Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2005), *19*, 261–276. Printed in the USA.

DOI: 10.1017/S0890060405050183

Shape decompositions and their algebras

DJORDJE KRSTIC
Alcatel, Calabasas, California 91302, USA
(Received October 16, 2004; Accepted May 18, 2005)

Abstract

Shapes play an important role in many human activities, but are rarely seen in their natural form as raw and unanalyzed. Rather, shapes come analyzed, structured in terms of their certain parts, forming shape decompositions. Different kinds of shape decompositions are developed, the most interesting among which are the decompositions that could be used as shape approximations. Two kinds of such decompositions, *discrete* and *bounded*, are examined in greater detail. Computations with shapes conducted in the framework of shape grammars and related shape algebras have been standard for over 3 decades. Similar computations are possible with analyzed shapes or shape decompositions. Different algebras to compute with shape decompositions are developed and compared to the shape algebras. The measure of their agreement determines how well the shapes are approximated by their decompositions.

Keywords: Shape Algebras; Shape Approximations; Shape Decompositions; Shape Decompositions Algebras; Shape Grammars

1. INTRODUCTION

Shapes are part of our everyday experience and play important roles in many human activities. Shapes come without apparent structure, therefore rendering any division into parts possible. However, any attempt to (verbally) describe a shape inevitably leads to structuring the shape in terms of its certain parts or to a shape *decomposition*. According to Stiny (1991), "These are defined in a variety of ways, but most simply as sets of shapes that show how their sums are divided into parts of certain kinds."

For example, the 3-dimensional shape in Figure 1a may be described as a *table* having *four legs* and a *board*, which leads to the decomposition (Fig. 1b).

The fact that we can name things around us structures our surroundings in a shape decomposition; whereas the simple naming of a shape structures it by recognizing the shape as being a part of itself.

For example, the original shape in Figure 1a is called a *table* in the description above. It should also be included in the decomposition so that all of the shapes in Figure 1 become elements of the latter.

Reprint requests to: Djordje Kristic, Alcatel, 22415 Kearny Street, Calabasas, CA 91302, USA. E-mail: George.Krstic@alcatel.com

Shapes are rarely perceived in their natural form as raw/unanalyzed, but rather approximated by their decompositions. An unknown shape may at first appear as raw/unanalyzed, but it gradually acquires a structure as one tries to understand it. As our understanding grows, we move from raw/unanalyzed shapes to structured/decomposed shapes.

Although important in many human activities, shapes are prime to design and fine arts. Designers use shapes to specify things to make: to provide information on how things function or how they should be built. To do so, designers differentiate some parts of their designs by naming and relating the parts. Shapes in designs are utilitarian in nature. They come structured in decompositions with ambiguities removed in order to avoid misunderstanding.

Ambiguities, although dismissed in designs, may well be at home in fine arts. Shapes of art may serve a different purpose than those of design. They need not be rationally understood. An art object may appear pure, unanalyzed, and open to any kind of interpretation by both the public and critics. This kind of ambiguity allows for multiple interpretations and adds to the richness of the object.

Although the outcomes of a designer's or artist's efforts may be different in nature (analyzed vs. raw shape), the design process may well be the same.

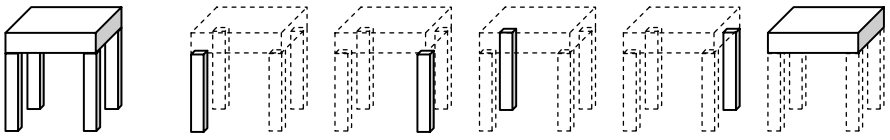

Fig. 1. Decomposition of a table: (a) the original shape and (b) its decomposition.

In the course of design, an unanalyzed shape may be seen in a certain way and accordingly decomposed into a certain set of parts. These may then be used to compose new shapes. The latter may again be seen as unanalyzed, only to be decomposed in some new way leading to new sets of parts. The process may be repeated until a design or an art object is created. Moving from analyzed to raw shapes creates an opportunity to see things in a different way, which then moves shapes back from raw to analyzed. The creative phases of the design process are characterized by shifts between raw/unanalyzed shapes and decomposed/analyzed ones. Stiny's (1994, 2001) shape computations with unexpected outcomes are good examples of such shifts at work.

Decompositions are used to describe shapes. They often appear in place of shapes functioning as their approximations. The purpose of this paper is to investigate how good such approximations are.

The investigation is conducted in the context of formal design theory that involves shape grammars and related shape algebras. Central to the theory are shape computations that attempt to formalize what designers do when they design. To use decompositions as shape approximations, tools for computing with decompositions are developed. In particular, algebras for decompositions similar to algebras for shapes are introduced; and the properties of such algebras are investigated and compared to the properties of shape algebras. The measure of their agreement determines how computations with decomposition may compare to the similar computations with shapes. This ultimately shows how well shapes are approximated by their decompositions.

1.1. A note on notation

Symbols like $\varnothing$, $\cap$, $\cup$, $\subseteq$, $\subset$, $\in$, $\notin$, and $\wp$ are used in their standard set-theory meaning: empty set, intersection, union, subset, proper subset, element of, not an element of, and power set, respectively. Some standard set-theory notions are assumed, such as the ordered set with related maximal and minimal elements, bounds, and closures, as well as the direct product and related functions and relations.

Computations are carried out in different *algebras*, where an algebra is seen as a set of objects that is closed under a set of operations. The elements of an algebra may satisfy certain equations, like distributive or associative laws, which are *axioms* of that algebra. Axioms may serve to distinguish classes of algebras such as rings, lattices, groups, Boolean algebras, and so forth.

New algebras, in particular subalgebras and quotient algebras, are constructed from the old ones. The former are subsets of an algebra that are algebras themselves. The latter are algebras whose elements are equivalence classes of some algebra, which has been partitioned with respect to an equivalence relation of a certain kind: *congruence*. Congruence preserves the operations of the algebra so that the result of computations carried on with the equivalence classes of some elements of the algebra is the equivalence class of the result of the same computation carried on with the elements themselves.

Algebras with the same kinds of operations could be related via certain mappings. If there is a mapping from an algebra to another algebra such that the mapping enumerates all of the elements of the latter algebra, and if it also preserves the operations of the algebras, then the latter algebra is *homomorphic* to the former one. If the mapping is also one to one, then the two algebras are *isomorphic*. A mapping defining isomorphism is a *surjection* because it enumerates all of the elements of the latter algebra and it is an *injection* because it is one to one; thus, it is a *bijection*.

We also make use of shape algebras (Stiny, 1991, 1992, 2001). These compute with shapes in a Boolean fashion and are also closed under geometric transformations that act on shapes. Shape algebras are seen as two-sorted (operating on objects of two different sorts), conveniently keeping both shapes and transformations in a single algebraic structure (Krstic, 1996, 1999, 2001). Shape algebra $(\boldsymbol{U_{ij}})$ computes with i-dimensional shapes occupying j-dimensional space forming the set U_{ij} $(i, j = 0, 1, 2, 3,$ and $i \leq j)$ and j-dimensional geometric transformations forming the set T_j.

The set of shapes U_{ij} has a lower bound, which is the empty shape denoted by 0, but has no upper bound. Shapes from U_{ij}, together with Boolean operations of sum $(+)$, product $(\cdot)$, and difference $(-)$, form a *generalized Boolean algebra*: a relatively complemented distributive lattice with a smallest element (Birkhoff, 1993). This establishes the *Boolean part* of $\boldsymbol{U_{ij}}$.

Transformations of T_j form a group establishing the *group part* of $\boldsymbol{U_{ij}}$. Transformations are combined with the aid of group operations and act on shapes of U_{ij} by a binary operation of group action that takes a shape $\boldsymbol{a}$ and a transformation $\boldsymbol{t}$ to produce a transformed shape $\boldsymbol{t}(\boldsymbol{a})$.

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

Shape algebras could be extended to other design-related objects such as labeled shapes, weighted shapes, or n-tuples of shapes. Such algebras still have the shape algebra structure, but the objects they operate on are not shapes.

1.2. Definition

Decompositions are used as shape descriptions that emphasize certain properties of shapes. There are different ways to define a decomposition of a shape depending mainly on the kind of information that is to be highlighted.

For example, decompositions defined as directed graphs containing shapes and spatial relations among them expose the details of how shapes are generated with a shape grammar (Knight, 1988).

Decompositions defined simply as sets of shapes will be used exclusively in this work. Although very elementary, the definition is general. Such decompositions do only the minimum of what decompositions should do: they structure unanalyzed shapes in terms of their certain parts and nothing else. That is, certain properties (parts) of a shape are chosen to represent the shape while others are neglected. This renders decompositions: shape approximations that suit certain purposes.

Shape parts are central to decompositions, and any shape, with the exception of shapes consisting of points, has infinitely many parts. A picture is literally worth *infinitely many* words. The set of all parts of a shape is the upper bound for all decompositions of that shape. This set has some interesting properties that reveal shape structure.

Let $B(\boldsymbol{a})$ denote the set that includes all of the parts of a shape $\boldsymbol{a}$. Any Boolean combination of parts of $\boldsymbol{a}$ carried out in the algebra in which $\boldsymbol{a}$ is defined yield another part of $\boldsymbol{a}$. Set $B(\boldsymbol{a})$ is a subalgebra of this algebra closed under its Boolean operations. If $\boldsymbol{a}$ is a shape from $\boldsymbol{U_{ij}}$, then $B(\boldsymbol{a})$ is a Boolean algebra (Stiny, 1980; Earl, 1997). Because $\boldsymbol{a}$ is not changed by the transformations of its symmetry group $S(\boldsymbol{a})$, the set $B(\boldsymbol{a})$ is closed under $S(\boldsymbol{a})$ in accordance with the following proposition.

PROPOSITION 1. *Set $B(\boldsymbol{a})$ is closed under the symmetry group of $\boldsymbol{a}$, or $S(\boldsymbol{a})$.* ■

The proof follows from the definition of the symmetry group of a shape and the order preserving nature of transformations. That is, if $\boldsymbol{t} \in S(\boldsymbol{a})$, than $\boldsymbol{t}(\boldsymbol{a}) = \boldsymbol{a}$, and if $\boldsymbol{x} \leq \boldsymbol{a}$, then $\boldsymbol{t}(\boldsymbol{x}) \leq \boldsymbol{t}(\boldsymbol{a})$. Consequently, $\boldsymbol{t}(\boldsymbol{x}) \leq \boldsymbol{a}$ so that $\boldsymbol{t}(\boldsymbol{x}) \in B(\boldsymbol{a})$.

This motivates the following definition: the *maximal structure of a shape* $\boldsymbol{a}$ denoted by $\boldsymbol{B}(\boldsymbol{a})$ is a two-sorted shape algebra with $B(\boldsymbol{a})$ as the Boolean part and $S(\boldsymbol{a})$ as the group part.

The maximal structure of $\boldsymbol{a}$ is clearly a proper subalgebra of $\boldsymbol{U_{ij}}$, which shows that shapes, like their algebras, have a dual nature. They have parts that are shapes from U_{ij} closed under transformations from T_j, where $B(\boldsymbol{a})$ and $S(\boldsymbol{a})$ are respective upper bounds. The interplay of structure and symmetry, which is central to design, emerges here in the very nature of the building blocks of designs: shapes.

Although it is possible to have decompositions containing infinitely many parts of a shape, like $B(\boldsymbol{a})$, only finite decompositions will be examined in this work. The former may be appealing to mathematicians, but the latter are central to the design practice.

DEFINITION 1. A finite nonempty set of shapes is a *decomposition*. ■

The set of all decompositions denoted by Δ_{ij} is a proper subset of $\wp(U_{ij})$. Set Δ_{ij} is clearly without an upper bound when ordered by $\subseteq$. It also lacks a lower bound, because the empty set is not a decomposition.

