Control
and Cybernetics
VOL. 6 (1977) No. 2

Planar grammars, parallel picture processing algorithms and their equivalence *)

by

ZENON KULPA

Polish Academy of Sciences
Institute of Biocybernetics and Biomedical Engineering
Warsaw, Poland

The paper outlines a formal proof of the processing equivalence of planar grammars (originated by Kirsch for picture description) and simple, fully nondeterministic picture processing algorithms consisting of position — invariant (parallel) local picture processing operations.

On the way to prove the main equivalence theorem, some other theorems on interesting properties of planar grammars are proved, namely theorems on connected — rules grammars, single — symbol — changing grammars, and on grammars with rules applicable at most one place on any picture.

1. Introduction

The purpose of this paper is to outline a formal proof of the processing equivalence of planar grammars (originated by Kirsch [3] for picture description and used by Dacey [16, 17]) and simple, fully nondeterministic picture processing algorithms (in the sense of Blikle, Mazurkiewicz [2]) consisting of position-invariant (parallel) local picture processing operations (Narasimhan [7], Rosenfeld [8], Kulpa [4]).

The equivalence we are to prove has been intuitively recognized by picture processing men [10] and for string grammars there have been presented some related equivalence results (Rosenfeld [9]).

After the first, short version of this paper [5] was written, it was realized that Rosenfeld [13] had obtained closely related results for planar grammars. Nevertheless, our work and Rosenfeld's [13] differ in some important aspect: he had considered a grammar as a device for defining a set of planar words (a planar language) and had proved an equivalence of (normal) planar grammars to planar grammars which can apply their rules in parallel (and, for avoiding paradoxes, all their rules must rewrite only one symbol at a time), that equivalence being in the sense

*) This paper was originally published in the vol. 25, entilted "Planar Grammars", of the Institute for Organization, Management and Control Sciences Reports (Warsaw 1975).

of equality of corresponding planar languages. On the contrary, we consider a grammar as a device defining some picture processing rule (sequential operation on pictures) i.e. a relation in the set of all planar words, and we show an equivalence (in some, loosely speaking, "processing" sense) of such understood grammars to parallel (position invariant), nondeterministic algorithms in the sense of Blikle and Mazurkiewicz [2]. So, the Rosenfeld's [13] results can be considered as a special case of ours. Moreover, our approach to the problem seems to be more formal and precise.

Analysis of some problems which had arisen in the course of proving the part 2 of Theorem 2 as in the first version [5] of this paper showed the need for some changes in the formulations as well as produced some new problems (discussed briefly in the Section 6).

On the way to prove the main theorem we also state two theorems (on equivalence properties of planar grammars) which appear to be interesting results by themselves.

The style of introducing planar grammars will be similar to Bielik's [1], and of the algorithms — to that of Blikle, Mazurkiewicz [2]. For brevity, the proofs will not be given in details, but only sketched.

2. Preliminaries

Let us recall briefly a notation for algebra of relations (after Blikle, Mazurkiewicz [2]). Let X, Y, Z, W be sets, then $R \subseteq X \times Y$, $Q \subseteq Y \times Z$ are (binary) *relations*; $R \cdot Q$ (shortly: RQ) denotes a *composition of relations*; xRy means $(x, y) \in R$; WR denotes an *image* of W under R and RW is a *coimage* of W. Thus, RY is a *domain* and XR a *range* of R. The same notation will also be used for functions. R^* denotes a (reflexive and transitive) *closure* of R; for any set X, I_X denotes an *identity* or diagonal relation in X (i.e. $aI_X b$ iff $a=b$ and $a, b \in X$). Thus, $I_W R$ is a *left restriction* and RI_W a *right restriction* of R to W.

We will denote by J the set of integers, call the set $U=J\times J$ a *raster* and any $(i, j) \in U$ a *point*. Two points (a, b), (i, j) are *adjacent* iff $|a-i|+|b-j|=1$. A subset S of U is *connected* if any two its points can be connected by a chain of consecutively adjacent points lying entirely in S. A function $d_{ij}\colon U \to U$ such that $d_{ij}((x, y)) = (x+i, y+j)$ is called a *displacement*.

