Combinatorial Topology
and
Distributed Computing

Maurice Herlihy

BROWN

Overview

Basic concepts of Combinatorial Topology

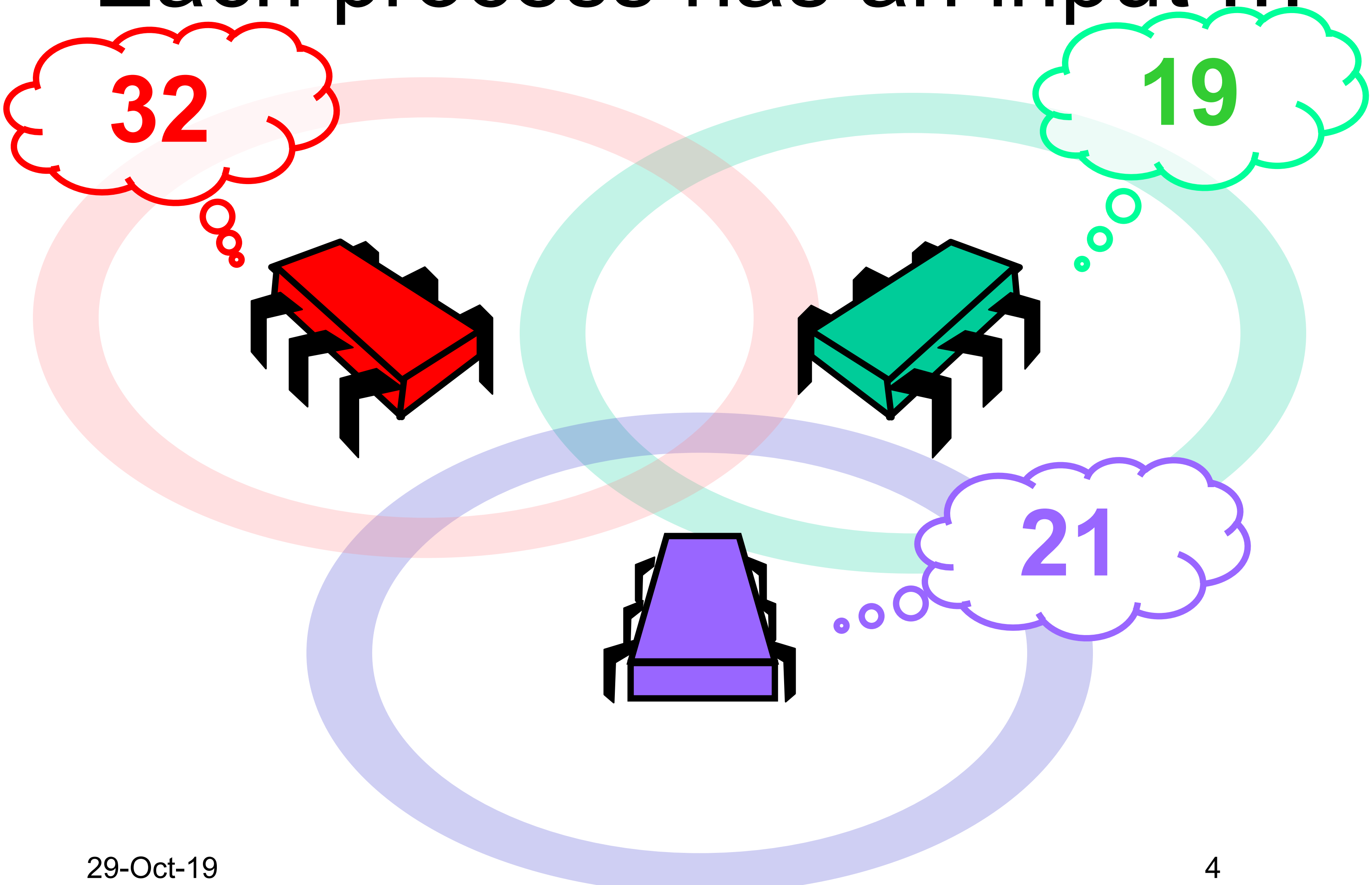
How they model distributed & concurrent computation

But first, two puzzles

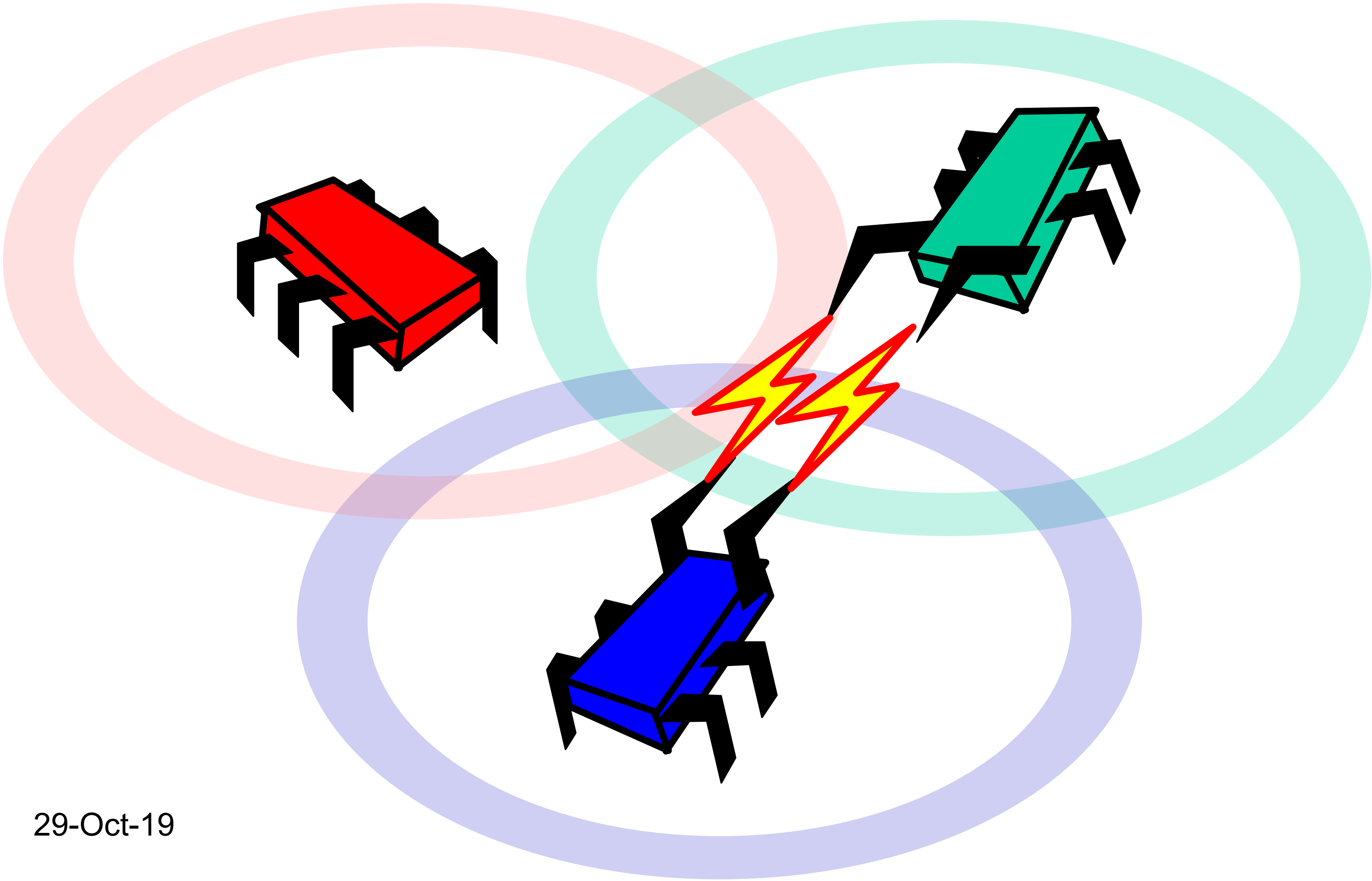
Consensus

k-set agreement

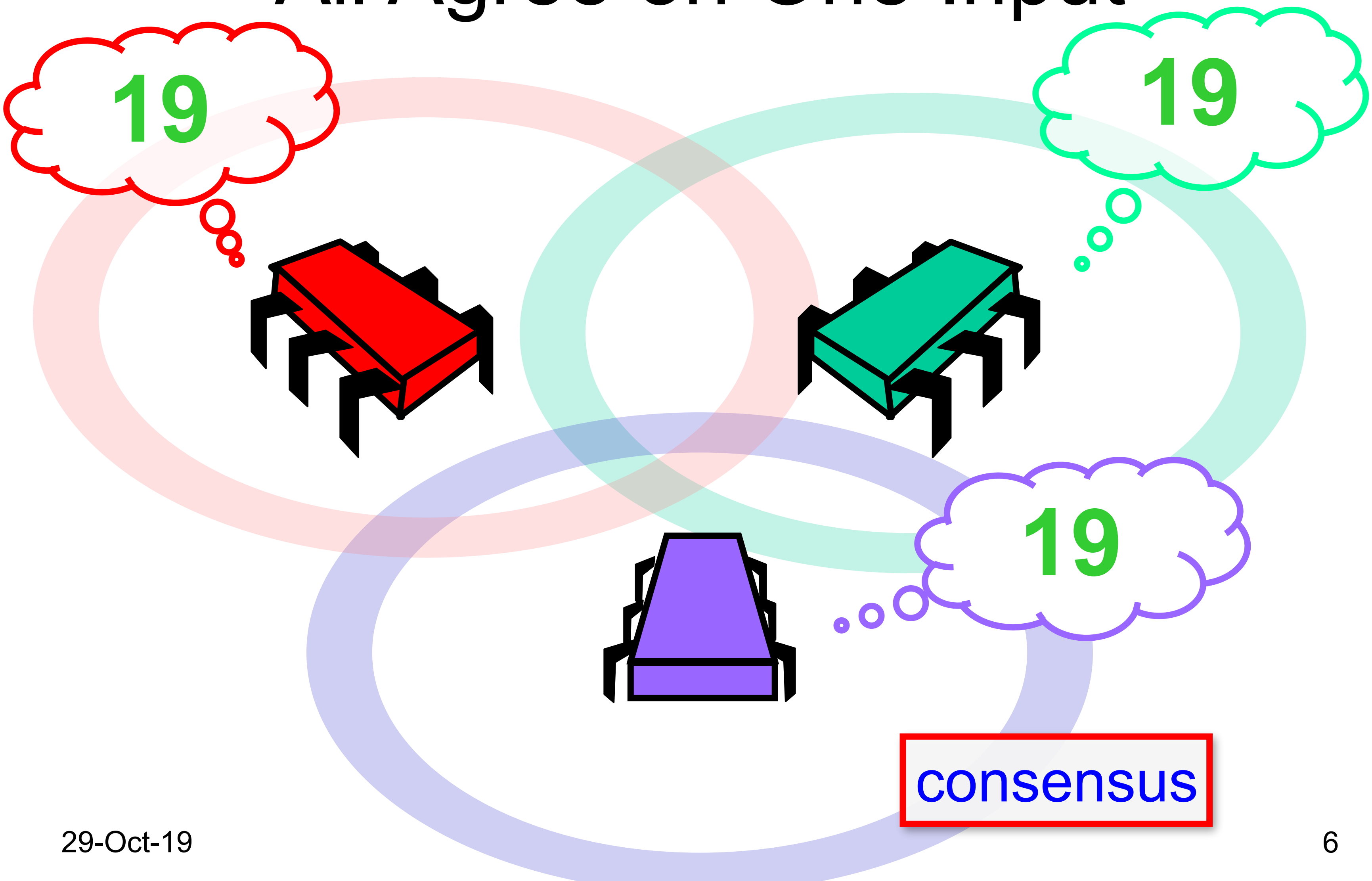
Each process has an input ...