DEFINITION 2. Let $\boldsymbol{a}$ be a shape, the set A is a *decomposition of* $\boldsymbol{a}$ whenever it is a decomposition and the sum of its elements is $\boldsymbol{a}$, or $\boldsymbol{a} = \Sigma A$. ■

Any decomposition is a decomposition of a shape because finite sums of shapes are guaranteed to be shapes.

The set of finite subsets of $B(\boldsymbol{a})$ that sum to $\boldsymbol{a}$ is the set of all decompositions of $\boldsymbol{a}$. Like Δ_{ij}, this set does not have an upper or a lower bound and is closed under the symmetry group of $\boldsymbol{a}$.

Decompositions structure shapes, and also structure their parts. In Figure 2a, the square is structured as a consequence of the structure of the double square in Figure 2b that contains it. Note that the location of the shapes in some coordinate system is indicated by cross marks.

The structure of an analyzed shape can be relativized to each of its parts in accordance with the following definition.

DEFINITION 3. If $\boldsymbol{a}$ is a shape, A its decomposition, and shape $\boldsymbol{b}$ a part of $\boldsymbol{a}$, then $\boldsymbol{b}$ is implicitly structured in decomposition $B = \{\boldsymbol{b} \cdot \boldsymbol{x} \mid \boldsymbol{x} \in A\}$, which is the *relativization* of the structure of $\boldsymbol{a}$ to $\boldsymbol{b}$ or the relativization of A to $\boldsymbol{b}$. ■

2. ALGEBRAS OF DECOMPOSITIONS

Computations with decompositions may be carried out in a similar way as computations with shapes. For example, the shapes in Figure 3a and b, when combined in the framework of $\boldsymbol{U_{12}}$, yield the shapes in Figure 3c, d, and e, which are their respective sum, product, and difference. It should

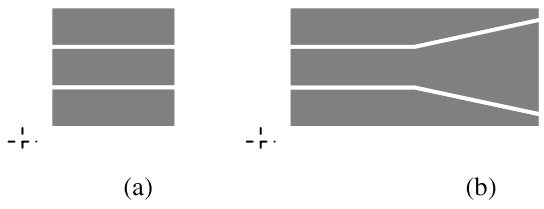

Fig. 2. (a,b) Decompositions: the structure of (b) is recognized in (a) whereas its elements are not.

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

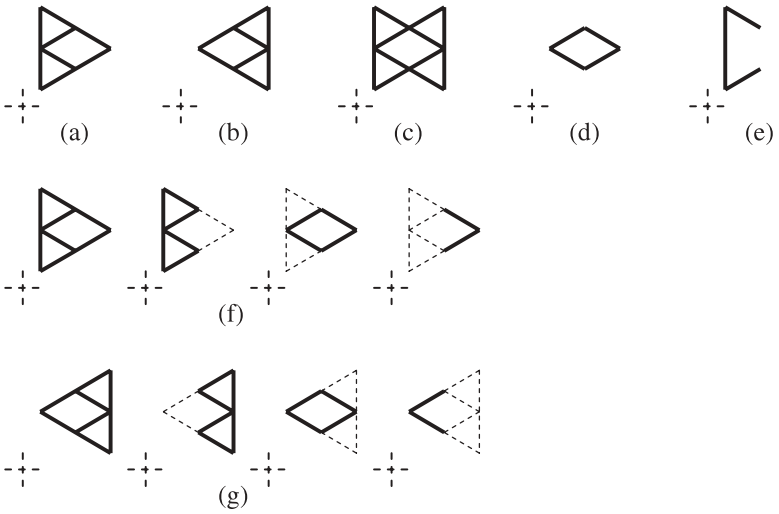

Fig. 3. Computing in $\boldsymbol{U_{12}}$: (a, b) shapes and their (c) sum, (d) product, and (e) difference, as well as their respective (f, g) decompositions.

be possible to duplicate these computations with the sets in Figure 3f and g, which are respective decompositions of the shapes in Figure 3a and b.

Computations with decompositions should be carried out in the framework of appropriate algebras belonging to the same family as shape algebras $\boldsymbol{U_{ij}}$. Such an algebra should be two-sorted, having decompositions as well as transformations that act on decompositions. The operations should include sum, product, and difference for decompositions, group operations for transformations, and a group action relating the two. More formally, an algebra for decompositions should satisfy the following general definition.

Definition 4. An algebra for decompositions is an extended shape algebra. Its Boolean part is a set of decompositions $X \subseteq \Delta_{ij}$, whereas its group part is a group of transformations that can act on shapes. The group action is extended to decompositions so that $\boldsymbol{t}(A) = \{\boldsymbol{t}(\boldsymbol{x}) | \ \boldsymbol{x} \in A\}$, where A is a decomposition and $\boldsymbol{t}$ is a transformation. ■

The definition above allows for specifying different algebras of decompositions, which may be tailored for specific applications. We will examine two algebras defined on the set of all decompositions Δ_{ij}: the set algebra of decompositions, and the complex algebra of decompositions. Based on these algebras, some other "smaller" algebras will be constructed and examined.

2.1. Set algebras of decompositions

If parts recognized in decompositions are paramount in computations, combining decompositions as sets is a viable option. Set operations of union, intersection, and difference may be taken as sum, product, and difference of decompositions. The operations preserve the set elements, and set inclusion establishes the ordering on Δ_{ij}. However, set Δ_{ij} of decompositions is not closed under $\cap$ and $-$. The intersection or difference of two decompositions may be empty, but there is no empty decomposition; a decomposition contains at least one shape, which may be 0. The set Δ_{ij} has to be augmented with the empty set to become an algebra.

Definition 5. A set algebra of decompositions $\boldsymbol{D_{ij}}$ satisfies Definition 4 and has the set $\Delta_{ij} \cup \{\varnothing\}$ together with operations $\cup$, $\cap$, and $-$ as the Boolean part, whereas its group part is the same as that of $\boldsymbol{U_{ij}}$. ■

This algebra, which was introduced by Stiny (1990, 1994), treats shapes that are elements of decompositions as symbolic objects without further dividing or summing them. The algebra works well if shapes have predetermined parts that remain fixed throughout a computation. The result of the computation contains only shapes that are elements of decompositions entering the computation. These are the atoms of the computation. For example, decompositions in Figure 4a, b, and c are respective sum, product, and difference of decompositions in Figure 3 computed in the framework of $\boldsymbol{D_{ij}}$. Note that no new shape has been created in the computations.

2.2. Complex algebras of decompositions

Set algebras compute with shapes as they would with any other object. Properties of shapes are irrelevant in $\boldsymbol{D_{ij}}$. In contrast, shape properties, in particular spatial properties, are most important in design. An algebra of decompositions that aspires to be an algebra of analyzed shapes should take into account spatial properties of shapes that are elements of decompositions. This can be achieved by letting the elements of one decomposition combine with the elements of another decomposition in computations. With no special preference, these combinations may be taken exhaustively.

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

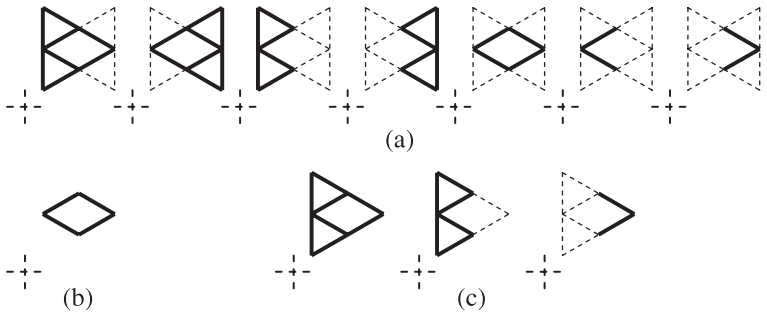

Fig. 4. Computing in $\boldsymbol{D_{12}}$: (a) sum, (b) product, and (c) difference of decompositions in Figure 3.

The operations of such an algebra are no more than extensions of shape operations ($+$, $\cdot$, and $-$) to direct products of decompositions.

Let A and B be two decompositions, then their sum $+'$ is defined as $A +' B = +(A \times B)$. "Polish" notation used here is an elegant way of writing $A +' B = \{\boldsymbol{x} + \boldsymbol{y} | \, \boldsymbol{x} \in A, \boldsymbol{y} \in B\}$. The other two operations product $\cdot'$ and difference $-'$ follow as $A \cdot' B = \cdot\ (A \times B)$ and $A -' B = -(A \times B)$.

Because decompositions are finite subsets (complexes) of some algebra of shapes $\boldsymbol{U_{ij}}$, they may be seen as elements of a complex algebra $\wp(\boldsymbol{U_{ij}})$ constructed on $\boldsymbol{U_{ij}}$. The algebra operates on the set $\wp(U_{ij}) - \{\varnothing\}$ with the set of all decompositions Δ_{ij} as a subset. The latter set is precisely the set of all finite complexes and forms a subalgebra in $\wp(\boldsymbol{U_{ij}})$. The general rule for operations of complex algebras (Gratzer, 1979) yields operations $+'$, $\cdot'$, and $-'$ as defined above.

Decompositions are ordered by $\leq'$ defined in accordance with the general definition for relations of complex structures (Gratzer & Whitney, 1978). That is, $A \leq' B$, if for every $\boldsymbol{x} \in A$ there exists $\boldsymbol{y} \in B$, such that $\boldsymbol{x} \leq \boldsymbol{y}$, and for every $\boldsymbol{y} \in B$ there exists $\boldsymbol{x} \in A$ such that $\boldsymbol{x} \leq \boldsymbol{y}$.

The set Δ_{ij} has a lower bound under the ordering $\leq'$. It is the decomposition of the empty shape $\{0\}$. However, Δ_{ij} does not have an upper bound.

DEFINITION 6. Complex algebra of decompositions $\wp'(\boldsymbol{U_{ij}})$ satisfies Definition 4 and has the set Δ_{ij} together with operation $+'$, $\cdot'$, and $-'$ as the Boolean part, whereas its group part is the same as that of $\boldsymbol{U_{ij}}$. ■

The new algebra handles shapes as spatial objects, not unlike $\boldsymbol{U_{ij}}$. For example, decompositions in Figure 5a, b, and c are the respective sum, product, and difference of decompositions in Figure 3 when computed in the framework $\wp'(\boldsymbol{U_{ij}})$. Note that the resulting shapes differ from the input ones: the former are combinations of the latter.

3. DECOMPOSITIONS AS SHAPE APPROXIMATIONS

Shape grammars and later shape algebras have a history of more than three decades as formal design theory tools. They have been established as formalizations of design practice. The algebras model what designers do when they draw and

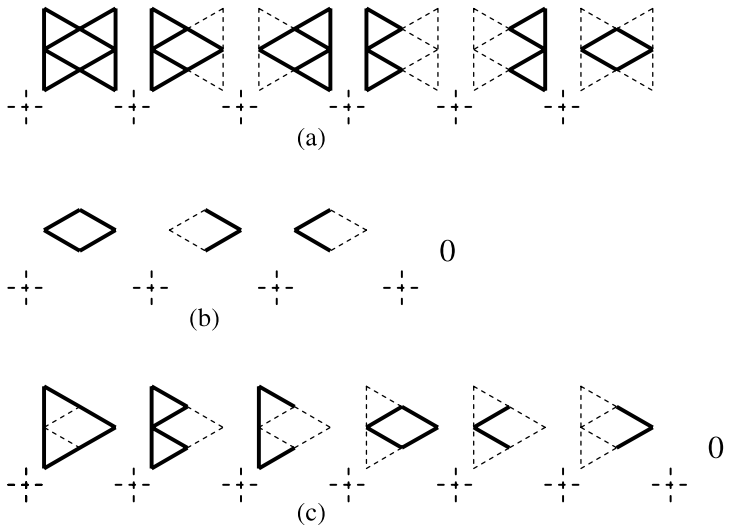

Fig. 5. Computing in $\wp'(\boldsymbol{U_{ij}})$: (a) sum, (b) product, and (c) difference of decompositions in Figure 3.

