# Contents

1. **Introduction**  
   1.1 Decision Tasks 8  
   1.2 Communication 9  
   1.3 Failures 9  
   1.4 Timing 10  
      1.4.1 Processes and Protocols 10  
   1.5 Chapter Notes 12

2. **Elements of Combinatorial Topology** 13  
   2.1 Basic Concepts 13  
   2.2 Distributed Computing 15  
   2.3 Chapter Notes 16  
   2.4 Exercises 16

3. **Manifolds, Impossibility, and Separation** 17  
   3.1 Manifold Complexes 17  
   3.2 Immediate Snapshots 18  
   3.3 Manifold Protocols 23  
   3.4 Set Agreement 24  
   3.5 Anonymous Protocols 28  
   3.6 Weak Symmetry-Breaking 29  
   3.7 Anonymous Set Agreement versus Weak Symmetry Breaking 30  
   3.8 Chapter Notes 35  
   3.9 Exercises 35

4. **Connectivity** 37  
   4.1 Consensus and Path-Connectivity 37  
   4.2 Consensus in Asynchronous Read-Write Memory 39  
   4.3 Set Agreement and Connectivity in Higher Dimensions 43
4.4 Set Agreement and Read-Write memory ..................................... 49
  4.4.1 Critical States ................................................. 53
4.5 Chapter Notes ..................................................................... 54
4.6 Exercises ........................................................................... 54

5 Colorless Tasks ........................................................................ 57
  5.1 Pseudospheres ..................................................................... 58
  5.2 Wait-Free Read-Write Memory ............................................ 61
    5.2.1 Read-Write Protocols and Pseudospheres ...................... 61
    5.2.2 Necessary and Sufficient Conditions ............................ 62
  5.3 Read-Write Memory with $k$-Set Agreement ......................... 64
  5.4 Decidability ........................................................................ 66
    5.4.1 Loop Agreement .................................................. 66
    5.4.2 Read-Write Memory .............................................. 68
    5.4.3 Read-Write Memory .............................................. 68
  5.5 Chapter Notes ..................................................................... 69
  5.6 Exercises ........................................................................... 70
Part I: Undergraduate Course
Chapter 3

Manifolds, Impossibility, and Separation

Theoretical distributed computing is primarily concerned with classifying which tasks can (or cannot) be solved using which 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

An $n$-dimensional complex is graph-connected if any two $n$-simplexes can be joined by a “chain” of $n$-simplexes in which each pair of neighboring simplexes has a common $(n-1)$-dimensional face. A complex $M$ is a manifold with boundary (sometimes called a pseudo-manifold with boundary) if it is graph connected, and each $(n-1)$-simplex in $M$ is contained in precisely one or two $n$-simplexes. An $(n-1)$ simplex in $M$ is an interior simplex if it is contained in exactly two $n$-simplexes, and a boundary simplex if it is contained in exactly one. The boundary subcomplex of $M$, denoted $\partial M$, is the subcomplex induced by its boundary simplexes. For brevity, we will use “manifold” as an abbreviation for “manifold with boundary”.

Fig. 3.1 shows a 2-dimensional manifold (with an empty boundary com-
3.2 Immediate Snapshots

The immediate snapshot model is a simplified model of computation whose protocol complexes are manifolds. 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.
3.2. **IMMEDIATE SNAPSHOTS**

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 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.

Now let us shift our attention from the operational model of schedules to the combinatorial model of vertexes, simplexes, 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 $\sigma$, the set of immediate snapshot executions defines a protocol complex $\mathcal{J}S(\sigma)$. Each vertex in this complex has the form $\langle P_i, \sigma_i \rangle$, where $P_i$ is the process taking the steps, and $\sigma_i$, the result of its snapshot, is a face of the input simplex $\sigma$. Each simplex in the protocol complex satisfies the following properties.

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

**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 $P_i \in \text{id}(\sigma_j)$, then $\sigma_i \subseteq \sigma_j$.

Fig. 3.3 shows the full immediate snapshot complex for three processes, highlighting the simplexes 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.

