Prof. Dr. Jürgen Dassow
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik

G R A M M A T I C A L

P I C T U R E

G E N E R A T I O N

Manuscript

Magdeburg, April 2011 – July 2011

Contents

Chapter 4

Collage Grammars

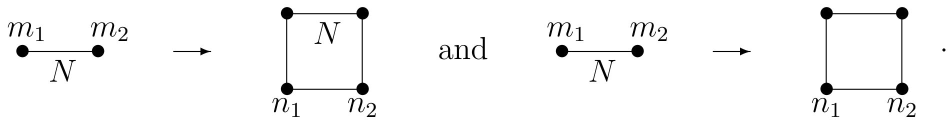
The two type of picture generating grammars, chain code picture grammar and Siromoney matrix grammar, have in common, that the grammar generate an object, a word or a matrix, and the object is transformed into a picture. In this chapter we consider collage grammars which work on pictures itself, i.e., we consider a special type of enriched pictures, where the enrichment allows the adding of pictures into pictures. This idea is a generalization of graph grammars where the derivation process consists in the replacement of edges/hyperedges by graphs. We illustrate the graph grammars by some examples without going into the detail, i.e., we omit formal definitions.

In this chapter, the nodes of a graph will be represented by a bullet *bullet* associated with the label/name of the node in the neighbourhood of the bullet, an edge (x, y) is given by a line connecting the bullets associated with x and y, and if an edge is labelled, the label is given above or below the line.

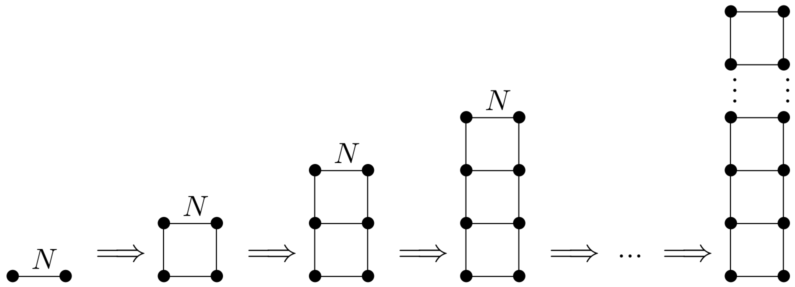
We start with graph grammars where an edge is replaced by a graph. Obviously, we have to know which edge has to be replaced by a graph according to a rule, however, it is also necessary to add some information, how the graph which replaces the edge is embedded in the graph where the replacement occurs. If we only replace an edge, we need two nodes of the graph which replaces the edge which are identified with the nodes of the replaced edge.

Thus the rules are of the following form: the left hand side is an edge e labelled by a nonterminal N and two nodes m_1 and m_1, and the right hand side is a graph $G = (V, E)$ with unlabelled edges and/or edges labelled by a nonterminal and two distinguished nodes n_1 and n_2. A derivation step from a graph $H_1 = (V_1, E_1)$ to a graph $H_2 = (V_2, E_2)$ consists in the cancellation of the edge e in a graph H_1, a renaming of all nodes of V such that the nodes of V and the nodes of H_1 have no name in common, n'_1 and n'_2 are the names of the distinguished nodes of G, and an identification of m_1 with n'_1 and of m_2 with n'_2, i.e., $V_2 = (V_1 \setminus \{m_1, m_2\}) \cup V$ and E_2 is the set of edges $(x, y) \in E_1$ with $x, y \in V_1 \setminus \{m_1, m_2\}$, (n'_i, y) with $(m_i, y) \in E_1$, $y \in V_1 \setminus \{m_1, m_2\}$ and $1 \leq i \leq 2$ (which exist already in H_1) and all edges of E_2. The language generated by a graph grammar consists of all graph with only unlabelled edges which can be obtained from a given initial graph by some derivation steps.

Let us consider the graph grammar given by the initial graph H which consists of an edge labelled by N and the two rules



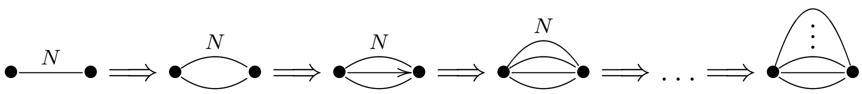
Omitting the labels/names of the nodes, we get the following derivation



which is unique up to the length. Thus the generated graph language consists of towers of squares of a arbitrary height ≥ 1.

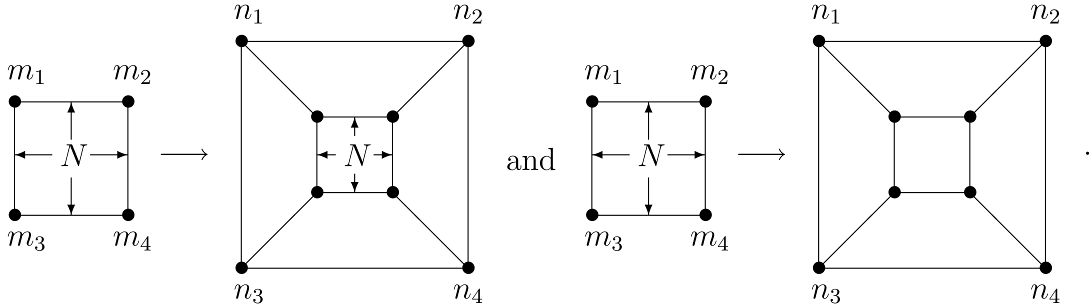
Our next examples also starts with the same initial graph and has the rules

We get the derivation

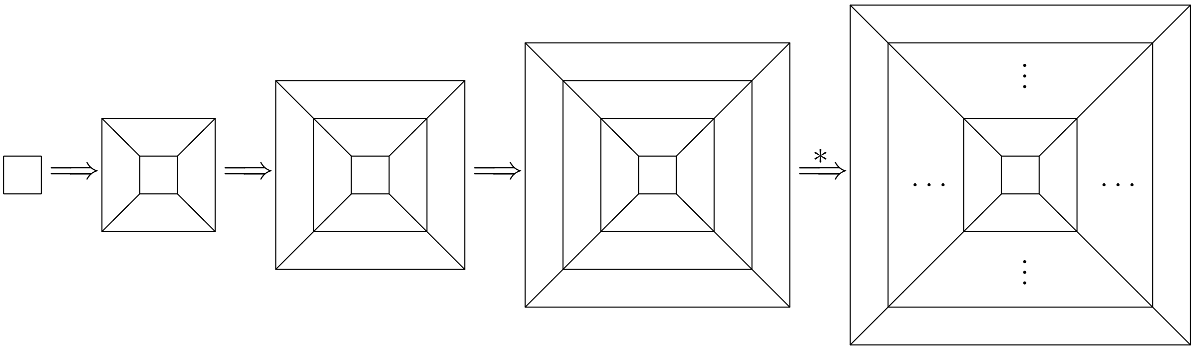


and hence the language consisting of all graphs with two nodes and an n-fold edge, where $n \geq 2$.