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

erase shapes, build, and modify models, or just move shapes around to produce new spatial relations. Grammars and algebras are capable of handling the emergence and ambiguity of shapes, which are important in the most creative, exploratory phase of a design process (Knight, 2003*a*, 2003*b*). In contrast, algebras of decompositions operate on sets of shapes, but these may or may not be seen as approximations of shapes. If decompositions are seen as shape approximations, the properties of their algebras should match those of shape algebras as closely as possible.

At minimum, an algebra for decompositions should be such that an appropriate shape algebra is homomorphic to it. This guaranties that the result of a computation carried on with decompositions of shapes is a decomposition of the result of the same computation carried on with the shapes themselves. Such symmetry between the two algebras is important if shapes in computations are to be approximated by their decompositions. Note that the mapping defining the homomorphism is the summation from Definition 2.

At maximum, algebras for shapes and decompositions should be isomorphic. At this point, symmetry between the algebras is complete and decompositions behave as shapes. Such an algebra of decompositions becomes a true algebra of analyzed shapes.

As an additional requirement, an algebra of decompositions should have enough power to inform the decision on whether two of its elements are analyzing the same shape. This renders the transition from an analyzed shape to the raw unanalyzed shape possible within the framework of the algebra.

3.1. Properties of set algebras of decompositions

Set algebras for decompositions compare favorably with shape algebras. All of the important properties of the latter, such as distributive and idempotent, are preserved by the former algebras. However, set algebras are less successful in satisfying minimum and additional requirements.

The minimum requirement holds for sums, but not for products and differences.

It is not possible to decide, by using operations and the relation of $\boldsymbol{D_{ij}}$, whether two of its different elements are decompositions of the same shape. The algebra sees decompositions as sets of shapes (Definition 1), rather then decompositions of shapes (Definition 2). It manipulates elements of decompositions as symbols disregarding their spatial properties.

This renders $\boldsymbol{D_{ij}}$ an unlikely choice if computations with analyzed shapes are an objective. However, $\boldsymbol{D_{ij}}$ is not without design applications.

Set algebras work well in situations where design is restricted to a finite kit of discrete parts, such that all shapes of interest are simple sums of the parts. Designing in the framework of a building system is a good example. Such systems usually offer kits of prefabricated building parts that may be assembled to produce a variety of buildings. Parts may range from structural members, like bearing walls or ceiling slabs, to architectural elements, such as partitions or facade articulations.

The set of all subsets of such a kit is a subalgebra of $\boldsymbol{D_{ij}}$, which is isomorphic with a certain subalgebra of $\boldsymbol{U_{ij}}$.

Let K_{ij} be a finite set of pairwise discrete shapes (no two of its elements share parts), $\wp(K_{ij})$ the set of all of its subsets, and $\Sigma\,\wp(K_{ij})$ the set of shapes obtained by summing the elements of $\wp(K_{ij})$. Because elements of K_{ij} are pairwise discrete shapes, combining them in products and differences yields no new shapes except for the empty shape. For example, if $\boldsymbol{a}, \boldsymbol{b} \in K_{ij}$ and $\boldsymbol{a} \neq \boldsymbol{b}$, then $\boldsymbol{a} \cdot \boldsymbol{b} = 0$ and $\boldsymbol{a} - \boldsymbol{b} = \boldsymbol{a}$, if $\boldsymbol{a} = \boldsymbol{b}$, then $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}$ and $\boldsymbol{a} - \boldsymbol{b} = 0$. New shapes can only be produced in sums so that $\Sigma\,\wp(K_{ij})$ is the set of all shapes that could be created from the shapes of K_{ij}. In contrast, set $\wp(K_{ij})$ enumerates all of the decompositions of the former shapes into the latter ones. Mapping Σ that connects the two sets has some interesting properties.

LEMMA 1. *Mapping* Σ *from* $\wp(K_{ij})$ *to* $\Sigma\,\wp(K_{ij})$ *is a bijection.* ■

Because $\Sigma\,\wp(K_{ij})$ is an image of $\wp(K_{ij})$, Σ is a surjection. To prove that Σ is an injection (one to one) we assume the opposite: there are two distinct decompositions in $\wp(K_{ij})$ that analyze the same shape in $\Sigma\,\wp(K_{ij})$. For the assumption to hold, there has to be at least one shape that is an element of one of the decompositions, but not an element of the other one. Say, decompositions A and B both analyze some shape $\boldsymbol{a}$, and shape $\boldsymbol{x}$ is an element of A, but not of B. Consequently, $\boldsymbol{x}$ is a part of ΣA, but not of ΣB. The latter holds because $\boldsymbol{x}$ does not share parts with any of the elements of B because they all belong to K_{ij} and so does $\boldsymbol{x}$. However, $\Sigma A = \Sigma B = \boldsymbol{a}$, which leads to a contradiction rendering $\boldsymbol{x}$ both a part of $\boldsymbol{a}$ and not part of $\boldsymbol{a}$. This refutes the assumption rendering Σ an injection. Because Σ is both a surjection and an injection it is a bijection, which completes the proof.

Sets $\wp(K_{ij})$ and $\Sigma\,\wp(K_{ij})$ are closed under set operations of $\boldsymbol{D_{ij}}$, and shape operations of $\boldsymbol{U_{ij}}$, respectively. They are Boolean parts of the two new algebras: decomposition algebra $\wp(\boldsymbol{K_{ij}}) \subset \boldsymbol{D_{ij}}$ and shape algebra $\Sigma\,\wp(\boldsymbol{K_{ij}}) \subset \boldsymbol{U_{ij}}$. The two new algebras are isomorphic in accordance with the following proposition.

PROPOSITION 2. *Algebras* $\wp(\boldsymbol{K_{ij}})$ *and* $\Sigma\,\wp(\boldsymbol{K_{ij}})$ *are isomorphic.* ■

The isomorphism is framed by the bijection Σ. We only need to show that Σ preserves the operations of algebras. Sum is preserved because $\boldsymbol{a} + \boldsymbol{b} = \Sigma A + \Sigma B = \Sigma(A \cup B)$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are shapes from $\Sigma\,\wp(\boldsymbol{K_{ij}})$ and A and B are their respective decompositions from $\wp(\boldsymbol{K_{ij}})$. The same can be shown for product and difference to complete the proof of the proposition.

The new *kit of parts* algebra $\wp(\boldsymbol{K_{ij}})$ computes with analyzed shapes in a meaningful way even though it does not

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

see them as spatial objects. In formal design theory $\wp(\boldsymbol{K_{ij}})$ algebras are used in systems like set grammars (Stiny, 1982) or structure grammars (Carlson et al., 1991), which work with predefined vocabularies of shapes. These systems have the same computational power as shape grammars and produce languages of designs similar to that generated by shape grammars. However, they cannot handle the emergence and ambiguity of shapes, which are important in creative phases of a design process. In contrast, shape grammars do that well (Knight, 2003*a*, 2003*b*).

3.2. Properties of complex algebras of decompositions

Unlike set algebras, complex algebras for decompositions do not preserve important properties of $\boldsymbol{U_{ij}}$ algebras. There are two reasons for this.

First, it is known in universal algebra that the identities $r = s$ valid in an algebra are also valid in its complex counterpart if and only if individual variables occur only once in both r and s (Guatam, 1957; Bleicher et al., 1973; Shafaat, 1974). However, most of the important identities of $\boldsymbol{U_{ij}}$ are not of this kind. In particular, these defining idempotent, $\boldsymbol{a} + \boldsymbol{a} = \boldsymbol{a}, \boldsymbol{a} \cdot \boldsymbol{a} = \boldsymbol{a}$, and distributive property, $\boldsymbol{a} \cdot (\boldsymbol{b} + \boldsymbol{c}) = (\boldsymbol{a} \cdot \boldsymbol{b}) + (\boldsymbol{a} \cdot \boldsymbol{c}), \boldsymbol{a} + (\boldsymbol{b} \cdot \boldsymbol{c}) = (\boldsymbol{a} + \boldsymbol{b}) \cdot (\boldsymbol{a} + \boldsymbol{c})$ are valid in $\boldsymbol{U_{ij}}$, but not in $\wp'(\boldsymbol{U_{ij}})$.

Second, the relation $\leq'$ is not a partial order, but a preorder. It is reflexive and transitive, but not antisymmetric. Consequently, Δ_{ij} is not a partially ordered set so that identities that are valid in $\boldsymbol{U_{ij}}$, under some condition(s) expressed in terms of $\leq$, may not be valid in $\wp'(\boldsymbol{U_{ij}})$. In particular, if $\boldsymbol{a} \leq \boldsymbol{b}$, then $\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{b}$ and $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}$, and if $\boldsymbol{a} \leq \boldsymbol{b}$ and $\boldsymbol{b} \leq \boldsymbol{a}$, then $\boldsymbol{a} = \boldsymbol{b}$ are valid in $\boldsymbol{U_{ij}}$, but not in $\wp'(\boldsymbol{U_{ij}})$.

Algebra $\wp'(\boldsymbol{U_{ij}})$, does better than $\boldsymbol{D_{ij}}$ when it comes to satisfying the minimum and the additional requirements above. The former holds if the differences are ignored, whereas the latter holds with no constraints.

Because it is possible to decide by using operations and the relation of $\wp'(\boldsymbol{U_{ij}})$ whether its two elements are decompositions of the same shape, the algebra sees decompositions as in Definition 2, that is as decompositions of shapes. This is handled by the following proposition.

PROPOSITION 3. *Let A and B be decompositions of shapes* $\boldsymbol{a}$ *and* $\boldsymbol{b}$*, respectively, then* $\boldsymbol{a} = \boldsymbol{b}$ *if and only if* $\Sigma'_{k=1,\ldots,m} A \leq' \Sigma'_{k=1,\ldots,n} B$ *and* $\Sigma_{k=1,\ldots,n} B \leq' \Sigma'_{k=1,\ldots,m} A$*, where m and n are cardinalities of A and B, respectively, and* Σ' *is the extension of the sum* $+'$ *of* $\wp'(\boldsymbol{U_{ij}})$*.* ■

Therefore, $\Sigma'_{k=1,\ldots,m} A = A +' A +' A +' \ldots$, where A appears m times on the right side of the equality. The proof then follows from the definition of $\leq'$ and the fact that $\boldsymbol{a} \in \Sigma'_{k=1,\ldots,m} A$ and $\boldsymbol{b} \in \Sigma'_{k=1,\ldots,n} B$. The latter is due to the following lemma and its corollary.

LEMMA 2. *Let* $A = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x}_n\}$ *be a decomposition, the sum* $S(k) = \Sigma'_{i=1,\ldots,k} A$*, where A appears k-times as an argument with* $k \leq n$*, contains an element* $\boldsymbol{y} = \Sigma_{i=1,\ldots,k} \boldsymbol{x}_i$*. Note that* Σ' *is the extension of* $+'$ *of* $\wp'(\boldsymbol{U_{ij}})$ *whereas* Σ *is the extension of* $+$ *of* $\boldsymbol{U_{ij}}$*.* ■

The proof is based on the definition of $+'$ where each element of one decomposition is combined in sum $+$ with each element of the other decomposition. Therefore, the lemma holds for $k = 2$ because $S(2) = A +' A$ contains shape $\boldsymbol{x}_1 + \boldsymbol{x}_2$. If we assume that it holds for k, $2 < k < n$, so that $S(k)$ contains shape $\boldsymbol{z} = \Sigma_{i=1,\ldots,k} \boldsymbol{x}_i$, then it is easy to show that it also holds for $k + 1$. Sum $S(k + 1) = S(k) +' A$ contains shape $\boldsymbol{z} + \boldsymbol{x_{k+1}} = (\Sigma_{i=1,\ldots,k} \boldsymbol{x}_i) + \boldsymbol{x_{k+1}} = \Sigma_{i=1,\ldots,k+1} \boldsymbol{x}_i$, which completes the proof by the induction.

