Synchronous Boolean Finite Dynamical Systems and Minimum Circuit Size Problem

Mitsunori Ogihara[1,a)] Kei Uchizawa[2,b)]

Abstract: We study synchronous Boolean finite dynamical systems (synchronous BFDSs) consisting of some finite number of objects, where a local transition function on each object is chosen from a simple basis B. Specifically, we focus on the case where the basis B is one of {AND}, {OR} and {XOR, NXOR}. We show that, in these settings, the following two problems are randomized polynomial-time reducible to Minimum Circuit Size Problem: (i) Given an n-object synchronous BFDS $\mathcal{F}$ and two state configurations $\boldsymbol{a}$ and $\boldsymbol{b}$, do there exist time steps s and t, such that the state configuration of $\mathcal{F}$ on $\boldsymbol{a}$ at step s is equal to the state configuration of $\mathcal{F}$ on $\boldsymbol{b}$ at step t? (ii) Given an n-object synchronous BFDS $\mathcal{F}$, an initial state configuration $\boldsymbol{a}$, and an integer t, is the state configuration sequence generated by $\mathcal{F}$ starting from $\boldsymbol{a}$ contains a cycle having length greater than or equal to t?

1. Introduction

The finite dynamical system is a system consisting of some finite number of objects. The objects have initial state assignments, and their states are updated over discrete time by a local state-update functions that take as input the states of the objects in the system. The system has been used as a mathematical model for time-dependent systems and can contain in itself other multi-object computational models, such as cellular and graph automata and Hopfield networks. Consequently, the behavior of finite dynamical system receives much attention of researchers, and much work has been done to explore its behavioral properties ([3], [4], [5], [9], [10], [11]).

In this paper, among the various settings of the finite dynamical systems, we consider a class of finite dynamical systems, called *synchronous boolean finite dynamical systems* (synchronous BFDSs, for short), and investigate the computational complexity of its specific behavioral properties.

For any positive integer n, a synchronous BFDS of n objects is an n-tuple $\mathcal{F} = (f_1, f_2, \ldots, f_n)$ such that $f_1, \ldots, f_n$ are boolean functions of n variables. Let $\mathcal{B}$ be a finite set of basis functions. We say that $\mathcal{F}$ *has basis* $\mathcal{B}$ if each function of $\mathcal{F}$ is chosen from the basis $\mathcal{B}$ [*1].

For an n-object synchronous BFDS $\mathcal{F} = (f_1, f_2, \ldots, f_n)$, we define a *state configuration* (or simply a *configuration*) of $\mathcal{F}$ as an n-dimensional boolean vector. We use the vector notation $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ to denote a state configuration, where $x_1, \ldots, x_n$ are boolean variables. The action of $\mathcal{F}$ on an state configuration $\boldsymbol{x}$ is defined as:

$$\mathcal{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x}))$$

In other words, the elements of $\mathcal{F}(\boldsymbol{x})$ are obtained by applying the n boolean functions $f_1, \ldots, f_n$ concurrently on the variables $x_1, \ldots, x_n$. Given an initial state configuration $\boldsymbol{x}^0 = (x_1^0, x_2^0, \ldots, x_n^0)$, the synchronous BFDS defines n sequences of boolean values $\{x_i^t\}$, $1 \le i \le n$ and $t \ge 0$ by iterative applications of $\mathcal{F}$ on the initial state configuration vector:

$$\text{for all } t \ge 0,\ \boldsymbol{x}^{t+1} = \mathcal{F}(\boldsymbol{x}^t),$$

where for all $t \ge 0$, $\boldsymbol{x}^t = (x_1^t, x_2^t, \ldots, x_n^t)$.

In the paper [12], we consider specifically the bases $\mathcal{B}$ that are chosen from function families AND, NAND, OR, NOR, XOR, and NXOR, and study the computational complexities of the following three decision problems:

(1) Convergence($\mathcal{B}$): Given a system $\mathcal{F}$ and an initial state configuration $\boldsymbol{a}$, decide whether the system converges to any fixed point.

