A Simple Constructive Computability Theorem for Wait-free Computation

Maurice Herlihy
Digital Equipment Corporation
Cambridge Research Laboratory
herlihy@crl.dec.com

Nir Shavit
Computer Science Department
Tel-Aviv University
shanir@math.tau.ac.il

Abstract

In modern shared-memory multiprocessors, processes can be halted or delayed without warning by interrupts, pre-emption, or cache misses. In such environments, it is desirable to design synchronization protocols that are wait-free: any processes that continues to run will finish the protocol in a fixed number of steps, regardless of delays or failures by other processes.

Not all synchronization problems have wait-free solutions. In this paper, we give a new, remarkably simple necessary and sufficient combinatorial condition characterizing the problems that have wait-free solutions using shared read/write memory.

We associate the range of possible input and output values for any synchronization problem with a high-dimensional geometric structure called a simplicial complex. We show that a synchronization problem has a wait-free solution if and only if its input complex can be continuously "stretched and folded" to cover its output complex. The key to the new theorem is a novel "simplex agreement" protocol, allowing processes to converge asynchronously to a common simplex of a simplicial complex. The proof exploits a number of classical results from algebraic and combinatorial topology.

STOC 94- 5/94 Montreal, Quebec, Canada
© 1994 ACM 0-89791-663-8/94/0005..$3.50

1 Introduction

A *decision task* is an input/output problem where N asynchronous processes start with input values, communicate either by shared memory or by message-passing, and halt with output values. Much research in this area has focused on identifying combinatorial conditions characterizing when such tasks are solvable. In the late 80's, Biran, Moran and Zaks [4] provided a pioneering graph theoretic characterization for decision tasks in an asynchronous message-passing system in which only a single processor could fail. This result was not substantially improved until early 1993, when three independent research teams, Borowsky and Gafni [5], Saks and Zaharoglou [13], and the present authors [11], succeeded in applying new combinatorial techniques to models that allow more than one failure. Saks and Zaharoglou used point-set topology and a form of Brouwer's fixed point theorem for the k-dimensional ball to prove that the longstanding open problem of wait-free k-set agreement [7] is unsolvable for $k \leq N-1$. Borowsky and Gafni used a novel simulation method and a variant of Sperner's Lemma to prove k-set agreement impossible with any number of faults $t \geq k$. The present authors presented a general topological characterization of t-faulty computation, and used it to derive impossibility results for both k-set agreement and the well-known renaming problem of Attiya et al. [3]. Our earlier characterization, however, has two limitations. First, the condition is not stated directly in terms of the task's input/output specification. Instead, it characterizes solvability in terms of the topological properties of an associated "full information complex," a geometric realization of a family of concurrent executions. Second, the sufficient condition is existential rather than constructive — one cannot easily derive an algorithm for a particular solvable task.

In this paper, we show that if we restrict our attention to the *wait-free* case, where up to $N-1$ processes can fail (or delay arbitrarily), then there is a remarkably concise necessary and sufficient condition for solvability. This new condition improves our earlier characterization in two ways: first, it provides a clean and intuitive mathematical condition for computability, stated *solely* in terms of a task's input/output specification, and second, the theorem is *constructive*: the combinatorial property characterizing solvability can be used directly to construct an algorithm.

2 Model

A collection of N sequential *processes* communicate by reading and writing variables in shared memory.[1] In modern shared-memory multiprocessors, processes can be halted or delayed without warning by interrupts, pre-emption, or cache misses. In such environments, it is desirable to design synchronization protocols that are wait-free: any processes that continues to run will finish the protocol in a fixed number of steps, regardless of delays or failures by other processes. To capture the formal properties of such environments, we make no fairness assumptions about processes. Up to $N-1$ out of N processes can halt, or display arbitrary variations in speed. In particular, one process cannot tell whether another has halted or is just running very slowly.

Elsewhere [11], we introduced a model in which task specifications are given using standard geometric formalisms from undergraduate-level algebraic topology. (Most of our technical definitions are taken from Spanier [14].) An initial or final state of a process is modeled as a *vertex*, $\vec{v}$, a point in some high-dimensional Euclidian space. Each vertex is labeled with a process id $id(\vec{v})$ and a value $value(\vec{v})$ (either input or output). A set of mutually compatible initial or final states is modeled as a *simplex*, the convex hull of a set of affinely-independent vertexes labeled with distinct process identifiers. Geometrically, a simplex is just the higher-dimensional analogue of a solid triangle or tetrahedron. The complete set of possible initial and final states are represented by sets of simplexes called *simplicial complexes* (or complexes).

Complexes have a dual nature: they are combinatorial objects (sets of sets of vertexes) as well as geometric or topological objects (point sets in Euclidian space). We use $\mathcal{A}^n$ to denote a (combinatorial) complex, and $|\mathcal{A}^n|$ to denote its (geometric) point-set in Euclidian space. Similarly, we use S^n to denote an $(n+1)$-process simplex, and $|S^n|$ its underlying point set. The number n is called the *dimension* of the simplex or complex. A simplex's set of identifiers is denoted by $ids(S^n)$.

A *simplicial map* $\mu : \mathcal{A}^n \to \mathcal{B}^n$ carries vertexes to vertexes such that every simplex in $\mathcal{A}^n$ maps to a simplex in $\mathcal{B}^n$. Any simplicial map defines a piece-wise linear map $|\mu| : |\mathcal{A}^n| \to |\mathcal{B}^n|$. A simplicial map is *color preserving* if $id(\mu(\vec{v})) = id(\vec{v})$. Henceforth, unless explicitly stated otherwise, all simplicial maps are assumed to be color preserving. The *star* of simplex S^m in complex $\mathcal{C}^n$, written $st(S^m)$, is the union of all $|T^n|$ such that $S^m \subset T^n$. The *open star*, written $\overset{\circ}{st}\,(S^m)$, is the interior of the star. If $S^m = (\vec{s}_0, \ldots, \vec{s}_m)$ and $T^\ell = (\vec{t}_0, \ldots, \vec{t}_\ell)$ are simplexes whose vertexes are affinely independent, their *join*, $S^m \cdot T^\ell$, is the $(m+\ell+1)$-simplex $(\vec{s}_0, \ldots, \vec{s}_m, \vec{t}_0, \ldots, \vec{t}_\ell)$.

