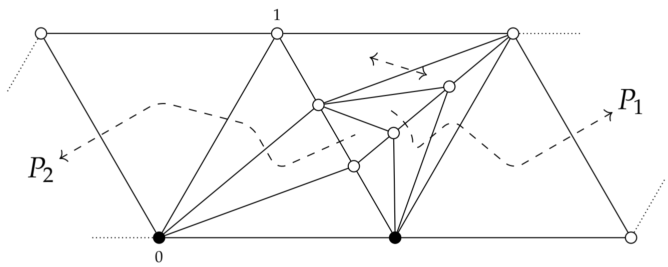
ALTA
**Institute for Algebra, Geometry,
Topology and their Applications**

Universität Bremen

Weak Symmetry Breaking and Simplex Path Demonochromatizing



Master Thesis

by Jan-Philipp Litza

REFEREES:
Prof. Dr. Dmitry Feichtner-Kozlov
Dr. Damien Imbs

SUBMISSION:
6th March 2015

Contents

Acknowledgements

First of all, I want to thank *Dmitry Feichtner-Kozlov* for showing up this very interesting connection between combinatorial topology and distributed computing, two fields that I much enjoy working in! If it was not for him, I would probably still be looking for the best topic to write this thesis about. Of course I am just as grateful for his following supervision, that encouraged and animated me each time necessary.

Many thanks to *Damien Imbs* for his high interest in the topic which not only resulted in his refereeing but also in many helpful tips, discussions and an always open door. Not many people take the job of the second supervisor as serious as he did!

For many helpful discussions, a huge incentive to start working seriously on the thesis as well as building me up when I was down, I want to thank *Alina Stürck*, who shared my struggle of writing a thesis.

What would a thesis be if it was not proofread? I want to thank *Tim Haga*, *Nils Przigoda*, and *Jan Senge* for pointing out typos, content-related errors, style mistakes and typesetting deficiencies.

Last but surely not least, my gratefulness and love goes to *Kathrin Stoltenberg*, who not only read my drafts numerous times and put up with my abstract field of study, never doubted the successful completion of my thesis and gave me strength all the way through, but also accompanied me from the very beginning of my studies to this very day and onwards. Thank you!

Notations and Conventions

$$\delta_{i,j} := \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

$$[n] := \{0, 1, \ldots, n\} \subset \mathbb{N}$$

$$[a, b] := \{x \in \mathbb{R} \mid a \leqslant x \leqslant b\}$$

$$n\text{-tuple } X := (x_1, x_2, \ldots, x_n)$$

$$[n]\text{-tuple } X := (x_0, x_1, x_2, \ldots, x_n)$$

$$\mathrm{pr}_A : A \times B \to A,\ (a, b) \mapsto a$$

$$\varphi(A') := \{\varphi(a) \mid a \in A'\} \qquad \text{for } \varphi\colon A \to B,\ A' \subseteq A$$

$$A - a := A \setminus \{a\} \qquad \text{for } a \in A$$

$$\#(s, x) := |\{i \mid x_i = s\}| \qquad \text{for a tuple } x$$

$$\#(s, B(X)) := |\{x \in X \mid B(x) = s\}| \qquad \text{for a set } X$$

$$A \triangle B := (A \cup B) \setminus (A \cap B) \qquad \text{for sets } A, B$$

$$\overline{b} := 1 - b \qquad \text{for a boolean value } b \in [1]$$

1 Motivation

In the field of distributed computing, one cares about a seemingly simple question:

> What can be computed by two or more processes by communicating with each other?

Of course, such a question can only be answered with more concrete specifications on what "computing" and "communicating" means. There are several models that define their exact meanings and result in different answers to the above question. Especially when we allow processes to crash or misbehave, this is a more complex question than one would expect at first glance, as can be seen by the amount of research that is being done since [PSL80] was one of the first papers to consider this kind of problem.

Soon, a new approach to the topic developed: Using combinatorial topology. In the form of graph connectivity it has first proven to be a helpful tool to model the communication of processes in [FLP85]. Since then, much more work has followed in this field, using well-known topological facts to prove the possibilities and mainly impossibilities of distributed computing. [SZ00] is one such example, linking the Brouwer fixed-point theorem to the impossibility of $n+1$ processes agreeing on any less than $n+1$ distinct values. We will study this application to distributed computing as well as the model of communication and computing in Section 2.2, after we will have introduced the needed basics of combinatorial topology in Section 2.1.

Of all the tasks that such processes could attempt to solve, we care about a particular one in Chapter 3, called the Weak Symmetry Breaking: Can $n+1$ processes, by communicating according to a given model, each decide on one of two options such that in the end each option is chosen at least once? We restrict the question to cases where all processes decide on a value, which is required by the possibility that processes can fail, in which case they do not decide on a value. Because this is unsolvable in some cases if every process starts with exactly the same parameters—think of all of them executing at the same time, which makes the state of every process indistinguishable from all the others—we loosen the task a little bit and allow them to have unique integral identifiers which they can compare.

By itself this task probably would not have gotten much attention. But a more realistic problem, K-renaming, can be solved for $K = 2n - 1$ if and only if one knows how to solve the Weak Symmetry Breaking [GRH06]. In K-renaming, $n+1$ processes start with process identifiers from a large namespace of size $N \gg n+1$ and want to "rename" themselves uniquely to a much smaller namespace of size $K + 1 \ll N$.

The Renaming Problem has been studied numerous times, and some confusion about its true answer arose. While early results showed that this task is (wait-free) solvable if $K \geqslant 2n$ and unsolvable if $K \leqslant n+1$ [ABND+90], the domain $n+1 < K < 2n$ was a bit unclear. After several supposed proofs that the task was unsolvable in all these intermediate cases, [CR12] were the first to show that it actually is solvable for $K = 2n - 1$ if $n + 1$ is not a prime power.

The reason *we* care about Weak Symmetry Breaking is its reformulation in the language of combinatorial topology. Along the way, very simple combinatorial structures called "simplex paths" arise that need to be "demonochromatized". This problem can be quite easily stated given the necessary terminology, as we will do in the beginning of Chapter 4. Its existing solutions that are surveyed later in that chapter, however, are more or less complex and hard to fully comprehend and verify. Furthermore, their earlier formulations used languages that differed from one another. We unified the terminology and fixed some smaller errors in our version.

Finally, we will begin to develop a possible alternative. Furthermore, we present a round complexity estimation not only for the first algorithm, for which it was done in [ACHP13] already, but also a novel one for the second algorithm that was developed in [Koz15].

2 Prerequisites

2.1 Combinatorial Topology

First we need to establish a basic framework of definitions. Those familiar with simplicial complexes can easily skip this section, which we will keep as short as possible, assuming that most readers are familiar with most of the concepts. Those who wish for a more detailed introduction may be interested in [Koz08, Chap. 1] where almost all of these concepts can be found with more detailed explanations, and in [HKR13, Chap. 3] for the parts more specific to our application.

While topology in general tries to distinguish important properties of a space, like whether two lines cross each other or not, from unimportant ones, like the angle at which they cross, combinatorial topology deals with simple tools that are all based on counting properties associated to spaces.

2.1.1 Simplicial complexes

The tool that is most important to us is the *simplicial complex*. In its abstract form, given a finite *ground set* $V(\mathcal{A})$, a simplicial complex on $V(\mathcal{A})$ is simply a subset of its power set $\mathcal{A} \subset 2^{V(\mathcal{A})}$ such that ABSTRACT SIMPLICIAL COMPLEX

(1) $\mathcal{A}$ is closed under taking subsets, i.e. $X \in \mathcal{A}$ if $X \subset Y \in \mathcal{A}$, and
(2) $\mathcal{A}$ contains all one-element subsets of $V(\mathcal{A})$, i.e. $\{v\} \in \mathcal{A}$ for all $v \in V(\mathcal{A})$.

The condition (2) keeps the ground set as small as possible and can also be formulated as $V(\mathcal{A}) = \bigcup \mathcal{A}$. Another simplicial complex $\mathcal{B}$ is called a *subcomplex* of $\mathcal{A}$ if $\mathcal{B} \subseteq \mathcal{A}$. In particular, $\mathcal{B}$ has to be based on a ground set contained in that of $\mathcal{A}$, i.e. $V(\mathcal{B}) \subseteq V(\mathcal{A})$. SUBCOMPLEX

Each element of $V(\mathcal{A})$ is called a *vertex*, each element $\sigma \in \mathcal{A}$ with $|\sigma| = n + 1$ a *simplex of dimension n*, an *n-simplex* or simply a *simplex*. We set $\dim(\sigma) := n = |\sigma| - 1$ and often denote an n-simplex by σ^n with a superscript n to indicate its dimension, especially VERTEX n-SIMPLEX when simplices of different dimensions are involved. When $\sigma \subseteq \tau \in \mathcal{A}$, then σ is called a *face of* τ, if $\emptyset \subsetneq \sigma \subsetneq \tau$ even a *proper face*. The other way round, τ is called a *co-face* of σ in this constellation. If a simplex has no co-faces, it is called a *facet* of $\mathcal{A}$. The dimension $\dim(\mathcal{A}) := \max_{\sigma\in\mathcal{A}} \dim(\sigma)$ of the whole simplicial complex $\mathcal{A}$ is the largest dimension of its simplices. If all facets have the same dimension n, the simplicial complex is called *pure* and the *codimension* of a simplex σ is $\operatorname{codim}(\sigma) := \dim(\mathcal{A}) - \dim(\sigma)$. The union of all simplices $\sigma \in \mathcal{A}$ with $\operatorname{codim}(\sigma) = 1$ that have only one co-face is called the *boundary* of $\mathcal{A}$, and we denote it by $\partial\,\mathcal{A}$. Note that in a pure simplicial complex, one co-face is the minimum for all simplices with a codimension of at least one.

(CO-)FACE
FACET
PURE
BOUNDARY

By abuse of notation, we will sometimes call a simplicial complex that only consists of a single n-simplex and all its faces an n-simplex as well and denote it simply by σ or σ^n. For example, $\partial\,\sigma$ is the union of all proper faces of σ.

Given two abstract simplicial complexes $\mathcal{A}$ and $\mathcal{B}$ on ground sets $V(\mathcal{A})$ and $V(\mathcal{B})$, respectively, every map $\varphi\colon V(\mathcal{A}) \to V(\mathcal{B})$ is called a *vertex map*. Only those vertex maps who preserve simplices, i.e. $\{\varphi(v_1), \cdots, \varphi(v_m)\}$ is a simplex of $\mathcal{B}$ if $\{v_1, v_2, \ldots, v_m\}$ is a simplex of $\mathcal{A}$, are called *simplicial maps*. We will also write $\varphi\colon \mathcal{A} \to \mathcal{B}$ if φ is a simplicial map, because thanks to this restriction, not only vertices are mapped to vertices but also simplices are mapped to simplices if we apply the usual convention for applying maps to subsets of their domains (see Notations and Conventions). A simplicial map that preserves dimensions, i.e. $\dim(\varphi(\sigma)) = \dim(\sigma)$ for every simplex σ, is called *rigid*.

VERTEX MAP
SIMPLICIAL MAPS
RIGID

If we have two simplicial maps $\varphi\colon V(\mathcal{A}) \to V(\mathcal{B})$ and $\psi\colon V(\mathcal{B}) \to V(\mathcal{A})$ such that $\psi \circ \varphi = \mathrm{id}_{V(\mathcal{A})}$ and $\varphi \circ \psi = \mathrm{id}_{V(\mathcal{B})}$, we call $\mathcal{A}$ and $\mathcal{B}$ *isomorphic* and write $\mathcal{A} \cong \mathcal{B}$. In this case, $\mathcal{A}$ and $\mathcal{B}$ have the same structure and are, somehow, interchangeable. This indicates that the actual ground set only plays a minor role and can be exchanged for any other set of the same size, as long as the simplices are modified accordingly.

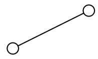
ISOMORPHIC

Before we continue with more terminology, we will have a look at a more visual representation of a simplicial complex. This time, we start with the simplices: A *geometric n-simplex* σ is the convex hull of $n+1$ affinely independent points $\{x_0, x_1, \ldots, x_n\}$ in $\mathbb{R}^d$ with $d \geqslant n+1$, called the *vertices* of σ. The convex hull of some points $\{y_0, y_1, \ldots, y_n\} \subset \mathbb{R}^d$ is defined as

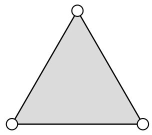
GEOMETRIC n-SIMPLEX

$$\operatorname{Conv}(\{y_0, y_1, \ldots, y_n\}) := \left\{ \sum_{i=0}^{n} t_i y_i \in \mathbb{R}^d \;\middle|\; \forall i \in [n]\colon t_i \in [0,1] \text{ and } \sum_{i=0}^{n} t_i = 1 \right\}$$

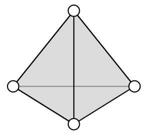
(a) A geometric 0-simplex.



(b) A geometric 1-simplex.



(c) A geometric 2-simplex.



(d) A geometric 3-simplex.

Figure 2.1: Examples of geometric simplices up to dimension 3.

and being affinely independent means that no point x_i is contained in the convex hull of the others. This way, a geometric 0-simplex is simply a point, a geometric 1-simplex is a line between two points, a geometric 2-simplex is a triangle and a geometric 3-simplex is a tetrahedron, as depicted in Figure 2.1, though not necessarily as regular as these examples. Note that every point of the convex hull of given points is uniquely determined by its *barycentric coordinates* $(t_0, t_1, \ldots, t_n)$ if we fix an order of the points.

BARYCENTRIC COORDINATES

We can define all the relations and properties we just assigned to abstract simplices in the same way for geometric simplices, in particular the *face* of a geometric n-simplex that is spanned by $\{x_0, x_1, \ldots, x_n\}$ is the convex hull of any subset of $\{x_0, x_1, \ldots, x_n\}$.

GEOMETRIC SIMPLICIAL COMPLEX

A collection of geometric simplices $\mathcal{K}$ is called a *geometric simplicial complex* if it satisfies two conditions:

(1) Every face of a geometric n-simplex $\sigma \in \mathcal{K}$ is contained in $\mathcal{K}$ as well, and
(2) the intersection $\sigma \cap \tau$ of any two geometric simplices $\sigma, \tau \in \mathcal{K}$ is a face of both of them (and thus contained in $\mathcal{K}$ by (1)).

Together these conditions are analogous to condition (1) of abstract simplicial complexes above. The union of all vertices of simplices in the complex is called *ground set* again and denoted by $V(\mathcal{K})$.

Just as we were able to exchange the ground set before, all that really matters to us know is the topology of $\mathcal{K}$, or more precisely, of the union of all simplices of $\mathcal{K}$, which we will call $|\mathcal{K}|$:

$$|\mathcal{K}| := \bigcup_{\sigma \in \mathcal{K}} \sigma$$

When we have two geometric simplicial complexes $\mathcal{K}$ and $\mathcal{H}$ with a homeomorphism $\varphi \colon |\mathcal{K}| \to |\mathcal{H}|$, i.e. a map that is bijective, continuous and whose inverse is continuous as well, then we call $\mathcal{K}$ and $\mathcal{H}$ *isomorphic* and write $\mathcal{K} \cong \mathcal{H}$ if the restriction of φ to a simplex of $\mathcal{K}$ is a simplex in $\mathcal{H}$ again. Because φ is bijective, this applies to its inverse as well.

We can easily construct an abstract simplicial complex $\mathcal{C}(\mathcal{K})$ from a geometric one $\mathcal{K}$, by setting the ground set $V(\mathcal{A})$ to be the union of the vertices of all simplices $\sigma \in \mathcal{K}$ and letting $\mathcal{C}(\mathcal{K})$ contain a subset $\{x_0, x_1, \ldots, x_n\}$ if and only if its convex hull is a simplex of $\mathcal{K}$.

STANDARD n-SIMPLEX

The inverse conversion is possible as well: Given an abstract simplicial complex $\mathcal{A}$ with $d := |V(\mathcal{A})| < \infty$, we can define a geometric simplicial complex $\mathcal{K} = \mathcal{G}(\mathcal{A})$ using the *standard n-simplex* Δ^n that is spanned by the standard basis vectors $\{e_1, e_2, \ldots, e_{n+1}\}$ of $\mathbb{R}^{n+1}$. All we have to do is construct Δ^{d-1}, which is conveniently spanned by d points, and define an injective (and by cardinality even bijective) map $p\colon V(\mathcal{A}) \to \{e_1, e_2, \ldots, e_d\}$. Then $\mathcal{K}$ is the subcomplex of Δ^{d-1} that contains a k-face $\sigma^k \in \Delta^{d-1}$ spanned by the vertices $\{x_0, x_1, \ldots, x_k\} \subseteq V(\Delta^{d-1})$ if and only if $p^{-1}(\{x_0, x_1, \ldots, x_k\}) \in \mathcal{A}$. Up to changing the ordering of basis vectors, this construction is unique. This ends up in the very high-dimensional space $\mathbb{R}^d$, and while lower dimensions would be possible, they require more effort. However, an abstract simplicial complex of dimension d has no equivalent geometric complex in $\mathbb{R}^n$ if $n \leqslant d$, because there are no geometric d-simplices in $\mathbb{R}^n$.

Combining these two constructions, we can exchange abstract and geometric simplicial complexes for each other, as we can build one from the other and vice versa. The following theorem gives the justification for this exchange:

Theorem 2.1. *Given an abstract simplicial complex $\mathcal{A}$ and a geometric simplicial complex $\mathcal{K}$, then*

(1) if $\mathcal{K} = \mathcal{G}(\mathcal{A})$, then $\mathcal{C}(\mathcal{K}) \cong \mathcal{A}$;
(2) if $\mathcal{A} = \mathcal{C}(\mathcal{K})$, then $\mathcal{G}(\mathcal{A}) \cong \mathcal{K}$.

Proof. (1) Because $V(\mathcal{C}(\mathcal{K}))$ is the union of all vertices of simplices of $\mathcal{K}$, which are all faces of Δ^{d-1} if $\dim(\mathcal{A}) = d$, we have $V(\mathcal{C}(\mathcal{K})) = \{e_1, e_2, \ldots, e_d\}$. Thus the bijection p chosen during the construction is actually a vertex map between $\mathcal{A}$ and $\mathcal{C}(\mathcal{K})$. It remains to show that p and p^{-1} are simplicial maps in order for $\mathcal{A}$ and $\mathcal{C}(\mathcal{K})$ to be isomorphic, which is trivial:
A subset $\sigma \subseteq V(\mathcal{C}(\mathcal{K})) = \{e_0, e_1, \ldots, e_d\}$ is a simplex of $\mathcal{C}(\mathcal{K})$ if and only if $\operatorname{Conv}(\sigma)$ is a simplex of $\mathcal{K}$, which in our case is exactly the case when $p^{-1}(\sigma) \in \mathcal{A}$.

(2) If we do one more step and construct $\mathcal{C}(\mathcal{G}(\mathcal{A}))$, we know from (1) that it is isomorphic to $\mathcal{A}$. Applying Lemma 2.2 on the next page, we are done. □

Lemma 2.2. *Given two abstract simplicial complexes $\mathcal{A}$ and $\mathcal{B}$ along with two geometric simplicial complexes $\mathcal{K}$ and $\mathcal{G}$ such that $\mathcal{A} = \mathcal{C}(\mathcal{K})$ and $\mathcal{B} = \mathcal{C}(\mathcal{G})$. Then every simplicial map $\varphi\colon V(\mathcal{A}) \to V(\mathcal{B})$ induces a continuous map $f\colon |\mathcal{K}| \to |\mathcal{G}|$ that maps simplices of $\mathcal{K}$ to simplices of $\mathcal{G}$.*

Proof. Remember from the definition of geometric simplices that every point x of an n-simplex σ with the vertices $\{x_0, x_1, \ldots, x_n\}$ can be represented using barycentric coordinates as

$$x = \sum_{i=0}^{n} t_i x_i \quad \text{with} \sum_{i=0}^{n} t_i = 1.$$

We use this notation to define the map f simplex-wise by setting

$$f(x) := \sum_{i=0}^{n} t_i \varphi(x_i) \quad \text{for } x \in \operatorname{Conv}(\sigma).$$

On a simplex-basis, this is obviously continuous, as the $\varphi(x_i)$ stay fixed and only the barycentric coordinates t_i change, which they do in a continuous manner. It also fits nicely together on simplex boundaries, because the (scaled) barycentric coordinates of the larger simplex have to match with the barycentric coordinates of its face. □

We will work with abstract simplicial complexes whenever possible, because their structure is much easier to handle, but visualize them using their geometric version. Also, sometimes it will not be possible to avoid using their geometric counterparts.

Note that the standard geometric n-simplex Δ^n, which we only introduced as a tool for one of the constructions above, is useful in many ways. For example, using barycentric coordinates again, one can see that there is one unique affine map from Δ^n to every geometric n-simplex σ^n, called its *characteristic map*, if we fix the ordering of the vertices of both simplices. This will be useful in order to not have to consider every possible geometric simplex, but only the standard simplices.

