Combinatorial Topology and Distributed Computing

Maurice Herlihy Dmitry Kozlov Sergio Rajsbaum

February 22, 2011

Contents

Part I: Undergraduate Course

Chapter 3

Manifolds, Impossibility, and Separation

Theoretical distributed computing is primarily concerned with classifying which tasks can, and which cannot be solved using given primitives. In this chapter, we present a simple impossibility result, showing that one cannot construct a protocol for the *set agreement* in the *round-by-round immediate snapshot* model of computation. We also show that set agreement is strictly stronger than the *weak symmetry-breaking* task: we can construct a protocol for weak symmetry-breaking from a "black box" that solves set agreement, but not vice-versa. We investigate these particular questions here because they can be addressed with a minimum of mathematical machinery. In later chapters, we generalize the techniques introduced here to address broader questions.

3.1 Manifold Complexes

We say, that a pure abstract simplicial complex of dimension n is *strongly connected* if any two n-simplices can be joined by a "chain" of n-simplices in which each pair of neighboring simplices has a common $(n-1)$-dimensional face.

Definition 3.1.1. A pure abstract simplicial complex $\mathcal{M}$ of dimension n is called a *pseudo-manifold with boundary* if it is strongly connected, and each $(n-1)$-simplex in $\mathcal{M}$ is contained in precisely one or two n-simplices.

An $(n-1)$ simplex in $\mathcal{M}$ is an *interior* simplex if it is contained in exactly two n-simplices, and a *boundary* simplex if it is contained in exactly one.

The *boundary* subcomplex of $\mathcal{M}$, denoted $\partial\,\mathcal{M}$, is the subcomplex consisting of all the simplices contained in its boundary $(n-1)$-simplices. For brevity, we will use "pseudomanifold" as an abbreviation for "pseudomanifold with boundary".

maurice: Do we want ∂ or Bd for boundary complex? If we do homology in later chapters would ∂ be confusing here?

The conditions in Definition 3.1.1 can be strengthened as follows.

Definition 3.1.2. Assume $\mathcal{M}$ is a pure abstract simplicial complex $\mathcal{M}$ of dimension n.

(1) $\mathcal{M}$ is called a *simplicial manifold* if the geometric realization of the link of every simplex σ is homeomorphic to a sphere of dimension $n-1-\dim\sigma$;

(2) $\mathcal{M}$ is called a *simplicial manifold with boundary if the geometric realization of the link of every simplex σ is either homeomorphic to a sphere or to a closed ball, in each case of dimension $n-1-\dim\sigma$.*

Note that in the special case when $\dim\sigma = n-1$ we have $n-1-\dim\sigma = 0$. The 0-dimensional sphere consists of two points, whereas the 0-dimensional ball consists of one point, so conditions of (1) and (2) of Definition 3.1.2 specialize precisely to the conditions of Definition 3.1.1. For brevity, we shall frequently omit the word "simplicial" and just write manifold, or manifold with boundary.

There is also the following standard topological notion.

Definition 3.1.3. Assume X is an arbitrary Hausdorff[1] topological space.

(1) X is called a *topological manifold* of dimension n, if every point of X has a neighborhood homeomorphic to an open ball of dimension n;

(2) X is called a *topological manifold with boundary of dimension n, if every point of X has a neighborhood homeomorphic to an open subset of Euclidean half-space:*

$$\mathbb{R}^n_+ = \{(x_1,\ldots,x_n)\in\mathbb{R}^n : x_n \geq 0\}.$$

The interior of X, denoted $\operatorname{Int} X$, is the set of points in X which have neighborhoods homeomorphic to an open ball of dimension n. The boundary of X, denoted $\partial\,X$, is the complement of $\operatorname{Int} X$ in X. The boundary points can be characterized as those points which land on the boundary hyperplane

[1] This is a technical condition from point-set topology, meaning every two points can be separated by disjoint open neighborhoods; it is needed to avoid all sorts perverse examples, and is always satisfied in the context of this book.

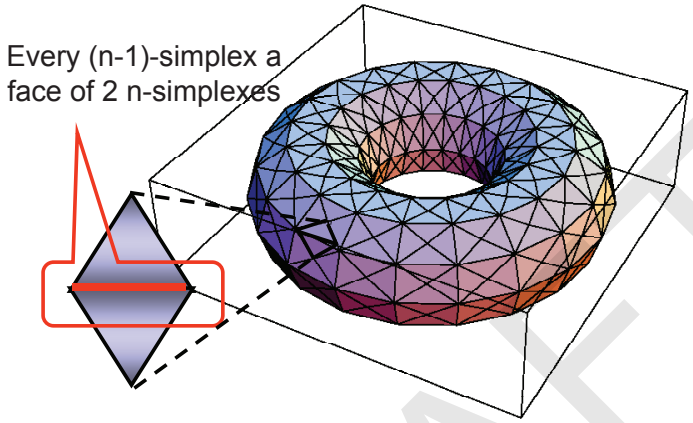

Figure 3.1: A simplicial manifold

$x_n = 0$ of $\mathbb{R}^n_+$ in their respective neighborhoods. If X is a manifold of dimension n with boundary, then $\operatorname{Int} X$ is a manifold of dimension n, and $\operatorname{Int} X$ is a manifold of dimension $n-1$.

We note that if $\mathcal{M}$ is a simplicial manifold with boundary, then its geometric realization is a topological manifold with boundary of the same dimension; moreover, the geometric realization of the boundary of $\mathcal{M}$ is precisely the boundary of $|\mathcal{M}|$. Figure 3.1 shows a 2-dimensional manifold (with an empty boundary complex). As you can see, a 2-dimensional manifold is a kind of a discrete approximation to a surface.

On the other hand, the geometric realization of the pseudomanifold does not have to be a manifold. Perhaps the simplest example is obtained if we take a simplicial 2-dimensional sphere, and then glue together the north and south poles,[2] see Figure ??. This is also called the *pinched torus*. Clearly,

[2]We assume that the poles are vertices of the simplicial complex, and that the mesh is fine enough, so that even after that gluing we still have a simplicial complex.

the condition of being a manifold fails at the glued poles, but the condition of being a pseudomanifold is still satisfied, since it is a condition for edges and triangles, and is untouched by vertices being glued together.

Few of the abstract simplicial complexes that arise naturally in the study of distributed computing are simplicial manifolds with boundary. We study them anyway primarily because they provide an excellent way to approach more complicated models. Manifolds have nice combinatorial properties not shared with more general classes of complexes. Later, we will see how to generalize the techniques developed here to other, more natural models of computation.

