A Sketch of Collage Grammars

F. Drewes, A. Habel, H.-J. Kreowski, and S. Taubenberger

Universität Bremen, Fachbereich 3, D–2800 Bremen (Germany)

E-mail: {drewes,habel,kreo}@informatik.uni-bremen.de

Abstract. The aim of this note is to sketch and to illustrate the basic features of collage grammars as a pattern-generating device based on hyperedge replacement.

1 Introduction

A collage consists of a set of parts and a sequence of pin-points. A part may be an arbitrary set of points (in an Euclidean space). Usually, parts are taken from some standard set of geometric objects like circles, triangles, polygons, polyhedra, etc. that have simple finite descriptions and are easy to deal with on graphical surfaces. In applications and future investigations parts may also get colours, textures, and the like. The pin-points are added for technical reasons: they are used to paste collages into collages. The union — also called overlay — of the parts of a collage yields the represented pattern. To generate collages from collages by the application of production rules, they are decorated with hyperedges in intermediate steps. A hyperedge is a labelled item with an ordered number of tentacles each of which is attached to a point. Serving as a place holder it may eventually be replaced by a decorated collage, provided that there is a transformation of the pin-points onto the attachment points of the hyperedge. This kind of hyperedge replacement establishes the rewrite steps of a collage grammar if the label of the replaced hyperedge and the replacing decorated collage form a production. So, a collage grammar generates a set of collages in the usual way of language generation. As collages represent patterns, a set of patterns is generated at the same time. By overlay of all generated patterns and by intersection of them one gets two particular patterns that are called upper and lower generated fractal. Whereas each generated collage and pattern still has a finite geometric description (provided that the right-hand sides of productions have a finite number of parts with finite descriptions), this may no longer be true for generated fractals because the generated languages may be infinite.

The idea of hyperedge replacement stems from the field of graph grammars, where hyperedge replacement is used very successfully as a context-free mechanism to generate graph languages (see, e.g, [Hab92]).

2 Preliminaries

In this section the basic notions and notations concerning collages are recalled (cf. [HK91, HKT93]). 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. Given sets X and Y, $Y - X$ denotes the complement of X in Y, and $X + Y$ denotes the disjoint union of X and Y. For sets X_i, $i \in I$, the disjoint union of the X_i is denoted by $\sum_{i\in I} X_i$. Familiarity with the basic notions of Euclidean geometry (see, e.g., Coxeter [Cox89]) is assumed. $\mathbb{R}$ denotes the set of real numbers and $\mathbb{R}^n$ the Euclidean space of dimension n for some $n \geq 1$. By $dist$ we denote the ordinary distance function $dist: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ on $\mathbb{R}^n$.

As we define below, a collage consists of a set of parts being geometric objects, and a sequence of so-called pin-points. To generate sets of collages, they are decorated with hyperedges in intermediate steps. Each hyperedge has a label and an ordered finite set of tentacles, each of which is attached to a point. A hyperedge is a place holder for a collage or — recursively — for another decorated collage. If it is replaced by a (decorated) collage, the replacing collage is transformed in such a way that the images of its pin-points match the points attached to the hyperedge.

1. A *collage* (in $\mathbb{R}^n$) is a pair $(PART, pin)$ where $PART \subseteq \wp(\mathbb{R}^n)$ is a set of *parts*, each $part \in PART$ being a set of points in $\mathbb{R}^n$, and $pin \in (\mathbb{R}^n)^*$ is a sequence of *pin-points*. The class of all collages is denoted by $\mathcal{C}$.

2. Let N be a set of *labels*. A *(hyperedge-)decorated collage* (over N) is a construct $C = (PART, EDGE, att, lab, pin)$ where $(PART, pin)$ is a collage, $EDGE$ is a set of *hyperedges*, $att : EDGE \to (\mathbb{R}^n)^*$ is a mapping, called the *attachment*, and $lab : EDGE \to N$ is a mapping, called the *labelling*. C is said to be *finite* if $PART$ and $EDGE$ are finite sets. The class of all decorated collages over N is denoted by $\mathcal{C}(\mathcal{N})$.

3. The components $PART$, $EDGE$, att, lab, and pin of a decorated collage C are also denoted by $PART_C$, $EDGE_C$, att_C, lab_C, and pin_C, respectively.

4. A collage can be seen as a decorated collage C without hyperedges, i.e., $EDGE_C = \emptyset$ and att_C as well as lab_C being the empty mappings. In this sense, $\mathcal{C} \subseteq \mathcal{C}(\mathcal{N})$. In the description of decorated collages without hyperedges, we will drop the components $EDGE_C$, att_C, and lab_C.

