Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 252 (2009) 77–91
www.elsevier.com/locate/entcs

On One-way One-bit O(One)-message Cellular Automata

Martin Kutrib[1] and Andreas Malcher[2]

Institut für Informatik, Universität Giessen
Arndtstr. 2, 35392 Giessen, Germany

Abstract

Cellular automata are one-dimensional arrays of interconnected interacting finite automata, also called cells. Here, we investigate one of the weakest class of cellular automata, namely the class of real-time one-way cellular automata. Additionally, we impose two restrictions on the inter-cell communication. First, the number of allowed uses of the links between cells is bounded. Moreover, the amount of information to be communicated in one time step is bounded by a constant being independent from the given automaton. In the weakest case, we consider cellular automata with one-way information flow where each cell receives one bit of information exactly once. In this case, we obtain a characterization of the regular languages for unary alphabets and the acceptance of non-context-free languages for non-unary alphabets. Next, a proper language hierarchy can be derived when increasing the number of allowed uses of the links between cells step by step. Finally, decidability problems of these restricted cellular automata are studied and undecidability of almost all problems can be proven even in the most restricted case of one-way one-bit communication.

Keywords: cellular automata, limited communication, formal languages, decidability.

1 Introduction

Devices of homogeneous, interconnected, parallel acting automata have widely been investigated from a computational capacity point of view. In particular, many results are known about *cellular automata* (see, for example, the surveys [1,2]) which are linear arrays of identical copies of deterministic finite automata, where the single nodes, which are sometimes called cells, are homogeneously connected to their both immediate neighbors. Additionally, they work synchronously at discrete time steps.

Clearly, the computational power of systems of parallel acting automata, here cellular automata, relies on the ability of the system to communicate information between single automata within given time and space constraints. Thus, it is essential to know how communication should be organized in order to employ cellular

[1] Email: kutrib@informatik.uni-giessen.de
[2] Email: malcher@informatik.uni-giessen.de

doi:10.1016/j.entcs.2009.09.015

automata optimally. Additionally, it would be interesting to understand in which way the number of communications as well as the amount of information communicated can be minimized or at least reduced.

Considering the (unrestricted) model of cellular automata, the state of each cell is communicated to its neighbors in every time step. That is, on the one hand the state is sent regardless of whether it is really required, and on the other hand, the number of bits sent is determined by the number of states. Much work has been done in studying restricted variants of cellular automata where the bandwidth of the communication links between two cells is bounded by some fixed constant. In the most restricted setting, only one bit of information is allowed to be communicated between two cells in one time step. However, despite their reduced communication these cellular automata are still powerful enough to accept unary as well as non-unary non-context-free (even non-semilinear) languages. For some classes it is additionally known that almost all of the commonly investigated decidability problems are undecidable. Results may be found in [3,4,8,9,10,13].

Another resource restricted in order to study the power of communication has been investigated in [11,12]. There, two-way cellular automata are considered where the number of proper state changes is bounded. There are strong relations to inter-cell communication. Roughly speaking, a cell can remember the states received from its neighbors. As long as these do not change, no communication is necessary. In [5,6] real-time one-way cellular automata have been studied where the communication is quantitatively measured by counting the number of uses of the communication links between cells. One measure studied is the maximal number of communications that may appear between each two cells. Reducing the number of communications in such a way that each two neighboring cells may communicate constantly often only, leads to devices which still can accept non-context-free (even non-semilinear) languages. Again, almost all of their decidability questions can be shown to be undecidable.

In this paper, we combine these two approaches and study real-time one-way cellular automata which are allowed to communicate only a fixed, finite number of bits per time step. Additionally, the communication links between two cells are allowed to be used constantly often only. In the most restricted case, we consider real-time one-way cellular automata which can communicate one bit of information between two cells exactly once.

In the next section, we present some basic notions and definitions, and introduce the classes of communication bounded cellular automata with respect to the bandwidth of the communication links as well as to the number of uses of these links. In Section 3, we show that the restriction to one bit and one communication is so strong that the regular languages can be characterized by these devices in the unary case. However, in the non-unary case non-regular languages can be accepted with two communications per cell. Next, a proper hierarchy on the number of communication steps is shown. That is, even in the setting of one-bit communication, devices with ℓ communications are more powerful than devices with $\ell - 1$ communications. Section 4 concerns decidability problems. First, we show a lemma which relates

languages accepted by real-time one-way cellular automata with a constant number of communications with languages accepted by real-time one-way cellular automata with a constant number of communications and one-bit communication. In this way, undecidability results for the cellular automata in question can be derived from known undecidability results.

2 Definitions and preliminaries

We denote the positive integers and zero $\{0, 1, 2, ...\}$ by $\mathbb{N}$. The empty word is denoted by λ, the reversal of a word w by w^R, and for the length of w we write $|w|$. For the number of occurrences of a subword x in w we use the notation $|w|_x$. We use $\subseteq$ for inclusions and $\subset$ for strict inclusions. In order to avoid technical overloading in writing, two languages L and L' are considered to be equal, if they differ at most by the empty word, i.e., $L - \{\lambda\} = L' - \{\lambda\}$. Throughout the article two devices are said to be *equivalent* if and only if they accept the same language.

A one-way cellular automaton is a linear array of identical deterministic finite state machines, sometimes called cells. Except for the rightmost cell each one is connected to its nearest neighbor to the right. We identify the cells by positive integers. The state transition depends on the current state of each cell and on the information which is currently sent by its neighbor. The information sent by a cell depends on its current state and is determined by so-called communication functions. The rightmost cell receives information associated with a boundary symbol on its free input line once during the first time step from the outside world. Subsequently, these input line is never used again. A formal definition is

Definition 2.1 A *one-way cellular automaton* (OCA) is a system $\langle S, F, A, B, \#, b_l, \delta\rangle$, where S is the finite, nonempty set of *cell states*, $F \subseteq S$ is the set of *accepting states*, $A \subseteq S$ is the nonempty set of *input symbols*, B is the set of *communication symbols*, $\# \notin S$ is the *boundary symbol*, $b_l : (S \cup \{\#\}) \to B \cup \{\perp\}$ is the *communication function* which determines the information *to be sent* to the left neighbor, where $\perp$ means *nothing to send*, and $\delta : S \times (B \cup \{\perp\}) \to S$ is the *local transition function*.

