Three-Processor Tasks Are Undecidable

Eli Gafni* eli@cs.ucla.edu

Elias Koutsoupias* elias@cs.ucla.edu

Abstract

We show that no algorithm exists for deciding whether a finite task for three or more processors is wait-free solvable in the asynchronous read-write shared-memory model. This impossibility result implies that there is no constructive (recursive) characterization of wait-free solvable tasks. It also applies to other shared-memory models of distributed computing, such as the comparison-based model.

Key words: asynchronous distributed computation, task-solvability, wait-free computation, contractibility problem

AMS subject classification: 68Q05, 68Q22

1 Introduction

A fundamental area in the theory of distributed computation is the study of asynchronous wait-free shared-memory distributed algorithms. Characterizing the class of distributed tasks that can be solved, no matter how "inefficiently", is an important step towards a complexity theory for distributed computation. A breakthrough was the demonstation by Fisher, Lynch, and Paterson [FLP85] that certain simple tasks, such as *consensus*, are not solvable. Subsequently, Biran, Moran, and Zaks [BMZ88] gave a complete characterization of the tasks solvable by two processors and of tasks that can be solved when only one processor can fail. Recently, three teams [BG93, HS93, SZ93] independently extended this result by providing powerful necessary conditions for task solvability which enabled them to

*Computer Science Department, UCLA

show that the *k-set agreement task* is not solvable for more than k processors. Finally, Herlihy and Shavit [HS94] gave a simple condition that is necessary and sufficient for a given task to admit a wait-free protocol. This condition was extended by Borowsky and Gafni [Bor95] to the more general model of asynchronous distributed computation of resiliency and set-consensus.

Here, we put the quest for complete characterization of solvable tasks to an abrupt end by showing that *there is no recursive characterization of wait-free tasks*. More precisely, we show that the problem of deciding whether a given finite task for three or more processors admits a wait-free protocol is undecidable. We also show that this holds for the comparison-based model (when processors can only compare their IDs). An immediate consequence of our result is that for any recursive function $f(s)$ there are finite *solvable* tasks of size (number of input-output tuples) s that cannot be solved by any protocol in less than $f(s)$ steps. Unfortunately, this may hamper the development of a "complexity theory" of asynchronous distributed computation.

Our proof exploits a surprising connection between distributed computation and topology. In particular, we give a reduction from the *contractibility problem* to the *task-solvability problem*. The contractibility problem asks whether a given loop of a simplicial complex is contractible, that is, whether it can be continuously transformed into a point.

The history of the contractibility problem goes back to Poincaré and Dehn at the beginning of the twentieth century (see [Sti93]). Dehn [Deh10] noticed that the contractibility problem is equivalent to the word problem of groups — given a word of a group as a product of its generators, decide whether it is equal to the identity. The relation between the contractibility of a loop and the word problem comes from the fact that a loop of a complex is contractible iff the corresponding word of the *fundamental group* of the complex is the identity. Dehn gave an algorithm (Dehn's algorithm) for the contractibility problem when the complex is a surface; for some recent interesting results for this special case see [DG95]. Attempts to extend Dehn's algorithm to higher dimensional manifolds made no substantial progress, however, for the very good reason that, as Novikov [Nov55] showed in 1995, the word problem is undecidable.

The equivalence between the contractibility problem and the word problem of groups is based on the fact that every group with a finite representation (with generators and relations) is the fundamental group of a finite simplicial complex. Since the fundamental group depends only on the 2-skeleton of a complex (the collection of all simplices of dimension 2 or less), it follows that the contractibility problem is undecidable even for 2-dimensional complexes. It is also known that every group with finite representation is the fundamental group of some 4-dimensional manifold [Mas67]. Thus, the contractibility problem is undecidable even for 4-dimensional manifolds.

Our main result is a reduction from the contractibility problem to the task-solvability problem. We outline the ideas behind this reduction here. The Herlihy-Shavit condition [HS96] for task solvability is that a task is solvable iff there is a chromatic subdivision of the input complex that maps simplicially to the output complex, is consistent with the input-output relation (carrier-preserving), and preserves colors. Here we consider a class of simple 3-processor tasks that is restricted to those whose input complex consists of a single triangle (2-simplex). In addition, these tasks have the property that whenever less than three processors participate, they must output a simplex of a fixed loop L of the output complex. The Herlihy-Shavit condition implies that if the task is solvable, then L is contractible. In fact, if we drop from the Herlihy-Shavit condition the restriction that the map must be color-preserving, the opposite would be true: L is contractible if the task is solvable. The difficult part of our reduction, then, is to extend this relation to the case of chromatic complexes and color-preserving simplicial maps. To do this, we proceed in stages. We first show that the contractibility problem remains undecidable for loops of length 3 of chromatic complexes. The final reduction is to take a chromatic complex with a loop of length 3 and transform it into a 3-processor task that is solvable iff the loop is contractible.

In Section 2, we discuss the solvability problem, present the Herlihy-Shavit condition, and define the special class of tasks that we consider in this paper. In Section 3, we discuss the contractibility problem and strengthen the result that the contractibility problem is undecidable for the special case of chromatic complexes and loops of length 3. We give a reduction from this stronger version of the contractibility problem to the task-solvability problem in Section 4. The results from Section 3 and the Herlihy-Shavit condition are then used to prove that the reduction works. We conclude by discussing some of the implications of our results.

2 The task-solvability problem

We will use standard terminology from algebraic topology (see [Mun84]). All complexes considered here are *finite* and *pure*, that is, all maximal simplices have the same dimension (usually 2-dimensional).

