An Overview of Synchronous Message-Passing and Topology

Maurice Herlihy

Brown University
Providence, RI 02912
herlihy@cs.brown.edu

Sergio Rajsbaum [1]

Compaq Computer Corporation
One Cambridge Center
Cambridge, MA 02142-1612
rajsbaum@crl.dec.com

Mark R. Tuttle

Compaq Computer Corporation
One Cambridge Center
Cambridge, MA 02142-1612
tuttle@crl.dec.com

Abstract

A slowly-growing number of computer scientists have found that ideas from topology can be used to analyze and understand problems in distributed computing. In this paper, we review one approach we have used in the past to write a succinct proof of the lower bound for the number of rounds needed to solve the k-set agreement problem in a synchronous, message-passing model of computation. The central idea in this approach is a simple combinatorial structure we call a *pseudosphere* in which each process from a set of processes is independently assigned a value from a set of values. Pseudospheres have a number of nice combinatorial properties, but their principal interest lies in the observation that the global states that arise in the synchronous, message-passing model can be viewed as simple unions of pseudospheres, and the fact that topological properties of unions of pseudospheres are so easy to prove. We choose this work to review because it is a simple example of how we model distributed systems with topology, and because it is the basis of on-going work to simplify the proof of this result.

Preprint submitted to Elsevier Preprint *June 6, 2000*

1 Introduction

Computer scientists have a long tradition of using ideas from topology in their work on problems from semantics and concurrency theory, but only recently have ideas from topology played a role in proving powerful new results in distributed computing. Beginning with a trio of papers independently proving the impossibility of solving the k-set agreement problem in asynchronous systems [BG93,HS99,SZ93], these ideas have been used to study other problems in many other models of computation [AR96,GK99,HR94,HR95,CHLT93,HRT98]. The purpose of this paper is to illustrate how topology is used to model computation in a distributed system, and how ideas from topology can be used to reason about distributed computation. We illustrate these ideas by sketching a recent proof [HRT98] we wrote of a known lower bound [CHLT93] on the number of rounds of communication needed to solve k-set agreement in a synchronous, message-passing model of computation. Our proof is the basis of work in progress to use topology to write proofs of this and other lower bounds that are as succinct as possible, deriving new topological tools for analyzing distributed computation along the way. Let us begin by illustrating why topology is a natural tool for proving the lower bound for k-set agreement, borrowing liberally from introduction to the original proof [CHLT93].

The *k-set agreement problem* [Cha93] is defined as follows. Each processor has a read-only input register and a write-only output register. Each processor begins with an arbitrary input value in its input register from a set V containing at least $k+1$ values $v_0, \ldots, v_k$, and nothing in its output register. A protocol solves k-set agreement if, in every execution, the nonfaulty processors halt after writing output values to their output registers that satisfy two conditions:

(i) *validity*: every processor's output value is some processor's input value, and

(ii) *agreement:* the set of output values chosen must contain at most k distinct values.

The first condition rules out trivial solutions in which a single value is hardwired into the protocol and chosen by all processors in all executions, and the second condition requires that the processors coordinate their choices to some degree. In the special case of $k = 1$, the 1-set agreement is equivalent to the well-known *consensus* problem [LSP82,PSL80,FL82,FLP85,Dol82,Fis83] in when all processors are required to choose the same output value. Consensus is known to be the "hardest" problem in distributed computing, in the sense that all other decision problems can be reduced to it.

[1] On leave from Instituto de Matemáticas, U.N.A.M., D.F. 04510, México, rajsbaum@math.unam.mx

We consider the k-set agreement problem in a *synchronous*, *message-passing* model with *crash failures*. In this model, n processors communicate by sending messages over a completely connected network. Computation in this model proceeds in a sequence of rounds. In each round, processors send messages to other processors, then receive messages sent to them in the same round, and then perform some local computation and change state. This means that all processors take steps at the same rate, and that all messages take the same amount of time to be delivered. Communication is reliable, but up to f processors can fail by crashing in the middle of a round. When a processor crashes, it sends some subset of the messages it is required to send in that round by the protocol, and then sends no messages in any later round.

The primary contribution of this paper is a (tight) lower bound on the amount of time required to solve k-set agreement. We prove that any protocol solving k-set agreement requires $\lfloor f/k \rfloor + 1$ rounds of communication, where f is the bound on the number of processors allowed to fail in any execution of the protocol. Since consensus is just 1-set agreement, this lower bound implies the famous lower bound of $f + 1$ rounds for solving consensus [FL82]. More important, the running time $r = \lfloor f/k \rfloor + 1$ demonstrates that there is a smooth but inescapable tradeoff among the number f of faults tolerated, the degree k of coordination achieved, and the time r the protocol must run.

Suppose P is a protocol that solves k-set agreement and tolerates the failure of f out of n processors, and suppose P halts in $r < \lfloor f/k \rfloor + 1$ rounds. This means that all nonfaulty processors have chosen an output value at time r in every execution of P. In addition, suppose $n \geq f + k + 1$, which means that at least $k + 1$ processors never fail. Our goal is to consider the *global states* that occur at time r in executions of P, and to show that in one of these states there are $k + 1$ processors that have chosen $k + 1$ distinct values, violating k-set agreement and showing that P could not possibly have solve k-set agreement in only r rounds.

Since consensus is a special case of k-set agreement, it is helpful to review the standard proof of the $f + 1$ round lower bound for consensus [FL82,DS83,Mer85,DM90] to see why new ideas from topology are needed for k-set agreement. Suppose that the protocol P is a consensus protocol, which means that in all executions of P all nonfaulty processors have chosen the same output value at time r. Two global states g_1 and g_2 at time r are said to be *similar* if some nonfaulty processor p has the same local state in both global states. The crucial property of similarity is that the decision value of any processor in one global state completely determines the decision value for any processor in all similar global states. For example, if all processors decide v in g_1, then certainly p decides v in g_1. Since p has the same local state in g_1 and g_2, and since p's decision value is a function of its local state, processor p also decides v in g_2. Since all processors agree with p in g_2, all processors decide v in g_2, and it follows