3.2 Immediate Snapshots

The *immediate snapshot* model is a simplified model of computation whose protocol complexes are manifolds with boundary. These executions are constrained, in the sense that they encompass only a subset of the interleavings possible in an asynchronous model. Nevertheless, any *impossibility* results that we prove for a restricted set of interleavings are valid for the less restricted model. It is easy to see why: solving a task in a distributed system means that the outputs should be valid in *every* execution. So if we can show a subset of executions where no valid decision is possible, then no valid decision is possible in general. Another way to formulate this observation is to imagine that executions are chosen by an "adversary" who always chooses the worst set of executions.

Consider an asynchronous system where $n+1$ processes share an $(n+1)$-element array m. When process P_i is scheduled to run, it writes its state to $m[i]$, and then atomically reads the entire array. We call such an atomic read a *snapshot*, and its result a *view*. For simplicity, we restrict how these steps can be interleaved. Each execution is divided into a sequence of *phases*. In each phase, select a set of processes that have not yet taken a step. All processes in that set simultaneously write, and then they simultaneously take a memory snapshot. Phases proceed until every process has been scheduled exactly once. Because each snapshot is scheduled immediately after the preceding write, we call this the *immediate snapshot* model.

Fig. 3.2 shows three examples of immediate snapshot executions. In each execution, there are three processes, P, Q, and R, where time runs from top to bottom. The bottom line of each table shows the result of each process's snapshot. For example, "PQ?" means the snapshot observed values written by P and Q, but not R. In the first execution, the processes

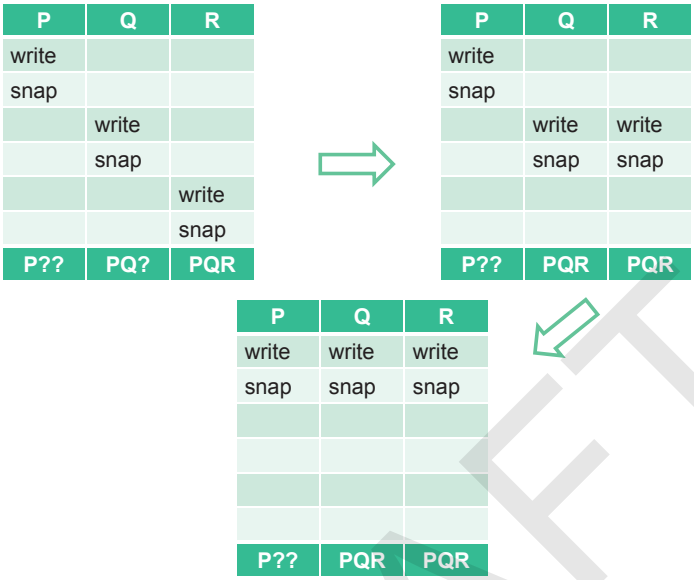

Figure 3.2: Immediate snapshot executions

are scheduled in distinct phases: first P, then Q, then R. In the second execution, as indicated by the curved arrow, we merge R and Q's phases. This perturbation changes the view of only process: Q's view changes, but P and R's views do not. In the third execution, we move both Q and R's phases to be simultaneous with P's. This perturbation, too, changes the view of only process: P's view changes, but Q and R's views do not.

dmitry: there is no curved arrow in the picture

Now let us shift our attention from the operational model of schedules to the combinatorial model of vertices, simplices, and complexes. Combinatorially, the observation that we can repeatedly perturb schedules so that only one process's view changes each time strongly suggests that the complex generated by all immediate snapshot schedules may be a manifold.

Each process's view at the end of an immediate snapshot execution is a face of the input simplex determined by which processes participated in the same or earlier phases. For input n-simplex σ, the set of immediate snapshot executions defines a subdivision of σ, called the *standard chromatic subdivision*, denoted $\mathrm{Ch}\,\sigma$. Each vertex in this subdivision is a pair $\langle P_i, \sigma_i \rangle$,

dmitry: when is the subdivision part proved?

maurice: we should prove subdivision in an appendix. I think there's a proof in the literature, but I need to check

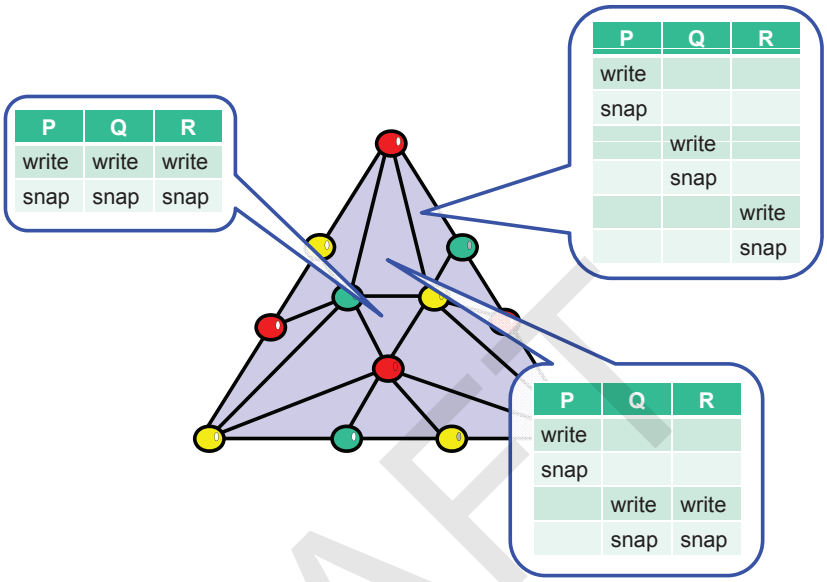

Figure 3.3: Three-process immediate snapshot complex

where P_i is the process taking the steps, and σ_i, the result of its snapshot, is a face of the input simplex σ. Each simplex in the protocol complex satisfies the following properties.

Property 3.2.1. Each process's write appears in its own view: $P_i \in \mathrm{id}(\sigma_i)$.

dmitry: include argument showing the equivalence with the original def

Property 3.2.2. Snapshots are ordered: for $0 \leq i, j \leq n$, either $\sigma_i \subseteq \sigma_j$ or vice-versa.

Property 3.2.3. Each snapshot is ordered immediately after its write: for $0 \leq i, j \leq n$, if $P_i \in \mathrm{id}(\sigma_j)$, then $\sigma_i \subseteq \sigma_j$.

Fig. 3.3 shows the standard chromatic subdivision of an input simplex for three processes, highlighting the simplices corresponding to the schedules shown in Fig. 3.2. Informally, we can see that this complex is a manifold, although such a claim requires proof.

maurice: Fig 3.2 should be redrawn

Recall, that whenever $\sigma = \{\vec{s}_0, \ldots, \vec{s}_n\}$ is an input simplex, where $\mathrm{id}(\vec{s}_i) = P_i$, we used $\mathrm{Face}_i\, \sigma = \sigma \setminus \{\vec{s}_i\}$ to denote the face of σ not labeled with P_i.