COROLLARY. *Let A be as above and let* $\boldsymbol{a}$ *be the shape A analyzes,* $\Sigma S(n)$ *then contains* $\boldsymbol{a}$*.* ■

By Definition 2, $\boldsymbol{a} = \Sigma A = \Sigma_{i=1,\ldots,n} \boldsymbol{x}_i$ whereas by the lemma above, $S(n)$ contains $\Sigma_{i=1,\ldots,n} \boldsymbol{x}_i$.

Note that the dual of Lemma 2 also holds, so that $P(k) = \Pi'_{i=1,\ldots,k} A$ contains an element $\boldsymbol{y} = \Pi_{i=1,\ldots,k} \boldsymbol{x}_i$, where Π' is the extension of $\cdot'$ whereas Π is the extension of $\cdot$.

Computations involving sums and products satisfy the minimum requirement. For example, the decompositions in Figure 5a and b analyze shapes in Figure 3c and d. The latter are the respective sum and product of the shapes in Figure 5a and b, whereas the former are the respective sum and product of their decompositions in Figure 5f and g.

Unfortunately, the minimum requirement is not satisfied by computations involving difference. For example, decomposition in Figure 5c analyzes shapes in Figure 3a. The latter is not the difference of the shapes in Figure 5a and b, although the decomposition is the difference of their decompositions in Figure 5f and g.

Although $\wp'(\boldsymbol{U_{ij}})$ treats shapes as spatial objects, the same way $\boldsymbol{U_{ij}}$ does, $\wp'(\boldsymbol{U_{ij}})$ is still a poor choice for computing with analyzed shapes. It does not satisfy the minimum requirement and most of the important identities (properties) of $\boldsymbol{U_{ij}}$.

However, as with $\boldsymbol{D_{ij}}$, special kinds of decompositions form a subalgebra in $\wp'(\boldsymbol{U_{ij}})$, which is capable of handling analyzed shapes in a meaningful way. The decompositions are singletons containing the shape they analyze and nothing else. Approximating a shape with itself does not add much to the shape description, so that an algebra of such decompositions should behave as a shape algebra. This is true due to the following result.

PROPOSITION 4. *Any identity* $r = s$ *valid in* $\boldsymbol{U_{ij}}$ *is valid in* $\wp'(\boldsymbol{U_{ij}})$ *if a singleton from* $\wp'(\boldsymbol{U_{ij}})$ *is assigned to each individual variable occurring more than once in r or s.* ■

This proposition is valid for any algebra and its complex counterpart; however, the proof in its generality is left to mathematicians. We will only show that it holds for one of the identities defining the distributivity of $\boldsymbol{U_{ij}}$. Let $A, B, C \in \wp'(\boldsymbol{U_{ij}})$. The identity $A \cdot' (B +' C) = (A \cdot' B) +' (A \cdot' C)$,

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

defining distributivity of $\cdot'$ over $+'$, is not valid in $\wp'(\boldsymbol{U_{ij}})$, because A occurs twice on the right-hand side. However, if A is a singleton $\{\boldsymbol{a}\}$, the identity becomes $\{\boldsymbol{a}\} \cdot' (B +' C) = (\{\boldsymbol{a}\} \cdot' B) +' (\{\boldsymbol{a}\} \cdot' C)$ ($'$). Elements of $\{\boldsymbol{a}\} \cdot' (B +' C)$ are, in accordance with definitions of the operations $\cdot'$ and $+'$, of the form $\boldsymbol{a} \cdot (\boldsymbol{x} + \boldsymbol{y})$, where $\boldsymbol{x} \in B$ and $\boldsymbol{y} \in C$. In contrast, all of the elements of $(\{\boldsymbol{a}\} \cdot' B) +' (\{\boldsymbol{a}\} \cdot' C)$ are of the form $(\boldsymbol{a} \cdot \boldsymbol{x}) + (\boldsymbol{a} \cdot \boldsymbol{y})$. Because $\boldsymbol{U_{ij}}$ is distributive, $\boldsymbol{a} \cdot (\boldsymbol{x} + \boldsymbol{y}) = (\boldsymbol{a} \cdot \boldsymbol{x}) + (\boldsymbol{a} \cdot \boldsymbol{y})$, so that identity ($'$) holds.

The algebra for singleton decompositions can now be constructed as follows. The set $A_{ij} \subset \Delta_{ij}$ of singleton decompositions together with operations $+'$, $\cdot'$, and $-'$ forms the Boolean part of a new algebra $\boldsymbol{A_{ij}}$. The group part and the operation of the group action of $\boldsymbol{A_{ij}}$ are the same as in $\wp'(\boldsymbol{U_{ij}})$.

Proposition 5. *The new algebra $\boldsymbol{A_{ij}}$ is a subalgebra of $\wp'(\boldsymbol{U_{ij}})$ isomorphic with $\boldsymbol{U_{ij}}$.* ∎

For $\boldsymbol{A_{ij}} \subseteq \wp'(\boldsymbol{U_{ij}})$ to hold A_{ij} should be closed under the operations of $\wp'(\boldsymbol{U_{ij}})$. Indeed, according to definitions of these operations we have $\{\boldsymbol{a}\} +' \{\boldsymbol{b}\} = \{\boldsymbol{a} + \boldsymbol{b}\}$, $\{\boldsymbol{a}\} \cdot' \{\boldsymbol{b}\} = \{\boldsymbol{a} \cdot \boldsymbol{b}\}$, and $\{\boldsymbol{a}\} -' \{\boldsymbol{b}\} = \{\boldsymbol{a} - \boldsymbol{b}\}$, where $\boldsymbol{a}, \boldsymbol{b} \in U_{ij}$ and $\{\boldsymbol{a}\}$, $\{\boldsymbol{b}\} \in A_{ij}$ ($'$). In set theory, a singleton (decomposition) can uniquely be specified as $\{\boldsymbol{a}\} = \{\boldsymbol{x} \mid \boldsymbol{x} = \boldsymbol{a}\}$, whereas (shape) $\boldsymbol{a}$ can be "recovered" from the singleton (decomposition) via $\cup\{\boldsymbol{a}\} = \boldsymbol{a}$. Therefore, there is one to one correspondence between shapes and their singleton decompositions. This correspondence preserves operations of $\boldsymbol{U_{ij}}$ as evident in identities ($'$), which completes the proof.

3.3. Algebras of analyzed shapes

Even though $\wp'(\boldsymbol{U_{ij}})$ treats shapes as spatial entities, the only analyzed shapes it can meaningfully handle are (trivial) singleton decompositions. This is rather disappointing because analyzed shapes play an important role not only in design, but also in our everyday lives. However, there is still some room for improvement of $\wp'(\boldsymbol{U_{ij}})$.

One can *normalize* operations of $\wp'(\boldsymbol{U_{ij}})$ to make a shape algebra homomorphic to it, the minimum requirement. The normalization relies on the concept of relativization of the structure of a shape to its subshape. In $\wp'(\boldsymbol{U_{ij}})$, the relativization can be expressed in accordance with the following proposition.

Proposition 6. *Let $\boldsymbol{a}$ be a shape, $A \in \wp'(\boldsymbol{U_{ij}})$ its decomposition, and shape $\boldsymbol{b}$ its part. Set $B = \{\boldsymbol{b}\} \cdot' A$ is a decomposition of $\boldsymbol{b}$, which is the relativization of A to $\boldsymbol{b}$.* ∎

Set $B = \{\boldsymbol{b}\} \cdot' A = \cdot (\{\boldsymbol{b}\} \times A) = \{\boldsymbol{x} \cdot \boldsymbol{y} \mid \boldsymbol{x} \in \{\boldsymbol{b}\}, \boldsymbol{y} \in A\} = \{\boldsymbol{b} \cdot \boldsymbol{x} \mid \boldsymbol{x} \in A\}$, which is, according to Definition 3, the relativization of A to $\boldsymbol{b}$.

If A is the result of a computation with decompositions and $\boldsymbol{b}$ is the result of the same computations carried on with the corresponding shapes, then B is the result of the normalized operation. If A is the result of a sum or a product, then $B = A$, which means that normalization does not affect these operations. The normalized sum $+^*$ and product $\cdot^*$ are the same as $+'$ and $\cdot'$. Normalization affects only the difference, which becomes $A -^* B = \{\Sigma A - \Sigma B\} \cdot' (A -' B) = A -' \{\Sigma B\}$, where A and B are decompositions. Set $A -^* B$ is clearly a decomposition of the difference of shapes analyzed by decompositions A and B. It has the structure of A, but not of B.

It is now possible to define a new *normalized* complex algebra of decompositions $\wp^*(\boldsymbol{U_{ij}})$, which is the same as $\wp'(\boldsymbol{U_{ij}})$ except for the Boolean operations, which are now normalized.

For example, the decompositions in Figure 6a, b, and c are the respective sum, product, and difference of decompositions in Figure 3 when computed in the framework $\wp^*(\boldsymbol{U_{ij}})$. Note that sum in Figure 6a and product in Figure 6b are the same as the ones computed in the framework of $\wp'(\boldsymbol{U_{ij}})$ (Fig. 5a, b).

The new algebra satisfies the minimum requirement: $\boldsymbol{U_{ij}}$ is homomorphic to it. However, the two algebras still have different properties.

These differences could be largely reduced if the ordering on decompositions is changed to reassemble that on shapes. Relation $\leq'$ is a preorder on Δ_{ij}, whereas $\leq$ is a partial order on U_{ij}. Universal algebra provides a procedure

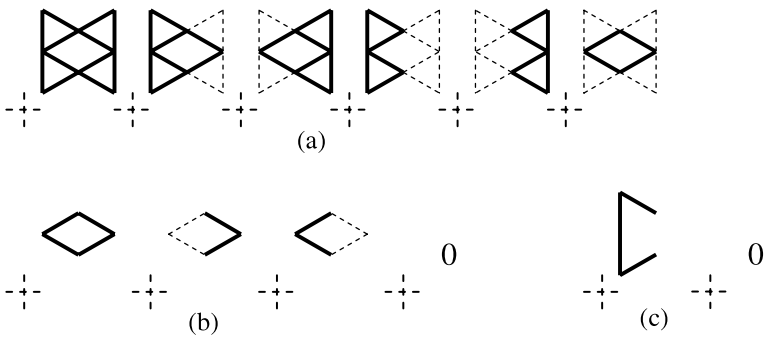

Fig. 6. Computing in $\wp^*(\boldsymbol{U_{ij}})$: (a) sum, (b) product, and (c) difference of decompositions in Figure 3.

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

for turning a preordered set into a partially ordered one (Vickers, 1989). The partially ordered set can then be extended to an algebra to handle analyzed shapes.

The procedure starts with equivalence relation $\equiv$ defined on Δ_{ij} in terms of preorder $\preceq'$. Two decompositions A and B are equivalent with respect to $\equiv$, or equivalent *modulo* $\equiv$, or $A \equiv B$, if and only if $A \preceq' B$ and $B \preceq' A$.

For example, decompositions in Figure 7a and b are equivalent modulo $\equiv$. Every element of the former decomposition is a part of shape c from the latter decomposition. In contrast, the empty shape from the former decomposition is a part of every element of the latter decomposition. Consequently, the latter includes the former decomposition in accordance with the definition of $\preceq'$. It is easy to verify that the relation also holds if the decompositions exchange places.

In the next step, Δ_{ij} is partitioned so that decompositions equivalent modulo $\equiv$ are grouped into subsets or *equivalence classes*. Each decomposition A belongs to a class $[A]_{\equiv}$ and the set of all equivalence classes is a *quotient set* $\Delta_{ij}/\equiv$.

Finally, set $\Delta_{ij}/\equiv$ is partially ordered by $\preceq'$ so that $[A]_{\equiv} \preceq' [B]_{\equiv}$ if and only if $A \preceq' B$ (Vickers, 1989).

Each equivalence class of $\Delta_{ij}/\equiv$ contains decompositions that have the following common properties.