5. Two decorated collages $C, C' \in \mathcal{C}(\mathcal{N})$ are said to be *isomorphic*, denoted by $C \cong C'$, if the underlying collages of C and C' are equal and if there is a bijective mapping $f : EDGE_C \to EDGE_{C'}$ such that $att_C(e) = att_{C'}(f(e))$ and $lab_C(e) = lab_{C'}(f(e))$ for all $e \in EDGE_C$.

While a hyperedge in a decorated collage is attached to some points according to our conventions, a (decorated) collage has got some pin-points. If there is a transformation that maps these pin-points to the attached points of the hyperedge, the hyperedge may be replaced by the transformed (decorated) collage. To define hyperedge replacement formally, we use three simple constructions on decorated collages: hyperedge removal, transformation, and addition.

1. **Removal** Let $C \in \mathcal{C}(\mathcal{N})$ and $B \subseteq EDGE_C$. Then the *removal* of B from C yields the decorated collage $C - B = (PART_C, EDGE_C - B, att, lab, pin_C)$ with $att(e) = att_C(e)$ and $lab(e) = lab_C(e)$ for all $e \in EDGE_C - B$.

2. **Transformation** Let $C \in \mathcal{C}(\mathcal{N})$ and let $t: \mathbb{R}^n \to \mathbb{R}^n$ be a mapping which will be referred to as a transformation. Then the *transformation* of C by t yields the decorated collage $t(C) = (t(PART_C), EDGE_C, att, lab_C, t(pin_C))$ with $att(e) = t(att_C(e))$ for all $e \in EDGE_C$.[1]

3. **Addition** Let $C \in \mathcal{C}(\mathcal{N})$ and $Y \subseteq \mathcal{C}(\mathcal{N})$. Then *adding* Y to C yields the decorated collage $C + Y = (PART_C \cup \bigcup_{R \in Y} PART_R, EDGE_C + \sum_{R \in Y} EDGE_R, att, lab, pin_C)$ with $att(e) = att_C(e)$ and $lab(e) = lab_C(e)$ for all $e \in EDGE_C$ and $att(e) = att_R(e)$ and $lab(e) = lab_R(e)$ for all $e \in EDGE_R$, $R \in Y$. (Note that $C + Y$ keeps the pin-points of C.)

4. **Replacement** Let $TRANS$ be a set of transformations. Let $C \in \mathcal{C}(\mathcal{N})$, $B \subseteq EDGE_C$, and let $(repl, trans)$ be a pair of mappings $repl: B \to \mathcal{C}(\mathcal{N})$, $trans: B \to TRANS$ with $att(e) = trans(e)(pin_{repl(e)})$ for all $e \in B$. Then the *replacement* of B in C through $(repl, trans)$ yields the decorated collage $REPL(C, repl, trans) = (C-B)+Y(B)$ where $Y(B) = \sum_{e \in B}\{trans(e)(repl(e))\}$ denotes the set of decorated collages determined by $(repl, trans)$.

In an intuitive sense hyperedge replacement works by removing some hyperedges, transforming associated decorated collages in such a way that the images of the pin-points match the points attached to the corresponding hyperedges, and adding the transformed decorated collages. Note that the pin-points may restrict the choice of possible transformations. Note also that the pin-points of the replacing collages loose their status. Examples that illustrate hyperedge replacement are given in the next section.

3 Collage grammars

Based on hyperedge replacement as introduced in the previous section, one can derive (decorated) collages from decorated collages by applying productions consisting of a label $A \in N$ and a decorated collage $R \in \mathcal{C}(\mathcal{N})$. Such a production may be applied to a hyperedge e with label A provided that there is a transformation from a given set $TRANS$ of admissible transformations that maps the pin-points of R to the attached points of e. The result of the application is obtained by replacing the hyperedge by the transformed image of R. More generally, several productions may be applied in parallel. Besides the ordinary notions of a derivation and a generated language two notions of generated fractals are introduced by the overlay and the intersection of all induced patterns of the generated language.