By the above explanation the application of a rule consists in a deletion of one edge and the adding of right hand side of a rule which is connected with the current graph only by two nodes. In order to get more general situations we extend the concept to hyperedge replacement, i.e., a hyperedge which is a set of edges is cancelled and the right hand side is connected via all nodes of the hyperedge. In the sequel, in drawings we "connect" the label of the hyperedge with the edges belonging to the hyperedge by a small arrow. As an example we consider the the following rules



Starting from the graph on the left hand side of the rules we get derivations of the form



4.1 Collage Grammars

We come to the definition of the basic concept of this chapter. We discuss only the two-dimensional case, but it is easy to extend the notion to higher dimensions.

The set of finite sequences of pairs real numbers or in geometrical terms of points of the $\mathbb{R}^2$ is denoted by $(\mathbb{R}^2)^*$.

The process of replacement according to a rule in a graph grammar uses the identification of the nodes of a hyperedge with the distinguished nodes of the replacing graph. Since we now consider geometrical objects in $\mathbb{R}^2$, additionally, we have to take into consideration the size of the objects and their position in the plane. Thus the hyperedges have to be positioned in the plane by their nodes which are points, and the object used for replacement has to have some distinguished points which can be identified with the nodes of the hyperedges. Therefore, the identification requires a mapping which maps the points of the hyperedge to the distinguished points. We shall restrict the set of possible mappings to the set of affine two-dimensional mappings.

Definition 4.1 *A* decorated collage *over a set Lab of labels is a quintuple*

$$C = (part, pin, E, att, lab),$$

where

- *part is a finite set of geometrical objects in* $\mathbb{R}^2$,
- $pin \in (\mathbb{R}^2)^*$ *and* $|pin| \geq 2$,
- E *is a set of hyperedges,*
- *att is a mapping which maps each* $e \in E$ *to an element* $att(e) \in (\mathbb{R}^2)^*$ *of length at least 2, and*
- *lab is a mapping which maps each* $e \in E$ *to an element* $lab(e) \in Lab$.

A collage *is a decorated collage with an empty set of hyperedges, i. e., it has the form* $(part, pin, \emptyset, \emptyset, \emptyset)$. *A collage is mostly only denoted by* $(part, pin)$.

For a decorated collage $C = (part, pin, E, att, lab)$ *the* picture *of* C, *denoted by* $pic(C)$, *is the set part.*

We omit a formal definition of a "geometrical object". In this paper we allow finite parts of curves, lines, polygons consisting of its borderlines only or polygons consisting of all their inner points (including the borderlines), etc. For other purposes other objects can be considered.

The letters of *pin* are called the pin points of C. They are the distinguished points of C which are identified if C replaces a hyperedge. The elements of $att(e)$, $e \in E$, are called attachment points and give the points which are used for identification, if e is replaced. The mapping lab gives the label of the hyperedges. We note that, by definition, there are no real edges which build the hyperedges; a hyperedge consists of its attachment points only (or - as drawn in examples - by edges from the label to the attachment points).

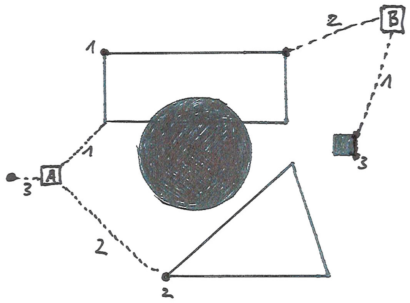


Figure 4.1: A decorated collage

An example of a decorated collage is given in Figure 4.1. Its set *part* consists of a complete circle, i. e., the points of the circle line and all its interior points, a complete small square, a triangle (without interior points), and a rectangle. The sequence *pin* consists of three points, denoted by 1 (a corner of the rectangle), 2 (a corner of the triangle), and 3 (a corner of the square). The collage contains two hyperedges e_1 and e_2 labelled by A and B, respectively. The mapping *att* is given by broken lines from A to the three points of $att(e_1)$ and from B to the two points of $att(e_2)$.

In order to define the derivation step, we first introduce rules for collages and then the application of such rules.

Definition 4.2 *A* context-free rule for collages *over a set Lab of labels is a pair* (A, C) *where* $N \in Lab$ *and* C *is a decorated collage over Lab.*

As usual we write $A \to C$ instead of (A, C).

Definition 4.3 *Let* $C_1 = (part_1, pin_1, E_1, att_1, lab_1)$ *and* $C_2 = (part_2, pin_2, E_2, att_2, lab_2)$ *be two decorated collages. Moreover, let e be an element of* E_1 *with* $lab(e) = A$*, and let* $A \to C$ *with* $C = (part, pin, E, att, lab)$ *be a context-free rule for collages. Then we say that* C_2 *is obtained or derived from* C_1 *by application of* $A \to C$*, written as* $C_1 \Longrightarrow C_2$*, if the following conditions hold:*
— there is an affine mapping t such that $t(pin) = att_1(e)$*,*
— $part_2 = part_1 \cup t(part)$*,*
— $pin_2 = pin_1$*,*
— E_2 *is the disjoint union of* $E_1 \setminus \{e\}$ *and* E*,*
— $att_2(e') = att_1(e')$ *for* $e' \in E_1 \setminus \{e\}$ *and* $att_2(e') = t(att(e'))$ *for* $e' \in E$*,*
— $lab_2(e') = lab_1(e')$ *for* $e' \in E_1 \setminus \{e\}$ *and* $lab_2(e') = lab(e')$ *for* $e' \in E$*.*

Intuitively, the application of a rule $A \to C$ with $C = (part, pin, E, att, lab)$ replaces a hyperedge e labelled by A in a decorated collage C_1 by the image of $part$ under an affine transformation which maps the pin point of C to the attachment points of the hyperedge. Moreover, the attachment points of the resulting decorated collage C_2 are those from hyperedges of C_1 which are different from e and the images of the attachment points of C under the affine transformation.

Definition 4.4 *i) A* context-free collage grammar *is a triple $G = (Lab, P, Z)$ where Lab is a set of labels, P is a finite set of context-free rules for collages, and Z is an initial decorated collage.*

ii) The collage language generated by a context-free collage grammar G *is defined as*

$$L_{col}(G) = \{pic(C) \mid Z \Longrightarrow^* C, C \text{ is a collage}\}$$

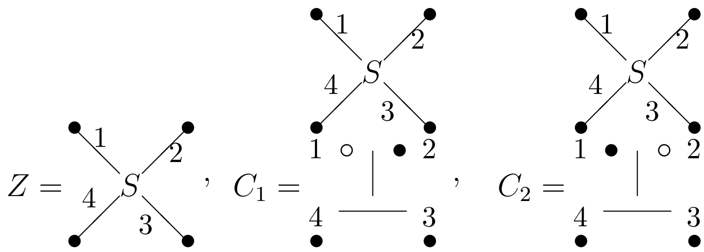
where $\Longrightarrow^*$ denotes the reflexive and transitive closure of $\Longrightarrow$.

We present some examples.

