A Generalized Asynchronous Computability Theorem

Eli Gafni
Computer Science Department, UCLA
eli@ucla.edu

Petr Kuznetsov
Télécom ParisTech
petr.kuznetsov@telecom-paristech.fr

Ciprian Manolescu *
Department of Mathematics, UCLA
cm@math.ucla.edu

ABSTRACT

We consider the models of distributed computation defined as subsets of the runs of the iterated immediate snapshot model. Given a task T and a model M, we provide topological conditions for T to be solvable in M.

When applied to the wait-free model, our conditions result in the celebrated Asynchronous Computability Theorem (ACT) of Herlihy and Shavit.

To demonstrate the utility of our characterization, we consider a task that has been shown earlier to admit only a very complex t-resilient solution. In contrast, our generalized computability theorem confirms its t-resilient solvability in a straightforward manner.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*relations between models*

Keywords

asynchronous computability, iterated models, characterization, topology

1. INTRODUCTION

This paper characterizes task solvablility in models of distributed computing, where processes communicate via reading from and writing to a shared memory. We treat a model as a set of *runs*, i.e., interleaving of read and write steps issued by different processes.

What do we mean by a characterization? We say that a task T is solvable in a model M, if there exists a *protocol* by which, in every run of M, each process taking sufficiently many steps eventually *outputs*, so that the outputs satisfy the task's specification with respect to the provided inputs. The conventional definition of solvability is therefore *operational*, based on the existence of a protocol. A topological characterization replaces the operational definition with the existence of a continuous map between topological spaces, capturing the sets of possible inputs and outputs of the task. This topological characterization may provide insights about the (in)solvability of the task that are not easy to grasp operationally [1, 21, 32].

In 1993, Herlihy and Shavit [21, 22] characterized read-write communication with no restrictions on the runs; this is referred to as the *wait-free* model. They formulated the Asynchronous Computability Theorem (ACT) stating that a task T is wait-free solvable *if and only if* there exists a simplicial, chromatic map from a subdivision of simplexes of an appropriately defined input simplicial complex to an appropriately defined output simplicial complex, satisfying the specification of T.

The original proof of ACT is given in [22] directly for the conventional read-write shared-memory model (referred to as the *standard shared-memory* and denoted SM), where memory consists of persistent objects which can be written and read by a given process arbitrarily often. However, the original proof can be simplified by casting the problem to the *iterated immediate-snapshots* (IIS) [3] shared memory model, in which processes march through a sequence of Immediate-snapshots (IS) tasks invoking the next one with their output from the previous one.

The IIS model can be thought of as convenient mathematical tool to analyze and understand distributed computing, contrasting with the more realistic but less convenient standard shared-memory model (SM). Casting distributed computation in SM to IIS is not unlike analyzing electromagnetic communication in the complex-number domain. We start with the real-world, we move the reasoning to an abstract mathematical world, and then we translate the results back to reality.

The proof of ACT in [22] can thus be simplified along the following lines:

(1) The wait-free runs in standard shared-memory can be simulated in the IIS model [3, 13], and the IIS model can be simulated by the standard shared-memory model [1]. Thus all we need is to characterize task solvability in IIS.

(2) If a task T is wait-free solvable in the IIS model, then there exists an integer k_T such that after the first k_T invocations of immediate snapshots, each process can output in T. This can be shown by a compactness argument or König's lemma (cf., e.g., [3]).

*CM was supported by NSF grant DMS-1104406.

PODC'14, July 15–18, 2014, Paris, France.

ACM 978-1-4503-2944-6/14/07 ...$15.00..

(3) Solving a task by k_T immediate snapshots can be interpreted topologically as constructing a simplicial map from the k_T-th standard chromatic subdivision of the input complex to the output complex. This is because each immediate snapshot can be represented by a standard subdivision [23, 25].

(4) A chromatic map from an arbitrary subdivision of a complex can be approximated by a chromatic map from an iterated standard chromatic subdivision. This is a chromatic version of the standard simplicial approximation theorem, and a proof can be found in Section 5 of [22]. It also follows from an operational argument in [3].

In this paper we extend this method of proof to models representing proper subsets of the runs of the wait-free standard shared-memory model.

To this end, we start with a formal definition of the very notion of solving a task in IIS. Surprisingly, no such definition appeared in the literature until now.[1] In particular, we introduce the notions of *participating* (taking at least one step) and *infinitely participating* (taking infinitely many steps) processes in an IIS run.

Further, to benefit from Step (1) in the list above, we need a mapping between SM and IIS, that preserves the notions of participating and infinitely participating sets. Furthermore, in addition to this forward simulation $F:\ \mathrm{SM} \to \mathrm{IIS}$ we need a backward simulation $B:\ \mathrm{IIS} \to \mathrm{SM}$, such that for all $M \subseteq \mathrm{SM}$ we have $B(F(M)) \subseteq M$. We also ask for the restriction of B to the image of F to preserve the notions of participating and infinitely participating sets. Given that the standard simulations [2, 3] do not meet these requirements, we employ a new two-way simulation presented in [4]. This applies to a large set of *adversarial* SM models [7, 24], specified by sets of processes that can be infinitely participating in a model run.

Thus, we can cast an adversarial SM model to its equivalent model in IIS. Moreover, the IIS model is, in a strict sense, richer than SM: multiple IIS runs collapse into a single SM run by the simulation [4]. Our characterization of IIS task solvability applies to any *sub-IIS model* (that is, a subset of the runs in the IIS model), including those that have no equivalents in SM.

Imitating Step (2) in sub-IIS models is in general impossible. For example, in the 1-resilient 3-process model, the task of 2-set agreement can be easily solved. However, every such solution has a run in which two processes fail and the remaining process never outputs [1, 21, 32]. In every 1-resilient extension of a finite prefix of this run, all infinitely participating processes output. Since such a prefix may have an arbitrary length, a uniform bound k_T on the number of steps sufficient to output in a run of the 1-resilient sub-IIS model does not exist. This observation is related to the fact that the model is *non-compact*, with respect to a metric that will be defined in Section 5.

Therefore, our main theorem, which we call the *generalized* asynchronous computability theorem (GACT), is a generalization of Step (3) in the outline above, without relying on Step (2). Instead, the characterization proposes to "approximate" a non-compact model by a sequence of compact models. The sequence converges to a superset of the target model. Thus, if the task is solvable by each compact model in the sequence, it is solvable by the target model. Each compact model is represented as subcomplex of a subdivided simplex. Hence GACT deals with a sub-complex of a subdivided simplex instead of a subdivided simplex (as ACT does), and instead of saying that there exists a single subdivision (as in ACT) it requires the existence of a sequence of sub-complexes.

Step (4) now applies individually to each subdivision in the sequence. In this paper we do not deal with this step, because the formulation would be too cumbersome, and because it is not necessary for our examples. Nevertheless, we believe that our theorem can be extended to arbitrary rather than standard chromatic subdivisions.

ACT turned out to be an essential tool in distributed computing [6, 12, 17–20]. We show that our GACT holds that promise too. We consider a task T, solvable t-resiliently, but (to our knowledge) only with a very involved algorithm [8]. In contrast, by applying the methods developed in this paper, we show that determining the t-resilient solvability of T is relatively simple.

The paper is organized as follows: In Section 2 we describe the IIS model and give some examples of sub-IIS models. In Section 3 we review some notions from combinatorial topology. In Section 4 we review the topological definition of a task, and explain what it means for a task to be solvable in a model. In Section 5 we describe sub-IIS models topologically. In Section 6 we prove our main result, GACT. In Section 7 we explain how GACT gives back the well-known ACT in the wait-free case. In Section 8 we introduce a new topological tool: a version of the simplicial approximation for infinite chromatic complexes. Using this tool, in Section 9 we show how GACT can be applied to a class of tasks called link connected; in particular, we use GACT to prove that a particular task can be solved in the t-resilient model. In Section 10 we recall some related work, and in Section 11 we draw the conclusions.

2. SUB-IIS MODELS

In this section, we describe our perspective on the *Iterated Immediate Snapshot* (IIS) model [3] and give examples of sub-IIS models.

2.1 The IIS model

Suppose we have $n+1$ processes $p_0, p_1, \dots, p_n$. A run r in IIS is a sequence of *non-empty* sets of processes $S_1 \supseteq S_2 \supseteq \dots$, with each $S_k \subseteq \{p_0, \dots, p_n\}$ consisting of those processes that participate in the kth iteration of immediate snapshot (IS). Furthermore, each S_k is equipped with an ordered partition: $S_k = S_k^1 \cup \cdots \cup S_k^{n_k}$ (for some $n_k \leq n$), corresponding to the order in which processes are invoked in the respective IS.

