The Impossibility of Approximate Agreement on a Larger Class of Graphs

Shihao Liu
Department of Computer Science, University of Toronto, Canada

Abstract

Approximate agreement is a variant of consensus in which processes receive input values from a domain and must output values in that domain that are sufficiently close to one another. We study the problem when the input domain is the vertex set of a connected graph. In asynchronous systems where processes communicate using shared registers, there are wait-free approximate agreement algorithms when the graph is a path or a tree, but not when the graph is a cycle of length at least 4. For many graphs, it is unknown whether a wait-free solution for approximate agreement exists.

We introduce a set of impossibility conditions and prove that approximate agreement on graphs satisfying these conditions cannot be solved in a wait-free manner. In particular, the graphs of all triangulated d-dimensional spheres that are not cliques, satisfy these conditions. The vertices and edges of an octahedron is an example of such a graph. We also present a family of reductions from approximate agreement on one graph to another graph. This allows us to extend known impossibility results to even more graphs.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Theory of computation → Computability

Keywords and phrases Approximate agreement on graph, wait-free solvability, triangulated sphere

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2022.22

Funding This work is supported by the Natural Science and Engineering Research Council of Canada.

Acknowledgements I want to thank my advisor Faith Ellen for her advice and many helpful discussions. I would also like to thank the anonymous reviewers for their helpful comments.

1 Introduction

Agreement problems have been extensively studied in the field of distributed computing. In particular, the *consensus* problem [19] requires that processes agree on a single input value. When the system is asynchronous and processes may crash, Fischer, Lynch, and Paterson [12] showed that consensus is unsolvable. Since then, many variants of consensus with milder agreement requirements have been studied in asynchronous systems.

One such variant is *approximate agreement*, where instead of agreeing on a single value, processes must output values that are sufficiently close to one another. This problem was introduced by Dolev, Lynch, Pinter, Stark and Weihl [11] and is related to synchronizing clocks in a distributed system. Attiya, Lynch, and Shavit [8] considered the approximate agreement problem where the domain is $\mathbb{R}$ and processes are required to output values that are within distance ε of one another. They showed that this problem has step complexity $\Theta(\log n)$ using single-writer registers in the asynchronous shared-memory setting. Their bound does not depend on ε nor the size of the input domain. Using multi-writer registers, Schenk [21] showed that this problem has step complexity $O(\log(M/\varepsilon))$, where M is the largest magnitude of any input value. Here, the complexity does not depend on the number of processes in the system. Mendes, Herlihy, Vaidya and Garg [17] considered approximate agreement on $\mathbb{R}^d$. They showed that this problem has a solution that tolerates up to f Byzantine failures in an asynchronous completely-connected message-passing system if

26th International Conference on Principles of Distributed Systems (OPODIS 2022).
Editors: Eshcar Hillel, Roberto Palmieri, and Etienne Rivière; Article No. 22; pp. 22:1–22:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and only if $n > \max\{3f, (d+1)f\}$. More recently, Attiya and Ellen [7] gave a wait-free algorithm with $O(\log n(\log n + \log(S/\varepsilon)))$ step complexity solving approximate agreement on $\mathbb{R}^d$ using multi-writer registers, where S is the maximum distance between any two inputs. They also proved two lower bounds for the this problem, $\Omega\Big(\min\left\{\frac{\log M}{\log\log M}, \frac{\sqrt{\log n}}{\log\log n}\right\}\Big)$ and $\frac{1}{2}\log_{\sqrt{2}+1}(S/\varepsilon)$.

Approximate agreement on graphs is the natural discrete variant of approximate agreement. Here the domain is the vertex set of some connected graph. Herlihy, Kozlov, and Rajsbaum [13] viewed approximate agreement on graphs as robot convergence tasks.

Approximate agreement on a path is equivalent to approximate agreement on a closed interval in $\mathbb{R}$. For some other graphs, such as trees, there are wait-free algorithms to solve approximate agreement using registers [1]. For other graphs, such as cycles of length at least 4, no wait-free algorithms using only registers exist [3, 10]. However, for many graphs it is unknown whether a wait-free solution for approximate agreement exists using only registers.

Approximate agreement on graphs

On a connected undirected graph $G = (V, E)$ (known to all processes), each process p_i begins the *approximate agreement* problem with an input value $x_i \in V$. At the end of the computation, each process outputs a value $y_i \in V$ such that the following two conditions are satisfied:

- agreement: different output values are adjacent in G, and
- (clique) validity: if the inputs form a clique in G, then the set of outputs is a subset of the set of inputs.

Different validity conditions for approximate agreement on graphs have been considered. The *shortest path validity* condition [3] requires each output value to lie on some shortest path between two input values. It ensures that each output is a vertex in the smallest convex subgraph containing all input vertices. A subgraph $H \subseteq G$ is *convex* if, for any two vertices u and v in H, all shortest paths (in G) between u and v are contained in H. The *minimal path validity* condition [18] requires output values to lie on some chordless path between two input values. Since every shortest path is also a chordless path, minimal path validity generalizes shortest path validity. The *clique validity* condition [1] is also known as *1-gathering validity*, it is a generalization of both minimal path validity and shortest path validity. Thus, a proof that there is no wait-free algorithm for approximate agreement on a graph with clique validity implies that there is no wait-free algorithm for approximate agreement on that graph with the other two validity conditions. Likewise, an algorithm solving approximate agreement on a graph with shortest path validity also guarantees minimal path validity and clique validity.

Positive results

Alistarh, Ellen, and Rybicki [3] showed that approximate agreement with shortest path validity has a wait-free solution on any graph of radius one or any *nicely bridged* graph in a shared memory system where processes communicate using registers. The family of nicely bridged graphs includes all chordal graphs. They also presented an approximate agreement algorithm on any connected graph that tolerates at most one process crash failure in the same model. This shows that approximate agreement has a wait-free solution among two processes on any connected graph.

Nowak and Rybicki [18] gave an approximate agreement algorithm on chordal graphs in an asynchronous message-passing model with at most f Byzantine processes, provided the number of processes is greater than $(\omega(G) + 1)f$, where $\omega(G)$ is the size of the largest clique in G. However, their algorithm only guarantees minimal path validity.

With clique validity, Alcántara, Castañeda, Flores-Peñaloza, and Rajsbaum [1] showed that approximate agreement has a wait-free solution using only registers when the graph G is a tree or, more generally, when the *clique graph* of G is a tree. The clique graph of G is the graph (V', E'), where vertices in V' are cliques in the graph G, and $(v', u') \in E'$ if and only if $v' \cap u' \neq \emptyset$ in the graph G.

Negative results

Approximate agreement has no wait-free solution using only registers on a cycle of length at least 4. Castañeda, Rajsbaum, and Roy [10] showed this by giving a reduction from 2-set agreement. Later, Alistarh, Ellen, and Rybicki [3] gave a direct combinatorial proof using Sperner's lemma. They also used a reduction to show that approximate agreement has no wait-free solution on a graph G, if the vertices of G can be labelled using the set $\{0, 1, 2\}$, such that

- G contains no triangle with three different labels, and
- G contains a cycle C in which exactly one node has label 1 and its two neighbours in C have labels 0 and 2.

We call such a labelling an *AER impossibility labelling*. In particular, any cycle of length $c \geq 4$ has an AER impossiblity labelling.

Ledent [16] conjectured that approximate agreement is not solvable in a wait-free manner on any graph whose complex of cliques is not contractible. This includes graphs consisting of the nodes and edges of an octahedron and an icosahedron and, more generally, any triangulated d-dimensional sphere, for $d \geq 1$, that is not a clique. Note that a triangulated 1-dimensional sphere is a cycle and a triangulated 2-dimensional sphere is a connected planar graph in which every edge is shared by exactly two triangles.

Our contribution

In Section 4.1, we show that approximate agreement on the octahedron graph has no wait-free solution using only registers, for $n \geq 4$ processes. In Section 4.2, we extend this result to any graph that satisfies a new set of impossibility conditions provided there are sufficiently many processes. Any cycle of length at least 4 satisfies these impossibility conditions. More generally, these impossibility conditions are satisfied by any graph (except a clique) consisting of the nodes and edges of a triangulated d-dimensional sphere, for $d \geq 1$. This includes the octahedron graph, which we show does not have an AER impossibility labelling.

In Section 5, we describe a simple reduction from approximate agreement on one graph to approximate agreement on another graph. As an application of this reduction, we show that the impossibility of wait-free approximate agreement on the stellated octahedron graph can be derived from the impossibility of wait-free approximate agreement on the octahedron graph.

Alistarh, Ellen and Rybicki [4] showed that extension-based proofs cannot be used to prove the impossibility of approximate agreement on a 4-cycle. In Section 6, we briefly discuss our generalization showing that extension-based proofs cannot be used to prove the impossibility of wait-free approximate agreement on any connected graph.

2 Iterated immediate snapshot model

We focus our attention on computation in the full-information (non-uniform) iterated immediate snapshot model [15].

In this model, a set P of n processes communicate by accessing an infinite sequence of shared single-writer atomic snapshot objects. Each *single-writer atomic snapshot* object has n components and supports two atomic operations, **update** and **scan**. Initially, each component of each snapshot object contains the value $-$. An **update**(x) performed by process p_i on a snapshot object changes the value of its i-th component to x, while a **scan** returns the current value of each component.

At any given time, the *state* of a process p_i consists of its process identifier, its current *view* of the system, and a bit indicating whether it has just performed an update or a scan. Initially, the view of process p_i is just its input value and p_i is poised to access the first snapshot object. Each process accesses each snapshot object in the sequence at most twice. The first time process p_i accesses a snapshot object, it performs an **update**. At its next step, p_i performs a **scan** on the same snapshot object and changes its state depending on the result it received from the **scan**. In the full-information setting, whenever process p_i performs an **update**, it writes its entire computation history into the i-th component of the snapshot object, and whenever it performs a **scan**, it changes its view to be the result of the scan. After changing its state, process p_i applies a *decision map* δ to its current state. If the state of p_i is mapped to $\bot$, then p_i is poised to access the next snapshot object in the sequence. Otherwise, the state of p_i is mapped to an output value y, which p_i outputs, and p_i cannot take any more steps. A *protocol* is specified by the decision map δ used by every process. A protocol is *wait-free* if each process takes a finite number of its own steps before its state is mapped to an output value by δ.