PROPOSITION 7. *Two decompositions are equivalent modulo* $\equiv$ *if and only if they share the sets of minimal and maximal elements.* ■

Let $\min A$ and $\max A$ denote sets of minimal and maximal elements of a decomposition A, and let B be a decomposition. We will first prove the following lemma.

LEMMA 3. *If* $A \equiv B$, *then* $\max A \subseteq B$, $\max B \subseteq A$, $\min A \subseteq B$, *and* $\min B \subseteq A$. ■

Let $\boldsymbol{x} \in \max A$. According to the definition of $\equiv$, $A \preceq' B$ and, according to definition of $\preceq'$, there is $\boldsymbol{y} \in B$ such that $\boldsymbol{x} \preceq \boldsymbol{y}$. According to the same definitions, $B \preceq' A$ and there is $\boldsymbol{z} \in A$ such that $\boldsymbol{y} \preceq \boldsymbol{z}$. Consequently, $\boldsymbol{x} \preceq \boldsymbol{y} \preceq \boldsymbol{z}$ (*) and because $\boldsymbol{x}$ is a maximal element of A and $\boldsymbol{z}$ is an element of A, the two have to be equal for (*) to hold. This makes $\boldsymbol{x}$ equal to $\boldsymbol{y}$ so that $\boldsymbol{x} \in B$, which proves that $\max A \subseteq B$. The proof of $\max B \subseteq A$ follows from the symmetry of $\equiv$. The proofs of the last two assertions may be done in the similar fashion to complete the proof of the lemma.

Let us now assume that $\max A = \max B$ and $\min A = \min B$. For every $\boldsymbol{x} \in A$ there is $\boldsymbol{y} \in \max A$ such that $\boldsymbol{x} \preceq \boldsymbol{y}$, and because $\max A = \max B$, $\boldsymbol{y} \in B$. Similarly, for every $\boldsymbol{v} \in B$ there is $\boldsymbol{u} \in \min B$ such that $\boldsymbol{u} \preceq \boldsymbol{v}$, and because $\min A = \min B$, $\boldsymbol{u} \in A$. Consequently, $A \preceq' B$ in accordance with the definition of $\preceq'$. The proof of $B \preceq' A$ may be obtained in the similar fashion so that $A \equiv B$, which completes the proof of the "if" part of the proposition.

Now assume that $A \equiv B$, but the two do not have a common set of maximal elements. For example, there is an element $\boldsymbol{x}$ maximal in A but not in B. According to Lemma 3, $\max A \subseteq B$ so that $\boldsymbol{x} \in B$. Because $\boldsymbol{x}$ is not maximal in B there is $\boldsymbol{y} \in \max B$ such that $\boldsymbol{x} \prec \boldsymbol{y}$. Again, according to lemma 3, $\max B \subseteq A$ so that $\boldsymbol{y} \in A$, and because $\boldsymbol{x}$ is maximal in A, $\boldsymbol{x} \prec \boldsymbol{y}$ does not hold. This refutes our assumption that $\max A \neq \max B$ so that $\max A = \max B$ holds. The proof that $\min A = \min B$ holds could be obtained in a similar fashion, which completes the proof of the "only if" part of the proposition.

COROLLARY 1. *Decompositions equivalent modulo* $\equiv$ *are analyzing the same shape.* ■

Let A and B be two decompositions such that $A \equiv B$. The shape that A analyzes is ΣA and because $\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{a}$ if $\boldsymbol{b} \preceq \boldsymbol{a}$, we can take $\Sigma \max A$ for ΣA. By the same token, the shape that B analyzes or ΣB is the same as shape $\Sigma \max B$. According to Proposition 7, $\max A = \max B$ so that $\Sigma A = \Sigma \max A = \Sigma \max B = \Sigma B$, which completes the proof.

COROLLARY 2. *There is a unique smallest representative* $\max A \cup \min A$ *for each equivalence class* $[A]_{\equiv}$, *where* A *is a decomposition.* ■

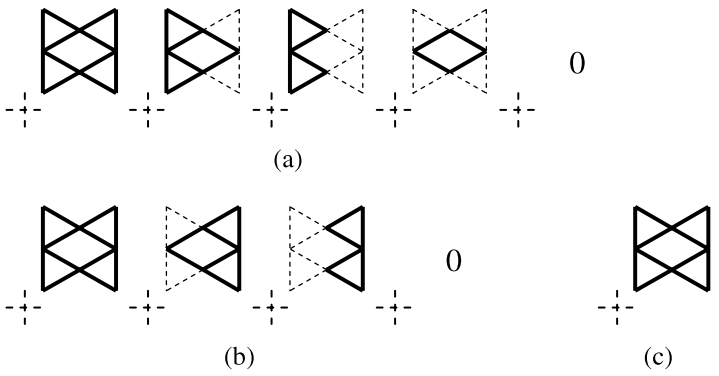

Fig. 7. (a,b) Decompositions are equivalent modulo $\equiv$. Every element of (a) is a part of (c) the shape included in (b). The empty shape in (a) is a part of every element of (b). Every element of (b) is a part of shape (c) included in (a). The empty shape in (b) is a part of every element of (a).

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

Note that the word smallest is related to the cardinality (number of elements) of decompositions.

For $\Delta_{ij}/\equiv$ to be extended to a quotient algebra, equivalence $\equiv$ needs to be a congruence on $\wp^*(\boldsymbol{U_{ij}})$.

PROPOSITION 8. *Equivalence relation $\equiv$ is a congruence on $\wp^*(\boldsymbol{U_{ij}})$.* ■

Let $A, B, C, D \in \wp^*(\boldsymbol{U_{ij}})$ and let $A \equiv B$ and $C \equiv D$. For $\equiv$ to be a congruence, substitution property has to hold for each operation of $\wp^*(\boldsymbol{U_{ij}})$. Substitution property for the sum is $A +^* C \equiv B +^* D$. Elements of $\max(A +^* C)$ are of the form $\boldsymbol{x} + \boldsymbol{y}$, where $\boldsymbol{x} \in \max A$ *and* $\boldsymbol{y} \in \max C$. In contrast, elements of $\max(B +^* D)$ are of the form $\boldsymbol{u} + \boldsymbol{v}$, where $\boldsymbol{u} \in \max B$ and $\boldsymbol{v} \in \max D$. Because, $\max A = \max B$ and $\max C = \max D$, in accordance with Proposition 7, $\max(A +^* C) = \max(B +^* D)$. Similarly, $\min(A +^* C) = \min(B +^* D)$, so that, according to Proposition 7, $A +^* C \equiv B +^* D$, which proves that the substitution property holds for $+^*$. It can be shown that the same property also holds for $\cdot^*$ and $-^*$, which renders $\equiv$ a congruence on $\wp^*(\boldsymbol{U_{ij}})$.

It is now possible to construct a quotient algebra $\wp^*(\boldsymbol{U_{ij}})/\equiv$, with elements that are classes of decompositions that share the shape they analyze and sets of minimal and maximal elements. The set of such classes is partially ordered with a smallest element, which is a one-element equivalence class of {0}, but with no greatest element.

Computations in $\wp^*(\boldsymbol{U_{ij}})/\equiv$ are carried on as in $\wp^*(\boldsymbol{U_{ij}})$. For example, computations in Figure 6 may be seen as computations in $\wp^*(\boldsymbol{U_{ij}},)/\equiv$, except that the decompositions are now seen as representatives of their equivalence classes. Consequently, any other representative of the same class may be used in the place of any of the decompositions. For example, decomposition in Figure 6a may be replaced with decomposition in Figure 8a because they belong to the same equivalence class. That is, they share sets of maximal and minimal elements, represented in Figure 8b and c, respectively.

Although all decompositions of an equivalence class are guaranteed to analyze the same shape, there are other classes in $\wp^*(\boldsymbol{U_{ij}})/\equiv$ with decompositions that analyze that very shape.

For example, decompositions in Figure 8a and d both analyze the shape in Figure 8b, but they do not belong to the same equivalence class. The former decomposition has a singleton containing the shape (Fig. 8b) as the set of maximal elements and the decomposition in Figure 8c as the set of minimal elements. In contrast, the decomposition in Figure 8d has c as the set of maximal elements and a singleton containing an empty shape as the set of minimal elements.

There is no one to one correspondence between shapes of $\boldsymbol{U_{ij}}$ and classes of $\wp^*(\boldsymbol{U_{ij}})/\equiv$ so that the two algebras are not isomorphic. However, any subalgebra of $\wp^*(\boldsymbol{U_{ij}})/\equiv$ whose elements enumerate shapes in a one to one fashion is isomorphic with $\boldsymbol{U_{ij}}$.

For example, the algebra $\boldsymbol{A_{ij}}$ of singleton decompositions can be constructed as a subalgebra of $\wp^*(\boldsymbol{U_{ij}})/\equiv$. For each singleton $\{\boldsymbol{a}\}$, sets of maximal and minimal elements are equal and both are equal to $\{\boldsymbol{a}\}$. This places $\{\boldsymbol{a}\}$ into an equivalence class of $\wp^*(\boldsymbol{U_{ij}})/\equiv$, with $\{\boldsymbol{a}\}$ the only element.

The following proposition sheds some light on the structure of decompositions that are building blocks of subalgebras of $\wp^*(\boldsymbol{U_{ij}})/\equiv$ that are isomorphic with $\boldsymbol{U_{ij}}$.

PROPOSITION 9. *Only decompositions that contain the shape they analyze and have exactly one minimal element qualify as the members of equivalence classes forming a subalgebra of $\wp^*(\boldsymbol{U_{ij}})/\equiv$ isomorphic with $\boldsymbol{U_{ij}}$.* ■

Members of the Boolean part of such a subalgebra are equivalence classes of decompositions, which according to Proposition 7, are characterized by the common sets of maximal and minimal elements. Proposition 9, however, requires these sets to be singletons. The following two lemmas prove this statement.

LEMMA 4. *No subset of $\wp(U_{ij})/\equiv$ with elements that are classes of decompositions characterized by nonsingleton sets of maximal elements is a subalgebra of $\wp^*(\boldsymbol{U_{ij}})/\equiv$ isomorphic with $\boldsymbol{U_{ij}}$.* ■

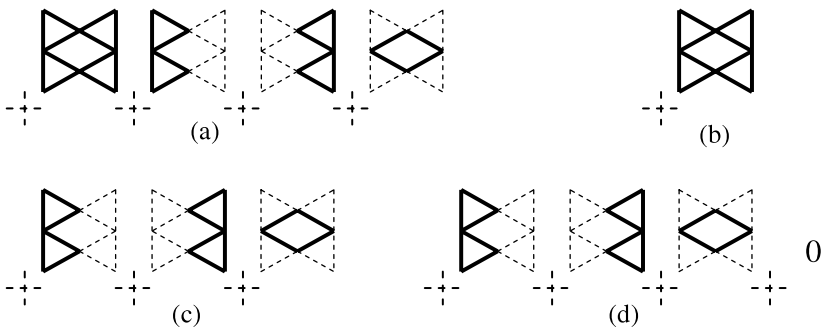

Fig. 8. (a) Decompositions and Figure 6a belong to the same equivalence class. (a, d) Decompositions both analyze (b) the shape, but they do not belong to the same equivalence class. Their sets of maximal and minimal elements are different: (b, c) for (a), and (c) and {0} for (d), respectively.

