An Axiomatic Approach to Computing the Connectivity of Synchronous and Asynchronous Systems

Maurice Herlihy* Sergio Rajsbaum† Mark Tuttle‡

Abstract

We present a unified, axiomatic approach to proving lower bounds for the k-set agreement problem in both synchronous and asynchronous message-passing models. The proof involves constructing the set of reachable states, proving that these states are highly connected, and then appealing to a well-known topological result that high connectivity implies that set agreement is impossible. We construct the set of reachable states in an iterative fashion using a round operator that we define, and our proof of connectivity is an inductive proof based on this iterative construction and simple properties of the round operator.

1 Introduction

The consensus problem [18] has received a great deal of attention. In this problem, $n+1$ processors begin with input values, and all must agree on one of these values as their output value. Fischer, Lynch, and Paterson [7] surprised the world by showing that solving consensus is impossible in an asynchronous system if one processor is allowed to fail. This leads one to wonder if there is any way to weaken consensus to obtain a problem that can be solved in the presence of $k-1$ failures but not in the presence of k failures. Chaudhuri [5] defined the k-set agreement problem and conjectured that this was one such problem, and a trio of papers [4, 13, 19] proved that she was right. The k-set agreement problem is a generalization of consensus, where we relax the requirement that processors agree on a single value: the set of output values chosen by the processors may contain as many as k distinct values, and not just 1.

Set agreement (and in particular consensus) has been studied in both synchronous and asynchronous models of computation, but mostly independently.

*Brown University, Computer Science Department, Providence, RI 02912; mph@cs.brown.edu.

†Instituto de Matemáticas, Universidad Nacional Autónoma de México, Ciudad Universitaria, D.F. 04510, Mexico; rajsbaum@math.unam.mx

‡HP Labs, One Cambridge Center, Cambridge, MA 02142; mark.tuttle@hp.com.

Indeed, prior proofs for these models appeared to have little in common, as reflected by the organization of a main textbook in the area [14], where the first part is devoted to synchronous systems and the second part of the book to asynchronous systems. Recent work has been uncovering more and more features and structure in common to both models e.g. [8, 12, 15, 16]. However, these results are in the form of transformations between models, or on proofs that have a similar structure in both models. Only [15] describes an abstract model that encompasses both models, with clearly identified properties that are needed to carry out consensus impossibility results. To go from consensus to set agreement a big step in complexity is encountered, since one must deal with higher dimensional topology instead of just graphs, as discovered by the trio of papers [4, 13, 19] mentioned above. The contribution of this paper is to present a new axiomatic approach were set consensus impossibility proofs can be derived in a uniform manner for both synchronous and asynchronous models.

All known proofs for the set agreement lower bound depend — either explicitly or implicitly — on a deep connection between computation and topology. These proofs essentially consider the simplicial complex representing all possible reachable states of a set agreement protocol, and then argue about the connectivity of this complex. These lower bounds for set agreement follow from the observation that set agreement cannot be solved if the complex of reachable states is sufficiently highly-connected. This connection between connectivity and set agreement has been established both in a generic way [11] and in ways specialized to particular models of computation [1, 4, 6, 10, 11, 12, 19]. Once the connection has been established, however, the problem reduces to reasoning about the connectivity of a protocol's reachable complex.

The primary contribution of this work is a new, substantially simpler proof of how the connectivity of the synchronous and asynchronous complexes evolve over time. Our proof depends on two key insights:

1. The notion of a *round operator* that maps a global state to the set of global states reachable from this state by one round of computation, an operator satisfying a few simple algebraic properties.

2. The notion of an *absorbing poset* organizing the set of global states into a partial order, from which the connectivity proof follows easily using the round operator's algebraic properties.

We believe this new approach has several novel and elegant features. First, we are able to isolate a small set of elementary combinatorial properties of the round operator that suffice to establish the connection with classical topology in a model-independent way. Second, these properties require only local reasoning about how the computation evolves from one round to the next. Finally, most connectivity arguments can be difficult to follow because they mix semantic, combinatorial, and topological arguments. Those arguments are cleanly separated here. The round operator definition relies on semantics: it is a combinatorial restatement of the properties of the synchronous model. Once the round operator is defined, however, we need no further appeals to properties of the original model. We reason in a purely combinatorial way about intersections of global states, and how they can be placed in a partial order. Once these combinatorial arguments are in place, we appeal directly to well-known theorems of topology to establish connectivity. These topology theorems are treated as "black boxes," in the sense that we apply them directly without any need to make additional topological arguments. Furthermore, our absorbing posets are very similar to shellable complexes e.g. [3] so we have uncovered yet one more link between the work of topologists and distributed computing.

For lack of space most of the proofs have been omitted, but appear in the full paper.