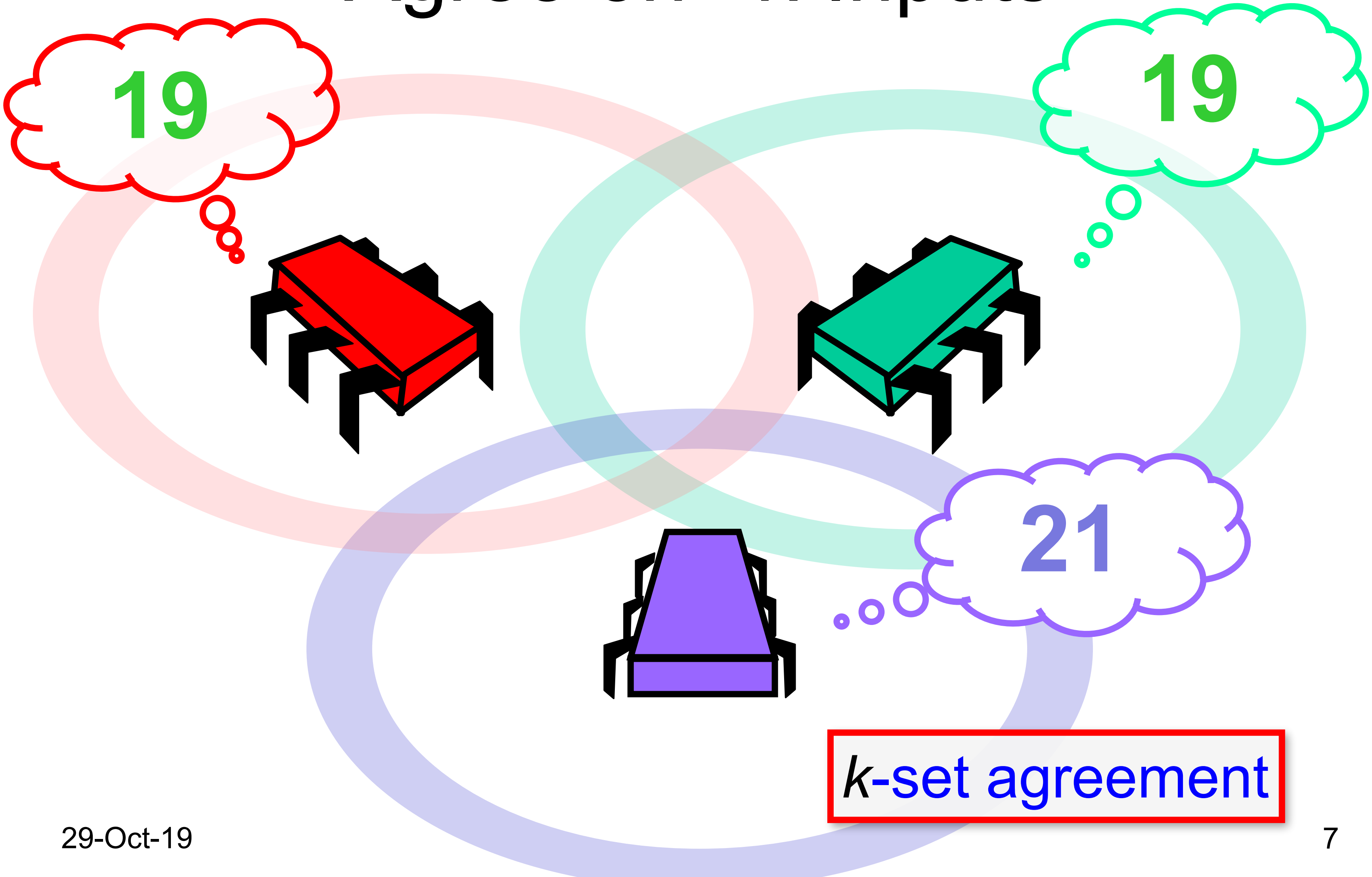
They Communicate …



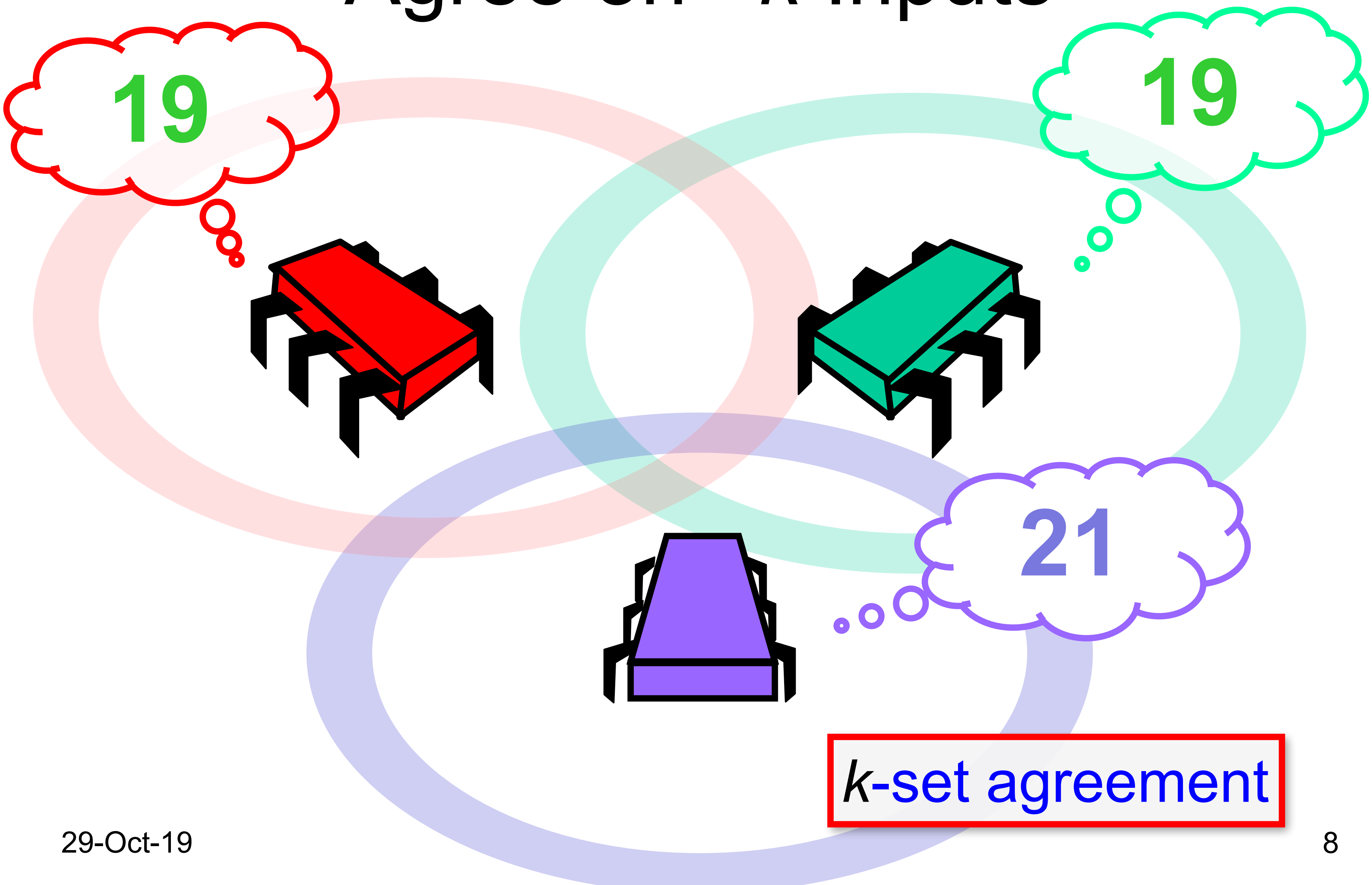
All Agree on One Input



Agree on · *k* Inputs



Agree on · *k* Inputs

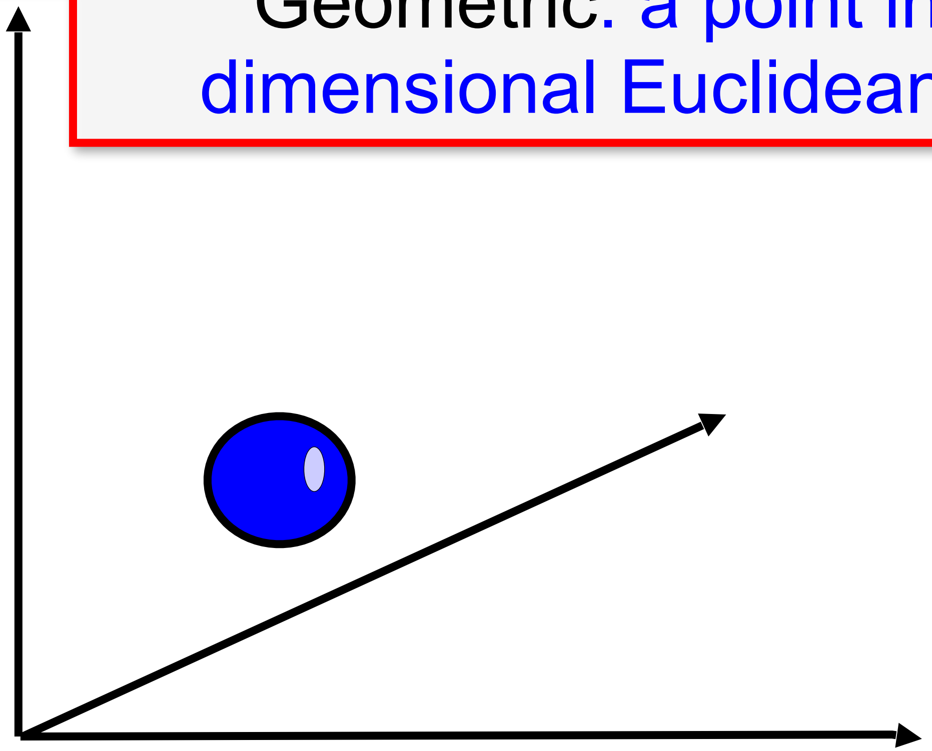


Combinatorial Topology
(standing on one foot)

