Geoinformatica (2014) 18:595–620
DOI 10.1007/s10707-013-0191-1

Comparing *G*-maps with other topological data structures

Patrick Erik Bradley · Norbert Paul

Received: 26 March 2010 / Revised: 11 January 2012 /
Accepted: 15 September 2013 / Published online: 26 October 2013

Abstract This article compares two approaches to storing spatial information: On the one hand there are topological datatypes where primitives and their connectivity are *explicitly* stored, on the other hand there is the *G*-maps-approach storing abstract "darts" and groups acting on these darts such that their orbits implicitly give the elements and topology of the stored space. First these concepts are mutually related from a categorial viewpoint and, second, their storage complexity is compared.

Keywords *G*-maps · Spatial modelling · Topology · Storage complexity

1 Introduction

Storage of spatial information is usually done by providing elementary shapes, such as lines, vertices, and faces, together with additional connectivity information turning these shapes into a topological space. Sometimes, these shapes—which are often called "primitives"—are only given in an implicit manner as, for example, in the definition of a polygon by a sequence of vertices in which case two consecutive vertices implicitly constitute an edge. The most generic topological data modeling concept is what the authors call "topological datatype": Explicit storage of all

P. E. Bradley
Institut für Photogrammetrie und Fernerkundung,
Karlsruhe Institute of Technology (KIT),
Kaiserstraße 12, 76128 Karlsruhe, Germany
e-mail: bradley@kit.edu

N. Paul (✉)
Geodätisches Institut Karlsruhe,
Karlsruhe Institute of Technology (KIT),
Kaiserstraße 7, 76128 Karlsruhe, Germany
e-mail: norbert.paul@kit.edu

primitives into one set and storing their incidence into a relation which defines the topology—the well-known incidence graph.

Every data structure which defines a topology for a finite set of elements implicitly stores that incidence relation, and, on the other hand, such incidence relation uniquely determines the topology. As incidence storage is asymptotically optimal, too, we consider it a canonical representative of all data models for topological modelling.

An extreme counter-example is the G-maps approach where all primitives are implicitly stored: The explicitly given objects are abstract "darts" upon which some involutionary groups act. A primitive is then a subset (or orbit) of these darts linked by all but one of these involutions and the topological information is found by a combination of set intersections together with additionally provided dimension information. There exists extensive literature on G-maps such as the original publication [7] and more recent works as [4]. The latter also mentions "implicit" versus "explicit" storage of G-map pyramids. However, such an "implicitly" stored G-map pyramid consists of an *explicitly* stored G-map together with additional data in order to generate the desired pyramid atop that G-map.

This article compares the different approaches with regard to effectiveness and efficiency. Effectiveness of a data structure means that there is a mapping from the instances of some abstract model of "real world" objects—in our case finite topological spaces—into instances of such data structure which distinguishes two different models. Hence, that mapping is essentially injective. If both the "real world" abstraction and the data structure can be considered categories this comparison can be easily done by means of category theory. Effectiveness of data modeling is then nothing else than having an essentially injective mapping (a functor) from the real world category into the data model's category. To consider G-maps a category, however, there must be a definition of the morphisms. This has not yet been done and so we will provide such morphisms here. The morphisms of topological datatypes, of course, are the continuous maps.

On the other hand, not only effectiveness but also efficiency of different data structures is an important issue. We consider here only storage space complexity and will not discuss time complexity of operators on these data structures. What we call "topological data types" are already optimal for arbitrary finite topological spaces, so their complexity will be compared to the storage complexity of G-maps of arbitrary dimension. As a new result a tight storage complexity lower bound will be provided. It turns out that G-maps are an extremely inefficient way to store spatial data.

This article first introduces the notion of G-maps in Section 2 and relates them to Coxeter groups. Then a definition for morphisms of G-maps is proposed so that now there is a category of G-maps. Later cells and cell-tuples of a G-map are introduced and the important result that the cell-tuple map from a G-map to its underlying topological space is surjective is proven. The section concludes with a characterisation of cellular G-map morphisms.

Section 3 introduces the notion of "topological datatype", an abstract relational representation of finite topological spaces. It is shown that the functor from G-maps to topological datatypes is not essentially surjective which means that there are topological datatypes which cannot be represented by a G-map. Therefore G-map is not an effective data model for arbitrary topological spaces. In particular, graphs which are an important class of topological spaces, in general, do not have a G-map representation.

Section 4 introduces the class of Alexandrov spaces which are exactly the spaces which can be modeled by topological datatypes. It is deduced that topological datatypes are storage-space optimal and, hence, a space-efficient model for topological models with storage-space complexity $\mathcal{O}(n^2)$ where n is the number of "primitives". On the other hand, by using the surjectivity result from Section 3, it turns out that G-maps are extremely space-inefficient: Their storage-space exceeds every polynomial upper bound.

2 Categories of *G*-maps

2.1 Motivation

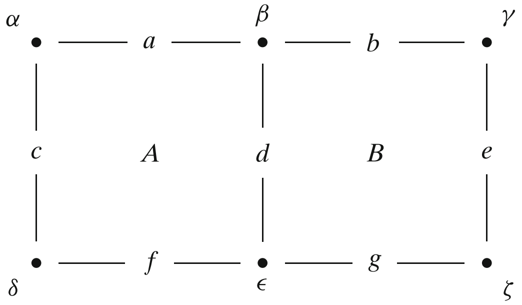
Spatial data is mostly modelled by defining and storing a topological structure. The topology is defined by specifying a partitioning of the space in question into areas, edges, and vertices. Areas are then bounded by edges, and edges are connected to vertices. Additional geometric information specifies the exact location of vertices and boundaries and thereby implicitly defines the shapes of the areas, too.

Such two-dimensional data structures can also be used to define volume objects by embedding them into 3D thus specifying volume surfaces. Such surface must then be cyclical and "wrap around" some "cavity" which represents the volume. These volume modelling concepts are called boundary-representation (or short "B-rep") models [8]. There are many different proposals for data structures in 2D, 3D, and even for arbitrary dimension. This raises the question what could be an adequate representation of higher-dimensional spatial data and whether there does exist an underlying principle common to all these spatial data structures. The main practical advantage of knowing such principle is that it might help to integrate different models such as geo-information, civil-engineering structures, and architectural models. Additionally, such a principle could be useful to assess modelling alternatives.

Now two spatial data models can be considered "canonical" because, first, they are a lossless representation of every finite topology and, second, they are optimal with respect to storage space needed [2, 9, 10]. We call one of these models "topological datatype" and it merely consists of the class of the so-called incidence graphs. Together with the continuous functions as morphisms they form a category **DTop**. The other model, the "relational complexes", is a category **DChainComp** where the edges of the incidence graph are labelled by integers to express additional orientation information: If, for example, a boundary face of some volume is looking "towards" that volume the corresponding face-volume-incidence may be labelled by -1, whereas, in case a boundary face looked "away from" such volume, the volume-face-incidence could be labelled by $+1$. This orientation labelling is arbitrary, but must be consistent as explained in [2]. Note that other labellings are possible, too. For example, "multi-incidence" information can be stored by attaching an incidence number to each edge of the incidence graph.

Curiously, a group-theoretic study of 2D spatial structure was undertaken in the study of arithmetic properties of Riemann surfaces, where the so-called "cartographic group" was introduced [11]. An implementation of this cartographic group can be found under the name 2-*map* in [6]. This generalises to N-*map* in [7].

Example 1 (Cartographer's Algebra) Assume some novice cartographer's first drawing of that simple map given below: Two neighbouring square countries A and B have a common boundary d.



Now, consider a path (β, d, A) in the incidence graph of that space. It starts with vertex β, steps up to the incident edge d and ends with the incident area A. Now, if we wanted to replace some element, say β, at this path by something else, there exists exactly one alternative path where only that given element is replaced, namely (ϵ, d, A). If, however, that other replacement element ϵ is to be replaced, too, the only possibility is returning to our original path.

Now each such vertex-edge-face path in our example has at each position at most one such replacement element. Note that A cannot be replaced in (α, a, A), because no other face than A is incident to the edge a.

There exist spaces where replacement elements in incidence paths are not unique. But if such uniqueness is given, then, instead of taking an element, one could also fix a *position* i for all these paths and then every incidence path has at most one "bypass" at its ith position. So there is a function α_i which either sends an incidence path to such bypass at position i or leaves the path unchanged if that bypass does not exist. Applying α_i again returns the original path. Such a function, where $\alpha_i(\alpha_i(x)) = x$ holds for all elements x, is called an *involution*.

If such involutions α_i exist for every dimension number i ranging from zero to the dimension n of the given space, they generate permutation groups which uniquely determine the space and, hence, can also be used as a topological data model for arbitrary dimension—an algebraic approach to spatial modelling which is called G-map [7].

Now, if one is to integrate spatial data from different sources, one has to know how to convert between data structures. As we aim at laying the grounds for a generic spatial data modelling tool, we advocate a discussion of the advantages and disadvantages of different approaches with respect to effectiveness and efficiency.

