International Journal of Theoretical Physics, Vol. 21, No. 12, 1982

# Some Remarks about the "Efficiency" of Polyautomata

**R. Vollmar**

*Technische Universität Braunschweig, Braunschweig, Federal Republic of Germany*

*Received May 8, 1981*

A proposal is made to measure the "efficiency" of cellular algorithms, implemented in cellular automata, roughly speaking by the ratio of the number of proper state changes and the product of time, and number of single automata. Such a definition is discussed in some detail. For some cellular algorithms lower bounds for their efficiency are given.

## INTRODUCTION

Probably pattern recognition will become one of the most important tasks array processors will be used for. Formal languages may be understood as patterns and cellular automata may serve as models for array processors.

Then it seems interesting to have a measure for the effort cellular automata have to spend recognizing different pattern classes. Also it would be useful to know, e.g., the needed energy.

Here a proposal is made to consider the number of proper state changes as a complexity measure and to define the efficiency of cellular automata as the ratio of the sum of such changes and the possible number of changes.

## NOTATIONS AND DEFINITIONS

The starting point is a one-dimensional bounded cellular space [in the notation of Smith (1970)] or [in the sense of Vollmar (1979)] a one-dimensional cellular automaton (ca).

It is a chain of identical finite, deterministic automata where each automaton is connected with its two immediate neighboring automata. A



0020-7748/82/1200-1007$03.00/0 

(deterministic) global transition function describes the global behavior of the ca in one transition step.

To use ca as language recognizers some conventions are needed: Let $L \subseteq X^+$ be a formal language. (In the following the empty word will be excluded.) A word $w \in X^+$ for which is to recognize whether $w \in L$ or not, is coded (one symbol per automaton) within a simple-connected region of single automata, called retina. The retina is bounded by two so-called border automata which do not change their (special) states. The first automaton of the retina, called accepting automaton, shows by assuming special states ($\alpha$ and $\omega$) the result of the recognition process ($w \in L$, or, respectively, $w \notin L$).

Such a ca is called a ca recognizer (for the language $L$). The covering of the retina and of the border automata with states at time $t=0$ is denoted $c_0^w$, the initial configuration attached to $w$. $c_0^w$ does not contain $\alpha$ and $\omega$. Starting with $c_0^w$ by repeated application of the global transition function a sequence of configurations $c_0^w, c_1^w, \ldots,$ the so-called propagation $\langle c_0^w \rangle$ is generated. The deterministic transition behavior of the ca implies for a given $c_0^w$ the existence of a unique propagation.

It is known that for the recognition process only a finite section (depending on the length of $w$) of the propagation is necessary. The "subpropagation" consisting of the first $l+1$ elements of $\langle c_0^w \rangle$ is called $l$ propagation and denoted $\langle c_0^w \rangle |_l$.

Two measures are considered:

(1) maxc($c_0^w$): maximum of the number of proper state changes over the automata in the retina during the recognition process

(2) sumc($c_0^w$): sum of the number of proper state changes of automata in the retina during the recognition process

*Proper* state changes are motivated among others by physical reasons: The energy to change the state of CMOS-chips is much higher than that of holding the state.

To formalize the measures some further notations have to be introduced:

In the following it is assumed that for a given word $w = w^1 \ldots w^n$ with $n \geqslant 1$, the symbol $w^i$ ($1 \leqslant i \leqslant n$) is coded to the state of the automaton with coordinate $i$—called automaton $i$ for short.

The current state of the automaton $i$ at time $t$ is denoted $c_t^w(i)$.

In the sequel so-called real-time ca recognizers will be considered: If for no input word of length $n$ a ca recognizer $\mathscr{C}$ for a language $L$ needs more than $n$ transition steps before accepting or rejecting the word, $\mathscr{C}$ is said to be a real-time ca recognizer for $L$.

*Definition 1.* Let $\mathscr{C}$ be a ca, $w \in X^+, c_0^w$ the initial configuration attached to $w$.

