Under consideration for publication in Math. Struct. in Comp. Science

Algebraic Spans†

Maurice Herlihy[1‡] and Sergio Rajsbaum[2§]

[1] *Computer Science Department, Brown University*
Providence RI 02912
herlihy@cs.brown.edu
[2] *Instituto de Matemáticas, U.N.A.M., Ciudad Universitaria, D.F. 04510, México*
rajsbaum@math.unam.mx

Received 26 January 2000

Topological methods have yielded a variety of lower bounds and impossibility results for distributed computing. In this paper, we introduce a new tool for proving impossibility results, based on a core theorem of algebraic topology, the *acyclic carrier theorem*, which unifies, generalizes, and extends earlier results.

1. Introduction

Combinatorial and topological methods have yielded a variety of lower bounds results for distributed computing, including general characterizations of the computational power of certain models (e.g. (BMZ90; HR94; HS93; HS94)), and the circumstances under which specific problems can be solved (e.g. (BG93a; BG93b; CHLT93; HR94; HS93; SZ93)); a recent survey appears in (HR99). In this paper, we introduce a new tool for proving impossibility results based on a core theorem of algebraic topology. Using the *acyclic carrier theorem* (Mun84, Th. 13.3), we unify, generalize, and extend earlier results. These new proofs are more succinct. Although the mathematical notions underlying this theorem are abstract, they are elementary, being fully covered in the first chapter of Munkres' standard textbook (Mun84).

This paper makes the following contributions.

— Earlier proofs (HR94; HS93) relied on a mixture of combinatorial and continuous arguments. In this paper, we show how to make these proofs completely algebraic, requiring no continuous mathematics. Some important constructs, such as the notion of a *span* (HS93), are restated in a more elegant algebraic form.

— For each task, set agreement and renaming, we prove a single, short theorem specifying an algebraic property that prevents a protocol from solving the task. These theorems

† A preliminary version of this paper appeared in the Proceedings of the 14th ACM Symposium on Principles of Distributed Computing (PODC '95), Aug. 20–23, Ottawa, Canada, 1995, pp. 90–99.
‡ Supported by NSF grant DMS-9505949.
§ Part of this work was done while visiting the Cambridge Research Laboratory of Compaq and the Laboratory for Computer Science of MIT. Partly supported by CONACYT and DGAPA Projects.

are quite general, yielding results in a variety of models. They imply the known results, and also yield the first impossibility results for renaming using set agreement primitives.

A more complete discussion of related work is postponed to Section 4. Here we expand some of the results presented in the tutorial (HR95), where a few more details about the topology techniques used here can be found.

Finally, we believe that these results further illustrate the benefits of formulating concepts and models from distributed computing in the language of algebraic topology, a mature branch of mainstream mathematics.

This paper is organized as follows. In Section 2 we present the distributed computing model. In Section 3, we present the algebraic topology background. In Section 4 we describe Algebraic Spans, the technique used to unify impossibility results. In Section 5 we present the applications to set agreement, while in Section 6, the applications to renaming. In the Appendix there are several examples to help the reader unfamiliar with algebraic topology, in order to make the paper more self-contained.

2. Distributed Computing Preliminaries

We consider a standard distributed system where processes cooperate to solve a shared task (e.g. (AW98; Lyn96)). They communicate with one another either by message passing or by accessing a shared memory. Informally, a *task* is a problem where each process starts with a private input value, communicates with the others by applying operations to shared objects, and halts with a private output value. A *protocol* is a program that solves a task in a concurrent system. A system may be *asynchronous*, placing no constraints on processes' relative speeds, or *synchronous*, requiring processes to run in lock-step. Processes may communicate by applying operations to shared objects, such as read/write memory, or objects with more powerful semantics. They may also communicate by message-passing. A protocol is *t-resilient* if it tolerates failures by t or fewer processes, and it is *wait-free* if it tolerates failures by n out of $n+1$ processes.

Each process begins a protocol in a distinguished *initial state*, and halts in one of a set of distinguished *final states*. The state of the system encompasses the states of the processes and the communication medium (shared memory or message passing). The system also begins a protocol in a distinguished *initial state*, and halts in one of a set of distinguished *final states*. A set of initial or final process states $s_0, \ldots, s_n$ is *mutually compatible* if there is an initial or final system state in which the i-th process has state s_i.

We model tasks and distributed systems using notions from combinatorial and algebraic topology (HS93). An initial or final state of a process is modeled as a *vertex*, a pair consisting of a process id (a name by which we identify the process) and a value (either input or output). We speak of the vertex as *colored* with the process id (we sometimes work with vertexes colored with other values). A set of $d+1$ mutually compatible initial or final states is modeled as a *d-dimensional simplex*, (or d-simplex). If the colors (not necessarily ids) of a simplex are distinct we say that it is *properly colored*. For the simple

three-process tasks used in examples and figures, we use P, Q, and R as process ids. In formal arguments, we use $P_0, \ldots, P_n$.

Let $S = (s_0, \ldots, s_p)$ and $T = (t_0, \ldots, t_q)$ be simplexes whose vertex sets are disjoint. The *join* of S and T, denoted $S{\cdot}T$, is the simplex $(s_0, \ldots, s_p, t_0, \ldots, t_q)$. If S has dimension p and T has dimension q, then $S \cdot T$ has dimension $p + q + 1$.

Any simplicial complex has a *geometric realization* as a point set in Euclidean space. A vertex corresponds to a point in Euclidean space. A simplex corresponds to the convex hull of affinely-independent vertexes. A complex corresponds to the union of its (geometric) simplexes, provided that any two (geometric) simplexes intersect either in a common face, or not at all. Any complex has a geometric realization in some Euclidean space of sufficiently high dimension. We refer to the point set occupied by a geometric realization of a complex as the *polyhedron* of that complex. Note that two distinct complexes may have the same polyhedron.

The complete set of possible initial (or final) states is represented by a set of simplexes, closed under containment, called a *simplicial complex* (or complex). The *dimension* of $\mathcal{C}$ is the dimension of a simplex of largest dimension in $\mathcal{C}$. We sometimes use superscripts to indicate dimensions of simplexes and complexes. The set of process ids associated with simplex S^n is denoted by $\mathit{ids}(S^n)$, and the set of values by $\mathit{vals}(S^n)$.

A *task specification* for $n + 1$ processes is given by an input complex $\mathcal{I}$, an output complex $\mathcal{O}$, and a relation Δ carrying each input n-simplex of $\mathcal{I}$ to a set of n-simplexes of $\mathcal{O}$. This map associates with each initial state of the system (an input n-simplex) the set of legal final states (output n-simplexes). For some models of computation, it is convenient to extend Δ to simplexes of lower dimension:

$$\Delta(S^m) = \bigcap \Delta(S^n)$$

where S^n ranges over all n-simplexes containing S^m. This definition has the following operational interpretation: $\Delta(S^m)$ is the set of legal final states in executions where only $m + 1$ out of $n + 1$ processes participate (the rest fail without taking any steps). A protocol *solves* a task if when the processes run their programs, they start with mutually compatible input values, represented by a simplex S^n, communicate with one another, and eventually halt with some set of mutually compatible output values, representing a simplex in $\Delta(S^n)$.

For example, in the *consensus* problem (FLP85), each process starts with a private input value, and all processes halt with some process's input value. Figure 1 shows the input and output complexes for two-process binary consensus. The input complex for consensus is constructed by assigning independent binary values to $n + 1$ processes (this complex can be shown to be homeomorphic to an n-sphere), and the output complex consists of two disjoint n-simplexes, corresponding to decision values 0 and 1.

Any protocol that solves a task has an associated *protocol complex* $\mathcal{P}$, in which each vertex is labeled with a process id and that process's final state (called its *view*). Each simplex thus corresponds to an equivalence class of executions that "look the same" to the processes at its vertexes. For $0 \leq m \leq n$, we understand $\mathcal{P}(S^m)$ for a given S^m in the input complex to be the complex generated by all executions starting in S^m, in which only the processes in $\mathit{ids}(S^m)$ take part (the rest fail without taking any steps).

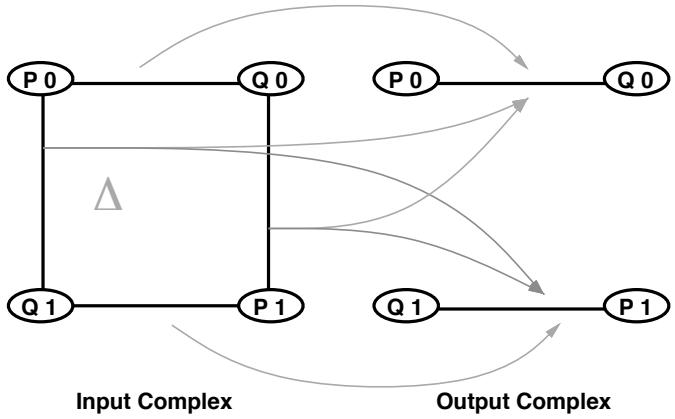

Fig. 1. Input and Output Complexes for 2-Process Consensus

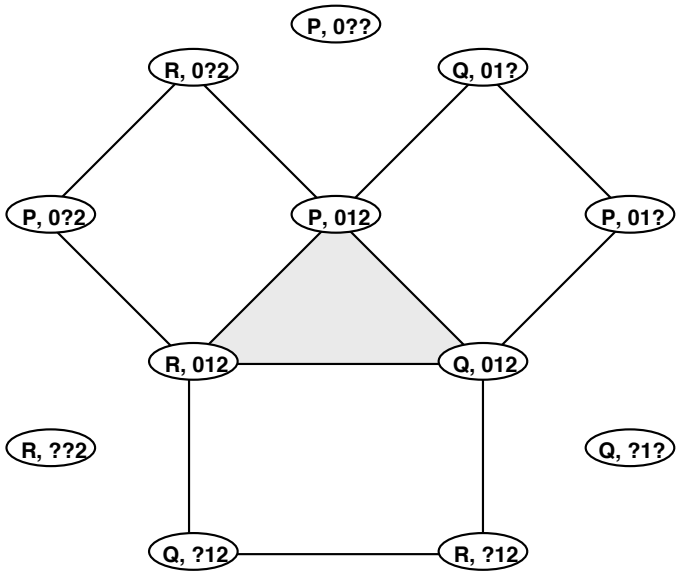

Fig. 2. Protocol Complex for One-Round Synchronous Protocol

The range of m for which $\mathcal{P}(S^m)$ is defined depends on the number of failures allowed by the model. If a simplex R is in $\mathcal{P}(S^m)$, we say that R is *reachable from* S^m.