A *subdivision* of a complex $\mathcal{A}^n$ is a complex $\mathcal{B}^n$ with a map ι carrying vertexes of $\mathcal{B}^n$ to points of $|\mathcal{A}^n|$ such that (1) if S^n is a simplex of $\mathcal{B}^n$ there is some simplex $T^n \in \mathcal{A}^n$ such that $\iota(S^n) \subset |T^n|$, and (2) the piece-wise linear map $|\iota| : |\mathcal{B}^n| \to |\mathcal{A}^n|$ is a homeomorphism. If S^m is a simplex of $\mathcal{B}^n$, $carrier(S^n)$ is the unique smallest T^ℓ such that $S^m \subset |T^\ell|$. A subdivision $\mathcal{B}^n$ is *chromatic* if for all S^m in $\mathcal{B}^n$, $ids(S^m) \subset ids(carrier(S^m))$. A simplicial map $\mu : \mathcal{B}^n \to \mathcal{C}^n$ between subdivisions of $\mathcal{A}^n$ is *carrier preserving* if for all $S^m \in \mathcal{B}^n$, $carrier(S^m) = carrier(\mu(S^m))$.

A task *specification* is given by an input complex $\mathcal{I}^n$, an output complex $\mathcal{O}^n$, and a map Δ carrying each input simplex of $\mathcal{I}^n$ to a set of simplexes of $\mathcal{O}^n$. This map associates with each initial state of the system (an input simplex) the set of legal final states (output simplexes). When $m < n$, $\Delta(S^m)$ indicates the legal final states in executions where only $m+1$ out of $n+1$ processes take steps (the rest fail before taking any steps). A *solution* to a task is a protocol in which the processes communicate by reading and writing a shared memory, and eventually halt with mutually compatible decision values. A *wait-free* solution is one which tolerates the failure of up to n out of $n+1$ processes.

For example, in the *renaming task* [3], $n+1$ processes with unique names from a large name space must choose unique names from a smaller name space. Figure 1 shows all the possible final states of the renaming task where three processors must choose unique names from an output space of four names. As shown, this particular complex is topologically equivalent to a torus. A task specification is shown schematically in Figure 2, in the form of a relation between simplexes of an input complex (shown here as a triangulated 2-sphere), and an output complex

[1] Our results also apply to message-passing systems in which fewer than half the processes can fail [1].

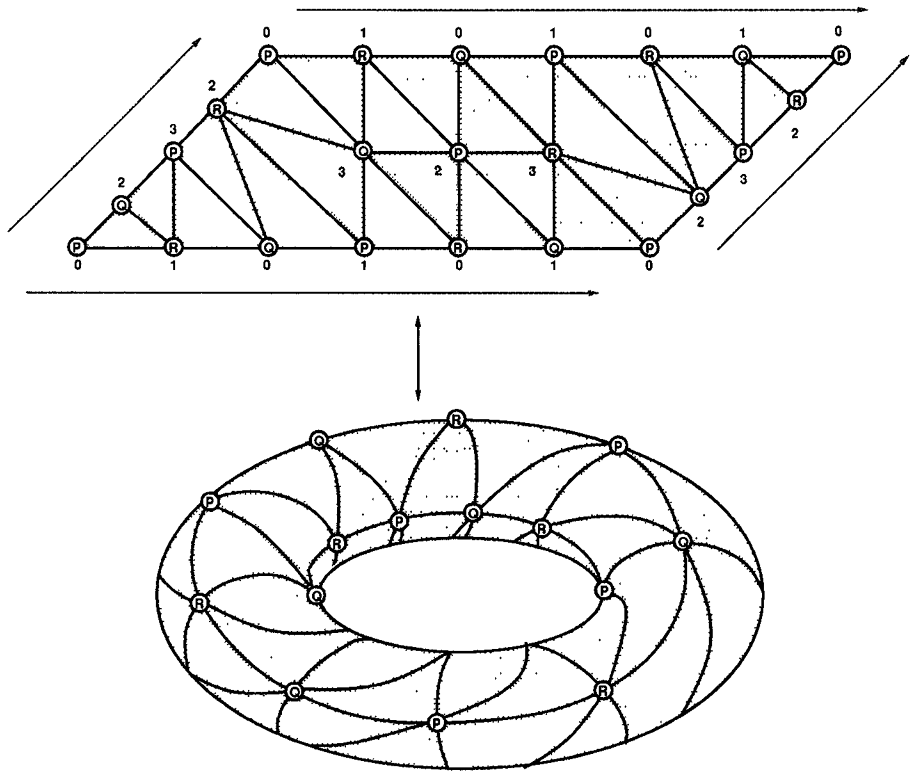

Figure 1: Output Complex for 3-Process Renaming with 4 Names

(shown here as a triangulated torus).

We are now ready to state our main theorem in its entirety.

Theorem 2.1 *A decision task $\langle \mathcal{I}^n, \mathcal{O}^n, \Delta \rangle$ has a wait-free solution using read-write memory if and only if there exists a chromatic subdivision $\sigma(\mathcal{I}^n)$ with a color-preserving simplicial map $\mu : \sigma(\mathcal{I}^n) \to \mathcal{O}^n$ such that for each simplex S^m in $\sigma(\mathcal{I}^n)$, $\mu(S^m) \in \Delta(carrier(S^m))$.*

This theorem is illustrated schematically in Figure 3 Informally, this theorem states that wait-free computation in read/write memory preserves topology. The simplicial map μ induces a continuous (piece-wise linear) map $|\mu|$ on the underlying point sets such that for each input simplex S^m, $|\mu|(|S^m|) \subset |\Delta(S^m)|$. A task is therefore solvable if and only if the input complex can be continuously "stretched" and "folded" so that each input simplex is carried into its corresponding set of output simplexes. This theorem has intriguing parallels to the classical simplicial approximation theorem [14][5.4.8], which states that any continuous map $|\mathcal{A}^n| \to |\mathcal{B}^n|$ can be approximated by a simplicial map from some subdivision of $\mathcal{A}^n$ to $\mathcal{B}^n$.