We denote by A a finite set called the *alphabet* and by T some subset of A called the *terminal alphabet*. Let $\# \notin A$ denotes a *blank symbol*, and let $A_{\#} = A \cup \{\#\}$.

Definition 1. An *abstract picture* (or planar word) over the alphabet A is any function $p\colon U \to A_{\#}$ such that pA is a bounded (i.e. finite in this case) set of points. The set of all abstract pictures over A we will denote by $\Pi(A)$, (or Π, if A is known).

A picture p is called *connected* if pA is connected. A *subpicture* is a function $I_N p$, where N is any bounded subset of U, and p is a picture. The set of all subpictures over A will be denoted by $\pi(A)$, (briefly: π). By $\pi_N(A)$ we denote the set of all subpictures over A with the domain N.

Let $\sigma \notin A$ denotes a so-called *starting symbol*, and let $A_\sigma = A \cup \{\sigma\}$, and $A_{\sigma\#} = A_\# \cup \{\sigma\}$.

For some $p \in \Pi(A)$ and $s \in A$, if $p \circ \{s\} \neq \emptyset$ we will speak that the symbol s *occurs in the picture* p.

If $s \in A$, then a function $h_s: \Pi(A) \to \Pi(A \setminus \{s\})$, such that $h_s(p)$ is a picture obtained from p by replacing all occurrences of s in p by $\#$, leaving other symbols unchanged, will be called the *s-erasing function*.

A *blank picture* is a picture β such that $U\beta = \{\#\}$, i.e. in β only blank symbols occur.

3. Planar grammars

DEFINITION 2. A generalized *planar grammar* is a system $G = (A_\sigma, T, R)$, where A, T: alphabets (as defined above), $R \subseteq \pi(A_\sigma) \times \pi(A_\sigma)$ a finite set of *rules*, such that for every $(u, v) \in R$, $uA_{\sigma\#} = vA_{\sigma\#}$ and $Uu \neq \{\#\}$ (i.e. the rule never creates anything on the completely empty place).

DEFINITION 3. For some $x, y \in \Pi(A_\sigma)$ we say that x *immediately produces* y (written $x \underset{G}{\Rightarrow} y$ or simply xGy) in G iff there exists a rule $r = (u, v) \in R$ and a displacement d_{ij} such that:

(1) $I_Q x = d_{ij} u$ and $I_Q y = d_{ij} v$,

(2) $I_{U \setminus Q} x = I_{U \setminus Q} y$, where $Q = d_{ij} uA_{\sigma\#} = d_{ij} vA_{\sigma\#}$.

That is, xGy if y can be obtained from x by replacement of some subpicture of x which is identical to (appropriately displaced) left side of the rule r, by the (appropriately displaced) right side of r, leaving the rest of x unchanged. A *starting picture* is a picture $p \in \Pi(A_\sigma)$ such that card $(p \circ \{\sigma\}) = 1$, i.e. in which there is exactly one occurrence of the strating symbol σ.

Let $\Sigma(A)$ denotes the set of all starting pictures over A. Note that $h_\sigma(\Sigma(A)) = \Pi(A)$ and although h_σ^{-1} is not a function, it is a relation; $h_\sigma^{-1} \subseteq \Pi(A) \times \Pi(A_\sigma)$.

DEFINITION 4. The *resulting relation* of the grmmar G is a relation $R_G = h_\sigma^{-1} I_{\Sigma(A)} G^* I_{\Pi(T)} \subseteq \Pi(A) \times \Pi(T)$, i.e. $xR_G y$ if there exists a picture $x' \in \Sigma(A)$ such that $h_\sigma(x') = x$ (i.e. x' is a picture x "augmented" by some single occurrence of the starting symbol σ) and $x' G^* y$, and y is a picture over the terminal alphabet T.