(2) PathIntersection($\mathcal{B}$): Given an n-object system $\mathcal{F}$ and two state configurations $\boldsymbol{a}$ and $\boldsymbol{b}$, do there exist time steps s and t, such that the state configuration of $\mathcal{F}$ on $\boldsymbol{a}$ at step s is equal to the state configuration of $\mathcal{F}$ on $\boldsymbol{b}$ at step t?

(3) CycleLength($\mathcal{B}$): Given a system $\mathcal{F}$, an initial state configuration $\boldsymbol{a}$, and an integer t, decide whether the state configuration sequence generated by the system starting from $\boldsymbol{a}$ contains a cycle having length greater than or equal to t. Note that the complement of this problem with $t = 2$ is Convergence($\mathcal{B}$).

We showed in [12] that the complexities of the three problems strongly depend on what functions $\mathcal{B}$ contains. More precisely, we prove that the three problems are each PSPACE-complete if the set $\mathcal{B}$ contains NAND, NOR or both AND and OR; but the

[1] Department of Computer Science, University of Miami 1365 Memorial Drive Coral Gables, FL 33146, USA
[2] Faculty of Engineering, Yamagata University, Jonan 4-3-16, Yonezawa, Yamagata, 992-8510 Japan
[a)] ogihara@cs.miami.edu
[b)] uchizawa@yz.yamagata-u.ac.jp

[*1] In general, the definition of bases allows mixing of a function family and a single Boolean function. See [12] for more detailed definition.

Convergence Problem is solvable in polynomial time, the Path Intersection Problem is in UP, and the Cycle Length Problem is in UP∩coUP if the set B is one of {AND}, {OR} and {XOR, NXOR}, which strongly suggests that these are unlikely to be even NP-hard in these cases (assuming P ≠ NP).

In this paper, we focus on the latter two problems, Path Intersection Problem and Cycle Length Problem, and strengthen the observation that these are unlikely to be NP-hard if the set B is one of {AND}, {OR} and {XOR, NXOR} by proving results that relate the problems to the Minimum Circuit Size Problem (MCSP) of Kabanets and Cai [8], where the MCSP asks the size of a smallest circuit that computes a given boolean function specified by the exact input-output table. More formally, we prove that Path Intersection Problem is in SZK, the class of languages having statistically zero-knowledge interactive proof system. This result implies that the problem is randomized polynomial-time reducible to the MCSP, since SZK $\subseteq \mathrm{RP}^{\mathrm{MCSP}}$ [2]. We also show that Cycle Length Problem is randomized polynomial-time reducible to MCSP. Since Kabanets and Cai [8] provide evidence that MCSP is unlikely to be NP-hard, e.g., in terms of strong circuit lower bounds of the linear exponential time E, the NP-hardness of these two problems would imply strong circuit lower bounds of E.

The rest of the paper is organized as follows: In Section 2.2, we formally define PathIntersection and CycleLength. We also give definitions of Minimum Circuit Size Problem and statistical zero-knowledge proof systems. In Section 3, we reduce Path Intersection Problem to MCSP. In Section 4, we reduce Cycle Length Problem to MCSP.

2. Preliminaries

2.1 Path Intersection Problem and Cycle Length Problem

For any n-state synchronous BFDS $\mathcal{F} = (f_1, f_2, \ldots, f_n)$, there are exactly 2^n possible state configurations. This implies that in an n-state synchronous BFDS, regardless of which initial state configuration $\boldsymbol{x}^0$ it starts, the state configuration sequence generated from $\boldsymbol{x}^0$ *enters a cycle*; that is, in the sequence there exist indices s and t, $0 \le s < t$, such that $\boldsymbol{x}^s = \boldsymbol{x}^t$. Clearly, for all such pairs (s, t), it holds:

$$\text{for all } i \ge 0,\ \boldsymbol{x}^{s+i} = \boldsymbol{x}^{t+i}.$$

Therefore, there is the smallest value of s for which there exists some $t > s$ such that $\boldsymbol{x}^s = \boldsymbol{x}^t$ and that, for that smallest value of s, there exists the smallest value of $t > s$ such that $\boldsymbol{x}^s = \boldsymbol{x}^t$. Let s_0 and t_0 respectively be the values of s and t thus defined. Then we have:

- $t_0 \le 2^n$ and
- for all i and j, $0 \le i < j \le t_0 - 1$, $\boldsymbol{x}^i \ne \boldsymbol{x}^j$.