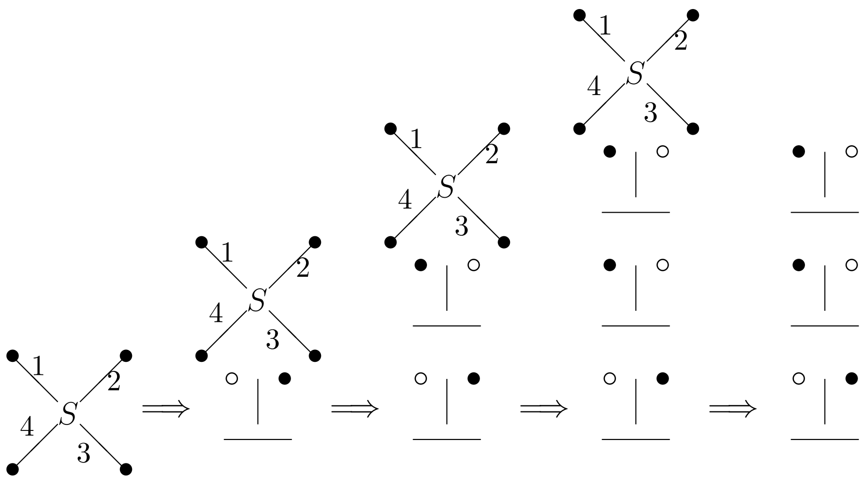
Example 4.5 *We consider the context-free collage grammar*

$$G_1 = (\{S\}, \{S \to C_1, S \to C_2, S \to (\emptyset, \emptyset, \emptyset, \emptyset, \emptyset\}, Z)$$

with

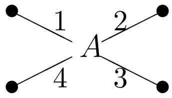


(the initial decorated collage is a hyperedge where its attachment points form a square; the right hand sides of the first two rules are a "face" with an open eye and a closed eye, which differ by the positions of the closed and open eyes, the right hand side of the third rule is the empty collage). A typical derivation is given by

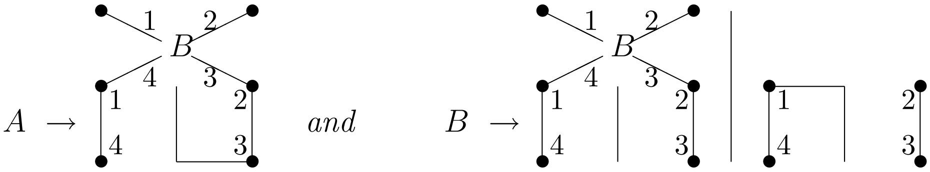


The language $L(G_1)$ generated by the collage grammar G_1 consists of "towers" of n "faces", $n \geq 0$.

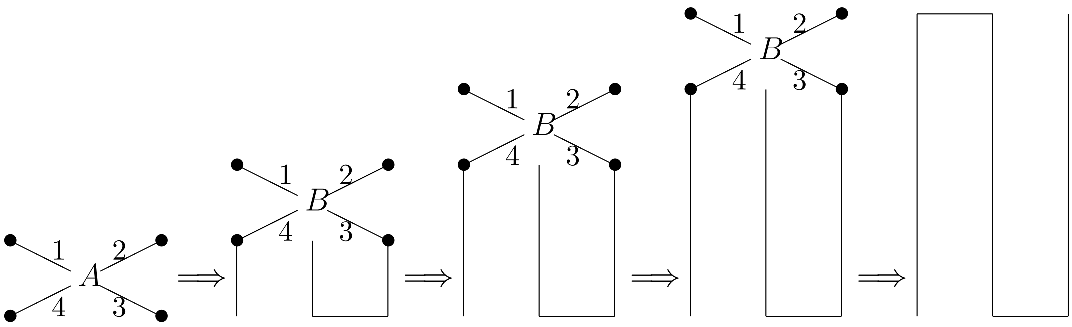
Example 4.6 *Let the collage grammar $G_2 = (\{A, B\}, P, Z)$ with the initial decorated collage*

$Z =$ 

and the set P consisting of the following three rules (where we write $A \rightarrow C_1|C_2$ instead of $A \rightarrow C_1, A \rightarrow C_2$)

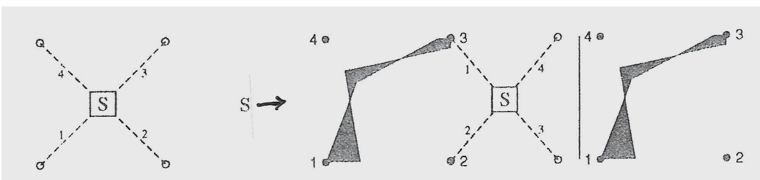


be given. A typical derivation is



It is easy to see that the language generated by G_2 is the set of all picture consisting of three parallel vertical lines of length n, $n \geq 2$, in distance 1 and two unit lines which connect the first vertical lines above and the second and third vertical line below. Therefore the generated pictures can be described as chain code pictures by the words $u^n r d^n r u^n$, $n \geq 2$.

Example 4.7 *We consider the context-free collage grammar $G_3 = (\{S\}, P, Z)$ where the initial decorated collage and the productions are given in the following line.*



In Figure 4.2 we see the decorated collages generated in one and two derivations steps, respectively, and the picture generated by fifty steps. The language consists of "spirals" similar to that shown in the right part of Figure 4.2

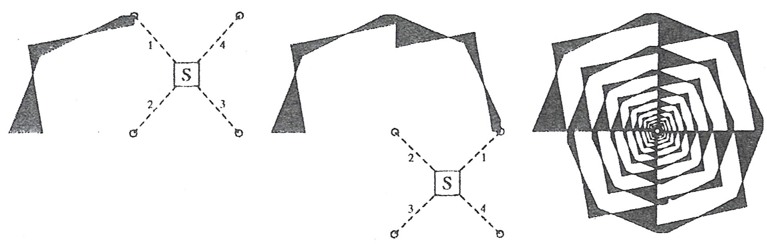


Figure 4.2: Some decorated collages generated by the grammar of Example 4.7

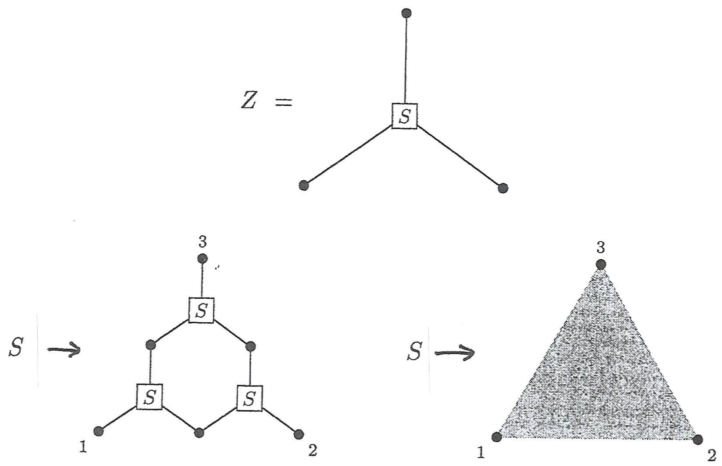


Figure 4.3: Initial decorated collage and productions of the grammar of Example 4.8

Example 4.8 *Let the context-free collage grammar $G_4 = (\{S\}, P, Z)$ be given, where the initial decorated collage Z and the elements of P are presented in the upper and lower row of Figure 4.3 The second rule produces a "black" triangle, i. e., all its inner points belong generated picture. The first rule can be interpreted as follows: a equal-sided triangle is divided into four equal-sided triangles (where the sides have half length of the original triangle); the three small triangles which have a corner in common with the original one correspond to the hyperedges and the triangle in the middle will remain empty whereas the other three can be divided further or turned to "black". In Figure 4.4 we present some of the generated pictures which are called Sierpiński triangles.*

We call a context-free collage grammar *linear*, if the initial decorated collage and all right hand sides of productions contain at least one hyperedge.