Note that the concept of the starting symbol σ and that of the starting picture brings near the processing and language-defining aspects of grammars. It also assures the validity of the part 2 of Theorem 2 below — without such a "single-point-marking" facility (which is given by the very definition in the language-defining formulation, see below and e.g. Milgram, Rosenfeld [6]) there exists no mechanism to ensure, for arbitrary pictures, the applicability of all rules at only one place (see also Sec. 6). But from practical point of view (simlarly as for the use of Theorem 1), it is not a problem: any grammar in the sense of [5] can be made like that in this paper, with the same resulting relation (in both senses!) if we augment it by the single

σ-erasing rule $\boxed{\sigma} \to \boxed{\#}$ (in more formal, but somewhat cumbersome form: $(\{((0,0),\sigma)\}, \{((0,0),\#)\})$, provided it is the only rule involving the use of σ (but see the proof of Theorem 2).

DEFINITION 5. Two grammars G and H are called *W-equivalent*, where W — some alphabet, iff $I_{\Pi(W)}\, R_G = I_{\Pi(W)}\, R_H$.

DEFINITION 6. A *planar language* defined by G is a set $L_G = \{\beta\}\, R_G$, i.e. a set of pictures produced by the grammar from the set of starting pictures having only one occurrence of a non-blank symbol, namely σ.

The planar language has therefore some similarity to an "impulse response" of the sequential "picture filter" defined by the grammar. As an impulse serves there the single σ symbol occurring on the otherwise blank background. Compare it with analogous interpretation of position-invariant linear operations (e.g. Rosenfeld [8], sec. 4.2—4.4).

As necessary explanation we want to point out here that although the above-discussed "σ-trick" allows us not distinguish formally between "language defining" and "picture processing" aspects of planar grammars, in practice this distinction has to be observed. For some "processing" grammar, e.g. extracting contour and inside points of black-white pictures, hardly defines any language. It can be fit into our format by augmenting it by the σ-erasing rule as above, but such a grammar defines the language consisting of only single blank picture. Thus investigation of its language-defining capabilities tells nothing about its useful application as a contour extractor. On the other side, the grammars like Kirsch'a right triangle grammar [1, 3] defining some classes of pictures, hardly can be used directly for some practically useful picture processing task (e.g. for recognition of triangles in the pictures), although their "resulting relations" and equivalent to them parallel picture processing algorithm can be constructed (see below).

The above formulations are, in a sense, noneffective, namely there is no effective way to check if some rule is in fact applicable to some arbitraty picture (because of infiniteness of the raster). It can be overcome if one requires the pictures to be connected or provides some endmarkers on the raster. Let us consider the first possibility.

Let us introduce a new, "visible-blank" symbol $\sqcup$, let $\sqcup \in A$ and denote by $T_{\sqcup}$ the new terminal alphabet $T_{\sqcup} = T \cup \{\sqcup\}$.

THEOREM 1. For every planar grammar $G = (A_\sigma, T, R)$ there exists a grammar $G' = (A_\sigma, T_{\sqcup}, R')$ such that:

(1) $R_G = R_{G'}\, h_{\sqcup}$ and

(2) $(\forall\, (u, v) \in R')\, (uA_{\sigma\#}) = vA_{\sigma\#}$ is connected) and

(3) if $X \in \Pi$ is connected, so is every y such that $xG'^{*}\, y$, i.e. G' is equivalent to G with respect to erasing of visible-blank symbol (1) and has rules connected (2) and all pictures produced in the course of derivation in G' are connected, provided the starting one was (3).

Proof. For every rule $(u, v) \in R$ change $\#$'s occurring in v to $\sqcup$'s. Then replace it by a set of rules by changing some (or all or none) of $\#$'s occurring in u to $\sqcup$'s in all possible combinations. Then make rules connected, replacing every one by a set of rules resulting by connecting its (possibly) disconnected parts by connected chains with all possible assignment of symbols from $A_{\sigma\#} \cup T_{\sqcup}$ in the left and, correspondingly, $A_\sigma \cup T_{\sqcup}$ in the right side (to the $\#$ symbol in the left corresponds the $\sqcup$ symbol in the right). The set R' results. Q.E.D.

So we will restrict ourselves from now on only to connected pictures and rules, without loss of (at least practical) generality. Although not explicitly stated, the theorem has also been assumed by Milgram, Rosenfeld [6].