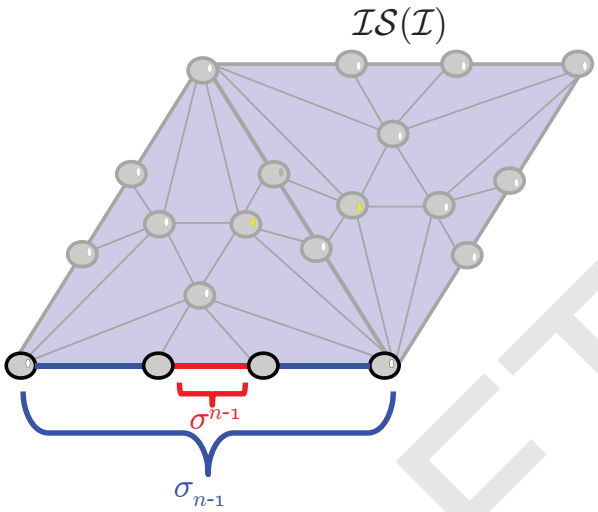

Figure 3.4: Proof of Theorem 3.2.4: Case 1

Furthermore, we let $\mathrm{Ch}\,\mathcal{I}$ denote the simplicial complex obtained from $\mathcal{I}$ by replacing every simplex by its chromatic subdivision.

Theorem 3.2.4. If the input complex $\mathcal{I}$ is a manifold, so is the immediate snapshot protocol complex $\mathrm{Ch}\,\mathcal{I}$.

dmitry: include the proof of strong connectedness and check other links, though this is still not enough for the subdivision part

Proof. Let σ^{n-1} be an $(n-1)$-simplex of $\mathrm{Ch}\,\sigma$. For ease of presentation, reindex the process IDs so that

$$\sigma^{n-1} = \{\langle P_0, \sigma_0\rangle, \ldots, \langle P_{n-1}, \sigma_{n-1}\rangle\} \text{ where } \sigma_i \subseteq \sigma_{i+1} \text{ for } 0 \leq i < n. \tag{3.2.1}$$

We must show that σ^{n-1} is a face of exactly one or two n-simplices.

The simplex σ_{n-1} has dimension either $n-1$ or n. If it has dimension exactly $n-1$, then σ_{n-1} is either an internal $(n-1)$-simplex of $\mathcal{I}$ or it is a boundary simplex of $\mathcal{I}$. There are three cases to consider.

- If σ_{n-1} is a boundary $(n-1)$-simplex of $\mathcal{I}$, then because $\mathcal{I}$ is a manifold, there is exactly one n-simplex σ that contains σ_{n-1}. There is exactly

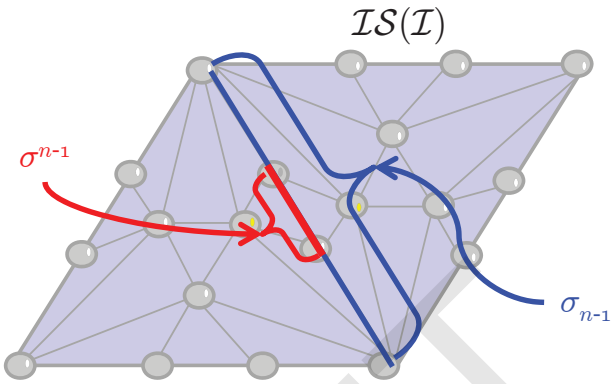

Figure 3.5: Proof of Theorem 3.2.4: Case 2

one n-simplex of $\mathrm{Ch}\,\mathcal{I}$, $\{\langle P_n, \sigma\rangle\} \cup \sigma_{n-1}$, that contains σ_{n-1}. (See Fig. 3.2.)

- If σ_{n-1} is an internal $(n-1)$-simplex of $\mathcal{I}$, then because $\mathcal{I}$ is a manifold, then there are exactly two n-simplices σ and σ' of $\mathcal{I}$ that contain σ_{n-1}. There are exactly two n-simplices of $\mathrm{Ch}\,\mathcal{I}$, $\{\langle P_n, \sigma\rangle\} \cup \sigma_{n-1}$, and $\{\langle P_n, \sigma'\rangle\} \cup \sigma_{n-1}$, that contain σ_{n-1}. (See Fig. 3.2.)

- If σ_{n-1} is an n-simplex of $\mathcal{I}$, suppose that P_n is in $\mathrm{id}(\sigma_\ell)$ but not $\mathrm{id}(\sigma_{\ell-1})$, for ℓ, where we understand σ_{-1} to be the empty set. Note that $\sigma_{\ell-1} \subset \sigma_\ell$, implying that $P_\ell \notin \mathrm{id}(\sigma_{\ell-1})$. For what values of σ_n is $\{\langle P_n, \sigma_n\rangle\} \cup \sigma^{n-1}$ an n-simplex of $\mathrm{Ch}\,\sigma$? $P_n \in \mathrm{id}(\sigma_n)$, hence $\vec{s}_n \in \sigma_n$. By Property 3.2.1,

$$\sigma_{\ell-1} \cup \{\vec{s}_n\} \subseteq \sigma_n \subseteq \sigma_\ell.$$

If

$$\sigma_{\ell-1} \cup \{\vec{s}_n\} \subset \sigma_n,$$

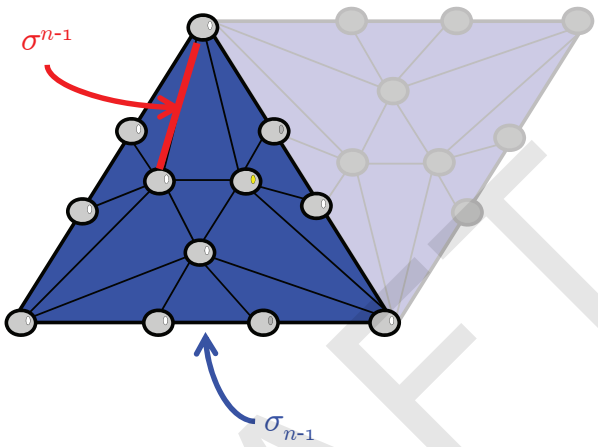

Figure 3.6: Proof of Theorem 3.2.4: Case 3

there is a process id P_i in $\mathrm{id}(\sigma_n) \setminus \mathrm{id}(\sigma_{\ell-1}) \cup \{\vec{s}_n\})$. By hypothesis, σ_i is a subset of σ_ℓ but not $\sigma_{\ell-1}$. Because these simplices are ordered by inclusion, it follows that $\sigma_i = \sigma_\ell$, and therefore $\sigma_n = \sigma_\ell$. It follows that σ_n can assume exactly two distinct values: $\sigma_{\ell-1} \cup \{\vec{s}_n\}$ or σ_ℓ, implying that σ^{n-1} is contained in exactly two n-simplices.