For example, Figures 4 and 5 show two simple tasks, one solvable and the other not. The first task, 2-process consensus[2], is not solvable, because two simplexes in the connected input complex must be mapped to distinct connected components of the output complex. The second task, called 2-process almost-consensus, is solvable. Although there is no simplicial map directly from the input to the output complex, subdividing the 1-simplex marked $(P0, Q1)$ does admit a map. A corresponding protocol appears in Figure 6.

3 Strategy

The necessity of Theorem 2.1 follows from earlier work. If a protocol exists, the subdivision is induced by the "coherent family of spans" constructed in our earlier paper [11], or by closely related constructions in [5, 13].

In this paper, we focus on sufficiency, constructing an explicit algorithm given the subdivision and the simplicial map. The basic intuition is that solving a decision task is really a form of *approximate agreement* [2, 8, 9, 10, 12], in which processes may start out preferring vertexes "far apart" on the output

[2]The 2-process consensus task requires processes with inputs in the range $\{0, 1\}$ to agree on one of their input values as a common output.

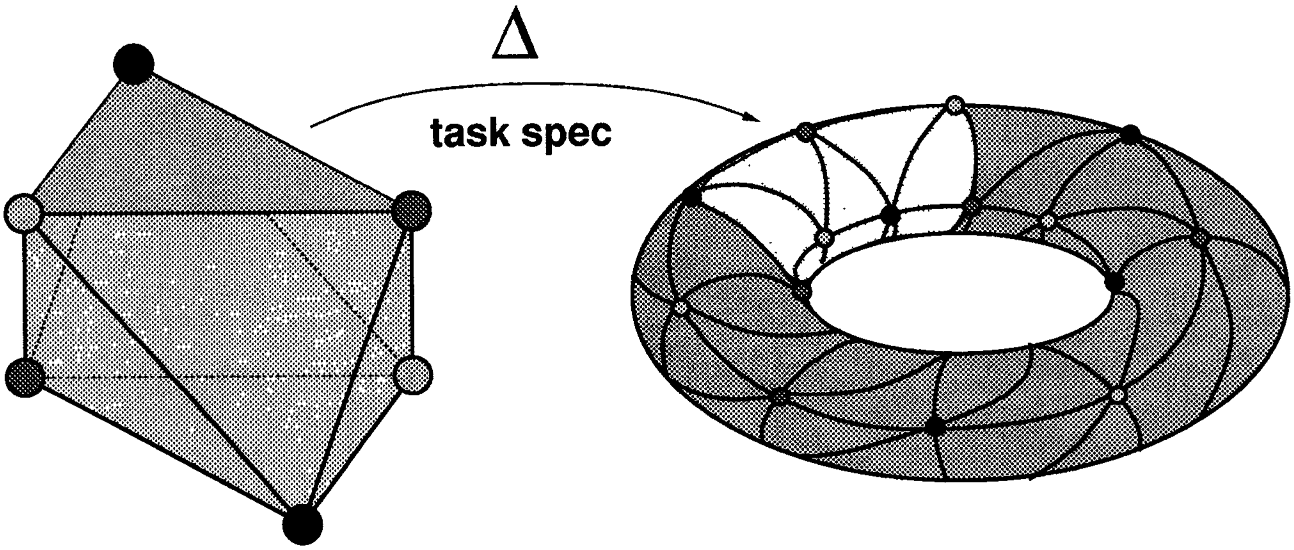

Figure 2: A Task Specification

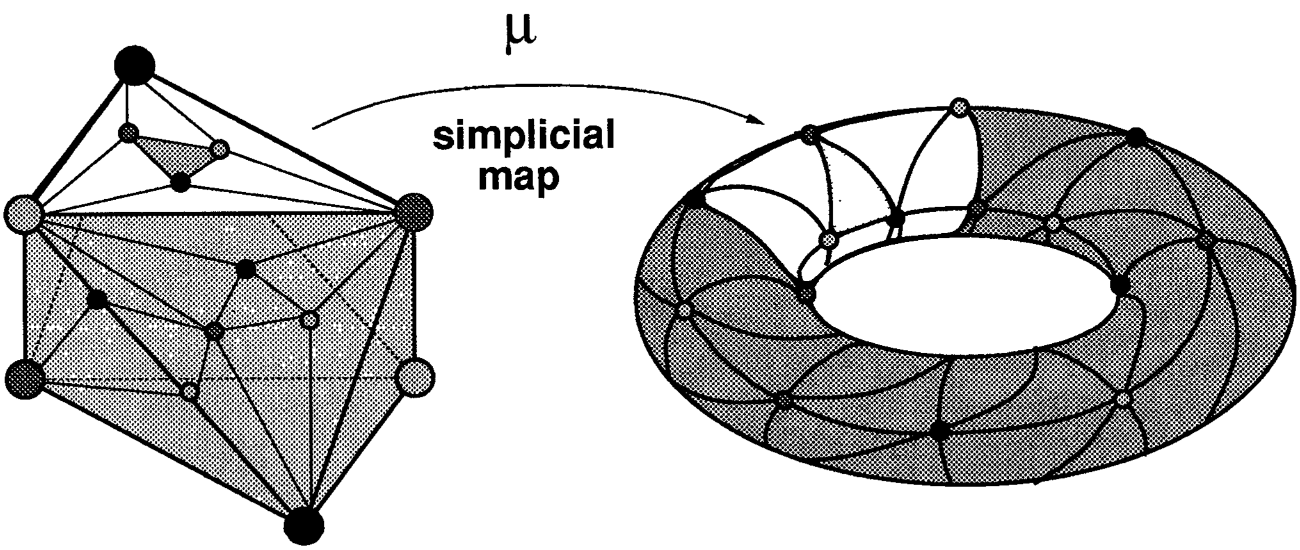

Figure 3: The Main Theorem

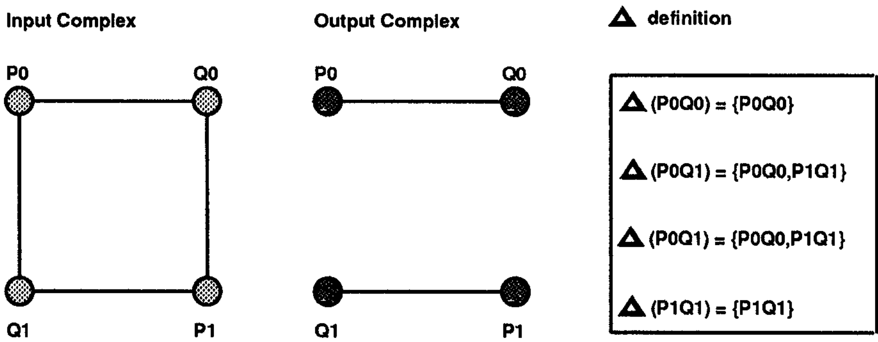

Figure 4: Two Process Consensus

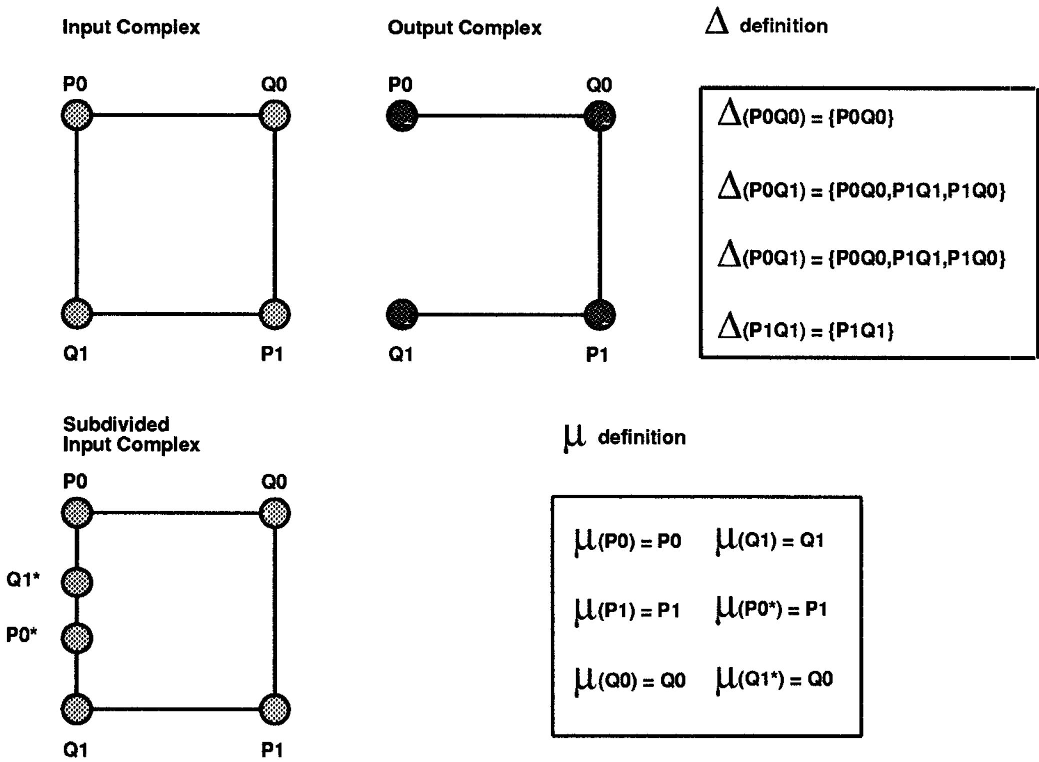

Figure 5: Two Process Almost-Consensus

```
Procedure almost-consensus for P
  initially input[P] = nil
  input[P] := P's input
  if my input is 1 then return 1
  if input[Q] != 1 then return 0
  return 1 /* vertex P0* */
```

```
Procedure almost-consensus for Q
  initially input[P] = nil
  input[Q] := Q's input
  if my input is 0 then return 0
  if input[P] != 0 then return 1
  return 0 /* vertex Q1* */
```