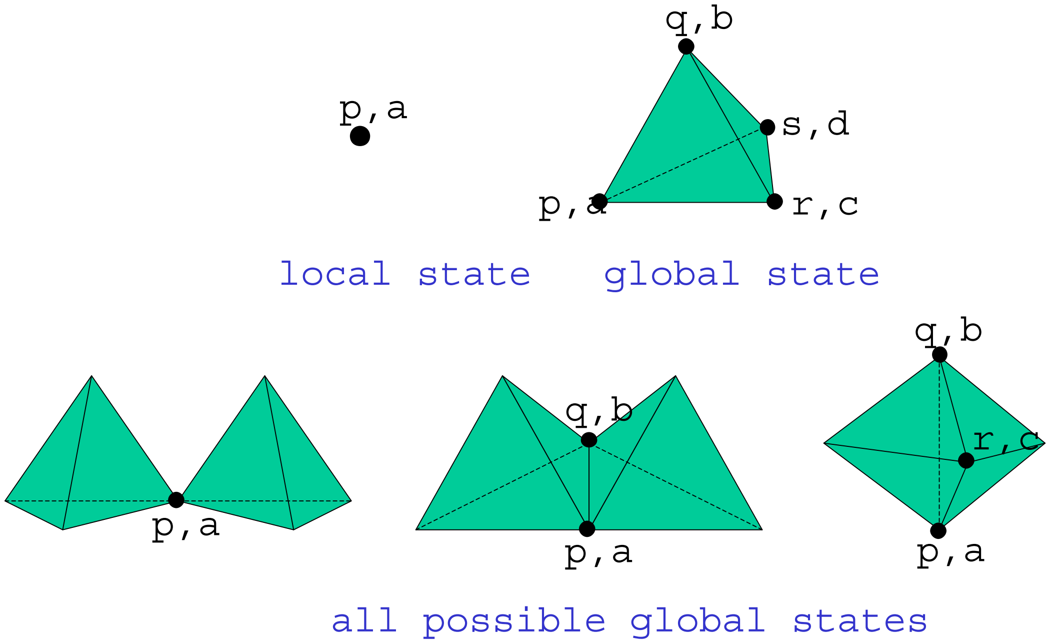

Fig. 1. Modeling global states with a simplicial complex.

that the decision value in g_1 determines the decision value in g_2. A *similarity chain* is a sequence of global states, $g_1, \ldots, g_\ell$, such that g_i is similar to g_{i+1}. A simple inductive argument shows that the decision value in g_1 determines the decision value in g_ℓ. The lower bound proof involves showing that two time r global states of P, one in which all processors start with 0 and one in which all processors start with 1, lie on a single similarity chain. Since there is a similarity chain from one state to the other, processors must choose the same value in both states, violating the definition of consensus.

The problem with k-set agreement is that the decision values in one global state do not determine the decision values in similar global states. If p has the same local state in g_1 and g_2, then p must choose the same value in both states, but the values chosen by the other processors are not determined. Even if $n-1$ processors have the same local state in g_1 and g_2, the decision value of the last processor is still not determined. The fundamental insight in all proofs of this lower bound [CHLT93,HRT98] is that k-set agreement requires considering all "degrees" of similarity at once—similarity to one processor, to two processors, to three processors—focusing on the number and identity of local states common to two global states. While this seems difficult—if not impossible—to do using conventional graph theoretic techniques like similarity chains, the notions of a simplex and a simplicial complex provides a compact way of capturing all degrees of similarity simultaneously, and are the basis of our proof.

A simplex is just the natural generalization of a triangle to n dimensions:

for example, a 0-dimensional simplex is a vertex, a 1-dimensional simplex is an edge linking two vertices, a 2-dimensional simplex is a solid triangle, and a 3-dimensional simplex is a solid tetrahedron. As illustrated in Figure 1, we can represent a local state for one processor p with a single vertex and a global state for four processors p, q, r, and s with a 3-dimensional simplex. We label a single vertex representing a processor's local state with the processor's name p and local state a, and we label a 3-dimensional simplex representing a global state for p, q, r, and s by labeling the vertexes corresponding to p, q, r, and s in the same way. Representing all global states as simplexes in this way, the intersection of two simplexes naturally captures the degree of similarity between the two corresponding global states. For example, referring again to Figure 1, two global states similar to p are represented by two simplexes intersecting only in p's vertex, two global states similar to p and q are represented by two simplexes intersecting in the edge between p and q, and two global states similar to p, q, and r are represented by two simplexes intersecting in the entire face containing p, q, and r.

Figure 2 shows the simplicial complexes — called *protocol complexes* — representing the global states reachable after zero, one, and two rounds of computation in a simple protocol in which each of three processors repeatedly sends its state to the others. Each process begins with a binary input. The first picture shows the possible global states after zero rounds: since no communication has occurred, each processor's state consists only of its input. It is easy to check that the simplexes corresponding to these global states form an octahedron. The next picture shows the complex after one round. Each triangle corresponds to a failure-free execution, each free-standing edge to a single-failure execution, and so on. The third picture shows the possible global states after three rounds.

The connection between these protocol complexes and k-set agreement is the following theorem. Let P be a protocol, and let $\mathcal{C}$ be the simplicial complex representing the set of global states reachable by following P for r rounds of computation. The theorem states that if $\mathcal{C}$ is $(k-1)$-connected, then P cannot solve k-set agreement in r rounds. Proving our lower bound reduces to reasoning about the connectivity of such simplicial complexes.

The key to our proof is the notion of a *pseudosphere*, a simplicial complex in which each process from a set of processes is independently assigned a value from a set of values. Pseudospheres have a number of nice combinatorial properties, but their principal interest lies in the observation that protocol complexes in the synchronous model can be characterized as simple unions of pseudospheres. Because of the simple combinatorial properties of pseudospheres, reasoning about these unions can be accomplished by straightforward combinatorial arguments.