A *configuration* of a one-way cellular automaton $\langle S, F, A, B, \#, b_l, \delta\rangle$ at time $t \geq 0$ is a description of its global state, which is actually a mapping $c_t : \{1, 2, \ldots, n\} \to S$, for $n \geq 1$. The operation starts at time 0 in a so-called *initial configuration*. For a given input $w = a_1 \cdots a_n \in A^+$ we set $c_{0,w}(i) = a_i$, for $1 \leq i \leq n$. During the course of its computation an OCA steps through a sequence of configurations, whereby successor configurations are computed according to the global transition function Δ: Let c_t, $t \geq 0$, be a configuration. Then its successor configuration $c_{t+1} = \Delta(c_t)$ is as follows. For $1 \leq i \leq n-1$, $c_{t+1}(i) = \delta(c_t(i), b_l(c_t(i+1)))$, and for the rightmost cell we set $c_1(n) = \delta(c_0(n), b_l(\#))$, $c_{t+1}(n) = \delta(c_t(n), \perp)$, for $t \geq 1$. Thus, the global transition function Δ is induced by δ, and the flow of information is one-way from right to left.

An input w is accepted by an OCA $\mathcal{M}$ if at some time i during its course of

computation the leftmost cell enters an accepting state. The *language accepted by* $\mathcal{M}$ is denoted by $L(\mathcal{M})$. Let $t : \mathbb{N} \to \mathbb{N}$, $t(n) \geq n$, be a mapping. If all $w \in L(\mathcal{M})$ are accepted with at most $t(|w|)$ time steps, then $\mathcal{M}$ is said to be of time complexity t. In the sequel we are particularly interested in OCAs having weak resources. So, we call the time complexity $t(n) = n$ *real time*.

In the following, we study the impact of communication in one-way cellular automata. In general, at every time step each link between two cells can be used to send information, and the communication function b_l can be chosen to be the identity mapping. Here, we limit the bandwidth of the links between the cells, that is, we bound the number of communication symbols by some constant being independent of the number of states. Furthermore, we measure the communication by the number of uses of the links between cells. It is understood that whenever a communication symbol not equal to $\perp$ is sent, a communication takes place. More precisely, the number of communications between cell $i+1$ and cell i up to time step t is defined by

$$\mathrm{com}(i,t) = |\{\, j \mid 0 \leq j < t \text{ and } b_l(c_j(i+1)) \neq \perp) \,\}| \,.$$

For computations we now consider the maximal number of communications between two cells. Let $c_0, c_1, \ldots, c_{t(|w|)}$ be the sequence of configurations computed on input w by some cellular automaton with time complexity $t(n)$, that is, the *computation on* w. Then we define

$$\mathrm{mcom}(w) = \max\{\, \mathrm{com}(i, t(|w|)) \mid 1 \leq i \leq |w| - 1 \,\}.$$

Let $f : \mathbb{N} \to \mathbb{N}$ be a mapping. If all $w \in L(\mathcal{M})$ are accepted with computations where $\mathrm{mcom}(w) \leq f(|w|)$, then $\mathcal{M}$ is said to be *max message bounded by* f. As mentioned before, we are particularly interested in OCAs having weak resources. Therefore, we restrict the communication drastically to one communication symbol and a constant max message bound. Clearly, the minimum is reached if we allow just one use of each link. We denote the class of OCAs which have at most $k \geq 1$ communication symbols and which are max communication bounded by some constant $\ell \geq 1$ by MC(ℓ)-OCA$_k$. For $\bigcup_{\ell \geq 1}$ MC(ℓ)-OCA$_k$ we write MC($O(1)$)-OCA$_k$. The family of all languages which are accepted by some device X in real time is denoted by $\mathscr{L}_{rt}(X)$.

3 Computational capacity

Concerning the constant max communication bound, in [6] it has been shown that the family $\mathscr{L}_{rt}$(MC($O(1)$)-OCA) contains the non-context-free languages $\{\, a_1^n a_2^n \cdots a_k^n \mid n \geq 1 \,\}$, $\{\, a^n b^m c^n d^m \mid n, m \geq 1 \,\}$, as well as the languages $\{\, a^n w \mid n \geq 1 \wedge w \in (b^* c^*)^k b^* \wedge |w|_b = n \,\}$, for all constants $k \geq 0$. All of

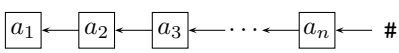

Fig. 1. Initial configuration of a one-way cellular automaton.

these languages are either semilinear or non-bounded. But in contrast to many other computational devices, for example certain multi-head finite automata, parallel communicating finite automata, certain parallel communicating grammar systems, MC($O(1)$)-OCAs can accept non-semilinear bounded languages, for example, the language $\{ a^n b^{n+\lfloor\sqrt{n}\rfloor} \mid n \geq 1 \}$ as well as the language $\{ a^{2^n} b^n c^{2^n+n} \mid n \geq 1 \}$ also belong to the family $\mathscr{L}_{rt}$(MC($O(1)$)-OCA) [5].

Concerning the bound on the number of communication symbols, two-way cellular automata as well as cellular automata with sequential input mode, so-called iterative arrays, have been investigated in [3,4,9,10]. The computational capacity of linear-time one-way cellular automata has been considered in [13].

Next, we study real-time OCAs with minimal communication, that is, the family $\mathscr{L}_{rt}$(MC(1)-OCA$_1$). We start with a minimal input alphabet containing one symbol only. It is known that even unrestricted massively parallel real-time OCAs cannot accept more unary languages than a single deterministic finite-state machine [7]. The following result shows that, to this end, one communication symbol and one use of the links suffice.

Theorem 3.1 *A unary language belongs to the family $\mathscr{L}_{rt}$(MC(1)-OCA$_1$) if and only if it is regular.*

Proof. Trivially, $\mathscr{L}_{rt}$(MC(1)-OCA$_1$) is included in $\mathscr{L}_{rt}$(OCA). Since every unary language belonging to $\mathscr{L}_{rt}$(OCA) is regular [7], the only if part follows.

