Mathematical Institute
Slovak Academy of Sciences, Bratislava

Preprint 13/1997

Ivan Korec

Real-time generation of primes by a one-dimensional cellular automaton with 9 states

October 6, 1997

PREPRINT SERIES

REAL-TIME GENERATION OF PRIMES BY A ONE-DIMENSIONAL CELLULAR AUTOMATON WITH 9 STATES

Ivan KOREC

Abstract. A 9-state one-dimensional cellular automaton is constructed which generates the primes in the following sense: The content of the 0-th cell at time t is equal to '1' if t is a prime, and is equal to '0' otherwise. The neighbourhood type of this CA is $(-1, 0, 1)$, i.e. the most usual one. At time $t = 0$ only the 0-th cell is in a non-quiescent state (here '0' is not the quiescent state). Further, a one-dimensional CA is constructed with the radius 10 but with two states only which also generates the primes. (At time $t = 0$ only the 1-st cell is in a non-quiescent state.) Also a generalized Pascal triangle with 75 distinct elements is constructed which generates the odd primes in a similar sense. Hence the primes can be real-time generated also by a 75-state one-dimensional CA with the neighborhood type $(-1, 1)$.

1. Introduction

We shall deal with one-dimensional cellular automata (abbreviation: 1D CA) in the classical sense; necessary definitions can be found in the next section. The main result of the present contribution is:

Claim 1. *There is a 9-state one-dimensional cellular automaton (with radius 1) which generates the primes in real time.*

(Three *claims* of the Introduction are repeated more formally as initial *theorems* in the Sections 3–5, Claim 4 as a *corollary* of Theorem 3.) By P. C. Fischer [Fi], generation of primes in real time is understood in the following sense: The content of the 0-th cell at time t is equal to '1' if t is a prime, and it is equal to '0' otherwise. The neighbourhood type of this CA is $(-1, 0, 1)$, i.e. the most usual one, and also the initial configuration is the simplest possible.

A similar result was obtained in [Fi], where, however, $37^3 = 50653$ (after reduction ≈ 30000) states are used. To be quite precise, in [Fi] one-dimensional iterative arrays are considered, but the difference in terminology is not substantial. In the solution presented here cells with negative numbers remain forever in the quiescent state, hence we can obtain a one-side infinite one-dimensional iterative array by a purely technical modification. A recent author's paper [K3] contains the analogy of Claim 1 for 11 states, and similar weaker analogies of some results below.

From Claim 1 we shall derive:

This research was partially supported by GA ČR No 201/95/0976 "HypercompleX" and by Grant 2/4034/97 of VEGA (SAV)

Claim 2. *There is a 2-state one-dimensional cellular automaton with radius* 10 *which generates the primes in real time.*

Radius 10 means that the neighbourhood type is $(-10, -9, \ldots, 10)$ (or a subset of that). Here two states '0', '1' will be used, and '0' is the quiescent state. Again, the content of the 0-th cell at time t is '1' if t is prime and '0' otherwise, and at the time $t = 0$ only a unique cell is in a non-quiescent state (i.e. '1' in this case). However, it cannot be the 0-th cell because its content must be '0'; the 1-st cell can be used for this purpose.

Further, we obtain a similar result for generalized Pascal triangles, abbreviation: GPT. They correspond to some computations of 1D CA with the neighborhood type $(-1, 1)$. (GPT are constructed similarly as the classical Pascal's triangle, but arbitrary binary operation is used instead of addition.)

Claim 3. *There is a generalized Pascal triangle with* 75 *distinct elements which generates the odd primes in real time.*

We have to consider here only odd primes because a column of a GPT has common elements either with its odd rows or with its even rows. (Various modifications with two distinguished columns are possible, but such complication seems to be unnecessary here.) We can construct another (rather trivial) GPT which recognizes $\{2\}$, and obtain:

Claim 4. *There is a* 75*-state one-dimensional cellular automaton with neighbourhood type* $(-1, 1)$ *which generates the primes in real time.*

All four claims above are proved by explicit construction of 1D CA, resp. GPT. At the end the lower bound 4 is proved for the situation of Claim 1, and some open problems are formulated.

2. Notation and technicalities

The set of nonnegative integers will be denoted by $\mathbb{N}$ and the set of all integers by $\mathbb{Z}$. The set of all nonempty words in an alphabet Σ will be denoted by Σ^+. The i-th symbol of a word w will be denoted by $w(i)$; the counting starts with 0. Length of a word w will be denoted by $|w|$. We shall use with single quotation marks ' ' around letters and words. It is suitable because to obtain more transparent figures we use also '.' and '/' as letters (or states).

Definition 1. *A one-dimensional CA is an ordered quadruple* $C = (S, N, f, q)$, *where*

S *is a finite set of states;*
$N = (a_1, \ldots, a_n)$, *the neighbourhood vector, is a finite sequence of pairwise distinct integers;*
$f : S^n \to S$ *is a local transition function;*
$q \in S$, *the quiescent state, satisfies* $f(q, \ldots, q) = q$.

A computation of one-dimensional CA C *is a function* $F : \mathbb{Z} \times \mathbb{N} \to S$ *such that*

$$F(z, t+1) = f(F(z + a_1, t), \ldots, F(z + a_n, t)) \tag{2.1}$$

for all $z \in \mathbb{Z}$, $t \in \mathbb{N}$.

The restrictions of F *to sets* $\mathbb{Z} \times \{t\}$, $t \in \mathbb{N}$ *will be called configurations, for* $t = 0$ *the initial configuration. A configuration (at time* t*) is called finite if* $F(z, t) = q$ *for all but finitely* $z \in \mathbb{Z}$.

We shall say that C is a one-dimensional CA of radius r if $|a_i| \leq r$ for all $i = 1, \ldots n$.

Notice that usually configurations of CA are defined at first, and then a computation of CA is defined as a sequence of configurations which satisfies some conditions ((2.1) in essential).

If a computation F of a 1D CA is given we can try to reconstruct the original 1D CA. It never can be done uniquely because, e.g., we can add unnecessary states and enlarge the neighborhood type. Even if we know S, N and q it may happen that (2.1) determines the local rule f only partially. However, if we are interested only in the computation F (and not in other computations of the CA) we can complete f arbitrarily. This will be our situation below. (Here we do not touch *algorithmic* problems like: how large piece of F must be used.)