[1]A transformation $t: \mathbb{R}^n \to \mathbb{R}^n$ has the following natural extensions:

- $t: \wp(\mathbb{R}^n) \to \wp(\mathbb{R}^n)$ by $t(part) = \{t(x) \mid x \in part\}$ for all $part \subseteq \mathbb{R}^n$,
- $t: \wp(\wp(\mathbb{R}^n)) \to \wp(\wp(\mathbb{R}^n))$ by $t(PART) = \{t(part) \mid part \in PART\}$ for all $PART \subseteq \wp(\mathbb{R}^n)$,
- $t: (\mathbb{R}^n)^* \to (\mathbb{R}^n)^*$ by $t(x_1 \cdots x_m) = t(x_1) \cdots t(x_m)$ for all $x_i \in \mathbb{R}^n$, $i = 1, \ldots, m$.

3.1 General assumption

For the rest of this note, let $TRANS$ be a set of transformations $t: \mathbb{R}^n \to \mathbb{R}^n$.

Remark. Various sets of transformations may be considered, e.g., the set of *isometries*, the set of *central dilatations*, the set of *similarity transformations*, or the set of *affine transformations*. Remember that a mapping $t: \mathbb{R}^n \to \mathbb{R}^n$ is said to be an *isometry* if for any $x, y \in \mathbb{R}^n$ we have $dist(t(x), t(y)) = dist(x, y)$. In $\mathbb{R}^2$, there are four basic types of isometries: translations, rotations, reflections, and glide reflections. A mapping $t: \mathbb{R}^n \to \mathbb{R}^n$ is said to be a *similarity transformation* (with *magnification factor* $c > 0$) if, for all $x, y \in \mathbb{R}^n$, $dist(t(x), t(y)) = c \cdot dist(x, y)$. A similarity can be accomplished in two stages: first a central dilatation (to make objects the right size) and then an isometry (to move objects to the right position). A similarity is a special case of an affine transformation which is composed of a non-singular linear transformation and a translation. For more details about geometrical transformations see, e.g., Coxeter [Cox89].

3.2 Definition (Productions and Derivations)

1. Let N be a set of labels. A *production* (over N) is a pair $p = (A, R)$ with $A \in N$ and $R \in \mathcal{C}(\mathcal{N})$. A is called the *left-hand side* of p and is denoted by $lhs(p)$. R is called the *right-hand side* and is denoted by $rhs(p)$.

2. Let $C \in \mathcal{C}(\mathcal{N})$, $B \subseteq EDGE_C$, and P be a set of productions (over N). Then a pair $(prod, trans)$ of mappings $prod: B \to P$ and $trans: B \to TRANS$ is called a *base* on B in C if $lab_C(e) = lhs(prod(e))$ and $att_C(e) = trans(e)(pin_{rhs(prod(e))})$ for all $e \in B$, and if $trans$ is the only transformation in $TRANS$ with the latter property.

3. Let $C, C' \in \mathcal{C}(\mathcal{N})$ and $(prod, trans)$ be a base on B in C. Then C *directly derives* C' *through* $(prod, trans)$ if $C' \cong REPL(C, repl, trans)$, where for all $e \in B$ we have $repl(e) = trans(e)(rhs(prod(e)))$. A direct derivation is denoted by $C \Longrightarrow C'$ or $C \Longrightarrow_P C'$.

4. A sequence of direct derivations of the form $C_0 \Longrightarrow_P C_1 \Longrightarrow_P \cdots \Longrightarrow_P C_k$ is called a *derivation* from C_0 to C_k and is denoted by $C_0 \Longrightarrow_P^* C_k$ or $C_0 \Longrightarrow^* C_k$.

Remark. The uniqueness requirement for $trans$ in Definition 3.2(2) makes sure that the result of applying a production to a hyperedge is uniquely determined.

Using the introduced concepts of productions and derivations, collage grammars and collage languages can be introduced in the usual way. Since collages represent patterns and the overlay as well as the intersection of patterns yields a pattern, collage grammars also specify fractal patterns.

3.3 Definition (Collage Grammars and Languages)

1. A *collage grammar* is a system $CG = (N, P, Z)$ where N is a finite set of *nonterminals*, P is a finite set of *productions* (over N) with finite right-hand sides, and $Z \in \mathcal{C}(\mathcal{N})$ is a finite decorated collage, called the *axiom*.

2. The *collage language generated by* CG consists of all collages which can be derived from Z by applying productions of P:

$$L(CG) = \{C \in \mathcal{C} \mid \mathcal{Z} \overset{*}{\underset{\mathcal{P}}{\Longrightarrow}} \mathcal{C}\}.$$

3. The *upper fractal pattern* generated by CG is given by the overlay of all generated collages:

$$fractal_u(CG) = \bigcup_{C \in L(CG)} \; \bigcup_{part \in PART_C} part.$$

4. The *lower fractal pattern* generated by CG is given by the intersection of all generated collages:

$$fractal_l(CG) = \bigcap_{C \in L(CG)} \; \bigcup_{part \in PART_C} part.$$

In the next sections we will see three examples of collage grammars and the correponding generated collages.