A *configuration* consists of the state of each process. Note that, in the full-information setting, we can determine the contents of the snapshot objects from the states of every process. An *initial configuration* is a configuration where processes are in their initial states (and the snapshot objects have their initial values). A process is *active* in a configuration if δ maps the state of that process to $\bot$. Likewise, a process is *terminated* in a configuration if δ maps the state of that process to an output value. A *terminal configuration* is a configuration where all processes are terminated.

An *execution* from a configuration C is defined by an alternating sequence $C_0, Q_1, C_1, Q_2, C_2, \ldots$ of configurations and subsets of processes, beginning with the configuration $C_0 = C$, such that, for each $k \geq 0$, Q_{k+1} is an non-empty subset of processes that are poised to access the same snapshot object in configuration C_k. Configuration C_{k+1} is the result of the processes in Q_{k+1} each performing an **update**, and then each performing a **scan**, starting from configuration C_k. Each execution from C induces a *schedule* from C, which is the sequence $Q_1, Q_2, \ldots$ of subsets of processes in the execution. Since each process only updates its corresponding component of each snapshot object, an execution is completely specified by its starting configuration and its schedule. If α is a finite schedule from configuration C, we use $C\alpha$ to denote the configuration at the end of the execution that induces α. In this case, we say that $C\alpha$ is *reachable* from C via the schedule α. Note that if C is a terminal configuration, then the empty schedule is the only possible schedule from C.

An execution is *Q-only* if its schedule consists of subsets of processes in Q. The schedule of a Q-only execution is called a Q-only schedule. If each active process in Q is poised to perform **update** on the same snapshot object in C, then a *1-round Q-only schedule* from C is a Q-only schedule where each active process in Q appears exactly once. When $Q = P$, we simply call this schedule a *1-round schedule.* From a configuration where all active processes are poised to perform **update** on the same snapshot object, every 1-round Q-only schedule can be extended to a 1-round schedule by appending a 1-round $(P \setminus Q)$-only schedule. The resulting schedule is called a *1-round Q-first schedule* (i.e. all occurrences of processes in Q

occur before any occurrence of a process in $P \setminus Q$). For $k \geq 1$, a k-round Q-first schedule β starting from C is a sequence of 1-round Q-first schedules $\beta_1, ..., \beta_k$, where β_1 starts from C and β_i starts from $C\beta_1 ... \beta_{i-1}$, for $2 \leq i \leq k$. Note that each terminal configuration reachable from an initial configuration C_0 is also reachable from C_0 via a k-round schedule, for some $k \geq 0$. (See Lemma 4.4 from [2].) Thus, it suffices to only consider such schedules.

Let C be a configuration reachable from some initial configuration via a k-round Q-first schedule, where $\emptyset \neq Q \subseteq P$. Then the *partial configuration C' of C induced by Q* consists of the of states of the processes in Q. We use $\pi(C')$ to denote the set of processes Q whose states appear in the partial configuration C'. The partial configuration C' can be viewed as a configuration in a system with a smaller set of processes: Suppose β is a k-round Q-first schedule starting from configuration C. Let β' be the restriction of β to the processes in Q. Then β' is a schedule starting from the partial configuration C' of C induced by Q and each process in Q has the same state in $C'\beta'$ and $C\beta$. Note that, for each process in Q, components corresponding to processes in $P \setminus Q$ all have value $-$ in both $C'\beta'$ and $C\beta$.

Two (partial) configurations C and C' are *indistinguishable to a set of processes* Q if $Q \subseteq \pi(C) \cap \pi(C')$ and the state of each process in Q is the same in both configurations. Consider any 1-round Q-first schedule β from both C and C'. If C and C' are indistinguishable to Q, then $C\beta$ and $C'\beta$ are indistinguishable to Q. Since we are restricting attention to full-information protocols, the converse is also true. It follows that, if $C\beta = C\beta'$, then $\beta = \beta'$.

If $\mathbb{K}$ is a collection of (partial) configurations (not necessarily induced by the same set of processes), $C \in \mathbb{K}$ is a (partial) configuration, and $\emptyset \subseteq Q \subseteq \pi(C)$, then we say Q *identifies C in* $\mathbb{K}$ if, for every other (partial) configuration $C' \in \mathbb{K}$ such that $Q \subseteq \pi(C')$, at least one process in Q has a different state in C and C'. When the collection $\mathbb{K}$ is clear from context, we simply say Q identifies C.

For any $r \geq 0$ and for any (partial) configuration C reachable from some (partial) initial configuration via some r-round $\pi(C)$-only schedule, let $\chi(C, \delta)$ denote the set of all possible (partial) configurations reachable via 1-round $\pi(C)$-only schedules starting from C and let $\chi^k(C, \delta)$ denote the set of all possible (partial) configurations reachable via k-round $\pi(C)$-only schedules starting from C. For a collection $\mathbb{K}$ of (partial) configurations, define $\chi(\mathbb{K}, \delta) = \bigcup_{C \in \mathbb{K}} \chi(C, \delta)$ and $\chi^k(\mathbb{K}, \delta) = \bigcup_{C \in \mathbb{K}} \chi^k(C, \delta)$.

3 A Computational Version of Sperner's Lemma

Our main results in Section 4 rely on a variant of a classical combinatorial tool known as Sperner's lemma. The original topological proofs for the impossibility of wait-free set agreement [14, 20, 9] all used Sperner's lemma or equivalent formulations. Later, Attiya and Castañeda [5] proved the impossibility of set agreement using purely combinatorial techniques, without using topology. Their argument implicitly applied elements of Sperner's lemma directly on executions. More recently, Alistarh, Ellen, and Rybicki [3] gave a combinatorial proof for the impossibility of approximate agreement on cycles (of length at least 4) using a generalization of Sperner's lemma to convex polygons.

In this section, we generalize Sperner's lemma and rephrase it as a self-contained statement about executions in the iterated immediate snapshot model. It makes no explicit mention of topology. However, we note that it is equivalent to a formulation of Sperner's lemma for manifolds, phrased in terms of simplicial complexes and subdivisions, that appears in [13] as Lemma 9.3.4.

Consider a protocol among $n \geq 2$ processes in the iterated immediate snapshot model. Let $2 \leq m \leq n$ and let $\mathbb{H}$ be a collection of (partial) initial configurations such that $|\pi(C)| = m$ for each $C \in \mathbb{H}$. For each $C \in \mathbb{H}$ and each subset of processes $Q \subseteq \pi(C)$, denote by $I(C, Q)$

the set of input values of processes in Q in the (partial) configuration C. The *boundary* of $\mathbb{H}$, denoted $\mathcal{B}(\mathbb{H})$, is the collection of all pairs (C,Q), where $Q \subsetneq \pi(C)$ is a subset of $m-1$ processes and C is identified by Q in $\mathbb{H}$. In other words, for each pair $(C,Q) \in \mathcal{B}(\mathbb{H})$, there is no other (partial) configuration $C' \in \mathbb{H}$ such that $Q \subsetneq \pi(C')$ and each process in Q has the same state in C and C'.

Throughout the remainder of this section, we let t be the maximum number of non-empty rounds taken by the protocol in executions starting from (partial) initial configurations in $\mathbb{H}$. Let $\mathbb{T} = \chi^t(\mathbb{H}, \delta)$ be the collection of all (partial) terminal configurations reachable from (partial) initial configurations in $\mathbb{H}$.

Definition 1 is a computational analogue of a Sperner labelling. Immediately afterwards, we give an example with $n = m = 3$ to help explain the definition.

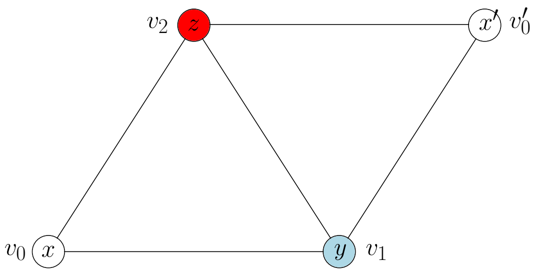
▶ **Definition 1.** *A collection $\mathbb{H}$ of (partial) initial configurations, each of which consists of the states of the same number of processes $m \geq 2$, satisfies the* computational Sperner conditions (CSC) *for the protocol if:*

- *CSC1: For each $(C,Q) \in \mathcal{B}(\mathbb{H})$, the processes in Q have different input values in C. In other words, $|I(C,Q)| = m-1$.*
- *CSC2: For each $(C,Q) \in \mathcal{B}(\mathbb{H})$, there are an odd number of pairs $(C',Q') \in \mathcal{B}(\mathbb{H})$ such that $I(C',Q') = I(C,Q)$.*
- *CSC3: For each $(C,Q) \in \mathcal{B}(\mathbb{H})$, for each subset $S \subseteq Q$, and for each S-first $\pi(C)$-only t-round schedule β starting from C, each process in S has output a value in $I(C,S)$ in the (partial) configuration $C\beta$.*
- *CSC4: For any $C \in \mathbb{H}$ and any subset $Q \subsetneq \pi(C)$ of $m-1$ processes, there is at most one other configuration $C' \in \mathbb{H}$ such that $Q \subsetneq \pi(C')$ and the configurations C and C' are indistinguishable to all processes in Q.*

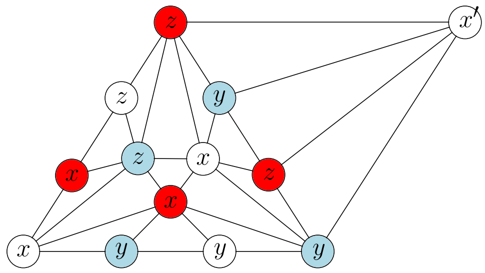
Let $\mathbb{H}$ be a collection consisting of two initial configurations C and C', where $\pi(C) = \pi(C') = \{p_0, p_1, p_2\}$ and C and C' are indistinguishable only to processes p_1 and p_2. Since p_0 has a different initial state in C and C', any subset of $\{p_0, p_1, p_2\}$ containing p_0 identifies C and C' in $\mathbb{H}$. Thus, $\mathcal{B}(\mathbb{H}) = \{(C, \{p_0, p_1\}), (C, \{p_0, p_2\}), (C', \{p_0, p_1\}), (C', \{p_0, p_2\})\}$. Let $x \neq x'$ be the input of process p_0 in configurations C and C'. Let v_0 and v_0' be its initial states in these two configurations. Let y and z be the inputs of processes p_1 and p_2 in both configurations and let v_1 and v_2 be their initial states. This is illustrated in Figure 1a, where the colors white, blue, and red represent the processes p_0, p_1, and p_2, respectively. The configurations $C = \{v_0, v_1, v_2\}$ and $C' = \{v_0', v_1, v_2\}$ are the two triangles. Since C and C' are indistinguishable to p_1 and p_2, these triangles share the edge $\{v_1, v_2\}$. The edges $\{v_0, v_1\}$ and $\{v_0, v_2\}$ in C and the edges $\{v_2, v_0'\}$ and $\{v_1, v_0'\}$ in C' form the boundary of this polygon.

