Available online at www.sciencedirect.com
ScienceDirect
Discrete Applied Mathematics 155 (2007) 2555–2566
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Hybrid one-dimensional reversible cellular automata are regular

Jesse Bingham[a,*], Brad Bingham[b]

[a] *Intel Corporation, JF2-04, 2111 NE 25th Avenue, Hillsboro, OR 97124, USA*
[b] *Department of Computer Science, University of British Columbia, 201-2366 Main Mall, Vancouver, B.C., Canada V6T 1Z4*

Received 13 September 2006; received in revised form 21 May 2007; accepted 6 July 2007
Available online 24 August 2007

Abstract

It is shown that the set of hybrid one-dimensional reversible cellular automata (CA) with the periodic boundary condition is a regular set. This has several important consequences. For example, it allows checking whether a given CA is reversible and the random generation of a reversible CA from the uniform distribution, both using time polynomial in the size of the CA. Unfortunately, the constant term in the resulting random generation algorithm is much too large to be of practical use. We show that for the less general case of null boundary (NB) CA, this constant can be reduced drastically, hence facilitating a practical algorithm for uniform random generation. Our techniques are further applied asymptotically to count the number of reversible NBCA.

Keywords: Reversible cellular automaton; Hybrid cellular automaton; Finite state automaton; Regular language

1. Introduction

Cellular automata (CA) are a class of discrete dynamical systems that have been applied to model a wide range of scientific phenomena, generate random data, perform computation, and many other applications [12,18,5,17]. A CA can be described as a set of *cells* embedded on a lattice, each of which can exist in a finite set of states. The system evolves by all cells updating their state according to some function of the states of the cells in a local neighborhood; this function is called the cell's *rule*. In this manner, a global configuration (i.e. mapping of cells to states) is taken to a successor configuration by the simultaneous update of all cells. This paper is concerned with *reversible* CA, which are CA having invertible global successor functions.

The class of CA we consider are called *hybrid*, meaning that each cell may employ a different rule for determining its next state. This contrasts with the *uniform* CA popularized by, e.g., Conway [4] and Wolfram [18], in which all cells use the same rule. Our CA are also characterized as one-dimensional (meaning the cells are embedded on a one-dimensional lattice), finite (referring to the number of cells), and nearest neighbor (meaning that each cell interacts with only its left and right neighbors). As our CA are finite and one-dimensional, they are naturally expressed as a finite string of rules.

In this paper, we show that the *reversible* CA form a regular set of strings. This result generalizes a previous result of Sarkar and Barua [15], which showed that hybrid CA involving only the two linear rules 90 and 150 are regular. We

[*] Corresponding author. Tel.: +1 503 264 4615; fax: +1 503 712 9121.
E-mail addresses: jesse.d.bingham@intel.com (J. Bingham), binghamb@cs.ubc.ca (B. Bingham).

0166-218X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2007.07.003

show that allowing *any* of the 256 possible nearest neighbor rules still preserves regularity. As in Sarkar and Barua's paper, we also show regularity holds regardless of which CA boundary condition is used.

Two important consequences of our regularity result are that there exists a linear time reversibility test algorithm, and that there exists a polynomial time algorithm that generates a random reversible CA from the uniform distribution. Unfortunately, only the former algorithm can be implemented in practice. This is because our regularity result is obtained by defining a nondeterministic automaton for the *complement* of the reversible CA, i.e. the language of irreversible CA. To obtain an automaton that accepts the reversible CA, we must determinize and complement, which causes an exponential blow up. The resulting automaton has $2^{(2^9)}$ states. This is not a show-stopper for the reversibility test algorithm, since it does not explicitly construct the entire automaton. On the other hand, the random generation algorithm must visit all $2^{(2^9)}$ states. In summary, the implied random generation algorithm may theoretically use polynomial time, but the hidden constant is astronomical.

To solve this problem, and hence produce a practical algorithm for uniform random generation, we focus on the null boundary (NB) condition. For this case, we show how the vast nondeterministic automaton for the more general periodic boundary (PB) condition can be reduced to a deterministic automaton having a mere nine states. This allows us to construct a *practical* algorithm that generates a random reversible CA from the uniform distribution. We have in fact implemented this algorithm; it generates a reversible CA with 200 cells in 90 sec on a contemporary laptop computer. The succinctness and determinism of the reduced automaton also allows us to *count* the number of NB reversible CA. We show that the number of such CA with n cells is $\Theta(\lambda^n)$, where $\lambda \approx 17.98$ (whereas there are 256^n possible CA with n cells).

The paper is organized as follows. In Section 2 we further touch on related work. Section 3 lays out the definitions and terminology used throughout the paper. Section 4 shows our main result on the regularity of PB reversible CA. Section 5 develops a greatly reduced automaton for NB reversible CA, which allows for a practical random generation algorithm in Section 6 and also for counting these CA in Section 7. The paper is concluded in Section 8.

2. Related work

A special class of CA involving only *linear* rules lend themselves to algebraic analysis. Linear rules are those that can be expressed as a modulo-2 sum (i.e. XOR logic). The global transition function of linear CA can be expressed as a matrix over the Galois field of two elements and is invertible if and only if this matrix is non-singular. *Additive* rules generalize linear rules somewhat in that they allow negation; these are explored thoroughly in the book of Pal Chaudhuri et al. [12].

Our results generalize previous results of Sarkar and Barua [15], who show that the set of reversible CA over the two linear rules 90 and 150 is regular, for both NB and PB conditions. Sarkar and Barua observe that the determinant of the transition matrix can be expressed by a multi-variate polynomial known as a *continuant*, which admits a recursive definition. Using this recursive definition, they obtain an inductive definition of all (90, 150) CA that yield a non-zero determinant, and hence are reversible. We note that this approach depends on the linearity of the rules 90 and 150; since we allow for nonlinear rules, we require a different technique. Sarkar and Barua also show that roughly 2/3 (resp. 1/3) of all null (resp. periodic) boundary CA with these two rules are reversible. Interestingly, it follows from our result of Section 7 that the ratio of reversible CA with *any* of the 256 nearest neighbor rules vanishes as n increases.

Our CA can be thought of as imposing the restriction on general n-bit finite state machines that the next state of each bit only depends on itself and its two neighbors. Another natural restriction yields the *feedback shift register* (FSR). Results pertaining to reversibility of FSR have been covered by Golomb [6]. In particular, it is shown that of the $2^{(2^n)}$ possible FSR, $2^{(2^{n-1})}$ are reversible. Golomb also provides several necessary and sufficient conditions for FSR reversibility.

There has been a wealth of work on reversibility of uniform, infinite CA (UICA). An early result by Moore [10] and Myhill [11] is the *Garden-of-Eden Theorem*, which states that UICA is surjective if and only if it is injective when restricted to finite configurations. Another important result is due to Richardson [14], and states that if the global successor function of a UICA is injective, then it must also be bijective (i.e. reversible). Amoroso and Patt have given an algorithm that decides reversibility of the (unique) one-dimensional UICA that uses a given rule [1]. Their algorithm, though not described in the language of automata theory, is similar to ours in that it essentially runs a finite state automaton (FSA) that attempts to construct distinct configurations that have the same successor. Our automata are more complex in a sense, since they must deal with different rules at each step. Also, for the case of PBCA, our automaton $\mathfrak{A}_{\mathsf{ica}}$ of Section 4 must remember some information about the left end of the configurations, so that it can check that they "wrap around" appropriately at the right end. Kari has shown that decidability of UICA reversibility does not generalize to higher dimensions; even for two dimensions, the problem becomes undecidable [8]. More recently, Sutner has shown that the *reachability* problem for one-dimensional reversible UICA can assume any recursively enumerable degree [16]. Here, reachability is the decision problem that asks, given two finite configurations, x and y, if the UICA ever reaches y when started in x. Toffoli and Margolus [17] provide an overview of results relating to reversible UICA and their connection to physics.