CHARACTERISTIC MAP

2.1.2 Maps, colors and labels

An important tool for modeling distributed computing are the so called *carrier maps*. Given two abstract simplicial complexes $\mathcal{A}$ and $\mathcal{B}$, a carrier map $\Phi\colon \mathcal{A} \to 2^{\mathcal{B}}$ maps each simplex of $\mathcal{A}$ to a subcomplex of $\mathcal{B}$ monotonically, i.e. $\Phi(\tau) \subseteq \Phi(\sigma) \subseteq \mathcal{B}$ if $\tau \subseteq \sigma \in \mathcal{A}$. Just like simplicial maps, carrier maps can be *rigid*, namely when the image of a d-simplex

CARRIER MAP

RIGID

STRICT is a pure subcomplex of dimension d. Furthermore, they can also be *strict* if for two simplices $\sigma, \tau \in \mathcal{A}$ we have $\Phi(\sigma \cap \tau) = \Phi(\sigma) \cap \Phi(\tau)$.

Note that because $\sigma \cap \tau \subseteq \sigma$ and $\sigma \cap \tau \subseteq \tau$, monotonicity of Φ already gives us $\Phi(\sigma \cap \tau) \subseteq \Phi(\sigma)$ and $\Phi(\sigma \cap \tau) \subseteq \Phi(\tau)$ and thus half of the equation defining a strict carrier map, namely the subset-part. Hence, we could define strict carrier maps equivalently by only requiring the superset-part $\Phi(\sigma \cap \tau) \supseteq \Phi(\sigma) \cap \Phi(\tau)$.

Given a carrier map $\Phi\colon \mathcal{A} \to 2^{\mathcal{B}}$, not every simplex $\tau \in \bigcup \Phi(\mathcal{A})$ is directly an image of a simplex of $\mathcal{A}$, but it is always contained in a subcomplex $\Phi(\sigma)$ for some $\sigma \in \mathcal{A}$. If Φ is not strict, there might be different $\sigma, \sigma' \in \mathcal{A}$ of minimal dimension with $\tau \in \Phi(\sigma) \cap \Phi(\sigma')$, but if it is strict, $\Phi(\sigma) \cap \Phi(\sigma') = \Phi(\sigma \cap \sigma')$, so $\sigma \cap \sigma'$ would be a simplex of smaller dimension whose image still contained τ. Thus, if Φ is strict, there is a unique $\sigma \in \mathcal{A}$

CARRIER of minimal dimension such that $\tau \in \Phi(\sigma)$. We call this σ the *carrier of* τ *(under* Φ*)* and denote it by $\operatorname{Car}(\tau, \Phi)$ — or $\operatorname{Car}(\tau)$ if no confusion over Φ can arise.

Obviously, we can compose two carrier maps $\Phi\colon \mathcal{A} \to 2^{\mathcal{B}}$ and $\Psi\colon \mathcal{B} \to 2^{\mathcal{C}}$ by setting

$$(\Psi \circ \Phi)(\sigma) := \Psi(\Phi(\sigma)) := \bigcup_{\tau \in \Phi(\sigma)} \Psi(\tau). \tag{2.1}$$

Given a simplicial map $\psi\colon \mathcal{B} \to \mathcal{C}$, the construction looks exactly the same, just replace Ψ with ψ. The other way round, if $\varphi\colon \mathcal{A} \to \mathcal{B}$, it is even simpler, because we need no unions whatsoever: $(\Psi \circ \varphi)(\sigma) := \Psi(\varphi(\sigma))$.

Proposition 2.3. *Let $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ be abstract simplicial complexes, $\Phi\colon \mathcal{A} \to 2^{\mathcal{B}}$ and $\Psi\colon \mathcal{B} \to 2^{\mathcal{C}}$ carrier maps and $\varphi\colon \mathcal{A} \to \mathcal{B}$ and $\psi\colon \mathcal{B} \to \mathcal{C}$ simplicial maps.*

(1) If Φ and Ψ are strict, so is $\Psi \circ \Phi$.
(2) If $f \in \{\Phi, \varphi\}$ and $g \in \{\Psi, \psi\}$ are rigid, so is $g \circ f$.

Proof. (1) Proving the strictness of $\Psi \circ \Phi$ is a matter of simply writing out the set operations involved, and we will skip it for brevity. It can be found for example in [HKR13, Prop. 3.4.6].

(2) We will first handle the most difficult case and let $f = \Phi$ and $g = \Psi$ both be rigid carrier maps. Since Φ is rigid, $\Phi(\sigma)$ is a pure d-dimensional complex if σ is a d-simplex of $\mathcal{A}$. And as Ψ is monotonic, it suffices to take the union over all d-simplices of $\Phi(\sigma)$ instead of all its simplices in Equation (2.1). But because Ψ is

rigid as well, the image of every d-simplex is a pure d-dimensional complex again, and so is the union of all these d-dimensional complexes, because no new facets appear by taking the union of subcomplexes.
Next, let $f = \Phi$ be a rigid carrier map and $g = \psi$ a rigid simplicial map. Φ maps a d-simplex to a pure d-dimensional complex, and ψ takes every simplex of that complex to a simplex of the same dimension, preserving faces because it operates on the vertex level. Thus the image of $\psi \circ \Phi$ is a pure d-dimensional complex itself.
More obviously, if $f = \varphi$ is a rigid simplicial map and $g = \Psi$ is a rigid carrier map, we do not need to take the union but simply compose the two maps. The image $\varphi(\sigma)$ has the same dimension as σ itself, and is mapped by Ψ to a pure complex of that dimension, so $\Psi \circ \phi$ is rigid.
Finally, $\psi \circ \varphi$ is of course rigid because the dimension of every simplex is preserved by both ψ and φ. □

When we have a simplicial map $\varphi\colon \mathcal{A} \to \mathcal{B}$ and a carrier map $\Phi\colon \mathcal{A} \to 2^{\mathcal{B}}$, we want to have some means of talking about their compatibility, and in fact there is only one sensible way to do that: φ is said to be *carried by* Φ if $\varphi(\sigma) \in \Phi(\sigma)$ holds for every simplex $\sigma \in \mathcal{A}$. In the same vein, another carrier map $\Psi\colon \mathcal{A} \to 2^{\mathcal{B}}$ can also be carried by Φ if it fulfills $\Psi(\sigma) \subseteq \Phi(\sigma)$ for every simplex $\sigma \in \mathcal{A}$. In both cases, we write $\varphi \subseteq \Phi$ and $\Psi \subseteq \Phi$, respectively. CARRIED BY

In a slightly different setup, if we are given three simplicial complexes $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ with carrier maps $\Phi_A\colon \mathcal{C} \to \mathcal{A}$ and $\Phi_B\colon \mathcal{C} \to \mathcal{B}$ and a simplicial map $\phi\colon \mathcal{A} \to \mathcal{B}$, we say that ϕ is *carrier-preserving* if $\mathrm{Car}(\phi(\sigma), \Phi_B) \subseteq \mathrm{Car}(\sigma, \Phi_A)$ for every simplex $\sigma \in \mathcal{A}$. CARRIER-PRESERVING

One more important concept is how we associate data to the vertices, as we have seen that the ground set is not suitable for this task because it can be exchanged and shuffled at any time. Given an abstract simplicial complex $\mathcal{A}$ and a set C of available values, a map $\chi\colon V(\mathcal{A}) \to C$ that assigns each vertex a value is called a *labeling*, and $\mathcal{A}$ is *labeled with C by* χ. If for every simplex $\sigma \in \mathcal{A}$ we have that $\chi|_\sigma$ is injective, meaning that no two vertices of a simplex have the same label, it is called a *coloring*, and $\mathcal{A}$ may be *colored* with C. If $|C| = \dim \mathcal{A}$, we call χ *minimal*. Especially in the context of distributed computing the colors might be called *names* as well. Of course, a single complex can have more than one labeling or coloring to attach more than one kind of data to the vertices. LABELING COLORING

If a vertex map $\varphi\colon \mathcal{A} \to \mathcal{B}$ between two complexes that are both colored with C (by $\chi_{\mathcal{A}}$ and $\chi_{\mathcal{B}}$ respectively) preserves these colors, i.e. $\chi(\varphi(\sigma)) = \chi(\sigma)$ for every simplex σ, we call φ *(χ-)chromatic*, *color-preserving* or *name-preserving*. Something similar applies to carrier maps: If $\Phi\colon \mathcal{A} \to 2^{\mathcal{B}}$ is a carrier map such that $\Phi(\sigma)$ is colored with $\chi_{\mathcal{A}}(\sigma)$ by $\chi_{\mathcal{B}}$ for every simplex $\sigma \in \mathcal{A}$, then Φ is called *chromatic*. This is mainly interesting for rigid carrier maps, though.

CHROMATIC

A simplex σ of a simplicial complex $\mathcal{A}$ labeled with χ is called *(χ-)monochromatic* or *c-monochromatic* if $\chi(\sigma) = \{c\}$. This counter-intuitive term—"chromatic" hinting at a coloring, whereas in our definition of a coloring no simplex can be monochromatic—stems from the fact that in literature, the term "coloring" is also used for what we call a labeling. What we call a coloring is then called a "proper coloring". However, we feel that less confusion arises if there are two strictly different terms used.

MONO-CHROMATIC

2.1.3 Subdivisions

SUBDIVISIONS

The most important way of modifying simplicial complexes are *subdivisions*. They are best defined using the geometric view of things: Given geometric simplicial complexes $\mathcal{K}$ and $\mathcal{H}$ that live in the same ambient space $\mathbb{R}^d$, we call $\mathcal{H}$ a subdivision of $\mathcal{K}$ if $|\mathcal{K}| = |\mathcal{H}|$ and each simplex of $\mathcal{K}$ is the union of finitely many simplices of $\mathcal{H}$. We can even restrict this to the union of finitely many simplices of the same dimension, because obviously higher dimensions would yield a larger union, while lower dimensions are already contained as faces. Intuitively, this means that we "cut" some (not necessarily all) simplices into finitely many new ones, but they still occupy the same space. Two very basic examples are depicted in Figure 2.2 on page 13.

Thanks to our earlier constructions, we can also apply this definition to abstract simplicial complexes without having to know exactly what the conditions in that realm are: If for two abstract simplicial complexes $\mathcal{A}$ and $\mathcal{B}$ we have that $\mathcal{A} \cong \mathcal{C}(\mathcal{K})$ and $\mathcal{B} \cong \mathcal{C}(\mathcal{H})$, and $\mathcal{H}$ is a subdivision of $\mathcal{K}$, then we call $\mathcal{B}$ a subdivision of $\mathcal{A}$ as well.

Note that every subdivision $\mathcal{H}$ of $\mathcal{K}$ induces a rigid and strict carrier map $\Psi\colon \mathcal{K} \to 2^{\mathcal{H}}$ that maps a simplex of $\mathcal{K}$ to all the simplices it has been subdivided into:

$$\Psi(\sigma) := \{\tau \in \mathcal{H} \mid \tau \subseteq \sigma\}.$$

That Ψ is rigid and strict directly follows from the fact that each simplex $\sigma \in \mathcal{K}$ is the union of finitely many simplices of $\mathcal{H}$, thus $|\sigma| = |\Psi(\sigma)|$.

This carrier map enables us to simply call a subdivision *chromatic*, if its induced carrier map is chromatic. Also, we now can speak about the carrier of a simplex in the subdivision. CHROMATIC SUBDIVISION

A more systematic approach than dividing only some simplices would not distinguish between different n-simplices, but only care about the dimension n. This is what we call a *boundary-consistent subdivision*: For every dimension $n \in \mathbb{N}$, let $\mathcal{S}_n$ be a subdivision of the standard geometric n-simplex Δ^n. These subdivisions have to be compatible: For $m < n$, let σ^m be a face of Δ^n, and let $\varphi\colon \Delta^m \to \sigma$ be its characteristic map, where we use the ordering of the standard basis for both simplices. Then the restriction of the subdivision $\mathcal{S}_n$ to σ^m has to agree with $\varphi(\mathcal{S}_m)$. This is what "boundary-consistent" should intuitively mean: If we know how to subdivide a low- and a high-dimensional simplex, the boundary of the high-dimensional simplex should be subdivided according to how we subdivide lower dimensions. BOUNDARY-CONSISTENT SUBDIVISION

Using such a boundary-consistent subdivision, we can define a *subdivision operator* $\mathcal{S}$ that subdivides every simplex σ of a geometric simplicial complex with fixed vertex ordering according to the characteristic map φ of σ, namely it replaces σ by $\varphi(\mathcal{S}_{\dim(\sigma)})$. From our examples in Figure 2.2 on page 13, both could be caused by a subdivision operator. However, we will see from their precise construction that only the standard chromatic subdivision in Figure 2.2(b) is a subdivision operator. SUBDIVISION OPERATOR

We will use a specific method to combine existing simplicial complexes into new ones and thereby create a subdivision: The *join* of two disjoint abstract simplices σ and τ, denoted $\sigma * \tau$, is their union $\sigma \cup \tau$. Equivalently, the join of two geometric simplices with affinely independent vertices $\{v_0, v_1, \ldots, v_n\}$ and $\{w_0, w_1, \ldots, w_m\}$ is the convex hull of the union of vertices. Similarly, the *join* of two simplicial complexes $\mathcal{A}$ and $\mathcal{B}$ is the complex of all possible joins of simplices: $\mathcal{A} * \mathcal{B} := \{\sigma * \tau \mid \sigma \in \mathcal{A}, \tau \in \mathcal{B}\}$. JOIN

As a special case, given an m-simplex σ, minimally colored with C by χ_σ, a chromatic subdivision $\Psi(\partial\sigma)$ of its boundary, and another m-simplex τ also minimally colored with C by χ_τ, the *cone over* $\Psi(\partial\sigma)$ *for* τ, denoted $\tau \circledast \Psi(\partial\sigma)$, is the simplicial complex defined as CONE

$$\bigcup\{\tau' * \Psi(\sigma') \mid \sigma' \in \partial(\sigma),\ \tau' \subseteq \tau \text{ and } \chi_\tau(\tau') \cap \chi_\sigma(\sigma') = \emptyset\}.$$

If we cone geometrically, we would assume τ as a scaled-down copy of σ with the same barycenter but mirrored point-wise at this barycenter. More precisely, given the vertices $v_0, v_1, \dots, v_m$ of σ, we place the vertices $w_0, w_1, \dots, w_m$ of τ at

$$w_i = \sum_{k=0}^{m} t_k v_k \quad \text{with } t_k = \begin{cases} \frac{m}{2m(m+1)} & \text{if } k = i, \\ \frac{2m+1}{2m(m+1)} & \text{else.} \end{cases}$$

One can easily check that the coefficients t_k sum up to 1, making the w_i lie inside the simplex σ, and that t_i is always lower than the other coefficients that are all equal, positioning w_i on the line from v_i through the barycenter, but on the far end as seen from v_i.

Proposition 2.4. *If σ is the minimal-colored simplicial complex containing a single simplex along with its faces, and $\Psi(\partial\sigma)$ is a chromatic subdivision of $\partial\sigma$, then $\Psi(\sigma) := \tau \circledast \Psi(\partial\sigma)$ is a chromatic subdivision of σ.*

The proof of this general statement is very technical and we refer to [CR10, the Appendix, p. 299 ff.] for its full presentation. While they used so called "divided images" instead of subdivisions, their method proof also works for subdivisions. It mainly involves two steps: First that $\tau \circledast \partial\sigma$ is a subdivision of σ and second that $\tau \circledast \Psi(\partial\sigma)$ is a subdivision of $\tau \circledast \partial\sigma$ and thus also of σ.

BASIC CHROMATIC SUBDIVISION

Here are two examples of cones that we will actually use extensively later on: Given an n-simplex σ we define the *basic chromatic subdivision $\beta(\sigma)$ of σ* as the cone over $\partial(\sigma)$ (i.e. with Ψ being the identity) for another n-simplex τ (Figure 2.2(a) on the next page). Note that this is no subdivision operator: We only subdivide the highest-dimensional simplex, not its boundary. If we did subdivide all faces, starting with the lowest dimensional, using cones, we would end up with the *standard chromatic subdivision* (Figure 2.2(b) on the facing page), denoted $\chi(\sigma)$. Note that for these two special cones, the proof that they actually are subdivisions can be simplified greatly by using Schlegel diagrams, as was done in [Koz12].

STANDARD CHROMATIC SUBDIVISION

Proposition 2.5. *Let $\beta(\sigma)$ be the basic chromatic subdivision of an n-simplex σ colored with Π by c. Then there is a bijection*

$$f\colon \{\sigma^n \in \beta(\sigma) \mid \dim(\sigma^n) = n\} \longleftrightarrow [1]^\Pi - \{0\}^\Pi.$$

Proof. The simplices (and in particular all n-simplices) of $\beta(\sigma)$ by definition all have the form $\tau' * \sigma'$ with $\tau' \subseteq \tau$ and $\sigma' \in \partial(\sigma)$, the two having no common colors. Because an

(a) Basic chromatic subdivision (b) Standard chromatic subdivision

Figure 2.2: Two chromatic subdivisions of a 2-simplex colored with green, red and blue.

n-simplex has $n+1$ vertices, every color is present exactly once, either in τ' or in σ'. We then simply set

$$f(\sigma)_i := \begin{cases} 1 \text{ if } i \in c(\tau'), \\ 0 \text{ if } i \in c(\sigma'). \end{cases} \qquad \text{for every } n\text{-simplex } \sigma \text{ and every } i \in \Pi$$

We already saw that this is injective, but it is also surjective by definition: $\beta(\sigma)$ is the union of the joins of all possible combinations of τ' and σ', allowing for σ' to be empty and $\tau' = \tau$, which would result in an all-1 tuple. Only the all-0 tuple must be excluded, because we choose $\sigma' \in \partial(\sigma)$, which excludes σ itself. □

In the subdivision $\Psi(\sigma)$ of a single n-simplex σ, we will call $\tau \in \Psi(\sigma)$ an *n-corner* if, for every dimension $1 \leqslant i \leqslant n$, there is an i-face τ^i of τ such that $\mathrm{Car}(\tau^i, \Psi)$ is an i-face of σ and they build an increasing chain $\tau^0 \subseteq \tau^1 \subseteq \cdots \subseteq \tau^n = \tau$. This seemingly artificial construction has the advantage that all n-corners of a chromatic subdivision have the same orientation [CR08, Lem. 2.4] and can be used to trace orientation through multiple iterations of coning, as we will do later in Chapter 3. n-CORNER

2.1.4 Pseudomanifolds and orientation

A pure simplicial complex of dimension n, no matter if abstract or geometric, is called an *n-dimensional pseudomanifold* if every of its $(n-1)$-simplices has exactly two co-faces, and if every two n-simplices σ and σ' can be connected by a series of n-simplices $\sigma = \sigma_1, \sigma_2, \ldots, \sigma_m = \sigma'$ such that $\sigma_i \cap \sigma_{i+1}$ is an $(n-1)$-simplex for all $i \in \{1, 2, \ldots, m-1\}$. Such a series is called a *simplex path* from σ to σ'. PSEUDO-MANIFOLD SIMPLEX PATH

Remember that in a pure simplicial complex, every simplex with codimension one has at least one co-face, and that those with exactly one were called the boundary. As they now have exactly two, a pseudomanifold has an empty boundary. We can loosen this restriction by only requiring that every $(n-1)$-simplex has at most two co-faces, and then call the complex a *pseudomanifold with boundary*.

Pseudomanifolds can have the interesting property of being orientable. To understand this concept, we first need a notion of an orientation on a simplex: Two orderings on the vertices of an n-simplex can differ by an even or by an odd permutation of the vertices, thus splitting all possible orderings in two equivalence classes. If $(v_0, v_1, \dots, v_n)$ is an ordering of the vertices, we write the corresponding equivalence class as $[v_0, v_1, \dots, v_n]$ and the other one as $-[v_0, v_1, \dots, v_n]$. These two classes are the possible *orientations* of the simplex.

ORIENTATION

We want an orientation of an n-simplex σ to induce an orientation on all of its $(n-1)$-faces. If the orientation of σ is $[v_0, v_1, \dots, v_n]$, we therefore define the induced orientation of the $(n-1)$-face $\sigma_i^{n-1} = \{v_0, v_1, \dots, v_n\} - v_i$ to be $(-1)^i[v_0, \dots, \hat{v}_i, \dots, v_n]$, where $\hat{v}_i$ means omission of v_i. In other words, we choose the naturally induced ordering to determine the equivalence class if i is even, and choose the opposite one if i is odd. Note that another representation of σ's orientation either preserves the parity of i or changes both, the parity of i and the equivalence class $[v_0, \dots, \hat{v}_i, \dots, v_n]$.

When the orientations assigned to two n-simplices intersecting in an $(n-1)$-face induce opposite orientations in this face, they are called *coherently oriented*, and if it is possible to assign an orientation to every n-simplex of a pseudomanifold such that every neighboring pair is coherently oriented, the pseudomanifold is called *coherently orientable* or—together with the orientation—*coherently oriented*. Note that this is only possible because in a pseudomanifold, every $(n-1)$-simplex has exactly two cofaces. If we were dealing with the simplex of a general complex and it had more cofaces, at least one orientation would necessarily be induced by more than one of them, simply because we only have two orientations at hand.