CSC1 says that every edge on the boundary of the polygon has endpoints with different inputs, so, in the example, x, x', y, and z are all different. CSC2 says that each pair of inputs that labels an edge on the boundary labels an odd number of such edges. In the example, each such pair labels exactly one edge on the boundary. CSC4 is a technical requirement (analogous to the pseudomanifold property of a simplicial complex) that says each edge occurs in at most two triangles.

The set of partial terminal configurations $\mathbb{T}$ reachable from C and C' by a protocol in the iterated immediate snapshot model can be represented by a subdivision of the two triangles [15]. This is illustrated in Figure 1b, where each triangle in the subdivision represents a reachable terminal configuration. The subdivision of a vertex v_i, which is just a vertex, corresponds to the $\{p_i\}$-only execution from v_i. The subdivision of an edge $\{v_i, v_j\}$



(a) Triangles representing configurations C and C'.



(b) A subdivision of the two triangles representing terminal configurations reachable from C and C' by a protocol.

corresponds to the $\{p_i, p_j\}$-only executions from the partial configuration $\{v_i, v_j\}$. Here, the output of a process in a terminal configuration labels the vertex corresponding to its state in this configuration. CSC3 requires this labelling to be a *Sperner labelling*: The subdivision of each vertex is labelled by the input of the vertex and the vertices of the subdivision of a boundary edge are each labelled by the input of an endpoint of the edge.

Informally, Theorem 2 says that if the collection $\mathbb{H}$ satisfies the computational Sperner conditions, then in at least one (partial) terminal configuration in $\mathbb{T}$, each process outputs a different value.

▶ **Theorem 2.** *Let $\mathbb{H}$ be a collection of (partial) initial configurations that satisfies the computational Sperner conditions for some protocol and let $m \geq 2$ be the number of processes represented by each (partial) configuration in $\mathbb{H}$. For each $(C, Q) \in \mathcal{B}(\mathbb{H})$, let $\mathbb{T}(C, Q)$ be the set of all (partial) terminal configurations T reachable from configurations in $\mathbb{H}$ in the iterated immediate snapshot model such that*

- *m different values are output by the m processes in T and,*
- *each value in $I(C, Q)$ is output by some process in T.*

Then $|\mathbb{T}(C, Q)|$ is odd and, hence, $|\mathbb{T}(C, Q)| \geq 1$.

The proof of Theorem 2 is deferred to the appendix. In the next section, we demonstrate how it could be easily applied to obtain impossibility results for wait-free computation in the iterated immediate snapshot model.

4 New Impossibility Results from Sperner's Lemma

We first look at approximate agreement on the octahedron graph. We show that there is no wait-free algorithm when the number of processes n is at least 4. We then extend our result to a larger class of graphs.

To help with the presentation, we recall some standard notions in graph theory. A *vertex coloring* is an assignment of colors (or labels) to each vertex of a graph such that no edge has endpoints with the same color. A *k-coloring* is a vertex coloring that uses at most k different colors. A graph is *k-colorable* if it has a *k-coloring*. The *chromatic number* of a graph is the smallest number k such that the graph is k-colorable. The *clique number* of a graph G, denoted $\omega(G)$, is the size of its largest clique. Note that the chromatic number of G is always at least as large as $\omega(G)$.

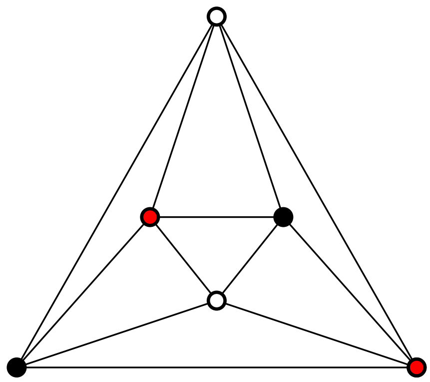


Figure 2 A 3-coloring of the octahedron graph.

4.1 The impossibility of approximate agreement on the octahedron graph

The octahedron graph (Figure 2) is obtained by taking the vertices and edges of an octahedron. It consists of a set of six vertices, $V = \{a_1, ..., a_6\}$, and a set of 12 edges. It is 3-colorable and its largest clique has size 3. To show that approximate agreement is unsolvable on the octahedron graph, it is sufficient to show that any protocol has an execution in which four processes output different values.

Consider a system with 4 processes p_0, p_1, p_2, p_3 and a 3-coloring of the octahedron graph, where we use $\{p_1, p_2, p_3\}$ as colors. We use $p(a_i)$ to denote the color of vertex a_i. For each triangle $\{a_i, a_j, a_k\}$ in the octahedron graph, let $C_{\{i,j,k\}}$ be the initial configuration where process $p(a_i)$ has input a_i, $p(a_j)$ has input a_j, $p(a_k)$ has input a_k, and p_0 has input a_1. Let $\mathbb{H}$ be the collection of all such configurations. (The choice of input vertex a_1 for process p_0 is arbitrary, but it has to be the same for all configurations in $\mathbb{H}$.)

The following two observations allow us to determine what pairs are in $\mathcal{B}(\mathbb{H})$.

▶ **Observation 3.** *Every edge in the octahedron graph is shared by exactly two triangles. If $\{a_i, a_j, a_k\}$ and $\{a_i, a_j, a_{k'}\}$ are both triangles in the octahedron graph, then configurations $C_{\{i,j,k\}}$ and $C_{\{i,j,k'\}}$ are indistinguishable to the processes in $\{p(a_i), p(a_j), p_0\}$. Since the edge $\{a_i, a_j\}$ is only shared by the two triangles $\{a_i, a_j, a_k\}$ and $\{a_i, a_j, a_{k'}\}$, there is no other configuration in $\mathbb{H}$ that is indistinguishable from configuration $C_{\{i,j,k\}}$ to the set of processes $\{p(a_i), p(a_j), p_0\}$.*

▶ **Observation 4.** *Each configuration $C_{\{i,j,k\}} \in \mathbb{H}$ is identified by the set $\{p_1, p_2, p_3\}$. In other words, in any other configuration $C_{\{i',j',k'\}} \in \mathbb{H}$, there is at least one process in $\{p_1, p_2, p_3\}$ that has a different input in $C_{\{i',j',k'\}}$ and in $C_{\{i,j,k\}}$.*

Observation 4 implies that $(C_{\{i,j,k\}}, \{p_1, p_2, p_3\}) \in \mathcal{B}(\mathbb{H})$ for all $C_{\{i,j,k\}} \in \mathbb{H}$. Observation 3 implies that $\{p_1, p_2, p_3\}$ is the only set of 3 processes that identifies $C_{\{i,j,k\}}$ in $\mathbb{H}$. Thus, $\mathcal{B}(\mathbb{H})$ contains only these pairs.

▶ **Lemma 5.** *For any wait-free protocol that solves approximate agreement on the octahedron graph, $\mathbb{H}$ satisfies the computational Sperner conditions.*

Proof. We prove that $\mathbb{H}$ satisfies the four conditions:

- CSC1: For any $(C_{\{i,j,k\}}, \{p_1, p_2, p_3\}) \in \mathcal{B}(\mathbb{H})$, processes p_1, p_2, p_3 received input values a_i, a_j, a_k in some order in configuration $C_{\{i,j,k\}}$. Hence, the three processes received three different input values.

- CSC2: Consider any two distinct pairs $(C_{\{i,j,k\}}, \{p_1, p_2, p_3\}), (C_{\{i',j',k'\}}, \{p_1, p_2, p_3\}) \in \mathcal{B}(\mathbb{H})$. Then, by definition, processes p_1, p_2, p_3 in the two configurations received vertices from two different triangles $\{a_i, a_j, a_k\}$ and $\{a_{i'}, a_{j'}, a_{k'}\}$ as input values (in some order). Hence, the set of input values $I(C, Q)$ is different for each pair $(C, Q) \in \mathcal{B}(\mathbb{H})$.
- CSC3: For any $(C_{\{i,j,k\}}, \{p_1, p_2, p_3\}) \in \mathcal{B}(\mathbb{H})$ and any subset $S \subseteq \{p_1, p_2, p_3\}$, the input values received by processes in S in configuration $C_{\{i,j,k\}}$ is a subset of the triangle $\{a_i, a_j, a_k\}$. Hence, in any protocol that solves approximate agreement on the octahedron graph, the clique validity condition ensures that the outputs of processes in S in any S-first execution starting from $C_{\{i,j,k\}}$ is a subset of the input values received by processes in S in configuration $C_{\{i,j,k\}}$.
- CSC4: Consider any $C_{\{i,j,k\}} \in \mathbb{H}$ and any 3 processes subset $Q \subsetneq \{p_0, p_1, p_2, p_3\}$. If $Q = \{p_1, p_2, p_3\}$, then Q identifies $C_{i,j,k}$ in $\mathbb{H}$ by Observation 4. In other words, no configuration in $\mathbb{H}$ is indistinguishable from $C_{i,j,k}$ to the set of processes $\{p_1, p_2, p_3\}$. Otherwise, without loss of generality, suppose $Q = \{p(a_i), p(a_j), p_0\}$. By Observation 3, the triangle $\{a_i, a_j, a_k\}$ shares the edge $\{a_i, a_j\}$ with exactly one other triangle $\{a_i, a_j, a_{k'}\}$ in octahedron graph, and $C_{\{i,j,k'\}}$ is the only configuration in $\mathbb{H}$ that is indistinguishable from $C_{\{i,j,k\}}$ to the set of processes $\{p(a_i), p(a_j), p_0\}$. ◀

