Topological Characterization of Task Solvability in General Models of Computation

Hagit Attiya
Department of Computer Science, Technion, Haifa, Israel

Armando Castañeda
Instituto de Matemáticas, Universidad Nacional Autónoma de México, Mexico

Thomas Nowak
Laboratoire Méthodes Formelles, Université Paris-Saclay, CNRS, ENS Paris-Saclay, France
Institut Universitaire de France, Paris, France

Abstract

The famous *asynchronous computability theorem* (ACT) relates the existence of an asynchronous wait-free shared memory protocol for solving a task with the existence of a simplicial map from a subdivision of the simplicial complex representing the inputs to the simplicial complex representing the allowable outputs. The original theorem relies on a correspondence between protocols and simplicial maps in round-structured models of computation that induce a compact topology. This correspondence, however, is far from obvious for computation models that induce a non-compact topology, and indeed previous attempts to extend the ACT have failed.

This paper shows that in *every* non-compact model, protocols solving tasks correspond to simplicial maps that need to be *continuous*. It first proves a *generalized* ACT for sub-IIS models, some of which are non-compact, and applies it to the *set agreement* task. Then it proves that in general models too, protocols are simplicial maps that need to be continuous, hence showing that the topological approach is *universal*. Finally, it shows that the approach used in ACT that equates protocols and simplicial complexes actually works for *every* compact model.

Our study combines, for the first time, combinatorial and point-set topological aspects of the executions admitted by the computation model.

2012 ACM Subject Classification Theory of computation → Distributed computing models

Keywords and phrases task solvability, combinatorial topology, point-set topology

Digital Object Identifier 10.4230/LIPIcs.DISC.2023.5

Related Version *Full Version*: `https://arxiv.org/abs/2301.13837`

Funding *Hagit Attiya*: partially supported by the Israel Science Foundation (grants 380/18 and 22/1425).
Armando Castañeda: partially supported by the DGAPA PAPIIT project IN108723.
Thomas Nowak: partially supported by the ANR grant ANR-21-CE48-0003.

1 Introduction

The celebrated topological approach in distributed computing relates task solvability to the topology of inputs and outputs of the task and the topology of the protocols allowed in a particular model of computation. This approach rests on three pillars. First, *configurations*, whether of inputs, outputs or protocol states, can be modeled as *simplexes*, which are finite sets. Second, the inherent *indistinguishability* of configurations is crisply captured by intersections between simplexes. Third, a *carrier map* captures the notion of the set of configurations that are reachable from a given configuration.

37th International Symposium on Distributed Computing (DISC 2023).
Editor: Rotem Oshman; Article No. 5; pp. 5:1–5:21
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

More concretely, in this approach, *tasks* are triples $T = (\mathcal{I}, \mathcal{O}, \Delta)$, where $\mathcal{I}$ and $\mathcal{O}$ are *simplicial complexes* modeling the inputs and outputs of the task, and Δ is a *carrier map* specifying the possible valid outputs, $\Delta(\sigma)$, for each input simplex $\sigma \in \mathcal{I}$. Similarly, *protocols* are triples $P = (\mathcal{I}, \mathcal{P}, \Xi)$, where $\mathcal{P}$ is the complex modeling the final configurations of the protocol, and Ξ is a carrier map specifying the reachable final configurations, $\Xi(\sigma)$, from σ.

With this perspective in mind, it is natural to conclude that a protocol maps final states (*i.e.*, states in final configurations) to outputs, and for the protocol to be correct, the mapping must be *simplicial*; that is, all outputs of final states in the same simplex $\tau \in \Xi(\sigma)$ (*i.e.*, in the same final configuration) must be in the same output simplex of $\Delta(\sigma)$. Thus, a protocol induces a simplicial map from $\mathcal{P}$ to $\mathcal{O}$. Moreover, since decisions of processes are only based on local information, it is natural to conclude the converse, *i.e.*, *any* simplicial map implies a protocol. (In general, a protocol specifies also the communication during an execution. However, solvability can consider only existence of a decision function, by assuming that the protocol is *full-information.*) This leads to the following purely topological solvability characterization: a protocol $P = (\mathcal{I}, \mathcal{P}, \Xi)$ solves a task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ *if and only if* there is a simplicial map $\delta : \mathcal{P} \to \mathcal{O}$ such that for every $\sigma \in \mathcal{I}$, we have $(\delta \circ \Xi)(\sigma) \subseteq \Delta(\sigma)$.

The discussion so far did not depend on a particular model of computation. Indeed, this approach seems universal and gives the impression that protocols and simplicial maps are the same, and that for all models, the solvability question can be reduced to the existence of a simplicial map. In fact, the correspondence between protocols and simplical maps seems so self-evident that frequently the characterization above seems to require no proof, and is introduced as a definition (*e.g.*, [13, Section 4.2.2; Definition 8.4.2]).

This approach works well for cases where the model of computation has a particular round structure[1] and it induces a *compact* topology so that the correspondence between protocols and simplicial maps holds. Roughly speaking, a compact topological space has no "punctures" or "missing endpoints", namely, it does not exclude any limit point. If a model of computation is specified as a set of infinite executions, then a compact model will contain all its "limit executions". For example, the *Iterated Immediate Snapshot* (IIS) model ensures that computation proceeds in sequence of (implicit) rounds; in each round, any of a *finite* set of possible schedules can happen. Thus, the model contains every infinite execution with this round structure. Models like IIS are sometimes called *oblivious* [7], and are known to induce *finite complexes with compact topology, where protocols and simplicial maps are the same.* Round-structured compact models have been extensively studied in the literature, and different techniques have been developed for them (*e.g.*, [4, 6, 10, 14]).

However, this approach is not true in all models, specifically, in non-compact ones. In a non-compact model, typically some "good" schedules only *eventually* happen, which then implies that the model is not limit-closed. Examples of a non-compact models are t-resilient *asynchronous* models where any process is guaranteed to eventually obtain information from at least $t-1$ other processes infinitely often, but the process can take an unbounded number of steps before that happens. This means, for example, that the model contains every infinite execution where a process runs solo for a *finite* number of steps and then obtains information from $t-1$ other processes, but it does not contain the infinite solo execution of the process, *i.e.*, the limit execution.

Challenging non-compact models have been mostly treated in the literature indirectly, through "compactification". Sometimes, compactification consists of considering only protocols with a concrete round structure, as is done in some chapters of [13]. In other cases, a

[1] Rounds may be explicit, like in the synchronous message-passing t-resilient model, or implicit, for example, modeled by *layers*, as in the Iterated Immediate Snapshot model.

computationally-equivalent round-structured compact model is analyzed instead (*e.g.*, [17,22]). This requires first to prove that the models are equivalent, through simulations in both directions, and then characterize solvability in the compact model. For impossibility results, it is also sometimes possible to identify a compact sub-model in which a problem of interest is still unsolvable [18].

Round-structured compact models have also served to analyze other compact models. In the famous *asynchronous computability theorem* (ACT) [15], fully characterizing solvability of the non-round-structured compact read/write wait-free shared memory model, a crucial step is showing equivalence with the IIS model. This restricts the solvability characterization to subdivisions, which are well-behaved topological spaces, making the ACT highly useful.

Some sub-IIS models, with subsets of IIS like t-resilient computations, are non-compact. For this reason, an attempt [11] to generalize the ACT to arbitrary sub-IIS models and tasks had to directly address non-compact models. The idea of this so-called *generalized* ACT is to somehow reuse the nice structure of IIS, modeling any sub-IIS model as a possibly infinite subdivision.

This raises two important questions that have not been explicitly investigated so far, which are addressed in this paper. (1) Can protocols in all models of distributed computation be captured as simplicial maps? (2) Can the topological approach be applied to all models of computation? The answers to these questions are not self-evident since there may be non-compact models or non-round-structured compact models, which cannot be compactified.

We note that there is already a recent negative answer to the first question: Godard and Perdereau [12] showed a non-compact model where consensus is unsolvable, but nevertheless, it has a simplicial map as described above. Roughly, it considers a sub-IIS model where a single infinite execution of IIS is removed. The resulting model is not compact. It turns out that the complex of a protocol is an *infinite* subdivision that is *disconnected*, hence, there is a simplicial map from it to the consensus output complex (which is disconnected too). But the map does not imply a consensus protocol, since intuitively, the decisions must be consistent as they approach the discontinuity of the removed execution. More specifically, the simplicial map does not imply a protocol because it is *not continuous*. Section 3 details the example based on their ideas. While continuity of simplicial maps is guaranteed in compact models, this is not the case in non-compact ones. This example demonstrates that the generalization in [11] is flawed as it misses the continuity property of simplicial maps. Godard and Perdereau also correct this problem *for the special case of two processes and the consensus task*.

Continuity of simplicial maps may seem trivial, but it was overlooked for long time, before [12]. Here, we further expose its importance.

We first study task solvability in the well-structured simplicial complexes induced by sub-IIS models. Our first contribution (Theorem 4.1) is to present a correct generalized ACT for any number of processes and arbitrary tasks. Our approach is motivated by the critical role of continuity. Our second contribution (Theorem 4.2) is to use our generalized ACT theorem in order to provide an impossibility condition for set agreement in sub-IIS models, where the continuity requirement of simplicial maps allows a natural generalization of the known impossibility conditions for round-structured compact models.

While this settles the questions for sub-IIS models, the questions for general models remain open. Our third contribution (Theorem 5.4) shows that the topological approach is applicable in *all* models of computation, if one requires simplicial maps to be continuous. Unlike the case of round-structured compact and sub-IIS models, proving the applicability of the topology approach to general non-compact models is not straightforward. It requires to combine *point-set topological* techniques [2] with *combinatorial topology* techniques [13].

We use this result in our fourth contribution: a proof that the approach described at the beginning of the introduction, equating protocols and simplicial maps that are not required to be continuous, is universal for compact models (Theorem 6.3). Namely, in every compact model, possibly non-round-structured, it is indeed the case that there is correspondence between protocols and simplicial maps, hence the approach works in all these cases. The proof of this result is far from trivial, and it uses *projective limits* from *category theory* [19].