A Vertex

Combinatorial: an element of a set

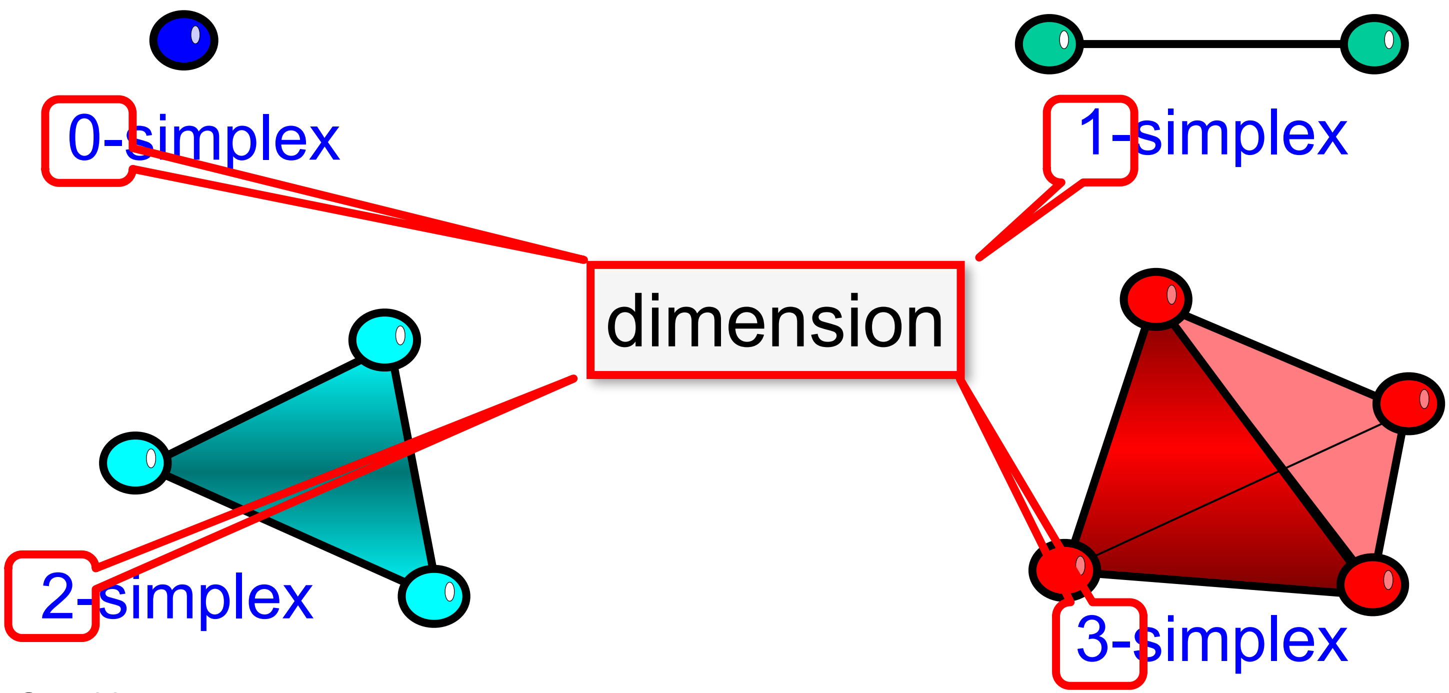
Geometric: a point in high-dimensional Euclidean Space



Simplexes

Combinatorial: a set of vertexes

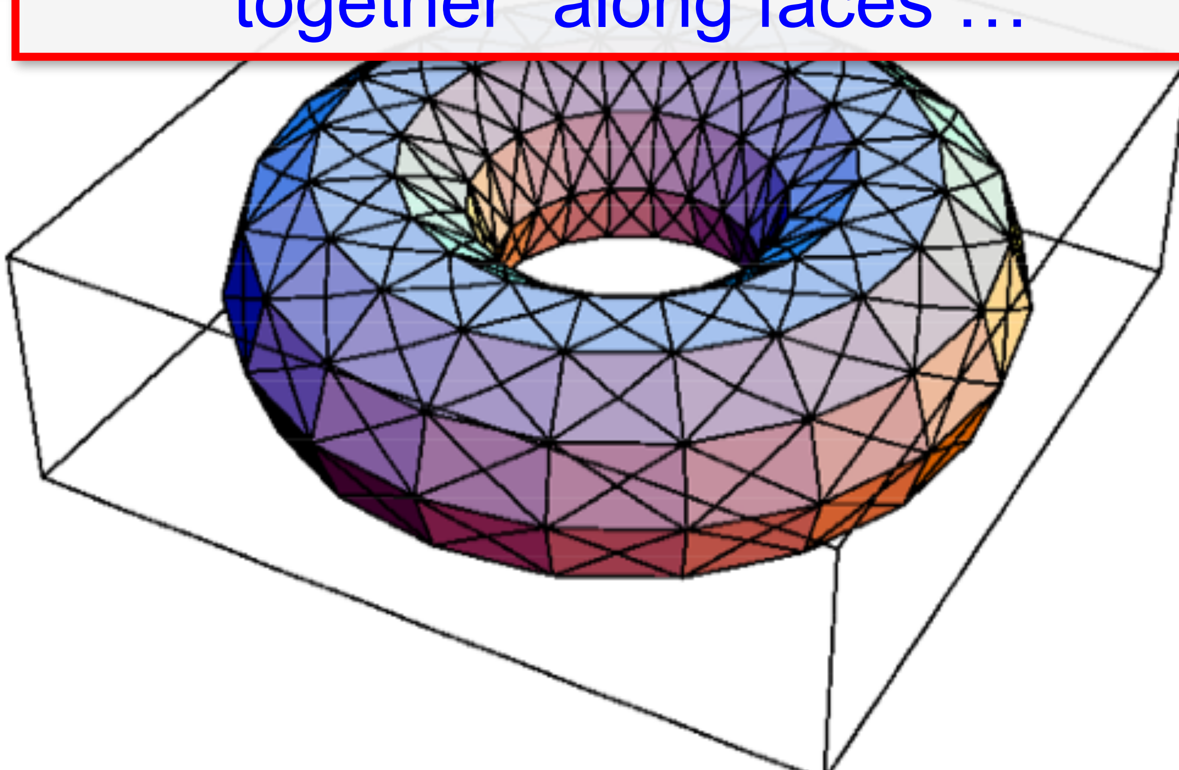
Geometric: convex hull of points in general position



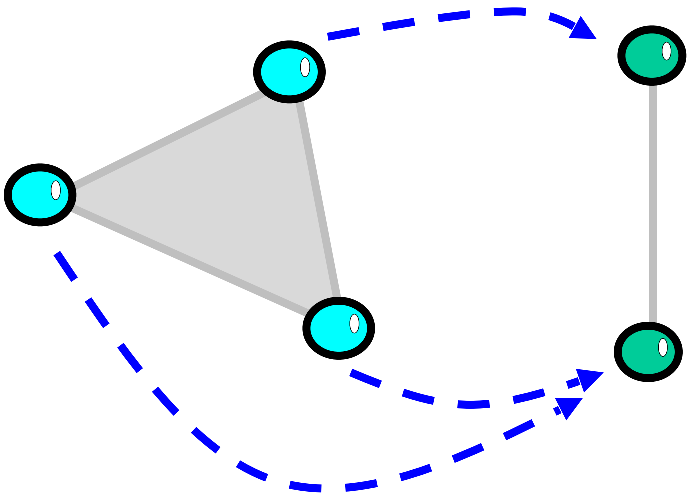
Simplicial Complex

Combinatorial: a set of simplexes close under inclusion.

