb{Z}}, F)$ but this is not necessary.

The new layer consists of states from set A_4. That is, the state set A of $(A^{\mathbb{Z}}, F)$ is replaced with

$$B = A \times A_4 = A_1 \times A_2 \times A_3 \times A_4,$$

and the new global rule G is defined accordingly. The states $E \times A_4$ will be called *error states* (motivated by Theorem 3.2) in the new cellular automaton.

Layer 4 consists of adding three different types of signals on top of cellular automaton $(A^{\mathbb{Z}}, F)$. One of these signal types is a vertical signal, which will be used to either block or let the other two signal types to pass. The signals will be defined in such a way that a vertical signal can remain in a blocking state indefinitely if, and only if, the answer to the question of Theorem 3.2 is positive.

The three signal types used on the new layer will be called *border* signals and *activation* signals of *type 1* and *type 2*. A detailed description of the interaction of these signal types is shown in Figures 6 and 7. An active border signal is represented by a double vertical line and an inactive border signal is represented by a single vertical line. An activation signal of type 1 is represented by a dashed arrow and an activation signal of type 2 is represented by a dotted arrow.

A border signal is a signal which travels only vertically. It is represented by using a ternary component in every cell with one of the values $\lozenge$, $\square$ and $\blacksquare$. An absent border signal is represented by value $\lozenge$ whereas a present border signal is represented by values $\square$ and $\blacksquare$. The border signal is said to be *inactive* or *active* if the component has value $\square$ or $\blacksquare$, respectively. The main idea is that an active border signal blocks the propagation of activation signals indefinitely if the cell does not enter an error state. If a cell i containing a border signal is in such a state that it blocks activation signals, then a signal coming from the left bounces back from the cell i and a

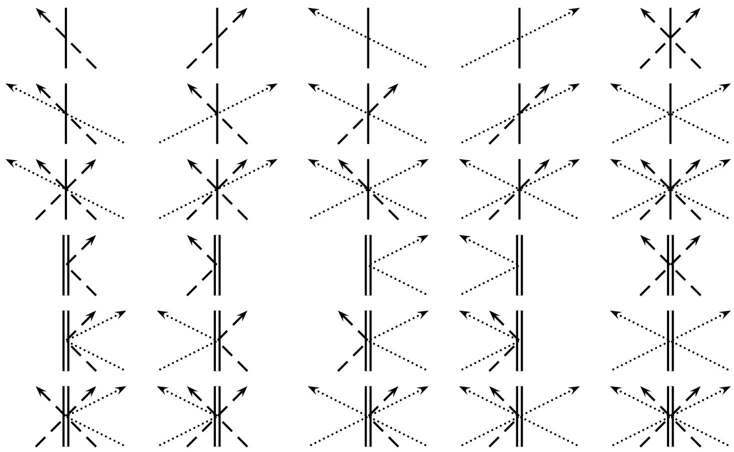


Figure 6: The interaction of different signal types when the cell containing the border signal is not in an error state.

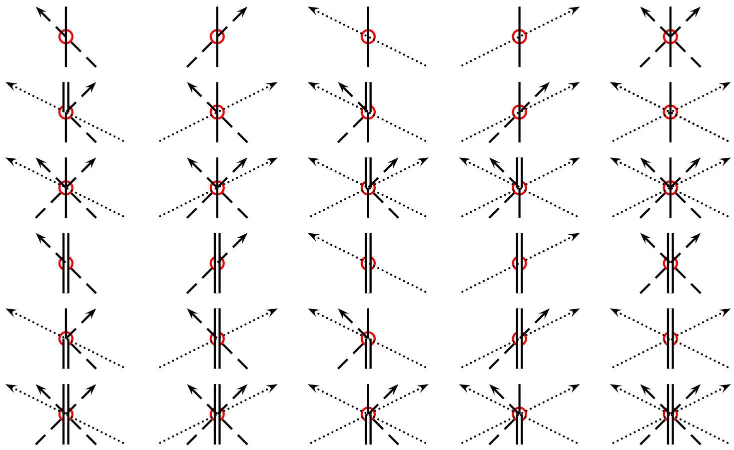


Figure 7: The interaction of different signal types when the cell containing the border signal is in an error state.

signal coming from the right bounces back from the cell $i+1$. For example, let two border signals be located in cells i and j with $i < j$ and assume that the signals are in an active state and the cells i and j do not enter error states. Then activation signals located in between the cells i and j will bounce back and forth between the cells $i+1$ and j.

An activation signal of type 1 is a signal which travels to the left or to the right one cell per one time step. An activation signal of type 2 is a signal which travels to the left or to the right two cells per one time step. The activation signals are used to change the states of border signals. The state of a border signal is changed if, and only if, the cell is in an error state and there is a single type 1 activation signal coming either from the left or from the right and a type 2 activation signal coming from the right. Let a cell in location i contain a border signal. If the cell i is in an error state and the cell $i+1$ contains a type 2 activation signal moving to the left, a type 1 activation signal moving to the right in the cell i bounces back from cell i and the state of the border signal is changed. If the type 1 activation signal is moving to the left in the cell $i+1$, it bounces back from cell $i+1$ and the state of the border signal is changed.

Formally, the state component of the fourth layer is expressed by elements of

$$A_4 = \{\Diamond, \square, \blacksquare\} \times \{\Diamond, \blacktriangleright, \blacktriangleleft, \blacklozenge\} \times \{\Diamond, \blacktriangleright, \blacktriangleleft, \blacklozenge\},$$

where the sets represent different signal types and their elements represent different combinations of a particular signal type.

A detailed description of the local rule for the signals can be found in Figures 6 and 7. The description of the local rule can be summarized as a following list of rules:

1. Any type of activation signal travels through an inactive border signal if no error state is present.

2. Any type of activation signal bounces off an active border signal if no error state is present.

3. A type 1 activation signal travels through any border signal if the cell is in an error state and no type 2 activation signal is coming from the right.