The context-free collage grammars of Examples 4.5, 4.6, and 4.7 are linear.

We present some basic properties of context-free collage grammars. We start with the decidability of the emptiness problem for context-free collage grammars.

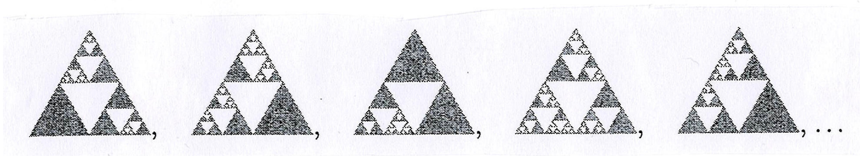


Figure 4.4: Some Sierpiński triangles

Theorem 4.9 *For a context-free collage grammar, it is decidable whether or not $L_{col}(G)$ is empty.*

Proof. We give a proof which is analogous to the word case.

Let the context-free collage grammar $G = (Lab, P, (part, pin, E, att, lab))$ be given. We define

$$M_0 = \{(A, pin') \mid A \to (part', pin', \emptyset, \emptyset, \emptyset) \in P\}.$$

Obviously, M_0 contains all labels A such that the occurrence of a hyperedge e labelled by A can be replaced in one step by a collage, i. e., by a terminal element, if pin can be mapped to $att(e)$. Moreover, for $i \geq 1$, we define M_i as the set of all pairs (A, pin''), where (A, pin'') in M_{i-1} or there is a rule $A \to (part'', pin'', E'', att'', lab'')$ such that, for any $e \in E''$, there are $(A', pin') \in M_{i-1}$ and a affine mapping t with $lab''(e) = A'$ and $t(pin') = att''(e)$. This means that we can obtain a collage from A after some steps. The first component of a pair (A, pin) in some M_i is taken from the finite set of labels; the second component is a sequence of pin points of a right hand side of a collage rule in P, i. e, it is from a finite set, too. Hence there is an index i such that $M_i = M_{i+1} = M_{i+2} = \ldots$ Now it follows that $L_{col}(G)$ is not empty if and only if for each $e \in E$ there are $(A', pin') \in M_i$ and a affine mapping t with $lab(e) = A'$ and $t(pin') = att(e)$, i.e., from Z we can derive a collage. □

Let $C = (part, pin, E, att, lab)$ be a decorated collage. By $\#(C)$ we denote the number of geometrical objects which build *part* and call it the size of C. The following theorem states that the sets of sizes of collages of infinite collage languages have no arbitrarily large gaps.

Theorem 4.10 *Let $G = (Lab, P, Z)$ be a context-free collage grammar such that*

$$M = \{\#(C) \mid Z \Longrightarrow^* C, C \text{ ia a collage}\}$$

is an infinite set. Then is a natural number c such that, for any natural number m, M has an non-empty intersection with $\{m, m+1, \ldots, m+c\}$.

Proof. Let the context-free collage grammar $G = (Lab, P, Z)$ be given. We define derivation trees as in the case of context-free word grammars. Then the inner nodes correspond to hyperedges. With an inner node we associate a pair (A, pin) if the hyperedge is labelled by A and we apply a collage rule $A \to (part, pin, E, att, lab)$ to the hyperedge.

We say that a derivation tree t is small, if the following conditions hold:

- There is no proper subtree t' of t such that the roots of t and t' are associated with the same pair.
- All proper subtrees of t are small.

Intuitively, this means that there is no subtree t' of t such that t' contains a proper subtree t'' such that the roots of t' and t'' are associated with the same pair.

We define T as the set of non-small derivation trees t according to G such that all proper subtrees of t are small. By definition, there is a node in t different from the root which is associated with the pair which is associated with the root. Obviously, the set T is finite. Let c be the maximal size of the yield of trees from T.

By the assumption, for any m there is a collage C in $L_{col}(G)$ with a size at least $m+c$. Let t be the derivation tree of C. Then it contains a proper subtree of T whose root is associated with (A, pin) and one of its inner nodes is also associated with (A, pin) and induces a small subtree (see the left part of Figure 4.5 where we only give the label). Then, for the corresponding hyperedges e and e', we have $s(pin) = att(e)$ and $s'(pin) =$

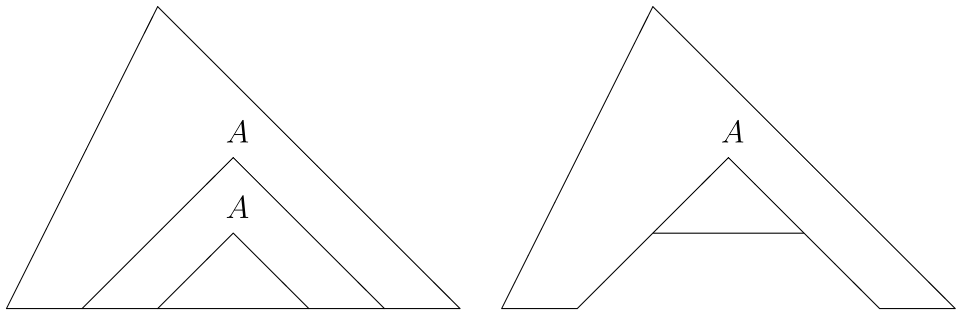


Figure 4.5: Substitution of a subtree of T

$att'(e')$ for some affine mappings s and s'. Therefore we can replace the tree of T by its small subtree and obtain a valid derivation (see the right part of Figure 4.5 which gives C_1. Obviously, $\#(C)$ and $\#(C_1)$ differ at most by c. We continue this process until we get a tree whose yield has a size mat most c. Thus we get a sequence of collage $C = C_0, C_1, C_2, \ldots C_k$ with $|\#(C_i) - \#(C_{i-1}| \leq c$ and $\#(C_k) \leq c$. Obviously, there is an i, $0 \leq i \leq k$, such that $\#(C_i) \in \{m, m+1, \ldots, m+c\}$. □

We now consider a special geometric property and show that it is decidable whether or not all pictures or at least one picture generated by a context-free collage grammar have this property.

We say that a picture p of a collage $C = (part, pin)$ contains a *ball*, if there is a point $(x, y) \in \mathbb{R}^2$ and a real number $r > 0$ such that all points $(x', y') \in \mathbb{R}^2$ with a distance $\leq r$ to (x, y) belong to an element of *part*.