Now, let $\mathcal{M}$ be some deterministic finite-state machine with a single input symbol a, state set S, initial state s_0, set of accepting states F, and transition function $\delta : S \times \{a\} \to S$. We construct a language equivalent real-time MC(1)-OCA$_1$ $\mathcal{M}' = \langle S', F', A', B', \#, b_l, \delta' \rangle$ by $S' = S \cup \hat{S} \cup \{a, q\}$, where $\hat{S} = \{ \hat{s} \mid s \in S \}$ is a copy of S, $F' = \{ \hat{s} \mid s \in F \}$, $A' = \{a\}$, $B' = \{1\}$,

$$b_l(x) = \bot, \text{ for } x \in \{a, q\} \cup S,$$
$$b_l(x) = 1, \text{ for } x \in \{\#\} \cup \hat{S},$$

and

$$\begin{aligned}
\delta'(a, \bot) &= \delta(s_0, a),\\
\delta'(s, \bot) &= \delta(s, a), \text{ for } s \in S,\\
\delta'(a, 1) &= \hat{s}_2, \text{ for } \delta(s_0, a) = s_2,\\
\delta'(s_1, 1) &= \hat{s}_2, \text{ for } \delta(s_1, a) = s_2 \text{ and } s_1 \in S,\\
\delta'(\hat{s}, \bot) &= q, \text{ for } \hat{s} \in \hat{S},\\
\delta'(q, \bot) &= q.
\end{aligned}$$

Basically, $\mathcal{M}'$ works as follows. During the first time step the rightmost cell receives the message 1 associated with the boundary symbol. Any cell simulates the finite state machine $\mathcal{M}$ until it receives the message 1 from its neighbor. When this happens, the cells simulate one more step of $\mathcal{M}$, but now change to the corresponding state of $\hat{S}$. Subsequently, they enter the state q, which is a rejecting sink

state not sending any information. Since only states from $\hat{S}$ are sending a message, every cell communicates exactly once. The set of accepting states is defined to be the subset of $\hat{S}$ that corresponds to the accepting states of $\mathcal{M}$. On input a^n the leftmost cell of $\mathcal{M}'$ enters a state from $\hat{S}$ once at time step n. It is an accepting state if and only if the finite-state machine $\mathcal{M}$ accepts the input a^n. □

The previous result raises the question whether we can characterize or possibly capture the general regular languages by MC(ℓ)-OCA$_1$s, for some constant ℓ. The next example reveals that one communication symbol and two messages per cell are sufficient to accept a non-regular language. On the other hand, it is shown in [3] that there are regular languages which are not accepted by any real-time CA with 1-bit communication. Thus, there are regular languages which cannot be accepted by any MC($O(1)$)-OCA$_1$.

Example 3.2 We describe the computation of an MC(2)-OCA$_1$ $\mathcal{M}$ on inputs of the form a^nbc^m. The acceptance of the language is governed by two signals. During the first time step the rightmost cell receives the message 1 associated with the boundary symbol and identifies itself to be the rightmost cell. During the second time step the cell with input symbol b (b-cell) sends a message. In this way, the unique a-cell with right neighboring b-cell can identify itself. Subsequently, the a-cell sends the message 1 with speed 1/2, and the rightmost cell sends the message 1 with maximal speed to the left. So, the slow signal starts at time step 2 in the rightmost a-cell, takes $2n-2$ further time steps to reach the leftmost cell, and, thus stays at time steps $2n$ and $2n+1$ in the leftmost cell. The fast signal is set up at time step 1 and takes $n+m$ further time steps to reach the leftmost cell. When both signals meet in a cell, that is, $n+m+1 = 2n+1$, an accepting state is entered. Therefore, the leftmost cell accepts if and only if $n = m$. Moreover, each cell sends at most two messages, and can be set up to send no more messages even for inputs of another form.

Now assume that the language accepted by $\mathcal{M}$ is regular. Since regular languages are closed under intersection, so is the language $L(\mathcal{M}) \cap a^*bc^* = a^nbc^n$, which is non-regular but context free. □

So, we know that MC(ℓ)-OCA$_1$s, for some constant $\ell > 1$, cannot characterize the regular languages. The next two results show that, for any $\ell \geq 1$, there are regular languages over a two-letter alphabet which are not accepted by any MC(ℓ)-OCA$_1$. Moreover, they reveal a tight infinite hierarchy of language classes dependent on the number of messages allowed to be sent by each cell. For all $\ell \geq 1$, we consider the witness languages

$$L_\ell = \{\, w \mid w \in \{a,b\}^+ \text{ and } |w|_b \geq \ell \,\}.$$

Theorem 3.3 *Let $\ell \geq 1$ be a constant. Then language L_ℓ does not belong to the family $\mathscr{L}_{rt}(MC(\ell-1)\text{-}OCA_1)$.*

Proof. In contrast to the assertion, we assume that L_ℓ is accepted by some real-time MC($\ell-1$)-OCA$_1$ $\mathcal{M}$ with state set S.

First, let $\delta(a, b_l(a)) = a_1$, $\delta(a_1, b_l(a_1)) = a_2$, $\delta(a_2, b_l(a_2)) = a_3$, and so on, and consider the sequence $a, a_1, a_2, \ldots$ The sequence becomes cyclic, say, after p time steps with a cycle length of q, that is, $a_p = a_{p+q}$. Clearly, the states in the cycle must not send messages, since otherwise the leftmost cell of input infixes a^i with $i \geq p + \ell q$ would send at least ℓ messages.