4. A type 2 activation signal always travels through any border signal if the cell is in an error state.

5. The state of the border signal is changed if, and only if, there is a single activation signal of type 1 coming either from the left or from the right and an activation signal of type 2 coming from the right. In this case the type 1 activation signal bounces back.

3.7 Undecidability

In this section it is concluded that the cellular automaton constructed in previous sections is sensitive to initial conditions if, and only if, the given reversible Turing machine does not halt on an empty tape. This follows from the fact that the contents of a simulation area which contains a halting Turing machine simulation can be used to construct a blocking word.

Recall that the cellular automaton $(B^{\mathbb{Z}}, G)$ was constructed in section 3.6 by adding an additional layer of signals to the cellular automaton $(A^{\mathbb{Z}}, F)$ of section 3.5.

Lemma 3.3. *If the question of Theorem 3.2 has a positive answer for the cellular automaton $(A^{\mathbb{Z}}, F)$, then the cellular automaton $(B^{\mathbb{Z}}, G)$ has a blocking word.*

Proof. Let $c_A \in A^{\mathbb{Z}}$ be such a configuration that $F^n(c_A)(0) \in A \backslash E$ for every $n \in \mathbb{N}$. Then, by the definition of an equicontinuity point, there exists such a positive integer k that the word $w = c_A[-k, k]$ is a blocking word and for every configuration $c \in A^{\mathbb{Z}}$ with $c[-k, k] = w$ condition $F^n(c)(0) \in A \setminus E$ holds.

Define word $w_{B,\blacksquare} \in B^*$ by setting

$$\begin{array}{ll} w_{B,\blacksquare}(i) = (w(i), \Diamond, \Diamond, \Diamond) & \text{if } 0 \leq i < k, \\ w_{B,\blacksquare}(i) = (w(i), \blacksquare, \Diamond, \Diamond) & \text{if } i = k \text{ and} \\ w_{B,\blacksquare}(i) = (w(i), \Diamond, \Diamond, \Diamond) & \text{if } k < i \leq |w| - 1. \end{array}$$

Now word $w_{B,\blacksquare}w_{B,\blacksquare}$ is a blocking word for the cellular automaton $(B^{\mathbb{Z}}, G)$. This follows from the fact that active border signals (i.e. values $\blacksquare$) in the centers of words $w_{B,\blacksquare}$ can never be changed inactive. □

Lemma 3.4. *If the question of Theorem 3.2 has a negative answer for the non-sensitive cellular automaton $(A^{\mathbb{Z}}, F)$, then for the cellular automaton $(B^{\mathbb{Z}}, G)$ and any word $u \in B^*$ there exists such a configuration $c \in \mathrm{Cyl}(u, 0)$ and such a positive integer t_u^+ that the configuration $G^t(c)$ does not contain any active border signals for any $t \geq t_u^+$.*

Proof. Let $w \in A^*$ be again a blocking word for the cellular automaton $(A^{\mathbb{Z}}, F)$. The blocking word exists because $(A^{\mathbb{Z}}, F)$ is non-sensitive. Define word $w_{B,\Diamond} \in B^*$ of the same length by setting

$$w_{B,\Diamond}(i) = (w(i), \Diamond, \Diamond, \Diamond)$$

whenever $0 \leq i < |w|$. The word $w_{B,\Diamond}$ is defined so that it does not contain any border signals.

Then the configuration $c \in B^{\mathbb{Z}}$ is constructed by first defining configuration c_n (where n is the number of border signals contained in the word u) so that

$$\begin{array}{ll} c_n(i) = w_{B,\Diamond}((i+1) \bmod |w_{B,\Diamond}|) & \text{if } i < 0, \\ c_n(i) = u(i) & \text{if } 0 \leq i < |u| \text{ and} \\ c_n(i) = w_{B,\Diamond}((i - |u|) \bmod |w_{B,\Diamond}|) & \text{if } |u| \leq i. \end{array}$$

That is, configuration c_n is such that word u is located in the origin and word $w_{B,\diamondsuit}$ repeatedly appears to the left and to the right of the occurrence of the word u. Moreover, all the border signals are located within domain $[0, |u|-1]$. Let the border signals be located in locations $b_1, b_2, \ldots, b_n$ where $b_i < b_{i+1}$, $0 \leq b_1$ and $b_n < |u|$. Also, every cell of c_n (and its modifications) will eventually enter an error state.

Second, the configuration c_n is modified iteratively so that all the border signals are changed inactive. Assume that c_k is such a configuration and t_k is such a time that $G^t(c_k)$ has neither active border signals nor activation signals in locations $(b_k, b_n]$, and further, all activation signals to the left of cell b_1 are moving to the left and all activation signals to the right of cell b_n are moving to the right for every integer $t \geq t_k$. That is, after t_k time steps b_k is the rightmost cell containing an active border signal. Furthermore, it is assumed that no activation signals pass through it to the right after time step t_k without a modification to the the initial configuration. Now the configuration c_k is modified (in two steps) further to produce configuration c_{k-1} where cell b_{k-1} contains the rightmost active border after $t_{k-1} > t_k$ time steps.

Recall that a border signal state is changed if, and only if, the cell containing the border signal is in an error state, there is a single type 1 activation signal coming either from the left or from the right and a type 2 activation signal coming from the right intersects the border signal at the same time.

1. Let $a > |u|$ be such an integer that $G^{t_k+a}(c_k)(b_k)$ is an error state. First, an activation signal of type 1 moving to the left is placed to the location $b_k + t_k + a + 1$ if, and only if, there is no type 1 signal coming from the left. That is, either an activation signal coming from the left is located in the cell b_k at time step $t_k + a$ or an activation signal coming from the right is located in the cell $b_k + 1$ at time step $t_k + a$.

