SET AGREEMENT IMPOSSIBILITY PROOF THROUGH COMBINATORIAL TOPOLOGY

ISTA Institute of Science and Technology Austria

Modern distributed algorithms,
Prof. Dan Alistarh,
February 2022

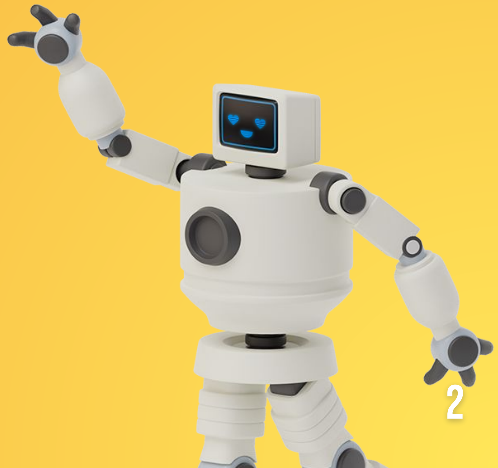
TEAM PRESENTATION

Shayan Talaei

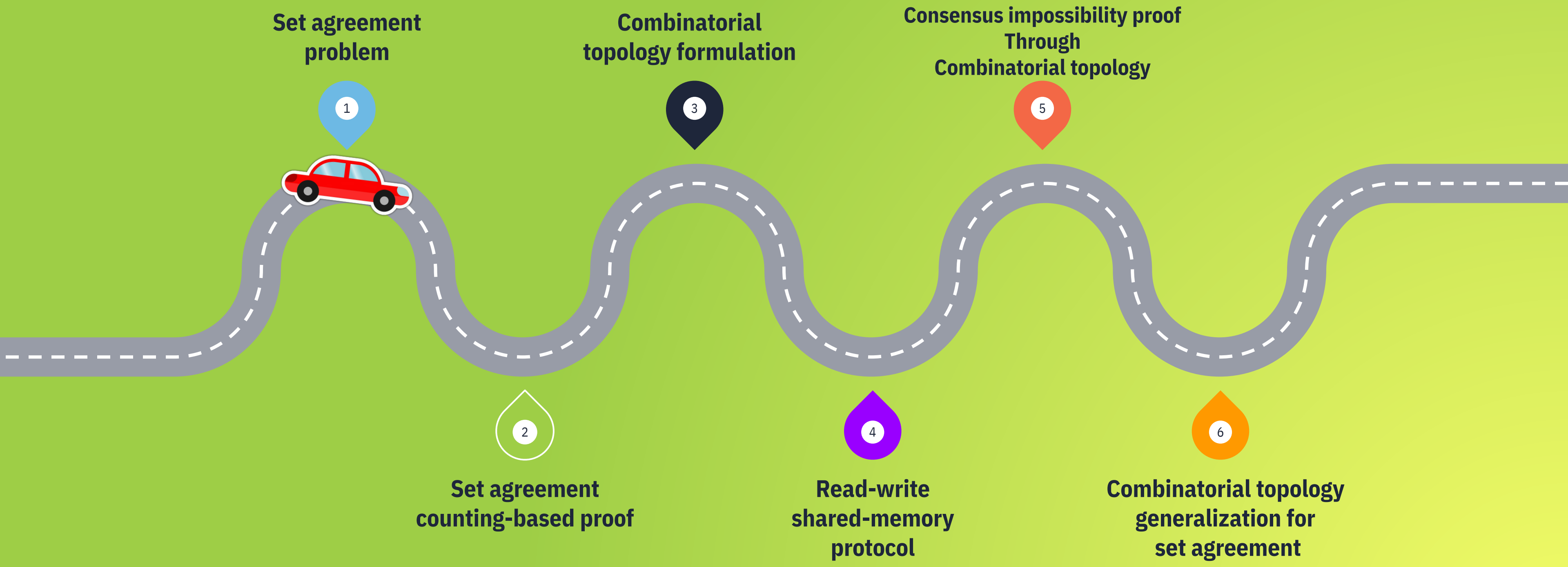
talaee.shayan@gmail.com

Matin Ansaripour

matinansaripour@gmail.com

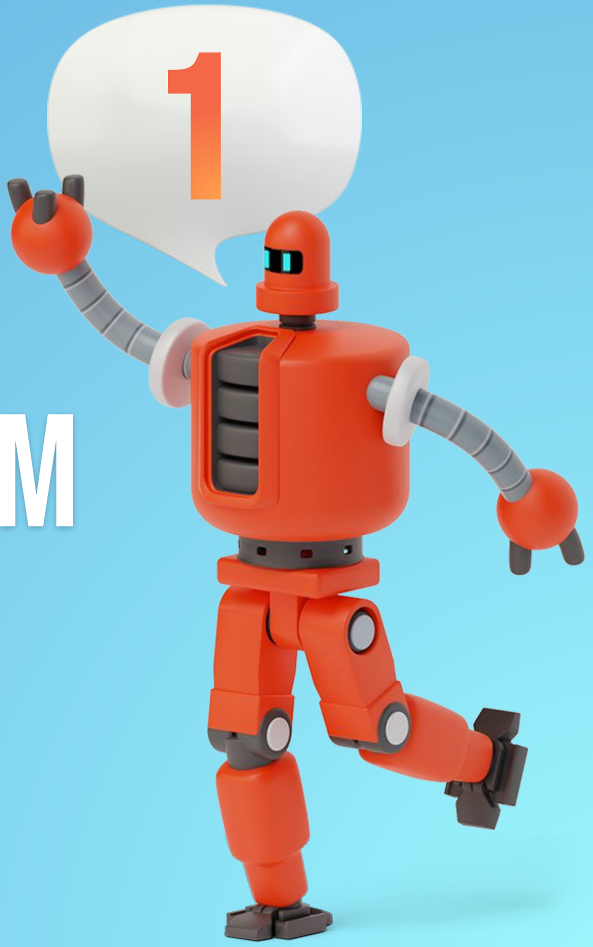


ROADMAP



1
Set agreement problem
2
Set agreement counting-based proof
3
Combinatorial topology formulation
4
Read-write shared-memory protocol
5
Consensus impossibility proof Through Combinatorial topology
6
Combinatorial topology generalization for set agreement

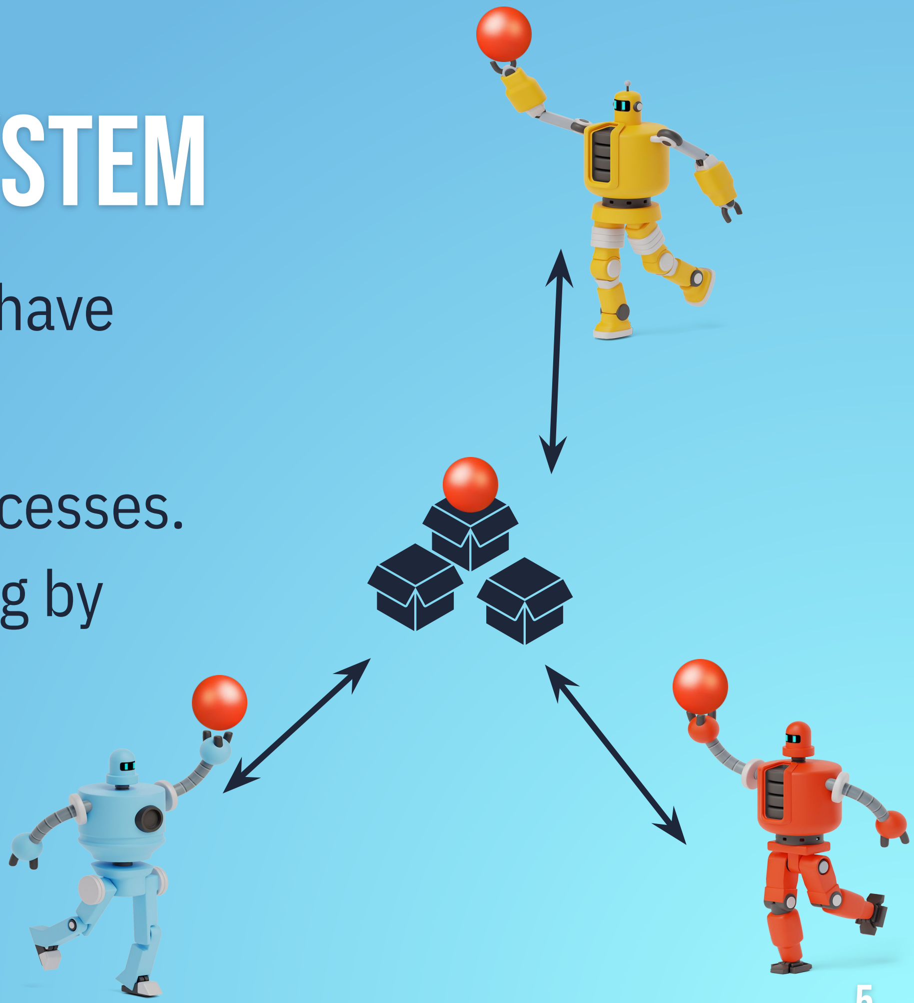
SET AGREEMENT PROBLEM



SHARED-MEMORY SYSTEM

In a shared-memory system we have the following properties.

- We have n asynchronous processes.
- Processes are communicating by writing and reading from the memory.
- A process may fail.



K-SET AGREEMENT PROBLEM

In k-set agreement problem, each process has a starting input.

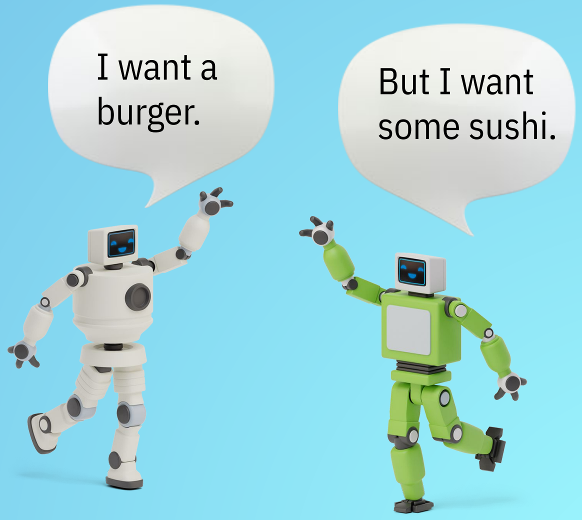
Each Process has to decide on a value such that:

k-Agreement

At most k different values are decided.

Validity

Every decided value is an input of a process.



In the consensus problem, we are trying to solve the k-set agreement when $k = 1$.