2 Preliminaries

2.1 Models

We consider two (standard) message-passing models, the *synchronous* and *asynchronous models*. In both models, we restrict our attention to computations with a round structure: the initial state of each processor is its input value, and computation proceeds in a sequence of rounds. In each round, each processor sends messages to other processors, receives messages sent to it by the other processors in that round, performs some internal computation, and changes state. We assume that processors are following a full-information protocol, which means that each processor sends its entire local state to every processor in every round. This is a standard assumption to make when proving lower bounds. A processor can fail by crashing in the middle of a round, in which case it sends its state only to a subset of the processors in that round. Once a processor crashes, it never sends another message after that.

In the synchronous model [2, 14], all processors execute round r at the same time, and processor P fails to receive a message from processor Q, then Q must have crashed, either in that round or in the previous round.

In the asynchronous model, there is no bound on processor step time nor on message delivery time, so a crashed processor cannot be distinguished from a slow processor. Our results, however, depend only on the unbounded message delivery time. Since our goal is to prove impossibility results, we are free to restrict our attention to executions in which processors take steps at a regular pace, and only message delivery times are delayed. In the behaviors we consider, messages from one processor to another are delivered in FIFO order, but when one message from P to Q is delivered, all outstanding messages from P to Q are delivered at the same time.

It is convenient to recast the asynchronous model in the following *omissions-failure* form. There are at most f potentially faulty processors. At each round, the nonfaulty processors broadcast their states to all processors (including the faulty processors). Each faulty processor broadcasts its state to some subset of the processors, and may omit to send to the others. Processors never crash. It can be shown that k-set agreement lower bounds in this omissions failure model

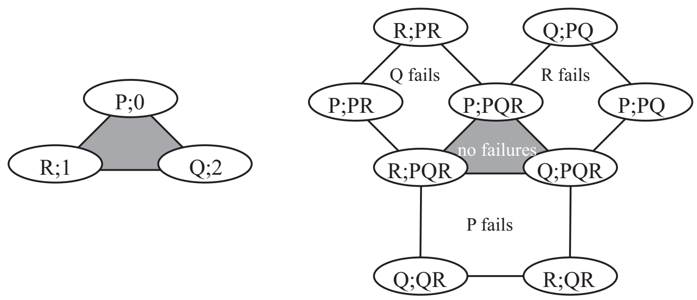

Figure 1: A global state S and the set $\mathcal{S}_1(S)$ of global states after one round from S.

carry over to the standard asynchronous crash-failure model; see [9] for a similar argument.

2.2 Combinatorial Topology

We represent the local state of a processor with a vertex labeled with that processor's id and its local state. We represent a global state as a set of labeled vertexes, labeled with distinct processors, representing the local state of each processor in that global state. In topology, a *simplex* is a set of vertexes, and a *complex* is a set of simplexes that is closed under containment. The *dimension* of a simplex is equal to its number of vertexes minus one. Applications of topology to distributed computing often assume that these vertexes are points in space and that the simplex is the convex hull of these points in order to be able to use standard topology results. As you read this paper, you might find it helpful to think of simplexes in this way, but in the purely combinatorial work done in this paper, a simplex is just a set of vertexes.

As an example, consider the simplex and complex illustrated in Figure 1. On the left side, we see a simplex representing an initial global state in which processors P, Q, and R start with input values 0, 2, and 1. Each vertex is labeled with a processor's id and its local state (which is just its input value in this case). On the right we see a complex representing the set of states that arise after one round of computation from this initial state if one processor is allowed to crash. The labeling of the vertexes is represented schematically by a processor id such as P and a string of processor ids such as PQ. The string PQ is intended to represent the fact that P heard from processors P and Q during the round but not from R, since R failed that round. (We are omitting input values on the right for notational simplicity.) The simplexes that represent states after one round are the 2-dimensional triangle in the center and the 1-dimensional edges that radiate from the triangle (including the edges of the triangle itself). The central triangle represents the state after a round in which no processor fails. Each edge represents a state after one processor failed. For example, the edge with vertexes labeled $P; PQR$ and $Q; PQ$ represent the global state after a round in which R fails by sending a message to P and not sending to Q: P heard from all three processors, but Q did not hear from R.

What we do in this paper is define round operators like the round operator $\mathcal{S}_1$ that maps the simplex S on the left of Figure 1 to the complex $\mathcal{S}_1(S)$ on the right, and then reason about the connectivity of $\mathcal{S}_1(S)$. Informally, connectivity in dimension 0 is just ordinary graph connectivity, and connectivity in higher dimensions means that there are no "holes" of that dimension in the complex. When we reason about connectivity, we often talk about the connectivity of a simplex S when we really mean the connectivity of the induced complex consisting of S and all of its faces. For example, both of the complexes in Figure 1 are 0-connected since they are connected in the graph theoretic sense. In fact, the complex on the left is also 1-connected, but the complex on the right is not since there are "holes" formed by the three cycles of 1-dimensional edges.