Figure 6: Protocols for Two Process Almost-Consensus

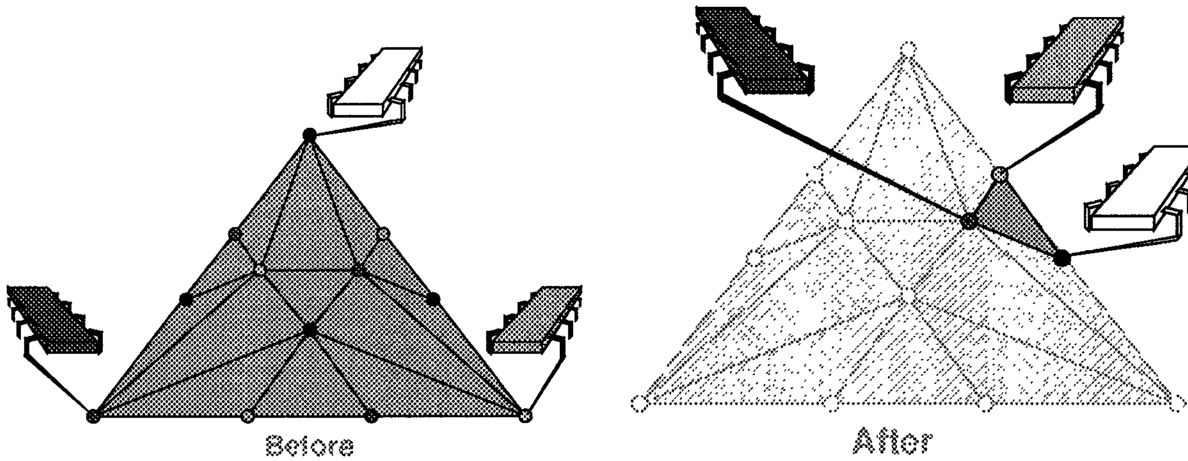

Figure 7: Simplex Agreement

complex, but after a process of negotiation eventually converge to the vertexes of a single output simplex. More formally:

Definition 3.1 The *simplex agreement task* $\langle \mathcal{I}^n, \mathcal{O}^n, \Delta \rangle$ has arbitrary input complex $\mathcal{I}^n$, output complex $\mathcal{O}^n = \sigma(\mathcal{I}^n)$, a chromatic subdivision of $\mathcal{I}^n$, and for all input simplexes S^m, $\Delta(S^m)$ is the set of m-simplexes in $\sigma(S^m)$.

Simplex agreement task is shown schematically in Figure 7. Given the necessity of Theorem 2.1, any algorithm that solves simplex agreement for an arbitrary chromatic subdivision of $\mathcal{I}^n$ is a universal algorithm. Our construction proceeds as follows.

- We introduce the *standard chromatic subdivision* of a complex $\mathcal{I}^n$, denoted $\chi(\mathcal{I}^n)$, and the *iterated* standard chromatic subdivision $\chi^k(\mathcal{I}^n)$. This subdivision is a color-preserving analogue of the classical barycentric subdivision.
- Simplex agreement on $\chi(\mathcal{I}^n)$ is solved by the "participating-set" algorithm of Borowsky and Gafni [6]. Simplex agreement on $\chi^k(\mathcal{I}^n)$ is solved by iterating that algorithm k times.
- If $\sigma(\mathcal{I}^n)$ is an arbitrary chromatic subdivision of $\mathcal{I}^n$, then there exists an integer K such that if $k > K$, there is a carrier-preserving simplicial map $\phi : \chi^k(\mathcal{I}^n) \to \sigma(\mathcal{I}^n)$.

Putting these results together, we have a universal algorithm. If the subdivision σ and simplicial map μ are given, then the value of k and the simplicial map ϕ may be computed off line. The processes first solve simplex agreement on $\chi^k(\mathcal{I}^n)$. A process that chooses vertex $\vec{v}$ then chooses as its output value $\mu(\phi(\vec{v}))$.

4 Standard Chromatic Subdivision

Let the simplex $S^n = (\vec{s}_0, \ldots, \vec{s}_n)$, where $id(\vec{s}_i) = P_i$, and $face_i(S^n)$ the subsimplex of S^n including all vertexes but $\vec{s}_i$.

Definition 4.1 In the *standard chromatic subdivision* of S^n, denoted $\chi(S^n)$, each n-simplex has the form $\{\langle P_0, S_0\rangle, \ldots, \langle P_n, S_n\rangle\}$, where S_i is a subsimplex of S^n, such that (1) $P_i \in ids(S_i)$, (2) for all S_i and S_j, one is a subsimplex of the other, and (3) if $P_j \in ids(S_i)$, then $S_j \subseteq S_i$.

The first and second subdivisions of S^2 are shown in Figure 8. Applying the standard chromatic subdivision repeatedly yields a subdivision $\chi^k(S^n)$. Applying it to every simplex in a complex $\mathcal{C}^n$ yields the complex $\chi^k(\mathcal{C}^n)$.

To show that $\chi(S^n)$ is a subdivision of S^n, we construct an explicit homeomorphism $\iota : |\chi(S^n)| \to |S^n|$. Assume inductively that there exist homeomorphisms $\iota_i : |\chi(face_i(S^n))| \to |face_i(S^n)|$. Let $S^n = (\vec{s}_0, \ldots, \vec{s}_n)$, $\vec{b} = \sum_{i=0}^{n}(\vec{s}_i/(n+1)$ the barycenter of S^n, and δ any value such that $0 < \delta < 1/n$. Define

$$\iota(\langle P_i, S^k\rangle) = \begin{cases} \iota_i(\langle P_i, S^k\rangle) & \text{If } S^k \subseteq face_i(S^n). \\ (1+\delta)\vec{b} - \delta\vec{s}_i & \text{If } S^k = S^n. \end{cases}$$

Because ι is a homeomorphism (proof omitted):

Lemma 4.1 *$\chi(S^n)$ is a subdivision of S^n.*

The *mesh* of a complex is the maximum diameter of any simplex. By analogy with the classical barycentric subdivision:

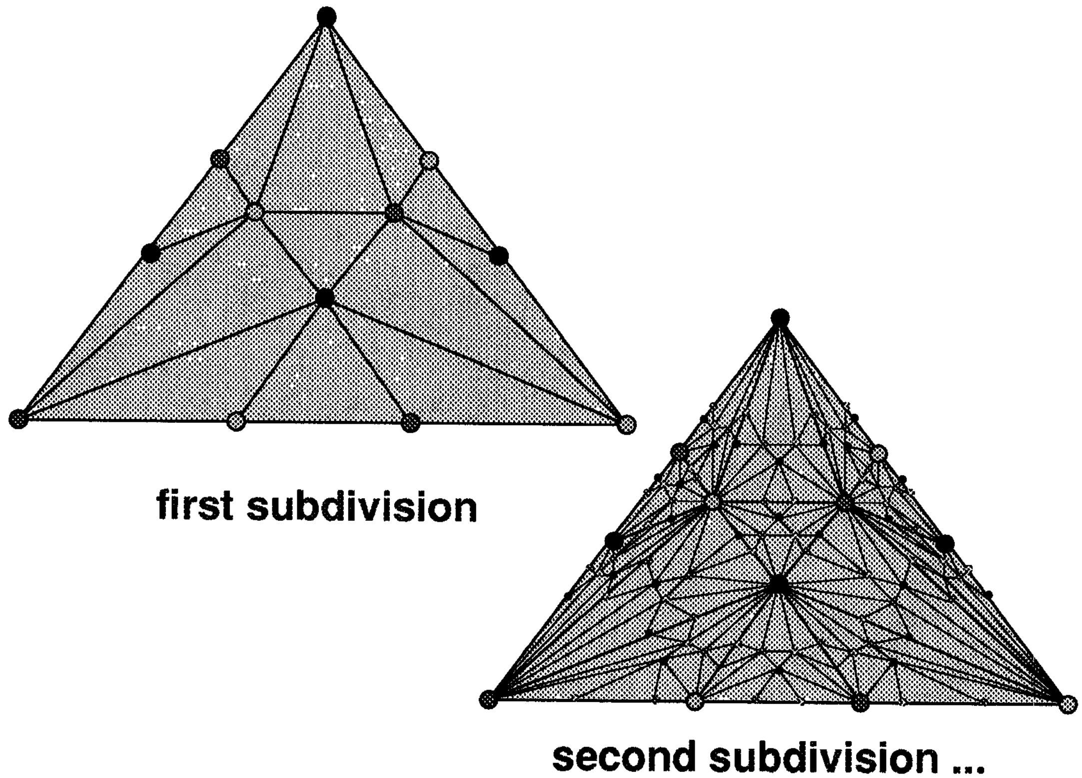

Figure 8: Standard Chromatic Subdivisions

Lemma 4.2 *For sufficiently small* δ, $mesh(\chi(S^n)) \leq \frac{n}{n+1} diam(S^n)$.

Lemma 4.2 implies that by taking sufficiently large k, $mesh(\chi^k(\mathcal{I}^n))$ can be made arbitrarily small.

We refer to the vertexes $\langle P_i, S^n \rangle$ as the *central vertexes* of the subdivision.

5 Simplex Agreement

Lemma 5.1 *There exists a wait-free solution to simplex agreement with input complex* $\mathcal{I}^n$ *and output complex* $\chi(\mathcal{I}^n)$, *the standard chromatic subdivision.*

Proof: Each process P_i must choose a subsimplex of S_i of S^n such that (1) $P_i \in ids(S_i)$, (2) for all S_i and S_j, one is a subset of the other, and (3) if $P_j \in ids(S_i)$, then $S_j \subseteq S_i$. This is exactly the *participating set* problem of Borowsky and Gafni [6], and their simple wait-free solution appears in Figure 9. ■

Lemma 5.2 *There exists a wait-free solution to simplex agreement with input complex* $\mathcal{I}^n$ *and output complex* $\chi^k(\mathcal{I}^n)$, *the iterated standard chromatic subdivision for any* $k > 0$.

Proof: Figure 10 shows an iterated version of the participating set algorithm. ■

6 Arbitrary Chromatic Subdivisions

Our main combinatorial result is to show that if $\sigma(S^n)$ is an arbitrary chromatic subdivision of S^n, then there exists a K such that for all $k \geq K$, there is a color and carrier-preserving simplicial map:

$$\phi : \chi^k(S^n) \rightarrow \sigma(S^n).$$

As a first step, we show that given a subdivision of a simplex, the result of "perturbing" a vertex within its carrier by a sufficiently small distance is still a subdivision.

Definition 6.1 Let $\sigma(S^n)$ be a subdivision of S^n. An *ϵ-perturbation* of $\sigma(S^n)$ is a complex $\sigma'(S^n)$ with a color and carrier-preserving simplicial map $\iota : \sigma(S^n) \rightarrow \sigma'(S^n)$, bijective on vertexes, such that for all $\vec{v}$, $|\vec{v} - \iota(\vec{v})| < \epsilon$.

Theorem 6.1 *If* $\sigma(S^n)$ *is a subdivision of* S^n, *then there exists* $\epsilon > 0$, *such that any* ϵ*-perturbation of* $\sigma(S^n)$ *is also a subdivision of* S^n.

Henceforth, all perturbations are assumed to be subdivisions. Note that $mesh(\sigma'(S^n)) \leq mesh(\sigma(S^n)) + 2\epsilon$.

```
Initially: f[i] = n+2; view_f[j] = null for j in {1..n+1}; S = empty;

procedure participating-set(i: process id; f: shared array);
   repeat
     f[i] := f[i]-1;
     for j := 1 to n+1 do view_f[j] := f[j] od;
     S := {j | view_f[j] <= f[i]};
   until |S| >= f[i];
   return S;
end participating-set;
```

Figure 9: The Participating Set Algorithm.

```
f[1..k][0..n], S[1..k][0..n], input[0..n]: shared array;
Initially for all r in {1..k} f[r][i] = n+2;
          S[r][i] = empty;

procedure simplex-agree(i: process_id;
                        my_vertex: vertex value;
                        k: refinement);

  input[i] := my_vertex;
  for r := 1 to k do
    S[r][i] := participating-set(i,f[r]);
    if r = 1
      then vertex[j,1] := <i,{input[k] | k in S[j,1]}>
      else vertex[j,r] := <i,{vertex[k,r-1] | k in S[j,r]}>
  return(mu(phi(vertex(i,k))));
```

Figure 10: The Iterated Participating Set Algorithm.