> *FLP Theorem. Consensus is impossible among n processes in asynchronous read/write shared memory if at least one of the processors can fail.*

Proof idea:

- Step 1: Bivalent Initialization

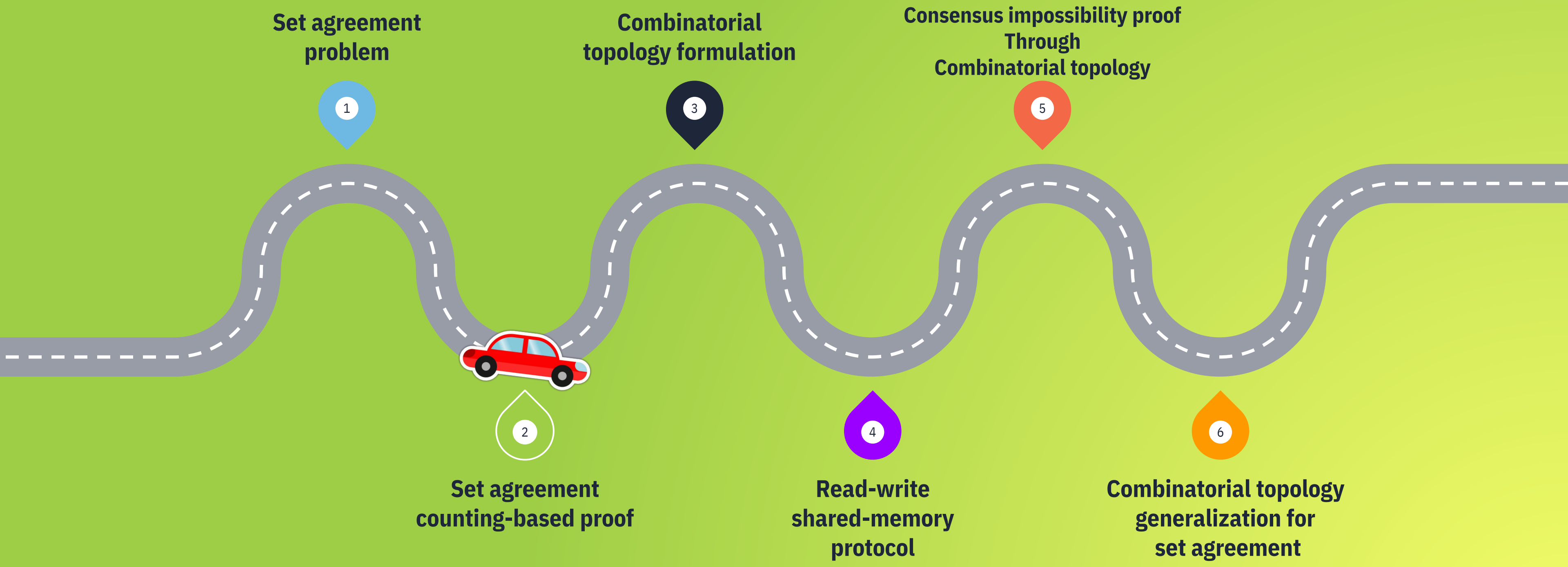
  Any consensus algorithm must have a bivalent initial configuration.

- Step 2: Bivalent Extension

  Every bivalent configuration has a bivalent extension.

No extension-based proof for k-set may exist.
(Alistarh, Aspnes, Ellen, Gelashvili, Zhu [STOC 2019])

ROADMAP


Set agreement problem
1
Set agreement counting-based proof
2
Combinatorial topology formulation
3
Read-write shared-memory protocol
4
Consensus impossibility proof Through Combinatorial topology
5
Combinatorial topology generalization for set agreement
6

SET AGREEMENT
COUNTING-BASED PROOF

SET AGREEMENT COUNTING-BASED PROOF

Theorem 1. There is no wait-free algorithm solving the (n-1)-set agreement task in an asynchronous shared memory system with n processes.

SET AGREEMENT COUNTING-BASED PROOF

Theorem 1. There is no wait-free algorithm solving the (n-1)-set agreement task in an asynchronous shared memory system with n processes.

If there is a wait-free algorithm to solve the k-set agreement for some $0 < k < n$, so we can solve (n-1)-set agreement.

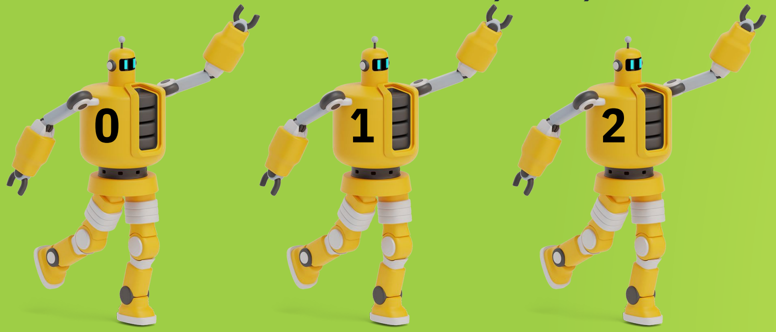
Corollary 1. For every $0 < k < n$, there is no wait-free algorithm solving the k-set agreement task in an asynchronous shared memory system with n processes.

Sketch of proof for Theorem 1:

For the sake of contradiction, assume there is a wait-free algorithm.

Let Cm, $0 < m \leq n$, be the set of all executions such that

- only the first m processes, p0, p1, ... , p(m-1) take steps.
- each pi has an input value i.
- all the values 0, ... , m-1 are decided.



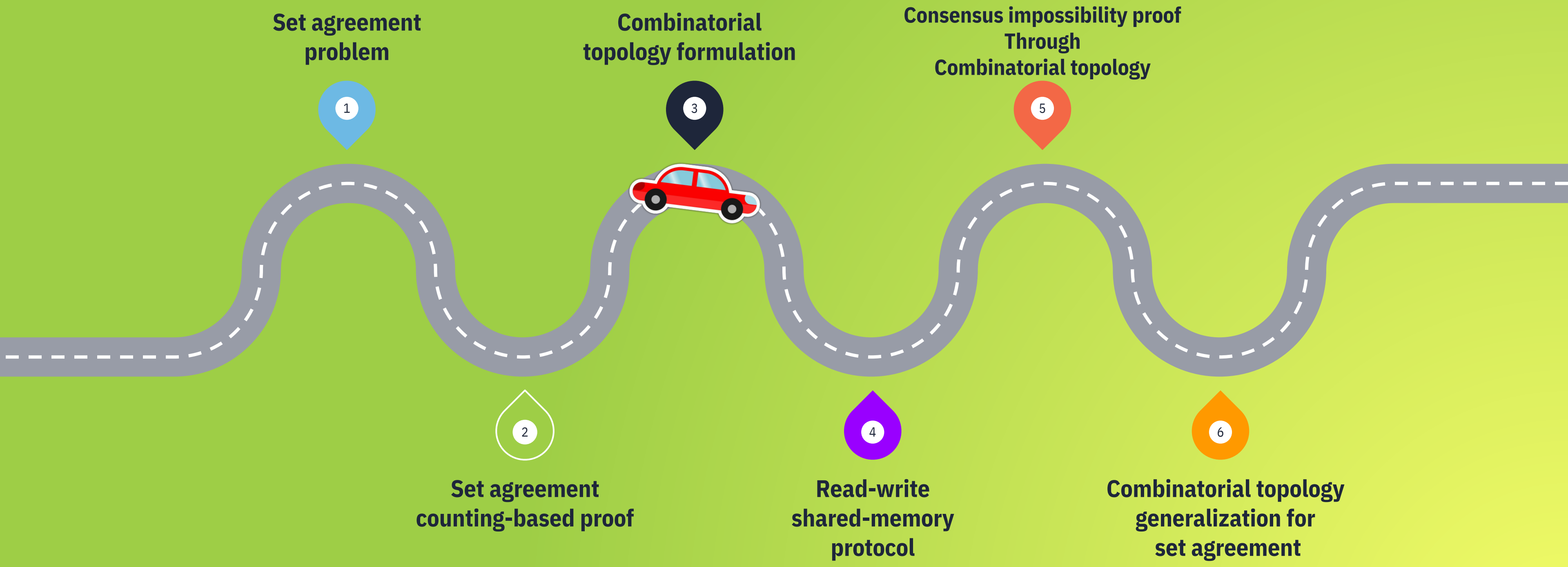
{0, 1, 2}

If $|C_n|$ is odd, then $C_n \neq \emptyset$. $\Rightarrow$ There is an execution in which n values are decided.

If $|C_n|$ is odd, then $C_n \neq \emptyset$. → There is an execution in which n values are decided.

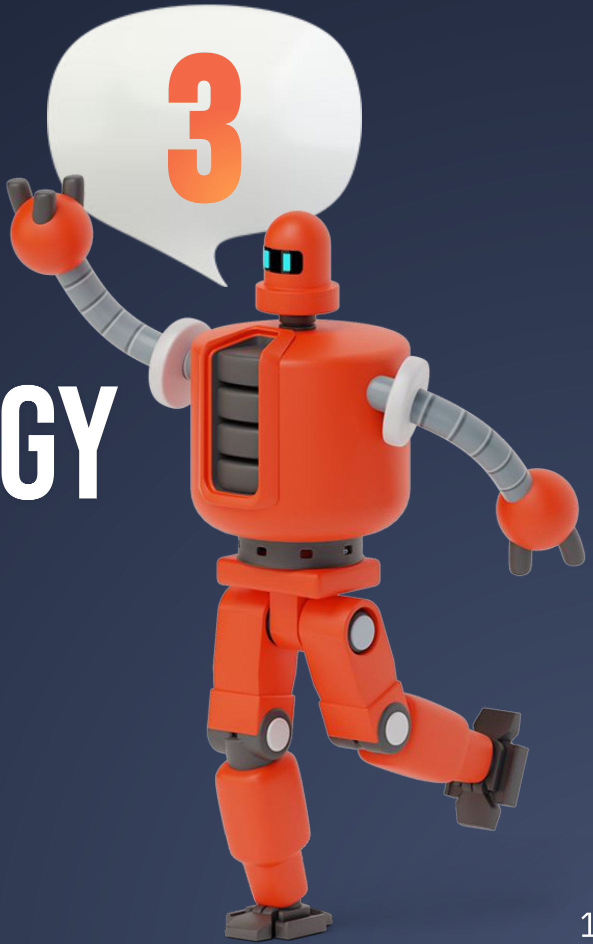
- By induction, we prove the size of C_m is odd for all $0 < m \leq n$.
- For $m = 1$ we just have a solo execution.
- Map $C(m-1)$ to a subset of C_m like A_m with the same parity.
- $|C_m \setminus A_m|$ is even.
- $|A_m|$ is odd (because $|C(m-1)|$ is odd).
- $|C_m \setminus A_m| + |A_m| = |C_m|$ is odd.