2.2 G-maps and representations

Let us first reproduce the definition from [7, Definition 1]:

Definition 1 An *n-G-map* is a tuple $X = (D, a_0, \ldots, a_n)$ such that

– D is a finite set of so-called *darts*.
– $a_i\colon D \to D$ is an involution for $i = 0, \ldots, n$.

– For i, j satisfying $0 \leq i + 2 \leq j \leq n$ the composition $a_i \circ a_j$ is an involution.

A *G-map* is an n-G-map for some $n \in \mathbb{N}$.

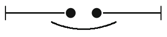
In the above definition, the a_i can be viewed as elements of the group S_D of permutations of the set D. The definition says something about the *order* of a composition of involutions which is defined as

$$\operatorname{ord}(a_i \circ a_j) = m_{ij} := \min\left\{k > 0 \mid (a_i \circ a_j)^k = 1\right\},$$

where 1 denotes the identity permutation $x \mapsto x$. The identity permutation is defined to be of order one: $m_{ii} = 1$.

Remark 1 In order to make illustrations, we introduce here Lienhardt's rules of depicting G-maps.

- A dart is depicted thus: •——⊣
- An orbit under $\langle a_0 \rangle$ is depicted in this way: •——⊣ ⊢——•
- An orbit under $\langle a_i \rangle$ for $i > 0$ is depicted by connecting the corresponding two darts by a number of i strokes. E.g. for $i = 1$:



- If a dart is fixed under a_i, then no corresponding line is drawn.

Lienhardt depicts the orbits under $\langle a_0 \rangle$ differently: namely as •——+——•We stay with our depiction for TEX-nical reasons...

Example 2 Figure 1 represents a triangle-shaped G-map with $a_0 \circ a_1$ having order 3.

The orders of the products $a_i \circ a_j$ are encoded in the so-called *Coxeter matrix* (m_{ij}), and we can make our first observation:

Lemma 1 *The Coxeter matrix of a G-map is symmetric: $m_{ij} = m_{ji}$ for all $i, j = 0, \ldots, n$.*

Proof We need to show that $m_{ij} = m_{ji}$. This holds true for general group-theoretic reasons:

$$(a_j a_i)^{m_{ji}} = 1 \tag{1}$$

$$(a_j a_i)^{m_{ij}} = a_j (a_i a_j)^{m_{ij}-1} a_i = a_j (a_i a_j)^{-1} a_i = a_j a_j^{-1} a_i^{-1} a_i = 1. \tag{2}$$

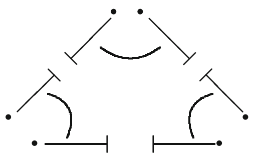


Fig. 1 A triangle-shaped 1-G-map: it has a permutation of order 3

Hence, m_{ji} divides m_{ij}. This follows from standard algebraic considerations: By definition of m_{ji} we have $m_{ji} \leq m_{ij}$, and division with remainder yields

$$m_{ij} = a \cdot m_{ji} + r$$

with $0 \leq r < m_{ji}$. If $r > 0$, then we would have the contradiction:

$$(a_j a_i)^r = (a_j a_i)^{a \cdot m_{ji} + r} = (a_j a_i)^{m_{ij}} = 1,$$

because m_{ji} is the smallest positive natural number m such that $(a_j a_i)^m = 1$. Hence, $r = 0$, and m_{ji} divides m_{ij} as asserted. By the same argument, one proves that m_{ij} divides m_{ji}. This implies the desired equality $m_{ij} = m_{ji}$. □

We will encounter the standard algebraic consideration in the proof above some further time below.

As a result of the observation above, we obtain from the definition of G-maps the following orders:

$$m_{ij} = \begin{cases} 1, & i = j \\ 2, & |i - j| > 1 \\ m_{ij} > 1, & |j - i| = 1, \end{cases} \tag{3}$$

the last case meaning that m_{ij} can be any natural number greater than one if $|j - i| = 1$. In the subgroup of S_D generated by the a_i, there are in general further relations depending on the action of each a_i on D.

In order to make a distinction from the permutation group above, we define the abstract group

$$W = \langle \alpha_0, \ldots, \alpha_n \rangle$$

presented by generators α_i and the relations given by

$$\mathrm{ord}(\alpha_i \alpha_j) = m_{ij},$$

where m_{ij} are numbers satisfying (3). For this choice of m_{ij}, W is a special case of a so-called *Coxeter group*.

Remark 2 These Coxeter groups can be infinite, and in fact many of them are infinite groups. On the other hand, the subgroup of S_D defined by the involutions a_i is certainly finite and satisfies in general more relations than those given by Eq. 3.

In the setting of Coxeter groups, the collection $M = (m_{ij})$ of scalars m_{ij} can be seen as the adjacency matrix of a graph with multiple edges. The *Coxeter graph* of W is defined by taking as vertices the α_i, $i = 0, \ldots, n$, and as edges those pairs (α_i, α_j) with $i \leq j$ such that $m_{ij} > 2$. The edges are labelled by m_{ij} in the case that $m_{ij} > 3$.

Condition (3) implies that the connected components of the Coxeter graph of W have the shape

$$\begin{tikzcd} \overset{\alpha_i}{\bullet} \arrow[r, dash, "m_{i+1}"'] & \overset{\alpha_{i+1}}{\bullet} \arrow[r, dash, "m_{i+2}"'] & \cdots \arrow[r, dash, "m_{i+r}"'] & \overset{\alpha_{i+r}}{\bullet} \end{tikzcd} \tag{4}$$

For this reason, we call a Coxeter group W *of segment type* if the Coxeter diagram of W has its connected components shaped as in Eq. 4.

Given a set D, a map $\phi\colon \{\alpha_0, \dots, \alpha_n\} \to S_D$ possibly extends to a map

$$\phi\colon W \to S_D,$$

where a word w in $\alpha_i, \{\alpha_i\}^{-1}$ is mapped to the corresponding word $\phi(w)$ in $a_i, \{a_i\}^{-1}$, where $a_i := \phi(\alpha_i)$. If such an extension exists, it is unique. It exists if

$$\operatorname{ord}(a_i a_j) \quad \text{divides} \quad \operatorname{ord}(\alpha_i \alpha_j).$$

In this case, the pair $(\{\alpha_0, \dots, \alpha_n\}, \phi)$ is called a *representation* of W. By abuse of notation, we will often write ϕ instead of $(\{\alpha_0, \dots, \alpha_n\}, \phi)$ when speaking of a representation of W. We then hope that it is understood from the context that the generators $\alpha_0, \dots, \alpha_n$ are fixed.

If W is of segment type, then $X_\phi = (D, a_0, \dots, a_n)$ is the G-map *associated* to ϕ. If further

$$\operatorname{ord}(\alpha_i \alpha_j) = \operatorname{ord}(a_i a_j) \tag{5}$$

for all i, j, then ϕ is called a *minimal* representation.

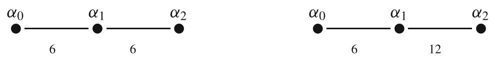
Let $X = (D, a_0, \dots, a_n)$ be an n-G-map. Then a Coxeter group $W = \langle \alpha_0, \dots, \alpha_n \rangle$ of segment type satisfying condition (5), yields a minimal representation

$$\phi_X\colon W \to S_D, \quad \alpha_i \mapsto a_i.$$

We call ϕ_X a *minimal representation of W associated to X*.

For a Coxeter group $W = \langle \alpha_0, \dots, \alpha_n \rangle$ there exists a canonical infinite space on which W acts. This space is the so-called *building* associated with W, and one usually has this space in mind when considering Coxeter groups. We believe that the study of the building associated should be included additionally to the study of the G-map X. Hence, a representation $\phi\colon W \to S_D$ should contain in general more information than the mere tuple $(D, \phi(\alpha_0), \dots, \phi(\alpha_n))$. However, it is not the scope of this article to study buildings of Coxeter groups, but to relate G-maps to topological datatypes.

Example 3 The images in Fig. 2 represent each the G-map $X = (D, a_0, a_1, a_2)$ whose Coxeter graph is given by



(6)

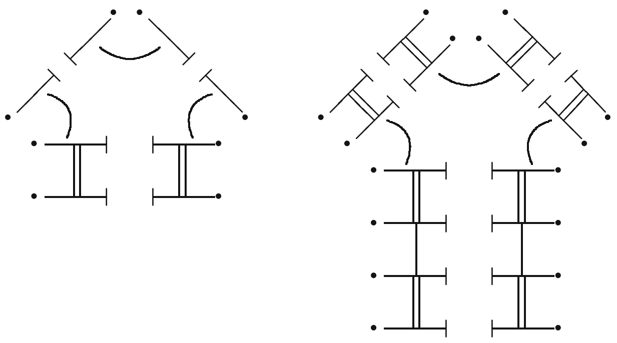


Fig. 2 Two triangles with sockets represented by G-maps

respectively. Observe that in the left G-map, the orbit of some darts under a_1a_2 have length 2 and 3. Hence, $6 = 2 \cdot 3$ is the order of a_1a_2. This can, of course, be checked easily by hand. The right G-map has orbits under a_1a_2 of length 4 and 6. This results in

$$\mathrm{ord}(\alpha_1\alpha_2) = \mathrm{lcm}(4, 6) = 12,$$

where lcm denotes the *least common multiple* of integers. This can again be checked by hand. The minimal representations are given by

$$W = \langle \alpha_0, \alpha_1, \alpha_2 \rangle \to \langle a_0, a_1, a_2 \rangle, \quad \alpha_i \mapsto a_i$$

where W is the Coxeter group with Coxeter graph given by Eq. 6, respectively.