Second, for $1 \leq i \leq \ell$, let $w_i = a^{r_1} b a^{r_2} b \cdots a^{r_i} b$, where

$$\begin{aligned} r_1 &= p + (\ell+1)|S|^{\ell} \text{ and} \\ r_{j+1} &= p + (\ell+1)|S|^{r_1+r_2+\cdots+r_j+j+\ell}. \end{aligned}$$

For $1 \leq j \leq i$, we are interested in the time steps at which the leftmost cell of the j-th sequence of adjacent a-cells in w_i (the j-th a-block) *cannot* send a message. With an eye towards further reasoning we consider inputs of the form $u_i w_i$, where $u_i \in \{a, b\}^{\ell}$ (cf. Fig. 2). The leftmost cell of the j-th a-block becomes cyclic at time step p. Since it does not send messages while in the cycle, it does not send messages at time steps t_j with $p \leq t_j \leq r_j - 1$. The b which follows the a-block could break the cycle at time r_j. In particular, there are $r_1 + r_2 + \cdots + r_{j-1} + j - 1 + \ell$ cells on the left of the j-th a-block. Between time p and $r_j = p + (\ell+1)|S|^{r_1+r_2+\cdots+r_{j-1}+j-1+\ell}$ the sequence of these cells evolves without receiving a message from the right. So, the computation of the entire sequence of cells becomes cyclic at time $p + |S|^{r_1+r_2+\cdots+r_{j-1}+j-1+\ell}$, at the latest. Therefore, up to time r_j the cycle has been passed through at least ℓ times which, in turn, implies that in the cycle there may occur no communication at all. Again, the b which follows the j-th a-block could break the cycle at time $r_j + 1$. Taking into account the arrival of a message sent by this b-cell, we derive that the leftmost a-cell of the j'-th a-block, for $1 \leq j' \leq j - 1$, does not send messages at time steps $p + |S|^{r_1+r_2+\cdots+r_{j-1}+j-1+\ell} \leq t_{j'} \leq r_{j'} + 1 + r_{j'+1} + 1 + \cdots + r_j - 1$.

We conclude that the leftmost a-cell of the i-th a-block *cannot* send a message at time steps $p, \ldots, r_i - 1$, and, for $1 \leq j < i$, the leftmost a-cell of the j-th a-block *cannot* send a message at time steps

$$\begin{aligned} &p, \ldots, r_j - 1, \\ &p + |S|^{r_1+r_2+\cdots+r_j+j+\ell}, \ldots, r_j + 1 + r_{j+1} - 1, \\ &p + |S|^{r_1+r_2+\cdots+r_{j+1}+j+1+\ell}, \ldots, r_j + 1 + r_{j+1} + 1 + r_{j+2} - 1, \\ &\vdots \\ &p + |S|^{r_1+r_2+\cdots+r_{i-1}+i-1+\ell}, \ldots, r_j + 1 + r_{j+1} + 1 + \cdots + r_i - 1. \end{aligned}$$

In particular, there are i time periods after time p during which the leftmost a-cell of the first a-block *can* send messages. We denote this periods as sets of time steps

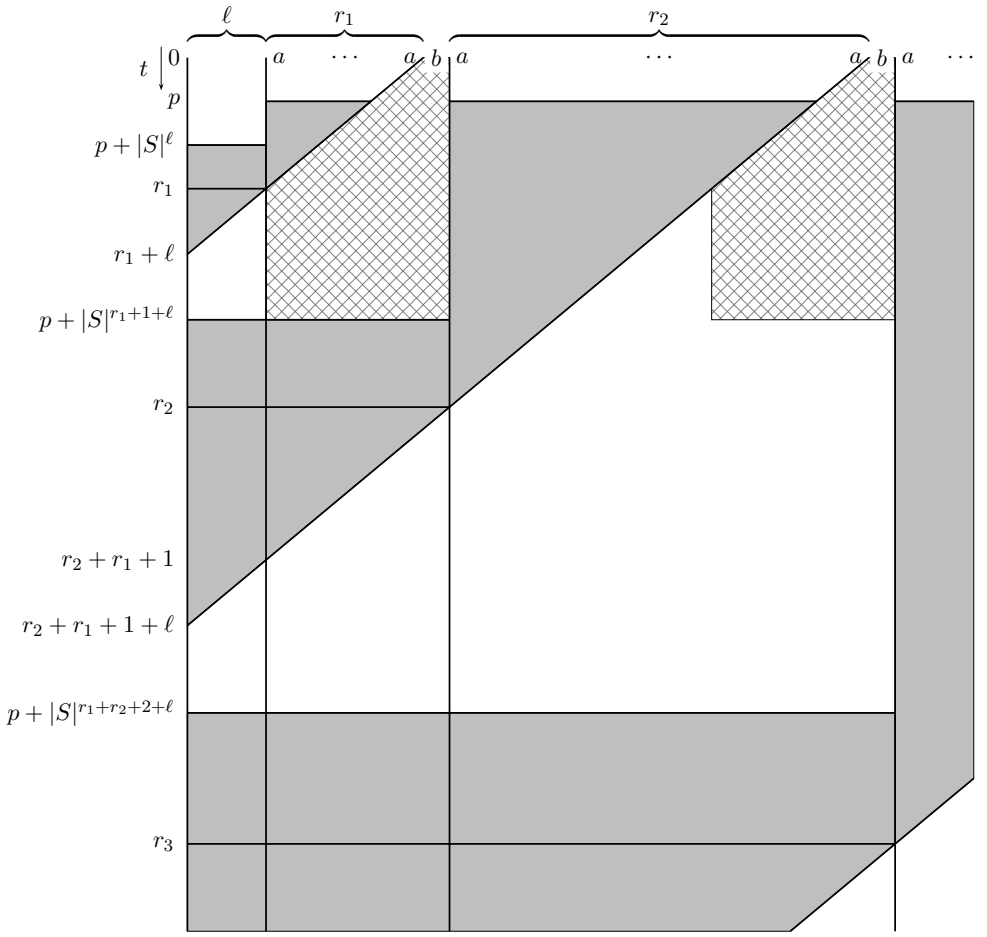

Fig. 2. Schematic space-time diagram of a computation on an input-prefix of $u_i w_i$. In gray-shaded regions there cannot be any communication. For example, the computations in the hatched regions are identical.