ROADMAP



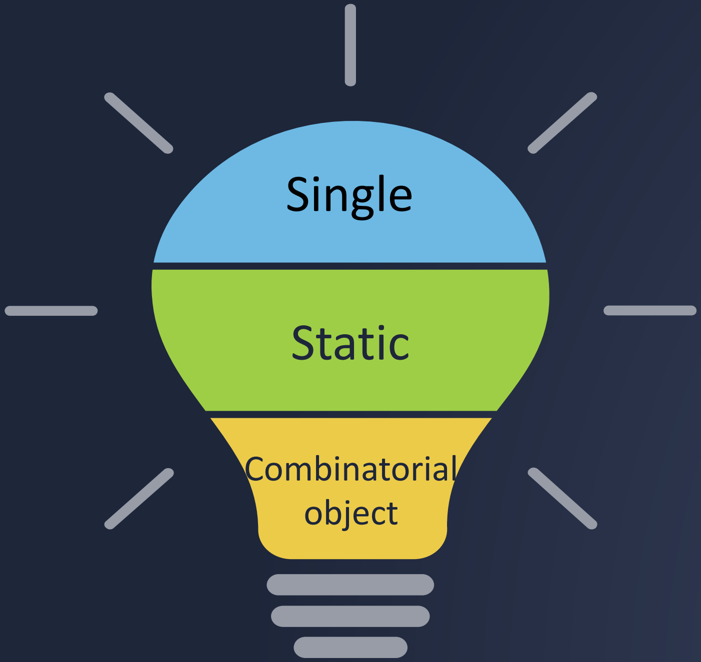
1
Set agreement problem
2
Set agreement counting-based proof
3
Combinatorial topology formulation
4
Read-write shared-memory protocol
5
Consensus impossibility proof Through Combinatorial topology
6
Combinatorial topology generalization for set agreement

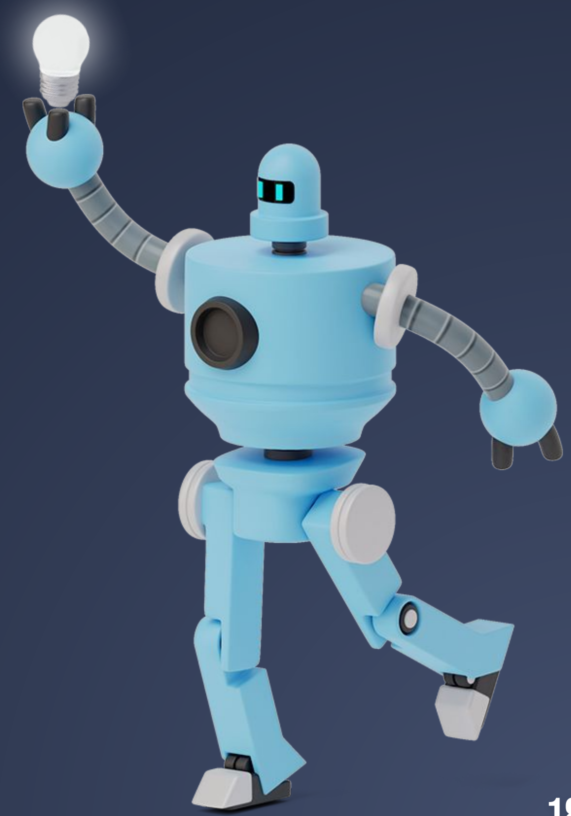
COMBINATORIAL TOPOLOGY FORMULATION



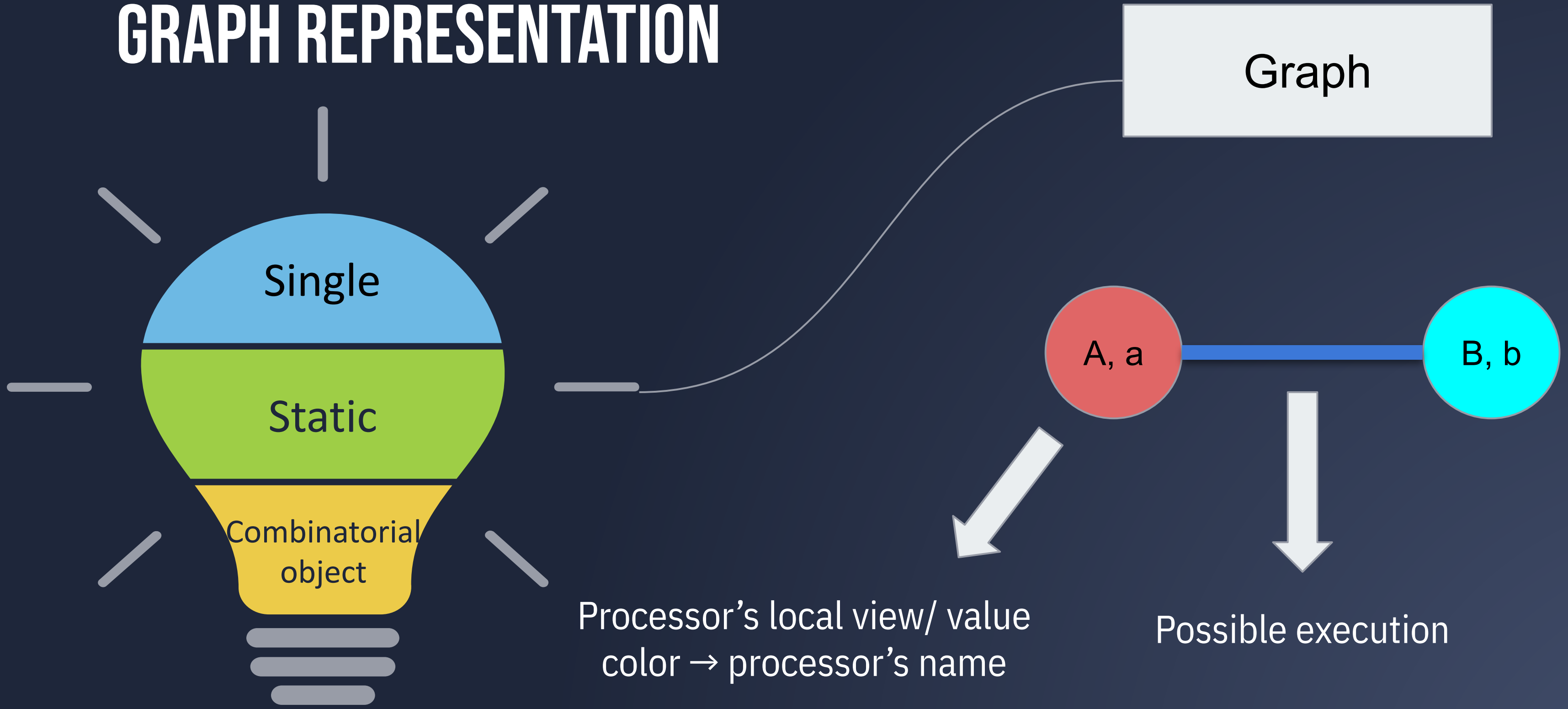
KEY IDEA

Capture all the essential properties of our system in a



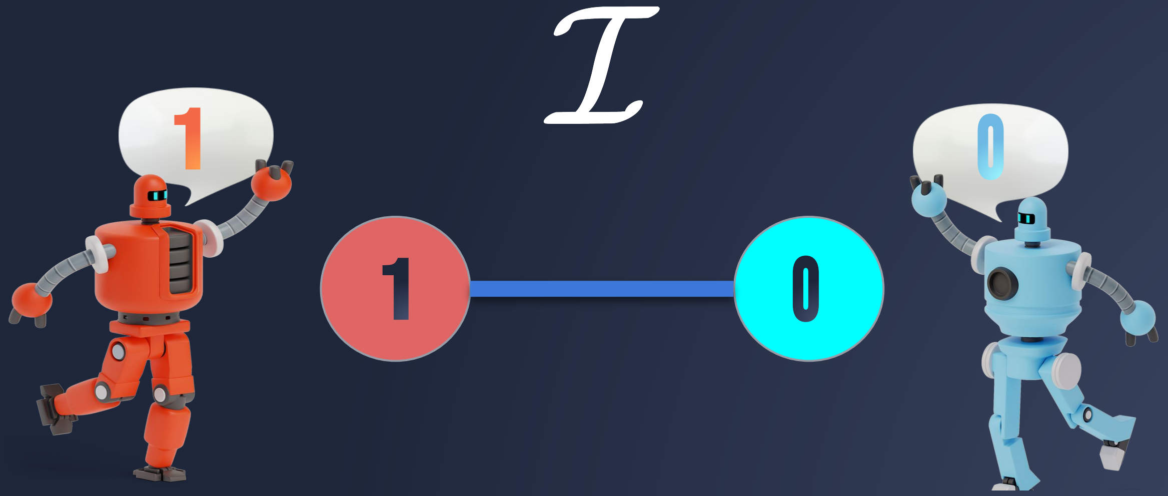


GRAPH REPRESENTATION



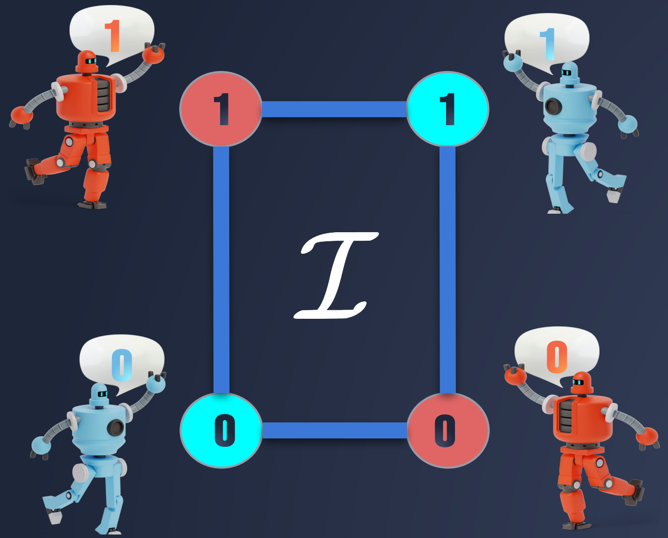
INPUT GRAPH $\mathcal{I}$

Vertices are colored by red and blue and labeled by input values.



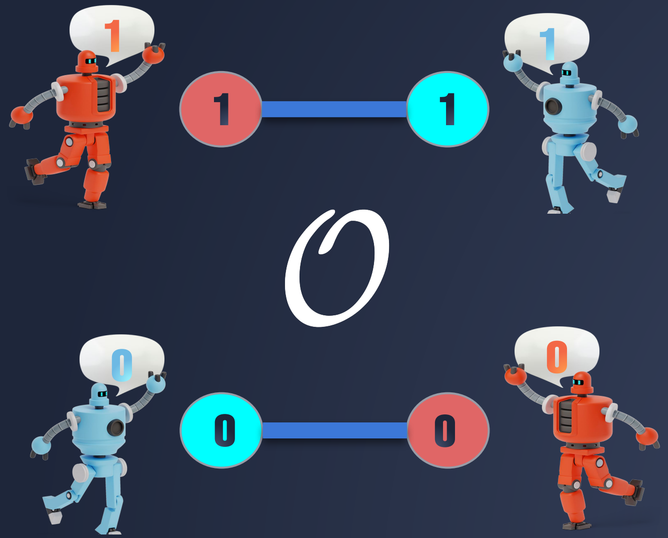
INPUT GRAPH $\mathcal{I}$

Vertices are colored by red and blue and labeled by input values.



