Towards a Topological Characterization of Asynchronous Complexity

(Extended Abstract)

Gunnar Hoest
M.I.T.
and
Nir Shavit
M.I.T. and Tel-Aviv University

This paper introduces the use of topological models and methods, formerly used to analyze computability, as tools for the quantification and classification of *asynchronous complexity*.

We present the first *asynchronous complexity theorem*, applied to decision tasks in the iterated immediate snapshot (IIS) model of Borowsky and Gafni. We do so by introducing a novel form of span called the *non-uniform chromatic subdivision*. Building on the framework of Herlihy and Shavit's topological computability model, our theorem states that the time complexity of any asynchronous algorithm is directly proportional to the level of non-uniform chromatic subdivisions necessary to allow a simplicial map from a task's input complex to its output complex.

To show the power of our theorem, we use it to derive two new results. The first includes tight upper and lower bounds on the time to acheive n process approximate agreement. Our bound of $\left\lfloor \log_d \frac{input-range}{\epsilon} \right\rfloor$ where $d = 3$ for two processes and $d = 2$ for 3 or more shows that the intriguing gap between the known lower and upper bounds implied by the work of Aspnes and Herlihy is not a technical coincidence. The second is a simple and purely geometric proof that the time complexity of solving the k process renaming problem of Attiya et al. when the number of names is $k(k+1)/2$ or more is $O(1)$.

More than the new bounds themselves, the importance of our asynchronous complexity theorem is that the algorithms and lower bounds it allows us to derive are intuitive and simple, with topological proofs that require no mention of concurrency at all.

1. INTRODUCTION

In the last few years, techniques of modeling and analysis based on classical algebraic topology [5; 9; 12; 15; 19; 18; 20; 21; 22; 23; 27] in conjunction with distributed simulation methods [9; 8; 7; 25] have brought about significant progress in our understanding of computability problems in an asynchronous distributed setting. We feel the time is ripe to extend these techniques to address *asynchronous complexity*.

This paper studies the class of problems called *decision tasks*, input/output problems in which N asynchronous processes start with input values, communicate via shared memory and halt with private output values. We focus on the iterated immediate snapshot memory model introduced by Borowsky and Gafni [9] as part of

---

`Contact author: Nir Shavit, shanir@theory.lcs.mit.edu.`
`This submission is for the long presentation track.`
`It is eligable for a student award as Gunnar Hoest is a full time student.`

their new simplified proof of the asynchronous computability theorem [23] [1]. The model is a restriction of atomic snapshot memory that guarantees that processes' scan operations return views that contain non-decreasing sets of the participating processes' inputs. We believe it is a good first candidate for topological modeling since it has a particularly nice geometric representation, and hence easily lends itself to topological analysis.

Keeping in style with Herlihy and Shavit's topological computability framework [23], our theorem states that the worst case time complexity for solving a decision task in the IIS model is equivalent to the minimal number of non-uniform chromatic subdivisions of the input complex necessary to allow a simplicial map from the input complex to the output complex. The theorem also immediately provides a matching upper bound given the subdivision and mapping.

The non-uniform chromatic subdivisions we introduce (See Figure 1 for examples) are a looser and more general form of standard chromatic subdivisions [23]. Unlike the iterated standard chromatic subdivions used in the computability work of [23; 9], they allow individual simplices in a complex to be subdivided a different number of times, while assuring that the subdivision of the complex as a whole remains consistent. Non-uniformity is a necessary property when analyzing complexity since the number of steps and hence the level of subdivision of an input simplex may differ from one set of inputs to the next, as for example in the approximate agreement problem. Taking just the complexity of the execution on the worst case inputs would make the complexity theorem useless since for example, for the approximate agreement problem Aspnes and Herlihy [1] show that for any k one can find a set of inputs that will require time k in the worst case.

The power of our theorem lies in its ability to allow one to reason about the complexity of problems in a purely geometric setting. As we show, the subdivisions of a complex are a clean and higher level way of thinking about the multitude of different length executions of a concurrent protocol. We found this geometric representation helpful and are sure that it will prove to be an invaluable tool for designing and analyzing concurrent algorithms.

We provide two example applications of our theorem. In Section 5 we show tight upper and lower bounds on the time to acheive n process approximate agreement. Our bound of $\lfloor \log_d \frac{input-range}{\epsilon} \rfloor$ where $d = 3$ for two processes and $d = 2$ for 3 or more shows that the intriguing gap between the known lower and upper bounds implied by the work of Aspnes and Herlihy [1] is not a technical coincidence. Then, in Section 6, we show a simple proof that the time complexity of solving the k process renaming problem of Attiya et al. [3] when the number of names is $k(k+1)/2$ or more is $O(1)$. This falls short of the $O(k)$ bound of Moir and Garay [16] in the read/write register model since known implementations of IIS from read/write registers take $\Theta(k^2)$ time. However, we believe the simplicity and intuitive appeal of our purely geometric proof offers some advantages.

Since there is quite a bit of mathematical machinery needed in order to understand the foundations on which we build our main theorem, we try in the next section to start out by presenting it on an intuitive level.

[1] This model was implicitly used by Herlihy and Shavit in their upper bound proof [22; 23].

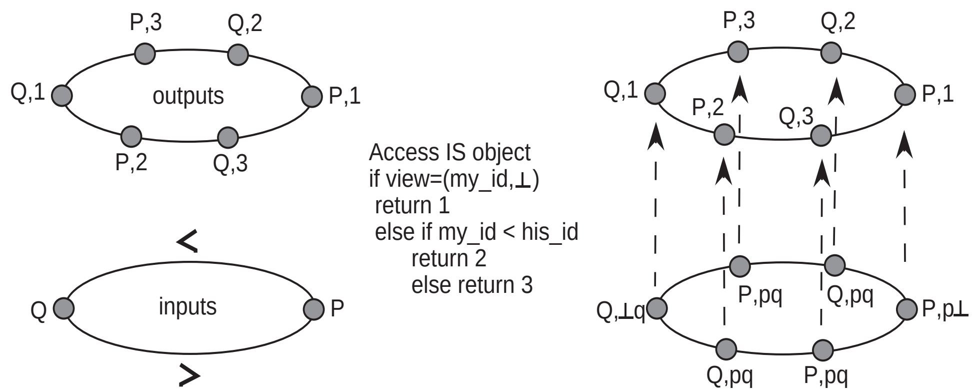

Fig. 1. Two Process Renaming

1.1 A bit of intuition

It is perhaps best to explain our theorem by way of example. Let us begin with a simple one. Consider any comparison based protocol solving the two process version of the Attiya et al. renaming problem with 3 names. The left hand side of Figure 1 describes the input and output complexes. Recall that the name space of processes is very large, that is, the input complex contains many vertices corresponding to many possible process names. However, since this is a comparison based protocol they can be represented by two vertices p and q and two edges (1-simplices) between them representing the case where $p > q$ and the case where $p < q$. The output complex is a cycle of 1-simplices the describes all the possible combinations of legal output values in two process executions of renaming with three output names.

Now, our complexity theorem says that the complexity of any protocol solving the renaming problem in IIS is exactly the number of chromatic subdivisions necessary to allow a simplicial map [2] from the subdivided input complex to the output complex. And indeed a single subdivision will allow such a map as depicted on the right hand side. Hence this problem is solvable in the IS model in exactly one step.

As will be shown in great detail later, the idea is that the added nodes of the subdivision actually capture the possible different executions that may result from processes accessing an IS object. This can be seen in the labeling of the vertexes that correspond to whether they saw the other process in the view returned from the IS object. The implied symmetric protocol appears in the middle of Figure 1.

Now, what happens when we go to three processes? Well, a single subdivision of a 2-simplex looks like the left hand side of Figure 2. Notice that with 6 possible output names we can color this complex so that all 2-simplices have different colors. In general, as we show in Section 6, one can color a subdivided n-simplex corresponding to the $N = n + 1$ process executions of a protocol with $N(N + 1)/2$ names, which implies that one can solve renaming in this case with a single IS object. This offers a partial explanation for Garay and Moir's ability to beat all known renaming upper bounds for the case where there are $N(N + 1)/2$ names [16].