▶ **Theorem 6.** *There is no wait-free protocol among 4 processes in the iterated immediate snapshot model that solves the approximate agreement problem on the octahedron graph.*

Proof. Consider a protocol that claims to solve approximate agreement on the octahedron graph. By Lemma 5, $\mathbb{H}$ satisfies the computational Sperner conditions. Since $\mathcal{B}(\mathbb{H})$ is nonempty, Theorem 2 implies that there exists a terminal configuration in which 4 different vertices are output by p_0, p_1, p_2, p_3. Since the largest clique in the octahedron has size 3, this contradicts the agreement condition. ◀

4.2 The impossibility of approximate agreement on a larger class of graphs

In this section, we define a class of graphs on which it is impossible to solve wait-free approximate agreement for sufficiently large number of processes. Given a point c in $\mathbb{R}^d$, a *sphere centered at* c is the set of all points equidistant from c in $\mathbb{R}^d$. It can be viewed as a subspace of dimension $d-1$ and, hence, is called a $(d-1)$-dimensional sphere. A *triangulation* of a $(d-1)$-dimensional sphere is a subdivision of the sphere into $(d-1)$-dimensional simplices, such that the intersection of any two simplices is either a common face of both simplices or empty. Our class of graphs includes the graph of any triangulated sphere that is not a clique. In particular, a cycle is the graph of a triangulated circle and the octahedron graph is the graph of a triangulated 2-dimensional sphere. We also compare our class of graphs to graphs that admit AER impossibility labellings and show that neither class contains the other.

Our class of graphs is defined by a set of *clique containment conditions*. The fact that every edge in the octahedron graph is shared by exactly two triangles was used in the previous section to show that a certain collection of initial configurations, $\mathbb{H}$, satisfies the computational Sperner conditions. We generalize this property to require that every clique of size $k-1$ in the graph is contained in exactly two cliques of size k, for some $k \geq 2$. In the proof of Theorem 6, we used Theorem 2 to show the existence an execution in which 4 processes output 4 different values. Since the octahedron graph contains no clique of size 4, this execution violates agreement. More generally, for any k-clique in the graph, we can use Theorem 2 to show the existence of an execution where the k-clique is a strict subset of the outputs. If this k-clique is a maximal clique in the graph, then agreement is violated.

The clique containment conditions

We say a graph G satisfy the *clique containment conditions* if there is a subgraph A of G and an integer k, where $2 \leq k$, such that the following hold:

1. every clique of size $k-1$ in A is contained in exactly two cliques of size k in A and
2. there is a clique of size k in A that is not contained in any clique of size $k+1$ in G.

The graph G of any triangulated d-dimensional sphere satisfies the first condition with $A = G$ and $k = d+1$ and, provided it is not a clique of size $k+2$, G also satisfies the second condition. In particular, the octahedron graph is the graph of a triangulated 2-dimensional sphere. Since it contains no clique of size 4, none of its cliques of size 3 is contained in a clique of size 4.

When $k = 2$, every graph G that satisfies our clique containment conditions also has an AER impossibility labelling. In this case, our first impossibility condition implies that A is a collection of disjoint cycles and our second impossibility condition implies that there exists an edge $\{u, v\}$ in A that is not contained in any triangle in G. Label u with value 1, v with value 2, and all other vertices in G with value 0. This gives an AER impossibility labelling of the graph G.

When $k \geq 3$, there are some graphs, in particular the octahedron graph, that satisfy the clique containment conditions, but do not have an AER impossibility labelling. For contradiction, suppose the octahedron graph has an AER impossibility labelling. Then it contains a cycle C of length at least 4 with three consecutive vertices labelled 0, 1, and 2. Since an AER impossibility labelling has no triangle with three different labels, these three vertices do not form a triangle. To finish labelling the rest of the graph so that there is no triangle with three different labels, observe that all other vertices can only receive the label 1. Hence, the cycle C contains at least two different vertices with the label 1, contradicting the definition of AER impossibility labelling.

There are also examples of graphs that have AER impossibility labellings, but do not satisfy the clique containment conditions. For example, consider the graph G' with ten vertices shown in Figure 3, where C is the cycle of length 5 in the middle of G'. Since each edge of G' is contained in some triangle, G' does not satisfy our second impossibility condition when $k = 2$. Since each edge of G' is contained in exactly one triangle, G' does not satisfy our first impossibility condition when $k = 3$. Note that G' has no clique of size greater than 3, hence G' does not satisfy our second impossibility condition when $k > 3$.

Clique containment conditions imply impossibility of approximate agreement

Consider a graph G that satisfies the clique containment conditions. We first construct a collection $\mathbb{H}$ of (partial) initial configurations for any protocol that claims to solve approximate agreement on G. Consider a subgraph $A = (V, E)$ of G and an integer k such that the clique containment conditions are satisfied. Let $a_1, ..., a_\ell$ denote the vertices in V, where $\ell = |V|$. Consider a system with n processes $p_0, p_1, ..., p_{n-1}$, where n is greater than the chromatic number of A. Consider an $n-1$ coloring of the graph A, where we use $\{p_1, ..., p_{n-1}\}$ as colors. We use $p(a_i)$ to denote the color of vertex a_i. For each k-clique $\{a_{i_1}, ..., a_{i_k}\}$ in the graph A, let $C_{\{i_1,...,i_k\}}$ be the (partial) initial configuration consisting of the states of processes $p(a_{i_1}), ..., p(a_{i_k})$ and p_0, such that process p_0 has input value a_1 and, for $1 \leq j \leq k$, process $p(a_{i_j})$ has input value a_{i_j}. Let $\mathbb{H}$ be the collection of all such (partial) configurations. As in Section 4.1, the input value assigned to p_0 is not important, as long as it is the same in all (partial) configurations in $\mathbb{H}$.

The following two observations allow us to determine what pairs are in $\mathcal{B}(\mathbb{H})$.

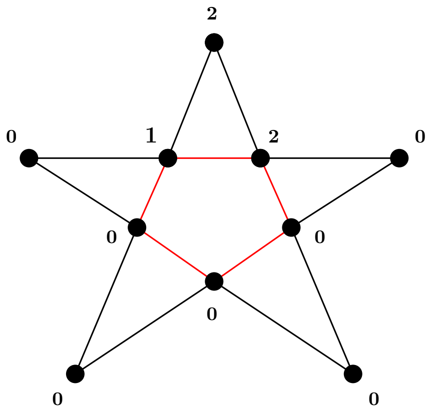


Figure 3 An AER impossibility labelling of a graph G' with ten vertices. The edges of the cycle C are colored in red.

▶ **Observation 7.** *If* $\{a_{i_1}, ..., a_{i_{k-1}}, a_{i_k}\}$ *and* $\{a_{i_1}, ..., a_{i_{k-1}}, a_{i'_k}\}$ *are both* k*-cliques of* A*, then partial configurations* $C_{\{i_1,...,i_k\}}$ *and* $C_{\{i_1,...,i'_k\}}$ *are indistinguishable to the processes in* $\{p(a_{i_1}), ..., p(a_{i_{k-1}}), p_0\}$*. Moreover,* $C_{\{i_1,...,i'_k\}}$ *is the only other partial configuration in* $\mathbb{H}$ *that is indistinguishable from* $C_{\{i_1,...,i_k\}}$ *to the set of processes* $\{p(a_{i_1}), ..., p(a_{i_{k-1}}), p_0\}$*.*

The second statement of Observation Observation 7 is true because, the first impossibility condition says that the $(k-1)$-clique $\{a_{i_1}, ..., a_{i_{k-1}}\}$ is contained in no other k-cliques of A.

▶ **Observation 8.** *Each partial configuration* $C_{\{i_1,...,i_k\}}$ *is identified by the set of processes* $\{p(a_{i_1}), ..., p(a_{i_k})\}$*. In other words, in any other configuration* $C_{\{i'_1,...,i'_k\}}$*, either* $\{p(a_{i_1}), ..., p(a_{i_k})\} \neq \{p(a_{i'_1}), ..., p(a_{i'_k})\}$*, or there is at least one process in* $\{p(a_{i_1}), ..., p(a_{i_k})\}$ *that has a different input in* $C_{\{i_1,...,i_k\}}$ *and in* $C_{\{i'_1,...,i'_k\}}$*.*

Observation 8 implies that $(C_{\{i_1,...,i_k\}}, \{p(a_{i_1}), ..., p(a_{i_k})\}) \in \mathcal{B}(\mathbb{H})$ for all $C_{\{i_1,...,i_k\}} \in \mathbb{H}$. Observation 7 implies that $\{p(a_{i_1}), ..., p(a_{i_k})\}$ is the only set of size k that identifies $C_{\{i_1,...,i_k\}}$ in $\mathbb{H}$. Thus $\mathcal{B}(\mathbb{H})$ contains only these pairs.

The next lemma is a generalization of Lemma 5, and has a similar proof.