Let $\mathcal{R}$ be the set of runs in IIS. Fix a run $r \in \mathcal{R}$, with $r = S_1, S_2, \dots$ as above. The processes $p_i \in S_1$ are called *participating*. If p_j appears in all the sets S_k, we say that p_j is *infinitely participating* in r. The sets of participating and infinitely participating processes in a run r are denoted $\mathit{part}(r)$ and $\infty\text{-}\mathit{part}(r)$, respectively.

If either $k = 0$ or $p_i \in S_k$ for some $k \geq 1$, then we define a set called the *kth view of p_i in the run r*, $\mathit{view}(p_i, k)$, recursively, as follows:

1. $\mathit{view}(p_i, 0) = \{p_i\}$;

[1] Some specific *compact* subsets of IIS runs were formally treated in [29].

2. For $k \geq 1$, the view of $p_i \in S_k^j \subseteq S_k$ is $view(p_i, k) = \{view(p_s, k-1) \mid p_s \in S_k^1 \cup \cdots \cup S_k^j\}$.

Our definitions can be interpreted operationally as follows. Every process proceeds through an infinite series of one-shot immediate snapshot (IS) instances [2]: $IS_1, IS_2 \ldots$. Then S_k is interpreted as the set of processes accessing memory IS_k, and each S_k^j is the set of processes obtaining the same *view* after accessing IS_k. Recall that in IS, the view of a process $p_i \in S_k^j$ is defined by the values written by the processes in $S_k^1 \cup \cdots \cup S_k^j$.

The original definition of IIS [3] can be thought of as the variant of our model, in which we impose the condition $S_1 = S_2 = \cdots = \{p_0, \ldots, p_n\}$, i.e., every process is infinitely participating. What is the advantage of our new, more refined, definition of IIS? It allows a run to be extended to more processes without changing the views of the already existing processes. For instance, in the run $r = \{\{p_0\}\}, \{\{p_0\}\}, \ldots$ we have $part(r) = \infty\text{-}part(r) = \{p_0\}$. In the run $r' = \{\{p_0\}, \{p_1\}\}, \{\{p_0\}, \{p_1\}\}, \{\{p_0\}, \{p_1\}\}, \ldots$ we have $part(r') = \infty\text{-}part(r') = \{p_0, p_1\}$. However, p_0 cannot tell whether it is in r or in r', because the corresponding views of p_0 are the same in both runs. In this situation, we say that r' is an extension of r.

Formally, we say that a run $r' = S'_1, S'_2, \ldots$ is an *extension* of a run $r = S_1, S_2, \ldots$, and we write $r \leq r'$, if (i) $S_j \subseteq S'_j$ for all j, and (ii) the views of the processes in $part(r)$ are the same in r' as in r. This defines a partial order on $\mathcal{R}$.

If r is a run, let $minimal(r)$ be the smallest run r_0 such that $r_0 \leq r$ (that is, for all $r' \leq r$, we have $r_0 \leq r'$). It is not difficult to see that $minimal(r)$ exists and is unique. We then define $fast(r) = \infty\text{-}part(minimal(r))$. We define $slow(r)$ to be the complement set of $fast(r)$.

Intuitively, $fast(r)$ is the largest set of processes that "see" each other (appear in each other's view) infinitely often in r. In other words, for all $p_i, p_j \in fast(r)$ and all $k \geq 0$, there exists $\ell \geq k$ such that $view(p_i, k)$ appears in $view(p_j, \ell)$.

2.2 Examples of models

We define a *sub-IIS model* M to be any subset of $\mathcal{R}$.

EXAMPLE 2.1. *The* wait-free *(or* completely asynchronous*) model WF is the set $\mathcal{R}$ itself. The interpretation of WF is that anything can happen (all sorts of step interleavings are allowed).*

EXAMPLE 2.2. *For $t \leq n$, the* t-resilient model Res_t *consists of the runs $r \in \mathcal{R}$ such that $|fast(r)| \geq n + 1 - t$. This is the model in which at most t processes are slow.*

EXAMPLE 2.3. *For $k \leq n + 1$, the* k-obstruction-free model OF_k *consists of all the runs r in which no more than k processes are fast, i.e., $|fast(r)| \leq k$. This model was previously discussed in [11], following a suggestion of Guerraoui.*

EXAMPLE 2.4. *More generally, consider the* model with adversary $\mathbb{A}$ *[7], which we denote by $M^{\text{adv}}(\mathbb{A})$. Here, $\mathbb{A}$ is any subset of the power set of $\{0, 1, \ldots, n\}$. We then define $M^{\text{adv}}(\mathbb{A})$ to consist of all runs r such that $slow(r) \in \mathbb{A}$.*

3. TOPOLOGICAL DEFINITIONS

Before moving forward, we need to review several notions from topology. We will assume that the reader has a basic knowledge of metric spaces (open sets, continuity, compactness), as in [31, Chapter 7].

3.1 Simplicial complexes

A good reference for the material in this section is Chapter 3 in [33].

A *simplicial complex* is a set V, together with a collection C of finite nonempty subsets of V such that:

(a) For any $v \in V$, the one-element set $\{v\}$ is in C;

(b) If $\sigma \in C$ and $\sigma' \subseteq \sigma$, then $\sigma' \in C$.

The elements of V are called *vertices*, and the elements of C are called a *simplices*. We usually drop V from the notation, and refer to the simplicial complex as C.

A simplicial complex C is called *finite* if the collection C is finite. A weaker notion is *locally finite*: C is said to be locally finite if every vertex of C belongs to only finitely many simplices in C. For simplicity, we will assume that our complexes are locally finite.

A subset of a simplex is called a *face* of that simplex.

A *subcomplex* of C is a subset of C that is also a simplicial complex.

The *dimension* of a simplex $\sigma \in C$ is its cardinality minus one. The k-skeleton of a complex C, denoted $\text{Skel}^k C$, is the subcomplex formed of all simplices of C of dimension k or less.

A simplicial complex C is called *pure* of dimension n if C has no simplices of dimension $> n$, and every k-dimensional simplex of C (for $k < n$) is a face of an n-dimensional simplex of C.

Given a simplex $\sigma \in C$, we denote by $\text{st}\, \sigma$ the *open star* of σ, that is, the set of all simplices in C that have σ as a face. The *closed star* of σ, denoted $\text{St}\, \sigma$, is the smallest simplicial complex that contains $\text{st}\, \sigma$. The difference $(\text{St}\, \sigma) \setminus (\text{st}\, \sigma)$ is called the *link* of σ.

Let A and B be simplicial complexes. A map $f : A \to B$ is called *simplicial* if it is induced by a map on vertices; that is, f maps vertices to vertices, and for any $\sigma \in A$, we have

$$f(\sigma) = \bigcup_{v \in \sigma} f(\{v\}).$$

A simplicial map f is called *noncollapsing* (or *dimension-preserving*) if $\dim f(\sigma) = \dim \sigma$ for all $\sigma \in A$.

Any simplicial complex C has an associated *geometric realization* $|C|$, defined as follows. Let V be the set of vertices in C. As a set, we let C be the subset of $[0, 1]^V = \{\alpha : V \to [0, 1]\}$ consisting of all functions α such that $\{v \in V \mid \alpha(v) > 0\} \in C$ and $\sum_{v \in V} \alpha(v) = 1$. For each $\sigma \in C$, we set $|\sigma| = \{\alpha \in |C| \mid \alpha(v) \neq 0 \Rightarrow v \in \sigma\}$. Each $|\sigma|$ is in one-to-one correspondence to a subset of $\mathbb{R}^n$ of the form $\{(x_1, \ldots, x_n) \in [0, 1]^n \mid \sum x_i = 1\}$. We put a metric on $|C|$ by $d(\alpha, \beta) = \sum_{v \in V} |\alpha(v) - \beta(v)|$.

Given a simplicial map $f : A \to B$, there is an associated continuous, piecewise linear map $|f| : |A| \to |B|$, defined by the formula

$$|f|(\alpha)(v') = \sum_{f(v) = v'} \alpha(v).$$

A nonempty complex C is called *k-connected* if, for each $m \leq k$, any continuous map of the m-sphere into $|C|$ can be extended to a continuous map over the $(m + 1)$-disk.

