IFIG
RESEARCH
REPORT

Institut für Informatik
JLU Gießen
Arndtstraße 2
D-35392 Giessen, Germany
Tel: +49-641-99-32141
Fax: +49-641-99-32149
mail@informatik.uni-giessen.de
www.informatik.uni-giessen.de

INSTITUT FÜR INFORMATIK

ITERATIVE ARRAYS WITH A WEE BIT ALTERNATION

Thomas Buchholz   Andreas Klein
Martin Kutrib

IFIG RESEARCH REPORT 9905
APRIL 1999

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN

IFIG Research Report

IFIG Research Report 9905, April 1999

Iterative arrays with a wee bit alternation

Thomas Buchholz[1] Andreas Klein[2]
Martin Kutrib[3]

Institute of Informatics, University of Giessen
Arndtstr. 2, D-35392 Giessen, Germany

Abstract. An iterative array is a line of interconnected interacting finite automata. One distinguished automaton, the communication cell, is connected to the outside world and fetches the input serially symbol by symbol. We are investigating iterative arrays with an alternating communication cell. All the other automata are deterministic. The number of alternating state transitions is regarded as a limited resource which depends on the length of the input.
We center our attention to real-time computations and compare alternating IAs with nondeterministic IAs. By proving that the language families of the latter are not closed under complement for sublogarithmic limits it is shown that alternation is strictly more powerful than nondeterminism. Moreover, for these limits there exist infinite hierarchies of properly included alternating language families. It is shown that these families are closed under boolean operations.

CR Subject Classification (1998): F.1, F.4.3, B.6.1, E.4

[1]E-mail: buchholz@informatik.uni-giessen.de
[2]E-mail: andreas.klein@informatik.uni-giessen.de
[3]E-mail: kutrib@informatik.uni-giessen.de

1 Introduction

Devices of interconnected parallel acting automata have extensively been investigated from a language theoretic point of view. The specification of such a system includes the type and specification of the single automata, the interconnection scheme (which sometimes implies a dimension to the system), a local and/or global transition function and the input and output modes. One-dimensional devices with nearest neighbor connections whose cells are deterministic finite automata are commonly called iterative arrays (IA) if the input mode is sequential to a distinguished communication cell.

Especially for practical reasons and for the design of systolic algorithms a sequential input mode is more natural than the parallel input mode of so-called cellular automata. Various other types of acceptors have been investigated under this aspect (e.g., the iterative tree acceptors in [6]).

In connection with formal language recognition IAs have been introduced in [5] where it was shown that the language families accepted by real-time IAs form a Boolean algebra not closed under concatenation and reversal. Moreover, there exists a context-free language that cannot be accepted by any d-dimensional IA in real-time. On the other hand, in [4] it is shown that for every context-free grammar a 2-dimensional linear-time IA parser exists. Compared with e.g., Turing machines there are essential differences in the recognition power. For example, the language of palindromes needs a lower bound of n^2 time steps for Turing machines but is acceptable in real-time by IAs.

In [7] a real-time acceptor for prime numbers has been constructed. Pattern manipulation is the main aspect in [1]. A characterization of various types of IAs by restricted Turing machines and several results, especially speed-up theorems, are given in [8, 9, 10].

Various generalizations of IAs have been considered. In [11] IAs are studied in which all the finite automata are additionally connected to the communication cell. Several more results concerning formal languages can be found e.g., in [12, 13, 14].

Sometimes completely nondeterministic arrays have been studied. In [3] arrays with restricted nondeterminism have been introduced. There it has been shown that the number of nondeterministic transitions can be reduced by a constant factor and that there exists an infinite hierarchy of properly included language families for necessarily sublogarithmic limits. Some closure properties for such families are given.

Here we continue the work initiated in [3] by making a further generalization step. We introduce arrays with restricted alternation. Our interest focuses on the question how much alternation is required, if at all, to enhance the power of a particular (nondeterministic) class. Thereby we are trying to identify the power and limitations of commonly known iterative arrays. In order to define alternations as limited resource we restrict the ability to perform alternating transitions to the communication cell, all the other automata are deterministic ones. Moreover, we limit the number of allowed alternating transitions which additionally have to appear at the beginning of the computation. Our attention is centered on real-time computations.

The basic notions and the model in question are defined in the next section. Section 3 is devoted to technical results mainly. By generalizing a method in [5] an equivalence relation is used to define a necessary condition for real-time languages. Another result states that for a given alternating iterative array one can always find another one that accepts the same language and that uses existential and universal states by turns at every time step. In Section 4 the closure under Boolean operations is investigated. Comparing alternating iterative arrays to nondeterministic ones for sublogarithmic limits in Section 5 it is shown that the former are strictly more powerful. The properness of the inclusion is proved at the hand of different closure properties. In particular the nondeterministic families are not closed under complement, a question left open in [3]. Finally we obtain infinite hierarchies of properly included language families varying the amount of allowed alternation.

2 Model and Notions

We denote the positive rational numbers by $\mathbb{Q}_+$, the integers by $\mathbb{Z}$, the positive integers $\{1, 2, \dots\}$ by $\mathbb{N}$, the set $\mathbb{N} \cup \{0\}$ by $\mathbb{N}_0$ and the powerset of a set S by 2^S. The empty word is denoted by ϵ and the reversal of a word w by w^R.

An iterative array with alternating communication cell is an infinite linear array of finite automata, sometimes called cells, each of them is connected to its both nearest neighbors to the left and to the right. For our convenience we identify the cells by integers. Initially they are in the so-called quiescent state. The input is supplied sequentially to the distinguished communication cell at the origin. For this reason we have two local transition functions. The state transition of all cells but the communication cell depends on the actual state of the cell itself and the actual states of its both neighbors. The state transition of the communication cell additionally depends on the actual input symbol (or if the whole input has been consumed on a special end-of-input symbol). The finite automata work synchronously at discrete time steps. Their states are partitioned into existential and universal ones. What makes a, so far, nondeterministic computation to an alternating computation is the mode of acceptance, which will be defined with respect to the partitioning. More formally:

Definition 1 *An* iterative array with alternating communication cell *(*A-IA*)* *is a system* $(S, \delta, \delta_{nd}, s_0, \#, A, F)$*, where*

a) S *is the finite, nonempty set of* states *which is partitioned into* existential S_e *and* universal S_u *states:* $S = S_e \cup S_u$*,*
b) A *is the finite, nonempty set of* input symbols,
c) $F \subseteq S$ *is the set of* accepting states,
d) $s_0 \in S$ *is the* quiescent state,
e) $\# \notin A$ *is the* end-of-input symbol,

f) $\delta : S^3 \to S$ *is the deterministic* local transition function for non-communication cells *satisfying* $\delta(s_0, s_0, s_0) = s_0$,

g) $\delta_{nd} : S^3 \times (A \cup \{\#\}) \to 2^S$ *is the* local transition function for the communication cell *satisfying* $\forall s_1, s_2, s_3 \in S, a \in A \cup \{\#\} : \delta_{nd}(s_1, s_2, s_3, a) \neq \emptyset$.