Geometric: simplexes "glued together" along faces …

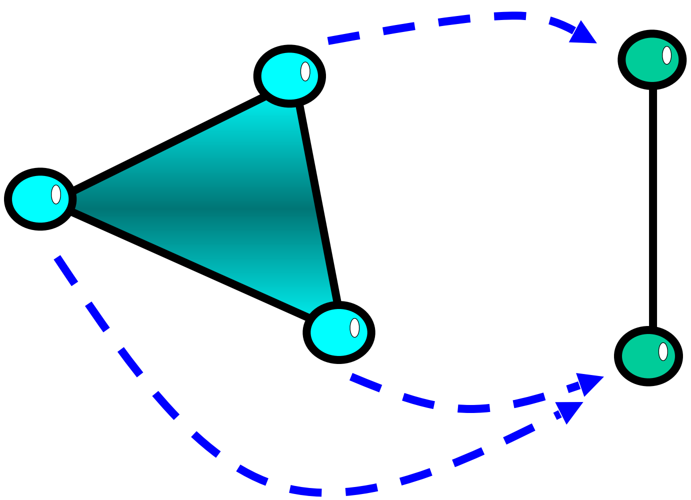


Simplicial Maps



Vertex-to-vertex map …

Simplicial Map

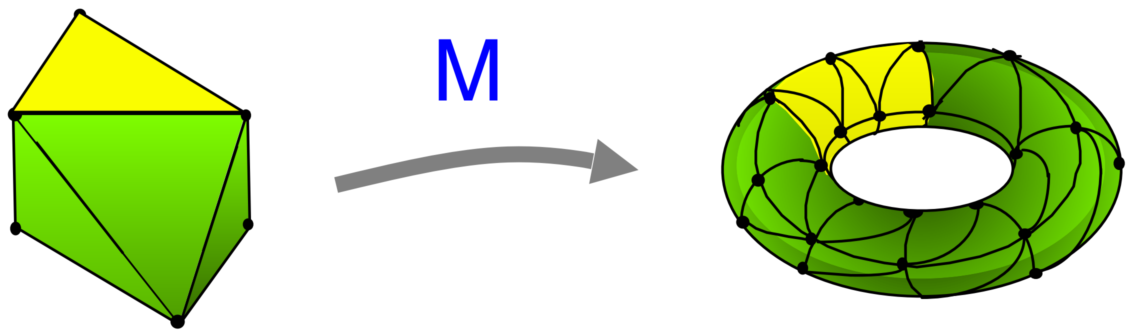


Vertex-to-vertex map …

that sends simplexes to simplexes

piece-wise linear map on geometric simplexes

Carrier Map

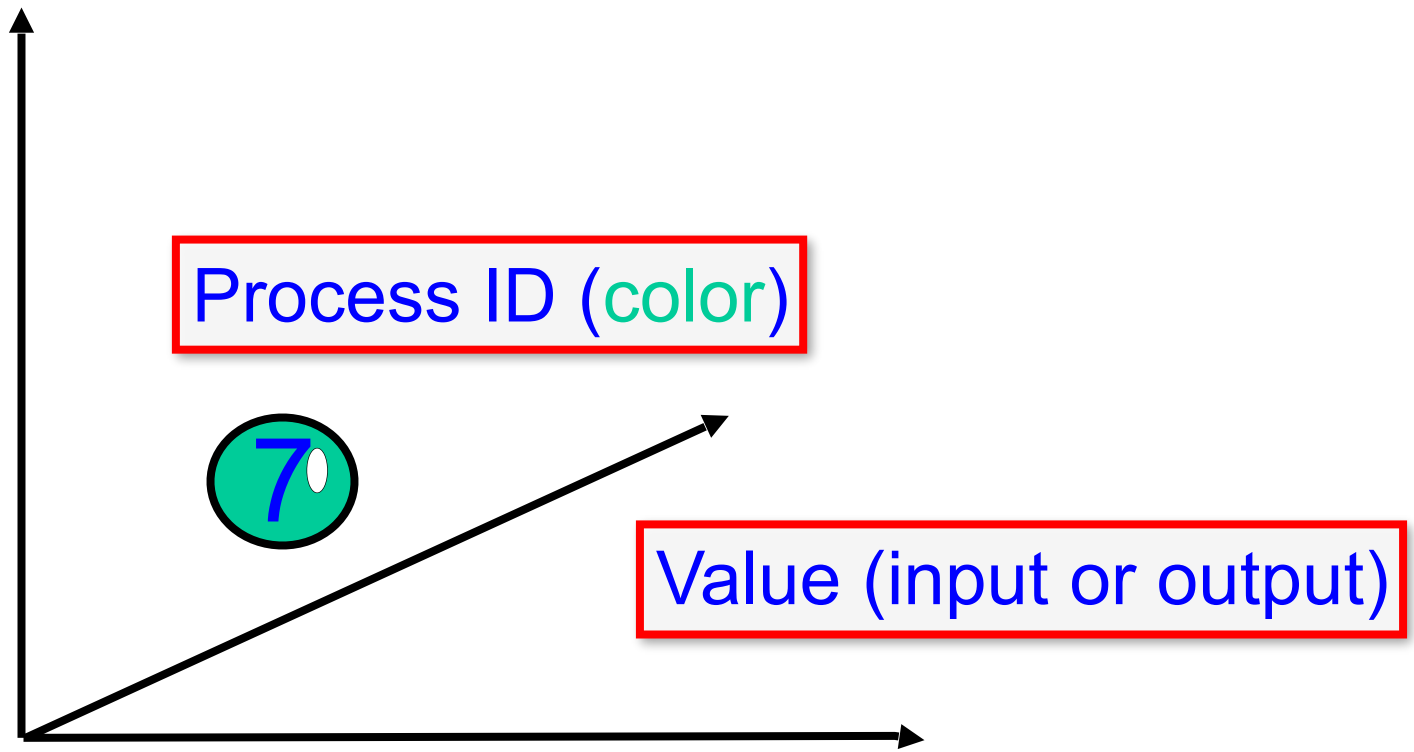


Maps simplex …

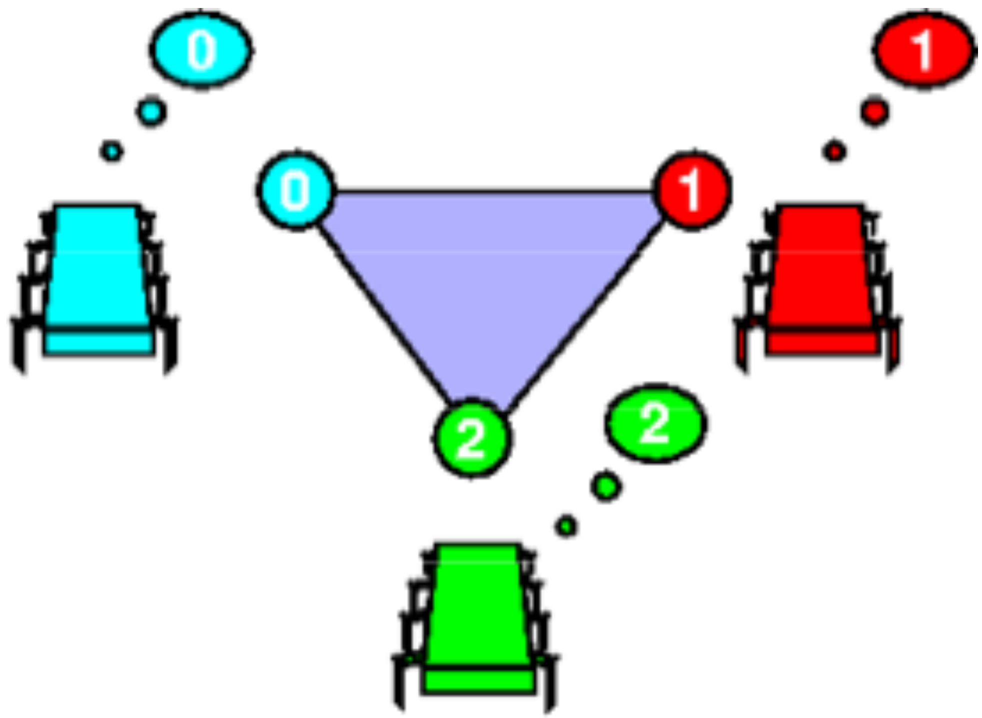
to subcomplex.

Preserves intersections: M (¾ Å ¿) = M (¾) Å M (¿)