We say that $\mathcal{F}$ on $\boldsymbol{x}$ *enters a cycle* (or *enters a loop*) at step s_0 and its cycle has *length* $t_0 - s_0$. We call s_0 the *tail length of* $\mathcal{F}$ *on* $\boldsymbol{x}$. We define $L_{\mathcal{F}}(\boldsymbol{x}^0)$ to be the length of the cycle $t_0 - s_0$.

We now formally define PathIntersection and CycleLength. Let $\mathcal{B}$ be a boolean function basis.

(1) PathIntersection($\mathcal{B}$) is the problem of deciding, given a synchronous BFDS $\mathcal{F}$ having basis $\mathcal{B}$ and two initial state configurations $\boldsymbol{a}$ and $\boldsymbol{b}$ of $\mathcal{F}$, whether there exist some s and t, $0 \le s, t \le 2^n - 1$, such that $\mathcal{F}^s(\boldsymbol{a}) = \mathcal{F}^t(\boldsymbol{b})$.

(2) CycleLength($\mathcal{B}$) is the problem of deciding, given a synchronous BFDS $\mathcal{F}$ having basis $\mathcal{B}$, an initial state configuration $\boldsymbol{a}$ of $\mathcal{F}$, and an integer t, whether the cycle length of $\mathcal{F}$ on $\boldsymbol{a}$, i.e., $L_{\mathcal{F}}(\boldsymbol{a})$, is greater than t.

The following lemma plays key role in our reductions presented in Section 3 and 4.

Lemma 1 ([12]). *Let $\mathcal{B}$ be one of* {AND}, {OR}, *and* {XOR, NXOR}. *Given an n-object synchronous BFDS $\mathcal{F}$ over basis $\mathcal{B}$, a state configuration $\boldsymbol{a} \in \{0,1\}^n$, and an integer $k \ge 0$, we can compute $\mathcal{F}^k(\boldsymbol{a})$ in time polynomial in $n + \log k$.*

2.2 Minimum Circuit Size Problem, statistical zero-knowledge interactive proof systems, and randomized reductions

We assume that the reader is familiar with introductory-level complexity classes (see, e.g., Hemaspaandra and Ogihara [7], for reference). The Minimum Circuit Size Problem (MCSP) [8] is defined as follows: Given a truth table of a k-variable Boolean function f (i.e., a string of n bits where $n = 2^k$) together with an integer m, decide whether f is computable by a boolean circuit of m or fewer gates. Kabanets and Cai [8] introduce the problem and show evidence that the problem is unlikely to be polynomial-time solvable and evidence that the problem is unlikely to be NP-hard. The papers [1], [2] show that several problems that are suspected to reside between P and NP, such as Graph Isomorphism and Factoring, are reducible to MCSP and show that SZK, the class of languages having statistically zero-knowledge interactive proof system, is probabilistic polynomial-time reducible to MCSP. Below we present this last result.

Let Π be a promise problem [6], that is, a membership problem defined over a polynomial-time decidable domain. Let Π_Y and Π_N be respectively be the set of Yes-instances and the set of No-instances of Π, where $\Pi_Y \cup \Pi_N \in \mathrm{P}$ and $\Pi_Y \cap \Pi_N = \emptyset$. A solution R to the promise problem Π satisfies: for all $x \in \Pi_Y \cup \Pi_N$, $x \in \Pi_Y$ if and only if $x \in R$.

Let S be a pair of probabilistic Turing machines P and V such that V is called the verifier and is is polynomial time-bounded and P is called the prover and is not polynomial time-bounded Given a common input x, P and V take turns in sending messages to each other. The computation lasts until V either accepts or rejects. Each machine computes its message based upon the input x, its internal computing history including the probabilistic choices, and the messages that have been so far sent between them. The pair (P, V) is said to be an *interactive proof system* for a promise problem $\Pi = (\Pi_Y, \Pi_N)$ if the following conditions are satisifed: (i) For all inputs x, the number of communication rounds is bounded by some fixed polynomial in $|x|$; (ii) For all $x \in \Pi_Y$, V accepts with probability at least $1 - 1/2^{|x|}$. (iii) For all $x \in \Pi_N$, for all probabilistic machines P^* including P, V accepts with probability at most $1/2^{|x|}$.