From the previous example we learn that the entries m_{ij} of the Coxeter matrix are given by "local" contributions as orbit lengths:

Lemma 2 *The matrix entry m_{ij} equals the* lcm *of all orbit lengths of a_ia_j.*

Proof Let $a = a_ia_j$. First notice that the length of the orbit $O = \langle a \rangle\, d$ of some $d \in D$ equals the order of the restriction $a|_O$ of the permutation a to the orbit O. This means we are done when we show that

$$\mathrm{ord}(a) = s := \mathrm{lcm}\,\{\mathrm{ord}(a|_O) \mid O \in \mathrm{Orb}(a)\},$$

where $\mathrm{Orb}(a)$ denotes the set of all orbits $\langle a \rangle\, d$ for every d in D. At any rate, $\mathrm{ord}(a)$ divides s. The reason is that

$$a^s = 1,$$

because s is a multiple of any $\mathrm{ord}(a|_O)$, and hence a^s takes any $d \in O$ to itself. Since any number t with $a^t(d) = d$ for all $d \in D$ is necessarily a common multiple of all $\mathrm{ord}(a|_O)$, it follows that s divides $t = \mathrm{ord}(a)$. Hence, $\mathrm{ord}(a) = s$. □

2.3 Morphisms and categories

So far, from a categorical viewpoint, G-maps exist only as objects. In order to form a category, it is necessary to define morphisms of G-maps. The aim of this subsection is to fill this gap and propose a definition of a category of G-maps.

Definition 2 A *morphism* $F\colon X \to X'$ of G-maps $X = (D, a_0, \ldots, a_n)$ and $X' = (D', a'_0, \ldots, a'_m)$ is a pair (Φ, f), where $\Phi(a_i) \in \{a'_0, \ldots, a'_m\}$ for $i = 0, \ldots, n$, and

$$f \circ a = \Phi(a) \circ f \tag{7}$$

for all $a \in S_D$, and $f\colon D \to D'$ is a mapping.

The first condition is to make sure that morphisms map every involution a_i to an involution a'_j, and condition (7) means that the diagram

$$\begin{tikzcd} D \arrow[r, "f"] \arrow[d, "a"'] & D' \arrow[d, "\Phi(a)"] \\ D \arrow[r, "f"'] & D' \end{tikzcd}$$

is commutative.

Definition 3 The *category of G-maps* is given by G-maps and morphisms of G-maps, and is denoted by G**-map**.

It is readily seen that G**-map** is indeed a category.

2.4 Cells and darts of G-maps

Lienhardt [7] defines the notion of "cell" in an n-G-map $X = (D, a_0, \dots, a_n)$ as being the $n-1$-G-map $(Wx, a_0, \dots, \check{a}_i, \dots, a_n)$ for some orbit Wx, $x \in D$. By $\check{a}_i$ we mean the omission of a_i. Since there is no topological structure on G-maps *a priori*, this could be considered an abuse of language. Apart from this, the definition does not "remember" which involution a_i of X has been omitted in the new G-map. In other words, if the actions of a_i and a_j on the orbit Wx coincide, there is no difference between the i-cell and the j-cell determined by x, even if $i \neq j$.

However, in terms of representations, there is a natural action of W on the orbit Wx defined by the representation $\phi\colon W \to S_D$. Omission of α_i yields the induced representation

$$\phi_i\colon \langle \alpha_0, \dots, \check{\alpha}_i, \dots, \alpha_n \rangle =: W_i \to S_{Wx}$$

whose associated G-map coincides with Lienhardt's notion of i-cell.

The problem above can be remedied by considering as i-cell the G-map

$$(Wx, a_0, \dots, 1, \dots, a_n), \tag{8}$$

where in the i-th place the a_i of X has been replaced by 1.

Definition 4 Let $X = (D, a_0, \dots, a_n)$ be G-map. Then the *i-cell $C_i(x)$ containing x* is the G-map according to expression Eq. 8. The G-map $X(x) = (Wx, a_0, \dots, a_n)$ is called the *connected component* of $x \in D$. The G-map X is called *connected*, if $X = X(x)$ for some $x \in D$.

Following Lienhardt, an element $x \in D$ is called a *dart*. A first observation is that

$$x \in \bigcap_{i=0}^{n} C_i(x), \tag{9}$$

where $C_i(x)$ are the cells of an n-G-map X for some given n. Hence, a dart $x \in D$ defines a unique *cell tuple*

$$\mathbf{C}(x) = (C_0(x), \dots, C_n(x))$$

which has the defining properties:

1. $\mathbf{K} = (K_0, \dots, K_n)$ is a tuple of i-cells K_i of the G-map X.
2. $K_i \cap K_{i+1} \neq \emptyset$ for $i \in \{1, \dots, n-1\}$.
3. $\mathbf{K}$ is of maximal length (i.e. X is an n-G-map).

Let $\mathrm{CT}(X)$ be the set of all cell tuples $\mathbf{K}$ of X. We obtain a map:

$$\mathbf{c}\colon X \to \mathrm{CT}(X), \quad x \mapsto \mathbf{C}(x)$$

Theorem 1 *For any n-G-map X, the cell tuple map* $\mathbf{c}$ *is surjective.*

This means that for every cell tuple $\mathbf{K}$ of X there is a dart $x \in D$ such that $\mathbf{K} = \mathbf{C}(x)$. The proof of this important theorem will depend on the following observation:

Lemma 3 *If* $|i - j| > 1$*, then the orbit* $\langle \alpha_i, \alpha_j \rangle d$ *has either one, two or four elements. In the case of four elements, neither* α_i *nor* α_j *has a fixed point.*

Proof Certainly, the orbit $O := \langle \alpha_i, \alpha_j \rangle d$ is not empty. Hence, it has at least one element. Since $\alpha_i \alpha_j$ is also an involution, if $|i - j| > 1$, it follows that the orbit has at most four elements.

Assume that O has more than one element. If α_i has a fixed point $a \in O$, then $b = \alpha_j(a)$ is different from a. Otherwise, there is another element $c \in O$. But then the involution $\alpha_i \alpha_j$ cannot take c to a or vice versa, a contradiction. Hence the orbits of $\langle \alpha_j \rangle$ under the action on O all have two elements. Since O is the union of orbits of $\langle \alpha_j \rangle$, it follows that O can have two or four, but not three, elements. □

Proof (Theorem 1) The statement is proven if

$$K_0 \cap \cdots \cap K_n \neq \emptyset$$

for any cell tuple $(K_0, \dots, K_n)$ We prove the assertion by induction on the length m of any sub-tuple $(K_i, \dots, K_j)$ of consecutive cells.

For $n = 0$, the statement is clear: any cell contains a dart.
For $n = 1$, the statement is also clear by property 2.

Assume for $n > 1$ the induction hypothesis that consecutive sub-tuples of length $1 \leq m \leq n - 1$ have a common dart. In particular, $K_0 \cap \cdots \cap K_{n-1}$ and $K_1 \cap \cdots \cap K_n$ are not empty. Also, $K_1 \cap \cdots \cap K_{n-1} \neq \emptyset$ contains a dart d and hence is of the form:

$$K_1 \cap \cdots \cap K_{n-1} = \langle \alpha_0, \alpha_2, \dots, \alpha_n \rangle d \cap \cdots \cap \langle \alpha_0, \dots, \alpha_{n-2}, \alpha_n \rangle d.$$

Each of the above orbits $\langle \alpha_0, \dots, \alpha_n \rangle d$ contains the orbit $\langle \alpha_0, \alpha_n \rangle d$ as a subset and hence

$$K_1 \cap \cdots \cap K_{n-1} \supseteq \langle \alpha_0, \alpha_n \rangle d =: O,$$

and Lemma 3 applies to O. Let $C \leq 4$ be the cardinality of O.

Case $C = 1$ Then $K_0 \cap O \cap K_n$ has a common dart.

Case $C = 2$ Assume $K_0 \cap O \ni d$ and $O \cap K_n \ni e$. One of the two involutions α_0 or α_n sends d to e. If α_0 sends d to e both darts lie in K_n and hence $O \cap K_n \ni d$ which is the common dart of all cells in the cell tuple. If however α_n sends d to e then both darts are in K_0 in which case e is the common dart in $K_0 \cap O \cap K_n$. Anyway $K_0 \cap O \cap K_n$ has a common dart.