For example, consider a model in which synchronous processes communicate by broadcasting messages, but a process can fail in the middle of a broadcast. Figure 2 shows the protocol complex for a three-process single-round protocol in which each process broadcasts its index to the others, and then halts. Each vertex in this figure is a possible final state of a non-faulty process (faulty processes are not shown), and simplexes indicate mutually compatible final states. The labels indicate the messages received: for example, "01?" indicates that messages were received from P and Q, but not R. The solid central triangle corresponds to the execution in which no process fails: each vertex is labeled

$\vec{s}$	a vertex
$\mathcal{K}$	a simplicial complex
$\|\mathcal{K}\|$	polyhedron of $\mathcal{K}$
S^m	an m-dimensional simplex
$C_i(\mathcal{K})$	i-th chain group of $\mathcal{K}$
∂	boundary homomorphism
α	a chain
$H_i(\mathcal{K})$	i-th homology group of $\mathcal{K}$
0	trivial single-element group

Fig. 3. Some notation used in this paper

with 012. Attached to the central triangle are 1-simplexes corresponding to executions in which one process fails, and disconnected from that triangle are the three 0-simplexes (vertexes) corresponding to executions in which two processes fail.

A *vertex map* carries vertexes of one complex to vertexes of another. A *simplicial map* is a vertex map that preserves simplexes. A simplicial map on properly colored complexes is *color preserving* if it associates vertexes of the same color. Notice that a color-preserving map preserves dimension. Let $\mathcal{P}$ be the protocol complex for a protocol. If S^n is an input simplex, let $\mathcal{P}(S^n) \subset \mathcal{P}$ denote the complex of final states reachable from the initial state S^n. A protocol solves a decision task $\langle \mathcal{I}^n, \mathcal{O}^n, \Delta \rangle$ if and only if there exists a color-preserving (i.e., process id-preserving) simplicial map $\delta : \mathcal{P} \to \mathcal{O}^n$, called a *decision map*, such that for every input simplex S^n, $\delta(\mathcal{P}(S^n)) \subset \Delta(S^n)$. We prove our impossibility results by exploiting the topological properties of the protocol complex and the output complex to show that no such map exists.

3. Algebraic Preliminaries

Our discussion closely follows that of Munkres (Mun84, Chapter 1). Let $\mathcal{K}$ be an n-dimensional simplicial complex, and $S = (\vec{s}_0, \ldots, \vec{s}_q)$ a q-simplex of $\mathcal{K}$. An *orientation* for S is an equivalence class of orderings on $\vec{s}_0, \ldots, \vec{s}_q$, consisting of one particular ordering and all even permutations of it. For example, an orientation of a 1-simplex $(\vec{s}_0, \vec{s}_1)$ is just a direction, either from $\vec{s}_0$ to $\vec{s}_1$, or vice-versa. An orientation of a 2-simplex $(\vec{s}_0, \vec{s}_1, \vec{s}_2)$ can be either "clockwise," as in $(\vec{s}_0, \vec{s}_1, \vec{s}_2)$ in Figure 4, or "counterclockwise," as in $(\vec{s}_0, \vec{s}_2, \vec{s}_1)$. By convention, simplexes are oriented in increasing subscript order unless explicitly stated otherwise.

A *q-chain* of $\mathcal{K}$ is a formal sum of oriented q-simplexes: $\sum_{i=0}^{\ell} \lambda_i \cdot S_i^q$, where λ_i is an integer. When writing chains, we typically omit q-simplexes with zero coefficients, unless they are all zero, when we simply write 0. We write $1 \cdot S^q$ as S^q and $-1 \cdot S^q$ as $-S^q$. For $q \geq 1$, we identify $-S^q$ with S^q having the opposite orientation.

The q-chains of $\mathcal{K}$ form a free Abelian group $C_q(\mathcal{K})$, called the q-th *chain group* of $\mathcal{K}$. Two q-chains are added by adding coefficients corresponding to the same simplexes. The identity element of the group is the q-chain with all coefficients equal to zero. The group $C_q(\mathcal{K})$ is freely generated by the elementary q-chains (one for each oriented q-simplex).

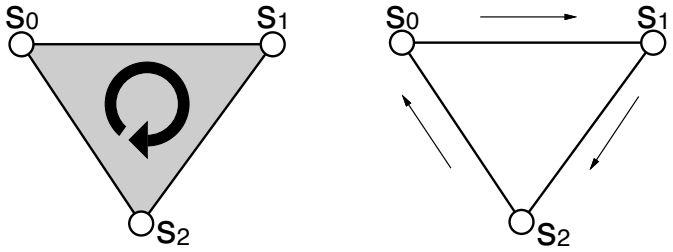

Fig. 4. Oriented Simplex and Boundary

For dimension -1, it is convenient to define $C_{-1}(\mathcal{K})$ to be Z, the infinite cyclic group Z of integers under addition.

A *boundary* $\partial_q : C_q(\mathcal{K}) \to C_{q-1}(\mathcal{K})$ is an homomorphism that satisfies

$$\partial_{q-1}\partial_q\alpha = 0, \tag{1}$$

and an *augmentation* $\partial_0 : C_0(\mathcal{K}) \to C_{-1}(\mathcal{K})$ is an epimorphism (i.e., a surjective homomorphism).

As usual, we use the following boundary homomorphism. Let $S^q = (\vec{s}_0, \ldots, \vec{s}_q)$ be an oriented q-simplex. Define $face_i(S^q)$, the i^{th} *face* of S^q, to be the $(q-1)$-simplex $(\vec{s}_0, \ldots, \hat{s}_i, \ldots, \vec{s}_q)$, where circumflex ($\wedge$) denotes omission. The boundary homomorphism $\partial_q : C_q(\mathcal{K}) \to C_{q-1}(\mathcal{K})$, $q > 0$, is defined on simplexes as follows:

$$\partial_q S^q = \sum_{i=0}^{q} (-1)^i \cdot face_i(S^q),$$

and extends additively to chains: $\partial_q(\alpha_0 + \alpha_1) = \partial_q\alpha_0 + \partial_q\alpha_1$. For $q = 0$, $\partial_0(\vec{s}) = 1$, and extend linearly.† It is not hard to check (or see (Mun84)) that ∂_q satisfies Equation 1. We sometimes omit subscripts from boundary operators. The boundary operator is illustrated in Figure 4.

A q-chain α is a *boundary* if $\alpha = \partial_{q+1}\beta$ for some $(q+1)$-chain β, and it is a *cycle* if $\partial_q\alpha = 0$. Equation 1 implies that every boundary is a cycle. A boundary is an element of $Im(\partial_{q+1})$, and a cycle is an element of $ker(\partial_q)$. Thus, equation 1 implies that the group $im(\partial_{q+1})$ is contained in the group $ker(\partial_q)$. Their quotient group is called the q^{th} *homology group*: ‡

$$H_q(\mathcal{K}) = ker(\partial_q)/im(\partial_{q+1}).$$

Informally, any q-cycle that is not also a boundary corresponds to a "hole" of dimension q. Conversely, if every q-cycle of $\mathcal{K}$ is a boundary, then $\mathcal{K}$ has no "holes" of dimension q, and $H_q(\mathcal{K})$ is the trivial group with just one element, denoted 0. We will later use one direction of this statement:

† Munkres (Mun84) uses ϵ for ∂_0.
‡ Munkres calls these the *reduced* homology groups (Mun84, p.71).

Remark 3.1. If $H_q(\mathcal{K}) = 0$ then for every q-cycle α there exists a $q+1$-chain β such that $\partial_{q+1}\beta = \alpha$.

More precisely, the elements of $H_q(\mathcal{K})$ are the left cosets of $im(\partial_{q+1})$ in $ker(\partial_q)$: an element of $H_q(\mathcal{K})$ has the form $\alpha + im(\partial_{q+1})$ for a q-cycle α. Thus, two q-cycles α, α' are *homologous* if they are in the same equivalence class in $H_q(\mathcal{K})$. Equivalently, α and α' are homologous if and only if $\alpha - \alpha'$ is a boundary.

If $H_0(\mathcal{K})$ is trivial (equal to the one-element group denoted by 0), then $\mathcal{K}$ is connected (there is a path of 1-simplexes connecting any two vertexes). Informally, if $H_q(\mathcal{K}) = 0$, then $\mathcal{K}$ has no "holes" of dimension q. If $H_q(\mathcal{K}) = 0$, for $q \leq k$, we say that $\mathcal{K}$ is *k-acyclic*. If $H_q(\mathcal{K}) = 0$ for every q, we say that $\mathcal{K}$ is *acyclic*.

The *chain complex* $C(\mathcal{K})$ is the sequence of groups and homomorphisms $\{C_q(\mathcal{K}), \partial_q\}$, $q \geq -1$. If $\mathcal{K}$ is of dimension n, then $C_q(\mathcal{K}) = 0$ for $n < q$.

Let $C(\mathcal{K}) = \{C_q(\mathcal{K}), \partial_q\}$ and $C(\mathcal{L}) = \{C_q(\mathcal{L}), \partial'_q\}$ be chain complexes for simplicial complexes $\mathcal{K}$ and $\mathcal{L}$. An *augmentation-preserving chain map* (or chain map) ϕ is a family of homomorphisms.

$$\phi_q : C_q(\mathcal{K}) \to C_q(\mathcal{L}),$$

such that $\partial'_q \circ \phi_q = \phi_{q-1} \circ \partial_q$, $q > 0$, and $\partial'_0 \circ \phi_0 = \partial_0$. Usually ϕ_{-1} is defined to be the identity. In this case the commutator rule for σ_0 is included in the general rule for σ_q.

The previous identities ensure that the chain map ϕ preserves cycles and boundaries: if α is a cycle or boundary, so is $\phi(\alpha)$. Notice that the identity chain map ι of $C(\mathcal{K})$ is a chain map, and the composition of two chain maps is a chain map.

Any simplicial map f from $\mathcal{K}$ to $\mathcal{L}$ induces a chain map $f_\#$ from $C(\mathcal{K})$ to $C(\mathcal{L})$ as follows.

$$(f_\#)_q(\alpha) = \begin{cases} (f(\vec{s}_0), f(\vec{s}_1), \ldots, f(\vec{s}_q)) & \text{if } \alpha \text{ is a simplex } (\vec{s}_0, \vec{s}_1, \ldots, \vec{s}_q) \\ 0 & \text{if } \alpha \text{ is a simplex } (\vec{s}_0, \vec{s}_1, \ldots, \vec{s}_p), p \neq q \\ \sum_i \lambda_i (f_\#)_q(S_i) & \text{if } \alpha \text{ is a chain } \sum_i \lambda_i S_i \end{cases}$$

This map is a homomorphism. For example, $(f_\#)_q(-S^q) = -(f_\#)_q(S^q)$, since interchanging two vertexes on the left interchanges two vertexes on the right. On the other hand, not every chain map is induced by a simplicial map; see the Appendix for an example.

Since $f_\#$ commutes with ∂ it induces a homomorphism $(f_*)_q$ from $H_q(\mathcal{K})$ to $H_q(\mathcal{L})$. We often omit subscripts and sharp signs from induced chain maps when the meaning is clear from context.