However, if we only have 5 names, the reader can convince herself that one needs

[2] Meeting the problem restrictions – i.e. the map must be symmetric for all cases where processors

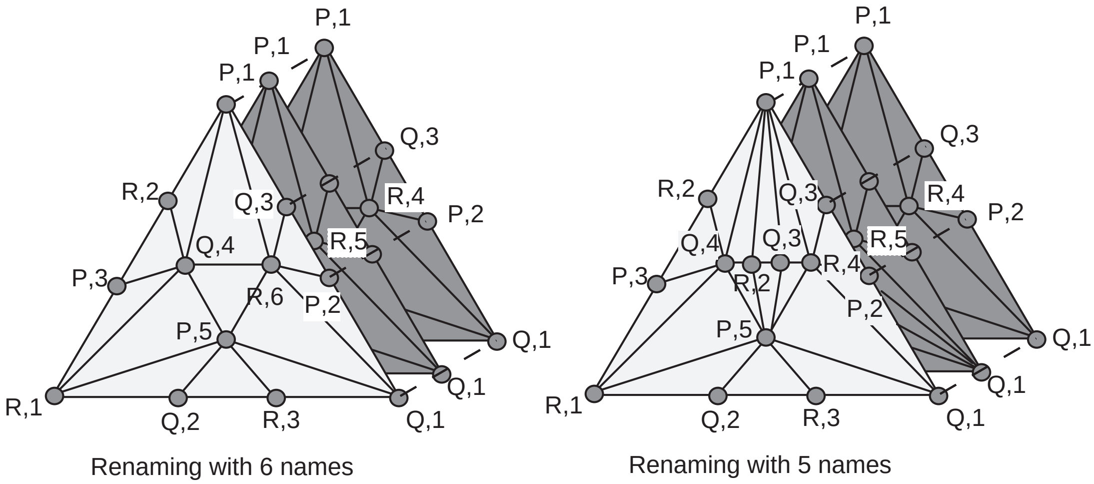

Fig. 2. Three Process Renaming with Five and Six Names

to subdivide once more in order to subdivide the 2-simplex so that all subsimplices can be colored in different names. The right hand side of Figure 2 shows how this can be done, and is an example of a non-uniform subdivision – some simplices are subdivided more than others. This figure implies that with 5 names the time complexity of solving 3 process renaming is two.

Finally, consider the approximate agreement problem, which we discuss in detail in Section 5. We are able to explain and close the upper/lower bound gap implied by Aspnes and Herlihy's approximate agreement work[1]. Figure 5, which shows the subdivisions induced by a three process approximate agreement protocol on some given input set. Aspnes and Herlihy derive their lower bound for any N process algorithm from a "bad" execution in which only the two processes with inputs farthest apart participate. Such an execution in the figure corresponds to a sequence of subdivisions of the triangle edge between P and Q. Since each subdivision introduces two new vertexes and splits the edge in three, one can only get a $\log_3$ bound. However, note that if one considers three processes, a 2-simplex and not just a path must be subdivided and and as we show, no matter how one subdivides it, there is always a path between P and Q that includes vertexes of R (marked by a darker color in the figure) that will be cut by at most a half in each subdivision, implying a tight $\log_2$ lower bound.

2. THE ITERATED IMMEDIATE SNAPSHOT MODEL

This section presents our non-uniform version of the Iterated Immediate Snapshot model of computation.

Borowsky and Gafni's immediate snapshot object (IS) [7] is by now a standard tool for arguing about asynchronous shared memory computation. It is essentially a restriction of standard atomic snapshot memory, in which a set of processors write concurrently and then immediately return a snapshot of memory. The object

---

have the same relative id order.

consists of a shared array of n cells, and supports a single external operation, called $writeread_i$, which writes a value to the i-th shared memory array cell, and subsequently returns a snapshot of the entire array. It guarantees that processes' scan operations return views that contain non-decreasing sets of the participating processes' inputs. A formal I/O automata based specification can be found in the Appendix.

The *iterated* IS model (IIS) was recently formulated as a computation model by Borowsky and Gafni [9]. The model assumes an infinite sequence $IS_0, IS_1, IS_2 \ldots$ of IS objects. In any given execution, all processes pass through *exactly* k IS objects in sequence, where the input to one object is the output to the next. At the end they apply a mapping function δ that returns the output value corresponding to a process' collected view.

We generalize the IIS model by introducing the concept of a *non-uniform protocol* in the IIS model. Unlike in the standard IIS model, each participating processor i accesses a possibly different number of IS objects, and then halts. Any such protocol can presented in normal form as in Figure 2 by properly picking the **is_final_state** predicate for each process.

Non-uniformity is necessary when building a complexity model since the number of steps taken by processes my differ based on the inputs, as in the approximate agreement problem. Assuming a uniform IIS model would mean that the complexity of the algorithm on any input is the complexity of the execution on the worst case inputs. This makes a complexity theorem useless since for example, for approximate agreement, Aspnes and Herlihy [1] show that for any k one can find a set of inputs that will require time k.

```
local_view := input_value; k := 0
forever do
 if is_final_state(local_view) return δ(local_view)
   else with IS_k  do
           local_view := writeread(local_view)
           k := k+1
od
```

Fig. 3. A Non-Uniform IIS Protocol in Normal Form

We can now define our complexity measure.[3] Let $\mathcal{P}$ be a protocol in the non-uniform IIS model, and let α be any execution of $\mathcal{P}$. Let k_i be the number of IS objects accessed by process i in α.

DEFINITION 2.1. *The time complexity of α is $\max_i k_i$, the maximal number of IS objects accessed by any process.*

Since the IIS model is equivalent to a restriction of regular atomic snapshot memory, in which all processes run in phases, in which writes by a group of processes are followed immediately by snapshots by the same group of processes, it follows that:

[3]Note that our model also lends itself naturally to analysis of "work."

LEMMA 2.2. *Any time complexity lower bound in the non-uniform IIS model is also a lower bound in the atomic snapshot model.*

It should be noted that the lower bounds obtained in this way may be crude, as they do not take into account executions that do not correspond to executions in the highly regular IIS model.

3. THE NON-UNIFORM CHROMATIC SUBDIVISION

The standard chromatic subdivision was introduced by Herlihy and Shavit [21; 22], and has since been used by a number of researchers. It is essentially a chromatic generalization of the standard barycentric subdivision from classical topology. For the sake of simplicity in this abstract we present here an informal definition. A formal definition and proof that the standard chromatic subdivision is a subdivision appear in the Appendix.

Let K^n be a pure chromatic complex, and let $S^n = (\vec{s}_0, \ldots, \vec{s}_n)$ be a simplex in K^n, where $id(\vec{s}_i) = i$, the id of process i. In the *standard chromatic subdivision* of S^n, denoted $\chi(S^n)$, each m-simplex, where $m \leq n$, has the form $(\langle 0, S_0 \rangle, \ldots, \langle m, S_m \rangle)$, where S_i is a face of S^n, such that (1) $i \in ids(S_i)$, (2) for all S_i and S_j, one is a face of the other, and (3) if $j \in ids(S_i)$, then $S_j \subseteq S_i$. The standard chromatic subdivision of K^n is just the union of all the $\mathcal{X}(S^n)$, as S^n ranges over all the n-simplices in K^n.

Applying the standard chromatic subdivision k times, where $k > 1$, yields a subdivision $\mathcal{X}^k(\mathcal{K}^n) = \mathcal{X}(\mathcal{X}^{k-1}(\mathcal{K}^n))$, which we call the kth iterated standard chromatic subdivision. Since the chromatic subdivision relation is transitive, $\mathcal{X}^k(\mathcal{K}^n)$ is a chromatic subdivision of $\mathcal{K}^n$. The number k is called the *level* of the subdivision. The 2-simplex with vertexes in $\{a, d, e\}$ in the lefthand side of Figure 4 is standard chromatically subdivided. So is its 1-simplex edge $\{a, e\}$. (Ignore the fact that the complex as a whole isn't).

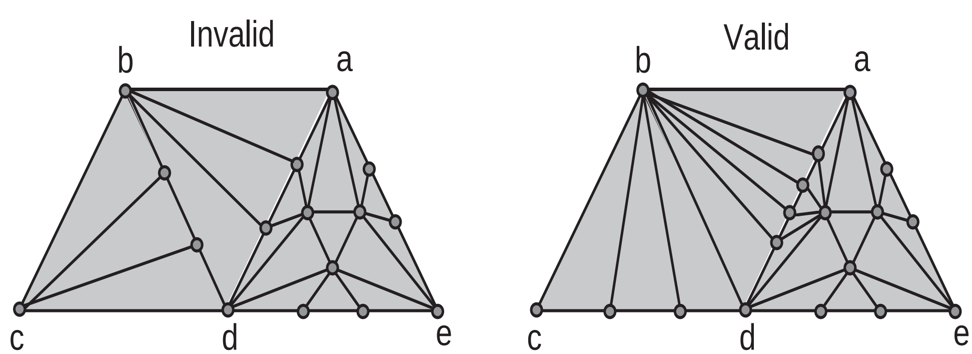

Fig. 4. Valid and Invalid Non-uniform Subdivisions

We introduce the concept of a *non-uniform chromatic subdivision*, a generalization of the standard chromatic subdivision in which the different simplices of a complex are not necessarily subdivided the same number of times. Informally, the non-uniform chromatic subdivision of a complex $\mathcal{K}^n$, $\widetilde{\mathcal{X}}(\mathcal{K}^n)$, is constructed by choosing, for each n-simplex in $\mathcal{K}^n$, a *single* face of the simplex (a face can be of any dimension and could also be the whole simplex) to which we apply the standard chromatic subdivision. We then induce the subdivision onto the rest of the simplex. This is best seen in Figure 4. Its right hand side shows a valid non-uniform chromatic subdivision of a complex where the simplex $\{b, c, d\}$'s subdivision is the result of subdividing its 1-face $\{c, d\}$ once and then inducing this subdivision onto the rest of the simplex. The left hand side is not a legal subdivision since the 2-simplex $\{a, b, d\}$ has two subdivided faces. The k-th level *non-uniform iterated chromatic subdivision* is generated by repeating this process k times, where only simplices in faces that were subdivided in round $k-1$ can be again subdivided in phase k. The complex on the right hand side of Figure 4 is an example of a valid non-uniform iterated chromatic subdivision.