Let $\mathcal{M}$ be an A-IA. A configuration of $\mathcal{M}$ at some time $t \geq 0$ is a description of its global state, which is actually a pair (w, c_t), where $w \in A^*$ is the remaining input sequence and $c_t : \mathbb{Z} \to S$ is a mapping that gives the actual states of the single cells. The configuration (w, c_0) at time 0 is defined by the input word w and the mapping $c_0(i) := s_0$, $i \in \mathbb{Z}$, while subsequent configurations are chosen according to the global transition Δ_{nd}: Let (w, c) be a configuration then the possible successor configurations (w', c') are as follows:

$$(w', c') \in \Delta_{nd}\big((w, c)\big) \iff \begin{array}{l} c'(i) = \delta\big(c(i-1), c(i), c(i+1)\big), i \in \mathbb{Z} \setminus \{0\}, \\ c'(0) \in \delta_{nd}\big(c(-1), c(0), c(1), a\big) \end{array}$$

where $a = \#$ and $w' = \epsilon$ if $w = \epsilon$, and $a = w_1$ and $w' = w_2 \cdots w_n$ if $w = w_1 \cdots w_n$. Thus, the global transformation Δ_{nd} is induced by δ and δ_{nd}. The i-fold composition of Δ_{nd} is defined as follows:

$$\Delta^0_{nd}\big((w, c)\big) := \big\{(w, c)\big\}, \qquad \Delta^{i+1}_{nd}\big((w, c)\big) := \bigcup_{(w', c') \in \Delta^i_{nd}((w, c))} \Delta_{nd}\big((w', c')\big)$$

The evolution of $\mathcal{M}$ is represented by its computation tree.

The *computation tree* $T_{\mathcal{M},w}$ of $\mathcal{M}$ under input $w \in A^+$ is a tree whose nodes are labeled by configurations. The root of $T_{\mathcal{M},w}$ is labeled by (w, c_0). The children of a node labeled by a configuration (w, c) are the nodes labeled by the possible successor configurations of (w, c). Thus, the node (w, c) has exactly $|\Delta_{nd}\big((w, c)\big)|$ children.

A configuration (w, c) is *accepting* iff $c(0) \in F$, it is *universal* iff $c(0) \in S_u$ and it is said to be *existential* iff $c(0) \in S_e$.

In order to define *accepting computations* on input words we need the notion of accepting subtrees.

Definition 2 *Let* $\mathcal{M} = (S, \delta, \delta_{nd}, s_0, \#, A, F)$ *be an* A-IA *and* $T_{\mathcal{M},w}$ *be its computation tree for an input word* $w \in A^n$, $n \in \mathbb{N}$. *A finite subtree* T' *of* $T_{\mathcal{M},w}$ *is said to be an* accepting subtree *iff it fulfills the following conditions:*

a) The root of T' *is the root of* $T_{\mathcal{M},w}$.

b) If a non-leaf node of T' *is labeled by an universal configuration then all its successors belong to* T'.

c) If a non-leaf node of T' *is labeled by an existential configuration then exactly one of its successors belongs to* T'.

d) The leafs of T' *are labeled by accepting configurations.*

From the computational point of view an accepting subtree is built by considering one possible successor (a guessed successor) if the communication cell is in an existential state and by considering all successors if the communication cell is in an universal state.

Now we are prepared to define the language accepted by an A-IA.

Definition 3 *Let $\mathcal{M} = (S, \delta, \delta_{nd}, s_0, \#, A, F)$ be an* A-IA.

a) *A word $w \in A^+$ is* accepted *by $\mathcal{M}$ iff there exists an accepting subtree of $T_{\mathcal{M},w}$.*

b) *$L(\mathcal{M}) = \{w \in A^+ \mid w$ is accepted by $\mathcal{M}\}$ is the* language accepted *by $\mathcal{M}$.*

c) *Let $t : \mathbb{N} \to \mathbb{N}$, $t(n) > n$, be a mapping. Iff for all $w \in L(\mathcal{M})$ there exists an accepting subtree of $T_{\mathcal{M},w}$ the height of which is less than $t(|w|)$, then L is said to be of* time complexity *t.*

An A-IA $\mathcal{M}$ has a *nondeterministic* communication cell if the state set consists of existential states only. An accepting subtree is now a list of configurations which corresponds to a possible computation path of $\mathcal{M}$. Iterative arrays with nondeterministic communication cell are denoted by G-IA.

A G-IA is deterministic if $\delta_{nd}(s_1, s_2, s_3, a)$ is a singleton for all states $s_1, s_2, s_3 \in S$ and all input symbols $a \in A \cup \{\#\}$. In these cases the course of computation is unique for a given input word w and, thus, the whole computation tree is a list of configurations. Deterministic iterative arrays are denoted by IA.

If the state set is a Cartesian product of some smaller sets $S = S_0 \times S_1 \times \cdots \times S_r$, we will use the notion *register* for the single parts of a state. The concatenation of a specific register of all cells forms a *track*.

The family of all languages which can be accepted by an A-IA with time complexity t is denoted by $\mathscr{L}_t$(A-IA). In the sequel we will use a corresponding notion for other types of acceptors. If $t(n)$ equals $n+1$ acceptance is said to be in real-time and we write $\mathscr{L}_{rt}$(A-IA). The *linear-time* languages $\mathscr{L}_{lt}$(A-IA) are defined according to $\mathscr{L}_{lt}(\text{A-IA}) := \bigcup_{k \in \mathbb{Q}_+,\, k>1} \mathscr{L}_{k \cdot n}(\text{A-IA})$.

