COLLAGE-ONE: A SYSTEM FOR EVALUATION AND VISUALIZATION OF COLLAGE GRAMMARS

F. Drewes, H.-J. Kreowski, N. Schwabe
Universität Bremen, Fachbereich 3, D–28334 Bremen (Germany)
E-mail: drewes,kreo,ns@informatik.uni-bremen.de

Abstract. COLLAGE-ONE is a system for evaluating and visualizing collage grammars, which are rule-based picture-generating devices of a context-free type. Pictures are represented as collages that consist of coloured parts, and they are derived by applying rules to collages, thereby producing pictures that get more and more detailed. COLLAGE-ONE provides a textual and graphical collage grammar editor, an evaluator of derivations, and a display of the derived pictures.

Key words. Collage grammar, visualization, picture generation

1. Introduction

One encounters quite a variety of rule-based and syntactic approaches to picture generation in the literature like shape grammars, cellular automata, chain-code picture languages, and L-systems with turtle interpretation (see, e.g., [3, 1, 4, 8, 2]). Also iterated function systems [6, 10], which are prominently used in fractal geometry, may be seen in this context. This paper is concerned with collage grammars as introduced by Habel, Kreowski, and Taubenberger [9, 12]. The framework of collage grammars may be considered as a generalization of iterated function systems with a context-free type of rules. It combines an attractive mathematical theory (see, e.g., [14, 15, 16]) with a promising generative power and a modelling spectrum ranging from fractals and tilings to branching structures, growing processes, and textures. The basic idea of collage grammars is to represent a picture as a collage consisting of (coloured) parts and to decorate a collage with hyperedges that span working areas and are subject to further refinement. In the refining process, hyperedges are replaced by other collages according to specified rules. In this way of derivation, one gets arbitrarily detailed and multi-coloured pictures even if one provides only simply shaped and uni-coloured parts by the rules.

The subject of this paper is COLLAGE-ONE, a system implementing collage grammars. It allows to edit a collage grammar either textually or graphically and to evaluate a specified derivation. The resulting sequence of pictures can be viewed on the screen and saved as high resolution PostScript files. The use, illustrations, and more technical details of COLLAGE-ONE as a system for evaluation and visualization of collage grammars are discussed in Section 3 while Section 2 recalls the basic features of collage grammars.

2. Collages and collage grammars

In this section, we merely recall the concept of collage grammars as introduced and studied in [9, 12], but allow colours in addition.

2.1. Notations

For a set X, $\wp(X)$ denotes the powerset of X, and X^* denotes the set of all finite sequences over X, including the empty sequence λ. The concatenation of two sequences u and v is denoted by $u \cdot v$. If $u_1, u_2, \ldots, u_n$ are sequences, then $\prod_{i=1\ldots n} u_i$ denotes $u_1 \cdot \ldots \cdot u_n$.

Given ordered sets X and Y, where the order is always denoted by $<$, $Y - X$ denotes the complement of X in Y, and the disjoint union is denoted by $X \sqcup Y$, where $x < y$ for all $x \in X, y \in Y$ and the orders on X and Y are retained.

Familiarity with the basic notions of Euclidean geometry (see, e.g., Coxeter [7]) is assumed. $\mathbb{R}$ denotes the set of real numbers and $\mathbb{R}^d$ the Euclidean space of dimension d for some $d \geq 1$.

2.2. Collages

Collages comprise coloured parts and pin points. For generating purposes, collages are decorated with hyperedges where a hyperedge is labelled and attached to a sequence of points. The picture represented by a collage is obtained by the overlay of the parts.