Later, we will show that these structures correspond in a natural way to the set of protocol complexes in the non-uniform IIS model of computation. In fact, it turns out that each non-uniform standard chromatic subdivision is equal to some protocol complex of non-uniform IIS (up to isomorphism).

DEFINITION 3.1. *Let $\mathcal{K}^n$ be a pure chromatic complex, where the colors are the numbers in $\mathcal{Z}_{n+1}$. Label each vertex $\vec{v}$ in $\mathcal{K}^n$ with $\langle i, \vec{v}\rangle$, where i is the color of $\vec{v}$. For each maximal simplex T^n in $\mathcal{K}^n$, decompose its complex of faces $\mathcal{T}^n$ into two arbitrarily chosen subcomplexes $\mathcal{C}$ and $\mathcal{S}$, such that $\mathcal{T}^n = \mathcal{C} \cdot \mathcal{S}$.* [4] *The vertices in $\mathcal{C}$ are refered to as* continuing, *and those in $\mathcal{S}$ as* stopped. *We require that these subcomplexes be chosen such that for all maximal simplices T_i^n, T_j^n in $\mathcal{K}^n$, we have that*

—$\mathcal{C}_i \cap \mathcal{T}_j \subseteq \mathcal{C}_j$
—$\mathcal{C}_i \cap \mathcal{S}_j = \emptyset$

We also require that there is at least one simplex T^n in $\mathcal{K}^n$ for which the set of continuing vertices is nonempty. The the non-uniform chromatic subdivision of $\mathcal{K}^n$ of level 1, $\widetilde{\mathcal{X}}^1(\mathcal{K}^n)$ is defined as

$$\widetilde{\mathcal{X}}^1(\mathcal{K}^n) = \bigcup_{T \in \mathcal{K}^n} \mathcal{S} \cdot \mathcal{X}(\mathcal{C})$$

A non-uniform chromatic subdivision of $\mathcal{K}^n$ of level k, which we typically denote $\widetilde{\mathcal{X}}^k(\mathcal{K}^n)$, can be obtained by applying the procedure described above iteratively k times, in such a way that, at each step, none of the continuing vertices are part of the set of stopped vertices from the previous step.

Informally speaking, a non-uniform chromatic subdivision of level k is one in which there is some simplex in $\mathcal{K}^n$ which is subdivided k times, but no simplex that is subdivided more than k times. Note that if we always choose $\mathcal{C}$ equal to $\mathcal{T}$, we get the iterated standard chromatic subdivision of $\mathcal{K}^n$, $\mathcal{X}^k(\mathcal{K}^n)$. Hence there for all $k \geq 0$ there exists some non-uniform iterated chromatic subdivision of level k. The Appendix includes a proof of the following lemma.

[4] The "$\cdot$" operator stands for a join [26; 23]

LEMMA 3.2. *Any non-uniform chromatic subdivision* $\widetilde{\mathcal{X}}^k(\mathcal{K}^n)$ *is a chromatic subdivision of* $\mathcal{K}^n$.

The level of subdivision necessary for the existence of a simplicial map from the input to the ouput complex of a decision task that agrees with the task specification can be interpreted as a topological measure of the task's time complexity. The following definition introduces some useful constructs for reasoning about this relationship.

DEFINITION 3.3. *Given a decision task* $\langle \mathcal{I}^n, \mathcal{O}^n, \Gamma \rangle$ *and a non-negative integer* k*, we say that* $\widetilde{\mathcal{X}}^k(\mathcal{I}^n)$ *is a* mapable *subdivision of the input complex, and* k *is a* mapable *level of subdivision if there exists some color and carrier preserving simplicial map* μ *from* $\widetilde{\mathcal{X}}^k(\mathcal{I}^n)$ *to* $\mathcal{O}^n$ *such that for all* T^m *in* $\widetilde{\mathcal{X}}^k(\mathcal{I}^n)$*,* $\mu(T^m) \in \Gamma(T^m)$.

This definition extends naturally to individual simplices as the map induces different levels of subdivision on the individual simplices in accordance with the idea that, in order to solve a decision task, some processes may have to do more computational work than others, and some inputs may require more computation than others.

4. AN ASYNCHRONOUS COMPLEXITY THEOREM

We can now state our main theorem.

THEOREM (TIME COMPLEXITY). *A decision task* $\langle \mathcal{I}^n, \mathcal{O}^n, \Gamma \rangle$ *has a wait-free protocol in the non-uniform IIS model with worst case time complexity* k_{S^m} *on input* S^m *iff there is a mapable non-uniform iterated chromatic subdivision* $\widetilde{\mathcal{X}}^k(\mathcal{I}^n)$ *with level* k_{S^m} *on* S^m.

Keeping in style with Herlihy and Shavit [23], the theorem simply states that solvability of a decision task $\langle \mathcal{I}^n, \mathcal{O}^n, \Gamma \rangle$ in the IIS model is equivalent to the existence of a color and carrier preserving simplicial map μ from some non-uniform iterated chromatic subdivision $\widetilde{\mathcal{X}}^k(\mathcal{I}^n)$ to $\mathcal{O}^n$ that agrees with the task specification Γ, that is, for all T^m in $\widetilde{\mathcal{X}}^k(\mathcal{I}^n)$, $\mu(T^m) \in \Gamma(T^m)$. The level k_{S^m} is a lower bound on the worst case time complexity of solving this task with inputs in S^m in the IIS model.

The theorem also immediately provides a matching upper bound given the subdivision and mapping. Simply run the normal form protocol of Figure 2. Since each process can locally store the subdivision and mapping, `is_final_state` just needs to test if a `local_view` is equal to some node v in the subdivison and if so return $\mu(v)$.

5. APPROXIMATE AGREEMENT

As an application of our complexity theorem, we will analyze the well-known approximate agreement task, in which each process i is given a real-valued input x_i, and is required to decide on some output y_i such that, for some predetermined $\epsilon > 0$, $\max y_i - \min y_i < \epsilon$, and for all i, $y_i \in [\min x_i, \max x_i]$. Aspnes and Herlihy [1] proved [5] a lower bound that implies a worst case time complexity of

[5] Though their proofs are for the read/write register model, they carry onto ours.

$\left\lfloor \log_3 \frac{\max x_i - \min x_i}{\epsilon} \right\rfloor$ and an upper bound of $\left\lfloor \log_2 \frac{\max x_i - \min x_i}{\epsilon} \right\rfloor$. This leaves a small but intriguing gap. We now show that this gap is not simply a technical fluke.

THEOREM 5.1. *Given $\epsilon > 0$, let $\{x_0, \ldots, x_n\}$ be a set of inputs to the approximate agreement problem for $n+1$ processes, where $n > 0$. The complexity of solving approximate agreement on this input set is exactly $\left\lfloor \log_d \frac{\max y_i - \min y_i}{\epsilon} \right\rfloor$ where $d = 3$ for two processes and $d = 2$ for three or more.*

Our theorem provides the matching upper bound algorithm and our lower bound by Lemma 2.2 applies to atomic snapshots as well. We hope to convince the reader that this is an excellant example of how topological modelling exposes subtle points which would otherwise be difficult to grasp.

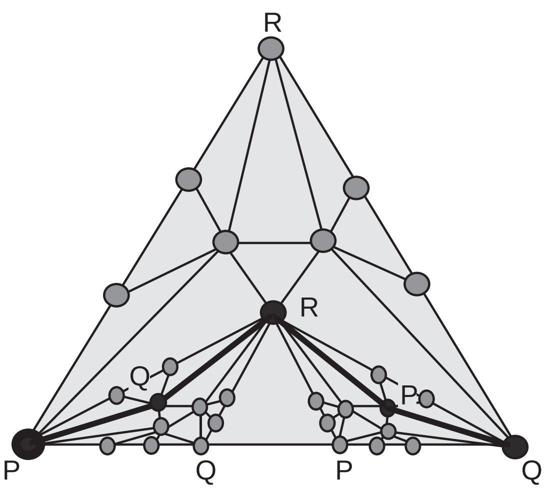

Fig. 5. Simplex Subdivided by an Approximate Agreement Protocol

The key intuition behind our ability to close the upper/lower gap is depicted in Figure5, which shows the subdivisions induced by a three process execution on some given input. Aspnes and Herlihy [1] derive their lower bound for any $n+1$ process algorithm from a "bad" execution in which only the two processes with inputs farthest apart participate. Such an execution in our model corresponds to a sequence of subdivisions of the edge between p and q. In the end each simplex on the subdivided path will have to map its p and q vertexes to output values that are ϵ apart. Since each subdivision introduces two new vertexes and splits the edge in three, in $\log_3$ such steps one can cut the distance among vertexes to ϵ. However, note that if one considers three process executions we run into a problem. No matter how we subdivide the 2-simplex, there is always a path between p and q that includes r's middle vertex (marked by a darker color), and will only be added a single vertex per subsimplex in each subdivision phase. Thus, the distance among outputs on this path can be cut by at most a half in each iteration, hence our tight $\log_2$ lower bound.

Of course, the upper and lower bound proofs need not mention the actual executions and all we need to do is argue about the geometry of the inputs and outputs and then apply our main complexity theorem.