In Rosenfeld [(13), Th. 2) a similar result concerning "language defining" aspect of planar grammars was obtained. Namely it was proved that for any planar grammar generating some language of (not necessarily connected) terminal pictures there exist a planar grammar generating a language containing only connected terminal pictures, that language constituting exactly the set of all connected components of the pictures generated by the first grammar.

4. Parallel picture processing algorithms

A *picture processing operation* (p.p.o.) is a function $\varphi: \Pi \to \Pi$. A p.p.o. $s_{ij} = \{(S, d_{ij}\, S) | S \in \Pi\} \subseteq \Pi \times \Pi$, where d_{ij}—displacement, is called a *shifting operation*.

DEFINITION 7. A *position-invariant* operation is a p.p.o. ψ such that for every shifting operation s_{ij} holds $\psi s_{ij} = s_{ij}\, \psi$.

DEFINITION 8. A (*parallel*) *local operation* is a p.p.o. λ such that there exists a bounded subset $N \subseteq U$ and the function:

$f_\lambda: \pi_N(A) \to A_\#$ such that for every $S \in \Pi$:

$\lambda(S) = \{((i,j), f_\lambda(I_N\, d_{ij}\, S)) | (i,j) \in U\}$.

The set of all parallel local operations will be denoted by $\Lambda(\Pi)$.

It is easy to see that every local operation λ is fully characterized by the function f_λ. Also every local operation is position-invariant (but not the converse). Such an operation can be performed in parallel, i.e. one can perform the computation prescribed by f_λ for every point simultaneously. Wothout loss of generality we may also assume N to be connected and including the point $(0, 0)$.

DEFINITION 9. A *simple, fully nondeterministic parallel picture processing algorithm* is a finite-control Mazurkiewicz's algorithm $\mathfrak{A} = (\Pi(A), \{a\}, a, L)$ (Blikle, Mazurkiewicz [2]) such that its set of instructions L looks as follows:

$$L = \{(\{(a, a)\}, \lambda_i) | \lambda_i \in \lambda_{\mathfrak{A}} \subseteq \Lambda(\Pi)\} \cup \{(\{(a, \varepsilon)\}, I_{\Pi(T)})\}.$$

Thus, instructions of the algorithm have the same initial and terminal label a and action consisting of some parallel local operation λ_i, except for one (terminating) instruction allowing the algorithm to stop "by end" if it produces some terminal picture.

Such an algorithm is fully characterized by the set $\lambda_{\mathfrak{A}}$ of parallel local p.p.o-s occurring in its instruction set L. Its flow of control is completely degenerated — at every step of the algorithm any applicable instruction from L is allowed to be performed (thus it works fully nondeterministically), and stopping is allowed at any moment when some terminal picture is generated.
Then the control flow of $\mathfrak{A}$ is practically the same as for the control of rule's application in a planar grammar.

The resulting relation of the algorithm can be obtained directly for any such algorithm $\mathfrak{A}$ from the formula (cf. [2]):

$$R_{\mathfrak{A}} = \Lambda_{\mathfrak{A}}^{*} \, I_{\Pi(T)}, \text{ where } \Lambda_{\mathfrak{A}} = \cup \lambda \mathfrak{A}.$$

Similarly as for grammars, we introduce here a σ-trick, allowing us to simplify the proof of Theorem 3. We do it by adding to the instruction set L of $\mathfrak{A}$ an instruction $l_\sigma = (\{(a, a)\}, h_\sigma)$, the σ-erasing function h_σ is a local p.p.o. with $f_{h_\sigma} = \{(\{((0, 0), \sigma)\}, \#)\}$; and writing the resulting relation as:

$$R'_{\mathfrak{A}'} = h_\sigma^{-1} \, I_{\Sigma(A)} \, R_{\mathfrak{A}'},$$

where $\mathfrak{A}'$ — the algorithm $\mathfrak{A}$ with L augmented by the σ-erasing instruction. It is evident, that $R'_{\mathfrak{A}'} = R_{\mathfrak{A}}$ for any $\mathfrak{A}$ not using the symbol σ in its instruction set, so that practically this trick can be considered to be the mere technicality. Some its theoretical consequences will be nevertheless briefly discussed in the Sec. 6 below.