Given an interactive proof system (P, V), an input x to the system, and one computational path π of P and V on input x from the start of computation to an end (that is, the moment V either accepts or rejects x), consider a string that encodes the entire portion of the path visible to V. Such a string consists of the messages

exchanged between P and V and the probabilistic choices made by V and excludes the probabilistic choices made by P. We call this encoding the *view of V with P* of path π. Let $View_{(P,V)}(x)$ denote the random variable representing the views of V with P as the prover on input x.

An interactive proof system is said to be *a statistical zero-knowledge interactive proof system* if there exists a probabilistic polynomial-time machine S that simulates the views of V with P as the prover with high probability as follows: There exists a polynomial $p(n)$ such that for all $x \in \Pi_Y$, the output S of x, denoted by $S(x)$, satisfies:

$$\sum_y \left|\Pr[S(x) = y] - \Pr[View_{(P,V)}(x) = y]\right| \leq \frac{1}{p(|x|)},$$

We define SZK to be the class of all promise problems with satistical zero-knowledge interactive proof systems. A promise problem Π (or a decision problem Π) is is *randomized polynomial-time reducible* to a decision prblem Q, if there exists a randomized polynomial-time oracle Turing machine N such that for all x, if $x \in \Pi_Y$, N^Q on input x accepts with probability greater than or equal to $1/2$, and if $x \in \Pi_N$, N^Q on input x rejects with probability 1. We define RP^Q to be the set of all promise problems randomized polynomial-time reducible to Q.

In [2], Allender and Das prove that every problem in SZK is randomized polynomial-time reducible to MCSP.

Theorem 1. $\mathrm{SZK} \subseteq \mathrm{RP}^{\mathrm{MCSP}}$.

3. PathIntersection and MCSP

In this section, we show that PathIntersection($\mathcal{B}$) and CycleLength($\mathcal{B}$) are reducible to MCSP when $\mathcal{B}$ is one of {AND}, {OR}, and {XOR, NXOR}.

For PathIntersection, we prove that the problem is randomized polynomial-time reducible to MCSP.

Theorem 2. *Suppose $\mathcal{B}$ is one of* {AND}*,* {OR}*, and* {XOR, NXOR}*. Then* PathIntersection($\mathcal{B}$) *belongs to* $\mathrm{RP}^{\mathrm{MCSP}}$.

Theorem 2 clearly follows from Theorem 1 and the following lemma.

Lemma 2. *Let $\mathcal{B}$ be one of* {AND}*,* {OR}*, and* {XOR, NXOR}*. Then* PathIntersection($\mathcal{B}$) *belongs to* SZK.

Proof. Since SZK is known to be closed under complement [13], it suffices to show that the complement of PathIntersection($\mathcal{B}$) belongs to SZK.

Let $x = (\mathcal{F}, \boldsymbol{a}_0, \boldsymbol{a}_1)$ be an input whose membership in PathIntersection($\mathcal{B}$) is to be tested. We design the following protocol to show that the problem is in SZK: Let n be the number of variables in the system $\mathcal{F}$. We may assume that the encoding of the input has length no more than n^3; i.e., $|x| \leq n^3$.

The prover and the verifier repeat the Steps 1-3 below $2n^3$ times. The verifier accepts if and only if none of the repetitions lead to rejection.

Step 1 The verifier uniformly selects $z \in \{0, 1\}$, and an integer k, $1 \leq k \leq 2^{n+2n^2}$, and sends $\boldsymbol{b} = \mathcal{F}^k(\boldsymbol{a}_z)$ to the prover. Note that, by Lemma 1, $\boldsymbol{b}$ is polynomial-time computable.

Step 2 If there exists an integer k' such that $\boldsymbol{b} = \mathcal{F}^{k'}(\boldsymbol{a}_0)$, the prover sends $c = 0$ to the verifier; otherwise, the prover sends $c = 1$ to the verifier.