Given a simplex S, a *labeling* of S from a set V is a new simplex constructed by replacing each vertex s of S with a pair (s, v), where $v \in V$.

Given a simplex S and a set V, we define the *pseudosphere* $\mathcal{P}(S, V)$ to be this set of labelings of S with elements of V. (We call $\mathcal{P}(S, V)$ a pseudosphere because is has some of the topological properties of a sphere.) The face S is called the *base simplex* of the pseudosphere, and given a simplex T of a pseudosphere $\mathcal{P}(S, V)$, we define *base*(T) to be the base simplex S of the pseudosphere.

The input complex for k-set agreement is $\mathcal{P}(S, V)$, the pseudosphere in which each vertex is labeled with an input from a set V, where $|V| > k$. The set of all reachable states of a protocol P with initial states $\mathcal{P}(S, V)$ is the protocol complex $\mathcal{C} = \mathcal{C}(\mathcal{P}(S, V))$. The fundamental connection between k-set agreement and connectivity is expressed in the following theorem (e.g. [11]):

Theorem 1: Let P be a protocol, and let $\mathcal{C}$ be its protocol complex. If $\mathcal{C}$ is $(k-1)$-connected, then P cannot solve k-set agreement.

Thus, our main task will be to prove that $\mathcal{C}$ is $(k-1)$-connected. Proving that a union of complexes is connected is made easier by the following theorem[1]. Notice that if A and B are complexes then both $A \cup B$ and $A \cap B$ are complexes.

Theorem 2 (Mayer-Vietoris): Let A, B be two complexes. Then $A \cup B$ is c-connected if A and B are c-connected and $A \cap B$ is $(c-1)$-connected.

Think about the special one-dimensional case of this statement: a graph that is the union of subgraphs A, B is 0-connected (connected in the graph theoretic sense) if A and B are 0-connected and $A \cap B$ is -1-connected (nonempty).

To prove that a complex $\mathcal{C}$ is c-connected, we split $\mathcal{C}$ into subcomplexes with less and less simplexes, and apply repeatedly the Mayer-Vietoris theorem. At the bottom of this recursion, we get complexes with just one simplex, and use the following fact.

[1] Actually this theorem is a well-known corollary of the Mayer-Vietoris sequence, which is described in most algebraic topology textbooks; see for example [20] Chapter 4, Section 6.

Theorem 3: A simplex of dimension at least ℓ is $(\ell-1)$-connected.

In this paper all we need to assume from topology is the previous two theorems. Both are very basic algebraic topology facts that appear in standard textbooks such as [17, 20].

3 Absorbing Posets and Round Operators

The *codimension* of two simplexes S_0 and S_1 is a measure of how much they have in common defined by

$$\mathit{codim}(S_0, S_1) = \max_i \{\dim(S_i) - \dim(\cap_j S_j)\}$$

where $\dim(\emptyset) = -1$ is the dimension of the empty simplex. Two useful properties of this definition are that if $S \subseteq T$ then

$$\mathit{codim}(S, T) = \dim(T) - \dim(S),$$

and if $S \subseteq X \subseteq T$ then

$$\mathit{codim}(S, T) = \mathit{codim}(S, X) + \mathit{codim}(X, T).$$

Let $\mathcal{S}$ be a nonempty set of simplexes, and $\preceq$ a partial order on $\mathcal{S}$.

Definition 4: We say that $(\mathcal{S}, \preceq)$ is an *absorbing poset* if for every two simplexes S and T in $\mathcal{S}$ with $T \not\preceq S$ there is a T_S in $\mathcal{S}$, $T_S \preceq T$ such that

$$S \cap T \subseteq T_S \cap T \quad (1)$$
$$\mathit{codim}(T_S, T) = 1 \quad (2)$$
$$\mathit{codim}(S \cap T, T_S) \leq \mathit{codim}(S, T). \quad (3)$$

The first two properties say that when considering pairwise intersections of simplexes — as we will frequently do in our Mayer-Vietoris arguments — pairs of high codimension are "absorbed" by pairs of low codimension, and we can restrict our attention to pairs of simplexes of codimension one. The third property just says that T_S satisfies the same property that S and T do, namely, $\mathit{codim}(S \cap T, X) \leq \mathit{codim}(S, T)$ for $X = S, T$. An absorbing poset is almost equivalent to a *shellable complex* [3]. In a shellable complex, Equations 2 and 3 apply only to principal faces ("facets") of the complex, while our construction allows one complex in $\mathcal{S}$ to be a proper face of another. It follows that every absorbing poset induces a shellable complex, but not vice-versa.

Lemma 5: If $\mathcal{A}$ is a set of simplexes such that every pair of simplexes has codimension 1, then $(\mathcal{A}, <)$ is an absorbing poset, where $<$ is any total order on $\mathcal{A}$.