Now we shall introduce generalized Pascal triangles (GPT). (The notion of Pascal's triangle was generalized in many ways; for example, many arithmetical ones are considered in [Bo]. However, we follow only [K1] and [K2], where a notion strongly related to 1D CA is given.) GPT are mappings of some subsets of $\mathbb{N} \times \mathbb{N}$ into finite sets. They are associated to some finite algebras and nonempty words analogously as the classical Pascal triangle can be associated to the (infinite) algebra $\langle \mathbb{N}; +, 0 \rangle$ and "the word" 1 (of length 1). More formally:

Definition 2. (i) *To every algebra $\mathcal{A} = (\mathbf{A}; *, \mathrm{o})$ such that $\mathrm{o} * \mathrm{o} = \mathrm{o}$ and to every word $w \in \mathbf{A}^+$ the partial function $G = \mathrm{GPT}(\mathcal{A}, w) : \mathbb{N} \times \mathbb{N} \to \mathbf{A}$ will be associated by the formula:*

$$G(x,y) = \begin{cases} \text{undefined} & \text{if} \quad x + y < |w| - 1, \\ w(x) & \text{if} \quad x + y = |w| - 1, \\ \mathrm{o} * G(0, y-1) & \text{if} \quad x = 0,\ y \geq |w|, \\ G(x-1, 0) * \mathrm{o} & \text{if} \quad y = 0,\ x \geq |w|, \\ G(x-1, y) * G(x, y-1) & \text{if} \quad x + y \geq |w|,\ x > 0,\ y > 0. \end{cases}$$

(ii) *A partial function G will be called GPT if it can be represented in the form $\mathrm{GPT}(\mathcal{A}, w)$ for a finite algebra $\mathcal{A}$ and $w \in \mathbf{A}^+$.*

(iii) *The i-th row of GPT G consists of all $G(x,y)$ with $x + y = i$ (it is a word). The i-th column of a GPT G consists of all $G(x,y)$ with $x - y = i$ (it is an infinite sequence). In both cases we consider the values in the order given by x.*

We shall explain the relationship between GPT and computations of 1D CA in its the simplest case only, for the neighborhood type $(-1, 1)$ and computations starting from configurations with non-quiescent state only in the 0-th cell. (We shall need exactly this case below.)

Computations of 1D CA can be considered as functions defined on $(1 + 1)$-dimensional discrete space-time $\mathbb{Z} \times \mathbb{N}$. For displaying computations of 1D CA we use the coordinate system with the space coordinate z horizontal, and the time coordinate t vertical, oriented downwards. For displaying GPT, the system of coordinates is chosen so that the whole GPT lies in the first (i.e., "positive") quadrant of the plane. The axis x is oriented right-downward, and the axis y is oriented left-downward. (Then point (iii) of Definition 2 is natural.) The length unit are chosen so that $x + y$ corresponds to the time coordinate t and $x - y$ to the space coordinate z mentioned above.

Then the domain of a GPT will be considered as a "light cone", which contains the whole interesting part of the CA computation; all positions outside of it are in the quiescent state q (resp. the constant o).

3. A 9-state 1D cellular automaton

Theorem 1. *There is a 9-state 1D CA (with radius 1) which has a computation F with the following properties:*

(1) $F(0,0) =$ '0' *and* $F(x,0) =$ '.' *for all* $x \neq 0$.
(2) *For all* $x < 0$ *and* $t \geq 0$ *it holds* $F(x,t) =$ '.' .
(3) *For all* $t \geq 0$ *we have* $F(0,t) =$ '1' *if* t *is a prime,* $F(0,t) =$ '0' *otherwise.*

Proof. The computation F is presented in Figure 1. (The necessary values of f which cannot be determined by the displayed piece of F are given in the lower part of the figure. A record $n|xyz{:}w$ in this part means that $f(x,y,z) = w$ is used in the n-th row for the first time. As we can see, the rows of F up to the 494-th one would be necessary to determine f sufficiently. Even from the whole F the local rule f is determined only partially; there are many local rules which give F.)

The alphabet of states of the constructed 1D CA will be

$$\{\text{'.'}, \text{'/'}, \text{'0'}, \text{'1'}, \text{'L'}, \text{'R'}, \text{'r'}, \text{'V'}, \text{'v'}\},$$

where '.' is used as the quiescent state. (Of course, the choice of symbols is not substantial. The alphabet is chosen to obtain the figures as transparent as possible. For example, the quiescent state '.' is chosen as similar to empty space; the other symbols will be explained later.)

The upper part of F contains some irregularities, and ought to be considered separately. It concerns mainly the rows up to the 27-th; they ought to be verified ad hoc. Two rightest 'R' in almost every row also arise here. However, starting from the 14-th row they do not influence the recognizing of primes; therefore they are not described below.

The computation F contains several kinds of signals which spread with distinct speeds. If "speed of light" (i. e. the maximal possible one) is ± 1 (where the sign + determines direction to the right) then the main signals and their speeds are:

(1) speed -1: '/' and 'L';
(2) speed 0: 'V' (and '1', only in the 0-th column);
(3) speed $\frac{1}{3}$: '1/'–'11'–'00L';
(4) speed 1: 'R'.

The signals behave in various ways at their crossings; they may disappear, form another signals, etc.

The most characteristic and most frequent signals are '/'; generation of them is the most substantial role of the other signals. $F(z,t) =$ '/' means that $z+t$ is composed. When '/' meets 'V' or 'R' it changes them temporarily to 'v' or 'r', respectively. We can consider 'v' as 'V' with '/', and similarly for 'r', 'R'. Analogously we can consider every '0' in the 0-th column as '1' with '/'; however, now '/' does not continue to (-1)-st column but it disappears. (Since the asymptotical density of composed integers is 1 we can see very many '0' in the 0-th column.) The signal '/' is not repeated when it meets 'L'; we can imagine that 'L' already contains '/'. (Here 'L' corresponds to a smaller divisor of the integer $z+t$.)