We mention that a picture of a collage $C = (part, pin)$ contains a ball if and only if there is an element of *part* which contains a ball. Hen

Theorem 4.11 *For a context-free collage grammar G it is decidable whether or not*
i) $L_{col}(G)$ contains a picture which contains a ball,
ii) any picture of $L_{col}(G)$ contains a ball.

Proof. Let the context-free collage grammar $G = (Lab, P, Z)$ be given. Obviously, if Z contains a ball, then any generated decorated collage and hence all pictures of $L_{col}(G)$ contain a ball.

Therefore we assume that Z does not contain a ball.

A picture $p \in L_{col}(G)$ contains a ball if there is a derivation $Z \Longrightarrow^* C \Longrightarrow C' \Longrightarrow^* C''$ with $pic(C'') = p$ such that C does not contain a ball and C' contains a ball. Hence the rule applied to C has the form $A \to B$ where B contains a ball.

i) In the proof of Theorem 4.9 we have constructed a set M_i containing those labels (with pin points) from which a collage can be generated. We modify this construction to get a set M of labels (with pin points) from which collages containing a ball can be generated.

In addition to the pairs (A, pin), we consider also triples $(A, pin, +)$. The third component announces whether we can generate a collage containing a ball. We set $(A, pin, +) \in M'_0$ if and only if $(A, pin) \in M_0$, i. e., there is a rule $A \to (part, pin, \emptyset, \emptyset, \emptyset)) \in P$, and *part* contains a ball. $(A, pin, +) \in M'_i$ if and only if there is a rule $A \to (part, pin, E, att, lab)$ such that

- *part* contains a ball and, for all $e \in E$, there is a pair $(A'', pin'') \in M_i$ such that $lab(e) = A''$ and $t'(pin'') = att(e)$ for some affine mapping t', or

- there is an edge $e \in E$ and $(A'pin', +) \in \bigcup_{i=0}^{i-1} M'_i$ with $lab(e) = A'$, $t(pin') = att(e)$ for some affine mapping t and, for all edges $e' \in E \setminus \{e\}$, there is a pair $(A'', pin'') \in M_i$ such that $lab(e') = A''$ and $t'(pin'') = att(e')$ for some affine mapping t'.

It is easy to see that we can generate a collage with ball starting with some rule $A \to (part, pin, E, att, lab))$ if and only if $(A, pin, +)$ in M'_i for some i. Moreover, $M = \bigcup_{i \geq 0} M'_i$ is finite and can be constructed (we construct in succession the sets M'_i and stop if $M'_j \setminus \bigcup_{i=0}^{j-1} M'_i$ is empty, i. e., no new element which can generate a ball is obtained.

Now we check whether the following situation holds for $Z = (part, pin, E, att, lab)$: there is an edge $e \in E$ and $(A'pin', +) \in \bigcup_{i=0}^{i-1} M'_i$ with $lab(e) = A'$, $t(pin') = att(e)$ for some affine mapping t and, for all edges $e' \in E \setminus \{e\}$, there is a pair $(A'', pin'') \in M_i$ such that $lab(e') = A''$ and $t'(pin'') = att(e')$ for some affine mapping t'. If the answer is yes, then we can generate a picture containing a ball. Otherwise no picture with ball can be generated.

ii) Again, we modify the proof for the decidability of the emptiness problem. We use triples $(A, pin, -)$ where the third component announces that a collage without ball can be generated. More formally, we define M''_0 as the set of triples $(A, pin, -)$ where rule $A \to (part, pin, \emptyset, \emptyset, \emptyset)) \in P$ exists such that *part* contains no ball. For $i \geq 1$, M''_i is the set of all triples $(A, pin, -)$ such that

- $(A, pin, -)$ is in M''_{i-1} or

- there is a rule $A \to (part, pin, E, att, lab) \in P$ where *part* contains no ball and, for all $e \in E$, there is a pair $(A', pin') \in M''_{i-1}$ such that $lab(e) = A'$ and $t(pin') = att(e)$ for some affine mapping t.

This ensures that a derivation starting with a rule $A \to (part, pin, E, att, lab) \in P$ and $(A, pin, -) \in M_i$ for some i, $i \geq 0$, can generate a collage without ball.

Obviously, there is an j such that $M_j = M_{j+1} = M_{j+2} = \ldots$

Therefore $L_{col}(G)$ contains a collage without ball if and only if the following condition holds for $Z = (part, pin, E, att, lab)$: for all edges $e \in E$, there is a pair $(A, pin) \in M_j$ such that $lab(e) = A$ and $t(pin) = att(e)$ for some affine mapping t. □

We now consider the extension to context-sensitive collage rules and grammars.

Definition 4.12 *i) A* context-sensitive collage rule *is a pair (L, R), where L is a decorated collage with a single hyperedge, i.e., $L = (part_L, pin_L, E_L, att_L, lab_L)$ with $E_L = \{e\}$ for some hyperedge e, and $R = (part_R, pin_R, E_R, att_R, lab_R)$ is a decorated collage. Again, we write $L \to R$ instead of (L, R).*

ii) The application of a context-sensitive collage rule $L \to R$ to a decorated collage $C = (part_C, pin_C, E_C, att_C, lab_C)$ is only possible if there is an hyperedge $e' \in E_C$ with $lab_C(e') = lab_L(e)$ and there is an affine mapping t with $t(att_L(e)) = att_C(e)$ and such that $t(part_L) \subset part_C$, and there is an affine mapping t' with $t'(pin_R) = att_L(e)$.

The application results in a decorated collage grammar C' which is obtained by applying the context-free free collage rule $lab_L(e) \to R$ to C with the mapping $t \circ t'$. We then write $C \Longrightarrow C'$.

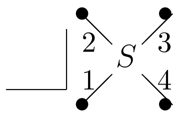
iii) A context-sensitive collage grammar *is a triple $G = (Lab, P, Z)$, where Lab is a finite set of labels, Z is a decorated collage over Lab, P is a finite set of context-sensitive collage rules $L \to R$ where L and R are decorated collages over Lab.*

iv) The collage language generated by a context-sensitive collage grammar *is defined by*

$$L_{col}(G) = \{pic(C) \mid Z \Longrightarrow^* C, C \text{ is a collage}\}$$

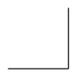
where $\Longrightarrow^*$ denotes the reflexive and transitive closure of $\Longrightarrow$.

We regard the following example. Let $G_5 = (\{S, A, B\}, P, Z)$ be the context-sensitive collage grammar with

$Z =$ 

and the productions given in Figure 4.6 (the attachment points and pin points are given by numbers in roman and italics, respectively)

A typical derivation according to G_5 is given in Figure 4.7.

It is easy to see that $L_{col}(G_5)$ consists of all lower triangles of angles  If we consider each angle as one geometrical element, for any picture p of $L_{col}(G_5)$, there is a number n such that p has $n(n+1)/2$ geometrical elements. By Theorem 4.10, it is easy to see that $L_{col}(G_5)$ cannot be generated by a context-free collage grammar. Thus we have the following theorem.