If $\phi, \psi : C(\mathcal{K}) \to C(\mathcal{L})$ are chain maps, then a *chain homotopy* from ϕ to ψ is a family of homomorphisms

$$D_q : C_q(\mathcal{K}) \to C_{q+1}(\mathcal{L}),$$

such that

$$\partial'_{q+1} D_q + D_{q-1} \partial_q = \phi_q - \psi_q.$$

Very roughly, if two chain maps are homotopic, then one can be deformed into the other; see Munkres (Mun84) for intuitive justification for this definition. In particular, the two chain maps induce the same homomorphism in homology.

Lemma 3.2. Let $\phi, \psi : C(\mathcal{K}) \to C(\mathcal{L})$ be chain maps and D a chain homotopy from ϕ to ψ. Then the chain $(\phi_q - \psi_q - D_{q-1}\partial_q)(S^q)$ of $C_q(\mathcal{L})$ is a cycle.

Proof.

$$\begin{aligned}\partial'_q(\phi_q - \psi_q - D_{q-1}\partial_q)(S^q) &= \partial'_q(\phi_q - \psi_q)(S^q) - \partial'_q D_{q-1}\partial_q(S_q) \\ &= \partial'_q(\phi_q - \psi_q)(S^q) - \partial'_q(\phi_q - \psi_q - \partial'_{q+1}D_q)(S^q) \\ &= \partial'_q\partial'_{q+1}S^q = 0\end{aligned}$$

□

For problems such as the *renaming* task defined below, we are interested in solutions that satisfy certain symmetry properties. We say that a simplicial map ρ from $\mathcal{K}$ to itself that permutes the vertexes of $\mathcal{K}$ is a *symmetry simplicial map.* The map ρ *preserves dimension*: it sends each q-simplex to a q-simplex. Moreover, ρ permutes the q-simplexes of $\mathcal{K}$: if $\rho(S_0^q) = \rho(S_1^q)$, then $S_0^q = S_1^q$. The i-fold composition of ρ is denoted ρ^i (also a symmetry simplicial map). The *orbit* of a simplex S^q consists of all q-simplexes S for which $\rho^i(S^q) = S$, for some i. The *q-orbits* partition the q-simplexes in equivalence classes. For example, if ρ is the identity symmetry, every orbit consists of a single simplex.

A *symmetry chain map* on $C(\mathcal{K})$, $\rho_\#$, is the chain map $\rho_\# : C(\mathcal{K}) \to C(\mathcal{K})$ induced by a symmetry simplicial map ρ. The i-fold composition of $\rho_\#$, $\rho_\#^i$, is also a chain map. Since ρ preserves dimension, for every oriented q-simplex S^q, $\rho_\#(S^q)$ is an oriented q-simplex, and since ρ is a permutation on the vertexes, $\rho_\#$ is a permutation on the oriented q-simplexes. Thus the oriented q-simplexes are also partitioned in orbits by $\rho_\#$: two oriented simplexes S_0^q, S_1^q are in the same orbit if and only if $\rho_\#^i(S_0^q) = S_1^q$, for some i.

To avoid cumbersome notation, we usually use ρ to denote both the symmetry simplicial map and the induced simplicial chain map, relying on context to avoid ambiguity.

Let ρ, $\tilde{\rho}$ be symmetry chain maps on $C(\mathcal{K})$ and $C(\mathcal{L})$, respectively. A chain map $\phi : C(\mathcal{K}) \to C(\mathcal{L})$ is *symmetric with respect to* $\rho, \tilde{\rho}$, or simply *symmetric*, when ρ and $\tilde{\rho}$ are understood, if $\tilde{\rho} \circ \phi = \phi \circ \rho$. Similarly, a chain homotopy D is symmetric if $\tilde{\rho} \circ D = D \circ \rho$. Notice that any chain map is symmetric with respect to the identity symmetry chain maps.

Definition 3.1. Let ρ, $\tilde{\rho}$ be symmetry chain maps on $C(\mathcal{K})$ and $C(\mathcal{L})$, respectively. A *symmetric acyclic carrier* from $\mathcal{K}$ to $\mathcal{L}$ is a function Σ that assigns to each simplex S^q of $\mathcal{K}$ a non-empty subcomplex of $\mathcal{L}$ such that

1 $\Sigma(S^q)$ is q-acyclic,
2 if S^p is a face of S^q, then $\Sigma(S^p) \subset \Sigma(S^q)$, and
3 $\Sigma(\rho(S)) = \tilde{\rho}(\Sigma(S))$.

A homomorphism $\phi : C_q(\mathcal{K}) \to C_{q'}(\mathcal{L})$ is *carried* by Σ if each simplex appearing with a non-zero coefficient in $\phi(S^q)$ is in the subcomplex $\Sigma(S^q)$.

The next theorem reduces to (Mun84, Th. 13.3), when $\rho, \tilde{\rho}$ are the identity; the proofs are similar.

Theorem 3.3 (Acyclic Carrier Theorem). Let Σ be a symmetric acyclic carrier from $\mathcal{K}$ to $\mathcal{L}$.

(1) There exists a symmetric chain map from $C(\mathcal{K})$ to $C(\mathcal{L})$ that is carried by Σ.
(2) If ϕ and ψ are two symmetric chain maps from $C(\mathcal{K})$ to $C(\mathcal{L})$ that are carried by Σ, then there exists a symmetric chain homotopy of ϕ to ψ that is also carried by Σ.

Proof. Let ∂ be the boundary of $C(\mathcal{K})$ and ∂' the boundary of $C(\mathcal{L})$, and $\rho, \tilde{\rho}$ the symmetry chain maps corresponding to ϕ, ψ.

(1) We construct the required chain map σ_d by induction on d. As usual, we define it on oriented d-simplexes and extend linearly.

Basis: To define σ_0, first pick a vertex $\vec{s}_0$ for each 0-orbit of $\mathcal{K}$. Let $\sigma_0(\vec{s}_0)$ be a vertex in $\Sigma(\vec{s}_0)$. Now, for each $\vec{s}_i \in \text{orbit}(\vec{s}_0)$, $\vec{s}_i = \rho^i(\vec{s}_0)$, let $\sigma_0(\vec{s}_i) = \tilde{\rho}^i(\sigma_0(\vec{s}_0))$. Notice that $\sigma_0(\vec{s}_i)$ is in $\Sigma(\vec{s}_i)$, because $\Sigma(\vec{s}_i) = \Sigma(\rho^i(\vec{s}_0))$, and by property (3) of Definition 3.1, $\Sigma(\rho^i(\vec{s}_0)) = \tilde{\rho}^i(\Sigma(\vec{s}_0))$. This defines σ_0 on the 0-simplexes, and extending linearly we get the desired homomorphism in dimension $d = 0$, that commutes with the boundary operators.

Assume inductively that σ has been correctly defined for dimensions smaller than d. Pick a representative S_0^d for each d-orbit. Now, by induction hypothesis, $\sigma_{d-1}(\partial S_0^d)$ is a well-defined $(d-1)$-chain, and it is in $\Sigma(S_0^d)$. This is because for each face S^{d-1} of S_0^d, $\sigma_{d-1}(S^{d-1})$ is in $\Sigma(S^{d-1})$, and hence it is in $\Sigma(S_0^d)$, by (2) of Definition 3.1. Also $\sigma_{d-1}(\partial S_0^d)$ is is a cycle, because $\partial'\sigma_{d-1}(\partial S_0^d) = \sigma_{d-2}\partial(\partial S_0^d) = 0$, by the induction hypothesis. Because $\Sigma(S_0^d)$ is acyclic, Remark 3.1 implies that we can choose a d-chain of $\Sigma(S_0^d)$, $\sigma_d(S_0^d)$, such that

$$\partial\sigma_d(S_0^d) = \sigma_{d-1}(\partial S_0^d). \tag{2}$$

This ensures that σ commutes with the boundary operators. For each $S_i^d = \rho^i(S_0^d)$, choose

$$\sigma_d(S_i^d) = \tilde{\rho}^i(\sigma_d(S_0^d)).$$

Hence, σ_d commutes with the boundary operators:

$$\begin{aligned}
\partial\sigma_d(S_i^d) &= \partial'\tilde{\rho}^i(\sigma_d(S_0^d)) \\
&= \tilde{\rho}^i(\partial'\sigma_d(S_0^d)) \\
&= \tilde{\rho}^i(\sigma_{d-1}(\partial S_0^d)) \\
&= \sigma_{d-1}(\rho^i(\partial S_0^d)) \\
&= \sigma_{d-1}(\partial S_i^d).
\end{aligned}$$

The penultimate step follows from the induction hypothesis. The step before that one follows from (2) above. Finally, $\sigma_d(S_i^d)$ is in $\Sigma(S_i^d)$ because $\sigma(S_0^d)$ is in $\Sigma(S_0^d)$ and $\Sigma(\rho(S_0^d)) = \tilde{\rho}^i(\Sigma(S_0^d))$.

(2) We construct the required chain homotopy D_q by induction on q, by defining it on the oriented q-simplexes, and then extend linearly.

Basis: Let $D_{-1} = 0$. If chain maps are required to be the identity on C_{-1}, this definition of D_{-1} would be the basis. In any case, to illustrate the ideas, we work through

the case of D_0. We define D_0 as follows. For each 0-orbit pick a vertex $\vec{s}_0$ of $\mathcal{K}$. Because

$$\begin{aligned}\partial'(\phi - \psi)(\vec{s}_0) &= \partial'\phi(\vec{s}_0) - \partial'\psi(\vec{s}_0) \\ &= \phi(\partial\vec{s}_0) - \psi(\partial\vec{s}_0) \\ &= 0\end{aligned}$$

$(\phi - \psi)(\vec{s}_0)$ is a cycle. Since ϕ, ψ are carried by Σ, both $\phi(\vec{s}_0)$, $\psi(\vec{s}_0)$ are in $\Sigma(\vec{s}_0)$, and hence, $(\phi - \psi)(\vec{s}_0)$ is in $\Sigma(\vec{s}_0)$. Since $\Sigma(\vec{s}_0)$ is 0-acyclic, by Remark 3.1, we can choose a 1-chain $D_0(\vec{s}_0)$ in $\Sigma(\vec{s}_0)$, such that $\partial' D_0(\vec{s}_0) = (\phi - \psi)(\vec{s}_0) = (\phi - \psi)(\vec{s}_0) - D_{-1}\partial(\vec{s}_0)$, and the chain homotopy definition is satisfied.

For every $\vec{s}_i \in \text{orbit}(\vec{s}_0)$, $\vec{s}_i = \rho^i(\vec{s}_0)$, choose $D_0(\vec{s}_i) = \tilde{\rho}^i(D_0(\vec{s}_0))$. Notice that $\partial' D_0(\vec{s}_i) = (\phi - \psi)(\vec{s}_i) - D_{-1}\partial(\vec{s}_i)$, since $D_{-1} = 0$ and

$$\begin{aligned}\partial' D_0(\vec{s}_i) &= \partial'\tilde{\rho}^i(D_0(\vec{s}_0)) \\ &= \tilde{\rho}^i\partial'(D_0(\vec{s}_0)) \\ &= \tilde{\rho}^i(\phi - \psi)(\vec{s}_0) \\ &= (\phi - \psi)\tilde{\rho}^i(\vec{s}_0).\end{aligned}$$

Also, $D_0(\vec{s}_i)$ is in $\Sigma(\vec{s}_i)$ because Σ is symmetric. Hence D_0 is carried by Σ. Finally, notice that $D_0(\rho(\vec{s}_i)) = \tilde{\rho}(D_0(\vec{s}_i))$, and hence D_0 is symmetric.