Case $C = 4$ Assume $O = \{a, b, c, d\}$, and $\alpha_0(a) = b$, $\alpha_0(c) = d$, $\alpha_n(a) = c$, and $\alpha_n(b) = d$. Assume further that $K_n \cap O \ni a$. Then:

(i) If $a \in K_0$, then a is our common dart and we are done.
(ii) If $b \in K_0$, then $\alpha_0(a) = b \in K_n$ and b is our common dart.
(iii) If $c \in K_0$, then $\alpha_n(c) = a \in K_0$ and we are in sub-case (i).
(iv) If $d \in K_0$, then $\alpha_n(d) = b \in K_0$, and we are in sub-case (ii) .

In each sub-case, it follows that $K_0 \cap O \cap K_n$ has a common dart, which proves the theorem. □

An important corollary of what has just been proven is, that the number of darts of a G-map is bounded from below by the number of cell-tuples. We will later use that property in the discussion of the storage-space complexity of G-maps.

The tuple map $\mathbf{c}$ is in general not injective (cf. Example 5). However, some important properties of G-maps can be read off $\mathbf{c}$. For example, according to [3, Lemma 2.1], it holds true for manifold-like G-maps that $\mathbf{c}$ is bijective.

Example 4 Figure 3 depicts a 1-G-map $X = (D, a_0, a_1)$ with two darts a and b. Both involutions a_0 and a_1 interchange the darts. Hence, $a_0 a_1$ is the identity, and we have

$$W = \langle \alpha_0 \rangle \times \langle \alpha_1 \rangle \cong \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}.$$

The assignment $W \to S_D, \alpha_i \mapsto a_i$ $(i = 0, 1)$ defines a minimal representation associated to X. The Coxeter graph of W consists of two vertices and no edges. The G-map itself is connected, and has a 0-cell $(D, 1, a_1)$ and a 1-cell $(D, a_0, 1)$. Notice that in this example, the orbits of $\langle \alpha_0 \rangle$ and $\langle \alpha_1 \rangle$ coincide. As the generators α_0 and α_1 of the Coxeter group W are distinct, the representations defining the 0-cell and the 1-cell are also distinct. This is a reason why cells are defined via representations of the whole Coxeter group W and not simply from W_0 or W_1. The associated G-map retains the information about which generator now acts trivially on the set of darts corresponding to a given cell.

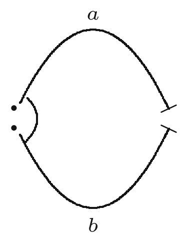


Fig. 3 A loop as a G-map

Example 5 The loop in Fig. 3 has the property that $\mathbf{c}$ is not injective. Namely, $a \neq b$, but $\mathbf{c}(a) = \mathbf{c}(b)$. In fact, there are precisely two cells, and both have as underlying sets $D = \{a, b\}$. Hence, $\mathbf{c}$ is not injective.

2.5 Properties of G-maps and their morphisms

A morphism of G-maps boils down to a map $\{\alpha_0, \ldots, \alpha_n\} \to \left\{\alpha'_0, \ldots, \alpha'_m\right\}$ which in turn uniquely corresponds to a map $\kappa\colon N := \{0, \ldots, n\} \to \{0, \ldots, m\}$ between the index sets. We call κ *contractive*, if for all $i \in N$ it holds true that $\kappa(i) \leq i$.

Definition 5 A morphism $F\colon X \to X'$ of G-maps is called *cellular*, if the induced map $\kappa\colon N \to M$ on index sets is contractive.

The following lemma for arbitrary morphisms of G-maps explains the above:

Lemma 4 *If $F\colon X \to X'$ is a morphism of G-maps, then $f\colon D \to D'$ takes any $y \in W_i x$, to $f(y) \in W_{\kappa(i)} f(x)$, where $\kappa\colon N \to M$ is the induced map between index sets.*

Proof Denote $\Phi\colon \{\alpha_0, \ldots, \alpha_n\} \to \left\{\alpha'_0, \ldots, \alpha'_m\right\}$ the map induced by F. Then $y \in W_i = \langle \alpha_0, \ldots, \check{\alpha}_i, \ldots, \alpha_n \rangle x$ is taken to

$$\begin{aligned} f(y) &\in \left\langle \Phi(\alpha_0), \ldots, \check{\Phi}(\alpha_i), \ldots, \Phi(\alpha_n) \right\rangle f(x) \\ &\subseteq \left\langle \alpha'_0, \ldots, \check{\alpha}'_{\kappa(i)}, \ldots, \alpha'_m \right\rangle f(x) \\ &= W_{\kappa(i)}\, f(x), \end{aligned}$$

which proves the assertion. □

Namely, if F is a morphism, then by Lemma 4, an i-cell is taken into some j-cell. If, moreover, F is cellular, then $j \leq i$ which is reminiscent of cellular maps $A \to B$ between cw-complexes: those take for all i the i-skeleton of A into that of B.

Definition 6 The G-maps together with the morphisms of G-maps form the category G**-map**. By taking only the cellular morphisms, one obtains the category G**-map**^{c}.

It is readily verified that G**-map**^{c} is indeed a category.

3 Topological datatypes from G-maps

Definition 7 A *topological datatype* is a pair (X, R), where X is a set and $R \subseteq X \times X$ is a binary relation on X. If (X, R) and (Y, S) are topological datatypes then a *continuous database map* $f\colon (X, R) \to (Y, S)$ is a map $f\colon X \to Y$ such that

$$(f \times f)(R) \subseteq S^*$$

where $f \times f\colon X \times X \to Y \times Y$ is the map

$$(x, y) \mapsto (f(x), f(y)),$$

and S^* is the reflexive and transitive closure of the relation S.

The category of topological datatypes with the continuous database maps as morphisms is called **DTop**. A topological datatype (X, R) determines a topological space $(X, \mathcal{T}_R)$ whose topology is given as

$$\mathcal{T}_R := \{A \subseteq X \mid \forall (a, b) \in R : a \in A \Rightarrow b \in A\}, \tag{10}$$

i.e. is generated by the binary relation R, cf. [2].

Example 6 The topological data type to Example 1 is (X, R), where

$$X = \{A, B, a, b, c, d, e, f, g, \alpha, \beta, \gamma, \delta, \epsilon, \zeta\}$$

and

$$R = \{(a, A), (c, A), (d, A), (f, A),$$
$$(b, B), (d, B), (e, B), (g, B),$$
$$(\alpha, a), (\beta, a), (\beta, b), (\gamma, b), \dots\}.$$

The topology $\mathcal{T}_R$ contains open sets like $\{A\}$, $\{B\}$, $\{d, A, B\}$, or $\{\alpha, a, c, A\}$. The relation R is also called "incidence relation". Every relation R' with $R \subseteq R' \subseteq R^*$ generates the same topology.

Let X be an object of G**-map**, and let $B_i := \mathcal{C}_i(X)$ the set of i-cells of X. The set $B := \mathcal{C}(X)$ of cells is defined to be the disjoint union of the sets B_i for all $i \in \mathbb{N}$.

We now define an incidence relation Δ on B as follows:

$$(x, y) \in \Delta \quad \Leftrightarrow \quad (x, y) \in B_{i-1} \times B_i \quad \text{and} \quad x \cap y \neq \emptyset,$$

where the intersection $x \cap y$ is understood as that of the corresponding sets of darts of X. In other words, cells x and y are in relation if and only if they are incident and their dimensions differ by one. Note that the cell tuples $(x_0, \dots, x_i, \dots, x_n)$ of the G-map are maximal sequences of cells such that two consecutive cells x_i and x_{i+1} are related by Δ.

Definition 8 The pair $(\mathcal{C}(X), \Delta)$ is called the *topological datatype* associated to the G-map X. It is also called the *cell space* of X.

The topological datatype carries the topological information of cell incidences in X. That is why it is also called *cell space*.

Let $F \colon X \to X'$ be a morphism of G-maps, and $B = \mathcal{C}(X)$, $B' = \mathcal{C}(X')$. From Lemma 4 it follows that there is a map $\mathcal{C}(F) \colon B \to B'$ induced by F. The question arises whether $\mathcal{C}(F)$ is a morphism $\mathcal{C}(F) \colon (B, \Delta) \to (B', \Delta')$ in **DTop**.

Recall that a map $\kappa \colon N \to M$ between sets of natural numbers is *monotonic* if $i < j$ implies $\kappa(i) \leq \kappa(j)$ for all $i, j \in N$.

Lemma 5 *The map $F \colon X \to X'$ induces a morphism $\mathcal{C}(F) \colon (B, \Delta) \to (B', \Delta')$ if and only if the map $\kappa \colon N \to M$ between index sets induced by f is monotonic.*

Proof We remind that the map $\mathcal{C}(F)$ is defined as

$$W_i\, d \mapsto W'_{\kappa(i)}\, f(d),$$

and that this assignment is well-defined by Lemma 4.