5. Equivalence proof

As it is clearly seen from the above, the only important differences between the rules of planar grammar (Sec. 3) and λ-action of parallel picture processing algorithm (Sec. 4) lie in the points:

(1) the rule can change several symbols at the point of its applicability, while the f_λ-function — only one;

(2) the rule applies at a given moment at one place (although may be applicable at several places); λ changes simultaneously all points that are to be changed (i.e. acts in parallel).

But note that any f_λ-function $f_\lambda \colon \pi_N(A) \to A_{\#}$, $(0, 0) \in N \subseteq U$, can be considered, according to its effect on the pictures, as a (finite!) set of rules:

$$r_\lambda = \{(u, v) \mid u, v \in \pi_N(A) \;\&\; v(0, 0) = f_\lambda(u) \neq u(0, 0) \;\&$$
$$\& \, (\forall (i, j) \in N) \, (i, j) \neq (0, 0) \Rightarrow u(i, j) = v(i, j)\},$$

i.e. rules with left parts being all the subpictures in the domain of f_λ for which f_λ really changes something in the picture $(f_\lambda(u) \neq u(0, 0))$, and right parts being equal to the left parts except for the symbol at the $(0, 0)$ point, which has to be equal to the value of f_λ for the rule's left part $(f_\lambda(u)=v(0,0))$. Therefore rules in r_λ all change only one symbol.

And vice versa, any rule $r=(u, v)$, $u, v \in \pi_N(A)$, which changes only one symbol, i.e. for which there exists a point $(i, j) \in N$ such that $u(i,j) \neq v(i,j)$ and $(\forall (m,n) \in \in N)\ (m,n) \neq (i,j) \Rightarrow u(m,n)=v(m,n)$, can be considered as a function:

$$f_r=\{(d_{-i,-j}\, u, v(i,j))\} \cup \{(d_{-i,-j}\, p, p(i,j)) \mid u \neq p \in \pi_N(A)\},$$

i.e. f_r domain includes all subpictures defined on the domain of the rule parts, but shifted as to place the symbol changed by r at the point (0, 0), and its values are respectively:

(a) the symbol changed by r for that single subpicture being the shifted rule's left part;

(b) the symbol occurring at the point (0, 0) of the (shifted) other subpictures in the domain of f_r.

That is, f_r changes something in the picture only at the place where appropriately shifted left part of the rule r occurs.

Therefore, the first step on our way to prove the equivalence will be:

THEOREM 2. For every planar grammar $G=(A_\sigma, T, R)$ there exists anA-equivalent grammar $G'=(A'_\sigma, T, R')$ such that, for every rule $q=(u, v) \in R'$:

(1) $(\exists!\,(m,n) \in Q)\ u(m,n) \neq v(m,n)$, i.e. a rule changes only one symbol;

(2) if $x \underset{q}{\Rightarrow} y$ then $(\exists!\, d_{ij})\ I_Q\, x = d_{ij}\, u$, i.e. a rule can be applied at most one place;

where $Q=d_{ij}\, uA'_{\#}$, and $\exists!$ quantifier means "there exists exactly one ...".

Proof. For (1) replace all rules changing more than one symbol by a sequence of one-sambol-changing rules having the same domain (as their originals) and introducing a set of new special "binding" nonterminals to assure blocking of the derivation if the new rules are not applied in the proper sequence as to give the result equivalent to an application of the single old rule. It is a technique analogical to that developed in the proof of the "two-point-rules" theorem by Bielik [1, 12]. Use also analogical argument as in Bielik [1, 12] to show the A-equivalence of such transformed grammar to the original one. The Rosenfeld's ([9,] p. 289) technique for asserting analogical fact for string grammars is not applicable to two dimensions.