In topology, a simplex is defined by a set of $n+1$ points, but in the theory of distributed computation, a simplex represents a consistent set of views of $n+1$ processors. The natural ordering of processors (according to their IDs) imposes structure on the complexes in that their simplices are ordered. This order defines a natural *coloring* of the vertices of the complex, where colors represent the rank of the ID of a processor. More precisely, a coloring of a n-dimensional simplicial complex is an assignment of colors $\{0, 1, \ldots, n\}$ to its vertices such that each vertex receives exactly one color and vertices of each simplex receive distinct colors. A *chromatic* simplicial complex is a simplicial complex together with a coloring.

A distributed task is a natural generalization of the notion of a (computable) function for the model of distributed computation. The computation of functions by a distributed system imposes such tight coordination of processors that only trivial functions can be computed wait-free by asynchronous distributed systems. Mainly for this reason, the study of distributed computation is focused on computing relations, a natural generalization of functions which requires less tight coordination of processors. In general, a distributed task is an input-output relation. Because in a distributed system some processors may take no steps at all, the task input-output relation must be defined on partial inputs and outputs. This requirement is captured nicely by assuming that the inputs form a chromatic simplicial complex. The vertices of a simplex of this complex denote the inputs to a subset of processors, the *participating* processors [HS96]. Similarly, the possible outputs of a distributed task form a chromatic simplicial complex.

Definition 1 *A distributed task for $n+1$ processors is a nonempty relation T between the simplices of two n-dimensional chromatic complexes I, O, $T \subset I \times O$, which preserves colors; that is, when $(A, B) \in T$, then A and B have the same colors (and therefore the same dimension).*

A distributed task is solvable when there is a distributed protocol such that the input to processor with ID k is a vertex of I with color k, its output is a vertex of O of color k, the set of the input vertices form a simplex $A \in I$, and the set of output vertices form a simplex $B \in O$ with $(A, B) \in T$. In other words, the participating processors get vertices of an input simplex and output vertices of a simplex of the output complex such that the input simplex and the output simplex form a pair of the task input-output relation. Each processor knows only its vertex, not the whole input simplex. Finding out the input simplex is usually impossible, because that task is equivalent to the consensus problem, which is not solvable. Of course, the notion of solvability depends on the computational model. Here, we consider the standard computational model of wait-free protocols for shared read-write memory. In a wait-free protocol, a processor must produce a valid output even when all other processors fail.

A typical distributed task is shown in Figure 1. The input complex I contains only one triangle $\{a, b, c\}$, and the output complex O is a subdivided triangle. The numbers on the vertices are colors. The input-output relation $T \subset I \times O$ contains the tuples $(\{a, b, c\}, \{x, y, z\})$ for all triangles $\{x, y, z\}$ of O (there are seven such triangles); it also contains all possible color-preserving tuples of simplices of the boundaries of I and O.

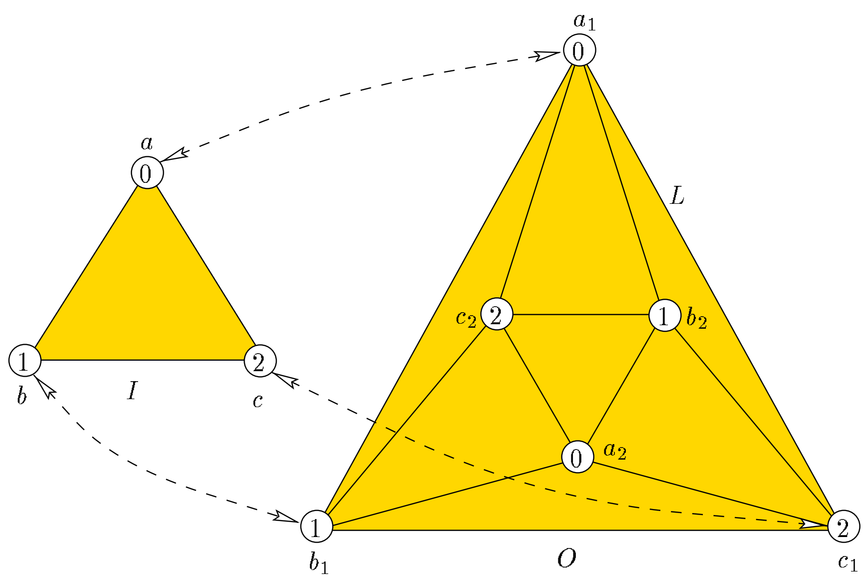

Figure 1: A standard inputless task.

A problem central to the theory of distributed computation is the characterization of the set of solvable tasks. This problem has a trivial negative answer: whether a 1-processor task is solvable is equivalent to whether the task, that is, the input-output relation, is recursive (computable); a similar observation was made in [JT92]. However, this is an unsatisfactory answer because it sheds no light on the difficulties inherent in distributed computation. Furthermore, many interesting distributed tasks are straightforward input-output relations. The interesting question, then, is whether a characterization of "simple" tasks exists. Here, we show that the answer for three or more processors remains negative, even for the simplest kind of tasks — finite tasks with a trivial input complex. For less than three processors, it is known that there exists a simple characterization for finite tasks of two processors that reduces the task-solvability problem to the connectivity properties of the output complex [BMZ88].

Our proof uses the Herlihy-Shavit condition for task solvability. Roughly speaking, this condition entails that a subcomplex of the output complex is "similar to" the input complex. To state the condition precisely, we need a few definitions: Consider a chromatic complex C and a subdivision C' of C (a subdivision of a complex is a refinement of it; see, for example, [Mun84, page 84]). For a simplex $A \in C'$, its carrier, $\mathrm{carrier}(A)$, is the smallest simplex of C that contains A. The complex C' is a *chromatic subdivision* of C if it is chromatic and its coloring has the property that each vertex $u \in C'$ has the color of some vertex of $\mathrm{carrier}(u)$.