A pseudosphere can be defined very simply, as illustrated in Figure 3. Start

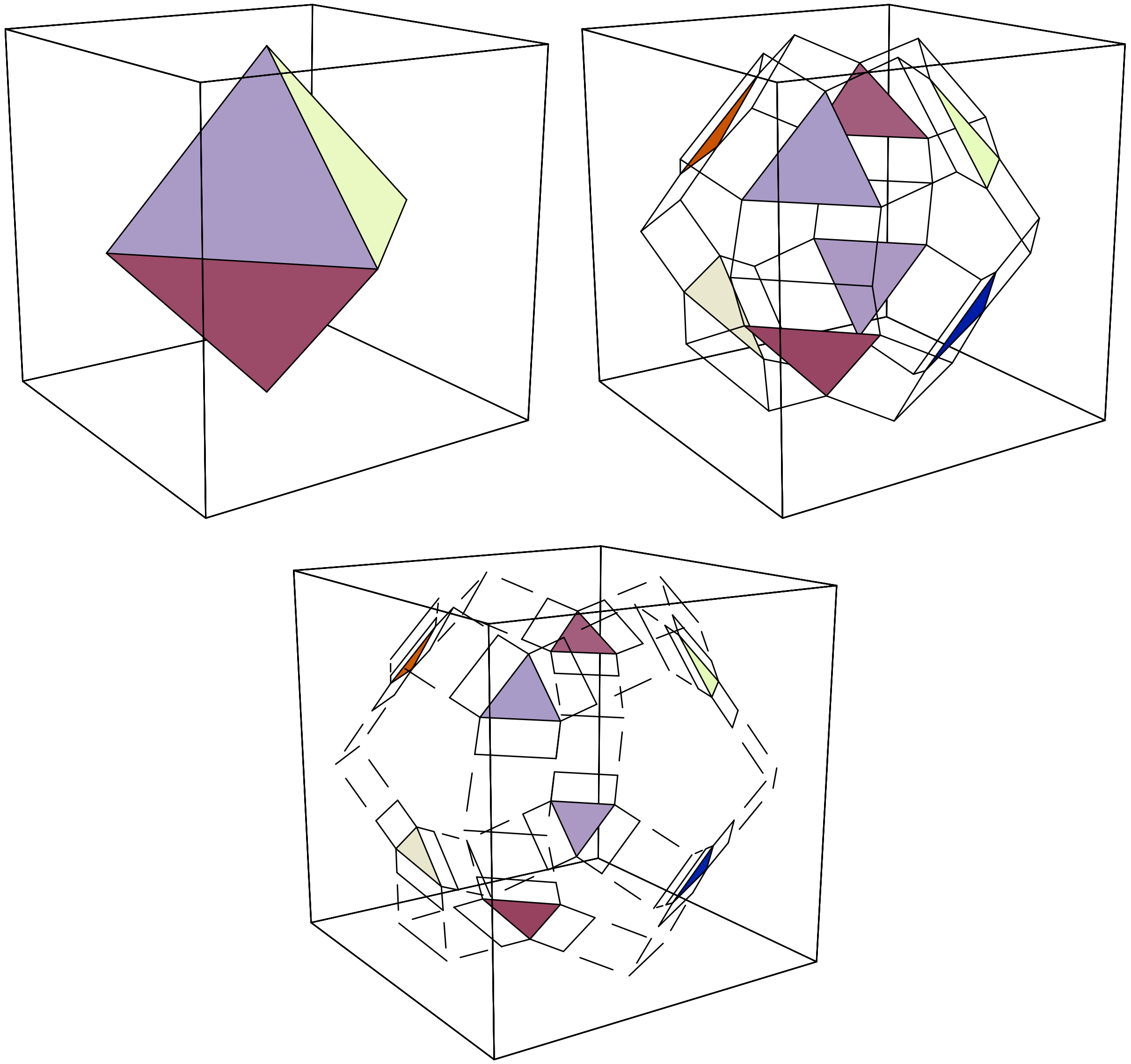

Fig. 2. Global states for zero, one, and two-round protocols.

with an n-dimensional simplex where each vertex is labeled with a process id, and choose a finite set of values taken from an arbitrary domain. The pseudosphere is the complex constructed by taking multiple copies of this simplex and independently labeling each vertex with a value from the domain. For example, Figure 3 shows how to construct a pseudosphere by independently assigning binary values to a set of three processes. The left-hand figure shows a triangle labeled with process ids P, Q, and R. The central figure shows an intermediate stage where two copies of the triangle are each labeled with zeros and ones. The right-hand figure shows the complete construction, where copies of the triangle are labeled with all combinations of zeros and ones. We

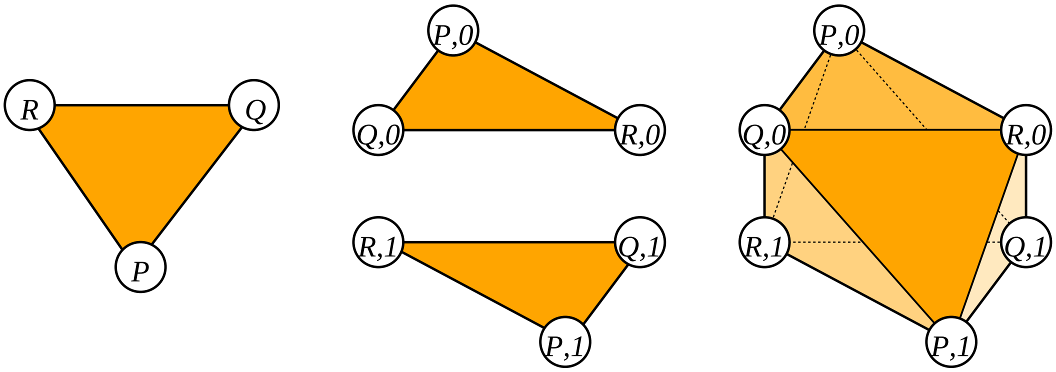

Fig. 3. Construction of a three-process binary pseudosphere.

can just as easily assign values from a larger set than $\{0,1\}$, although the result is harder to illustrate. We call this construct a pseudosphere because it is easily shown that the result of assigning binary values to $n+1$ processes is topologically equivalent to an n-dimensional sphere.

The collection of initial global states for consensus or k-set agreement clearly forms a pseudosphere whose vertices are labeled with input values. For example, the right-hand figure in Figure 3 is the input complex for three-process consensus. The basic insight underlying the work presented in this paper is that protocol complexes in the synchronous model have natural representations as unions of pseudospheres, except that the vertices are labeled failure information instead of input values. Reasoning about these protocol complexes reduces to the purely combinatorial problem of reasoning about unions of pseudospheres. We express the one-round executions as the union of pseudospheres. An r-round execution is constructed by inductively replacing each simplex in the single-round execution with the union of pseudospheres produced by the $(r-1)$-round protocol. The protocol complex produced by this iterative construction represents only a subset of the global states reachable in the model, but this set is large enough to prove the desired results for consensus, k-set agreement, renaming, and so on.

2 Basic Topology

A *vertex* $\vec{v}$ is a point in a high-dimensional Euclidian space. Vertexes $\vec{v}_0,\ldots,\vec{v}_n$ are *affinely independent* if $\vec{v}_1-\vec{v}_0,\ldots,\vec{v}_n-\vec{v}_0$ are linearly independent. An *n-dimensional simplex* (or *n-simplex*) $S^n=(\vec{s}_0,\ldots,\vec{s}_n)$ is the convex hull of a set of $n+1$ affinely-independent vertexes. For example, a 0-simplex is a vertex, a 1-simplex a line segment, a 2-simplex a solid triangle, and a 3-simplex a solid tetrahedron. Where convenient, we use superscripts to indicate dimensions of simplexes. We say that the $\vec{s}_0,\ldots,\vec{s}_n$ *span* S^n. By convention, a simplex of dimension $d<0$ is an empty simplex. Simplex S^m is a (proper) *face* of T^n if the vertexes of S^m are a (proper) subset of the vertexes of T.