(a) The number of proper state changes (or simply changes) of an automaton $i$ $(1 \leq i \leq |w|)$[1] before the moment $l$ is defined by[2]

$$s_l(i) := |\{t / c_t^w(i) \neq c_{t+1}^w(i) \wedge t < l\}|$$

(b) The maximal number of changes for the $l$ propagation $\langle c_0^w \rangle|_l$ is defined by

$$\mathrm{maxc}(\langle c_0^w \rangle|_l) := \max_{1 \leq i \leq |w|} \{s_l(i)\}$$

(c) The sum of changes for the $l$ propagation $\langle c_0^w \rangle|_l$ is defined by

$$\mathrm{sumc}(\langle c_0^w \rangle|_l) := \sum_{i=1}^{|w|} s_l(i)$$

Analogous definitions are used for propagations.

*Definition 2.* Let $\mathcal{C}$ be a ca recognizer for $L \subseteq X^+$, $f$: $\mathbb{N} \to \mathbb{N}$[3] a function, $w$, $c_0^w$ as above.

(a) A propagation $\langle c_0^w \rangle$ is called $f$-max-(sum-)change-bounded if the following holds:

$$\mathrm{maxc}(\langle c_0^w \rangle) \leq f(|w|), \quad \text{or, respectively,} \quad \mathrm{sumc}(\langle c_0^w \rangle) \leq f(|w|)$$

(b) $\mathcal{C}$ is called $f$-max-(sum-)change-bounded if it holds that

$$\forall w \in X^+ : \langle c_0^w \rangle \text{ is } f\text{-max-(sum-)change-bounded}$$

The main tool to prove the results will be modified crossing sequences [following Hennie (1965)]. In this connection the used ca has to be fixed: In the sequel ca $\mathcal{C} = (A, 1, H_1, F)$ will be considered. $A$ denotes the state set of the single automata the ca is composed of, the dimension is 1, $H_1$ denotes the so-called von Neumann neighborhood which is characterized by the direct connection of the two immediately neighboring automata to each automaton, and $F$ stands for the global transition function.

*Definition 3.* (a) The "spatial" restriction of an $l$ propagation $\langle c_0^w \rangle|_l$ containing two neighboring automata $i, i+1$ with $1 \leq i \leq |w|$ is denoted $\langle c_0^w(i, i+1) \rangle|_l$ and called $l$-propagation section. If the restriction to one

[1] $|w|$ denotes the length of word $w$.
[2] $|M|$ denotes the cardinality of $M$. From the context the meaning of $|\cdots|$ will become clear.
[3] $\mathbb{N}$ denotes the set of natural numbers (without zero).

automaton $i, 1 \leq i \leq |w|$, is considered, the notation $l$-propagation 1-section $\langle c_0^w(i) \rangle|_l$ is used.

(b) Two $l$-propagation sections $\langle c_0^w(i, i+1) \rangle|_l$ and $\langle c_0^w(j, j+1) \rangle|_l$ are called equal if the following holds:

$$\forall t \in \{0, 1, \ldots, l\}: \left[c_t^w(i) = c_t^w(j) \wedge c_t^w(i+1) = c_t^w(j+1)\right]$$

## RESULTS

In this section some results will be cited (and proved in part) which should make clear that there are some reasons to consider the proposed measure coincident with some intuitive ideas about "parallelism inherent in problems."

As a first approach for the definition of efficiency of ca the ratio of the number of changes of the "busiest" automaton and the time needed to recognize a language will be discussed. But it is clear that of greater interest will be the total amount of load of the ca. At a first thought one would conjecture that there is no great difference between the results of such measurements, i.e., that the max-change number and the appropriately adapted sum-change number are equal (at least at the order of magnitude) —in analogy to a result of Hennie (1965) for Turing machines. But the following holds:

*Proposition 1.* The class of languages recognizable by $f$-max-change-bounded real-time ca recognizers with $f(|w|) = k$, $k \in \mathbb{N}$ fixed, is properly included in the class of languages recognizable by $O(|w|)$-sum-change-bounded real-time ca recognizers.[4]

The proposition which will not be proved here states that real-time ca recognizers which need a constant number of proper state changes *in the mean* are properly more powerful than real-time ca recognizers with a constant number of changes per single automaton. Therefore it seems meaningful to define the efficiency of a ca recognizer $\mathcal{C}$ recognizing words $w \in X^+$ as

$$\Pi(\mathcal{C}, w) := \mathrm{sumc}\left(\langle c_0^w \rangle|_{t(w)}\right) \cdot \left(|w| \cdot t(w)\right)^{-1}$$

where $t$: $X^+ \to \mathbb{N}$ gives the time needed for the recognition of words.