For the induction step, assume a symmetric D_j carried by Σ is defined in dimensions j less than d. Pick a representative S_0^d for each d-orbit. By the same calculations that lead to Lemma 3.2, $(\phi - \psi - D_{d-1}\partial)(S_0^d)$ is a cycle. Moreover, it is in $\Sigma(S^d)$, because ϕ, ψ are carried by Σ, and because so is D_{d-1} (by induction hypothesis), and by property (2) of the definition of acyclic carrier. Because $\Sigma(S_0^d)$ is acyclic, we can choose a $(d+1)$-chain $D_d(S_0^d)$ in $\Sigma(S_0^d)$ such that

$$\partial' D_d(S_0^d) = (\phi - \psi - D_{d-1}\partial)(S_0^d).$$

For each $S_i^d = \rho^i(S_0^d)$ in the same orbit, choose $D_d(S_i^d) = \tilde{\rho}^i(D_d(S_0^d))$. Thus,

$$\begin{aligned}\partial' D_d(S_i^d) &= \partial'\tilde{\rho}^i(D_d(S_0^d)) \\ &= \tilde{\rho}^i(\partial' D_d(S_0^d)) \\ &= \tilde{\rho}^i(\phi - \psi - D_{d-1}\partial)(S_0^d) \\ &= (\phi - \psi - D_{d-1}\partial)(S_i^d).\end{aligned}$$

The last step follows from the induction hypothesis. Finally, it is easy to verify that $D_d \circ \rho = \tilde{\rho} \circ D_d$.

□

As a consequence, we have the following special case of the Acyclic Carrier Theorem, which says that when Σ is dimension preserving a chain homotopy carried by Σ is trivial and hence two chain maps carried by Σ are equal:

Remark 3.4. If $\phi, \psi : C(\mathcal{K}) \to C(\mathcal{L})$ are both carried by Σ, D is a chain homotopy, Σ is dimension preserving, i.e., for each S^q in $\mathcal{K}$, $q = dim(S^q) = dim(\Sigma(S^q))$, then $C_{q+1}(\Sigma(S^q)) = 0$, $D_i = 0$ for all i, and ϕ and ψ are equal chain maps.

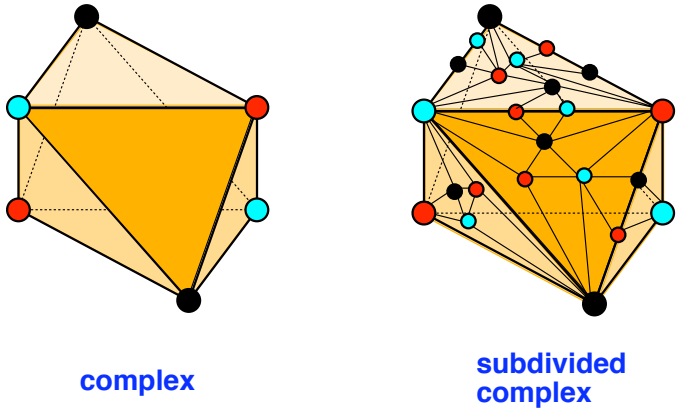

Fig. 5. A Subdivided Complex

4. Algebraic Spans

We can use the acyclic carrier theorem to establish a variety of impossibility results. Our basic strategy is the following. We assume that we have a protocol with complex $\mathcal{P}$ that solves a task $\langle \mathcal{I}, \mathcal{O}, \Delta \rangle$ in a particular model of computation. Let S^ℓ be an input simplex, $\mathcal{S}^\ell$ the complex of its faces, and $\mathcal{P}$ a protocol. For a variety of models of computation, prior research has shown that $\mathcal{P}(S^\ell)$ is $f(\ell)$-acyclic, where f is a model-dependent function. We exploit such results to establish the existence of an acyclic carrier Σ from $\mathcal{S}^\ell$ to $\mathcal{P}$. Then the acyclic carrier theorem guarantees the existence of a chain map $\sigma : C(\mathcal{S}^\ell) \to C(\mathcal{P})$ carried by Σ, which we call an *algebraic span*. The decision map $\delta : \mathcal{P} \to \mathcal{O}$ is a simplicial map, and therefore induces a chain map $\delta : C(\mathcal{P}) \to C(\mathcal{O})$. The composition of δ and σ is also a chain map:

$$\delta \circ \sigma : C(\mathcal{S}^\ell) \to C(\mathcal{O}).$$

We then use properties about Δ to show that $\mathcal{S}^\ell$ and $\mathcal{O}$ are topologically "incompatible," implying that this chain map cannot exist, and thus deriving a contradiction.

We now discuss how a variety of prior lower bound results can all be given a common reformulation in the language of chain complexes and acyclic carriers.

Informally, a *subdivision* of a complex is a way of "chopping up" each of its simplexes into smaller simplexes, as illustrated in Figure 5. Any subdivision of a complex has the "same topology" as the original complex; in particular, the homology groups are unchanged. Much of the earlier work in this area has focused on some notion of subdivision.

Herlihy and Shavit (HS93) considered wait-free protocols in which $n+1$ processes communicate by reading and writing a shared memory. They showed that it is possible to subdivide the input complex so that there exists a simplicial map, called a *span*, from the subdivision to the protocol complex. We will refer to this notion of span as a *geometric* span. They then used the existence of geometric spans to derive a number of impossibility results. These results can be extended to show that a geometric span exists on the input subcomplex $\mathcal{I}_t$ containing the vertexes colored with process ids $P_0, \ldots, P_t$.

Herlihy and Rajsbaum (HR94) considered wait-free protocols using stronger primitives characterized by their ability to solve the (m, j)-agreement task (Cha93), a generalization of consensus (FLP85). They showed that in this model, a geometric span exists only for a subcomplex of the input complex.

Herlihy, Rajsbaum, and Tuttle (HRT97) introduced the notion of a *pseudosphere* (discussed more below), a simple combinatorial structure that can be used to analyze message-passing models, both synchronous and asynchronous. Earlier work on synchronous message passing includes the "Bermuda Triangle" construction of Chaudhuri, Herlihy, Lynch, and Tuttle (CHLT93).

In this paper, we show how these results can be unified by replacing the geometric language of subdivisions and simplicial maps with the more abstract algebraic language of chain complexes and acyclic carriers. To illustrate this remark, we focus first on the wait-free geometric span of Herlihy and Shavit (HS93). Let $\mathcal{P}(S^m)$ denote the subcomplex of $\mathcal{P}$ corresponding to executions where only the processes in $ids(S^m)$ participate, and they start with inputs S^m. Establishing the existence of geometric spans required a combination of combinatorial and continuous arguments:

1 $\mathcal{P}(S^m)$ is acyclic,
2 $\mathcal{P}(S^m)$ is simply connected,[§]
3 inductively use these two facts to construct a family of continuous maps of the input complex, and
4 apply simplicial approximation to transform these continuous maps into the desired simplicial maps on subdivisions of the input complex.

Reformulating this result in algebraic terms yields a simpler derivation: The function Σ_{WF} that assigns to each input simplex S^m, $0 \leq m \leq n$, the protocol subcomplex $\mathcal{P}(S^m)$ is an an acyclic carrier from $\mathcal{I}$ to $\mathcal{P}$. First, as stated above, it is known that every $\mathcal{P}(S^m)$ is acyclic, and second, if S^p is a face of S^q then $\Sigma_{WF}(S^p) \subset \Sigma_{WF}(S^q)$. The Acyclic Carrier Theorem guarantees the existence of an *algebraic span* $\sigma : C(\mathcal{I}) \to C(\mathcal{P})$, which we use for the impossibility results.

The geometric and algebraic notions of span are related as follows. Any geometric span, reinterpreted as a chain map, is an algebraic span. Although algebraic spans are more abstract, they are simpler in several ways. It is easier to establish the existence of an algebraic span: the second, third, and fourth steps of the derivation are unnecessary. The geometric span is not unique — it is easily seen that there are an infinite number of permissible subdivisions and simplicial maps. By contrast, the Acyclic Carrier Theorem implies that algebraic spans are unique up to chain homotopy. On the other hand, algebraic spans are a weaker notion than geometric spans, since they do not depend on the protocol complex being simply connected. Nevertheless, as we show in this paper, the weaker notion is sufficient to prove the set agreement and renaming impossibility results.

Attiya and Rajsbaum (AR96) take an alternative approach to proving lower bounds for wait-free read/write memory using a combinatorial notion called a "divided image."

[§] A space is *simply connected* if its fundamental group is trivial (implying that every loop on the space can be continuously deformed to a point).

5. Set Agreement

In the $(n+1,k)$-agreement task (Cha93)[¶], each of $n+1$ processes starts with a private input value from some set *vals*, $|vals| \geq n+1$, communicates with the others by applying operations to shared objects, and then halts after choosing a private output value. Each process is required to choose some process's input value, and the set of values chosen should have size at most k. This problem independently was shown to have no t-resilient solution in read/write memory by Borowsky and Gafni (BG93a) and by Herlihy and Shavit (HS93), and no wait-free solution, by Saks and Zaharoglou (SZ93). A variety of impossibility results for implementing $(n+1,k)$-agreement from (m,j)-agreement were given by Borowsky and Gafni (BG93b), and by Herlihy and Rajsbaum (HR94).

We assume in this section that we have a protocol with complex $\mathcal{P}$ that solves $(n+1,k)$-agreement in a particular model of computation.

Theorem 5.1. Suppose we have a protocol for $(n+1,k)$-agreement, with protocol complex $\mathcal{P}$, a properly colored simplex S^ℓ, $\ell \leq n$, with colors $vals(S^\ell)$, and an acyclic carrier Σ from $\mathcal{S}^\ell$ to $\mathcal{P}$ such that

$$vals(\delta(\Sigma(S))) \subseteq vals(S) \tag{3}$$

for all simplexes S in $\mathcal{S}^\ell$. We claim that $k \geq \ell+1$.

Proof. Assume by way of contradiction that there exists a protocol with complex $\mathcal{P}$ solving $(n+1,k)$-agreement, and an acyclic carrier Σ from $\mathcal{S}^\ell$ to $\mathcal{P}$, $k \leq \ell \leq n$, satisfying Property (3).