   Second, an activation signal of type 2 is set to be located in the cell $b_k+2(t_k+a)+1$. These two different activation signals meet the border signal in cell b_k and together change it inactive. Integer a may need to be greater than $|u|$ to make sure that an added activation signal of any type does not hit any earlier stage border signals in cells b_i, where $i > k$, while they are still in an active state.

2. A band of e (i.e. sufficiently many) cells in the locations $[b_k + 2(t_k + a)+2, b_k+2(t_k+a)+1+e]$ are set not to contain any activation signals of type 2 moving to the left. Then the type 1 activation signal that was used to change the border signal in the cell b_k and all the activation signals contained between the border signals in the cells b_{k-1} and b_k and moving to the right (at time step $t_k + a$) can move beyond the last border signal in the cell b_n without changing the already inactive border signals back to active. Any activation signal moving to the left (at time step $t_k + a$) either crosses the border signal in cell b_{k-1} (when it changes inactive or enters an error state) while moving to the left

or collides with it and starts moving to the right. In any case, due to finiteness of u, in a finite number of time steps all the activation signals of type 1 or 2 that would cross the border signal in cell b_{k-1} while moving to the right have also done so. Within this time there should be no type 2 activation signals coming from the right.

Clearly, e can be chosen to be a finite positive integer because word u contains only finitely many activation signals bouncing back and forth between the border signals.

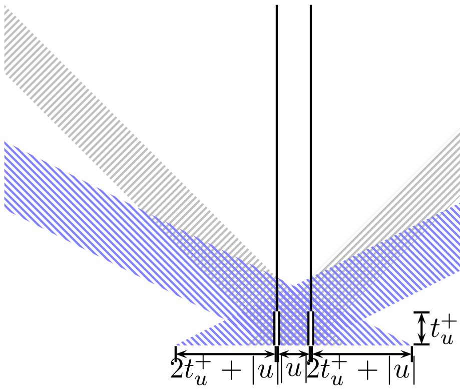
Let the modified configuration (produced from c_k) be denoted by c_{k-1}. Now, for some positive integer $t_{k-1} > t_k$ configuration $G^t(c_{k-1})$ contains neither active border signals nor activation signals in locations $(b_{k-1}, b_n]$, and further, all activation signals to the left of cell b_1 are moving to the left and all activation signals to the right of cell b_n are moving to the right for every integer $t \geq t_{k-1}$ and no activation signals meet with border b_{k-1} when it enters an error state. Now this iterative procedure is repeated n times to change the state of each border signal inactive and to allow enough time pass for all the activation signals to move beyond all the border signals. Eventually this procedure gives the configuration $c = c_0$.

Assume that the active border signals in the configuration c are erased in t_u^+ time steps. Then only $2t_u^+ + |u|$ cells to the left and right of word u contain activation signals. That is, the configuration c was chosen in such a way that the activation signals appear only in the locations shown in Figure 8(a). Then for some positive integer t_u^+ configurations $G^t(c)$ do not contain any active border signals for $t \geq t_u^+$. □

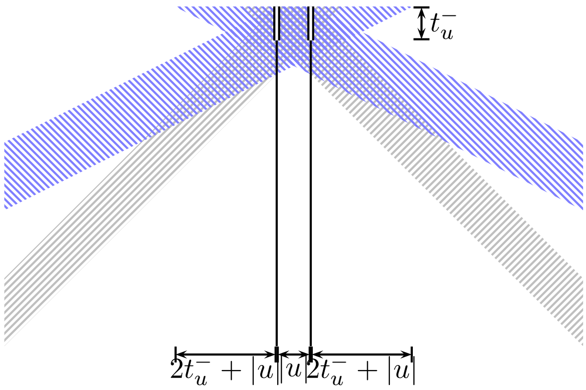
An example on the usage of the activation signals to change the border signals inactive is shown in Figure 9. The state of a border signal is changed if the cell containing the border signal enters an error state and there is a single activation signal of type 1 coming either from the left or from the right and an activation signal of type 2 coming from the right.

Because the signal interaction on layer 4 is almost identical for both the forward rule and the inverse rule, a similar lemma holds for the inverse rule also. The only difference is that because an activation signal of type 2 has the "activation property" only when it is coming from the right, the iterative process places the required signals to the left from the word occurrence instead of placing them to the right when working with the inverse rule. To say it more practically, the set of rules in Figure 7 is invariant with respect to a rotation by 180 degrees. For the inverse rule, the locations of the signals of layer 4 are shown in Figure 8(b).

Lemma 3.5. *If the question of Theorem 3.2 has a negative answer for the non-sensitive cellular automaton* $(A^{\mathbb{Z}}, F)$*, then for the cellular automaton* $(B^{\mathbb{Z}}, G)$ *and any word* $u \in B^*$ *there exists such a configuration* $c \in \mathrm{Cyl}(u, 0)$ *and such a positive integer* t_u^- *that the configuration* $G^{-t}(c)$ *does not contain any active border signals for any* $t \geq t_u^-$.



(a) All active border signals contained in u can be changed inactive after t_u^+ applications of the forward rule by choosing the contents of the $2t_u^+ + |u|$ cells to its left and to its right.



(b) All active border signals contained in u can be changed inactive after t_u^- applications of the inverse rule by choosing the contents of the $2t_u^- + |u|$ cells to its left and to its right.

Figure 8: If the question of Theorem 3.2 has a negative answer, all border signals contained in the word u can be changed inactive in a finite number of time steps. Double lines represent the locations between which active border signals may appear. Hash fill with positive slope and hash fill with negative slope represent the possible locations for activation signals of type 1 and type 2, respectively.