COHERENTLY ORIENTABLE

However, because we mostly are dealing with chromatic pseudomanifolds, we can use a much simpler version of orientability:

Lemma 2.6. *Let the n-dimensional pseudomanifold $\mathcal{A}$ be minimally colored. Then $\mathcal{A}$ is orientable if and only if there is a map*

$$\varepsilon\colon \{\sigma^n \in \mathcal{A} \mid \dim \sigma = n\} \to \{-1, +1\}$$

assigning either -1 *or* $+1$ *to every n-simplex in a way that no two neighboring n-simplices are assigned the same value.*

Proof. Without loss of generality, we assume the colors to be $[n]$. By considering not orderings of the vertices themselves but orderings of their colors, we simply call the orientation $[0, 1, \ldots, n]$ "$+1$" and its opposite "-1". Now by the minimality of the cover, the common face $\sigma_i^{n-1} = \tau_j^{n-1}$ of two neighboring n-simplices σ and τ is missing vertices of the same color in both, so actually $i = j$. Thus, the induced orientations of the face differs if and only if the coloring orientations of σ and τ differ. □

It is well known that the subdivision $\mathcal{H}$ of an orientable pseudomanifold $\mathcal{K}$ is still an orientable pseudomanifold. While the pseudomanifold-property follows almost directly from $|\mathcal{K}| = |\mathcal{H}|$, it is more difficult to see and that the orientability of a pseudomanifold $\mathcal{K}$ actually only depends on the underlying topological space $|\mathcal{K}|$ as well. In fact, such combinatorial invariants are what combinatorial topology is all about. But as this is the result of the more involved theory of homology, its proof would go beyond the scope of this thesis and we will skip it. We instead refer to [Koz08] and [Mun84] for a more detailed discussion of these matters.

Instead, to conclude this chapter on the basics of combinatorial topology that we require, we note a special property of the basic chromatic subdivision:

Lemma 2.7. *Given an oriented n-simplex* σ *with an arbitrary binary labeling B, first coning its boundary with another n-simplex* τ *and then coning the boundary of* τ *with yet another n-simplex* γ *yields a complex in which we can label* γ *with 0,* $\partial(\tau)$ *with 1 and get exactly one 0-monochromatic n-simplex (namely* γ*) that has the same orientation as* σ *without modifying the boundary of* σ*.*

Proof. If we ignore the labeling, these are just two basic chromatic subdivisions executed consecutively. That way it is obvious that $\partial \sigma$ stays untouched. Furthermore, even if the inner simplex of the first basic chromatic subdivision (the one that corresponds to the tuple $(1, \ldots, 1)$ in Proposition 2.5 on page 12, in this case τ) had another orientation than the simplex that was subdivided (σ in this case), doing the subdivision a second time on this simplex would reverse the orientation again, making sure that γ has the same orientation σ had.

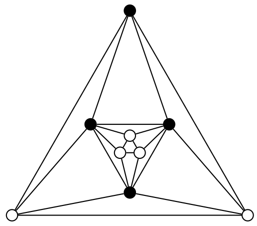


Figure 2.3: After two appropriate basic chromatic subdivisions, there is an isolated 0-monochromatic n-simplex in the middle, indicated using white for 0-vertices and black for 1-vertices.

It remains to show that there are no 0-monochromatic simplices other than γ. We know that every simplex of a basic chromatic subdivision contains at least one vertex from the simplex the cone was taken with. Thus, every simplex in $\beta(\sigma) \setminus \tau$ cannot be 0-monochromatic, because it contains a vertex of τ which is 1-monochromatic.

Also, except the innermost, every simplex in a basic chromatic subdivision contains at least on vertex from the original outer simplex. Applying this to the second subdivision yields that every simplex in $\beta(\tau) \setminus \gamma$ is not 0-monochromatic, again, because τ is 1-monochromatic. □

An example of this construction in dimension two can be seen in Figure 2.3. Note that while the construction may yield 1-monochromatic simplices, we will always cone the example with a 0-monochromatic simplex, making the 1-monochromatic ones disappear.

2.2 Distributed Computing

Now we need to apply the framework of combinatorial topology to the problems of distributed computing. We limit ourselves to the parts relevant to our goal of analyzing the weak symmetry breaking. A good introduction to the whole theory, covering other types of problems as well, can be found in [HKR13].

2.2.1 Model

To establish some basic terminology: In a distributed computing *system*, there are $n + 1 > 1$ *processes* $\{p_0, p_1, \ldots, p_n\}$ that execute a *protocol* in order to solve a *task* by *communicating*. Each process p_i is said to have the *(process) ID* or *name* i and may or may not be assigned an additional *input value* $v_i \in V^{in}$. After finishing the protocol, each process p_i (that did not crash) returns an *output value* $o_i \in V^{out}$.

The "processes" can be thought of as computers communicating via a network, processor cores of a single computer communicating via shared memory, or even humans communicating by writing letters. While we will detail the meaning of protocol and task later on, we will start with the assumptions made concerning the environment we are in: What constraints and freedoms apply to the processes, communication and protocol?

There are several models for distributed computing, each considering different situations. We will assume a model that has the following properties:

Asynchronous The processes of the system act asynchronously, meaning that not all processes do a computation step at the same time, but as fast as they can or want. Because we do not care about actual speeds, this property is characterized by the varying relative speeds of the processes: While one process may still be executing the first step of the protocol, another one might already be finished. Furthermore, the relative speeds can change during the execution of the protocol.

Wait-Free No process is allowed to wait for another process to finish with some part of the protocol. This includes passive waiting (doing nothing until a message arrives) as well as active waiting (looping through the same actions until the situation changes). Note that due to the asynchronicity, it is equivalent to assume that up to $n - 1$ processes may silently crash, i.e. they stop doing anything and fall silent eternally, at any time during the execution, because a very slow process cannot be distinguished from a crashed process if no other process is allowed to wait until it sends a message.

Rank-Symmetric As our model is wait-free, it is possible that any proper subset of processes fails. Consider an execution where only processes p_i with $i \in I_1 \subseteq [n]$ participate (i.e. all others have crashed). Given another subset $I_2 \subseteq [n]$ with $|I_1| = |I_2|$, let $r\colon I_1 \to I_2$ be an order-preserving bijection. We then require that $o_i = o_{r(i)}$ for all $i \in I_1$. This property prevents trivial decisions based solely on the process' own ID: By the wait-free property, no process can be sure to know all participating processes, which makes it impossible to anticipate if, for example, the own process ID is the lowest.

2.2.2 Tasks

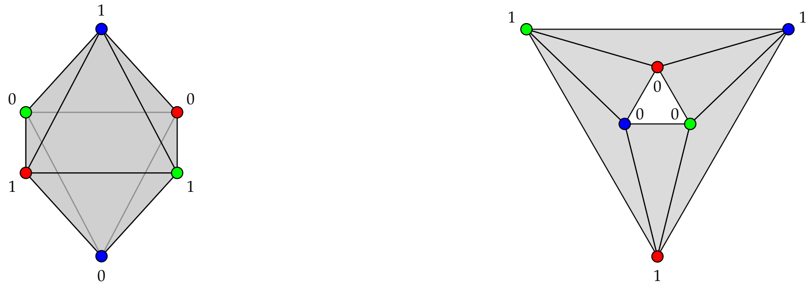
TASK

To analyze the computability of a problem or, as we will call it from now on, a *task*, we need to formalize what we even mean by that term. First of all, we need to model the initial situation: Which combinations of input values, if any, are allowed to start the task. This is where our simplicial complex comes in handy: Given the set $\Pi = \{p_0, p_1, \dots, p_n\}$ of processes and a set V^{in} of values that a single process can start with, a pure, n-dimensional simplicial complex $\mathcal{I}$ on the ground set $S \subseteq \Pi \times V^{in}$ is called an *input complex* if it is colored with Π and labeled with V^{in} by the respective projections pr_Π and $\mathrm{pr}_{V^{in}}$. The processes $p_{i_1}, p_{i_2}, \dots, p_{i_m}$ are allowed to start the task with the input values $v_{i_1}, v_{i_2}, \dots, v_{i_m}$ if and only if there is a simplex $\{(p_{i_1}, v_{i_1}), (p_{i_2}, v_{i_2}), \dots, (p_{i_m}, v_{i_m})\} \in \mathcal{I}$. We call this simplex a *configuration*.

INPUT COMPLEX

CONFIGURATION

Modeling the initial situation as a simplicial complex makes sense, because the subset of a configuration is a configuration again, it just does not assign a value to every process. This situation is the same as when all processes have values assigned, but some of them crash before starting, thus the input values of the others do not matter for the result of the few finishing first. Furthermore, every possible combination of assignments to some processes has to be expandable to an assignment to all processes, thus $\mathcal{A}$ has to be a pure complex of dimension n.



(a) No further restrictions. (b) No three processes can have the same value.

Figure 2.4: Two possible input (or output) complexes for three processes each having a boolean value. Each vertex represents the assignment of one value to one process, where the value is denoted by the label and the process which it is assigned to by the color. Possible combinations of assignments are indicated by lines and gray triangles.

Let us have an example for this. If a task that is to be solved by three processes, $\Pi = \{p_0, p_1, p_2\}$, requiring each of them to have a boolean input value, $V^{in} = [1]$, and imposing no further restrictions, we have $S = \{p_1, p_2, p_3\} \times [1]$ and

$$\mathcal{A} = \left\{ A \subseteq S \,\middle|\, |\operatorname{pr}_\Pi A| = |A| \right\}.$$

However, if the task requires that not all three processes have the same input value, S would stay the same but $\mathcal{A}$ would shrink to

$$\left\{ A \subseteq S \,\middle|\, |\operatorname{pr}_\Pi A| = |A| \wedge |\operatorname{pr}_{V^{in}} A| > 1 \right\}.$$

Both complexes are visualized in Figure 2.4.

Similarly, we can construct an *output complex* $\mathcal{O}$ to define what combinations of output values are admissible, which again is pure n-dimensional, colored by Π but this time labeled by V^{out}. Both of our examples for input complexes could be output complexes as well, and in fact, the second one is exactly the output complex for the Weak Symmetry Breaking task for three processes, as we will see in Chapter 3. OUTPUT COMPLEX

Given some input configuration for each of the participating processes, the task specifies what each process may output after some computation and communication. This is encoded in a name-preserving (i.e. pr_Π-chromatic) carrier map $\Delta \colon \mathcal{I} \to 2^{\mathcal{O}}$. Note that by using a carrier instead of a simplicial map, tasks that allow more than one output value for a given input value can be formulated.

2.2.3 Protocols

Now that we know how to describe the task we are supposed to solve, how do we model the process of its solution? An important insight is that the communication and the computation can theoretically be separated completely thanks to our requirement of wait-freeness: If the processes simply communicate everything they know, either from earlier communications or from their input values, then every process can compute the state and decisions of all other processes that already completed the protocol after completing the communication phase itself. We will exclusively use such *full information protocols*.

The communication between processes is based on $n + 1$ so called *registers* $\{m_0, \ldots, m_n\}$ that store arbitrary values, one for each process. Each process p_i can write only to its own register m_i, replacing its previous contents, but read all registers at once. All registers are initialized with the special value $\perp$, that cannot be written by any process. Thus, a register contains $\perp$ if and only if it has not been written to yet by its associated process, meaning that this process has not yet reached a point in the protocol where it would write to the register.

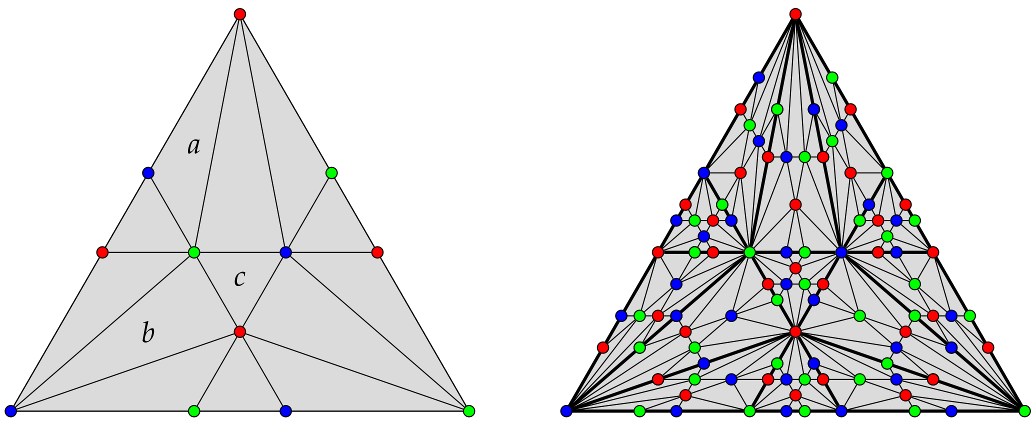
LAYERS The communication happens in *rounds* or *layers*, as we will call them. Each layer has its own set of registers, and in each layer, some processes simultaneously write to their according registers of this round and read a snapshot of all of this round's registers immediately after. Since we only consider full information protocols, no computation happens between the rounds, thus each process writes its input value v_i in its first round. Later, in its i-th round, it writes the snapshot it read in its $(i-1)$-th round. Finally, after taking part in a predetermined number of rounds K, each process p_i *decides* on an output value o_i. This is called the *(layered) Immediate Snapshot (IS) model*. IS MODEL

The IS model is not very realistic, as we forbid a process to inspect the value of registers corresponding to earlier rounds, which is hardly justifiable in real-world applications. It is, however, equivalent to more realistic models in terms of task solvability, as is proven in [HKR13, Chap. 14], thus its usage does not make the results any less useful.

To formalize this model in terms of combinatorial topology, we want to define a subdivision operator of the input complex whose application corresponds to one round of communication. Modeling a layer as a subdivision makes sense, because depending on the order in which the processes execute their rounds, different configurations of the system are possible, more than there were at the beginning, and thus more simplices should be present. Using not just a subdivision but a subdivision operator is necessary, because a process only knows about those other processes $\Pi' \subseteq \Pi$ that executed their rounds earlier or in parallel, not about the ones that follow. Thus the subdivision of the subcomplex consisting only of simplices labeled with Π' must be the same as the restriction of the subdivision of the whole input complex to those simplices labeled only with Π'—which is exactly what the usage of a subdivision operator guarantees.

We will start with a simple case: Imagine an input complex $\mathcal{I} = \sigma^n$ consisting of only one n-simplex, and set $K = 1$. If every process p_i only executes one round and has only one possible input value v_i, all that matters is in which order they execute their round. We will not do a full formal construction and proof it. Rather, we refer to [Koz12] and will illustrate some key examples of execution orders for three processes $\{\text{red}, \text{green}, \text{blue}\}$ using Figure 2.5(a) as a possible protocol complex to illustrate that it probably is the right choice. The execution can be imagined as the processes "moving around" on the vertices of their color, starting at the corner vertices.

First, consider the execution order where red, blue and green consecutively execute their rounds. Red does not know about any other process, thus it has to stay on its corner vertex by the property of the subdivision operator. Then blue executes and can choose any blue vertex on the line between the red and blue corner vertex. At the same time, it has to choose a blue vertex of a simplex that contains the red corner vertex, because



(a) (b)

Figure 2.5: The protocol complex $\mathcal{P}$ for three processes, indicated by different colors, after (a) one or (b) two rounds of immediate snapshots. Some simplices of (a) are labeled for reference. The edges of (a) inside (b) are drawn thicker to highlight that (b) is indeed a subdivision of (a).

red already decided on this one and the simplices represent the possible configurations. Thus choosing its own blue corner vertex would not be possible and it has to choose the blue vertex of the upper left simplex labeled with a in the figure. Finally, it is green's turn, and though green is free to choose any simplex in the whole complex because it executes last and knows that all processes participate, it is bound to choose the green vertex of a because this is the only vertex contained in a simplex together with the red and blue vertices that were already fixed. In total, a models exactly this order of executions. Note that green cannot distinguish this execution order from one where first blue and then red executed. Luckily, the green vertex of a covers both cases, and in fact, all co-faces of a vertex represent configurations that are indistinguishable to that vertex's process.

Another example: What if blue executes first, and red and green follow simultaneously? Red and green cannot distinguish this situation from a fully sequential execution like we just had it. So red would choose the same vertex that it would choose if green had run before, and vice versa. We end up with the vertices of the simplex b.

Finally, if all three processes executed the round simultaneously, the same argument applies: They cannot distinguish this situation from a fully sequential run where the two others, thus they choose the vertices of the simplex c.

All in all, Figure 2.5(a) on page 21 seems to fit the requirements for the protocol complex for $K = 1$, $n = 2$ and $\mathcal{I} = \sigma^n$. We see that it is the standard chromatic subdivision of $\mathcal{I}$, and could argue for every other dimension in exactly the same way.

If $K > 1$, every following round starts at a simplex of the protocol complex of the previous round and subdivides it in exactly the same way. This is because it does not matter if the first process that executes the second round does so before the last one executes the first round, or vice versa, as the different rounds have different sets of registers. The same holds true for larger input complexes, because which specific input values we started with does not matter, assignments and values from outside this simplex will not simply appear during the execution. As such, the complex in Figure 2.5(b) on page 21 can be considered as the protocol complex of both, an input complex $\mathcal{I} = \sigma^2$ with $K = 2$ or an input complex $\mathcal{I} = \chi(\sigma^2)$ with $K = 1$. Thus the protocol complex after each processes executed K rounds of IS is the K-th standard chromatic subdivision of $\mathcal{I}$, so we have the protocol complex $\mathcal{P} = \chi^K(\mathcal{I})$.

Along with the protocol complex, we have the protocol carrier map $\Theta = \chi^K$ that gives us all possible configurations after an execution of the protocol for a given input configuration. Θ is strict ($\Theta(\sigma) \cap \Theta(\sigma') = \Theta(\sigma \cap \sigma')$), because the executions/simplices that are contained in both $\Theta(\sigma)$ and $\Theta(\sigma')$ are those where all processes of $\sigma \cap \sigma'$ ran without knowledge of the other processes and thus could not distinguish between the input configurations σ and σ' or even only $\sigma \cap \sigma'$.

The final decision of an output value is modeled by the *decision map*, a chromatic simplicial map $\delta\colon \mathcal{P} \to \mathcal{O}$. Obviously, such a map does not always exist, otherwise we would not need to bother about solvability of tasks. To be exact, a protocol $\Theta\colon \mathcal{I} \to \mathcal{P}$ *solves a task* $\Delta : \mathcal{I} \to \mathcal{O}$ if there is a decision map δ such that $\delta \circ \Theta$ is carried by Δ. Roughly speaking, a task is solved by a protocol if at the end of every possible protocol execution, every process can choose an output such that the configuration of output values is allowed by the task for the input values supplied. DECISION MAP

Note that this model is as good as (almost) any other, as proven [BG93]. For other models, the protocol complex $\mathcal{P}$ and protocol map Θ could look different, but wouldn't yield other solvability results.

2.2.4 Computability

This combinatorial setup enables us to characterize, when a protocol exists for a task. This was originally done by [HS99] in a very handy theorem:

Theorem 2.8 (Anonymous computability theorem [HS99, Thm. 6.3])**.** *A symmetric decision task $(\mathcal{I}, \mathcal{O}, \Delta)$ has a wait-free anonymous protocol using read-write memory if and only if there exists an integer K and a color-preserving simplicial map*

$$\delta\colon \chi^K(\mathcal{I}) \to \mathcal{O}$$

symmetric under permutation, such that δ is carried by $\Delta \circ \mathrm{Car}(*, \chi^K)$.

Here χ^K is the K-th standard chromatic subdivision. In the same paper, we get another helpful property, that enables us to do other chromatic subdivisions as well, because they are "contained" in the standard chromatic subdivision if we only apply it often enough:

Theorem 2.9 ([HS99, Thm. 5.29])**.** *If $\mathcal{B}$ is a chromatic subdivision of a complex $\mathcal{A}$, then there exists $K \geqslant 0$ and a chromatic and carrier-preserving simplicial map $\chi^K(\mathcal{A}) \to \mathcal{B}$.*

Note that while back then they proved the theorem for anonymous protocols, which must not depend on the process ID at all, it easily transferred to rank-symmetric protocols as well by requiring that δ is not symmetric, but rank-symmetric, i.e. symmetric on the faces of $\mathcal{I}$ under rank-preserving permutations of subsets of $\mathcal{I}$. The equivalence comes from the fact that instead of restricting the use of the process ID to comparisons, one could also understand them as inputs from an infinite set, making $\mathcal{I}$ contain infinitely many copies of more or less the same input complex with different process ID assignments. If you know how to subdivide a single input subcomplex rank-symmetrically, you can subdivide the whole infinite input complex.