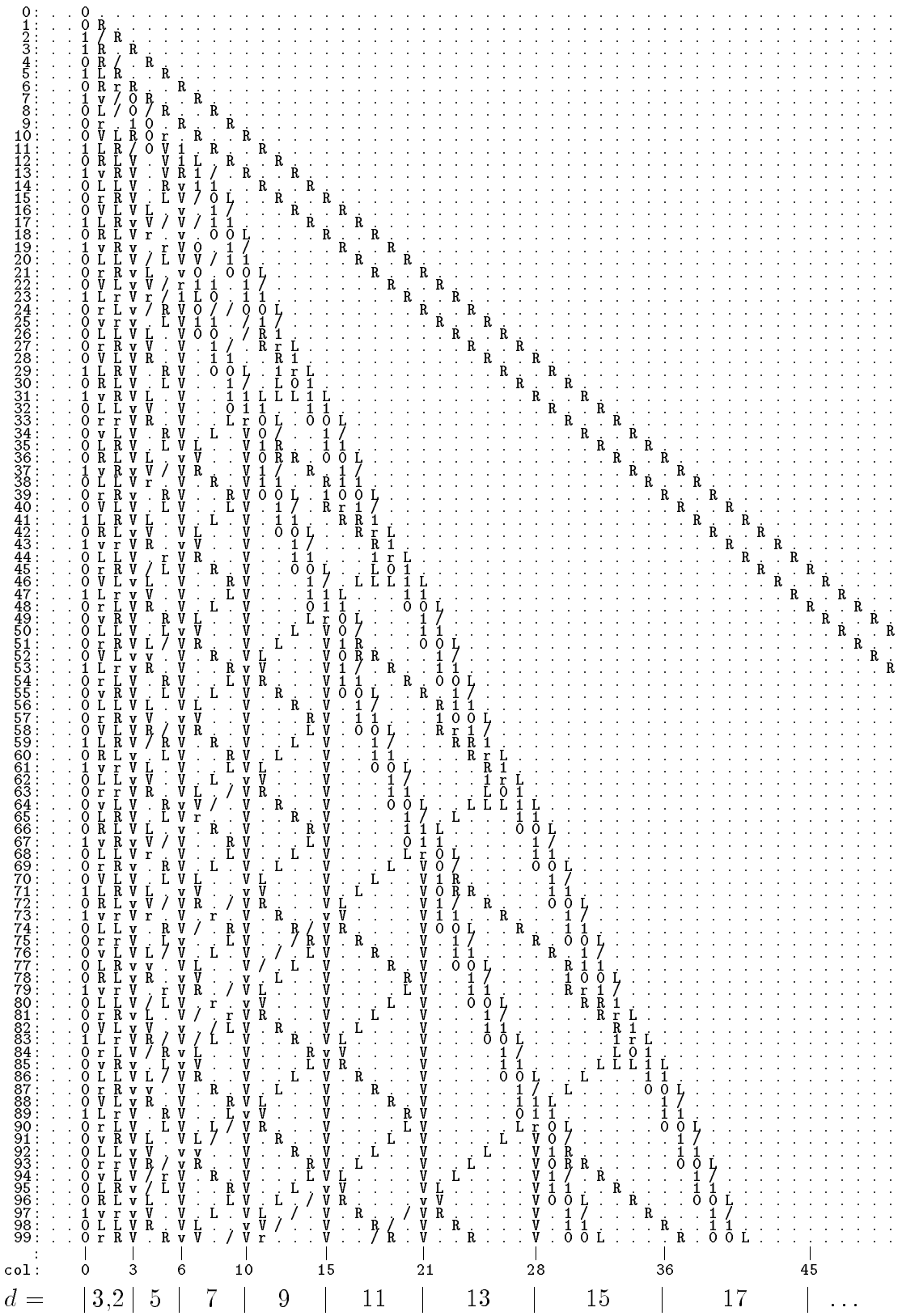

Figure 1. Real-time generation of primes by a 9-state 1D CA.

To every odd integer $d > 3$ a segment between two consecutive 'V' is associated (the segments are labelled during the computation). In this segment divisibility by d is recognized with help of the signals 'L' and 'R'. So the signal "composed" is formed which runs to the left as '/'. The divisors $d = 2$ and $d = 3$ are considered separately in the second and the first cell, respectively. When we remove the '/' signals from the right the area consisting of the columns 0–4 would be periodical with period 6 (and the integers $6k \pm 1$ would be declared as "primes").

The symbols 'V' which separate the segments for divisors $d > 3$ are displaced using signals '1/'–'11'–'00L', 'L', and 'R'. There are two instances of the first signal. (In a large scale each of them covers an arc of a quadratic parabola, and not a half-line. This is because the signals are shifted during crossings.) The other two signals run between these instances, 'R' to the right, and 'L' to the left. By crossing of 'R' with '1/'–'11'–'00L' the signal 'L' arises (and 'R' disappears). By crossing of 'L' with '1/'–'11'–'00L' the signals 'R' and 'V' arise; also 'L' continues. Further this signal 'L' reflects between just created 'V' and the previously created 'V'. Again, by every reflection 'L' is changed to 'R' or conversely. A special delay is arranged on the left. So multiples of an odd integer $d > 3$ are recognized, and at appropriate moments the signals '/' are formed. The smallest integer recognized here as a multiple of $d > 5$ is $\frac{d.(d+5)}{2}$ (in particular, 42 for $d = 7$). Recognition of multiples of 5 starts from 20 (in the irregular part of F).

We have explained how the necessary values of the corresponding local rule f can be read from Figure 1. Notice than the values of f presented in the lower part (and some other values) can be obtained from the values used sooner when temporary changes caused by crossing with '/' are considered. For example, in the row 494 we need $f(\text{'r'}, \text{'/'}, \text{'/'}) = \text{'r'}$, which can be derived from $f(\text{'R'}, \text{'.'}, \text{'.'}) = \text{'R'}$.

Together values of f for 344 triples of arguments are necessary; the remaining $9^3 - 344 = 385$ triples never occur as subwords of the rows of F. Hence the corresponding values of f can be arbitrary. □

4. A 2-state 1D CA with radius 10

Theorem 2. *There is a 2-state 1D CA with radius* 10 *which has a computation* H *with the following properties:*

(1) $H(0,1) = \text{'1'}$ *and* $H(z,0) = \text{'0'}$ *for all* $z \neq 1$.
(2) *For all* $z < 0$ *and* $t \geq 0$ *it holds* $H(z,t) = \text{'0'}$.
(3) *For all* $t \geq 0$ *we have* $H(0,t) = \text{'1'}$ *if* t *is a prime, and* $H(0,t) = \text{'0'}$ *otherwise.*

Proof. Let F be the computation from Theorem 1 (and Figure 1) and let the function F_{10} be defined by the formula

$$F_{10}(z,t) = \begin{cases} \text{'!'} & \text{if } F(z,t) = \text{'.'} \text{ and } 0 < z < t \\ F(z,t) & \text{otherwise.} \end{cases}$$

It means that all '.' in the supports of the rows of F are replaced by '!'. An initial segment of F_{10} is presented in the left-hand part of Figure 2. We can easily see that F_{10} is a computation of a 10-state cellular automaton.