For $n \in \mathbb{N}$ we define

$$\Pi(\mathcal{C}, n) := \max_{w \in X^+ \wedge |w| = n} \Pi(\mathcal{C}, w)$$

[4] $f = O(g)$ means that there exists a $k \in \mathbb{N}$, such that almost everywhere $f \leq kg$ holds.

The maximum is chosen to take into consideration, e.g., the maximal needed energy during the recognition of words of length $n$. It should be mentioned that—in analogy to definitions for Turing machines—complexity is (implicitly) defined for algorithms, i.e., for ca recognizers, and not for languages.

For $\Pi$ defined in this way the following holds:

$$n^{-1} \leqslant \Pi(\mathcal{C}, n) \leqslant 1$$

By definition the second inequality is clear and the first one follows immediately by the fact that before the end of the recognition process at each moment at least one automaton must change its state; otherwise by the deterministic behavior a stable configuration is reached.

From proposition 1 of Vollmar (1981) it follows immediately that the regular languages (without the empty word) are recognizable by $|w|$-sum-change-bounded real-time ca recognizers. From the same proposition it is derivable that the languages $\{a^k X a^k / k \in \mathbb{N}\}$ and $\{a^k X b^k Y c^k / k \in \mathbb{N}\}$ are accepted by $O(|w|)$-sum-change-bounded real-time ca recognizers. This implies the following:

*Proposition 2.* (a) To any regular language $L$ (without the empty word) there exist real-time ca recognizers $\mathcal{C}$ for $L$ with $\Pi(\mathcal{C}, n) = n^{-1}$.

(b) There exists a language $L \in \mathfrak{L}_2 \setminus \mathfrak{L}_3$ for which a real-time ca recognizer $\mathcal{C}$ exists with $\Pi(\mathcal{C}, n) = O(n^{-1})$.

(c) There exists a language $L \in \mathfrak{L}_1 \setminus \mathfrak{L}_2$ for which a real-time ca recognizer $\mathcal{C}$ exists with $\Pi(\mathcal{C}, n) = O(n^{-1})$.

This means that for these examples the capabilities of ca have only been used in a very restricted manner—and this coincides with the intuition that for such "simple" languages parallel working automata are "too powerful."

But there exist examples of problems which can be solved by ca only with great effort. First it is shown that there are languages which can only be recognized with an efficiency "close to" 1, and then the firing squad synchronization problem (fssp) is discussed which can only be solved with an efficiency in the order of $n^{-1} \ln n$.

Two lemmas are needed:

*Lemma 1.* Let $\mathcal{C} = (A, 1, H_1, F)$ be a real-time ca recognizer for a language $L \subseteq X^+$, $w \in X^+$ with $|w| = n$. Let $r: \mathbb{N} \to \mathbb{N}$ be a function such that for almost all $n$ it holds that $r(n) \leqslant n$. Then the number of different $n$-propagation sections $\langle c_0^w(i, i+1) \rangle |_n, 1 \leqslant i \leqslant n$, with $< r(n)$ changes is bounded by

$$[|A| \cdot (n+1)]^{2r(n)}$$

*Proof.* To estimate the number of different $n$-propagation 1-sections $\langle c_0^w(i)\rangle|_n$, where $1 \leq i \leq n$, vectors of $n+1$ components are considered, where at $r(n)$ places elements of $A$ are contained; the other components are filled by the respectively following and at the end by the respectively preceding element. The $n$-propagation 1-sections $\langle c_0^w(i)\rangle|_n$ with $< r(n)$ changes are contained in this set of vectors. The number of elements in this set is $\leq \binom{n+1}{r(n)}|A|^{r(n)}$, which is $\leq (|A|\cdot(n+1))^{r(n)}$. From this the assertion follows. ■

*Lemma 2.* Let $\mathscr{C} = (A, 1, H_1, F)$ be a real-time ca recognizer for $L := \{uYYu^R / u \in \{a, b\}^+\}$. Let $w, \bar{w} \in L$ with $|w| = |\bar{w}| = n$, and let $c_0^w$, $c_0^{\bar{w}}$ be the initial configurations attached to $w$, $\bar{w}$, respectively. For $w = w_1 w_2$ and $\bar{w} = \bar{w}_1 \bar{w}_2$ with $|w_1| = |\bar{w}_1| = j$, where $1 \leq j < n/2 - 1$ holds: If $\langle c_0^w(j, j+1)\rangle|_n$ and $\langle c_0^{\bar{w}}(j, j+1)\rangle|_n$ are equal, then $w_1 = \bar{w}_1$.