Theorem 4.13 *There is a collage language generated by a context-sensitive collage grammar, which cannot be generated by a context-free collage grammar.* □

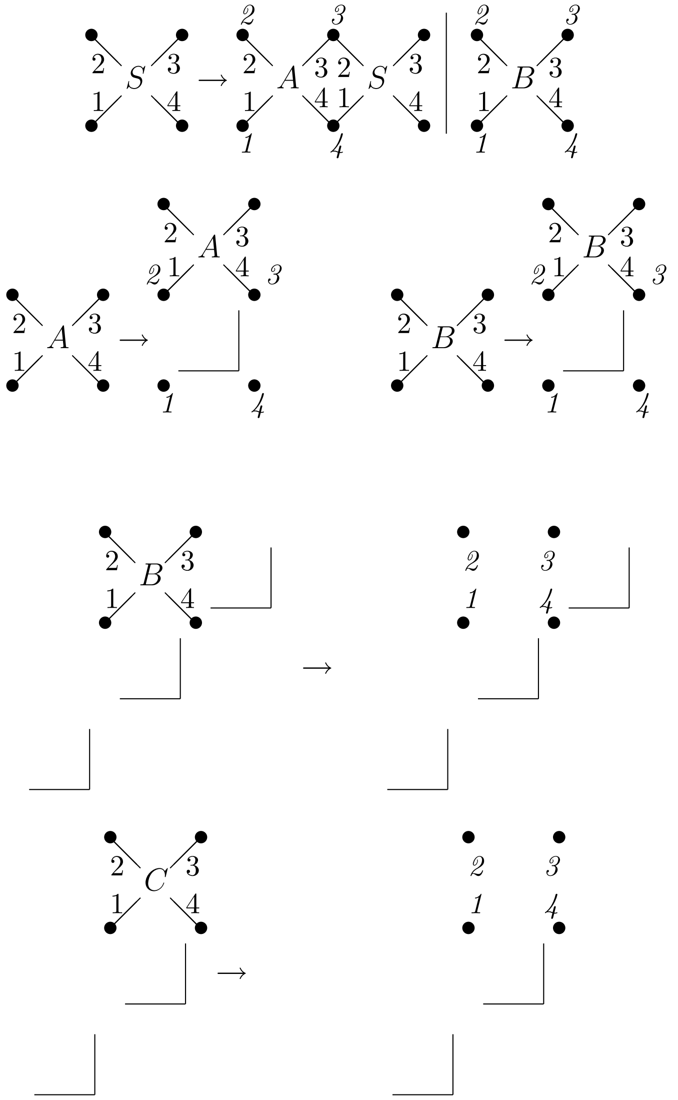


Figure 4.6: Rules of the context-sensitive collage grammar G_5

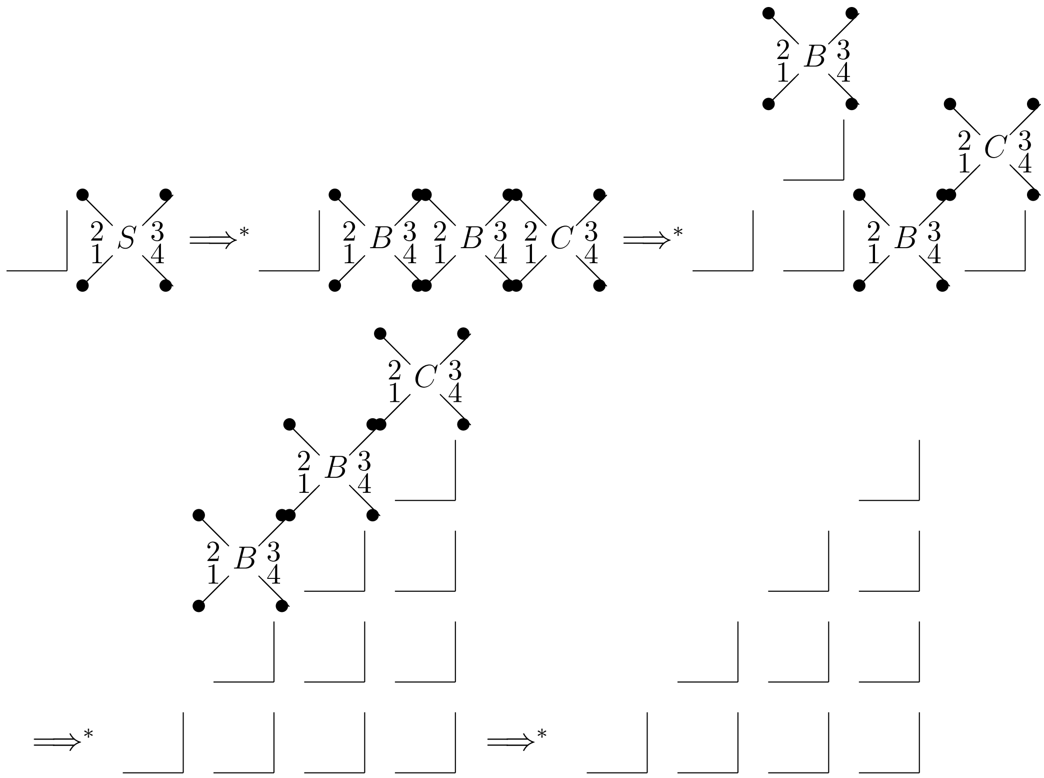


Figure 4.7: Typical derivation according to G_5

4.2 Collage Grammars with Chain Code Pictures as Parts

As one can see from the Examples 4.5, 4.7, and 4.8, collage grammars can generate pictures which cannot be generated by chain code picture grammars (and Siromoney matrix grammars) because such grammars generate only pictures consisting of a finite set of unit lines in the grid $\mathbb{Z}^2$. Therefore we restrict collage grammars in such a way that they also generate only chain code pictures. This allows us to do a more precise comparison of chain code picture grammars and collage grammars.

The restriction to objects of $\mathbb{Z}^2$ requires a restriction to special affine transformations, too, since arbitrary affine transformations can map a point of $\mathbb{Z}^2$ to a point not contained in $\mathbb{Z}^2$. Therefore we restrict in this paper to translations which map $\mathbb{Z}^2$ onto $\mathbb{Z}^2$.

Definition 4.14 *i) A decorated collage* $(part, pin, E, att, lab)$ *is called a decorated chain code collage (decorated cc-collage, for short), if part is a finite set of unit lines of the grid* $\mathbb{Z}^2$*, pin and* $att(e)$ *for all* $e \in E$ *belong to* $(\mathbb{Z}^2)^*$

ii) A context-free collage grammar $G = (Lab, P, Z)$ *is called a chain code collage grammar (cc-collage grammar, for short), if* Z *and all right hand sides of rules of* P *are decorated cc-collages.*

Example 4.6 gives an example for a cc-grammar if we interpret the parts consisting of unit lines (in distance 1.

Above we noticed that there are collage languages which are not chain code picture languages. But as example we mentioned collage languages which contain objects which cannot be described by chain codes. We now strengthen the above remark to collage languages which contain only chain code pictures.