Proposition 1 (Herlihy-Shavit) *A task $T \subset I \times O$ is solvable wait-free iff there exists a subdivision I' of I and a color-preserving simplicial map $\mu : I' \mapsto O$ such that for each simplex $A \in I'$ there exists a simplex $B \in O$ with $\mu(A) \subset B$ and $(\mathrm{carrier}(A), B) \in T$.*

A map μ that satisfies the above condition will be called *carrier-preserving* and *color-preserving*.

Proposition 1 provides a powerful tool for checking whether a particular task is solvable. For example, by applying the Herlihy-Shavit condition, we can conclude immediately that the task of Figure 1 is wait-free solvable. To see this, notice that in this case, we can take the subdivision I' to be the output complex and the map μ to be the identity map.[1] If, however, we create a "hole" in the output complex by removing the triangle $\{a_2, b_2, c_2\}$, the resulting task is not solvable; intuitively, the map μ cannot create a "torn" image of I.

The main result of our paper is to show that the condition of Proposition 1 is not constructive, namely, there is no effective way to find I' from T; computing μ is easy, since one can try all possible simplicial maps from I' to O. We will restrict our attention to the simple case of tasks of three

[1] Strictly speaking, I' is combinatorially homeomorphic to O and μ is this homeomorphism.

processors, $n = 2$. In this case, the simplices are triangles and the simplicial complexes are of dimension 2. For this dimension, our intuition about topological facts is usually correct; exactly the opposite is true for higher dimensions. We introduce one further simplification: We will deal only with tasks where the input complex consists of only one triangle. Furthermore, for each proper face of the input triangle there is exactly one possible output. In particular, there is a loop L of the output complex that has length 3 such that when less than three processors participate in the execution, the processors must output a simplex of L, and this simplex is unique because of the coloring requirements. When all three processors participate, the output can be any simplex of the output. We will call such a task a *standard inputless task* (O, L). The task of Figure 1 is an example of such a task. Since the input to each processor is fixed, we interpret a standard inputless task as follows: processors do not really get any input; rather, they simply execute a protocol in order to "agree on" some triangle of the output complex O. This could be trivially achieved (by agreeing on a triangle in advance), except for the difficulty that when some processors do not participate, the output simplex must belong to the loop L.

For a standard inputless task, the Herlihy-Shavit condition can be restated as "the task is solvable iff there is a chromatic subdivision I' of a triangle I and a color-preserving simplicial map μ that maps the boundary of I' to the loop L and that can be extended over I'." The coloring restrictions imply that the simplicial map μ maps the boundary of I' only once around L. Putting it differently, the requirement that μ is color-preserving guarantees that it is also carrier-preserving.

If we disregard colors for the moment, a standard inputless task is solvable iff there is a carrier-preserving simplicial map μ from the boundary of a subdivided triangle I' to the loop L which can be extended over the whole triangle. This condition shows the close connection between task solvability and the contractibility problem, because such I' and μ exist iff the loop L of the output complex O is contractible (we will elaborate on this connection in Section 4). It is not, however, immediate that this observation holds for the special case of chromatic complexes and color-preserving simplicial maps. Here, we extend this connection to the chromatic case by a series of reductions.

3 The contractibility problem

Let X be a topological space. A *loop* L of X is a continuous map from the 1-sphere $S = \{x \in \mathbf{R}^2 : |x| = 1\}$ to X. Two loops L and L' are *homotopic*, when L can be continuously deformed to L'. More precisely, L and L' are said to be homotopic if there exists a continuous map $F : S \times [0, 1] \mapsto X$, such that $F(x, 0) = L(x)$ and $F(x, 1) = L'(x)$ [Mun84, page 103]. A *nonsingular* loop is one without self-intersections (when the map is an injection). A loop L is *null-homotopic*, or *contractible*, when it is homotopic to a constant loop; the image of a constant loop is a point. Equivalently, loop L is null-homotopic when it can be continuously deformed to a point [ST80, page 158]. For example, in Figure 2 the loop L_1 is null-homotopic, while the loop L_2 is not.

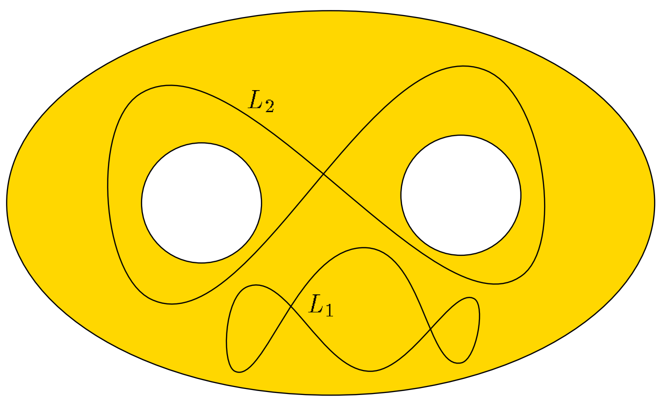

Figure 2: Contractible (L_1) and non-contractible (L_2) loops.

Let C be a simplicial complex, that is, a collection of simplices in the Euclidean space R^m. The polytope $|C|$ of C is the underlying Euclidean space consisting of the union of the simplices of C. A loop of a complex C is a simplicial loop of its polytope $|C|$. Thus, the image of a loop is a sequence of edges (v_1, v_2), $(v_2, v_3), \ldots, (v_{k-1}, v_k)$, (v_k, v_1) (the image of a null-homotopic loop is simply a vertex v). We usually do not distinguish between the loop of a complex and its image (as we have done many times so far); so, for example, we can refer to a simplex of the loop when we really mean a simplex of the image of it.