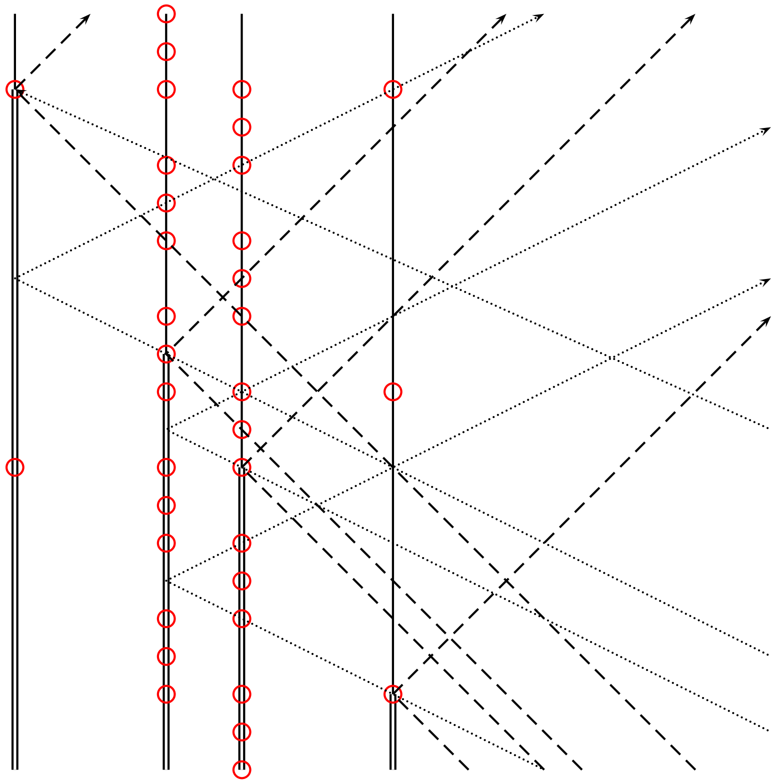


Figure 9: Any finite number of border signals can be changed to inactive state in which they can remain indefinitely long if every cell containing a border signal will enter an error state. Error states in the cells containing border signals are denoted by circles.

Theorem 3.6. *The cellular automaton $(B^{\mathbb{Z}}, G)$ is sensitive if, and only if, the question of Theorem 3.2 has a negative answer for the cellular automaton $(A^{\mathbb{Z}}, F)$.*

Proof. If the answer to the question of Theorem 3.2 is positive, then there exists an equicontinuity point for the cellular automaton $(A^{\mathbb{Z}}, F)$ with a cell that does not enter an error state ever. Then, according to Lemma 3.3, there exists a blocking word for the cellular automaton $(B^{\mathbb{Z}}, G)$. Hence, the cellular automaton $(B^{\mathbb{Z}}, G)$ is not sensitive.

If the answer to the question of Theorem 3.2 is negative, then the cellular automaton $(B^{\mathbb{Z}}, G)$ is sensitive by Lemma 3.4, because at some point border signals no longer block activation signals. □

Undecidability of sensitivity now follows from Theorem 3.2.

Corollary 3.7. *It is undecidable whether a given reversible one-dimensional cellular automaton is sensitive.*

4 Undecidability of topological mixing and transitivity

In this section it is described how the cellular automaton constructed in section 3 can be modified in such a way that it will be topologically mixing and topologically transitive if, and only if, the original cellular automaton (of section 3) is sensitive. The idea is to modify the cellular automaton in such a way that unless the original cellular automaton has a blocking word sequence, the contents of a simulation area (and actually any finite pattern in the original cellular automaton) can be shifted freely to the left and to the right. The shift effect is achieved by adding new states, called shift signals, that advance diagonally and do not affect the computation with the original states but act as a "filling" material. However, the computation with the original states does affect the propagation of the shift signals. To be precise, if a left shift signal encounters an active border signal (as defined in section 3.6), it is changed to a right shift signal. Similarly, if a right shift signal encounters an active border signal, it is changed to a left shift signal. Therefore, the modified cellular automaton (with shift signals) is mixing if, and only if, blocking words do not exist for the original cellular automaton $(B^{\mathbb{Z}}, G)$.

4.1 Shift signals

Let the cellular automaton constructed in section 3 be again denoted by $(B^{\mathbb{Z}}, G)$. The state set is modified in such a way that the modified cellular automaton is topologically mixing if, and only if, the original cellular automaton is sensitive.

First, the state set B is swapped to the cartesian product B^N for some $N > r$, where r is the radius of the original local rule g. Because type 2 activation signals move with a speed of two cells per one time step, the value of r is at least 2. The value of N is not fixed until the proofs of Lemmas 4.3 and 4.4 where the reason for using a certain value for N is also seen. The local rule is modified accordingly by considering the N-tuples of states to form a single configuration consisting of the original states in a natural, sequential way. In terms of symbolic dynamics, the cellular automaton, which is modified this way, could be denoted by $((B^N)^{\mathbb{Z}}, \gamma_N \circ G \circ \gamma_N^{-1})$ using the notation in [17]. The mapping γ_N rearranges the cell structure by replacing every N consecutive cells with a vector containing the states as elements.

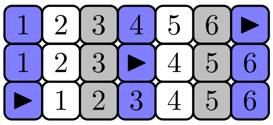
The goal of this change is to reduce the effective radius of the local rule. That is, the Nth iteration of the new local rule has radius r. In the modified cellular automaton N consecutive cells from the original cellular automaton $(B^{\mathbb{Z}}, G)$ occupy a single cell in the new cellular automaton.

Second, the state set B^N is extended with additional states ▶ and ◀. State ▶ represents a *left shift signal* and state ◀ represents a *right shift signal*. The left shift signals and the right shift signals, together called *shift signals*, are empty place holders which are used to shift the location of states belonging to the original state set. A left shift signal can travel three cells to the right per one time step (as shown in Figure 10(a)) and a right shift signal can travel three cells to the left per one time step (as shown in Figure 10(b)).