There is a natural way to restrict the alternation of the arrays. One can limit the number of allowed alternating state transitions of the communication cell. Note, here we do not limit the number of alternations (i.e., transitions from an universal to an existential state or vice versa) but the number of time steps at which alternating transitions may occur. For this reason a deterministic local transition function $\delta_d : S^3 \times (A \cup \{\#\}) \to S$ for the communication cell is provided and the global transition induced by δ and δ_d is denoted by Δ_d. Let $f : \mathbb{N} \to \mathbb{N}_0$ be a mapping that gives the number of allowed alternating transitions dependent on the length of the input.

The resulting system $(S, \delta, \delta_{nd}, \delta_d, s_0, \#, A, F)$ is a fA-IA (f alternating IA) if starting with the initial configuration (w, c_0) the possible configurations at some time t are given by the global transition as follows:

$\{(w, c_0)\}$ if $t = 0$, $\Delta_{nd}^t\big((w, c_0)\big)$ if $t \leq f(|w|)$ and

$$\bigcup_{(w',c') \in \Delta_{nd}^{f(|w|)}((w,c_0))} \Delta_d^{t-f(|w|)}\big((w', c')\big) \quad \text{otherwise}$$

Observe that all alternating transitions have to be applied before the deterministic ones. Up to now we have f not required to be computable at all. Of course for almost all applications we will have to do so but some of our general results can be developed without such a requirement.

3 Equivalence Classes and Normalization

Definition 4 *Let $L \subseteq A^*$ be a language over an alphabet A and $l \in \mathbb{N}$ be a constant. Two words w and w' are* l-equivalent *with respect to L iff $ww_l \in L \iff w'w_l \in L$ for all $w_l \in A^l$. The* number of l-equivalence classes of words of length n with respect to L *are denoted by $N(n, l, L)$ (i.e. $|ww_l| = n$).*

The following lemma gives a necessary condition for a language to be real-time acceptable by an fA-IA.

Lemma 5 *Let $f : \mathbb{N} \to \mathbb{N}_0$, $f(n) \leq n$, be a mapping. If $L \in \mathscr{L}_{rt}(f\text{A-IA})$ then there exist constants $p, q \in \mathbb{N}$ such that $N(n, l, L) \leq p^{l \cdot q^{f(n)}}$.*

Proof. Let $\mathcal{M} = (S, \delta, \delta_{nd}, \delta_d, s_0, \#, A, F)$ be a real-time fA-IA which accepts L. We define $q := \max\big\{|\delta_{nd}(s_1, s_2, s_3, a)| \;\big|\; s_1, s_2, s_3 \in S, a \in A\big\}$.

In order to determine an upper bound to the number of l-equivalence classes at first we consider the possible configurations of $\mathcal{M}$ after reading all but l input symbols. The remaining computation depends on the last l input symbols and the states of the cells $-l-1, \ldots, 0, \ldots, l+1$. For the $2l+3$ states there are $|S|^{2l+3}$ different possibilities. Let $p_1 := |S|^5$ then due to $|S|^{2l+3} = |S|^{2l} \cdot |S|^3 = \big(|S|^2\big)^l \cdot |S|^3 \leq \big(|S|^2\big)^l \cdot \big(|S|^3\big)^l = \big(|S|^2 \cdot |S|^3\big)^l = p_1^l$ we have at most p_1^l different possibilities.

Now we consider the computation trees of $\mathcal{M}$. Since the number of alternating steps is bounded by $f(n)$ in each computation tree there are at most $q^{f(n)}$ internal nodes that are labeled by existential or universal configurations (all the others are part of the deterministic computation) we have to distinguish $2^{q^{f(n)}}$ different labelings. Each computation tree of finite height has at most $q^{f(n)}$ leafs. Each leaf at level $n-l$ can be labeled with one of the p_1^l different configurations. Since the number of equivalence classes is not affected by the last l input symbols altogether one can distinguish $(p_1^l)^{q^{f(n)}} \cdot 2^{q^{f(n)}}$ different computation trees of height $n-l$. Correspondingly, there are at most $p_1^{l \cdot q^{f(n)}} \cdot 2^{q^{f(n)}}$ classes. For a suitable $p \in \mathbb{N}$ this is less than $p^{l \cdot q^{f(n)}}$. □

If $\mathcal{M}$ is a fG-IA for the number of equivalence classes we need not to take the labelings into account. Thus, we obtain less than $p_1^{l \cdot q^{f(n)}}$ classes.

Now we are going to extend the previous lemma. The question is how the number of l-equivalence classes is affected if we concatenate each word of L by another arbitrary l symbols from A.

Lemma 6 *Let $f : \mathbb{N} \to \mathbb{N}_0$, $f(n) \leq n$, be an increasing mapping that satisfies $f(2n) \leq 2f(n)$. If the number of l-equivalence classes with respect to a language $L \subseteq A^*$ is not bounded according to Lemma 5 then $L \bullet A^l \notin \mathscr{L}_{rt}(f\text{G-IA})$.*

Proof. At first we prove $N(n+l+1, 2 \cdot l+1, L \bullet A^l) = N(n, l, L)$.

From $ww_l \in L$ for an arbitrary $w_l \in A^l$ it follows $ww_l \bullet w'_l \in L \bullet A^l$ for all $w'_l \in A^l$ and $w'w_l \in L$. From $w'w_l \in L$ it follows $w'w_l \bullet w'_l \in L \bullet A^l$ for all $w'_l \in A^l$.

Conversely, let w and w' be $(2l+1)$-equivalent with respect to $L \bullet A^l$. From $ww_l \bullet w'_l \in L \bullet A^l$ for an arbitrary $w \in A^l$ and all $w'_l \in A^l$ it follows $ww_l \in L$ and $w'w_l \bullet w'_l \in L \bullet A^l$. From the latter we obtain $w'w_l \in L$.

Secondly, there exist n and l such that we have $N(n, l, L) > p^{l \cdot q^{f(n)}}$ for every $p, q \in \mathbb{N}$, since the number of l-equivalence classes with respect to L is not bounded according to Lemma 5 (i.e., $L \notin \mathscr{L}_{rt}(f\text{G-IA})$).

On the other hand, a real-time fG-IA can distinguish at most $p^{(2 \cdot l+1) \cdot q^{f(n+l+1)}}$ equivalence classes with respect to $L \bullet A^l$. Since $l < n$ it follows $p^{(2 \cdot l+1) q^{f(n+l+1)}} \leq p^{(2 \cdot l+1) q^{2f(n)}} \leq p'^{l \cdot q'^{f(n)}} < N(n, l, L) = N(n+l+1, 2 \cdot l+1, L \bullet A^l)$.