Let $\sigma = \{\vec{s}_0, \ldots, \vec{s}_n\}$ be an input simplex, where $\text{id}(\vec{s}_i) = P_i$. Recall that $\text{Face}_i \sigma = \sigma \setminus \{\vec{s}_i\}$ is the face of $\sigma$ not labeled with $P_i$.

**Theorem 3.2.4.** If the input complex $\mathcal{I}$ is a manifold, so is the immediate snapshot protocol complex $\mathcal{J}S(\mathcal{I})$.

**Proof.** Let $\sigma^{n-1}$ be an $(n-1)$-simplex of $\mathcal{J}S(\sigma)$. For ease of presentation, reindex the process ids so that

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

We must show that $\sigma^{n-1}$ is a face of exactly one or two $n$-simplexes.

The simplex $\sigma_{n-1}$ has dimension either $n-1$ or $n$. If it has dimension exactly $n-1$, then $\sigma_{n-1}$ 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 $\sigma_{n-1}$ is a boundary $(n-1)$-simplex of $\mathcal{I}$, then because $\mathcal{I}$ is a manifold, then is exactly one $n$-simplex $\sigma$ that contains $\sigma_{n-1}$. There is exactly one $n$-simplex of $\mathcal{J}S(\mathcal{I})$ $\{\langle P_n, \sigma \rangle\} \cup \sigma_{n-1}$, that contains $\sigma_{n-1}$. (See Fig. 3.2)

- If $\sigma_{n-1}$ is an internal $(n-1)$-simplex of $\mathcal{I}$, then because $\mathcal{I}$ is a manifold, then there are exactly two $n$-simplexes $\sigma$ and $\sigma'$ of $\mathcal{I}$ that contain $\sigma_{n-1}$. There are exactly two $n$-simplexes of $\mathcal{J}S(\mathcal{I})$ $\{\langle P_n, \sigma \rangle\} \cup \sigma_{n-1}$, and $\{\langle P_n, \sigma' \rangle\} \cup \sigma_{n-1}$, that contain $\sigma_{n-1}$. (See Fig. 3.2)
• If \( \sigma_{n-1} \) is an \( n \)-simplex of \( \mathcal{I} \), suppose that \( P_n \) is in \( \text{ids}(\sigma_\ell) \) but not \( \text{ids}(\sigma_{\ell-1}) \), for \( \ell \), where we understand \( \sigma_{-1} \) to be the empty set. Note that \( \sigma_{\ell-1} \subset \sigma_\ell \), implying that \( P_\ell \not\in \text{ids}(\sigma_{\ell-1}) \). For what values of \( \sigma_n \) is \( \{ \langle P_n, \sigma_n \rangle \} \cup \sigma_{n-1} \) an \( n \)-simplex of \( \mathcal{I}\mathcal{S}(\mathcal{I}) \)? \( P_n \in \text{ids}(\sigma_n) \), hence \( \bar{s}_n \in \sigma_n \).

By Property 3.2.1

\[ \sigma_{\ell-1} \cup \{ \bar{s}_n \} \subseteq \sigma_n \subseteq \sigma_\ell. \]

If

\[ \sigma_{\ell-1} \cup \{ \bar{s}_n \} \subset \sigma_n, \]

there is a process id \( P_i \) in \( \text{ids}(\sigma_n) \setminus \text{ids}(\sigma_{\ell-1} \cup \{ \bar{s}_n \}) \). By hypothesis, \( \sigma_i \) is a subset of \( \sigma_\ell \) but not \( \sigma_{\ell-1} \). Because these simplexes are ordered by inclusion, it follows that \( \sigma_i = \sigma_\ell \), and therefore \( \sigma_n = \sigma_\ell \). It follows that \( \sigma_n \) can assume exactly two distinct values: \( \sigma_{\ell-1} \cup \{ \bar{s}_n \} \) or \( \sigma_\ell \), implying that \( \sigma_{n-1} \) is contained in exactly two \( n \)-simplexes.
The proof of Theorem 3.2.4 uncovered an interesting fact about the immediate snapshot complex: the boundary \((n - 1)\)-simplexes are precisely those simplexes where one process does not appear in any of the others’ views.