4 Examples

We now present some examples that we hope illustrate the concept of collage grammars.

4.1 Carpets

Figure 1 depicts the axiom of the collage grammar $carpet = (\{\mathsf{C}\}, \{p^1_{car}, p^2_{car}\}, Z_{car})$. Its

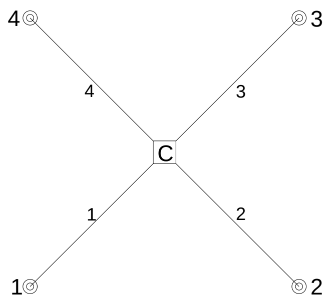

Figure 1: The axiom of the grammar *carpet*.

productions are shown in Figure 2 in a kind of Backus-Naur form. Both right-hand sides contain as parts the boundaries of seven rectangles of different size that intersect only in their boundaries and that (incompletely) divide the square defined by the pin-points. The right-hand side of p^1_{car} contains seven hyperedges in addition (labelled with C), each of which spans one of the rectangular parts.

The derivations in this grammar rely on the set of all affine transformations in two-dimensional space, $AFF(2)$. One of the possible derivations is shown in Figure 3. Each

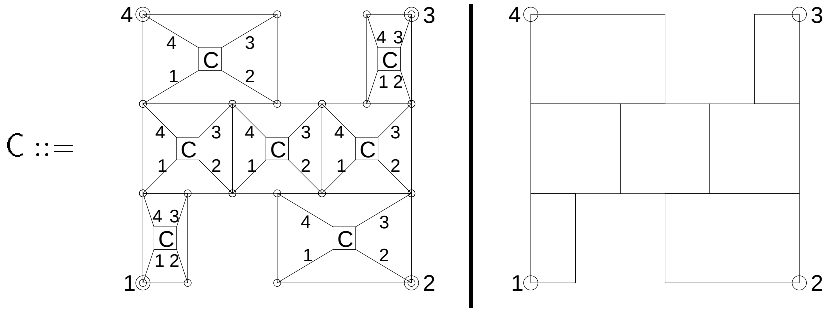

Figure 2: The productions of the grammar *carpet*.

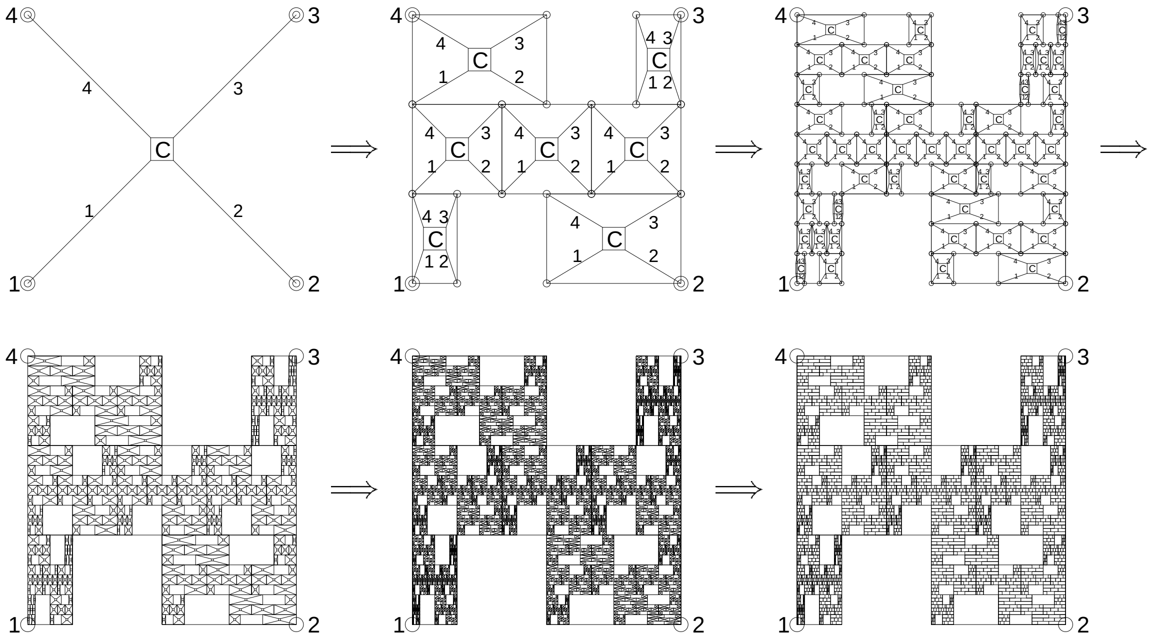

Figure 3: A derivation in *carpet*.