For (2), construct the new grammar adding as new nonterminals the new symbol ε and the "primed" original symbols (i.e. all symbols from $A\# \cup \{\varepsilon\}$). Then modify all rules to have exactly one symbol at the left and exactly one at the right hand side replaced by its primed counterpart, and replace all occurrences of σ in them by the new symbol ε. Introduce a set of new two-point rules (not involving the σ symbol) only "moving" a prime over the picture, add a rule $\boxed{\sigma} \to \boxed{\varepsilon'}$ and, for all $t \in T$, a rule $\boxed{t'} \to \boxed{t}$. All the above assures that all the new rules are applicable exactly at one place in the picture (the rule $\boxed{\sigma} \to \boxed{\varepsilon'}$ due to the existence of only one σ-symbol at the start of derivation, and other rules — due to occurrence of at most one primed symbol in the picture at any given moment of derivation). The resulting grammar is evidently A-equivalent to the original one. The single-point-marking facility provided by the σ-trick of Definition 4 is there absolutely necessary to produce the single "prime" in the picture, serving to mark the place where a rule can be applied. This method is equivalent to Rosenfeld's ([9], Th. 5).

To combine the properties (1) and (2) in the single grammar (A-equivalent to G) is also evidently possible. For, in the proof of (1), newly introduced rules are by the very construction applicable at only one place (where their predecessors in the old-rule-modelling sequence were applied, writing down the "binding" symbols). Thus, following the construction of (2) by that of (1) does not affect the only-one-place applicability of the rules constructed for (2). Q.E.D.

THEOREM 3. (the equivalence). For any planar grammar $G=(A_\sigma, T, R)$ there exists a simple fully nondeterministic parallel picture algorithm $\mathfrak{U}=(\Pi(A'), \{a\}, a, L)$ such that it is A-equivalent to G, and conversely — for any such an algorithm there exists an A'-equivalent planar grammar G.

Proof. The first part directly follows from the properties of r_λ and f_r constructs discussed before and by the use of Theorem 2. Indeed, to construct an algorithm A-equivalent to the given grammar it suffices to construct firstly, using Theorem 2, an A-equivalent (to the given one) grammar with one-symbol-changing rules applicable at one place each. Such rules can be directly modelled by the f_λ-functions (see the construction of f_r) and, due to the one-place applicability of the rules, parallel application of corresponding f_r-functions (in appropriate λ-operations) has the same effect as the application of the rules themselves. Therefore the algorithms with such constructed λ-operations as actions of its instructions will have the same resulting relation (with respect to pictures over the alphabet A).

The proof of the second part is less immediate. It requires one to construct a planar grammar which in effect applies the f_λ-functions of $\mathfrak{A}$ "in parallel" (Rosenfeld [9]), by modelling with appropriate rules, for every λ_i of $\mathfrak{U}$, a "square spiral" extending scan of the current picture, marking the places where f_{λ_i} changes a symbol, and after recognizing that the whole picture has been scanned (the use of Theorem 1 is indispensable) — scanning backwards, changing symbols at marked points and ending at the beginning of the scan with a condition (a special symbol) allowing to start a scan for another λ_i.

The whole scanning process starts at the beginning from the single σ symbol. The rules modelling the f_λ-functions of the algorithm are directly obtainable by the construction of r_λ above. The whole set of rules modelling the scan of the picture constitutes a so-called "automaton imitator" introduced by Milgram, Rosenfeld [6] and also used in Rosenfeld [13]. Q.E.D.

Note that the construction of the parallel algorithm equivalent to some planar grammar as proposed in the proof above, goes through assuring firstly applicability of grammar rules at one place only, so for every such a rule a simple parallel operation can be constructed, which then also changes in effect only one point in the picture. It is counter-practical, because the most important practical advantage of parallel algorithms over serially operating grammars lies exactly in the ability of the operations of the algorithms to make changes at several raster points simultaneously. Of course, for the theoretical purposes it is no trouble; such a construction allows to prove the theorem in a rather simple way. Some problems which nevertheless arise from that aspiring at simplicity are discussed in the next section.

6. Some open problems

The σ-trick introduced in the paper simplifies the argument, but has some not completely trivial consequences. Indeed, note firstly that without it, the part 2 of Theorem 2 is no longer valid — there is no possibility to produce exactly one "primed" symbol on the picture. Nevertheless, it is possible to realize by appropriate grammar rules the operation $h_\sigma^{-1} I_{\Sigma(A)}$; in words: "mark exactly one blank point in the picture with the symbol σ". For in every (non-blank) picture several well-defined unique points exist, e.g. the leftmost non-blank point in the uppermost non-blank row of the non-blank component of the picture (remember that using Theorem 1, we consider only connected pictures, i.e. having only one non-blank component). So it is possible to model by appropriate rules the automaton imitator [6 13] finding such a point and marking the single blank point in its neighbourhood.