We conclude this related work section by noting that our automata constructions are rooted in the linear time sink algorithm of the first author's masters thesis [2]. Here a *sink* is a contiguous group of CA cells that can potentially get "stuck" in constant states as the CA evolves.

3. Preliminary definitions

For any set A, we let A^* denote the set of all finite strings over A, and for any $n \geqslant 0$, we let A^n denote the set of all strings of length n in A^*. For $x \in A^n$, we denote by x_i the ith symbol in x where x_1 is the first symbol, i.e. x may be written as $x_1, \ldots, x_n$ where n is the length of x. Let $\mathbb{B} = \{0, 1\}$ denote the bits. We will typically use lowercase letters from the beginning of the alphabet (a, b, etc.) to denote elements of $\mathbb{B}$ and lowercase letters from the end of the alphabet (w, x, etc.) to denote strings of $\mathbb{B}^*$.

A function $r : \mathbb{B}^3 \to \mathbb{B}$ is called a *rule*. We will use r to denote a single rule, and ρ to denote a string of rules. It is traditional to identify a rule r with its *rule number*, which is defined by $r(111)2^7 + r(110)2^6 + \cdots + r(000)2^0$. The rule number is always in the range $\{0, \ldots, 255\}$.

A *one-dimensional nearest neighbor hybrid*[1] cellular automaton, hereafter simply *CA*, is a nonempty finite string of rules $\rho_1, \ldots, \rho_n$. The length of this string n is called the CA's *size*. A *configuration* of a CA of size n is an element of $\mathbb{B}^n$. Intuitively, a size n CA ρ can be thought of as linear arrangement of n *cells*, each storing a single bit; the bit at each cell is determined by the current configuration. Time evolves discretely and at each time step the cells are all updated in parallel as follows. If x is the current configuration, the next configuration is such that the ith cell gets the bit $\rho_i(x_{i-1}x_i x_{i+1})$. Thus, cells $i-1$ and $i+1$ are considered *neighbors* of cell i (as is cell i itself), since they are the only cells that can influence the next bit at cell i. An obvious issue here is how one deals with the non-existent cells 0 and $n+1$. One may either wrap the CA so that cells 1 and n become neighbors, resulting in the PB condition, or cells 0 and $n+1$ can be treated as being constantly the bit 0, resulting in the NB condition. Either approach gives rise to a *successor function*, which determines how the rules are used to map a CA configuration to a successor configuration.

Definition 1 (*PB successor function η*). Given a *CA* $\rho_1, \ldots, \rho_n$ where $n \geqslant 2$, the *PB successor function* $\eta_\rho : \mathbb{B}^n \to \mathbb{B}^n$ is defined by $\eta_\rho(x) = y$ where, for each $1 \leqslant i \leqslant n$,

$$y_i = \begin{cases} r_i(x_{i-1}x_i x_{i+1}) & \text{if } 1 < i < n, \\ r_1(x_n x_1 x_2) & \text{if } i = 1, \\ r_n(x_{n-1}x_n x_1) & \text{if } i = n. \end{cases}$$

Definition 2 (*NB successor function η^0*). Given a *CA* $\rho_1, \ldots, \rho_n$ where $n \geqslant 2$, the *NB successor function* $\eta^0_\rho : \mathbb{B}^n \to \mathbb{B}^n$ is defined by $\eta^0_\rho(x) = y$ where, for each $1 \leqslant i \leqslant n$,

$$y_i = \begin{cases} r_i(x_{i-1}x_i x_{i+1}) & \text{if } 1 < i < n, \\ r_1(0 x_1 x_2) & \text{if } i = 1, \\ r_n(x_{n-1}x_n 0) & \text{if } i = n. \end{cases}$$

In this paper we are concerned with a certain class of CA called *reversible*. A CA ρ is said to be *PB reversible* (resp. *NB reversible*) if η_ρ (resp. η^0_ρ) is a permutation. If ρ is not reversible, then there must exist distinct configurations x and y having the same successor; in this case we call the pair (x, y) *irreversibility witnesses* for ρ, qualifying with "PB" or "NB" to indicate which configuration successor function is under consideration.

[1] a.k.a. *nonuniform*.

A rule r is called *balanced* if it has the same number of 0's and 1's in its truth table; i.e. $|r^{-1}(1)| = |r^{-1}(0)| = 4$. We let R denote the set of all balanced rules; by simple counting we find that $|R| = \binom{8}{4} = 70$. As the following lemma asserts, a CA can be PB reversible only if all its rules are balanced.

Lemma 1. *If CA ρ is PB reversible, then all its rules are balanced.*

Proof. Suppose rule ρ_i is not balanced; without loss of generality let us assume that ρ_i has k 1's in its truth table where $4 < k \leqslant 8$. It follows that for exactly $k2^{n-3} > 2^{n-1}$ configurations x, the ith component of $\eta_\rho(x)$ is 1. Since there are only 2^{n-1} configurations having ith component 1, there must exist two distinct configurations y and z such that $\eta_\rho(y) = \eta_\rho(z)$. □

The analogous result for NB reversibility does not quite hold. If a $\rho_1, \ldots, \rho_n$ is NB reversible, it follows that rules $\rho_2, \ldots, \rho_{n-1}$ must be balanced, but ρ_1 and ρ_n need not be. Nevertheless, for consistency, we will from here on only consider CA over the balanced rules R; i.e. we are effectively redefining the notion of CA from being an arbitrary nonempty string of rules to being a nonempty element of R^*. We make this restriction as it makes some of our constructions cleaner; however, none of our results depend fundamentally on it.

We conclude this section by noting that the NB successor function can be viewed as a special case of the PB successor function as follows. For any CA $\rho_1, \ldots, \rho_n$ we have that η^0_ρ is equal to $\eta_{\rho'}$, where ρ' is the same as ρ except that the first and last rules are tweaked to be independent of their missing neighbors. Indeed, one *could* define a NBCA to be any sequence of rules such that the first (resp. last) are independent of their left (resp. right) arguments. We found it convenient to use our successor function definitions instead.

4. Regularity of PB reversible CA

In this section, we show that the set of PB reversible CA is a regular language. To this end, we define an FSA $\mathfrak{A}_{\mathsf{ica}}$ that accepts the language of PB irreversible CA. Since regular languages are closed under complement, this demonstrates our claim. We start by recalling some basic definitions from automata theory.

An FSA is a 5-tuple $A = (S, \Sigma, \delta, I, F)$ where S is a finite set of *states*, Σ is a finite *alphabet*, $\delta \subseteq S \times \Sigma \times S$ is called the *transition relation*, $I \subseteq S$ are the *initial states*, and $F \subseteq S$ are the *accepting states*. Given a string $w_1, \ldots, w_\ell \in \Sigma^*$, a *run* of A (on w) is a sequence $s_0, \ldots, s_\ell$ of states such that $s_0 \in I$ and $(s_{i-1}, w_i, s_i) \in \delta$ for all $1 \leqslant i \leqslant \ell$. A run is called *accepting* if its final state is in F. The *language* of A, denoted by $L(A)$, is the set of all strings in Σ^* that have accepting runs. We say that A is *deterministic* if $|I| = 1$ and for all $s \in S$ and $\sigma \in \Sigma$ there exists exactly one $s' \in S$ such that $(s, \sigma, s') \in \delta$; otherwise A is called *nondeterministic*. When A is deterministic, we will treat δ as a function $\delta : S \times \Sigma \to S$.