PROOF. We first establish the lower bound. Let $\mathcal{P}$ be a protocol that solves approximate agreement with worst case complexity k_S on S, where S is any input simplex of dimension $n > 0$. Then the asynchronous time complexity theorem states that there is some mapable non-uniform chromatic subdivision $\widetilde{\mathcal{X}}^k(\mathcal{I})$, with level k_S on S. We will show that $k_S > \left\lfloor \log_d \frac{|S|}{\epsilon} \right\rfloor$. The proof uses the following lemma.

LEMMA 5.2. *Let $l \leq k$. Label the vertices of $\widetilde{\mathcal{X}}^l(S)$ with real numbers in a way that agrees with the intial value labeling of S, and let l_S be the level of $\widetilde{\mathcal{X}}^l(S)$. Then*

$$|\widetilde{\mathcal{X}}^l(S)| \geq \frac{|S|}{d^{l_S}}$$

PROOF. Suppose for simplicity of argument that $l = l_S$. We first give the proof for the case of two processes where $d = 3$. By definition of $|S|$, there is a 1-simplex $U = (\vec{u_0}, \vec{u_1})$ in $\mathcal{S}$ such that $|U| = |S|$. [6] The complex $\widetilde{\mathcal{X}}^l(U)$ contains at most 3^l 1-simplices, denoted $U_1, \ldots, U_M$, where $M \leq 3^k$. These form a continuous path from $\vec{u}_0$ to $\vec{u}_1$, the endpoints of which are labeled with $val(\vec{u}_0)$ and $val(\vec{u}_1)$, respectively. So the best we can do is cut $|U|$ in 3^l pieces. The triangle inequality tells us that $|U| \leq \sum_{i=1}^{M} |U_i| \leq M \max_i |U_i| \leq 3^k \max_i |U_i|$. Hence $\max_i |U_i| \geq |U|/3^l = |S|/3^l$. The lemma follows, since $\max_i |U_i| \leq |\widetilde{\mathcal{X}}^l(S)|$.

We now prove the case where $d = 2$. We argue by induction on l. The case $l = 0$ is trivial. Now suppose the claim is true for $l-1$. By definition of $|\widetilde{\mathcal{X}}^{l-1}(S)|$, there is a 1-simplex $U = (\vec{u_0}, \vec{u_1})$ in $\mathcal{S}$ such that $|U| = |\widetilde{\mathcal{X}}^{l-1}(S)|$. U is a face of some 2-simplex $U' = (\vec{u_0}, \vec{u_1}, \vec{u}_2)$. Suppose that the next levels of non-uniform chromatic subdivision does not subdivide U completely. Then there is some 1-simplex T in the non-uniform subdivision of U' with $|T| \geq |U'|/2$. Since $|U'| = |\widetilde{\mathcal{X}}^{l-1}(S)|$ and $T| \leq |\widetilde{\mathcal{X}}^l(S)|$, the lemma follows by induction. Suppose the next level of subdivision does subdivide U' completely. Then this subdivision has an internal vertex $\vec{m}_2$, colored with $id(\vec{u}_2)$, and two neighboring 1-simplices $T_0 = (\vec{u}_0, \vec{m}_2)$ and $T_1 = (\vec{m}_2, \vec{u}_1)$. Then the triangle inequality tells us that $|U| \leq |T_0| + |T_1| \leq 2 \max_i |T_i|$ It follows that $|\widetilde{\mathcal{X}}^l(S)| \geq |\widetilde{\mathcal{X}}^{l-1}(S)|/2$. The lemma follows by induction. □

Suppose now that there exists a simplicial map $\mu : \widetilde{\mathcal{X}}^k(\mathcal{I}) \to \mathcal{O}$ such that, for all simplices T in $\widetilde{\mathcal{X}}^k(\mathcal{I})$, $\mu(T) \in \Gamma(carrier(T))$. We can associate this map with a labeling of the vertices in $\widetilde{\mathcal{X}}^k(\mathcal{I})$ as follows. Label each vertex $\vec{v}$ in $\widetilde{\mathcal{X}}^k(\mathcal{I})$ with $val(\mu(\vec{v}))$. This labeling agrees with the input value labeling of $\mathcal{I}$, since for any vertex $\vec{v}$, the task specification requires that for any simplex S_0 that contains $\vec{v}$, it must be the case that $\mu(\vec{v}) \in |S_0|$, where $|S_0|$ is the range of the value labels on S_0. Choose two neighboring simplices S_0 and S_1 containing $\vec{v}$ such that $|S_0| \cap |S_1| = val(\vec{v})$. It follows that $\mu(\vec{v}) = val(\vec{v})$. Now let T be any simplex in $\widetilde{\mathcal{X}}^k(\mathcal{I})$. By definition of μ, $\mu(T)$ is a simplex in $\mathcal{O}$, and hence $|\mu(T)| < \epsilon$. It follows that $|T| =$

[6] Note that we use script notation such as $\mathcal{S}$to denote the complex of faces of a simplex.

$|\mu(T)| < \epsilon$, and hence that $|\widetilde{\mathcal{X}}^k(\mathcal{I})| < \epsilon$, where $|\widetilde{\mathcal{X}}^k(\mathcal{I})|$ is equal to $max_{T \in \widetilde{\mathcal{X}}^k(\mathcal{I})}|T|$. Clearly, for any input simplex S, it follows that the labels on the restriction of $\widetilde{\mathcal{X}}^k(\mathcal{I})$ to S, $\widetilde{\mathcal{X}}^k(S)$, have range less than ϵ. The previous lemma then states that $\epsilon > |\widetilde{\mathcal{X}}^k(S)| \geq \frac{|S|}{d^{k_S}}$. We conclude that

$$k_S > \left\lfloor \log_d \frac{|S|}{\epsilon} \right\rfloor$$

To prove the upper bound we now construct a mapable non-uniform chromatic subdivision $\widetilde{\mathcal{X}}^k(\mathcal{I})$ of the input complex with level $k_S = \left\lfloor \log_d \frac{|S|}{\epsilon} \right\rfloor$ on each input simplex S. As argued above, the requirement that the subdivision be mapable is equivalent to saying that there is a vertex labeling of $\widetilde{\mathcal{X}}^k(\mathcal{I})$ that agrees with the initial value labeling of $\mathcal{I}$ with the additional property that $|\widetilde{\mathcal{X}}^k(\mathcal{I})| < \epsilon$. For each level of subdivision k, for each maximal simplex T in the current non-uniform subdivision $\widetilde{\mathcal{X}}^k(\mathcal{I})$, choose the maximal face S of T such that $|S| < \epsilon$ as stopped vertices, the rest are continuing. If the dimension of $\mathcal{C}$ is 1, label the new vertices in $\widetilde{\mathcal{X}}^{k+1}(\mathcal{C}) = \mathcal{X}(\mathcal{C})$ with $(2 \min val(\mathcal{C}) + \max val(\mathcal{C}))/3$ and $(\min val(\mathcal{C}) + 2 \max val(\mathcal{C}))/3$. Otherwise, label the new vertices with $(\min val(\mathcal{C}) + \max val(\mathcal{C}))/2$. It is clear from this construction that, at each step, for all simplices S in $\mathcal{I}$ we have that, if $|\widetilde{\mathcal{X}}^k(S)| > \epsilon$, then either $|\widetilde{\mathcal{X}}^{k+1}(S)| = |\widetilde{\mathcal{X}}^k(S)|/d)$, or $|\widetilde{\mathcal{X}}^{k+1}(S)| < \epsilon$. It follows that the level k_S of $\widetilde{\mathcal{X}}^k(\mathcal{I})$ on S is $\left\lfloor \log_d \frac{|S|}{\epsilon} \right\rfloor$. We conclude from the asynchronous time complexity theorem that there is a wait-free protocol that solves approximate agreement with worst case time complexity $\left\lfloor \log_d \frac{|S|}{\epsilon} \right\rfloor$ on input S where $d = 3$ for two processes and $d = 2$ for three or more.

6. RENAMING

In this section we use the asynchronous complexity theorem to analyze the complexity of the renaming task of Attiya et al. [3], in which at most $n+1$ processes are given unique input names taken from a large name space, and must choose unique output names taken from a smaller name space.

A protocol is *comparison-based* if the only operations a process can perform on process ids is to test for equality and order; that is, given two process ids P and Q, a process can test for $P = Q$, $P \leq Q$, and $P \geq Q$, but cannot examine the structure of the identifiers in any more detail (e.g., it cannot test whether P is prime). We will only consider comparison-based protocols in this section.

Let $\mathcal{A}$ and $\mathcal{B}$ be complexes where each vertex is labeled with a process id,and possibly with a value. $\mathcal{B}$ is a *recoloring* of $\mathcal{A}$ if there exists a bijective simplicial map (*not* color-preserving)

$$\rho : \mathcal{A} \rightarrow \mathcal{B}$$

that is (1) order-preserving on process ids: if $id(\vec{u}) < id(\vec{v})$ then $id(\rho(\vec{u})) < id(\rho(\vec{v}))$, and (2) value-preserving: if $val(\vec{v})$ is defined then $val(\vec{v}) = val(\rho(\vec{v}))$.