The above looks like a contradiction, but really it is not — for the automaton imitator mechanism for doing the marking *must* use rules *which cannot be guaranteed to apply in exactly one place* for an arbitrary picture. Then there exists one more problem: there is no possibility to start exactly one automaton imitator searching the unique point to mark. In general, several such imitators will work concurrently on the picture. They must be so organized as to not interfere with each other (it is generally impossible, for the arbitrary number of such automata will require infinitely many new symbols in the alphabet) or so constructed as to destroy each other except for the only one surviving (seems possible).

Similar remarks apply also to algorithms. The σ-trick introduced for them looks, at the first view at least, giving them an additional computing power, because the marking operation $h_\sigma^{-1} I_{\Sigma(A)}$ is evidently not local. But note that the class of operations on pictures defined by the algorithms discussed is wider than the class of local p.p.o.-s, as it can be shown by a relatively simple example.

Then it is conjectured that there exists an $\mathfrak{U}$-algorithm (without σ-trick introduced) which realizes the single-point-marking operation $h_\sigma^{-1} I_{\Sigma(A)}$. The argument can proceed similarly as for grammars above, by construction of appropriate automaton imitator built from algorithm instructions, and the problem of several parallely "growing" imitators seems even easier to resolve here, due to uniform grow of them.

The above serves to argument that the σ-trick is really a technicality. But an alternative idea comes into one's mind, namely to prove directly the equivalence of grammars and algorithms, both without the σ-trick introduced. Such a proof evidently cannot use the part 2 of Theorem 2 as its step, due to its invalidity in this case. It seems to be possible, although the proof will be more complicated and, probably, will nevertheless have to use the above-discussed single-marking automation imitators. To end the reasoning, let us "officially" state:

Conjecture. Generalized planar grammars and simple, fully nondeterministic parallel picture processing algorithms, both of the form not using the σ-trick (or any its equivalent), are also equivalent with respect to their picture processing power.

Practically however, the result of Theorem 3 seems to be equally sufficient.

The distinct problem is to prove the equivalence of the above-defined fully nondeterministic algorithms and general picture-processing ones (in the sense of [2]). The problem seems to be solvable with no principal troubles. With the equivalence just proved, it brings us to the area of "programmed picture grammars" investigated by some authors, e.g. Swain, Fu [15].

7. Conclusions

The above results can encourage some adherents of linguistic methods in picture recognition ("all the processing can be made grammatically!"), but it also is an argument for their opponents ("the algorithms do it better than your intricate and unnatural grammars!") — see [10]. The author, although promotes the linguistic (or structural) trend in pattern recognition, in this case inclines to the second opinion. If grammars can be of any use to describing or processing pictures on the raster level, another apparatus, e.g. that by Siromoney et al. [11] looks better suited here than the above-described Kirsch's planar grammars are.

The author argues that also the notion of programmed grammars (like that of [15]) can be more naturally and effectively investigated in the framework of algorithms rather than grammars.

From another point of view, the results obtained can be considered as a generalized version of the equivalence of sequential and parallel picture processing operations (see Rosenfeld, Pfaltz [14]).

It looks, however, that the area of essentially sequential operations on pictures needs more practically adequate paradigm than that of planar grammars or that proposed in [14]. An encouragement to search for such a paradigm is, may be, the main positive result of this paper.

Acknowledgments

The author greatly acknowledges prof. A. Rosenfeld for pointing him out the existence of the paper [13] and its similarity to his results. Also thanks are due to Mr. A. Bielik for very stimulating discussions on the subject of the paper.