**Corollary 3.2.5.**

\[ \partial \mathcal{I}S(I) = \mathcal{I}S(\partial I). \]

### 3.3 Manifold Protocols

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 \(I\) is a manifold, so is \(\mathcal{M}(I)\).

**Property 3.3.2.** \(\mathcal{M}(\partial I) = \partial \mathcal{M}(I)\).
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
3.4. SET AGREEMENT

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 vertexes \(V\) and a set of edges \(E\). The degree of a vertex, \(\deg(\vec{v})\), is the number of edges to which \(\vec{v}\) belongs.

**Lemma 3.4.1.** In any graph \(G = (V, E)\), the sum of the degrees of the vertexes 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. \(\square\)

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

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

Suppose we color an \(n\)-simplex \(\sigma = (\vec{s}_0, \ldots, \vec{s}_n)\) with \(n + 1\) distinct colors, and then color a subdivision \(\text{Div} \sigma\) 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 \(\Delta^n\) be a subdivision of \(\Delta\), and let \(\chi : \text{Div} \Delta^n \to \Delta\) be a simplicial map such that \(\chi(\vec{v}) \in \text{Car}(\vec{v}, \Delta^n)\). The map \(\chi\) maps an odd number of simplexes of \(\text{Div} \Delta^n\) onto \(\Delta^n\).

**Proof.** We argue by induction on the dimension \(n\). When \(n\) is zero, the complexes \(\Delta^0\) and \(\text{Div} \Delta^0\) are the same, \(\chi\) is the identity map, and the claim is trivial.

Assume the result for \(n - 1\). By the induction hypothesis,

\[\chi : \text{Div} \text{Face}_i \Delta^n \to \text{Face}_i \Delta^n\]

sends an odd number of \((n - 1)\)-simplexes to \(\text{Face}_n \Delta^n\). The subdivision \(\text{Div} \Delta^n\) is a manifold, so by Lemma 3.4.3 \(\chi\) sends an odd number of \(n\)-simplexes of \(\text{Div} \Delta^n\) onto \(\Delta^n\). \(\square\)

**Lemma 3.4.4 (Sperner’s Lemma for Manifolds).** Let \(\Delta^n = \{\vec{d}_0, \ldots, \vec{d}_n\}\) be an \(n\)-simplex, and \(\text{Face}_i \Delta^n\) the \((n - 1)\)-face of \(\Delta^n\) 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 $\chi$ sends an odd number of $(n-1)$-simplexes of $\mathcal{B}$ onto $\text{Face}_n \Delta^n$, then $\chi$ sends an odd number of $n$-simplexes of $\mathcal{M}$ onto $\Delta^n$.

**Proof.** Define $G$ to be the graph whose vertexes are indexed by the $n$-simplexes of $\mathcal{M}$, with the addition of one more “external” vertex $\vec{e}$. There is an edge between two vertexes if their corresponding simplexes share a common $(n-1)$-face such that $\chi$ sends that face onto $\text{Face}_n \Delta^n$. There is also an edge from the external vertex $\vec{e}$ to every $n$-simplex $\sigma$ such that $\sigma$ has an $(n-1)$ face in $\mathcal{B}$, and $\chi$ 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 $\sigma$. We claim that $\vec{v}_\sigma$ has an odd degree if and only if $\chi$ maps $\sigma$ onto $\{0, \ldots, n-1\}$. There are three cases to consider. First, if $\chi$ does not map any $(n-1)$ face of $\sigma$ onto $\{0, \ldots, n-1\}$, then $\vec{v}_\sigma$ has degree 0. Suppose $\chi$ does map a $(n-1)$ face of $\sigma$ onto $\{0, \ldots, n-1\}$. Second, if $\chi$ maps the remaining vertex to
3.4. \textit{SET AGREEMENT}

Figure 3.8: Dual Graph of Subdivision with “external” node

a value in \(\{0, \ldots, n - 1\}\), then \(\chi\) maps exactly two \((n - 1)\)-faces of \(\sigma\) onto \(\{0, \ldots, n - 1\}\). 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 \(\chi\) maps the remaining vertex to \(n\), then \(\chi\) maps exactly one \((n - 1)\)-face of \(\sigma\) onto \(\{0, \ldots, n - 1\}\), implying that \(\vec{v}_\sigma\) has degree 1.