by:

$$
\begin{aligned}
T_1 &= \{r_1, \ldots, p + |S|^{r_1+1+\ell} - 1\} \\
T_2 &= \{r_1 + 1 + r_2, \ldots, p + |S|^{r_1+r_2+2+\ell} - 1\} \\
&\vdots \\
T_{i-1} &= \{r_1 + r_2 + \cdots + r_{i-1} + i - 2, \ldots, p + |S|^{r_1+r_2+\cdots+r_{i-1}+i-1+\ell} - 1\} \\
T_i &= \{r_1 + r_2 + \cdots + r_i + i - 1, r_1 + r_2 + \cdots + r_i + i\}
\end{aligned}
$$

Next, we claim that the leftmost a-cell of the first a-block must send at least one message in each period T_j, $1 \leq j \leq i$, which will prove the theorem by setting $i = \ell$. To this end, we fix the input prefixes u_i to be $b^{\ell-i}a^i$, and consider the inputs $u_i w_i$, for $1 \leq i \leq \ell$. All these inputs belong to L_ℓ and, thus, have to be accepted by $\mathcal{M}$.

On input $u_i w_i$, the leftmost cell of $\mathcal{M}$ cannot enter an accepting state before time step $\ell + r_1 + 1 + r_2 + 1 + \cdots + r_i$. Otherwise the input $u_i a^{r_1} b a^{r_2} b \cdots a^{r_i} a \notin L_\ell$ would

be accepted, too. The leftmost a-cell of the i-th a-block cannot send a message from time p to $r_i - 1$. At the latter time all the $\ell + r_1 + 1 + r_2 + 1 + \cdots + r_{i-1} + 1$ cells to its left run through cycles and do not send messages. So, in order to obey real time, the leftmost a-cell of the i-th a-block has to send a message at time r_i or $r_i + 1$. This message has to break the cycles of all the $\ell + r_1 + 1 + r_2 + 1 + \cdots + r_{i-1} + 1$ cells to the left. It can arrive at the leftmost cell at time $\ell + r_1 + r_2 + \cdots + r_i + i - 1$ at the earliest and one time step later at the latest. So, the leftmost a-cell of the first a-block has to send a message at time $r_1 + r_2 + \cdots + r_i + i - 1$ at the earliest (say, this is case Ai) and one time step later at the latest (this is case Bi). Similarly, on an extended input of the form $u_i a^{r_1} b a^{r_2} b \cdots a^{r_i+m} b$, for all $m \geq 1$, the leftmost a-cell of the first a-block has to send a message at time $r_1 + r_2 + \cdots + r_i + i - 1 + m$ or one time step later.

This already shows the claim for $i = 1$.

Now, assume that the claim is true for some $i \geq 1$. Due to the deterministic behavior of $\mathcal{M}$, we conclude that on input $u_{i+1} w_{i+1}$ the leftmost a-cell of the first a-block sends a message in each period $T_1, \ldots, T_{i-1}$. In addition, we know from above, that it sends a message in period T_{i+1}. In case Ai, we observe that the message sent in period T_i (on input $u_i w_i$) is independent of a possible right neighbor of the rightmost symbol b (it could be another symbol or the boundary). So, we obtain also a message in period T_i on input $u_{i+1} w_{i+1}$. This shows the claim for $i+1$ and case Ai.

Assume for a moment that for case Bi there is no message in period T_i on input $u_{i+1} w_{i+1}$. Depending on whether case A_{i+1} or B_{i+1} applies, the leftmost a-cell of the first a-block sends the period T_{i+1} message at time $r_1 + r_2 + \cdots + r_{i+1} + i$ or $r_1 + r_2 + \cdots + r_{i+1} + i + 1$. For case $A(i+1)$ we consider the input $u_{i+1} a^{r_1} b a^{r_2} b \cdots a^{r_i + r_{i+1} + 1} b$, and for case $B(i+1)$ the input $u_{i+1} a^{r_1} b a^{r_2} b \cdots a^{r_i + r_{i+1} + 2} b$. Both inputs do not belong to L_ℓ. According to the computation on the extended input for i, we obtain for $m = r_{i+1}$ and $m = r_{i+1} + 1$ that the leftmost a-cell of the first a-block sends also a message at time $r_1 + r_2 + \cdots + r_{i+1} + i$ or $r_1 + r_2 + \cdots + r_{i+1} + i + 1$. So, in any case a wrong input is accepted. The contradiction shows the claim for $i + 1$ and case Bi and, thus, concludes the induction and the proof. □

In order to derive the infinite tight hierarchy it remains to be shown that language L_ℓ is accepted by some MC(ℓ)-OCA$_1$ in real time.

Theorem 3.4 *Let $\ell \geq 1$ be a constant. Then language L_ℓ belongs to the family $\mathscr{L}_{rt}$(MC(ℓ)-OCA$_1$).*

Proof. Basically, a real-time MC(ℓ)-OCA$_1$ $\mathcal{M}$ accepting L_ℓ works as follows. Each cell with input symbol b sends a message to the left. Whenever a cell receives a message a finite internal counter is increased up to ℓ, and the message is forwarded to the left. After having counted up to ℓ a cell accepts and absorbs any further incoming message. More formally, let $\mathcal{M} = \langle S, F, A, B, \#, b_l, \delta \rangle$, where $S = \{a, b\} \cup$

$\{0,1,\ldots,\ell\} \cup \{\hat{0},\hat{1},\ldots,\hat{\ell}\}$, $F = \{\ell,\hat{\ell}\}$, $A = \{a,b\}$, $B = \{1\}$,

$$b_l(x) = \bot, \text{ for } x \in \{a,b\} \cup \{0,1,\ldots,\ell\},$$
$$b_l(x) = 1, \text{ for } x \in \{\#\} \cup \{\hat{0},\hat{1},\ldots,\hat{\ell}\},$$

and

$$\begin{aligned}
\delta(a,\bot) &= \delta(a,1) = 0,\\
\delta(b,\bot) &= \delta(b,1) = \hat{1},\\
\delta(i,\bot) &= \delta(\hat{i},\bot) = i, \text{ for } 0 \le i < \ell,\\
\delta(i,1) &= \delta(\hat{i},1) = \widehat{i+1}, \text{ for } 0 \le i < \ell,\\
\delta(\ell,\bot) &= \delta(\hat{\ell},\bot) = \ell,\\
\delta(\ell,1) &= \delta(\hat{\ell},1) = \ell.
\end{aligned}$$