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

Let $V_{ij} \subseteq \wp(U_{ij})/\equiv$ be such a subset and let $[A]_\equiv$ be one of its elements, where $A = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n\}$, $\max A = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_k\}$, $2 \leq k \leq n$, and A analyzes shape $\boldsymbol{a}$. If we take a finite sum $\Sigma^*_{i=1,\ldots n}[A]_\equiv = \Sigma'_{i=1,\ldots n}[A]_\equiv = [\Sigma'_{i=1,\ldots n} A]_\equiv = [S(n)]_\equiv$ then, according to the corollary of Lemma 2, $\boldsymbol{a}$ is an element of $S(n)$. Because $\boldsymbol{U_{ij}}$ is homomorphic to $\wp^*(\boldsymbol{U_{ij}})$, both $S(n)$ and A are analyzing the same shape, $\boldsymbol{a}$. Consequently, $\max S(n) = \{\boldsymbol{a}\}$, and because $k \geq 2$ $\max A = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_k\} \neq \max S(n)$ so that $[A]_\equiv \neq [S(n)]_\equiv$. This means that if $[S(n)]_\equiv$ is an element of V_{ij} and V_{ij} is a subalgebra $\wp^*(\boldsymbol{U_{ij}})/\equiv$, then V_{ij} is not isomorphic with $\boldsymbol{U_{ij}}$ because shape $\boldsymbol{a}$ is represented by the two different elements: $[A]_\equiv$ and $[S(n)]_\equiv$ of V_{ij}. If $[S(n)]_\equiv$ is not an element of V_{ij}, then V_{ij} is not a subalgebra of $\wp^*(\boldsymbol{U_{ij}})/\equiv$, which completes the proof.

LEMMA 5. *No subset of $\wp(U_{ij})/\equiv$ with elements that are classes of decompositions characterized by nonsingleton sets of minimal elements is a subalgebra of $\wp^*(\boldsymbol{U_{ij}})/\equiv$ isomorphic with $\boldsymbol{U_{ij}}$.* ■

The proof is similar to that of Lemma 4 except that $P(n) = \Pi'_{i=1,\ldots,n} A$ and the dual of Lemma 2 are used in the place of $S(n)$ and Lemma 2.

According to the lemmas above, only subsets of $\wp(U_{ij})/\equiv$ with elements that are equivalence classes of decompositions characterized by singleton sets of minimal and maximal elements may be subalgebras of $\wp^*(\boldsymbol{U_{ij}})/\equiv$ isomorphic with $\boldsymbol{U_{ij}}$. If $\max A$ is a singleton, it has to be equal to $\{\boldsymbol{a}\}$; however, for $\min A$ we have more options. For example, in $\boldsymbol{A_{ij}}$, $\min A$ is the same as $\max A$, namely $\{\boldsymbol{a}\}$.

Another example of a subalgebra of $\wp^*(\boldsymbol{U_{ij}})/\equiv$ isomorphic with $\boldsymbol{U_{ij}}$ is based on equivalence classes of decompositions that recognize the shape they analyze, as well as the fact that the empty shape is a part of any shape. It may contain other parts too, but has, at least, to recognize the two. Such a decomposition satisfies Proposition 9 with minimal element 0. For each shape $\boldsymbol{a} \in \boldsymbol{U_{ij}}$ there is exactly one equivalence class $[\{\boldsymbol{a}, 0\}]_\equiv$ of such decompositions that corresponds to $\boldsymbol{a}$. Set $C_{ij} \subset \Delta_{ij}/\equiv$ of such equivalence classes can be elevated to an algebra in accordance with the following result.

PROPOSITION 10. *Set C_{ij} together with operations $+^*$, $\cdot^*$, and $-^*$, is the Boolean part of an algebra $\boldsymbol{C_{ij}}$, which is a subalgebra $\wp^*(\boldsymbol{U_{ij}})/\equiv$ isomorphic with $\boldsymbol{U_{ij}}$.* ■

Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be shapes from $\boldsymbol{U_{ij}}$, A and B their respective decompositions, and $\boldsymbol{t}$ a transformation from the group part of $\wp^*(\boldsymbol{U_{ij}})/\equiv$. In addition, let A and B be members of some equivalence classes of C_{ij}.

Because $[A]_\equiv \in C_{ij}$, $\{\boldsymbol{a}, 0\} \subseteq A$ so that $\max A = \{\boldsymbol{a}\}$ and $\min A = \{0\}$. According to Corollary 2 of Proposition 7, there is a unique representative of $[A]_\equiv$, such that $[A]_\equiv = [\max A \cup \min A]_\equiv = [\{\boldsymbol{a}, 0\}]_\equiv$. Consequently, there is only one equivalence class in C_{ij} containing decompositions of $\boldsymbol{a}$, each class of C_{ij} corresponds to only one shape, and for every shape in U_{ij} there is a corresponding class in C_{ij}. The correspondence is a bijection, thus, enumerating all of the elements of U_{ij} and C_{ij}. It needs to be shown that the correspondence also preserves the operations and transformations of $\boldsymbol{C_{ij}}$ (or $\boldsymbol{U_{ij}}$).

For the sum we have $[A]_\equiv +^* [B]_\equiv = [\{\boldsymbol{a}, 0\}]_\equiv +^* [\{\boldsymbol{b}, 0\}]_\equiv = [\{\boldsymbol{a} + \boldsymbol{b}, \boldsymbol{a}, \boldsymbol{b}, 0\}]_\equiv = [\{\boldsymbol{a} + \boldsymbol{b}, 0\}]_\equiv \in C_{ij}$ in accordance with Corollary 2 of Proposition 7. The result of the sum of equivalence classes is the equivalence class of the sum of corresponding shapes. The same is true for products and differences: $[A]_\equiv \cdot^* [B]_\equiv = [\{\boldsymbol{a}, 0\}]_\equiv \cdot^* [\{\boldsymbol{b}, 0\}]_\equiv = [\{\boldsymbol{a} \cdot \boldsymbol{b}, 0\}]_\equiv \in C_{ij}$ and $[A]_\equiv -^* [B]_\equiv = [\{\boldsymbol{a}, 0\}]_\equiv -^* [\{\boldsymbol{b}, 0\}]_\equiv = [\{\boldsymbol{a} - \boldsymbol{b}, 0\}]_\equiv \in C_{ij}$. This renders $\boldsymbol{C_{ij}}$ a subalgebra of the Boolean part of $\wp^*(\boldsymbol{U_{ij}})/\equiv$ isomorphic with the Boolean part of $\boldsymbol{U_{ij}}$.

Set C_{ij} is closed under transformations of $\wp^*(\boldsymbol{U_{ij}})/\equiv$ and the correspondence above preserves the transformations. We have $\boldsymbol{t}([A]_\equiv) = \boldsymbol{t}([\{\boldsymbol{a}, 0\}]_\equiv) = [\{\boldsymbol{t}(\boldsymbol{a}), 0\}]_\equiv \in C_{ij}$. Consequently, $\boldsymbol{C_{ij}}$ and $\boldsymbol{U_{ij}}$ are isomorphic, which completes the proof.

For example, each of the decompositions in Figure 3 may be augmented with an empty shape and used to compute in the framework of $\boldsymbol{C_{ij}}$ as illustrated in Figure 9.

Algebra $\boldsymbol{C_{ij}}$ is the most interesting of the algebras constructed here. It computes with decompositions that are nontrivial shape approximations and behaves like a shape algebra. The shape approximations of $\boldsymbol{C_{ij}}$ are rather general ones. They may contain any set of parts of a shape provided that the shape itself and the empty shape are included. This renders $\boldsymbol{C_{ij}}$ an excellent choice as a framework for computations with decompositions seen as shape approximations. Computations carried on with shapes in the framework of $\boldsymbol{U_{ij}}$ algebra can now be repeated with analyzed shapes in the framework of $\boldsymbol{C_{ij}}$ algebra.

For example, the result $[\{\boldsymbol{a} + \boldsymbol{b}, \boldsymbol{a}, \boldsymbol{b}, 0\}]_\equiv$ of the sum above seems like a reasonable explanation of the sum of shapes $\boldsymbol{a}$ and $\boldsymbol{b}$. One should expect both $\boldsymbol{a}$ and $\boldsymbol{b}$ to be preserved by the sum. If we add two new parts $\boldsymbol{c} \leq \boldsymbol{a}$ and $\boldsymbol{d} \leq \boldsymbol{b}$ to the decompositions above, the same computation becomes the following:

$$[\{\boldsymbol{a}, \boldsymbol{c}, 0\}]_\equiv +^* [\{\boldsymbol{b}, \boldsymbol{d}, 0\}]_\equiv = [\{\boldsymbol{a} + \boldsymbol{b}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} + \boldsymbol{d}, \boldsymbol{c}, \boldsymbol{d}, 0\}]_\equiv.$$

The new parts are preserved, whereas their sum $\boldsymbol{c} + \boldsymbol{d}$, also included, extends the computation to the new parts. The two principles, preservation and extension, inform computations in $\boldsymbol{C_{ij}}$. A computation preserves as much as possible of the original parts, but also extends (to) these parts.

It has been believed that algebras $\boldsymbol{A_{ij}}$ and $\boldsymbol{C_{ij}}$ are the only subalgebras of $\wp^*(\boldsymbol{U_{ij}})/\equiv$ isomorphic with $\boldsymbol{U_{ij}}$ (Krstic, 2004); however, Proposition 9 seems to allow for construction of some other such algebras. This is based on the fact that for a given shape $\boldsymbol{a}$ we can create an equivalence class $[\{\boldsymbol{a}, f(\boldsymbol{a})\}]_\equiv$ to represent $\boldsymbol{a}$. Operator f (on the Boolean part of $\boldsymbol{U_{ij}}$) has to be chosen so that $f(\boldsymbol{a})$ is a part of $\boldsymbol{a}$. For a set of such equivalence classes to be an algebra isomorphic

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

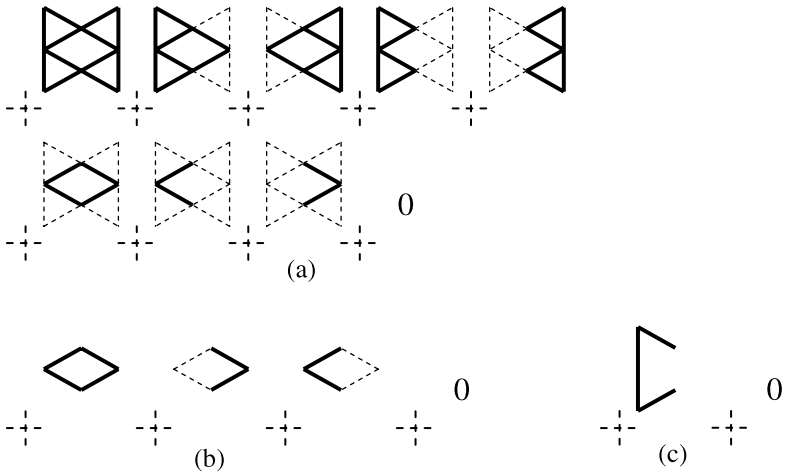

Fig. 9. Computing in C_{ij}: (a) sum, (b) product, and (c) difference of decompositions in Figure 3, each augmented with an empty shape.

with U_{ij}, operator f has to satisfy the following five conditions:

$$f(a) \leq a,$$

$$f(a) + f(b) = f(a + b),$$

$$f(a) \cdot f(b) = f(a \cdot b),$$

$$f(a) - b = f(a - b), \quad \text{and}$$

$$t(f(a)) = f(t(a)),$$

where a and b are shapes from the Boolean part of U_{ij} and t is a transformation from the group part of U_{ij}.

Because every shape is guaranteed to be a part of itself and to have 0 as a part we can take operator f to be an identity $f(a) = a$ or a constant $f(a) = 0$ and get algebras A_{ij} and C_{ij}, respectively. In both cases, f satisfies all five of the conditions above. We may also try some other ways of defining f.

For example, we can pick parts of a by using some fixed shape c so that $f(a) = a \cdot c$, or alternatively $f(a) = a - c$. In both cases, f satisfies first four of the above conditions, rendering the related sets of equivalence classes, algebras isomorphic, with the Boolean part of U_{ij}. However, f fails the fifth condition, and the related sets are not closed under transformations of the group part of U_{ij}.