*Proof.* A transition of each automaton is done in dependence on the state of this automaton and of the states of the two immediate neighbors. If $\mathscr{C}$ works with the initial configuration attached to $w_1\bar{w}_2$, then at each moment the state of the automaton $j$ and the states of its two neighbors have not changed with respect to the initial configuration attached to $w_1 w_2$ (since by assumption the $n$-propagation sections $\langle c_0^w(j, j+1)\rangle|_n$ and $\langle c_0^{\bar{w}}(j, j+1)\rangle|_n$ are equal). Therefore $\mathscr{C}$ also accepts $w_1\bar{w}_2$. According to the construction of $L$ then it must hold that $w_1 = \bar{w}_1$. ■

*Proposition 3.* Let $\phi(n) = o(n^{2-\epsilon})$[5] with an arbitrary, but fixed $\epsilon > 0$. Then there does not exist a $\phi(n)$-sum-change-bounded real-time ca recognizer $\mathscr{C} = (A, 1, H_1, F)$ for $L := \{uYYu^R / u \in \{a, b\}^+\}$, or, respectively, for $L' := \{uYYu / u \in \{a, b\}^+\}$.

*Proof.* The assertion will be proved for the language $L$; but with the same words the proof for $L'$ can be given. The proof will follow the following plan: First the number of elements from $L$ of length $n$ is determined for which an acceptance can be done with less than $\lceil kn^{1-\epsilon}\rceil$ changes in two automata. Then the number of corresponding words is counted which can be accepted in such a way that at least two automata from the region $\lceil n/4\rceil, \ldots, n/2 - 1$ have less than $\lceil kn^{1-\epsilon}\rceil$ changes. Finally it is shown that the ratio of this number and the number of all words of $L$ tends to 0 for $n \to \infty$. In the following let $\epsilon$ and $k$ be arbitrary, but fixed numbers with $\epsilon > 0$ and $0 < k < 1$ and let $w \epsilon (\{a, b\} \cup \{Y\})^+$ with $|w| = n$. From Lemma 1 the number of different $n$-propagation sections $\langle c_0^w(i, i+1)\rangle|_n$ is $\leq (|A|\cdot(n+1))^{2\lceil kn^{1-\epsilon}\rceil}$. From Lemma 2 words from $L$ with the same

[5] $f = o(g)$ means that $\lim_{n\to\infty} f(n)/g(n) = 0$.

$n$-propagation section $\langle c_0^w(i,i+1)\rangle|_n$, where $i<n/2-1$, must possess the same initial subword of length $i$. There are in $L$ $2^{n/2-1-i}$ different words which have the same subword of length $i$. Therefore the number of words from $L$ with $<\lceil kn^{1-\epsilon}\rceil$ changes by each of the automata $i$ and $i+1$ is

$$\leqslant 2^{n/2-1-i}(|A|\cdot(n+1))^{2\lceil kn^{1-\epsilon}\rceil}$$

This implies that there exist

$$\leqslant \lceil n/4\rceil 2^{n/2-1-\lceil n/4\rceil}\cdot(|A|\cdot(n+1))^{2\lceil kn^{1-\epsilon}\rceil}$$

different words in $L$ of length $n$ which are accepted by at least two automata at the positions $\lceil n/4\rceil,\ldots,n/2-1$ with $<\lceil kn^{1-\epsilon}\rceil$ changes. (If all automata in this region would need more than $\lceil kn^{1-\epsilon}\rceil$ changes, then the assertion would also be true.) But the number of words in $L$ of length $n$ equals $2^{n/2-1}$. The sequence

$$\lceil n/4\rceil 2^{\lceil n/4\rceil}\cdot[|A|\cdot(n+1)]^{2\lceil kn^{1-\epsilon}\rceil}\cdot 2^{1-n/2}$$

tends to 0 for $n\to\infty$. From this follows the assertion. ■

*Remark.* The proof follows the lines of a proof of Trachtenbrot (1977) with which it is shown that a Turing machine for recognition of palindromes needs time $O(n^2)$. The essential difference lies in the modification of crossing sequences.