Now let us denote $U = \{u_{\texttt{.}}, u_{\texttt{/}}, u_{\texttt{0}}, \ldots, u_{\texttt{v}}, u_{\texttt{!}}\}$, where

$u_{\texttt{.}} = \text{'000000'}$, $u_{\texttt{/}} = \text{'011000'}$, $u_{\texttt{0}} = \text{'001000'}$, $u_{\texttt{1}} = \text{'011110'}$, $u_{\texttt{L}} = \text{'011010'}$,
$u_{\texttt{R}} = \text{'010110'}$, $u_{\texttt{r}} = \text{'000110'}$, $u_{\texttt{V}} = \text{'011100'}$, $u_{\texttt{v}} = \text{'001110'}$, $u_{\texttt{!}} = \text{'001010'}$.

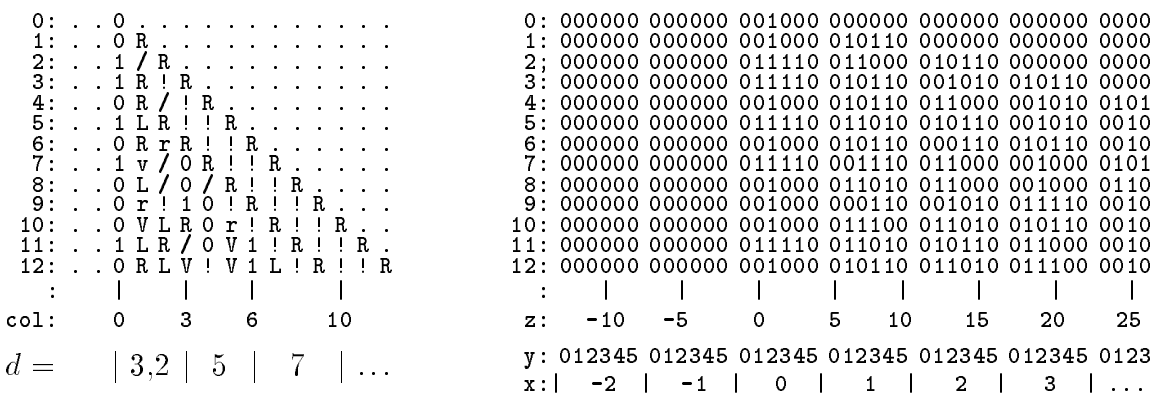

Figure 2. The computations F_{10} and H.

Let the *kernel* of a word $w \in U$ be the minimal subword of w which contains all '1' of w.

Let the function $H : \mathbb{Z} \times \mathbb{N} \to \{0,1\}$ be constructed so that every value $\mathtt{x}$ of F_{10} will be replaced by $u_{\mathtt{x}}$; six values of H arise from one value of F_{10}. The columns of H are enumerated so that the leftest column which contains '1' obtains the number 0. More formally, let for all $x \in \mathbb{Z}$, $0 \le y < 6$ and $t \in \mathbb{N}$

$$H(6x+y,t) = u_{F_{10}(x,t)}[y+1] \tag{4.1}$$

where $u[i]$ denotes the i-th symbol of the word $u = u_1 \dots u_6$. The function H and enumeration of its columns are presented in the right-hand part of Figure 2. Bigger spaces between some columns only show how H was formed; they cannot be immediately locally reconstructed from H itself. (Below we shall see that in some important cases they can be locally reconstructed.)

The function H obviously satisfies the conditions (1)–(3) of Theorem 2. It remains to prove that H is a computation of a cellular automaton of radius 10, i.e. that for every $z \in \mathbb{Z}, t \in \mathbb{N}$ the value $H(z,t+1)$ is uniquely determined by the word

$$W_{z,t} = H(z-10,t)H(z-9,t)\dots H(z+9,t)H(z+10,t).$$

Let $z = 6x+y$, $0 \le y < 6$. The word $W_{z,t}$ contains the kernels of the words

$$u_{F_{10}(x-1,t)}, \quad u_{F_{10}(x,t)}, \quad u_{F_{10}(x+1,t)}. \tag{4.2}$$

(We can also see that $u_{F_{10}(x-1,t)}u_{F_{10}(x,t)}u_{F_{10}(x+1,t)}$ is a subword of $0W_{z,t}0$.)

Now assume that besides $W_{z,t}$ also y is given. (We need not know x and z.) Then we can reconstruct the words (4.2), and hence the symbols $F_{10}(x-1,t)$, $F_{10}(x,t)$, $F_{10}(x+1,t)$. Then we can determine $F_{10}(x,t+1)$ by the local rule for F_{10}, and finally $H(z,t+1)$ by the formula (4.1).

It remains to determine y from $W_{z,t}$. It is not always possible (example: $W_{z,t}$ consisting of '0' only). However, if it is impossible then we have $F_{10}(x,t+1) =$ '.' and hence $H(z,t+1) =$ '0'.

Informally, $W_{z,t}$ is a subword of a word from U^+, and to determine y means to determine the boundaries between the used elements of U. As soon as these boundaries are known we can determine the words, then $F_{10}(x-1,t)$, $F_{10}(x,t)$, $F_{10}(x+1,t)$, then $F_{10}(x,t+1)$, and finally $H(z,t+1)$.