□

The proof of Theorem 3.2.4 uncovered an interesting fact about the immediate snapshot complex: the boundary $(n-1)$-simplices are precisely those simplices where one process does not appear in any of the others' views.

Corollary 3.2.5.

$$\partial\,\mathrm{Ch}\,\mathcal{I} = \mathrm{Ch}\,\partial\,\mathcal{I}.$$

3.3 Manifold Protocols

dmitry: here we need to decide whether we go with manifold or with pseudo-manifold

The immediate snapshot protocol is an example of a *manifold protocol*. A protocol $\mathcal{M}(\cdot)$ is a manifold protocol if it satisfies the following properties.

Property 3.3.1. If $\mathcal{I}$ is a manifold, so is $\mathcal{M}(\mathcal{I})$.

Property 3.3.2. $\mathcal{M}(\partial\mathcal{I}) = \partial\,\mathcal{M}(\mathcal{I})$.

Manifold protocols are closed under composition.

Many of the protocols we consider are the result of composing multiple instances of simpler protocol executions, which we call *round-by-round* composition. A *round operator* is the carrier map that represents one computational "step" by each process. Here, an immediate snapshot protocol is a round operator, and a *round-by-round immediate snapshot* protocol, is the result of composing multiple immediate snapshot protocols, each using a separate shared array.

3.4 Set Agreement

Recall that in the *k-Set Agreement* task, each process starts with a private input value, communicates with the others, and then halts after choosing a private output value. Each process is required to choose some process's input, and at most k distinct values may be chosen. For brevity we use *set agreement* as shorthand for $(n+1)$-process n-set agreement. We now demonstrate that no manifold protocol can solve set agreement.

First, we need some simple combinatorial lemmas. Recall that a graph is a 1-dimensional complex given by a set of vertices V and a set of edges E. The *degree* of a vertex, $\deg(\vec{v})$, is the number of edges adjacent to $\vec{v}$.

Lemma 3.4.1. In any graph $G = (V, E)$, the sum of the degrees of the vertices is twice the number of edges:

$$2|E| = \sum_{\vec{v}\in V} \deg(\vec{v}).$$

Proof. Each edge $e = \{\vec{v}_0, \vec{v}_1\}$ adds one to the degree of $\vec{v}_0$ and one to the degree of $\vec{v}_1$, contributing two to the sum of the degrees. □

Corollary 3.4.2. Any graph has an even number of vertices of odd degree.

As noted earlier, an $(n+1)$-*coloring* of a complex $\mathcal{C}$ is a map $\chi : \mathcal{C} \to \Delta^n$, where Δ^n is an n-simplex. We say that χ sends a simplex σ *onto* Δ if every vertex in Δ^n is the image of a vertex in σ.

Suppose we color an n-simplex $\sigma = (\vec{s}_0, \ldots, \vec{s}_n)$ with $n+1$ distinct colors, and then color a subdivision $\operatorname{Div}\sigma$ so that each vertex is colored with a color from its carrier. It turns out that at least one simplex in the subdivision must have $n+1$ distinct colors.

Lemma 3.4.3 (Sperner's Lemma).
Let $\operatorname{Div}\Delta^n$ be a subdivision of Δ, and let $\chi : \operatorname{Div}\Delta^n \to \Delta$ be a simplicial map such that $\chi(\vec{v}) \in \operatorname{Car}(\vec{v}, \Delta^n)$. The map χ maps an odd number of simplices of $\operatorname{Div}\Delta^n$ onto Δ^n.

Proof. We argue by induction on the dimension n. When n is zero, the complexes Δ^0 and $\operatorname{Div}\Delta^0$ are the same, χ is the identity map, and the claim is trivial.

Assume the result for $n-1$. By the induction hypothesis,

$$\chi : \operatorname{Div}\operatorname{Face}_i \Delta^n \to \operatorname{Face}_i \Delta^n$$

sends an odd number of $(n-1)$-simplices to $\operatorname{Face}_n \Delta^n$. The subdivison $\operatorname{Div}\Delta^n$ is a manifold, so by Lemma 3.4.4, χ sends an odd number of n-simplices of $\operatorname{Div}\Delta^n$ onto Δ^n. □

dmitry: I am confused why the graph edges on Figure 3.8 go over two boundary edges of the triangle

dmitry: What if we go over the n-1 skeleton instead

Lemma 3.4.4 (Sperner's Lemma for Pseudomanifolds).
Let $\Delta^n = \left\{\vec{d}_0, \ldots, \vec{d}_n\right\}$ be an n-simplex, and $\operatorname{Face}_i \Delta^n$ the $(n-1)$-face of Δ^n that contains every vertex except $\vec{d}_i$. Let $\mathcal{M}$ be an n-dimensional manifold, $\mathcal{B}$ its boundary complex, and $\chi : \mathcal{M} \to \Delta^n$ an $(n+1)$-coloring. If χ sends an odd number of $(n-1)$-simplices of $\mathcal{B}$ onto $\operatorname{Face}_n \Delta^n$, then χ sends an odd number of n-simplices of $\mathcal{M}$ onto Δ^n.

Proof. Define G to be the graph whose vertices are indexed by the n-simplices of $\mathcal{M}$, with the addition of one more "external" vertex $\vec{e}$. There is an edge between two vertices if their corresponding simplices share a common $(n-1)$-face such that χ sends that face onto $\operatorname{Face}_n \Delta^n$. There is also an edge from the external vertex $\vec{e}$ to every n-simplex σ such that σ has an $(n-1)$-face in $\mathcal{B}$, and χ sends that face onto $\{0, \ldots, n-1\}$. (See Figs. 3.7 and 3.8.)