A *simplicial complex* (or complex) is a set of simplexes closed under containment and intersection. The *dimension* of a complex is the highest dimension of any of its simplexes. In this paper all the complexes of dimension n are *full* in the sense that every simplex is contained in some n-simplex. $\mathcal{L}$ is a *subcomplex* of $\mathcal{K}$ if every simplex of $\mathcal{L}$ is a simplex of $\mathcal{K}$. The *m-skeleton* of $\mathcal{K}$, denoted $skel^m(\mathcal{K})$, is the subcomplex consisting of all simplexes of $\mathcal{K}$ of dimension at most m. A map $\mu : \mathcal{K} \to \mathcal{L}$ carrying vertexes to vertexes is *simplicial* if it also carries simplexes to simplexes. Two complexes $\mathcal{K}$ and $\mathcal{L}$ are *isomorphic*, written $\mathcal{K} \cong \mathcal{L}$, if there is a surjective and one-to-one simplicial map $\iota : \mathcal{K} \to \mathcal{L}$.

Informally, a complex is k-connected if it has no holes in dimensions k or less. More precisely,

Definition 2.1 *A complex $\mathcal{K}$ is* k-connected *if every continuous map of the k-sphere to $\mathcal{K}$ can be extended to a continuous map of the $(k+1)$-disk [Spa66, p. 51]. (By convention, a complex is* (-1)-connected *if it is nonempty, and every complex is* k-connected *for $k < -1$.)*

This definition says that a complex is 0-connected if it is connected in the graph-theoretic sense. The following theorem is an elementary consequence of the Mayer-Vietoris sequence [Mun84, p. 142]. It allows us to reason about a complex's connectivity in terms of the connectivity of its components.

Theorem 2.2 *If $\mathcal{K}$ and $\mathcal{L}$ are complexes such that $\mathcal{K}$ and $\mathcal{L}$ are k-connected, and $\mathcal{K} \cap \mathcal{L}$ is nonempty and $(k-1)$-connected, then $\mathcal{K} \cup \mathcal{L}$ is k-connected.*

3 Model

A set of $n+1$ sequential *processes* communicate by sending messages to one another. At any point, a process may *crash*: it stops and sends no more messages. There is a bound f on the number of processes that can fail. In the *synchronous* model, processes take steps at the same rate, and messages take the same amount of time to be delivered, and message delivery is reliable and FIFO.

Each process starts with an *input value* taken from a set V, and then executes a deterministic *protocol* in which it repeatedly receives one or more messages, changes its local state, and sends one or more messages. After a finite number of steps, each process chooses a *decision value* and halts. At any instant, a process's local state is given by its *view*: the input value and the the sequence of messages received so far. A protocol is uniquely determined by its *message function* and its *decision function.* The message function determines which messages a process should send in a given state, and the decision function determines which output value a process should choose in a given state (if any). A protocol is a *full-information protocol* [Had83,FL82,PSL80] if the message function causes each process to send its entire local state when it sends a message. We can assume without loss of generality that all protocols $\mathcal{P}$ we consider are *full-information* protocols [Had83,FL82,PSL80,DM90].

In the k-set agreement task [Cha91], processes are required to (1) choose a decision value after a finite number of steps, (2) choose as decision value some process's input value, and (3) collectively choose no more than k distinct decision values. When $k = 1$, this problem is usually called *consensus*.

We now show how to apply concepts from combinatorial topology to this model. An initial local state of process P is modeled as a vertex $\vec{v} = \langle P, v \rangle$ labeled with P's process id and initial value v. An initial global state is modeled as an n-simplex $S^n = (\langle P_0, v_0 \rangle, \ldots, \langle P_n, v_n \rangle)$, where the P_i are distinct. We use $\mathit{ids}(S^n)$ to denote the set of process ids associated with S^n, and $\mathit{vals}(S^n)$ the set of values. The set of all possible initial global states forms a complex, called the *input complex*.

Any protocol has an associated *protocol complex* $\mathcal{P}$, defined as follows. Each vertex is labeled with a process id and a possible view for that process. A set of vertexes $\langle P_{i_0}, v_{i_0} \rangle, \ldots, \langle P_{i_d}, v_{i_d} \rangle$ spans a simplex of $\mathcal{P}$ if and only if there is some protocol execution in which $P_{i_0}, \ldots, P_{i_d}$ finish the protocol with respective views $v_{i_0}, \ldots, v_{i_d}$. Each simplex thus corresponds to an equivalence class of executions that "look the same" to the processes at its vertexes. The protocol complex $\mathcal{P}$ depends both on the protocol and on the timing and failure characteristics of the model.

We use $\mathcal{P}(S^m)$ to denote the subcomplex of $\mathcal{P}$ corresponding to executions in which only the processes in $\mathit{ids}(S^m)$ participate (the rest fail before sending any messages). If $m < n - f$, then there are no such executions, and $\mathcal{P}(S^m)$ is empty. More generally, if $\mathcal{I}$ is a subcomplex of the input complex, then we define $\mathcal{P}(\mathcal{I})$ to be the union of $\mathcal{P}(S^m)$ for all S^m in $\mathcal{I}$. A protocol *solves* k-set agreement if the protocol's decision map δ carries vertexes of $\mathcal{P}$ to values in V such that if $\vec{p} \in \mathcal{P}(S^n)$, then $\delta(\vec{p}) \in \mathit{vals}(S^n)$.

4 Pseudospheres

Informally, a pseudosphere is a combinatorial structure in which each process from a set of processes is independently assigned a value from a set of values.

Definition 4.1 *Let $S^m = (\vec{s}_0, \ldots, \vec{s}_m)$ be a simplex and $U_0, \ldots, U_m$ be a sequence of finite sets. The* pseudosphere *$\psi(S^m; U_0, \ldots, U_m)$ is the following complex. Each vertex is a pair $\langle \vec{s}_i, u_i \rangle$, where $\vec{s}_i$ is a vertex of S^m and $u_i \in U_i$. Vertexes $\langle \vec{s}_{i_0}, u_{i_0} \rangle, \ldots, \langle \vec{s}_{i_\ell}, u_{i_\ell} \rangle$ span a simplex of $\psi(S^m; U_0, \ldots, U_m)$ if and only if the $\vec{s}_i$ are distinct. A pseudosphere in which all U_i equal U is simply written $\psi(S^m; U)$.*

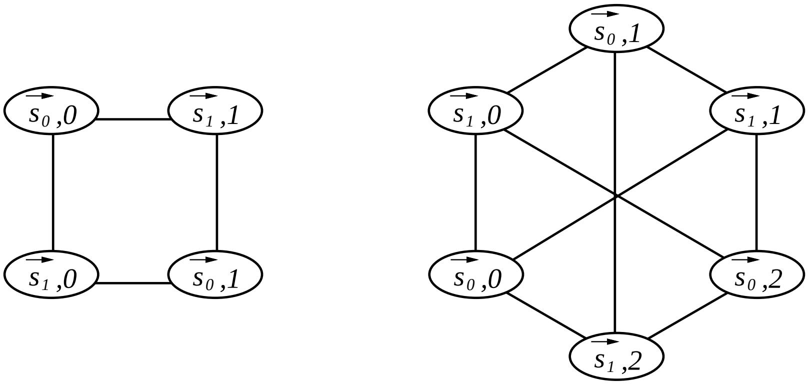

Fig. 4. Pseudospheres $\psi(\{P_0, P_1\}\,; \{0,1\})$ and $\psi(\{P_0, P_1\}\,; \{0,1,2\})$.