▶ **Lemma 9.** *For any wait-free algorithm that solves approximate agreement on the graph* G*,* $\mathbb{H}$ *satisfies the computational Sperner conditions.*

Proof. We prove $\mathbb{H}$ satisfies the four conditions:

- CSC1: For any $(C_{\{i_1,...,i_k\}}, \{p(a_{i_1}), ..., p(a_{i_k})\}) \in \mathcal{B}(\mathbb{H})$, processes $p(a_{i_1}), ..., p(a_{i_k})$ received vertices of the k-clique $\{a_{i_1}, ..., a_{i_k}\}$ as input values in some order. Hence, the k processes received k different input values in $C_{\{i_1,...,i_k\}}$.
- CSC2: Consider any two distinct pairs $(C_{\{i_1,...,i_k\}}, \{p(a_{i_1}), ..., p(a_{i_k})\}), (C_{\{i'_1,...,i'_k\}}, \{p(a_{i'_1}), ..., p(a_{i'_k})\}) \in \mathcal{B}(\mathbb{H})$. Then, by definition, processes $p(a_{i_1}), ..., p(a_{i_k})$ in configuration $C_{\{i_1,...,i_k\}}$ and processes $p(a_{i'_1}), ..., p(a_{i'_k})$ in configuration $C_{\{i'_1,...,i'_k\}}$ received vertices from two different k-cliques $\{a_{i_1}, ..., a_{i_k}\}$ and $\{a_{i'_1}, ..., a_{i'_k}\}$ as input values. Hence, the set of input values $I(C, Q)$ is different for each pair $(C, Q) \in \mathcal{B}(\mathbb{H})$.
- CSC3: For any $(C_{\{i_1,...,i_k\}}, \{p(a_{i_1}), ..., p(a_{i_k})\}) \in \mathcal{B}(\mathbb{H})$ and any subset $S \subseteq \{p(a_{i_1}), ..., p(a_{i_k})\}$, the set of inputs received by processes in S in $C_{\{i_1,...,i_k\}}$ is a subset of the k-clique $\{a_{i_1}, ..., a_{i_k}\}$. Hence, in any protocol that solves approximate agreement on graph G, the clique validity condition ensures that the outputs of processes in S in any S-first execution starting from $C_{\{i_1,...,i_k\}}$ is a subset of input values received by processes in S in $C_{\{i_1,...,i_k\}}$.

- CSC4: Consider any $C_{\{i_1,...,i_k\}} \in \mathbb{H}$ and any $(k-1)$-process subset $Q \subsetneq \pi(C_{\{i_1,...,i_k\}}) = \{p(a_{i_1}), ..., p(a_{i_k}), p_0\}$. If $Q = \{p(a_{i_1}), ..., p(a_{i_k})\}$, then, by Observation 8, Q identifies $C_{\{i_1,...,i_k\}}$ in $\mathbb{H}$. In other words, for any other partial configuration $C_{\{i'_1,...,i'_k\}} \in \mathbb{H}$ such that $Q \subsetneq \pi(C_{\{i'_1,...,i'_k\}})$, at least one process in Q has a different state in $C_{\{i_1,...,i_k\}}$ and in $C_{\{i'_1,...,i'_k\}}$. Otherwise, without loss of generality, suppose $Q = \{p(a_{i_1}), ..., p(a_{i_{k-1}}), p_0\}$. By Observation 7, the k-clique $\{a_{i_1}, ..., a_{i_k}\}$ shares the $(k-1)$-clique $\{a_{i_1}, ..., a_{i_{k-1}}\}$ with exactly one other k-clique $\{a_{i_1}, ..., a_{i_{k-1}}, a_{i'_k}\}$. Furthermore, $C_{\{i_1,...,i'_k\}}$ is the only configuration in $\mathbb{H}$ such that $Q \subsetneq \pi(C_{\{i_1,...,i'_k\}})$ and $C_{\{i_1,...,i'_k\}}$ is indistinguishable from $C_{\{i_1,...,i_k\}}$ to the set of processes Q. ◀

We are now ready to prove the impossibility of a wait-free solution to the approximate agreement problem on graphs that satisfy the clique containment conditions.

▶ **Theorem 10.** *Let G be a graph that satisfies the clique containment conditions with subgraph A and integer k. Then there is no wait-free protocol among n processes in the iterated immediate snapshot model that solves approximate agreement on G when n is greater than the chromatic number of A.*

Proof. Consider a protocol that claims to solve approximate agreement on the graph G. Pick a k-clique $\{a_{i_1}, ..., a_{i_k}\}$ in A that is not contained in any $(k+1)$-clique in G. By Observation 8, $(C_{i_1,...,i_k}, \{p(a_{i_1}), ..., p(a_{i_k})\}) \in \mathcal{B}(\mathbb{H})$. Then, by Theorem 2, there exists a partial terminal configuration T in which $k+1$ different values are output, including each value in $\{a_{i_1}, ..., a_{i_k}\}$. Since the clique $\{a_{i_1}, ..., a_{i_k}\}$ is maximal, the $k+1$ values output by processes in T is not a clique in the graph G. This contradicts the agreement condition. ◀

5 More Impossibility Results from Reductions

In this section, we describe a simple reduction from approximate agreement on one graph to another graph. These reductions allow us to extend our impossibility result to even more graphs.

Let $G = (V, E)$ and $G' = (V', E')$ be undirected graphs. We say that a vertex map $\psi : V \to V'$ is a *clique map* if, for every clique κ in G, $\phi(\kappa)$ is a clique in G'.

▶ **Theorem 11.** *Let $G = (V, E)$ and $G' = (V', E')$ be graphs for which there exists clique maps $\psi : V \to V'$ and $\psi' : V' \to V$, such that $\psi'(\psi(u)) = u$ for all $u \in V$. Then, if approximate agreement on G' has a wait-free solution among n processes, so does approximate agreement on G.*

Proof. Let $\mathcal{A}'$ be a wait-free protocol solving approximate agreement on the graph G'. We construct a wait-free protocol $\mathcal{A}$ solving approximate agreement on the graph G as follows: each process with input x runs the approximate agreement algorithm $\mathcal{A}'$ on G' using $\psi(x)$ as its input. If y' is the output it obtained from this execution of $\mathcal{A}'$, then it outputs $\psi'(y')$.

By the agreement property of $\mathcal{A}'$, the set of outputs in each execution of $\mathcal{A}'$ is a clique κ' in G'. Since ψ' is a clique map, the set of outputs in each execution of $\mathcal{A}$, $\psi'(\kappa')$, is a clique in G. Hence, $\mathcal{A}$ satisfies agreement.

To see that $\mathcal{A}$ satisfies validity, suppose the set of inputs in some execution of $\mathcal{A}$ is a clique κ in G. Since ψ is a clique map, it follows that $\psi(\kappa)$ is a clique in G'. By validity of $\mathcal{A}$, the set of outputs in this execution of $\mathcal{A}$ is a subset $\kappa' \subseteq \psi(\kappa)$. Thus, $\psi'(\kappa') \subseteq \psi'(\psi(\kappa)) = \kappa$.

Hence, $\mathcal{A}$ solves approximate agreement on G in a wait-free manner among n processes. ◀

We present two applications of Theorem 11. Let $G = (V, E)$ be the 5-cycle, let $G' = (V', E')$ be the graph in Figure 3, and let C be the cycle of length 5 in the middle of G'. Let $\psi : V \to V'$ be the clique map that maps G onto C. Let $\psi' : V' \to V'$ be the clique map such that $\psi'(\psi(u)) = u$ for each vertex $u \in V$ and $\psi'(v) = w$ for each $v \in V' \setminus C$, where w is a vertex on the 5-cycle such that $\psi(w)$ is adjacent to v in G'. Since approximate agreement on the 5-cycle has no wait-free solution among $n \geq 3$ processes, Theorem 11 implies that approximate agreement on G' has no wait-free solution among $n \geq 3$ processes. Note that in Section 4.2 we showed that G' has an AER impossibility labelling, which gives another proof of this result.

The *stellated octahedron* is obtained by attaching a tetrahedron to each face of the octahedron. More formally, let $H = (V, E)$ be the octahedron graph. We can obtain the graph of the stellated octahedron graph $H' = (V', E')$ from H as follows: $V \subseteq V'$, $E \subseteq E'$, and, for each triangle $\{v_i, v_j, v_k\}$ in H, there is a new vertex $v_{\{i,j,k\}} \in V'$ and three new edges $\{v_{\{i,j,k\}}, v_i\}, \{v_{\{i,j,k\}}, v_j\}, \{v_{\{i,j,k\}}, v_k\} \in E'$. Then $\psi : V \to V'$, which maps each vertex $v \in V$ to $v \in V'$, is a clique map. Likewise, let $\psi' : V' \to V$ map each vertex $v \in V \subsetneq V'$ to v and map each vertex $v' \in V' \setminus V$ to a vertex $u \in V \subsetneq V'$ that is adjacent to v' in the stellated octahedron graph. Then ψ' is a clique map such that $\psi'(\psi(v)) = v$ for all $v \in V$. Combining Theorem 6 and Theorem 11 gives the following result.

▶ **Corollary 12.** *There is no wait-free protocol among 4 processes in the iterated immediate snapshot model that solves the approximate agreement problem on the stellated octahedron graph.*

6 Extension-Based Proofs

The notion of extension-based proofs was introduced by Alistarh, Aspnes, Ellen, Gelashvili, and Zhu [2]. It describes a class of impossibility proofs that includes valency arguments. Extension-based proofs are defined as an interaction between a *prover* and any protocol that claims to solve a task in a wait-free manner. The prover repeatedly queries the protocol while it attempts to construct a faulty or infinite execution of the protocol. It is known that extension-based proofs cannot be used to prove the impossibility of $(n-1)$-set agreement [2] and approximate agreement on 4-cycle [4]. In contrast, combinatorial proofs of these impossibility results exist [14, 20, 9, 3]. In the full version of our paper, we show that extension-based proofs cannot be used to prove the impossibility of approximate agreement on any connected graph.

▶ **Theorem 13.** *For any connected graph G, there is no extension-based proof of the impossibility of a wait-free solution for approximate agreement on G among $n \geq 3$ processes.*

7 Futher work

We conclude by discussing a few open problems about approximate agreement on graphs.

- Is there a wait-free protocol using registers for approximate agreement on the octahedron graph for $n = 3$ processes? When $n \geq 4$, Theorem 6 implies that no wait-free algorithm exists. The algorithm by Alistarh, Ellen, and Rybicki that tolerates one crash failure [3] solves approximate agreement in a wait-free manner for $n = 2$ processes. We know that extension-based proofs are not powerful enough to obtain any impossibility result for wait-free approximate agreement on graphs. Thus, to prove that approximate agreement on the octahedron graph for $n = 3$ processes is impossible, a reduction or a combinatorial approach is required.
- For any graph G that satisfies the clique containment conditions, what is the largest number of processes for which there is a wait-free protocol using registers that solves approximate agreement on G?