OUTPUT GRAPH $\mathcal{O}$

Vertices are colored by red and blue and labeled by output values.

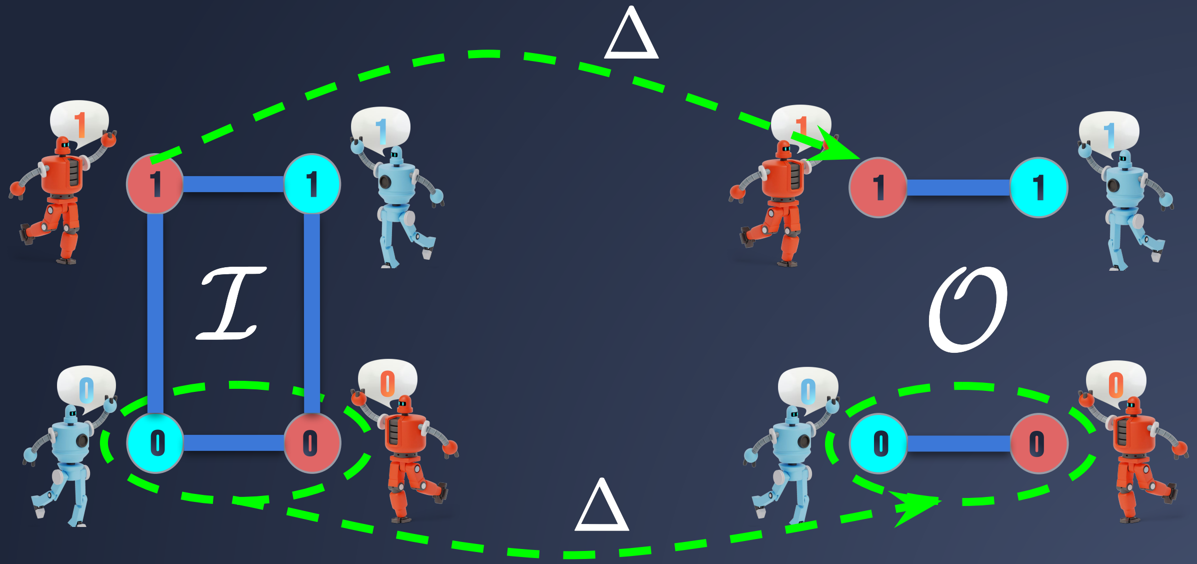


TASK $(\mathcal{I}, \mathcal{O}, \Delta)$

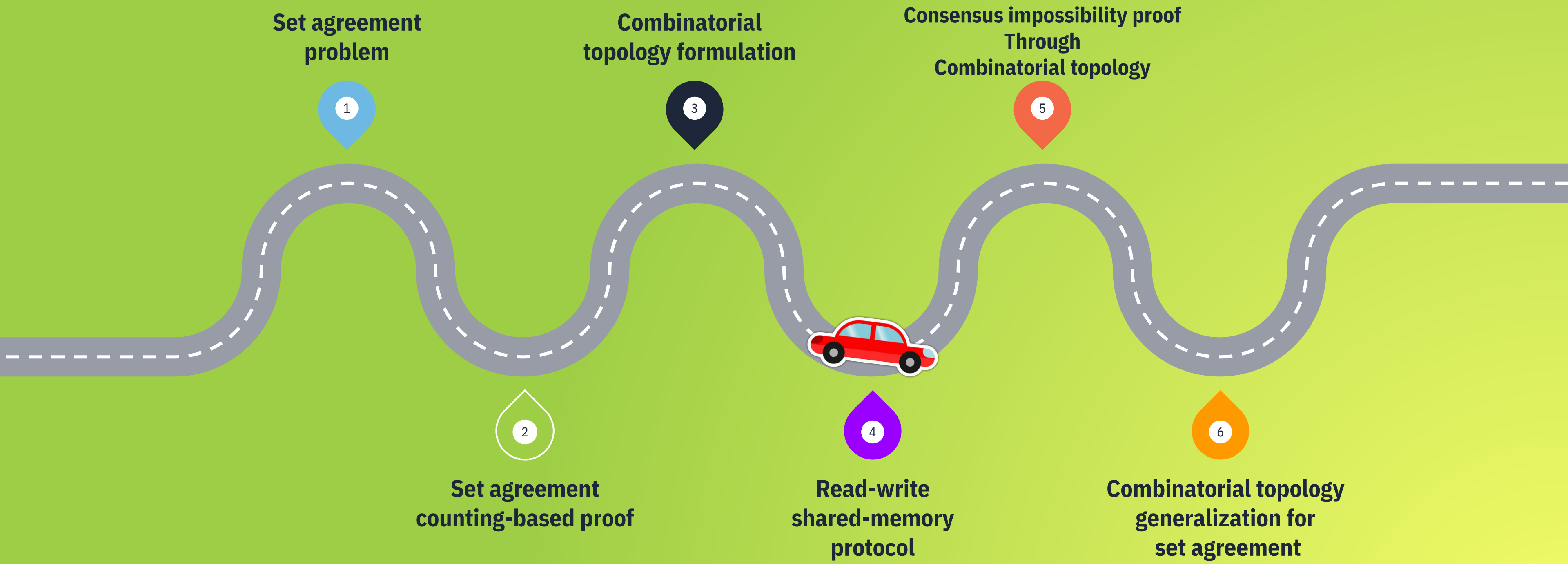
Δ is a name-preserving carrier map from $\mathcal{I}$ to $\mathcal{O}$.

Map each simplex in $\mathcal{I}$
to a subgraph of $\mathcal{O}$

- Solo-executions
- Same inputs

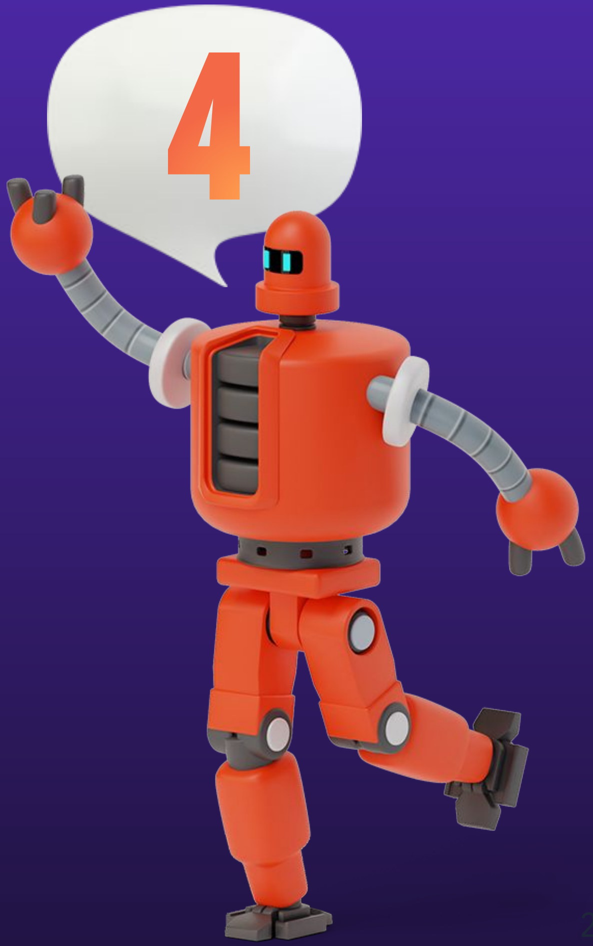


ROADMAP



Set agreement problem
1
Set agreement counting-based proof
2
Combinatorial topology formulation
3
Read-write shared-memory protocol
4
Consensus impossibility proof Through Combinatorial topology
5
Combinatorial topology generalization for set agreement
6

READ-WRITE SHARED-MEMORY PROTOCOL



PROTOCOL

How to define an algorithm from a processor's perspective?

I. Start from an initial local view → $\mathcal{I}$ the input graph

II. Perform some steps using a protocol $\Big\downarrow \Xi$ the strict carrier map

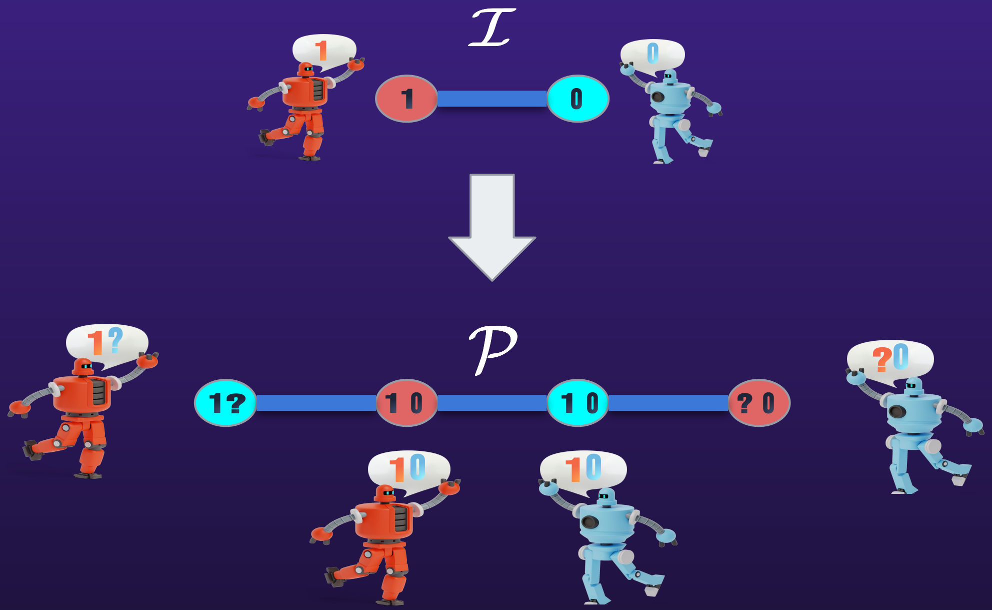
III. Until reach to a set of local views → $\mathcal{P}$ the *protocol graph*

IV. Output based on your local view → δ the *decision map*

Note that Ξ and $\mathcal{P}$ are independent from the task.Therefore, solvability of the task using the protocol is down to finding δ properly.