We call this construct a pseudosphere because if S^n is an n-dimensional simplex, then $\psi(S^n; \{0,1\})$ is homeomorphic to an n-dimensional sphere. Pseudospheres are important because every complex considered here is either a pseudosphere or the union of pseudospheres. Because any process can start with any input from V, the input complex to k-set agreement is the pseudosphere $\psi(P^n; V)$, where P^n is a simplex whose vertexes are labeled with the $n+1$ distinct process ids.

Lemma 4.2 *Pseudospheres satisfy the following simple combinatorial properties.*

(i) *If U is a singleton set, then $\psi(S^m, U) \cong S^m$.*

(ii) *Let $S^m = (\vec{s}_0, \ldots, \vec{s}_m)$, and $S^{m-1} = (\vec{s}_0, \ldots, \widehat{\vec{s}_i}, \ldots \vec{s}_m)$, where circumflex denotes omission. If $U_i = \emptyset$, then*

$$\psi(S^m; U_0, \ldots, U_m) \cong \psi(S^{m-1}; U_0, \ldots, \widehat{U}_i, \ldots, U_m).$$

(iii) $\psi(S_0; U_0, \ldots, U_m) \cap \psi(S_1; V_0, \ldots, V_m) \cong \psi(S_0 \cap S_1; U_0 \cap V_0, \ldots, U_m \cap V_m)$.

The next theorem shows how to exploit the nice combinatorial properties of pseudospheres. It states that if applying a protocol to a single simplex preserves connectivity below some dimension, then applying that protocol to any input pseudosphere also preserves that degree of connectivity. It is actually a theorem in topology, and so it applies to any model of computation.

Theorem 4.3 *Let $\mathcal{P}$ be a protocol, S^m be a simplex, and c be a constant. If for every face S^ℓ of S^m and for every sequence $V_0, \ldots, V_\ell$ of singleton sets the protocol complex $\mathcal{P}(\psi(S^\ell; V_0, \ldots, V_\ell))$ is $(\ell - c - 1)$-connected, then for every sequence $U_0, \ldots, U_m$ of nonempty sets the protocol complex $\mathcal{P}(\psi(S^m; U_0, \ldots, U_m))$ is $(m - c - 1)$-connected.*

A consequence of this theorem is that any n-dimensional pseudosphere is $(n-1)$-connected (just let $\mathcal{P}$ be the trivial protocol in which each process halts immediately):

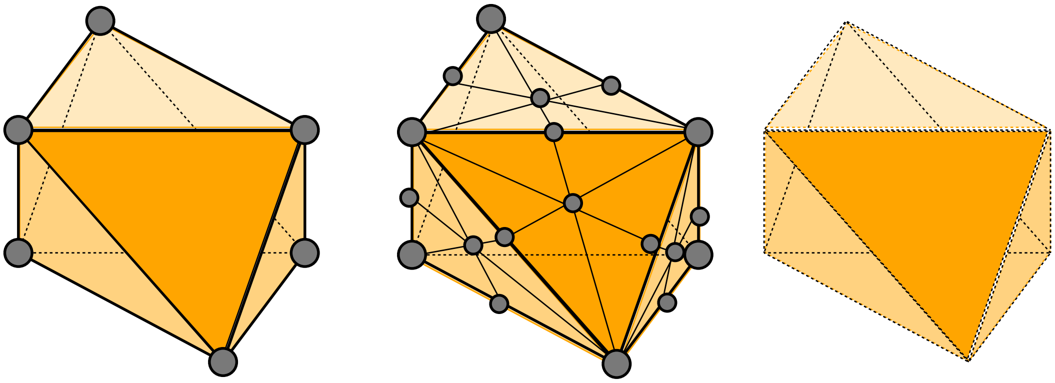

Fig. 5. Simplicial complex, subdivision, and polyhedron

Corollary 4.4 *If $U_0, \ldots, U_m$ are all nonempty, then $\psi(S^m; U_0, \ldots, U_m)$ is $(m-1)$-connected.*

Naively, one might think that S^m is always m-connected, but note that although the empty simplex has dimension -1, it is not (-1)-connected. We can generalize Theorem 4.3 to multiple pseudospheres.

Theorem 4.5 *Let $\mathcal{P}$ be a protocol satisfying the precondition of Theorem 4.3, and let $A_0, \ldots, A_\ell$ be a sequence of finite sets. If $\cap_{i=0}^{\ell} A_i \neq \emptyset$ then*

$$\mathcal{P}\left(\bigcup_{i=0}^{\ell} \psi(S^m; A_i)\right) \text{ is } (m-c-1)\text{-connected.}$$

Letting $\mathcal{P}$ be the trivial protocol in which each process decides its input:

Corollary 4.6 *If $A_0, \ldots, A_\ell$ is a sequence of finite sets such that $\cap_{i=0}^{\ell} A_i \neq \emptyset$ then*

$$\bigcup_{i=0}^{\ell} \psi(S^m; A_i) \text{ is } (m-1)\text{-connected.}$$

5 Connectivity vs k-Set Agreement

The notion of k-connectivity lies at the heart of all known lower bounds for k-set agreement. In this section, we prove a general theorem linking $(k-1)$-connectivity with impossibility of k-set agreement. This theorem is model independent in the sense that it depends on the connectivity properties of protocol complexes, not on explicit timing or failure properties of the model. This result was originally stated elsewhere [HR94], but for the sake of making this paper self-contained, we present the full proof here.

The point-set occupied by a complex $\mathcal{C}$ is called its *polyhedron*, and is denoted by $|\mathcal{C}|$. Any simplicial map $\phi : \mathcal{A} \to \mathcal{B}$ induces a piece-wise linear map $|\phi| : |\mathcal{A}| \to |\mathcal{B}|$ that agrees with ϕ on vertexes of $\mathcal{A}$.