We shall look for maximal nonempty subwords of $W_{z,t}$ which contain no '00' inside and no '0' at the ends. If such a word is surrounded by by '00' from both

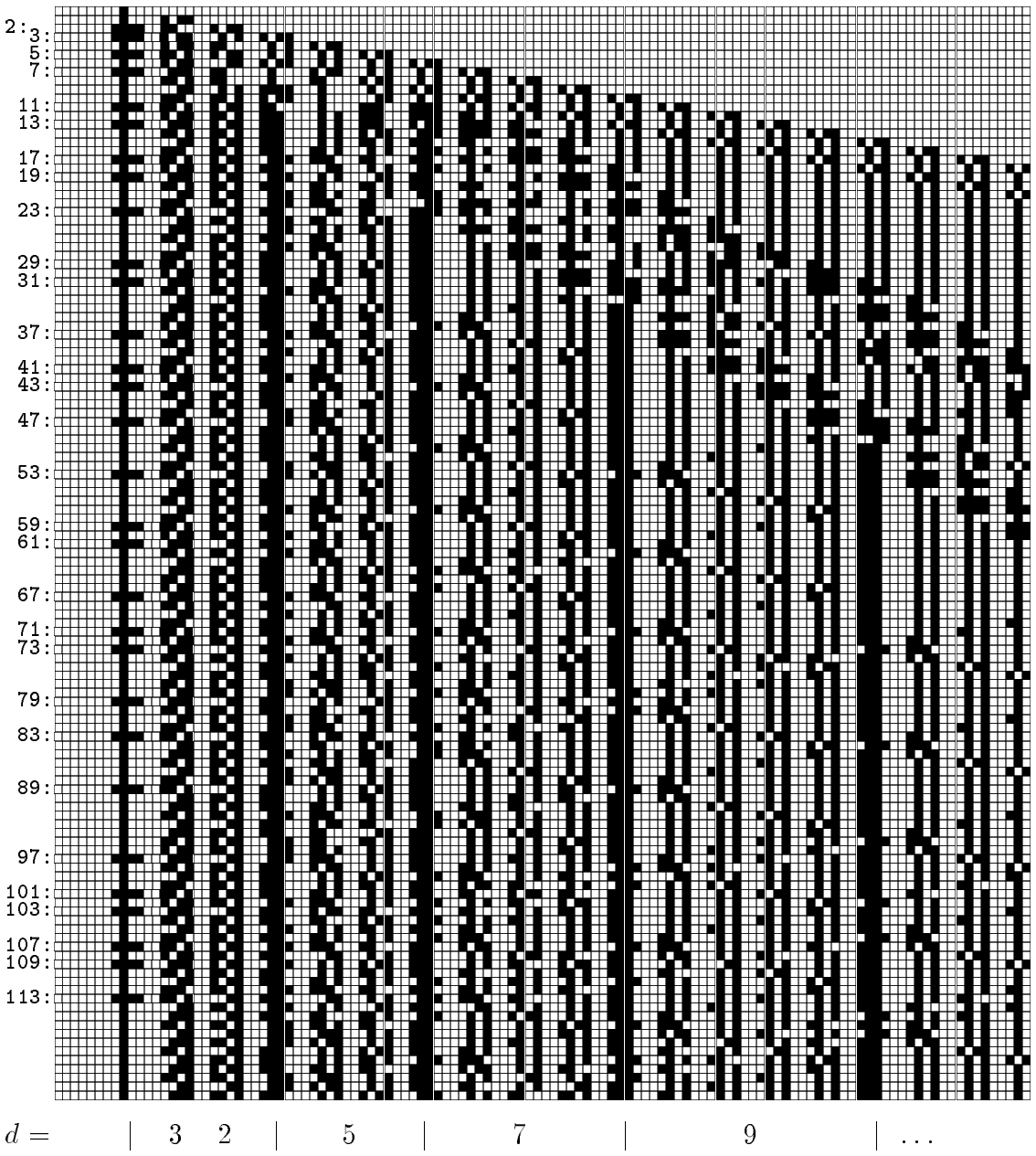

Figure 3. Real-time generation of the primes by a 2-state 1D CA with $r \leq 10$.

sides it must be the kernel of a word $u_{\mathtt{x}}$ which was really used in the construction of H. For the words '1111', '1101', '1011' we know that even without this surrounding, and for '101' and '111' surrounding by '0' suffice. Let us call such subwords of $W_{z,t}$ *safe kernels.*

If $W_{z,t}$ contains no safe kernel then $H(z, t+1) =$ '0'. If it contains exactly one safe kernel, then it must be the kernel of $u_{\mathtt{0}}$, $u_{\mathtt{1}}$, or $u_{\mathtt{R}}$. In these cases y obviously can be determined.

Each of the words '1111', '1101', '1011', '101', or '1' is the kernel of exactly one element of U. Therefore if any of them is among safe kernels of $W_{z,t}$ then y (informally: the position of spaces if $W_{z,t}$ is chosen from Figure 2) is uniquely determined. Each of the words '111' or '11' is the kernel of two elements of U. Therefore if it is among the safe kernels of $W_{z,t}$ it leaves us two possibilities for y, the correct one, and one shifted ± 1 or ± 2 modulo 6. The incorrect possibilities are distinct for distinct elements of $\{$'V','v','/','r'$\}$. Therefore if at least two distinct

elements of $\{$'V','v','/','r'$\}$ are among $F(x-1,t)$, $F(x,t)$, $F(x+1,t)$ then y is uniquely determined.

If $W_{z,t}$ contains two safe kernels '111' with '000' between them then these kernels must correspond to 'vv' because 'VV' is not contained in (any row of) F_{10}. It remains to distinguish the cases with 'rr' and '//'. Notice that 'rr' occurs in F_{10} only as a subword of '0rr'. In this case $W_{z,t}$ contains '000000' in its left half. It cannot happen for '//'. □

Figure 3 contains a segment of H. To obtain a more transparent figure, black squares are used instead of '1' and empty squares are used instead of '0'. The rows with primes are enumerated only, and the areas for the divisors are approximately shown.

5. A generalized Pascal triangle

Theorem 3. *There is an algebra $\mathcal{A} = (\mathbf{A}; *, \text{'.'})$ such that '.' is its idempotent,* $\mathrm{card}(A) = 75$ *and for some $w \in \mathbf{A}$ the function $G = \mathrm{GPT}(\mathcal{A}, w)$ has the following properties:*

(1) $G(0,0) \neq$ '.'.
(2) *For all $x, y \in \mathbb{N}$, if $x - y < -1$ then $G(x,y) =$ '.' .*
(3) *For all $t \in \mathbb{N}$ we have $G(t, t+1) =$ '1' if $2t+1$ is a prime, and $G(t,t+1) =$ '0' otherwise.*

Proof. The requested GPT G is displayed in Figure 4. However, the elements '.', '0', '1' from the theorem are replaced by '..', '.0', '.1'. (A "leading dot" is used similarly as sometimes leading zeros are used in writing integers.) Figure 4 is obtained simply by suitable gluing the elements of F from Figure 1 into ordered pairs. (Technically, spaces are made between not glued elements, and ' ' are not used.) Therefore G clearly reaches at most 9^2 distinct values. However, some pairs never occur as values of G, and need not be included into $\mathbf{A}$. The computer simulation found only 75 values of G, and it can be proved that no new pairs occur later.