Thus, $L \bullet A^l \notin \mathscr{L}_{rt}(f\text{G-IA})$ by Lemma 5. □

In order to reduce the technical effort for proofs it is often useful to be able to start with devices that meet a certain normal-form. For our purposes it is convenient to consider iterative arrays which are *alternation normalized* as follows: $s_0 \in S_e$ and $\forall\, s_1, s_2, s_3 \in S, a \in A \cup \{\#\} : \delta_{nd}(s_1, s_2, s_3, a) \subseteq S_e$ if $s_2 \in S_u$ and $\delta_{nd}(s_1, s_2, s_3, a) \subseteq S_u$ if $s_2 \in S_e$.

Thus the communication cell changes continually from an existential state into an universal state and vice versa.

Lemma 7 *Let $f : \mathbb{N} \to \mathbb{N}_0$, $f(n) \leq n+1$, and $t : \mathbb{N} \to \mathbb{N}$, $t(n) > n$, be two mappings. If $L \in \mathscr{L}_t(f\text{A-IA})$ then exists an alternation normalized fA-IA which accepts L with time complexity t.*

Proof. Let $L \in \mathscr{L}_t(f\text{A-IA})$ and let $\mathcal{M} = (S, \delta, \delta_{nd}, \delta_d, s_0, \#, A, F)$ be some fA-IA which accepts L with time complexity t. Denote by S_e resp. S_u the existential resp. universal states of $\mathcal{M}$. Now we are going to construct an alternation normalized fA-IA $\mathcal{M}' = (S', \delta', \delta'_{nd}, \delta'_d, s_0, \#, A, F')$ which simulates $\mathcal{M}$ without any loss of time. Suppose for a moment that $s_0 \in S_e$, i.e., the quiescent state of $\mathcal{M}$ is existential. The situation $s_0 \in S_u$ is handled afterwards.

Let $A_\# := A \cup \{\#\}$. For $s \in S$ define $X_s := \{(l, r, a, p) \mid l, r, p \in S \wedge a \in A_\# \wedge p \in \delta_{nd}(l, s, r, a)\}$. Thus, the fourth component p of a quadruple (l, r, a, p) from X_s contains a possible successor state of the communication cell of $\mathcal{M}$ under the assumption that it is in state s and its left resp. right neighbor is in state l resp. r whereby the input symbol a is consumed. By X we denote the union of all such X_s, i.e., $X = \bigcup_{s \in S} X_s$.

Let $S' := S \cup (S \times X \times \{e, u\} \times \{+, -, 0\})$ be the set of states and partition it into the existential states $S'_e := S \cup (S \times X \times \{e\} \times \{+, -, 0\})$ and the universal ones $S'_u := S' \setminus S'_e$. So a state of $\mathcal{M}'$ is existential iff it belongs to S or its third component (register) contains e. Clearly, by continually changing the content of the third register during the nondeterministic transitions the alternation normalization can be ensured. Therefore, if $\mu \in \{e, u\}$ we will later use the notation $\bar{\mu}$ for the complementary symbol, i.e. $\bar{\mu} = u$ if $\mu = e$ and $\bar{\mu} = e$ if $\mu = u$.

Before formally defining the local transition functions of $\mathcal{M}'$ we will explain what behavior they are intended to realize. The deterministic transitions of the communication cell of $\mathcal{M}'$ as well as the transitions of the deterministic cells are direct simulations of the corresponding ones of $\mathcal{M}$. So the only crux is the arrangement of the nondeterministic transitions of $\mathcal{M}'$ to which we will give our attention in the following.

Apart from the very first time step the communication cell of $\mathcal{M}'$ consists of four registers, i.e., it state belongs to $S \times X \times \{e, u\} \times \{+, -, 0\}$. The first register contains the actually simulated state s of the communication cell of $\mathcal{M}$. The second register enables $\mathcal{M}'$ to simulate an existential (resp. universal) transition of $\mathcal{M}$ even if the communication cell of $\mathcal{M}'$ is in a universal (resp. existential) state.

Therefore, it contains a nondeterministically determined quadruple (l, r, a, p) from X_s, which was chosen in the previous time step. During that time step $\mathcal{M}'$ has performed an existential (resp. universal) transition. Thus, if the content of the quadruple matches the actual situation (i.e., l resp. r are the states of the left resp. right neighbor of the communication cell and a is the fetched input symbol) by easily extracting p as successor state of s the existentiality (resp. universality) of the previous transition can be exploited. If otherwise the quadruple do not match the actual situation the communication cell switches into a rejecting (resp. accepting) state. Hence wrong guesses does neither become effective in the computation of $\mathcal{M}'$ if the previous transition was existential nor if it was universal.

The third register ensures the alternation normalization as explained above and the fourth register assists rejection $(-)$ resp. acception $(+)$ which is required in the case of a wrongly guessed quadruple. Correspondingly, the set of final states is defined by $F' := F \cup (F \times X \times \{e, u\} \times \{0\}) \cup (S \times X \times \{e, u\} \times \{+\})$.

Now let $s_1, s_2, s_3 \in S$, $x \in X$, $\mu \in \{e, u\}$, $\beta \in \{+, -, 0\}$, and $a \in A_{\#}$. In the sequel the local transition functions are defined partially only. Observe that in a not covered situation they are always considered to map on their second argument.

The local transition function δ' of the deterministic cells simply simulates δ using the content of the first register (of the communication cell) if necessary:

$$\delta'(s_1, s_2, s_3) := \delta'(s_1, s_2, (s_3, x, \mu, \beta)) := \delta'((s_1, x, \mu, \beta), s_2, s_3) := \delta(s_1, s_2, s_3).$$

Similarly, the deterministic local transition function δ'_d of the communication cell behaves:

$$\delta'_d(s_1, s_2, s_3, a) := \delta'_d(s_1, (s_2, x, \mu, 0), s_3, a) := \delta_d(s_1, s_2, s_3, a).$$

The nondeterministic local transition function δ'_{nd} is now designed as follows. During the first nondeterministic time step the communication cell is equipped with the four registers (observe that s_0 is existential in both $\mathcal{M}$ and $\mathcal{M}'$). So,

$$\delta'_{nd}(s_1, s_2, s_3, a) := \{(s', x', u, 0) \mid s' \in \delta_{nd}(s_1, s_2, s_3, a) \wedge x' \in X_{s'}\}.$$