To satisfy the fifth condition we may choose f so that it involves transformations. For example, $f(a) = a \cdot \tau(a)$, or alternatively $f(a) = a - \tau(a)$, where τ is a transformation defined in a coordinate system local to a. When a is transformed, so is its local coordinate system. If a is represented by an equivalence class $[\{a, a \cdot \tau(a)\}]_{\equiv}$, then $t(a)$ is represented by $[\{t(a), t(a) \cdot t \circ \tau \circ t^{-1}(t(a))\}]_{\equiv} = [\{t(a), t(a) \cdot t \circ \tau(a)\}]_{\equiv}$, where $\circ$ is a group composition. Alternatively, if a is represented by $[\{a, a - \tau(a)\}]_{\equiv}$, then $t(a)$ is represented by $[\{t(a), t(a) - t \circ \tau(a)\}]_{\equiv}$. Although f satisfies the first and fifth condition, it fails the second and the third one. We may choose for τ to be defined in a global coordinate system. Thus, if a is represented by $[\{a, a \cdot \tau(a)\}]_{\equiv}$, or by $[\{a, a - \tau(a)\}]_{\equiv}$, then $t(a)$ is represented by $[\{t(a), t(a) \cdot \tau(t(a))\}]_{\equiv} = [\{t(a), t(a) \cdot \tau \circ t(a)\}]_{\equiv}$ or $[\{t(a), t(a) - \tau \circ t(a)\}]_{\equiv}$, respectively. This again satisfies the first four conditions but fails the last one because $\tau \circ t(a)$ may differ from $t \circ \tau(a)$.

Although all of the possibilities for f have not been exhausted, we may speculate with a great certainty that $f(a) = a$ and $f(a) = 0$ are the only solutions that satisfy all of the conditions above. Thus, we have the following conjecture.

CONJECTURE 1. *Algebras A_{ij} and C_{ij}, which are subalgebras of $\wp^*(U_{ij})/\equiv$ isomorphic with U_{ij}, are the only such subalgebras.* ■

Conjecture 1 singles out both: algebra C_{ij}, as the only nontrivial algebra for analyzed shapes, and also decompositions that recognize the shape they analyze as well as the empty shape as the only meaningful shape approximations.

3.4. Properties of decompositions

Thus far, different algebras of decompositions were constructed and analyzed. Now, attention will be shifted to different decompositions of shapes emerging with these algebras. Seven algebras D_{ij}, $\wp'(U_{ij})$, $\wp^*(U_{ij})$, $\wp^*(U_{ij})/\equiv$, A_{ij}, $\wp(K_{ij})$, and C_{ij}, have been defined. The first four compute with general/unstructured decompositions, whereas the remaining three use different kinds of structured decompositions.

The structure of decompositions unfolds on two levels: local, exposing the relations between the elements of a

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

decomposition; and global, relating these elements to the parts of the shape analyzed by the decomposition. If a decomposition is to be an approximation of a shape, then its structure on the global level is of the most importance. All parts of the shape should be represented by the elements of the decomposition. There are infinitely many shape parts, but only finitely many elements so that the relation between the two is many to one.

For a given shape $\boldsymbol{a}$, its part $\boldsymbol{x}$, and its decomposition A, this relation is as follows. There may exist two elements $\boldsymbol{y}$ and $\boldsymbol{z}$ in A such that $\boldsymbol{y}$ is a part of $\boldsymbol{x}$ and $\boldsymbol{x}$ is a part of $\boldsymbol{z}$, or $\boldsymbol{y} \leq \boldsymbol{x} \leq \boldsymbol{z}$. Decomposition A approximates part $\boldsymbol{x}$ by means of elements $\boldsymbol{y}$ and $\boldsymbol{z}$. If $\boldsymbol{y}$ exists, then $\boldsymbol{x}$ is *bottomed* by $\boldsymbol{y}$, or $\boldsymbol{x}$ has at least properties of $\boldsymbol{y}$. If $\boldsymbol{z}$ exists, then $\boldsymbol{x}$ is *topped* by $\boldsymbol{z}$, or $\boldsymbol{x}$ has at most properties of $\boldsymbol{z}$. If both elements exist, then $\boldsymbol{x}$ is *bounded* by $\boldsymbol{y}$ and $\boldsymbol{z}$, or $\boldsymbol{x}$ has at least properties of $\boldsymbol{y}$ and at most properties of $\boldsymbol{z}$. If neither $\boldsymbol{y}$ nor $\boldsymbol{z}$ exist, then $\boldsymbol{x}$ is not recognized by decomposition A. All elements of A, when seen as parts of $\boldsymbol{a}$, are bounded by themselves.

Algebra $\boldsymbol{A}_{ij}$ works with *singleton* decompositions that are singleton sets of Δ_{ij}. These contain only the shape they decompose and nothing else. Each part of the shape is trivially represented, topped, by the shape itself (i.e., it has at most properties of the shape).

In set algebras, where shape parts are of the most importance, singletons inform that the shape is a part of itself. Singletons are less informative in complex algebras. The former depend on shape structures, but singletons by themselves do not structure shapes. Singletons and their algebras simply demonstrate that complex algebras behave well in a boundary case. They do so by validating all of the identities of $\boldsymbol{U}_{ij}$, as shown in Proposition 4.

Algebra $\wp(\boldsymbol{K}_{ij})$ uses *discrete* decompositions, or subsets of a finite set K_{ij} of pairwise discrete shapes. No two elements of such a decomposition have common parts. Approximating shapes with discrete decompositions is thrifty, with no redundancy or ambiguity. The only shape parts that are bounded are the elements of a decomposition. All other parts are topped by only one element, bottomed by one and possibly more elements, or not recognized at all. Any part of the shape that consists of proper parts of two or more elements of the decomposition is not recognized.

The set in Figure 10a is an example of a discrete decomposition of the shape in Figure 3a, whereas the shapes in Figure 10b, c, and d are its proper parts. The shape in Figure 10b is topped by the first element of decomposition a, shape c is bottomed by both the second and the third element of decomposition a, whereas shape d is not recognized by a.

Discrete decompositions are good representations of finished designs ready to be assembled from the parts that are elements of the decompositions.

By including the sums of the original elements, a discrete decomposition could extend to a hierarchical structure. Hierarchies are often used in design and elsewhere because their straightforward structure shows not only the parts of a shape, but also the way the parts are put together to assemble the shape.

The algebra of analyzed shapes $\boldsymbol{C}_{ij}$ computes with *bounded* decompositions, which are, as shown earlier, the only meaningful shape approximations. These recognize the shape they analyze and the empty shape and can include other shape parts besides the two. At a minimum, no other parts are included. Every shape part is guaranteed to be bounded (represented) by such a decomposition. However, at a minimum, this representation is a rather trivial one. At most, a part has properties of the shape itself, and at least, no properties at all. The other elements a bounded decomposition may contain contribute to a finer shape part representation. Shapes in Figure 5b and c are examples of bounded decompositions.

The two shapes recognized in a minimum bounded decomposition are important from the algebraic point of view. They work in computations to preserve structures and shape parts recognized by decompositions.

For example, if two shapes $\boldsymbol{a}$ and $\boldsymbol{b}$ are analyzed by their respective bounded decompositions A and B and if the product $A \cdot^{*} B$ is taken, then shape $\boldsymbol{a}$ from A combines with elements of B to preserve the structure recognized by B, whereas shape $\boldsymbol{b}$ from B combines with the elements of A to preserve the structure recognized by A. If $\boldsymbol{b} \leq \boldsymbol{a}$, then $\boldsymbol{a}$ preserves elements of B as well. Similarly, $\boldsymbol{b}$ preserves elements of A if $\boldsymbol{b} \leq \boldsymbol{a}$. The empty shape has the same function in sums: the one in A preserves elements of B and vice versa. Both $\boldsymbol{a}$ and $\boldsymbol{b}$ assure that the product, sum, or difference of A and B contains the shape it analyzes: $\boldsymbol{a} + \boldsymbol{b}$, $\boldsymbol{a} \cdot \boldsymbol{b}$, or $\boldsymbol{a} - \boldsymbol{b}$, respectively.

Bounded decomposition may serve as the basis for creating more structured decompositions of shapes. They may satisfy additional conditions to become lattices, topologies, groups, and Boolean algebras.

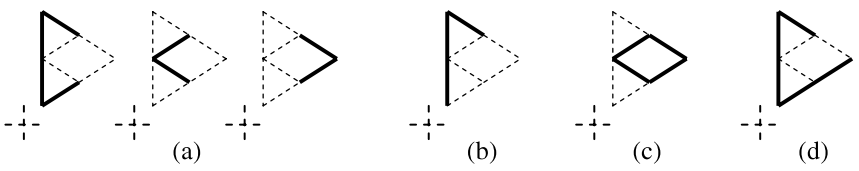

Fig. 10. (a) A discrete decomposition of the shape in Figure 3a, with shape parts that are (b) topped, (c) bottomed, or (d) not recognized by decomposition (a).

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

4. DISCUSSION

If algebras for shapes are good models of how shapes are used in design, then algebras of decompositions may inform on how analyzed shapes are utilized. Table 1 sorts all of the algebras for decompositions constructed here and highlights the ones suitable for computing with analyzed shapes. It also exposes relationships among the algebras and shows how they relate to shape algebras.

Unstructured decompositions are not suitable as approximations of shapes. Neither set nor complex algebras of general decompositions could be used to duplicate computations with shapes. This could only be achieved with decompositions structured in some ways. Two such structures were distinguished: discrete decompositions and bounded decompositions.

The structure of a discrete decomposition is minimum in the sense that there is no overlap between the elements. The same amount of material would be used for making all of the elements of a discrete decomposition as for making the shape itself.

However, bounded decompositions are far from being minimum because they always contain the shape they analyze and may contain some other parts as well. Although their structure comes from purely algebraic considerations, it is not without intuitive appeal. The presence of both, the shape analyzed by a decomposition and the empty shape, provides certain context for the other shape parts recognized by the decomposition.

The former shape assures that the parts are always seen in the context of the whole. This local context is given explicitly by the fact that the whole is a member of the decomposition. If a description is associated with a bounded decomposition, it contains the name of the shape analyzed by the decomposition. In this sense, bounded decompositions are *named*.

The empty shape in a bounded decomposition implies the global context in which the analyzed shape is placed. It shifts attention from shape parts to the shape surroundings, which are implied by the absence of the parts. In an associated description, the empty shape may map to a description of the shape surroundings.

Algebra $\boldsymbol{C_{ij}}$ suggests that both local and global contexts are necessary for decompositions to be successfully used as shape approximations. How intuitive is this? Do we analyze (see) shapes with their parts always related to both the whole and its surroundings? These questions are interesting in their own right, but need to be decided by cognitive psychologists.

For our part, future research should concentrate on moving from general decompositions to ones tailored to specific applications. This may lead to decompositions with more elaborate structures.

As mentioned earlier, both discrete and bounded decompositions could be extended to more structured entities. The elements of such decompositions may form different algebras. The latter could then be combined within the framework of algebras capable of preserving the algebraic structure of decompositions.

For example, Stiny (1994, 2001) computes with decompositions that are topologies and Boolean algebras, whereas Krstic (1996) does it with decompositions that are lattices, hierarchies, topologies, Boolean algebras, and Boolean algebras with operators. Such decompositions are able to handle a variety of problems, from continuity of computations in shape grammars, to recasting spatial computations with shapes into nonspatial ones with symbols.