As far as we know, non-compact models have been directly studied only in [8, 9, 11, 12, 20]. A full combinatorial solvability characterization for two-process consensus under synchronous general message-loss failures appears in [9]. For the case of two processes, these models are all sub-IIS, hence this work is the first that directly studies non-compact models. Then, [11] attempted to generalize ACT to general sub-IIS models and tasks, for any number of processes. The solvability of two-process consensus is studied again in [12], now from a combinatorial topology perspective, where it is shown that the attempt in [11] is flawed. That paper also provides an alternative full topological solvability characterization for two-process consensus. Recently, sub-IIS models were studied through *geometrization* [8], *i.e.*, using a mapping from IIS executions to points in the Euclidean space, which in turn induces a topology. The geometrization is used to derive a full solvability characterization for *set agreement* in sub-IIS models, and it generalizes the two-process consensus solvability characterization of [12]. A solvability characterization for consensus (only) in general models, for any number of processes, is presented in [20]. It is derived using point-set topology techniques from [2], without combining them with combinatorial topology. Recent formalizations [1, 3] for proofs based on valency arguments show that for some tasks, *e.g.*, set agreement and renaming, impossibility cannot be shown by inductively constructing infinite executions. This means that arguments regarding the final protocol states are necessary in order to prove impossibility. Our results indicate that such proofs can be carried within combinatorial topology, in general models of computation.

In summary, our contributions are:

1. A generalized ACT for arbitrary sub-IIS models (Theorem 4.1).
2. An application of the generalized ACT to set agreement (Theorem 4.2).
3. A proof that if simplicial maps from $\mathcal{P}$ to $\mathcal{O}$ are required to be continuous, the topological approach works for every model of computation (Theorem 5.4).
4. A proof that the usual topological approach where simplicial maps are not required to be continuous works for every compact model (Theorem 6.3).

2 Preliminaries

This section presents the elements of combinatorial topology and point set topology used in further sections, and defines tasks, system models and task solvability.

We start by fixing some basic notation. We denote by Π the set of processes and let $n = |\Pi|$. For any function $f : X \to Y$ and subsets $A \subseteq X$ and $B \subseteq Y$, we denote by $f[A]$ the image of the set A under f and by $f^{-1}[B]$ the inverse image of the set B under f.

2.1 Elements of Combinatorial Topology and Decision Tasks

To be the most general possible, we use the language of colored tasks [13, Definition 8.2.1], to study one-shot distributed decision tasks like consensus or set agreement. We use the standard concepts in [13] with the only difference that simplicial complexes might be infinite, *i.e.*, a possibly infinite sets of finite sets. Definitions of concepts like simplicial and carrier maps, geometric realization, standard chromatic subdivision and more appear in the Appendix. Here we just recall the definition of tasks.

A *decision task* is a triple $T = (\mathcal{I}, \mathcal{O}, \Delta)$ such that:

- $\mathcal{I}$, the *input complex*, is a *finite* pure chromatic simplical complex of dimension $n-1$, whose vertices are additionally labeled by a set of inputs V^{in}. Each simplex of $\mathcal{I}$ specifies private inputs for the processes that appear in the simplex.
- $\mathcal{O}$, the *output complex*, is a *finite* pure chromatic simplical complex of dimension $n-1$, whose vertices are additionally labeled by a set of inputs V^{out}. As above, each simplex of $\mathcal{O}$ specifies private outputs for the processes in the simplex.
- Δ is a chromatic carrier map from $\mathcal{I}$ to $\mathcal{O}$, $\Delta(\sigma)$, that *specifies* the valid outputs for every input simplex σ in $\mathcal{I}$. Namely, when the inputs are the ones specified in σ, the outputs in any simplex of $\Delta(\sigma)$ are allowed.

Whenever the complex is understood from the context, we will denote by $v(p, x)$ the unique vertex of the complex with color $p \in \Pi$ and label x.

2.2 Elements of Point-Set Topology

In addition to combinatorial topology, we employ point-set topology [5], *i.e.*, the general mathematical theory of closeness, convergence, and continuity. The topologies that we define here are described by *metrics*, which are distance functions $d : X \times X \to [0, \infty)$ that satisfy:

1. Positive definiteness: $d(x, y) = 0$ if and only if $x = y$
2. Symmetry: $d(x, y) = d(y, x)$
3. Triangle inequality: $d(x, z) \le d(x, y) + d(y, z)$

A set equipped with a metric is called a *metric space*. The most basic metric is the *discrete metric*, which is defined by:

$$d(x,y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$$

That is, the discrete metric can only give the information whether two elements are equal, but implies no finer-grained notion of closeness.

A central notion in point-set topology are *open sets*, which are subsets $O \subseteq X$ such that

$$\forall x \in O \quad \exists \varepsilon > 0 \colon \quad B_\varepsilon(x) \subseteq O$$

where $B_\varepsilon(x) = \{y \in X \mid d(x, y) < \varepsilon\}$ is the open ball with radius ε around x. With respect to the discrete metric, every subset $O \subseteq X$ is open. This follows from the fact that the open ball with radius $1/2$ around x is equal to $B_{1/2}(x) = \{x\}$, *i.e.*, only contains x itself.

The general definition of a *topological space* is a nonempty set X together with a *topology*, *i.e.*, a set $\mathcal{O} \subseteq 2^X$ of subsets of X that is closed under arbitrary unions and finite intersections. The elements of $\mathcal{O}$ are called the open sets of the space. With the above definition, every metric induces a topology.

A particular class of metrics that we use in this paper is that of *ultrametrics*. They satisfy the stronger ultrametric triangle inequality: $d(x, z) \le \max\{d(x, y), d(y, z)\}$ for all $x, y, z \in X$. The discrete metric is an example of an ultrametric. In an ultrametric space, two open balls are either disjoint or one is a subset of the other, as is shown by the following folklore lemma:

▶ **Lemma 2.1.** *Let X be an ultrametric space. For all $x, y \in X$ and all $\delta, \varepsilon > 0$, one of the following is true: (1) $B_\delta(x) \cap B_\varepsilon(y) = \emptyset$, (2) $B_\delta(x) \subseteq B_\varepsilon(y)$, (3) $B_\varepsilon(y) \subseteq B_\delta(x)$.*

The *morphisms* of topological spaces X and Y are *continuous* functions, namely, those functions $f : X \to Y$ such that any inverse image of an open set is open. In metric terms, this means that for every $x \in X$ and every $\varepsilon > 0$ there exists a $\delta > 0$ such that $d_X(x, x') < \delta$

implies $d_Y(f(x), f(x')) < \varepsilon$ for all $x' \in X$. Here, we denoted by d_X the metric on X and by d_Y the metric on Y. All constant functions are continuous, as are all *locally constant* functions, *i.e.*, functions $f : X \to Y$ that are constant in some open ball $B_\varepsilon(x)$ with positive radius $\varepsilon > 0$ for every $x \in X$.

Topologies for standard set-theoretic constructions can be defined from their individual parts. For instance, the product topology of a countable collection of metric spaces X_i can be described by the metric $d : X \times X \to [0, \infty)$ with

$$d(x,y) = \sum_{i\in\mathbb{N}} 2^{-i} \frac{d_i(x_i, y_i)}{1 + d_i(x_i, y_i)} \quad .$$

We use the product metric to extend the notion of indistinguishability of local views of configurations (these concepts are formally defined in Section 2.3) to a metric on infinite executions. It has the following property:

▶ **Lemma 2.2** ([5, § 2.3, Proposition 4]). *Let $(X_i)_{i\in\mathbb{N}}$ be a countable collection of metric spaces and let $X = \prod_{i\in\mathbb{N}} X_i$ be their product equipped with the product metric. For all metric spaces Y and all functions $g : Y \to X$, the following are equivalent:*

1. *The function g is continuous.*
2. *The function $\pi_i \circ g$ is continuous for all $i \in \mathbb{N}$ where $\pi_i : X \to X_i$ is the projection on the component i.*

The disjoint-union topology of the disjoint union $X = \bigsqcup_{i\in I} X_i$ is described by the metric $d : X \times X \to [0, \infty)$ with $d(x, y) = d_i(x, y)$ if there is an index $i \in I$ such that both x and y are elements of X_i, and $d(x, y) = 2$ else. We use the disjoint-union metric to get a global metric from those defined for the local views of each process, with the following property:

▶ **Lemma 2.3** ([5, § 2.4, Proposition 6]). *Let $(X_i)_{i\in I}$ be a collection of metric spaces and let $X = \bigsqcup_{i\in I} X_i$ be their disjoint union equipped with the disjoint-union metric. For all metric spaces Y and all functions $g : X \to Y$, the following are equivalent:*

1. *The function g is continuous.*
2. *The function $g \circ \varphi_i$ is continuous for all $i \in I$ where $\varphi_i : X_i \to X$ is the embedding of X_i into X.*

2.3 System Model

Let $T = (\mathcal{I}, \mathcal{O}, \Delta)$ any task. Since our goal is to give a very general characterization of task solvability, we work with an abstract system model that hides most of the operational details, such as semantics of shared registers or guarantees of message delivery. We instead focus on the structure of the set of executions induced by the local *indistinguishability* relations, *i.e.*, by the processes' local views. We further assume that actions taken by processes do not influence the set of possible executions. That is, we assume the existence of *full-information* executions, on which we base our characterization. A full-information execution is a sequence of configurations. A configuration is a vector with the process states and the state of the environment (*e.g.*, shared memory, messages in transit) in its entries. In a full-information execution, every process relays all the information it gathered to all other processes whenever it can. This includes its input value, the order and contents of events it perceived, and the information relayed to it by others. In particular, we assume that there are no size constraints on messages or shared memory.

Formally, let Exec be the set of full-information executions of n processes in which initial configurations are chosen according to the input complex $\mathcal{I}$. We assume the existence of projection functions $\pi_p : \mathsf{Exec} \to \mathsf{View}_p$ from executions to sequences of local views of

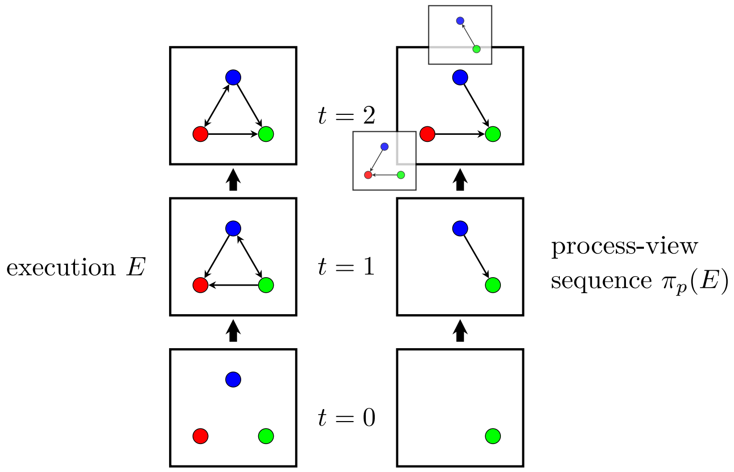

Figure 1 Prefix of a full-information execution E (left) and process-view projection $\pi_p(E)$ (right) of a synchronous message-passing system with dynamic communication graphs. The depicted prefix includes the initial configuration as well as the first two communication rounds. Process p is the green (lower right) process. Initially, after round 0, process p only knows its own initial value. After the first round, process p also knows the blue (upper) process's initial value as well as the fact that directed edge from the blue to the green process was present in the communication graph of the first round. After the second round, process p learned the initial value of the red (lower left) process, its own incoming edges of the second round's communication graph, as well as the views of the blue and the red processes after the first round.

process p. These sequences can be finite or infinite. Its element with index t contains the local view of process p right after its t^{th} step in the execution. The set of process views of executions in which process p is correct will be denoted by CView_p.