Assume in what follows that $x, y \in B$ be such that $\Delta(x, y)$ holds. Then $x = W_{i-1}\, d$, $y = W_i\, d$ for some $d \in D$. It follows that the cells

$$\mathcal{C}(F)(x) = W'_{\kappa(i-1)}\, f(d), \quad \mathcal{C}(F)(y) = W'_{\kappa(i)}\, f(d).$$

are incident. The question is whether they are related by the reflexive and transitive closure of Δ'.

If $\mathcal{C}(F)$ is a morphism in **DTop**, then consequently there is for all such pairs x, y a chain $\mathcal{C}(F)(x) = z_0, \dots, z_m = \mathcal{C}(F)(y) \in B'$ satisfying

$$(z_{j-1}, z_j) \in \Delta'$$

for all $j = 1, \dots, m$, where possibly $m = 0$. This condition implies $\kappa(i-1) < \kappa(i)$, or $\kappa(i-1) = \kappa(i)$ in the case $m = 0$. Hence κ is monotonic.

Assume now that κ is monotonic. If $\kappa(i-1) = \kappa(i)$, then

$$\mathcal{C}(F)(x) = \mathcal{C}(F)(y).$$

The other case is that $\kappa(i) = \kappa(i-1) + r$ with $r \geq 1$. Then

$$z_\rho := W'_{\kappa(i-1)+\rho}\, d, \quad \rho = 0, \dots, r$$

defines a chain in B' such that $(z_{\rho-1}, z_\rho) \in \Delta'$ for $\rho = 1, \dots, r$. Applied to all such pairs $x, y \in B$, this means that $\mathcal{C}(F)$ is a morphism in **DTop**. □

Definition 9 Let $F\colon X \to X'$ be a morphism of G-maps. It is said to be *continuous* if $\kappa\colon N \to M$ is monotonic. It is a *cellular map*, if it is a cellular morphism and continuous. The category of G-maps with continuous morphisms will be denoted by G-**map**$^{\text{cont}}$.

A consequence of Lemma 5 is that there is a functor

$$\mathcal{C}\colon\ G\textbf{-map}^{\text{cont}} \to \textbf{DTop}$$

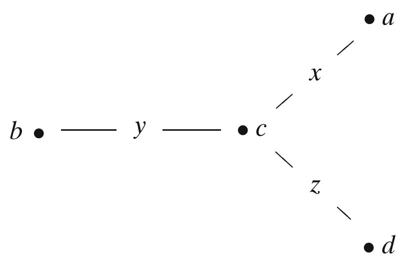
defined in the natural way as above. We remark that the cellular maps between G-maps play a similar role as the cellular maps between cell complexes.

Remark 3 The functor $\mathcal{C}$ above is not essentially surjective, as not every object of **DTop** can be the cell space of a G-map X. One reason for this is that in the space $(\mathcal{C}(X), \Delta)$ an i-cell of X can only be related to an $i+1$-cell, whereas in **DTop** this restriction does not hold.

One might suspect that with a different functor one would be able to model all topological spaces using G-maps. However, the following example will show that G-maps do not cover all topological spaces: They are restricted to manifolds.

Example 7 (Non-Involutionary Cell Tuples) Let the following topological space $(X, \mathcal{T}_X)$ be generated by the relation

$$R = \{(a, x), (c, x), (b, y), (c, y), (d, z), (c, z)\}.$$



Then the elements of that relation are also the cell tuples. Now, the cell tuples which correspond to the element c are $\{(x, c), (y, c), (z, c)\}$. Therefore it must be represented by at least three darts, because the map from the darts to the cell tuples is surjective.

Assume now there was a 1-G-map (D, a_0, a_1) that generates $(X, \mathcal{T}_X)$. Let d be the dart that represents vertex c. Then $c = C_0(d) = \langle a_1 \rangle d = \{d, a_1(d)\}$ because the orbit of an element with respect to one involution can only contain the element itself and its image. But this is a contradiction, because we have seen that $C_0(d)$ must have at least three elements. Therefore no 1-G-map can represent the above graph.

Note that the above example shows that a graph in general does not have an adequate representation as a 1-G-map.

4 Complexity of G-maps and topological datatypes

In this section we discuss the asymptotic storage complexity of G-maps and incidence based topological datatypes.

Storage complexity of a topological data structure DS will be expressed in terms of asymptotic growth of some "space function" $s_{DS} : \mathbb{N} \to \mathbb{N}$ where $s_{SD}(n)$ is the worst case storage space needed to store any topological space $(X, \mathcal{T}_X)$ with a finite point set of cardinality $n = |X|$ using DS. The problem of representing a general finite topological space by some data structure is known to be in $\mathcal{O}(n^2)$ and this complexity bound is optimal.

First we will introduce an important characterisation of finite topological spaces:

Definition 10 (Alexandrov Space) A topological space $(X, \mathcal{T}_X)$ with point set X and topology $\mathcal{T}_X$ is called *Alexandrov space* iff the intersection of every set $\mathcal{S}$ of open sets $\mathcal{S} \subseteq \mathcal{T}_X$ is again open, hence iff

$$\forall\, \mathcal{S} \subseteq \mathcal{T}_X \;:\; \bigcap_{S \in \mathcal{S}} S \in \mathcal{T}_X$$

holds.

This condition is stricter than the corresponding condition for general topological spaces where only *finite* sets of open sets need to have an open intersection. However, a finite topological space only has finitely many open sets and therefore is always an Alexandrov space.

Note that in general a subspace of $\mathbb{R}^n$ is not an Alexandrov space. We now state other well-known characteristics of Alexandrov spaces that we use later [1]:

Theorem 2 (Characteristics of Alexandrov Spaces) *Let $(X, \mathcal{T}_X)$ be a topological space. The following statements are equivalent:*

1. *$(X, \mathcal{T}_X)$ is an Alexandrov space*
2. *Each point $x \in X$ has a unique minimal neighbourhood* $\mathrm{Star}(x) \in \mathcal{T}_X$.
3. *The set $\mathcal{C}_X = \{X \setminus A \mid A \in \mathcal{T}_X\}$ of the closed sets in $(X, \mathcal{T}_X)$ is a topology for X. The space $(X, \mathcal{C}_X)$ is called* dual space *of $(X, \mathcal{T}_X)$.*
4. *There is a relation R that generates $\mathcal{T}_X$ according to Eq. 10.*

Note that the minimal neighbourhood $\mathrm{Star}(x)$ of x is the closure of $\{x\}$ in the dual space. Additionally, if a relation R generates an Alexandrov space its transpose R^T generates the dual space. Let us discuss that relation in more detail:

Definition 11 (Specialisation Preorder) Let $\mathcal{T}_X$ be a topology for X. Then $\mathcal{T}_X$ defines a relation $x \prec y :\Leftrightarrow x \in \mathrm{cl}\{y\}$ ("x is in the closure of y"), called the *specialisation preorder* in $(X, \mathcal{T}_X)$.

The specialisation preorder is known to be transitive and reflexive (hence called "preorder"), and it is intimately related to Alexandrov topologies because each Alexandrov topology is uniquely determined by its specialisation preorder. Other topologies may have the same specialisation preorder but then they are not of the Alexandrov type. If the relation R that generates the topology is not a preorder, the specialisation preorder of the generated topology is R^*, the reflexive and transitive closure of R. So each finite topology, by being of Alexandrov type, can be encoded by storing this preorder or any other relation the reflexive and transitive closure of which is that preorder. This immediately gives a storage-space upper bound of $\mathcal{O}(n^2)$.

This complexity upper bound is easily achieved by assuming w.l.o.g. the elements of the space are $\{1, \ldots, n\}$ and then store the incidence matrix by a word of length $n \cdot (n + 1)$ composed by the characters {`B`, `0`, `X`}. Each row in that matrix is a word of length $n + 1$ which starts with the character `B` as "begin row". Then the ith entry in row j is represented by the character `X` if i is incident with j and it is `0` otherwise. It is easily seen that the matrix is of size $n \cdot (n + 1)$.

Example 8 Let $\bullet\!\longleftrightarrow\!\circ$ be an edge. Then we say $1 = \bullet$, $2 = \longleftrightarrow$, and $3 = \circ$. The incidence relation is

$$R = \{(\bullet, \longleftrightarrow), (\longleftrightarrow, \circ)\} = \{(1, 2), (3, 2)\},$$

and the incidence matrix is

$$\begin{pmatrix} 0 & 0 & 0 \\ X & 0 & X \\ 0 & 0 & 0 \end{pmatrix}.$$

This matrix is then stored as the word `B000BX0XB000`.

4.1 Complexity of incidence-based data structures

The above mentioned complexity upper bound is also a lower bound:

Theorem 3 *Every data structure which can store all topologies for each arbitrary finite set X has a worst case storage-space complexity $\Omega(n^2)$, where n is the size of X.*

This is interesting because **DTop** is only slightly different to preorders in accepting arbitrary relations R on X (incidence relations). Remember that then the transitive and reflexive closure R^* is its specialisation preorder. Although accepting arbitrary relations can save much storage space it does not improve the asymptotic growth: The size of R then still grows quadratically with the size of X in the worst cases.