More formally, let $COLOUR$ be a fixed set of *colours* and N be a fixed set of *labels*. Then a *(hyperedge-decorated) collage* consists of a sequence $PART_C$ of *parts*, a sequence pin_C of *pin points*, an ordered set $EDGE_C$ of *hyperedges*, and mappings $lab_C : EDGE_C \rightarrow N$ and $att_C : EDGE_C \rightarrow (\mathbb{R}^d)^*$ assigning a *label* and a sequence of *attachment points* respectively to each hyperedge. Each *part* in $PART_C$ is a pair $(P, colour)$ consisting of a set P of points in $\mathbb{R}^d$ together with a *colouring* mapping $colour\colon part \rightarrow COLOUR$. In the following we will not distinguish explicitly between a part *part* and its set P of points.

The class of all collages over N is denoted by $\mathcal{C}(N)$. The class of *terminal collages*, i.e. collages without hyperedges, is denoted by $\mathcal{C}$.

Let $PART_C = part_1 \cdots part_k$. Then the *picture* of C is the union of all parts, $pict(C) = \bigcup_{i=1}^{k} part_i$, together with the colouring mapping $colour\colon pict(C) \rightarrow COLOUR$ given as follows: If $p \in part_{i_0}$ for some i_0, it keeps its colour provided that $p \notin \bigcup_{i=i_0+1}^{k} part_i$. Intuitively, one may think of a part as an—often uni-coloured—geometric object of an elementary form like a cube, a tetrahedron, a ball etc. Nevertheless the picture underlying a collage may get arbitrarily detailed and multi-coloured because the collage may comprise many parts of different forms and colours. If parts overlap, one has to face a colouring conflict that we resolve by assuming that a part with higher index overlays a part with lower index. This explains why we assume a sequence of parts rather than a set, and results in the definition of $pict(C)$ above. The definition is somewhat related to the well-known *painter's algorithm* commonly used to solve the hidden-surface problem in computer graphics (see, for example, [5, p. 478]).

2.3. Hyperedge replacement

The replacement of a hyperedge e of a collage C by a collage R works as follows. In a first step, R is transformed such that the transformed pin points match the attachment points of e. Afterwards, the hyperedge is replaced by the transformed collage yielding a refinement of C. In the explicit construction, the sequence of parts of C is concatenated with the sequence of transformed parts, e is removed, and the hyperedges of R are added disjointly. The remaining hyperedges keep their labels and attachment points. The added hyperedges keep their labels and get attached to their transformed attachment points. Moreover, the pin points of C are kept. In the same way, a set of hyperedges can be replaced by corresponding collages simultaneously. In the formal definition given below, we define first the transformation of a collage and then the replacement.

Let $t\colon \mathbb{R}^d \to \mathbb{R}^d$ be a transformation on $\mathbb{R}^d$ and C be a collage. Then there is a *transformed collage* $t(C)$ constructed as follows: $PART_{t(C)} = t(PART_C)$, $pin_{t(C)} = t(pin_C)$, $EDGE_{t(C)} = EDGE_C$, $lab_{t(C)} = lab_C$, and $att_{t(C)}\colon EDGE_{t(C)} \to (\mathbb{R}^d)^*$ defined by $att_{t(C)}(e) = t(att_C(e))$ for each $e \in EDGE_{t(C)} = EDGE_C$.[1]

Let C be a collage and $B \subseteq EDGE_C$ with m elements $b_1 < \cdots < b_m$. Let $R_1, \ldots, R_m$ be collages and $t_1, \ldots, t_m$ be transformations on $\mathbb{R}^d$ with $t_i(pin_{R_i}) = att_C(b_i)$ for $i = 1, \ldots, m$. Then the *replacement* of B by $R_1, \ldots, R_m$ with respect to $t_1, \ldots, t_m$ yields the collage X constructed as follows: $PART_X = PART_C \cdot \prod_{i=1}^m PART_{t_i(R_i)}$, $pin_X = pin_C$, and $EDGE_X = (EDGE_C - B) \sqcup \bigsqcup_{i=1}^m EDGE_{t_i(R_i)}$ where each hyperedge of X gets its label and attachment points from the component it stems from.

In the following, we restrict our attention to the set AFF of affine transformations on $\mathbb{R}^d$.