A step is defined as a possibility to irrevocably decide. That is, the t^{th} step of process p is process p's t^{th} possibility to decide a value (or not) in the execution. We allow processes to decide in their initial state, *i.e.*, in their step with index $t = 0$. Step counts are local to a process and need not be synchronized among processes. A process that only has finitely many steps is called *faulty* in the execution. For an execution $E \in \mathsf{Exec}$ we write $\mathsf{Correct}(E) \subseteq \Pi$ for the set of correct (non-faulty) processes in the execution. A *participating* process $p \in \mathsf{Part}(E) \subseteq \Pi$ is one that takes at least one step. We have that $\mathsf{Correct}(E) \subseteq \mathsf{Part}(E)$. We write $\mathsf{Init}_p(E) \in V^{\text{in}}$ for the initial value of process p in execution E. The concrete forms of executions and local views depend on the specifics of the computational model. Fig. 1 depicts an example execution and process-view sequence.

A (decision) *protocol* is a function from local views to $V^{\text{out}} \cup \{\perp\}$ with $\perp \notin V^{\text{out}}$ such that decisions are irrevocable: if some view is mapped to a decision value $v \in V^{\text{out}}$, then all its successor views are also mapped to v. A process p thus has at most one decision value in every execution E, which we denote by $\mathsf{Decision}_p(E) \in V^{\text{out}}$. A protocol *solves* a task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ if it satisfies the following two conditions in every execution $E \in \mathsf{Exec}$:

- Every correct process $p \in \mathsf{Correct}(E)$ has a decision value in E.
- We have $\{v(p, \mathsf{Decision}_p(E)) \mid p \in \mathsf{Correct}(E)\} \in \Delta(\sigma)$ where $\sigma = \{v(p, \mathsf{Init}_p(E)) \mid p \in \mathsf{Part}(E)\}$.

Example: Lossy-Link Model. The lossy-link model [21] is a synchronous computation model with $n = 2$ processes, p and q, that communicate via message passing. The communication graph can change from round to round. In each round, the adversary chooses one of three

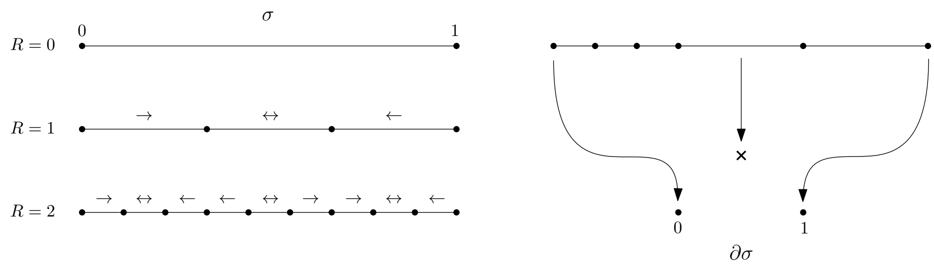

Figure 2 Subdivisions in the IIS model for two processes.

communication graphs: $\leftarrow$, $\rightarrow$, or $\leftrightarrow$. In a round with communication graph $\leftarrow$, only the message from the right to the left processes arrives, the other message is lost. In a round with communication graph $\rightarrow$, only the message from the left to the right processes arrives, the other message is lost. In a round with communication graph $\leftrightarrow$, both messages arrive and no message is lost. In a full-information execution, each process starts out by sending its initial value and then records all received messages in subsequent rounds, relaying this information to the other process. In this model, there is no notion of faulty processes; the only source of uncertainty is the communication. We thus have $\mathsf{Part}(E) = \mathsf{Correct}(E) = \{p, q\} = \Pi$ for every full-information execution E, and thus $\mathsf{CView}_p = \mathsf{View}_p$ and $\mathsf{CView}_q = \mathsf{View}_q$.

Since both processes are correct in every execution, both processes are participating, *i.e.*, $\mathsf{Part}(E) = \Pi$.

3 The Need of Continuity

This section explains the need of continuity of simplicial maps to model protocols in non-compact models. This is done using in part the example in [12] showing a flaw in the attempt to generalize the ACT [11].

For a system with two processes, left and right, the compact IIS model can be equivalently defined as the lossy-link model described in Section 2.3. Thus, IIS for two processes consists of all infinite sequences of communication graphs $\leftarrow$, $\rightarrow$, or $\leftrightarrow$, each graph specifying the communication that occurs in a round. A sub-IIS model is any subset of IIS.

Let us consider an *inputless* version of the consensus task where the left process has fixed input 0 and the right process has fixed input 1. Then, the input complex of the task $\mathcal{I}$ is the complex made of the edge $\sigma = \{0, 1\}$ and its faces (processes are identified with their inputs), the output complex $\mathcal{O}$ has simplexes $\{0\}$ and $\{1\}$, and Δ maps σ to $\mathcal{O}$, and each $\{i\}$ to itself. Complexes $\mathcal{I}$ and $\mathcal{O}$ will be denoted σ and $\partial\sigma$, respectively.

The topology of the IIS executions is well understood: the complex modeling all configurations at the end of round R is a *finite* subdivision of the input complex σ (basically a subdivision of the real interval $[0, 1]$), and as R increases, the subdivision gets finer. Concretely, it is the R-th standard chromatic subdivision. Figure 2(left) shows the subdivisions for the first two rounds, where, for example, the left-most and right-most edges of the second subdivision correspond to the configurations at the end of the finite solo executions $\rightarrow, \rightarrow$ and $\leftarrow, \leftarrow$, respectively, where a process does not hear from the other, and the central edge corresponds to $\leftrightarrow, \leftrightarrow$, where processes hear from each other.

A key property of round-structured compact models like IIS is that, for any protocol solving a task, there is a *finite* round R such that all correct processes make a decision at round R, at the latest (assuming $\mathcal{I}$ is finite). With this property, it is simple to see that consensus is impossible in IIS (see the right side of Figure 2):

Figure 3 A possible subdivision for the sub-IIS model M_1.

1. For any round R, the complex corresponding to the *decided states* of a hypothetical protocol P, is a finite subdivision $\mathcal{K}$ of σ, *i.e.*, $|\mathcal{K}| = |\sigma|$. (Recall that $|\mathcal{K}|$ is the geometric realization of $\mathcal{K}$.) The subdivision might be irregular because processes might make decisions at different rounds; processes keep running after decision, hence an edge models infinitely many infinite executions, all of them sharing the finite prefix where the decisions are made.
2. P must map each vertex (state) of $\mathcal{K}$ to an output in $\partial\sigma$, with the restriction that the left-most vertex must be mapped to 0 and the right-most vertex must be mapped to 1, as they correspond to solo executions, hence, by validity of the consensus task (*i.e.*, $\Delta(\{i\}) = \{i\}$), the process that only sees its input is forced to decide it.
3. Since P solves consensus, it induces a simplicial map $\delta : \mathcal{K} \to \partial\sigma$, which, as $\mathcal{K}$ is finite, *necessarily* induces a continuous map $|\delta| : |\mathcal{K}| \to |\partial\sigma|$. The map $|\delta|$ is ultimately a continuous map $|\sigma| \to |\partial\sigma|$ that maps the boundary of σ to itself.
4. Finally, this continuous map does not exist because $|\sigma|$ is solid whereas $|\partial\sigma|$ is disconnected.

The argument above goes from protocols to simplicial maps. In models like IIS, the other direction is also true. Namely, for any given task $T = (\mathcal{I}, \mathcal{O}, \Delta)$, for any complex $\mathcal{K}$ related to $\mathcal{I}$ that satisfies some model-dependent properties, any simplicial map from $\mathcal{K}$ to $\mathcal{O}$ that agrees with Δ, induces a protocol for T. Thus, to show that a task is solvable in two-process IIS, it suffices to exhibit a finite, possibly irregular, subdivision of the input complex, in the style of the one in Figure 2(left), and a simplicial map that is valid for the task.

The main aim of [11] is to generalize the approach above that equates simplicial maps and protocols to arbitrary sub-IIS models, in order to exploit the already known topology of IIS. The high-level idea is that the complexes that model a sub-IIS model are still subdivisions but not necessarily of the input complex, and not necessarily finite.

Let us consider first the sub-IIS model M_1 with all infinite executions of the form $\leftarrow$ followed by any infinite sequence with $\leftarrow$, $\rightarrow$, or $\leftrightarrow$ (intuitively right goes first), or $\rightarrow$ followed by any infinite sequence with $\leftarrow$, $\rightarrow$, or $\leftrightarrow$ (intuitively left goes first). It can be seen that consensus is solvable in this model: since $\leftrightarrow$ cannot happen in the first round, the process that receives no message in the first round is the "winner". Figure 3 shows an irregular subdivision that models all executions of M_1; for example, the right-most edge corresponds to all executions of M_1 with prefix $\leftarrow, \leftarrow$. Intuitively, in the subdivision, in some executions processes decide in round one (represented by the edge at the left), and in the remaining executions processes decide in round two (represented by the three edges at the right). Clearly, there is a simplicial map from such a disconnected subdivision to $\partial\sigma$ that agrees with consensus. This simplicial map induces a consensus protocol for M_1.

The argument above works well because the model is compact, hence finite subdivisions are able to capture all its executions. However, in non-compact models, some executions can only be modeled through infinite subdivisions, which implies that simplicial maps are not necessarily protocols.

Consider now the sub-IIS model M_2 obtained by removing from IIS the infinite execution E described by the sequence $\leftrightarrow, \leftarrow, \leftarrow, \ldots$ This model is not compact because it contains any infinite execution with a *finite* prefix (of any length) of E, but it does not contain E itself, the limit execution. As said, a crucial property of non-compactness is that the executions of

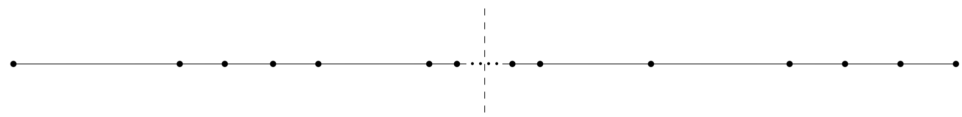

Figure 4 An schematic representation of an infinite subdivision for the sub-IIS model M_2.

the model cannot be captured by a finite subdivision. Intuitively, an edge can only model executions that have a common finite prefix of E of length x, but in M_2 there are executions with a prefix larger than x, hence these executions are not captured by the edge; if the subdivision is finite, there are necessarily executions that are not modeled by any edge.