Let $\mathcal{O}_{vals(\mathcal{S}^\ell)}$ denote the subcomplex of $\mathcal{O}$ consisting of simplexes whose decision values are all in $vals(\mathcal{S}^\ell)$. Let $\pi : C(\mathcal{O}_{vals(\mathcal{S}^\ell)}) \to C(\mathcal{S}^\ell)$ be the chain map induced by the simplicial map sending $\langle P_i, v_j \rangle$ to the vertex of $\mathcal{S}^\ell$ with value v_j. This map is well-defined because each simplex in $\mathcal{O}$ is labeled with at most k different values, and $\mathcal{S}^\ell$ contains simplexes with k or fewer distinct values, since $k \leq \ell$.

The Acyclic Carrier Theorem (with ρ just the identity symmetry) guarantees that there exists a chain map $\sigma : C(\mathcal{S}^\ell) \to C(\mathcal{P})$ carried by Σ. Let δ be the chain map corresponding to the decision map of $\mathcal{P}$.

$$C(\mathcal{S}^\ell) \xrightarrow{\sigma} C(\mathcal{P}) \xrightarrow{\delta} C(\mathcal{O}) \xrightarrow{\pi} C(\mathcal{S}^\ell).$$

By (3), the image of the composition $\delta \circ \sigma : C(S^\ell) \to C(\mathcal{O})$ is contained in $C(\mathcal{O}_{vals(\mathcal{S}^\ell)})$. Hence, one may form $\phi : C(\mathcal{S}^\ell) \to C(\mathcal{S}^\ell)$, the composition of σ, δ, and π. Let Φ be the acyclic carrier from $\mathcal{S}^\ell$ to itself that sends each simplex S^i to S^i and all its faces, $\Phi(S^i) = \mathcal{S}^i$. Let ι be the identity chain map on $\mathcal{S}^\ell$. Thus ι is carried by Φ. Property 3 implies that Φ also carries ϕ. Because $dim(S^i) = i = dim(\Phi(S^i))$, Remark 3.4 implies that the two maps are equal. Therefore $\iota(S^\ell) = \phi(S^\ell) = S^\ell$.

In each execution, however, no more than ℓ values are chosen, since $k \leq \ell$, implying that every simplex S in the image of δ has at most ℓ different values. Because $\ell \leq n$, $\mathcal{P}$ contains simplexes of dimension ℓ. Every such simplex S^ℓ in $\mathcal{P}$ is sent by the composition

[¶] Originally called k-set agreement

of the simplicial maps corresponding to δ and π to a simplex in $\mathcal{S}^\ell$ of dimension less than ℓ. The chain map $\pi \circ \delta$ thus sends every ℓ-chain of $C(\mathcal{P})$ to the 0 chain, so $\pi \circ \delta \circ \sigma(S^\ell) = 0$, and $\phi(S^\ell) = 0$, a contradiction. □

Now we show how to apply Theorem 5.1 to various models of computation. We rely on a number of facts about the acyclicity of various protocol complexes. These facts are proved elsewhere, and they rely on techniques beyond the scope of this paper.

Fact 5.2 ((HS93; HS99)). Let $\mathcal{P}$ be an $(n+1)$-process protocol using wait-free asynchronous read/write memory. For any input simplex S^m, where $0 \leq m \leq n$, $\mathcal{P}(S^m)$ is acyclic.

Let S^n be an input simplex in which each process's input value is distinct, and let $\mathcal{S}^n$ be the complex consisting of S^n and its faces. For every S^m in $\mathcal{S}^n$, define $\Sigma_{WF}(S^m) = \mathcal{P}(S^m)$. It is easy to check that $\Sigma_{WF}(S^m)$ satisfies the conditions for an acyclic carrier. By Theorem 5.1, if $\mathcal{P}$ solves $(n+1,k)$-agreement in a wait-free system, then $k \geq n+1$:

Corollary 5.3 ((BG93a; HS93; SZ93)). There is no wait-free $(n+1,n)$-agreement protocol in read/write memory.

Protocol complexes for t-resilient computations are similar.

Fact 5.4. Let $\mathcal{P}$ be an $(n+1)$-process protocol using t-resilient asynchronous read/write memory, for $t < n$. For any input simplex S^m, where $n - t \leq m \leq n$, $\mathcal{P}(S^m)$ is $(t - n + m - 1)$-acyclic.

Let S^n be an input simplex in which $P_0, \ldots, P_t$ have distinct input values. We can express S^n as the join of two simplexes: S^t, colored with $P_0, \ldots, P_t$, and S^{n-t}, colored with the remaining processes. Let $\mathcal{S}^t$ be the complex consisting of S^t and its faces. For every simplex S^ℓ in $\mathcal{S}^t$, define $\Sigma_t(S^\ell)$ to be $\mathcal{P}(S^\ell \cdot S^{n-t})$, where the $(n-t+\ell+1)$-simplex $S^\ell \cdot S^{n-t}$ is the join of S^ℓ and S^{n-t}. As noted, $\Sigma_t(S^\ell)$ is ℓ-acyclic, and satisfies the conditions for an acyclic carrier (with the trivial symmetry operator).

To satisfy Property 3, the following "pre-processing" stage can be added to any $(n+1,t)$-agreement protocol to ensure that every decision value is the input to a participating low-order process. Processes $P_0, \ldots, P_t$ are called the *low-order* processes, and the remaining $P_{t+1}, \ldots, P_n$ are called the *high-order* processes. Before executing the protocol, each low-order process writes its input to a shared array, and each high-order process repeatedly reads that array until a low-order value appears. Because there are $t+1$ low-order processes, and only t failures, every high-order process will eventually observe a low-order value. It then replaces its own input value with that low-order value, and then proceeds to execute the protocol. To show that no t-resilient $(n+1,t)$-agreement protocol exists, it suffices to show that no such protocol exists in which each value is the input to some participating low-order process.

Σ_t is an acyclic carrier from $\mathcal{S}^t$ to $\mathcal{P}(S^n)$, satisfying Property 3, and we conclude from Theorem 5.1 that if $\mathcal{P}$ solves $(n+1,k)$-agreement in a t-resilient system, then $k \geq t+1$ (which can be proved by reduction from the wait-free case, Corollary 5.3, using the BG-Simulation (BG93a; LR96)):

Corollary 5.5. There is no t-resilient $(n+1,t)$-agreement protocol in which processes communicate by a shared read/write memory.

For fixed m and j, $m \geq j > 0$, an (m,j)-*consensus object* provides two operations: *propose*(i) adds the value i to the set of input values, and *choose*() returns a previously-proposed input value. No more than m input values can be proposed, and no more than j distinct values can be returned.

Consider the function

$$J(u) = j \cdot \left\lfloor \frac{u}{m} \right\rfloor + \min(j, u \bmod m) - 1.$$

There is a simple t-resilient $(n+1, J(t+1)+1)$-agreement protocol if processes share a read/write memory and (m,j)-agreement objects. Processes $P_0, \ldots, P_t$ propose their values to $\lfloor \frac{t+1}{m} \rfloor + (t+1) \bmod m$ (m,j)-consensus objects (as few objects as possible), and collectively choose most $J(t+1)+1$ values. Each such process writes its choice to a register. The remaining $n-t$ processes simply wait for one of $P_0, \ldots, P_t$ to write its value. Since only t processes can fail, this wait is bounded. This upper bound can be shown to be tight using the following fact (a refinement of the result of (HR94)).

Fact 5.6. Let $\mathcal{P}$ be an $(n+1)$-process protocol using t-resilient asynchronous read/write memory, for $t < n$, extended with objects that solve (m,j)-agreement. For any input simplex S^m, where $n-t \leq m \leq n$, $\mathcal{P}(S^m)$ is $(J(t-n+m-1)-1)$-acyclic.

For $0 \leq i \leq J(t+1)$, we partition the processes into sets G_i, and choose a representative Q_i from each G_i:

$$G_i = \{P_{i\cdot(m-j+1)}, \ldots, P_{\min(n,(i+1)\cdot(m-j+1)-1)}\}$$
$$Q_i = P_{i\cdot(m-j+1)}.$$

All but the last G_i has $m-j+1$ elements. The Q_i are called *principal processes*. As in the t-resilient read-write case, processes $P_0, \ldots, P_t$ are called the *low-order* processes, and the remaining $P_{t+1}, \ldots, P_n$ are called the *high-order* processes. (In wait-free models, there are no high-order processes.) Without loss of generality, we can precede any $(n+1,k)$-consensus protocol with the same "pre-processing step" used in the t-resilient read-write case, ensuring that every value chosen by the protocol is an input to a participating low-order process.

Let S^n be an input simplex in which every process in G_i has the same value v_i, but $v_i \neq v_j$ for $i \neq j$. We can express S^n as $L \cdot H$, where L is labeled with low-order processes, and H with high-order processes. The low-order simplex L can itself be expressed as $S_0 \cdot \cdots \cdot S_{J(t+1)}$, where $ids(S_i) = G_i$. Finally, let $S^{J(t+1)}$ be the face of S^n labeled with the principal processes.

Let $\mathcal{S}^{J(t+1)}$ be the complex consisting of $S^{J(t+1)}$ and its faces. For every simplex S^ℓ in $\mathcal{S}^{J(t+1)}$, define $\sigma(S^\ell)$ to be the join of S_i, for $i \in ids(S^\ell)$. For each simplex S^ℓ in $\mathcal{S}^{J(t+1)}$, let

$$\Sigma_{(m,j)}(S^\ell) = \mathcal{P}(\sigma(S^\ell) \cdot H).$$

For each S^ℓ, $\Sigma_{(m,j)}(S^\ell)$ is ℓ-acyclic, so $\Sigma_{(m,j)}$ is an acyclic carrier. Property 3 is satisfied

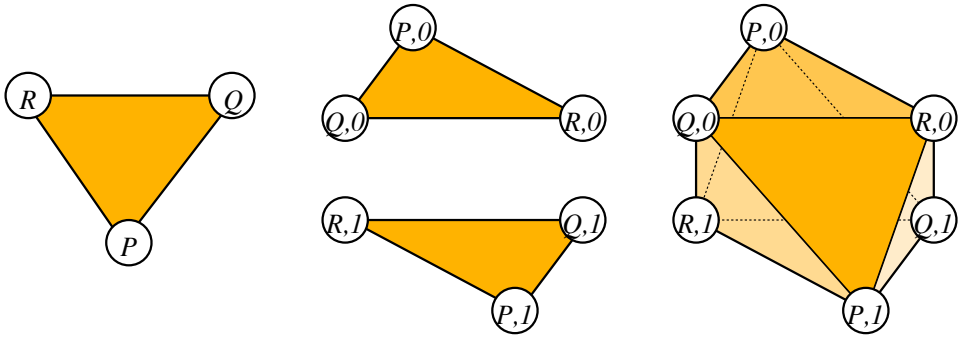

Fig. 6. Construction of a three-process binary pseudosphere.

because each process in G_i has the same input as some principal process in S^ℓ (by hypothesis), and so does each process in H (by preprocessing).

Corollary 5.7. There is no t-resilient $(n+1, J(t+1))$-agreement protocol if processes share a read/write memory and (m, j)-agreement objects.