We define a nondeterministic FSA $\mathfrak{A}_{\mathsf{ica}} = (S_{\mathsf{ica}}, R, \delta_{\mathsf{ica}}, I_{\mathsf{ica}}, F_{\mathsf{ica}})$ such that $L(\mathfrak{A}_{\mathsf{ica}})$ is precisely the *irreversible* periodic CAs. $\mathfrak{A}_{\mathsf{ica}}$ has states of the form (v, ab, cd, ef, gh), where v, a, b, c, d, e, f, g, and h are all bits. Intuitively, when $\mathfrak{A}_{\mathsf{ica}}$ is in the state (v, ab, cd, ef, gh), this means that the string of rules $\rho_1, \ldots, \rho_\ell$ seen so far are such that there exists a bit-string $x_0, \ldots, x_{\ell+1}$ with $x_0 x_1 x_\ell x_{\ell+1} = abcd$ and a bit-string $y_0, \ldots, y_{\ell+1}$ with $y_0 y_1 y_\ell y_{\ell+1} = efgh$ that make all the ρ_i's agree, i.e. $\rho_i(x_{i-1} x_i x_{i+1}) = \rho_i(y_{i-1} y_i y_{i+1})$ for all $i \in 1, \ldots, \ell$. Furthermore, if $v = 1$, then $x_0, \ldots, x_{\ell+1}$ and $y_0, \ldots, y_{\ell+1}$ are known to have differed in at least one bit. In short, $\mathfrak{A}_{\mathsf{ica}}$ attempts to construct PB irreversibility witnesses for the string of rules it receives as input.

Formally, the components S_{ica}, δ_{ica}, and F_{ica} are as follows:

- $S_{\mathsf{ica}} = \mathbb{B} \times (\mathbb{B}^2)^4$.
- δ_{ica} is the set of all transitions of the form

$$((v, ab, cd, ef, gh), r, (v', ab, dd', ef, hh'))$$

such that $r(cdd') = r(ghh')$ and

$$v' = \begin{cases} 1 & \text{if } cdd' \neq ghh', \\ v & \text{otherwise.} \end{cases}$$

- $I_{\mathsf{ica}} = \{(0, ab, ab, ef, ef) : a, b, e, f \in \mathbb{B}\}$.
- $F_{\mathsf{ica}} = \{(1, ab, ab, ef, ef) : a, b, e, f \in \mathbb{B}\}$.

Our key result is now proven in Theorem 1; its supporting Lemma 2 below formalizes the above intuition.

Theorem 1. $\rho \in L(\mathfrak{A}_{\mathsf{ica}})$ *if and only if* ρ *is PB irreversible.*

Proof. ($\Rightarrow$) Let $s_0, \ldots, s_n$ be an accepting run of $\mathfrak{A}_{\mathsf{ica}}$ on $\rho_1, \ldots, \rho_n$ and let $(v_i, a_ib_i, c_id_i, e_if_i, g_ih_i) = s_i$. Since $v_n = 1$, there exists $x_0, \ldots, x_{n+1}, y_0, \ldots, y_{n+1} \in \mathbb{B}$ satisfying the three conditions of Lemma 2. Since $s_n \in F_{\mathsf{ica}}$, we have that $a_nb_n = c_nd_n$ and $e_nf_n = g_nh_n$, and hence $x_0x_1 = x_nx_{n+1}$ and $y_0y_1 = y_ny_{n+1}$. These observations and the third item of Lemma 2 yield $\rho_1(x_nx_1x_2) = \rho_1(y_ny_1y_2)$, $\rho_n(x_{n-1}x_nx_1) = \rho_n(y_{n-1}y_ny_1)$, and $\rho_i(x_{i-1}x_ix_{i+1}) = \rho_i(y_{i-1}y_iy_{i+1})$ for all $i \in \{2, \ldots, n-1\}$. This implies that $\eta_\rho(x_1, \ldots, x_n) = \eta_\rho(y_1, \ldots, y_n)$, and hence ρ is irreversible.

($\Leftarrow$) Suppose ρ is irreversible, and let $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ be CA configurations such that $x_1, \ldots, x_n \neq y_1, \ldots, y_n$ and $\eta_\rho(x_1, \ldots, x_n) = \eta_\rho(y_1, \ldots, y_n)$. Let j be the first number in the sequence $n, 1, 2, \ldots, n-1$ such that $x_j \neq y_j$, and then let $k = 2$ if $j \in \{n, 1, 2\}$; otherwise let $k = j$. The following can be verified to be an accepting run of $\mathfrak{A}_{\mathsf{ica}}$ on ρ

$$\begin{aligned}
&(0, x_nx_1, x_nx_1, y_ny_1, y_ny_1),\\
&(0, x_nx_1, x_1x_2, y_ny_1, y_1y_2),\\
&\vdots\\
&(0, x_nx_1, x_{k-2}x_{k-1}, y_ny_1, y_{k-2}y_{k-1}),\\
&(1, x_nx_1, x_{k-1}x_k, y_ny_1, y_{k-1}y_k),\\
&(1, x_nx_1, x_kx_{k+1}, y_ny_1, y_ky_{k+1}),\\
&\vdots\\
&(1, x_nx_1, x_nx_1, y_ny_1, y_ny_1). \qquad \square
\end{aligned}$$

Lemma 2. *If* $\mathfrak{A}_{\mathsf{ica}}$ *has a run on* $\rho_1, \ldots, \rho_\ell$ *ending with a state of the form* $(1, ab, cd, ef, gh)$ *then there exists* $x_0, \ldots, x_{\ell+1}, y_0, \ldots, y_{\ell+1} \in \mathbb{B}$ *such that*

(1) $x_0x_1 = ab$, $x_\ell x_{\ell+1} = cd$, $y_0y_1 = ef$, *and* $y_\ell y_{\ell+1} = gh$;
(2) $x_0, \ldots, x_\ell \neq y_0, \ldots, y_\ell$;
(3) for all $1 \leqslant i \leqslant \ell$ we have $\rho_i(x_{i-1}x_ix_{i+1}) = \rho_i(y_{i-1}y_iy_{i+1})$.

Proof. Let $s_0, \ldots, s_\ell$ be a run of $\mathfrak{A}_{\mathsf{ica}}$ on $\rho_1, \ldots, \rho_\ell$ and let $(v_i, a_ib_i, c_id_i, e_if_i, g_ih_i) = s_i$, and assume that $v_\ell = 1$. Let k be minimal such that $v_k = 1$; note that $k \geqslant 1$. Let $x_{\ell+1} = d_\ell$, and for all $0 \leqslant i \leqslant \ell$, let $x_i = c_i$. Similarly, let $y_{\ell+1} = h_\ell$ and for all $0 \leqslant i \leqslant \ell$, let $y_i = g_i$. Now $s_0 \in I_{\mathsf{ica}}$ and $(s_{i-1}, \rho_i, s_i) \in \delta_{\mathsf{ica}}$ for all $1 \leqslant i \leqslant \ell$ imply that $x_0x_1 = c_0c_1 = c_0d_0 = a_0b_0 = a_1b_1 = \cdots = a_\ell b_\ell = ab$ and $y_0y_1 = g_0g_1 = g_0h_0 = e_0f_0 = e_1f_1 = \cdots = e_\ell f_\ell = ef$; also we observe that $x_\ell x_{\ell+1} = c_\ell d_\ell = cd$ and $y_\ell y_{\ell+1} = g_\ell h_\ell = gh$. Thus, item (1) of the lemma statement holds by construction. Now since $v_k = 1 \neq v_{k-1}$, we have that $c_{k-1}c_kd_k \neq g_{k-1}g_kh_k$ from the definition of δ_{ica}. Thus, $x_{k-1}x_kx_{k+1} \neq y_{k-1}y_ky_{k+1}$ and item (2) follows. Finally, item (3) also follows from the definition of δ_{ica}. $\square$

This brings us to a main result of the paper.