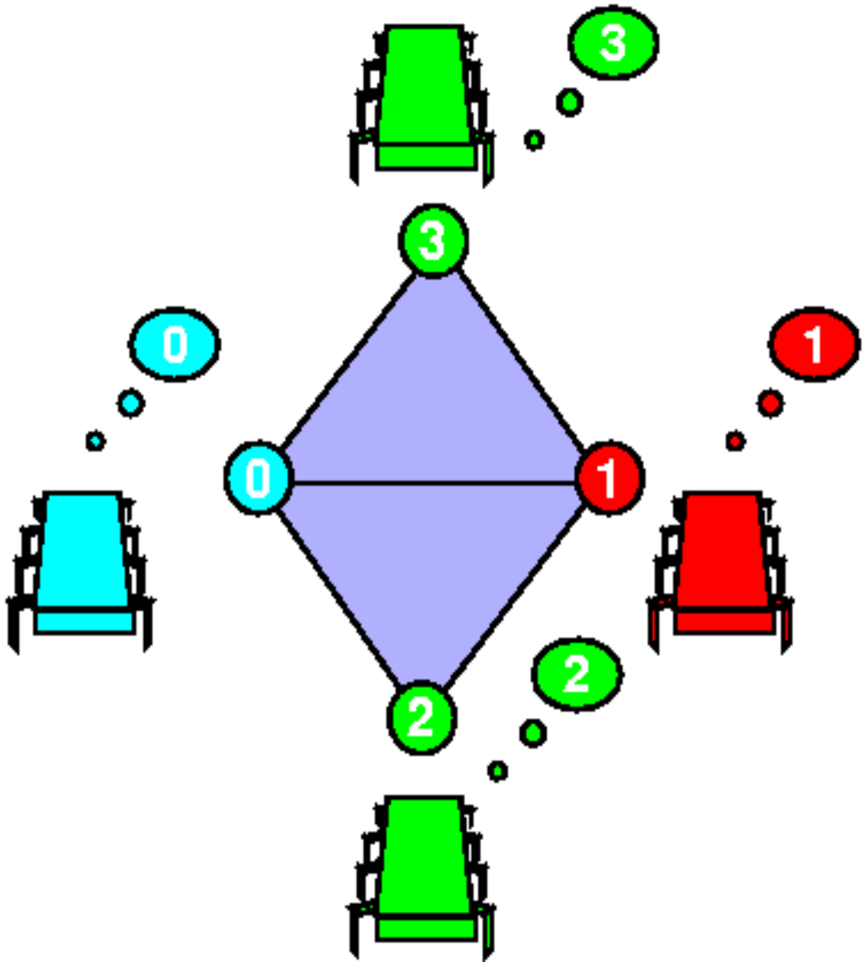
Vertex = Process State



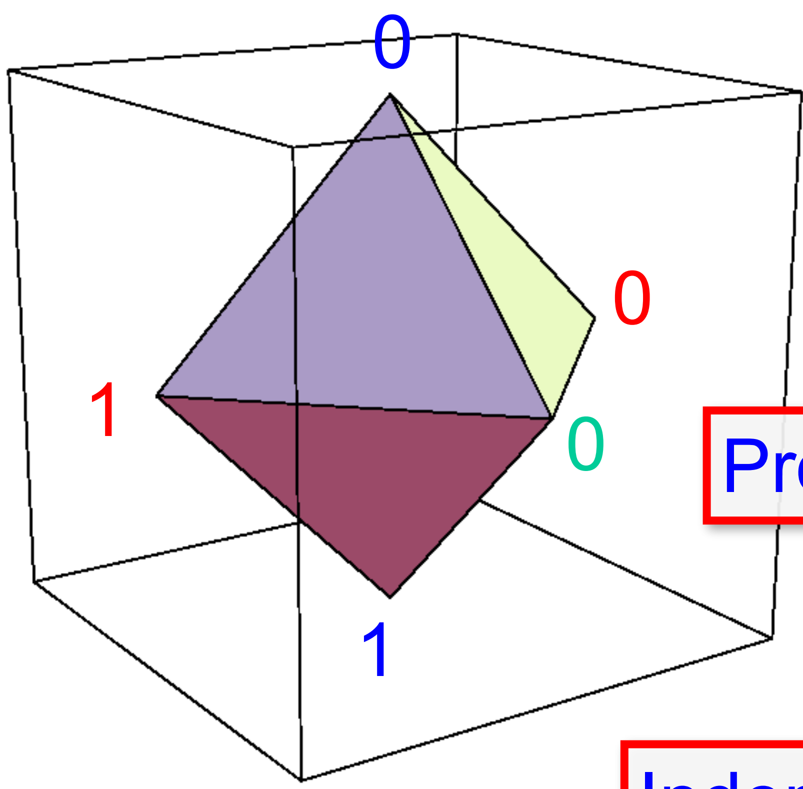
Simplex = Global State



Complex = Global States



Input Complex for Binary Consensus

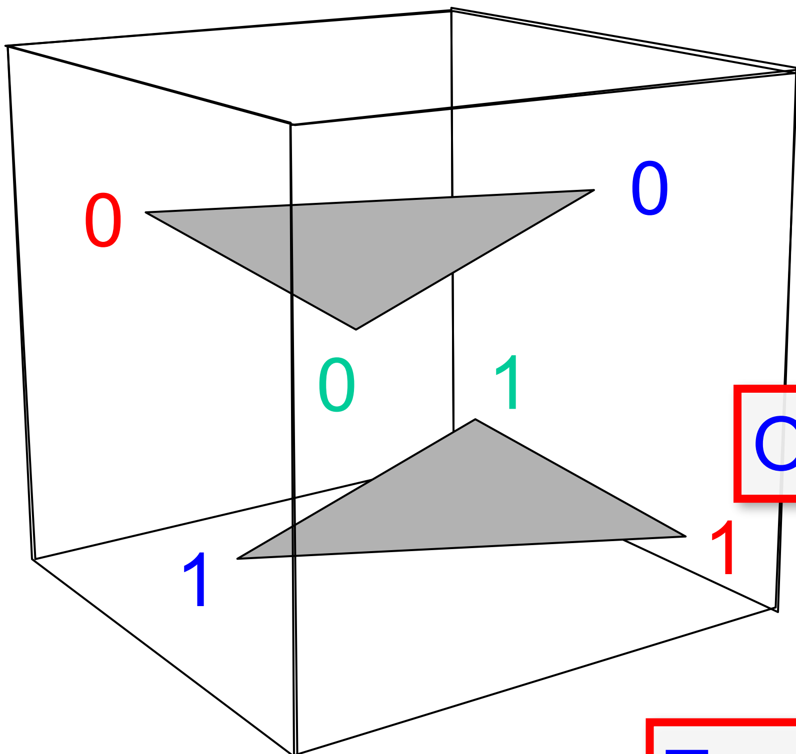


All possible initial states

Processes: red, green, blue

Independently assigned 0 or 1

Output Complex for Binary Consensus

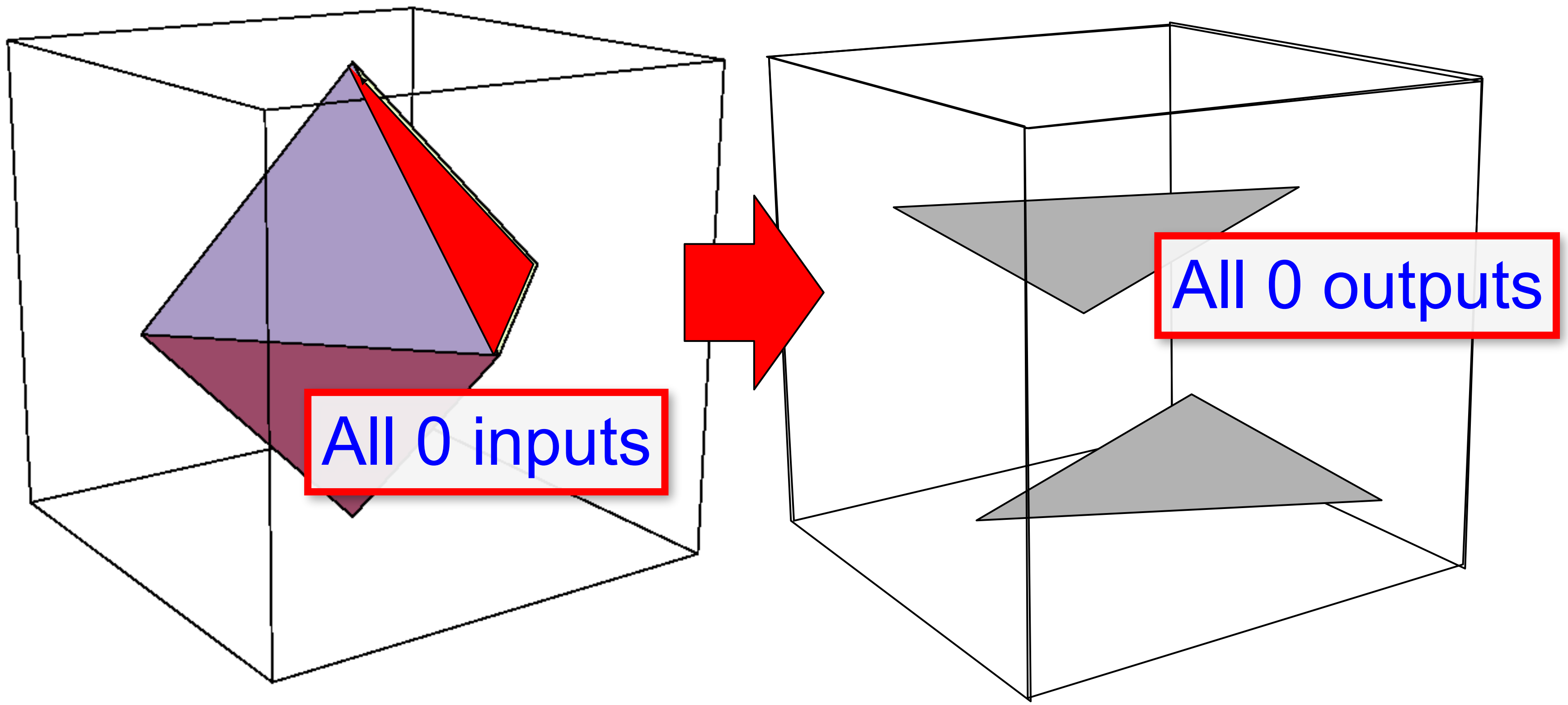


All possible final states

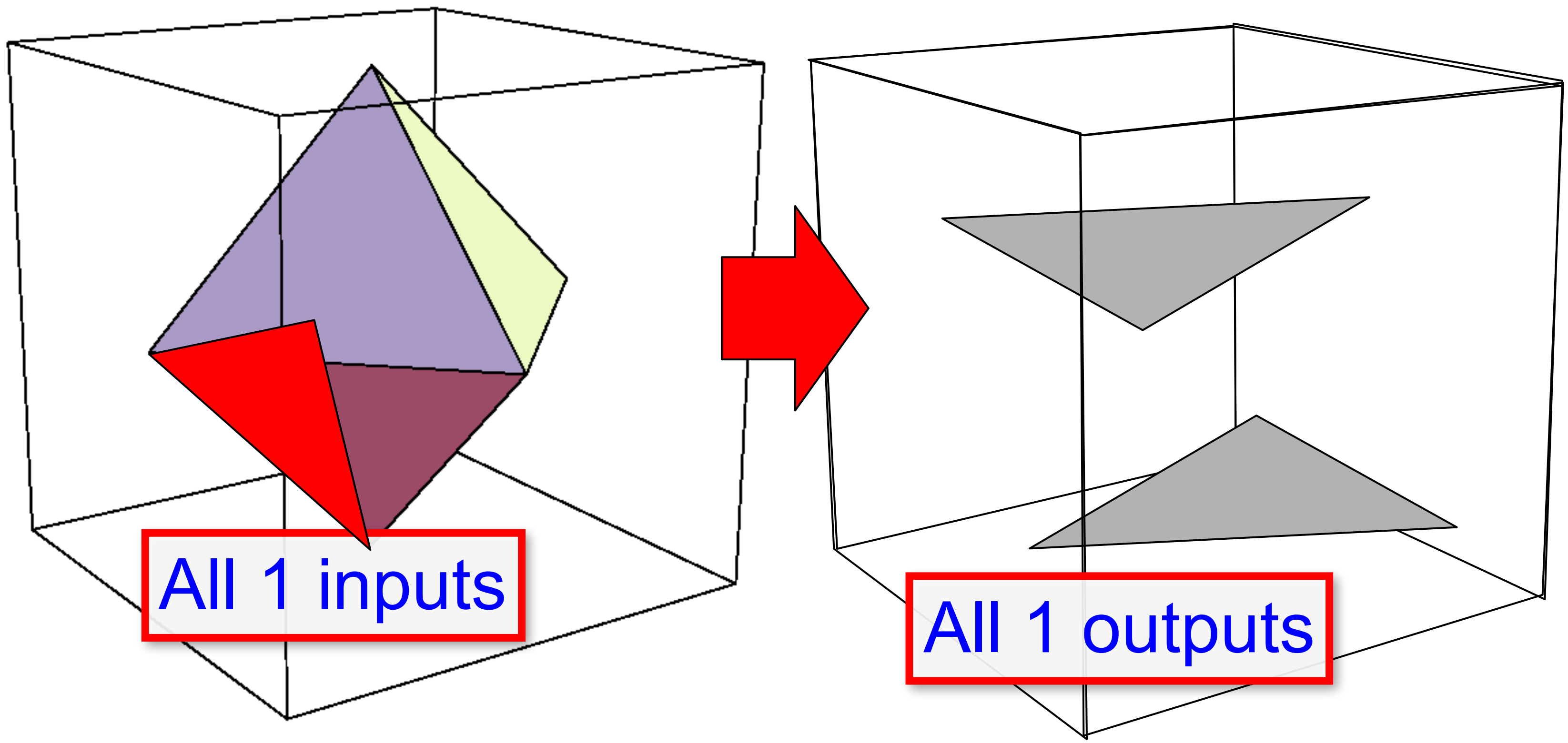
Output values all 0 or all 1

Two disconnected simplexes

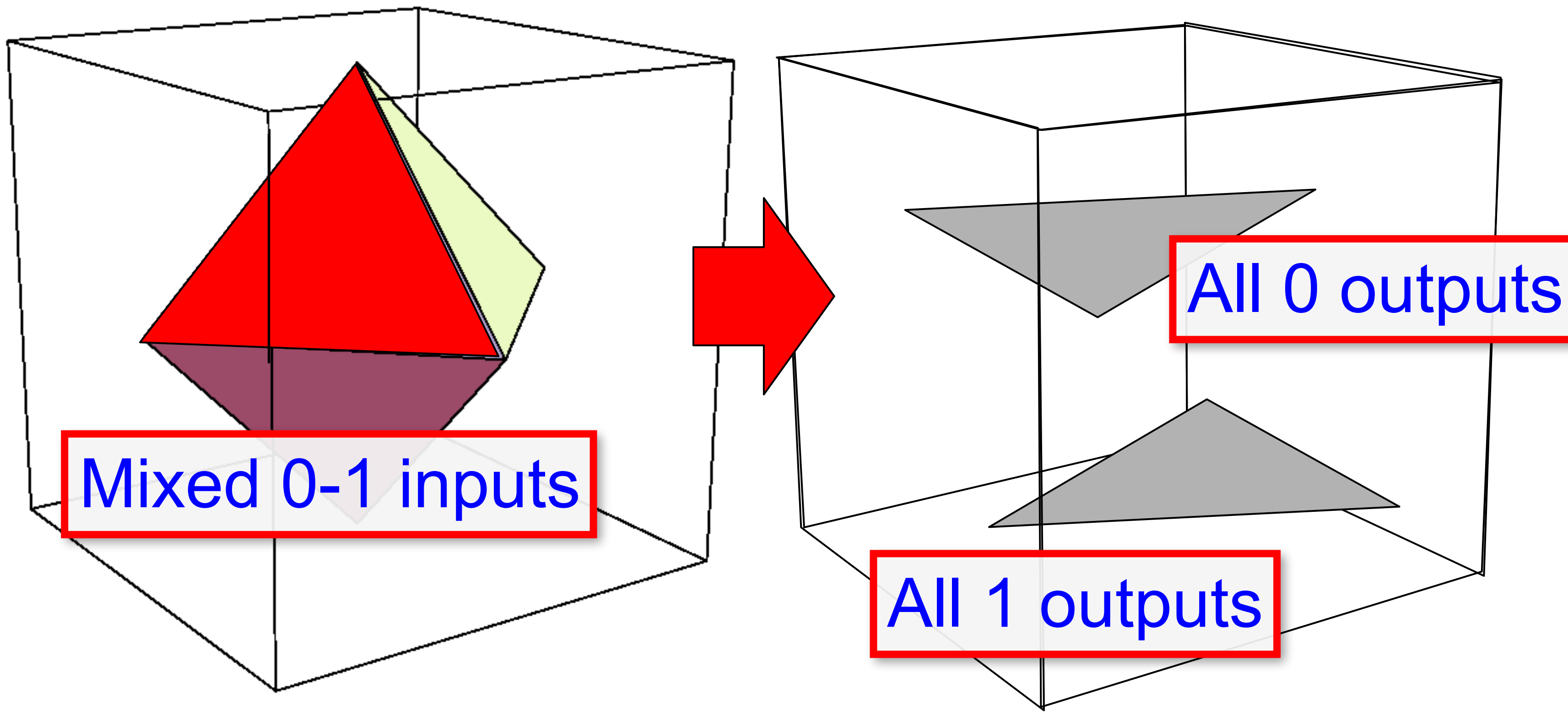
Carrier Map for Consensus



Carrier Map for Consensus

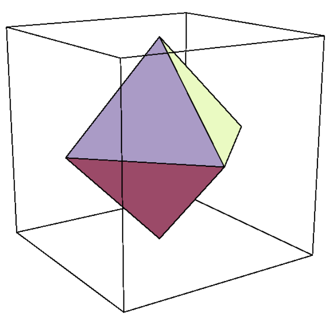


Carrier Map for Consensus

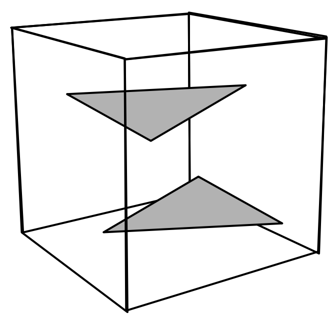


Task Specification

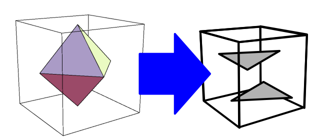
(I, O, ¢)



Input complex



Output complex



Carrier map

Protocol

```
view = my input value;
for (i = 0; i < r; i++) {
    broadcast view;
    view += messages received;
    }
return δ(view)
```

Finite program

Protocol

```
view = my input value;
for (i = 0; i < r; i++) {
    broadcast view;
    view += messages received;
    }
return δ(view)
```

Start with input value

Protocol

```
view = my input value;
for (i = 0; i < r; i++) {
    broadcast view;
    view
    }
return δ(view)
```

Run for fixed number of rounds

Protocol

```
view = my input value;
for (i = 0; i < r; i++) {
    broadcast view;
    view += messages received;
    }
return δ(view)
```

Send current view to others

Protocol

```
view = my input value;
for (i = 0; i < r; i++) {
    broadcast view;
    view += messages received;
    }
return δ(view)
```

Concatenate messages received to view

(full-information protocol)

Protocol

```
view = my input value;
for (i = 0; i < r; i++) {
    broadcast view;
    view += messages received;
    }
return δ(view)
```