2.4. Collage grammar

Hyperedge replacement yields the derivation process of collage grammars. A pair $A \to R$ with a label A and a collage R is called a *production*. Let P be a set of productions. Let C be a collage and $B \subseteq EDGE_C$ with m elements $b_1 < \cdots < b_m$. Let, finally, C' be the replacement of B by collages $R_1, \ldots, R_m$ with respect to transformations $t_1, \ldots, t_m \in AFF$. Then we say that C *derives directly* C' (over P), denoted by $C \Longrightarrow_P C'$, provided

[1] Given a transformation $t\colon \mathbb{R}^d \to \mathbb{R}^d$, there are natural extensions to sequences and sets of points as well as to sequences of sets of points: $t(p_1 \cdots p_k) = t(p_1) \cdots t(p_k)$ for $p_1 \cdots p_k \in (\mathbb{R}^d)^*, t(S) = \{t(p) \,|\, p \in S\}$, for $S \subseteq \mathbb{R}^d$, and $t(S_1 \cdots S_l) = t(S_1) \cdots t(S_l)$ for $S_1, \ldots, S_l \subseteq \mathbb{R}^d$. For coloured parts $part = (P, colour)$ we let $t(part) = (t(P), colour \circ t^{-1})$, where $\circ$ denotes functional composition.

that the pair $lab_C(b_i) \to R_i$ is in P for $i = 1, \ldots, m$. Sometimes it is reasonable to accept only specific productions in certain derivation steps rather than always all productions. For this purpose, we assume a set $\mathcal{P}$ of sets of productions, called a set of *tables*, and denote $C \Longrightarrow_P C'$ also by $C \Longrightarrow_{\mathcal{P}} C'$ provided that $P \in \mathcal{P}$. Accordingly, we get *derivations* $C \Longrightarrow_{\mathcal{P}}^* C'$, where $\Longrightarrow_{\mathcal{P}}^*$ is the transitive and reflexive closure of $\Longrightarrow_{\mathcal{P}}$. The idea is to be able to apply productions from different tables in different derivation steps, but to use only one table in each single step.

A set N of labels, a set $\mathcal{P}$ of tables, and a start collage Z form a *collage grammar* CG. The language of CG comprises all terminal collages derivable from Z using tables in $\mathcal{P}$:

$$L(CG) = \{C \in \mathcal{C} \mid Z \Longrightarrow_{\mathcal{P}}^* C\}.$$

2.5. Derivation specification

The derivation process of a collage grammar is highly non-deterministic in general, which is quite a desirable property because the intended collage languages are usually infinite, but must be generated from a single start collage. In contrast to that we want to display the collage that is the result of a single derivation in COLLAGE-ONE. Hence, we need facilities to select derivations out of the potentially infinitely many choices. Explicitly, we get rid of the non-determinism in the following way.

First, we restrict ourselves to collage grammars with the uniqueness property. This means, whenever a production can be applied to a hyperedge, the involved transformation is required to be unique. This is guaranteed, for example, if one deals with affine transformations and the pin points of right hand sides of productions each span a d-dimensional area. Second, the sequence of tables to be applied must be specified. This yields already a single derivation if there is at most one production for each label in each table and if one chooses maximum parallelism as mode of replacement. Third, probabilities are assigned to the productions such that the sum of probabilities of productions for the same label in the same table is less than or equals 1. Now, in a derivation step, a production for the specified table is chosen randomly for each hyperedge according to its probability. For this purpose, COLLAGE-ONE employs a random number generator. In the case that the sum of probabilities is less than 1, some hyperedges may not be replaced in some steps.

3. Collage-One by example

In this section some typical examples of collage grammars and their visualization in COLLAGE-ONE are briefly discussed. Rather than showing every detail the aim is to provide the reader with an intuition about the main features of collage grammars and how these features are reflected in COLLAGE-ONE.