Theorem 2. *The set of PB reversible CA is regular.*

Proof. Follows from Theorem 1 and the fact that regular languages are closed under complement. $\square$

Theorem 2 tells us that there exists an algorithm that tests if a given CA is PB reversible in linear time. To obtain this algorithm, we must determinize $\mathfrak{A}_{\mathsf{ica}}$ using the well-known subset construction [13]. The resulting automaton, $\mathfrak{A}'_{\mathsf{ica}}$, has $2^{|S_{\mathsf{ica}}|} = 2^{512}$ states, so clearly it cannot be constructed using the time and space resources of any real computer. However, we may test if ρ is PB reversible by constructing only the run $s_0, \ldots, s_n$ of $\mathfrak{A}'_{\mathsf{ica}}$ on ρ (rather than constructing $\mathfrak{A}'_{\mathsf{ica}}$ in its entirety), and checking that s_n is rejecting, i.e. that ρ is *not* irreversible. Since each state of $\mathfrak{A}'_{\mathsf{ica}}$ can naturally be encoded as a bit-string of length 512, this algorithm's space and time requirements fall well within the realm of real computers. Indeed, we have implemented this algorithm and it can easily handle CAs of size 300 in less than 15 sec.

5. A small deterministic automaton for NB reversible CA

In this section, we develop a *deterministic* FSA $\widehat{\mathfrak{A}}_{\mathsf{nb}}$, with a mere nine states, with language being the set of NB reversible CA.

To develop $\widehat{\mathfrak{A}}_{\mathsf{nb}}$, we first define a deterministic automaton $\mathfrak{A}_{\mathsf{nb}}$ having the same language and 1025 states. Roughly, we then show that only a few states of $\mathfrak{A}_{\mathsf{nb}}$ ever actually occur on accepting paths, which allows us to reduce $\mathfrak{A}_{\mathsf{nb}}$ to $\widehat{\mathfrak{A}}_{\mathsf{nb}}$. One can see $\mathfrak{A}_{\mathsf{nb}}$ as being obtained from $\mathfrak{A}_{\mathsf{ica}}$ as follows. First, we observe that when dealing with the NB condition, we do not need to "remember" the first couple bits of the irreversibility witnesses. Hence the second and fourth components of the states of $\mathfrak{A}_{\mathsf{ica}}$ can be removed. Next, the subset construction [13] is used to determinize and complement $\mathfrak{A}_{\mathsf{ica}}$. Finally, we can in fact remove states having 0 as the first component from the state sets, which brings us to $\mathfrak{A}_{\mathsf{nb}}$. The states of $\mathfrak{A}_{\mathsf{nb}}$ are sets of elements of $(\mathbb{B}^2)^2$, which are best seen as graphs with vertices $\mathbb{B}^2$. Rather than formally derive $\mathfrak{A}_{\mathsf{nb}}$ from $\mathfrak{A}_{\mathsf{ica}}$ in this manner, we define the former without mention of the latter. The proof that $L(\mathfrak{A}_{\mathsf{nb}})$ is the set of NB reversible CA; however, does have a similar flavor to that of the proof of Theorem 1.

Let $\mathbb{G}$ be the set of undirected graphs on the four vertices $\mathbb{B}^2$. We allow loops in the graph of $\mathbb{G}$, and we will write an edge as a pair (ab, cd), where it is understood that such pairs are unordered. Since there are $\binom{4}{2} = 6$ possible non-loop edges and four possible loops in the graphs of $\mathbb{G}$, we have that $|\mathbb{G}| = 2^{6+4} = 1024$.

The deterministic FSA $\mathfrak{A}_{\mathsf{nb}} = (S_{\mathsf{nb}}, R, \delta_{\mathsf{nb}}, \{\rhd\}, F_{\mathsf{nb}})$ is defined as follows:

- $S_{\mathsf{nb}} = \mathbb{G} \cup \{\rhd\}$, where $\rhd$ is a state not in $\mathbb{G}$.
- $\delta_{\mathsf{nb}} : S_{\mathsf{nb}} \times R \to S_{\mathsf{nb}}$ is defined such that if $g \in \mathbb{G}$, then $\delta_{\mathsf{nb}}(g, r)$ is the graph g' with edges specified as follows:
    - *Newly born edge*. Whenever $a, b \in \mathbb{B}$ are such that $r(ab0) = r(ab1)$, $(b0, b1)$ is an edge of g'.
    - *Propagated edge*. Whenever (ab, cd) is an edge of g and $r(abe) = r(cdf)$ for some $e, f \in \mathbb{B}$, (be, df) is an edge of g'.
- Also, $\delta_{\mathsf{nb}}(\rhd, r)$ is the graph g' such that whenever $a, b, c, d \in \mathbb{B}$ are such that $r(0ab) = r(0cd)$ and $ab \neq cd$, we have (ab, cd) is an edge of g'.
- F_{nb} is the set of all graphs in $\mathbb{G}$ containing none of the edges (00, 00), (00, 10), or (10, 10).

Note that the above notions of newly born and propagated edges are not inherent to a graph alone, they are really with respect to a transition (g, r, g'). Also, an edge in g' might having multiple "reasons" for existing, and could very well be both newly born and propagated.

Theorem 3. *$\rho \in L(\mathfrak{A}_{\mathsf{nb}})$ if and only if ρ is an NB reversible CA.*

Proof. ($\Rightarrow$) Suppose ρ is an NB irreversible CA, and let $\boldsymbol{x}$ and $\boldsymbol{y}$ be distinct configurations such that $\eta^0_\rho(\boldsymbol{x}) = \eta^0_\rho(\boldsymbol{y})$. We define $x_0 = y_0 = x_{n+1} = y_{n+1} = 0$. Let $g_0, \ldots, g_n$ be the run of $\mathfrak{A}_{\mathsf{nb}}$ on $\rho_1, \ldots, \rho_n$ (this run is unique since $\mathfrak{A}_{\mathsf{nb}}$ is deterministic). Let $1 \leqslant k' \leqslant n$ be minimal such that $x_{k'} \neq y_{k'}$ and let $k = \max(k', 2)$. We show by induction that $(x_i x_{i+1}, y_i y_{i+1})$ is an edge of g_i for all $k - 1 \leqslant i \leqslant n$. If $k = 2$, then $(x_1 x_2, y_1 y_2)$ is an edge of g_1, by the definition of $\delta_{\mathsf{nb}}(\rhd, \rho_1)$. Otherwise, if $k > 2$, then $(x_{k-1} x_k, y_{k-1} y_k)$ is a newly born edge of (g_{k-1}, ρ_k, g_k). For the inductive step, assume that $(x_i x_{i+1}, y_i y_{i+1})$ is an edge of g_i for some $i \geqslant k - 1$. Then clearly $(x_{i+1} x_{i+2}, y_{i+1} y_{i+2})$ is a propagated edge of $(g_i, \rho_{i+1}, g_{i+1})$. It follows from our claim that $(x_n 0, y_n 0)$ is an edge of g_n, hence $g_n \notin F_{\mathsf{nb}}$ and $\rho_1, \ldots, \rho_n \notin L(\mathfrak{A}_{\mathsf{nb}})$.