A *subdivision* of a simplicial complex C is a simplicial complex C' such that:

1. The vertices of C' are points of $|C|$.
2. For any $\sigma' \in C'$, there exists $\sigma \in C$ such that $\sigma' \subset |\sigma|$.
3. The piecewise linear map $|C'| \to |C|$ mapping each vertex of C' to the corresponding point of C is a homeomorphism.

In particular, every complex C admits a *barycentric subdivision* $\mathrm{Bary}(C)$, defined as follows. The vertices of $\mathrm{Bary}(K)$ are the barycenters of the simplices of C (in the geometric realization). The simplices of $\mathrm{Bary}(K)$ correspond to ordered sequences $(\sigma_0, \dots, \sigma_m)$ of simplices of C, where σ_i is a face of σ_{i+1}; the barycenters of σ_i are then the vertices of the corresponding simplex in $\mathrm{Bary}(K)$.

By iterating this construction k times we obtain the kth barycentric subdivision, $\mathrm{Bary}^k(C)$.

3.2 Chromatic complexes

We now turn to the chromatic complexes used in distributed computing, and recall some notions from [22].

Fix $n \geq 0$. The *standard n-simplex* $\mathbf{s}$ has $n+1$ vertices, in one-to-one correspondence with $n+1$ *colors* $0, 1, \dots, n$. A face $\mathbf{t}$ of $\mathbf{s}$ is specified by a collection of vertices from $\{0, \dots, n\}$. We view $\mathbf{s}$ as a complex, with its simplices being all possible faces $\mathbf{t}$. Note that the open star of a face $\mathbf{t}$ is $\mathrm{st}\,\mathbf{t} = \{\mathbf{t}' \mid \mathbf{t} \subseteq \mathbf{t}'\}$, while the closed star of any face is the whole simplex $\mathbf{s}$.

A *chromatic complex* is a simplicial complex C together with a noncollapsing simplicial map $\chi : C \to \mathbf{s}$. Note that C can have dimension at most n. We usually drop χ from the notation. We write $\chi(C)$ for the union of $\chi(v)$ over all vertices $v \in C$. Note that if $C' \subseteq C$ is a subcomplex of a chromatic complex, it inherits a chromatic structure by restriction.

In particular, the standard n-simplex $\mathbf{s}$ is a chromatic complex, with χ being the identity.

Every chromatic complex C has a *standard chromatic subdivision* $\mathrm{Chr}\,C$. Let us first define $\mathrm{Chr}\,\mathbf{s}$ for the standard simplex $\mathbf{s}$. The vertices of $\mathrm{Chr}\,\mathbf{s}$ are pairs $(i, \mathbf{t})$, where $i \in \{0, 1, \dots, n\}$ and $\mathbf{t}$ is a face of $\mathbf{s}$ containing i. We let $\chi(i, \mathbf{t}) = i$. Further, $\mathrm{Chr}\,s$ is characterized by its n-simplices; these are the $(n+1)$-tuples $((0, \mathbf{t}_0), \dots, (n, \mathbf{t}_n))$ such that:

(a) For all $\mathbf{t}_i$ and $\mathbf{t}_j$, one is a face of the other;

(b) If $j \in \mathbf{t}_i$, then $\mathbf{t}_j \subseteq \mathbf{t}_i$.

The geometric realization of $\mathbf{s}$ can be taken to be the set $\{\mathbf{x} = (x_0, \dots, x_n) \in [0,1]^{n+1} \mid \sum x_i = 1\}$, with the vertex i corresponding to the point $\mathbf{x}^i$ with i coordinate 1 and all other coordinates 0. Then, we can identify a vertex $(i, \mathbf{t})$ of $\mathrm{Chr}\,\mathbf{s}$ with the point

$$\frac{1}{2k-1}\mathbf{x}_i + \frac{2}{2k-1}\Big(\sum_{\{j \in \mathbf{t} \mid j \neq i\}} \mathbf{x}_j\Big) \in |\mathbf{s}| \subset \mathbb{R}^{n+1},$$

where k is the cardinality of $\mathbf{t}$. (Compare [22, Definition 5.7].) Thus, $\mathrm{Chr}\,\mathbf{s}$ becomes a subdivision of $\mathbf{s}$ and the geometric realizations are identical: $|\mathbf{s}| = |\mathrm{Chr}\,\mathbf{s}|$.

Next, given a chromatic complex C, we let $\mathrm{Chr}\,C$ be the subdivision of C obtained by replacing each simplex in C with its chromatic subdivision. Thus, the vertices of $\mathrm{Chr}\,C$ are pairs (p, σ), where p is a vertex of C and σ is a simplex of C containing p. If we iterate process this m times we obtain the m^{th} chromatic subdivision, $\mathrm{Chr}^m\,C$.

Let A and B be chromatic complexes. A simplicial map $f : A \to B$ is called a *chromatic map* if for all vertices $v \in A$, we have $\chi(v) = \chi(f(v))$. Note that chromatic map is automatically noncollapsing. A chromatic map has chromatic subdivisions $\mathrm{Chr}^m f : \mathrm{Chr}^m A \to \mathrm{Chr}^m B$. Under the identifications of topological spaces $|A| \cong |\mathrm{Chr}^m A|, |B| \cong |\mathrm{Chr}^m B|$, the continuous maps $|f|$ and $|\mathrm{Chr}^m f|$ are identical.

A *chromatic multi-map* between A and B is a map $\Delta : A \to 2^B$ that, for any $m \leq n$, takes every m-simplex of A to a pure m-dimensional subcomplex of B, such that: (i) For every simplex σ of A, we have $\chi(\sigma) = \chi(\Delta(\sigma))$, and (ii) For all simplices $\sigma, \tau \in A$, we have $\Delta(\sigma \cap \tau) \subseteq \Delta(\sigma) \cap \Delta(\tau)$. In particular, if σ' is a face of σ, then $\Delta(\sigma') \subseteq \Delta(\sigma)$.

4. TASKS

4.1 Definitions

A *task* $T = (\mathcal{I}, \mathcal{O}, \Delta)$ on $n+1$ processes $\{p_0, \dots, p_n\}$ consist of two finite, pure n-dimensional chromatic complexes $\mathcal{I}$ and $\mathcal{O}$, together with a chromatic multi-map $\Delta : \mathcal{I} \to 2^{\mathcal{O}}$. The *input complex* $\mathcal{I}$ specifies the possible input values, the *output complex* $\mathcal{O}$ specifies the possible output values, and Δ describes which output values are allowed for a given input. The colors specify to which process each input or output value corresponds.

A task is called *input-less* if the input complex is the standard simplex $\mathbf{s}$, colored by the identity. Then each process starts with input only its own id.[2]

4.2 Affine tasks

Many examples of input-less tasks can be constructed as follows. Let $L \subseteq \mathrm{Chr}^k\,\mathbf{s}$ be a pure n-dimensional subcomplex of the k^{th} chromatic subdivision of $\mathbf{s}$, for some k. For each face $\mathbf{t} \subseteq \mathbf{s}$, the intersection $L \cap \mathrm{Chr}^k\,\mathbf{t}$ is a subcomplex of $\mathrm{Chr}^k\,\mathbf{s}$; we assume that this subcomplex is pure of the same dimension as $\mathbf{t}$ (and possibly empty).

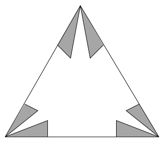
We define an input-less task $(\mathbf{s}, L, \Delta)$ by setting $\Delta(\mathbf{t}) = L \cap \mathrm{Chr}^k\,\mathbf{t}$ for any face $\mathbf{t} \subseteq \mathbf{s}$. Tasks constructed like this are called *affine*. To depict an affine task, we can simply draw the corresponding complex L.

By abuse of notation, we will usually write L for the affine task $(\mathbf{s}, L, \Delta)$. We chose the name *affine* because if we have a task L as above, the geometric realizations of the simplices of L can be depicted as lying on affine subspaces of $\mathbb{R}^n$. Similar terminology appears in algebraic geometry, where one talks about affine varieties.