Step 3 If $c \neq z$, the verifier rejects immediately.

We show that the above protocol forms an interactive proof system.

First suppose that no pair (s, t) exists such that $\mathcal{F}^s(\boldsymbol{a}_0) = \mathcal{F}^t(\boldsymbol{a}_1)$. There is exactly one $z' \in \{0, 1\}$ such that for some $k', 0 \leq k' \leq 2^{n+2n^2}$, it holds that $\mathcal{F}^{k'}(\boldsymbol{a}_{z'}) = \boldsymbol{b}$. That unique value of z' should be equal to the value of z. The prover has only to compute this z' by using exhaustive search for example, and send it to the verifier, who must accept it upon receiving it it according to the protocol. Thus, in this case, the verifier can be made to accept with probability 1.

Next, suppose that there is a pair (s, t) such that $\mathcal{F}^s(\boldsymbol{a}_0) = \mathcal{F}^t(\boldsymbol{a}_1)$. This means that the cycle that the system enters when started from $\boldsymbol{a}_0$ is the same as the cycle that the system enters when started from $\boldsymbol{a}_1$. Let H be the set of all configurations appearing in this cycle and let $\ell = \|H\|$ be the length of the cycle. For $z \in \{0, 1\}$, let g_z be the smallest integer g such that $\mathcal{F}^g(\boldsymbol{a}_z) \in H$. We have that all of g_0, g_1, and ℓ are at most 2^n. Then, since the value of k is greater than 2^n, for each choice of c, by selecting the value of k according to the protocol, the verifier generates a distribution over H. For each $z \in \{0, 1\}$ and for each $h \in H$, let $q_z(h)$ be the probability that h is chosen as $\boldsymbol{a}_z$. Then for each z, $q_z(h)$ is either $2^{-n-n^2} \cdot \lfloor (2^{n+2n^2} - g_z)/\ell \rfloor$ or $2^{-n-n^2} \cdot \lceil (2^{n+2n^2} - g_z)/\ell \rceil$.

Since $0 \leq g_z \leq 2^n$ and $0 \leq \ell \leq 2^n$, we have

$$\begin{aligned}
2^{-n-2n^2} \cdot \lfloor (2^{n+2n^2} - g_z)/\ell \rfloor &\geq 2^{-n-2n^2} \cdot ((2^{n+2n^2} - g_z)/\ell - 1) \\
&\geq 2^{-n-2n^2} \cdot ((2^{n+2n^2} - 2^n)/\ell - 1) \\
&\geq 2^{-n-2n^2} \cdot (2^{n+2n^2} - 2^n - \ell)/\ell \\
&\geq 2^{-n-2n^2} \cdot (2^{n+2n^2} - 2^n - 2^n)/\ell \\
&= 2^{-n-2n^2} \cdot (2^{n+2n^2} - 2^{n+1})/\ell \\
&= 1/\ell - 2^{-2n^2+1}/\ell \\
&\geq 1/\ell - 2^{-2n^2+1}, \text{ and} \\
2^{-n-2n^2} \cdot \lceil (2^{n+2n^2} - g_z)/\ell \rceil &< 2^{-n-2n^2} \cdot ((2^{n+2n^2} - g_z)/\ell + 1) \\
&\leq 2^{-n-2n^2} \cdot (2^{n+2n^2}/\ell + 1) \\
&= 1/\ell + 2^{-n-2n^2} \\
&< 1/\ell + 2^{-2n^2+1}.
\end{aligned}$$

This gives that for all $h \in H$,

$$|q_0(h) - q_1(h)| < 2 \cdot 2^{-2n^2+1} = 2^{-2n^2+2}.$$

A strategy of a prover can be described as follows: given $h \in H$ select $c = 0$ with probability $\alpha(h)$ and $c = 1$ with probability $1 - \alpha(h)$, where $0 \leq \alpha(h) \leq 1$. Then the probability that the prover is able to guess the value of z correctly is:

$$\frac{1}{2}\left(\sum_{h \in H} q_0(h)\alpha(h) + q_1(h)(1 - \alpha(h))\right)$$