step in this derivation is maximum parallel, that is, in each step all hyperedges are replaced. Moreover, only one of the productions is used in every step. Beginning with the axiom Z_{car}, the first production is applied to the single initial hyperedge, yielding the right-hand side of the production. Hence the resulting decorated collage has seven hyperedges, which are affinely transformed images of the axiom. The first production is now applied again, by using affine transformations that map the pin-points of its right-hand sides to the attached nodes of the respective hyperedges. In the last step, p^2_{car} is applied to all hyperedges, yielding the final result of the derivation.

The collages generated by *carpet* are approximations of $fractal_u(carpet)$, which in turn is a variant of the well-known Sierpinski carpet.

4.2 Sierpinski triangle

As another example, consider the collage grammar $triangle = (\{\mathsf{T}\}, \{p^1_{tri}, p^2_{tri}\}, Z_{tri})$, whose axiom and productions are depicted in Figures 4 and 5. The right-hand side of the first production does not contain any parts. Its hyperedges span rectangles whose edges are half the size of the edges of the rectangle given by the pin-points. Two of them occupy the lower-left and -right quarters of that rectangle and the third one is placed in the middle of its top row.

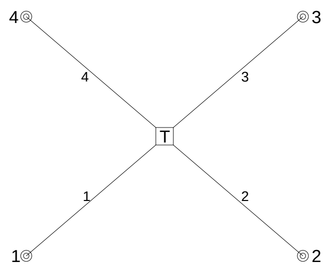

Figure 4: The axiom of the grammar *triangle*.

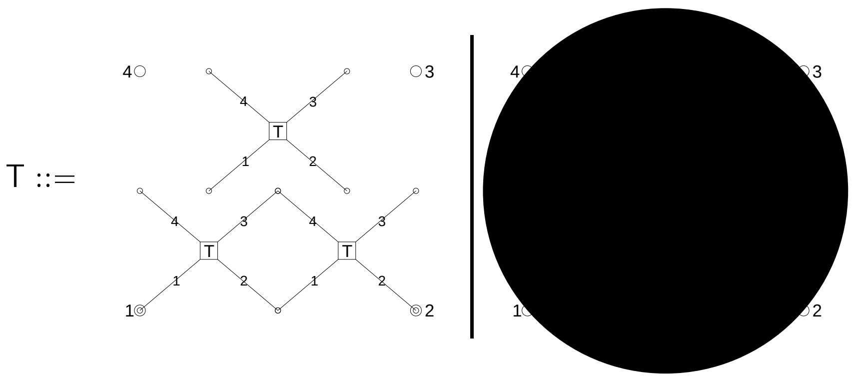

Figure 5: The productions of the grammar *triangle*.

The first nine collages generated by maximum parallel derivations of length one up to nine are shown in Figure 6. Obviously, the lower fractal generated by *triangle*, i.e., the intersection of all the generated collages, is the Sierpinski triangle.

Note that there is no explicit triangle used in the grammar at all. The only reference to a triangular structure lies in the relative positions of the hyperedges in p^1_{tri}.