Proof: For any simplexes S and T in $\mathcal{A}$ such that $S < T$, pick $T_S = S$. Substituting S for T_S, it is easy to check that the three conditions of Definition 4 are satisfied:

$$S \cap T \subseteq S \cap T$$
$$\mathit{codim}(S, T) = 1$$
$$\mathit{codim}(S \cap T, S) \leq \mathit{codim}(S, T).$$

3.1 Axioms

A *simplicial operator* $\mathcal{Q}$ is a family of maps. Each map $\mathcal{Q}_\ell$ carries a simplex of dimension $m \geq \ell$ to a nonempty set of simplexes, where each simplex has dimension at most m. The subscript ℓ is the operator's *degree*. For $\ell < 0$, it is convenient to define $\mathcal{Q}_\ell(S)$ to be the empty set. Note that $\mathcal{Q}_\ell(\emptyset) = \emptyset$ for all ℓ, and $\mathcal{Q}_0(S) \neq \emptyset$ for any nonempty simplex S.

Simplicial operators extend naturally to sets of simplexes. If $\mathcal{A}$ is a set of simplexes,

$$\mathcal{Q}_\ell(\mathcal{A}) = \bigcup_{A \in \mathcal{A}} \mathcal{Q}_\ell(A). \tag{4}$$

The exact meaning of the operator will vary from model to model. In the synchronous message-passing model, ℓ is the number of processors that can crash in each round. In the asynchronous model, ℓ is the number of processors that remain partially silent in each round.

We use $\mathcal{Q}_k \mathcal{Q}_\ell(S)$ to denote the composition of $\mathcal{Q}_k$ and $\mathcal{Q}_\ell$ applied to S, $\mathcal{Q}_\ell^r(S)$ to denote the r-fold composition of $\mathcal{Q}_\ell$ applied to S, and $\|\mathcal{Q}_\ell^r(S)\|$ to denote the simplicial complex induced by the set $\mathcal{Q}_\ell^r(S)$ (i.e., closed under containment).

The first axiom says that the states reachable after the failure of ℓ processors are reachable after the failure of even more processors.

Axiom 1:

$$\mathcal{Q}_\ell(S) \subseteq \mathcal{Q}_m(S)$$

when $\ell \leq m$.

The next axiom describes multi-round executions. We introduce a model-specific, integer-valued linear function ϕ. Informally, $\phi(f)$ is the number of failures needed in a round to hide the existence of f faulty processors. We will see that in the synchronous model, faulty processors crash, so $\phi(f) = 0$. In the asynchronous model, faulty processors fail to send messages, so $\phi(f) = f$.

Axiom 2: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{Q}_k^r \mathcal{Q}_\ell(S_0)\| \cap \|\mathcal{Q}_k^r \mathcal{Q}_\ell(S_1)\| = \|\mathcal{Q}_{k-\phi(c)}^r \mathcal{Q}_{\ell-c}(S_0 \cap S_1)\|.$$

The right-hand-side of this equation is the set of states for processors that cannot tell whether the initial state was S_0 or S_1. The processors that can tell the difference must be silenced in the first round, requiring an extra c failures, and must be kept silent for the remaining rounds, requiring $\phi(c)$ extra failures in each subsequent round.

Axiom 3: For every simplex S, $\mathcal{Q}_\ell(S)$ is an absorbing poset.

4 Theorems and Lemmas

Lemma 6: Let $i \geq j$. For all $r > 0$, if $S \subseteq T$,

$$\|\mathcal{Q}_i^r \mathcal{Q}_j(S)\| \subseteq \|\mathcal{Q}_{i+\phi(c)}^r \mathcal{Q}_{j+c}(T)\|$$

where $c = \mathit{codim}(S, T)$.

Proof: By Axiom 2,

$$\|\mathcal{Q}_k^r \mathcal{Q}_\ell(S_0)\| \cap \|\mathcal{Q}_k^r \mathcal{Q}_\ell(S_1)\| = \|\mathcal{Q}_{k-\phi(c)}^r \mathcal{Q}_{\ell-c}(S_0 \cap S_1)\|$$

where $c = \mathit{codim}(S_0, S_1)$, implying that

$$\|\mathcal{Q}_{k-\phi(c)}^r \mathcal{Q}_{\ell-c}(S_0 \cap S_1)\| \subseteq \|\mathcal{Q}_k^r \mathcal{Q}_\ell(S_0)\|$$

The claim follows by setting $S = S_0 \cap S_1$, $T = S_0$, $i = k - \phi(c)$, and $j = \ell - c$.