Definition 6.2 Two chromatic subdivisions $\rho(S^n)$ and $\sigma(S^n)$ are *independent* if, for every $\vec{r}$ in $\rho(S^n)$, and every $\vec{s}_0, \ldots, \vec{s}_k$ in $\sigma(S^n)$ such that $id(\vec{s}_i) \neq id(\vec{r})$ for $0 \leq i \leq k$, $\vec{r}$ is affinely independent of $\vec{s}_0, \ldots, \vec{s}_k$.

Theorem 6.2 *If $\rho(S^n)$ and $\sigma(S^n)$ are chromatic subdivisions of S^n, then $\rho(S^n)$ has an ϵ-perturbation independent of $\sigma(S^n)$.*

Definition 6.3 If $\iota : \sigma(S^n) \rightarrow \sigma'(S^n)$ is an ϵ-perturbation, and $\phi : \mathcal{A} \rightarrow \sigma(S^n)$ a simplicial map, the compostion $\phi' = \phi \circ \iota$ is called an *ϵ-perturbation* of ϕ.

Lemma 6.3 (Spanier 2.1.25) *A set of vertexes $\vec{v}_0, \ldots, \vec{v}_m$ belong to a common m-simplex if and only if*

$$\bigcap_{i=0}^{m} \overset{\circ}{st}\ (\vec{v}_i) \neq \emptyset.$$

Definition 6.4 Let $\mathcal{B}$ be a colored complex, and $\mathcal{C}$ a subcomplex of $\mathcal{B}$. The *partial chromatic* subdivision $\chi(\mathcal{B}, \mathcal{C})$ is defined as follows: each simplex in $\chi(\mathcal{C}, \mathcal{B})$ has the form $C \cdot B$, where $C \in \chi(\mathcal{C})$ and $carrier(C, \chi(\mathcal{C})) \cdot B \in \mathcal{B}$. The iterated partial chromatic subdivision $\chi^{\ell}(\mathcal{B}, \mathcal{C})$ is defined inductively.

Lemma 6.4 *If $\phi : \mathcal{A} \rightarrow \mathcal{B}$ is a color-preserving simplicial map, then there exists a color-preserving simplicial map $\psi : \chi(\mathcal{A}) \rightarrow \chi(\mathcal{B}, \mathcal{C})$.*

Proof: Let $carrier(\vec{v}, \chi(\mathcal{A})) = X \cdot Y$, where X is the largest face of the carrier such that $\phi(X) \in \mathcal{C}$. If $id(\vec{v}) \in ids(Y)$, define $\psi(\vec{v}) = \phi(\vec{v})$, and otherwise, define $\psi(\vec{v})$ to be the unique central vertex of $\chi(\phi(X))$ with the same id as $\vec{v}$.

We first check that ψ is simplicial on simplexes $S^m = (\vec{s}_0, \ldots, \vec{s}_m)$ where $\phi(carrier(S^m, \chi(\mathcal{A}))) \in \mathcal{C}$. The
simplexes $carrier(\vec{s}_0, \chi(\mathcal{A})), \ldots carrier(\vec{s}_m, \chi(\mathcal{A}))$ are ordered by inclusion (in some order), and so are the simplexes $X_i = \phi(carrier(\vec{s}_i, \chi(\mathcal{A})))$, and any set of central vertexes labeled with distinct colors spans a simplex.

It remains to note that if $\psi(S^m)$ is a simplex, and $S^m \cdot \vec{v} \in \chi(\mathcal{A})$, where $\psi(\vec{v}) \notin \mathcal{C}$, then $\psi(S^m \cdot vv) = \psi(S^m) \cdot \phi(\vec{v})$ is also a simplex. ■

A simple inductive argument yields:

Lemma 6.5 *If $\phi : \mathcal{A} \rightarrow \mathcal{B}$ is a color-preserving simplicial map, then there exists a color-preserving simplicial map $\psi : \chi^{\ell}(\mathcal{A}) \rightarrow \chi^{\ell}(\mathcal{B}, \mathcal{C})$, for all $\ell \geq 0$.*

Theorem 6.6 *If $\sigma(S^n)$ is an arbitrary chromatic subdivision of S^n, then there exists a K such that for all $k \geq K$, there is a carrier-preserving simplicial map:*

$$\phi : \chi^k(S^n) \rightarrow \sigma(S^n).$$

Proof: We first argue inductively by dimension n. When $n = 0$, the property is trivial, so assume inductively that we have such a map for all faces of $\chi^k(S^n)$.

We next given an inductive construction for extending this map into the interior of $\chi^k(S^n)$. We have a three-part induction hypothesis. For each i between 0 and n,

1. There is a subdivision $\tau_i(S^n)$, independent of $\sigma(S^n)$, with a color and carrier-preserving simplicial map $\psi_i : \chi^{\ell_i}(S^n) \rightarrow \tau_i(S^n)$ for some $\ell_i \geq 0$.
2. $\tau_i(S^n)$ contains a subcomplex $\mathcal{X}_i$ with a color and carrier-preserving simplicial map $\phi_i : \mathcal{X}_i \rightarrow \sigma(S^n)$.
3. Every simplex $T \in \tau_i(S^n)$ can be expressed as $X \cdot Y$, $X \in \mathcal{X}_i$, $dim(X) \geq i$, and for every $\vec{x} \in X$ and $\vec{y} \in Y$, $\vec{y} \in \overset{\circ}{st}\ (\phi_i(\vec{x}), \sigma(S^n))$.

In the base case, when $i = 0$, $\mathcal{X}_0 = \emptyset$, $\ell_0 = k$. The first condition is satisfied because Theorem 6.2 ensures that $\chi^k(S^n)$ has an ϵ-perturbation $\tau_0(S^n)$ independent of $\sigma(S^n)$. The remaining conditions are vacuous.