- Find a wait-free protocol using registers for approximate agreement on graphs whose complex of cliques is contractible or show that no such algorithm exists. Does knowing that a graph is contractible help to solve approximate agreement? More generally, find a decidable characterization of the class of graphs on which approximate agreement has a wait-free solution using registers. Note that there exists a topological characterization of tasks that have wait-free solutions using registers [14], but this characterization is not decidable.

References

1 Manuel Alcántara, Armando Castañeda, David Flores-Peñaloza, and Sergio Rajsbaum. The topology of look-compute-move robot wait-free algorithms with hard termination. *Distrib. Comput.*, 32(3):235–255, 2019.

2 Dan Alistarh, James Aspnes, Faith Ellen, Rati Gelashvili, and Leqi Zhu. Why extension-based proofs fail. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing*, (STOC), pages 986–996, 2019.

3 Dan Alistarh, Faith Ellen, and Joel Rybicki. Wait-free approximate agreement on graphs. In *Proceeding of the 28th International Colloquium on Structural Information and Communication Complexity*, (SIROCCO), pages 87–105, 2021.

4 Dan Alistarh, Faith Ellen, and Joel Rybicki. Wait-free approximate agreement on graphs, 2021. `arXiv:2103.08949`.

5 Hagit Attiya and Armando Castañeda. A non-topological proof for the impossibility of k-set agreement. In *Proceedings of the 13th International Symposium on Stabilization, Safety, and Security of Distributed Systems*, (SSS), pages 108–119, 2011.

6 Hagit Attiya and Faith Ellen. *Impossibility results for distributed computing.* Morgan et Claypool Publishers, 2014.

7 Hagit Attiya and Faith Ellen. The step complexity of multidimensional approximate agreement. In *Proceedings of the 26th International Conference on Principles of Distributed Systems*, (OPODIS), pages 25:1–25:12, 2022.

8 Hagit Attiya, Nancy Lynch, and Nir Shavit. Are wait-free algorithms fast? *J. ACM*, 41(4):725–763, 1994.

9 Elizabeth Borowsky and Eli Gafni. Generalized flp impossibility result for t-resilient asynchronous computations. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, (STOC), pages 91–100. Association for Computing Machinery, 1993.

10 Armando Castañeda, Sergio Rajsbaum, and Matthieu Roy. Convergence and covering on graphs for wait-free robots. *Journal of the Brazilian Computer Society*, 24(1):1–15, 2018.

11 Danny Dolev, Nancy Lynch, Shlomit Pinter, Eugene Stark, and William Weihl. Reaching approximate agreement in the presence of faults. *Journal of the ACM*, 33(3):499–516, 1986.

12 Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, 1985.

13 Maurice Herlihy, D. N. Kozlov, and Sergio Rajsbaum. *Distributed computing through combinatorial topology.* Morgan Kaufmann, 2014.

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

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

16 Jérémy Ledent. Brief announcement: variants of approximate agreement on graphs and simplicial complexes. In *Proceedings of the 40th Annual ACM Symposium on Principles of Distributed Computing*, (PODC), pages 427–430, 2021.

17 Hammurabi Mendes, Maurice Herlihy, Nitin Vaidya, and Vijay K Garg. Multidimensional agreement in byzantine systems. *Distributed Computing*, 28(6):423–441, 2015.

18 Thomas Nowak and Joel Rybicki. Byzantine Approximate Agreement on Graphs. In *Proceedings of the 33rd International Symposium on Distributed Computing*, (DISC), pages 29:1–29:17, 2019.

19 M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980.

20 Michael Saks and Fotios Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. *SIAM Journal on Computing*, 29(5):1449–35, 2000.

21 E. Schenk. Faster approximate agreement with multi-writer registers. In *Proceedings of the 36th IEEE Annual Symposium on Foundations of Computer Science*, (FOCS), pages 714–723, 1995.

A Proof of Theorem 2

In this appendix, we give a complete proof of Theorem 2. We begin with a few lemmas describing technical properties of the iterated immediate snapshot model.

▶ **Lemma 14.** *Let C be a (partial) configuration and $S \subseteq \pi(C)$ be a nonempty subset of processes. Let $\beta = B_1, ..., B_\ell, ..., B_r$ be a 1-round S-first schedule from C, where $B_1 \cup \cdots \cup B_\ell$ is the set of active processes in S. Then for any 1-round schedule $\beta' = B'_1, ..., B'_m$ such that $C\beta$ and $C\beta'$ are indistinguishable to all processes in S, $B_1, ..., B_\ell$ is a prefix of β'.*

Proof. Consider the smallest k such that $B_k \neq B'_k$. Note that $B_k \cup \cdots \cup B_r = B'_k \cup \cdots \cup B'_m$. Suppose $B_1, ..., B_\ell$ is not a prefix of β'. Then $k \leq \ell$ and $B_k \subseteq S$.

If there exists some process $q \in B'_k \setminus B_k$, then every process in $B'_k \cup \cdots \cup B'_m$ will see the update by process q during β'. However, processes in $B_k \subseteq B'_k \cup \cdots \cup B'_m$ will not see the update by process q during β. Hence, all processes in $B_k \subseteq S$ distinguish between $C\beta$ and $C\beta'$.

So suppose that $B'_k \subsetneq B_k$. Let $q \in B_k \setminus B'_k$. Then every process in B_k will see the update by process q during β. However, processes in B'_k will not see the update by process q during β'. Hence, all processes in $B'_k \subseteq S$ distinguish between $C\beta$ and $C\beta'$. ◀

The next result is a restatement of Lemma 8.4 from [6]. Its proof is similar.

▶ **Lemma 15.** *Let C be a (partial) configuration. For any (partial) configuration $D \in \chi(C, \delta)$ and any subset $Q \subsetneq \pi(C)$ of $|\pi(C)| - 1$ processes, there is at most one other (partial) configuration $D' \in \chi(C, \delta)$ such that D and D' are indistinguishable to processes in Q. Moreover, Q identifies D in $\chi(C, \delta)$ if and only if D is reached from C via a 1-round Q-first schedule.*

Proof. Consider the the 1-round schedule $\beta = B_1, B_2, ..., B_r$ such that $D = C\beta$. Let p be the only process in $\pi(C) \setminus Q$ and let $B_1, ..., B_\ell$ be the longest Q-only prefix of β. Consider an arbitrary 1-round schedule $\beta' = B'_1, ..., B'_m$, such that $C\beta$ and $C\beta'$ are indistinguishable to processes in Q. By Lemma 14, $B'_1, ..., B'_\ell = B_1, ..., B_\ell$.

Case 1: β is Q-first.

If p is not active, then $\ell = r = m$ and, hence, $\beta' = \beta$. Otherwise, $\ell = r - 1$ and $B_r = \{p\}$. Hence $B'_1, ..., B'_{r-1} = B_1, ..., B_{r-1}$ and $B'_r = B_r = \{p\}$, so $\beta' = \beta$. In both cases, Q identifies $D = C\beta$.

Case 2: β is not Q-first.

Then process p is active in C. By definition of ℓ, $B_{\ell+1}$ is the block containing process p.

Case 2.1: $B_{\ell+1} = \{p\}$.

Then $\ell+1 < r$ because β is not Q-first. If there exists a process $q \in B'_{\ell+1} \setminus (B_{\ell+2} \cup \{p\})$, then every process in $B'_{\ell+1} \cup \cdots \cup B'_m$ will see the update by process q during β'. However, processes in $B_{\ell+2} \subsetneq B'_{\ell+1} \cup \cdots \cup B'_m$ will not see the update by process q during β. Hence, all processes in $B_{\ell+2} \subseteq S$ distinguish between $C\beta$ and $C\beta'$, contradicting the definition of β'. Therefore, $B'_{\ell+1} \subseteq B_{\ell+2} \cup \{p\}$.

Likewise, if there exists a process $q \in (B_{\ell+2} \cup \{p\}) \setminus B'_{\ell+1}$, then every process in $B_{\ell+2} \cup \cdots \cup B_r$ will see the update by process q during β. However, processes in $B'_{\ell+1}$ will not see the update by process q during β' and, hence, are able to distinguish between $C\beta$ and $C\beta'$. Since no process in Q can distinguish between $C\beta$ and $C\beta'$, this implies that either $B'_{\ell+1} = B_{\ell+2} \cup \{p\}$ or $B'_{\ell+1} = \{p\}$.

Case 2.2: $B_{\ell+1} \neq \{p\}$.

Then $B_{\ell+1} \cap S \neq \emptyset$. If there exists $q \in B'_{\ell+1} \setminus B_{\ell+1}$, then every process in $B'_{\ell+1} \cup \cdots \cup B'_m$ will see the update by q during β'. However, processes in $B_{\ell+1} \cap Q \subseteq B'_{\ell+1} \cup \cdots \cup B'_m$ will not see the update by q during β. Hence, all processes in $B_{\ell+1} \cap Q$ distinguish between $C\beta$ and $C\beta'$, contradicting the definition of β'. Therefore, $B'_{\ell+1} \subseteq B_{\ell+1}$.

Likewise, if there exists $q \in B_{\ell+1} \setminus B'_{\ell+1}$, then every process in $B_{\ell+1} \cup \cdots \cup B_r$ will see the update by q during β. However, processes in $B'_{\ell+1} \cap Q$ will not see the update by q during β' and, hence, are able to distinguish $C\beta$ from $C\beta'$. Since no process in Q can distinguish between $C\beta$ and $C\beta'$, this implies that either $B'_{\ell+1} = B_{\ell+1}$, or $B'_{\ell+1} \cap Q = \emptyset$. Note that $B'_{\ell+1} \cap Q = \emptyset$ implies that $B'_{\ell+1} = \{p\}$.

Thus process p has exactly two possible states in $C\beta'$. For each of these states of p, applying Lemma 14 with $S = \pi(C)$ gives a unique schedule. ◀

We can apply Lemma 15 repeatedly to each round of an execution to show the following result.