To show that task solvability is undecidable, we will use the standard technique of reducing a known undecidable problem to it. In our case, this problem is the contractibility problem [Sti93].

Definition 2 *The contractibility problem is defined as follows: given a simplicial complex C and a loop L of C, is L null-homotopic?*

For this definition to be complete, we need to fix the representation of C and L. Since we are only interested in whether the problem is decidable, the details of the representation are not important. For our purposes here, we assume that C and L are given explicitly by their simplices.

There is an important connection between the homotopic properties of loops and group theory, through the fundamental group of a complex. In particular, a loop is null-homotopic iff the corresponding word of the fundamental group is equal to the identity. This connection between contractibility and group theory results in the following proposition.

Proposition 2 *The contractibility problem is undecidable for 2-dimensional complexes.*

This folklore result is based on the fact that for every group G with a finite representation with generators and relations, there exists a finite simplicial complex with fundamental group G. This complex can be easily constructed from G (see, for example, [Sti93, page 129]). In fact, something stronger holds: each group G is the fundamental group of a 4-dimensional simplicial manifold [Mas67, pages 143–144]. This means that the contractibility problem is undecidable for 4-dimensional manifolds. In contrast, for 2-dimensional manifolds (e.g., sphere, torus, projective plane), it is decidable [Deh10]. Some recent work on this special case has led to a linear-time algorithm for almost all 2-dimensional manifolds [DG95]. The contractibility problem for 3-dimensional manifolds is, to our knowledge, still unresolved; however, it is known that not every group with finite representation can be the fundamental group of a 3-dimensional manifold.

Notice also that Proposition 2 refers to 2-dimensional complexes. This is based on the fact that the fundamental group of a complex of any dimension is identical to the fundamental group of its 2-skeleton.

Since every group can be the fundamental group of a complex, the contractibility problem is equivalent to the word problem of groups. The word problem asks whether a word of a group (as a product of its generators) is equal to the identity [Sti93, page 46]. Novikov [Nov55] showed that the word problem is undecidable: there exists a group G such that no algorithm can decide whether a word of this group is equal to the identity (for a textbook proof see [Rot95, chapter 12]). Notice that the group G need not be part of the input, although for our purposes the weaker version of the result when the group is part of the input will suffice.

We will make use of a stronger version of Proposition 2. We first observe that the contractibility problem is undecidable for *link-connected* 2-dimensional complexes. A simplicial complex is link-connected when the link of every vertex is connected (the link of a vertex is the subcomplex induced by its adjacent vertices). To see that the contractibility problem remains undecidable for link-connected complexes, notice that it is undecidable for the 2-skeleton of 4-manifolds, and clearly these complexes are link-connected. Therefore we have the stronger proposition:

Proposition 3 *The contractibility problem is undecidable for link-connected 2-dimensional complexes.*

Link-connectivity must be preserved by all our reductions, but we will not use it until the last part (Lemma 3) of the proof of the main result.

The plan for reducing this undecidable problem to the task-solvability problem is as follows: First, we strengthen Proposition 3 to chromatic complexes and loops of length 3. A chromatic complex together with a loop of length 3 defines a standard inputless task. Using the Herlihy-Shavit condition, we then show that this task is solvable iff the loop is contractible.

We begin by showing that Proposition 3 holds for nonsingular loops (i.e., loops without self-intersections).

Lemma 1 *The contractibility problem is undecidable for nonsingular loops of link-connected 2-dimensional complexes.*

Proof. Given a link-connected 2-dimensional simplicial complex C and a loop L of C, we create a new complex C' and a singular loop L' of C' such that L is null-homotopic iff L' is null-homotopic. The idea is that C' can be produced by attaching an annulus (ring) A to C: one boundary of A is identified with the loop L, and the other boundary is a nonsingular loop L' (see Figure 3). The annulus A is free of self intersections except for points of L.

We claim that L is contractible in C iff L' is contractible in C'. But first we need a definition. A topological space Y is a *deformation retract* of a topological space $X, Y \subset X$, iff there is a continuous map $f : X \times [0, 1] \mapsto X$

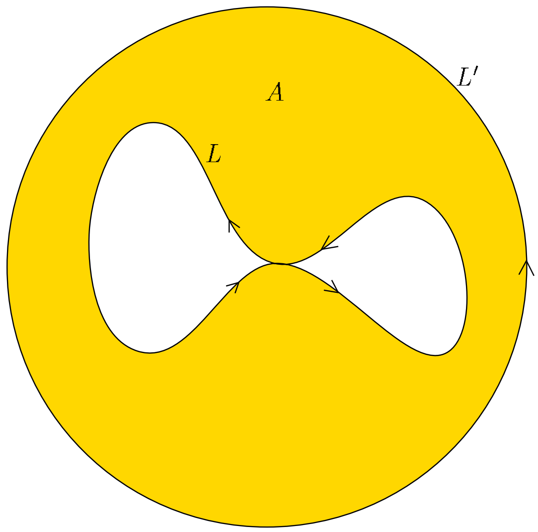

Figure 3: Reduction to nonsingular loop.

such that for all $x \in X$, $f(x, 0) = x$ and $f(x, 1) \in Y$, and for all $y \in Y$ and all t, $f(y, t) = y$.

If Y is a deformation retract of X then Y and X have the same homotopy type [Mun84, page 108]. It is clear that $|C|$ is a deformation retract of $|C'|$: f gradually collapses the annulus $|A|$ to the loop $|L|$ keeping $|C|$ fixed. It follows that $|C|$ and $|C'|$ have the same homotopy, and therefore L is contractible in C iff it is also contractible in C'. The claim follows from the fact that L and L' are homotopic in C'.