($\Leftarrow$) Let $g_0, \ldots, g_n$ be the run of $\mathfrak{A}_{\mathsf{nb}}$ on $\rho_1, \ldots, \rho_n$, and suppose that $g_n \notin F_{\mathsf{nb}}$. Let $(x_n 0, y_n 0)$ be an edge of g_n, which must therefore exist. Note that $(x_n 0, y_n 0)$ is a propagated edge in (g_{n-1}, ρ_n, g_n), since newly born edges must involve vertices that disagree in their second components. A propagated edge is "caused" by an edge in the previous state of $\mathfrak{A}_{\mathsf{nb}}$, let us call this edge the *parent*. Let

$$(x_k x_{k+1}, y_k y_{k+1}), \ldots, (x_{n-1} x_n, y_{n-1} y_n), (x_n 0, y_n 0)$$

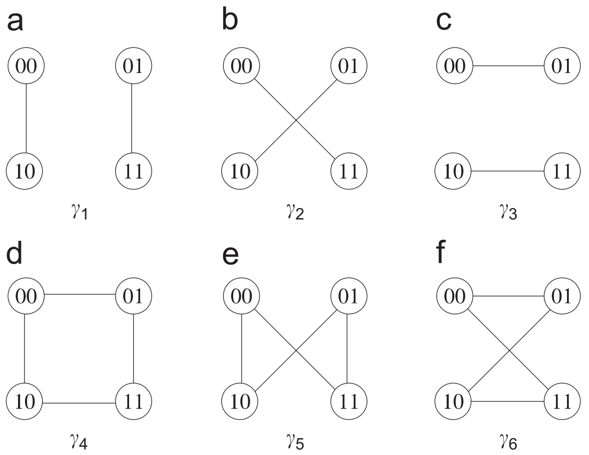

Fig. 1. Six viable graphs.

be a maximal sequence of edges such that $(x_{i-1}x_i, y_{i-1}y_i)$ is a parent of (x_ix_{i+1}, y_iy_{i+1}) in (g_{i-1}, ρ_i, g_i) for all $k < i \leqslant n$, where x_{n+1} and y_{n+1} are defined to be 0. We case split on the value of k, noting that $k < n$:

- $k = 1$. We claim that $\boldsymbol{x} = x_1, \ldots, x_n$ and $\boldsymbol{y} = y_1, \ldots, y_n$ are NB irreversibility witnesses. To see that $\boldsymbol{x} \neq \boldsymbol{y}$, we note that since (x_1x_2, y_1y_2) is an edge in g_1 we have that $x_1x_2 \neq y_1y_2$, from the definition of $\delta_{\mathsf{nb}}(\rhd, \rho_1)$. To see that $\eta^0_\rho(\boldsymbol{x}) = \eta^0_\rho(\boldsymbol{y})$, note that since (x_ix_{i+1}, y_iy_{i+1}) is propagated for all $1 < i \leqslant n$, we have that

$$\rho_i(x_{i-1}x_ix_{i+1}) = \rho_i(y_{i-1}y_iy_{i+1}) \tag{1}$$

holds for such i. That (1) holds for $i = 1$ follows from the definition of $\delta_{\mathsf{nb}}(\rhd, \rho_1)$.
- $k > 1$. Note that since (x_kx_{k+1}, y_iy_{k+1}) is a newly born edge in (g_{k-1}, ρ_k, g_k), we have that $x_{k+1} \neq y_{k+1}$, $x_k = y_k$, and there exists $a \in \mathbb{B}$ such that $\rho_k(ax_kx_{k+1}) = \rho_k(ay_ky_{k+1})$; let us define $x_{k-1} = y_{k-1} = a$. Also define $x_i = y_i = 0$ for all $0 \leqslant i < k-1$. We claim that $\boldsymbol{x} = x_1, \ldots, x_n$ and $\boldsymbol{y} = y_1, \ldots, y_n$ are NB irreversibility witnesses. Clearly $\boldsymbol{x} \neq \boldsymbol{y}$, since $x_{k+1} \neq y_{k+1}$ and $k+1 \leqslant n$. To see that $\eta^0_\rho(\boldsymbol{x}) = \eta^0_\rho(\boldsymbol{y})$, we show (1) for all $1 \leqslant i \leqslant n$ by case splitting on i:
    - $1 \leqslant i < k$. Since (x_kx_{k+1}, y_ky_{k+1}) is newly born, it follows that $x_k = y_k$. Therefore, $x_i = y_i$ for all $0 \leqslant i \leqslant k$ and (1) holds for all $1 \leqslant i < k$.
    - $i = k$. We previously defined x_{k-1} and y_{k-1} such that (1) holds for $i = k$.
    - $k < i \leqslant n$. Since (x_ix_{i+1}, y_iy_{i+1}) is propagated for all $k < i \leqslant n$, we have that (1) holds for all such i. □

Note that $|S_{\mathsf{nb}}| = 1 + 2^{10} << 2^{512}$, the latter being the number of states arising if one determinizes $\mathfrak{A}_{\mathsf{ica}}$. In the following, we show that there are only six graphs in $\mathbb{G}$ that ever appear on accepting runs of $\mathfrak{A}_{\mathsf{nb}}$. This observation allows us to define a reduced automaton $\widehat{\mathfrak{A}}_{\mathsf{nb}}$ with a mere nine states (namely, these six graphs along with $\rhd$ and two new states $\spadesuit$ and $\Diamond$).

Here we define several properties of graphs $g \in \mathbb{G}$. g is said to be *viable* if it is one of the six graphs of Fig. 1; we let $\mathbb{V}$ denote the set containing these six graphs. A *loop* is an edge of the form (u, u) for some $u \in \mathbb{B}^2$. A 3-*cycle* is a set of three edges $\{(u, v), (v, w), (w, u)\}$ such that u, v, and w are all distinct. We say a graph is *doomed* if it contains a loop or a 3-cycle; note that no viable graph is doomed, though there exist graphs that are neither viable nor doomed.

It turns out that only viable graphs can appear on accepting runs of $\mathfrak{A}_{\mathsf{nb}}$, with the possible exception of the last state; this is captured by the following lemma.

Lemma 3. *Let $g_0, \ldots, g_n$ be an accepting run of $\mathfrak{A}_{\mathsf{nb}}$. Then g_i is viable for all $1 \leqslant i < n$.*

Proof. If $n = 1$ then the lemma holds vacuously, thus we assume $n \geqslant 2$. On the contrary, suppose at least one graph $g_1, \ldots, g_{n-1}$ is not viable, then let k be minimal such that g_k is not viable. It follows that $g_{k-1} \in \{\rhd\} \cup \mathbb{V}$, thus, by Lemmas 4 and 5, we have that g_k is doomed. Thus, by inductively applying Lemma 6, $g_{k+1}, \ldots, g_n$ are all not in