▶ **Lemma 16.** *Let C be a (partial) initial configuration and let t be the maximum number of non-trivial rounds taken by any $\pi(C)$-only execution starting from C. For any $0 \leq r \leq t$ and any (partial) configuration $T \in \chi^r(C, \delta)$, a set of $|\pi(C)| - 1$ processes $Q \subsetneq \pi(C)$ identifies T in $\chi^r(C, \delta)$ if and only if T is reachable via a Q-first schedule from C.*

Proof. Let $\beta_1, ..., \beta_r$ be the r-round schedule such that $T = C\beta_1...\beta_r$. Consider any subset $Q \subseteq \pi(C)$ of $|\pi(C)| - 1$ processes.

First suppose that $\beta_1, ..., \beta_r$ is not Q-first. Let $k \leq r$ be the largest index such that β_k is not Q-first. The remaining 1-round schedules $\beta_{k+1}, ..., \beta_r$ are all Q-first. Let $C' = C\beta_1...\beta_{k-1}$ and let $D = C'\beta_k \in \chi(C', \delta)$. Then, Lemma 15 says that there exists exactly one other $D' \in \chi(C', \delta)$ such that D and D' are indistinguishable to processes in Q. Since $\beta_{k+1}, ..., \beta_r$ are Q-first, it follows that $D\beta_{k+1}...\beta_j$ and $D'\beta_{k+1}...\beta_j$ are indistinguishable to processes in Q, for each $k + 1 \leq j \leq r$. Hence, Q does not identify $T = C\beta_1...\beta_r$ in $\chi^r(C, \delta)$.

Now suppose that $\beta_1, ..., \beta_r$ is Q-first. We inductively show that Q identifies $C\beta_1...\beta_i$ in $\chi^i(C, \delta)$ for all $0 \leq i \leq r$. For the base case, since C is the only configuration in $\chi^0(C, \delta)$, Q identifies C. For the inductive case, let $i < r$ and assume that Q identifies $D = C\beta_1...\beta_i$ in $\chi^i(C, \delta)$. Let $E = D\beta_{i+1}$ and let $E' \in \chi^{i+1}(C, \delta)$ be such that $E \neq E'$. Then $E' \in \chi(D', \delta)$ for some $D' \in \chi^i(C, \delta)$. If $D \neq D'$, then by the inductive hypothesis, some process $q \in \pi(C)$ distinguishes between D and D' and, hence, distinguishes between E and E'. If $D = D'$, then, since β_{i+1} is Q-first, Lemma 15 implies that Q identifies E in $\chi(D, \delta)$. Hence, Q identifies $C\beta_1...\beta_{i+1}$ in $\chi^{i+1}(C, \delta)$. ◀

▶ **Lemma 17.** *Let $m \geq 2$ and let $\mathbb{K}$ be a collection of (partial) configurations such that $|\pi(C)| = m$ for all $C \in \mathbb{K}$. Suppose that, for any $C \in \mathbb{K}$ and any subset $Q \subsetneq \pi(C)$ of $m-1$ processes, there is at most one other $C' \in \mathbb{K}$ such that $Q \subsetneq \pi(C')$ and the (partial) configurations C and C' are indistinguishable to all processes in Q. Then, for any $D \in \chi(\mathbb{K}, \delta)$ and any subset $R \subsetneq \pi(D)$ of $m-1$ processes, there is at most one other $D' \in \chi(\mathbb{K}, \delta)$ such that $R \subsetneq \pi(D')$ and the (partial) configurations D and D' are indistinguishable to all processes in R.*

Proof. Consider a (partial) configuration $D \in \chi(\mathbb{K}, \delta)$ and a subset $R \subsetneq \pi(D)$ of $m-1$ processes. By definition, $D = C\beta$ for some (partial) configuration $C \in \mathbb{K}$ and some 1-round $\pi(C)$-only schedule β.

First suppose that β is not R-first. Let p be the only process in $\pi(D) \setminus R$. Then process p and at least one process in R are active in C. Consider any $D' \in \chi(\mathbb{K}, \delta)$ such that $R \subsetneq \pi(D')$ and D' is indistinguishable from D to all processes in R. Then $D' \in \chi(C', \delta)$ for some $C' \in \mathbb{K}$. Note that $\pi(C) = \pi(D)$ and $\pi(C') = \pi(D')$. Suppose that $C' \neq C$. Since β is not R-first, the scan of some active process $q \in R$ sees the update by every active process in $\pi(D)$ during β. If $\pi(D') \neq \pi(D)$, then $p \notin \pi(D')$ since $R \subsetneq \pi(D') \neq \pi(D) = R \cup \{p\}$. In this case, then q distinguishes between D and D', because q sees the update by process p during β. Hence, $\pi(D') = \pi(D)$. If there is a process that is active in C and has a different state in $C' \neq C$, then q distinguishes between D and D'. Hence, every process that is active in C has the same state in C'. Since $C' \neq C$, there is a process q' that is terminated in C and has a different state in C'. Since p is active in C and q' is not, $q' \neq p$ and, hence, $q' \in R$. Thus, q' is a process in R that distinguishes between D' and D. This contradicts the definition of D'. Therefore, $C' = C$. Then, by Lemma 15, either $D' = D$ or $D' \neq D$ is the unique (partial) configuration in $\chi(C, \delta)$ that is indistinguishable from D to processes in R.

Now suppose that β is R-first. By assumption, there is at most one other configuration $C' \in \mathbb{K}$, such that $R \subsetneq \pi(C')$ and C' is indistinguishable from C to all processes in R. It follows that $C\beta$ and $C'\beta$ are indistinguishable to all processes in R. Since β is R-first, R identifies $C\beta$ in $\chi(C, \delta)$ and R identifies $C'\beta$ in $\chi(C', \delta)$. Hence for any other 1-round schedule $\beta' \neq \beta$, at least one process in R distinguishes between $C\beta$ and $C\beta'$, and at least one process in R distinguishes between $C\beta$ and $C'\beta'$. For any configuration $C'' \in \mathbb{K}$ such that $C'' \neq C, C'$, there is at least one process $q \in R$ that distinguishes between C'' and C. Since q also distinguishes between $C\beta$ and $C''\beta''$ for any 1-round schedule β'', it follows that $C'\beta$ is the only configuration in $\chi(\mathbb{K}, \delta)$ that is indistinguishable from $C\beta$ to processes in R. ◀

The rest of this section is devoted to proving Theorem 2.

▶ **Theorem 2.** *Let $\mathbb{H}$ be a collection of (partial) initial configurations that satisfies the computational Sperner conditions for some protocol and let $m \geq 2$ be the number of processes represented by each (partial) configuration in $\mathbb{H}$. For each $(C, Q) \in \mathcal{B}(\mathbb{H})$, let $\mathbb{T}(C, Q)$ be the set of all (partial) terminal configurations T reachable from configurations in $\mathbb{H}$ in the iterated immediate snapshot model such that*

- *m different values are output by the m processes in T and,*
- *each value in $I(C, Q)$ is output by some process in T.*

Then $|\mathbb{T}(C, Q)|$ is odd and, hence, $|\mathbb{T}(C, Q)| \geq 1$.

The proof is by strong induction on m. Let $m \geq 2$ and assume that the claim is true for all m' such that $2 \leq m' < m$. Let t be the maximum number of non-trivial rounds taken by the protocol in executions starting from (partial) initial configurations in $\mathbb{H}$ and let $\mathbb{T} = \chi^t(\mathbb{H}, \delta)$. If $\mathcal{B}(\mathbb{H})$ is empty, then there is nothing to prove. So assume $\mathcal{B}(\mathbb{H})$ is nonempty. Fix an arbitrary $(C, Q) \in \mathcal{B}(\mathbb{H})$. Define a graph $G = (\mathbb{T} \cup \{w\}, E)$ as follows:

- There is an edge in E between (partial) terminal configurations T and T' if and only if T and T' are indistinguishable to a subset $Q' \subseteq \pi(T) \cap \pi(T')$ of $m-1$ processes and $I(C,Q)$ is the set of values output in T (and hence in T') by the processes in Q'.
- There is an edge in E between a (partial) terminal configuration T and vertex w if and only if there is a subset $Q' \subsetneq \pi(T)$ of $m-1$ processes that identifies T in $\mathbb{T}$ and $I(C,Q)$ is the set of values output in T by the processes in Q'.

▶ **Lemma 18.** *For each (partial) terminal configuration $T \in \mathbb{T}$ adjacent to w in G, there is some $(C',Q') \in \mathcal{B}(\mathbb{H})$ such that $I(C',Q') = I(C,Q)$, T is reachable from C' via a unique schedule β, and β is Q'-first.*

Proof. Let T be adjacent to w in G. Then there exists a subset $Q' \subsetneq \pi(T)$ of $m-1$ processes that identifies T in $\mathbb{T}$ and $I(C,Q)$ is the set of values output in T by the processes in Q'. Since $\mathbb{T} = \chi^t(\mathbb{H},\delta)$, it follows that $T = C'\beta_1...\beta_t$ for some (partial) initial configuration $C' \in \mathbb{H}$ and some t-round $\pi(C')$-only schedule $\beta_1,...,\beta_t$ starting from C'. Since we are considering the full-information iterated immediate snapshot model, this schedule $\beta_1,...,\beta_t$ is unique. Since $\chi^t(C',\delta) \subseteq \mathbb{T}$, Q' also identifies T in $\chi^t(C',\delta)$. Hence, by Lemma 16, $\beta_1,...,\beta_t$ is Q'-first. If some other (partial) initial configuration $C'' \in \mathbb{H}$ is indistinguishable from C' to all processes in Q', then $C''\beta_1...\beta_t$ is indistinguishable from $C'\beta_1...\beta_t$ to all processes in Q'. This contradicts the fact that Q' identifies $C'\beta_1...\beta_t$ in $\mathbb{T}$. Therefore, Q' identifies C' in $\mathbb{H}$. Hence, $(C',Q') \in \mathcal{B}(\mathbb{H})$.