Let $\vec{v}_\sigma$ be the vertex corresponding to the n-simplex σ. We claim that $\vec{v}_\sigma$ has an even degree if and only if χ maps σ onto $\{0, \ldots, n-1\}$. There are three cases to consider. First, if χ does not map any $(n-1)$-face of σ onto $\{0, \ldots, n-1\}$, then $\vec{v}_\sigma$ has degree 0. Suppose χ does map a $(n-1)$-face of σ onto $\{0, \ldots, n-1\}$. Second, if χ maps the remaining vertex to a value in $\{0, \ldots, n-1\}$, then χ maps exactly two $(n-1)$-faces of σ onto

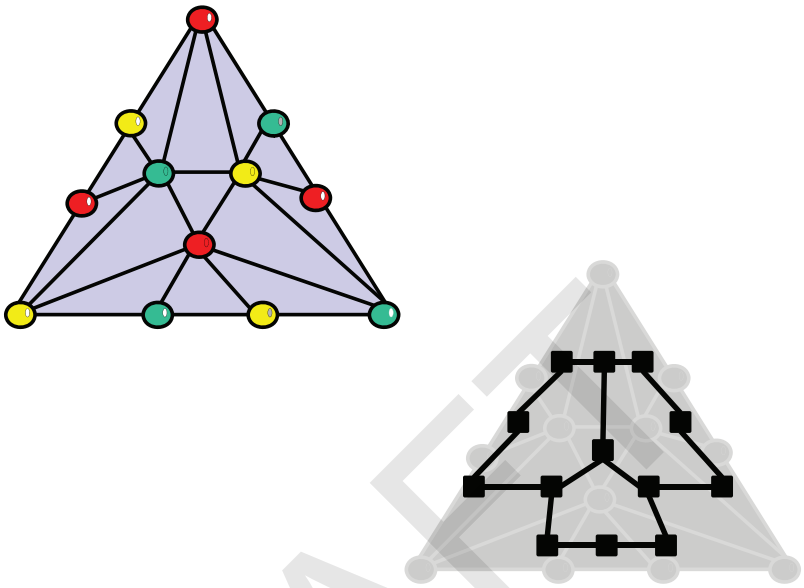

Figure 3.7: Dual Graph

$\{0, \ldots, n-1\}$. Each such face corresponds to an edge linking $\vec{v}_\sigma$ either to a neighboring n-simplex (for internal faces) or to the external vertex $\vec{e}$ (for boundary faces). It follows that $\vec{v}_\sigma$ has degree 2. Finally, if χ maps the remaining vertex to n, then χ maps exactly one $(n-1)$-face of σ onto $\{0, \ldots, n-1\}$, implying that $\vec{v}_\sigma$ has degree 1.

Moreover, $\vec{e}$ has odd degree. By hypothesis, χ sends an odd number of boundary $(n-1)$-simplices onto $\{0, \ldots, n-1\}$, producing an odd number of edges at $\vec{e}$.

By Lemma 3.4.1 G has an even number of vertices of odd degree. Since the external node $\vec{e}$ has odd degree, there must be an odd number of other vertices $\vec{v}_\sigma$ with odd degree. Each of these vertices corresponds to an n-simplex that χ maps onto Δ^n. □

Theorem 3.4.5. Any manifold protocol $\mathcal{M}$ where each process halts with an input value has an odd number of executions in which the processes choose $n+1$ distinct inputs.

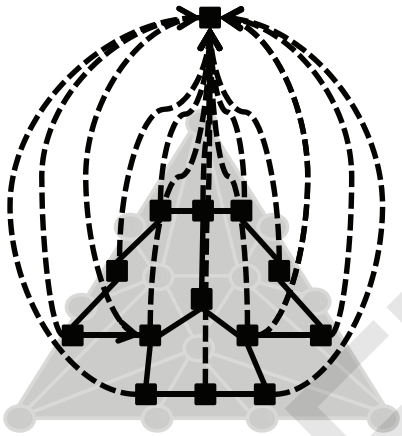

Figure 3.8: Dual Graph of Subdivision with "external" node

Proof. The input complex consists of a single simplex $\sigma = \{\vec{s}_0, \ldots, \vec{s}_n\}$, where $\vec{s}_i$ is labeled with process ID P_i: $\mathrm{id}(\vec{d}_i) = P_i$. The protocol $\mathcal{M}$ solves set agreement if and only if there is a simplicial map $\delta : \mathcal{M}(\sigma) \rightarrow \sigma$ sending each vertex of the protocol complex to the input vertex labeled with the value chosen. To solve set agreement, the coloring δ cannot map any n-simplex onto σ. By way of contradiction, we will show that δ must map an odd number of n-simplices onto σ.

We argue by induction. When n is zero, there is only one process, one possible execution, and one possible value to choose.

Assume the claim for dimension $n-1$. Because $\mathcal{M}$ is a manifold task,

$$\partial\,\mathcal{M}(\sigma) = \bigcup_{P_i \in \mathrm{id}(\sigma)} \mathcal{M}(\mathrm{Face}_i\,\sigma)$$

Process P_i can choose P_j's input only in executions where P_j takes a step, so χ sends no $(n-1)$-simplices of $\mathcal{M}(\mathrm{Face}_i\,\sigma)$ onto $\mathrm{Face}_n\,\sigma$ for $i \neq n$. By the induction hypothesis, $\chi : \mathcal{M}(\mathrm{Face}_n\,\sigma) \rightarrow \mathrm{Face}_n\,\sigma$ sends an odd number of

$(n-1)$-simplices onto $\mathrm{Face}_n\,\sigma$. In total, χ sends an odd number of $(n-1)$-simplices of $\partial\,\mathcal{M}(\sigma)$ onto $\mathrm{Face}_n\,\sigma$. By Lemma 3.4.4, χ sends an odd number of n-simplices of $\mathcal{M}(\sigma)$ onto σ. □

Theorem 3.4.5 implies that for any manifold protocol, any possible decision map has an odd, in particular non-zero, number of executions in which $n+1$ processes choose $n+1$ distinct values, contradicting the specification of the set agreement task.

Corollary 3.4.6. There is no protocol for set agreement in the round-by-round immediate snapshot model.

3.5 Anonymous Protocols

So far, we have assumed that there are $n+1$ processes with given unique IDs in the range $0,\ldots,n$. In practice, it is reasonable to assume that processes have unique IDs, but not that their IDs are taken from such a small name space. Instead, IDs are typically taken from a much larger name space, such as Unix process identifiers or Internet addresses, both 32-bit numbers. For impossibility results, the size of the name space is unimportant: any task that cannot be solved if IDs are taken from a small name space also cannot be solved if IDs are taken from a larger name space. For algorithms, however, it may be possible to abuse the small name space assumption to derive trivial protocols.