Figure 4 contains a schematized infinite subdivision $\mathcal{K}$ that indeed captures all executions of M_2. Intuitively, there are infinitely many edges that get closer and closer to the point that represents the removed execution E (depicted as a vertical dashed line at the center), but no edge actually "crosses" it (as E is not in M_2). Thus the simplicial complex $\mathcal{K}$ is disconnected, and there is a simplical map from $\mathcal{K}$ to $\partial\sigma$ that agrees with consensus. Although all executions are captured in the infinite subdivision, such a simplicial map *does not* imply a protocol. The intuition is that there is a sudden jump in the decisions around E, which ultimately implies that the decision in executions that are similar enough to the removed limit execution E are not consistent, namely, they cannot be produced by a protocol.

It turns out that the topological space $|\mathcal{K}|$ is actually a subdivision of $|\sigma|$: in the limit, $|\mathcal{K}| = |\sigma|$. Thus, the infinite subdivision $\mathcal{K}$ describes a space that is not disconnected! Moreover, for any infinite subdivision that models M_2, the space associated with it is connected, *i.e.*, this is an invariant of the model of computation. Any simplicial map that intends to capture a protocol should consider that $|\mathcal{K}| = |\sigma|$. This is precisely captured by demanding that the induced map $|\mathcal{K}| \to |\mathcal{O}|$ must be continuous (hence smooth around E). Therefore, there is no continuous map $|\sigma| \to |\partial\sigma|$ that maps the boundary of σ to itself, and indeed consensus is not solvable in this model [9, Theorem III.8].

Formalizing this seemingly simple observation in arbitrary models of computation is not obvious, and it requires a combination of combinatorial topology techniques and point-set topology techniques, as is done in the following sections. Intuitively, distance functions in point-set topology are used to equip protocol complexes with a topology that in turn yields a correspondence between continuous simplicial maps and protocols.

4 Proof of the Generalized Asynchronous Computability Theorem with an Application to Set Agreement

In this section we use the definitions and notation of Gafni, Kuznetsov, and Manolescu [11] for sub-IIS models. They introduced the notion of *terminating subdivisions* of the input complex $\mathcal{I}$ of a task. The idea is to repeatedly subdivide all simplexes via the standard chromatic subdivision, except those that are already marked as terminated. The terminated simplexes model configurations where processes have decided.

Formally, a terminating subdivision $\mathscr{T}$ is specified by a sequence of chromatic complexes $\mathcal{I}_0, \mathcal{I}_1, \ldots$ and a sequence of subcomplexes $\Sigma_0 \subseteq \Sigma_1 \subseteq \ldots$ such that for all $k \geq 0$: (1) Σ_k is a subcomplex of $\mathcal{I}_k$ (each $\mathcal{I}_k$ is a *non-uniform* subdivision [16]) and (2) $\mathcal{I}_0 = \mathcal{I}$ and $\mathcal{I}_{k+1}$ is obtained from $\mathcal{I}_k$ by the *partial* chromatic subdivision in which the simplexes in Σ_k are not further subdivided (the terminated simplexes), and each simplex $\tau \notin \Sigma_k$ is replaced with its standard chromatic subdivision $\mathrm{Chr}\,\tau$. Precisely, we replace a simplex σ in $\mathcal{I}_k$ by a

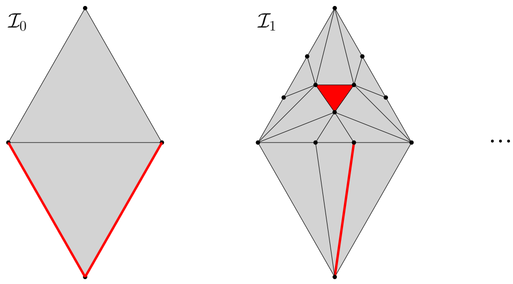

Figure 5 First two complexes of a three-process terminating subdivision.

coarser subdivision than $\operatorname{Chr}\sigma$. Whereas the vertices of $\operatorname{Chr}\sigma$ are pairs (p, σ') with $p \in \Pi$ and $\sigma' \subseteq \sigma$, in $\mathcal{I}_{k+1}$ we consider the pairs (p, σ') of that form such that either $\sigma' \notin \Sigma_k$, or σ' consists of a single vertex in Σ_k.

Figure 5 schematizes a terminating subdivision where $\mathcal{I}$ is made of two triangles and terminated simplexes are marked in red.

A simplex of Σ_k, for some k, is called *stable*. The simplicial complex $K(\mathscr{T})$ is the union of all Σ_k; $K(\mathscr{T})$ might be infinite.

The vertices of $K(\mathscr{T})$ are naturally embedded in the geometric realization of $\mathcal{I}$ by their definition as a vertex of the repeated chromatic subdivision $\operatorname{Chr}^k \mathcal{I}$ (recall that $|\operatorname{Chr}^k \mathcal{I}| = |\mathcal{I}|$, for every $k \geq 0$). In particular, we identify the geometric realization $|K(\mathscr{T})|$ with a subset of $|\mathcal{I}|$. Every IIS execution can be described as an infinite sequence of simplexes $\sigma_0, \sigma_1, \ldots$ such that $\sigma_k \in \operatorname{Chr}^k \mathcal{I}$, for every $k \geq 0$.

A terminating subdivision is *admissible* for a sub-IIS model M if $K(\mathscr{T})$ covers all executions of M, namely, for each execution $\sigma_0, \sigma_1, \ldots$ of M, there is a k such that $|\sigma_k| \subseteq |\tau|$, for some terminated simplex $\tau \in \Sigma_k$.

▶ **Theorem 4.1.** *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 simplicial map $\delta\colon K(\mathscr{T}) \to \mathcal{O}$ 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 $\delta(\tau) \in \Delta(\sigma)$.*

(c) *$|\delta|$ is continuous.*

Proof sketch. ($\Rightarrow$): This direction consisting in showing that the geometric realization of the map δ as constructed in the proof of Gafni, Kuznetsov, and Manolescu [11, Theorem 6.1] is continuous by generalizing the proof given by Godard and Perdereau [12, Theorem 33] for the consensus task with two processes.

($\Leftarrow$): We modify the protocol that is constructed in the proof of Gafni, Kuznetsov, and Manolescu [11, Theorem 6.1] for process p to decide in round k if the set

$$B_k(v) = \{w \in V(K(\mathscr{T})) \mid d(v, w) \leq D_k \wedge \chi(w) = p\}$$

only contains vertices that are mapped to the same output vertex by δ, where v is the view of the process in round k. This condition eventually holds since the subset topology on the geometric realization of the output vertices $V(\mathcal{O})$ is discrete; thus, δ is locally constant. ◀

We now use Theorem 4.1 to derive a condition for the impossibility of $(n-1)$-set agreement task in IIS-sub models. Recall that in this task each process is required to eventually decide an input value (termination) of a process participating in the execution (validity) such that no more than $n-1$ distinct values are decided (agreement).

Let $\Pi = \{p_0, p_1, \dots, p_{n-1}\}$. For simplicity, we focus on the *inputless* version of the set agreement task, where each process p_i, $0 \le i \le n-1$, has fixed input i in every execution, and thus the task is the triple $T = (\mathcal{I}, \mathcal{O}, \Delta)$, where the input complex $\mathcal{I}$ is made of all faces of simplex $\sigma = \{0, 1, \dots, n-1\}$, and for simplicity it is denoted σ, the output complex $\mathcal{O}$, denoted $\partial\sigma$, is the complex with all *proper* faces of σ, and Δ maps every proper face $\sigma' \subset \sigma$ to the complex with all faces of σ', and maps σ to $\partial\sigma$.

▶ **Theorem 4.2.** *Let M be an IIS-sub model such that for any termination subdivision $\mathscr{T}$ of σ that is admissible for M, $|\sigma| = |K(\mathscr{T})|$. Then, $(n-1)$-set agreement is impossible in M.*

5 Characterization of Task Solvability in General Models

In this section we present a topological solvability characterization in general models, hence showing that the topology approach is applicable in all models of computation. As anticipated, the characterization demands simplicial maps to be continuous, which is particularly relevant if complexes are infinite. Differently from sub-IIS models in the previous section, where continuity naturally arises in protocols complexes as they are subdivisions (hence embedded in a Euclidean space), the general case requires to equip protocol complexes with a topology, which is used to capture continuity.

Recall that the set of process views of executions in which process p is correct is denoted CView_p. These are the executions in which we demand process p to decide on an output value. We always have $\mathsf{CView}_p \subseteq \mathsf{View}_p$. For every process p we define a topology on the set CView_p of correct process-p local-view sequences induced by the distance function

$$d_p(\alpha, \beta) = 2^{-T_p(\alpha,\beta)}$$

where $T_p(\alpha, \beta)$ is defined as the smallest index at which the local views in the process-view sequences α and β differ. If no such index exists, then $T_p(\alpha, \beta) = \infty$. This means that the distance between two process-view sequences is smaller the later the process can detect a difference between the two. If α and β do not differ in any index, then $\alpha = \beta$ and $d_p(\alpha, \beta) = 2^{-\infty} = 0$. A variant of this distance function, which considers complete executions instead of local views, was introduced by Alpern and Schneider [2].

We first establish that the distance function d_p is an ultrametric.

▶ **Lemma 5.1.** *The distance function d_p is an ultrametric on CView_p.*

In the next lemma, we establish the fundamental fact that the decision functions for process p are exactly the continuous functions $\mathsf{CView}_p \to V(\mathcal{O})$ when using d_p on CView_p and the discrete metric on $V(\mathcal{O})$.

▶ **Lemma 5.2.** *Let $\delta_p : \mathsf{CView}_p \to V(\mathcal{O})$ be a function. The following are equivalent:*

1. *There is a protocol such that process p decides the value $\delta_p(\alpha)$ in every execution $E \in \mathsf{Exec}$ with local view $\pi_p(E) = \alpha \in \mathsf{CView}_p$.*
2. *The function δ_p is continuous when equipping CView_p with the topology induced by d_p and $V(\mathcal{O})$ with the discrete topology.*

To formulate and prove our characterization for the solvability of tasks in general models, we define a structure that combines the notions of chromatic simplicial complexes and the notion of point-set topology of sequences of local views. Formally, a *topological chromatic simplicial complex* is a chromatic simplicial complex whose set of vertices is equipped with a topology. A vertex map between two topological chromatic simplicial complexes is a *morphism* if it is continuous, chromatic, and simplicial.