A minor issue is that the annulus A must be constructed explicitly. We give here one such construction. Let $(x_0, x_1), (x_1, x_2), \ldots, (x_{k-1}, x_k), (x_k, x_0)$ be the edges of L (some of them may be identical when part of the loop retraces itself). The boundary of annulus A identified with L contains vertices $y_0, y_1, \ldots, y_k$ such that y_i will be identified with x_i. The other boundary, L', of annulus A contains distinct vertices $z_0, z_1, \ldots, z_k$. The triangles of annulus A are $\{y_i, y_{i+1}, z_i\}$ and $\{y_{i+1}, z_i, z_{i+1}\}$, for $i = 0, 1, \ldots, k$. We have to verify that these are indeed triangles (i.e., all vertices are distinct) and that annulus A is free of self-intersections except for points in L. Some of the vertices x_i of L may be identical, because the loop L may cross or even retrace itself. However, since (x_i, x_{i+1}) is an edge of L, it follows that y_i

and y_{i+1} are distinct and therefore that the given triangulation of annulus A is valid. It is also easy to verify that annulus A has no self-intersections outside L.

Finally, we have to verify that the new complex C' is link-connected. It is clear that the links of vertices not in L are connected. Consider now the link $\mathrm{lk}(x_i)$ of a vertex $x_i \in L$. Since C is link-connected, every vertex of $C \cap \mathrm{lk}(x_i)$ is connected through $\mathrm{lk}(x_i)$ to x_{i-1} and to x_{i+1}. In particular, x_{i-1} and x_{i+1} are connected through $\mathrm{lk}(x_i)$ (or they are identical). Similarly, every vertex in $A \cap \mathrm{lk}(x_i)$ is connected to x_{i-1} or to x_{i+1}. Therefore, $\mathrm{lk}(x_i)$ is connected. $\square$

This lemma allow us to consider only nonsingular loops. We may sometimes treat a nonsingular loop L of a complex C as the 1-dimensional subcomplex of C consisting of the edges of L. We are now ready to strengthen Proposition 3 to chromatic complexes and loops of length 3.

Theorem 1 *The contractibility problem is undecidable for loops of length 3 of link-connected 2-dimensional chromatic complexes.*

Proof. Consider a link-connected 2-dimensional simplicial complex C and a nonsingular loop L of it. We will show how to produce a chromatic complex C' and a loop $L' \subset C'$ of length 3.

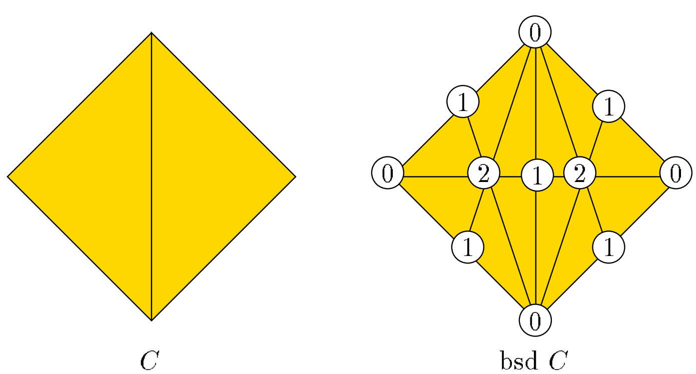

Figure 4: The chromatic barycentric subdivision.

Producing a chromatic complex is easy. Let bsd C denote the *barycentric subdivision* of the simplicial complex C [Mun84, page 85]. We can color the simplicial complex bsd C with three colors as shown in Figure 4. Original vertices of C are colored with 0, vertices on the edges — with carrier an edge — with 1, and the remaining vertices — with carrier a triangle — with color 2. With this coloring, bsd C becomes a chromatic complex. The nonsingular loop L of C corresponds to a nonsingular loop bsd L of the chromatic complex bsd C. Clearly, L is null-homotopic in C iff bsd L is null-homotopic in bsd C'. Notice also that the vertices of bsd L have colors 0 or 1.

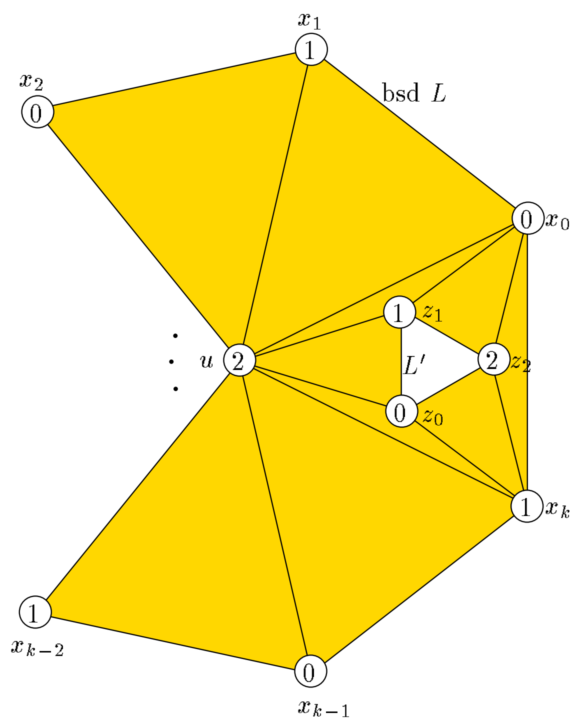

Figure 5: Reduction to loops of length 3.