To rule out such spurious solutions, we say a protocol is *anonymous* if each process's decision value depends only on its inputs and on how its steps are interleaved with the others', but not on that process's ID. Consider a task $(\mathcal{I},\mathcal{O},\Delta)$. Formally, a protocol complex $\mathcal{P}(\mathcal{I})$ is *symmetric* if any permutation π of the process IDs induces a simplicial map $\pi:\mathcal{P}(\mathcal{I})\to\mathcal{P}(\mathcal{I})$, sending $\langle P_i,v\rangle \mapsto \langle\pi(P_i),v\rangle$. All complexes considered in this section are symmetric. Recall that the protocol has a decision function $\delta:\mathcal{P}(\mathcal{I})\to\mathcal{O}$, carried by Δ. The protocol is *anonymous* if relabeling process IDs, but leaving inputs or interleavings unchanged does not affect the processes' same output values. We can represent this relation by the following *commutative diagram*:

dmitry: add the condition that S_n acts on O as well

$$\begin{array}{ccc} \mathcal{P}(\mathcal{I}) & \xrightarrow{\delta} & \mathcal{O} \\ {\scriptstyle\pi}\downarrow & & \downarrow{\scriptstyle\pi} \\ \mathcal{P}(\mathcal{I}) & \xrightarrow{\delta} & \mathcal{O} \end{array}$$

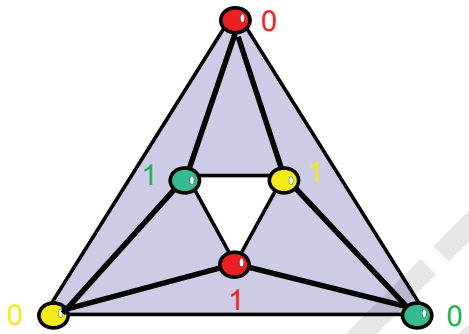

Figure 3.9: Output complex for Weak Symmetry-Breaking

Starting with a vertex in the upper left-hand corner, applying maps along both directed paths yields the same result in the lower right-hand corner.

3.6 Weak Symmetry-Breaking

maurice: Add process IDs to Fig 3.9

The *Weak Symmetry-Breaking* task ensures that if all processes participate, they can be divided into two non-empty groups. Specifically, in every execution in which all $n+1$ processes participate, at least one process decides *true* and at least one decides *false*. Figure 3.9 shows the output complex for 3-process weak symmetry-breaking: this complex is an *annulus*, which is a topological disk with a hole in the center.

maurice: define crosspolytope?

In general, the output complex for weak symmetry-breaking task with $n+1$ processes is an n-dimensional simplicial manifold with boundary. It can be constructed by taking the boundary of the $(n+1)$-dimensional crosspolytope and then removing a pair of opposite n-simplices. In particular, the

boundary of the output complex always consists of two simplicial $(n-1)$-spheres. Weak symmetry-breaking will turn out to be a useful building-block for constructing other protocols.

Formally, the weak symmetry-breaking task takes as input complex a single n-simplex and its faces, and the following output complex:

- each vertex $\langle P_i, b\rangle$ is labeled with a process ID and a Boolean value, and
- a set of vertices forms a simplex if (1) their process IDs are distinct, and (2) fewer than $n+1$ vertices have the same Boolean value.

If each process has a unique ID in the range $0, \ldots, 2n-1$, then weak symmetry-breaking has a trivial protocol: each process decides the parity of its ID. To rule out such uninteresting solutions, we require that any protocol that implements weak symmetry-breaking be anonymous.

3.7 Anonymous Set Agreement versus Weak Symmetry Breaking

dmitry: number sides of the triangles on Figure 3.10

We say a task $\mathcal{T}$ *implements* a task $\mathcal{S}$ in a particular model if one can construct a protocol for $\mathcal{S}$ using read-write memory and a finite number of "black boxes" solving $\mathcal{T}$. If $\mathcal{T}$ implements $\mathcal{S}$, but not vice-versa, then we say that $\mathcal{S}$ is *weaker* than $\mathcal{T}$. Otherwise, they are *equivalent*.

We consider two tasks, *anonymous* set agreement and weak symmetry-breaking in a round-by-round model. We will show that anonymous set agreement implements weak symmetry-breaking, but not vice-versa, so the latter is weaker than the former. This is an example of a *separation* result.

maurice: must define linearizable

Fig. 3.12 shows how set agreement implements weak symmetry-breaking. The processes share a set, initially empty, of process IDs (Line 1). This set provides a linearizable $\mathsf{put}(x)$ method that places x in the set, along with the ability to iterate over IDs in the set. An iteration yields every ID put in the set before the iteration started, and it may yield IDs put in the set while the iteration is in progress. This set might be implemented as a simple array. The processes also share an anonymous set agreement object (Line 2). This object provides a single $\mathsf{decide}(x)$ method, that runs an anonymous set agreement protocol with input x, and returns a decision value.

Each process calls the anonymous set agreement object's $\mathsf{decide}()$ method, using its own id as input (Line 4). It stores the result in the set (Line 5). The process then iterates through the set (Line 6), returning *true* if it finds its own ID in the set, and *false* otherwise.

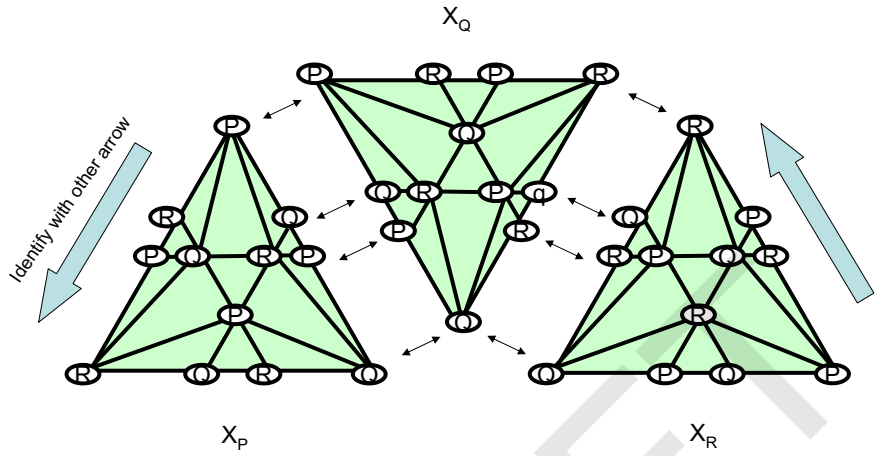

Figure 3.10: One-round Moebius task protocol complex for 3 processes.

Lemma 3.7.1. If all $n+1$ processes participate, some process decides *true*.

Proof. The process whose id is first to be put in the set will observe its own ID and return *true*. □

Lemma 3.7.2. If all $(n+1)$ processes participate, some process decides *false*.

Proof. If all $n+1$ processes decide *true*, then $n+1$ distinct inputs were chosen as decision values, violating the Set Agreement specification. □

Lemma 3.7.3. The protocol of Fig. 3.9 is anonymous.