PROTOCOL GRAPH $\mathcal{P}$

💡 *Key Idea* → finding **all possible local views** starting from an input graph $\mathcal{I}$ using a defined **protocol**



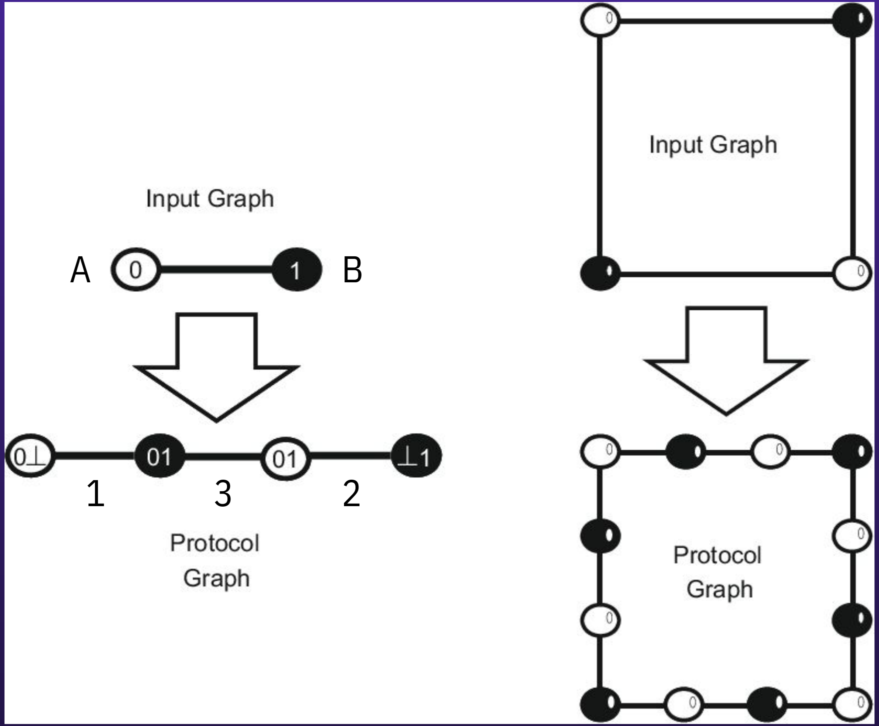
PROTOCOL GRAPH $\mathcal{P}$

Layered read-write protocol

In each round first write and then read

What happens if
1. A reads before write of B?
2. B reads before write of A?
3. Both writes be before both reads?

Simply each edge subdivides into three edges.

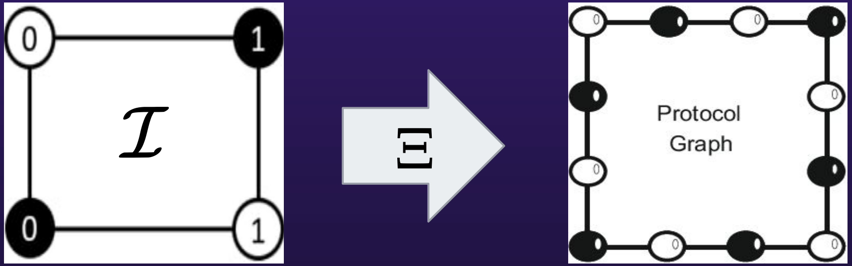


PROTOCOL GRAPH $\mathcal{P}$

Layered read-write protocol

In a single round, each edge $\xrightarrow{\text{subdivides}}$ three edges

Connected input graph $\xrightarrow{\text{Read-write shared-memory protocol}}$ Connected protocol graph

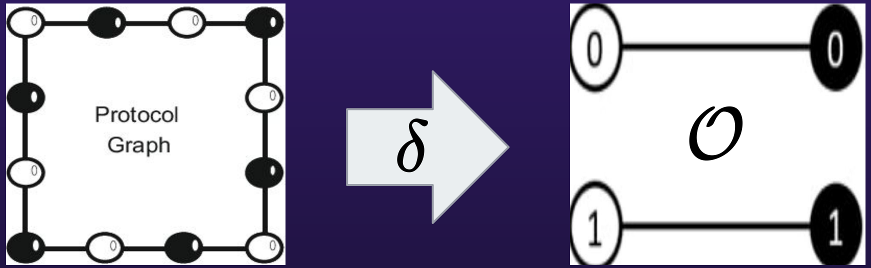


DECISION MAP δ

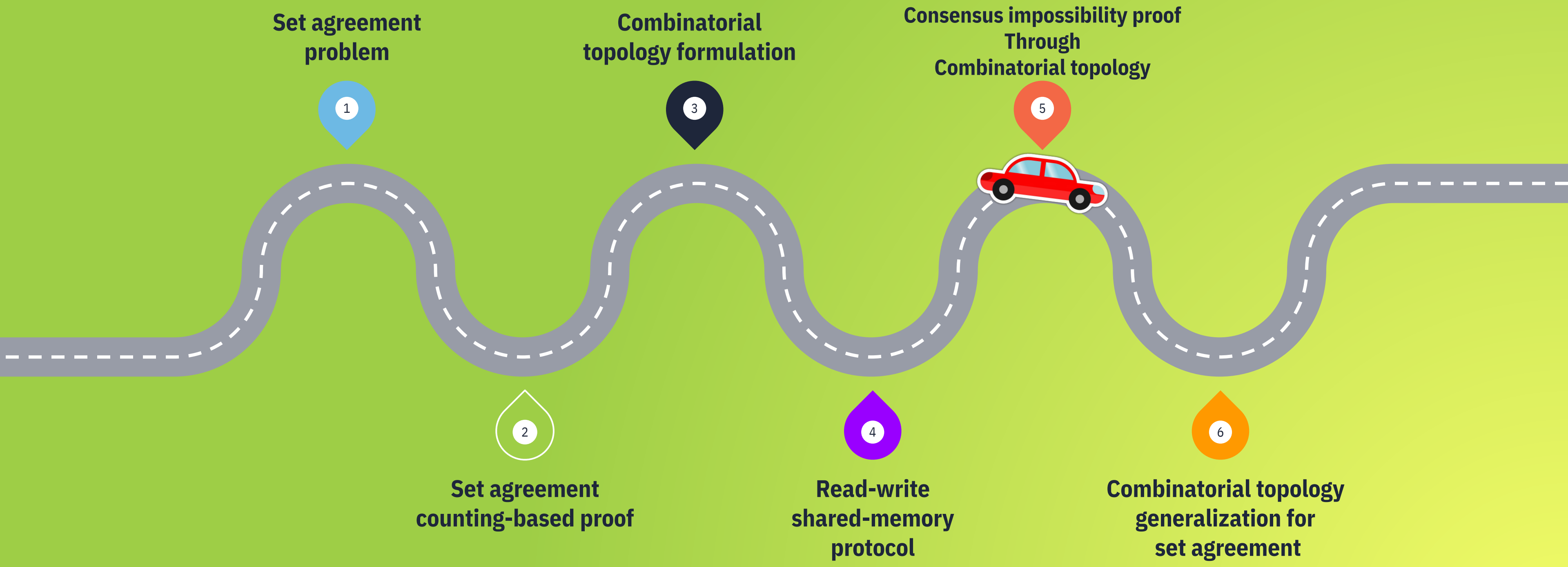
Decision map is a *simplicial map*, which maps each simplex of $\mathcal{P}$, vertices and edges, to a simplex of $\mathcal{O}$, i.e.,

- Vertices: each local view $\longrightarrow$ Output value
- Edges: each possible execution $\longrightarrow$ Valid execution in output graph

Connected protocol graph $\xrightarrow{\text{Decision map}}$ Connected output graph

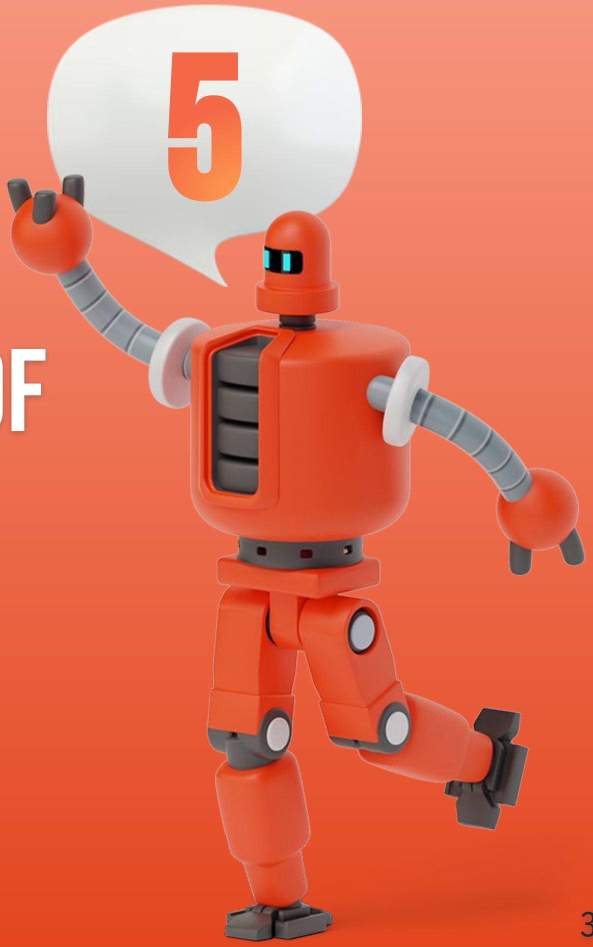


ROADMAP



Set agreement problem
1
Set agreement counting-based proof
2
Combinatorial topology formulation
3
Read-write shared-memory protocol
4
Consensus impossibility proof Through Combinatorial topology
5
Combinatorial topology generalization for set agreement
6

CONSENSUS IMPOSSIBILITY PROOF
THROUGH
COMBINATORIAL TOPOLOGY

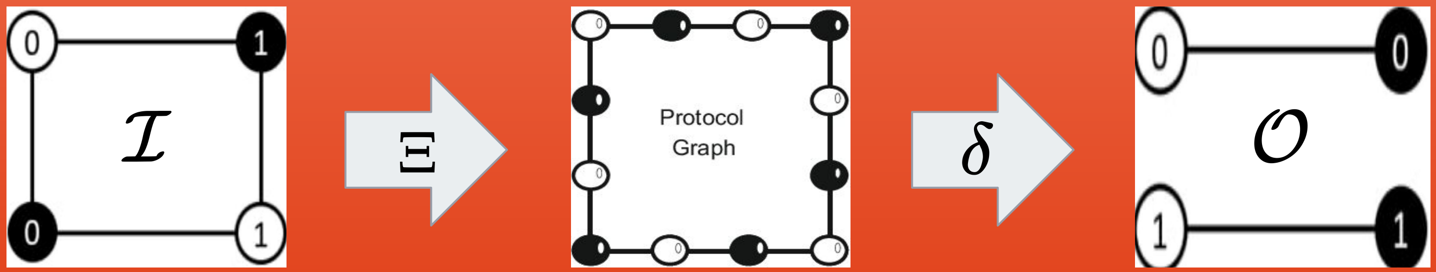


CONNECTIVITY

Connected input graph $\xrightarrow{\text{Read-write shared-memory protocol}}$ Connected protocol graph

Connected protocol graph $\xrightarrow{\text{Decision map}}$ Connected output graph

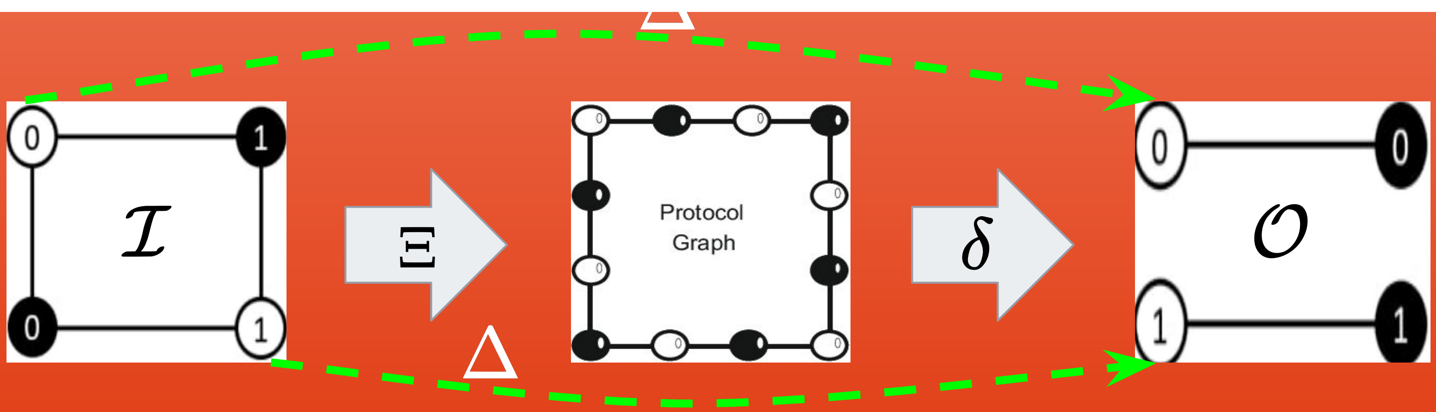
Read-write shared memory cannot carry a disconnected carrier map Δ.