A *subdivision* of a complex $\mathcal{A}$ is a complex $\mathcal{B}$ such that (1) each simplex of $\mathcal{B}$ is contained in a simplex of $\mathcal{A}$, and (2) each simplex of $\mathcal{A}$ is the union of finitely many simplexes of $\mathcal{B}$ [Mun84, p. 83]. This definition implies that $|\mathcal{A}| = |\mathcal{B}|$. If $\vec{b}$ is a vertex of $\mathcal{B}$, the *carrier* of $\vec{b}$ in $\mathcal{A}$, denoted $\mathit{carrier}(\vec{b}, \mathcal{A})$, is the smallest simplex of $\mathcal{A}$ that contains $\vec{b}$. Figure 5 illustrates a complex, a subdivision of that complex, and their underlying polyhedron.

We will need a step-by-step method for constructing subdivisions. Let $\mathcal{C}$ be a complex, and $\vec{w}$ a point with the property that any ray emanating from $\vec{w}$ intersects $|\mathcal{C}|$ in at most one point. Define the *cone* $\vec{w} \cdot \mathcal{C}$ to be the collection of all simplexes of the form $(\vec{w}, \vec{s}_0, \ldots, \vec{s}_k)$, where $(\vec{s}_0, \ldots, \vec{s}_k)$ is a simplex of $\mathcal{C}$, together with all faces of such simplexes. This cone is itself a complex, having $\mathcal{C}$ as a subcomplex [Mun84, p. 44]. Let σ be a subdivision of $\mathit{skel}^{\ell-1}(\mathcal{C})$, and $S^\ell_0, \ldots, S^\ell_L$ the ℓ-simplexes of $\mathit{skel}^\ell(\mathcal{C})$. For $0 \leq i \leq L$, let $\vec{w}_i$ be an interior point of $|S^\ell_i|$. Each cone $\vec{w}_i \cdot \sigma(S^\ell_i)$ is a subdivision of S^ℓ_i, and the union of these cones as i ranges from 0 to L is a subdivision of $\mathit{skel}^\ell(\mathcal{C})$ that agrees with σ on the $(\ell - 1)$ skeleton [Mun84, p. 85]. The result is called the subdivision of $\mathit{skel}^\ell(\mathcal{C})$ obtained by *starring* σ. The subdivision shown in Figure 5 is the result of successive starring.

We use the following variant of Sperner's Lemma [Lef49, Lemma 5.5]:

Lemma 5.1 (Sperner's Lemma) *Let $\sigma(S^n)$ be a subdivision of simplex S^n. If $F : \sigma(S^n) \to S^n$ is a map sending each vertex of $\sigma(S^n)$ to a vertex in its carrier, then there is at least one n-simplex $T^n = (\vec{t}_0, \ldots, \vec{t}_n)$ in $\sigma(S^n)$ such that the $F(\vec{t}_i)$ are all distinct.*

We also exploit the following extension lemma, which appears in Glaser [Gla70, Theorem IV.2].

Lemma 5.2 *Let $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ be complexes such that $\mathcal{A} \subset \mathcal{B}$, and $f : |\mathcal{B}| \to |\mathcal{C}|$ is a continuous map such that f restricted to $|\mathcal{A}|$ is simplicial. There exists a subdivision τ of $\mathcal{B}$ such that $\tau(\mathcal{A}) = \mathcal{A}$, and a simplicial map $\phi : \tau(\mathcal{B}) \to \mathcal{C}$ extending the restriction of f to $|\mathcal{A}|$.*

Theorem 5.3 *Let $V = \{v_0, \ldots, v_k\}$ be a set of $k + 1$ possible input values, and $\mathcal{P}$ a protocol with input complex $\psi(P_0, \ldots, P_n; V)$. If $\mathcal{P}$ has the property that for every n-dimensional pseudosphere $\psi(P_0, \ldots, P_n; U)$, where U is a nonempty subset of V, $\mathcal{P}(\psi(P_0, \ldots, P_n; U)$ is $(k - 1)$-connected, then $\mathcal{P}$ cannot solve k-set agreement.*

Theorems 4.3 and 5.3 imply

Corollary 5.4 *If $\mathcal{P}(\mathcal{S}^m)$ is $(m - (n - k) - 1)$-connected for all m where with $n - f \leq m \leq n$, then $\mathcal{P}$ cannot solve k-set agreement in the presence of f failures.*

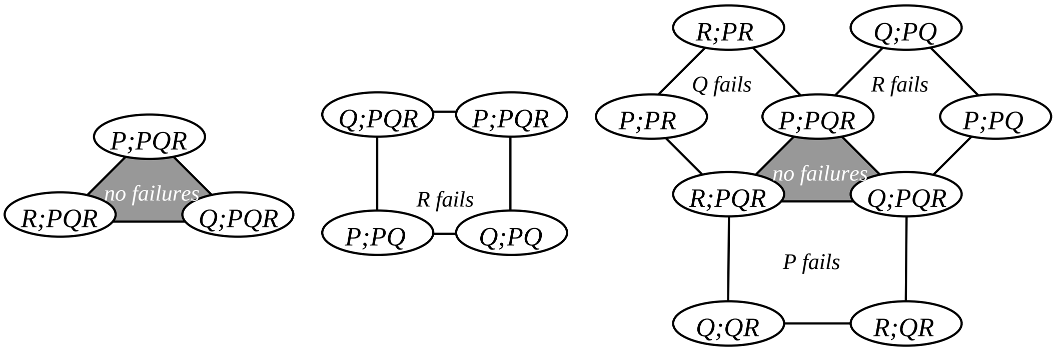

Fig. 6. Construction of a one-round three-process protocol complex.

6 Synchronous Computation

We now define the r-round synchronous protocol complex $\mathcal{S}^r(S^m)$. Here too, we consider only a subset of all possible executions: executions in which no more than k processes fail in any round. We are interested in executions where no more than k processes fail in any round. Informally, we will show that the one-round protocol complex is the union of pseudospheres, where each pseudosphere corresponds to the set of executions in which a fixed set of processes fail. For example, Figure 6 illustrates the possible executions of a one-round protocol for three processes, P, Q, and R, starting from a fixed input simplex, in which no more than one process fails. Here, each vertex is labeled with a process, followed by the processes from which it has received messages. The figure on the left represents the execution in which in which no processes fail: this is a (degenerate) pseudosphere in which each process receives the same set of messages. The figure in the middle represents the executions in which R alone fails. This complex is a pseudosphere: P and Q independently do or do not receive a message from R. The figure on the right represents the entire one-faulty protocol complex. It is the union of the failure-free pseudosphere with the three single-failure pseudospheres.