For the next case if $s_2 \in S_\mu$ (and $\beta = 0$) an existential (resp. universal) transition of $\mathcal{M}$ can be simulated directly since the communication cell of $\mathcal{M}'$ is in an existential (resp. universal) state, too:

$$\delta'_{nd}(s_1, (s_2, x, \mu, 0), s_3, a) := \{(s', x', \bar{\mu}, 0) \mid s' \in \delta_{nd}(s_1, s_2, s_3, a) \wedge x' \in X_{s'}\}.$$

If $s_2 \notin S_\mu$, ($\beta = 0$), and there exists a $p \in S$ such that $x = (s_1, s_3, a, p)$, then p is a possible successor state of s_2 (in $\mathcal{M}$):

$$\delta'_{nd}(s_1, (s_2, x, \mu, 0), s_3, a) := \{(p, x', \bar{\mu}, 0) \mid x' \in X_p\}.$$

Since x has been determined during an existential (resp. universal) time step the extraction of p from x during the succeeding time step (which is universal (resp. existential)) actually corresponds to the simulation of an existential (resp. universal) transition.

Finally if $s_2 \notin S_\mu$, ($\beta = 0$), and $x \neq (s_1, s_3, a, p)$ for all $p \in S$, then x has not been guessed appropriately and in case of an existential (resp. universal) transition during the previous time step rejection (resp. acception) has to occur:

$$\delta'_{nd}(s_1, (s_2, x, \mu, 0), s_3, a) := \{(s_2, x, \bar{\mu}, \beta') \mid \beta' = - \text{ if } \mu = e \wedge \beta' = + \text{ if } \mu = u\}.$$

It remains to show how $\mathcal{M}'$ can be constructed if $s_0 \in S_u$, i.e., the communication cell of $\mathcal{M}'$ has to simulate a universal transition of $\mathcal{M}$ during the first (existential) time step. Obviously, using sufficiently many tracks on which every possible successor state is simulated in parallel solves this problem. Clearly, then a configuration is accepting iff the configurations on all tracks are accepting. □

4 Closure under Boolean Operations

Lemma 8 *Let $f : \mathbb{N} \to \mathbb{N}_0$, $f(n) \leq n$, and $t : \mathbb{N} \to \mathbb{N}$, $t(n) > n$, be two mappings then $\mathscr{L}_t(f\text{A-IA})$ is closed under union and intersection.*

Proof. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two t-time fA-IAs. By Lemma 7 we may assume that $\mathcal{M}_1$ and $\mathcal{M}_2$ are alternation normalized. Due to the normalized behavior we can construct a t-time fA-IA $\mathcal{M}'$ that simulates $\mathcal{M}_1$ and $\mathcal{M}_2$ on different tracks in parallel. It is easy to see that the computation tree of $\mathcal{M}'$ contains an accepting subtree if $\mathcal{M}_1$ or $\mathcal{M}_2$ accept simply by considering the corresponding track only. The closure under union follows.

In order to find an accepting subtree for the intersection we have to use the successor that contains guesses of $\mathcal{M}_1$ and $\mathcal{M}_2$ which lead to acceptance in existential steps, respectively. Clearly in universal steps all successor configurations of $\mathcal{M}'$ contain all successor configurations of $\mathcal{M}_1$ and $\mathcal{M}_2$ and vice versa. The closure under intersection follows. □

The comparison between nondeterministic and alternating IAs in the next section is done at the hand of closure under complement. It is easy to prove the closure of A-IAs but hard to disprove it for G-IAs. Here is the easy part:

Lemma 9 *Let $f : \mathbb{N} \to \mathbb{N}_0$, $f(n) \leq n$, be a mapping then $\mathscr{L}_{rt}(f\text{A-IA})$ is closed under complement.*

Proof. The meaning of an existential transition is that there must exist one successor configuration which leads to acceptance. In order to accept the complement this can be replaced by the meaning that all successor configurations do not lead to acceptance. On the other hand, the meaning of an universal step that all successors must lead to acceptance can be replaced by the meaning that one successor does not lead to acceptance. The negation in the new meaning is simply realized as follows: if the communication cell has consumed the whole input it now accepts if it would have rejected before and vice versa. Thus, final and non-final configurations are exchanged. □

5 Alternating Hierarchy

5.1 Comparison with Nondeterministic Iterative Arrays

In the following we incorporate some results of a previous work [3] concerning IAs with nondeterministic communication cell.

In order to define an important language let $f : \mathbb{N} \to \mathbb{N}_0$ be an increasing mapping such that $f \in o(\log)$. We define another mapping $h : \mathbb{N} \to \mathbb{N}$ by $h(n) := 2^{f(n)}$. It is increasing since f is. Moreover, since $f \in o(\log)$ for all $k \in \mathbb{Q}_+$ it holds $\lim_{n\to\infty} \frac{h(n)}{n^k} = \lim_{n\to\infty} \frac{2^{f(n)}}{2^{\log(n)\cdot k}} = 0$ and therefore $h \in o(n^k)$. Especially for $k = \frac{1}{2}$ it follows that the mapping $m(n) := \max\{n' \in \mathbb{N}_0 \mid (h(n)+1)\cdot(n'+1) \leq n\}$ is unbounded, and for large n we obtain $m(n) > h(n)$. The following language depends on f only.

$$\begin{aligned} L_f := \{\$^r w_1 \$ w_2 \$ \cdots \$ w_j ¢ y ¢ \mid\ & \exists\, n \in \mathbb{N} : j = h(n) \land w_i \in \{0,1\}^{m(n)}, 1 \leq i \leq j, \\ & \land r = n - (h(n)+1)\cdot(m(n)+1) \\ & \land \exists\, 1 \leq i' \leq j : w_{i'} = y^R\} \end{aligned}$$

The words of length n of L_f consist of $2^{f(n)}$ subwords w_i and one subword y which is the reversal of one of the w_i. The number of subwords is fixed for a given n. The lengths of the subwords is as large as possible.

The next theorem follows immediately from a theorem shown in [3] in order to prove a nondeterministic hierarchy.