Lemma 7: If $(\mathcal{S}, \preceq)$ is an absorbing poset, and S, T, and T_S are defined as in Definition 4, then

$$\mathit{codim}(S \cap T, T_S \cap T) < \mathit{codim}(S, T).$$

Proof: Because $S \cap T \subseteq T_S \cap T$, $\mathit{codim}(S \cap T, T_S \cap T)$ is just the number of vertexes in $T_S \cap T$ but not in $S \cap T$.

There are two cases to consider. First, suppose there is a vertex in T but not in T_S. It follows that

$$\mathit{codim}(S \cap T, T_S \cap T) < \mathit{codim}(S \cap T, T) \leq \mathit{codim}(S, T).$$

Second, suppose instead that $T \subset T_S$. Because T and T_S are distinct, there is vertex in T_S but not in T. It follows that

$$\mathit{codim}(S \cap T, T) < \mathit{codim}(S \cap T, T_S).$$

By Equation 3,

$$\mathit{codim}(S \cap T, T_S) \leq \mathit{codim}(S, T).$$

Combining these inequalities yields the bound.

The next lemma states that every state reachable with a certain number of failures is also reachable with more failures.

Lemma 8:

$$\mathcal{Q}_j^r\mathcal{Q}_k(S) \subseteq \mathcal{Q}_\ell^r\mathcal{Q}_m(S).$$

when $j \leq \ell$ and $k \leq m$.

Proof: We argue by induction on $r \geq 0$. When $r = 0$, the claim follows from Axiom 1.

Suppose $r > 0$. Since $\mathcal{Q}_j^{r-1}\mathcal{Q}_k(S) \subseteq \mathcal{Q}_\ell^{r-1}\mathcal{Q}_m(S)$ by the induction hypothesis, we have

$$\mathcal{Q}_j^r\mathcal{Q}_k(S) = \mathcal{Q}_j\mathcal{Q}_j^{r-1}\mathcal{Q}_k(S) \subseteq \mathcal{Q}_j\mathcal{Q}_\ell^{r-1}\mathcal{Q}_m(S) \subseteq \mathcal{Q}_\ell\mathcal{Q}_\ell^{r-1}\mathcal{Q}_m(S) = \mathcal{Q}_\ell^r\mathcal{Q}_m(S).$$

□

Lemma 9: Let $(\mathcal{S}, \preceq)$ be an absorbing poset, and let $T \in \mathcal{S}$ be a maximal simplex with respect to $\preceq$. We claim that the following sets are both absorbing posets: $(\mathcal{L}, \preceq)$, where $\mathcal{L} = \{L | L \in \mathcal{S} - \{T\}\}$, and $(\mathcal{M}, \preceq)$, where $\mathcal{M} = \{T\}$.

Lemma 10: Let $(\mathcal{A}, \preceq)$ be an absorbing poset containing more than one simplex, and let $A \in \mathcal{A}$ be a maximal simplex with regards to $\preceq$. For each $B \neq A$ in $\mathcal{A}$, there exists a $A_B \in \mathcal{A}$ satisfying the three conditions of Definition 4. We claim that the set

$$\mathcal{B} = \{A_B \cap A | B \in \mathcal{A} - \{A\}\}$$

is an absorbing poset for any total order $<$ on the elements of $\mathcal{A} - \{A\}$

Lemma 11: If every simplex in $\mathcal{Q}_k^r\mathcal{Q}_\ell(\mathcal{A})$ has dimension at least d, then so does every simplex in $\mathcal{A}$.

Lemma 12: Let $\mathcal{Q}_k^r\mathcal{Q}_\ell$ be a composition of simplicial operators where $k \geq \ell$. If $(\mathcal{S}, \preceq)$ is an absorbing poset then for every two simplexes S and T in $\mathcal{S}$ with $T \not\preceq S$ there is a T_S in $\mathcal{S}$ with $T_S \preceq T$, such that

$$\|\mathcal{Q}_k^r\mathcal{Q}_\ell(S)\| \cap \|\mathcal{Q}_k^r\mathcal{Q}_\ell(T)\| \subseteq \|\mathcal{Q}_k^r\mathcal{Q}_\ell(T_S)\| \cap \|\mathcal{Q}_k^r\mathcal{Q}_\ell(T)\|$$
$$\mathit{codim}(T_S, T) = 1$$
$$\mathit{codim}(S \cap T, T_S) \leq \mathit{codim}(S, T).$$

Lemma 13: If $(\mathcal{A}, \preceq)$ is an absorbing poset where ℓ is the minimum dimension of any simplex in $\mathcal{A}$, then $\|\mathcal{A}\|$ is $(\ell - 1)$-connected.

Theorem 14: Let $\mathcal{Q}_k^r\mathcal{Q}_\ell$ be a composition of simplicial operators where $k \geq \ell$, and $(\mathcal{A}, \preceq)$ an absorbing poset. If every simplex in $\mathcal{Q}_k^r\mathcal{Q}_\ell(\mathcal{A})$ has dimension at least ℓ, then $\|\mathcal{Q}_k^r\mathcal{Q}_\ell(\mathcal{A})\|$ is $(\ell - 1)$-connected.