4.1. A note on implementation

Although this work is theoretical, the results obtained could lead to practical applications. It would be advantageous to

Table 1. *Algebras for decompositions*

Algebra		Decompositions		Properties					
Name	Symbol	Type	Structure	Subalgebra of	Homomorphic Image	Isomorphic With	Additional Condition	Identities of U_{ij} Preserved	Shape Approximations
Set	D_{ij}	General	None	NA	NA	NA	No	Yes	No
Complex	$\wp'(U_{ij})$	General	None	NA	NA	NA	Yes	No	No
Normalized complex	$\wp^*(U_{ij})$	General	None	NA	U_{ij}	NA	Yes	No	No
Quotient	$\wp^*(U_{ij})/\equiv$	Equivalence class	None	NA	U_{ij}	NA	Yes	Yes	No
Kit of parts	$\wp(K_{ij})$	Discrete	Discrete elements	D_{ij}	$\Sigma\,\wp(K_{ij})$	$\Sigma\,\wp(K_{ij})$	No	Yes	Yes
Singletons	A_{ij}	Singleton	$\{a\}$ $[\{a\}]_{\equiv}$	$\wp'(U_{ij})$ $\wp^*(U_{ij})$ $\wp^*(U_{ij})/\equiv$	U_{ij}	U_{ij}	Yes	Yes	Yes (trivial)
Analyzed shapes	C_{ij}	Bounded	$\{a, \ldots 0\}$ $[\{a, 0\}]_{\equiv}$	$\wp^*(U_{ij})/\equiv$	U_{ij}	U_{ij}	Yes	Yes	Yes

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

be able to experiment with shape decompositions used as shape approximations, or analyzed shapes. This way the assumptions we make about shapes and their parts will be included in computations with shapes and affect the results of such computations.

Major domains of implementation for shape algebras are shape grammars (production systems), which use rewriting rules to generate designs. Grammars were introduced over three decades ago by George Stiny and James Gips, and were successfully used to generate new design styles and analyze the existing styles (Stiny, 2001). Computer implementations of shape grammars are many, ranging from general grammar interpreters (Krishnamurti, 1982; Krishnamurti, & Giraud, 1986; Chase, 1989; Piazzalunga & Fitzhorn, 1998; Tapia, 1999) to ones specific to certain design styles (Flemming, 1987; Agrawal & Cagan, 1998).

Algebras developed here should serve as a framework for a new kind of grammars for analyzed shapes (shape decompositions), thus extending the shape grammar formalism. Computer implementations of such grammars could be based on the existing shape grammar interpreters because operations of algebras $\wp'(\boldsymbol{U}_{ij})$, $\wp^*(\boldsymbol{U}_{ij})$, $\wp^*(\boldsymbol{U}_{ij})/\equiv$, $\boldsymbol{A}_{ij}$, and $\boldsymbol{C}_{ij}$ are extensions of shape operations ($+$, $\cdot$, and $-$) to direct products of decompositions. Such an operation can be computed by computing a number of shape operations.

For example, to compute the sum or product of decompositions A and B in $\wp'(\boldsymbol{U}_{ij})$, $\wp^*(\boldsymbol{U}_{ij})$, or $\wp^*(\boldsymbol{U}_{ij})/\equiv$, we need to compute card$A$card$B$ corresponding shape operations, where card$A$ and card$B$ are cardinalities of the respective decompositions. The same number of shape operations is needed to compute a difference in $\wp'(\boldsymbol{U}_{ij})$. However, for difference $A - B$ in $\wp^*(\boldsymbol{U}_{ij})$ or $\wp^*(\boldsymbol{U}_{ij})/\equiv$, only card$A$ shape operations is needed if the shape B analyzes is known. If this shape is not known, another card$B$ operations is needed to compute ΣB, which amounts to total of card$A$ + card$B$ operations for the difference. To transform a decomposition A we need card$A$ shape transformations. In $\wp^*(\boldsymbol{U}_{ij})/\equiv$, the number of shape operations needed to compute a single analyzed shape operation could be further reduced if the unique smallest representatives for equivalence classes are used (Corollary 2 of Proposition 7), which for $\boldsymbol{C}_{ij}$ will amount to just one shape operation. This is not surprising because $\boldsymbol{C}_{ij}$ and $\boldsymbol{U}_{ij}$ are isomorphic. However, by using the smallest representatives in $\boldsymbol{C}_{ij}$, all of the distinguished shape parts that are not included in the representatives are lost in computations. Therefore, there is no advantage in computing with decompositions instead of shapes themselves. Even without using the smallest representatives, computations could be reduced because bounded decompositions of $\boldsymbol{C}_{ij}$ are guaranteed to contain empty shape so that shape operations involving it may be omitted. For the sum or product of decompositions A and B in $\boldsymbol{C}_{ij}$ we need (card$A$ − 1)(card$B$ − 1) shape operations; and for the difference $A - B$ or transformation $\boldsymbol{t}(A)$, card$A$ − 1 shape operations. Note that in the case of difference above there is no need to compute ΣB because it is already an element of B, and one only needs to keep track of it. The latter task is not difficult because of isomorphism between $\boldsymbol{C}_{ij}$ and $\boldsymbol{U}_{ij}$. If one knows ΣA and ΣB, then computing $\Sigma(A + B)$ is simple because it is equal to $\Sigma A + \Sigma B$. The same is true for products and differences, $\Sigma(A \cdot B) = \Sigma A \cdot \Sigma B$ and $\Sigma(A - B) = \Sigma A - \Sigma B$.

A shape generation with a shape grammar starts with an initial shape, which is then gradually changed by recursive applications of shape rules. Similarly, an analyzed shape generation should start with an initial analyzed shape (shape decomposition). To change an analyzed shape C with a shape rule $A \rightarrow B$, which replaces an analyzed shape A with an analyzed shape B, one needs a transformation $\boldsymbol{t}$ under which A becomes a part of C, or $\boldsymbol{t}(A) \leq' C$. If such $\boldsymbol{t}$ exists, then a transformed analyzed shape $\boldsymbol{t}(A)$ is removed from C and replaced with the transformed analyzed shape $\boldsymbol{t}(B)$ to create a new analyzed shape C'. This can be accomplished by computing four operations from an algebra for analyzed shapes, in accordance with the following expression.

$$C' = [C - \boldsymbol{t}(A)] + \boldsymbol{t}(B).$$

To implement the expression above in $\boldsymbol{C}_{ij}$, one needs to compute card$A$ − 1 of shape operations for $\boldsymbol{t}(A)$, card$B$ − 1 for $\boldsymbol{t}(B)$, card$C$ − 1 for $C - \boldsymbol{t}(A)$ and another (card$C$ − 1) (card$B$ − 1) for the sum, which totals card$B$card$C$ + card$A$ − 2 of shape operations. This is $k = 1/4$ (card$B$card$C$ + card$A$ − 2) times more shape operations per rule application than is needed in standard shape grammars. Note that $k = 1$ if decompositions in the computations are of cardinality 2, which means at they are the smallest representatives. This is expected because $\boldsymbol{C}_{ij}$ and $\boldsymbol{U}_{ij}$ are isomorphic. If we take n for an average number of elements for decompositions in computations, then $k = (n/2)^2$ is an approximation with less then 2% error for $n > 50$. Computation time is proportional to the square of cardinalities of decompositions in computations. The time could be reduced with the aid of parallel computing. Operations in $\boldsymbol{C}_{ij}$ are extensions of shape operations to direct products of decompositions and are done componentwise, which may be computed in parallel.

Subshape evaluations necessary to determine transformation $\boldsymbol{t}$ are computationally the most complex operations a shape grammar interpreter faces. However, no complexity is added for evaluating decompositions of $\boldsymbol{C}_{ij}$. Because $\boldsymbol{C}_{ij}$ and $\boldsymbol{U}_{ij}$ are isomorphic, $\boldsymbol{t}(A) \leq' C$ if and only if $\boldsymbol{t}(\Sigma A) \leq \Sigma C$ so that shapes ΣA and ΣC could be evaluated instead of their decompositions A and C.

We have not elaborated on implementations of algebras $\boldsymbol{A}_{ij}$, $\boldsymbol{D}_{ij}$, and $\wp(\boldsymbol{K}_{ij})$ as there are no computational issues with these algebras. Decompositions of the first algebra are, for any practical application, shapes themselves whereas operations of the last two algebras are set operations, which are easier to implement than standard shape operations.

https://doi.org/10.1017/S0890060405050183 Published online by Cambridge University Press

REFERENCES

Agrawal, M., & Cagan, J. (1998). A blend of different tastes: the language of coffee makers. *Environment and Planning B: Planning and Design 25*, 205–226.

Birkhoff, G. (1993). *Lattice Theory*. Providence, RH: American Mathematical Society.

Bleicher, M.N., Schneider, H., & Wilson, R.L. (1973). Permanence of identities on algebras. *Algebra Universalis 3*, 72–93.

Carlson, C., Woodbury, R., & McKelvey, R. (1991). An introduction to structure and structure grammars. *Environment and Planning B: Planning and Design 18*, 417–426.

Chase, S.C. (1989). Shapes and shape grammars: from mathematical model to computer implementation. *Environment and Planning B: Planning and Design 16*, 215–242.

Earl, C. (1997). Shape boundaries. *Environment and Planning B: Planning and Design 24*, 668–687.

Flemming, U. (1987). The role of shape grammars in analysis and creation of designs. In *Computability of Designs* (Kaley, Y.E., Ed.), pp. 245–272. New York: Wiley.

Gratzer, G. (1979). *Universal Algebra*. Providence, RI: American Mathematical Society.

Gratzer, G., & Whitney, S. (1978). Infinitary varieties of structures closed under the formation of complex structures [Abstract]. *Notices of the American Mathematical Society 25*, A224.

Guatam, N. (1957). The validity of equations of complex algebras. *Archives Mathematic Logik Grundlagenforch 3*, 117–124.

Knight, T. (1988). Comparing designs. *Environment and Planning B: Planning and Design 7*, 73–110.

Knight, T. (2003*a*). Computing with emergence. *Environment and Planning B: Planning and Design 30*, 125–155.

Knight, T. (2003*b*). Computing with ambiguity. *Environment and Planning B: Planning and Design 30*, 165–180.

Krishnamurti, R. (1982). *SGI: A Grammar Interpreter* [Research Report]. Centre for Configurational Studies, The Open University, Milton Keynes.

Krishnamurti, R., & Giraud C. (1986). Towards a shape editor: the implementation of a shape generation system. *Environment and Planning B: Planning and Design 13*, 391–403.

Krstic, D. (1996). *Decompositions of shapes*. PhD Thesis. University of California, Los Angeles.

Krstic, D. (1999). Constructing algebras of design. *Environment and Planning B: Planning and Design 26*, 45–57.

Krstic, D. (2001). Algebras and grammars for shapes and their boundaries. *Environment and Planning B: Planning and Design 28*, 151–162.

Krstic, D. (2004). Computing with analyzed shapes. In *Design Computing and Cognition '04* (Gero, J.S., Ed.), pp. 397–416. Dordrecht: Kluwer Academic.

Piazzalunga, U., & Fitzhorn, P.I. (1998). Note on three-dimensional shape grammar interpreter. *Environment and Planning B: Planning and Design 25*, 11–33.

Shafaat, A. (1974). On varieties closed under construction of power algebras. *Bulletin Australian Mathematical Society 11*, 213–218.

Stiny, G. (1980). Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design 7*, 243–251.

Stiny, G. (1982). Spatial relations and grammars. *Environment and Planning B: Planning and Design 9*, 113–114.

Stiny, G. (1990). What is a design. *Environment and Planning B: Planning and Design 17*, 97–103.

Stiny, G. (1991). The algebras of design. *Research in Engineering Design 2*, 171–181.

Stiny, G. (1992). Weights. *Environment and Planning B: Planning and Design 19*, 413–430.

Stiny, G. (1994). Shape rules: closure, continuity, and emergence. *Environment and Planning B: Planning and Design 21*, S49–S78.

Stiny, G. (2001). How to calculate with shapes. In *Formal Engineering Design Synthesis* (Antonson, E.K., & Cagan J., Eds.), pp. 24–60. Cambridge: Cambridge University Press.

Tapia, M.A. (1999). A visual implementation of a shape grammar system. *Environment and Planning B: Planning and Design 26*, 59–73.

Vickers, S. (1989). *Topology Via Logic*. New York: Cambridge University Press.

Djordje Krstic holds a BA and MA from Belgrade University and a PhD from UCLA. He taught at Belgrade University and is currently a Principal Software Engineer at Alcatel USA. Dr. Kristic has been performing research in the field of computational design theory since 1988. His main research interests are in shape grammars, algebras of design, and space syntax.