Lemma 4.15 *There is a linear cc-collage grammar* G *such that* $L_{col}(G) \notin \mathcal{CCP}(CF)$.

Proof. We consider the linear cc-collage grammar G_2 of Example 4.6. We have shown in Example 4.6 that it generates $L_2 = bccp(L_2')$ with $L_2 = \{u^n r d^n r u^n \mid n \geq 2\}$.

Let us assume that L_2 is generated by a context-free chain code picture grammar $G = (N, \pi, P, S)$. Since G has to generate an infinite set of words, there is a derivation

$$S \Longrightarrow^* vAv' \Longrightarrow^* vxAyv' \Longrightarrow^* vxwyv' \in \pi^*.$$

Let $sh(vxwyv') = (p_1, p_2)$ and $sh(x) = (q_1, q_2)$. Note that $-3 \leq p_1 \leq 3$ because the width of the pictures is 3. Since we also have the derivations

$$S \Longrightarrow^* vAv' \Longrightarrow^* uxAyv \Longrightarrow^* ux^nAy^nv \Longrightarrow^* ux^nwy^nv \in \pi^*$$

with $n \geq 2$, we get $q_1 = 0$ (otherwise $sh(ux^nwy^nv) = sh(uxwyv) = (p_1 + (n-1)q_1, p_2 + (n-1)q_2)$ where $3 \leq p_1 + (n-1)q_1 \leq 3$ is not satisfied). Analogously, if $q_2 \neq 0$, then we obtain pictures where the vertical lines have different length. Thus $sh(x) = (0, 0)$. Analogously, we can prove that $sh(y) = (0, 0)$. Therefore G is a normal grammar (see Definition 2.17. Then $bccp(G)$ is finite by Corollary 2.20. However, this contradicts the fact that L_2 contains infinitely many pictures. $\square$

We mention that L_2 can be generated by the regular extended chain code picture grammar $G'_2 = (\{S, A\}\{u, d, r, l, \uparrow, \downarrow\}, P, S)$ with

$$P = \{S \to u \uparrow r \downarrow dru \uparrow ll \downarrow A,\ A \to u \uparrow r \downarrow d \uparrow r \downarrow u \uparrow ll \downarrow A,\ A \to urd \uparrow r \downarrow u\}$$

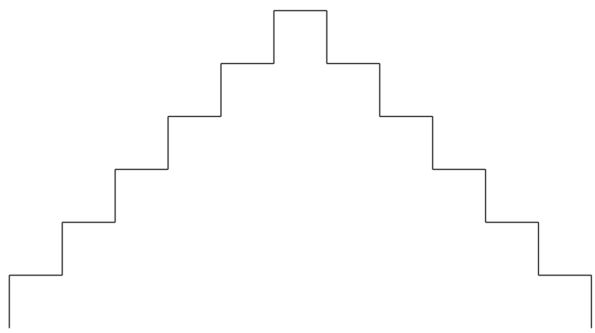
Lemma 4.16 *There is a chain code picture language $L \in \mathcal{CCP}(LIN)$ such that L cannot be generated by a context-free cc-collage grammar.*

Proof. We do not give a complete proof; we only illustrate the idea.

The linear chain code picture grammar

$$G = (\{S\},\ \pi,\ \{S \to urSrd,\ S \to urd\},\ S$$

generates the language L of all pictures of the form



which are described by words $(ur)^n urd(rd)^n$.

Let us assume that there is a context-free cc-collage grammar G' generating L. Let n be sufficiently large and consider the picture $p = bccp((ur)^n urd(rd)^n)$. We consider a derivation tree t for p. Using the arguments of the proof of Theorem 4.10 we can substitute a subtree tree t' of t by a smaller subtree t''. However, the size of the geometrical figures is changed by the substitution. Therefore the picture obtained by the substitution does not only consist of unit lines. This contradicts the structure of the pictures in L. □

We give now two lemmas which relate extended chain code picture grammars and cc-collage grammars.

Lemma 4.17 *For any regular extended chain code picture grammars G, a linear cc-collage grammar G' with $bccp(G) = L_{col}(G')$ can be constructed.* □

Proof. We only give the proof for regular chain code picture grammars (since we only need it for this case in the sequel). The generalization to extended regular chain code picture grammars is left to the reader.

Let $G = (N, \pi, P, S)$ be given where all productions of P are of the $A \to wB$ or $A \to w$ with $A, B \in N$ and $w \in \pi^*$. Let

$$Q = \{sh(w) \mid A \to wB \in P \text{ or } A \to w \in P\}.$$

With a word wB with $dccp(w) = ((0,0), p, z)$ and $q \in Q$, we associate the decorated collages $col(wB, q) = (p, (0,0)z, \{e\}, att, lab)$ with $att(e) = z(z+q)$ and $lab(e) = B$. Moreover with a word wB with $dccp(w) = ((0,0), p, z)$, we associate the decorated collage $col(w) = (p, (0,0)z, \emptyset, ßemptyset, \emptyset)$.

We construct the linear cc-collage grammar $G' = (N \cup \{S'\}, P', Z)$ where

- $Z = (\emptyset, \emptyset, \{e\}, att, lab)$ with $att(e) = (0,0)(1,1)$ and $lab(e) = S'$
- all rules $S' \to (\emptyset, (0,0)(1,1), \{e\}, att, lab)$ with $att(e) = (0,0)q$ and $lab(e) = S$ are in P',
- for any rule $A \to wB \in P$ and any $q \in Q$, the collage rule $A \to col(wB, q)$ belongs to P',
- for any rule $A \to w$, the collage rule $A \to col(w)$ belongs to P'.

It is easy to prove by induction on the length of the derivations that

- $A \Longrightarrow w \in \pi^*$ implies $A' \Longrightarrow col(w)$ where A' is the decorated collage consisting of a single hyperedge with label A and attachment point which allow the application of $A \to col(w)$,
- $A \Longrightarrow^x B \Longrightarrow^* w \in \pi^*$ in G implies $A' \Longrightarrow col(xB) \Longrightarrow^* dccp(w)$ in G' where A' is the decorated collage consisting of a single hyperedge with label A and attachment point which allow the application of $A \to col(w, B, q)$ (note that this does not depend on q since the attachment points of A' have to fit to the pin points of $col(wB, q)$ which only depend on w).

Taking into consideration that the rules applicable to Z generate exactly decorated collages which consist of a single hyperedge labelled by S a attachment point which allow the start of a derivation, it follows that $bccp(G) \subseteq L_{col}(G')$.

Conversely, we prove by induction on the length of a derivation that $A' \Longrightarrow^* C$ with a collage C in G' (where A' corresponds to A as in the first part of the proof) implies $A \to w$ with $dccp(w) = pic(C)$. From this we get $L_{col}(G') \subseteq dccp(G)$.

Both shown inclusions yield $L_{col}(G') = dccp(G)$. □

Analogously, one can show the following statement.