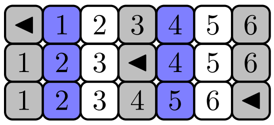
The shift signals are not allowed to appear on arbitrarily many consecutive cells. The state ◀ signifying a right shift signal can be located only on cells at locations i, where $i \bmod 3 = 1$. Similarly, the state ▶ signifying a left shift signal can be located only on cells at locations i, where $i \bmod 3 = 2$. Therefore, at least the cells in locations i, where $i \bmod 3 = 0$, are in states from the original state set. With these constraints, the original local rule can be used to compute the next configuration from the original states found between the shift signals. At least every third cell does not contain a shift signal and therefore the radius of the new local rule remains finite.

The collisions of the shift signals are defined differently from the description given in section 3.1. This follows from the fact that the shift signals form a disjoint subset of the state set and their locations are restricted. If a shift signal does not encounter an active border signal, it travels a straight path as shown in Figure 11(a). If a shift signal encounters an active border signal, it is swapped to a shift signal travelling to the opposite direction (as shown in Figure 11(b)). In other words, a shift travels a straight path if, and only if, it does not bounce back from an active border signal. If the shift signal encounters an active border signal, it is bounced back.

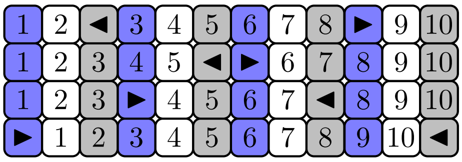
However, because shift signals are restricted to only certain locations whereas a border signal can be located anywhere, it needs to be clarified where a left signal is swapped to a right signal and vice versa. In short, a left shift signal that would intersect with an active border signal is replaced with a right shift signal in the first possible location to the left of the border signal. Similarly, a right shift signal that would intersect with an active



(a) A left shift signal is used to move states of the original state set to the left.



(b) A right shift signal is used to move states of the original state set to the right.



(c) The possible locations for the shift signals are restricted so that they can intersect while preserving the number of cells in the original states.

Figure 10: Use of the shift signals. Both shift signals work as a filling material moving elements of the original state set. For simplicity, the states of the original state set are denoted only by the numbers of their relative positions.

border signal is replaced with a left shift signal in the first possible location to the right of the border signal. The case of the left shift signal is shown in Figure 11(c). The collisions are defined in a similar way to right shift signals. If the shift signal is located between two active border signals whose distance is less than the shift signal's movement amount, then the new location and movement direction is determined repeatedly in a natural way.

To enforce the constraint on the locations of shift signals, the state set of the cellular automaton is further modified from B^N to

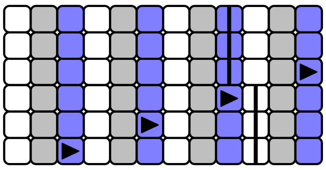
$$C = B^N \times \left(B^N \cup \{◄\}\right) \times \left(B^N \cup \{►\}\right)$$

and the new local rule is defined accordingly. Let the new global rule be denoted by H. Then the new cellular automaton is $(C^{\mathbb{Z}}, H)$, which was constructed by first joining consecutive cells to form N-tuples and second by adding the shift signals.

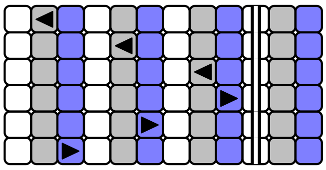
In terms of the original cellular automaton $(B^{\mathbb{Z}}, G)$, on every time step a shift signal shifts N states to the left or to the right by N cells.

4.2 Undecidability

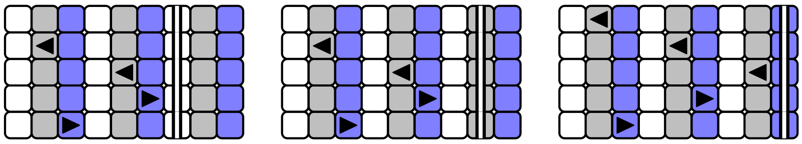
In this section it is shown that the cellular automaton constructed in previous sections is topologically mixing if, and only if, the given reversible Turing machine does not halt on an empty tape.



(a) A shift signal can travel through an inactive border signal.



(b) A shift signal bounces back from an active border signal.



(c) The different cases of a left shift signal bouncing back.

Figure 11: Shift signals travel a straight path if, an only if, they do not encounter active border signals.

If the given reversible Turing machine halts, it is possible to construct a blocking word which cannot be moved with the shift signals in the cellular automaton constructed in section 4.1. To be exact, if the Turing machine halts, there can exist a border signal which always remains in the active state. Then the shift signals simply bounce away from the border signal without moving it and the cellular automaton is not even sensitive.

If the Turing machine does not halt, no blocking word sequence exists and eventually all simulation areas can be moved to the left and to the right at will (by modifying the initial configuration). If the Turing machine does not halt, all the active border signals in a finite segment can be changed to inactive border signals. Once all the border signals are in inactive state, the contents of any finite segment can be shifted to the left or to the right by setting sufficiently many shift signal states to the initial configuration.

Lemma 4.1. *Suppose the question of Theorem 3.2 has a negative answer for the cellular automaton* $(A^{\mathbb{Z}}, F)$*. Then for the cellular automaton* $(C^{\mathbb{Z}}, H)$ *and any word* $u \in C^*$ *there exists such a configuration* $c \in \mathrm{Cyl}(u, 0)$ *and a positive integer* t_u^+ *that for any* $t \geq t_u^+$ *the configuration* $H^t(c)$ *does not contain any active border signals.*

Proof. The proof is similar to that of Lemma 3.4. The shift signals only disperse away from the location of the word u when all the active border signals have been changed inactive. □

Similarly, Lemma 4.2 follows.