Proof We will show that a set X of size n has at least $2^{\frac{n^2}{4}}$ different topologies. But then each binary encoding of all topologies must have at least this number of different distinguishable states and therefore its worst case number of bits in a bit string that encodes the topology is $\log 2^{\frac{n^2}{4}} = \frac{n^2}{4}$. Together with $\frac{n^2}{4} \in \Omega(n^2)$ this proves the statement.

To prove the above lower bound on number of topologies we first assume with no loss of generality that $n = |X|$ is even. Then we partition X into two subsets E and V of equal size $\frac{n}{2}$. The set E will be the sets of edges of a hypergraph, and the set V becomes its vertices. A hypergraph is similar to a graph, only that an edge may connect an arbitrary number (including zero) of vertices.

Then we define the edge-vertex incidence by a relation $R \subseteq V \times E$. As $V \times E$ is of size $\frac{n}{2} \cdot \frac{n}{2} = \frac{n^2}{4}$ there are $2^{\frac{n^2}{4}}$ different such relations.

First note that R is transitive. If two such relations R and S are different, they also generate different topologies: Without loss of generality we assume $(v, e) \in R \setminus S$. By $v \neq e$, $(v, e) \notin S$, and S transitive, the minimal neighbourhood $\mathrm{Star}_S(v)$ of v in $(X, \mathfrak{T}_S)$, does not contain e. However, $\mathrm{Star}_S(v)$ then violates Eq. 10 with relation R and, hence, is not open in $(X, \mathfrak{T}_R)$. Therefore both topologies are different.

So we have at least $2^{\frac{n^2}{4}}$ different topologies for X. □

4.2 Dimension

We will give a formal definition of "dimension" for finite T_0 Alexandrov spaces. This notion of dimension also coincides with the topological Krull dimension "dim" [5] of T_0 Alexandrov spaces.

We briefly remind that a topological space $(X, \mathfrak{T})$ is said to satisfy the T_0-separability axiom, iff each two different points $a, b \in X, a \neq b$ have neighbourhoods $a \in U_a \in \mathfrak{T}$ and $b \in U_b \in \mathfrak{T}$, such that $a \notin U_b$ or $b \notin U_a$. It can be easily shown that an Alexandrov space is T_0 iff it is generated by an acyclic relation R. Then its specialisation preorder R^* is a partial order. T_0 is the weakest of a family of separability axioms and the only one a non-discrete Alexandrov space can satisfy. The next stronger T_1-separability axiom says that for a and b there exist two neighbourhoods such that $a \notin U_b$ **and** $b \notin U_a$ hold from which follows by the Alexandrov property that $\{a\}$ and $\{b\}$ are open: Take all open sets U_b which contain b. Now, by T_1, for each other point a there exists such an open set U_b not containing

a. By the Alexandrov property the intersection of all these open sets is open. But that intersection contains b and no other point. Therefore a T_1 Alexandrov space is discrete (in the topological sense) or, in other words, not very interesting.

Now we can define "dimension" for these spaces:

Definition 12 (Vertices) Let $(X, \mathcal{T})$ be a finite T_0-space, then we call the set $X_0 := \{x \in X \mid \mathrm{cl}\{x\} = \{x\}\}$ the *vertices* of that space.

The subspace of the vertices $(X_0, \mathcal{T}|_{X_0})$ is the maximal discrete closed subset of X. It consists of all minimal elements with respect to the specialisation preorder.

If we remove all vertices from X we get a remaining subspace $X \setminus X_0$ which is also T_0 and therefore has its own set of vertices. If we denote the vertices of X by X_0, the vertices of $X \setminus X_0$ are correspondingly denoted by $(X \setminus X_0)_0$. These are then the edges of X. This removal can be done repeatedly until our space is completely exhausted:

Definition 13 (Edges, Faces, Volumes, etc.) Let $(X, \mathcal{T})$ be a finite space, then we call the set $X_1 := (X \setminus X_0)_0$ the *edges* of $(X, \mathcal{T})$. For a number $i \in \mathbb{N}$, we define $X_{i+1} := (X \setminus \cup_{j=0}^{i} X_j)_0$. The set X_2 is usually called the "faces" of $(X, \mathcal{T})$, and X_3 are its "volumes".

After having removed the vertices from a space X the dimension of each element shifts down by one—its edges, for example, become the new vertices.

We say that i is the *dimension* of an element $x \in (X, \mathcal{T})$ if x is in X_i, and we call the maximal dimension of an element within that space the *dimension* of the space. Note that this "dimension of elements", in general, contradicts intuition: an "edge", for example, may have three endpoints.

We will now see that based on this we get an effective data structure for some T_0 spaces:

Theorem 4 (Generalised B-Rep) *A non-empty finite T_0-space $(X, \mathcal{T}_X)$ of dimension d is uniquely determined by a sequence*

$$X_0 \; R_1 \; X_1 \; R_2 \; \cdots \; R_d \; X_d$$

of non-empty and mutually disjoint sets X_i and right-total relations $R_i \subseteq X_{i-1} \times X_i$ such that $X = \bigcup_{i=0}^{n} X_i$ and $R = \bigcup_{i=1}^{n} R_i$.

Proof The relation R is indeed a relation on X, and hence $(X, \mathcal{T}_R)$ a topology defined by the given sequence.

We first have to prove that X_0 is the set of vertices and, second, that the remaining sequence $X_1 \; R_2 \; \cdots \; R_d \; X_d$ defines the space $(X, \mathcal{T}_R) \setminus X_0$. Then by induction it is clear that this remaining sequence defines a space of dimension $d - 1$. The induction starts with a "sequence" X_0 of length 0 which with its empty relation $\emptyset$ generates the discrete space $(X_0, \mathcal{T}_\emptyset)$.

So let $v \in X_0$ be arbitrarily chosen. Take an arbitrary pair $(a, b) \in R$. Then $b \neq v$ because none of the relations has X_0 on its right hand side and all other sets are disjoint from X_0. Therefore $b \in X \setminus \{v\}$ and so the implication $a \in X \setminus \{v\} \Rightarrow b \in$

$X \setminus \{v\}$ in Eq. 10 holds. Therefore $X \setminus \{v\}$ is open. But then the complement $\{v\}$ is closed and so v is a vertex.

Now take a $w \in X_i$ with $i \neq 0$. We show that w is not a vertex by showing that $\{w\}$ is not closed. First by $i \neq 0$ there is a set X_{i-1} related to X_i by a right-total relation R_i. But as X_{i-1} is not empty and R_i is right-total, there is also a pair (v, w) in R_i with $v \in X_{i-1}$. As X_i is disjoint to X_{i-1} we have $v \neq w$ and therefore $v \in X \setminus \{w\}$. But then the implication $v \in X \setminus \{w\} \Rightarrow w \in X \setminus \{w\}$ does not hold and so the set does not satisfy the condition for open sets according to Eq. 10 and, hence, is not open. But then its complement $\{w\}$ is not closed and therefore w is not a vertex.

Therefore all elements in X_0 are vertices of $(X, \mathfrak{T}_R)$ and all other elements of X are not.

We will later need the property that the set X_0 is closed: As all singleton sets $\{v\}$ for $v \in X_0$ are closed and the union of an arbitrary set of closed sets in an Alexandrov space is closed, the set X_0, which is the union of these singletons, is closed, too.

To show that $(X, \mathfrak{T}_R) \setminus X_0$ is generated by the subsequence $X_1\ R_2 \cdots R_d\ X_d$ we have to show that a set is open in the subspace iff it is open in the space generated by that subsequence.

Let S be the relation $R \setminus R_1$ and $(Y, \mathfrak{T}_S)$ be the space generated by the remaining sequence.

First, the point sets obviously coincide so $X \setminus X_0 = Y$. What remains is to show that the topologies are equal.

$\mathfrak{T}_S \subseteq \mathfrak{T}_R|_Y$:

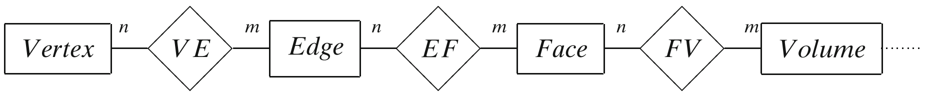
If a set U is open in $(Y, \mathfrak{T}_S)$ then for every $(a, b) \in S = \bigcup_{i=2}^{d} R_i$ the implication $a \in U \Rightarrow b \in U$ holds. But then this is also true in the original space, because for every $(a, b) \in R_1$ the statement $a \in U$ is wrong as then a would be in X_0 which cannot be the case. Therefore the implication $a \in U \Rightarrow b \in U$ is also true for $(a, b) \in R_1$. Therefore U is open in the entire space $(X, \mathfrak{T}_R)$. By $U \cap Y = U$ the set U is also open in $\mathfrak{T}_R|_Y$.