The protocol complex $\mathcal{P}$ is a (possibly infinite) topological chromatic simplicial complex defined as follows. The set of vertices of $\mathcal{P}$ is the disjoint union $V(\mathcal{P}) = \bigsqcup_{p\in\Pi} \mathsf{CView}_p$ of the correct local-view spaces. The vertices from CView_p are colored with the process name p. We equip the set of vertices with the disjoint-union topology, *i.e.*, the finest topology that makes all embedding maps $\iota_p : \mathsf{CView}_p \to V(\mathcal{P})$ continuous.

A set σ of vertices of $\mathcal{P}$ is a simplex of $\mathcal{P}$ if and only if the local views are consistent with the views of correct processes in an execution, *i.e.*, if it is of the form

$$\sigma = \{\pi_p(E) \mid p \in P\}$$

for some execution $E \in \mathsf{Exec}$ and some set $P \subseteq \mathsf{Correct}(E)$.

The execution map $\Xi : \mathcal{I} \to 2^{\mathcal{P}}$ is defined by mapping every input simplex in $\mathcal{I}$ to the local views of correct processes of executions in which the initial values of participating processes are as in the input simplex. Formally,

$$\Xi(\sigma) = \big\{\{\pi_p(E) \mid p \in P\} \mid E \in \mathsf{Compatible}(\sigma) \wedge P \subseteq \mathsf{Correct}(E)\big\}$$

where $\mathsf{Compatible}(E) \subseteq \mathsf{Exec}$ denotes the set of executions that are compatible with the initial values described by the input simplex σ. That is,

$$\mathsf{Compatible}(\sigma) = \{E \in \mathsf{Exec} \mid \mathsf{Part}(E) \subseteq \chi[\sigma] \wedge \forall p \in \mathsf{Part}(E)\colon \mathsf{Init}_p(E) = \ell(\pi_p(\sigma))\}$$

where $\pi_p(\sigma)$ denotes the unique vertex of the simplex σ with the color p, if it exists, and $\ell(\pi_p(\sigma))$ is the input (label) of vertex $\pi_p(\sigma)$. The execution map Ξ assigns a subcomplex of $\mathcal{P}$ to every input simplex σ in $\mathcal{I}$. As in the classical finite-time setting [13, Definition 8.4.1], the next lemma shows that it is a carrier map:

▶ **Lemma 5.3.** *The execution map Ξ is a carrier map such that $\mathcal{P} = \bigcup_{\sigma\in\mathcal{I}} \Xi(\sigma)$.*

In contrast to the classical finite-time setting, however, the execution map is not necessarily rigid. Whether it is depends on whether any finite execution prefix can be extended to a fault-free execution. This is not the case, *e.g.*, in many synchronous models. If Ξ is not rigid, then, by definition, it is *a fortiori* not chromatic. It does, however, satisfy the inclusion

$$\big\{\chi(v) \mid v \in V(\Xi(\sigma))\big\} \;\subseteq\; \chi[\sigma]$$

for all input simplices $\sigma \in \mathcal{I}$. In other words, the colors of $\Xi(\sigma)$ are included in the colors of σ; no new process names appear. It turns out that the stronger assumptions of rigidity or chromaticity are not necessary to show our solvability characterization.

▶ **Theorem 5.4.** *The task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ is solvable if and only if there exists a continuous chromatic simplicial map $\delta : \mathcal{P} \to \mathcal{O}$ such that $\delta \circ \Xi$ is carried by Δ.*

Proof. ($\Rightarrow$): Assume that there is a protocol that solves task T. Define the vertex map $\delta : \mathcal{P} \to \mathcal{O}$ by setting $\delta(\alpha)$ to be the vertex of $\mathcal{O}$ with color p and label v where p is the unique process such that $\alpha \in \mathsf{CView}_p$ and v is the decision value of process p in an execution with local-view sequence α when executing the protocol.

The map δ is continuous on each individual CView_p by Lemma 5.2. By Lemma 2.3, it is thus continuous on their disjoint union $\mathcal{P}$. The map δ is chromatic since the color of the vertex $\alpha \in \mathsf{CView}_p$ is p, as is the color of $\delta(\alpha)$.

To prove that δ is simplicial, let φ be a simplex of $\mathcal{P}$. Then, by definition, there exists an execution $E \in \mathsf{Exec}$ and a set $P \subseteq \mathsf{Correct}(E)$ such that $\varphi = \{\pi_p(E) \mid p \in P\}$. Set $\sigma = \{v(p, \mathsf{Init}_p(E)) \mid p \in \Pi\}$ and $\tau = \{v(p, \mathsf{Decision}_p(E) \mid p \in \mathsf{Correct}(E)\}$. Then, since the protocol solves task T, we have $\tau \in \Delta(\sigma)$. By definition of δ, we then have $\delta[\varphi] = \{v(p, \mathsf{Decision}_p(E)) \mid p \in P\} \subseteq \tau \in \Delta(\sigma) \subseteq \mathcal{O}$, which means that $\delta[\varphi] \in \mathcal{O}$ and hence that δ is simplicial.

It remains to prove that $\delta \circ \Xi$ is carried by Δ. So let $\sigma \in \mathcal{I}$ and $\tau \in (\delta \circ \Xi)(\sigma)$. We need to show that $\tau \in \Delta(\sigma)$. By the definitions of Ξ and δ, there exists an execution $E \in \mathsf{Compatible}(\sigma)$ and a set $P \subseteq \mathsf{Correct}(E)$ such that $\tau = \{v(p, \mathsf{Decision}_p(E)) \mid p \in P\}$. Since the protocol solves task T, we have $\tau' = \{v(p, \mathsf{Decision}_p(E)) \mid p \in \mathsf{Correct}(E)\} \in \Delta(\sigma')$ where $\sigma' = \{v(p, \mathsf{Init}_p(E)) \mid p \in \mathsf{Part}(E)\}$. Since $\tau \subseteq \tau'$ and $\Delta(\sigma')$ is a simplicial complex, we deduce that $\tau \in \Delta(\sigma')$. Now, because $E \in \mathsf{Compatible}(\sigma)$, we have $\mathsf{Part}(E) \subseteq \chi[\sigma]$ and $\sigma' \subseteq \sigma$. It thus follows that $\tau \in \Delta(\sigma') \subseteq \Delta(\sigma)$ because Δ is a carrier map.

($\Leftarrow$): The restriction δ_p of δ to the set CView_p is continuous because δ is. By Lemma 5.2 there hence exists a protocol such that every process p decides the value $\ell(\delta(\pi_p(E))) \in V^{\mathrm{out}}$ for every execution E in which p is correct.

Let $E \in \mathsf{Exec}$ be any execution and define the sets $\sigma = \{v(p, \mathsf{Init}_p(E)) \mid p \in \mathsf{Part}(E)\}$ and $\tau = \{v(p, \mathsf{Decision}_p(E)) \mid p \in \mathsf{Correct}(E)\}$. To show that the protocol solves task T, it remains to show that $\tau \in \Delta(\sigma)$. Since $\delta \circ \Xi$ is carried by Δ, it suffices to prove $\tau \in (\delta \circ \Xi)(\sigma)$. Setting $\varphi = \{\pi_p(E) \mid p \in \mathsf{Correct}(E)\}$, we have $\tau = \delta[\varphi]$. We are thus done if we show $\varphi \in \Xi(\sigma)$. But this follows from $E \in \mathsf{Compatible}(\sigma)$, which is true by construction of σ. ◀

6 Relationship to the Classical Finite-Time Approach

In this section, we formalize the relationship between our infinite protocol complex used for the general solvability characterization in Theorem 5.4 and the classically studied finite-time protocol complexes. Besides demonstrating that the classical formalism is a special case of ours, we show the finite-time approach is sufficient for all compact models. More specifically, we show that it is possible to restrict the study to finite-time protocols if the computational model is compact. Formally, a topological space is *compact* if every open cover has a finite subcover. Many computational models that are defined by safety predicates are compact. We use the concept of projective limit from category theory [19] to formalize the relationship between finite-time and infinite-time complexes. In particular, we show that the infinite-time complex is the projective limit of the finite-time complexes if the model is compact.

Finite-Time Complexes. For every nonnegative integer T, we define the time-T protocol complex $\mathcal{P}|_T$ as follows:

- The set of vertices of $\mathcal{P}|_T$ is the disjoint union of the sets $\mathsf{CView}_p|_T$ where p varies in the set Π of processes.
- The set $\mathsf{CView}_p|_T$ is defined as the set of open balls of radius $\varepsilon = 2^{-T}$ in CView_p. These balls are either identical or disjoint by Lemma 2.1.
- All vertices of $\mathsf{CView}_p|_T$ are colored with p.
- A set of vertices of $\mathcal{P}|_T$ is a simplex of $\mathcal{P}|_T$ if and only if there is a simplex of $\mathcal{P}$ that is formed by choosing one element in each vertex of the set.
- The topology on $V(\mathcal{P}|_T)$ is the discrete topology.

This definition makes $\mathcal{P}|_T$ a topological chromatic simplicial complex. As a chromatic simplicial complex, it is isomorphic to the classical finite-time construction of protocol complexes [13]. The finite-time execution map $\Xi_T : \mathcal{I} \to 2^{\mathcal{P}|_T}$ is defined by

$$\Xi_T(\sigma) = \{B_T[\tau] \mid \tau \in \Xi(\sigma)\}$$

where $B_T(\alpha) = \{\beta \in V(\mathcal{P}) \mid d(\alpha, \beta) < 2^{-T}\}$ is the function that takes each vertex α of $\mathcal{P}$ to the open 2^{-T}-ball in which it is included.

Projective Limits. We will show that, if the model is compact, then $\mathcal{P}$ is the limit of the $\mathcal{P}|_T$ in a precise sense. For this, we use the notion of projective limits from category theory [19], which we introduce in this subsection.

A *category* is a class of *objects* and a class of *morphisms* between objects. Every morphism $f : X \to Y$ is assigned a domain object X and a codomain object Y. For compatible morphisms $f : X \to Y$ and $g : Y \to Z$, the composition $g \circ f$ is a morphism $X \to Z$. The composition operator is required to be associative. For every object X, the existence of an identity morphism $\mathrm{id}_X : X \to X$ is required. The identity morphism satisfies $f \circ \mathrm{id}_x = f$ for all morphism $f : X \to Y$ with domain X and $\mathrm{id}_X \circ g = g$ for all morphisms $g : Z \to X$ with codomain X.