Lemma 4.2. *Suppose the question of Theorem 3.2 has a negative answer for the cellular automaton* $(A^{\mathbb{Z}}, F)$*. Then for the cellular automaton* $(C^{\mathbb{Z}}, H)$ *and any word* $u \in C^*$ *there exists such a configuration* $c \in \mathrm{Cyl}(u, 0)$ *and a positive integer* t_u^- *that for any* $t \geq t_u^-$ *the configuration* $H^{-t}(c)$ *does not contain any active border signals.*

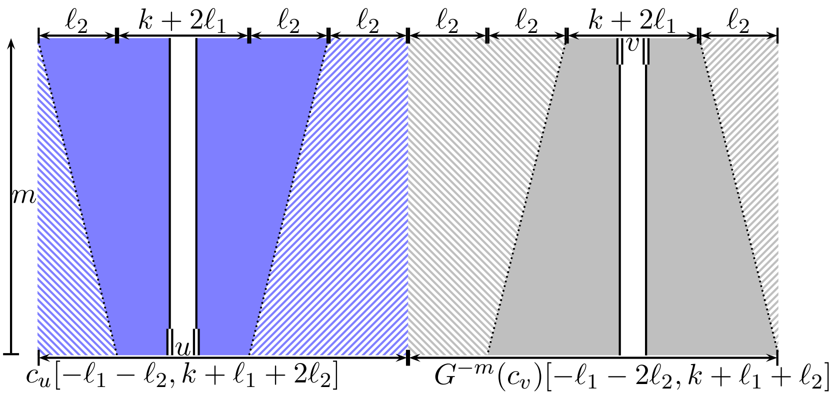


Figure 12: The configuration constructed in the proof of Lemma 4.3. The areas denoted by solid and line fill do not contain border signals. Activation signals are not found on the areas denoted by line fill.

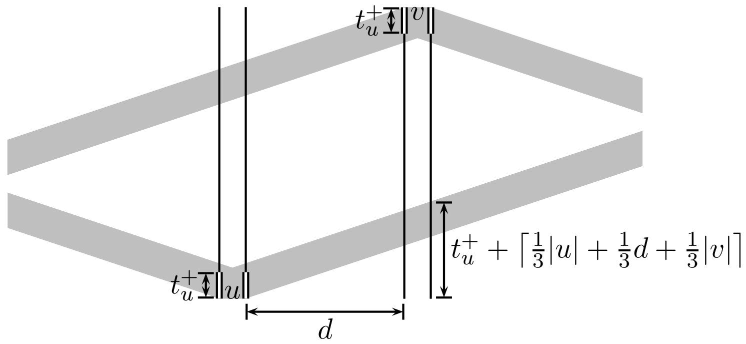


Figure 13: The shift signals originating from the word u pass through the future domain of word v in $t_u^+ + \lceil \frac{1}{3}|u| + \frac{1}{3}d + \frac{1}{3}|v| \rceil$ time steps. The solid gray areas denote the locations in which the shift signals may appear.

Lemma 4.3. *Let $u \in C^*$ and $v \in C^*$ be two words of equal length k and assume that the question of Theorem 3.2 has a negative answer for the cellular automaton $(A^{\mathbb{Z}}, F)$ so that the value $t = \max(t_u^+, t_v^-)$ exists. Then there exists such a positive integer m_0 and a configuration $c_m \in C^{\mathbb{Z}}$ for any positive integer $m \geq m_0$ that*

1. $c_m \in \mathrm{Cyl}(u, 0)$,
2. $H^m(c_m) \in \mathrm{Cyl}(v, 4\lceil t/N \rceil + 3k + 4\lceil rm/N \rceil)$ *and*
3. $H^i(c_m)$ *does not contain any active border signals for* $i \in [t, (m-t)]$.

The idea of the proof is to place the words u and v with a suitable distance so that the contents of one word does not affect the contents of another. That is, the activation signals dispersing away from word u do not have time to reach the position of word v in the "condensed" cell structure.

Proof. Let $\ell_1 = 2\lceil t/N \rceil + k$, $\ell_2 = \lceil rm/N \rceil$ and $\ell = \ell_1 + 2\ell_2$, where k is the length of words u and v and r is the radius of the cellular automaton $(B^{\mathbb{Z}}, G)$.

Multiple $N\ell_1 = 2N\lceil t/N \rceil + Nk$ gives an upper bound for the number of states from B which need to be redefined around the words to change the active border signals inactive. That is, ℓ_1 is the equivalent bound of the new cell structure to the bounds "$2t_u^+ + |u|$" and "$2t_u^- + |u|$" in Figures 8(a) and 8(b). The bound ℓ_2 is simply chosen in a suitable way to have enough space between the two words.

Let c_u and c_v be the configurations constructed with the method of the proof of Lemma 4.1 and its analogy for the inverse rule, respectively. Let $d = 2\ell_1 + 4\ell_2$. That is,

$$d = 4\lceil t/N \rceil + 2k + 4\lceil rm/N \rceil$$

and it is the distance between the domain of u and the future domain of v. Then, the configuration c_m is constructed by first setting

$$\begin{array}{ll} c_m(i) = c_u(i) & \text{if } i < k + \ell, \text{ and} \\ c_m(i) = G^{-m}(c_v)(i-d) & \text{if } k + \ell \leq i. \end{array}$$

Second, those shift signals which are to be eventually located within word v are placed to suitable locations in the initial configuration.