$$
\begin{aligned}
&= \frac{1}{2}\left(\sum_{h\in H} q_1(h) + (q_0(h) - q_1(h))\alpha(h)\right) \\
&= \frac{1}{2}\sum_{h\in H} q_1(h) + \frac{1}{2}\sum_{h\in H}(q_0(h) - q_1(h))\alpha(h) \\
&= \frac{1}{2} + \frac{1}{2}\sum_{h\in H}(q_0(h) - q_1(h))\alpha(h) \\
&\le \frac{1}{2} + \frac{1}{2}\sum_{h\in H}|q_0(h) - q_1(h)|\alpha(h) \\
&\le \frac{1}{2} + \frac{1}{2}\sum_{h\in H}|q_0(h) - q_1(h)| \\
&\le \frac{1}{2} + \ell \cdot 2^{-2n^2+2} \\
&\le \frac{1}{2} + 2^n \cdot 2^{-2n^2+2} \\
&\le \frac{1}{2} + 2^{-n^2}
\end{aligned}
$$

Since the protocol is repeated $2n^3$ times, the probability that the prover is able to guess correctly the value of z in all repetitions is no more than $(1/2 + 2^{-n^2})^{2n^3} < 2^{-n^3}$. Since the length of input encoding is no more than n^3, we have that the probability that the verifier accepts is less than $2^{-|x|}$ as desired. Thus, the problem has an interactive proof system.

To prove that this system is a zero-knowledge interactive proof system, consider a simulator that mimics the action of the verifier by selecting z and k accordingly and generates $c = z$ as the message from the provider. For every positive instance, the distribution of the view thus generated is identical to the distribution generated by the prover defined in the above. According to [13], this argument is sufficient to show that PathIntersection($\mathcal{B}$) is in SZK. □

4. CycleLength and MCSP

We here consider CycleLength. Although we do not know that CycleLength is in SZK, we can directly prove that the problem is reducible to MCSP following an argument similar to the ones in [1], [2].

The following theorem plays a key role in the reductions.

Theorem 3 ([1]). *Let L be a language of polynomial density such that, for some $\epsilon > 0$, for every $x \in L$, $KT(x) \ge |x|^\epsilon$. Let $f(y, x)$ be computable uniformly in time polynomial in $|x|$. There exists a polynomial-time probabilistic oracle Turing machine N and polynomial q such that for any n and y*

$$\Pr_{|x|=n,s}[f(y, N^L(y, f(y, x), s)) = f(y, x)] \ge \frac{1}{q(n)},$$

where x is chosen uniformly at random and s denotes the internal coin flips of N.

In [1], it is also shown that MCSP is the desired language L that satisfies the conditions given in the theorem.

Using Theorem 3, we now show that CycleLength belongs to $\mathrm{ZPP}^{\mathrm{MCSP}}$.

Theorem 4. *Suppose $\mathcal{B}$ is one of* {AND}, {OR}, *and* {XOR, NXOR}. *Then,* CycleLength($\mathcal{B}$) *belongs to* $\mathrm{ZPP}^{\mathrm{MCSP}}$.

Proof. Let $\mathcal{F}$ be a system and $\boldsymbol{a}$ be an initial configuration, as an input of CycleLength. We compute

$$\boldsymbol{w} = \mathcal{F}^{2^n}(\boldsymbol{a}).$$

Clearly, $\boldsymbol{w}$ is a configuration on the cycle originated from $\boldsymbol{a}$.

We define f as a function that takes as input $\mathcal{F}$, $\boldsymbol{x} \in \{0,1\}^n$ and an integer k and outputs the configuration $\mathcal{F}^k(\boldsymbol{x})$:

$$f(\boldsymbol{x}, k) = \mathcal{F}^k(\boldsymbol{x}).$$

By Lemma 1, f is computable in time polynomial in $\log k$. Thus, by Theorem 3, there exists a probabilistic oracle Turing machine N and polynomial q such that for any n and $\boldsymbol{x}$,

$$\Pr_{k,s}[f(\boldsymbol{x}, N^{\mathrm{MCSP}}(\boldsymbol{x}, f(\boldsymbol{x}, k), s)) = f(\boldsymbol{x}, k)] \ge \frac{1}{q(n)}, \tag{1}$$