THEOREM (TIME COMPLEXITY FOR COMPARISON-BASED PROTOCOLS). *A decision task $\langle \mathcal{I}^n, \mathcal{O}^n, \Gamma \rangle$ has a wait-free protocol in the IIS model with worst case time complexity k_{S^m} on input S^m iff there is a mapable non-uniform iterated chromatic subdivision $\widetilde{\mathcal{X}}^k(\mathcal{I}^n)$ with level k_{S^m} on S^m such that any recoloring $\rho : \widetilde{\mathcal{X}}^k(S_0^m) \rightarrow$*

$\widetilde{\mathcal{X}}^k(S_1^m)$ *induces a recoloring* $\rho' : \mu(\widetilde{\mathcal{X}}^k(S_0^m)) \to \mu(\widetilde{\mathcal{X}}^k(S_1^m))$ *such that for every face* T *of* S_0^m, $\rho'(\mu(\widetilde{\mathcal{X}}^k(T))) = \mu(\widetilde{\mathcal{X}}^k(\rho(T)))$

The additional condition captures the notion that the behavior of comparison-based protocols does not change if processes are renamed in an order-preserving way. The proof of this theorem is almost identical to the proof of Theorem 4.1, except that it is necessary to check at each step that the equivalence-under-recoloring property continues to hold.

THEOREM 6.2. *The complexity of solving* $(n+1)(n+2)/2$*-renaming in the comparison-based non-uniform IISS model is* 1.

The following corollary follows immediately since we can implement ISS objects in time $\Theta(n)$ in the atomic snapshot model and time $\Theta(n^2)$ in the read-write model.

COROLLARY 6.3. *Let* $\{x_0, \ldots, x_n\}$ *be a set of inputs to the* $(n+1)(n+2)/2$*-renaming problem. The complexity of solving* $(n+1)(n+2)/2$*renaming is at most* $O(n)$ *in the atomic snapshot model and at most* $O(n^2)$ *in the read-write model.*

PROOF. (Of Theorem 6.2) That 1 is a lower bound on the complexity of solving renaming follows from the fact that the symmetry requirement of renaming prevents each processor from simply deciding on its own value, which forces each process to take at least one step before deciding. We now construct a simplicial map from the standard chromatic subdivision $\mathcal{X}(\mathcal{I})$ to the output complex $\mathcal{O}$ of renaming that agrees with the task specification. Let $S = (\vec{s}_0, \ldots, \vec{s}_n)$ be any simplex in $\mathcal{I}$. We now specify the map μ. First, for all $\vec{s}_i$ in $\mathcal{X}(skel^0(S))$ ($skel^m(C)$ for any simplex or complex C is the complex of its faces of dimension at most m), we let $\mu(\vec{s}_i) = \langle id(\vec{s}_i), 1\rangle$. Now suppose μ has been specified for all vertices that lie on $\mathcal{X}(skel^{m-1}(S))$ such that $val(\mu(\mathcal{X}(skel^{m-1}(S)))) = \{1, \ldots, m(m+1)/2\}$. We now show how to define μ for the extra vertices in $\mathcal{X}(skel^m(S))$. Let S^m be any m-face of S. $\mathcal{X}(S^m)$ contains $m+1$ vertices not in $\mathcal{X}(skel^m(S))$ that lie on an m-simplex $T^m = (\vec{t}_0, \ldots, \vec{t}_m)$. Let π be a permutation of $\{0, \ldots, m\}$ such that $\vec{t}_{\pi(i)}$ has the ith largest process id. We define $\mu(\vec{t}_i) = \left\langle id(\vec{t}_i), \pi(i) + m(m+1)/2 + 1\right\rangle$. It is clear that the map defined in this way is simplicial from $\mathcal{X}(S)$ to $\mathcal{O}$, and that for all simplices T in $\mathcal{X}(S)$, we have that $\mu(T) \in \Gamma(carrier(T))$, as required. Let $\rho : \widetilde{\mathcal{X}}^k(S) \to \widetilde{\mathcal{X}}^k(S')$ be and recoloring of S. Then since ρ is simplicial and preserves the order of process ids , it maps the subdivided m-faces of S to the subdivided m-faces of S'. For any m, the image of μ on the vertices in the subdivided m-skeleton that are not in the subdivided $m-1$-skeleton is disjoint from its image on the $m-1$-skeleton. Moreover, the map is defined solely in terms of the ordering of the ids of these vertices. It follows that ρ induces a recoloring $\rho' : \mu(\widetilde{\mathcal{X}}^k(S)) \to \mu(\widetilde{\mathcal{X}}^k(S'))$ that satisfies the conditions of Theorem 6.1. It follows that there is a renaming protocol of complexity 1. □

REFERENCES

[1] J. Aspnes, M. P. Herlihy, Wait-Free Data Structures in the Asynchronous PRAM Model. *Proceedings of the 3rd Annual ACM Symposium on Principles of Distributed Computing*, pages 377–408, July 1991. Also appeared as technical report.

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

[3] H. Attiya, A. Bar-Noy, D. Dolev, D. Peleg, and R. Reischuk. Renaming in an asynchronous environment. *Journal of the ACM*, July 1990.

[4] H. Attiya, N. Lynch and N. Shavit, Are Wait-Free Algorithms Fast? In *Journal of the ACM*, Vol. 41, No. 4 (July 1994), pages 725-763.

[5] H. Attiya and S. Rajsbaum. A combinatorial topology framework for wait-free computability. Preprint, 1995.

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

[7] E. Borowsky and E. Gafni. Immediate Atomic Snapshots and Fast Renaming. *Proceedings of the 12th Annual ACM Symposium on Principles of Distributed Computing*, pages 41–51, August 1993.

[8] E. Borowsky and E. Gafni. Generalized flp impossibility result for t-resilient asynchronous computations. In *Proceedings of the 1993 ACM Symposium on Theory of Computing*, May 1993.

[9] E. Borowsky and E. Gafni. A Simple Algorithmically Reasoned Characterization of Wait-free Computations. *Draft*, July 1996.

[10] E. Borowsky and E. Gafni. The set consensus hierarchy. Unpublished manuscript, November 1993.

[11] S. Chaudhuri. Agreement is harder than consensus: set consensus problems in totally asynchronous systems. In *Proceedings of the Ninth Annual ACM Symosium on Principles of Distributed Computing*.

[12] S. Chaudhuri, M.P. Herlihy, N. Lynch, and M. Tuttle. Tight Bounds for k-Set Agreement. In *Proceedings of the 33rd ACM Symposium on Foundations of Computer Science*, October 1993.

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

[14] M. Fischer, N.A. Lynch, and M.S. Paterson. Impossibility of Distributed Commit with one faulty process. Journal of the ACM,32(2), April 1985.

[15] E. Gafni and E. Koutsoupias. 3-processor tasks are undecidable. PODC 95, Brief announcement, page 271.

[16] M. Moir and J. Garay. Long-Lived Renaming Improved and Simplified, *Proc. 10th International Workshop on Distributed Algorithms (WDAG '96)*, LNCS (1151), Springer-Verlag, pp. 287-303, Bologna, Italy, October 1996.

[17] M. P. Herlihy. Wait-Free Synchronization. *ACM Transactions on Programming Languages and systems,* Vol.. 11, No.1, Jan 1991, Pages 124-129.

[18] M. Herlihy and S. Rajsbaum. Algebraic spans. *Proceedings of the 14th Annual ACM Symposium on Principles of Distributed Computing*, pages 90-99, August 1995.

[19] M. P. Herlihy and S. Rajsbaum. Set Consensus Using Arbitrary Objects. In *Proceedings of the 12th Annual ACM Symposium on Principles of Distributed Systems*, August 1994, Los Angeles.

[20] M.P. Herlihy and S. Rajsbaum. On the Decidability of Distributed Decision Problems, To appear in STOC'97.

[21] M.P. Herlihy and N. Shavit. The asynchronous computability theorem for t-resilient tasks. In *Proceedings of the 1993 ACM Symposium on Theory of Computing*, Pages 111–120, May 1993. Full version available as a DEC technical report.

[22] M. P. Herlihy and N. Shavit. A Simple Constructive Computability Theorem for Wait-free Computation.In *Proceedings of the 26th Annual Symposium on Theory of Computing*, Pages 243-252, May 23-25, 1994.

[23] M. P. Herlihy and N. Shavit. The Topological Structure of Asynchronous Computability. Technical Report, Submitted for Journal publication, Brown University, 1995.

[24] N.A. Lynch And M.R. Tuttle. An Introduction To Input/Output Automata. MIT/LCS/TM-373, MIT Laboratory For Computer Science, Nov 1988.

[25] Nancy Lynch and Sergio Rajsbaum. On the Borowsky-Gafni simulation algorithm. In *Proceedings of the Fourth ISTCS Israel Symposium on Theory of Computing and Systems*, pages 4–15, Jerusalem, Israel, June 1996. IEEE Computer Society. Also, short version appears in *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, Philadelphia, PA, page 57, May 1996.

[26] J.R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley, Cambridge, 1996.

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

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

7. APPENDIX

This section presents the missing proofs in the article body. We begin with a formal specification of Browsky and Gafni's Immediate Snapshot object.