For example, consider the task of *total order* L^{ord}, defined as follows. For each permutation α of $\{0, 1, \dots, n\}$, there is a unique n-simplex σ_α in the second chromatic subdivision $\mathrm{Chr}^2\,\mathbf{s}$ with the property that the vertex of σ_α colored i is in the interior of an i-dimensional face of $\mathbf{s}$. For example, for 3

[2]Note that in the definition of a multi-map we allowed images to be empty. This is somewhat non-standard, as it means that processes in a task do not have to output. If one prefers to avoid that, for every task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ we can construct a new, equivalent task $T^+ = (\mathcal{I}^+, \mathcal{O}^+, \Delta^+)$ as follows. We let $\mathcal{I}^+ = \mathcal{I}$. The output complex $\mathcal{O}^+$ is obtained from $\mathcal{O}$ by adding extra vertices $v_0, \dots, v_n$ (with v_i corresponding to "no output" for the process i); moreover, for each simplex σ in $\mathcal{O}$, we add an n-simplex σ^+ in $\mathcal{O}^+$ by adjoining vertices v_i for the colors i not represented in σ. Finally, we let $\Delta^+(\tau) = (\Delta(\tau))^+$.

processes, the six simplices of the form σ_α are those shown here:



The total order task is the affine task associated to the complex $L^{\text{ord}} \subseteq \text{Chr}^2 \mathbf{s}$ is the union of all the $(n+1)!$ simplices of the form σ_α. The name total order refers to the fact that the possible outputs (when all $n+1$ processes are running) are in one-to-one correspondence with the total orderings (i.e., permutations) of the set of processes $\{0, 1, \dots, n\}$, similar to *one-shot total-order broadcast* [15] where every process proposes broadcasts its identifier and the processes agree on the order in which the identifiers are delivered.

4.3 Views with input

Let $r = S_1, S_2, \dots$ be a run in IIS. Recall that in Section 2.1 we defined the kth view of a process in the run, $view(p_i, k)$. We now generalize this to allow arbitrary inputs.

Let $\mathcal{I}$ be a pure n-dimensional chromatic complex, and let $\omega \in \mathcal{I}$ be an n-dimensional simplex. The kth view of p_i in the run r starting with input ω is denoted $view(p_i, \omega, k)$ and defined recursively as follows:

1. $view(p_i, \omega, 0) = \{(p_i, v)\}$, where v is the vertex colored i in the simplex ω;
2. For $k \geq 1$, the view of $p_i \in S_k^j \subseteq S_k$ is $view(p_i, \omega, k) = \{view(p_s, \omega, k-1) \mid p_s \in S_k^1 \cup \cdots \cup S_k^j\}$.

4.4 Task Solvability

In a sub-IIS model, informally, a task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ is solvable in M if for all runs $r \in M$, the infinitely participating processes output, and their output is a subsimplex of the allowed outputs for the participating processes. An output is the result of a *protocol.* For us, when dealing with solvability rather than complexity, a protocol is just a partial map from views to outputs. Thus, requiring an infinitely participating process to output means requiring that eventually it will have a view that is mapped by the protocol to an output value.

We define the set $\mathcal{V} = \mathcal{V}(\mathcal{I})$ to consist of all possible $view(p_i, \omega, k)$ in all runs $r \in \mathcal{R}$, for all processes p_i, simplices $\omega \in \mathcal{I}$, and integers $k \geq 0$. Formally, a protocol Π for the task T is a map from a subset of $\mathcal{V}$ to the set of vertices in the output complex $\mathcal{O}$.

DEFINITION 4.1. *A task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ is* solvable *in a sub-IIS model M if there exists a protocol Π for T such that for all $r \in M$ (with $r = S_1, S_2, \dots$ as before):*

1. *For each p_i, and for each n-dimensional simplex $\omega \in \mathcal{I}$, there exist k_0 and a vertex v of $\mathcal{O}$ colored i, such that:*
   - *For all $k < k_0$, $view(p_i, \omega, k) \notin domain(\Pi)$;*
   - *For all $k \geq k_0$ such that $p_i \in S_k$ exists, we have $\Pi(view(p_i, \omega, k)) = v$.*

   (This condition is satisfied vacuously if p_i is not infinitely participating, because we can find k_0 such that p_i did not take k_0 steps in r, so $p_i \notin S_k$ for $k \geq k_0$.)
2. *For all k, $\{\Pi(view(p_i, \omega, k)) \mid view(p_i, \omega, k) \in domain(\Pi)\}$ is a sub-simplex of a simplex in $\Delta\big(\omega \cap \chi^{-1}(part(r))\big)$.*

In every run $r \in M$, condition (1) above requires every infinitely participating to eventually produce an output, and condition (2) requires the produced output to respect the task specification Δ given the inputs of participating processes.

4.5 Example: solving tasks in sub-IIS

Note that our definition of task solvability in sub-IIS models brings illuminating subtleties that were not observed in the conventional SM model. Consider a sub-IIS model M and the corresponding model $M_{fast} = \{r \mid \exists r' \in M, r = minimal(r')\}$. If a task T is solvable in M then it is obviously solvable in M_{fast}, but not necessarily vice-versa. Indeed, consider the obstruction-free model $OF = OF_1$, consisting of runs with a single fast process. Obviously, the total order task L^{ord} cannot be solved in OF, because in runs r where the process in $fast(r)$ is always ahead of the rest ($S_k = fast(r)$ for all k), the rest of the processes essentially proceed *wait-free.* In contrast, we can easily solve L^{ord} in OF_{fast} using commit-adopt [9] (implemented in IIS).

5. TOPOLOGICAL INTERPRETATION

Recall that $\mathcal{R}$ denotes the set of runs in IIS. We put a metric on $\mathcal{R}$ as follows. Given runs $r, r' \in \mathcal{R}$, we let $k = k(r, r')$ denote the largest $k \geq 0$ such that the first k steps of r and r' are identical. (In particular, we let $k = \infty$ when $r = r'$.) We set the distance between r and r' to be $d(r, r') = 1/(1+k)$. It is easy to see that $d(r, r')$ is a metric that captures how "close" the two runs are.

Recall that a metric space is compact if every open cover has a finite subcover; or, equivalently, if any infinite sequence has a convergent subsequence. (See [31, Section 7.7], for example.) For future reference, we mention:

LEMMA 5.1. *The metric space $\mathcal{R}$ is compact.*

PROOF. Let $r[1] = (S[1]_1, S[1]_2, \dots)$, $r[2] = (S[2]_1, S[2]_2, \dots)$, $\dots$ be an infinite sequence of runs. (Each $S[i]_k$ is a set equipped with an ordered partition.) There are only finitely many possibilities for the first step S_1^j, so we can find a subsequence $(r[1,1], r[1,2], \dots)$ of $(r[1], r[2], \dots)$ such that the first step in each $r[1,i]_q$ is a constant choice S_1, with a constant partition $S_1 = S_1^1 \cup \cdots \cup S_1^{n_1}$. From the subsequence $(r[1,1], r[1,2], \dots)$ we can extract a further subsequence $(r[2,1], r[2,2], \dots)$ such that the second step is a constant choice $S_2 = S_2^1 \cup \cdots \cup S_2^{n_2}$, and so on. Hence, the diagonal subsequence $(r[1,1], r[2,2], r[3,3], \dots)$ converges to $r = (S_1, S_2, \dots)$. Thus, every sequence in $\mathcal{R}$ has a converging subsequence. $\square$

The metric space $\mathcal{R}$ is not easy to visualize. We can however get a partial understanding by focusing on the views of the fast processes in each run.

Consider the standard chromatic subdivisions of the n-simplex $\mathbf{s}$. Recall that a run in IIS can be identified with a sequence of simplices $\sigma_0, \sigma_1, \sigma_2, \dots$, with $\sigma_k \in \text{Chr}^k \mathbf{s}$ and $|\sigma_{k+1}| \subset |\sigma_k|$ [23,25].

Note that every run converges to a point of the geometric realization $|\mathbf{s}|$, so there is a natural, continuous map $\pi : \mathcal{R} \to |\mathbf{s}|$, which we call the *affine projection.* The information captured in $p = \pi(r)$ exactly consists of the views of the fast processes in r. In fact, each point $\pi(r) \in |\mathbf{s}|$ can be identified with the minimal run *minimal*(r).

There is a canonical coloring map $\chi : |\mathbf{s}| \to 2^{\{0,1,\ldots,n\}}$ that extends the colorings on all chromatic subdivisions $\mathrm{Chr}^m\, \mathbf{s}$ to $|\mathbf{s}|$. Precisely, given a point $p \in |\mathbf{s}|$, we let $\chi(p)$ be the minimal subset $A \subseteq \{0, 1, \ldots, n\}$ such that p lies in a simplex σ of a chromatic subdivision $\mathrm{Chr}^k\, \mathbf{s}$ with $\chi(\sigma) = A$. It is easy to see that $\chi(p) = \mathit{fast}(r)$, for any r such that $\pi(r) = p$.