The 0-th column of F is now distributed into the 0-th column of G (even rows; the first components) and the (-1)-st column of G (odd rows; the second components). So it is arranged that (3) holds. □

As a consequence of Theorem 3 we can obtain:

Corollary 4. *There is a 75-state one-dimensional cellular automaton with neighbourhood type $(-1,1)$ and its computation K such that:*

(1) $K(z,0) =$ '.' *for all $z \in \mathbb{Z} \setminus \{0,1\}$.*
(2) *For all $z < 0$ and $t \geq 0$ it holds $K(z,t) =$ '.'.*
(3) *For all $t \geq 0$ we have $K(0,t) =$ '1' if t is a prime, and $K(0,t) =$ '0' otherwise.*

Proof. The computation K can be composed from two GPT, a trivial one which generates only $\{2\}$, and G from Theorem 3. The elements of any of them will fill the gaps between neighbour elements of the other (like black and white fields in the chessboard); the area outside GPT will be filled by quiescent state. □

Remarks. 1. Our GPT G was obtained directly from F. The bound 75 can be diminished by a modification of G. For example, some elements of the form '/X' can be identified with '.X' because '/' in this pair would exclude even integers t.

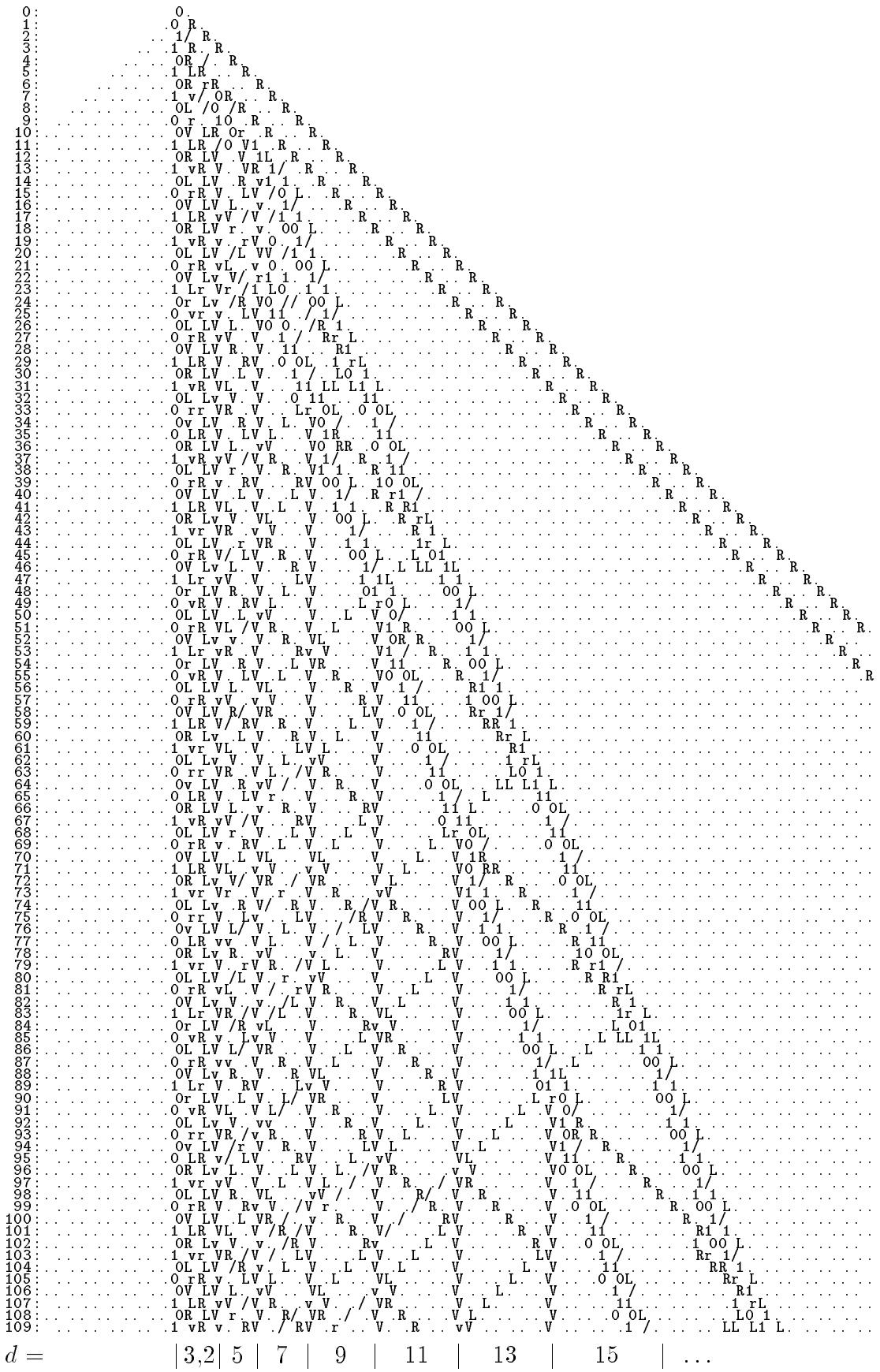

Figure 4. A generalized Pascal triangle which generates odd primes.

2. The corollary can be reformulated (with the constant 75) also to one-way

CA, i.e. 1D CA with neighbourhood type $(0,1)$ or $(-1,0)$; see [CHY] (and [Dy]). However, in these cases the result cannot be found in a fixed cell; it ought to be read in the leftest non-quiescent cell. Further, several states will correspond to composed integers.

6. A lower bound for usual 1D CA

We shall consider the same type of 1D CA as in Section 3 (i.e., those with radius 1). The symbol '.' will be the quiescent state; two further states will be '0' and '1', the other states are not specified. A computation of such 1D CA will be called *normal* if it satisfies the conditions (1) and (2) of Theorem 1.