Proof. No step of the protocol, including the anonymous set agreement subroutine, depends on any participants' process ID. □

Corollary 3.7.4. Anonymous set agreement implements weak symmetry-breaking.

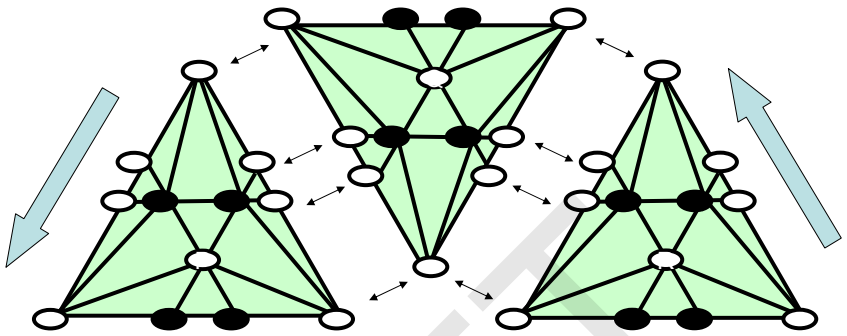

Figure 3.11: Weak Symmetry-breaking from one-round Moebius task protocol

```
1  Set<ID> output;          // set of IDs
2  anonSetAgree sa;         // anonymous set agreement
3  boolean choose(ID me) {
4    ID choice = sa.decide(me);
5    output.put(choice);
6    foreach (ID id in output) {
7      if (id == me) return true;
8    }
9    return true;
10 }
```

Figure 3.12: Implementing weak symmetry-breaking from set agreement

For the other direction, we wish to show that weak symmetry-breaking cannot implement set agreement. We will prove this claim indirectly, by constructing a manifold task that implements weak symmetry-breaking. If weak symmetry-breaking could implement set agreement, then we could replace the weak symmetry-breaking objects with their manifold task implementations, yielding a set agreement protocol, contradicting Theorem 3.4.5.

We introduce a new task $\mathcal{M}(\cdot)$, called the *Moebius* task. First, we construct the 2-dimensional Moebius task. As illustrated in Fig. 3.10, for each input 2-simplex σ, take three "copies" ξ_0, ξ_1, ξ_2 of $\mathrm{Ch}\,\sigma$. We call $\mathrm{Face}_i\,\xi_i$ the *external face* of ξ_i (even though it is technically a complex), and $Face_j\xi_i$, for $i \neq j$, the *internal faces*. We then identify (that is, "glue together") $\mathrm{Face}_1\,\xi_1$ and $\mathrm{Face}_1\,\xi_2$, $\mathrm{Face}_2\,\xi_0$ and $\mathrm{Face}_2\,\xi_2$, and $\mathrm{Face}_3\,\xi_0$ and $\mathrm{Face}_3\,\xi_1$. The resulting complex is a manifold. It is easy to check that $\mathcal{M}(\partial\,\sigma) = \partial\,\mathcal{M}(\sigma) = \mathrm{Ch}\,\partial\,\sigma$.

dmitry: mark these ξ's on the figure

Because this complex is a manifold, it cannot solve 2-set agreement. As illustrated in Fig. 3.11, however, it can solve weak symmetry-breaking. We color each vertex with black and white "pebbles" (that is, *true* or *false* values) as follows. For each central simplex of ξ_i, color each node black except for the one labeled with P_i. For the central simplex of each external face $\mathrm{Face}_i\,\xi_i$, color the central $(2N-2)$-simplex black. The rest are white. It is easy to check that (1) no 2-simplex is monochrome, and (2) the protocol is anonymous because the coloring on the boundary is symmetric. It follows that the 2-dimensional Moebius task *separates* weak symmetry-breaking and anonymous set agreement, in the sense that it can implement one, but not the other.

Now we generalize this construction to even dimensions. Let $n = 2N$. For each input n-simplex σ, take $n+1$ "copies" $\xi_0, \ldots, \xi_n$ of $\mathrm{Ch}\,\sigma$. As before, we call the complex $\mathrm{Face}_i\,\xi_i$ the *external face* of ξ_i and $Face_j\xi_i$, for $i \neq j$, the *internal faces*.

If U is a set of process IDs, the *rank* of an ID P_i is the number of IDs in U smaller than P_i. For each j, $0 \leq j \leq n$, let $\pi_j : \Pi \setminus \{j\} \to \Pi \setminus \{j\}$ be the map sending the ID with rank r in $\Pi \setminus \{j\}$ to the ID with rank $r + N \bmod 2N$.

For each i, and each $j \neq i, \pi_j(i)$, identify the internal face $\mathrm{Face}_j\,\xi_i$ with $\mathrm{Face}_j\,\xi_{\pi_j(i)}$. Because $\pi_j(i) \neq i$, $\pi_j(i) \neq j$, and $\pi_j(\pi_j(i)) = i$, each $(2N-1)$-simplex in each internal face lies in exactly two $(2N)$-simplices, so the result is a manifold. (This why this construction works only in even dimensions.)

Let σ be an input n-simplex. The Moebius task's carrier map carries each proper face τ of σ to $\mathrm{Ch}\,\tau$. It carries σ itself to all n-simplices of $\mathcal{M}(\sigma)$.

Theorem 3.7.5. The Moebius task cannot solve Set Agreement.

Proof. The one-round Moebius task is a manifold protocol, so composing the Moebius task with itself, with immediate snapshot, or with any other manifold task yields a manifold task. The claim follows from Theorem 3.4.5. □

To show this task solves weak symmetry breaking, we again color the edges with black and white pebbles so that no simplex is monochrome, and the coloring on the boundary is symmetric. For the central simplex of each ξ_i, color each node black except for the one labeled with P_i. For the central simplex of each external face ξ_{ii}, color the central $(2N-2)$-simplex black. The rest are white.

Every $(2N-1)$-simplex ξ in ξ_i intersects both a face, either internal or external, and a central $(2N-1)$-simplex. If ξ intersects an internal face, then the vertices on that face are white, and the vertices on the central simplex are black. If ξ intersects the internal face, then it intersects the white node of the central simplex of ξ_i, and a black node of the central simplex of ξ_{ii}.

Corollary 3.7.6. Set agreement implements weak symmetry breaking but not vice-versa.

The techniques studied here illustrate how combinatorial and algorithmic techniques complement one another: combinatorial techniques are often effective to prove impossibility, while algorithmic techniques are convenient to show that something is possible.