A special case of a sub-IIS model is a set of runs of the form $\pi^{-1}(S)$, where $S \subseteq |\mathbf{s}|$. We call such models *geometric*, because they can be easily visualized as associated subsets of $|\mathbf{s}|$. Notice that all models in Examples 2.1-2.4 are geometric. However, our main results will apply equally well to *all* (not necessarily geometric) sub-IIS models.

6. THE GENERALIZED ASYNCHRONOUS COMPUTABILITY THEOREM

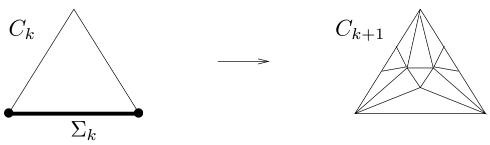
6.1 Terminating subdivisions

Let C be a chromatic complex. Consider standard chromatic subdivisions $\mathrm{Chr}^m\, C$ for $m > 0$ (Section 4), and recall that the vertices of $\mathrm{Chr}^m\, C$ can be identified with a subset of the vertices in $\mathrm{Chr}^{m+1}\, C$.

A *terminating subdivision* $\mathscr{T}$ of C is specified by a sequence of chromatic complexes $C_0, C_1, C_2, \ldots$, and a sequence of subcomplexes $\Sigma_0 \subseteq \Sigma_1 \subseteq \Sigma_2 \subseteq \ldots$ such that for all $k \geq 0$:

(i) Σ_k is a subcomplex of C_k;

(ii) $C_0 = C$, and C_{k+1} is obtained from C_k by taking the partial chromatic subdivision in which the simplices in Σ_k are "terminated", i.e., not further subdivided. Precisely, we replace a simplex σ in C_k by a coarser subdivision than $\mathrm{Chr}(\sigma)$. Whereas the vertices of $\mathrm{Chr}(\sigma)$ are pairs (p, τ) with τ being a face of σ and p a vertex of τ, in C_{k+1} we consider the pairs (p, τ) of that form such that either $\tau \notin \Sigma_k$, or τ consists of a single vertex in Σ_k. For example, if Σ_k is zero-dimensional, then $C_{k+1} = \mathrm{Chr}^1\, C_k$; if C_k is the standard 2-dimensional simplex and Σ_k is one of its 1-dimensional faces, we have:



A simplex of Σ_k for some k is called a *stable simplex* in the subdivision $\mathscr{T}$; such a simplex remains unchanged in all further complexes $C_{k+1}, C_{k+2}, \ldots$. The union $\cup_k \Sigma_k$ of stable simplices in $\mathscr{T}$ forms a chromatic complex, which we denote by $K(\mathscr{T})$; it usually has infinitely many vertices. Observe that the geometric realization $|K(\mathscr{T})|$ can be identified with a subset of $|C|$.

In particular, if there exists k such that $\Sigma_k = C_k$, then we must have $C_k = C_{k+1} = \ldots$, and $\mathscr{T}$ is just a finite subdivision of C; in this case, all the simplices in C_k are stable, and $|K(\mathscr{T})| = |C|$. At the other extreme, if Σ_k is empty for all k, then $\mathscr{T}$ consists of the k^{th} chromatic subdivisions of C for all k; in this case, no simplices are stable, and $K(\mathscr{T})$ is empty.

Stable simplices intend to model processes that have produced outputs and thus, intuitively, do not need to communicate among themselves any longer. Therefore, stable simplices are not further subdivided. However, processes with outputs keep participating in the computation: simplices that contain non-stabilized vertices continue to be subdivided. This will allow us to formulate the conditions of task solvability in non-compact sub-IIS models.

6.2 The main result

We are now ready to formulate and prove our main result: a characterization of taks solvability in sub-IIS models.

Let $M \subseteq \mathcal{R}$ be a sub-IIS model on $n+1$ processes. Let $\mathscr{T}$ be a terminating subdivision of a pure n-dimensional chromatic complex $\mathcal{I}$, and let $\chi : \mathcal{I} \to \mathbf{s}$ be the coloring map. Let $\rho : |\mathcal{I}| \to |\mathbf{s}|$ be the geometric realization (piecewise linear extension) of the map χ. Note that ρ maps vertices of $\mathrm{Chr}^k\, \mathcal{I}$ to vertices of $\mathrm{Chr}^k\, \mathbf{s}$ of the same color.

Recall that each vertex v in $\mathscr{T}$ belongs to $\mathrm{Chr}^k(\sigma)$ for some $k \geq 0$ and some n-dimensional simplex σ of $\mathcal{I}$. Thus, $\rho(v)$ is a vertex of $\mathrm{Chr}^k(\mathbf{s})$. If we have a simplex $\tau \in K(\mathscr{T})$, then $\rho(|\tau|)$ is the convex hull of $\rho(v)$ for $v \in \tau$.

We say that $\mathscr{T}$ is *admissible* for M if for any run $r \in M$ (viewed as a sequence of simplices $\sigma_0, \sigma_1, \ldots$ in $\mathbf{s}$) and for every n-dimensional simplex ω in $\mathcal{I}$, there exists $k > 0$ and a stable simplex $\tau \in K(\mathscr{T})$ such that $|\tau| \subseteq |\omega|$ and $|\sigma_k| \subseteq \rho(|\tau|)$. The intuition here is that every run of M with inputs ω should eventually land in a simplex of $\mathscr{T}$.

THEOREM 6.1 (GACT). *A sub-IIS model M solves a task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ if and only if there exists a terminating subdivision $\mathscr{T}$ of $\mathcal{I}$ and a chromatic map $\delta : K(\mathscr{T}) \to \mathcal{O}$ such that:*

(a) $\mathscr{T}$ is admissible for the model M;

(b) For any simplex σ of $\mathcal{I}$, if τ is a stable simplex of $\mathscr{T}$ such that $|\tau| \subseteq |\sigma|$, then $\delta(\tau) \in \Delta(\sigma)$.

PROOF. "$\Rightarrow$" : Suppose M solves T using a protocol Π. By induction on the recursion that defines *view*(p_i, ω, k), it is easy to see that the k^{th} view of p_i in a run $r \in M$ (with input ω) corresponds to a vertex $v \in \mathrm{Chr}^k(\omega) \subseteq \mathrm{Chr}^k(\mathcal{I})$ with $\chi(v) = \{p_i\}$.

We construct a terminating subdivision $\mathscr{T}$ with desirable properties as follows. We proceed with the standard subdivisions $\mathrm{Chr}^k(\mathcal{I})$ for $k = 0, 1, 2, \ldots$, and we examine all runs $r \in M$. At the k^{th} stage we take the inductively constructed C_k, whose vertices are a subset of the vertices of $\mathrm{Chr}^k(\mathcal{I})$. We then terminate those simplices for which Π has given an output: A simplex σ of C_k (with $|\sigma| \subseteq |\omega|$) is included in Σ_k if there exists a run $r \in M$ such that the vertices of σ are of the form $v_i = \mathit{view}(p_i, \omega, k)$ for that run, and the outputs $\Pi(v_i)$ exist (that is, $v_i \in \mathit{domain}(\Pi)$). Then Σ_k determines C_{k+1}.

Given a simplex $\sigma \in \Sigma_k$ with vertices v_i, we set $\delta(\sigma)$ be the simplex with vertices $\Pi(v_i)$.

Part (a) (admissibility of $\mathscr{T}$) follows from the construction: Given any run $r \in M$ and a top-dimensional simplex ω in $\mathcal{I}$, pick k such that all the processes infinitely participating in r have produced output at the k^{th} step when they are given input from ω. Let σ_k be the corresponding simplex of $\mathrm{Chr}^k\, \mathbf{s}$. If $\rho^{-1}(|\sigma_k|) \cap |\omega|$ is an embedded simplex of C_k, then it is necessarily a stable simplex (because all the processes have output), and we are done. If $\rho^{-1}(|\sigma_k|) \cap |\omega|$

is not an embedded simplex of C_k, then, by construction, it is contained in a simplex of C_k that was terminated before (because some of the processes have produced outputs at an earlier time).

Part (b) follows from the fact that M solves T using Π.