Finally, to produce a complex C' and a loop L' of length 3, we employ the reduction of Lemma 1: C' is the result of attaching a chromatic annulus A to the nonsingular loop bsd L. Let (x_0, x_1), (x_1, x_2), $\ldots$, (x_{k-1}, x_k), (x_k, x_0) be the edges of bsd L. One boundary of the chromatic annulus A is identified with bsd L, while the other boundary L' contains 3 vertices, z_0, z_1, z_2, with colors 0, 1, and 2, respectively. There is also an internal vertex u of A with color 2. The chromatic annulus A is shown in Figure 5 (again, numbers on vertices indicate colors). We omit its precise description here since the reader can easily derive it from the figure. It remains to verify that A is an annulus without self-intersections, and this follows directly from the fact that bsd L is nonsingular. Note that this is the only place where we need Lemma 1. We could actually use a simpler construction by letting L' to be the loop (x_0, u, x_k), but the construction of Figure 5 is consistent with the proof of Lemma 1.

An argument identical with that of the proof of Lemma 1 establishes that the complex C' is link-connected and that L' is null-homotopic iff L is null-homotopic. The theorem follows from Lemma 1. □

The requirement that loop L has length 3 is a "technical" detail. We could prove our main result by simply considering loops L where, instead of an edge (z_i, z_{i+1}), there is a chromatic path between z_i and z_{i+1}. The restriction to loops of length 3 results in simpler constructions and proofs, however.

4 Reduction to task-solvability

To show that task-solvability for three processors is undecidable, we will reduce the stronger version of the contractibility problem of Theorem 1 to the task-solvability problem. The reduction is straightforward. Given a link-connected 2-dimensional chromatic complex C and a loop L of length 3, the output is the standard inputless task $T = (C, L)$. We will show that the loop L is contractible in C iff the standard inputless task (C, L) is solvable.

The proof is based on the two following lemmata:

Lemma 2 *Let $T = (C, L)$ be a standard inputless task. Loop L is contractible in C iff there is a subdivision I' of the input triangle I and a simplicial map $\psi : I' \mapsto C$ that is carrier-preserving.*

Proof. Notice first that we require neither that I' be a chromatic subdivision nor that μ be color-preserving. It follows directly from the definition of null-homotopic loops that loop L is contractible in C iff there is a *continuous map* ϕ from a disk B to $|C|$ that maps *homeomorphically* the boundary of the disk to $|L|$. Since the triangle I is homeomorphic to a disk, L is contractible in C iff there is a continuous map from I to $|C|$ that maps its boundary to the loop $|L|$ homeomorphically (and, therefore, simplicially). The problem with this definition is that ϕ is a continuous map, not a simplicial one.

However, a fundamental result from algebraic topology, the Simplicial Approximation Theorem [Mun84, page 89], allows us to replace the continuous map ϕ with a simplicial one. By the Simplicial Approximation Theorem, there is a subdivision I' of the triangle I and a simplicial map $\psi : I' \mapsto C$ that *approximates* ϕ. It suffices, therefore, to verify that ψ is also carrier-preserving. By the definition of simplicial approximations, for each point x of I, $\psi(x)$ is a vertex of the smallest simplex of C that contains $\phi(x)$. Since ϕ maps simplicially the boundary of I to $|L|$, all points of an edge E of I are mapped to the same edge $\phi(E)$ of L. Thus, the vertices of I' with carrier E are mapped by ψ to vertices of $\phi(E)$, which shows that ψ is carrier-preserving. $\square$

Lemma 2 shows the close connection between the contractibility of loops and the solvability of tasks. However, it only requires that the map ψ be carrier-preserving, while the Herlihy-Shavit condition requires the map to be chromatic too. The following lemma shows that this is not a problem.

Lemma 3 *Let $T = (C, L)$ be a standard inputless task, where C is link-connected. If there exists a subdivision I' of the input triangle I and a carrier-preserving simplicial map $\psi : I' \mapsto C$, then there exists a chromatic subdivision A of I and a simplicial map $\mu : A \mapsto C$ that is both carrier-preserving and color-preserving.*

Proof. The proof here is an adaptation of the proof of a similar result in [HS96, Lemma 5.21]. The basic idea is that the colors of C induce a coloring of I'. A vertex $u \in I'$ is assigned the color of its image $\psi(u) \in C$. We call such a coloring of I' ψ-induced. This coloring makes ψ a color-preserving map. However, such a coloring may not make I' a chromatic complex, because two adjacent vertices u_1 and u_2 of I' may receive the same color. Because ψ is a simplicial map, this happens only when these vertices are mapped to the same node, in which case, we say that the edge $\{u_1, u_2\}$ is monochromatic. Similarly, we say that a triangle is monochromatic when all its vertices are mapped to the same vertex.

Let A be a subdivision of I such that there is a carrier-preserving simplicial map $\mu : A \mapsto C$ such that the number of monochromatic simplices of A with the μ-induced coloring is minimum. We claim that A has no monochromatic edges or triangles. Suppose that this not the case. We will reach a contradiction by exhibiting a subdivision A' of A — and therefore of I — with one monochromatic simplex less than A.

Consider first the case when A with the μ-induced coloring has a monochromatic triangle $\{u_1, u_2, u_3\}$, and therefore $\mu(u_1) = \mu(u_2) = \mu(u_3)$. Let p be a vertex in the link of $\mu(u_1)$; such a vertex always exists because the complex C is pure. Consider now the subdivision A' of A where the triangle $\{u_1, u_2, u_3\}$ is subdivided into three triangles $\{u_1, u_2, c\}$, $\{u_1, c, u_3\}$, and $\{c, u_2, u_3\}$, where c is the barycenter of $\{u_1, u_2, u_3\}$. Consider also the map $\mu' : A' \mapsto C$ that agrees with μ on the vertices of A and $\mu'(c) = p$. But then A' with the μ'-induced coloring has one monochromatic simplex (the triangle $\{u_1, u_2, u_3\}$) less than A with the μ-induced coloring.