Lemma 4.18 *For any linear cc-collage grammar G', there is e regular extended chain code picture grammar G' such that $L_{col}(G) = bccp(G)$.* □

The Lemmas 4.17 and 4.18 allows us to transfer some statements from regular (extended) chain code picture grammars/languages to (cc-collage) grammars/languages.

Theorem 4.19 *The equivalence problem*

Given: *collage grammars G_1 and G_2*
Question: *Does $L_{col}(G) = L_{col}(G_2)$ hold?*

is undecidable for linear cc-collage grammar (hence for context-free collage grammar, too).

Proof. Assume that the equivalence problem is decidable for linear cc-collage grammars.

Let two regular chain code picture grammars G_1 and G_2 be given. We construct the linear cc-collage grammars G_1' and G_2' with $bccp(G_1) = L_{col}(G_1')$ and $bccp(G_1) = L_{col}(G_1')$. Obviously, G_1 and G_2 are (picture) equivalent if and only if G_1' and G_2' are equivalent. Hence our assumption implies the decidability of the equivalence problem for regular chain code picture grammars. This contradicts Theorem 2.23. □

Using the same proof idea and Theorem 2.29, we can show the following statement.

Theorem 4.20 *It is undecidable whether or not a linear cc-collage grammar generates*
(a) a simple curve,
(b) a closed simple curve,
(c) a regular picture, (d) an Eulerian picture,
(e) a Hamiltonian picture,
(f) a tree. □

Bibliography

[1] K. Culik II and J. Kari, Digital images. In: [22], Vol. III, 599–616

[2] J. Dassow and F. Hinz, Decision problems and regular chain code picture languages. *Discrete Appl. Math.* **45** (1993) 29–49.

[3] F. Drewes, *Grammatical Picture Generation.* Springer-Verlag, Berlin, 2006.

[4] F. Drewes and H.-J. Kreowski, Picture generation by collage grammars. In: H. Ehrig, G. Engels, H.-J. Kreowski and G. Rozenberg (Eds.), *Handbook of Graph Grammars and Computing by Graph Transformations*, Vol. II, World Scientific, Singapore, 1999, 397–457.

[5] S. Ewert and A. P. J. van der Walt, Random context picture grammars. *Publicationes Math.* **54** (1999) 763–786.

[6] S. Ewert and A. P. J. van der Walt, Generating pictures using random permitting context. *Internat. J. Pattern Recognition* **12** (1999) 939–950.

[7] H. Freeman, On the encoding of arbitrary geometric configurations. *IRE Transactions on Electronic Computers* **10** (1961) 260–268.

[8] H. Freeman, Computer processing of line-drawing images. *Computer Surveys* **6** (1974) 57–97.

[9] D. Giammarresi, Finite state recognazability for two-dimensional languages: a brief survey. In: J. Dassow, G. Rozenberg and A. Salomaa (Eds.), *Developments in Language Theory II*, World Scientific, Singapore, 1997, 299–308.

[10] D. Giammarresi and A. Restivo, Two-dimensional languages. In: [22], 215–267.

[11] A. Habel and H.-J. Kreowski, Collage grammars. In: H. Ehrig, H.-J. Kreowski and G. Rozenberg (Eds.), *Proc. 4^{th} Internat. Workshop Graph Grammars and their Application to Computer Science*, Lecture Notes in Computer science 532, Springer-Verlag, Berlin, 1991, 411–329.

[12] G. T. Herman and G. Rozenberg, *Developmental Systems and Languages.* North-Holland, Amsterdam, 1975.

[13] F. Hinz and E. Welzl, Regular chain code picture languages with invisible lines. Techn Report 252, IIG, TU Graz, 1988.

[14] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Complexity.* Addison-Wesley, Reading, 1979.

[15] K. Inoue and I. Takanami, A survey of two-dimensional automata theory. In: J. Dassow and J. Kelemen (Eds.), *Machines, Languages, and Complexity*, Lecture Notes in Computer Science 381, Springer-Verlag, Berlin, 1989, 72–91.

[16] Ch. Kim, Complexity and decidability for restricted classes of picture languages. *Theor. Comp. Sci.* **73** (1990) 295–311.

[17] D. C. Kozen *Automata and Computability.* Springer-Verlag, New York, 1997.

[18] H. W. Maurer, G. Rozenberg and E. Welzl, Using string languages to describe picture languages. *Inform. Control* **54** (1982) 155–185.

[19] P. Pruzinkiewicz, M. Hammel, J. Hanan and R. Mech, Visual models of plant development. In: [22], Vol. III, 1997, 535–597.

[20] P. Pruzinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants.* Springer-Verlag, New York, 1990.

[21] G. Rozenberg and A. Salomaa, *The Mathematical Theory of L Systems.* Academic Press, New York, 1980.

[22] G. Rozenberg and A. Salomaa (Eds.), *Handbook of Formal Languages.* Vol. I – III, Springer-Verlag, Berlin, 1997.

[23] A. Rosenfeld and R. Siromoney, Picture languages - a survey. *Languages of Design* **1** (1993) 229–245.

[24] I. H. Sudborough and E. Welzl, Complexity and decidability of chain code picture languages. *Theor. Comp. Sci.* **36** (1985) 175–202.

[25] C. Kim, Picture Iteration and Picture Ambiguity. *J. Comput. Syst. Sci.* **40** (1990) 289–306.

[26] A. Rosenfeld, Isotonic grammars, parallel grammars, and picture grammars. In: *Machine Intelligence* 6 (Eds. B. Meltzer and D. Michie), Edinburgh Univ. Press, 1971, 281–294.

[27] A. Rosenfeld, *Picture Languages – Formal Models for Picture Recognition.* Academic Press, New York, 1979.

[28] A. Rosenfeld, Array grammars. In: *Graph-Grammars and Their Application to Computer Science*; Springer, Berlin, 1986, 67–70.

[29] A. Saoudi, K. Rangarajan and V. R. Dare, Finite images generated by GL-systems. In: [36], 181–190.

[30] R. Siromoney On equal matrix languages. *Inform. Control* **14** (1969) 135–151.

[31] R. Siromoney, K. G. Subramanian and V. Rajkumar Dare, Infinite arrays and controlled deterministic table 0L array systems. Theor. Comput. Sci. **33** (1984) 3–11.

[32] R. Stiebe Picture Generation Using Matrix Systems. Proc. Intern. Meeting Young Computer Scientists, Bratislava, 1990, 251–260

[33] R. Stiebe Picture generation using matrix systems. *Journal of Information Processing and Cybernetics (EIK)* **28** (1992) 311–327.

[34] R. Stiebe, Subimage problems for Siromoney matrix languages. In: Proc. 14. Theorietag Automaten und formale Sprachen, 2004, Univ. Potsdam, 123–128.

[35] R. Stiebe, Subimage problems for Siromoney matrix languages. Manuscript, 2006.

[36] P. S.-P. Wang (Ed.), *Array Grammars, Patterns and Recognizers.*World Scientific, 1989.

[37] P. S.-P. Wang, Three-dimensional sequential/parallel universal array grammars and object pattern analysis. *ICPIA* (1992) 305–312.