A sequence $(X_T)_{T\geq 0}$ of objects of a category can be transformed into an *inverse system* by specifying a family $(f_{S,T})_{0\leq S\leq T}$ of morphisms $f_{S,T} : X_T \to X_S$ such that $f_{T,T} = \mathrm{id}_{X_T}$ and $f_{R,T} = f_{R,S} \circ f_{S,T}$ for all $0 \leq R \leq S \leq T$. The *projective limit* of the sequence is then an object X together with morphisms $\pi_T : X \to X_T$ such that $\pi_S = f_{S,T} \circ \pi_T$ for all $0 \leq S \leq T$ and with the universal property that for any other such object Y and morphisms $\psi_T : Y \to X_T$, there exists a unique morphism $u : Y \to X$ such that the following diagram commutes for all $0 \leq S \leq T$:

$$\begin{tikzcd}
 & Y \arrow[ddl, "\psi_T"'] \arrow[d, "u"] \arrow[ddr, "\psi_S"] & \\
 & X \arrow[dl, "\pi_T"] \arrow[dr, "\pi_S"'] & \\
X_T \arrow[rr, "f_{S,T}"'] & & X_S
\end{tikzcd}$$

For every pair of integers S and T, $0 \leq S \leq T$, define the vertex maps $f_{S,T} : \mathcal{P}|_T \to \mathcal{P}|_S$ by setting $f_{S,T}(B)$ to be the unique open 2^{-S}-ball of $\mathcal{P}|_S$ in which the open 2^{-T}-ball B of $\mathcal{P}|_T$ is included. These are morphisms between topological chromatic simplicial complexes and they satisfy $f_{R,T} = f_{R,S} \circ f_{S,T}$ for all $0 \leq R \leq S \leq T$. This makes the sequence of the $\mathcal{P}|_T$ an inverse system.

▶ **Lemma 6.1.** *The projective limit of the sequence of complexes $\mathcal{P}|_T$ exists.*

We can equip the set of executions with the metric $d(E, E') = 2^{-K}$ where $K = \inf\{k \geq 0 \mid E_k \neq E'_k\}$, which measures how many configurations are identical in two execution prefixes [2]. With this topology on Exec, the projection maps $\pi_p : \mathsf{Exec} \to \mathsf{View}_p$ are continuous. In fact, continuity of the map means that each local view needs to be determined by some finite prefix of the execution. We have the following lemma:

▶ **Lemma 6.2.** *If Exec is compact, then $V(\mathcal{P})$ is compact as well, and $\mathcal{P}$ is the projective limit of the $\mathcal{P}|_T$.*

Sufficiency of Finite-Time Complexes for Compact Models. We can now state the fact that finite-time protocol complexes are sufficient to study compact models.

▶ **Theorem 6.3.** *If $V(\mathcal{P})$ is compact, then the following are equivalent:*
1. *The task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ is solvable.*
2. *There is a continuous chromatic simplicial map $\delta : \mathcal{P} \to \mathcal{O}$ such that $\delta \circ \Xi$ is carried by Δ.*
3. *There is a time T such that there exists a chromatic simplicial map $\delta_T : \mathcal{P}|_T \to \mathcal{O}$ such that $\delta_T \circ \Xi_T$ is carried by Δ.*
4. *The task $T = (\mathcal{I}, \mathcal{O}, \Delta)$ is solvable in a bounded number of local steps per process.*

On the other hand, if $V(\mathcal{P})$ is not compact, then the equivalence in Theorem 6.3 need not hold, as is shown by the example in Section 3.

7 Conclusion

We put together combinatorial and point-set topological arguments to prove a generalized asynchronous computability theorem, which applies also to non-compact computation models. This relies on showing that in non-compact models, protocols solving tasks correspond to simplicial maps that need to be *continuous*. We show an application to the *set agreement* task. We also show that the usual finite-time protocol complex, where protocols and simplicial maps are the same, suffices for *all* compact models.

It would be interesting to find other computation models and tasks where our techniques, and the generalized ACT, in particular, can be applied. Another intriguing direction for future research is to characterize which computation models lead to non-compact topological objects.

References

1. Dan Alistarh, James Aspnes, Faith Ellen, Rati Gelashvili, and Leqi Zhu. Why extension-based proofs fail. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC 2019)*, pages 986–996. ACM, New York, 2019.
2. Bowen Alpern and Fred B. Schneider. Defining liveness. *Information Processing Letters*, 21(4):181–185, 1985.
3. Hagit Attiya, Armando Castañeda, and Sergio Rajsbaum. Locally solvable tasks and the limitations of valency arguments. *Journal of Parallel and Distributed Computing*, 176:28–40, 2023. doi:10.1016/j.jpdc.2023.02.002.
4. Hagit Attiya and Sergio Rajsbaum. The combinatorial structure of wait-free solvable tasks. *SIAM Journal on Computing*, 31(4):1286–1313, 2002. doi:10.1137/S0097539797330689.
5. Nicolas Bourbaki. *General Topology. Chapters 1–4.* Springer, Heidelberg, 1989.
6. Armando Castañeda, Pierre Fraigniaud, Ami Paz, Sergio Rajsbaum, Matthieu Roy, and Corentin Travers. A topological perspective on distributed network algorithms. *Theoretical Computer Science*, 849:121–137, 2021.
7. Étienne Coulouma, Emmanuel Godard, and Joseph Peters. A characterization of oblivious message adversaries for which consensus is solvable. *Theoretical Computer Science*, 584:80–90, June 2015. doi:10.1016/j.tcs.2015.01.024.
8. Yannis Coutouly and Emmanuel Godard. A topology by geometrization for sub-iterated immediate snapshot message adversaries and applications to set-agreement. In Rotem Oshman, editor, *Proceedings of the 37th International Symposium on Distributed Computing (DISC 2023)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, 2023. To appear.
9. Tristan Fevat and Emmanuel Godard. Minimal obstructions for the coordinated attack problem and beyond. In *Proceedings of the 25th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2011)*, pages 1001–1011. IEEE, New York, 2011.
10. Pierre Fraigniaud, Ran Gelles, and Zvi Lotker. The topology of randomized symmetry-breaking distributed computing. In Avery Miller, Keren Censor-Hillel, and Janne H. Korhonen, editors, *Proceedings of the 40th ACM Symposium on Principles of Distributed Computing (PODC 2021)*, pages 415–425. ACM, New York, 2021. doi:10.1145/3465084.3467936.
11. Eli Gafni, Petr Kuznetsov, and Ciprian Manolescu. A generalized asynchronous computability theorem. In Shlomi Dolev, editor, *Proceedings of the 33rd ACM Symposium on Principles of Distributed Computing (PODC 2014)*, pages 222–231. ACM, New York, 2014.
12. Emmanuel Godard and Eloi Perdereau. Back to the coordinated attack problem. *Mathematical Structures in Computer Science*, 30(10):1089–1113, 2020. doi:10.1017/S0960129521000037.
13. Maurice Herlihy, Dmitry N. Kozlov, and Sergio Rajsbaum. *Distributed Computing Through Combinatorial Topology.* Morgan Kaufmann, Waltham, 2014. URL: https://store.elsevier.com/product.jsp?isbn=9780124045781.

14 Maurice Herlihy and Sergio Rajsbaum. Simulations and reductions for colorless tasks. In Darek Kowalski and Alessandro Panconesi, editors, *Proceedings of the 31st ACM Symposium on Principles of Distributed Computing (PODC 2012)*, pages 253–260. ACM, New York, 2012. `doi:10.1145/2332432.2332483`.

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

16 Gunnar Hoest and Nir Shavit. Toward a topological characterization of asynchronous complexity. *SIAM Journal on Computing*, 36(2):457–497, 2006. `doi:10.1137/S0097539701397412`.

17 Petr Kuznetsov, Thibault Rieutord, and Yuan He. An asynchronous computability theorem for fair adversaries. In Calvin Newport and Idit Keidar, editors, *Proceedings of the 37th ACM Symposium on Principles of Distributed Computing (PODC 2018)*, pages 387–396. ACM, New York, 2018. URL: `https://dl.acm.org/citation.cfm?id=3212765`.

18 Ronit Lubitch and Shlomo Moran. Closed schedulers: a novel technique for analyzing asynchronous protocols. *Distributed Computing*, 8:203–210, 1995.

19 Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, Heidelberg, 2nd edition, 1987.

20 Thomas Nowak, Ulrich Schmid, and Kyrill Winkler. Topological characterization of consensus under general message adversaries. In *Proceedings of the 38th ACM Symposium on Principles of Distributed Computing (PODC 2019)*, pages 218–227. ACM, New York, 2019. `doi:10.1145/3293611.3331624`.

21 Nicola Santoro and Peter Widmayer. Time is not a healer. In B. Monien and R. Cori, editors, *Proceedings of the 6th Annual Symposium on Theoretical Aspects of Computer Science (STACS 1989)*, pages 304–313. Springer, Heidelberg, 1989. `doi:10.1007/bfb0028994`.

22 Vikram Saraph, Maurice Herlihy, and Eli Gafni. Asynchronous computability theorems for t-resilient systems. In Cyril Gavoille and David Ilcinkas, editors, *Proceedings of the 30th International Symposium on Distributed Computing (DISC 2016)*, pages 428–441. Springer, Heidelberg, 2016. `doi:10.1007/978-3-662-53426-7_31`.

A Additional Details for Section 2 (Preliminaries)

A *simplicial* complex is a (possibly infinite) set V along with a (possibly infinite) collection $\mathcal{K}$ of finite subsets of V closed under containment *i.e.*, if $\sigma \in \mathcal{K}$ then $\sigma' \in \mathcal{K}$, for any $\sigma' \subseteq \sigma$. An element of V is called a *vertex* of $\mathcal{K}$, and the vertex set of $\mathcal{K}$ is denoted by $V(\mathcal{K})$. Each set in $\mathcal{K}$ is called a *simplex*. A subset of a simplex is called a *face* of that simplex. The *dimension of a simplex* σ, denoted $\dim \sigma$, is one less than the number of elements of σ, *i.e.*, $|\sigma| - 1$. The *dimension of a complex* is the smallest integer that upper bounds the dimension of any of its simplexes, or ∞ if there is no such bound. A simplex σ in $\mathcal{K}$ is called a *facet* of $\mathcal{K}$ if σ is not properly contained in any other simplex. A complex is *pure* if all its facets have the same dimension. We will focus on pure complexes, either finite or infinite.