For functions F with the domain $\mathbb{Z} \times \mathbb{N}$ we define

$$\mathrm{FirstErr}(F) = \inf\big\{t \in \mathbb{N} \mid (t \in \mathbb{P} \wedge F(0,t) \neq 1) \vee (t \notin \mathbb{P} \wedge F(0,t) \neq 0)\big\},$$

where $\mathbb{P}$ denotes the set of primes; we shall only apply this definition for normal 1D CA computations. Notice that $\mathrm{FirstErr}(F)$ can be either an element of $\mathbb{N}$ or $\infty = \inf \emptyset$. The condition (3) of Theorem 1 corresponds to $\mathrm{FirstErr}(F) = \infty$. Finally, for every integer $n \geq 3$ we define

$$\mathrm{FirstErr}(n) = \max\big\{\,\mathrm{FirstErr}(F) \;\big|\; F \text{ is a normal computation of a 1D CA with } n \text{ states}\big\}.$$

Now Theorem 1 can be expressed by the formula $\mathrm{FirstErr}(9) = \infty$.

Claim 5. $\mathrm{FirstErr}(3) = 38$.

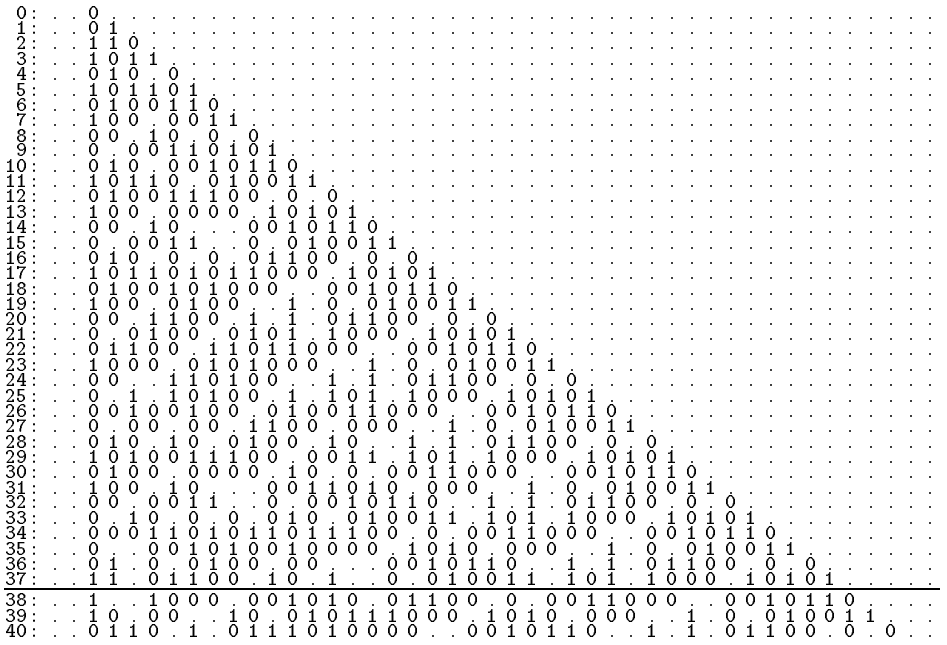

Figure 5. The computation F_3 which shows $\mathrm{FirstErr}(3) \geq 38$.

Proof. By a computer computation; it took about 8 minutes on PC 80386. In essential, all 3^{27} possible local rules were considered. However, many group of them could be considered together, and so only about 97000 cases were indeed considered. A 3-state normal computation F_3 such that $\mathrm{FirstErr}(F_3) = 38$ is presented in Figure 5. (Of course, F_3 itself shows only $\mathrm{FirstErr}(3) \geq 38$.) $\square$

As an immediate corollary we obtain:

Theorem 4. *At least* 4 *states are necessary for real-time generation of primes in the sense of Theorem 1.*

It seems difficult to prove a better lower bound by a similar direct computation. To compute FirstErr(4), 4^{64} local rules ought to be considered. Notice only that FirstErr(4) ≥ 97.

7. Remarks and problems

The main related problem is to diminish the gap between the upper bound 9 in Theorem 1 and the lower bound 4 in Theorem 4. We can also try to diminish the radius $r = 10$ in Theorem 2, and to find a lower bound for it. (Maybe, the value $r = 9$ could be obtained with a slight modification of H or even without changing it at all.)

Another problem concern the necessary computation space (measured as the length of the segment which contains all non-quiescent cells):

Conjecture 1. *The space* $O(\sqrt{t}\,)$ *suffices for real-time generation of the primes by 1D CA.*

Here the idea of sieve of Eratosthenes probably again can be applied. However, counting up to d must be performed in the space $O(\log d)$. Further, only prime divisors ought to be considered. Of course, the number of states can be much larger than above.

Problem 1. *Does the space* $O((\log t)^k)$ *(for a fixed* k*) suffice for real-time generation of the primes by 1D CA?*

Problem 2. *Is there a Turing machine which generates the primes in real time?*

Here we ask the signals to appear in a fixed position of a fixed tape; another possibility is a special (real time) output tape. The problem can be stated for various kinds of Turing machines (and also for linear time instead of real time).

References

[Bo] B. A. Bondarenko, *Generalized Pascal triangles and pyramids, their fractals, graphs and applications (in Russian)*, "Fan", Tashkent, 1990.

[CHY] Culik, K. II — Hurd, L. P. — Yu, S., *Computation theoretic aspects of cellular automata*, Physica D **45** (1990), 357–378.

[Dy] Dyer, C. R., *One-way bounded cellular automata*, Information and Control **44** (1980), no. 3, 261–281.

[Fi] Fischer, P. C., *Generation of primes by a one-dimensional real-time iterative array*, J. ACM **12** (1965), no. 3, 388–394.

[K1] I. Korec, *Generalized Pascal triangles*, In: K. Halkowska and S. Stawski, ed.: Proceedings of the V Universal Algebra Symposium, Turawa, Poland, May 1988, World Scientific, Singapore, 1989, pp. 198–218.