Since $\beta_1...\beta_t$ is Q'-first, by CSC3, the outputs of processes in Q' in T is a subset of $I(C',Q')$. Since T is adjacent to w, $I(C,Q)$ is the set of values output by processes in Q' in T. Hence we know $I(C,Q) \subseteq I(C',Q')$. However, by CSC1, $|I(C,Q)| = |I(C',Q')| = m-1$. Thus, $I(C,Q) = I(C',Q')$. ◀

▶ **Lemma 19.** *For each $(C',Q') \in \mathcal{B}(\mathbb{H})$ such that $I(C',Q') = I(C,Q)$, there are an odd number of (partial) terminal configurations in $\mathbb{T}$ reachable from C' that are adjacent to w in G.*

Proof. If $m = 2$, then $|Q'| = m-1 = 1$. Let q be the only process in Q'. Let $\beta_1,...,\beta_t$ be a t-round $\{q\}$-first schedule starting from C'. By Lemma 16, $C'\beta_1...\beta_t$ is identified by $\{q\}$ in $\mathbb{T}$. By CSC3, process q outputs its own input in $C'\beta_1...\beta_t$. Hence, $C'\beta_1...\beta_t$ is adjacent to w in G. Furthermore, by Lemma 18, if a (partial) terminal configuration T adjacent to w is reachable from C' via a schedule $\beta'_1,...,\beta'_t$, then $\beta'_1,...,\beta'_t$ is $\{q\}$-first. Since $m = 2$, there is only one t-round $\{q\}$-first schedule starting from C. Hence, $C'\beta_1...\beta_t$ is the only (partial) terminal configuration in $\mathbb{T}$ reachable from C' that is adjacent to w.

Now suppose $m > 2$. Consider the partial initial configuration D' of C' induced by the set of processes Q'. Let $\mathbb{H}'$ be the collection consisting of the single partial configuration D', let $\mathbb{T}' = \chi^t(D',\delta)$, and let $m' = m-1 = |\pi(D')|$. Note that, $\mathbb{H}'$ satisfies CSC4 and every subset of $\pi(D')$ identifies D' in $\mathbb{H}'$. Hence, $\mathcal{B}(\mathbb{H}')$ consists of the pairs (D',R') for all subsets $R' \subsetneq Q'$ of $m'-1$ processes.

Because $(C',Q') \in \mathcal{B}(\mathbb{H})$ and $\mathbb{H}$ satisfies CSC1, each process in Q' has a different input value in C'. Therefore, for each pair $(D',R') \in \mathcal{B}(\mathbb{H}')$, each process in $R' \subsetneq Q'$ has a different value in the partial configuration D' and, hence, $\mathbb{H}'$ satisfies CSC1. Moreover, the set of inputs $I(D',R')$ is different for each pair $(D',R) \in \mathbb{H}'$, so $\mathbb{H}'$ satisfies CSC2.

Let $\alpha' = \alpha'_1,...,\alpha'_t$ be any t-round Q'-only schedule starting from D' and let p be the only process in $\pi(C')$ that is not in $\pi(D') = Q'$. We inductively define $\phi(\alpha')$ to be the t-round $\pi(C')$-only Q'-first schedule $\alpha_1,...,\alpha_t$ starting from C', where $\alpha_i = \alpha'_i\{p\}$ if p is active in $C'\alpha_1...\alpha_{i-1}$, and $\alpha_i = \alpha'_i$ otherwise. Each process in the set Q' has the same state in $D'\alpha'$ and $C'\phi(\alpha')$ and, thus, outputs the same value in both (partial) configurations.

Consider any $(D', R') \in \mathcal{B}(\mathbb{H}')$ and any subset $S \subseteq R'$. If α' is a t-round S-first Q'-only schedule starting from D', then $\phi(\alpha')$ is a t-round S-first (and Q'-first) $\pi(C')$-only schedule starting from C'. Since $\mathbb{H}$ satisfies CSC3, each process in S outputs a value in $I(C', S) = I(D', S)$ in $C'\phi(\alpha')$. Each process in S outputs the same value in $D'\alpha'$ and $C'\phi(\alpha')$, so $\mathbb{H}'$ also satisfies CSC3. Therefore, $\mathbb{H}'$ satisfies all four computational Sperner conditions.

Fix an arbitrary pair $(D', R') \in \mathcal{B}(\mathbb{H}')$. Let $\hat{\mathbb{T}}'$ be the set of all partial terminal configuration in $\mathbb{T}'$ such that m' different values are output by the processes in Q'. Let $T' \in \hat{\mathbb{T}}'$ and let β be the t-round Q'-only schedule such that $T' = D'\beta$. Since $(C', Q') \in \mathcal{B}(\mathbb{H})$, $\phi(\beta)$ is Q'-first, and $\mathbb{H}$ satisfies CSC3, it follows that all values output by processes in Q' in the (partial) configuration $C'\phi(\beta)$ are elements of $I(C', Q')$. Each process in Q' outputs the same value in T' and $C'\phi(\beta)$, so all values output by processes in Q' in partial configuration T' are elements of $I(C', Q') = I(C, Q)$. By CSC1, $|I(C, Q)| = m - 1 = m'$, so $I(C, Q)$ is the set of values output by the processes in Q' in partial configuration T'. Hence each value in $I(D', R') \subseteq I(C, Q)$ is output by some process in T'. By the inductive hypothesis of Theorem 2 applied to $\mathbb{H}'$, it follows that $|\hat{\mathbb{T}}'|$ is odd.

Since $\phi(\beta)$ is Q'-first, Lemma 16 says that Q' identifies $C'\phi(\beta)$ in $\chi^t(C', \delta)$. Since $(C', Q') \in \mathcal{B}(\mathrm{H})$, we know that Q' also identifies C' in $\mathbb{H}$. Hence, for any other (partial) configuration $C'' \in \mathbb{H}$, some process $q \in Q'$ distinguishes between C'' and C'. Since the protocol is full-information, it follows that q also distinguishes between $C''\beta$ and $C'\beta$. This implies that Q' identifies $C'\phi(\beta)$ in $\mathbb{T}$. Hence, $C'\phi(\beta)$ is adjacent to w.

Consider any (partial) terminal configuration $T \in \mathbb{T}$ reachable from C' that is adjacent to w. By Lemma 18, there is a $\pi(C')$-only Q'-first schedule α starting from C' such that $T = C'\alpha$. Since α is Q'-first and $Q' = \pi(C') \setminus \{p\}$, it follows that $\alpha = \phi(\alpha')$ for some Q'-only schedule α' starting from D'. Since $D'\alpha'$ is indistinguishable from T to processes in Q' and $I(C, Q)$ is the set of values output by processes in Q' in T, it follows that $I(C, Q)$ is also the set of values output by processes in Q' in $D'\alpha'$ and, thus, $D'\alpha' \in \mathbb{T}'$. Hence $|\mathbb{T}'|$ is at least the number of (partial) terminal configurations in $\mathbb{T}$ reachable from C' that are adjacent to w in G.

Now consider any two different partial terminal configurations $T', T'' \in \mathbb{T}'$. Then there exists two different schedules β', β'' starting from D' such that $T' = D'\beta'$ and $T'' = D'\beta''$. Since the protocol is full-information, some process $q \in \pi(D') = Q'$ distinguishes between $D'\beta'$ and $D'\beta''$. Hence, q distinguishes between $C'\phi(\beta')$ and $C'\phi(\beta'')$. Both $C'\phi(\beta')$ and $C'\phi(\beta'')$ are (partial) terminal configurations adjacent to w. Thus $|\mathbb{T}'|$ is at most the number of (partial) terminal configurations in $\mathbb{T}$ reachable from C' that are adjacent to w in G. Therefore, $|\mathbb{T}'|$ is number of (partial) terminal configurations in $\mathbb{T}$ reachable from C' that are adjacent to w in G. Since $|\mathbb{T}'|$ is odd, the statement of the lemma follows. ◀

The next lemma follows from Lemma 18, Lemma 19, and the fact that $\mathbb{H}$ satisfies CSC2.

▶ **Lemma 20.** *Vertex w has odd degree in G.*

Proof. By Lemma 18, each (partial) terminal configuration T adjacent to w in G is reachable from C' for some $(C', Q') \in \mathcal{B}(\mathbb{H})$ such that $I(C', Q') = I(C, Q)$. By CSC2, there are an odd number of pairs $(C', Q') \in \mathcal{B}(\mathbb{H})$ such that $I(C', Q') = I(C, Q)$. For each such pair (C', Q'), Lemma 19 tells us that there are an odd number of (partial) terminal configurations in $\mathbb{T}$ reachable from C that are adjacent to w in G. Thus, w has odd degree in G. ◀

The collection $\mathbb{H}$ of initial configurations satisfies CSC4. Thus, applying Lemma 17 t times shows that, for each (partial) terminal configuration $T \in \mathbb{T} = \chi^t(\mathbb{H}, \delta)$ and each subset $Q' \subsetneq \pi(T)$ of $m - 1$ processes, there is at most one other (partial) terminal configuration

$T' \in \mathbb{T}$ such that $Q' \subsetneq \pi(T')$ and T and T' are indistinguishable to all processes in Q'. If there is such a (partial) terminal configuration T' and $I(C,Q)$ is the set of values output in T by the processes in Q', then T is adjacent to T' in G. If Q' identifies T in $\mathbb{T}$ and $I(C,Q)$ is the set of values output in T by the processes in Q', then T is adjacent to w in G. Hence, the degree of T in graph G is the number of $(m-1)$-element subsets $Q' \subsetneq \pi(T)$ such that $I(C,Q)$ is the set of values output in T by the processes in Q'.

Let $\hat{\mathbb{T}}$ be the set of all (partial) terminal configurations in $\mathbb{T}$ with odd degree in G. By the handshaking lemma, every graph must have an even number of odd degree vertices and, by Lemma 20, w has odd degree. Thus, $|\hat{\mathbb{T}}|$ is odd.

Let $T \in \mathbb{T}$. Recall that $|\pi(T)| = m$ and, by CSC1, $|I(C,Q)| = m-1$. If $I(C,Q)$ is the set of values output by $\pi(T)$ in T, then T has degree 2. If $I(C,Q)$ is a proper subset of the set of values output by $\pi(T)$ in T, then T has degree 1. Otherwise, T has degree 0.

Hence $\hat{\mathbb{T}}$ is the set of all (partial) terminal configurations $T \in \mathbb{T}$ such that m different values are output and each value in $I(C,Q)$ is output by some process. Therefore, $\hat{\mathbb{T}} = \mathbb{T}(C,Q)$. This concludes the proof of Theorem 2.