Considering this small change, we can combine these two theorems to the following by additionally using the equivalences of the IS with the read-write memory model along with our accustomed notation:

Corollary 2.10. *A rank-symmetric decision task* $(\mathcal{I}, \mathcal{O}, \Delta)$ *has a wait-free rank-symmetric protocol using IS if there exists a chromatic subdivision* $\mathcal{P} := \Psi(\mathcal{I})$ *of* $\mathcal{I}$ *and a decision map* $\delta \colon \mathcal{P} \to \mathcal{O}$ *such that* $\delta \circ \Psi$ *is carried by* Δ*.*

3 Weak Symmetry Breaking

We will now consider the application of the theory from Chapter 2 to find a protocol that solves the task of *Weak Symmetry Breaking (WSB)*:

> Each of $n+1$ processes is assigned a unique process number and has to decide on a boolean output value just by comparing its value with the others, such that if all processes participate, each value is output by at least one processes.

In our terms, this means that the input complex $\mathcal{I}$ consists of a single n-simplex along with all its faces, because there is no input except the process numbers, which are handled implicitly by the rank-symmetry. For simplicity, we will assume the processes are named with $\Pi = [n]$.

As we want to output a boolean value, we have $V^{out} = [1]$ and the output complex $\mathcal{O}$ has a ground set of $V(\mathcal{O}) = [1]^{\Pi} = [1]^{[n]}$. A subset $\sigma \subseteq V(\mathcal{O})$ with $|\operatorname{pr}_{\Pi}(\sigma)| = |\sigma|$ is a simplex if $\dim \sigma < n$ or if it is not monochromatic, that is $\operatorname{pr}_{V^{out}} \sigma = V^{out} = [1]$. An example of dimension two was depicted in the previous chapter in Figure 2.4(b) on page 19.

Finally, the task's carrier map $\Delta\colon \mathcal{I} \to \mathcal{O}$ is the maximal name-preserving map. It is therefore not really of interest, as there is only one possible input configuration, and as long as it is contained in $\mathcal{O}$, every output configuration is allowed for this single input configuration.

Theorem 3.1 ([CR12, Thm. 7.2]). *If $n+1$ is not a prime power, then there exists a rank-symmetric wait-free WSB protocol for $n+1$ processes.**

The rest of this thesis will be dedicated to showing and comparing some ways of proving this theorem using the Anonymous Computability Theorem (or, more precisely,

*Note that the converse is true as well, as proven in [CR10], but not of interest to this thesis.

Corollary 2.10 on page 24) by constructing a protocol complex $\mathcal{P}$ as a subdivision of $\mathcal{I}$. We do not need to worry about the decision map $\delta\colon \mathcal{P} \to \mathcal{O}$ being carried by the task's carrier map Δ because Δ is maximal. As the essential information in δ is whether a process p_i decides 0 or 1, we understand $B = \mathrm{pr}_{V^{out}} \circ \delta$ as a labeling of $\mathcal{P}$. Obviously, this means that no simplex $\sigma \in \mathcal{P}$ can be B-monochromatic in the sense that $|B(\sigma)| = 1$.

3.1 Subdivision

On our way to $\mathcal{P}$, we will subdivide $\mathcal{I}$ in various ways, always associating boolean values via a map called B. This will be done in two steps: The first generates an oriented subdivision in which all B-monochromatic n-simplices with same B-coloring can be paired and have different orientation. The second one then resolves these monochromaticities by finding a path of even length between two monochromatic simplices of different orientation and modifies it to contain no monochromatic simplices at all. While the second step can be done in different ways, the first step as well as the whole setup is taken from [CR12, Sect. 5].

CONTENT

Let $\mathcal{L}^n$ be a chromatic pseudomanifold with a binary labeling $B\colon V(\mathcal{L}^n) \to [1]$, coherently oriented by $D\colon \mathcal{L}^n \to [1]$. The *content* of $\mathcal{L}^n$, $\mathcal{C}(\mathcal{L}^n)$, is the sum of its monochromatic n-simplices counted by orientation:

$$\mathcal{C}(\mathcal{L}^n) = \sum_{\substack{\sigma \in \mathcal{L}^n \\ \dim(\sigma)=n \\ |B(\sigma)|=1}} (-1)^{B(\sigma)\cdot n} D(\sigma)$$

Our subdivision will be constructed in a way that for integers $k_0, k_1, \ldots, k_{n-1}$ with $k_0 \in \{0, -1\}$ it has a content of exactly

$$\mathcal{C} = 1 + \sum_{i=0}^{n-1} \binom{n+1}{i+1} k_i. \tag{3.1}$$

In order for the second step to resolve all monochromaticities, we have to have equally many positively and negatively oriented B-monochromatic n-simplices, so we want to have $\mathcal{C} = 0$. Number theory tells us, that (without our restriction on k_0) this is exactly the case if $\binom{n+1}{1}, \binom{n+1}{2}, \ldots, \binom{n+1}{n}$ are relatively prime, i.e. they have a greatest common divisor of 1 [DD99]. This in turn is given if $n+1$ is not a prime power [Dic19, p. 274]. Finally, our restriction on k_0 does not interfere with the solvability, because by

$\binom{n+1}{1} = \binom{n+1}{n}$ only the sum $k_0 + k_{n-1}$ needs to be fixed. Thus, our construction only works if $n+1$ is not a prime power.

We construct the subdivision Ψ inductively by subdividing and B-labeling the faces of $\mathcal{I}$ by dimension. The goal is to have exactly $|k_i|$ 0-monochromatic i-simplices in the subdivision of every i-face of $\mathcal{I}$, such that a final coning over $\Psi(\partial\mathcal{I})$ will generate a 0-monochromatic n-simplex for each of them, oriented according to $\operatorname{sgn}(k_i)$.

We start with the 0-faces, which are obviously copied into the subdivision without changes, and set their B-labeling to $1 + k_0$. Given the subdivision Ψ up to dimension $i-1$, an i-face σ^i of $\mathcal{I}$ is subdivided as follows: Let τ^i be a 1-monochromatic i-simplex, then build the cone over $\Psi(\partial\sigma^i)$ for τ^i and orient it coherently in a way that at least one* i-corner has the orientation $(-1)^{i+1}$. Up to now, this subdivision of σ^i has no 0-monochromatic i-simplex, because every i-simplex contains at least one of the vertices of τ^i, which is 1-monochromatic. So we subdivide it such that exactly $|k_i|$ 0-monochromatic n-simplices of orientation $\operatorname{sgn}(k_i)$ are created. While there might be more clever ways to do so, Lemma 2.7 on page 15 gives us an easy one, by subdividing existing non-monochromatic simplices of appropriate orientation twice. Another, more global way with an upper bound on the number of subdivisions needed is described in [ACHP13, Sect. 4.1]. Finally, copy this subdivision to all other i-faces of $\mathcal{I}$, preserving the rank of process IDs.

Figure 3.1 on the following page illustrates the operations that take place. That example uses $n = 5$ and solves the diophantine equation 3.1 by $k_0 = -1$, $k_1 = 1$, $k_2 = 1$, $k_3 = -2$ and $k_4 = 0$. It thus has its outer vertices labeled $1 - k_0 = 0$, has $|k_1| = 1$ positively oriented 1-simplex in every 1-face, and $|k_2| = 1$ positively oriented 2-simplex. Note that the subdivision is rank-symmetric on the boundary. It could thus be copied to all other 2-faces of the 5-simplex that it is itself a 2-face of (not depicted) while preserving the order of the outer vertices.

After the subdivision up to dimension $n-1$ has been constructed, simply cone over $\Psi(\partial\mathcal{I})$ for a 0-monochromatic n-simplex τ^n and orient the whole complex such that τ^n is oriented positively. Apart from τ^n itself, this introduces one 0-monochromatic n-simplex for every 0-monochromatic i-simplex in each of the $\Psi(\sigma^i)$, and because every n-simplex contains at least one vertex from τ^n, no 1-monochromatic n-simplices can occur. We now only have to check the orientations of the 0-monochromatic n-simplices.

*Actually, all i-corners have the same orientation [CR08, Lem. 2.4], but we only need to have one oriented this way.

Therefor, remember that in every dimension i, in subdivision of σ^i at least one i-corner was oriented as $(-1)^{i+1}$. Let ρ^i be this i-corner. By its definition, there is a sequence $\rho^0 \subsetneq \rho^1 \subsetneq \cdots \subsetneq \rho^i$ of faces of ρ^i such that for every $j \in [i]$, $\rho^j \subseteq \Psi(\sigma^j)$ where σ^j is a j-face of σ^i and thus also of $\mathcal{I} = \sigma^n$. Now each of these faces ρ^j generates an n-simplex $\tau_j * \rho^j$ in $\tau^n \circledast \Psi(\partial \sigma^{n-1})$ where τ_j is an $(n-j-1)$-face of τ^n, and each consecutive pair of faces ρ^j, ρ^{j+1} generates a pair of n-simplices $\tau_j * \rho^j, \tau_{j+1} * \rho^{j+1}$ that shares an $(n-1)$-face, namely $\tau_{j+1} * \rho^j$. Because τ_0 is itself an $(n-1)$-face of τ^n, $\tau_0 * \rho^0$ and τ^n share an $(n-1)$-face as well, yielding an n-simplex path from τ^n to $\tau_i * \rho^i$ of length $i+2$. Because τ^n is oriented positively and an even length n-simplex path means that the end simplices are oriented contrarily, this makes $\tau_i * \rho^i$ have the same orientation as ρ^i had in $\Psi(\sigma^i)$, i.e. $(-1)^{i+1}$.

Now it is fairly obvious that every other i-simplex in $\Psi(\sigma^i)$ has the same orientation as its n-dimensional counterpart, because every i-simplex path inside $\Psi(\sigma^i)$ induces an n-simplex path inside $\Psi(\sigma^n)$ of the same length, in particular those connecting the 0-monochromatic simplices in $\Psi(\sigma^i)$ to their respective i-corner.

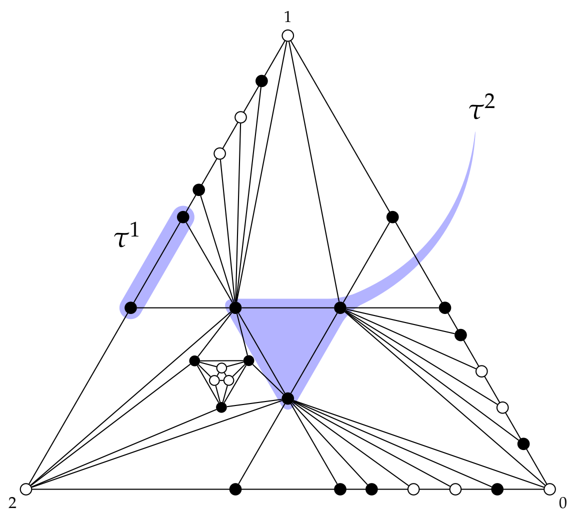


Figure 3.1: The rank-symmetric subdivision of a 2-face of σ^5 corresponding to the choice $k_0 = -1, k_1 = 1, k_2 = 1$ (and $k_3 = -2, k_4 = 0$, which do not yet have an effect in this dimension).

3.2 Simplex Paths

Having subdivided the input complex $\mathcal{I}$, we now have multiple 0-monochromatic n-simplices, one half negatively and one half positively oriented. This enables us to iteratively pick two 0-monochromatic simplices of opposite orientation and find an n-simplex path between them. This path will be of even length because the end simplices have opposite orientations and passing from one simplex to the next of the path flips the orientation each time. If we furthermore require that the path generally does not contain any other monochromatic simplices, it can then be modified such that it contains no more B-monochromatic simplices of any kind. How exactly this modification is done will be handled in the next chapter.

If we are interested in minimizing the number of protocol layers or, equivalently, the number $K \geqslant 0$ of iterated standard chromatic subdivisions applied to the input complex that is guaranteed to exist by Theorem 2.9 on page 23, the application of whatever algorithm we choose to all the paths would be the most costly step [ACHP13, Sect. 5 f.]. That is because finding non-intersecting (or "disjoint") simplex paths in the current complex means finding disjoint paths in a graph that does not have very predictable and nice properties, which is a known problem that is hard to solve [Kar75].

But intersecting paths cannot be processed independently, because different subdivisions might be applied to the simplices of their intersection by the two parallel invocations of the following algorithms. So to avoid having to find disjoint paths, we must find and demonochromatize one path after the other, at worst causing one simplex to be subdivided again and again.

4 Simplex Path Demonochromatizing

We will now describe methods of subdividing a simplex path without changing its boundary (in order to preserve the rank-symmetry of the subdivision from Chapter 3) such that they do not contain any monochromatic simplices.

In contrast to the previous step in Chapter 3, which required that our dimension n is chosen such that $n+1$ is not a prime power, this problem can be considered and solved in any dimension $n \geqslant 2$.

Simplex paths, just as simplicial complexes, can be thought of in at least two ways: Geometrically and combinatorially. While for subdivisions we typically think about the geometric version, proving that we actually eliminate monochromaticities is easier in the combinatoric model.

GEOMETRIC SIMPLEX PATH

Given an arbitrary simplicial complex K, a *geometric simplex path in K of length ℓ* is an ℓ-tuple $\Sigma = (\sigma_1, \ldots, \sigma_\ell)$ of n-simplices of K such that $\sigma_{i,i+1} := \sigma_i \cap \sigma_{i+1}$ is an $n-1$-simplex of K for each $i \in \{1, 2, \ldots, \ell - 1\}$. We will sometimes also call the subcomplex of K containing only the simplices of Σ along with its faces a geometric simplex path as well.

ATOMIC

When dealing with a simplicial complex K binarily-labeled by B, a geometric simplex path in *standard form* (also called *atomic*) is of even length and has only two B-monochromatic simplices σ_1 and σ_ℓ, which are 0-monochromatic.

ABSTRACT SIMPLEX PATH

An n-dimensional *(abstract) simplex path* $P = (I, C, V)$ of length ℓ is a triple of

- an $[n]$-tuple $I \in [1]^{n+1}$, describing the binary labeling of the „first" n-simplex,
- an $(\ell-1)$-tuple $C \in [n]^{\ell-1}$, specifying a series of vertices C_i to serially „flip" over their opposite faces, and
- an $(\ell-1)$-tuple $V \in [1]^{\ell-1}$, giving the binary labeling of the vertex created by each flip in C.

The only requirement we impose on these tuples is that C satisfies a „no back-flip" condition: $C_i \neq C_{i+1}$ for all $i \in \{1, 2, \ldots, \ell - 2\}$.

For convenience, we will define

$$e_i^j := \begin{cases} I_i & \text{if } \operatorname{last}(i, C(1, j-1)) = \infty \\ V_{\operatorname{last}(i, C(1, j-1))} & \text{else} \end{cases}$$

where $\operatorname{last}(i, C(i, j-1)) := \max\{k \in \{1, \ldots, j-1\} \mid V_k = i\}$. This way, e_i^j gives the binary label of the vertex named i after j flips. We will call $R^j := \left(e_0^j, \ldots, e_n^j\right)$ the *j-th simplex of the path P*, which is b-monochromatic if $e_i^j = b$ for all $i \in \{0, 1, \ldots, n\}$. Note that $R^1 = I$.

We will care about *atomic* simplex paths, meaning that ℓ is even and the only monochromatic simplices are R^1 and R^ℓ, which are 0-monochromatic. In this case, we can shorten the definition to $P = (C, V)$ as $I = (0, \ldots, 0)$ is mandatory.

Quite obviously, any binarily-labeled geometric simplex path can be converted into a unique abstract simplex path: Consider a geometric simplex path $\Sigma = (\sigma_1, \ldots, \sigma_\ell)$ colored with $[n]$ by Π and binarily labeled by B. We define its *associated abstract simplex path* $P = (I, C, V)$ by setting

ASSOCIATED ABSTRACT SIMPLEX PATH

$I_i := B(v_i)$ for $i \in [n]$, where v_i is the unique vertex of σ_1 with $\Pi(v_i) = i$,
$C_j := \Pi(\sigma_{j+1} \setminus \sigma_j) = \Pi(\sigma_j \setminus \sigma_{j+1})$ for $j \in \{1, 2, \ldots, \ell - 1\}$, and
$V_j := B(\sigma_{j+1} \setminus \sigma_j)$ for $j \in \{1, 2, \ldots, \ell - 1\}$.

While the inverse transformation is possible as well, it is not unique because some vertices actually may or may not be the same in the geometric version (cf. Figure 4.1 on the following page). Thus, when we define a subdivision we will first define it on the geometric version, and then analyze its effect on and further work with the abstract version.

What follows are two different constructive proofs of the following theorem:

Theorem 4.1. *Given an atomic geometric simplex path Σ labeled binarily by B and colored by Π, there is a Π-chromatic subdivision $\Psi(\Sigma)$ such that $\partial\Psi(\Sigma) = \partial\Sigma$ and $\Psi(\Sigma)$ contains no B-monochromatic simplices.*

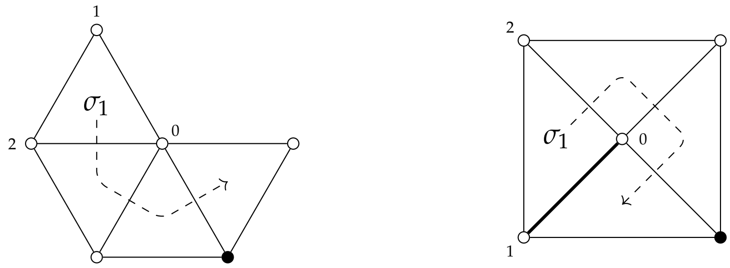


Figure 4.1: Two 2-dimensional geometric simplex paths of length 4 with binary labeling represented using the colors black and white, sharing the same associated abstract simplex path $I = (0, 0, 0)$, $C = (1, 2, 1)$ and $V = (0, 1, 0)$.

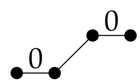


Figure 4.2: Height graph of the paths in Figure 4.1. If nothing else is denoted, the lower left vertex is always at position $(1, 0)$.

HEIGHT GRAPH

The workings and effectiveness of the algorithms will be illustrated using what is called the *height graph* of an abstract simplex path $P = (I, C, V)$, a partially marked graph embedded in $\mathbb{N}_0^2$ with the vertices $\{(i, h_i) \in \mathbb{N}_0^2 \,|\, i \in [\ell] - 0\}$ where $h_i := \#(1, B(\sigma_i))$ is the number of vertices of σ_i labeled with 1 by B. Edges exist between each (i, h_i) and $(i + 1, h_{i+1})$ for $i = 1, \ldots, \ell - 1$. If we have $h_i = h_{i+1}$, we furthermore mark the edge between (i, h_i) and $(i + 1, h_{i+1})$ with V_i. When the heights differ and the edge has a slope, we know that $V_i = 1$ if the slope is rising, and $V_i = 0$ if it is falling, thus we do not need to label all edges. See Figure 4.2 for an example.

For consistency, we will also use the shortcut $h_{i,i+1} := \#(1, B(\sigma_{i,i+1}))$ for the connecting $(n - 1)$-faces. If we need to denote, which path the height of a simplex or face is associated to, we will write $h_i(P)$ for the height at position i of path P.

4.1 Algorithm by Castañeda and Rajsbaum [CR12]

The earlier method of the two that we will present originated from [CR12] and was heavily modified and shortened in [ACHP13]. We will follow the modified version for its smaller case differentiation, but deviate and fall back to the original version from [CR12] in cases where we deem the modified version insufficient. Note though, that our naming conventions differ from both. Most importantly, our first simplex is called

σ_1 instead of σ_0, and we will use the number of vertices labeled with 1 instead of those labeled with 0.

The algorithm will use only one kind of subdivision: Assume an n-dimensional geometric simplex path Σ of length $\ell \geqslant 4$, labeled with $[1]$ by B and colored with $[n]$ by Π, along with its associated abstract simplex path $P = (I, C, V)$, and a quadruple (m, D, Q, s) such that

- $2 \leqslant m \leqslant \ell - 2$, specifying at which simplex boundary the expansion takes place,
- D is an $[n]$-tuple $(d_0, \dots, d_{C_m - 1}, -, d_{C_m + 1} \dots, d_n)$ of boolean values, giving the binary labeling of the newly created vertices,
- Q is a t-tuple of values from $\Pi - C_m = [n] - C_m$ that defines how the new path routes through the subdivision, such that
  - $q_1 = C_{m-1}$ and $q_t = C_{m+1}$,
  - for every $l \in [n] - C_m$ we have that $\#(l, Q)$ is even,
  - for all $1 \leqslant i < j \leqslant s$ and all $s + 1 \leqslant i < j \leqslant t$, there exists $l \in [n] - C_m$ such that $\#(l, Q[i, j])$ is odd, and finally
- $1 \leqslant s < t - 1$ defines when we flip over $\sigma_{m,m+1}$ and thus splits Q into a part that happens in the first and a part that happens in the second original simplex.