*Corollary*. For a real-time ca recognizer $\mathcal{Q}$ for $L$ ($L'$) there holds

$$\Pi(\mathcal{Q},n)>O(n^{-\epsilon})$$

The two languages $L$ and $L'$ also have the same complexity from the point of property specification as Salomaa (1973) states about slightly different languages.

Incidentially it is derivable from the result stated above that a pattern transformation problem, namely, the reflection of an arbitrary word, can only be done in ca with a number of changes "close to" $n^2$, too.

Defining $\Pi(\mathcal{Q},n)$ for ca in an obvious analogy to the one for ca recognizers, a lower bound for the number of proper state changes for the solution of the fssp is derived.

Solutions of this and of modified problems may be widely used not only as universal methods for the synchronization of nets of automata but also in connection with problems of the recognition of formal languages and of pattern transformation.

It is well known that the minimal time to solve the "classical" fssp, i.e., the one for a one-dimensional ca of $n$ automata with $H_1$ neighborhood and the general at one end, is $2n-2$.

*Proposition 4.* Let $\mathfrak{A}$ be a ca which solves the fssp in minimal time. The number of proper state changes for $\mathfrak{A}$ is at least $O(n \ln n)$.

*Proof.* From the assumptions for the fssp it is clear that the transition from the moment $2n-3$ to $2n-2$ must change the states of all automata. Since a deterministic behavior is assumed, to reach this at the preceding step at least one automaton in the $H_1$-neighborhood must have changed its state. This implies that then at least $\lceil n/3 \rceil$ changes have happened. During $k$ steps each automaton may be influenced by $|H_k| = 2k+1$ automata. Therefore at the moment $2n-2-k$ in the $H_k$-neighborhood of each automaton at least one state change must have occurred, i.e., at this moment at least $\lceil n \cdot (2k+1)^{-1} \rceil$ changes must have taken place. Therefore the number of changes is greater than

$$\sum_{k=0}^{2n-3} \lceil n \cdot (2k+1)^{-1} \rceil \geqslant 2n + \sum_{k=1}^{\lceil n/2 \rceil} \lceil n \cdot (2k+1)^{-1} \rceil$$

$$\geqslant (n/2)(3 + \ln[n/2])$$

*Remark.* The proof follows a communication of Schönhage (1980).

*Corollary .* For a ca $\mathfrak{A}$ which solves the fssp in minimal time there holds

$$\Pi(\mathfrak{A}, n) \geqslant O(n^{-1} \cdot \ln n)$$

It can be shown that the sum of changes is not altered (in the order of magnitude) for solutions of the fssp which need a longer time. Seutter (1981) has shown that the minimal time solution of Waksman (1966) has a sum of $O(n^2)$ changes, and one of the solutions of Balzer (1967) which needs $<3n$ steps has a sum of $O(n \cdot \log n)$ changes.

## CONCLUSION

It is hoped that $\Pi$ reflects the parallelism inherent in problems and that it is therefore reasonable to use $\Pi$ to select from a collection of tasks those to be processed by array processors.

## ACKNOWLEDGMENTS

I am indebted to J. G. Pecht and to H. Szwerinski for helpful discussions.

## REFERENCES

Balzer, R. M. (1967). "An 8-State Minimal Time Solution to the Firing Squad Synchronization Problem," *Information and Control*, **10**, 22–42.

Hennie, F. C. (1965). "One-Tape, Off-Line Turing Machine Computations," *Information and Control*, **8**, 553–578.

Salomaa, A. (1973). *Formal Languages*. Academic Press, New York and London.

Schönhage, A. (1980). Personal communication.

Seutter, F. (1981). "Zustandsänderungszahlen bei speziellen Synchronisationsproblemen." Diploma Thesis, Technical University Braunschweig, Braunschweig.

Smith III, A. R. (1970). "Cellular Automata and Formal Languages, *Proceedings of Eleventh Annual IEEE Symposium on Switching and Automata Theory*, pp. 216–224.

Trachtenbrot, B. A. (1977). *Algorithmen und Rechenautomaten*. VEB Deutscher Verlag der Wissenschaften, Berlin.

Vollmar, R. (1979). *Algorithmen in Zellularautomaten*. Teubner, Stuttgart.

Vollmar, R. (1981). "On Cellular Automata with a Finite Number of State Changes, *Computing, Supplementum*, **3**,181–191.

Waksman, A. (1966). "An Optimum Solution to the Firing Squad Synchronization Problem," *Information and Control*, **9**, 66–78.