Herlihy, Rajsbaum, and Tuttle (HRT97) have investigated certain well-structured "round-by-round" executions of the standard asynchronous message-passing models. To show that no protocol exists for the t-resilient asynchronous message-passing model, it suffices to show that no protocol exists in the "round-by-round" subset of that model.

Fact 5.8 ((HRT97)). Let $\mathcal{P}$ be an $(n+1)$-process protocol using t-resilient "round-by-round" asynchronous message-passing read/write memory. For any input simplex S^m, where $n - t \leq m \leq n$, $\mathcal{P}(S^m)$ is $(t - n + m - 1)$-acyclic.

By a construction essentially identical to the one given above for shared memory,

Corollary 5.9. There is no t-resilient $(n+1, t)$-agreement protocol in the message-passing model.

Now consider a model in which $n+1$ processes execute synchronously in rounds. In each round, each process broadcasts a message to the others, but up to k processes per round can fail by halting. A process can fail when its broadcast is partially complete. Without loss of generality, we can restrict our attention to *full-information* protocols in which each process broadcasts its complete state to the others. We use $\mathcal{P}^r(S)$ to denote the complex resulting from an r-round synchronous full-information protocol.

A *pseudosphere* $\psi(S^n; U)$ is a complex defined in terms of a simplex S^n, where each vertex is labeled with a process id, and a finite non-empty set U. The pseudosphere is the complex constructed by taking multiple copies of S^n and independently labeling each vertex with a value from U. For example, Figure 5 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 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 were introduced in a recent paper by Herlihy, Rajsbaum, and Tuttle (HRT97).

Pseudospheres are useful because the collection of initial global states for $(n+1,k)$-agreement forms a pseudosphere whose vertexes are labeled with input values. For example, the right-hand figure in Figure 5 is the input complex for three-process binary consensus.

Fact 5.10 ((HRT97)). For all $n \geq 0$, and all non-empty U, the pseudosphere $\psi(S^n;U)$ is $(n-1)$-connected.

Fact 5.11 ((HRT97)). Let $\mathcal{P}^r$ be the protocol complex for an r-round synchronous full-information protocol in which k or fewer processes can fail in each round, and let $\psi(S^n;U)$ be an input complex. If $n \geq rk+k$, $\mathcal{P}^r(\psi(S^n;U))$ is $(k-1)$-acyclic.

Let S^{k-1} be a simplex whose values are labeled from 0 to $k-1$, and $\mathcal{S}^{k-1}$ the complex consisting of S^{k-1} and its faces. For each simplex S^ℓ in S^{k-1}, the pseudosphere $\psi(S^n;\mathit{ids}(S^\ell))$ is the subcomplex of the input complex in which all input values are taken from $\mathit{ids}(S^\ell)$. Define

$$\Sigma_r(S^\ell) = \mathcal{P}^r(\psi(S^n;\mathit{ids}(S^\ell)))$$

The map Σ_r is an acyclic carrier, and because each input value in $\Sigma_r(S^\ell)$ is in $\mathit{vals}(S^\ell)$, it satisfies Property 3.

Corollary 5.12 ((HRT97)). If $n \geq f+k$, then there is no synchronous f-resilient k-agreement protocol taking $\lfloor f/k \rfloor$ or fewer rounds.

Each of these lower bounds is known to be tight.

6. Renaming

In the renaming task (Attiya *et al.* (ABND$^+$90)), $n+1$ processes with unique names taken from a large name space must choose unique names taken from a small name space. More precisely, in the $(n+1,K)$-renaming task, the processes are given unique *input names* in the range $0,\ldots,N$, and are required to choose unique *output names* in the range $0,\ldots,K$, where $n \leq K < N$.

To rule out the trivial solution where P_i chooses output name i, we are interested in protocols for which a process's choice is independent of its process id. Let π be a permutation of the process ids. The permutation π acts on any labeled simplex by replacing each occurrence of a process id P in the label with the process id $\pi(P)$.

$$\pi(\langle P_i, e_i\rangle) = \langle \pi(P_i), \pi(e_i)\rangle$$

If the label is a view of an execution e, then $\pi(e)$ is the execution in which each occurrence of P is replaced by $\pi(P)$ (the same interleaving, but processes are renamed).

A complex $\mathcal{C}$ is *symmetric* if the vertex map induced by π is simplicial. (To avoid

cumbersome notation, we use π to denote both the permutation and the various maps it induces, relying on context to avoid ambiguity.) If $\mathcal{A}$ and $\mathcal{B}$ are symmetric complexes, then a simplicial map $\phi : \mathcal{A} \to \mathcal{B}$ is *symmetric under permutation* if $\pi(\phi(\vec{v})) = \phi(\pi(\vec{v}))$ for any permutation π. A task specification $\langle \mathcal{I}, \mathcal{O}, \Delta \rangle$ is *symmetric* if $\mathcal{I}$ and $\mathcal{O}$ are symmetric, and for all $S^n \in \mathcal{I}$, $\Delta(\pi(S^n)) = \pi(\Delta(S^n))$. In short, the problem specification depends only on input values, not process ids.

Definition 6.1. A protocol $\mathcal{P}$ is *anonymous* if the decision map δ is symmetric under permutation: for every simplex T in $\mathcal{P}(S^n)$, and for any permutation π, $\pi(\delta(T)) = \delta(\pi(T))$.

This condition can be summarized by the following commutative diagram.

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

The permutation π also acts on chain complexes, where the anonymity condition can be summarized as follows.

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

We restrict our attention to anonymous protocols.

In this section, we use symmetry arguments to give general lower bounds on renaming, according to the following strategy. We show that if an $(n+1, K)$-renaming protocol has an acyclic carrier Σ from $\mathcal{S}^\ell$ to $\mathcal{P}$ with the property that the protocol behaves "symmetrically" on the boundary of $\Sigma(\mathcal{S}^\ell)$, then $K \geq 2\ell$. We use the following particular symmetries.

Let $S^\ell = (\vec{s}_0, \ldots, \vec{s}_\ell)$ be a simplex where each $\vec{s}_i$ is labeled with process id P_i, let $\mathcal{S}^\ell$ be the complex of all its faces, and $\dot{\mathcal{S}}^{\ell-1}$ the complex of its proper faces (i.e. $\mathcal{S}^\ell$ minus S^ℓ). In what follows we use the "rotation" permutation ρ, on the process ids $P_0, \ldots, P_\ell$, sending P_i to $P_{i+1 \bmod \ell+1}$, and leaving the other process ids fixed. This permutation acts on simplexes and protocol complexes

$$\rho : \mathcal{S}^\ell \to \mathcal{S}^\ell \qquad \rho : \mathcal{P} \to \mathcal{P}$$

by $\rho(\vec{s}_i) = \alpha(\vec{s}_i) = \vec{s}_{i+1 \bmod \ell+1}$, and $\rho\langle P_i, e_i \rangle = \langle \rho(P_i), \rho(e_i) \rangle$. Because $\mathcal{P}$ is assumed to be anonymous, both of these maps are simplicial. The induced chain maps on the chain complexes $C(\mathcal{S}^\ell)$ and $C(\mathcal{P})$ are symmetry maps.

The proof is based on the following, purely topological lemma.

Lemma 6.1. If $\phi : C(\mathcal{S}^\ell) \to C(\mathcal{S}^\ell)$ is a symmetric chain map with respect to ρ, then $\phi(\partial S^\ell) = k \cdot \partial S^\ell$, for $k \equiv 1 \pmod{\ell+1}$.

Proof. Let $\iota : C(\mathcal{S}^\ell) \to C(\mathcal{S}^\ell)$ be the identity chain map. Consider any symmetric

acyclic carrier Σ from $\mathcal{S}^\ell$ to itself that carries both ϕ and ι, for example, one that assigns all of $\mathcal{S}^\ell$ to every simplex of $\mathcal{S}^\ell$. The Acyclic Carrier Theorem implies that there is a symmetric chain homotopy D between ϕ and ι.

In particular, by Lemma 3.2, $(\phi - \iota - D\partial)(face_0(S^\ell)) \in C_{\ell-1}(\mathcal{S}^\ell)$ is a cycle of $\mathcal{S}^\ell$. Since the group of $(\ell-1)$-cycles of $\mathcal{S}^\ell$ is infinite cyclic generated by ∂S^ℓ (as discussed in the Appendix),

$$(\phi - \iota - D\partial)(face_0(S^\ell)) = k \cdot \partial S^\ell, \tag{4}$$

for some integer k.

Note that, for even ℓ,

$$\rho^i(face_0(S^\ell)) = (-1)^i face_i(S^\ell), \tag{5}$$

while if ℓ is odd,

$$\rho^i(face_0(S^\ell)) = face_i(S^\ell). \tag{6}$$

To check this remark observe that

$$\rho(face_0(S^\ell)) = \rho(\vec{s}_1, \ldots, \vec{s}_\ell) = (\vec{s}_2, \ldots, \vec{s}_\ell, \vec{s}_0).$$

We can write both cases 5 and 6 in one equation by taking p to be the reverse parity of ℓ ($p = 0$ if ℓ is odd, and 1 if ℓ is even):

$$\rho^i(face_0(S^\ell)) = (-1)^{i \cdot p} face_i(S^\ell). \tag{7}$$

Hence, using Equation 7 and the definition of boundary,

$$\rho(\partial S^\ell) = (-1)^{p+1} \partial S^\ell. \tag{8}$$

because $\rho(\partial S^\ell) = \rho(\Sigma_{i=0}^\ell (-1)^i \cdot face_i(S^\ell))$, and using Equation 7 this is equal to

$$\begin{aligned} \Sigma_{i=0}^\ell (-1)^i \cdot \rho(face_i(S^\ell)) &= \Sigma_{i=0}^\ell (-1)^i \cdot (-1)^p face_{i+1 \bmod \ell+1}(S^\ell) \\ &= (-1)^{p+1} \partial S^\ell. \end{aligned}$$

Thus, Equation 8 yields

$$\rho^i(\partial S^\ell) = (-1)^{i(p+1)} \partial S^\ell. \tag{9}$$

By definition, $\phi(\partial S^\ell) = \phi \Sigma_{i=0}^\ell (-1)^i \cdot face_i(S^\ell)$. Thus, by Equation 7,

$$\begin{aligned} \phi(\partial S^\ell) &= \phi \Sigma_{i=0}^\ell (-1)^{i(p+1)} \rho^i(face_0(S^\ell)) \\ &= \Sigma_{i=0}^\ell \phi (-1)^{i(p+1)} \rho^i(face_0(S^\ell)), \\ &= \Sigma_{i=0}^\ell \rho^i (-1)^{i(p+1)} \phi(face_0(S^\ell)) \end{aligned}$$

by symmetry of ϕ. By Equation 4,

$$= \Sigma_{i=0}^\ell \rho^i (-1)^{i(p+1)} (k \cdot \partial S^\ell + (\iota + D\partial)(face_0(S^\ell))).$$

By Equation 9,

$$= k(\ell+1) \cdot \partial S^\ell + \Sigma_{i=0}^\ell \rho^i (-1)^{i(p+1)} (\iota + D\partial)(face_0(S^\ell)).$$

By Equation 7, symmetry of ι and ∂,

$$= k(\ell+1)\cdot\partial S^\ell + \Sigma_{i=0}^{\ell}(\iota + D\partial)((-1)^i face_i(S^\ell)).$$

Since ι is the identity,

$$= k(\ell+1)\cdot\partial S^\ell + \partial S^\ell + D\partial\partial S^\ell,$$

and the proof follows from $\partial\partial = 0$. □

Informally, this lemma says that any map from S^ℓ to itself that is symmetric on the boundary must "wrap" the boundary around itself a non-zero number of times.