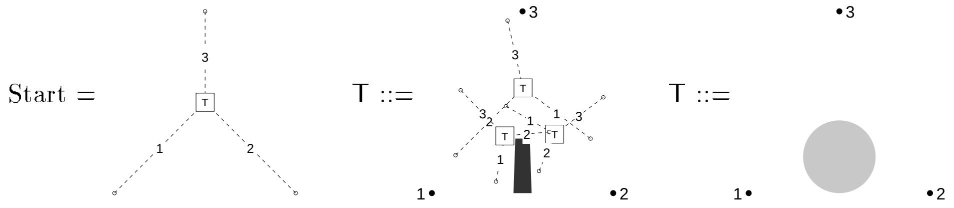

Fig. 1. A collage grammar generating tree-like structures.

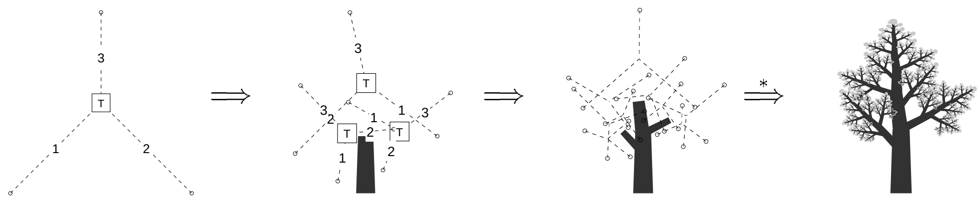

Fig. 2. A parallel derivation in the tree grammar.

Interaction with the system is quite simple. If given a grammar file together with a derivation specification, the system generates a derivation as described in 2.5. The generated derivation depends on the built-in random number generator if rules with probabilities are used. The collages given by the right-hand sides of rules and those of the derivation steps are held in memory as consecutive so-called pages (first the rules as they appear in the grammar, then the derivation steps in their natural order). These pages can be selected and are then displayed in the main window. Furthermore, there is a zooming facility that allows to magnify or shrink the representation on the screen. Pressing a *reload*-button causes a new derivation to be computed, probably based on different probabilistic choices and thus yielding a different sequence of collages.

We shall first consider a simple collage grammar which employs only two rules. The intention is to generate a tree-like structure similar to the images of artificial plants known from graphical interpretations of L-systems (see [8]). The start collage is a single hyperedge labelled with T. There are two tables, each of which consists of a single rule. The start collage and the two rules are depicted in a kind of Backus-Naur form in Figure 1. The right-hand side of the first rule consists of three pin points, a part which can be thought of as the trunk of the tree to be generated, and three hyperedges. The hyperedge in the middle will cause the trunk to extend during subsequent derivation steps, whereas the other two hyperedges yield branches of the tree. The second rule is a terminating one—it contains no hyperedges. Note that the part of the first right-hand side is darker than the one of the second. Unfortunately, in this black-and-white printout no coloured parts can be shown.

Figure 2 shows a derivation. Here, the derivation specification says that table 1 has

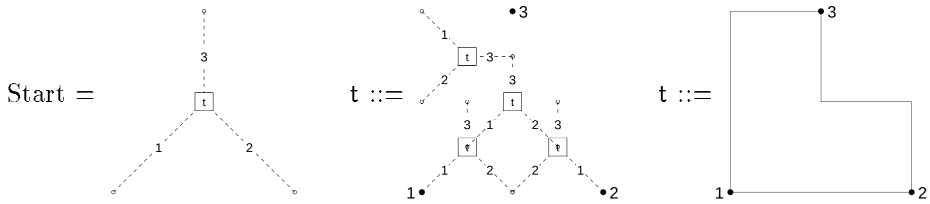

Fig. 3. A collage grammar generating tilings.