EDGE EXPANSION

Then the *edge expansion* (m, D, Q, s) of P is constructed as follows: Do the basic chromatic subdivision of $\sigma_{m,m+1}$ and call it S. Then extend B by setting $B(v_i) = d_i$ where v_i is the unique vertex of the interior of S with $\Pi(v) = i$. Afterwards, set v and v' to be the unique vertices of $\sigma_m \setminus \sigma_{m+1}$ and $\sigma_{m+1} \setminus \sigma_m$, respectively, and build $S \circledast \{v, v'\}$. Replace the old σ_m and σ_{m+1} by this cone. Note that neither the other simplices nor the boundary of Σ are changed by this subdivision.

Using the presentation from Proposition 2.5 on page 12, every $(n-1)$-simplex of the subdivision of σ_m can be described as $v * (o_0, o_1, \dots, o_{C_m - 1}, -, o_{C_m + 1}, \dots, o_n)$ or directly as $(o_0, o_1, \dots, o_{C_m - 1}, 0, o_{C_m + 1}, \dots, o_n)$ where $o_i \in [1]$. The same holds true for σ_{m+1} by coning with v' or by putting a 1 at the C_m-th position.

We define U to be the t-tuple of binary $[n]$-tuples with $u_1 = (\delta_{j, C_{m-1}})_{j=0}^{n}$ and the transition between u_{i-1} and u_i being exactly a flip the q_i-th component. This gives us the final geometric path

$$\tilde{\Sigma} := \big(\underbrace{\sigma_1, \dots, \sigma_{m-1},}_{\text{before } \sigma_m} \underbrace{v * u_1, \dots, v * u_s,}_{\text{in } \sigma_m} \underbrace{v' * u_s, \dots, v' * u_{t-1},}_{\text{in } \sigma_{m+1}} \underbrace{\sigma_{m+2}, \dots, \sigma_\ell}_{\text{after } \sigma_{m+1}} \big) .$$

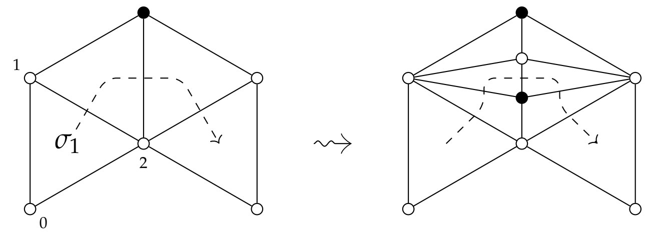


Figure 4.3: An edge expansion with input data $m = 2$, $D = (1, -, 0)$, $Q = (0, 2, 2, 0)$ and $s = 2$ is applied to a geometric simplex path Σ of length $\ell = 2$ whose associated abstract simplex path is $I = (0, 0, 0)$, $C = (0, 1, 0)$, and $V = (1, 0, 0)$.

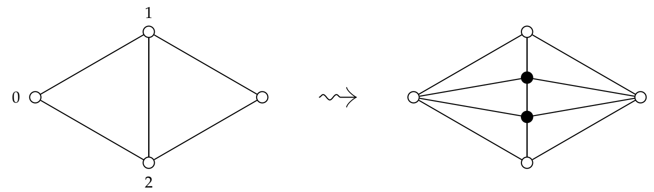


Figure 4.4: Special case of applying edge expansion to a simplex path of length $\ell = 2$.

Note that u_s appears twice: The simplices right before and after flipping over $\sigma_{m,m+1}$ share the same $(n-1)$-face in the basic chromatic subdivision of $\sigma_{m,m+1}$.

Concerning the associated abstract simplex path, we have the following changes:

$$\begin{aligned}\tilde{I} &= I,\\ \tilde{C} &= (C_1, \ldots, C_{m-1}, q_2, \ldots, q_s, C_m, q_{s+1}, \ldots, q_{t-1}, C_{m+1}, \ldots, C_\ell),\\ \tilde{V} &= (V_1, \ldots, V_{m-2}, w_1, \ldots, w_s, V_m, w_{s+1}, \ldots, w_{t-1}, V_{m+1}, \ldots, V_\ell)\end{aligned}$$

where

$$w_i := \begin{cases} e^i_{q_i} & \text{if } \#(q_i, (q_1, \ldots, q_i)) \text{ is even,} \\ d_{q_i} & \text{if } \#(q_i, (q_1, \ldots, q_i)) \text{ is odd.} \end{cases}$$

For an example see Figure 4.3, where an edge expansion is applied to a two-dimensional geometric simplex path of length four, turning it into a path of length six.

Trivial path Before the regular algorithm begins, let us handle a special case to know where we are heading: The trivial atomic path of length $\ell = 2$. No matter the dimension n, all we have to do to eliminate the two monochromatic n-simplices is do an edge expansion with $m = 0$, $D = (-, 1, \ldots, 1)$ (assuming $C_1 = 0$, otherwise put the dash

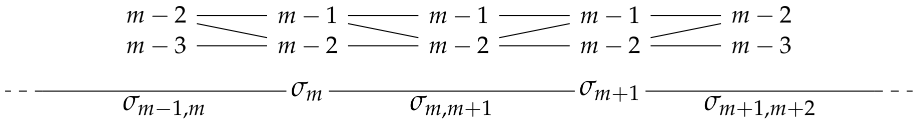


Figure 4.5: The restrictions on the number of 1-vertices (heights) of the few n-vertices and connecting faces preceding the subdivision point. The numbers at the top are possible heights and are connected with each other if that particular combination is possible. At the bottom, the simplex path is indicated by its dual complex.

at the appropriate position C_m) and any Q and s. Even though the definition of edge expansion requires $\ell \geqslant 4$ and $m \geqslant 2$, we can do the same for these values, except the embedding into the earlier and later parts of the path ceases to apply. Obviously, the subdivision contains no monochromatic n-simplices of any kind, as every n-simplex contains at least one vertex from the subdivision's interior (all labeled 1) and one from $\sigma_1 \triangle \sigma_2$ (all labeled 0). See Figure 4.4 on page 34 for an example in dimension 2.

Subdividing point In any other atomic geometric simplex path S with associated abstract path P, we first have to find a *subdividing point m*, defined as the minimal m such that

SUBDIVIDING POINT

$$\#(0, B(\sigma_{m+1,m+2})) \geqslant n + 2 - m \quad \iff \quad \#(1, B(\sigma_{m+1,m+2})) = h_{m+1,m+2} \leqslant m - 2.$$

Note that if such an m exists, because m is minimal this implies that $h_{m,m+1} \geqslant m - 2$. By considering that the two faces $\sigma_{m,m+1}$ and $\sigma_{m+1,m+2}$ share $n - 1$ vertices and thus each only has one vertex the other does not, it is obvious that

$$\begin{aligned} 1 \geqslant \overbrace{h_{m+1,m+2}}^{\leqslant m-2} - \overbrace{h_{m,m+1}}^{\geqslant m-2} &\geqslant -1 \\ \implies m - 2 \geqslant h_{m+1,m+2} \geqslant h_{m,m+1} - 1 &\geqslant m - 3. \end{aligned} \tag{4.1}$$

This gives us bounds on the height of quite some n-simplices and their faces. They are summed up in Figure 4.5, which can be constructed from right to left using Equation (4.1) and the facts that *a)* an $(n-1)$-face can only have at most one 1-vertex less but never more than its n-dimensional co-faces, and *b)* any σ_i can have at most $i - 1$ vertices labeled with 1, because from the starting simplex σ_1 that was 0-monochromatic, each step introduces at most one new 1-vertex.

The subdivision point m is guaranteed to exist, as the number of vertices labeled with 1 is bounded not only by $n+1$, as no n-simplex must be 1-monochromatic, but also by $\frac{\ell}{2}$: We already established that $h_i \leqslant i-1$. By symmetry, at least h_i flips have to follow the vertex σ_i to reach the 0-monochromatic end-simplex σ_ℓ, so $i + h_i \leqslant \ell$. Thus, for $i = m+2$,

$$\ell \geqslant m + 2 + h_{m+2} \geqslant m + 2 + h_{m+1,m+2} \geqslant m + 2 + m - 3 = 2m - 1.$$

Considering that ℓ is even and m is an integer, $m \leqslant \frac{\ell+1}{2}$ also implies $m \leqslant \frac{\ell}{2}$.

4.1.1 Case 1: Asymmetric*

If $h_m \neq h_{m+1}$, we apply an edge expansion at position m and call the resulting subdivision Ψ. Our aim is to produce two 0-monochromatic n-simplices in $\Psi(\sigma_m \cup \sigma_{m+1})$ so that we can connect one to each end of the original simplex path in order to receive two shorter paths. Because $h_m \neq h_{m+1}$, exactly one of the two vertices in $\sigma_m \triangle \sigma_{m+1}$ is labeled with zero, the other with one (hence the name "asymmetric"), thus every monochromatic $(n-1)$-simplex in $\Psi(\sigma_{m,m+1})$ induces exactly one monochromatic n-simplex in $\Psi(\sigma_m \cup \sigma_{m+1})$. In Figure 4.5 on page 35 we can see that $h_m \neq h_{m+1}$ implies $h_{m,m+1} = m - 2$. Let $\Pi_0 = \{i_1, \dots, i_{n-(m-2)}\}$ be the names assigned to the 0-vertices of $\sigma_{m,m+1}$. We then color the subdivision using $d_j = 1$ if $j \in \Pi_0 \setminus \{i_1\}$ and $d_j = 0$ otherwise. This will generate exactly two 0-monochromatic $(n-1)$-simplices in $\Psi(\sigma_{m,m+1})$, which is best explained in the language of Proposition 2.5 on page 12: Their corresponding boolean tuples have zeros at places $i_2, \dots, i_{n-(m-2)}$, units at all positions not in Π_0 and either a zero or unit in position i_1. These two $(n-1)$-simplices share an $(n-2)$-face, which means that their corresponding 0-monochromatic n-simplices in $\Psi(\sigma_m \cup \sigma_{m+1})$ share an $(n-1)$-face and are thus oppositely oriented. We can therefore construct shortest paths of even length from each end to one of the two newly created monochromatic simplices and call them P_1 and P_2. Note that we do not generate any 1-monochromatic n-simplices, because all vertices named i_1 in the subdivision are labeled with 0.

*In [ACHP13], case A lists two conditions that cannot be fulfilled simultaneously, while case B treats one case that could have been addressed more easily in the same way as an earlier one. Thus, our cases 1 and part of 2 correspond to their case A, whereas the other part of our case 2 together with our case 3 correspond to their case B. Compared with [CR12], our case 1 maps to their cases C and D, our case 2 to their cases A and E, and our case 3 to their case B.

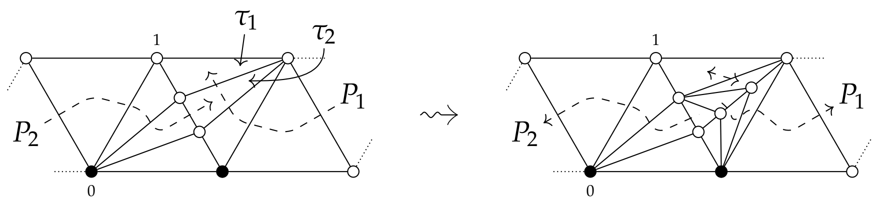


Figure 4.6: Example of resolving an intersection of P_1 with P_2 in the case 1 by subdividing a face of τ_2 in two dimensions, where $m = 3$.

Intersection resolution Both new 0-monochromatic n-simplices lie "on the same side" of $\sigma_{m,m+1}$, i.e. they share the same vertex from $\sigma_m \triangle \sigma_{m+1}$. So obviously, one of the two paths P_1 and P_2 crosses $\Psi(\sigma_{m,m+1})$, meaning that there is an $(n-1)$-simplex $\sigma \in \Psi(\sigma_{m,m+1})$ whose two cofaces are contained in the path, while the other does not. We will call the non-crossing path P_1, the crossing one P_2 and their end-simplices τ_1 and τ_2, respectively. Note that P_2 crosses $\Psi(\sigma_{m,m+1})$—without loss of generality—right before ending in its newly created 0-monochromatic n-simplex τ_2 (cf. Figure 4.6). Now we cannot guarantee that P_1 and P_2 are intersection-free, which, if they are not, would prevent invoking the whole algorithm recursively on them while still making sure it terminates. If they intersect, however, we can assume by our construction that they intersect in τ_2, and due to the minimal length of the paths we know that τ_2 is then not only the last simplex of P_2, but also the second last simplex of P_1 as it is neighboring to τ_1. Furthermore, we can be sure that the unique face of τ_2 that is crossed by P_2 is not crossed by P_1, because it is a face of the subdivision of $\sigma_{m,m+1}$ which is only crossed by P_1 as we already established. Vice versa, the two faces of τ_2 that are crossed by P_1 are not crossed by P_2.

For easier indices, we reverse the paths P_1 and P_2 and let $(\tilde{C}, \tilde{V})$ be the abstract and $(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_{\tilde{\ell}})$ the geometric form of P_1. Then we apply an edge expansion to the face $\tilde{\sigma}_{2,3}$ and label it with $d_i = 1 - \delta_{i,C_m} - \delta_{i,\tilde{C}_1}$, i.e. label every vertex with 1 except for those colored with C_m and $\tilde{C}_1$.

An important aspect for all our algorithmic work is that we do not really care about the actual names of the vertices. We cared when first subdividing in Chapter 3 in order to be rank-symmetric, but we no longer do now, because we do not modify the boundary anyway. Thus for every step we do, we can reassign the names at our discretion by applying some permutation on $[n]$, thereby making it much easier to talk about the simplices of a subdivision in terms of the boolean tuples from Proposition 2.5 on page 12.

We will thus assume for now that $C_m = 0$ and $\tilde{C}_1 = 1$. Note that these are two different vertices, because otherwise P_1 and P_2 would share their first two simplices. Then the labeling can be written more easily as $D = (0, 0, 1, \ldots, 1)$. As can be seen in Figure 4.6 on page 37, this labeling enables us to let P_2 end in the 0-monochromatic simplex $(1, 0, \ldots, 0)$ whose only inner vertex is named $C_m = 0$. Note that this modified version of P_2 has the same length as P_2 had originally. It furthermore generates a trivial atomic path starting at τ_1 and ending at the simplex $(0, 1, 0, \ldots, 0)$ after a single flip $\tilde{C}_1 = 1$.

We now need to deal with the rest of P_1. Assume by $S_{[n]}$-action that $\tilde{C}_2 = 2$, as it can be neither $\tilde{C}_1 = 1$ by the no-backflip condition, nor $C_m = 0$ because P_1 does not cross $\Psi(\sigma_{m,m+1})$. If $\tilde{C}_3 \in \{\tilde{C}_1, C_m\} = \{0, 1\}$ like in our figure, we are in luck and simply connect the 0-monochromatic simplex $(1, 1, 0, \ldots, 0)$ to the original end of P_1 by the path

$$\tilde{C}' := (2, \{0, 1\} - \tilde{C}_3, \tilde{C}_3, \tilde{C}_4, \ldots, \tilde{C}_{\tilde{\ell}}).$$

Note that flipping $\tilde{C}_2 = 2$ brings us on the other side of $\tilde{\sigma}_{2,3}$ where nothing is 0-monochromatic anymore, hence this new path is atomic and of the same length as P_1 has originally been. Note additionally that this case always applies in two dimensions.

If in higher dimensions $\tilde{C}_3 \notin \{\tilde{C}_1, C_m\} = \{0, 1\}$, we assume by $S_{[n]}$-action that $\tilde{C}_3 = 3$. Starting at $(1, 1, 0, \ldots, 0)$ might now yield a too long path, because $\tilde{C}_3 = 3$ would have to be flipped twice. Thus, we also set $d_3 = 0$ in the edge expansion and then construct the new path by starting at the simplex $(0, 1, 0, 1, 0, \ldots, 0)$ and work our way to the original end of P_1 by setting

$$\tilde{C}' := (2, 1, 3, \tilde{C}_4, \ldots, \tilde{C}_{\tilde{\ell}}).$$

The path is atomic again, however, we now also have the 0-monochromatic simplices

$$\alpha_1 = (1, 1, 0, 1, 0, \ldots, 0), \qquad \alpha_3 = (1, 0, 0, 1, 0, \ldots, 0),$$
$$\alpha_2 = (1, 1, 0, 0, 0, \ldots, 0), \qquad \alpha_4 = (0, 0, 0, 1, 0, \ldots, 0).$$

Luckily, as can be seen from the boolean tuple representation, α_1 and α_2 share an $(n-1)$-face and thus form a trivial atomic path, as do α_3 and α_4.

Length calculation We will now have a look at the lengths of the paths P_1 and P_2, which both need to be smaller than ℓ in order for a recursion of this algorithm to finish at some point. As we have seen, their lengths do not change if we have to resolve an intersection, which we can thus ignore and only consider the first edge expansion we have done. The path starting at σ_1 contains $m-1$ simplices before it even enters our subdivision. The first simplex of the new path inside the subdivision (which is its m-th simplex in total) will be $\tilde{\sigma}_m = (0, \ldots, 0, 1, 0, \ldots, 0)$ in boolean tuple representation, where the unit is at position C_{m-1}. The two monochromatic simplices have at most m units in their representation: $\sigma_{m,m+1}$ contains $m-2$ vertices labeled with 1, which have to be exchanged for their inner counterparts. One more unit may be needed if the monochromatic simplices are located in σ_{m+1}, which means crossing $\sigma_{m,m+1}$. Another one may occur if the inner instead of the outer vertex named i_1 is chosen.

Now it is important to note that if the monochromatic simplices lie in σ_{m+1}, then $h_{m+1} < h_m$ which according to Figure 4.5 on page 35 implies that $h_m = m-1$. This means that $V_i = 1$ for all $1 \leqslant i \leqslant m-1$, in particular $V_{m-1} = 1$ and thus $e^m_{C_{m-1}} = 1$. Hence, the vertex named C_{m-1} of the both monochromatic simplices is the inner one. Summed up, this means that if we have a unit at position C_m in the boolean tuple representation of the monochromatic simplices, then we also have one at C_{m-1}. This makes $\tilde{\sigma}_m$ have at most $m-1$ positions that need to be flipped in order to reach any of the two monochromatic simplices, meaning that one is even as close as $m-2$ additional flips, resulting in a total length of at most $m+m-2 = 2m-2 \leqslant \ell - 2$ for the new path that starts at σ_1.

Symmetrically, the first simplex of the path starting at σ_ℓ that lies inside our subdivision is its $(\ell - m)$-th simplex, its representation as boolean tuple having exactly two units, this time at positions C_{m+1} and C_m. By symmetry, a similar argument as before holds true concerning the equivalence of having to flip over $\sigma_{m,m+1}$, and the outer vertex named C_{m+1} being labeled with 1. Thus, we again have a path of length at most $\ell - m + m - 2 = \ell - 2$ that starts at σ_ℓ.

4.1.2 Case 2: Symmetric 0

If $h_m = h_{m+1} = h_{m,m+1} \in \{m-1, m-2\}$ and thus $B(\sigma_m \triangle \sigma_{m+1}) = \{0\}$ (hence the name "Symmetric 0"), we only need to generate one 0-monochromatic $(n-1)$-simplex in the subdivision of $\sigma_{m,m+1}$ in order to have two 0-monochromatic n-simplices in $\Psi(\sigma_m \cup \sigma_{m+1})$, one on either side of $\sigma_{m,m+1}$. So again we do an edge expansion with $m = m$, and again, if $\Pi_0 = \{i_1, \ldots, i_{n-(m-1)}\}$ are the names of the 0-vertices in $\sigma_{m,m+1}$, we set $d_j = 1$ if $j \in \Pi_0$ and $d_j = 0$ otherwise. Note that i_1 is not treated in a special way this time. We end up with exactly one 0-monochromatic $(n-1)$-simplex in $\Psi(\sigma_{m,m+1})$ and

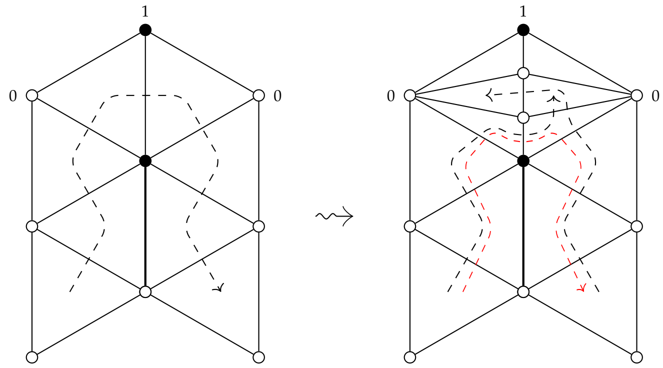


Figure 4.7: Problematic intersection in case 2 for the input path $C = (2, 0, 1, 0, 1, 0, 2)$, $V = (1, 0, 1, 0, 0, 0, 0)$: The black paths would normally intersect unresolvably in two simplices, which is why we instead choose the red path.

thus two neighboring 0-monochromatic n-simplices in $\Psi(\sigma_m \cup \sigma_{m+1})$, but no 1-monochromatic ones, since every n-simplex of the subdivision contains exactly one vertex from $\sigma_m \triangle \sigma_{m,m+1}$, which are both labeled with 0.