However, a shift signal exiting one word must not affect the formation of the second word. Therefore, the number of time steps m is bounded from below by equation

$$m \geq t_u^+ + \left\lceil \frac{1}{3}|u| + \frac{1}{3}d + \frac{1}{3}|v| \right\rceil + t_v^-$$

which comes from the fact that after $\lceil \frac{1}{3}|u| + \frac{1}{3}d + \frac{1}{3}|v| \rceil$ time steps the left shift signals (which have slope $\frac{1}{3}$) exiting the domain of word u have passed through the future domain of word v as shown in Figure 13. Because $d = 2\ell_1 + 4\ell_2$, it follows it would be sufficient to have condition

$$\begin{aligned} & t_u^+ + \left\lceil \frac{1}{3}k + \frac{1}{3}\left(2(2\lceil t/N \rceil + k) + 4(\lceil rm/N \rceil)\right) + \frac{1}{3}k \right\rceil + t_v^- \\ & \leq t + \frac{1}{3}k + \frac{1}{3}\lceil (4t/N + 2 + 2k + 4rm/N + 4) \rceil + \frac{1}{3}k + t + 3 \\ & \leq \frac{4}{3}k + \frac{1}{6}m + 3t + 6 \leq m, \end{aligned}$$

where $N = 8r$, hold. The previous inequality would hold if

$$m - \frac{1}{6}m \geq \frac{4}{3}k + 3t + 6$$

which would hold if $m \geq 2k + 3t + 6$. Therefore, it can be chosen that $m_0 = 2k + 3t + 6$.

Finally, no active border signals appear in the time window $[t, (m-t)]$ because the activation signals originally found in the configuration c_u do not have enough time to meet with the border signals located in the word

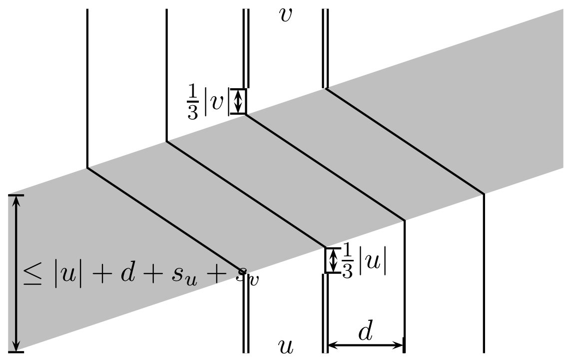


Figure 14: To have words u and v in the same position, at most $|u| + d + s_u + s_v$ left shift signals are required. This shift effect can be achieved in $|u| + d + s_u + s_v + \left\lceil \frac{1}{3} \max(|u|, |v|) \right\rceil$ time steps, where d is the distance between the occurrences of the word u and the word v without adding any shift signals to the initial configuration. Expressions s_u and s_v denote the number of shift signals contained in u and v, respectively.

v. Similarly, the signals that change the border signals inactive in c_v do not have enough time to meet with the border signals located in the word u. This follows from the fact that the distance even from the "seam" location to either one of the words is $N(\ell_1 + 2\ell_2) > 4m$ in terms of the old cell structure where a type 2 activation signal (i.e. the fastest signal that possibly matters) travels with a speed of two cells per one time step. In short, no active border signals are generated in other locations than in the domains of u and v and none on the interval $[t, (m - t)]$ because the words are chosen to appear far enough from each other. □

Lemma 4.4. *Let $u \in C^*$ and $v \in C^*$ be two words of equal length k and assume that the question of Theorem 3.2 has a negative answer for the cellular automaton $(A^{\mathbb{Z}}, F)$ so that the value $t = \max(t_u^+, t_v^-)$ exists. Then there exists such a positive integer n_0 and a configuration $c_n \in C^{\mathbb{Z}}$ for any positive integer $n \geq n_0$ that*

1. $c_n \in \mathrm{Cyl}(u, 0)$ *and*

2. $H^n(c_n) \in \mathrm{Cyl}(v, 0)$.

Proof. Let c be the configuration given by Lemma 4.3 and which therefore depends on integer $m = n$. The configuration c_n is constructed by modifying configuration c by adding shift signals between the states in c. Let $\ell_1 = 2\lceil t/N \rceil + k$, $\ell_2 = \lceil rn/N \rceil$ and $\ell = \ell_1 + 2\ell_2$, where k is the length of words u and v and r is the radius of the cellular automaton $(B^{\mathbb{Z}}, G)$. Let the number of shift signals contained in word u and v be s_u and s_v, respectively. The distance between the word occurrences is again $d = 2\ell_1 + 4\ell_2$.

An upper bound for the number s of left shift signals required to shift the word v to appear in the original domain of word u is given by condition

$$\begin{aligned} s &\le k + d + s_u + s_v \\ &= k + 2\ell_1 + 4\ell_2 + s_u + s_v \\ &= k + 2(2\lceil t/N \rceil + k) + 4\lceil rm/N \rceil + s_u + s_v \\ &\le k + 4t/N + 4 + 2k + 4rn/N + 4 + s_u + s_v \\ &\le 5k + 4t/N + 4rn/N + 8. \end{aligned}$$

The number s of shift signals is restricted only by equation

$$n \ge t + \left\lceil \frac{1}{3} k \right\rceil + s + t$$

which follows from the time window enforced by the appearance of the active border signals as shown in Figure 14. The coefficient $\frac{1}{3}$ follows from the fact that a left shift signal has slope $\frac{1}{3}$. However, the equation holds if

$$n \ge t + k + 5k + 4t/N + 4rn/N + 8 + t = 6k + 2t + 4t/N + 4rn/N + 8.$$

By fixing the constant N to have value $8r$ as already in the proof of Lemma 4.3, it follows from the previous equation that the shift signals can be used if $n \ge 12k + 6t + 16$ and if the constraint m_0 of Lemma 4.3 holds also. Now the bound n_0 is given by

$$n_0 = 12k + 6t + 8 \ge \max(12k + 6t + 16, 2k + 3t + 6).$$

For any $n \ge n_0$ the new configuration can be constructed by modifying the configuration given by Lemma 4.3 by adding sufficiently many left shift signals to the left of the cell in the location $-3t$. □

Theorem 4.5. *For reversible one-dimensional cellular automata, the sets of topologically mixing and non-sensitive cellular automata are recursively inseparable.*

Proof. Assume that the question of Theorem 3.2 has a negative answer for the cellular automaton $(A^{\mathbb{Z}}, F)$. Then, by Lemma 4.4, the cellular automaton $(C^{\mathbb{Z}}, H)$ is topologically mixing.