TWO-PROCESS CONSENSUS UNSOLVABILITY

Two-process consensus Δ is a disconnected carrier map.

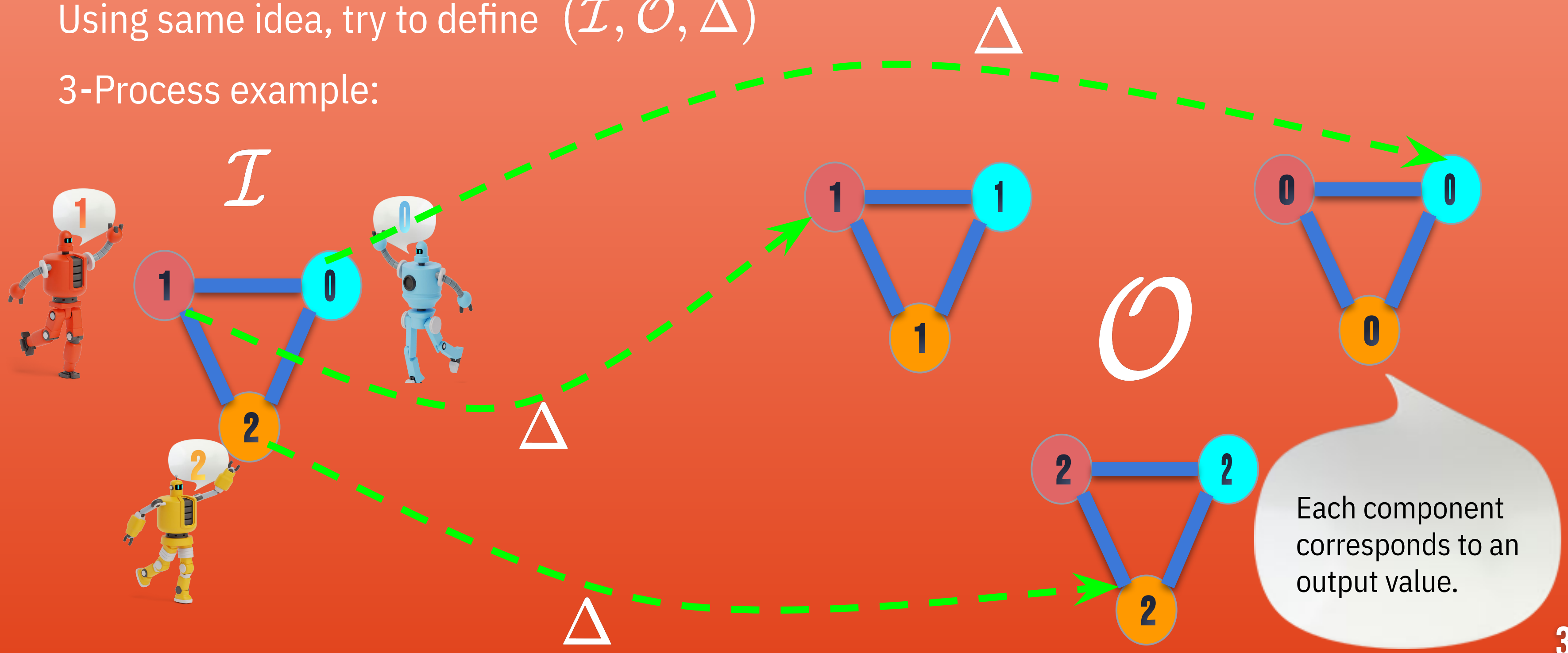
Two-process consensus using read-write shared memory is not possible!



N-PROCESS CONSENSUS UNSOLVABILITY

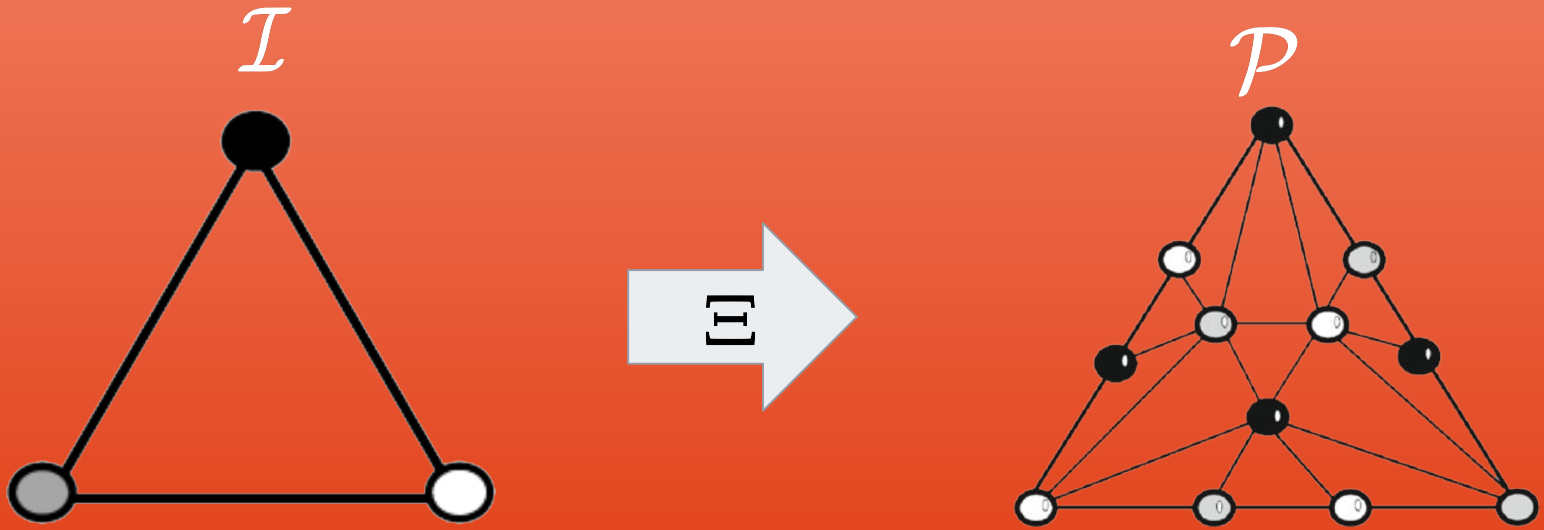
Using same idea, try to define $(\mathcal{I}, \mathcal{O}, \Delta)$

3-Process example:



READ-WRITE SHARED MEMORY PROTOCOL GRAPH

- Vertices → processors' local view
- Edges → possible executions

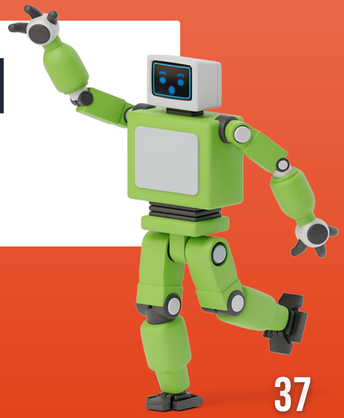


N-PROCESS CONSENSUS UNSOLVABILITY

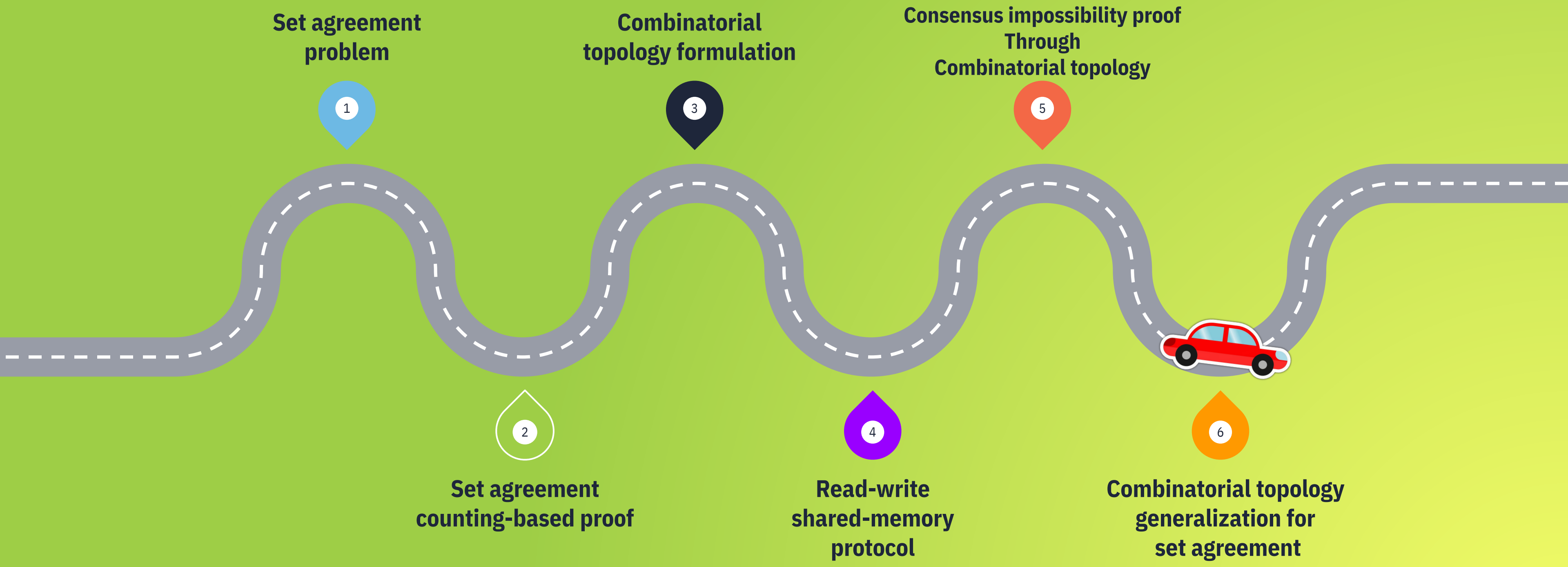
Two-process consensus Δ is a disconnected carrier map.