7.1 Immediate Snapshots

Formally, we can specify IS objects as I/O automata [24]. Let D be any data type (set of values). Let $\perp$ be some value not in D. Define $\vartheta(D)$ to be the data type $(D \cup \{\perp\})^{n+1}$, the set of all $n+1$-arrays whose entries are either an element of D or $\perp$. We index $\vartheta(D)$ using the numbers in $\mathcal{Z}_{n+1}$. The IS automaton for $n+1$ processes and data type D, referred to as IS_D^{n+1}, is then defined as follows.

Signature:

Input:
$inv\_writeread(v)_i,\ v \in D$

Output:
$ret\_writeread(S)_i,\ S \in \vartheta(D)$

Internal:
$update_{\mathcal{U}},\ \mathcal{U} \subseteq \mathcal{Z}_{n+1}$

States:

$memory \in \vartheta(D)$ for some i, initially $(\perp, .., \perp)$
$inv\_value \in \vartheta(D)$ for some i, initially $(\perp, .., \perp)$
$ret\_value \in \vartheta^2(D)$ for some i, initially $(\perp, .., \perp)$
$interface \in \{inv, ret, \perp\}^{n+1}$, initially $(\perp, .., \perp)$

Transitions:

$inv\_writeread(v)_i$
Effect:
$inv\_value(i) := v$
$interface(i) := inv$

$update_{\mathcal{U}}$
Precondition:
$\forall i \in \mathcal{U} : interface(i) = inv$
Effect:
$\forall i \in \mathcal{U} : memory(i) := v$
$\forall i \in \mathcal{U} : ret\_value(i) := memory$
$\forall i \in \mathcal{U} : interface(i) := ret$

$ret\_writeread(S)_i$
Precondition:
$interface(i) = ret$
$memory(i) = S$
Effect:
$interface(i) := \perp$
$memory(i) := \perp$

Fig. 6. The Immediate Snapshot Specification

7.2 Definition and proof of the standard chromatic subdivision

Our definition of the standard chromatic subdivision taken from [23] is rather ad hoc. In this section, we will fill in this gap by providing a complete definition of the standard chromatic subdivision, along with the missing proof that it is in fact a subdivision.

Let $\mathcal{K}^n$ be a pure n-dimensional chromatic complex, where the colors are the numbers in $\mathcal{Z}_{n+1}$. Label each vertex $\vec{v}$ in $\mathcal{K}^n$ with $\langle i, \vec{v}\rangle$, where i is the color of $\vec{v}$.

In order to define the standard chromatic subdivision of $\mathcal{K}^n$, we inductively define a sequence of subdivisions $\mathcal{L}_p$ of the skeletons of $\mathcal{K}^n$, where $0 \leq p \leq n$ as follows. Let $\mathcal{L}_0 = skel^0(\mathcal{K}^n)$. Now suppose that $\mathcal{L}_{p-1}$ is a chromatic subdivision of the $p-1$-skeleton of $\mathcal{K}^n$, and that each vertex $\vec{v}$ in $\mathcal{L}_{p-1}$ is labeled $\langle i, S^{q_i}\rangle$, where S^{q_i} is some simplex in $skel^{p-1}(\mathcal{K}^n)$ such that $T^r = (\vec{t_0}, \ldots, \vec{t_r})$ is a simplex in $\mathcal{L}_{p-1}$ iff $ids(T^r) \subseteq ids(carrier(T^r)$, and for all $1 \leq i,j \leq r$, $id(t_i) \neq id(t_j)$, and the follwing conditions hold:

—$id(\vec{t_i}) \in ids(val(\vec{t_i}))$
—$val(\vec{t_i})$ is a face of $val(\vec{t_j})$ or vice versa
—$id(\vec{t_j}) \in ids(val(\vec{t_i})) \Rightarrow val(\vec{t_j})$ is a face of $val(\vec{t_i})$

Let $S^p = (\vec{s_0}, \ldots, \vec{s_p})$ be a p-simplex in K^n. The set $Bd(S^p)$ is the polytope of a subcomplex of the $p-1$-skeleton of K, and hence of a subcomplex of $\mathcal{L}_{p-1}$, which we denote $\mathcal{L}_{Bd(S^p)}$. Let $\vec{b}$ be the barycenter of S^p, and let δ be some positive real number such that $0 < \delta < 1$. For each $1 \leq i \leq p$, define $\vec{m_i}$ to be the point $(1+\delta)\vec{b} - \delta\vec{s_i}$. These points are called the midpoints of S^p. Label $\vec{m_i}$ with $\langle i, S^p\rangle$. Let M_{S^p} be the set of midpoints of S^p. We define $\mathcal{L}_{S^p}$ to be the union of $\mathcal{L}_{Bd(S^p)}$ and all the faces of all chromatic p-simplices $T^p = (\vec{t_0}, \ldots, \vec{t_p})$, such that for all $1 \leq i, j \leq p$: $\vec{t_i} \in skel^0(\mathcal{L}_{Bd(S^p)}) \cup M_{S^p}$, and the following conditions hold:

—$id(\vec{t_i}) \in ids(val(\vec{t_i}))$
—$val(\vec{t_i})$ is a face of $val(\vec{t_j})$ or vice versa
—$id(\vec{t_j}) \in ids(val(\vec{t_i})) \Rightarrow val(\vec{t_j})$ is a face of $val(\vec{t_i})$

We now define $\mathcal{L}_p$ to be the complex consisting of the union of the complexes $\mathcal{L}_{S^p}$, as S^p ranges over all the p-simplices of $\mathcal{K}^n$. The Appendix includes a proof of the following lemma that states that this structure make sense mathematically, that is, that it is in fact a subdivision of the p-skeleton of $\mathcal{K}^n$.

Lemma 7.1. *For all $0 \leq p \leq n$, $\mathcal{L}_p$ is a chromatic subdivision of $skel^p(\mathcal{K}^n)$.*

Proof. We argue by induction. The case $p = 0$ is trivial. So suppose $p > 0$, and suppose the claim holds for $\mathcal{L}_0, \ldots, \mathcal{L}_{p-1}$. We will first prove that $\mathcal{L}_p$ is a chromatic simplicial complex. To that end, we prove the following auxiliary lemma.

Lemma 7.2. *For all p-simplices S^p in $\mathcal{K}^n$, $\mathcal{L}_{S^p}$ is a chromatic simplicial complex.*

Proof. We must show that $\mathcal{L}_{S^p}$ is closed under containment and intersection. Let U^q be a simplex in $\mathcal{L}_{S^p}$, and let V^r be a face of T^q, where $0 \leq r \leq q \leq p$. If U^q is in $\mathcal{L}_{Bd(S^p)}$, then so is V^r, since $\mathcal{L}_{Bd(S^p)}$ is a complex (since $\mathcal{L}_p$ is a subdivision and hence a complex by assumption). Hence V^r is in $\mathcal{L}_{S^p}$. Suppose U^q is not contained in $\mathcal{L}_{Bd(S^p)}$. Then U^q must be the face of a p-simplex T^p as described above. By definition of $\mathcal{L}_{S^p}$, all the faces of T^p, and hence all faces of U^q, must be in $\mathcal{L}_{S^p}$. It follows that $\mathcal{L}_{S^p}$ is closed under containment.

Let $U_1^{q_1}$, $U_2^{q_2}$ be simplices in $\mathcal{L}_{S^p}$, and suppose their intersection, denoted V^r, is nonempty. If $U_1^{q_1}$, $U_2^{q_2}$ are both in $\mathcal{L}_{Bd(S^p)}$, it follows immediately that V^r is in $\mathcal{L}_{Bd(S^p)}$ and hence in L_{S^p}. Similarly, if $U_1^{q_1}$ is in $\mathcal{L}_{Bd(S^p)}$ but $U_2^{q_2}$ is not, then $V^r = U_1^{q_1} \cap U_2^{q_2} = U_1^{q_1} \cap (U_2^{q_2} \cap |\mathcal{L}_{Bd(S^p)}|)$. Note that $U_2^{q_2} \cap |\mathcal{L}_{Bd(S^p)}|$ is a simplex in $\mathcal{L}_{Bd(S^p)}$, since all the criteria given above are satisfied. Hence it follows that V^r is in $\mathcal{L}_{Bd(S^p)}$, and hence in $\mathcal{L}_{S^p}$. If neither $U_1^{q_1}$ nor $U_2^{q_2}$ is in $\mathcal{L}_{Bd(S^p)}$, then since all faces of $U_1^{q_1}$ and $U_2^{q_2}$ are in $\mathcal{L}_{S^p}$, then so is V^r. It follows that $\mathcal{L}_{S^p}$ is closed under intersection, and hence is a simplicial complex. That $\mathcal{L}_{S^p}$ is chromatic follows from the fact that we only include chromatic simplices in $\mathcal{L}_{S^p}$ in our construction. Note that $\mathcal{L}_{p-1}$ and hence $\mathcal{L}_{Bd(S^p)}$ are chromatic by assumption. □

Notice that for all distinct p-simplices S^p, T^p we have that $|\mathcal{L}_{S^p}| \cap |\mathcal{L}_{T^p}| = S^p \cap T^p$, which is a simplex in $skel^{p-1}(\mathcal{K}^n)$, and hence is the polytope of a subcomplex of $\mathcal{L}_{p-1}$, and hence of both $\mathcal{L}_{S^p}$ and $\mathcal{L}_{T^p}$. It follows that $\mathcal{L}_p$ is a simplicial complex [26]. It remains to show that $\mathcal{L}_p$ is a chromatic subdivision. To this end, we must first show that every simplex in $\mathcal{L}_p$ is contained in some simplex in $skel^p(\mathcal{K}^n)$, and that every simplex in $skel^p(\mathcal{K}^n)$ is the union of finitely many simplices in $\mathcal{L}_p$. Now, it is clear from our construction that any simplex T_q in $\mathcal{L}_p$ is contained in some simplex S^p in $skel^p(K^n)$. Also, since for all simplices S^p in $skel^p(K^n)$, the set of midpoints is finite, and $\mathcal{L}_{p-1}$ is a subdivision of $skel^{p-1}(\mathcal{K}^n)$ by assumption, it follows that S^p is the union of finitely many simplices in $\mathcal{L}_p$. Hence $\mathcal{L}_p$ is a subdivision. This subdivision is chromatic, since $\mathcal{L}_{p-1}$ is chromatic by assumption, and since the colors used to color the midpoints of any simplex S^p are exactly the colors used to color S^p.