Let $\mathcal{K}$ be a complex and σ be a simplex of it. The *star* of σ in $\mathcal{K}$ is the complex $\operatorname{st} \sigma = \{\tau \in \mathcal{K} \mid \sigma \subseteq \tau\}$.

Let $\mathcal{K}$ and $\mathcal{L}$ be complexes. A *vertex map* from $\mathcal{K}$ to $\mathcal{L}$ is a function $h : V(\mathcal{K}) \to V(\mathcal{L})$. If h also carries simplexes of $\mathcal{K}$ to simplexes of $\mathcal{L}$, it is called a *simplicial map*.

For two complexes $\mathcal{K}$ and $\mathcal{L}$, if $\mathcal{K} \subseteq \mathcal{L}$, we say $\mathcal{K}$ is a *subcomplex* of $\mathcal{L}$. Given two complexes $\mathcal{K}$ and $\mathcal{L}$, a *carrier map* $\Phi : \mathcal{K} \to 2^{\mathcal{L}}$ maps each simplex $\sigma \in \mathcal{K}$ to a subcomplex $\Phi(\sigma)$ of $\mathcal{L}$, such that for every two simplexes τ and τ' in $\mathcal{K}$ that satisfy $\tau \subseteq \tau'$, we have $\Phi(\tau) \subseteq \Phi(\tau')$. We say that Φ is *rigid* if for every $\sigma \in \mathcal{K}$, $\Phi(\sigma)$ is pure of dimension $\dim \sigma$.

A *geometric realization* of a complex $\mathcal{K}$ is an embedding of the simplexes of $\mathcal{K}$ into a real vector space such that, roughly speaking, intersections of simplexes are respected. All geometric realizations of a complex are topologically equivalent, *i.e.*, homeomorphic. Thus,

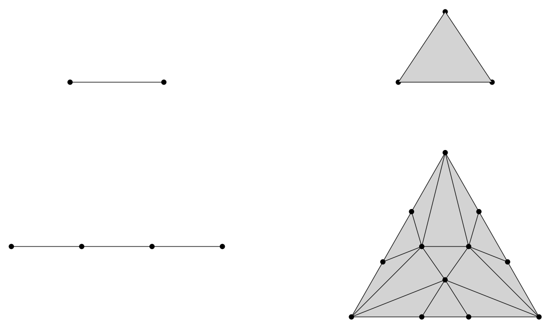

Figure 6 The standard chromatic subdivision of an edge and of a triangle.

we speak of *the* geometric realization of $\mathcal{K}$, which is denoted $|\mathcal{K}|$. The standard construction sets $|\mathcal{K}|$ equal to the set of functions $\alpha : V(\mathcal{K}) \to [0,1]$ such that $\{v \in V(\mathcal{K}) \mid \alpha(v) > 0\}$ is a simplex of $\mathcal{K}$ and $\|\alpha\|_1 = \sum_{v \in V(\mathcal{K})} \alpha(v) = 1$. The 1-norm induces a metric on $|\mathcal{K}|$ that makes its diameter equal to 1 if $\mathcal{K}$ has more than one vertex. Any simplicial map $h : \mathcal{K} \to \mathcal{L}$ induces a function $|h| : |\mathcal{K}| \to |\mathcal{L}|$. If the complexes are finite, then $|h|$ is necessarily continuous, and there is no guarantee of that otherwise.

A *coloring* of a complex $\mathcal{K}$ is a function $\chi : V(\mathcal{K}) \to \Pi$. The coloring is *chromatic* if any two distinct vertices of the same facet of $\mathcal{K}$ have distinct colors. A *chromatic* complex is a simplicial complex equipped with a chromatic coloring. A *labeling* of a complex $\mathcal{K}$ is a function $\ell : V(\mathcal{K}) \to L$, where L is a set. The set L will be a set of inputs, outputs or process states. Below, we will consider chromatic and labeled complexes such that each vertex is uniquely identified by its color together with its label, namely, for any two distinct vertices u and v, $(\chi(u), \ell(u)) \neq (\chi(v), \ell(v))$. For any vertex v of any such complex, we let denote by $v(p, x)$ the unique vertex of the complex with color $p \in \Pi$ and label $x \in L$.

Let $\mathcal{K}$ be a chromatic complex. The *standard chromatic subdivision* of $\mathcal{K}$, denoted $\operatorname{Chr}\mathcal{K}$, is the chromatic complex whose vertices have the form (p, σ), where $p \in \Pi$, σ is a face of a facet of $\mathcal{K}$ and $p \in \chi(\sigma)$. A set $\{(p_0, \sigma_0), (p_1, \sigma_1), \ldots, (p_s, \sigma_s)\}$ is a simplex of $\operatorname{Chr}\mathcal{K}$ if and only if $\sigma_0 \subseteq \sigma_1 \subseteq \ldots \subseteq \sigma_s$ and for all $0 \leq q, r \leq s$, if $q \in \chi(\sigma_r)$ then $\sigma_q \subseteq \sigma_r$. The chromatic coloring χ for $\operatorname{Chr}\mathcal{K}$ is defined as $\chi(p, \sigma) = p$. Figure 6 contains the standard chromatic subdivision of an edge, a 1-dimensional simplex, and a triangle, a 2-dimensional simplex. The k-th standard chromatic subdivision, $\operatorname{Chr}^k\mathcal{K}$, is obtained by iterating k times the standard chromatic subdivision. The standard chromatic subdivision is indeed a subdivision: $|\operatorname{Chr}^k\mathcal{K}| \cong |\mathcal{K}|$, for every $k \geq 0$.

A simplicial map $h : V(\mathcal{K}) \to V(\mathcal{L})$ is *chromatic* if it carries colors, *i.e.*, $\chi(v) = \chi(h(v))$, for every vertex v of $\mathcal{K}$. A carrier map $\Phi : \mathcal{K} \to 2^{\mathcal{L}}$ is *chromatic* if $\Phi(\sigma)$ is pure and chromatic of dimension $\dim\sigma$, and each facet of it has colors $\chi(\sigma)$.

▶ **Lemma 2.1.** *Let X be an ultrametric space. For all $x, y \in X$ and all $\delta, \varepsilon > 0$, one of the following is true: (1) $B_\delta(x) \cap B_\varepsilon(y) = \emptyset$, (2) $B_\delta(x) \subseteq B_\varepsilon(y)$, (3) $B_\varepsilon(y) \subseteq B_\delta(x)$.*

Proof. Assume that both (1) and (2) are false. We will prove that then (3) is true.

Let $v \in B_\varepsilon(y)$. We need to show that $v \in B_\delta(x)$. Since (1) is false, there exists a $z \in B_\delta(x) \cap B_\varepsilon(y)$. Applying the ultrametric triangle inequality twice, we have:

$$\begin{aligned} d(v,x) &\leq \max\{d(v,z), d(z,x)\} \leq \max\{d(v,y), d(y,z), d(z,x)\} \\ &< \max\{\varepsilon, \varepsilon, \delta\} = \max\{\varepsilon, \delta\} \end{aligned}$$

It remains to prove that $\varepsilon \leq \delta$ so that $\max\{\varepsilon, \delta\} = \delta$ and $v \in B_\delta(x)$.

Suppose by contradiction that $\varepsilon > \delta$. Since (2) is false, there exists a $u \in B_\delta(x) \setminus B_\varepsilon(y)$. But then we have

$$\begin{aligned} d(u,y) &\leq \max\{d(u,z), d(z,y)\} \leq \max\{d(u,x), d(x,z), d(z,y)\} \\ &< \max\{\delta, \delta, \varepsilon\} = \varepsilon \ , \end{aligned}$$

which means that $u \in B_\varepsilon(y)$, a contradiction to the choice of u. ◀

B Additional Details for Section 4 (Proof of the Generalized Asynchronous Computability Theorem with an Application to Set Agreement)

▶ **Theorem 4.1.** *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 simplicial map $\delta\colon K(\mathscr{T}) \to \mathcal{O}$ 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 $\delta(\tau) \in \Delta(\sigma)$.*

(c) *$|\delta|$ is continuous.*

Proof. ($\Rightarrow$): We prove that the geometric realization of the map δ as constructed in the proof of Gafni, Kuznetsov, and Manolescu [11, Theorem 6.1] is continuous by generalizing the proof given by Godard and Perdereau [12, Theorem 33] for the consensus task with two processes.

Let $x \in |K(\mathscr{T})|$ and $\varepsilon > 0$. We show the existence of an $\eta > 0$ such that:

$$\forall y \in |K(\mathscr{T})|\colon \quad d(x,y) < \eta \implies d\big(|\delta|(x), |\delta|(y)\big) < \varepsilon \tag{1}$$

Let σ be the minimal stable simplex in $K(\mathscr{T})$ such that $x \in |\sigma|$. Since $K(\mathscr{T})$ is locally finite, the star $\operatorname{st}\sigma = \{\tau \in K(\mathscr{T}) \mid \sigma \subseteq \tau\}$ is finite. Let k be the smallest round number such that $\operatorname{st}\sigma \subseteq \Sigma_k$. Denote by D_k the diameter of the geometric realization of simplices in $\operatorname{Chr}^k \mathcal{I}$ and choose $\eta = \varepsilon D_k$.

We show (1) in the geometric realization of every simplex $\tau \in \operatorname{st}\sigma$. By the choice of k, we have $\tau \in \operatorname{Chr}^r \mathcal{I}$ for some $0 \leq r \leq k$. Let $y \in |\tau|$ and denote by α the barycentric coordinates of x with respect to τ and by β the barycentric coordinates of y with respect to τ, *i.e.*, $x = \sum_{v\in\tau} \alpha(v)\cdot v$ and $y = \sum_{v\in\tau} \beta(v)\cdot v$ with $\alpha, \beta \geq 0$ and $\|\alpha\|_1 = \|\beta\|_1 = 1$. Here, we identified each vertex $v \in \tau$ with its position in the geometric realization $|K(\mathscr{T})|$. We then have:

$$d(x,y) = \|x-y\|_1 = \operatorname{diam}|\tau| \cdot \sum_{v\in\tau} |\alpha(v) - \beta(v)| \geq D_k \cdot \sum_{v\in\tau} |\alpha(v) - \beta(v)|$$

By definition of the geometric realization $|\delta|$, we have

$$|\delta|(x) = \sum_{v\in\tau} \alpha(v)\cdot\delta(v)$$

where, again, we identify the vertex $\delta(v)$ with its position in geometric realization $|\mathcal{O}|$. Since $\delta(v)$ is a vertex of $\mathcal{O}$ for every vertex $v \in \tau$, we have

$$d\big(|\delta|(x), |\delta|(y)\big) \leq \sum_{v\in\tau} |\alpha(v) - \beta(v)| \leq \frac{d(x,y)}{D_k} < \frac{\eta}{D_k} = \varepsilon \ ,$$

which shows (1) and concludes the proof of continuity of $|\delta|$.