Theorem 6.2. Suppose we have a protocol for $(n+1, K)$-renaming, an acyclic carrier Σ from $\mathcal{S}^\ell$ to $\mathcal{P}$ that is symmetric with respect to ρ, such that

$$ids(\Sigma(S)) = ids(S), \tag{10}$$

for all proper faces S of S^ℓ. Then $K \geq 2\ell$.

Proof. Assume by way of contradiction that $K < 2\ell$. Consider the output complex $\mathcal{O}$ of the $(n+1, K)$-renaming task, and its subcomplex $\mathcal{O}(S^\ell)$ encompassing vertexes with ids $P_0, \ldots, P_\ell$. Thus, a vertex $\langle P_i, v\rangle$ of $\mathcal{O}$ is labeled with a process id P_i and an output name v in $0, \ldots, K$. Let $\pi : \mathcal{O} \to \mathcal{S}^\ell$ be the simplicial map $\pi\langle P_i, v\rangle = \vec{s}_j$, where $j = (i + (v \bmod 2)) \bmod \ell + 1$. Let π also denote the induced chain map.

The simplicial map π does not send any ℓ-simplex of $\mathcal{O}(S^\ell)$ to S^ℓ. This is because π sends an ℓ-simplex of $\mathcal{O}(S^\ell)$ to S^ℓ only if the processes $P_0, \ldots, P_\ell$ have chosen all even or all odd output names, which is impossible because the range $0, \ldots, 2\ell-1$ does not contain $\ell+1$ distinct even or distinct odd names. It follows that, on $\mathcal{O}(S^\ell)$, the homomorphism (of the chain map π) $\pi_\ell = 0$.

We have the following sequence of chain maps.

$$C(\mathcal{S}^\ell) \xrightarrow{\sigma} C(\mathcal{P}) \xrightarrow{\delta} C(\mathcal{O}) \xrightarrow{\pi} C(\mathcal{S}^\ell),$$

where σ is the chain map whose existence is guaranteed by the Acyclic Carrier Theorem. Let $\phi : C(\mathcal{S}^\ell) \to C(\mathcal{S}^\ell)$ be the composition of σ, δ, and π. Notice that all vertexes in the image of σ are labeled with ids from $ids(S^\ell)$, by Equation 10. Thus, the same holds for the image of $\delta\cdot\sigma$, since δ is color preserving; i.e., this image is contained in $\mathcal{O}(S^\ell)$. It follows from $\pi_\ell = 0$ on $\mathcal{O}(S^\ell)$ that $\phi_\ell(S^\ell) = 0$, and hence

$$\phi_{\ell-1}(\partial S^\ell) = 0, \tag{11}$$

since ϕ commutes with ∂.

We claim that the chain map ϕ is symmetric. Recall that the rotation permutation ρ sends each process P_i in $ids(S^\ell)$ to $P_{i+1 \bmod \ell+1}$, and leaves the rest unchanged. It operates on output complexes in the usual way:

$$\rho(\langle P_i, v_i\rangle) = \begin{cases} \langle P_{i+1 \bmod \ell+1}, v_i\rangle & \text{if } P_i \in ids(S^\ell) \\ \langle P_i, v_i\rangle) & \text{otherwise} \end{cases}$$

We have the following commutative diagram of symmetric chain maps:

$$\begin{array}{ccccccc} C(\mathcal{S}^\ell) & \xrightarrow{\sigma} & C(\mathcal{P}) & \xrightarrow{\delta} & C(\mathcal{O}) & \xrightarrow{\pi} & C(\mathcal{S}^\ell) \\ \downarrow{\scriptstyle \rho} & & \downarrow{\scriptstyle \rho} & & \downarrow{\scriptstyle \rho} & & \downarrow{\scriptstyle \rho} \\ C(\mathcal{S}^\ell) & \xrightarrow{\sigma} & C(\mathcal{P}) & \xrightarrow{\delta} & C(\mathcal{O}) & \xrightarrow{\pi} & C(\mathcal{S}^\ell) \end{array}$$

We check that each rectangle commutes: σ is symmetric by the Acyclic Carrier Theorem, δ is symmetric because the protocol is anonymous, and π is symmetric by construction. It follows that ϕ is symmetric:

$$\begin{aligned} \rho\phi &= \rho\pi\delta\sigma \\ &= \pi\rho\delta\sigma \\ &= \pi\delta\rho\sigma \\ &= \pi\delta\sigma\rho = \phi\rho. \end{aligned}$$

Lemma 6.1 implies that $\phi(\partial S^\ell) = k \cdot \partial S^\ell$, for $k \neq 0$, contradicting Equation 11. □

Now we consider some of the acyclic carriers described in Section 5. For asynchronous read/write memory, consider a protocol for $(n+1, K)$-renaming, with protocol complex $\mathcal{P}$, and let $\mathcal{I}$ be the corresponding input complex. Recall from 5.2 that for any input simplex $S^n \in \mathcal{I}$ there is an acyclic carrier Σ_{WF} from $\mathcal{S}^n$ to $\mathcal{P}$. This carrier, for a single input simplex, does not satisfy the symmetry requirements of Theorem 6.2. Pick, for example, S^n labeled with process ids $P_0, \ldots, P_n$ and the vertex of P_i labeled with input name i. It does satisfy the requirement

$$ids(\Sigma_{WF}(S)) = ids(S), \tag{12}$$

for all proper faces S of S^n, by definition. But Σ_{WF} is not symmetric w.r.t. $\rho, \tilde{\rho}$, because $\tilde{\rho}$ sends an execution e to an execution $\alpha(e)$ with the same input values, e.g., if S_0^{n-1} is labeled with $P_1, \ldots P_n$, $\tilde{\rho}$ would send an $n-1$ simplex $S \in \Sigma_{WF}(S_0^{n-1})$ to a $n-1$ simplex S' in $\Sigma_{WF}(S_1^{n-1})$, where S_1^{n-1} is labeled with $P_0, P_2, \ldots P_n$, and such that S, S' have the same input values. But such a simplex is not in $\Sigma_{WF}(S_1^{n-1})$, since all simplexes in this carrier have input values taken from $0, 2, \ldots n$, while S has input values $1, \ldots, n$.

We can, however, construct a symmetric carrier by "gluing together" the carriers from a number of input simplexes as shown in Figure 7. Notice that this complex is a subdivided simplex (and hence acyclic), and that input names are assigned symmetrically around the boundary.

Definition 6.2. Let $S^n = (\vec{s}_0, \ldots, \vec{s}_n)$, where $id(\vec{s}_i) = P_i$. The *standard chromatic subdivision* of S^n, denoted $\chi(S^n)$, contains all vertexes of the form $\langle P_i, S \rangle$ for $S \subseteq S^n$ and $P_i \in ids(S)$. A set of vertexes form a simplex if and only if (1) the process ids are distinct, and (2) if S_i, S_j correspond to two vertexes of the set, then $S_j \subseteq S_i$ or $S_i \subseteq S_j$.

It is trivial to check that $\chi(S^n)$ is indeed a complex. (In fact, it can be shown that it is a subdivision of S^n.)

We now construct a complex $\chi'(S^\ell) \subseteq \mathcal{I}$, isomorphic to $\chi(S^\ell)$, by assigning input names

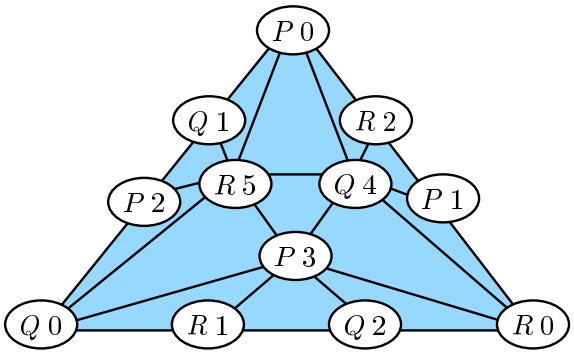

Fig. 7. Symmetric input subcomplex for renaming

to vertexes of $\chi(S^\ell)$. The input names are defined inductively. The unique vertex of $\chi'(S^0)$ has input name 0. Assume inductively that we have assigned $f(m-1) = m(m+1)/2$ input names to the vertexes in $\chi'(S^{m-1}) = \chi'(face_m(S^m))$. The rotation map that sends P_i to $P_{i+1 \bmod m+1}$ induces a bijective simplicial map

$$\rho : face_i(S^m) \to face_{i+1}(S^m)$$

by $\rho(\vec{s}_i) = \vec{s}_{i+1 \bmod m+1}$, and also

$$\rho : \chi(face_i(S^m)) \to \chi(face_{i+1}(S^m)),$$

by $\rho\langle P_i, S_i\rangle = \langle \rho(P_i), \rho(S_i)\rangle$. Every vertex $\vec{v} \in \chi(\dot{\mathcal{S}}^{m-1})$ (the boundary of S^m) is equal to $\rho^i(\vec{u})$, for some $\vec{u} \in \chi(face_m(S^m))$. Assign each vertex $\vec{v} \in \chi'(\dot{\mathcal{S}}^{m-1})$ the same input value as $\vec{u}$. Finally, for each P_i, the vertex $\langle P_i, S^m\rangle$ is the only interior vertex labeled with P_i. Assign this vertex the input value $f(m-1)+i$.

This construction uses $n(n+1)/2$ input names, by solving the recursion $f(0) = 1$, $f(m) = f(m-1) + m + 1$. Any renaming protocol for $2n+1$ input names can be transformed into a protocol for a larger number of input names simply by using the shared-memory renaming protocol of Bar-Noy and Dolev (BND89) to reduce the number of names to $2n+1$, and therefore the impossibility of $(n+1, K)$-renaming for $O(n^2)$ input names implies impossibility for $2n+1$ input names.

Corollary 6.3. There is no wait-free $(n+1, 2n-1)$-renaming protocol in read/write memory (HS93).

A similar argument yields:

Corollary 6.4. There is no t-resilient $(n+1, 2t-1)$-renaming protocol in read/write memory.

If processes share read/write memory and (m, j)-consensus objects, then it is not known whether a symmetric carrier $\Sigma_{m,j}$ can be chosen so that $ids(\Sigma_{m,j}(S^\ell)) = ids(S^\ell)$. This condition is clearly satisfied, however, when $m = n+1$, and $j > (n+1)/2$.