Length calculation When we now construct the shortest paths from σ_1 to each of the new monochromatic simplices, the same analysis as above shows that their lengths are $m - 1 + h_{m,m+1}$ and $m - 1 + h_{m,m+1} + 1 = m + h_{m,m+1}$ for the monochromatic simplex in σ_m and σ_{m+1}, respectively. Note that because of $h_{m,m+1} \leqslant h_m \leqslant m - 1$, both lengths are shorter than ℓ and thus useful for the recursion. But no matter the value of m, we see that the shorter P_1, the one that leads to the simplex inside σ_m, is of even length if and only if $h_{m,m+1} = m - 1$, whereas it is odd if $h_{m,m+1} = m - 2$. If $h_{m,m+1} = m - 1$ we are in luck, because no intersection of P_1 with P_2, the shortest path of even length from σ_ℓ to the other monochromatic simplex, can occur, as they both stay on "their side" of $\sigma_{m,m+1}$.

Intersection resolution However, if $h_{m,m+1} = m - 2$, intersections could occur because both paths have to cross the face $\sigma_{m,m+1}$. When $C_{m-1} \neq C_{m+1}$ we can either easily avoid any intersection by choosing the shortest path appropriately, or the occurring intersections can be resolved in the same way as in case 1. But in the event that $C_{m-1} = C_{m+1}$, an example of which can be seen in Figure 4.7, there is no easy way to construct an intersection-free path that is still shorter than ℓ. Thus in this very special case, we only construct one path $\tilde{P}$ that starts at σ_1, ends at σ_ℓ, and whose length is still exactly ℓ. Inside our subdivision, it immediately crosses the face $\sigma_{m,m+1}$, corresponding to the parameter $Q = (C_{m-1}, C_{m+1})$ of the edge expansion. If we assume $C_{m-1} = C_{m+1} = 1$ and $C_m = 0$ by $S_{[n]}$-action, the only two n-simplices of the subdivision contained in this new path, we will call them τ_1 and τ_2, correspond to the tuples $(0, 1, 0, \dots, 0)$ and $(1, 1, 0, \dots, 0)$. Thus, they each contain $n-1$ vertices of the original simplex $\sigma_{m,m+1}$. Then τ_1 and τ_2 have to be non-monochromatic, as we will see in a moment, thus $\tilde{P}$ is atomic. Both 0-monochromatic n-simplices of our subdivision in turn are not part of $\tilde{P}$ and can be handled separately as a trivial atomic path.

If τ_1 and τ_2 were 0-monochromatic then $h_{m,m+1} = 1$, which would imply $h_m = 1$ and $m = 3$. Since $V_1 = 1$ we necessarily have $V_{m-1} = V_2 = 0$, otherwise $h_3 \geqslant 2 > 1$, and thus we know that $e^m_{C_{m-1}} = e^m_1 = 0$. But because τ_1 and τ_2 have a unit at position 1 in their boolean tuple representation, they contain the inner vertex of the subdivision named 1 which is labeled oppositely with 1, which contradicts the assumption that they were 0-monochromatic.

We now have to confirm that this special case will not reoccur recursively bnforever. We note that $\tilde{h}_{m,m+1} := \#(1, B(\tau_1 \cap \tau_2)) = h_{m,m+1} \pm 1$, depending on V_{m-1}. If $V_{m-1} = 0$, then, as detailed above, $\tilde{h}_{m,m+1} = h_{m,m+1} + 1 = m - 2 + 1 = m - 1$. Thus, this special case cannot occur, because it requires $\tilde{h}_{m,m+1} = m - 2$. Otherwise, $\tilde{h}_{m,m+1} = h_{m,m+1} - 1 = m - 3$, in which case $\tilde{m} = m - 1$ holds for the new subdivision point $\tilde{m}$.

4.1.3 Case 3: Symmetric 1

If neither of the previous cases applies, we know that $h_{m,m+1} \neq h_m = h_{m+1}$, which implies $h_{m,m+1} = h_{m-1,m} = m - 2$ and $h_m = h_{m+1} = m - 1$ according to Figure 4.5 on page 35. Regrettably, in this situation we have that $B(\sigma_m \triangle \sigma_{m+1}) = \{1\}$ (hence the name "Symmetric 1"), meaning that simply doing an edge expansion with $m = m$ will not create any 0-monochromatic simplices at all.

Therefore, we will first apply an edge expansion at $m = m - 1$. This expansion works exactly as detailed in case 1, because we have that $h_{m-1} = h_{m-1,m} = m - 2 < m - 1 = h_m$. Merely the length bound of the path joining one of the newly created 0-monochromatic simplices (we will call it σ') with σ_ℓ might not hold true, but this is no big loss because we only need it until it crosses $\sigma_{m,m+1}$ after at most $m - 1$ simplices. The bound on the other path joining σ_1 obviously does hold true, because we even applied the expansion one simplex earlier.

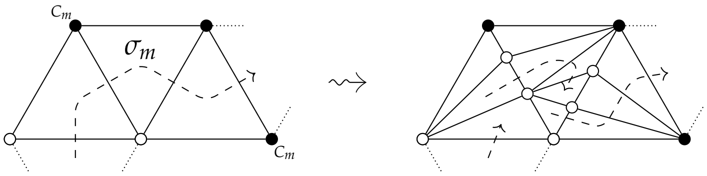


Figure 4.8: Edge expansion in case 3 (symmetric 1).

After this expansion, we can be sure that the coface of $\sigma_{m,m+1}$ in the subdivision of σ_m has a 0-vertex named C_m, because the vertex of σ_m named C_m was labeled with 1 and the expansion flips every 1-vertex on the outside into a 0-vertex on the inside. Thus, we can now apply another edge expansion, this time on $\sigma_{m,m+1}$ but again following the rules of case 1 (remember that the C_m-vertex of σ_{m+1} is still labeled with 1). One of the new 0-monochromatic simplices will as usual be joined by a path to σ_ℓ, while the other one is connected to σ'. By doing the same considerations regarding the length of a path connecting a 0-monochromatic simplex inside the edge expansion to the next vertex outside twice, we deduce that this path also has a length of at most $2m-2$, leaving us with three atomic paths to process instead of two. An example of dimension 2 can be seen in Figure 4.8.

4.1.4 Repeat

We saw in an exhaustive case analysis that—except for one special case—one application of this algorithm splits its input path into two or three new atomic paths, all shorter by at least two simplices than the original one. In the special case, the algorithm yields only one path of the same length as its input, but we saw that the next application of the algorithm on this path will not run into this special case again.

Thus, if we apply the algorithm often enough, only paths of length two will remain, which we dealt with on page 34. All in all, we now have one way to eliminate two 0-monochromatic n-simplices of opposite orientation in the subdivision we constructed in Chapter 3 at a time, and by repeating the recursion to eliminate all 0-monochromatic n-simplices.

A quick word on the round complexity of this method: If we ignore the special case for a moment, we know that any path of length $\ell \leqslant 2(n+1)$ will be eliminated by at most $n+1$ iterations of the algorithm, because its length is reduced by two in each invocation. Even if we consider the special case, we will need at most twice as many iterations, as we know that after every special case a regular iteration follows. Furthermore, each iteration applies the edge expansion at most twice, so all in all, a path of length $\ell \leqslant 2(n+1)$ is eliminated by at most $4(n+1)$ edge expansions. Longer paths can, however, be easily chopped into chunks of length at most $2(n+1)$ by one preprocessing step that is similar to the cases we already dealt with, as shown in [ACHP13, Claim 1, p. 198]. Hence, this method needs at most $4(n+1)+2 \in \mathcal{O}(n)$ subdivisions to eliminate a single path of arbitrary length.

4.2 Algorithm by Kozlov [Koz15]

We will now take an approach that was first introduced in [Koz15]. It uses the height graph extensively, applying various subdivisions in order to reduce the maximal height of the path until it only consists of 0-monochromatic simplices. Apart from edge expansions, we will also need another kind: The *vertex expansion* of an n-dimensional geometric simplex path Σ of length $\ell \geqslant 3$, labeled with $[1]$ by B and colored with $[n]$ by Π, along with its abstract simplex path $P = (I, C, V)$, is given by a triple (m, D, Q) such that

VERTEX EXPANSION

- $2 \leqslant m \leqslant \ell - 1$, specifying which simplex is subdivided,
- D is an $[n]$-tuple $(d_0, \dots, d_n)$ of boolean values, giving the labeling of the newly created vertices, and
- Q is a t-tuple of values from Π that defines how the new path routes through the subdivision, such that
  - $q_1 = C_{m-1}$ and $q_t = C_m$,
  - for every $l \in [n]$, we have that $\#(l, Q)$ is even, and
  - for all $1 \leqslant i < j \leqslant t$ except $(i,j) = (1,t)$, there exists an $l \in [n]$ such that $\#(l, Q[i,j])$ is odd.

Note that the only real difference to the requisites of the edge expansion (defined on page 33) is that there is no splitting of Q into two parts. This makes sense when we really construct the vertex expansion: It is the basic chromatic subdivision of σ_m, whereas in the edge expansion we subdivided $\sigma_{m,m+1}$. The rest is completely analogous: We label the new vertex colored $i \in [n]$ with d_i and reroute our path according to Q just like we did earlier, except that this time there is no crossing the face $\sigma_{m,m+1}$. Hence, we end up with the new path

$$\tilde{\Sigma} = (\sigma_1, \dots, \sigma_{m-1}, u_1, \dots, u_{t-1}, \sigma_{m+1}, \dots, \sigma_\ell)$$

where the u_i as before uniquely describes a simplex of the subdivision of σ_m by specifying for each color whether we chose the old outer or the new inner vertex of this color, exchanging inner and outer of the color q_i in the transition from u_{i-1} to u_i. This time, the abstract simplex path is modified to the following:

$$\begin{aligned}
\tilde{I} &= I \\
\tilde{C} &= (C_1, \dots, C_{m-1}, q_2, \dots, q_{t-1}, C_m, \dots, C_\ell) \\
\tilde{V} &= (V_1, \dots, V_{m-2}, w_1, \dots, w_{t-1}, V_m, \dots, V_\ell) \\
&\text{where } w_i := \begin{cases} e^i_{q_i} & \text{if } \#(q_i, (q_1, \dots, q_i)) \text{ is even} \\ d_{q_i} & \text{else.} \end{cases}
\end{aligned}$$

Compared to the edge expansion, the only difference is the absence of the flip across $\sigma_{m,m+1}$, all other formulas stay the same.

ADMISSIBLE

Furthermore, we will use the notion of admissible paths, which are concatenations of atomic paths: An abstract simplex path $P = (I, C, V)$ is *admissible* if $R^0 = R^\ell = (0, \dots, 0)$, ℓ is even, no simplex is 1-monochromatic and when considering all pairs (R^i, R^{i+1}) for even $2 \leqslant i \leqslant \ell - 2$ either both or no simplices of the pair are 0-monochromatic. Obviously, atomic paths themselves are admissible as well. The goal of considering these paths is that if we have a function that takes an atomic path as input and yields an admissible path as output, we can split its admissible output into atomic parts and re-apply the function.

4.2.1 Exhaustive expansion

While in the last section we carefully constructed our subdivision such that they introduce exactly two new 0-monochromatic n-simplices and no 1-monochromatic ones, we will relax this now by specifying exactly which kinds of new monochromatic simplices to allow.

Given two abstract simplex paths P and $\tilde{P}$, where $\tilde{P}$ is a vertex expansion of P, we associate the $[1] \times [n]$ array* A to them, where $A_{0,*} := R^m(P)$ and $A_{1,*} := D$:

$$A = \begin{array}{|c|c|c|c|} \hline e^m_0(P) & e^m_1(P) & \cdots & e^m_n(P) \\ \hline d_0 & d_1 & \cdots & d_n \\ \hline \end{array}$$

*An array is essentially a matrix whose indexing starts at 0.

As we know from Proposition 2.5 on page 12, every n-simplex in the subdivided version of σ_m in $\tilde{P}$ can be uniquely described by mapping each color to a boolean value that determines whether it contains the inner or outer vertex of that color. We will thus interpret this $[1] \times [n]$ array as all possible chromaticity tuples of n-simplices of σ_m's subdivision: Take the boolean $[n]$-tuple α associated to an n-simplex $\tilde{\sigma}$ in the subdivision, and look at $A_{\alpha(i),i}$ to know what boolean label its vertex with color i has. Together, $A(\alpha) := (A_{\alpha(0),0}, \dots, A_{\alpha(n),n})$ describes the whole coloring of $\tilde{\sigma}$. If it contains only zeros then $\tilde{\sigma}$ is 0-monochromatic, if it contains only ones then it is 1-monochromatic.

Exactly the same procedure is possible for the edge expansion if we set $A_{1,i} := d_i$ only for $i \in [n] - C_m$ and $A_{1,C_m} := V_m = e^{m+1}_{C_m}(P)$. That way, all but the C_m-th entry describe the situation inside $\sigma_{m,m+1}$ and the C_m-th entry chooses between the two vertices of $\sigma_m \triangle \sigma_{m+1}$:

$A =$

$e^m_0(P)$	$\cdots$	$e^m_{C_m-1}(P)$	$e^m_{C_m}(P)$	$e^m_{C_m+1}(P)$	$\cdots$	$e^m_n(P)$
d_0	$\cdots$	d_{C_m-1}	$e^{m+1}_{C_m}(P)$	d_{C_m+1}	$\cdots$	d_n

For example, the array associated to the edge expansion done in Figure 4.3 on page 34 looks like this:

$A =$

1	0	0
1	0	0

Note that it is impossible to choose one entry from each column such that they all have the same value, as was to be expected because the expansion did not yield any monochromatic simplices.

We will call a simplex (whose corresponding $[n]$-tuple is α) *on-path* if there is an (and *off-path* if there is no) index i satisfying ON-/OFF-PATH

(1) if $\tilde{P}$ is a vertex expansion, then $1 \leqslant i \leqslant t-1$,
(2) if $\tilde{P}$ is an edge expansion and
 (2.a) $\alpha(C_m) = 0$, then $1 \leqslant i \leqslant s$ or
 (2.b) $\alpha(C_m) = 1$, then $s \leqslant i \leqslant t-1$,
(3) and in every case

$$\alpha(j) = \#(j, (q_1, \dots, q_i)) \mod 2$$

for all $j \in [n]$ if $\tilde{P}$ is a vertex expansion or for all $j \in [n] - C_m$ if $\tilde{P}$ is an edge expansion.

Ignoring conditions (1) and (2) for a moment, we will have a closer look at condition (3) to bridge it to our geometric intuition. We recall that $\alpha(j) = 1$ means that in the simplex associated to α, the vertex colored with j is one of the newly added inner ones, while a value of 0 means that it is an outer vertex. We also recall that $Q[1,i] = (q_1, \dots, q_i)$ specifies the routing inside the expansion up to the i-th "flip", and that we start outside of our subdivision before doing the first flip defined by q_1. Since we stay inside our subdivision the hole time, having flipped a vertex colored by j an odd number of times means that it is currently an inner vertex, while an even number of flips means it is an outer vertex—which is exactly what the equation says.

Condition (1) really does not say much, it just ensures that we still are in the subdivision (recall that q_t, the last flip, already leaves the subdivision and flips to σ_{m+2}). Condition (2) is a bit more interesting, as it takes into account in which part of the edge expansion the simplex is located, which is encoded in the C_m-th column of A and thus represented by $\alpha(C_m)$.

So all in all, the combinatorial definition says exactly what our geometric intuition demanded for the terms "on-path" and "off-path" and, on top of that, even gives us the index at which the monochromatic simplex is located in the new path if it is on-path:

Lemma 4.2. *Given a vertex or edge expansion with parameters (m, D, Q) or (m, D, Q, s), respectively, that turns a simplex path P into $\tilde{P}$, and an on-path simplex σ along with its parameter i from the definition of on-path, then the position of σ in $\tilde{P}$ is*

- *$m - 1 + i$ in cases (1) and (2.a) from the definition of on-path, or*
- *$m + i$ in case (2.b).*

Now, let $M(A) := \{\alpha\colon [n] \to [1] \mid A(\alpha) \text{ is monochromatic}\}$. If $M(A) = \emptyset$, everything is fine as no new monochromatic n-simplices arose from the expansion. Otherwise, we could have two situations: Either $M(A)$ describes two "opposite" monochromatic simplices, one 0-monochromatic and one 1-monochromatic, in which case exactly half of the entries of A are 0. Or else all elements of $M(A)$ are b-monochromatic, with $b \in [1]$. They cannot mix in this case because more than one b-monochromatic simplex in $M(A)$ means more than $n+1$ entries in A being b, hence less than $n+1$ entries are $1-b$, not allowing for any (1-b)-monochromatic simplices.

Note that two simplices with boolean tuple representation α_1 and α_2 share an $(n-1)$-face if and only if α_1 and α_2 differ in the image of exactly one input, i.e. there is an $i \in [n]$ such that $\alpha_1(i) \neq \alpha_2(i)$ and $\alpha_1(j) = \alpha_2(j)$ for every other $j \in [n] - i$. We can think of all

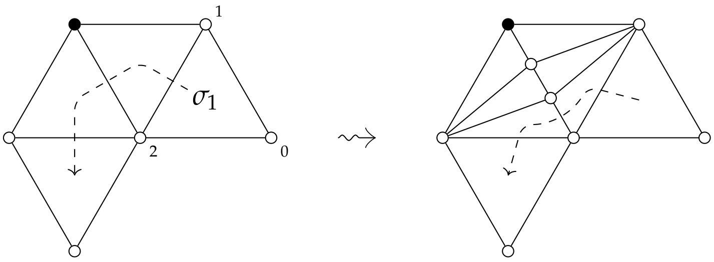


Figure 4.9: Edge expansion on the path from Figure 4.3 on page 34, this time with $m = 2$, $D = (0, -, 0)$, $Q = (0, 0)$ and $s = 1$.

simplices of the subdivision as a graph by setting the vertices to be the simplices and drawing an edge whenever the simplices share an $(n-1)$-face. In the same way, we have a graph consisting of the tuple representations, which actually is the same. Because the tuples are boolean, this is exactly the 1-skeleton of a hypercube if we include the normally invalid tuple $(0, \ldots, 0)$. Obviously, because $M(A)$ is a subset of all these tuples, it is also a subgraph of the hypercube.

EXHAUSTIVE EXPANSION

We now call an expansion *exhaustive* if all simplices $\alpha \in M(A)$ are either on-path or can be matched with another neighboring monochromatic simplex of the same kind that is also off-path in order to form a trivial atomic path that can be resolved as such (see page 34). Combinatorially, the expansion $\tilde{P}$ is called exhaustive if there exists a perfect graph-matching on $\{\alpha \in M(A) \mid \alpha \text{ is off-path}\}$.[*]

Example 4.3. We will consider one more example that is a bit more interesting than the one in Figure 4.3 on page 34: Consider the atomic abstract simplex path $I = (0, 0, 0)$, $V = (1, 0, 0)$, $C = (0, 1, 0)$, which is the same as previously, but this time do an edge expansion with input data $m = 2$, $D = (0, -, 0)$, $Q = (0, 0)$ and $s = 1$. Without any geometric intuition, we can easily build the associated array just from the definitions:

$$A = \begin{array}{|c|c|c|} \hline e_1^2 & e_2^2 & e_3^2 \\ \hline d_1 & V_2 & d_2 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

We can see that $M(A)$ contains four elements:

$$M(A) = \{(1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}$$

[*] While the definition in [Koz15, Def. 4.3] seems to allow matching on-path with off-path simplices, this is actually strictly forbidden!

This is consistent with the geometric picture that can be seen in Figure 4.9 on page 47, which contains four monochromatic simplices in the subdivision. Two of them are part of the new simplex path, while the other two are not but are neighboring, so this expansion should be exhaustive. Indeed, simply matching $(1,0,1)$ with its neighbor $(1,1,1)$ satisfies all conditions. To see this, we need to check the on-path condition for all four vertices:

- For $\alpha = (1,0,0)$ we have $\alpha(C_m) = \alpha(C_2) = \alpha(1) = 0$ and thus by condition (2.a) $i = 1$. Thus $(q_1,\dots,q_i) = (q_1) = (0)$ and $\#(j,(0)) = \delta_{j,0} = \alpha(j)$ for all $j \in [2] - C_m = \{0,2\}$. So $(1,0,0)$ is on-path, which is consistent with it being unmatched.
- For $\alpha = (1,0,1)$ we also have $i = 1$ and $(q_1) = (0)$, but $\#(2,(0)) = 0 \neq 1 = \alpha(2)$, thus $(1,0,1)$ is off-path and must be matched, which it was.
- For $\alpha = (1,1,0)$ we have $\alpha(C_m) = 1$ and thus by condition (2.b) $i = 1$. This gives $(q_1,\dots,q_i) = (0)$ again, and $\alpha(j) = \#(j,(0)) = \delta_{j,1}$ holds for all $j \in [2] - C_m = \{0,2\}$. As such, $(1,1,0)$ is on-path and consequently was not matched.
  Note that this is where we need to make an exception for C_m: The parameter Q of an edge expansion can never contain C_m, thus $\#(C_m,(q_1,\dots,q_i)) = 0 \mod 2$ no matter what Q we are given. If we did not make an exception, this would mean that all simplices in the subdivision of σ_{m+1} cannot be on-path, which obviously is not what we want.
- Finally, for $\alpha = (1,1,1)$ we again have $i = 1$ by case (2.b), but regardless of the choice of i we always have $\#(2,(q_1,\dots,q_i)) = 0 \neq 1 = \alpha(2)$ because Q only contains zeros. Thus $(1,1,1)$ is off-path and was matched.