$\mathfrak{T}_S \supseteq \mathfrak{T}_R|_Y$:

From the definition of subspace a set U is open in the subspace Y iff there exists an open set U_X in the original space such that $U_X \cap Y = U$. But as X_0 is closed in $(X, \mathfrak{T}_R)$ the set Y as its complement is open and hence $U = U_X \cap Y$ is open in $(X, \mathfrak{T}_R)$. But then for every pair $(a, b) \in R$ the implication $a \in U \Rightarrow b \in U$ holds. But then that implication holds for every pair (a, b) in a subset of R such as the relation S. Therefore U is in $\mathfrak{T}_S$.

So we have shown that a generalised b-rep sequence of length d can generate a topological space of dimension d. □

Note that the chain of relations presented in Theorem 4 is a generalisation of the classical b-rep-structure to arbitrary dimension, as denoted by the following ER-schema:



This schema, however, establishes some restrictions on spaces that can be stored in this manner. For example it establishes a static upper bound of the dimension. The authors, however, prefer a relational schema where all "primitives" are collected into *one* entity type with an integer attribute specifying the dimension instead of using a distinct type for each dimension.

4.3 Simplices

We now present an example class of topological spaces which have a G-map representation and which we will use to compute a complexity lower bound for G-map representations of topological spaces. With Eq. 9 in Section 2.4 we have already introduced potential dart candidates for G-maps:

Definition 14 (Cell Tuple) Let $(X, \mathcal{T})$ be a finite T_0-space. A *cell tuple* of $(X, \mathcal{T})$ is a tuple $(x_0, \ldots, x_n) \in X^{n+1}$ of maximal length such that $i \neq j \implies x_i \neq x_j$ (the tuple is injective) and $i < j \implies x_i \in \mathrm{cl}\{x_j\}$.

As opposed to the definition in Section 2.4, which is based on G-maps, we have given here a purely topological definition of cell tuples which is independent of any representation of the topology. However, one easily sees that there is a correspondence of both definitions: Two cells of a G-map are incident in the associated topological space iff their dart sets are not disjoint and that $a \in \mathrm{cl}\{b\}$ means, a is incident to b. The other properties are just taken from the G-map-based definition.

An element x_i is of dimension i iff it is at the ith position in each cell tuple $(x_0, \ldots, x_i, \ldots, x_n)$ which contains that element. By Theorem 1 the map from darts to the cell tuples is surjective in G-maps. This means that in a G-map the number of darts cannot be less than the number of cell tuples.

By *combinatorial simplex* we mean the topological space where the elements are the non-empty subsets of some set $\{v_0, \ldots, v_n\}$ of "vertices" of size $n + 1$. The dimension of such an element is equal to the set size minus 1, and the dimension of a simplex grows linearly with the number of its vertices.

The topology of that space is defined by the $\subset$ relation: an element represented by $\{v_{i_0}, \ldots, v_{i_n}\}$ is incident to an element $\left\{v_{j_0}, \ldots, v_{j_m}\right\}$ if it is a subset: $\left\{v_{i_0}, \ldots, v_{i_n}\right\} \subseteq \left\{v_{j_0}, \ldots, v_{j_m}\right\}$. For example the singletons $\{a\}$ and $\{b\}$ are each of dimension 0 and they are both incident with the "edge" $\{a, b\}$ of dimension 1.

Example 9 (Triangle) A 2-simplex is the space $(2^V \setminus \emptyset, \mathcal{T}_\subset)$ of the non-empty subsets of $V = \{v_0, v_1, v_2\}$. The set $\{v_1\}$, for example, is the vertex attached to the edges $\{v_0, v_1\}$ and $\{v_1, v_2\}$. The face is represented by V itself. An example cell tuple is the sequence $(\{v_1\}, \{v_0, v_1\}, \{v_0, v_1, v_2\})$. Note that two consecutive cells in the tuple differ by exactly one additional element.

So in every subsequence A_{i-1}, A_i, A_{i+1} within a cell tuple of a simplex the set A_{i+1} has exactly two more elements than A_{i-1}. Let $\{u, w\}$ be these two additional elements. Then A_i can either be $\{u\} \cup A_{i-1}$ or $\{w\} \cup A_{i-1}$. Hence the two sequences $\ldots, A_{i-1}, \{u\} \cup A_{i-1}, A_{i+1}, \ldots$ and $\ldots, A_{i-1}, \{w\} \cup A_{i-1}, A_{i+1}, \ldots$ are connected by an involution a_i such that

$$a_i(\ldots, A_{i-1}, \{w\} \cup A_{i-1}, A_{i+1}, \ldots) = \ldots, A_{i-1}, \{u\} \cup A_{i-1}, A_{i+1}, \ldots .$$

Now if we compose two such involutions a_i and a_j with $|i - j| \geq 2$ and apply this composition $a_i \circ a_j$ to a sequence

$$\ldots, A_{i-1}, A_i, \ldots, A_k, \ldots, A_j, A_{j+1}, \ldots$$

then, by the set A_k between A_i and A_{j-1}, there is also at most one possibility to replace both A_i and A_j by another A'_i and A'_j which is what $a_i \circ a_j$ does. Applying $a_i \circ a_j$ again will return the original sequence. Therefore the set of all cell tuples of a combinatorial simplex together with these involutionary replacements constitutes a G-map to which the simplex itself is the associated space. This shows that a G-map representation of a simplex exists.

This is generally true for every cell complex which constitutes a manifold or a manifold with boundary [3]. Historically, it is this observation that lead to the concept of G-maps.

We will now use combinatorial simplices to show that there are essentially more cell tuples (hence also more darts of a generating G-map) than incidences (edges in the incidence graph). So this example will show that G-maps are not optimal with respect to storage space.

Example 10 Let S_n be the n-dimensional combinatorial simplex with vertices $V = \{v_0, \ldots, v_n\}$. This space consists of the set of non-empty subsets of V. There are $2^{n+1} - 1$ such subsets and hence our encoding of the topology of these subsets can be done by an incidence matrix with no more than $(2^{n+1} - 1)^2 \leq 2^{2n+2} = 4^{n+1}$ entries.

Now all cell tuples of S_n start with a singleton set $(\{x_i\}, \ldots)$, hence there are $n + 1$ different possibilities to start such a cell tuple. Each $i + 1$-cell following an i-cell X_i is obtained by adding a new vertex to X_i which is not yet member of X_i. Each cell X_i at the ith position of a cell tuple, hence, is a set of $i + 1$ vertices and so there are $n - i$ possible continuations of that partial cell tuple. So the total number of cell tuples of an n-simplex is $(n + 1)!$ which is more than $4 \cdot 4^n$.

Now if we store the above example by a G-map and assumed that each dart consumes constant storage space we get the following storage complexity lower bound:

Theorem 5 *The worst case number of darts $d(n)$ in a cell-tuple representation of an arbitrary topological T_0-space with n points and no dimension upper bound is asymptotically bigger than $\mathcal{O}(p)$ for any polynomial p.*

Again, this means that, in general, an *explicit* storage of cell tuple representatives is far from the optimal $\mathcal{O}(n^2)$.

Proof By Theorem 1 we know that there cannot be more cell tuples than darts. So we count the cell tuples of our example space and their number gives a lower bound on the number of darts.

We consider Example 10, an i-simplex. It consists of $n = 2^{(i+1)} - 1$ cells and has $(i + 1)!$ cell tuples. Hence $n + 1 = 2^{i+1}$, or $i = \log_2(n + 1) - 1$. So the i-Simplex has $d(n) = (\log_2(n + 1) - 1)!$ cell tuples if it has n cells.

We want to show that

$$\limsup_{n\to\infty} \frac{d(n)}{n^k} = \infty$$

where $n > 0$ and k is an arbitrary constant polynomial degree.

Assume that

$$q(n) = \frac{(\log_2(n+1)-1)!}{n^k}$$

for an arbitrary polynomial degree k had an upper bound. If this were true then the expression we get by replacing n by 2^{n+1} would also have an upper bound. First we define the constant $c := 2^{2k}$ and then do the aforementioned replacement to get

$$\begin{aligned} r(n) = q(2^{n+1}) &= \frac{(\log_2(2^{n+1}+1)-1)!}{(2^{n+1})^k} \\ &\geq \frac{(\log_2(2^n+1))!}{(2^k)^{n+1}} \\ &\geq \frac{(\log_2(2^n))!}{(2^k)^{n+1}} \\ &\geq \frac{n!}{(2^{2k})^n} \\ &= \frac{n!}{c^n}. \end{aligned}$$

It is well known that $\frac{n!}{c^n}$ for a constant c has no upper bound and so q can have no upper bound, either. □

So, when cell tuples or darts of a *G*-map are stored explicitly and each such element consumed a constant amount of space they already need much more memory than storing the incidence relation would do. By now we have not yet considered the additional information about which partitioning represents which element in the topological space.

But enumerating the cell tuples is not enough—their connectivity must also be stored.

4.4 Complexity of involutions

Note that a *G*-map always has an associated space and with this space it is possible to generate a corresponding *G*-map. So every topological data model may then be considered a, possibly compressed, representation of a *G*-map. For this reason this discussion is not about the space complexity of the *G*-map storage problem but of the costs of *explicitly* storing darts and involutions of some given *G*-map in the straightforward manner its definition immediately suggests. This involves storing the involutions on the n darts, too, which also give a significant contribution to the storage complexity.