Read-write shared memory cannot carry a disconnected carrier map Δ.

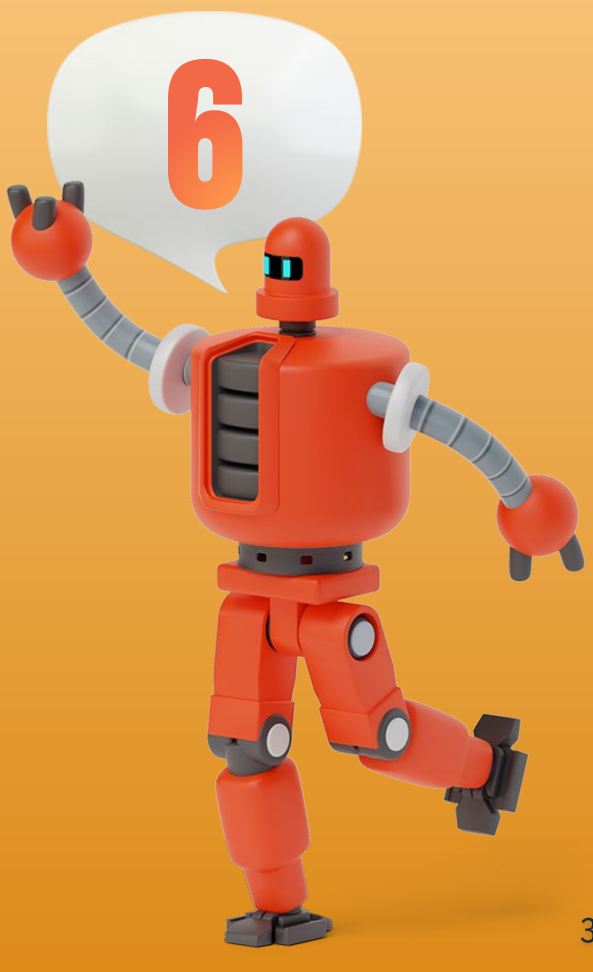
N-process consensus using read-write shared memory is not possible!



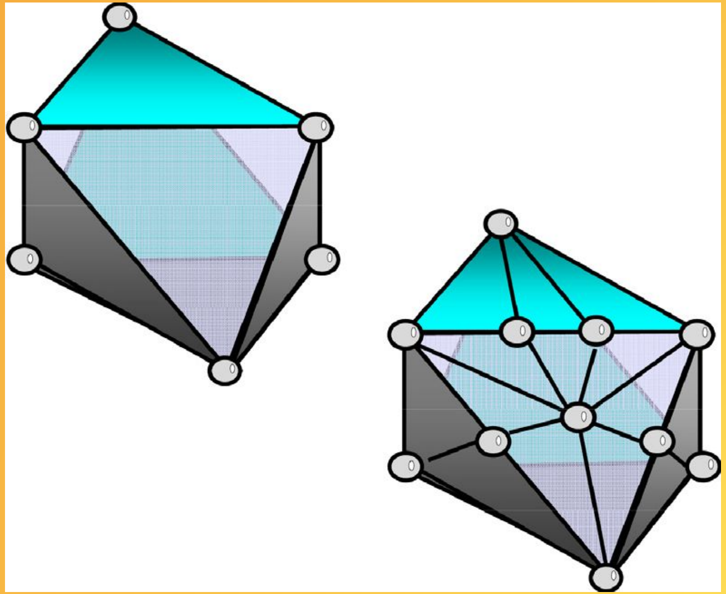
ROADMAP


1
Set agreement problem
2
Set agreement counting-based proof
3
Combinatorial topology formulation
4
Read-write shared-memory protocol
5
Consensus impossibility proof Through Combinatorial topology
6
Combinatorial topology generalization for set agreement

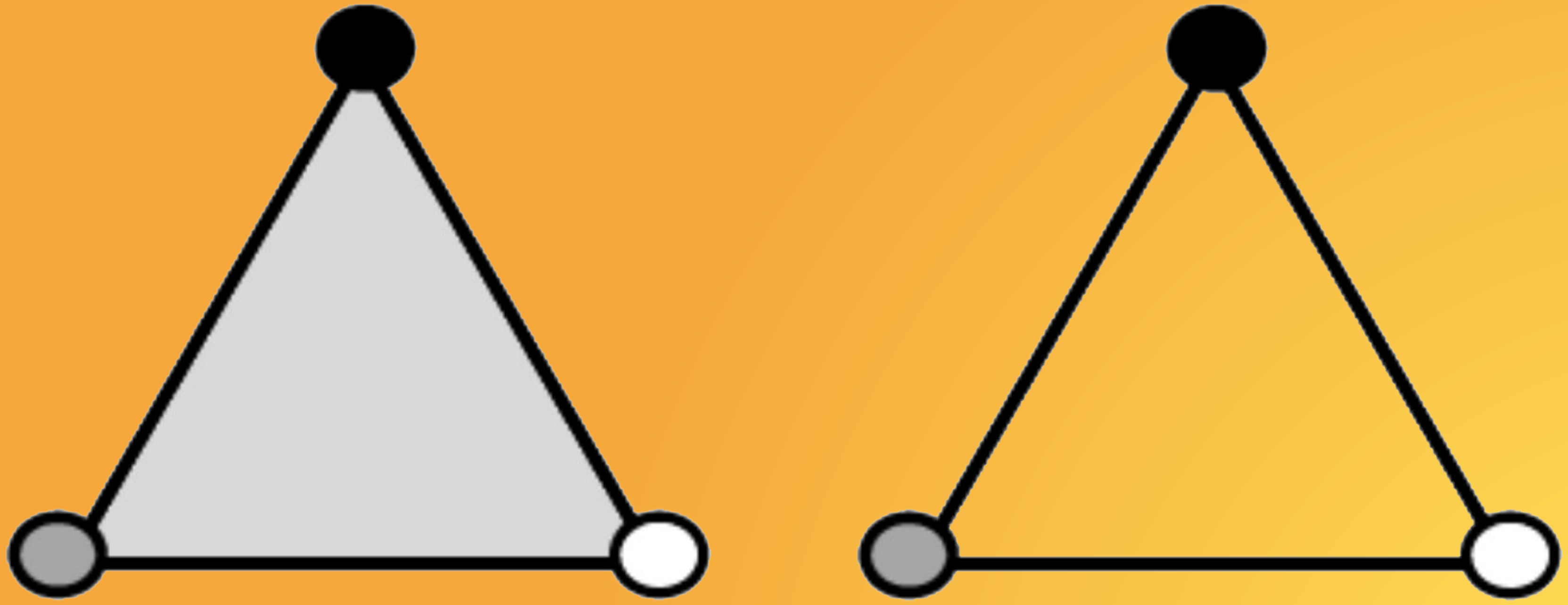
COMBINATORIAL TOPOLOGY GENERALIZATION FOR SET AGREEMENT



How to generalize the combinatorial topology approach for k-set agreement?



First, we have to extend models and definitions to higher dimensions.



The above models have different meanings.

TWO DIFFERENT VIEWS

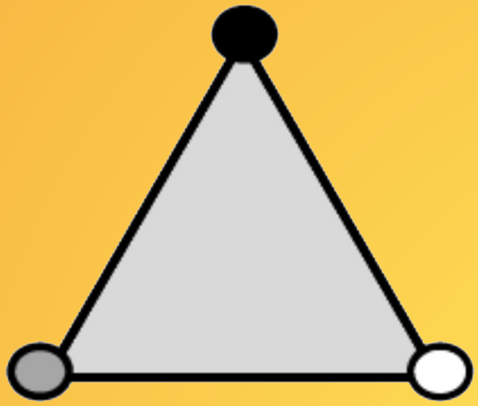
- **The combinatorial view:** A subset of vertices is called a simplex. A simplex X is said to have dimension |X|-1 .

$$\{1, 2, 3\}$$

TWO DIFFERENT VIEWS

- **The combinatorial view:** A subset of vertices is called a simplex. A simplex X is said to have dimension |X|-1 .
- **The geometric view:** a geometric simplex of dimension n, is the convex hull of some affinely independent points in $\mathbb{R}^d$ (d≥n).

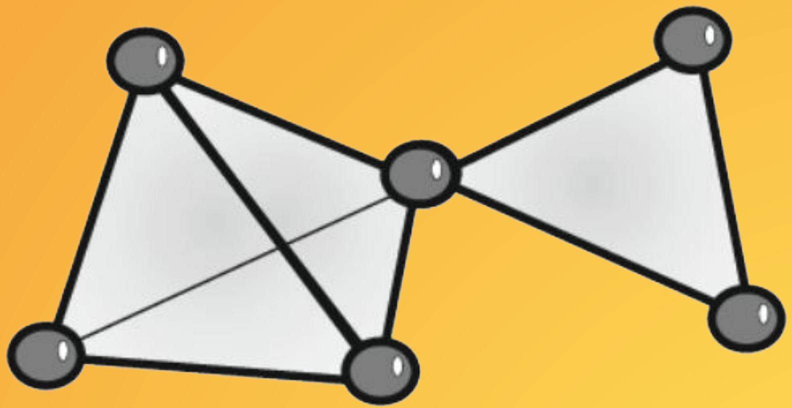
$$\{1, 2, 3\}$$



Abstract simplicial complex

Let S be the set of the vertices. A family T of finite subsets of S, we say that T is an abstract simplicial complex on S if the following are satisfied:

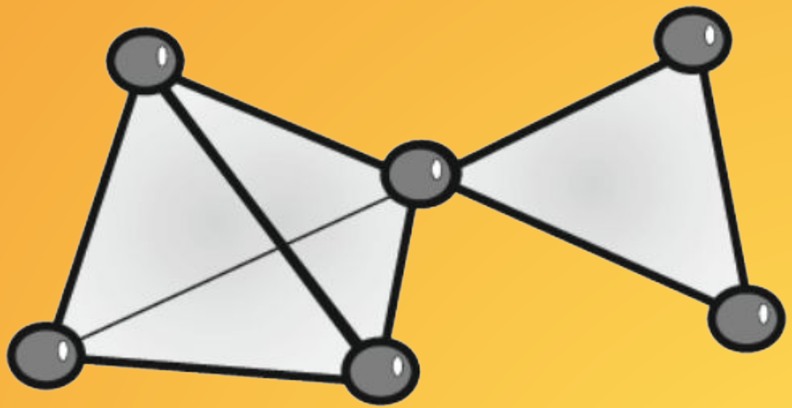
1. If $X \in T$, and $Y \subseteq X$, then $Y \in T$.
2. $\{v\} \in T$ for all $v \in S$.