References

1. BIELIK A.: Sterowana generacja słów języków planarnych. (Controlled generation of planar languages words, in Polish). M. Sc. Thesis supervised by Z. Kulpa. Warsaw 1973 (unpublished).
2. BLIKLE A., Mazurkiewicz A.: An algebraic approach to the theory of programs, algorithms, languages and recursiveness. Proc. of Symposium and Summer School "Math. Found. of Computer Science" (MFCS-72), Warsaw 1972.
3. KIRSCH R. A.: Computer interpretation of English text and picture patterns. *IEEE Trans. on Electron. Comp.* **EC-13**, 4 (1964).
4. KULPA Z.: A picture processing system PICTURE ALGOL 1204. Proc. 7th Yugoslav Internat. Symp. FCIP 72, Bled 1972.
5. KULPA Z.: On the equivalence of planar grammars and parallel picture processing algorithms. Proc. Symp. and Summer School "Math. Found. of Computer Science" (MFCS-74), Jadwisin (Poland) 1974, published as: Lecture notes in computer science, vol. 28. Berlin 1975, Springer-Verlag.
6. MILGRAM D. L., Rosenfeld A.: Array automata and array grammars. Information Processing 71 (IFIP 71 Congress Proc.) Amsterdam 1972 North Holland.
7. NARASIMHAN R.: Labeling schamata and syntactic description of pictures. *Inform. a. Control* 7 (1964) 151—179.
8. ROSENFELD A.: Picture processing by computer. New York 1969, Acad. Press.
9. ROSENFELD A.: Isotonic grammars, parallel grammars, and picture grammars. In: Machine Intelligence 6. Eds. B. Meltzer, D. Michie. Edinburgh 1971, Edinburgh Univ. Press.
10. Panel discussion on "Are picture grammars of any use in scene analysis?" In: Graphic languages. Eds. F. Nake, A. ROSENFELD. Amsterdam 1972, North Holland.
11. SIROMONEY G., SIROMONEY R., KRITHIVASAN K.: Picture languages with array rewriting rules. *Inform. a. Control* **22** (1973).
12. BIELIK A.: A theorem of the two-point-rules planar grammars. In: Planar grammars. Repts. of the Inst. of Organization Management and Control Sciences, series B, vol. 25, Warsaw 1975.
13. ROSENFELD A.: Array grammars normal forms. *Inform. a. Control* **23** (1973) 173—182.
14. ROSENFELD A., Pfaltz J. L.: Sequential operations in digital picture processing. *J.ACM* (1966) 471—494.
15. SWAIN P. H., FU K. S.: Stochastic programmed grammars for syntactic pattern recognition. *Pat. Recog.* (1972) 83—100.
16. DACEY M. F.: The syntax of a triangle and some other figures. *Pat. Recog.* 1 (1970).
17. DACEY M. F.: Poly- a two-dimensional language for a class of polygons. *Pat. Recog.* 2 (1971).

Received, December 1976.

Gramatyki planarne, równoległe algorytmy przetwarzania obrazów i ich równoważność

Naszkicowano formalny dowód równoważności (w sensie algorytmicznym) tzw. gramatyk planarnych (wprowadzonych przez Kirscha dla opisu obrazów wizualnych) i prostych, całkowicie niedeterministycznych algorytmów przetwarzania obrazów, algorytmów składających się z tzw. lokalnych, niezależnych od położenia (równoległych), operacji na obrazach.

W toku konstruowania dowodu głównego twierdzenia o równoważności udowodniono także kilka innych twierdzeń o pewnych ciekawych własnościach gramatyk planarnych, a mianowicie: twierdzenia o gramatykach z regułami spójnymi, o gramatykach zmieniających każdą regułą tylko jeden symbol obrazu i o gramatykach, których reguły są stosowalne w co najwyżej jednym miejscu obrazu.

Планарные грамматики. параллельные алгоритмы преобразования изображений и их эквивалентность

В статье представлено формальное доказательство эквивалентности (в алгоритмическом смысле) так называемых планарных грамматик (введеных Киршем для описания визуальных изображений) и простых вполне недетерминистических алгоритмов преобразования изображений. Эти алгоритмы состоят из так называемых покальных, независимых от положения (параллельных) операции на изображениях.

В ходе построения доказательства основной теоремы об эквивалентности доказаны некоторые другие теоремы об интересных свойствах планарных грамматик, а именно — теоремы о грамматиках со связными правилами, о грамматиках изменяющих каждым правилом только один символ изображения и о грамматиках правила которых могут применяться не больше чем в одном месте изображения.