finally, apply task-specific decision map to view

Protocol Complex

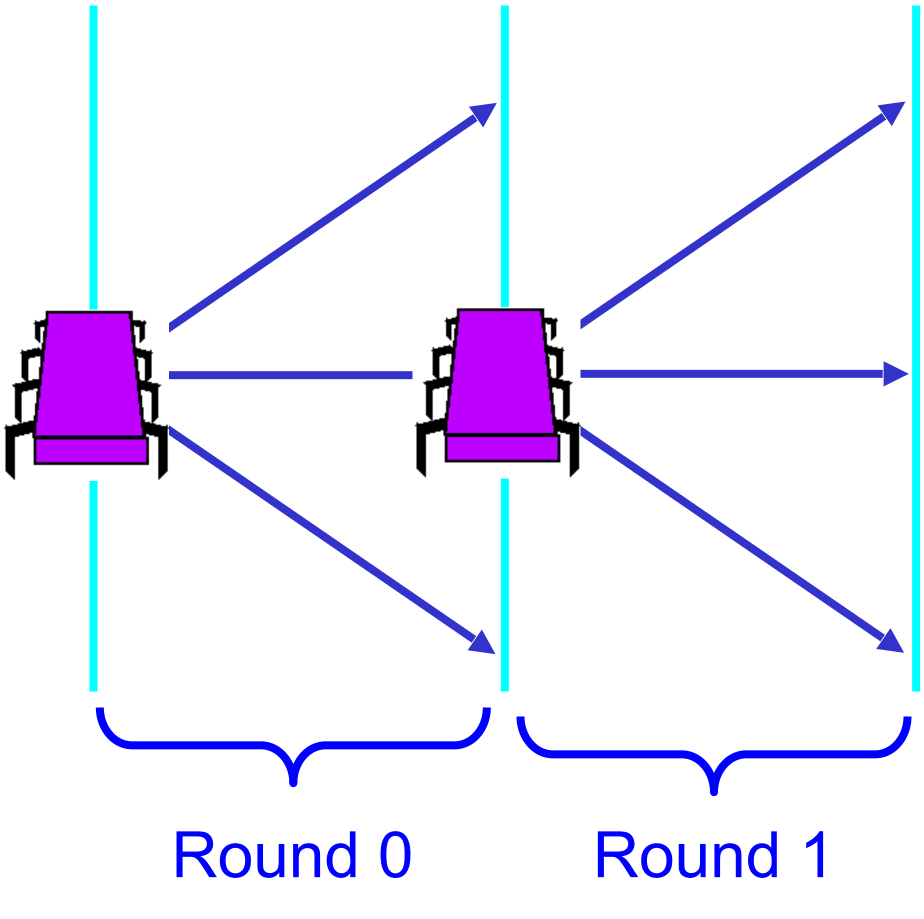
Vertex: process ID, view

Complete log of messages sent & received

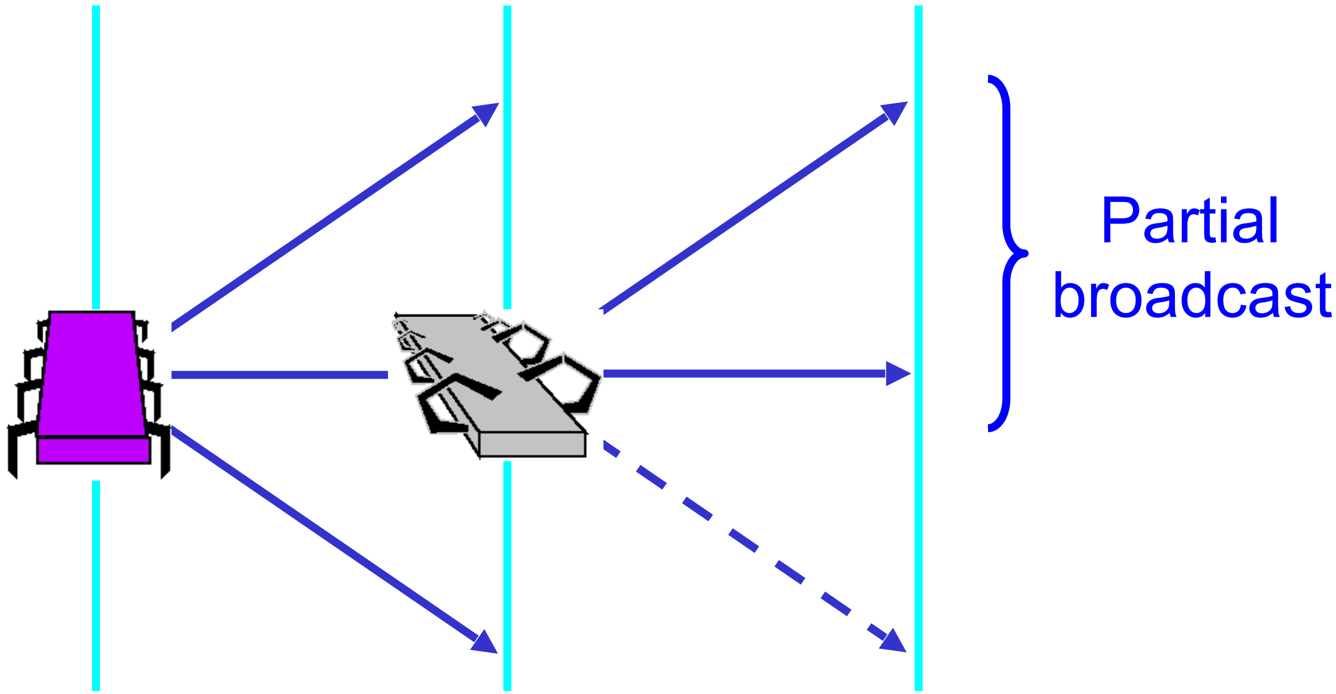
Simplex: compatible set of views

Each execution defines a simplex

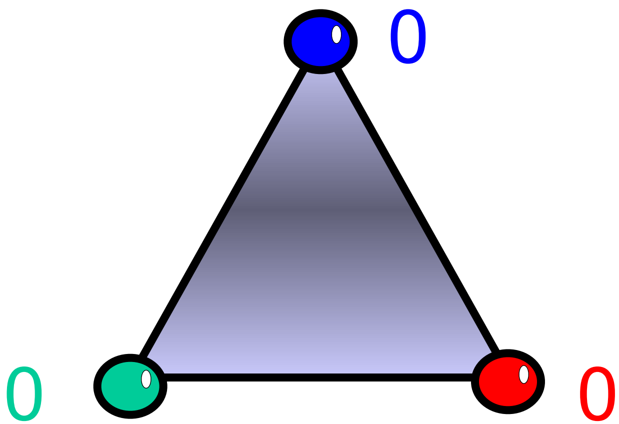
Example: Synchronous Message-Passing



Failures: Fail-Stop



Single Input: Round Zero

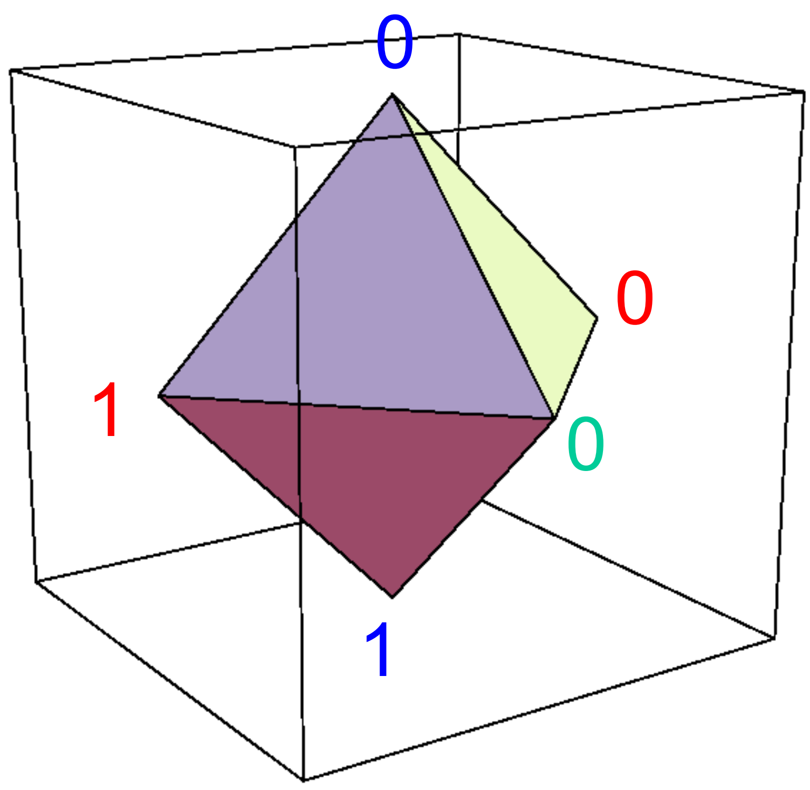


No messages sent