We are now ready to give our definition of the standard chromatic subdivision of a complex $\mathcal{K}^n$.

Definition 7.3. *The* standard chromatic subdivision *of* $\mathcal{K}^n$*, denoted* $\mathcal{X}(\mathcal{K}^n)$*, is the complex* $\mathcal{L}_n$*.*

7.3 Proof of the non-uniform chromatic subdivision

Proof. (Of Lemma 3.2) We will prove that, for any complex $\mathcal{K}^n$, any non-uniform chromatic subdivision $\widetilde{\mathcal{X}}^1(\mathcal{K}^n)$ of level 1 as defined above is a chromatic subdivision of $\mathcal{K}^n$. Since the chromatic subdivision relation is transitive, it follows that $\widetilde{\mathcal{X}}^k(\mathcal{K}^n)$ is a chromatic subdivision of $\mathcal{K}^n$ for any $k \geq 0$.

We first show that $\widetilde{\mathcal{X}}^1(\mathcal{K}^n)$ is a chromatic simplicial complex. Let T^n be any maximal simplex in $\mathcal{K}^n$. Let $\mathcal{D} = (\mathcal{S} \cdot \mathcal{X}(\mathcal{C})$. Then, since $\mathcal{X}(\mathcal{C})$ is a subdivision of $\mathcal{C}$ by the previous lemma, we have that $\mathcal{D}$ is a chromatic complex, since $ids(\mathcal{C}) \cap ids(\mathcal{S}) = \emptyset$, and starring two chromatic complexes that share no colors give rise to a chromatic simplicial complex [26]. Now, for any pair of intersecting maximal simplices T_i^n, T_j^n, we have that $|\mathcal{D}_i| \cap |\mathcal{D}_j|$ is the polytope of a subcomplex of both $\mathcal{D}_i$ and $\mathcal{D}_j$, since we made sure that $\mathcal{C}_i \cap \mathcal{T}_j \subseteq \mathcal{C}_j$ and $cC_i \cap \mathcal{S}_j = \emptyset$. Hence $\widetilde{\mathcal{X}}^1(\mathcal{K}^n)$ is a chromatic simplicial complex.

It is clear from our construction that any simplex in $\widetilde{\mathcal{X}}^1(\mathcal{K}^n)$ is contained in some simplex T^n in $\mathcal{K}^n$. It remains to be shown that any simplex U^m in $\mathcal{K}^n$ is the union of finitely many simplices in $\widetilde{\mathcal{X}}^1(\mathcal{K}^n)$. U^m is a face of some maximal T^n in $\mathcal{K}^n$. By construction, $\widetilde{\mathcal{X}}^1(T^n)$ contains finitely many simplices, and hence so does the subcomplex $\widetilde{\mathcal{X}}^1(U^m)$. The union of the simplices in $\widetilde{\mathcal{X}}^1(U^m)$ equals U^m, and hence U^m is the union of finitely many simplices. It follows that $\widetilde{\mathcal{X}}^1(\mathcal{K}^n)$ is a subdivision. We have already shown that $\widetilde{\mathcal{X}}^1(\mathcal{K}^n)$ is a chromatic complex. In order to establish that it is a chromatic subdivision, we must show that for all simplices S^m in $\widetilde{\mathcal{X}}^1(\mathcal{K}^n)$, $ids(S^m) \subseteq ids(carrier(S^m))$. Now, S^m is contained in the restriction of $\widetilde{\mathcal{X}}^1(\mathcal{K}^n)$ to $carrier(S^m)$, and so $ids(S^m) \subseteq ids(\widetilde{\mathcal{X}}^1(carrier(S^m)))$. We claim that $ids(\widetilde{\mathcal{X}}^1(carrier(S^m))) \subseteq ids(carrier(S^m))$. Let $\mathcal{C}$ be the complex defined by the continuing vertices in $carrier(S^m)$, and $\mathcal{S}$ be the complex defined by the stopped vertices. Then the complex of faces of $carrier(S^m)$ is equal to $\mathcal{C} \cdot \mathcal{S}$. It is clear that $ids(\widetilde{\mathcal{X}}^1(\mathcal{S})) = ids(\mathcal{S}) \subseteq ids(\mathcal{S})$, and it follows from lemma 4.1 that $ids(\widetilde{\mathcal{X}}^1(\mathcal{C})) = ids(\mathcal{X}(\mathcal{C})) \subseteq ids(\mathcal{S})$. The claim follows, since $ids(\widetilde{\mathcal{X}}^1(carrier(S^m))) = ids(\widetilde{\mathcal{X}}^1(\mathcal{C})) \cup ids(\widetilde{\mathcal{X}}^1(\mathcal{S}))$. We conclude that $\widetilde{\mathcal{X}}^1(\mathcal{K}^n)$ is a chromatic subdivision of $\mathcal{K}^n$. □

7.4 The proof of our main theorem

We now give a proof of our main asynchronous time complexity theorem. We will first define the concept of a protocol complex in the non-uniform IIS model, and show that the set of such complexes is equal to the set of non-uniform chromatic subdivisions of the associated input complex.

Given a decision task $\langle \mathcal{I}^n, \mathcal{O}^n, \Gamma \rangle$ and a solution protocol $\mathcal{P}$ of worst case time complexity k, we define the corresponding uninterpreted protocol complex of the $n+1$-process IIS model, denoted $\mathcal{P}^k$, as follows: Each vertex $\vec{v} \in \mathcal{P}^k$ is labeled with a process ID and a local state such that there is some execution α of the protocol in which process $id(\vec{v})$ halts with local state $val(\vec{v})$. A simplex $T^m = (\vec{t_0}, \ldots, \vec{t_m})$ is in $\mathcal{P}^k$ if there is an execution α of the protocol in which each process $id(\vec{t_i})$ halts with local state $val(\vec{t_i})$ for all $0 \leq i \leq m$. The subcomplex of $\mathcal{P}^k$ generated by the executions that start from $\mathcal{J}^m$ is denoted $\mathcal{P}^k(\mathcal{J}^m)$.

LEMMA 7.4. *The protocol complex $\mathcal{P}^1$ of any non-uniform IS protocol of time complexity* 1 *with input complex $\mathcal{I}$ is equal to some non-uniform chromatic subdivision $\widetilde{\mathcal{X}}^1(\mathcal{I}^n)$.*

PROOF. We will establish a one-to-one correspondence between the set of protocol complexes on $\mathcal{I}$ of worst case time complexity 1 and the set of non-uniform chromatic subdivisions of $\mathcal{I}$. We have already described the process of generating a non-uniform chromatic subdivision in definition 3.1. Here we show how to generate the protocol complex $\mathcal{P}^1$ of any protocol $\mathcal{P}$ of worst case time complexity 1.

Consider any maximal input simplex T in $\mathcal{I}$. Some of the processors will (provided that they participate), decide on their input values, while others will access the IS object. These disjoint sets of vertices span two disjoint subcomplexes $\mathcal{S}$ and $\mathcal{C}$ of the complex $\mathcal{T}$ of faces of T such that $\mathcal{C} \cdot \mathcal{S} = \mathcal{T}$. Since any one process must, upon seeing a given input, either access the IS object or not, we have that, for all maximal simplices T_i, T_j in $\mathcal{I}$

—$\mathcal{C}_i \cap \mathcal{T}_j \subseteq \mathcal{C}_j$
—$\mathcal{C}_i \cap \mathcal{S}_j = \emptyset$

Since $\mathcal{P}$ has worst case time complexity 1, there must be at least one simplex T in $\mathcal{I}$ for which the set of continuing processes is nonempty. We refer to the vertices in $\mathcal{S}$ as stopped, and those in $\mathcal{C}$ as continuing. The protocol complex $\mathcal{P}^1(\mathcal{S})$ is clearly equal to $\mathcal{S}$, since any subset of the processes in $\mathcal{S}$ may participate. Since the processes in $\mathcal{C}$ are completely oblivious of those in $\mathcal{S}$ and vice versa, the protocol complex on $\mathcal{T}$ is equal to all combinations of simplices in $\mathcal{P}^1(\mathcal{S})$ with those in $\mathcal{P}^1(\mathcal{C})$, that is, $\mathcal{P}^1(\mathcal{T}) = \mathcal{P}^1(\mathcal{S}) \cdot \mathcal{P}^1(\mathcal{C})$. It remains to determine $\mathcal{P}^1(\mathcal{C})$. We use the following lemma.