($\Leftarrow$): We modify the protocol that is constructed in the proof of Gafni, Kuznetsov, and Manolescu [11, Theorem 6.1] for process p to decide in round k if the set

$$B_k(v) = \{w \in V(K(\mathscr{T})) \mid d(v, w) \leq D_k \wedge \chi(w) = p\}$$

only contains vertices that are mapped to the same output vertex by δ, where v is the view of the process in round k. This condition eventually becomes true since the subset topology on the geometric realization of the output vertices $V(\mathcal{O})$ is discrete, and thus δ is locally constant. ◀

▶ **Theorem 4.2.** *Let M be an IIS-sub model such that for any termination subdivision $\mathscr{T}$ of σ that is admissible for M, $|\sigma| = |K(\mathscr{T})|$. Then, $(n-1)$-set agreement is impossible in M.*

Proof. Let M be a sub-IIS model. By Theorem 4.1, if $(n-1)$-set agreement is solvable in model M, there is a (possibly infinite) terminating subdivision $\mathscr{T}$ of σ and a chromatic simplicial map $\delta : K(\mathscr{T}) \to \partial\sigma$ such that (1) $\mathscr{T}$ is admissible for M, (2) for every input simplex $\sigma' \subseteq \sigma$, if τ is a stable simplex of $\mathscr{T}$ such that $|\tau| \subseteq |\sigma'|$, then $\delta(\tau) \in \Delta(\sigma')$, and (3) $|\delta|$ is continuous.

Let us suppose that $|\sigma| = |K(\mathscr{T})|$, namely, $K(\mathscr{T})$ subdivides σ. Thus, for each face $\sigma' \subseteq \sigma$, $|\sigma'| = |K(\sigma')|$, where $K(\sigma')$ denotes the terminating subdivision of σ'. Consider the identity map $g : |\sigma| \to |K(\mathscr{T})|$. Clearly, g is continuous, with $g(|\sigma'|) = |K(\sigma')|$. Consider the function $f = |\delta| \circ g : |\sigma| \to |\partial\sigma|$. Since $|\delta|$ and g are continuous, the function f is continuous too. We argue that $f(|\sigma'|) \subseteq |\sigma'|$, for every proper face $\sigma' \subset \sigma$. Consider any proper face $\sigma' \subset \sigma$. We have that (a) $g(|\sigma'|) = |K(\sigma')|$, by definition of g, (b) for any stable simple $\tau \in \mathscr{T}$ with $|\tau| \subseteq |\sigma'| = |K(\sigma')|$, $\delta(\tau) \in \Delta(\sigma')$, by the properties of δ, and (c) $\Delta(\sigma') = \sigma'$, by definition of Δ. We thus conclude that $f(|\sigma'|) \subseteq |\sigma'|$.

The following lemma is direct consequence of Lemma 4.3.5 in [13], and proves below the impossibility of $(n-1)$-set agreement whenever $|K(\mathscr{T})| = |\sigma|$.

▶ **Lemma B.1.** *There is no continuous map $f : |\sigma| \to |\partial\sigma|$ such that for every proper face $\sigma' \subset \sigma$, $f(|\sigma'|) \subseteq |\sigma'|$.*

One can understand Lemma B.1 as a continuous version of the discrete Sperner's lemma. Intuitively, it states that if a continuous map $f : |\sigma| \to |\partial\sigma|$ maps the boundary of σ to itself (*i.e.*, $f(|\sigma'|) \subseteq |\sigma'|$, for each $\sigma' \subset \sigma$), similar to Sperner's lemma hypothesis, then f cannot exist because the mapping cannot be extended to the interior of σ, since $|\sigma|$ is solid whereas $|\partial\sigma|$ has a hole.

As explained above, Theorem 4.1 and assumption $|K(\mathscr{T})| = |\sigma|$ imply that if $(n-1)$-set agreement is solvable in M, then there exists a continuous map $f : |\sigma| \to |\partial\sigma|$ such that for every proper face $\sigma' \subset \sigma$, $f(|\sigma'|) \subseteq |\sigma'|$. Such continuous map f contradicts Lemma B.1. Therefore, $(n-1)$-set agreement is impossible in M. ◀

C Additional Details for Section 5 (Characterization of Task Solvability in General Models)

▶ **Lemma 5.1.** *The distance function d_p is an ultrametric on* CView_p.

Proof. If $\alpha = \beta$, then $d_p(\alpha, \beta) = 0$. If $d_p(\alpha, \beta) = 0$, then $T_p(\alpha, \beta) = \infty$, which means that there is no index at which they differ by definition, *i.e.*, $\alpha = \beta$. This shows that d_p is positive definite.

The symmetry condition $d_p(\alpha, \beta) = d_p(\beta, \alpha)$ holds since the definition of $T_p(\alpha, \beta)$ is symmetric in α and β.

We now prove the ultrametric triangle inequality by showing:

$$T_p(\alpha,\gamma) \geq \min\left\{T_p(\alpha,\beta), T_p(\beta,\gamma)\right\}$$

Assume by contradiction that $T_p(\alpha,\gamma) < \min\left\{T_p(\alpha,\beta), T_p(\beta,\gamma)\right\}$. Set $t = T_p(\alpha,\gamma)$. Since $t < \infty$ and $t < T_p(\alpha,\beta)$, all local views up to index t coincide in both sequences α and β. Likewise, all local views up to index t coincide in both sequences β and γ. But then, by transitivity of the equality relation on local views, all local views up to index t coincide also in the two sequences α and γ, which means $T_p(\alpha,\gamma) > t = T_p(\alpha,\gamma)$; a contradiction. ◀

▶ **Lemma 5.2.** *Let $\delta_p : \mathsf{CView}_p \to V(\mathcal{O})$ be a function. The following are equivalent:*
1. *There is a protocol such that process p decides the value $\delta_p(\alpha)$ in every execution $E \in \mathsf{Exec}$ with local view $\pi_p(E) = \alpha \in \mathsf{CView}_p$.*
2. *The function δ_p is continuous when equipping CView_p with the topology induced by d_p and $V(\mathcal{O})$ with the discrete topology.*

Proof. ($\Rightarrow$): To show that δ_p is continuous, we will show that the inverse image of any singleton $\{o\} \subseteq V(\mathcal{O})$ is open with respect to d_p. This then implies that the inverse image of any subset $O \subseteq V(\mathcal{O})$, *i.e.*, of any subset of $V(\mathcal{O})$ that is open with respect to the discrete topology, is open with respect to d_p.

Let $\alpha \in \delta_p^{-1}[\{o\}]$. Because process p decides the value o in the local view α, there exists an index T at which this decision has already happened in α. Choose $\varepsilon = 2^{-T}$. Now let $\alpha' \in \mathsf{CView}_p$ with $d_p(\alpha,\alpha') < \varepsilon$. Then, by definition of d_p, the local views of α and of α' are indistinguishable for process p up to and including index T. But then, process p needs to have decided value o at index T in local view α' as well. We thus have $\delta_p(\alpha') = o$, which means that $\alpha' \in \delta_p^{-1}[\{o\}]$. Therefore, the inverse image $\delta_p^{-1}[\{o\}]$ is open with respect to d_p.

($\Leftarrow$): We define the protocol for process p in the following way. Decide value $o \in V(\mathcal{O})$ in the t^{th} step if the set of all local views in CView_p that are indistinguishable from the current execution in the first t steps of process p is included in the inverse image $\delta_p^{-1}[\{o\}]$.

Let $E \in \mathsf{Exec}$ be an execution with $p \in \mathsf{Correct}(E)$. We will show that process p decides value $o = \delta_p(\alpha)$ where $\alpha = \pi_p(E)$. By definition of o, we have $\alpha \in \delta_p^{-1}[\{o\}]$. By continuity of δ_p, the inverse image $\delta_p^{-1}[\{o\}]$ is an open set in CView_p. There hence exists an $\varepsilon > 0$ such that $\alpha' \in \delta_p^{-1}[\{o\}]$ for all $\alpha' \in \mathsf{CView}_p$ with $d_p(\alpha,\alpha') < \varepsilon$. It remains to show that process p eventually decides the value o and that it does not decide any other value in execution E. Setting $T = \lceil \log_2 \varepsilon \rceil$, we see that, by design of the protocol's decision rule, process p has decided value o at the latest in step number T. To show that process p does not decide any other value than o, it suffices to observe that $d(\alpha,\alpha) = 0 < 2^{-t}$ for every $t \geq 0$ and $\alpha \in \delta_p^{-1}[\{o\}]$. ◀

▶ **Lemma 5.3.** *The execution map Ξ is a carrier map such that $\mathcal{P} = \bigcup_{\sigma \in \mathcal{I}} \Xi(\sigma)$.*

Proof. We first prove that Ξ is a carrier map. Let $\sigma \subseteq \tau$. We need to prove that $\Xi(\sigma) \subseteq \Xi(\tau)$. We first show that $\mathsf{Compatible}(\sigma) \subseteq \mathsf{Compatible}(\tau)$. Let $E \in \mathsf{Compatible}(\sigma)$. Then $\mathsf{Part}(E) \subseteq \chi[\sigma] \subseteq \chi[\tau]$ since $\sigma \subseteq \tau$, which means that $E \in \mathsf{Compatible}(\tau)$ because the second condition in the definition is fulfilled since $\pi_p(\sigma) = \pi_p(\tau)$ for all $p \in \chi[\sigma]$. But then, for every $\varphi \in \Xi(\sigma)$, we also have $\varphi \in \Xi(\tau)$ since every E in the definition of $\Xi(\sigma)$ is also valid for $\Xi(\tau)$.

To prove $\mathcal{P} = \bigcup_{\sigma\in\mathcal{I}} \Xi(\sigma)$, it suffices to show $\mathsf{Exec} = \bigcup_{\sigma\in\mathcal{I}} \mathsf{Compatible}(\sigma)$. The inclusion of $\mathsf{Compatible}(\sigma)$ in Exec is immediate by its definition. So let $E \in \mathsf{Exec}$ be any execution. We need to show the existence of a simplex $\sigma \in \mathcal{I}$ such that $E \in \mathsf{Compatible}(\sigma)$. For this, it suffices to choose $\sigma = \{v(p, \mathsf{Init}_p(E)) \mid p \in \mathsf{Part}(E)\}$. This set is an input simplex since the initial values in Exec are chosen according to $\mathcal{I}$ by definition. ◀