Moreover, \(\vec{e}\) has odd degree. By hypothesis, \(\chi\) sends an odd number of boundary \((n - 1)\)-simplexes 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 vertexes of odd degree. Since the external node \(\vec{e}\) has odd degree, there must be an odd number of other vertexes \(\vec{v}_\sigma\) with odd degree. Each of these vertexes corresponds to an \(n\)-simplex that \(\chi\) maps onto \(\{0, \ldots, n - 1\}\).

\(\square\)

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

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 \( P_i: \text{id}(\vec{a}_i) = P_i \). The protocol \( \mathcal{M} \) solves set agreement if and only if there is a simplicial map \( \delta: \mathcal{M}(\sigma) \to \sigma \) sending each vertex of the protocol complex to the input vertex labeled with the value chosen. To solve set agreement, the coloring \( \delta \) cannot map any \( n \)-simplex onto \( \sigma \). By way of contradiction, we will show that \( \delta \) must map an odd number of \( n \)-simplexes onto \( \sigma \).

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,

\[
\partial \mathcal{M}(\sigma) = \bigcup_{P_i \in \text{ids}(\sigma)} \mathcal{M}(\text{Face}_i \sigma)
\]

Process \( P_i \) can choose \( P_j \)'s input only in executions where \( P_j \) takes a step, so \( \chi \) sends no \((n - 1)\)-simplexes of \( \mathcal{M}(\text{Face}_i \sigma) \) onto \( \text{Face}_n \sigma \) for \( i \neq n \). By the induction hypothesis, \( \chi: \mathcal{M}(\text{Face}_n \sigma) \to \text{Face}_n \sigma \) sends an odd number of \((n - 1)\)-simplexes onto \( \text{Face}_n \sigma \). In total, \( \chi \) sends an odd number of \((n - 1)\)-simplexes of \( \partial \mathcal{M}(\sigma) \) onto \( \text{Face}_n \sigma \). By Lemma 3.4.4, \( \chi \) sends an odd number of \( n \)-simplexes of \( \mathcal{M}(\sigma) \) onto \( \sigma \).

Theorem 3.4.5 implies that for any manifold protocol, any possible decision map has an odd (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 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,
3.6. **WEAK SYMMETRY-BREAKING**

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 \(\pi\) 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 here are symmetric. Recall that the protocol has a decision function \(\delta : \mathcal{P}(\mathcal{I}) \to \mathcal{O}\), carried by \(\Delta\). 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:

\[
\begin{align*}
\mathcal{P}(\mathcal{I}) & \xrightarrow{\delta} \mathcal{O} \\
\downarrow{\pi} & \downarrow{\pi} \\
\mathcal{P}(\mathcal{I}) & \xrightarrow{\delta} \mathcal{O}
\end{align*}
\]

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

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*. Fig. 3.9 shows the output complex for 3-process weak symmetry-breaking. This complex is an *annulus*: a disk with a hole in the center. 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 vertexes forms a simplex if (1) their process IDs are distinct, and (2) fewer than \(n + 1\) vertexes have the same Boolean value.

If each process has a unique ID in the range \(0, \ldots, n+1\), 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

We say a task $T$ implements a task $S$ in a particular model if one can construct a protocol for $S$ using read-write memory and a finite number of “black boxes” solve $T$. If $T$ implements $S$, but not vice-versa, then we say that $S$ is weaker than $T$. 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.

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 \( \text{put}() \) 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 \( \text{decide}(x) \) method, that runs an anonymous set agreement protocol with input \( x \), and returns a decision value.