LEMMA 7.5. *Let $\mathcal{C}$ be an input complex in the uniform 1-shot IS model. The corresponding protocol complex equals $\mathcal{X}(C)$.*

PROOF. Consider any input simplex S in $\mathcal{C}$, not necessarily maximal, in which process i starts with input $v_i \in V_0$. Let α be an execution in the 1-shot ISS model with these inputs. Suppose all the processes in $ids(S)$ participate. Let D be the set of processes that decide in α. We assume D is nonempty. Each process $i \in D$ decides on a value $S_i^1 \in V_1$, since the $ret\_writeread(v)_i$ action returns a snapshot containing a subset of the inputs entered before the $update_{\mathcal{U}}$ action in which v_i is first written into *memory*. This value is an encoding of a subsimplex of S. We must show that $T = \{\langle i, S_i^1\rangle \mid i \in D\}$ is a $|D|-1$-simplex in $\mathcal{X}(S)$, and hence in $\mathcal{X}(\mathcal{C})$.

We say that v_i is written to *memory* in action $update_{\mathcal{U}}$ if $memory[i]$ was equal to $\perp$ before $update_{\mathcal{U}}$, but is equal to v_i after this action. Note that no cell in *memory* is ever reset, so once i's input value is written to *memory*, $memory[i]$ will not be reset during the rest of the execution. In the 1-shot ISS model here, any process' input value v_i is written to *memory* at most once. It follows that for any two different actions $update_{\mathcal{U}}$, $update_{\mathcal{U}'}$, the index sets $\mathcal{U}, \mathcal{U}'$ are disjoint.

Since v_i must be written to *memory* before *memory* is copied to $ret\_value[i]$ (both these events occur, in the given order, in an $update_{\mathcal{U}}$ action), it follows that $i \in ids(S_i^1)$ for all i. Now suppose v_i is written to *memory* by the action $update_{\mathcal{U}_i}$, and v_j is written to *memory* by the action $update_{\mathcal{U}_j}$. Suppose $update_{\mathcal{U}_i}$ occurs after $update_{\mathcal{U}_j}$. Since no *memory* cells are ever reset, it follows that the *memory* version that is written to $ret\_value[j]$ during $update_{\mathcal{U}_j}$ is a prefix of the version that is written to $ret\_value[i]$ during $update_{\mathcal{U}_i}$, that is, if a cell equals $\perp$ in the *memory* version written to $ret\_value[i]$, then the same is true for the version written to $ret\_value[j]$. Hence $S_j^1 \subseteq S_i^1$. The case where $update_{\mathcal{U}_j}$ occurs after $update_{\mathcal{U}_i}$ is similar, and in this case we have $S_i^1 \subseteq S_j^1$. Finally, if v_i and v_j are written to *memory* by the same action $update_{\mathcal{U}_i} = update_{\mathcal{U}_j}$, then it follows that, since in $update_{\mathcal{U}_i}$ the values of all processes whose index is in $\mathcal{U}_i = \mathcal{U}_j$ are written to *memory* before any writes to ret_value are made, $S_i^1 = S_j^1$. Finally, suppose $j \in ids(S_i^1)$. This implies that $v_j = S_j^0$ was written to *memory* during $update_{\mathcal{U}_i}$, or in an earlier action $update_{\mathcal{U}_j}$. In either case, $S_j^1 \subseteq S_i^1$. Thus the criteria in the definition of the standard chromatic subdivision are satisfied, and we conclude that T is a $|D|-1$-simplex in $\mathcal{X}(S^n)$.

Let S^m be a simplex in $\mathcal{X}(\mathcal{I})$. Let $S^{m'} = carrier(S^m, \mathcal{I})$. We can write S^m as $\{\langle i, S_i\rangle \mid i \in M\}$ for some $M \subseteq \{0, \ldots, m'\}$, where $\forall i,j \in M : S_i \subseteq S^{m'}$, $i \in ids(S_i)$, $S_i \subseteq S_j$ or $S_j \subseteq S_i$, and $j \in ids(S_i) \Rightarrow S_j \subset S_i$. We must construct a corresponding execution α of the 1-shot ISS model, that is, an execution in which process i halts with decision value S_i for all $i \in M$. We proceed as follows. Partition the set M into a collection of nonempty *concurrency classes* of process indices, $\mathcal{U}_1$, $\ldots, \mathcal{U}_k$ for some k such that any two process indices i, j are in the same concurrency class iff $S_i = S_j$. We can define a total order $\prec$ on the set of concurrency classes as follows. Let $\mathcal{U}_x, \mathcal{U}_y$ be distinct concurrency classes. Then $\mathcal{U}_x \cap \mathcal{U}_y\ \emptyset$. Since both classes are nonempty, we can pick an element from each, say $S_i \in \mathcal{U}_x$ and $S_j \in \mathcal{U}_y$. By assumption, $S_i \neq S_j$. Then either $S_i \subset S_j$ or $S_j \subset S_i$. In the first case, let $\mathcal{U}_x \prec \mathcal{U}_y$, and in the second case, let $\mathcal{U}_y \prec \mathcal{U}_x$.

Now use this ordered partition to define a partition $\mathcal{U}'_1, \ldots, \mathcal{U}'_k$ of the set M' as follows. We partition the elements of $M \cap M'$ as before. For each concurrency class $\mathcal{U}$ of M, define a concurrency class $\mathcal{U}'$ of M' as follows. $\mathcal{U}'$ is the union of $\mathcal{U}$ and all $i \in M' - M$ such that $\mathcal{U}$ is the least concurrency class (as determined by $\prec$) such that $\forall j \in \mathcal{U},\ i \in S_j$. Note that this is a partition of all of M' since $S^{m'} = carrier(S^m, \mathcal{I})$. This partition gives us a new collection of concurrency classes $\mathcal{U}'_1, \ldots, \mathcal{U}'_k$.

We are now ready to construct α. First position the $update_{\mathcal{U}'_i}$ actions in increasing order according to the $\prec$ ordering. For each concurrency class $\mathcal{U}'_x$, position the $inv\_writeread(v_i)_i$ actions of all i such that $i \in \mathcal{U}'_x$ immediately before the $update_{\mathcal{U}'_x}$ action (their internal ordering does not matter). Similarly, position the $ret\_writeread(S_i)_i$ actions of all i such that $i \in \mathcal{U}'_x$ and $i \in M$ immediately after the $update_{\mathcal{U}'_x}$ action, but before the $inv\_writeread(v_j)_j$ actions associated with the next concurrency class. Processes i whose index is not in M' take no steps in α. Processes i whose index is not in M do not execute any $ret\_writeread$ actions. By construction, each deciding process i decides S_i in α, as required. □

We immediately conclude that the protocol complex on T is equal to $\mathcal{S} \cdot \mathcal{X}(\mathcal{C})$. The entire protocol complex, then, is simply the union

$$\mathcal{P}^1 = \bigcup_{T \in \mathcal{I}} \mathcal{S} \cdot \mathcal{X}(\mathcal{C})$$

It is clear that the generation procedure described is equivalent to the one given in definition 3.1, and hence that the set of protocol complexes of worst case time complexity 1 is equal to the set of non-uniform chromatic subdivisions of level 1. the lemma follows.

Lemma 7.6. *The protocol complex $\mathcal{P}^k$ on $\mathcal{I}$ of any protocol in the non-uniform IIS with worst case time complexity k is equal to some non-uniform iterated chromatic subdivision $\widetilde{\mathcal{X}}^k(\mathcal{I})$.*

Proof. By the previous lemma, we know that the claim is true for $k = 1$. The protocol complex $\mathcal{P}^k$ of a protocol complex of a non-uniform protocol with worst case time complexity k can be constructed by applying the procedure described in the previous lemma iteratively k times, in such a way that, at each step, none of the continuing vertices are part of the set of stopped vertices from the previous step. This requirement is necessary, since a process can stop and decide only once. It follows from the previous lemma that the proecdure for generating the set of protocol complexes corresponding to non-unifrom protocols with worst case time complexity k is equal to that for generating the set of non-uniform iterated chromatic subdivisions of level k. Hence these sets are equal, and the lemma follows. □

We now give the proof of Theorem 4.1.

Proof. (Of Theorem 4.1) Given a decision task $\langle \mathcal{I}^n, \mathcal{O}^n, \Gamma \rangle$. It follows immediately from the previous lemma that any non-uniform protocol complex $\mathcal{P}^k$ with worst case complexity k_{S^m} on input S^m and decision map $\delta : \mathcal{P}^k \to \mathcal{O}^n$ that agrees with Γ corresponds to a non-uniform iterated chromatic subdivision $\widetilde{\mathcal{X}}^k(\mathcal{I}^n)$ with level k_{S^m} on S^m together with a simplicial map $\mu : \widetilde{\mathcal{X}}^k(\mathcal{I}^n) \to \mathcal{O}^n$ that agrees with Γ, and vice versa. The theorem follows. $\square$