"$\Leftarrow$" : Conversely, suppose there exists a terminating subdivision $\mathscr{T}$ and a map $\delta : K(\mathscr{T}) \to \mathcal{O}$ as in the statement of the theorem. We construct a protocol Π by which M solves T. Suppose we have a run $r \in M$, corresponding to a sequence of simplices $\sigma_0 \subseteq \sigma_1 \subseteq \sigma_2 \subset \dots$ Since $\mathscr{T}$ is admissible for M, for each input ω there exists a stable simplex τ such that $|\tau| \subseteq |\omega|$ and $|\sigma_k| \subseteq \rho(|\tau|)$ for all $k \gg 0$. Given a process $p_i \in \infty\text{-}part(r)$, we can assign it as output value the vertex of $\delta(\tau)$ that has color p_i. Now, p_i may obtain an output value (necessarily the same as before) through another run, at a different step k. We take the minimum over all such k, to obtain the value k_0 needed in the definition of Π. Condition (b) implies that Π solves T. □

7. THE WAIT-FREE MODEL

For the wait-free model WF, let us see how we can derive the original Asynchronous Computability Theorem of [22]. Indeed, Theorem 6.1 has the following:

COROLLARY 7.1. *A task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ is solvable in the wait-free model if and only if there exists $k \geq 0$ and a chromatic map $\eta : \mathrm{Chr}^k \mathcal{I} \to \mathcal{O}$ such that, for any simplex $\tau \subseteq \mathcal{I}$ and any subsimplex σ of $\mathrm{Chr}^k \tau \subset \mathrm{Chr}^k \mathcal{I}$, we have $\eta(\sigma) \in \Delta(\tau)$.*

PROOF. If $\eta : \mathrm{Chr}^k \mathcal{I} \to C$ exists, solvability of T follows from GACT because $\mathrm{Chr}^k \mathcal{I}$ (with all the vertices terminated at the k^{th} step) is a terminating subdivision that is admissible for WF.

Conversely, suppose that T is wait-free solvable. GACT provides a terminating subdivision $\mathscr{T}$ that is admissible for WF, and a map $\delta : K(\mathscr{T}) \to \mathcal{O}$. For each run $r \in WF = \mathcal{R}$ and top-dimensional input simplex $\omega \in I$, there is a k such that $|\sigma_k|$ is contained in a stable simplex τ of $\mathscr{T}$ with $|\tau| \subseteq |\omega|$. Since there are finitely many possibilities for ω, we can find a $k = k(r)$ that works for all ω. Let $\mathcal{R}_k$ be the set of runs r for which $k(r) \leq k$. We have inclusions $\mathcal{R}_0 \subseteq \mathcal{R}_1 \subseteq \mathcal{R}_2 \subseteq \dots$, and each $\mathcal{R}_k$ is open in the metric on $\mathcal{R}$ introduced in Section 5. We know from Lemma 5.1 that the set $\mathcal{R}$ is compact. Hence, the open cover $\mathcal{R}_k$ of $\mathcal{R}$ admits a finite subcover; i.e., there exists k such that $\mathcal{R}_k = \mathcal{R}$. We now define the desired map $\eta : \mathrm{Chr}^k \mathcal{I} \to \mathcal{O}$ by setting $\eta(\sigma) = \delta(\tau)$, where $\tau \in K(\mathscr{T})$ is the minimal simplex with $|\sigma| \subseteq |\tau|$. □

As stated in [22], ACT characterizes solvability in terms of a map from an arbitrary colored subdivision of $\mathcal{I}$ to the output complex. That any colored subdivided simplex can be approximated by $\mathrm{Chr}^k(\mathcal{I})$ for some k large enough is a purely topological result, proved in [22], and which can be used here verbatim. (This corresponds to Step 4 in the outline of the proof of ACT from the Introduction.)

8. SIMPLICIAL APPROXIMATION

To be able to apply GACT, we need a tool for constructing chromatic maps between two chromatic complexes A and B (subject to some boundary conditions). In many cases, it is easier to first construct a continuous map $f : |A| \to |B|$. Standard results in algebraic topology (reviewed in Subsection 8.1 below) say that after replacing A by a fine enough subdivision, we can deform f into a geometric realization of a simplicial map. Such a map may collapse the dimension of simplices, so it is not always clear how to turn it into a chromatic map. However, if we impose an additional condition (link-connectedness for the target), we will show in Subsection 8.2 that one can do the approximation using chromatic maps.

8.1 Classical results

Let A and B be simplicial complexes and let $f : |A| \to |B|$ be a continuous map between their geometric realizations. If A' is a subdivision of A, a simplicial map $\phi : A' \to B$ is called a *simplicial approximation* to f if for every $x \in |A| = |A'|$ and $\sigma \in B$ we have

$$f(x) \in |\sigma| \;\Rightarrow\; |\phi|(x) \in |\sigma|.$$

Roughly, the simplicial approximation theorem says that every continuous map between simplicial complexes can be approximated by a simplicial map. There are several versions of this in the literature. For finite simplicial complexes, we have:

THEOREM 8.1. *Let A and B be simplicial complexes such that A is finite, and let $f : |A| \to |B|$ be a continuous map. Then:*

(a) There exists an integer N such that for all $n \geq N$, the map f admits a simplicial approximation $\phi : \mathrm{Bary}^n(A) \to B$.

(b) Furthermore, if we have a subcomplex $C \subseteq A$ such that the restriction of f to C is the geometric realization of a simplicial map $g : C \to B$, then the approximation ϕ can be taken so that the restriction of ϕ to $|C|$ equals $|g|$.

Part (a) of this result is a special case of Theorem 8 in [33, p.128]. The theorem is stated in [33] in more generality, for pairs of simplicial complexes. Part (b) above follows from this more statement, taking into account Lemma 1 in [33, p.126].

In this paper we will need a different variant of the simplicial approximation theorem, one that applies without the hypothesis that A is finite:

THEOREM 8.2. *Let A and B be simplicial complexes, and let $f : |A| \to |B|$ be a continuous map. Then:*

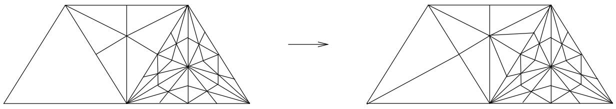
(a) There exists a subdivision A' of A such that the map f admits a simplicial approximation $\phi : A' \to B$.

(b) Furthermore, if we have a subcomplex $C \subseteq A$ such that the restriction of f to C is the geometric realization of a simplicial map $g : C \to B$, then the approximation ϕ can be taken so that the restriction of ϕ to $|C|$ equals $|g|$.

Theorem 8.2 is mentioned in the remarks at the bottom of p.128 in [33]; see [34] or [28] for a more complete treatment.

Note that, in the case when A is countable and locally finite, we can deduce Theorem 8.2 from Theorem 8.1 as follows. Let us write A as a union of finite simplicial complexes $A_1 \subseteq A_2 \subseteq \dots$ We construct the subdivision A' and the map ϕ inductively. Suppose we found a simplicial approximation

$\phi_k : \mathrm{Bary}^{n_k}(A_k) \to B$ for the restriction of f to A_k. Consider the restriction of f to $|A_{k+1}| = |\mathrm{Bary}^{n_k}(A_{k+1})|$. We extend the approximation ϕ_k to A_{k+1} by using part (b) of Theorem 8.1, applied to $\mathrm{Bary}^{n_k}(A_{k+1})$ and B. The result is a simplicial approximation $\phi_{k+1} : \mathrm{Bary}^{n_{k+1}}(A_{k+1}) \to B$ for some $n_{k+1} \geq n_k$, such that $|\phi_k| = |\phi_{k+1}|$ on $|A_k|$. The desired approximation $\phi : A' \to B$ has $|\phi| = |\phi_k|$ on each $|A_k|$. A subtle point here is the construction of the subdivision A', which is getting finer and finer as we go towards infinity. In principle, we would like A' to be $\mathrm{Bary}^{n_k}(A_k)$ on each $|A_k| \setminus |A_{k-1}|$. This is not a simplicial complex, but we can turn it into one by introducing additional simplices, as shown in the figure:



The local finiteness of A ensures that there is an upper bound on the number of times we have to subdivide each simplex.

8.2 Chromatic approximations

Let us go back to Theorem 8.1. Note that if A is a chromatic complex, then instead of the barycentric subdivisions $\mathrm{Bary}^n(A)$, one could take standard chromatic subdivisions $\mathrm{Chr}^n(A)$. (The same proof applies.) However, we cannot a priori conclude that the simplicial approximation is a chromatic simplicial map. For example, if the continuous map f collapses a simplex of A to a single vertex in B, then any simplicial approximation would do the same, but on the other hand chromatic maps are non-collapsing.

Nevertheless, we can avoid collapsing by assuming that the following property (for the target complex B):

Definition 8.3 (Definition 4.14 in [22]). *A pure n-dimensional complex B is called* link-connected *if for all simplices $\sigma \in B$, the link of σ in B is $(n-\dim(\sigma)-2)$-connected.*

For example, the output complex L^{ord} for the total order task on three processes is not link-connected, because the link (in L^{ord}) of a vertex of $\mathbf{s}$ is not connected.