Table 1
A tabular representation of the function $\delta_{\mathsf{nb}} : S_{\mathsf{nb}} \times R \to S_{\mathsf{nb}}$, restricted to the domain $(\{\rhd\} \cup \mathbb{V}) \times R$

r			$\rhd$	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6
00001111	15	240	◇	γ_3	γ_3	◇	◇	γ_3	◇
00010111	23	232	◇	◇	γ_4	◇	◇	◇	◇
00011011	27	228	♤	◇	γ_6	♤	◇	◇	♤
00011101	29	226	◇	◇	◇	◇	◇	◇	◇
00011110	30	225	♤	γ_3	◇	♤	♤	◇	◇
00100111	39	216	◇	♤	γ_6	◇	◇	♤	◇
00101011	43	212	♤	◇	γ_4	◇	◇	◇	◇
00101101	45	210	◇	γ_3	♤	◇	◇	♤	◇
00101110	46	209	♤	◇	◇	◇	◇	◇	◇
00110011	51	204	γ_3	◇	γ_3	γ_3	◇	◇	γ_3
00110101	53	202	γ_1	◇	◇	γ_4	◇	◇	◇
00110110	54	201	γ_2	♤	◇	γ_6	♤	◇	◇
00111001	57	198	γ_2	◇	♤	γ_6	◇	◇	♤
00111010	58	197	γ_1	◇	◇	γ_4	◇	◇	◇
00111100	60	195	γ_3	γ_3	◇	γ_3	γ_3	◇	◇
01000111	71	184	◇	◇	◇	◇	◇	◇	◇
01001011	75	180	♤	γ_3	◇	♤	♤	◇	◇
01001101	77	178	◇	◇	γ_4	◇	◇	◇	◇
01001110	78	177	♤	◇	γ_6	♤	◇	◇	♤
01010011	83	172	γ_3	◇	◇	γ_4	◇	◇	◇
01010101	85	170	γ_1	◇	γ_1	γ_1	◇	◇	γ_1
01010110	86	169	γ_2	◇	γ_5	γ_5	◇	◇	γ_5
01011001	89	166	γ_2	◇	γ_5	γ_5	◇	◇	γ_5
01011010	90	165	γ_1	γ_3	γ_2	γ_1	γ_4	γ_6	γ_5
01011100	92	163	γ_3	◇	◇	γ_4	◇	◇	◇
01100011	99	156	γ_3	♤	◇	γ_6	♤	◇	◇
01100101	101	154	γ_1	◇	γ_5	γ_5	◇	◇	γ_5
01100110	102	153	γ_2	◇	γ_2	γ_2	◇	◇	γ_2
01101001	105	150	γ_2	γ_3	γ_1	γ_2	γ_6	γ_4	γ_5
01101010	106	149	γ_1	◇	γ_5	γ_5	◇	◇	γ_5
01101100	108	147	γ_3	◇	♤	γ_6	◇	◇	♤
01110001	113	142	♤	◇	γ_4	◇	◇	◇	◇
01110010	114	141	◇	♤	γ_6	◇	◇	♤	◇
01110100	116	139	♤	◇	◇	◇	◇	◇	◇
01111000	120	135	◇	γ_3	♤	◇	◇	♤	◇

Each row is labelled with the (transposed) truth table of some $r \in R$, i.e. r is expressed as the bit-string $r(111)r(110)\ldots r(000)$, along with the rule numbers for both r and the negation of r; both rules always have the same row, allowing for a 2-fold reduction in table height. If $\delta_{\mathsf{nb}}(g, r) \in \mathbb{V}$ for the column/row for g/r, then the corresponding entry is $\delta_{\mathsf{nb}}(g, r)$; if $\delta_{\mathsf{nb}}(g, r)$ is doomed and accepting (resp. rejecting), this is specified by ♤ (resp. ◇). Hence the table demonstrates Lemma 5.

F_{nb} (and also all doomed). Since $k < n$, this implies that $g_n \notin F_{\mathsf{nb}}$, which contradicts $g_0, \ldots, g_n$ being an accepting run. □

Lemma 4. *For any $r \in R$, either $\delta_{\mathsf{nb}}(\rhd, r) \in \mathbb{V}$ or $\delta_{\mathsf{nb}}(\rhd, r)$ is doomed.*

Proof. Let $g' = \delta_{\mathsf{nb}}(\rhd, r)$. The edges of g' only depend on the equivalence classes induced on $U = \{000, 001, 010, 011\}$ by r. If r maps three (or four) elements of U to the same value, g' contains a 3-cycle and is doomed. Left to consider is the case when r maps two elements of U to 1 and two to 0; here we find $g' \in \{\gamma_1, \gamma_2, \gamma_3\} \subset \mathbb{V}$. □

Lemma 5. *For any $r \in R$ and $g \in \mathbb{V}$, either $\delta_{\mathsf{nb}}(g, r) \in \mathbb{V}$, or $\delta_{\mathsf{nb}}(g, r)$ is doomed.*

Proof. Table 1 shows the $\delta_{\mathsf{nb}}(g, r)$ for each $g \in \mathbb{V}$ and $r \in R$, hence proving this lemma by exhaustive enumeration. □

Lemma 6. *For any doomed $g \in \mathbb{G}$ and $r \in R$, $\delta_{\mathsf{nb}}(g, r)$ is doomed and not in F_{nb}.*

Proof. Let $g' = \delta_{\mathsf{nb}}(g, r)$. Suppose g has a loop (xy, xy). Then $(y0, y0)$ is a propagated edge in (g, r, g'), and hence g' has an odd cycle. Now suppose g has a 3-cycle ab, cd, ef. Note that exactly two of b, d, and f must be equal, without loss of generality we assume that $b = d \neq f$. If $r(ab0) = r(cd0)$, then $(b0, d0)$ is a loop in g' and we are done. Otherwise either $r(ab0) = r(ef0)$ or $r(cd0) = r(ef0)$, without loss of generality let us assume the former. Then $(b0, f0) = (00, 10)$ is an edge in g', hence $g' \notin F_{\mathsf{nb}}$. We case-split:

- There exists $xy \in \{ab, cd, ef\}$ such that $r(xy1) = r(ab0)$. Then $(y1, 00)$ and $(y1, 10)$ are edges in g' and hence g' has a 3-cycle.
- If there does not exist such a xy, then it must be that $r(ab1) = r(cd1) = r(ef1) = r(cd0)$, and it follows that $(b1, d1)$ is a edge in g'. Since $b = d$, g' has a loop.

In both cases we find that g' is doomed. □

We now define the deterministic automaton $\widehat{\mathfrak{A}}_{\mathsf{nb}} = (\widehat{S}_{\mathsf{nb}}, R, \widehat{\delta}_{\mathsf{nb}}, \{\rhd\}, \widehat{F}_{\mathsf{nb}})$. The states of $\widehat{\mathfrak{A}}_{\mathsf{nb}}$ include two new states $\spadesuit$ and $\diamondsuit$, which, respectively, represent all doomed states that are accepting (in $\mathfrak{A}_{\mathsf{nb}}$) and all doomed states that are not accepting (in $\mathfrak{A}_{\mathsf{nb}}$).

- $\widehat{S}_{\mathsf{nb}} = \mathbb{V} \cup \{\rhd, \spadesuit, \diamondsuit\}$.
- $\widehat{F}_{\mathsf{nb}} = \{\spadesuit\} \cup (\mathbb{V} \cap F_{\mathsf{nb}}) = \{\spadesuit, \gamma_2, \gamma_3, \gamma_6\}$.
- $\widehat{\delta}_{\mathsf{nb}}$ is defined by

$$\widehat{\delta}_{\mathsf{nb}}(g, r) = \begin{cases} \delta_{\mathsf{nb}}(g, r) & \text{if } g \in \mathbb{V} \cup \{\rhd\} \text{ and } \delta_{\mathsf{nb}}(g, r) \in \mathbb{V}, \\ \spadesuit & \text{if } g \in \mathbb{V},\ \delta_{\mathsf{nb}}(g, r) \notin \mathbb{V} \text{ and } \delta_{\mathsf{nb}}(g, r) \in F_{\mathsf{nb}}, \\ \diamondsuit & \text{if } g \in \mathbb{V},\ \delta_{\mathsf{nb}}(g, r) \notin \mathbb{V} \text{ and } \delta_{\mathsf{nb}}(g, r) \notin F_{\mathsf{nb}}, \\ \diamondsuit & \text{if } g \in \{\spadesuit, \diamondsuit\}. \end{cases}$$

Theorem 4. *$\rho \in L(\widehat{\mathfrak{A}}_{\mathsf{nb}})$ if and only if ρ is an NB reversible CA.*

Proof. We show that $L(\widehat{\mathfrak{A}}_{\mathsf{nb}}) = L(\mathfrak{A}_{\mathsf{nb}})$; the result follows by Theorem 3.