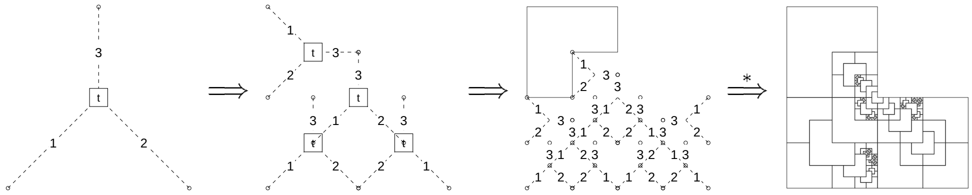

Fig. 4. A derivation with probabilities in the tiling grammar.

to be applied in the first eight steps. Afterwards, table 2 is applied once. Note that, intuitively, the middle hyperedge of the first right-hand side is mirrored, which yields a somewhat less regular shape than the one that would have been encountered otherwise.

The second example, whose start collage and rules can be seen in Figure 3, also has two rules, but this time the first table contains both rules whereas, as before, the second contains only the terminating one. Since both rules have the same left-hand side, probabilities must be specified for them in the first table. Here, both rules are applied with equal probability. Hence, in contrast to the first example, a derivation specification does not determine the resulting collage in a unique way (as long as the first step employs table 1). Two derivations that comply with one and the same derivation specification may result in different collages. A sample derivation is given in Figure 4.

Finally, we can use the rules of the two previous grammars together with further ones to design a collage grammar generating the sort of "spirals" a (clipped) sample of which

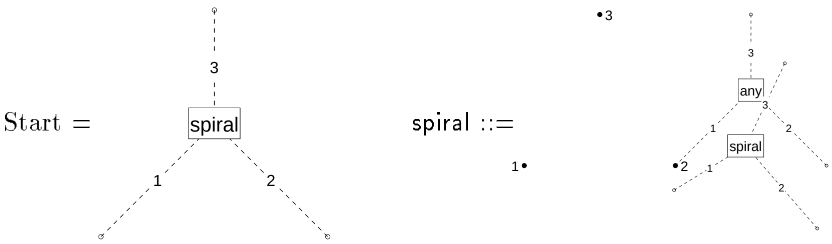

Fig. 5. A collage grammar that generates a spiral consisting of "any" kind of elements.

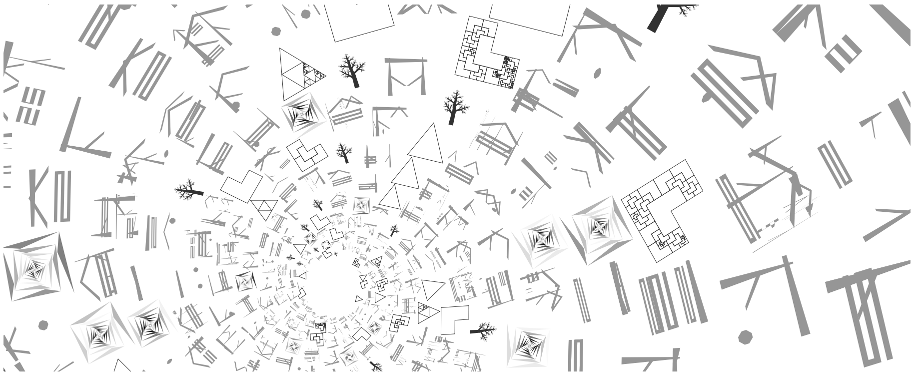

Fig. 6. A spiral with some sub-collages as elements.

is depicted in Figure 6. The overall structure is obtained using the rule in Figure 5. Other rules, each equipped with a certain probability, replace the hyperedges labelled with `any` by hyperedges that give rise to the generation of one of the individual collages the spiral is composed of. The strange looking symbols are generated by a simple collage grammar that arranges five different parts by the use of probabilities.