A variant of Theorem 8.2 for chromatic maps is proved in [22, Lemma 4.21] under the assumptions that A and B are chromatic complexes, B is link-connected, A is a finite subdivision of the standard simplex, and C is the boundary of A. The conclusion is that the map g can be taken to be chromatic. Furthermore, Theorem 5.29 in [22] shows that, under the same assumptions, the subdivision A' of A can be taken to be a standard chromatic subdivision; this yields a chromatic variant of Theorem 8.1.

One can generalize the results of Herlihy and Shavit to the setting of infinite complexes:

Theorem 8.4. *Let A and B be chromatic simplicial complexes, and let $f : |A| \to |B|$ be a continuous map. Suppose that A is countable and locally finite, and that B is link-connected. Then:*

(a) There exists a subdivision A' of A such that the map f admits a chromatic simplicial approximation $\phi : A' \to B$.

(b) Furthermore, if we have a subcomplex $C \subseteq A$ such that the restriction of f to C is the geometric realization of a chromatic simplicial map $g : C \to B$, then the approximation ϕ can be taken so that the restriction of ϕ to $|C|$ equals $|g|$.

Proof. We do this inductively on the skeleta of A. Suppose we have defined the map ϕ on a subdivision the k-skeleton $\mathrm{Skel}^k(A)$. We apply [22, Lemma 4.21] to the restriction of f to each $(k+1)$-simplex σ of A, mapped to the $(k+1)$-skeleton of B. (Observe that if B is link-connected, then so are its skeleta.) We obtain a simplicial approximation to f on σ, agreeing with the already constructed approximation on the boundary of σ. In the process we have to subdivide the simplices σ' in $\mathrm{Skel}^k(A)$, and each σ' is on the boundary of several $(k+1)$-simplices σ. However, by local finiteness, we can find a sufficiently fine subdivision that works for all $\sigma \supset \sigma'$. By continuing this ad infinitum, we obtain the approximation ϕ. Furthermore, if we have a subcomplex C as in part (b), then at each step we arrange so that the approximation agrees with the one defined on the corresponding skeleton of C. $\square$

9. LINK-CONNECTED TASKS

9.1 A general result

The following proposition is an extension of the work of Herlihy and Shavit from [22] to the case of *infinite* chromatic complexes.

Proposition 9.1. *Let M be a sub-IIS model, and $\mathscr{T}$ a terminating subdivision of $\mathcal{I}$ that is admissible for M. Suppose we have a task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ such that the complexes $\Delta(\tau)$ are link-connected, for any $\tau \in \mathcal{I}$. Then, the task T is solvable in M if and only if there exists a continuous map $f : |K(\mathscr{T})| \to |\mathcal{O}|$ such that $f(|K(\mathscr{T})| \cap |\tau|) \subseteq |\Delta(\tau)|$ for all $\tau \in \mathcal{I}$.*

Proof. If $\mathscr{T}$ and $\mathscr{T}'$ are terminating subdivisions of $\mathbf{s}$, we say that $\mathscr{T}'$ is a *stable refinement* of $\mathscr{T}$ if $|K(\mathscr{T})| = |K(\mathscr{T}')|$, and every simplex of $\mathscr{T}'$ is contained in a simplex of $\mathscr{T}$; i.e., $K(\mathscr{T}')$ should be a subdivision of $K(\mathscr{T})$. Note that if $\mathscr{T}$ is admissible for a model M, then so is $\mathscr{T}'$.

Given the continuous map f, we shall construct a simplicial, chromatic approximation $\delta : K(\mathscr{T}') \to \mathcal{O}$ as needed to apply GACT; here, $\mathscr{T}'$ is a stable refinement of $\mathscr{T}$.

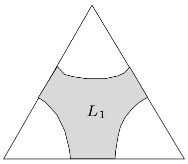
We first construct a chromatic subdivision K' of $K(\mathscr{T})$, whose vertices are not necessarily in the standard chromatic subdivisions of $\mathcal{I}$, and a chromatic map $\delta' : K' \to \mathcal{O}$ (an approximation to f) such that δ' is carrier-preserving: $\delta'(\sigma) \in \Delta(\tau)$ when $|\sigma| \subseteq |\tau|$. We do this inductively on $d \geq 0$: For each d, we define the values of δ' on the simplices that are contained in d-dimensional faces of $|\mathcal{I}|$. Suppose we have defined δ' for $d-1$, and pick a d-dimensional face τ of $|\mathcal{I}|$. The restriction of f to $|K(\mathscr{T})| \cap |\tau|$ can be approximated by a simplicial map from a subdivision of $K(\mathscr{T})$, extending the already constructed δ' on the $(d-1)$-dimensional boundary. Further, since $K(\mathscr{T})$ is locally finite (by definition) and $\Delta(\tau)$ is link-connected, it follows from Theorem 8.4 that we can arrange for δ' to preserve colors.

Thus, we find a sufficiently fine stable refinement $\mathscr{T}'$ of $\mathscr{T}$ and a chromatic, carrier-preserving map $g : K(\mathscr{T}') \to K'$. We then set $\delta = \delta' \circ g$ and apply Theorem 6.1.

Conversely, if T is solvable in M, we can apply GACT and obtain a terminating subdivision $\mathscr{T}$ and a chromatic map $\delta : K(\mathscr{T}) \to \mathcal{O}$. The desired continuous map f is the geometric realization of δ. $\square$

9.2 An example of GACT in action

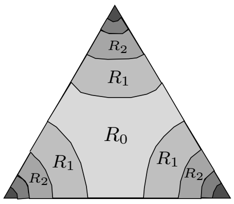
Consider the t-resilient model Res_t from Example 2.2. Let L_t be the affine task with output complex consisting of all the simplices σ in the second chromatic subdivision $\mathrm{Chr}^2\,\mathbf{s}$ such that no vertex of σ is on an $(n-t-1)$-dimensional face of $\mathbf{s}$. For example, when $n = 2$ and $t = 1$, the output complex for L_1 looks like:



PROPOSITION 9.2. *The task L_t is solvable in the model Res_t.*

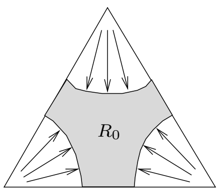
PROOF. Note that for each face $\mathbf{t} \subseteq \mathbf{s} = \mathcal{I}$, the complex $\Delta(\mathbf{t})$ for the task L_t is link-connected. Therefore, it suffices to find a terminating subdivision $\mathscr{T}$ and a continuous map f with the properties required in Proposition 9.1.

For $n \geq 0$, let $\tilde{R}_n \subset |\mathbf{s}|$ be the union of (the geometric realizations of) all the simplices $\sigma \subset \mathrm{Chr}^{n+2}\,\mathbf{s}$ such that no vertex of σ is on an $(n-t-1)$-dimensional face of $\mathbf{s}$. Let $R_0 = |L_t|$ and, for $n > 0$, let R_n be the closure of $\tilde{R}_n - \tilde{R}_{n-1}$. The union of all R_n's is the complement of the $(n-t-1)$-skeleton of $\mathbf{s}$:



The terminating subdivision $\mathscr{T}$ is as follows: It starts with $\Sigma_0 = \Sigma_1 = \emptyset$, so that $C_0 = \mathbf{s}, C_1 = \mathrm{Chr}^1\,\mathbf{s}, C_2 = \mathrm{Chr}^2\,\mathbf{s}$. We then let Σ_2 be the subcomplex of $\mathrm{Chr}^2\,\mathbf{s}$ supported in the region R_0. This defines C_2 so that on the complement of R_0 it consists of the simplices in $\mathrm{Chr}^3\,\mathbf{s}$. After this, we terminate all the simplices in R_1. Now on the complement of $R_0 \cup R_1$ we have the fourth chromatic subdivision. We then terminate the simplices in R_2, and so on. By construction, eventually, every simplex contained in any R_k is stable in this terminating subdivision. Since the affine projection $\pi(Res_t)$ is contained in the union $|K(\mathscr{T})|$ of all the R_k's, we deduce that $\mathscr{T}$ is admissible for Res_t.