Each processes calls the anonymous set agreement object’s \( \text{decide}() \) method, using its own id as input (Line 3). It stores the result in the set (Line 4). The process then iterates through the set (Line 5), returning \text{true} \) if it finds its own ID in the set, and \text{false} \) otherwise.

\textbf{Lemma 3.7.1.} If all \((n + 1)\) processes participate, some process decides \text{true}.

\textbf{Proof.} The process whose id is first to be put in the set will observe its own
Figure 3.11: Weak Symmetry-breaking from one-round Moebius task protocol

```java
Set<ID> output;  // set of IDs
anonSetAgree sa;  // anonymous set agreement

boolean choose(ID me) {
  ID choice = sa.decide(me);
  output.put(choice);
  foreach (ID id in output) {
    if (id == me) return true;
  }
  return true;
}
```

Figure 3.12: Implementing weak symmetry-breaking from set agreement
3.7. ANONYMOUS SET AGREEMENT VERSUS WEAK SYMMETRY BREAKING

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.

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 \(M(\cdot)\), called the Moebius task. First, we construct the 2-dimensional Moebius task. As illustrated in Fig. 3.10 for each input 2-simplex \(\sigma\), take three “copies” \(\xi_0, \xi_1, \xi_2\) of \(\text{Ch} \sigma\). We call \(\text{Face}_i \xi_i\) the external face of \(\xi_i\) (even though it is technically a complex), and \(\text{Face}_j \xi_i\), for \(i \neq j\), the internal faces. We then identify (that is “glue together”) \(\text{Face}_1 \xi_0\) and \(\text{Face}_1 \xi_2\), \(\text{Face}_2 \xi_1\) and \(\text{Face}_2 \xi_3\), and \(\text{Face}_3 \xi_1\) and \(\text{Face}_3 \xi_2\). The resulting complex is a manifold. It is easy to check that \(M(\partial \sigma) = \partial M(\sigma) = \text{Ch} \partial \sigma\).

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 \(\xi_i\), color each node black except for the one labeled with \(P_i\). For the central simplex of each external face \(\text{Face}_i \xi_i\), 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 \(\sigma\), take \(n+1\) “copies” \(\xi_0, \ldots, \xi_n\) of \(\text{Ch} \sigma\). As before,
we call the complex Face\(_i\)\(\xi_i\) the external face of \(\xi_i\) 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 \(\pi_j : \Pi \setminus \{j\} \rightarrow \Pi \setminus \{j\}\) be the map sending the ID with rank \(r\) in \(\Pi \setminus \{j\}\) to the ID with rank \(r + N \mod 2N\).

For each \(i\), and each \(j \neq i\), \(\pi_j(i)\), identify the internal face Face\(_j\)\(\xi_i\) with 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 \((2N)\)-simplexes, so the result is a manifold. (This why this construction works only in even dimensions.)

Let \(\sigma\) be an input \(n\)-simplex. The Moebius task’s carrier map carries each proper face \(\tau\) of \(\sigma\) to Ch\(\tau\). It carries \(\sigma\) itself to all \(n\)-simplexes of \(M(\sigma)\).

**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 \(\xi_i\), color each node black except for the one labeled with \(P_i\). For the central simplex of each external face \(\xi_{i\i}\), color the central \((2N - 2)\)-simplex black. The rest are white.

Every \((2N - 1)\)-simplex \(\xi\) in \(\xi_i\) intersects both a face, either internal or external, and a central \((2N - 1)\)-simplex. If \(\xi\) intersects an internal face, then the vertexes on that face are white, and the vertexes on the central simplex are black. If \(\xi\) intersects the internal face, then it intersects the white node of the central simplex of \(\xi_i\), and a black node of the central simplex of \(\xi_{i\i}\).

**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 [4] and to Saks and Zaharoughu [17], 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 [9].

3.9 Exercises

Exercise 3.1. Count the number of simplexes in $I\mathcal{S}(\sigma^n)$.

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

Exercise 3.3. Compute the Euler characteristic of $I\mathcal{S}(\sigma^n)$.

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 $\text{Set}<\text{ID}>$ 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