4.2.2 Moves

The actual algorithm will use *moves*, comparable to the case analysis done in Section 4.1. They are special types of edge or vertex expansions. In contrast to the earlier approach, where we first fixed a position in the path, this time the moves apply to different positions, depending on the type of move.

Summit move If, in an atomic path $P = (C,V)$ of length ℓ, we have a position $2 \leqslant m \leqslant k-1$ such that $h_m > h_{m-1}$ and $h_m > h_{m+1}$, we call this position a *summit*, referring to the height graph at this position (cf. Figure 4.11 on the next page). Obviously, the path needs to be at least 3 simplices long in order for such an m to exist,

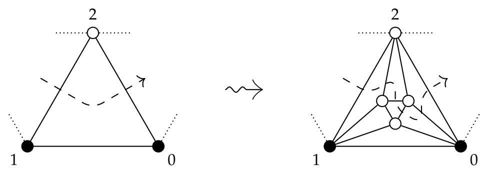


Figure 4.10: A summit move in a two-dimensional path. Only the relevant simplex σ_m of the path is shown.

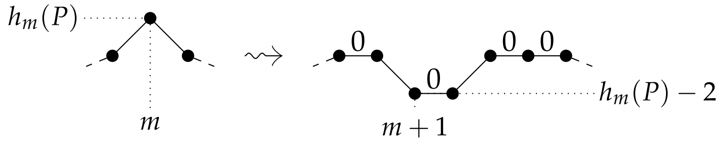


Figure 4.11: Height graph before and after a summit move, read off Figure 4.10.

and because it is supposed to be atomic we can even assume it to contain 4 simplices. This implies that $h_m \geqslant 2$, because if $h_m = 1$ then $h_{m-1} = h_{m+1} = 0$, contradicting the atomicity of the path. Furthermore, the even length of the path enables us to assume that m is odd: If it is not, we simply reverse the path, which makes the same simplex have an odd index.

Up to $S_{[n]}$-action, we can assume that $(C_{m-1}, C_m) = (0, 1)$. Furthermore, because $h_m > h_{m-1}$, we know that $V_{m-1} = 1$ and thus $e_0^m = 1$. Analogously, $h_m > h_{m+1}$ gives us $V_m = 0$ and $e_1^m = 1$. Because there has to be at least one vertex labeled with 0 in the simplex R^m, it might as well be the one colored with 2, so we set $e_2^m = 0$. This makes the simplex look something like this:

$$R^m = (1, 1, 0, e_3^m, \ldots, e_n^m).$$

This simplex is subdivided using a vertex expansion with the input data

$$m := m, \qquad D := (0, 0, 0, \overline{e_3^m}, \ldots, \overline{e_n^m}), \qquad Q := (0, 1, 2, 0, 2, 1).$$

An example can be seen in Figure 4.10.

Notice how our knowledge of three vertices is sufficient to fully analyze the effect of this move on the path. In particular, we note that the summit move is an exhaustive expansion, because we can see in the array

$A =$

1	1	0	e_2^m	$\cdots$	e_n^m
0	0	0	$\overline{e_2^m}$	$\cdots$	$\overline{e_n^m}$

that $M(A) = \{\underbrace{(1, 1, 0, e_3^m, \ldots, e_n^m)}_{=:\alpha_1}, \underbrace{(1, 1, 1, e_3^m, \ldots, e_n^m)}_{=:\alpha_2}\}$

where the last part is identical, thus the two 0-monochromatic simplices are neighbors. If $h_m = 2$ then $e_3^m = \cdots = e_n^m = 0$, which makes both simplices on-path because $\#(j,(q_1,q_2)) = \#(j,(0,1)) = \alpha_1(j)$ and $\#(j,(q_1,q_2,q_3)) = \#(j,(0,1,2)) = \alpha_2(j)$ (choosing $i_1 = 2$ and $i_2 = 3$, respectively) for all $j \in [n]$. If $h_m > 2$, however, $e_j^m = \alpha_1(j) = \alpha_2(j) = 1$ for at least one $j \geqslant 3$. But because Q only contains 0, 1 and 2, condition (3) for on-path simplices can never be fulfilled, hence both are off-path.

Furthermore, we can see that the path resulting from the summit move is admissible: It has length $\ell + 4$, which is even because ℓ is even, and contains either both or none of the new 0-monochromatic simplices, as we have just seen. We now have to show that if it contains them, i.e. if $h_m = 2$, then their indices fall into the same "bucket" of consecutive even-odd index-pairs. This follows from Lemma 4.2 on page 46, because one of the simplices has index $m + i_1 - 1 = m + 1$ and the other $m + i_2 - 1 = m + 2$ in the new path. Because the orientation of the path was chosen such that m is odd, $m + 1$ and $m + 2$ fall into the same even-odd-pair, thus the result of a summit move is admissible.

Finally, let us see what impact the summit move has on the height graph of the path. The only inner vertices of the subdivision are those named 0, 1 and 2 because Q only contains these three names. We can thus ignore all others in our height consideration because they are the same as before the move, which enables us to use the generic example in dimension two from Figure 4.10 on page 49 as general argument for all dimensions, which yields the height graph change in Figure 4.11 on page 49. In particular we notice that locally the maximal height of the path was reduced by one.

Generic plateau move Let $P = (C, V)$ again be an abstract atomic simplex path. If there is a position $3 \leqslant m \leqslant \ell - 2$ such that $h_m = h_{m+1} > h_{m-1}$, we call this position a *plateau*, again referring to the height graph. Notice that $h_m \geqslant 2$ because $m - 1 \geqslant 2$ and hence σ_{m-1} cannot be monochromatic.

By $S_{[n]}$-action we can again assume that $(C_{m-1}, C_m) = (0, 1)$ and because of $h_m > h_{m-1}$ also that $V_{m-1} = 1$. Furthermore, $h_m = h_{m+1}$ gives us that $V_m = e_{C_m}^m = e_1^m$. We now decide that we want $C_{m+1} = 2$, where 2 could really be any other name by $S_{[n]}$-action except 1 which is prevented by the no-backflip condition, and 0, which will be handled later by the special plateau move.

Then the generic plateau move is an edge expansion with the parameters

$$m := m, \qquad D := (0, -, e_2^m, \ldots, e_n^m), \qquad s := 2, \qquad Q := (0, 2, 0, 2).$$

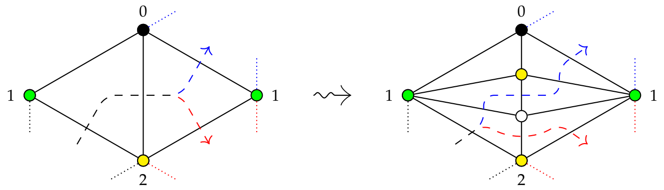


Figure 4.12: Two-dimensional generic (blue path) and special (red path) plateau moves. The colors yellow and green represent binary labels of the respective vertices, where the same color is labeled with the same value, but different colors may be labeled differently. White and black have their usual meaning of a vertex being labeled with 0 or 1.

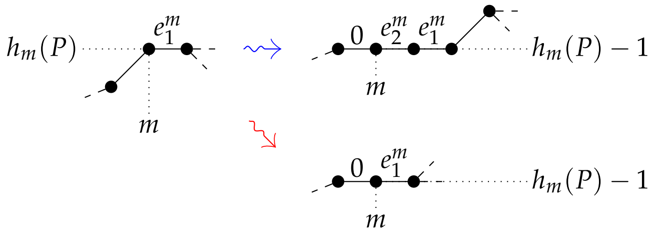


Figure 4.13: Excerpt from the height graph of a path before and after applying the generic (top) and special (bottom) plateau move, read off Figure 4.12.

Again we can see that this move is exhaustive if we have a look at the associated array:

$$A = \begin{array}{|c|c|c|c|c|} \hline 1 & e_1^m & e_2^m & \cdots & e_n^m \\ \hline 0 & e_1^m & e_2^m & \cdots & e_n^m \\ \hline \end{array}$$

We note that if not $e_1^m = \cdots = e_n^m =: b$, then $M(A)$ is empty. However, $b = 0$ is impossible because then $h_m = 1$, contradicting our choice of m, and $b = 1$ is impossible as well because it would make R^i 1-monochromatic. Hence, $M(A)$ is empty and the expansion is exhaustive.

The resulting path is also atomic, because its length $\ell + 2$ is even and no new monochromatic simplices have been introduced as we just saw, thus the only monochromatic simplices still are the end simplices that are untouched by the expansion, thanks to our choice of m.

With the same argument as before that only the three vertices with the lowest names are ever flipped and in fact only the vertex labeled with 0 has another binary label in the interior that on the boundary, we can use the two-dimensional illustration in Figure 4.12 to derive the general rule on what happens to the height graph, as can be seen in Figure 4.13. In particular we note that the rising edge of the height graph was pushed towards the end of the path.

Special plateau move In the same scenario that the generic plateau move requires, we also could have that $C_{m+1} = 0$ (instead of $C_{m+1} = 2$ as above). In this case, we need to slightly adjust our parameters for the edge expansion: m and D are the same as above, but we set $Q := (0, 0)$ and $s := 1$.

Note that our considerations concerning the exhaustiveness of the expansion and the atomicity of the resulting path hold almost unaltered (of course this time the length does not change at all, which makes the new path of even length as well), as does the argument for constructing the height graph from the two-dimensional visualization in Figure 4.12 on page 51, resulting in the height graph in Figure 4.13. Compared to the generic plateau move, the special one even has the advantage that it does not extend the path and could locally reduce the height if $V_{m+1} = 0$, which would mean that a "widened summit" would be flattened by this move.

With these three moves, we can already achieve part of our goal:

LOW ADMISSIBLE PATH

Lemma 4.4. *Given an admissible simplex path P, there is a sequence of exhaustive vertex or edge expansions that transforms P into a* low admissible path*, that is, an admissible path with maximal height 1.*

Proof. Assume that P is an admissible path that cannot be reduced to a low admissible path. Obviously, P cannot be low itself, so $\max_i h_i(P) \geqslant 2$. Furthermore, we want P to be a "lowest" path that is not reducible, that is, one of the smallest with respect to the reverse lexicographic order of their height graphs: Given two paths P and $\tilde{P}$ of lengths ℓ and $\tilde{\ell}$, we start from the right by comparing $h_\ell(P)$ and $h_{\tilde{\ell}}(\tilde{P})$. If they differ, we order P and $\tilde{P}$ accordingly, if they are the same we proceed to $h_{\ell-1}(P)$ and $h_{\tilde{\ell}-1}(\tilde{P})$ and repeat the comparison. We simply write $P > \tilde{P}$ if this comparison yields that P is greater than $\tilde{P}$.

First of all, P cannot contain a summit. If it did, we could apply the summit move to obtain another admissible path $\tilde{P}$. The new path $\tilde{P}$ must not be reducible to a low admissible path either, because otherwise P would be too, as the summit move is an exhaustive vertex expansion. But as we can see in Figure 4.11 on page 49, the height graphs of P and $\tilde{P}$ are identical after the simplices m and $m + 4$ respectively, and we have $h_m(P) > h_m(P) - 1 = h_{m+4}(\tilde{P})$ which makes $P > \tilde{P}$, contradicting our assumption that P is minimal.

But if P has no summits, it must have a plateau: Pick $2 \leqslant m \leqslant \ell - 1$ as the first position such that $\max_i h_i(P) = h_m(P)$. This implies that $h_{i-m}(P) < h_m(P)$ and because there are no summits $h_{m+1}(P) = h_m(P)$. Hence, we have a plateau at position m and can apply the appropriate plateau move to get a new path $\tilde{P}$. Again, $\tilde{P}$ must not be reducible to a low admissible path, because otherwise P would be as well by first reducing it to $\tilde{P}$. If we needed to apply a generic plateau move, the height graphs of P and $\tilde{P}$ are identical from $m+1$ and $m+3$ onwards, respectively, as can be seen in Figure 4.13 on page 51. They first differ in $h_m(P) > h_m(P) - 1 = h_{m+2}(\tilde{P})$, again contradicting our minimality assumption. If we needed to apply a special plateau move, the height graphs even are completely identical except in position m, and according to Figure 4.13 we have $h_m(P) > h_m(P) - 1 = h_m(\tilde{P})$.

Hence, there is no such path P, which means that all admissible paths are reducible to low admissible paths using exhaustive vertex or edge expansions. □

4.2.3 Low admissible paths

While the three moves are powerful enough to reduce arbitrary admissible paths to low admissible paths, they all require that $\max_i h_i \geqslant 2$, which makes them unsuitable for further reducing low admissible paths. We will therefore continue with other moves to turn low admissible paths into admissible 0-paths, that is, to paths of even length containing only 0-monochromatic simplices.

In the following, we always start with a low atomic path P of length ℓ. The path being low has several consequences, in particular that every simplex except the first and the last has exactly one vertex labeled with 1. This vertex always has the same name, because changing to another name would require two flips, one that flips the old name to 0 and another one that flips the new name to 1, and thus generate an intermediate simplex that has either height 0 or 2, both impossible in a low atomic path. In terms of our combinatorics, this means that after applying $S_{[n]}$-action, we can assume that $C_1 = 0$, $C_{\ell-1} = 0$ and

$$R^2(P) = R^3(P) = \cdots = R^{\ell-1}(P) = (1, 0, \ldots, 0).$$

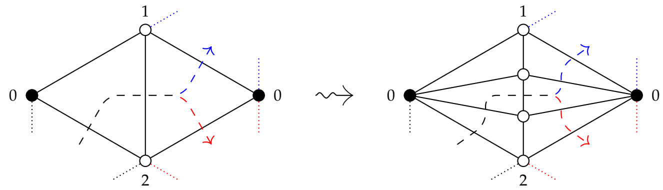


Figure 4.14: The move to flatten a unit applied to a two-dimensional path (clipped to the relevant simplices σ_m and σ_{m+1}). The red part represents $C_{m+1} = 1$ whereas the blue part corresponds to $C_{m+1} = 2$.

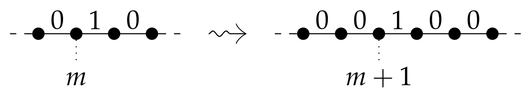


Figure 4.15: Height graph change when flattening a unit, valid for both choices of C_{m+1}

This enables us to determine the labeling part V of the abstract simplex path completely from its flipping part C:

$$V_i = \delta_{C_i,0} \quad \text{for } 1 \leqslant i \leqslant \ell - 1.$$

Flatten a unit Let us choose an even $4 \leqslant m \leqslant \ell - 4$ such that $V_m = 1$ if it exists. The purpose of this move is to change the parity of m from even to odd. Note that while earlier we were able to simply reverse the path to change the parity of the position we applied the summit move on, this is impossible here because we now consider the parity of a shared $(n-1)$-face which is preserved when reversing the path.

We can assume that $(C_{m-1}, C_m) = (1, 0)$ and $C_{m+1} \in \{1, 2\}$ using $S_{[n]}$-action, which implies by our earlier considerations that $V_{m-1} = V_{m+1} = 0$. The move to flatten a unit is an edge expansion with the parameters

$$m := m, \qquad D := (-, 0, \ldots, 0), \qquad s := 2, \qquad Q := \begin{cases} (1,2,2,1) & \text{if } C_{i+1} = 1, \\ (1,2,1,2) & \text{if } C_{i+1} = 2. \end{cases}$$

As we can easily see using the associated array, this expansion does not yield any monochromatic simplices at all, thus it is exhaustive:

$$A = \begin{array}{|c|c|c|c|} \hline 1 & 0 & \cdots & 0 \\ \hline 1 & 0 & \cdots & 0 \\ \hline \end{array}$$

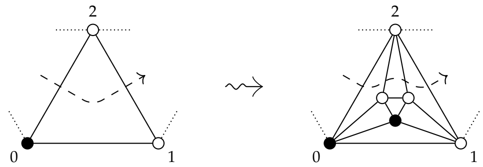


Figure 4.16: The move to eliminate a unit in a two-dimensional path. Only the relevant simplex σ_m of the path is shown.

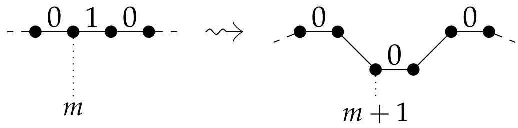


Figure 4.17: Height graph before and after eliminating a unit.

And because it increases the length by two, we end up with a still admissible (and even atomic) path. Also, we succeeded at our goal of making the flip to a 1 occur at an odd position, as can be seen in Figures 4.14 and 4.15 on page 54.

Eliminate a unit We assume the same situation as for the move that flattens a unit, only this time we require m to be odd. By applying that move, we can easily force it to be odd if needed. Eliminating a unit then is a vertex expansion with the parameters*

$$m := i, \qquad D := (0, 0, 1, 1, \ldots, 1), \qquad Q := (1, 0, 1, 0).$$

This expansion creates exactly two 0-monochromatic simplices, as can be seen using the associated array:

$A =$

1	0	0	$\cdots$	0
0	0	1	$\cdots$	1

$$M(A) = \{\underbrace{(1, 0, 0, \ldots, 0)}_{\alpha_1}, \underbrace{(1, 1, 0, \ldots, 0)}_{\alpha_2}\}$$

They are both on-path: For α_1 choosing $i = 3$ yields $(q_1, \ldots, q_i) = (1, 0, 1)$ and thus $\#(j, (1, 0, 1) = \delta_{j,1} = \alpha_1(j)$ for all $j \in [n]$. For α_2 we can choose $i = 2$ and get $\#(j, (1, 0)) = \delta_{j,0} + \delta_{j,1} = \alpha(j)$. This is consistent with the geometric picture given in Figure 4.16.

*[Koz15] differs by setting $Q := (0, 1, 0, 1)$, which is illegal because $C_{m-1} = 1 \neq 0 = q_1$.

$\alpha(0)$	$\alpha(1)$	$\alpha(2)$	$\alpha(3)$	i	$(q_1,\dots,q_i)$	
1	1	0	0	2	$(1,0)$	on-path, first half
1	1	0	1	3	$(1,0,3)$	($\alpha(C_m)=0 \Rightarrow i \leqslant s$)
1	0	1	1	4	$(1,0,3,1)$	on-path, second half
1	1	1	1	3	$(1,0,3)$	($\alpha(C_m)=1 \Rightarrow i \geqslant s$)
1	0	0	0			off-path pair 1
1	0	0	1			
1	0	1	0			off-path pair 2
1	1	1	0			

Table 4.18: Analysis of monochromatic simplices occurring in the move to shorten generic zeros.

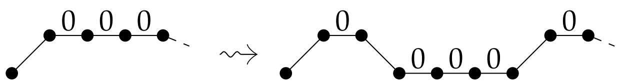


Figure 4.19: Height graph before and after shortening generic zeros.

Together, the moves to flatten and to eliminate units enable us to reduce every low admissible path to a form where the height graph contains no horizontal edges labeled with 1. We now only need to deal with horizontal edges labeled with 0, which is where our last moves applies:

Shorten generic zeros Assume that $V_2 = V_3 = V_4 = 0$ (and remember that $V_1 = 1$ necessarily). This implies that $1 \notin \{C_2, C_3, C_4\}$ as per our earlier considerations that $R^i(P) = (1,0,\dots,0)$ for all $i \geqslant 2$ after applying $S_{[n]}$-action. We can thus assume that $C = (0,1,2,C_4,\dots)$ and $C_4 \in \{1,3\}$. In analogy to the plateau move, we first assume that $C_4 = 3$. In that case we apply an edge expansion with the parameters

$$m := 3, \qquad D := (0,0,-,0,1,\dots,1), \qquad Q := (1,0,3,1,0,3), \qquad s := 3.$$

The relatively large number of zeros in D already indicates that this expansion will create quite a few 0-monochromatic simplices, and indeed we have eight of them:

$$A = \begin{array}{|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & \cdots & 1 \\ \hline \end{array}$$

$$M(A) = \{(1,\alpha(1),\alpha(2),\alpha(3),0,\dots,0) \mid \alpha(1),\alpha(2),\alpha(3) \in [1]\}$$

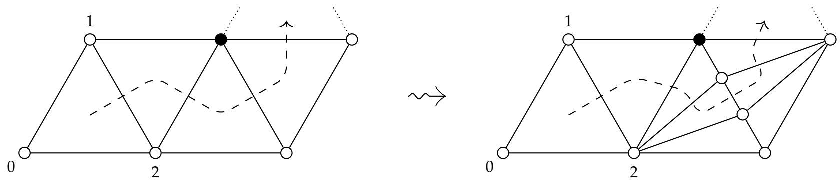


Figure 4.20: The move shortening special zeros applied to a two-dimensional path (clipped to the relevant simplices σ_1 to σ_4).