States with a hat send messages. During the first time step the internal counters are initialized and the b-cells change to a state that sends a message. The message associated with the boundary symbol is ignored. Roughly speaking, the counters indicate how many symbols b a cell is aware of. So, the counters of the b-cells are initialized with 1. Moreover, cells that are aware of ℓ symbols b accept and ignore further messages from the right. Since a cell sends a message if and only if its counter is increased, any cell sends at most ℓ messages. □

4 Decidability problems

This section is devoted to investigating decidability problems. Despite their sparse communication even for the structurally weak real-time MC($O(1)$)-OCAs several problems are known to be undecidable. These results have been obtained in [6] by reduction of Hilbert's tenth problem. Interestingly, the same is true if one trades *bounded* languages for a number of communications that is at least logarithmic in the length of the input [5]. Here we derive that the problems remain undecidable even for devices with drastically reduced communications, that is, each two neighboring cells may communicate constantly often only where, in addition, the set of communication symbols is reduced to a singleton. Needless to say, the undecidability carries over to all the other stronger devices.

Lemma 4.1 *Let $\mathcal{M} = \langle S,F,A,B,\#,b_l,\delta\rangle$ be a real-time MC($O(1)$)-OCA and $\$ \notin A$ be a new symbol. Then a real-time MC($O(1)$)-OCA$_1$ $\mathcal{M}'$ accepting the language $\{w\$^{(|B|+2)(|w|+1)}v \mid v \in \{\$, A(A \cup \{\$\})^*\}, w \in L(\mathcal{M})\}$ can effectively be constructed.*

Proof. The main task of the construction is to cut down the communication between each two cells to one bit in every time step. Obviously, the given OCA $\mathcal{M}$ may communicate one symbol from $B \cup \{\bot\}$ between each two cells in every time step. Thus, the principal idea of the construction is to insert a certain number of new symbols for each input symbol at the end of the input and to use these new

symbols, that is, the additional time provided by them, to encode the information to be sent.

First, we observe that $\mathcal{M}$ can be modified by introducing an additional time step such that in the first time step no communication takes place but the communication associated with the boundary symbol. Moreover, we may assume that the communication symbol associated with the boundary symbol is a new unique one, say $b_{\#}$. Next, we add $m = |B| + 2$ symbols \$ for each input symbol and the boundary symbol. We encode $B \cup \{\bot, b_{\#}\}$ by the following function γ. Let $B = \{b_1, b_2, \ldots, b_{|B|}\}$. Then $\gamma(\bot) = 0^m$, $\gamma(b_{\#}) = 1^m$, and $\gamma(b_i) = 10^i 10^{m-i-2}$, for $i = 1, 2, \ldots, |B|$.

Let $\mathcal{M}' = \langle S', F', A \cup \{\$\}, B', \#, b'_l, \delta' \rangle$ be an OCA where B' consists of the sole element 1 only. We start by considering inputs of the form $w\$^{m(|w|+1)}\$$, where $w \in A^+$, that is, $v = \$$. In the first time step the only communication is the one associated with the boundary symbol. The remaining cells are not communicating. In this way, the rightmost \$-cell can identify itself. In each of the next m time steps every \$-cell but the rightmost one sends a 1 to its left neighbor as long as itself receives a 1 from its right neighbor. Since the rightmost \$-cell does not communicate at time step 1, the $(m+1)$st \$-cell from the right does not receive a 1 at time m. In the next time step, this cell starts a signal C by sending a 1 to its left neighbor which is forwarded with maximal speed to the left by all other \$-cells. It should be remarked that by setting up an internal counter every cell can identify the $(m+1)$st time step.

The cells carrying an input symbol from the set A do not communicate in the first $m+1$ time steps. (Due to the above modification of the given OCA $\mathcal{M}$ this will not cause any delay.) The $|w|$-th cell from the left receives the encoding $\gamma(b_{\#}) = 1^m$ during the time steps $2, 3, \ldots, m+1$, and can simulate one step of $\mathcal{M}$ suitably. After the $(m+1)$st time step, the leftmost $|w|$ cells simulate the computation of $\mathcal{M}$ by sending the encodings of the communication symbols in m time steps, whereby a 1 is sent for a 1, and nothing is sent for a 0. With an eye towards checking the correct number of \$ symbols we provide an additional time step $m+1$.

In addition to the simulation, a signal D with speed $1/(m+1)$ is started in the $|w|$-th cell at time $2(m+1)$. In order to realize a signal with that speed, an internal counter is increased at every time step. When the counter reaches $m+1$, a message 1 is sent to the left neighboring cell which now starts a counting up to $m+1$, and so on. The signal D reaches the leftmost cell at time $2(m+1) + (|w|-1)(m+1) = (|w|+1)(m+1)$. The signal D cannot interfere with with sending encodings, because the latter are sent at every time steps $1, 2, \ldots, m$ and the former is sent at most at time steps $m+1$. Since $\mathcal{M}$ obeys real time plus one time step, every cell which has been passed by D can be obliged to stop its simulation of $\mathcal{M}$. It now forwards any incoming 1 to the left.

Let us come back to the \$-cells. The $(m+1)$st \$-cell from the right has started the signal C at time $(m+1)$. This signal reaches the $|w|$-th cell at time step $m(|w|+1)+2$ and is forwarded to the left by the remaining cells. Thus, it reaches the leftmost cell at time step $m(|w|+1)+2+|w|-1 = (m+1)(|w|+1)$, that is,

at the same time step at which signal D reaches the cell. In this case, the input is accepted if the cell, in addition, simulates an accepting state of $\mathcal{M}$. See Fig. 3 for a schematic computation.

Next, we want to show that $\mathcal{M}'$ performs only a constant number of communications between each two cells. To this end, we observe that each cell communicates at most m messages during the first $m+1$ time steps. Due to the encoding chosen, every communication symbol of $\mathcal{M}$ requires to send two messages in $\mathcal{M}'$. Finally, the signal D as well as the signal C each cause one additional communication step per cell. Altogether, $\mathcal{M}'$ is a real-time MC($O(1)$)-OCA$_1$.