We now assume that no triangle of A is monochromatic but that there is a monochromatic edge $\{u_1, u_2\}$ on the boundary of A. Then, u_1 and u_2 belong to exactly one triangle of A. Let b be the third vertex of this triangle. We can construct a subdivision A' of A by subdividing the triangle $\{u_1, u_2, b\}$ into two triangles $\{u_1, c, b\}$ and $\{c, u_2, b\}$, where c is the barycenter of $\{u_1, u_2\}$. Consider the extension $\mu' : A' \mapsto C$ such that μ' agrees with μ on all vertices of A and $\{\mu'(c), \mu'(u_1)\}$ is an edge of the loop L. Because μ' is carrier-preserving, we have again reached a contradiction since A with the μ-induced coloring has one monochromatic simplex more than A' with the μ'-induced coloring.

The last, and more complicated, case to consider is when A has a monochromatic edge $\{u_1, u_2\}$ that is not in its boundary. This is the only place where we must require that complex C be link-connected. The edge $\{u_1, u_2\}$ belongs to exactly two triangles. Let a and b be the remaining vertices of these two triangles. Since μ is a simplicial map, $\mu(a)$ belongs to the link of $\mu(u_1)$ in C. Let p be a vertex in the link of $\mu(u_1)$ (not necessarily distinct for $\mu(a)$). But then the fact that C is link-connected implies that there is path with edges (p_1, p_2), $(p_2, p_3), \ldots, (p_{k-1}, p_k)$ in the link of $\mu(u_1)$ that connects $\mu(a) = p_1$ and $p = p_k$. We can always choose a non-empty path, even when $p = a$, because C is pure. Similarly, there is a path with edges (q_1, q_2), $(q_2, q_3), \ldots, (q_{k-1}, q_k)$ which connects $\mu(b)$ to $q = q_l$. This suggests the following subdivision A' of A: The triangle $\{u_1, u_2, a\}$ is subdivided into triangles $\{\bar{p}_i, \bar{p}_{i+1}, u_1\}$ and $\{\bar{p}_i, \bar{p}_{i+1}, u_2\}$, $i = 1, 2, \ldots, k-1$. The vertices $\bar{p}_i$, $i = 2, 3, \ldots, k$, are new and distinct, and $\bar{p}_k$ is the barycenter of $\{u_1, u_2\}$. Similarly, the triangle $\{u_1, u_2, b\}$ is subdivided into triangles $\{\bar{q}_i, \bar{q}_{i+1}, u_1\}$ and $\{\bar{q}_i, \bar{q}_{i+1}, u_2\}$, $i = 1, 2, \ldots, l-1$, where $\bar{q}_l = \bar{p}_k$. Consider also the extension $\mu' : A' \mapsto C$ such that μ' agrees with μ on A and $\mu'(\bar{p}_i) = p_i$, $i = 1, 2, \ldots, k$ and $\mu'(\bar{q}_i) = q_i$, $i = 1, 2, \ldots, l$. Using the fact that C is chromatic, it is easy to verify that B with the μ'-induced coloring has one monochromatic simplex less than A with the μ-induced coloring. □

An alternative proof of the Lemma 3 can be obtained by employing the Convergence Algorithm of Borowsky and Gafni [Bor95]. We outline this proof here. By the Herlihy-Shavit condition, it suffices to show that the task T is solvable. The protocol consists of two phases. In the first phase, processors "converge" on a simplex of I'. Let x_i be the vertex of I' where processor i converges. If the color of $\mu(x_i)$ is i, then the processor i stops and outputs $\mu(x_i)$. Obviously, at least one processor stops in this phase. Although the remaining processors do not know the output of the stopped processors, they know a simplex of C that contains the outputs of stopped processors. In the second phase, the remaining processors converge in C in the link of the output of all stopped processors; each of the remaining processors starts at a vertex of its color and, if possible, a vertex of the loop L. Since C is link-connected and chromatic, the remaining processors can indeed converge. Thus T is solvable.

We can now prove the main theorem of this paper.

Theorem 2 *The task-solvability problem for three or more processors in the read-write wait-free model is undecidable.*

Proof. By Theorem 1, there is no algorithm to decide, given a standard inputless task $T = (C, L)$, whether the loop L is contractible in C, when C is link-connected. However, by Lemmata 2 and 3, the loop L is contractible iff there is a chromatic subdivision I' of the input triangle I and a color-preserving and carrier-preserving simplicial map $\mu : I' \mapsto C$. This is precisely the Herlihy-Shavit condition for T to be solvable, and therefore L is contractible in C iff T is solvable. Hence, task solvability is undecidable for three processors.

This immediately implies that the solvability problem for more than three processors is also undecidable: Consider, for example, tasks where there is only one possible output of all but the first three processors; such a task is solvable iff the subtask for the first three processors is solvable. □

Biran, Moran, and Zaks [BMZ88] define a slightly different model of distributed computation in which the processors must produce a valid output only if all of them complete their protocol. For this model, the input-output relation contains only n-dimensional simplices. For each task $T \subset I \times O$, it is easy to construct an equivalent task T' in the model of [BMZ88]: The input for task T' to processor i may be a special value p_i that indicates that the processor does not "participate" in T. In that case, the processor must output a special value q_i. Otherwise, the input is a vertex of I and the output a vertex of O in such a way that the input-output relation of processors whose inputs are not special values is identical to T. It is easy to see that T is solvable iff T' is solvable in the model of [BMZ88]. This immediately implies the following.