($\supseteq$) Let $g_0, \ldots, g_n$ be an accepting run of $\mathfrak{A}_{\mathsf{nb}}$ on $\rho_1, \ldots, \rho_n$. If $g_n \in F_{\mathsf{nb}}$, then by Lemma 3, $g_1, \ldots, g_{n-1} \in \mathbb{V}$. If $g_n \in \mathbb{V}$, then $g_0, \ldots, g_n$ is the run of $\widehat{\mathfrak{A}}_{\mathsf{nb}}$ on ρ, and is accepting. If $g_n \notin \mathbb{V}$, then $g_0, \ldots, g_{n-1}, \spadesuit$ is the run of $\widehat{\mathfrak{A}}_{\mathsf{nb}}$ on ρ, and is also accepting.

($\subseteq$) Let $\hat{g}_0, \ldots, \hat{g}_n$ be an accepting run of $\widehat{\mathfrak{A}}_{\mathsf{nb}}$ on $\rho_1, \ldots, \rho_n$. If $\hat{g}_i \in \mathbb{V}$ for all $1 \leqslant i \leqslant n$, then clearly $\hat{g}_0, \ldots, \hat{g}_n$ is also an accepting run of $\mathfrak{A}_{\mathsf{nb}}$. Otherwise, let $k \geqslant 1$ be minimal such that $\hat{g}_k \notin \mathbb{V}$. It follows that $g_0, \ldots, g_{k-1}$ is a run of $\mathfrak{A}_{\mathsf{nb}}$ on $\rho_1, \ldots, \rho_{k-1}$. If $k = n$, then $\hat{g}_n = \spadesuit$, since $\hat{g}_n$ is accepting, and since $\hat{g}_{n-1} \in \mathbb{V}$, we have that $\delta_{\mathsf{nb}}(\hat{g}_{n-1}, \rho_n) \in F_{\mathsf{nb}}$ and therefore $\mathfrak{A}_{\mathsf{nb}}$ accepts $\rho_1, \ldots, \rho_n$. If $k < n$ it follows that $\hat{g}_{k+1} = \cdots = \hat{g}_n = \diamondsuit$, which contradicts $\hat{g}_0, \ldots, \hat{g}_n$ being an accepting run. □

6. Fast random uniform generation of NB reversible CA

In this section we present an algorithm that, given n, generates a random NB reversible CA of size n using $\mathcal{O}(n^2)$ time, and generates each NB reversible CA with equal probability (i.e. the generation is from the uniform distribution). Our algorithm is an instance of the approach of Hickey and Cohen [7] for generating random strings from a context free grammar, applied to the regular language of NB reversible CA. As observed by Hickey and Cohen, it is easy to generate a "random" string; one simply traverses the automaton, at each state selecting one of the transitions out of the state with equal probability. However, this is very unlikely to result in a uniformly distributed selection, and certainly would not for our automaton $\widehat{\mathfrak{A}}_{\mathsf{nb}}$. For instance, Table 1 suggests that the additive rules 90, 105, 150, and 165 appear in NB reversible CA with a higher frequency than other rules, since they never lead to a doomed state; thus transitions on these rules should be followed with higher probability.

To solve this problem, one must assign "weights" to the transitions of $\widehat{\mathfrak{A}}_{\mathsf{nb}}$ that dictate the probability that the transition should be followed. This weight depends on the number of rules left to generate. Hence, this weighting depends on the number of rules generated so far.

```
function RANDOM_NB_REVERSIBLE_CA($n$)
    compute and store the values of count
    $s := \triangleright$
    for $i := 1, \ldots, n$ do
        select $r \in R$ with probability $\mathsf{prob}(s, r, i)$
        $\rho_i := r$
        $s := \widehat{\delta}_{\mathsf{nb}}(s, \rho_i)$
    end for
    return $\rho_1 \ldots \rho_n$
end function
```

Fig. 2. An algorithm that generates a random NB reversible CA from the uniform distribution.

Let $\mathbb{N}_n = \{0, \ldots, n\}$, and let $\mathbb{N} = \{0, 1, \ldots\}$. Given n, our uniform generation algorithm first constructs a table for the function count : $\widehat{S}_{\mathsf{nb}} \times \mathbb{N}_n \to \mathbb{N}$, which is defined inductively by

$$\mathsf{count}(g, i) = \begin{cases} 0 & \text{if } i = n \text{ and } g \notin \widehat{F}_{\mathsf{nb}}, \\ 1 & \text{if } i = n \text{ and } g \in \widehat{F}_{\mathsf{nb}}, \\ \sum\limits_{r \in R} \mathsf{count}(\widehat{\delta}_{\mathsf{nb}}(g, r), i + 1) & \text{if } i < n. \end{cases}$$

The algorithm also relies on a function prob : $\widehat{S}_{\mathsf{nb}} \times (\mathbb{N}_n \backslash \{0\}) \times R \to [0, 1]$, which depends on count:

$$\mathsf{prob}(s, i, r) \;=\; \frac{\mathsf{count}(\widehat{\delta}_{\mathsf{nb}}(s, r), i)}{\mathsf{count}(s, i - 1)}.$$

The random generation algorithm is given in Fig. 2. Given input $n \geqslant 1$ RANDOM_NB_REVERSIBLE_CA first computes a table storing the value $\mathsf{count}(s, i)$ for each $(s, i) \in \widehat{S}_{\mathsf{nb}} \times \mathbb{N}_n$ We note that the number of entries in the table is only $|\widehat{S}_{\mathsf{nb}}|n{=}9n$. Next, starting with at the initial state $\triangleright$, RANDOM_NB_REVERSIBLE_CA follows n transitions of $\widehat{\mathfrak{A}}_{\mathsf{nb}}$, at each state selecting the next rule according to prob. The sequence of rules generated in this manner yields an NB reversible CA $\rho_1, \ldots, \rho_n$, selected uniformly from the set of all NB reversible CA of size n. That RANDOM_NB_REVERSIBLE_CA indeed generates NB reversible CA from the uniform distribution follows from Hickey and Cohen [7], hence we do not provide a proof. Though both the computation of the count table and the generation of the random CA require $\mathcal{O}(n)$ operations, some of these operations involve addition operations performed on very large integers (i.e. those in the image of count). These integers involve $\mathcal{O}(n)$ bits, and hence a single addition costs time $\mathcal{O}(n)$. The net result is that RANDOM_NB_REVERSIBLE_CA takes $\mathcal{O}(n^2)$ time. There are no prohibitive hidden constants here, for instance our implementation generates a reversible CA with 200 cells in 90 seconds on a contemporary laptop computer.

7. Counting NB reversible CA

In this section, we derive a formula involving matrix powers that gives the number of NB reversible CA for a given size n. We then find the asymptotic growth of this formula.

Let us define the following row vector s, 6×6 matrix T, and column vector f.

$$s = [12 \;\; 12 \;\; 12 \;\; 0 \;\; 0 \;\; 0],$$

$$T = \begin{bmatrix} 0 & 0 & 16 & 0 & 0 & 0 \\ 4 & 4 & 4 & 8 & 8 & 8 \\ 4 & 4 & 4 & 8 & 8 & 8 \\ 0 & 0 & 2 & 2 & 0 & 2 \\ 0 & 0 & 2 & 2 & 0 & 2 \\ 2 & 2 & 2 & 0 & 12 & 0 \end{bmatrix}, \quad f = \begin{bmatrix} 24 \\ 24 \\ 24 \\ 12 \\ 12 \\ 12 \end{bmatrix}.$$

These values are used in our formula that counts the number of NB reversible CA, presented in the following theorem.