6.1 Single-Round Protocols

Let $\mathcal{S}^1(S^n)$ be the complex of one-round executions of an $(n+1)$-process protocol with input simplex S^n in which at most k processes fail. It is the union of complexes $\mathcal{S}^1_K(S^n)$ of one-round executions starting from S^n in which *exactly* the processes in K fail. Given a set K of process ids, let $S^n \backslash K$ be the face of S^n labeled with the process ids *not* in K. Our next result says that $\mathcal{S}^1_K(S^n)$ is a pseudosphere, which means that $\mathcal{S}^1(S^n)$ is a union of pseudospheres:

Lemma 6.1 *If $m \geq n - f$ and K is a subset of ids(S^m) of size at most $f - (n - m)$, then*

$$\mathcal{S}^1_K(S^m) \cong \psi(S^m \backslash K; 2^K).$$

The one-round complex is a union of pseudospheres in the synchronous model (Lemma 6.1). To compute the connectivity of this union using Theorem 2.2, we need to understand the intersections. The next lemma shows that these intersections have a simple structure: they are themselves the union of pseudospheres. Order the process sets lexicographically: the empty set first, followed by singleton sets, followed by two-element sets, and so on. Let $K_0, \ldots, K_\ell$ be the sequence of sets of process ids less than or equal to K_ℓ, listed in lexicographic order.

Lemma 6.2 *Let $m \geq n - f$ and let $K_0, \ldots, K_k$ be the subsets of ids(S^m) of size at most $f - (n-m)$ arranged in lexicographical order. If $K_0, \ldots, K_\ell$ is a prefix of this sequence, then*

$$\bigcup_{i=0}^{\ell-1} \mathcal{S}^1_{K_i}(S^m) \cap \mathcal{S}^1_{K_\ell}(S^m) = \bigcup_{p \in K_\ell} \psi(S^m \backslash K_\ell; 2^{K_\ell - \{p\}}).$$

Let $\mathcal{S}^1(S^n)$ denote the protocol complex for a one-round synchronous $(n+1)$-process protocol with input simplex S^n where no more than than k processes fail.

Lemma 6.3 *$\mathcal{S}^1(S^m)$ is $(m - (n-k) - 1)$-connected if $m \geq (n-f) + k$ and $n \geq 2k$.*

6.2 Multi-Round Protocols

Let $\mathcal{S}^r(S^n)$ be the protocol complex for an r-round synchronous $(n+1)$-process protocol with input simplex S^n where no more than than k processes fail in each round. We can decompose this complex as follows. Let $K_0, \ldots, K_\ell$ be a sequence of sets of k or fewer process ids in lexicographic order. Recall that $\mathcal{S}^1_{K_i}(S^n) = \psi(S^n \backslash K_i; 2^{K_i})$ is the complex of one-round executions in which exactly the processes in K_i fail. The set of r-round executions in which exactly the processes in K_i fail in the first round can be written as $\mathcal{S}^{r-1}_i(\mathcal{S}^1_{K_i}(S^n))$, where $\mathcal{S}^{r-1}_i$ is the complex for an $(r-1)$-round, $(f - |K_i|)$-faulty, $(n - |K_i| + 1)$-process full-information protocol. The $\mathcal{S}^{r-1}_i$ are considered distinct protocols because the $\mathcal{S}^1_{K_i}(S^n)$ have varying dimensions. Taking the union over all the K_i, we have

$$\mathcal{S}^r(S^n) = \bigcup_{i=0}^{\ell} \mathcal{S}^{r-1}_i(\mathcal{S}^1_{K_i}(S^n)).$$

The connectivity of a protocol $\mathcal{P}$ depends on the *degree* of the protocol. Consider the multi-round executions of $\mathcal{P}$ in which f_i is the maximum number of processes that fail at round i. The *degree* of $\mathcal{P}$ is the minimum f_i for any round. Define $\widetilde{\mathcal{S}}^{r-1}_\ell$ to be the protocol identical to $\mathcal{S}^{r-1}_\ell$ except that it fails at most $k-1$ processes in its first round. While $\mathcal{S}_\ell^{r-1}$ has degree k, $\widetilde{\mathcal{S}}_\ell^{r-1}$ has degree $k-1$. Our next result implies that intersections of the complexes comprising $\mathcal{S}^r$ are equivalent to $\widetilde{\mathcal{S}}_\ell^{r-1}$ applied to a union of pseudospheres, which makes it possible to use Theorem 2.2 to analyze the connectivity of $\mathcal{S}^r$.

Lemma 6.4

$$\bigcup_{i=0}^{\ell-1} \mathcal{S}_i^{r-1}(\mathcal{S}_{K_i}^1(S^n)) \cap \mathcal{S}_\ell^{r-1}(\mathcal{S}_{K_\ell}^1(S^n)) = \widetilde{\mathcal{S}}_\ell^{r-1}\left(\bigcup_{j\in K_\ell} \psi(S^n \backslash K_\ell; 2^{K_\ell - \{j\}})\right).$$

Define

$$\mathcal{S}_K^P(S^m) = \begin{cases} \mathcal{T} \text{ if } \mathit{ids}(S^m) \subseteq P \text{ and } P - \mathit{ids}(S^m) \subseteq K \\ \emptyset \ \text{ otherwise} \end{cases}$$

where $\mathcal{T}$ is the complex of one-round executions of the full-information protocol in which only the processes in K fail, starting with the processes in P and the input simplex S^m. The condition $\mathit{ids}(S^m) \subseteq P$ says that initial inputs are provided for some of the processes, and the condition $P - \mathit{ids}(S^m) \subseteq K$ says that processes for which no input is provided can be considered to have failed immediately before having sent a single message. In general, define

$$\mathcal{S}^P_{K_r,K_{r-1},\ldots,K_1}(S^m) = \mathcal{S}^{P_r}_{K_r}\mathcal{S}^{P_{r-1}}_{K_{r-1}}\cdots\mathcal{S}^{P_1}_{K_1}(S^m) \qquad \text{where } P_i = P - \bigcup_{j=1}^{i-1} K_j.$$

A consequence of these definitions is that the complex $\mathcal{S}^P_{K_r,K_{r-1},\ldots,K_1}(S^m)$ is the empty set unless $\mathit{ids}(S^m) \subseteq P$ and $P - \mathit{ids}(S^m) \subseteq K_1$.