where k is chosen uniformly at random from $0 \le k \le 2^n$, and s denotes the internal coin flips of N. By choosing h, $0 \le h \le 100 \cdot 2^n$, uniformly at random, we make $100q(n)$ independent executions of $N^{\mathrm{MCSP}}(\boldsymbol{w}, f(\boldsymbol{w}, h), s))$. The inequality (1) and the Chernoff bound imply that with high probability we can obtain a pair of integers h and h' such that $h \ne h'$ and

$$\mathcal{F}^h(\boldsymbol{w}) = \mathcal{F}^{h'}(\boldsymbol{w}) \tag{2}$$

Since we can check in polynomial time whether Eq. (2) holds, we can avoid errors. Let $z = |h' - h|$. Clearly, the cycle length $L_{\mathcal{F}}(\boldsymbol{a})$ is a factor of z.

Since Factoring is known to be in $\mathrm{ZPP}^{\mathrm{MCSP}}$ [1], we can obtain all the prime factors of z. Let $d_1, d_2, \ldots, d_m$ be the prime factors of z, which may not be pairwise distinct. Note that $m \le \log n$. We now apply the following test for z: For every i, $1 \le i \le m$, check whether $\mathcal{F}^{z/d_i}(\boldsymbol{w}) \ne \mathcal{F}^z(\boldsymbol{w})$ holds. If z passes the test, then z is clearly the desired cycle length, and so we output z. Otherwise, that is, there exists an index i' such that $\mathcal{F}^{z/d_{i'}}(\boldsymbol{w}) = \mathcal{F}^z(\boldsymbol{w})$, then we set $z = z/d_{i'}$, and apply the test for the new z with the factors other than $d_{i'}$. We repeat the procedure until z passes the test. Since $m \le \log n$, our entire algorithm runs in polynomial time, and thus the theorem follows. □

References

[1] E. Allender, H. Buhrman, M. Koucký, D. van Melkebeek, and D. Ronneburger. Power from random strings. *SIAM Journal on Computing*, **35**(6):1467–1493, 2006.

[2] E. Allender and B. Das. Zero knowledge and circuit minimization. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science*, pages 25–32, 2014.

[3] C. L. Barrett, H. S. Mortveit, C. M. Reidys. Elements of a theory of simulation II: Sequential dynamical systems. *Applied Mathematics and Computation*, **107**(2–3):121–136, 2000.

[4] C. L. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. Complexity of reachability problems for finite discrete dynamical systems. *Journal of Computer and System Sciences*, **72**(8):1317–1345, 2006.

[5] C. L. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, R. E. Stearns, and P. T. Tošić. Gardens of Eden and fixed points in sequential dynamical systems. In *Proceedings of Discrete Models: Combinatorics, Computation, and Geometry*, pages 95–110, 2001. American Mathematics Society.

[6] S. Even, A. L. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography, Information and Control, **6**(2): 159–173, 1984.

[7] L. A. Hemaspaandra and M. Ogihara. *A Complexity Theory Companion*. Springer-Verlag, 2001.

[8] V. Kabanets and J. Cai. Circuit minimization problem. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 73–79, 2000.
[9] S. Kosub. Dichotomy results for fixed-point existence problems for boolean dynamical systems. *Mathematics in Computer Science*, **1**(3):487–505, 2008.
[10] S. Kosub. and C. M. Homan. Dichotomy Results for Fixed Point Counting in Boolean Dynamical Systems. In *Proceedings of the Tenth Italian Conference on Theoretical Computer Science*, pages 163–174, 2007.
[11] R. Laubenbacher and B. Pareigis. Equivalence relations on finite dynamical systems. *Advances in Applied Mathematics*, **26**(3):237–251, 2001.
[12] M. Ogihara and K. Uchizawa. Computational Complexity Studies of Synchronous Boolean Finite Dynamical Systems. *Proc. of the 12th Annual Conference on Theory and Applications of Models of Computation*, pages 87–98, 2015.
[13] T. Okamoto. On Relationships between statistical zero-knowledge proofs. *Journal of Computer and System Sciences*, **60**(1): 47–108, 2000.