5 The Synchronous Model

We assume a standard synchronous message-passing model with crash failures [2, 14]. The system has $n+1$ processors, and at most f of them can crash in any given execution. Each processor begins in an initial state consisting of its input value, and computation proceeds in a sequence of rounds. In each round, each processor sends messages to other processors, receives messages sent to it by the other processors in that round, performs some internal computation, and changes state. We assume that processors are following a full-information protocol, which means that each processor sends its entire local state to every processor in every round. This is a standard assumption to make when proving lower bounds. A processor can fail by crashing in the middle of a round, in which case it sends its state only to a subset of the processors in that round. Once a processor crashes, it never sends another message after that

A simplex X is *between* two simplexes T and R if $T \subseteq X \subseteq R$. We use $[T : R]$ to denote the set of simplexes between T and R.

Definition 15: Given simplexes S, T, and R, the *pseudosphere* $\mathcal{P}(S, [T : R])$ is the set of all possible labelings of S with simplexes between T and R.

We call this set a pseudosphere because the induced complex has some of the topological properties of a sphere. The simplex S is called the *base simplex* of the pseudosphere, and given a simplex X of a pseudosphere $\mathcal{P}(S, [T : R])$, we define *base*(X) to be S.

Given a simplex S and a set D of processors, let $F = S/D$ be the face of S obtained from S by deleting the vertexes labeled with processors in D. The set of states reachable from S by one round of synchronous computation in which the processors in D fail can be represented by the pseudosphere $\mathcal{P}(F, [F : S])$, the set of all possible labelings of F with simplexes between F and S.

Next, we define the failure operator. Given a simplex S and an integer $\ell \geq 0$, the *ℓ-failure operator* $\mathcal{F}_\ell(S)$ maps S to the set of all faces F of S with *codim*$(F, S) \leq \ell$, which is the set of all faces obtained by deleting at most ℓ vertexes from S. This models the sets of at most ℓ processors that can fail in one round of computation from S.

Definition 16: For every integer $\ell \geq 0$, the *synchronous round operator* $\mathcal{S}_\ell(S)$ is defined by

$$\mathcal{S}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(F, [F : S]).$$

We now check that the synchronous round operator satisfies our axioms.

Lemma 17: $\mathcal{S}_\ell$ satisfies Axiom 1:

$$\mathcal{S}_\ell(S) \subseteq \mathcal{S}_m(S)$$

when $\ell \leq m$.

Proof: Since $\ell \leq m$ implies $\mathcal{F}_\ell(S) \subseteq \mathcal{F}_m(S)$, it follows that

$$\mathcal{S}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(F, [F : S]) \subseteq \bigcup_{F \in \mathcal{F}_m(S)} \mathcal{P}(F, [F : S]) = \mathcal{S}_m(S).$$

In this model, the integer-valued linear function ϕ is simply $\phi(f) = 0$.

Lemma 18: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{S}_k^r \mathcal{S}_\ell(S_0)\| \cap \|\mathcal{S}_k^r \mathcal{S}_\ell(S_1)\| \subseteq \|\mathcal{S}_k^r \mathcal{S}_{\ell-c}(S_0 \cap S_1)\|.$$

Lemma 19: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{S}_k^r \mathcal{S}_{\ell-c}(S_0 \cap S_1)\| \subseteq \|\mathcal{S}_k^r \mathcal{S}_\ell(S_0)\| \cap \|\mathcal{S}_k^r \mathcal{S}_\ell(S_1)\|$$

Corollary 20: $\mathcal{S}_\ell$ satisfies Axiom 2: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{S}_k^r \mathcal{S}_\ell(S_0)\| \cap \|\mathcal{S}_k^r \mathcal{S}_\ell(S_1)\| = \|\mathcal{S}_k^r \mathcal{S}_{\ell-c}(S_0 \cap S_1)\|.$$

To show that $\mathcal{S}_\ell$ satisfies Axiom 3, we impose a partial order on simplexes of $\mathcal{S}_\ell(S)$. Recall that

$$\mathcal{S}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(F, [F : S]).$$

This expression suggests a lexicographic order. We will combine a total order on simplexes F in $\mathcal{F}_\ell(S)$ with a partial order on simplexes of each $\mathcal{P}(F, [F : S])$.