Corollary 6.5. There is no wait-free $(n+1, 2j-1)$-renaming protocol if processes share a read/write memory and $(n+1, j)$-consensus objects.

This result is new.

We do not analyze renaming lower bounds for synchronous message-passing systems, since it is known that $\log n$ rounds are necessary and sufficient for wait-free $(n+1, n+1)$-renaming (HT90) using comparison-based protocols.

Appendix. Appendix

This appendix gives some simple examples of chain groups, chain maps and chain homotopies, for readers unfamiliar with algebraic topology.

Examples

Let $S^2 = (\vec{s}_0, \vec{s}_1, \vec{s}_2)$ be a 2-simplex (a "solid" triangle). Let $\dot{\mathcal{S}}^1$ be the oriented complex of its proper faces (a "hollow" triangle); $\dot{\mathcal{S}}^1$ includes three 0-simplexes (vertexes): $\vec{s}_0$, $\vec{s}_1$, and $\vec{s}_2$, and three 1-simplexes: $S^1_i = \mathit{face}_i(S^2)$, $0 \leq i \leq 2$:

$$S^1_0 = (\vec{s}_1, \vec{s}_2), S^1_1 = (\vec{s}_0, \vec{s}_2), S^1_2 = (\vec{s}_0, \vec{s}_1),$$

where the vertexes are ordered as indicated in each S^1_i. Thus,

$$\partial S^1_0 = \vec{s}_2 - \vec{s}_1, \partial S^1_1 = \vec{s}_2 - \vec{s}_0, \partial S^1_2 = \vec{s}_1 - \vec{s}_0.$$

The 0-th chain group of $\dot{\mathcal{S}}^1$, $C_0(\dot{\mathcal{S}}^1)$, is generated by the $\vec{s}_i$, meaning that all 0-chains have the form

$$\lambda_0 \cdot \vec{s}_0 + \lambda_1 \cdot \vec{s}_1 + \lambda_2 \cdot \vec{s}_2,$$

where the λ_i's are integers. The first chain group, $C_1(\dot{\mathcal{S}}^1)$, is generated by the S^1_i, and all 1-chains have the form

$$\lambda_0 \cdot S^1_0 + \lambda_1 \cdot S^1_1 + \lambda_2 \cdot S^1_2.$$

Since $\dot{\mathcal{S}}^1$ contains no simplexes of higher dimension, the higher chain groups are trivial.

Let us calculate the groups of 1-cycles of $\dot{\mathcal{S}}^1$. For a 1-chain $\lambda_0 \cdot S^1_0 + \lambda_1 \cdot S^1_1 + \lambda_2 \cdot S^1_2$, we have that $\partial(\lambda_0 \cdot S^1_0 + \lambda_1 \cdot S^1_1 + \lambda_2 \cdot S^1_2)$ is equal to $\lambda_0 \cdot \partial(S^1_0) + \lambda_1 \cdot \partial(S^1_1) + \lambda_2 \cdot \partial(S^1_2)$, because ∂ is a homomorphism. Thus, the last equation is equal to $\lambda_0 \cdot (\vec{s}_2 - \vec{s}_1) + \lambda_1 \cdot (\vec{s}_2 - \vec{s}_0) + \lambda_2 \cdot (\vec{s}_1 - \vec{s}_0)$. And this is equal to 0 if and only if $\lambda_0 = \lambda_2 = -\lambda_1$. Therefore, the group of 1-cycles is generated by the cycle $S^1_0 - S^1_1 + S^1_2$, and is isomorphic to the infinite cyclic group of integers under addition.

It is easy to generalize this argument to prove that the group of n-cycles of $\dot{\mathcal{S}}^n$ is also infinite cyclic. In fact, since the group of $n+1$-chains is trivial, $H_n(\dot{\mathcal{S}}^n)$ is also infinite cyclic ($\dot{\mathcal{S}}^n$ has a hole, and one can go around it k times, for any integer k).

The rotation map $\rho : \dot{\mathcal{S}}^1 \to \dot{\mathcal{S}}^1$ defined by $\rho(\vec{s}_i) = \vec{s}_{i+1 \bmod 3}$ is a simplicial map. Therefore, it induces a chain map (abusing notation, we call it the same) $\rho : C(\dot{\mathcal{S}}^1) \to C(\dot{\mathcal{S}}^1)$. The chain map ρ is defined on simplexes as follows. For $0 \leq i \leq 2$, $\rho_0(\vec{s}_i) =$

$\vec{s}_{i+1 \bmod 3}$. In dimension 1, we have

$$\begin{aligned}\rho_1(S_0^1) &= (\vec{s}_2, \vec{s}_0) = -S_1^1 \\ \rho_1(S_1^1) &= (\vec{s}_1, \vec{s}_0) = -S_2^1 \\ \rho_1(S_2^1) &= (\vec{s}_1, \vec{s}_2) \ \ = S_0^1\end{aligned}$$

To verify that ρ is a chain map, it suffices to check that $\rho_0(\partial S_0^1) = \rho_0(\vec{s}_2 - \vec{s}_1) = \vec{s}_0 - \vec{s}_2 = \partial\rho_1(S_0^1)$, and similarly for the S_1^1 and S_2^1.

The identity simplicial map $\iota : \dot{\mathcal{S}}^1 \to \dot{\mathcal{S}}^1$ also induces a chain map $\iota : C(\dot{\mathcal{S}}^1) \to C(\dot{\mathcal{S}}^1)$. We now show that ι and ρ are chain homotopic, by displaying both an acyclic carrier, and the chain homotopy D. An acyclic carrier Σ for ι and ρ is the following: we want that $\Sigma(\vec{s}_i)$ includes both $\iota(\vec{s}_i) = \vec{s}_i$ and $\rho(\vec{s}_i) = \vec{s}_{i+1}$, so $\Sigma(\vec{s}_i)$ is the complex consisting of S_{i+2}^1 and its vertexes. And $\Sigma(S_i^1)$ is the subcomplex of $\dot{\mathcal{S}}^1$ containing $\iota(S_i^1) = S_i^1$, $\rho(S_i^1) = S_{i+1}^1$) (i.e. two edges), and their vertexes. Both ι and ρ are carried by Σ, and both $\Sigma(\vec{s}_i)$ and $\Sigma(S_i^1)$ are acyclic (being contractible). The chain homotopy D is given by $D_0(\vec{s}_i) = (-1)^{i+1} S_{i+2 \bmod 3}^1$, and $D_1(S_i^1) = 0$. It is easily verified that

$$(D\partial + \partial D)(S) = (\iota - \rho)(S).$$

Although every simplicial map induces a chain map, some chain maps are not induced by any simplicial map. Consider the chain map defined by $\phi(\vec{s}_i) = \vec{s}_i$, and $\phi(S_i^1) = S_i^1 + (-1)^i \sum_{j=0}^{2} (-1)^j S_j^1$ (notice that $\sum_{j=0}^{2} (-1)^j S_j^1 = \partial S^2$). Thus, $\phi(\partial S^2) = 4 \cdot \partial S^2$, so this map "wraps" the boundary around itself four times, something no simplicial map could do. This map is not chain homotopic to ι, although $(\phi - \iota)(S)$ is a cycle for every simplex S.

One of the referees pointed out to us that Lemma 6.1 can be viewed as a special case of the Equivariant Hopf theorem (tDP82; tD87).

Appendix. Acknowledgments

We are grateful to the anonymous referees for many helpful suggestions.

References

H. Attiya, A. Bar-Noy, D. Dolev, D. Peleg, and R. Reischuk. Renaming in an asynchronous environment. *Journal of the ACM*, 37(3):524–548, July 1990.

H. Attiya and S. Rajsbaum. A combinatorial topology framework for wait-free computability. In *Proc. 10th International Workshop on Distributed Algorithms*, volume 1151 of *Lecture Notes in Computer Science*, pages 321–343. Springer-Verlag, 1996.

H. Attiya and J. Welch. *Distributed Computing: Fundamentals, Simulations and Advanced Topics*. McGraw–Hill, 1998. First Edition.

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*, pages 91–100, May 1993.

E. Borowsky and E. Gafni. The implication of the Borowsky-Gafni simulation on the set consensus hierarchy. Technical Report 930021, UCLA Computer Science Dept., 1993.

O. Biran, S. Moran, and S. Zaks. A combinatorial characterization of the distributed 1-solvable tasks. *Journal of Algorithms*, 11:420–440, 1990.

A. Bar-Noy and D. Dolev. Shared memory vs. message passing in an asynchronous distributed environment. In *Proceedings of the 8th Annual ACM Symposium on Principles of Distributed Computing*, pages 371–382, August 1989.

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

S. Chaudhuri, M.P. Herlihy, N. Lynch, and M.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, October 1993.

M. Fischer, N.A. Lynch, and M.S. Paterson. Impossibility of distributed commit with one faulty process. *Journal of the ACM*, 32(2):374–382, April 1985.

M.P. Herlihy and S. Rajsbaum. Set consensus using arbitrary objects. In *Proceedings of the 13th Annual ACM Symposium on Principles of Distributed Computing*, pages 324–333, August 1994.

M.P. Herlihy and S. Rajsbaum. *A Primer on Algebraic Topology and Distributed Computing*, volume 1000 of *Lecture Notes in Computer Science*, pages 203–217. Springer-Verlag, 1995.

M.P. Herlihy and S. Rajsbaum. New perspectives in distributed computing. In *Proc. of the 24th Mathematical Foundations of Computer Science*, volume 1672 of *Lecture Notes in Computer Science*, pages 170–186. Springer-Verlag, September 1999.

M. Herlihy, S. Rajsbaum, and M. Tuttle. Unifying synchronous and asynchronous message-passing models. In *Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing*, pages 133–142. ACM, June 1997.

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*, pages 111–120, May 1993.

M.P. Herlihy and N. Shavit. A simple constructive computability theorem for wait-free computation. In *Proceedings of the 1994 ACM Symposium on Theory of Computing*, pages 243–252, May 1994.

M.P. Herlihy and N. Shavit. The topological structure of asynchronous computability. In *Journal of the ACM*, 1999. to appear.

M. P. Herlihy and Mark R. Tuttle. Wait-free computation in message-passing systems: Preliminary report. In *Proceedings of the 9th Annual ACM Symposium on Principles of Distributed Computing*, pages 347–362. ACM, August 1990.

Nancy Lynch and Sergio Rajsbaum. On the borowsky-gafni simulation algorithm. In *4th Israeli Symposium on Theory of Computing and Systems*, pages 4–15, June 1996.

Nancy Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, 1996. First Edition.

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

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*, pages 101–110, May 1993.

T. tom Dieck. Transformation groups. *de Gruiter Studies in Mathematics*, 8:126, 1987.

T. tom Dieck and T Petrie. Publ. maths. ihes. *de Gruiter Studies in Mathematics*, 56:337–377, 1982.