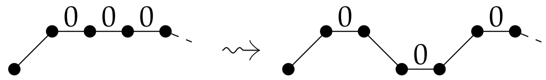


Figure 4.21: Height graph before and after shortening special zeros.

Half of these are on-path, half of them are not but pairable, as the analysis in Table 4.18 on page 56 shows, thus the move to shorten generic zeros is exhaustive. It is also admissible, because the 0-monochromatic ones appear at positions

$$(m-1+2, m-1+3, m+3, m+4) = (4,5,6,7)$$

in $\tilde{P}$ according to Lemma 4.2 on page 46 and thus comprise three atomic paths.

Regrettably, this is the only move that cannot adequately be projected down to two dimensions, because $|\{q_i \mid 1 \leqslant i \leqslant 6\} \cup \{C_m\}| = 4 > 3$, but the height graph is depicted in Figure 4.19 on page 56.

Shorten special zeros In the same situation as the previous move, assume that $C_4 = 1$. We then apply an edge expansion with

$$m := 3, \qquad D := (0,0,-,1,1,\dots,1), \qquad Q := (1,0,0,1), \qquad s := 2.$$

Under these circumstances, we only need to deal with four 0-monochromatic simplices:

$$A = \begin{array}{|c|c|c|c|c|c|c|} \hline 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & 0 & 1 & 1 & \cdots & 1 \\ \hline \end{array}$$

$$M(A) = \{(1, \alpha(1), \alpha(2), 0, \dots, 0) \mid \alpha(1), \alpha(2) \in [1]\}$$

Their analysis is much easier: The two vertices of $M(A)$ starting with $(1,1,\dots)$ are both on-path, because for $i = 2$ we have $\#(j,(1,0)) = \delta_{j,0} + \delta_{j,1} = \alpha(j)$ for all $j \in [n]-2$, whereas the two starting with $(1, 0, \ldots)$ are off-path as no prefix of Q contains an odd number of zeros but an even number of ones. Thus, again according to Lemma 4.2 on page 46, the 0-monochromatic simplices are at positions $m - 1 + 2 = 4$ and $m + 2 = 5$, which (together with the still even length) makes the resulting path admissible and the expansion exhaustive.

4.2.4 Reducibility of all paths

Theorem 4.5. *For every admissible abstract simplex path P there is a sequence of exhaustive edge or vertex expansions transforming it into a constant 0-path.*

Proof. Using Lemma 4.4 on page 52 we can transform P into a low admissible path if needed, enabling us to assume that $\max_i h_i = 1$. We then work on each atomic path individually, applying the move to flatten a unit as long as there are any even units (positions with $V_m = 1$, $m > 1$ even) and then applying the move to eliminate a unit. This again yields multiple atomic paths, on each of which we apply the move to shorten zeros until no single atomic path has a length greater than four.

Now up to $S_{[n]}$-action, there is only one atomic abstract simplex path of length four, namely $P = ((0, 1, 0), (1, 0, 0)$. It can be split further into two paths of length two by an edge expansion with parameters

$$m := 2, \qquad D := (0, -, 1, \ldots, 1), \qquad Q := (0, 0), \qquad s := 1.$$

We see that

$$A = \begin{array}{|c|c|c|c|c|} \hline 1 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & 1 & \cdots & 1 \\ \hline \end{array}$$

$$M(A) = \{(1, 0, 0, \ldots, 0), (1, 1, 0, \ldots, 0)\}$$

Both monochromatic simplices are on-path with the choice of $i = 1$, putting them at positions $m - 1 + 1 = 2$ and $m + 1 = 3$ according to Lemma 4.2 on page 46. As the resulting path still has length four, all simplices are 0-monochromatic. □

As we saw earlier, the edge and vertex expansions do not modify the boundary of a geometric simplex path, and all exhaustive expansions generate superfluous 0-monochromatic simplices only in pairs. Hence, every atomic geometric simplex path can be subdivided such that every monochromatic n-simplex is 0-monochromatic and paired to a neighboring 0-monochromatic n-simplex. These trivial atomic paths were already handled on page 34, thus we eliminated all monochromaticities in the path, proving Theorem 4.1 on page 31.

4.2.5 Complexity

We will have a quick look at how many subdivisions or, equivalently, moves it takes to reduce a low admissible path to a 0-path.

First, we needed to change the parity of some positions i with $V_i = 1$ using the move to flatten a unit. Simply because only half the positions could possibly have the wrong parity, an upper bound for the number of applications of this move is $\ell/2$, each extending the path by two, so we end up with a path of length at most $\ell + 2\,\ell/2 = 2\ell$.

Then we applied the move to eliminate a unit to every position with $V_i = 1$. The number of these positions did not change by the application of the moves to flatten a unit, so we might as well consider their number in the original path. Because we are dealing with low admissible paths, at most every second flip can have $V_i = 1$, thus again, we apply this move at most $\ell/2$ times, each extending by two simplices again, resulting in a path of length at most $2\ell + 2\,\ell/2 = 3\ell$.

Finally, we needed to shorten the zeros. As this move is always applied to the start of a path, we see that each application of that moves generates a path of length 4 and an atomic tail that is two simplices shorter than the input was, thus we can apply this move at most $\ell/2$ times as well.

These resulting paths of length 4 can be turned first into trivial atomic paths and then into non-monochromatic simplices using two consecutive subdivisions, thus every admissible path can be demonochromatized using $3/2\ell + 2$ subdivisions.

The complexity of reducing an arbitrary atomic path to a low admissible path is more difficult to estimate. First, we note that we decrease the height graph "layer by layer", each time reducing its maximum height by one. The number of simplices with the maximal height h is limited by $\ell - 2h - 1$, because at least h_i flips have to precede as well as follow the simplex σ_i.

Thus, to reduce the height of the path by one, we might have to execute the summit move $\ell/2 - h$ times, because every simplex could be the top of a summit. Or we might

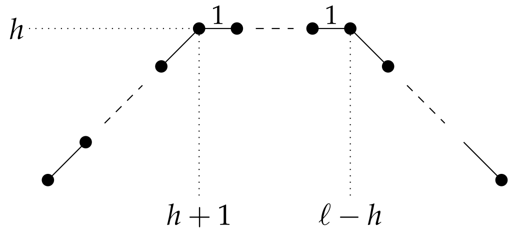


Figure 4.22: Height graph of a simplex path that needs the most subdivisions to turn it into a low admissible path.

have to execute the plateau move $\ell - 2h - 1$ times, if there are no summits at all, and the summit move once at the end after we made a summit out of the plateau. While a combination of these moves might be necessary as well, the number of moves will surely lie in between. So reducing the height by one takes at most $\ell - 2h$ subdivisions. Regrettably, it also extends the path by up to $2\ell - 4h + 2$ simplices: Each summit move extends the path by four, each (generic) plateau move by two simplices.

This shows that reducing the maximal height one more can be much more complex, as the path length could have doubled in the worst case. Our estimation is that it takes $\mathcal{O}(\ell^h)$ subdivisions to reduce an atomic path to a low admissible path, where h is the maximal height of the original path of length ℓ. Note that h itself is bounded by $n-1$, so you could also assume a worst case complexity of $\mathcal{O}(\ell^{n-1})$.

An example where this actually applies can be easily constructed: Given $h > 2$ and $\ell \geqslant 2h+2$, set $n = h+1$. The abstract atomic path then is constructed with

$$V = (\overbrace{1,\dots,1}^{\ell-h-1},\overbrace{0,\dots,0}^{h})$$
$$C = (1,2,\dots,h,C_{h+1},\dots,C_{\ell-1-h},1,2,\dots,h)$$

where $C_{h+1},\dots,C_{\ell-1-h}$ can be chosen arbitrarily from $\{1,\dots,h\}$ (obeying the no-backflip condition). Note that $C_i \neq 0$ and thus $e_0^i = 0$ for all $i \in \{1,\dots,\ell\}$. Hence no simplex is 1-monochromatic, making the path atomic if and only if ℓ is even. The resulting height graph is depicted in Figure 4.22.

5 A Global Approach

While both of the algorithms in Chapter 4 pick some point in the path and modify it locally in order to make the original path more beneficial in some terms, be it length or height, we will now attempt a more global approach to demonochromatizing simplex paths.

If we were trying to optimize the number of standard chromatic subdivisions needed to eliminate monochromatic simplices in a path, it seems beneficial to use as much of each subdivision as possible. A full standard chromatic subdivision is not possible, because one of the conditions on the subdivisions we make is that the boundary of the path must not be modified. But we are allowed to do a vertex expansion or basic chromatic subdivision of each n-simplex in the path, as it only affects its interior, and we are allowed to subdivide the gluing $(n-1)$-simplices afterwards. The result of both subdivisions combined can be seen in Figure 5.1 for dimension two. Note that for reasons of clarity we did not subdivide the first and last n-simplices as well as the first and last gluing $(n-1)$-simplices. They are all 0-monochromatic anyway and we gain nothing from subdividing them.

Concerning the coloring on the newly created vertices, we simply choose to color all of them with zero, because this causes a maximal amount of 0-monochromatic n-simplices in the subdivision. This way, chances are high that we can construct a complete pairing

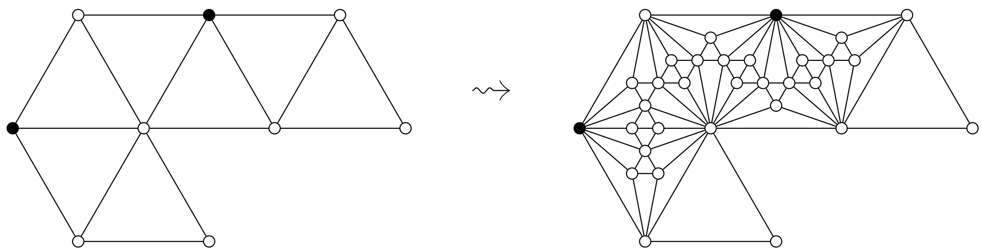
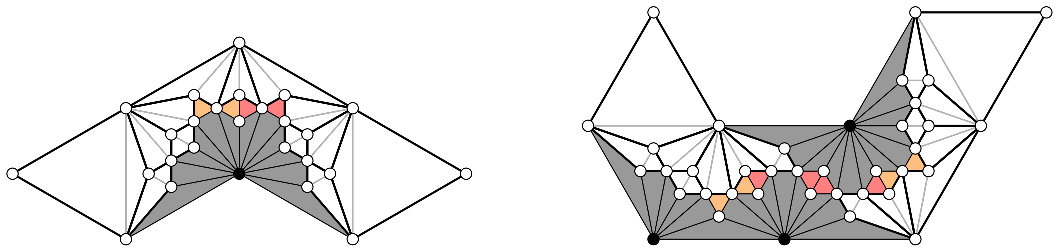


Figure 5.1: The global subdivision of a simplex path of dimension 2.



(a) The simplices in the orange and red pairs are not neighboring, but connecting them by a simple and regular path is possible.

(b) The four red simplices in the middle cannot be matched in a way simple enough to make the further steps trivial.

Figure 5.2: Two problematic situations that can arise. Non-0-monochromatic simplices are grayed out. A possible matching of neighboring simplices is indicated by a surrounding border and lighter strokes separating them, whereas matchings of non-neighboring simplices are indicated by colors. While other matchings are possible, none will be perfect.

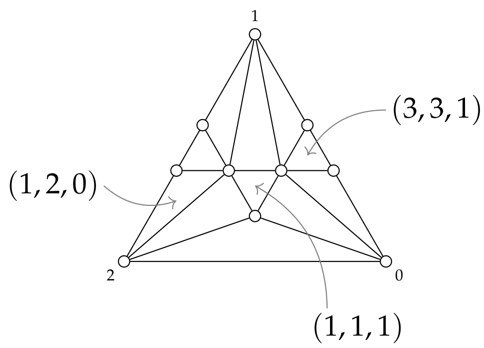


Figure 5.3: Examples of the naming convention in the subdivision of one simplex.

of neighboring 0-monochromatic n-simplices, which we already know how to deal with (cf. Figure 4.4 on page 34).

Unfortunately, this naïve approach does not work by itself in many cases. Even in two dimensions, special cases occur, as they do in Figure 5.2. We will not be able to give full solutions to these problems, but begin to develop terminology and possible directions to head to.

In analogy to Proposition 2.5 on page 12, we can describe every n-simplex in our subdivision by an $[n]$-tuple with values from $[3]$ (or equivalently a map $\alpha\colon [n] \to [3]$), where 0 and 1 have their usual meaning of outer and inner vertices, whereas 2 and 3 indicate that the vertex with the corresponding name is chosen to be in one of the two edge expansions. An example of this notation is given in Figure 5.3.

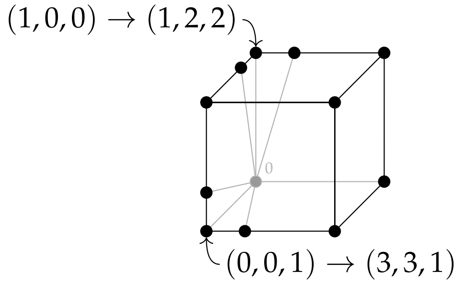


Figure 5.4: Neighboring relations of the global subdivision in two dimensions as a graph. In the hypercube of dimension three, two vertices at unit vectors have been replaced by hypercubes of dimension two minus their origin.

Obviously not all of these combinations are valid. Let us assume that the names that are flipped are 0 and n (if we look at the m-th simplex of an abstract simplicial path, this is equivalent to $C_{m-1} = 0$ and $C_m = n$), making their opposite faces the ones that get subdivided by the edge expansion. Then the set of all n-simplices can be written as

$$\begin{aligned}
&(\ \ ([1]^{[n]} - (0,\dots,0)) && \text{(basic chromatic subdivision)} \\
&\cup \{1\} \times \{0,2\}^{[n]-0} && \text{(edge expansion of } \sigma_{m-1,m}\text{)} \\
&\cup \{0,3\}^{[n-1]} \times \{1\} && \text{(edge expansion of } \sigma_{m,m+1}\text{)} \\
&)\setminus\{(1,0,\dots,0),(0,\dots,0,1)\} && \text{(these get subdivided by the edge expansions)}
\end{aligned}$$

We note that there are $2^{n+1} - 3 + 2^n - 1 + 2^n - 1 = 2^{n+2} - 5$ n-simplices in the subdivision of each original n-simplex.

Sadly, the neighboring conditions are not as easy as before, which makes an approach like the exhaustive expansion much more difficult. As with the basic chromatic subdivision, two given simplices share an $(n-1)$-face when their corresponding tuples differ in only one position. But because we now have a more difficult set, simply changing the tuple at one position does not necessarily yield a valid simplex again. Considered as a graph, we earlier dealt with a hypercube minus the origin for the basic chromatic subdivision. Now, we have to replace the entry and exit simplices $(1,0,\dots,0)$ and $(0,\dots,0,1)$ with hypercubes of one dimension lower (again minus their origin). An example can be seen in Figure 5.4.

6 Conclusion, Outlook

We have seen how to construct a chromatic subdivision of a single n-simplex that contains n-simplices-monochromatic, but equally many of the two possible orientations, in Chapter 3. These n-simplices were matched by so called atomic simplex paths that we continued to subdivide in two different ways in Sections 4.1 and 4.2, ending up without any n-simplices-monochromatic.

Both algorithms to subdivide the simplex paths had a similar structure: They both worked in iterations to modify the input path locally, where one iteration of the one cuts the input path into several shorter ones, while one iteration of the other lowers the maximal height of the input path. Both had to deal with many corner cases, and while the first one that we described in Section 4.1 is more efficient most of the time in terms of how many subdivisions and thus rounds it needs, that is $\mathcal{O}(n)$ compared to $\mathcal{O}(\ell^{n-1})$, the one described in Section 4.2 is easier to understand as a whole and more convincingly reveals that it really works in every case.

Finally, we saw that another approach might be possible as well which would not modify the input path locally, but subdivide it globally in more or less the same way. This approach would promise an even easier and more convincing way to eliminate the monochromaticities of atomic simplex paths. However, issues remain with that approach that could be approached in future work.

Furthermore, we note again that the overall strategy of first constructing monochromatic simplices of opposite orientation and then matching them with simplex paths has its limits in terms of round complexity set by the difficulty of finding these simplex paths without intersections. If the global difficulty of $\mathcal{O}(n^{q+3})$ shown in [ACHP13] was to be seriously decreased, a totally new strategy has to be developed that does not rely on simplex paths.

References

[ABND+90] H. Attiya, A. Bar-Noy, D. Dolev, D. Peleg, R. Reischuk. "Renaming in an Asynchronous Environment." In: *J. ACM* 37.3 (July 1990), pp. 524–548. DOI: 10.1145/79147.79158.

[ACHP13] H. Attiya, A. Castañeda, M. Herlihy, A. Paz. "Upper Bound on the Complexity of Solving Hard Renaming." In: *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing*. PODC '13. Montréal, Québec, Canada: ACM, 2013, pp. 190–199. DOI: 10.1145/2484239.2484251.

[BG93] E. Borowsky, E. Gafni. "Immediate Atomic Snapshots and Fast Renaming." In: *Proceedings of the Twelfth Annual ACM Symposium on Principles of Distributed Computing*. PODC '93. Ithaca, New York, USA: ACM, 1993, pp. 41–51. DOI: 10.1145/164051.164056.

[CR08] A. Castañeda, S. Rajsbaum. "New Combinatorial Topology Upper and Lower Bounds for Renaming." In: *Proceedings of the Twenty-seventh ACM Symposium on Principles of Distributed Computing*. PODC '08. Toronto, Canada: ACM, 2008, pp. 295–304. DOI: 10.1145/1400751.1400791.

[CR10] A. Castañeda, S. Rajsbaum. "New Combinatorial Topology Bounds for Renaming: The Lower Bound." English. In: *Distributed Computing* 22.5-6 (2010), pp. 287–301. DOI: 10.1007/s00446-010-0108-2.

[CR12] A. Castañeda, S. Rajsbaum. "New Combinatorial Topology Bounds for Renaming: The Upper Bound." In: *J. ACM* 59.1 (Mar. 2012), 3:1–3:49. DOI: 10.1145/2108242.2108245.

[DD99] J. B. Dence, T. P. Dence. *Elements of the Theory of Numbers*. Academic Press, 1999.

[Dic19] L. E. Dickson. *History of the Theory of Numbers*. Carnegie Institution of Washington, 1919.

[FLP85] M. J. Fischer, N. A. Lynch, M. S. Paterson. "Impossibility of Distributed Consensus with One Faulty Process." In: *J. ACM* 32.2 (Apr. 1985), pp. 374–382. DOI: 10.1145/3149.214121.

[GRH06] E. Gafni, S. Rajsbaum, M. Herlihy. "Subconsensus Tasks: Renaming is Weaker Than Set Agreement." In: *Proceedings of the 20th International Conference on Distributed Computing*. DISC'06. Stockholm, Sweden: Springer-Verlag, 2006, pp. 329–338. DOI: 10.1007/11864219_23.

[HKR13] M. Herlihy, D. N. Kozlov, S. Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Morgan Kaufmann, 2013. DOI: 10.1016/B978-0-12-404578-1.00017-6.

[HS99] M. Herlihy, N. Shavit. "The Topological Structure of Asynchronous Computability." In: *J. ACM* 46.6 (Nov. 1999), pp. 858–923. DOI: 10.1145/331524.331529.

[Kar75] R. M. Karp. "On the computational complexity of combinatorial problems." In: *Networks* 5 (1975), pp. 45–68.

[Koz08] D. N. Kozlov. *Combinatorial Algebraic Topology*. Vol. 21. Algorithms and computation in mathematics. Springer, 2008. DOI: 10.1007/978-3-540-71962-5.

[Koz12] D. N. Kozlov. "Chromatic Subdivision of a Simplicial Complex." In: *Homology, Homotopy and Applications* 14.2 (2012), pp. 197–209. DOI: 10.4310/HHA.2012.v14.n2.a12.

[Koz15] D. N. Kozlov. "Weak symmetry breaking and abstract simplex paths." In: *Mathematical Structures in Computer Science* FirstView (Feb. 2015), pp. 1–31. DOI: 10.1017/S0960129514000085.

[Mun84] J. R. Munkres. *Elements of Algebraic Topology*. Menlo Park, CA: Addison-Wesley, 1984.

[PSL80] M. Pease, R. Shostak, L. Lamport. "Reaching Agreement in the Presence of Faults." In: *J. ACM* 27.2 (Apr. 1980), pp. 228–234. DOI: 10.1145/322186.322188.

[SZ00] M. Saks, F. Zaharoglou. "Wait-Free k-Set Agreement is Impossible: The Topology of Public Knowledge." In: *SIAM J. Comput.* 29.5 (Mar. 2000), pp. 1449–1483. DOI: 10.1137/S0097539796307698.