Theorem 10 *Let $f : \mathbb{N} \to \mathbb{N}_0$ and $g : \mathbb{N} \to \mathbb{N}_0$ be two increasing mappings such that $f \in o(\log)$ and $g \in o(f)$ then $L_f \in \mathscr{L}_{rt}(f\text{G-IA})$ and $L_f \notin \mathscr{L}_{rt}(g\text{G-IA})$.*

Since for $g \in o(f)$ the language L_f is not a real-time gG-IA language but, on the other hand, it can be accepted in real-time by a fG-IA, and the number of guesses can be reduced by a constant factor [3] one obtains the following corollary. Moreover, it holds for A-IAs too, since our approximation of the numbers of equivalence classes are identical regardless of whether or not nondeterministic or alternating IAs are in question:

Corollary 11 *Let $f : \mathbb{N} \to \mathbb{N}_0$ and $g : \mathbb{N} \to \mathbb{N}_0$ be two increasing mappings such that $f \in o(\log)$ then $L_f \in \mathscr{L}_{rt}(g\text{G-IA}) \Longrightarrow g \in \Omega(f)$ and $L_f \in \mathscr{L}_{rt}(g\text{A-IA}) \Longrightarrow g \in \Omega(f)$.*

The next theorem is the main result of the present section. It states that under some preconditions the real-time alternating IAs are strictly more powerful than the real-time nondeterministic IAs. For the proof we need a closure property concerning marked iteration.

Definition 12 *Let L be a language over an alphabet A and $\bullet \notin A$ be a distinguished marking symbol. The language $(L\bullet)^+$ is the* marked iteration *of L.*

Here we have to require f to be in some sense computable. This can be done in terms of deterministic real-time IA languages. It should be mentioned that the family $\mathscr{L}_{rt}(\text{IA})$ is very rich.

Theorem 13 *Let $f : \mathbb{N} \to \mathbb{N}_0$ be an increasing, unbounded mapping such that $f \in o(\log)$ and $\{a^{f(m)}b^{m-f(m)} \mid m \in \mathbb{N}\} \in \mathscr{L}_{rt}(\text{IA})$ then $\mathscr{L}_{rt}(f\text{G-IA}) \subset \mathscr{L}_{rt}(f\text{A-IA})$.*

Proof. Since a fG-IA is just a fA-IA with only existential states we have the inclusion $\mathscr{L}_{rt}(f\text{G-IA}) \subseteq \mathscr{L}_{rt}(f\text{A-IA})$.

It remains to show $\mathscr{L}_{rt}(f\text{G-IA}) \neq \mathscr{L}_{rt}(f\text{A-IA})$. The idea is to prove the inequality at the hand of different closure properties.

By Lemma 9 the family $\mathscr{L}_{rt}(f\text{A-IA})$ is closed under complement. We are going to show that $\mathscr{L}_{rt}(f\text{G-IA})$ is not closed under complement.

In order to do so suppose $\mathscr{L}_{rt}(f\text{G-IA})$ is not closed under marked iteration which will be shown by Lemma 14.

Let $L \in \mathscr{L}_{rt}(f\text{G-IA})$ be a language over an alphabet A. If $\overline{L}$ does not belong to $\mathscr{L}_{rt}(f\text{G-IA})$ we are done.

Assume now $\mathscr{L}_{rt}(f\text{G-IA})$ is closed under complement and let $\mathcal{M}$ be a fG-IA that accepts $\overline{L}$ in real-time. Now we construct a real-time fG-IA $\mathcal{M}'$ that accepts $\overline{(L\bullet)^+}$.

In [2, 3] the real-time simulation of stacks by deterministic IAs has been shown. Thereby the communication cell contains the symbol at the top of the stack. We will use the ability of IAs to simulate such data structures at the construction.

One deterministic regular task of $\mathcal{M}'$ is to check whether the input is of the form $x_1\bullet x_2\bullet \cdots \bullet x_k\bullet$ where $x_i \in A^+$, $1 \leq i \leq k$. All words that do not fit this form are accepted.

A word $x_1 \bullet x_2 \bullet \cdots \bullet x_k \bullet$ belongs to $\overline{(L\bullet)^+}$ iff at least one x_i, $1 \leq i \leq k$, belongs to $\overline{L}$. In order to accept such words $\mathcal{M}'$ simulates $\mathcal{M}$ on x_1 directly and additionally during its nondeterministic steps the $f(|x_i|)$ nondeterministic steps of $\mathcal{M}$ on input x_i for $i > 1$. Since f is increasing $\mathcal{M}'$ has at least as many nondeterministic steps as $\mathcal{M}$. The guessing is done by choosing nondeterministically one of the (finite) local transition functions at each time step and pushing it onto a stack.

When the direct simulation of $\mathcal{M}$ on x_1 succeeds the job of $\mathcal{M}'$ is done. Otherwise it starts the following task every time a $\bullet$ appears in the input.

A signal is sent through the stack which copies the content of the stack to a second stack cell by cell. Additionally, $\mathcal{M}'$ simulates $\mathcal{M}$ on the next subword x_i. In order to simulate a nondeterministic step one mapping is popped from the second stack (leaving the first stack unchanged) and is applied to the local configuration. So the communication cell can simulate a nondeterministic step of $\mathcal{M}$ deterministically by applying a previously nondeterministically determined deterministic local transition. Again, if one of the simulations succeeds $\mathcal{M}'$ accepts otherwise it rejects.

Up to now we kept quiet about a crucial point. The number $f(|x_i|)$ of simulated nondeterministic transitions may be incorrect. Therefore, the decision of $\mathcal{M}'$ depends on corresponding verifications additionally: In order to perform this task an acceptor for the language $L' = \{a^{f(m)}b^{m-f(m)} \mid m \in \mathbb{N}\}$ is simulated in parallel whenever a $\bullet$ appears in the input. Thereby an input symbol a is assumed for each nondeterministic step (up to the guessed time $f(|x_i|)$) and an input symbol b for each deterministic step (up to the end of input x_i). So the number x resp. y of simulated nondeterministic resp. deterministic transitions corresponds to a word $a^x b^y$ belonging to L' iff there exists an $m \in \mathbb{N}$ such that $x = f(m)$ and $y = m - f(m)$. Thus, iff $|x_i| = x + y = f(m) + m - f(m) = m$. Altogether $\mathcal{M}'$ accepts $\overline{(L\bullet)^+}$ in real-time.

Since we have assumed that $\mathscr{L}_{rt}(f\text{G-IA})$ is closed under complement it follows $(L\bullet)^+ \in \mathscr{L}_{rt}(f\text{G-IA})$. But we have supposed $\mathscr{L}_{rt}(f\text{G-IA})$ is not closed under marked iteration. From the contradiction it follows that $\mathscr{L}_{rt}(f\text{G-IA})$ is not closed under complement if $\mathscr{L}_{rt}(f\text{G-IA})$ is really not closed under marked iteration. This will be proved in the next theorem. □

The next lemma has already been used to prove a previous one.