Corollary 1 *The task-solvability problem for the model of Biran, Moran, and Zaks [BMZ88] is undecidable for three or more processors.*

Another interesting variant of the shared read-write memory model is the comparison-based model where processors cannot access directly their IDs but can only compare them [HS96]. A typical task for this model is the *renaming* task: the input (name) to each processor is a distinct member of a set S of size m, and the output must be a distinct member of a smaller set of size k. In the comparison-based model, the input to a processor is not a vertex of the input complex I but instead some value associated with the vertex. Different vertices may have the same value. Similarly there are values associated with the vertices of the output complex. This generalization in the definition of tasks is necessary for the comparison-based model to be different from the model we have considered so far; otherwise, when a processor gets as input a vertex of the input complex I, it can immediately determine its color and the rank of its ID. This suggests a trivial reduction from task-solvability to the comparison-based model task-solvability: Given a task $T \subset I \times O$, construct a comparison-based model task T' with the same input-output tuples where the value of each vertex is the vertex itself. Then, all values are distinct, and a processor can infer its color from its input. It follows that T is solvable iff T' is solvable in the comparison-based model.

Corollary 2 *The task-solvability problem for three or more processors in the comparison-based model is undecidable.*

Recently, Herlihy and Rajsbaum [HR] proposed an interesting extension of Theorem 2 to the models of resiliency and set-consensus. Using the contractibility problem, they showed that the task-solvability problem for these models is also undecidable in general.

5 Conclusion

Let us define the size of a task to be the number of its input-output tuples. Theorem 2 implies that for *any recursive function* $f(s)$, there are solvable tasks of size s whose protocols require at least $f(s)$ steps. This is indicative of the difficulty involved in developing a robust complexity theory for asynchronous distributed computation. The analogy for traditional complexity theory would be that the finite languages, a proper subset of regular languages, are nonrecursive! However, it may still be possible to develop a notion of complexity of distributed tasks that is independent of the task size. An intriguing open problem is finding a solvable "natural" task whose protocol requires, for example, exponential number of steps. Of course, one could use the reductions given in this paper to produce such a task, but that task could not be considered natural.

The Herlihy-Shavit condition (despite the title of [HS94]) is not constructive. Our results here cast some doubt on its applicability as a necessary and sufficient condition for task solvability. On one hand, the best way to show that a task is solvable is to provide a distributed algorithm that solves the given task. On the other hand, showing that a task is not solvable often employs other weaker conditions that are easier to apply than the Herlihy-Shavit condition (e.g, Sperner's Lemma or homology). However, our work does show how powerful the Herlihy-Shavit condition is, because no weaker condition would enable us to derive the results of this paper. Ironically, although our work exposes the inherent weakness of the Herlihy-Shavit condition, to our knowledge our work is the only work that makes full use of it.

Acknowledgments: The possibility that the Herlihy-Shavit condition might not be constructive was suggested by Shlomo Moran in a conversation with the first author in 1994. We would like to thank Geoffrey Mess from the UCLA Mathematics Department for providing useful pointers to the literature.

References

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

[BMZ88] Ofer Biran, Shlomo Moran, and Shmuel Zaks. A combinatorial characterization of the distributed tasks which are solvable in the presence of one faulty processor. In *Proceedings of the 7th ACM*

Symposium on Principles of Distributed Computing, pages 263–275, 1988.

[Bor95] Elizabeth Borowsky. *Capturing the power of resiliency and set consensus in distributed systems.* PhD thesis, University of California, Los Angeles, 1995.

[Deh10] Max Dehn. Über die topologie des dreidimensional raumes. *Math. Ann.*, 69:137–168, 1910. English translation by John Stillwell in: Max Dehn, *Papers on Group Theory and Topology*, Springer-Verlag, New York, 1987.

[DG95] Tamal K. Dey and Sumanta Guha. Optimal algorithms for curves on surfaces. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 266–274, Milwaukee, Wisconsin, 23–25 October 1995.

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

[GK95] Eli Gafni and Elias Koutsoupias. 3-processor tasks are undecidable. In *Proceedings of the 14th Annual Symposium on Principles of Distributed Computing*, page 271, August 1995.

[HR] Maurice Herlihy and Sergio Rajsbaum. The decidability of distributed decision tasks. Manuscript.

[HS93] Maurice Herlihy and Nir Shavit. The asynchronous computability theorem for t-resilient tasks. In *Proceedings of the 25th ACM Symposium on the Theory of Computing*, pages 111–120, 1993.

[HS94] Maurice Herlihy and Nir Shavit. A simple constructive computability theorem for wait-free computation. In *Proceedings of the 26th ACM Symposium on the Theory of Computing*, pages 101–110, 1994.

[HS96] Maurice Herlihy and Nir Shavit. The topological structure fo asynchronous computability. Technical Report CS-96-03, Department of Computer Science, Brown University, 1996.

[JT92] Prasad Jayanti and Sam Toueg. Some results on the impossibility, universality and decidability of consensus. In *Proceedings of the 6th Workshop on Distributed Algorithms*, pages 68–84, 1992.

[Mas67] William S. Massey. *Algebraic Topology: An Introduction*. Harcourt, Brace and World, New York, 1967.

[Mun84] James R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

[Nov55] P. S. Novikov. On the algorithmic unsolvability of the word problem in group theory (in Russian). *Trudy Mat. Inst. Steklov*, 44:143, 1955.

[Rot95] Joseph J. Rotman. *An Introduction to the Theory of Groups*. Graduate texts in mathematics. Springer-Verlag, New York, 4th edition, 1995.

[ST80] H. Seifert and W. Threlfall. *A Textbook in Topology*. Academic Press, New York, 1980.

[Sti93] John Stillwell. *Classical Topology and Combinatorial Group Theory*. Graduate texts in mathematics. Springer-Verlag, New York, 2nd edition, 1993.

[SZ93] Michael Saks and Fotios Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. In *Proceedings of the 26th ACM Symposium on the Theory of Computing*, pages 101–110, 1993.