We assume a total order $\leq_{\text{id}}$ on processor ids, which induces a total order on the vertexes of a simplex. We begin by imposing a lexicographic total order on the faces F of S. First we order the faces by decreasing dimension, so that large faces occur before small faces. Then we order faces of the same dimension with a rather arbitrary rule based on our total order on processor ids: we order F before G if the smallest processor id labeling vertexes in F and not G comes before the smallest processor id labeling G and not F. Formally:

Definition 21: Define the total order $<_f$ on the faces of a simplex S by $F <_f G$ if

1. $\dim(F) > \dim(G)$ or
2. $\dim(F) = \dim(G)$ and $p_F <_{\text{id}} p_G$ where

$$p_F = \min\{\mathit{ids}(F) - \mathit{ids}(G)\} \quad \text{and} \quad p_G = \min\{\mathit{ids}(G) - \mathit{ids}(F)\}.$$

Define $F \leq_f G$ if $F <_f G$ or $F = G$.

Next we order the simplexes in a pseudosphere $\mathcal{P}(F, [F : S])$ using the following face ordering: we order A before B if, for each vertex v of the base simplex F, the face of S labeling v in A comes before the face of S labeling v in B. Formally:

Definition 22: Define the partial order $\preceq_p$ on the simplexes of a pseudosphere $\mathcal{P}(F, [F : S])$ by $A \preceq_p B$ if and only if $A_v \leq_f B_v$ for each vertex v in F, where A_v and B_v are the simplexes labeling the vertex v in A and B.

Now we order $\mathcal{S}_\ell(S)$ lexicographically using the face and pseudosphere orders: we order the simplexes in a pseudosphere $\mathcal{P}(F, [F : S])$ before the simplexes in a pseudosphere $\mathcal{P}(G, [G : S])$ if F is ordered before G in the face ordering, and we order the simplexes within a single pseudosphere using the pseudosphere ordering. Formally:

Definition 23: Define the partial order $\preceq_r$ on the simplexes in $\mathcal{S}_\ell(S)$ by $A \preceq_r B$ if and only if

1. *different pseudospheres:* $\mathit{base}(A) <_f \mathit{base}(B)$ or
2. *same pseudosphere:* $\mathit{base}(A) = \mathit{base}(B)$ and $A \preceq_p B$

Theorem 24: $\mathcal{S}_\ell$ satisfies Axiom 3: For every simplex S, $(\mathcal{S}_\ell(S), \preceq_r)$ is an absorbing poset.

Theorem 25: Assume $n + 1 \geq f + k + 1$. No synchronous protocol for k-set agreement halts in fewer than $\lfloor f/k \rfloor + 1$ rounds in the presence of f crash failures.

Proof: Suppose there is a protocol that halts in fewer than $\lfloor f/k \rfloor + 1$ rounds, and assume without loss of generality that it halts in exactly $r = \lfloor f/k \rfloor$ rounds in every execution. Consider the subset of executions in which at most k processors halt in every round. For the input complex $\mathcal{P}(S, V)$, the set of final states of such executions is $\mathcal{S}_k^r(\mathcal{P}(S, V))$. Every simplex in this complex has dimension at least k. By Theorem 14, this complex is $(k-1)$-connected, and by Theorem 1, the protocol cannot solve k-set agreement.

6 Asynchronous Model

Informally, the asynchronous round operator $\mathcal{A}_\ell(S)$ is defined as follows. There are at most ℓ faulty processors in each round, although the set of faulty processors can change from round to round. Faulty processors never crash, but they can omit sending messages. In each round, all nonfaulty processors send their states to all the processors (including faulty ones), while the faulty processors send messages to an arbitrary subset of processors (perhaps none).

Definition 26: For every integer $\ell \geq 0$, the *asynchronous round operator* $\mathcal{A}_\ell(S)$ is defined by

$$\mathcal{A}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(S, [F : S]).$$

At each asynchronous round every processor is labeled with states that include all nonfaulty processors and some subset of faulty processors. Compare with Definition 16. Notice that every simplex in $\mathcal{A}_\ell(S)$ has the same dimension as S.

We now check that the asynchronous round operator satisfies our axioms.

Lemma 27: $\mathcal{A}_\ell$ satisfies Axiom 1:

$$\mathcal{A}_\ell(S) \subseteq \mathcal{A}_m(S)$$

when $\ell \leq m$.

Proof: Since $\ell \leq m$ implies $\mathcal{F}_\ell(S) \subseteq \mathcal{F}_m(S)$, it follows that

$$\mathcal{A}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(S, [F : S]) \subseteq \bigcup_{F \in \mathcal{F}_m(S)} \mathcal{P}(S, [F : S]) = \mathcal{A}_m(S).$$

In this model, the integer-valued linear function ϕ is simply $\phi(f) = f$.