Geometric simplicial complex

A geometric simplicial complex K in $\mathbb{R}^d$ is a collection of geometric simplices, such that

1. For every X $\in$ K, any convex hull of a subset from X's vertices is also in K.
2. For all X, Y $\in$ K, their intersection X∩Y is a convex hall is in each of them.



Defining the relation between geometric and combinatorial views

Given a geometric simplicial complex K, we define the underlying abstract simplicial complex C(K) as follows:

Defining the relation between geometric and combinatorial views

Given a geometric simplicial complex K, we define the underlying abstract simplicial complex C(K) as follows:

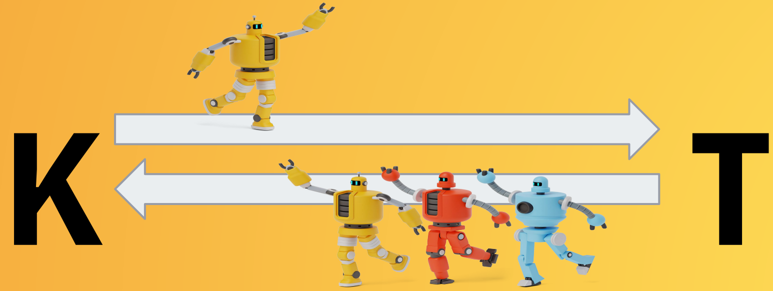
- Take the union of all the sets of vertices of the simplices of K as the vertices of C(K).
- For each simplex of K in the form of the convex hull of {v0, v1, ... , vn}, take the set
  {v0, v1, ... , vn} to be a simplex of C(K).

Defining the relation between geometric and combinatorial views

Given a geometric simplicial complex K, we define the underlying abstract simplicial complex C(K) as follows:

- Take the union of all the sets of vertices of the simplices of K as the vertices of C(K).
- For each simplex of K in the form of the convex hull of {v0, v1, ... , vn}, take the set
  {v0, v1, ... , vn} to be a simplex of C(K).

For a abstract simplicial complex T, there exist many geometric simplicial complexes K, such that C(K) = T.



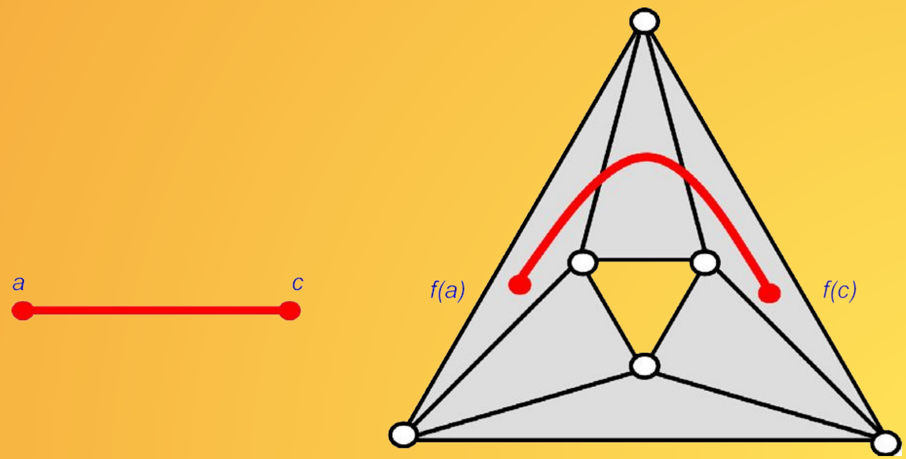
Let K be a geometric simplicial complex K and T be a abstract simplicial complex such that $C(K) = T$.

Let K be a geometric simplicial complex K and T be a abstract simplicial complex such that $C(K) = T$.

- $|K|$ is the union of its simplices, called its polyhedron.
- $|T| = |K|$.

k-connectivity

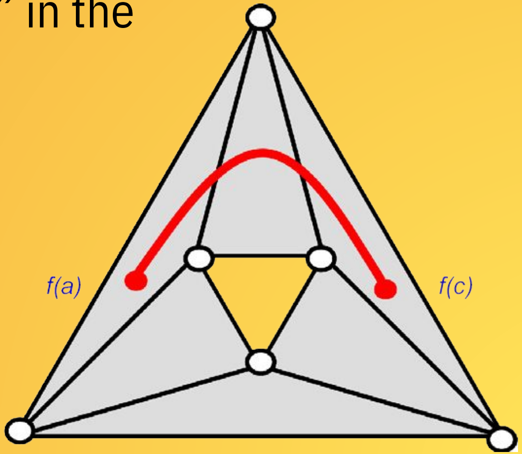
Let k be any positive integer. The complex K is k-connected if, for all $0 \leq l \leq k$, and continuous map $f : S^l \to |K|$ can be extended to $F : D^{(l+1)} \to |K|$, where the sphere S^l is the boundary of the disk $D^{(l+1)}$.



k-connectivity

Let k be any positive integer. The complex K is k-connected if, for all $0 \le l \le k$, and continuous map $f : S^l \to |K|$ can be extended to $F : D^{(l+1)} \to |K|$, where the sphere S^l is the boundary of the disk $D^{(l+1)}$.

One way to think about that there is no k-dimensional “hole” in the complex.

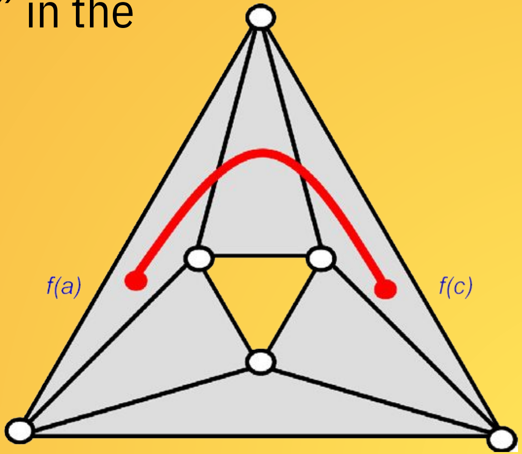


k-connectivity

Let k be any positive integer. The complex K is k-connected if, for all $0 \leq l \leq k$, and continuous map $f : S^l \rightarrow |K|$ can be extended to $F : D^{(l+1)} \rightarrow |K|$, where the sphere S^l is the boundary of the disk $D^{(l+1)}$.

One way to think about that there is no k-dimensional “hole” in the complex.

A carrier map from a k-connected complex G to a complex H is k-connected if the image of G under the map is still k-connected.



Impossibility of k-set agreement proof using combinatorial topology

Theorem 2. Let I be an input complex for k-set agreement. If (I, O, D) is an (n+1)-process k-set agreement task, and (I, P, m) is a protocol such that m is (k-1)-connected for simplices in I, then (I, P, m) cannot solve the k-set agreement task (I, O, D).

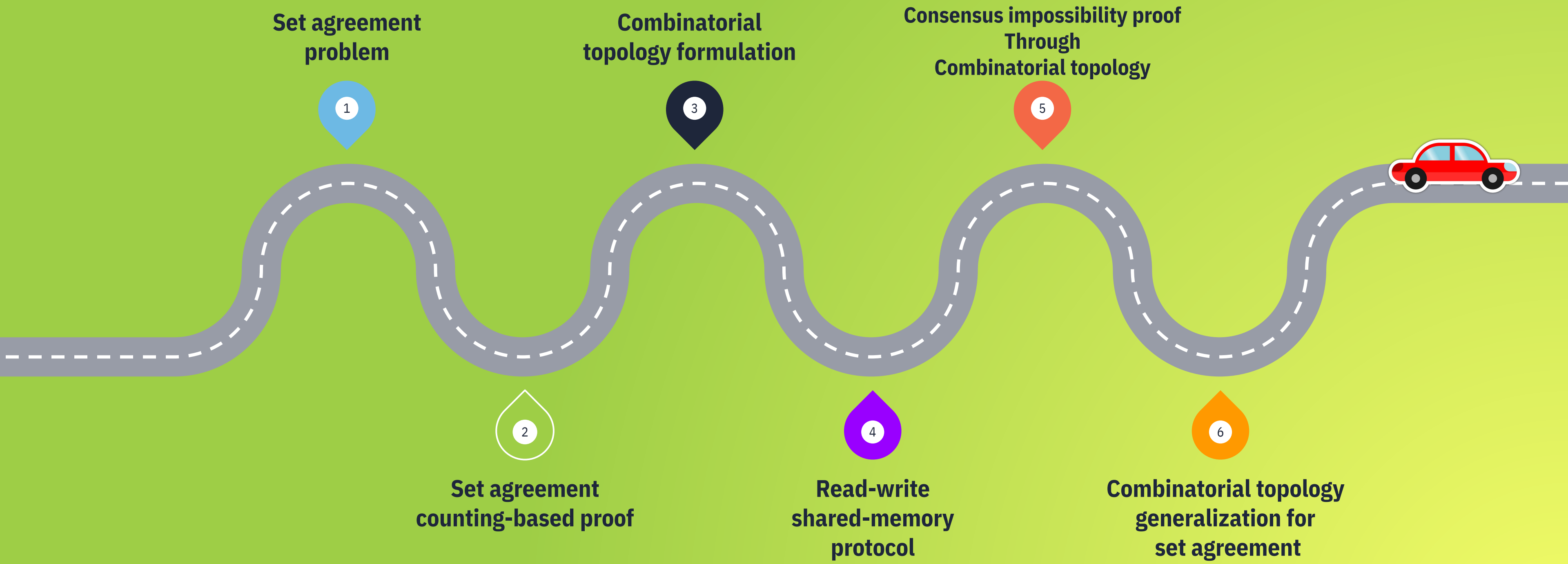
Impossibility of k-set agreement proof using combinatorial topology

Theorem 2. Let I be an input complex for k-set agreement. If (I, O, D) is an (n+1)-process k-set agreement task, and (I, P, m) is a protocol such that m is (k-1)-connected for simplices in I, then (I, P, m) cannot solve the k-set agreement task (I, O, D).

Proof idea:

1. m is (k-1)-connected.
2. P is (k-1)-connected.
3. The image of I under D is not (k-1)-connected.
4. There is no simplicial map from P to a subset of D(I).

ROADMAP


Set agreement problem
1
2
Set agreement counting-based proof
Combinatorial topology formulation
3
4
Read-write shared-memory protocol
Consensus impossibility proof Through Combinatorial topology
5
6
Combinatorial topology generalization for set agreement

ANY QUESTIONS?

You can find us at:

- matinansaripour@gmail.com
- talaee.shayan@gmail.com

RESOURCES

[1] Attiya H., Paz A. (2012) Counting-Based Impossibility Proofs for Renaming and Set Agreement. In: Aguilera M.K. (eds) Distributed Computing. DISC 2012. Lecture Notes in Computer Science, vol 7611. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33651-5_25

[2] Herlihy, M., Kozlov, D., & Rajsbaum, S. (2013). Distributed computing through combinatorial topology. Newnes.