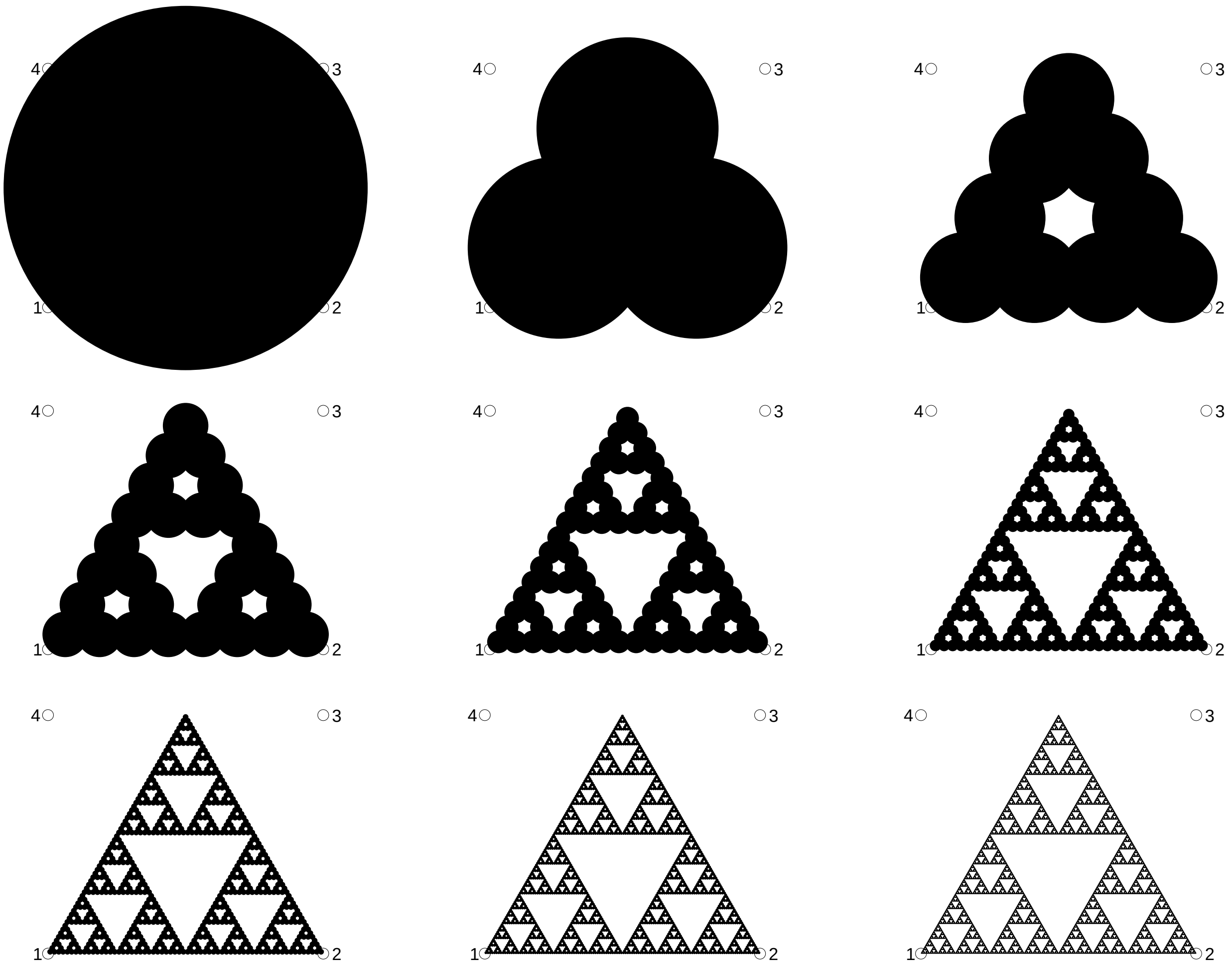

Figure 6: Some collages generated by the grammar *triangle*.

4.3 Bullets

As a last example, consider the grammar $bullets = (\{\mathsf{B}\}, \{p^1_{bu}, p^2_{bu}\}, Z_{bu})$, with productions as shown in Figure 7. (Its axiom is a single hyperedge labelled with B that spans a square.) The two hyperedges in the right-hand side of the second production are slightly rotated and span a square half the size of the one given by the pin-points.

Again, we show the first collages generated by maximum parallel derivations (see Figure 8). For this grammar the upper fractal is the interesting one: it is the limit of the sequence of collages generated by maximum parallel derivations of increasing length.

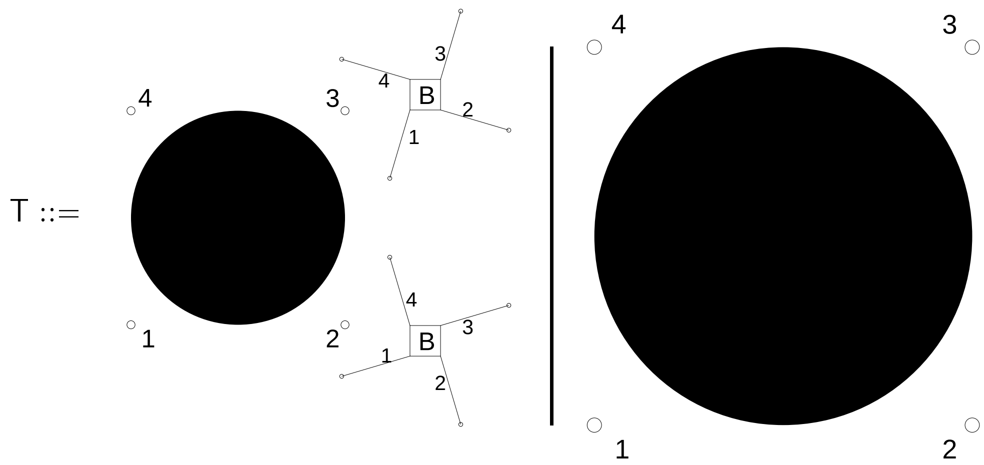

Figure 7: The productions of the grammar *bullets*.