Concerning the accepted inputs we know that an input is accepted if and only if the signals C and D meet in the leftmost cell, which simulates an accepting state of $\mathcal{M}$. The simulation of $\mathcal{M}$ can only be accepting if initially the encoding of the unique symbol $b_{\#}$ is communicated. This, in turn, can only be done by a \$-cell. So, any accepted input starts with a prefix $w \in L(\mathcal{M})$ followed by a number of \$-symbols. Moreover, since the signal C has to arrive in due time, it has to be set up appropriately. Since in the first $m+1$ time steps A-cells do not communicate whereas all \$-cells try to send the encoding 1^m, all but the rightmost of the number of \$-cells send at least one message. Thus, all but the rightmost \$-cell can identify an A-cell to the right within the first $m+1$ time steps and then block the computation. So, any accepted input is either of the form $w\$^{m(|w|+1)}\$$ or $w\$^{m(|w|+1)}xu$, where $w \in L(\mathcal{M})$, $x \in A$, and $u \in A(A \cup \{\$\})^*$. Conversely, every input of one of these forms in accepted. □

It is straightforward to generalize this construction for cellular automata with two-way communication. Furthermore, since the construction increases the number of communication steps per cell only linearly, it is easy to see that the construction

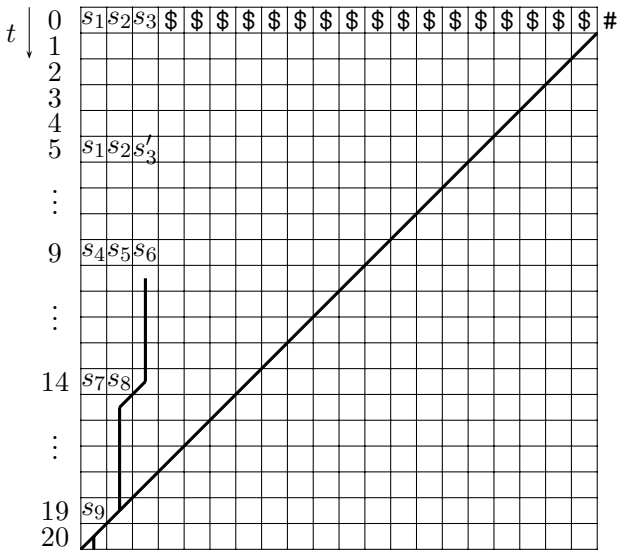

Fig. 3. Schematic simulation of an MC($O(1)$)-OCA $\mathcal{M}$ with $|B| = 2$ on input $s_1 s_2 s_3$, which evolves in its relevant part in the next time steps to $s_1 s_2 s_3'$, $s_4 s_5 s_6$, $s_7 s_8$, and s_9 at time steps $5, 9, 14, 19$. At time step 20 an accepting state of $\mathcal{M}'$ is entered if the simulated state s_9 is an accepting state in $\mathcal{M}$.

also works for SC(f)-CAs and MC(f)-CAs where f is not necessarily a constant function. Here SC(f)-CA denotes the class of cellular automata whose total number of communications (appearing anywhere in the array) during a computation is bounded by f. For a proper definition of these models we would like to refer to [6,5].

Now, we can use Lemma 4.1 to reduce the undecidability problems for MC($O(1)$)-OCAs shown in [6] to MC($O(1)$)-OCA$_1$.

Theorem 4.2 *Given an arbitrary real-time MC($O(1)$)-OCA$_1$ $\mathcal{M}$, it is undecidable whether $L(\mathcal{M})$ is empty.*

Proof. The main idea to prove the undecidability of the emptiness problem in [6] is to construct a real-time MC($O(1)$)-OCA $\mathcal{M}$ which accepts a certain language $L(p)$. From this fact the emptiness of $L(p)$ can be related to the non-existence of solutions for Hilbert's tenth problem whose known undecidability implies the undecidability of emptiness for real-time MC($O(1)$)-OCAs.

By Lemma 4.1 we can construct a real-time MC($O(1)$)-OCA$_1$ $\mathcal{M}'$ for a given real-time MC($O(1)$)-OCA $\mathcal{M}$ over the alphabet A which accepts the language

$$L(\mathcal{M}') = \{w\$^{m(|w|+1)}v \mid v \in \{\$, A(A \cup \{\$\})^*\}, w \in L(\mathcal{M})\}$$

for a suitable m depending on $\mathcal{M}$. Since $L(\mathcal{M})$ is empty if and only if $L(\mathcal{M}')$ is empty, we obtain the undecidability of emptiness for real-time MC($O(1)$)-OCA$_1$s.□

Theorem 4.3 *Emptiness, finiteness, infiniteness, equivalence, inclusion, regularity, and context-freedom are undecidable for arbitrary real-time MC($O(1)$)-OCA$_k$s with $k \geq 1$.*

Proof. It is sufficient to show all claims for $k = 1$. At first, we show that the undecidability of emptiness implies the undecidability of inclusion and equivalence as well. Let $\mathcal{M}$ be an arbitrary real-time MC($O(1)$)-OCA$_1$ and $\mathcal{M}'$ be a real-time MC($O(1)$)-OCA$_1$ accepting the empty language. If we could decide the inclusion or equivalence of $\mathcal{M}$ and $\mathcal{M}'$, we could decide the emptiness of $\mathcal{M}$ as well which is a contradiction to Theorem 4.2.

Next, by using Lemma 4.1 we construct a real-time MC($O(1)$)-OCA$_1$ $\mathcal{M}'$ for a given real-time MC($O(1)$)-OCA $\mathcal{M}$ over the alphabet A which accepts the language $L(\mathcal{M}') = \{w\$^{m(|w|+1)}v \mid v \in \{\$, A(A \cup \{\$\})^*\}, w \in L(\mathcal{M})\}$ for a suitable m depending on $\mathcal{M}$. Since $L(\mathcal{M}')$ is finite (infinite) if and only if $L(\mathcal{M}')$ is empty (non-empty), we obtain the undecidability of finiteness and infiniteness for real-time MC($O(1)$)-OCA$_1$s from the undecidability of emptiness for real-time MC($O(1)$)-OCAs shown in [6].