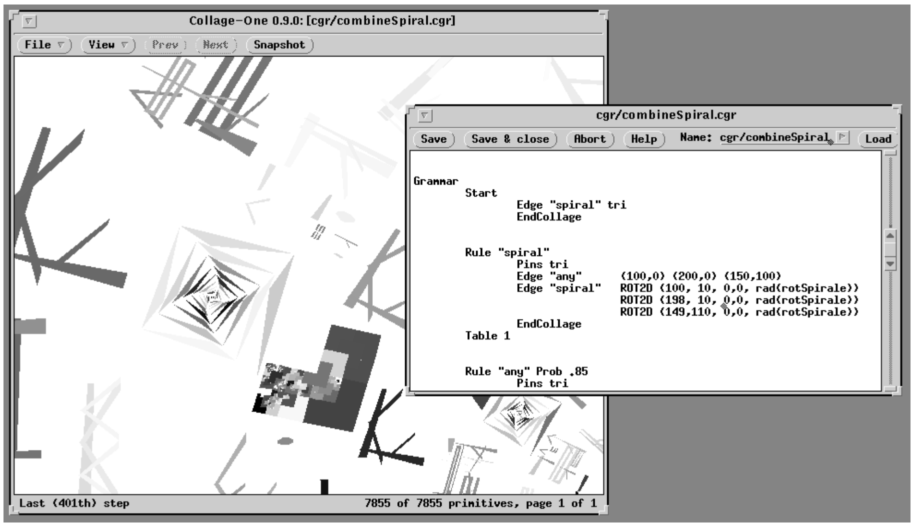

Fig. 7. The COLLAGE-ONE displaying window and textual editor.

The COLLAGE-ONE system as it appears during the development of this example is shown in Figure 7. The program is written in C++ and runs on Sun SPARCstations under Solaris/OpenWindows and also on PCs under the Linux operating system. The implementation is very close to the formal definition of collage grammars; for example, there are C++ classes implementing hyperedges, parts, collages, and grammars. On the right of Figure 7 the textual editor showing part of the currently processed grammar file can be seen. The PostScript output enjoys some additional features which make it particularly well-suited for LaTeX documents. A number of LaTeX commands allow to change the appearance of hyperedges, parts, pin points, etc. in quite a flexible manner.

As mentioned before, in addition to the textual editor a comfortable graphical editor for collages and collage grammars is available. It was developed at the Computer Science Department of the University of Bremen by Jutta Bildstein and Lutz Zegartowski as part of their diploma thesis and allows to edit 3-dimensional as well as 2-dimensional collages and collage grammars. Figure 8 shows the main window of the editor. It consists of a working area containing the collage being composed and a list of icons representing other collages that have already been created.

4. Conclusion

In this paper, we have presented the characteristics of COLLAGE-ONE as a system for evaluation and visualization of collage grammars. It may be worthwhile to relate the

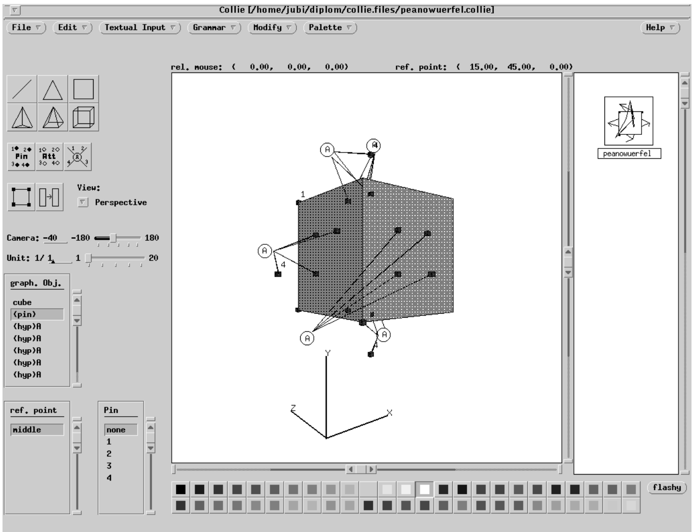

Fig. 8. The graphical collage and collage grammar editor.

concept of collage grammars to other grammatical concepts of picture generation like shape grammars [1, 2] or CP-graph grammars [11]. For a discussion of the relationship of hypergraphs and CP-graphs, which is interesting also from the point of view of collage grammars, see [13, pages 487 and 490].