First we briefly introduce the log-function we will use:

Definition 15 (Integer Logarithm) The function

$$\log : \mathbb{N} \to \mathbb{N}, n \mapsto \log n := \begin{cases} \lfloor \log_2 n \rfloor + 1 & : n > 0 \\ 1 & : n = 0 \end{cases}$$

is called the *integer logarithm*.

This function denotes the minimum number of bits needed to store an integer n in binary. It is at least 1 for storage of the number 0 in a single bit.

It is easy to see that every function $f : \{1, \ldots, n\} \to \{1, \ldots, n\}$ can be stored by some data structure of at most $\mathcal{O}(n \log n)$ amount of memory which hence is also a storage complexity upper bound for an involution acting on a set D of n elements. We assume here $D = \{1, \ldots, n\}$ and store such a mapping as a bit word which roughly says:

“`At most` $2^{\log n}$ `entries will follow:`$(f(1), f(2), \ldots, f(n))$”.

The prefix, announcing the number of entries is a bit word ‘`0..01`’ of length $\log(n + 1)$ consisting of $\log n$ leading ‘`0`’s and one terminating ‘`1`’ and is of same length as each entry $f(i)$ to follow. Each entry at position i is the binary representation of $f(i)$—a bit word of length $\log(n + 1)$. There are n such entries and so the total length is $\log(n + 1) + n \log(n + 1)$ which is in $\mathcal{O}(n \log n)$.

For example, the function

$$f : \{1, \ldots, 4\} \to \{1, \ldots, 4\},$$

$$f(1) := 4,\ f(2) := 2,\ f(3) := 4,\ f(4) := 1$$

is stored as

`001 100 010 100 001.`

The first ‘`001`’ denotes that each entry consists of three digits.

This complexity upper bound is also asymptotically optimal for involutions. The proof is simple: Count all involutions on a given set of size n, provide a lower bound on that number, and then each data structure must take at least that number of distinguishable states which is, if stored in binary, $2^{\text{minimal length of that structure}}$.

It is well known that the number $I(n)$ of involutions on a set D of size n satisfies the following recursive formula [12]:

$$I(0) = 1$$

$$I(1) = 1$$

$$I(n + 2) = I(n + 1) + (n + 1) \cdot I(n).$$

For big numbers (where “big” means $n \geq 2$) we can rewrite the above formula:

$$I(n) = I(n - 1) + (n - 1) \cdot I(n - 2).$$

However, $I(n)^2$ is bounded from below by $(n-1)!$, as one easily induces by

$$\begin{aligned}
I(1)^2 &= 1 \geq 0! \\
I(2)^2 &= 4 \geq 1! \\
I(n+1)^2 &= (I(n) + n \cdot I(n-1))^2 \\
&\geq I(n)^2 + n^2 \cdot I(n-1)^2 \\
&\geq (n-1)! + n^2 \cdot (n-2)! && \text{by induction} \\
&= (n-1) \cdot (n-2)! + n^2 \cdot (n-2)! \\
&= (n^2 + n - 1) \cdot (n-2)! \\
&\geq (n^2 - n) \cdot (n-2)! && \text{as } n \text{ is "big"} \\
&= n \cdot (n-1) \cdot (n-2)! = n!.
\end{aligned}$$

If some binary encoding of a data structure is able to distinguish between at least $\lfloor\sqrt{(n-1)!}\rfloor$ different instances it must have at least $\lfloor\sqrt{(n-1)!}\rfloor$ different states and therefore a worst case bit-length of

$$L(n) \geq \left\lfloor \frac{1}{2}\log((n-1)!) \right\rfloor.$$

But then

$$\begin{aligned}
2 \cdot L(n) \geq \log((n-1)!) \geq \log\left(\frac{n-1}{2}\right)^{\frac{n-1}{2}} &= \frac{n-1}{2}\log\frac{n-1}{2} \\
\in \Omega\left(\frac{n-1}{2}\log\frac{n-1}{2}\right) &= \Omega((n-1)\log(n-1)) \\
= \Omega(n\log n)
\end{aligned}$$

holds. Therefore, by $L(n) \in \Omega(n\log n)$, we get that $\mathcal{O}(n\log n)$ is the optimal storage complexity of involutions acting on n elements.

If a topological space has n elements, by the proof of Theorem 5, the number of darts can become up to $(\log(n+1)-1)!$ and so explicit storage of a G-Map would cost

$$\Omega(d(n)\log d(n))$$

with

$$d(n) = (\log(n+1)-1)!$$

because it would store at least one involution on all darts. The additional storage space $d(n)$ for the darts themselves can be neglected by

$$f \in \mathcal{O}(f\log f)$$

together with the fact

$$f \in \mathcal{O}(g) \implies \Omega(f+g) = \Omega(g).$$

Remember that the number of darts of a G-map representing an arbitrary topological space of n elements alone exceeds every polynomial—in particular the optimal $\mathcal{O}(n^2)$. In addition, storage of the involutions even makes the situation worse. So explicitly storing darts and involutions of a G-map can become very expensive if no dimension limit is imposed.

We admit that in spatial data modelling the dimension number of a space has a relatively small fixed upper bound in almost all practical cases. This dimension, in general, is much less than the number of stored elements. On the other hand, such excessive growth of storage space with dimension does not only mean that storage space is unnecessarily wasted. It also means that higher dimensional data may have many redundancies when stored as G-maps.

5 Conclusion

A definition of morphisms for G-maps was given and used to show that G-maps are, in fact, an effective data structure for some topological data. However, their being particularly designed for manifold data makes storage of non-manifold topological data using G-maps at least "unnatural" when not impossible.

The main shortcoming of G-maps, however, is their extreme verbosity. They have been advocated as storage schema for topological data at arbitrary dimension. But if, indeed, dimension is unbounded G-map storage complexity as a function of the number of "spatial primitives" exceeds every polynomial and is clearly not optimal.

The discussion presented in this article started with the authors' observation that G-maps have a tendency to becoming extremely complex even in the case of usual 3D volume modelling, an observation which is also reported by other authors like, for example, [6, Section 1.2, p. 151]. Anyway, dynamically created cell tuples could be very useful for implementing topological algorithms—it is only their use as explicit storage schema which is discussed here. Such use of G-maps in algorithms, however, should be done with care because if such algorithm iterates all cell tuples the observed space complexity immediately turns into time complexity.

Acknowledgements The authors acknowledge support from the DFG-projects BR 3513/3, and BR 2128/12 and many valuable comments from the anonymous reviewers.

References

1. Alexandroff P (1937) Diskrete Räume. Mat Sb 44(2):501–519
2. Bradley PE, Paul N (2010) Using the relational model to capture topological information of spaces. Comput J 53(1):69–89
3. Brisson E (1993) Representing geometric structures in d dimensions: topology and order. Discret Comput Geom 9:387–426
4. Grasset-Simon C, Damiand G, Lienhardt P (2006) nD generalized map pyramids: definition, representations and basic operations. Pattern Recogn 39(4):527–538
5. Hartshorne R (1977) Algebraic Geometry. Springer, New York
6. Kraemer P, Cazier D, Bechmann D (2009) Extension of half-edges for the representation of multiresolution subdivision surfaces. Visual Comput 25:149–163
7. Lienhardt P (1994) N-dimensional generalized combinatorial maps and cellular quasi-manifolds. Int J Comput Geom Appl 4:275–232
8. Mäntylä M (1988) An introduction to solid modeling. Computer Science, Rockville, MD

9. Paul N (2008) Topologische Datenbanken für Architektonische Räume. Dissertation, Universität Karlsruhe
10. Paul N (2010) Basic topological notions and their relation to BIM. In: Underwood J, Isikdag U (ed) Handbook of research on building information modeling and construction informatics—information science reference, pp 451–472
11. Schneps L (ed) (1994) The Grothendieck theory of Dessins d'Enfants. Cambridge University Press, Cambridge, MA
12. Sloane NJA (ed) (2012) Number of self-inverse permutations on n letters, also known as involutions; number of Young tableaux with n cells. In: On-line encyclopedia of integer sequences. http://www.research.att.com/~njas/sequences/A000085

Patrick Erik Bradley received a Ph.D. in mathematics from Karlsruhe University, Germany, in 2002 with an emphasis in non-archimedean geometry. Afterwards, he joined the Architectural Faculty of Karlsruhe University where he became a researcher on building stock dynamics and building information modelling. He is currently a researcher at the Institute of Photogrammetry and Remote Sensing at Karlsruhe Institute of Technology, Germany.

Norbert Paul graduated from the University of Karlsruhe, Germany 1996 with a Diploma of Architecture and Urban Planning. After several years as programmer for geoinformation systems and later for embedded controllers he joined the Architectural Department of the Karlsruhe University in 2002 as a researcher on building information modeling. There he received the Ph.D. for his dissertation on Topological Databases. He now works for the Geodetic Institute (GIK) at the KIT in Karlsruhe, Germany as a researcher in spatial modeling.