For the induction step, assume the hypothesis for $i - 1$. Let $\mathcal{Y}_{i-1}$ be the largest complex containing only vertexes *not* in $\mathcal{X}_{i-1}$. The open stars of the vertexes in $\sigma(S^n)$ form an open cover for $|S^n|$. Because $\tau_{i-1}(S^n)$ and $\sigma(S^n)$ are independent, every simplex in $Y \in \mathcal{Y}_{i-1}$ has an open cover by sets of the form $\overset{\circ}{st}\ (\vec{s}, \sigma(S^n))$ where $id(\vec{s}) \in ids(Y)$. Because $|Y|$ is compact, this open cover has a Lebesgue number. Let λ_{i-1} be the minimum of the Lebesgue numbers for for all such Y (which exists because $\mathcal{Y}_{i-1}$ is finite). Choose q large enough to ensure that $mesh(\chi^q(\mathcal{Y}_{i-1})) < \lambda_{i-1}/9$. Let $\ell_i = \ell_{i-1} + q$. By Lemma 6.4, we can extend ψ_{i-1} to a simplicial map

$$\Psi : \chi^{\ell_i}(S^n) \rightarrow \chi^q(\tau_{i-1}(S^n), \mathcal{Y}_{i-1})$$

Pick $\epsilon < \lambda_{i-1}/9$. By Theorem 6.2, there exists $\tau_i(S^n)$, an ϵ-perturbation of $\chi^q(\tau_{i-1}(S^n), \mathcal{Y}_{i-1})$ independent of $\sigma(S^n)$, and $\psi_i : \chi^{\ell_i}(S^n) \rightarrow \tau_i(S^n)$, an ϵ-perturbation of Ψ. This perturbation adds at most $2\lambda_{i-1}/9$ to the diameter of any simplex in $\mathcal{Y}_{i-1}$:

$$mesh(\tau_i(\mathcal{Y}_{k-1})) \leq mesh(\chi^q(\mathcal{Y}_{k-1})) + \frac{2\lambda_{i-1}}{9} \leq \frac{\lambda_{i-1}}{3}.$$

For every simplex T^n in $\tau_i(S^n)$, $T^n = X \cdot Y$, where $X \in \tau_i(\mathcal{X}_{i-1})$ (a perturbation of $\mathcal{X}_{i-1}$), and $Y \in \tau_i(\mathcal{Y}_{i-1})$.

$$diam(\bigcup_{\vec{y} \in Y} st(\vec{y}, Y)) \leq 3 \cdot mesh(\tau_i(\mathcal{Y}_{i-1})) < \lambda_{i-1}.$$

Because λ_{i-1} is a Lebesgue number, there is some $\vec{s} \in \sigma(S^n)$ such that the star of every vertex in Y lies in $\overset{\circ}{st}\ (\vec{s}, \sigma(S^n))$. In particular, for at least one $\vec{u} \in Y$, $id(\vec{s}) = id(\vec{y})$. Let $\mathcal{X}_i$ be the largest complex containing only these $\vec{u}$ together with vertexes of $\mathcal{X}_{i-1}$. Define $\phi_i : \mathcal{X}_i \rightarrow \sigma(S^n)$ to send $\vec{x} \in \mathcal{X}_{i-1}$ to $\phi_{i-1}(\vec{x})$, and each remaining $\vec{u}$ to its matching $\vec{s}$. This map is color and carrier-preserving by construction. If X^m is a simplex in $\mathcal{X}_i$, $X^m = U \cdot V$, where U has the property given above, and $V \in \mathcal{X}_{i-1}$. By the induction hypothesis,

$$U \subset \bigcap_{\vec{v} \in V} \overset{\circ}{st}\ (\phi_i(\vec{v}), \sigma(S^n)).$$

By construction,

$$U \subset \bigcap_{\vec{u} \in U} \overset{\circ}{st}\ (\phi_i(\vec{u}), \sigma(S^n)),$$

so ϕ_i is simplicial by Lemma 6.3.

The desired map ϕ is the composition of ϕ_n and ψ_n. ■

Acknowledgments

The authors would like to thank Sergio Rajsbaum for his comments.

References

[1] H. Attiya, A. Bar-Noy, and D. Dolev. Sharing memory robustly in message-passing systems. In *Proceedings of the 9th Annual ACM Symposium on Principles of Distributed Computing*, pages 377–408, August 1990.

[2] H. Attiya, N. Lynch, and N. Shavit. Are wait-free algorithms fast? In *Proceedings of the 31st Annual Symposium on the Foundations of Computer Science*, October 1990.

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

[4] O. Biran, S. Moran, and S. Zaks. A combinatorial characterization of the distributed tasks which are solvable in the presence of one faulty processor. In *Proceedings of the 7th Annual ACM Symposium on Principles of Distributed Computing*, pages 263–275, August 1988.

[5] 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.

[6] E. Borowsky and E. Gafni. Immediate atomic snapshots and fast renaming. In *Proceedings of the 12th Annual ACM Symposium on Principles of Distributed Computing*, August 1993.

[7] 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.

[8] D. Dolev, N.A. Lynch, S.S. Pinter, E.W. Stark, and W.E. Weihl. Reaching approximate agreement in the presence of faults. *Journal of the ACM*, 33(3):499–516, July 1986.

[9] A. Fekete. Asymptotically optimal algorithms for approximate agreement. In *Proceedings of the 5th Annual ACM Symposium on Principles of Distributed Computing*, August 1986.

[10] M.P. Herlihy. Impossibility results for asynchronous PRAM. In *Proceedings of the 2nd Annual Symposium on Parallel Algorithms and Architectures*, July 1991.

[11] M.P. Herlihy and N. Shavit. The asynchronous computability theorem for t-resilient tasks. In *Proceedings of the 1993 ACM Symposium on Theory of Computing*, May 1993.

[12] S. Mahaney and F.B. Schneider. Inexact agreement: Accuracy, precision, and graceful degredation. In *Proceedings of the 4th Annual ACM Symposium on Principles of Distributed Computing*, August 1985.

[13] M. Saks and F. Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. In *Proceedings of the 1993 ACM Symposium on Theory of Computing*, May 1993.

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