3.8 Chapter Notes

The immediate snapshot model is due to Borowsky and Gafni [6] and to Saks and Zaharoughu [25], who called them *block executions*. Borowsky and Gafni also showed that the round-by-round immediate snapshot model is equivalent to the standard read-write memory model.

The separation between weak symmetry-breaking and anonymous set agreement is adapted from Gafni, Rajsbaum, and Herlihy [13].

3.9 Exercises

Exercise 3.1. Count the number of simplices in $\mathrm{Ch}\,\sigma$, for an n-simplex σ.

Exercise 3.2. Count the number of simplices in the output complex for $(n+1)$-process weak symmetry-breaking.

Exercise 3.3. Compute the Euler characteristic of $\mathrm{Ch}\,\sigma$, for an n-simplex σ.

Exercise 3.4. Bridges of Königsberg problem. Hint: use reasoning similar to the proof of Lemma 3.4.1.

Exercise 3.5. Using read-write memory, implement the Set<ID> object used in Fig. 3.9. You may assume IDs are integers in the range $1, \ldots, N$, for some $N > n + 1$. Do not worry about efficiency.

Bibliography

[1] Yehuda Afek, Hagit Attiya, Danny Dolev, Eli Gafni, Michael Merritt, and Nir Shavit. Atomic snapshots of shared memory. *Journal of the ACM*, 40(4):873–890, 1993.

[2] Hagit Attiya, Amotz Bar-Noy, Danny Dolev, David Peleg, and Rudiger Reischuk. Renaming in an Asynchronous Environment. *Journal of the ACM*, July 1990.

[3] Hagit Attiya, Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Bounds on the time to reach agreement in the presence of timing uncertainty. *J. ACM*, 41:122–152, January 1994.

[4] Hagit Attiya and Jennifer Welch. *Distributed Computing Fundamentals, Simulations, and Advanced Topics Second Edition.*

[5] Ofer Biran, Shlomo Moran, and Shmuel Zaks. A combinatorial characterization of the distributed tasks which are solvable in the presence of one faulty processor. In *PODC '88: Proceedings of the seventh annual ACM Symposium on Principles of distributed computing*, pages 263–275, New York, NY, USA, 1988. ACM.

[6] Elizabeth Borowsky and Eli Gafni. Generalized FLP impossibility result for t-resilient asynchronous computations. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 91–100, New York, NY, USA, 1993. ACM.

[7] Elizabeth Borowsky and Eli Gafni. A Simple Algorithmically Reasoned Characterization of Wait-Free Computations (Extended Abstract). In *PODC '97: Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing*, pages 189–198, New York, NY, USA, 1997. ACM.

[8] S. Chaudhuri. Agreement Is Harder Than Consensus: Set Consensus Problems in totally asynchronous systems. In *Proceedings Of The Ninth Annual ACM Symosium On Principles of Distributed Computing*, pages 311–234, August 1990.

[9] Soma Chaudhuri, Maurice Herlihy, Nancy A. Lynch, and Mark R. Tuttle. A Tight Lower Bound for k-Set Agreement. In *In Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, pages 206–215, 1993.

[10] Carole Delporte-Gallet, Hugues Fauconnier, Rachid Guerraoui, and Andreas Tielmann. The Disagreement Power of an Adversary. In Idit Keidar, editor, *Distributed Computing*, volume 5805 of *Lecture Notes in Computer Science*, chapter 6, pages 8–21. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2009.

[11] M. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility Of Distributed Commit With One Faulty Process. *Journal of the ACM*, 32(2), April 1985.

[12] Eli Gafni and Elias Koutsoupias. Three-Processor Tasks Are Undecidable. *SIAM J. Comput.*, 28(3):970–983, 1999.

[13] Eli Gafni, Sergio Rajsbaum, and Maurice Herlihy. Subconsensus Tasks: Renaming Is Weaker Than Set Agreement. In *Distributed Computing, 20th International Symposium, Stockholm, Sweden, September 18-20, 2006, Proceedings(DISC)*, volume 4167 of *Lecture Notes in Computer Science*, pages 329–338. Springer, 2006.

[14] Maurice Herlihy and Sergio Rajsbaum. The decidability of distributed decision tasks (extended abstract). In *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 589–598, New York, NY, USA, 1997. ACM.

[15] Maurice Herlihy and Sergio Rajsbaum. The topology of shared-memory adversaries. In *Proceeding of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, PODC '10, pages 105–113, New York, NY, USA, 2010. ACM.

[16] Maurice Herlihy, Sergio Rajsbaum, and Mark Tuttle. An Axiomatic Approach to Computing the Connectivity of Synchronous and Asynchronous Systems. *Electron. Notes Theor. Comput. Sci.*, 230:79–102, 2009.

[17] Maurice Herlihy, Sergio Rajsbaum, and Mark R. Tuttle. Unifying synchronous and asynchronous message-passing models. In *PODC '98: Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, pages 133–142, New York, NY, USA, 1998. ACM.

[18] Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *J. ACM*, 46(6):858–923, 1999.

[19] Flavio Junqueira and Keith Marzullo. A framework for the design of dependent-failure algorithms: Research Articles. *Concurr. Comput. : Pract. Exper.*, 19(17):2255–2269, 2007.

[20] Flavio P. Junqueira and Keith Marzullo. Designing Algorithms for Dependent Process Failures. Technical report, 2003.

[21] Dimitry Kozlov. *Combinatorial Algebraic Topology*, volume 21 of *Algorithms and Computation in Mathematics*. Springer, 1 edition, October 2007.

[22] M. C. Loui and H. H. Abu-Amara. *Memory requirements for agreement among unreliable asynchronous processes*, volume 4, pages 163–183. JAI press, 1987.

[23] Yoram Moses and Sergio Rajsbaum. A Layered Analysis of Consensus. *SIAM J. Comput.*, 31:989–1021, April 2002.

[24] James Munkres. *Elements of Algebraic Topology*. Prentice Hall, 2 edition, January 1984.

[25] Michael Saks and Fotios Zaharoglou. Wait-Free k-Set Agreement is Impossible: The Topology of Public Knowledge. *SIAM Journal on Computing*, 29(5):1449–1483, 2000.

[26] Michael Saks and Fotios Zaharoglou. Wait-Free k-Set Agreement is Impossible: The Topology of Public Knowledge. *SIAM J. Comput.*, 29(5):1449–1483, 2000.

[27] Francis Sergeraert. The Computability Problem In Algebraic Topology. *Adv. Math*, 104:1–29, 1994.

[28] Edwin H. Spanier. *Algebraic topology*. Springer-Verlag, New York, 1981.

[29] John Stillwell. *Classical Topology and Combinatorial Group Theory*. Springer, 2nd edition, March 1993.