Currently, several extensions of the system and the theory are investigated. The following are of particular interest in connection with COLLAGE-ONE.

- **BIZARR goes to Hollywood** is a system for the generation of animations based on methods for syntactic picture generation including cellular automata, L- and Map-L-systems, iterated function systems, and collage grammars. It allows the combination of these methods and employs a ray tracer combined with an MPEG generator as its back-ends. The combination of COLLAGE-ONE with this system extends the application spectrum of collage grammars to the production of high quality 3-D animations.
- **Context-sensitive and type-0 grammars:** So far, we have limited ourselves to context-free collage grammars, both in theory and implementation. Context-sensitive collage grammars would provide a way to specify hyperedge replacement that may only occur if a given context is present. Type-0 grammars would also allow to replace parts.

Acknowledgement. We thank the referees for pointing out possible improvements of the paper.

References

1975

[1] J. Gips. *Shape Grammars and Their Uses, Artificial Perception, Shape Generation, and Computer Aesthetics.* Birkhäuser Verlag, Basel/Stuttgart, 1975.

[2] G. Stiny. *Pictorial and Formal Aspects of Shapes and Shape Grammars.* Birkhäuser Verlag, Basel/Stuttgart, 1975.

1978

[3] R.C. Gonzales, M.G. Thomason. *Syntactic Pattern Recognition.* Addison-Wesley, Reading, Mass., 1978.

1982

[4] H.A. Maurer, G. Rozenberg, E. Welzl. Using string languages to describe picture languages. *Information and Control* 54, 155–185, 1982.

1986

[5] D. Hearn, M.P. Baker. *Computer Graphics.* Prentice Hall International, Englewood Cliffs, New Jersey, 2nd edition, 1986.

1988

[6] M. Barnsley. *Fractals Everywhere.* Academic Press, Boston, 1988.

1989

[7] H.S.M. Coxeter. *Introduction to Geometry.* John Wiley & Sons, New York, Wiley Classics Library, second edition, 1989.

1990

[8] P. Prusinkiewicz, A. Lindenmayer. *The Algorithmic Beauty of Plants.* Springer-Verlag, New York, 1990.

1991

[9] A. Habel, H.-J. Kreowski. Collage grammars. In H. Ehrig, H.-J. Kreowski, G. Rozenberg, editors, *Proc. Fourth Intl. Workshop on Graph Grammars and Their Application to Comp. Sci.*, volume 532 of *Lecture Notes in Computer Science*, 411–429. Springer, 1991.

1992

[10] H.-O. Peitgen, H. Jürgens, D. Saupe. *Chaos and Fractals. New Frontiers of Science.* Springer-Verlag, New York, 1992.

1993

[11] E. Grabska. Theoretical concepts of graphical modeling. Part two: CP graph grammars and languages. *Machine GRAPHICS and VISION* 2, 149–178, 1993.

[12] Annegret Habel, Hans-Jörg Kreowski, Stefan Taubenberger. Collages and patterns generated by hyperedge replacement. *Languages of Design* 1, 125–145, 1993.

1994

[13] E. Grabska. Theoretical concepts of graphical modeling. Part three: State of the art. *Machine GRAPHICS and VISION* 3, 481–512, 1994.

1995

[14] F. Drewes, A. Habel, H.-J. Kreowski, S. Taubenberger. Generating self-affine fractals by collage grammars. *Theoretical Computer Science* 145, 159–187, 1995.

1996

[15] F. Drewes, H.-J. Kreowski. (Un)decidability of properties of pictures generated by collage grammars. *Fundamenta Informaticae*, to appear.

[16] F. Drewes. Language theoretic and algorithmic properties of d-dimensional collages and patterns in a grid. *Journal of Computer and System Sciences*, to appear.