[K2] ———, *Generalized Pascal triangles, their relation to cellular automata and their elementary theories*, J. Dassow, A. Kelemenová (Eds.): Development in theoretical computer science, Proceedings of the 7th IMYCS, Smolenice 92, November 16-20, 1992, Gordon and Breach Science Publishers, 1994, pp. 59–70.

[K3] ———, *Real-time generation of primes by a one-dimensional cellular automaton with 11 states*, I. Prívara, P. Ružička (Eds.): Mathematical Foundations of Computer Science 1977, Bratislava, August 25–29, 1997, Springer LNCS 1295, 1997, pp. 358–367.

Mathematical Institute, Slovak Academy of Sciences,
Štefánikova 49, 81473 Bratislava, Slovakia

This preprint series was founded in 1996. Its purpose is to present manuscripts of submitted or unpublished papers and reports of fellows of the Mathematical Institute of the Slovak Academy of Sciences in Bratislava and its Košice branch. The authors are fully responsible for the content of the preprints. Postal addresses:

Mathematical Institute
Slovak Academy of Sciences
Štefánikova 49
SK–814 73 Bratislava
SLOVAKIA

Tel./Fax (+421 7) 5249 7316

Mathematical Institute
Slovak Academy of Sciences
Grešákova 6
SK–040 01 Košice
SLOVAKIA

Tel./Fax (+421 95) 6228291

PostScript files of preprints are available at WWW address:
`http://www.mat.savba.sk/preprints`

Latest preprints of the series:

13/1997 *Ivan Korec*: Real-time generation of primes by a one-dimensional cellular automaton with 9 states

12/1997 *Ivan Korec*: Arithmetical operations strongest with respect to the first order definability

11/1997 *Otokar Grošek, Karol Nemoga, Marcel Zanechal*: Why use bijective S-boxes in GOST-algorithm

10/1997 *Beloslav Riečan*: On the L^p space of observables

9/1997 *Ladislav Mišík Jr., Tibor Žáčik*: A formula for calculation of metric dimension of converging sequences

8/1997 *Anatolij Dvurečenskij, Maria Gabriella Graziano*: Commutative BCK-algebras and lattice ordered groups

7/1997 *Anatolij Dvurečenskij, Maria Gabriella Graziano*: On representations of commutative BCK-algebras

6/1997 *Sylvia Pulmannová, Karl Svozil*: Ideals in ortholattices, Bell inequalities and simultaneously definite properties

5/1997 *Camille Debiève, Miloslav Duchoň, Beloslav Riečan*: Moment problem in some ordered vector spaces

4/1997 *Miloslav Duchoň, Beloslav Riečan*: On the Kurzweil–Stieltjes integral in ordered spaces

3/1997 *Anatolij Dvurečenskij, Maria Gabriella Graziano*: Remarks on representations of minimal clans

2/1997 *Ivan Korec*: Elementary definability from Pascal's triangle modulo p and the set of e-th powers

1/1997 *Ján Jakubík*: Lexicographic products of half linearly ordered groups

34/1996 *Emília Halušková*: Direct limits of monounary algebras

33/1996 *Ivan Korec*: A list of arithmetical structures strongest with respect to the first order definability

32/1996 *Ivan Korec*: Definability of Pascal's triangles modulo 4 and 6 and some other binary operations from their associated equivalence relations

31/1996 *Anatolij Dvurečenskij*: Measures and states on BCK-algebras

30/1996 *Ján Haluška*: Uncertainty and tuning in music

29/1996 *Ján Haluška*: On numbers 256/243, 25/24, 16/15

28/1996 *Ján Haluška*: COMMA 32 805/32 768

27/1996 *Beloslav Riečan*: On the sum of observables in MV algebras of fuzzy sets

(continued on inside back cover)

26/1996 *Beloslav Riečan*: On the strong law of large numbers for weak observables in MV algebras

25/1996 *Ivan Korec*: Algebraical constructions of reversible generalized Pascal triangles and one–dimensional cellular automata

24/1996 *Peter Vojtáš*: Uncertain reasoning with floating connectives

23/1996 *Ivan Korec*: Definability of addition from multiplication and neighbourhood relation and some related results

22/1996 *Ivan Korec*: Open problems more or less related to generalized Pascal triangles

21/1996 *Sylvia Pulmannová*: On connections among some orthomodular structures

20/1996 *Pekka Lahti, Sylvia Pulmannová*: Coexistent observables and effects in quantum mechanics

19/1996 *Ivan Korec*: Definability of arithmetical operations from Pascal's triangle modulo n and arithmetical functions

18/1996 *Ivan Korec*: Undecidability and uniform definability in classes of structures related to Pascal's triangles modulo n

17/1996 *Beloslav Riečan*: Probability theory on some ordered structures

16/1996 *Beloslav Riečan*: Weak observables in MV algebras

15/1996 *Anatolij Dvurečenskij, Hee Sik Kim*: On connections between BCK-algebras and difference posets

14/1996 *Anatolij Dvurečenskij, Karl Svozil*: Product of partition logics, orthoalgebras and automata

13/1996 *Robin L. Hudson, Sylvia Pulmannová*: Chaotic expansion of elements of the universal enveloping algebra of a Lie algebra associated with a quantum stochastic calculus

12/1996 *Sylvia Pulmannová*: Congruences in partial abelian semigroups

11/1996 *Karol Nemoga, Štefan Schwarz*: An explicit description of the set of all normal bases generators of a finite field

10/1996 *Ján Jakubík*: Subdirect product decompositions of MV-algebras

9/1996 *Ivan Korec*: Reversibility in generalized Pascal triangles and binary reversibility in one-dimensional cellular automata

8/1996 *Jozef Bobok, Milan Kuchta*: X-minimal patterns and generalization of Sharkovskiĭ's theorem

7/1996 *Jarmila Hedlíková, Sylvia Pulmannová*: Generalized difference posets and orthoalgebras

6/1996 *Ján Borsík, Roman Frič*: Pointwise convergence fails to be strict

5/1996 *Miroslav Ploščica*: Affine completions of distributive lattices

4/1996 *Ján Haluška*: Equal temperament and Pythagorean tuning: a geometrical interpretation in the plane

3/1996 *Winfried Just, Peter Vojtáš*: On matrix rapid filters

2/1996 *Anatolij Dvurerečenskij*: Fuzzy set representations of some quantum structures

1/1996 *Miloslav Duchoň, Beloslav Riečan*: On the product of semigroup valued measures