Lemma 28: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{A}_k^r \mathcal{A}_\ell(S_0)\| \cap \|\mathcal{A}_k^r \mathcal{A}_\ell(S_1)\| \subseteq \|\mathcal{A}_{k-c}^r \mathcal{A}_{\ell-c}(S_0 \cap S_1)\|.$$

Lemma 29: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{A}_{k-c}^r \mathcal{A}_{\ell-c}(S_0 \cap S_1)\| \subseteq \|\mathcal{A}_k^r \mathcal{A}_\ell(S_0)\| \cap \|\mathcal{A}_k^r \mathcal{A}_\ell(S_1)\|.$$

Corollary 30: $\mathcal{A}_\ell$ satisfies Axiom 2: Let $k \geq \ell$. For all $r > 0$, if $c = \mathit{codim}(S_0, S_1)$,

$$\|\mathcal{A}_k^r \mathcal{A}_\ell(S_0)\| \cap \|\mathcal{A}_k^r \mathcal{A}_\ell(S_1)\| = \|\mathcal{A}_{k-c}^r \mathcal{A}_{\ell-c}(S_0 \cap S_1)\|.$$

To show that $\mathcal{A}_\ell$ satisfies Axiom 3, we impose a partial order on simplexes of $\mathcal{A}_\ell(S)$. Recall that

$$\mathcal{A}_\ell(S) = \bigcup_{F \in \mathcal{F}_\ell(S)} \mathcal{P}(S, [F : S]).$$

If $F' \subseteq F$, then $\mathcal{P}(S, [F' : S]) \subseteq \mathcal{P}(S, [F : S])$, so we can restrict our attention to faces of codimension ℓ. Unlike in the synchronous model, where simplexes have varying dimensions, all simplexes in this set are labelings of S, and all have dimension n.

We use the same total order $\leq_{\text{id}}$ on processor ids, the same total order $<_f$ on the faces of a simplex S. Next we order the simplexes in $\mathcal{A}_\ell(S)$ using this face ordering: we order A before B if, for each vertex v of the base simplex S the face of F labeling v in A comes before the face of F labeling v in B. Formally:

Definition 31: Define the partial order $\preceq_p$ on the simplexes of $\mathcal{A}_\ell(S)$ by $A \preceq_p B$ if and only if $A_v \leq_f B_v$ for each vertex v in S, where A_v and B_v are the simplexes labeling the vertex v in A and B.

Theorem 32: $\mathcal{A}_\ell$ satisfies Axiom 3: For every simplex S, $(\mathcal{A}_\ell(S), \preceq_p)$ is an absorbing poset.

Theorem 33: No asynchronous protocol for k-set agreement exists in the presence of k crash failures.

References

[1] 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, October 1996.

[2] Hagit Attiya and Jennifer Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. McGraw–Hill, 1998.

[3] Anders Björner and Michelle L. Wachs. Shellable nonpure complexes and posets, i. *Transactions of the American Mathematical Society*, 348(4):1299–1327, 1996.

[4] E. Borowsky and E. Gafni. Generalized flp impossibility result for t-resilient asynchronous computations. In *Proceedings of the 1993 ACM Symposium on Theory of Computing*, May 1993.

[5] 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.

[6] Soma Chaudhuri, Maurice Herlihy, Nancy A. Lynch, and Mark R. Tuttle. Tight bounds for k-set agreement. *Journal of the ACM (JACM)*, 47(5):912–943, 2000.

[7] 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.

[8] Eli Gafni. Round-by-round fault detectors (extended abstract): unifying synchrony and asynchrony. In *Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, pages 143–152. ACM Press, 1998.

[9] Maurice Herlihy and Lucia D. Penso. Tight bounds for k-set agreement with limited scope failure detectors. In Faith Fich, editor, *Distributed Computing, 17th International Conference, DISC 2003, Sorrento, Italy, October 1-3, 2003 Proceedings*, volume 2848 of *Lecture Notes in Computer Science*, pages 279–291. Springer, 2003.

[10] 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.

[11] Maurice Herlihy and Sergio Rajsbaum. Algebraic spans. *Mathematical Structures in Computer Science*, 10(4):549–573, August 2000. Special Issue: Geometry and Concurrency.

[12] Maurice Herlihy, Sergio Rajsbaum, and Mark R. Tuttle. Unifying synchronous and asynchronous message-passing models. In *Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, pages 133–142. ACM Press, 1998.

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

[14] Nancy Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[15] Yoram Moses and Sergio Rajsbaum. A layered analysis of consensus. *SIAM Journal of Computing (SICOMP)*, 31(4):989–1021, 2002.

[16] Achour Mostéfaoui, Sergio Rajsbaum, and Michel Raynal. The synchronous condition-based consensus hierarchy. In Rachid Guerraoui, editor, *Distributed Computing, 18th International Conference, DISC 2004, Amsterdam, Netherlands, October 4-8, 2004 Proceedings*, Lecture Notes in Computer Science. Springer, 2004.

[17] J.R. Munkres. *Elements Of Algebraic Topology*. Addison Wesley, Reading MA, 1984. ISBN 0-201-04586-9.

[18] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980.

[19] 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.

[20] E.H. Spanier. *Algebraic Topology*. Springer-Verlag, New York, 1966.