Lemma 14 *Let $f : \mathbb{N} \to \mathbb{N}_0$ be an increasing, unbounded mapping such that $f \in o(\log)$ and $\{a^{f(m)}b^{m-f(m)} \mid m \in \mathbb{N}\} \in \mathscr{L}_{rt}(\text{IA})$ then $\mathscr{L}_{rt}(f\text{G-IA})$ is not closed under marked iteration.*

Proof. By Theorem 10 L_f belongs to $\mathscr{L}_{rt}(f\text{G-IA})$. Now we are going to show that the marked iteration $(L_f\bullet)^+$ of L_f does not belong to $\mathscr{L}_{rt}(f\text{G-IA})$ from which the lemma follows.

Assume in contrast there exists a fG-IA $\mathcal{M} = (S, \ldots)$ which accepts $(L_f\bullet)^+$ in real-time. We consider words $x_1 \bullet x_2 \bullet \cdots \bullet x_k \bullet \in (L_f\bullet)^+$ for a $k \in \mathbb{N}$. Let x_k be an arbitrary word in L_f and n_k be its length: $n_k := |x_k|$. Since m

is an unbounded mapping we can find a smallest $n_i \in \mathbb{N}$ such that $m(n_i) \geq |x_i \bullet x_{i+1} \bullet \cdots \bullet x_k \bullet|$ respectively, for $1 \leq i \leq k-1$. Obviously, there exist words of length n_i in L_f. Let x_i be one of them respectively. For the lengths l_i of the subwords $x_i \bullet \cdots x_k \bullet$ we obtain $l_k = n_k + 1$ and for $1 \leq i \leq k-1$: $l_i = n_i + 1 + l_{i+1}$.

In what follows let k_j be appropriated constants. Since $h(n_i) \leq m(n_i)$ and $r_i \leq m(n_i)$ it holds $n_i = \big(m(n_i)+1\big)\big(h(n_i)+1\big) + r_i \leq \big(m(n_i)+1\big)^2 + m(n_i) \leq k_8 \cdot m(n_i)^2$. For l_i we obtain:

$$\begin{aligned} l_i &= r_i + \big(h(n_i)+1\big)\big(m(n_i)+1\big) + 1 + l_{i+1} \\ &\leq k_5 \cdot l_{i+1} + \big(h(n_i)+1\big)\big(k_5 \cdot l_{i+1} + 1\big) + 1 + l_{i+1} \text{ since } r_i \leq m(n_i) \leq k_5 \cdot l_{i+1} \\ &\leq k_6 \cdot h(n_i) \cdot l_{i+1} \\ &\leq k_6 \cdot h(k_7 \cdot l_{i+1}^2) \cdot l_{i+1} \\ &\leq k_i' \cdot l_{i+1}^{\epsilon} l_{i+1} \text{ since } h(n) \in o(n^{\epsilon/2}) \text{ for all } \epsilon \in \mathbb{Q}_+ \end{aligned}$$

It follows $l_1 \leq k_1' \cdot l_2^{1+\epsilon} \leq \cdots \leq k_1' \cdot \ldots \cdot k_{k-1}' \cdot l_k^{(1+\epsilon)^{k-1}}$.

If we choose $\epsilon \in \mathbb{Q}_+$ such that $(1+\epsilon)^{k-1} < 2$ then for large n we obtain that $l_1 \leq \frac{1}{2} \cdot l_k^2 = \frac{1}{2} \cdot (n_k+1)^2 \leq n_k^2$.

Thus for processing $x_1 \bullet \cdots x_k \bullet$ $\mathcal{M}$ performs at most $f(n_k^2)$ nondeterministic transitions. Since $f \in o(\log)$ there exists $k_1 \in \mathbb{N}$ such that $k_1 \cdot f(n_k) \geq f(n_k^2)$ for large n_k. Therefore, for large n at most $k_1 \cdot f(n_k)$ nondeterministic steps are performed by $\mathcal{M}$. (note that k_1 does not depend on k).

Now we consider the equivalence classes that appear if we cut $x_1 \bullet \cdots x_k \bullet$ after the first symbol ¢ in x_i respectively. Since $x_2 \bullet \cdots x_k \bullet$ is at most as long as the y_1 in x_1 we have $N(|x_1 \bullet \cdots x_k \bullet|, 2|y_1|+1, L_f \bullet A^{|y_1|})$ different equivalence classes for the cut in x_1. By Lemma 6 this number equals $N(|x_1|, |y_1|, L_f)$. By Corollary 11 there exists a constant such that at least $k_2 \cdot f(n_1)$ guesses are necessary in order to accept languages with such a number of equivalence classes. Define $q_m := \max\{|\delta_{nd}(s_1, s_2, s_3, a)| \mid s_1, s_2, s_3 \in S, a \in A\}$. Thus, the computation of $\mathcal{M}$ on input $x_1 \bullet$ contains at least $q_m^{k_2 \cdot f(n_1)}$ different paths.

Now we consider all computation paths of $\mathcal{M}$. For all $x_1 \in L_f$ there exists a class of paths that are accepting for words of the form $x_1 \bullet \cdots$ Since for computations on $x_1 \bullet$ there are at least $q_m^{k_2 \cdot f(n_1)}$ different paths we have now at least $q_m^{k_2 \cdot f(n_1)}$ disjoint classes.

If we cut $x_1 \bullet \cdots x_k \bullet$ after the first symbol ¢ in x_2, again, it results in $N(|x_1 \bullet x_2|, |y_2|, L_f)$ equivalence classes for which $k_2 \cdot f(n_2)$ different computations paths are necessary. These paths are all in the same class for x_1. Therefore, every class contains at least $q_m^{k_2 \cdot f(n_2)}$ paths. Since at least $q_m^{k_2 \cdot f(n_1)}$ classes are disjoint there are at least $q_m^{k_2 \cdot f(n_1)} \cdot q_m^{k_2 \cdot f(n_2)}$ different paths.