Define a *failure pattern* to be a sequence $\sigma = \sigma_r, \ldots, \sigma_1$ of integers representing upper bounds on the number of processes allowed to fail in each of the first r rounds of the full-information protocol. The failure pattern of most interest to us will be the failure pattern in which k processes fail in each round. We say that the failure pattern σ has *degree* k if it is a nondecreasing sequence of integers

$$k = \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_r$$

beginning with k. We say that a sequence $\Sigma = K_r, \ldots, K_1$ of process sets *satisfies* σ if $|K_i| \leq \sigma_i$ for each i, and we write $\Sigma \sim \sigma$. Generalizing $\mathcal{S}^P_{K_r,K_{r-1},\ldots,K_1}(S^m)$, we define

$$\mathcal{S}^P_\sigma(S^m) = \bigcup_{\Sigma\sim\sigma} \mathcal{S}^P_\Sigma(S^m)$$

to be the complex of r-round executions of the full-information protocol with failure pattern σ, starting with the processes in P and the input simplex S^m. We say that $\mathcal{S}^P_\sigma(S^m)$ has degree k if σ has degree k.

Lemma 6.5 *Let $\sigma = (k_r, k_{r-1}, \ldots, k_1)$ be a failure pattern and P be a set of processors. Let $\tau = (k_r, k_{r-1}, \ldots, k_2)$ and $\tau' = (k_r, k_{r-1}, \ldots, k_2 - 1)$, and let $K_1, \ldots, K_k$ be the subsets of P of size at most k_1 listed in lexicographical order.*

$$\bigcup_{i=0}^{\ell-1} \mathcal{S}_{\tau}^{P-K_i}(\mathcal{S}_{K_i}^{P}(S^m)) \cap \mathcal{S}_{\tau}^{P-K_\ell}(\mathcal{S}_{K_\ell}^{P}(S^m))$$

$$= \mathcal{S}_{\tau'}^{P-K_\ell}\left(\bigcup_{p \in K_\ell} \psi(S^m \backslash K_\ell; 2^{K_\ell - \{p\}})\right).$$

Lemma 6.6 *Let $\sigma = (k_r, \ldots, k_1)$ be a failure pattern of degree k, and let P be a set of processors of size n. If $n \geq k_r + \cdots + k_1 + k$ and $ids(S^m) \subseteq P$, then $\mathcal{S}_{\sigma}^{P}(S^m)$ is $(m - (n - k) - 1)$-connected.*

The connectivity of this protocol complex implies the lower bound for synchronous k-set agreement [CHLT93]:

Theorem 6.7 *If $n \geq f + k$, then any synchronous f-resilient k-set agreement protocol requires $\lfloor f/k \rfloor + 1$ rounds. If $n < f + k$, then any synchronous f-resilient k-set agreement protocol requires $\lfloor f/k \rfloor$ rounds.*

Proof. If $n - k \geq f$, then $\mathcal{S}^{\lfloor f/k \rfloor}(\mathcal{I})$ is $(k-1)$-connected. If $n - k < f$, then $\mathcal{S}^{\lfloor f/k \rfloor - 1}(\mathcal{I})$ is $(k-1)$-connected. Either way, Theorem 5.3 states that the protocol cannot solve k-set agreement.

References

[AR96] Hagit Attiya and Sergio Rajsbaum. The combinatorial structure of wait-free solvable tasks. In *Proceedings of the 10th International Workshop on Distributed Algorithms*, volume 1151 of *Lecture Notes in Computer Science*, pages 322–343. Springer-Verlag, Berlin, October 1996.

[BG93] Elizabeth Borowsky and Eli Gafni. Generalized FLP impossibility result for t-resilient asynchronous computations. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 91–100, May 1993.

[Cha91] Soma Chaudhuri. Towards a complexity hierarchy of wait-free concurrent objects. In *Proceedings of the 3rd IEEE Symposium on Parallel and Distributed Processing*, December 1991.

[Cha93] Soma Chaudhuri. More choices allow more faults: Set consensus problems in totally asynchronous systems. *Information and Computation*, 105(1):132–158, July 1993.

[CHLT93] Soma Chaudhuri, Maurice Herlihy, Nancy Lynch, and Mark R. Tuttle. A tight lower bound for k-set agreement. In *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, pages 206–215, November 1993.

[DM90] Cynthia Dwork and Yoram Moses. Knowledge and common knowledge in a Byzantine environment: Crash failures. *Information and Computation*, 88(2):156–186, October 1990.

[Dol82] Danny Dolev. The Byzantine generals strike again. *Journal of Algorithms*, 3(1):14–30, March 1982.

[DS83] Danny Dolev and H. Raymond Strong. Authenticated algorithms for Byzantine agreement. *SIAM Journal on Computing*, 12(3):656–666, November 1983.

[Fis83] Michael J. Fischer. The consensus problem in unreliable distributed systems (a brief survey). In Marek Karpinsky, editor, *Proceedings of the 10th International Colloquium on Automata, Languages, and Programming*, pages 127–140. Springer-Verlag, 1983.

[FL82] Michael J. Fischer and Nancy A. Lynch. A lower bound for the time to assure interactive consistency. *Information Processing Letters*, 14(4):183–186, June 1982.

[FLP85] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty processor. *Journal of the ACM*, 32(2):374–382, April 1985.

[GK99] Eli Gafni and Elias Koutsoupias. Three-processor tasks are undecidable. *SIAM Journal on Computing*, 28(3):970–983, 1999.

[Gla70] L. C. Glaser. *Geometrical Combinatorial Topology*, volume 1. Van Nostrand Reinhold, New York, 1970.

[Had83] Vassos Hadzilacos. A lower bound for Byzantine agreement with fail-stop processors. Technical Report TR–21–83, Harvard University, 1983.

[HR94] Maurice Herlihy and Sergio Rajsbaum. Set consensus using arbitrary objects. In *Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing*, pages 324–333, August 1994.

[HR95] Maurice Herlihy and Sergio Rajsbaum. Algebraic spans. In *Proceedings of the 14th Annual ACM Symposium on Principles of Distributed Computing*, pages 90–99, August 1995. *Mathematical Structures in Computer Science*, to appear.

[HRT98] Maurice Herlihy, Sergio Rajsbaum, and Mark R. Tuttle. Unifying synchronous and asynchronous message-passing models. In *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing*, pages 133–142. ACM, June 1998.

[HS99] Maurice P. Herlihy and Nir Shavit. The topological structure of asynchronous computability. *Journal of the ACM*, November 1999.

[Lef49] S. Lefschetz. *Introduction to Topology.* Princeton University Press, Princeton, New Jersey, 1949.

[LSP82] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, July 1982.

[Mer85] Michael Merritt. Notes on the Dolev-Strong lower bound for byzantine agreement. Unpublished manuscript, 1985.

[Mun84] J. R. Munkres. *Elements Of Algebraic Topology.* Addison Wesley, Reading MA, 1984.

[PSL80] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, 1980.

[Spa66] Edwin H. Spanier. *Algebraic Topology.* Springer-Verlag, New York, 1966.

[SZ93] Michael Saks and Fotis Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 101–110, May 1993. *SIAM Journal on Computing*, to appear.