It remains to construct the continuous map $f : |K(\mathscr{T})| \to |L_t| = R_0$. We let the restriction of f to R_0 be the identity, and map everything else onto the boundary $R_0 \cap R_1$ using radial projection away from the $(n-t-1)$-skeleton of $\mathbf{s}$:



Observe that radial projection preserves boundaries, so Proposition 9.1 applies. $\square$

REMARK 9.3. *An alternative, operational solution of task L_t via t-resilient atomic-snapshots was given by the first author in [8]. More precisely, the Red-Yellow-Green algorithm in [8, Section 4] specifies an intricate simulation scheme that allows for solving L_t.*

10. RELATED WORK

The topological conditions of wait-free task solvability were expressed by Herlihy and Shavit [21, 22] in the form of ACT. In the restricted case of *colorless* tasks that, roughly, can be defined without taking process identifiers in mind, Herlihy and Rajsbaum [19, 20] derived task solvability conditions in adversarial shared-memory models [7]. This paper proposes a characterization of generic (not necessarily colorless) tasks in any (not necessarily adversarial) sub-IIS model.

The IIS model was introduced by Borowsky and Gafni [3] and shown to precisely capture the standard chromatic subdivision of the input complex [23, 25]. Due to the elegance of its topological representation, IIS has been widely used topological reasoning about distributed computing [1, 3, 16, 21, 22]. In [3, 13], IIS has been shown equivalent to SM in terms of task solvability. Rajsbaum et al. [29] and, more recently, Raynal and Stainer [30] relate proper subsets of sub-IIS and sub-SM models restricted using specific failure detectors. A recent paper [4] extends these equivalences to arbitrary sub-SM and sub-IIS models, thus justifying the choice of IIS as a model of study.

The difficulty of dealing with certain problems in certain non-compact models, such as consensus and t-resilience, has been studied before by Lubitsch and Moran [26], Brit and Moran [5], Moses and Rajsbaum [27]. By deriving topological solvability conditions for any task and any sub-IIS model, this paper brings this work to a higher level of generality. The continuous space $|\mathbf{s}|$ has appeared previously in the work of Saks and Zaharoglou [32] where it was used to derive the impossibility wait-free set agreement.

11. CONCLUDING REMARKS

We presented a version of a generalization of ACT. Other versions may be possible through the relation between simplicial and continuous maps, as well as through defining terminating subdivisions not necessarily with respect to $\mathrm{Chr}^m(\mathbf{s})$, in analogy to the sufficiency condition of ACT [22]. We chose the simplest version which still provides us with the benefit of producing a "topological solution" to the given task.

The main technical challenge we faced was to define and view the IIS model directly, rather than just through the prism of the simulations from the standard (non-iterated) model [3, 13]. This brought forth a coherent view of IIS, as well as exposed the richness of the model.

The generic IIS models considered in this paper are just arbitrary subsets of the various possible interleaving of reads and writes, which is an extension with respect to the previous attempts to model sub-IIS computations [29, 30]. Yet, distributed computing refers also to the availability of one-shot objects, e.g., consensus, k-set agreement, etc. Of course, we can produce a sub-IIS model which is equivalent to having consensus, or any other simple object. An open question is whether our framework embedded in a large enough dimension can capture "all of distributed computing," at least with respect to terminating computations. In particular, what will be the sets of runs that correspond to the availability of the Möbius task [14] or a task from the family of 0-1 exclusion [10]? We know that, for instance, the "symmetric" task on n processes from [14] can also be formulated as a regular task on $2n-1$ processes, hence increasing the task dimension does help here. Our speculation is that any computability question for a "reasonable" one-shot problem in distributed computing is equivalent to a question of task solvability in a sub-IIS model.

Acknowledgement

We are in debt to Robert F. Brown for helpful discussions in the very early stages of this research.

12. REFERENCES

[1] E. Borowsky and E. Gafni. Generalized FLP impossibility result for t-resilient asynchronous computations. In *STOC*, pages 91–100, May 1993.

[2] E. Borowsky and E. Gafni. Immediate atomic snapshots and fast renaming. In *PODC*, pages 41–51, 1993.

[3] E. Borowsky and E. Gafni. A simple algorithmically reasoned characterization of wait-free computation (extended abstract). In *PODC*, pages 189–198, 1997.

[4] Z. Bouzid, E. Gafni, and P. Kuznetsov. Live equals fast in iterated models. *CoRR*, abs/1402.2446, 2014. http://arxiv.org/abs/1402.2446.

[5] H. Brit and S. Moran. Wait-freedom vs. bounded-freedom in public data structures. *J. UCS*, 2(1):2–19, 1996.

[6] A. Castañeda, M. Herlihy, and S. Rajsbaum. An equivariance theorem with applications to renaming. In *LATIN*, pages 133–144, 2012.

[7] C. Delporte-Gallet, H. Fauconnier, R. Guerraoui, and A. Tielmann. The disagreement power of an adversary. *Distributed Computing*, 24(3-4):137–147, 2011.

[8] E. Gafni. On the wait-free power of iterated-immediate-snapshots. Unpublished manuscript, online at `http://www.cs.ucla.edu/~eli/eli/wfiis.ps`, 1998.

[9] E. Gafni. Round-by-round fault detectors (extended abstract): Unifying synchrony and asynchrony. In *PODC*, 1998.

[10] E. Gafni. The 0-1-exclusion families of tasks. In *OPODIS*, pages 246–258, 2008.

[11] E. Gafni. Free-for-all execution: Unifying resiliency, set-consensus, and concurrency. Unpublished manuscript, online at `http://www.cs.ucla.edu/~eli/eli/concurrency25.pdf`, 2008.

[12] E. Gafni and E. Koutsoupias. Three-processor tasks are undecidable. *SIAM J. Comput.*, 28(3):970–983, 1999.

[13] E. Gafni and S. Rajsbaum. Distributed programming with tasks. In *OPODIS*, pages 205–218, 2010.

[14] E. Gafni, S. Rajsbaum, and M. Herlihy. Subconsensus tasks: Renaming is weaker than set agreement. In *DISC*, pages 329–338, 2006.

[15] V. Hadzilacos and S. Toueg. A modular approach to fault-tolerant broadcasts and related problems. Technical Report TR 94-1425, Department of Computer Science, Cornell University, May 1994.

[16] M. Herlihy, D. N. Kozlov, and S. Rajsbaum. *Distributed Computing Through Combinatorial Topology.* Morgan Kaufmann, 2014.

[17] M. Herlihy and S. Rajsbaum. The decidability of distributed decision tasks (extended abstract). In *STOC*, pages 589–598, 1997.

[18] M. Herlihy and S. Rajsbaum. Algebraic spans. *Mathematical Structures in Computer Science*, 10(4):549–573, 2000.

[19] M. Herlihy and S. Rajsbaum. Concurrent computing and shellable complexes. In *DISC*, pages 109–123, 2010.

[20] M. Herlihy and S. Rajsbaum. The topology of distributed adversaries. *Distributed Computing*, 26(3):173–192, 2013.

[21] M. Herlihy and N. Shavit. The asynchronous computability theorem for t-resilient tasks. In *STOC*, pages 111–120, May 1993.

[22] M. Herlihy and N. Shavit. The topological structure of asynchronous computability. *J. ACM*, 46(2):858–923, 1999.

[23] D. N. Kozlov. Chromatic subdivision of a simplicial complex. *Homology, Homotopy and Applications*, 14(1):1–13, 2012.

[24] P. Kuznetsov. Understanding non-uniform failure models. *Bulletin of the EATCS*, 106:53–77, 2012.

[25] N. Linial. Doing the IIS. Unpublished manuscript, 2010.

[26] R. Lubitch and S. Moran. Closed schedulers: A novel technique for analyzing asynchronous protocols. *Distributed Computing*, 8(4):203–210, 1995.

[27] Y. Moses and S. Rajsbaum. A layered analysis of consensus. *SIAM J. Comput.*, 31(4):989–1021, 2002.

[28] J. R. Munkres. *Elements of algebraic topology.* Addison-Wesley Publishing Company, Menlo Park, CA, 1984.

[29] S. Rajsbaum, M. Raynal, and C. Travers. The iterated restricted immediate snapshot model. In *COCOON*, pages 487–497, 2008.

[30] M. Raynal and J. Stainer. Increasing the power of the iterated immediate snapshot model with failure detectors. In *SIROCCO*, pages 231–242, 2012.

[31] H. L. Royden. *Real analysis.* Macmillan Publishing Company, New York, third edition, 1988.

[32] M. Saks and F. Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. *SIAM J. on Computing*, 29:1449–1483, 2000.

[33] E. H. Spanier. *Algebraic topology.* McGraw-Hill Book Co., New York, 1966.

[34] J. H. C. Whitehead. Simplicial Spaces, Nuclei and m-Groups. *Proc. London Math. Soc.*, S2-45(1):243, 1939.