Proceeding inductively we conclude that there are at least $q_m^{k_2 \cdot f(n_1)} \cdot \ldots \cdot q_m^{k_2 \cdot f(n_k)} \geq \big(q_m^{k_2 \cdot f(n_k)}\big)^k$ different paths. To realize the paths $\mathcal{M}$ at least needs to perform $k \cdot k_2 \cdot f(n_k)$ nondeterministic steps (here we need $q_m > 1$ what follows since f is unbounded). For a k such that $k \cdot k_2 > k_1$ we get a contradiction because $\mathcal{M}$ performs at most $k_1 \cdot f(n_k)$ nondeterministic transitions. □

Corollary 15 *Let $f : \mathbb{N} \to \mathbb{N}_0$ be an increasing mapping such that $f \in o(\log)$ then $\mathscr{L}_{rt}(f\text{G-IA})$ is not closed under complement.*

5.2 The Hierarchy

In [3] the following nondeterministic hierarchies have been shown: Let $f : \mathbb{N} \to \mathbb{N}_0$, $f \in o(\log)$, and $g : \mathbb{N} \to \mathbb{N}_0$, $g \in o(f)$, be two increasing mappings such that $\forall\, m, n, \in \mathbb{N} : f(m) = f(n) \Longrightarrow g(m) = g(n)$. If $L = \{a^{g(m)}b^{f(m)-g(m)} \mid m \in \mathbb{N}\}$ belongs to the family $\mathscr{L}_{lt}(\text{IA})$ then $\mathscr{L}_{rt}(g\text{G-IA}) \subset \mathscr{L}_{rt}(f\text{G-IA})$.

By the results of the previous subsection we obtain an alternating hierarchy, too.

Theorem 16 *Let $f : \mathbb{N} \to \mathbb{N}_0$, $f \in o(\log)$, and $g : \mathbb{N} \to \mathbb{N}_0$, $g \in o(f)$ be two increasing mappings such that $\forall\, m, n, \in \mathbb{N} : f(m) = f(n) \Longrightarrow g(m) = g(n)$. If $\{a^{g(m)}b^{f(m)-g(m)} \mid m \in \mathbb{N}\} \in \mathscr{L}_{lt}(\text{IA})$ and $\{a^{f(m)}b^{m-f(m)} \mid m \in \mathbb{N}\} \in \mathscr{L}_{rt}(\text{IA})$ then $\mathscr{L}_{rt}(g\text{A-IA}) \subset \mathscr{L}_{rt}(f\text{A-IA})$.*

Proof. Due to the assumption $L := \{a^{g(m)}b^{f(m)-g(m)} \mid m \in \mathbb{N}\} \in \mathscr{L}_{lt}(\text{IA})$ a real-time fG-IA can limit its nondeterministic transitions up to the guessed time step $g(n)$ and verify its guess. For a deterministic real-time IA language this technique has been used in the proof of Lemma 14. It is known that deterministic linear-time IAs can be sped up to $2 \cdot n$ time [9]. Since $f \in o(\log)$ we can assume $f \leq \frac{n}{2}$ and, hence, during n time steps a $(2 \cdot n)$-time IA for L can be simulated.

By this constructibility property and for structural reasons we obtain $\mathscr{L}_{rt}(g\text{A-IA}) \subseteq \mathscr{L}_{rt}(f\text{A-IA})$. Since g is of order $o(f)$ but by Corollary 11 it has to be of order $\Omega(f)$ in order to accept L_f in real-time we conclude $L_f \notin \mathscr{L}_{rt}(g\text{A-IA})$.

On the other hand by Theorem 10 L_f belongs to $\mathscr{L}_{rt}(f\text{G-IA})$. We obtain $\mathscr{L}_{rt}(f\text{G-IA}) \not\subseteq \mathscr{L}_{rt}(g\text{A-IA})$.

By Theorem 13 it holds $\mathscr{L}_{rt}(f\text{G-IA}) \subset \mathscr{L}_{rt}(f\text{A-IA})$.

It follows $\mathscr{L}_{rt}(g\text{A-IA}) \subset \mathscr{L}_{rt}(f\text{A-IA})$. □

At a first glance the preconditions of the hierarchy seem to be rather complicated but the following natural functions meet them. Let $i > 1$ be a constant then $f(n) := \log^i(n)$ and $g(n) := \log^{i+1}(n)$ ($\log^i$ denotes the i-fold composition of log).

References

[1] Beyer, W. T. *Recognition of topological invariants by iterative arrays.* Technical Report TR-66, MIT, Cambridge, Proj. MAC, 1969.

[2] Buchholz, Th. and Kutrib, M. *Some relations between massively parallel arrays.* Parallel Computing 23 (1997), 1643–1662.

[3] Buchholz, Th., Klein, A., and Kutrib, M. *Iterative arrays with limited nondeterministic communication cell*. IFIG Report 9901, Institut für Informatik, Universität Gießen, Gießen, 1999.

[4] Chang, J. H., Ibarra, O. H., and Palis, M. A. *Parallel parsing on a one-way array of finite-state machines*. IEEE Transactions on Computers C-36 (1987), 64–75.

[5] Cole, S. N. *Real-time computation by n-dimensional iterative arrays of finite-state machines*. IEEE Transactions on Computers C-18 (1969), 349–365.

[6] Čulik II, K. and Yu, S. *Iterative tree automata*. Theoretical Computer Science 32 (1984), 227–247.

[7] Fischer, P. C. *Generation of primes by a one-dimensional real-time iterative array*. Journal of the ACM 12 (1965), 388–394.

[8] Ibarra, O. H. and Jiang, T. *On one-way cellular arrays*. SIAM Journal on Computing 16 (1987), 1135–1154.

[9] Ibarra, O. H. and Palis, M. A. *Some results concerning linear iterative (systolic) arrays*. Journal of Parallel and Distributed Computing 2 (1985), 182–218.

[10] Ibarra, O. H. and Palis, M. A. *Two-dimensional iterative arrays: Characterizations and applications*. Theoretical Computer Science 57 (1988), 47–86.

[11] Seiferas, J. I. *Iterative arrays with direct central control*. Acta Informatica 8 (1977), 177–192.

[12] Seiferas, J. I. *Linear-time computation by nondeterministic multidimensional iterative arrays*. SIAM Journal on Computing 6 (1977), 487–504.

[13] Smith III, A. R. *Real-time language recognition by one-dimensional cellular automata*. Journal of Computer and System Sciences 6 (1972), 233–253.

[14] Terrier, V. *On real time one-way cellular array*. Theoretical Computer Science 141 (1995), 331–335.