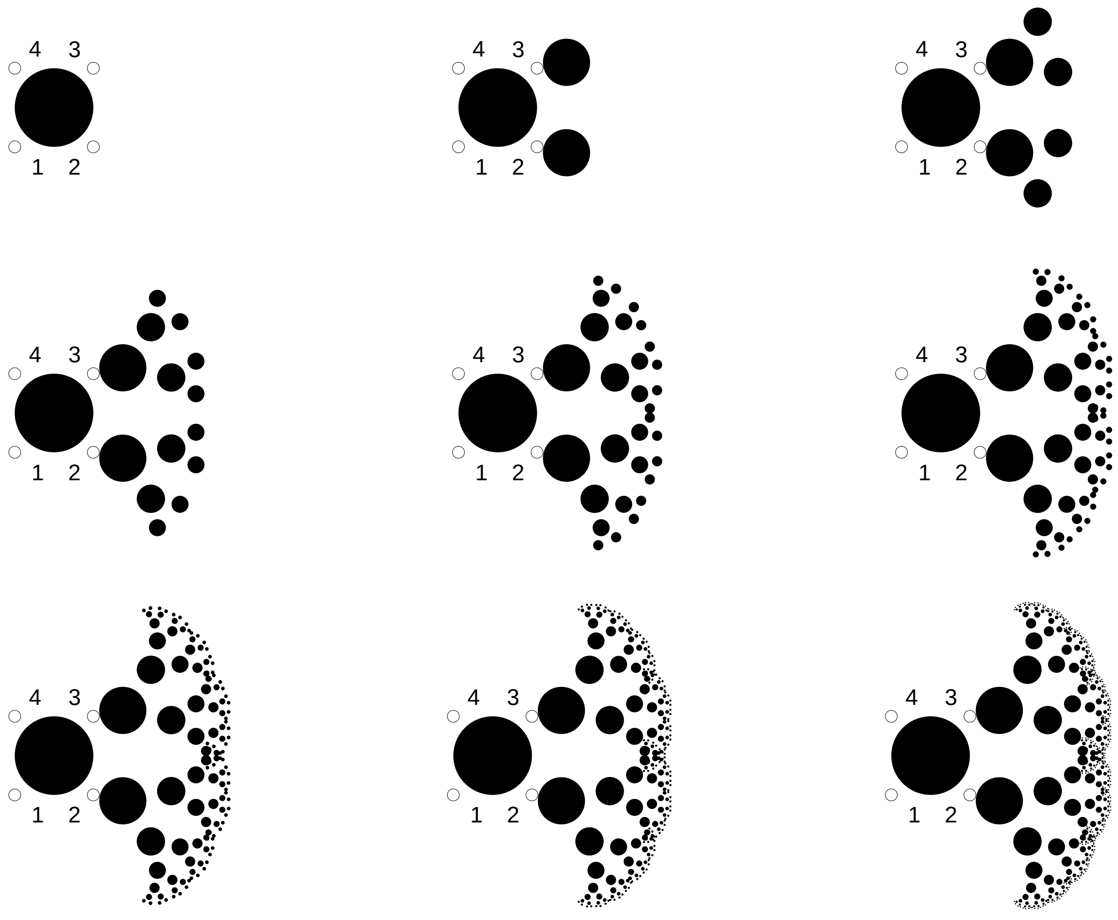

Figure 8: Some collages generated by the grammar *bullets*.

5 Conclusion

We conclude this introductory note about collage grammars by summarizing some of the properties of this generating mechanism. We also give the respective references to the literature.

In [HK91, HKT93] collage grammars (invented by Habel and Kreowski in [HK88]) are studied with main emphasis on context-freeness and fixed-point results. It is shown that collage grammars are context-free in much the same sense as hyperedge-replacement graph grammars (cf. [Hab92]) are. Due to context-freeness, every derivation in a collage grammar that starts with a single hyperedge can be divided into the first step (yielding the right-hand side of the applied production) and the subderivations originating from the hyperedges in the right-hand side. Then the replacement of these hyperedges by the generated collages of the subderivations yields the same result as the initial derivation. (Actually, in order to obtain this result we have to consider *proper* collage grammars. Mainly, a proper collage grammar is one in which the uniqueness requirement formulated in Definition 3.2(2) is always satisfied.)