Theorem 5. *The number of NB reversible CA of size $n \geqslant 2$ is $sT^{n-2}f$.*

Proof. Viewing $\widehat{\mathfrak{A}}_{\mathsf{nb}}$ as a multi-graph, the number of NB reversible CA of size n is simply the number of walks from $\triangleright$ to an element of $\widehat{F}_{\mathsf{nb}} = \{$ ♤ $, \gamma_2, \gamma_3, \gamma_6\}$ in $\widehat{\mathfrak{A}}_{\mathsf{nb}}$; this follows from Theorem 4 and the fact that $\widehat{\mathfrak{A}}_{\mathsf{nb}}$ is deterministic. The proof is essentially an application of the well-known fact that the number of walks of length n between pairs of vertices in a multi-graph is given by the adjacency matrix raised to the nth power [3].

Let $walks(m, \gamma_i)$ be the number of distinct walks of length $m \geqslant 1$ from $\triangleright$ to γ_i. We claim that $walks(m, \gamma_i)$ is the ith component of sT^{m-1}. For $m = 1$, the claim follows by inspection of Table 1. For $m > 1$, we recall that any walk from $\triangleright$ to γ_i cannot involve ♤ or $\Diamond$ and only visits $\triangleright$ initially, by the definition of $\widehat{F}_{\mathsf{nb}}$. Thus,

$$walks(m, \gamma_i) \;=\; \sum_{1 \leqslant j \leqslant 6} walks(m-1, \gamma_j) |\{r \in R : \widehat{\delta}_{\mathsf{nb}}(\gamma_j, r) = \gamma_i|,$$

which is $(sT^{m-2})T$, again by inspection of Table 1.

Any walk from $\triangleright$ to $\widehat{F}_{\mathsf{nb}}$ can clearly only visit elements of $\mathbb{V}$ internally, hence the total number of such walks of length n is

$$\sum_{1 \leqslant j \leqslant 6} walks(n-1, \gamma_j) |\{r \in R : \widehat{\delta}_{\mathsf{nb}}(\gamma_j, r) \in \widehat{F}_{\mathsf{nb}}|,$$

which by the above claim is

$$\sum_{1 \leqslant j \leqslant 6} (sT^{n-2})_j |\{r \in R : \widehat{\delta}_{\mathsf{nb}}(\gamma_j, r) \in \widehat{F}_{\mathsf{nb}}|.$$

By inspection of Table 1, this is equal to $sT^{n-2}f$. □

We now determine the asymptotic growth of this formula.

Theorem 6. *The number of NB reversible CA of size n grows as* $\Theta(\lambda^n)$ *where* $\lambda \approx 17.98$.

Proof. Given 6×6 matrix A, let $\|A\| = sT\hat{A}f$, where $\hat{A}$ is the matrix of entry-wise absolute values of A. It is straightforward to verify that $\|\cdot\|$ is a matrix norm [9, Chapter 5]. By Theorem 5, the number of NB reversible CA of size n is $sT^{n-2}f = \|T^{n-3}\|$, since T contains no negative entries. According to Meyer [9, p. 619],

$$\lambda^k = \lim_{k \to \infty} \|T^k\|,$$

where λ is the spectral radius of T. Thus, $\|T^{n-3}\|$ grows as $\Theta(\lambda^{n-3})$. Since λ is constant, we have $\lambda^n = \Theta(\lambda^{n-3})$, hence the number of NB reversible CA is $\Theta(\lambda^n)$. Numerically, we have computed λ to be approximately 17.98. □

We conclude this section by contrasting the number of NB reversible CA, $\Theta(\lambda^n)$, with the total number of CA of size n, which is exactly 256^n. Clearly the ratio of NB reversible CA vanishes as n goes to infinity.

8. Conclusions and future work

We have presented an automaton construction to prove that the set of PB reversible CA are regular. Also we have developed a very simple automaton for NB reversible CA, and used this automaton to define a practical algorithm for random uniform generation of NB reversible CA, and also to count the NB reversible CA. An immediate avenue of future work is to provide a sufficiently simple automaton for PB reversible CA to allow for similar results for such CA.

Our automata constructions might be applicable to deciding reversibility of infinite hybrid one-dimensional CA. Such CA might be specified as the bi-infinite concatenation of a finite string of rules ρ, i.e. $\ldots \rho\rho\rho \ldots$. We also wonder if our approach can be generalized to handle higher dimensional finite hybrid CA.

Finally, we note that Table 1 can be used to make observations about how frequently different rules appear in NB reversible CA. For instance, we see that the four additive rules {90, 165, 105, 150} never lead to a doomed state, and hence appear with high frequency. On the other hand, the eight rules {29, 226, 46, 209, 71, 184, 116, 139} always lead to the rejecting doomed state $\Diamond$, and hence *never* appear in NB reversible CA.

Acknowledgment

We thank Dale Olesky for his insights with regard to Section 7.

References

[1] S. Amoroso, Y.N. Patt, Decision procedures for surjectivity and injectivity of parallel maps for tessellation structures, J. Comput. System Sci. 6 (1972) 448–464.

[2] J. Bingham, Genetic evolution of nonlinear cellular automata for built-in self test of combinational circuit, Master's Thesis, University of Victoria, 2001.

[3] J.A. Bondy, U.S.R. Murty, Graph Theory with Applications, Elsevier, New York, 1976.

[4] J.H. Conway, Winning Ways for your Mathematical Plays, vol. 2, Academic Press, London, 1982 (Chapter 25).

[5] N. Ganguly, B.K. Sikdar, A. Deutsch, G. Canright, P. Pal Chaudhuri, A survey on cellular automata, Technical Report, Center for High Performance Computing, Dresden University of Technology, 2003.

[6] S.W. Golomb, Shift Register Sequences, Aegean Park Press, 1981.

[7] T. Hickey, J. Cohen, Uniform random generation of strings in a context-free language, SIAM J. Comput. 12 (4) (1983) 645–655.

[8] J. Kari, Reversibility and surjectivity problems of cellular automata, J. Comput. System Sci. 48 (1994) 149–182.

[9] C.D. Meyer, Matrix Analysis and Applied Linear Algebra, Society for Industrial and Applied Mathematics, 2000.

[10] E.F. Moore, Machine models of self-reproduction, in: Proceedings of the 14th Symposium on Applied Mathematics, 1962, pp. 17–33.

[11] J. Myhill, The converse to Moore's Garden-of-Eden Theorem, Proc. Amer. Math. Soc. 14 (1963) 685–686.

[12] P. Pal Chaudhuri, D. Roy Chowdhury, S. Nandi, S. Chattopadhyay, Additive Cellular Automata: Theory and Applications, vol. 1, Wiley-IEEE Computer Society Press, 1997.

[13] M.O. Rabin, D. Scott, Finite automata and their decision problems, IBM J. Res. Develop. 3 (2) (1959) 114–125.

[14] D. Richardson, Tessellations with local transformations, J. Comput. System Sci. 5 (1972) 373–388.

[15] P. Sarkar, R. Barua, The set of reversible 90/150 cellular automata is regular, Discrete Appl. Math. 84 (1–3) (1998) 199–213.

[16] K. Sutner, The complexity of reversible cellular automata, Theoret. Comput. Sci. 325 (2) (2004) 317–328.

[17] T. Toffoli, N. Margolus, Invertible cellular automata: a review, Physica D 45 (1990) 229–253 (Reprint with corrections and annotations, December 1996).

[18] S. Wolfram, A New Kind of Science, Wolfram Media, 2002.