We turn to show that $L(\mathcal{M}')$ is regular if and only if $L(\mathcal{M})$ is finite. Clearly, $L(\mathcal{M}')$ is regular, if $L(\mathcal{M})$ is finite. To obtain that $L(\mathcal{M}')$ is not regular if $L(\mathcal{M})$ is infinite, we assume that $L(\mathcal{M}')$ is regular. Then, the intersection of $L(\mathcal{M}')$ with the regular language $A^*\$^*$ gives the regular language $L'(\mathcal{M}') = \{w\$^{m(|w|+1)+1} \mid w \in L(\mathcal{M})\}$. By an obvious application of the pumping lemma for regular languages, we can show that $L'(\mathcal{M}')$ is not regular. This gives the contradiction and shows the undecidability of regularity.

Finally, let $\mathcal{M}_1$ be an arbitrary real-time MC($O(1)$)-OCA accepting a language over some alphabet A. It is not difficult to construct a real-time MC($O(1)$)-OCA $\mathcal{M}_2$ accepting the language $\{w\mathtt{¢}^{|w|} \mid w \in \mathcal{M}_1\}$ where $\mathtt{¢} \notin A$. By using Lemma 4.1 we can construct a real-time MC($O(1)$)-OCA$_1$ $\mathcal{M}'$ which accepts the language

$$L(\mathcal{M}') = \{w\mathtt{¢}^{|w|}\mathtt{\$}^{m(2|w|+1)}v \mid v \in \{\mathtt{\$}, (A \cup \{\mathtt{¢}\})(A \cup \{\mathtt{¢}, \mathtt{\$}\})^*\}, w \in L(\mathcal{M}_1)\}$$

for a suitable constant m. If $L(\mathcal{M}_1)$ is finite, then $L(\mathcal{M}')$ is regular and, thus, context free. Next, we want to show that $L(\mathcal{M}')$ is not context free, if $L(\mathcal{M}_1)$ is infinite. By way of contradiction, we assume that $L(\mathcal{M}')$ is context free. Then the intersection of $L(\mathcal{M}')$ with the regular language $A^*\mathtt{¢}^*\mathtt{\$}^*$ gives the context-free language $L'(\mathcal{M}') = \{w\mathtt{¢}^{|w|}\mathtt{\$}^{m(2|w|+1)+1} \mid w \in L(\mathcal{M}_1)\}$. By an obvious application of the pumping lemma for context-free languages, one can show that $L'(\mathcal{M}')$ is not context free. This gives the contradiction and shows that $L(\mathcal{M}_1)$ is finite if and only if $L(\mathcal{M}')$ is context free. In this way, we obtain the undecidability of testing context-freedom for real-time MC($O(1)$)-OCA$_1$s from the undecidability of finiteness for real-time MC($O(1)$)-OCAs shown in [6]. □

Clearly, the undecidability carries over to, for example, MC($O(1)$)-CA$_k$ with two-way communication and the models SC($O(n)$)-OCA$_k$ and SC($O(n)$)-CA$_k$ having a different measure on the number of communications (cf. [6,5]). In particular, the following corollary shows the undecidability of almost all decidability problems for k-bit one-way cellular automata studied in [13].

Corollary 4.4 *Emptiness, finiteness, infiniteness, equivalence, inclusion, regularity, and context-freedom are undecidable for arbitrary real-time OCA$_k$s with $k \geq 1$.*

References

[1] Kutrib, M., *Cellular automata – a computational point of view*, in: *New Developments in Formal Languages and Applications*, Springer, 2008 pp. 183–227.

[2] Kutrib, M., *Cellular automata and language theory*, in: *Encyclopedia of Complexity and System Science*, Springer, 2009 in press.

[3] Kutrib, M. and A. Malcher, *Fast cellular automata with restricted inter-cell communication: Computational capacity*, in: *Theoretical Computer Science (IFIP TCS2006)*, IFIP **209** (2006), pp. 151–164.

[4] Kutrib, M. and A. Malcher, *Fast iterative arrays with restricted inter-cell communication: Constructions and decidability*, in: *Mathematical Foundations of Computer Science (MFCS 2006)*, LNCS **4162** (2006), pp. 634–645.

[5] Kutrib, M. and A. Malcher, *Bounded languages meet cellular automata with sparse communication*, in: *Descriptional Complexity of Formal Systems (DCFS 2009)*, 2009.

[6] Kutrib, M. and A. Malcher, *Cellular automata with sparse communication*, in: *Implementation and Application of Automata (CIAA 2009)*, LNCS **5642** (2009), pp. 34–43.

[7] Seidel, S. R., *Language recognition and the synchronization of cellular automata*, Technical Report 79-02, Department of Computer Science, University of Iowa, Iowa City (1979).

[8] Umeo, H., *Linear-time recognition of connectivity of binary images on 1-bit inter-cell communication cellular automaton*, Parallel Comput. **27** (2001), pp. 587–599.

[9] Umeo, H. and N. Kamikawa, *A design of real-time non-regular sequence generation algorithms and their implementations on cellular automata with 1-bit inter-cell communications*, Fund. Inform. **52** (2002), pp. 257–275.

[10] Umeo, H. and N. Kamikawa, *Real-time generation of primes by a 1-bit-communication cellular automaton*, Fund. Inform. **58** (2003), pp. 421–435.

[11] Vollmar, R., *On cellular automata with a finite number of state changes*, Computing **3** (1981), pp. 181–191.

[12] Vollmar, R., *Some remarks about the 'efficiency' of polyautomata*, Internat. J. Theoret. Phys. **21** (1982), pp. 1007–1015.

[13] Worsch, T., *Linear time language recognition on cellular automata with restricted communication*, in: *LATIN 2000: Theoretical Informatics*, LNCS **1776** (2000), pp. 417–426.