The context-freeness result yields also an equational characterization of the generated languages as fixed-points of sets of equations. A similar fixed-point result is obtained for the generated upper fractal.

In an extended version of this note ([DHKT93], to be presented on the *Developments in Formal Language Theory* conference in Turku, Finland) the relations between upper and lower fractals of collage grammars on the one hand and self-affinity on the other are investigated. Especially, the so-called *increasing* and *decreasing* Sierpinski grammars — special collage grammars — are studied. It turns out that these generate self-affine upper respectively lower fractals. More precisely, the upper fractals of increasing Sierpinski grammars and the lower fractals of decreasing ones are self-affine. One example is the grammar for the Sierpinski triangle shown in the previous section. It is furthermore proved that every self-affine fractal can be obtained as the lower fractal of a suitably chosen decreasing Sierpinski grammar. What is more, the grammar constructed to show this result contains only collages with very simple parts. Hence these collage grammars can be seen as handy descriptions of such complex geometric objects like self-affine fractal images.

Future studies of collage grammars should include the following topics (among others):

(1) Overlay and intersection, as used in the definition of the upper and lower fractal are somewhat rough ways to get fractals from potentially infinite sets of patterns. More sophisticated limit constructions may yield different results.

(2) The results for lower fractals of decreasing Sierpinski grammars seem to be related to results in the area of so-called iterated function systems (see, e.g., Barnsley [Bar88]). This connection should be investigated.

(3) A comparison of collage grammars with other syntactic approaches to fractal geometry, like the work by Culik II and Dube [CD90, CD91], would be interesting.

(4) Moreover, one should compare collage grammars with other syntactic devices for the generation of pictures, like chain code picture languages (see, e.g., Maurer, Rozen-

berg, and Welzl [MRW82], Dassow [Das89], Kim [Kim90]), graphical interpretation of L-systems (see, e.g., Prusinkiewicz and Lindenmayer [PL90]) and cellular automata (see, e.g., Toffoli and Margolus [TM88]).

References

[Bar88] M. Barnsley. *Fractals Everywhere.* Academic Press, Boston, 1988.

[CD90] K. Culik II, S. Dube. Affine automata and related techniques for generation of complex images. *Lecture Notes in Computer Science* 452, 224–231, 1990. To appear in Theoretical Computer Science.

[CD91] K. Culik II, S. Dube. Balancing order and chaos in image generation. *Lecture Notes in Computer Science* 510, 600–614, 1991.

[Cox89] H.S.M. Coxeter. *Introduction to Geometry.* Edition, Wiley Classics Library Edition, John Wiley & Sons, Inc., New York, second edition, 1989.

[Das89] J. Dassow. Graph-theoretic properties and chain code picture languages. *Journ. Inform. Process. Cybern. EIK.* 25, 423–433, 1989.

[DHKT93] F. Drewes, A. Habel, H.-J. Kreowski, S. Taubenberger. Generating self-affine fractals by collage grammars, 1993. To appear.

[Hab92] A. Habel. *Hyperedge Replacement: Grammars and Languages*, volume 643 of *Lecture Notes in Computer Science.* Springer, 1992.

[HK88] A. Habel, H.-J. Kreowski. Pretty patterns produced by hyperedge replacement. In H. Göttler, H.J. Schneider, editors, Graph-Theoretic Concepts in Computer Science, *Lecture Notes in Computer Science* 314, 32–45, 1988.

[HK91] 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., *Lecture Notes in Computer Science* 532, 411–429, 1991.

[HKT93] A. Habel, H.-J. Kreowski, S. Taubenberger. Collages and patterns generated by hyperedge replacement. *Languages of Design* 1, 125–146, 1993.

[Kim90] C. Kim. Picture iteration and picture ambiguity. *Journal of Computer and System Sciences* 40, 289–306, 1990.

[MRW82] H.A. Maurer, G. Rozenberg, E. Welzl. Using string languages to describe picture languages. *Information and Control* 54, 155–185, 1982.

[PL90] P. Prusinkiewicz, A. Lindenmayer. *The Algorithmic Beauty of Plants.* Springer-Verlag, New York, 1990.

[TM88] T. Toffoli, N. Margolus. *Cellular Automata Machines.* MIT Press, Massachusetts, 1988.