Assume that the answer to the question of Theorem 3.2 is positive. Then the blocking word constructed in the proof of Lemma 3.3 can be modified to be used in the new cell structure of the cellular automaton $(C^{\mathbb{Z}}, H)$. The word remains blocking because shift signals simply turn away from states in C containing active border signals from B as vector elements. Hence, the cellular automaton is not sensitive to initial conditions. □

Corollary 4.6. *The following dynamical properties are undecidable for reversible one-dimensional cellular automata:*

1. *sensitivity to initial conditions,*

2. topological mixing and

3. topological transitivity.

Because a reversible cellular automaton has dense periodic points, the cellular automaton of Theorem 4.5 is chaotic if, and only if, it is transitive. Because transitivity was shown to be undecidable in the reversible case, the undecidability of Devaney's chaos follows.

Corollary 4.7. *It is undecidable whether a given reversible one-dimensional cellular automaton is chaotic according to Devaney.*

A cellular automaton is transitive if, and only if, it has a dense orbit. Transitivity was seen to be an undecidable property, so undecidability of Knudsen's chaos follows. In particular, in the case of reversible cellular automata, Devaney's and Knudsen's definitions of chaos are equivalent.

Corollary 4.8. *It is undecidable whether a given reversible one-dimensional cellular automaton is chaotic according to Knudsen.*

5 Conclusions

It was shown that sensitivity to initial conditions, topological mixing and topological transitivity are undecidable properties for reversible one-dimensional cellular automata. The cellular automaton construction in the reduction was such that the cellular automaton is either topologically mixing or non-sensitive. Therefore the sets of topologically mixing cellular automata and non-sensitive cellular automata are recursively inseparable. It was previously known that sensitivity is undecidable for irreversible cellular automata. However, no undecidability results regarding topological mixing or topological transitivity were known previously even for irreversible cellular automata. Sensitivity, topological mixing and topological transitivity are known to be decidable for linear cellular automata over $\mathbb{Z}_m$ [3].

Because the set of periodic points of a reversible cellular automaton is dense, undecidability of Devaney's chaos followed from the simultaneous undecidability of transitivity and sensitivity. Because a one-dimensional cellular automaton is chaotic with respect to Knudsen's definition if, and only if, it is transitive, undecidability of Knudsen's chaos followed also.

6 Acknowledgements

The financial support by the Fund of Yrjö, Vilho and Kalle Väisälä is gratefully acknowledged. The author is grateful to Professor J. Kari for guidance, useful discussions and advices.

References

[1] C. H. Bennett. Logical reversibility of computation. *IBM Journal of research and development*, 6:525–532, 1973.

[2] F. Blanchard and P. Tisseur. Some properties of cellular automata with equicontinuity points. *Annales de l'Institute Henri Poincaré*, 36(5):562–582, 2000.

[3] G. Cattaneo, E. Formenti, G. Manzini, and L. Margara. Ergodicity, transitivity, and regularity for linear cellular automata over $\mathbb{Z}_m$. *Theoretical Computer Science*, 233:147–164, 2000.

[4] G. Cattaneo and L. Margara. Topological definitions of chaos applied to cellular automata dynamics. In *MFCS 1998*, volume 1450 of *Lecture Notes in Computer Science*, pages 816–824. Springer-Verlag, 1998.

[5] B. Codenotti and L. Margara. Transitive cellular automata are sensitive. *Amer. Math. Monthly*, 103:58–62, 1996.

[6] M. D'amico, G. Manzini, and L. Margara. On computing the entropy of cellular automata. *Theoretical Computer Science*, 290:1629–1646, 2003.

[7] R. L. Devaney. *An introduction to chaotic dynamical systems*. Addison-Wesley, 1989.

[8] B. Durand, E. Formenti, and G. Varouchas. On undecidability of equicontinuity classification for cellular automata. *Discrete Mathematics and Theoretical Computer Science*, pages 117–128, 2003.

[9] G. Hedlund. Endomorphisms and automorphisms of shift dynamical systems. *Math. Systems Theory*, 3:320–375, 1969.

[10] J. Kari. The nilpotency problem of one-dimensional cellular automata. *SIAM Journal on Computing*, 21:571–586, 1992.

[11] J. Kari. Reversibility and surjectivity problems of cellular automata. *Journal of Computer System Sciences*, 48:149–182, 1994.

[12] J. Kari. Theory of cellular automata: A survey. *Theoretical Computer Science*, 334:3–33, 2005.

[13] J. Kari and N. Ollinger. Periodicity and immortality in reversible computing. In *MFCS 2008*, volume 5162 of *Lecture Notes in Computer Science*, pages 419–430. Springer-Verlag, 2008.

[14] S. C. Kleene. A symmetric form of Gödel's theorem. *Indagationes Mathematicae*, 12:244–246, 1950.

[15] C. Knudsen. Chaos without nonperiodicity. *The American Mathematical Monthly*, 101:563–565, 1994.

[16] P. Kurka. Languages, equicontinuity and attractors in cellular automata. *Ergodic Theory and Dynamical Systems*, 17:417–433, 1997.

[17] D. Lind and B. Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.

[18] J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.

Turku
Centre *for*
Computer
Science

Lemminkäisenkatu 14 A, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematics

Åbo Akademi University
- Department of Computer Science
- Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration
- Institute of Information Systems Sciences

ISBN 978-952-12-2238-2
ISSN 1239-1891