View is input value

Same as input simplex

Round Zero Protocol Complex

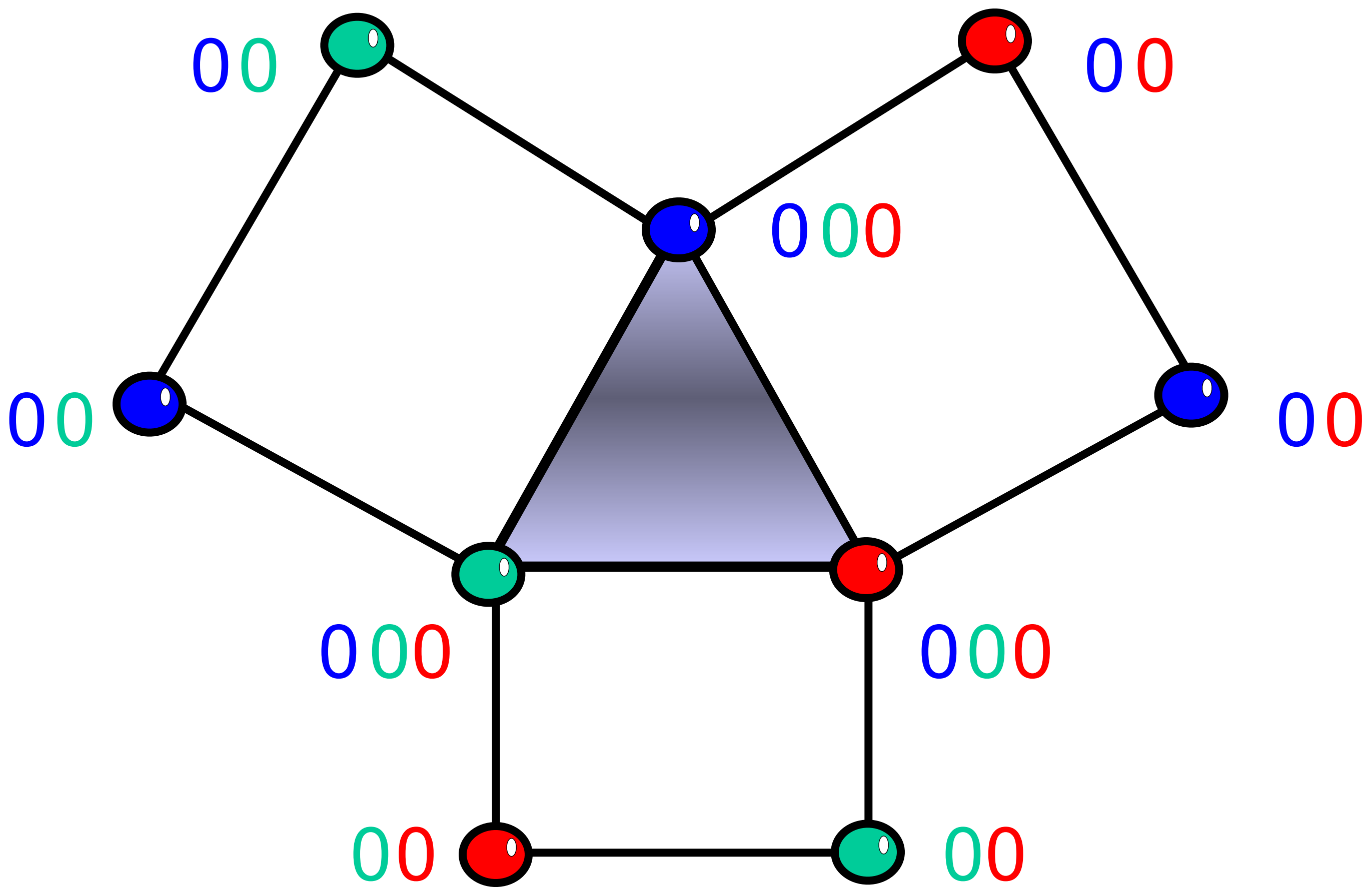


No messages sent

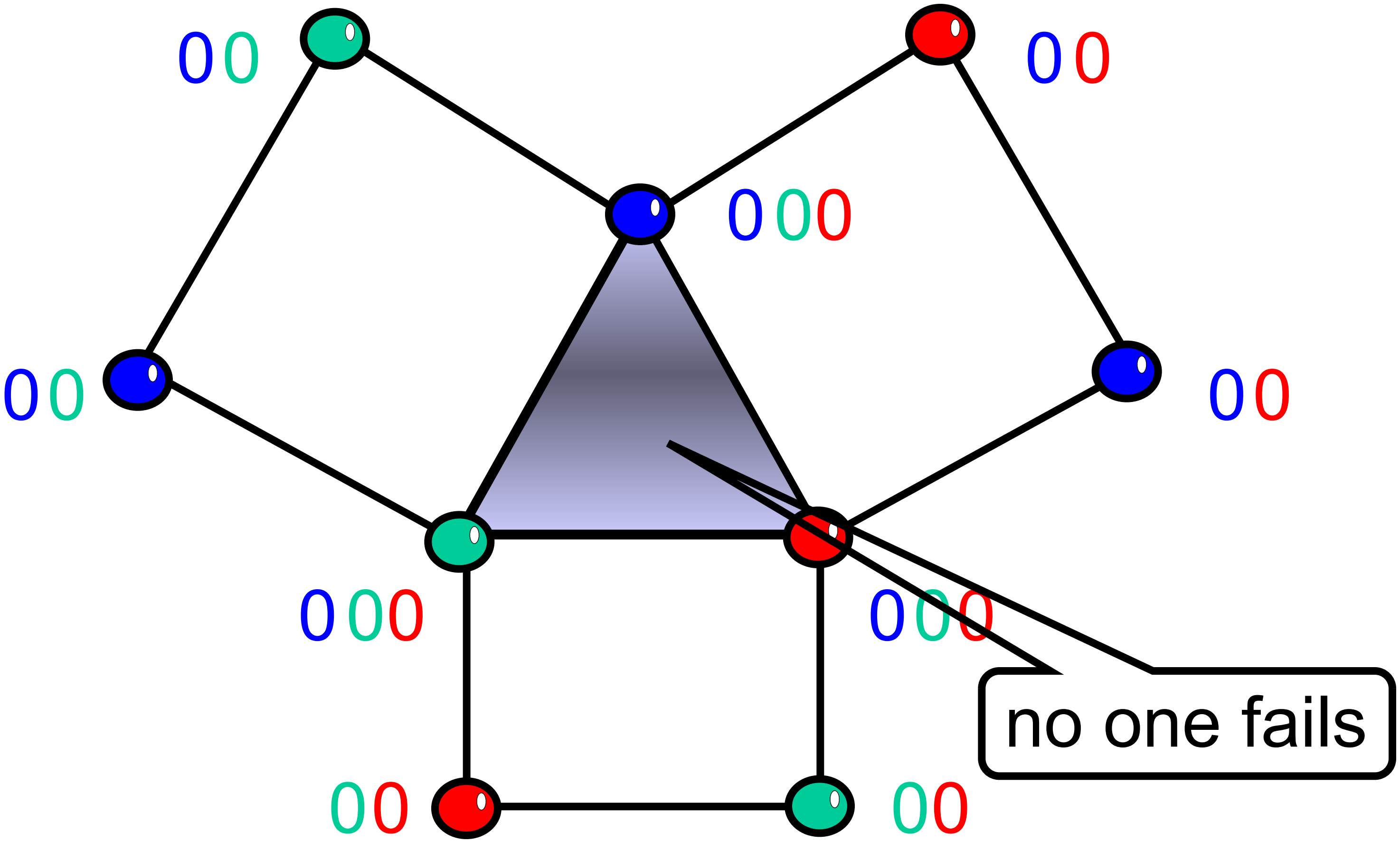
View is input value

Same as input complex

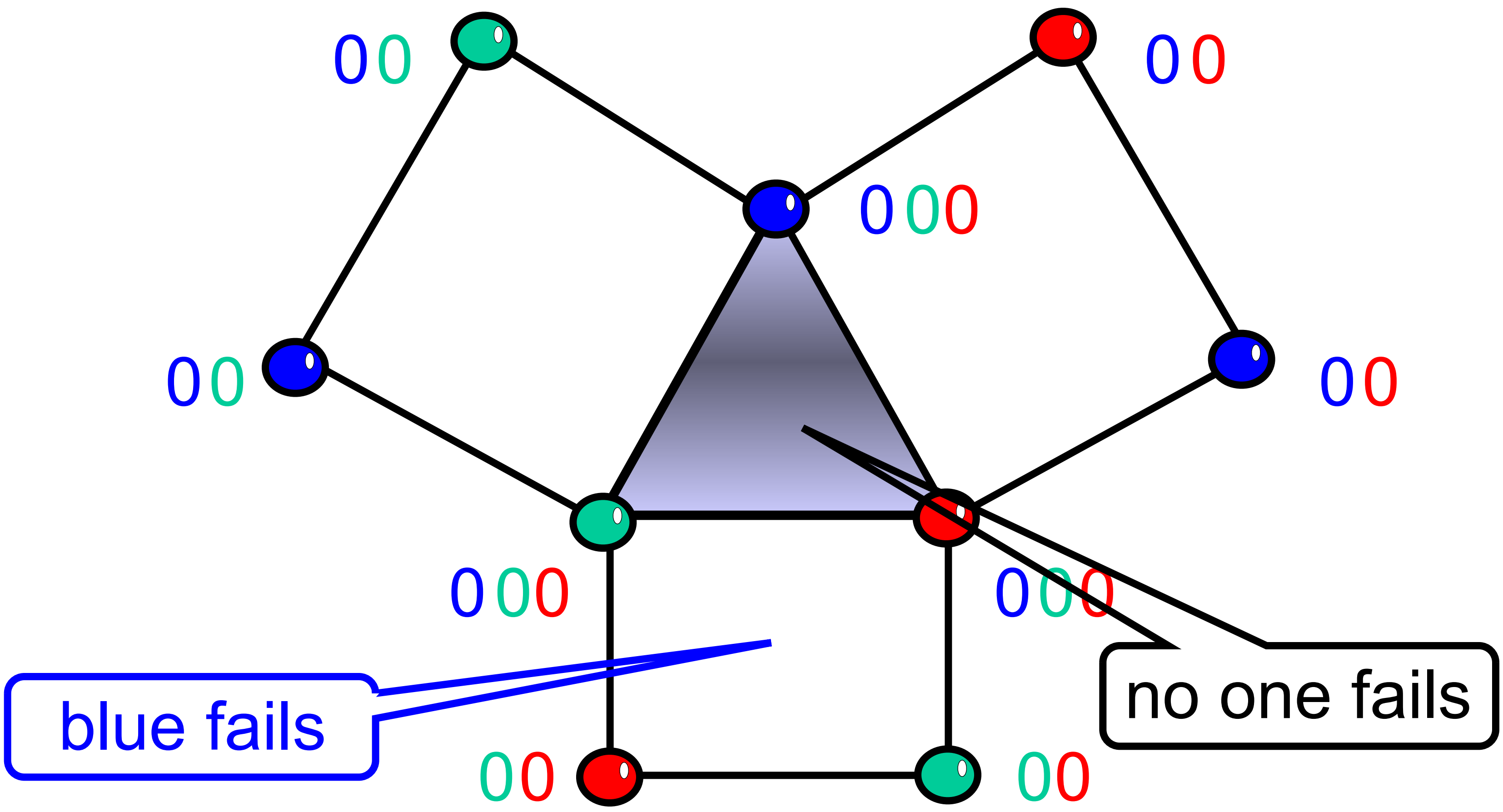
Single Input: Round One



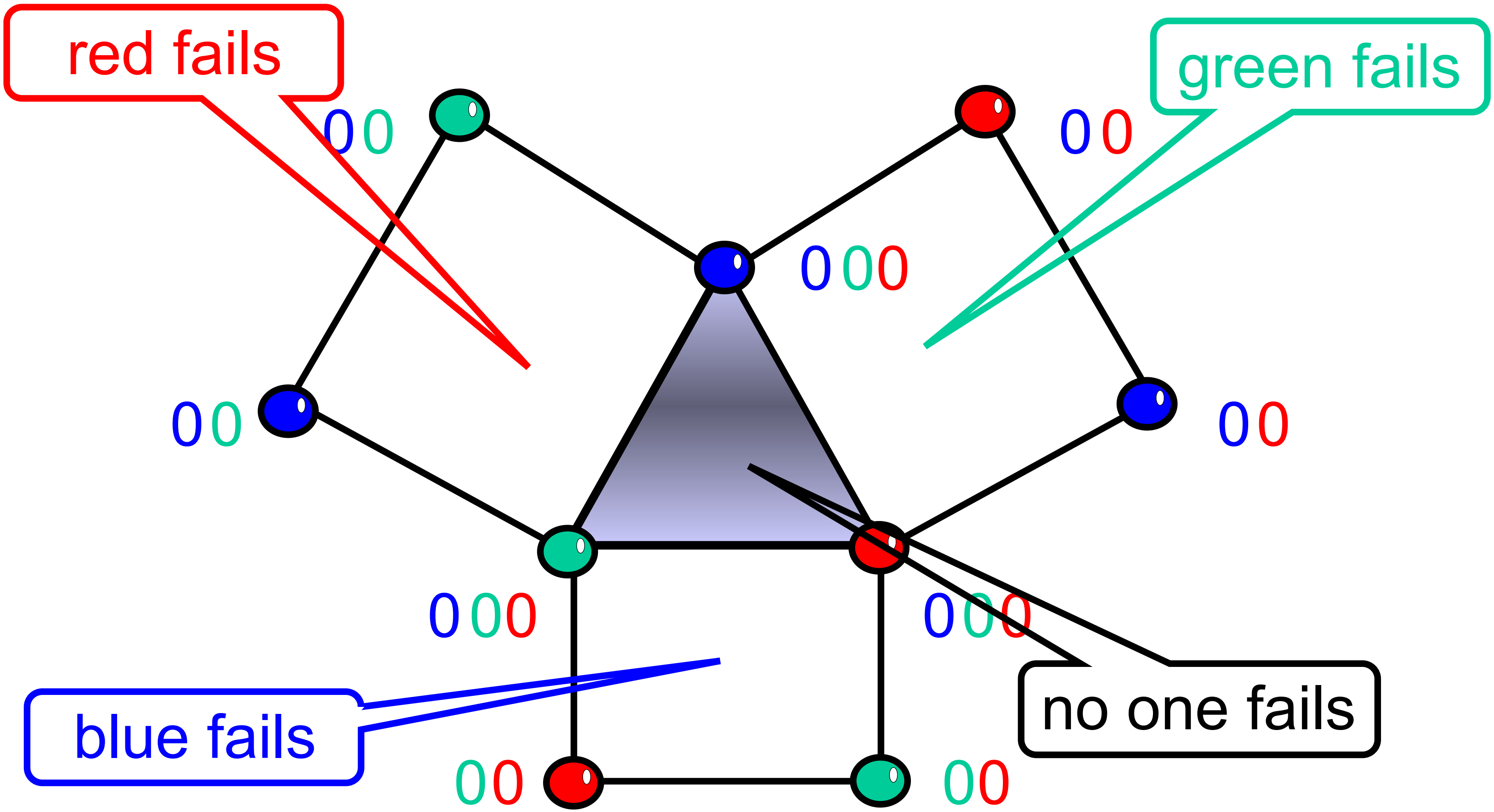
Single Input: Round One



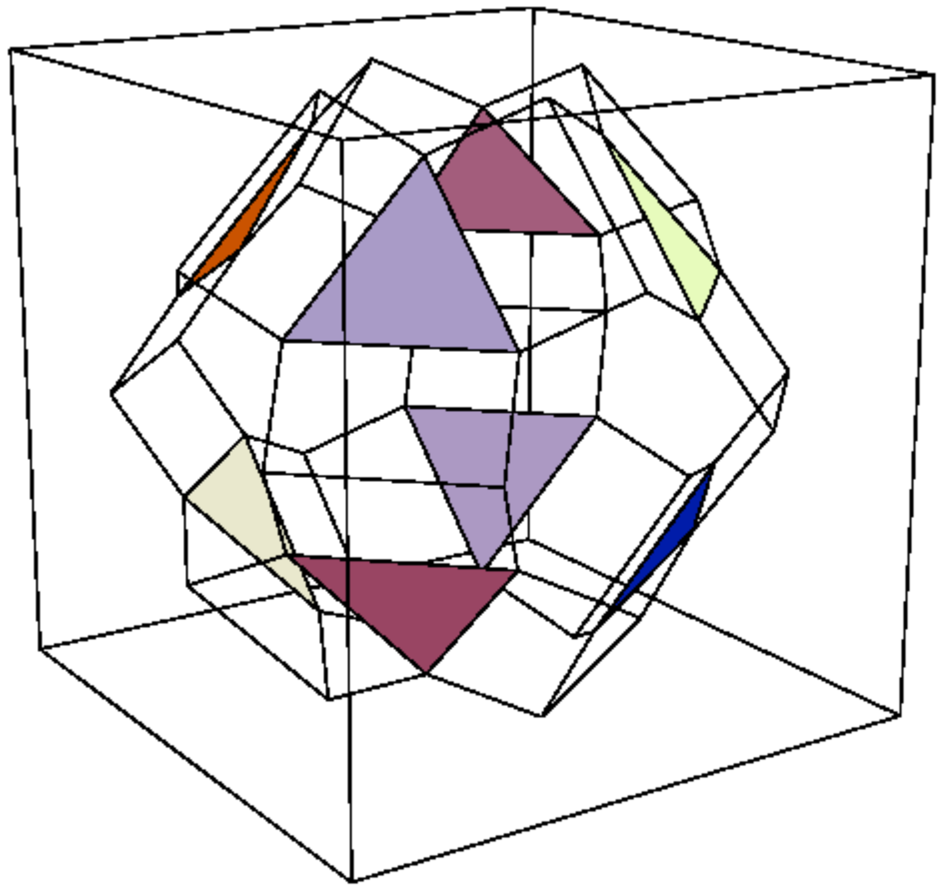
Single Input: Round One



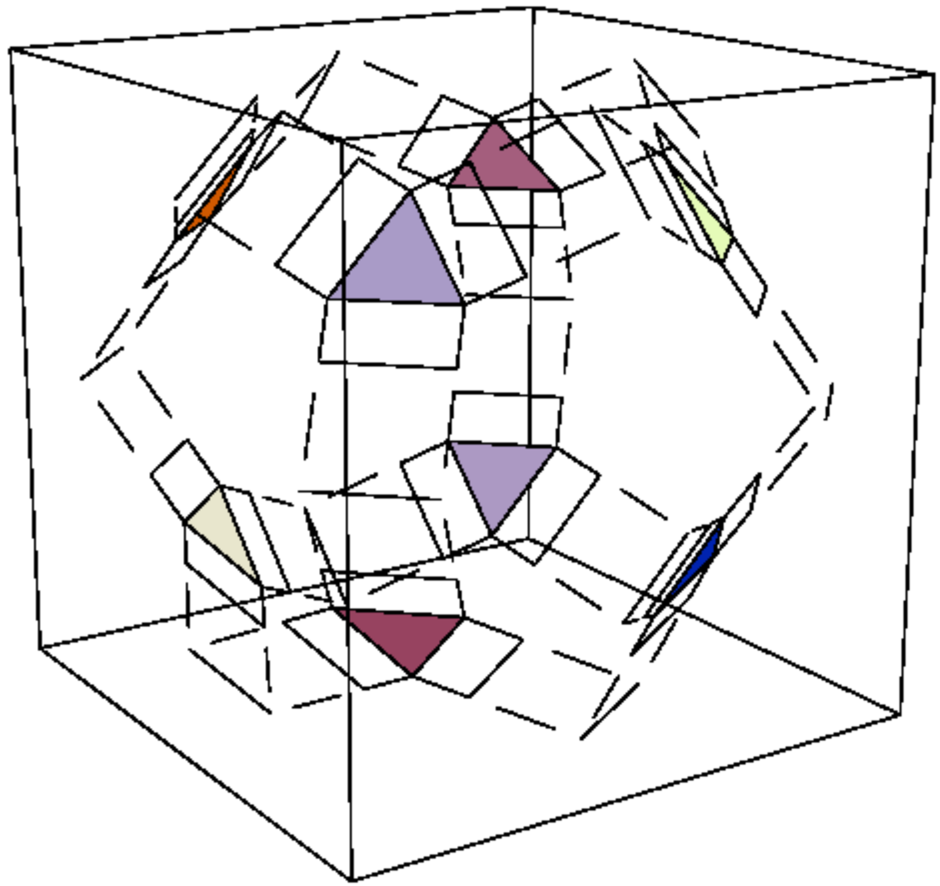
Single Input: Round One



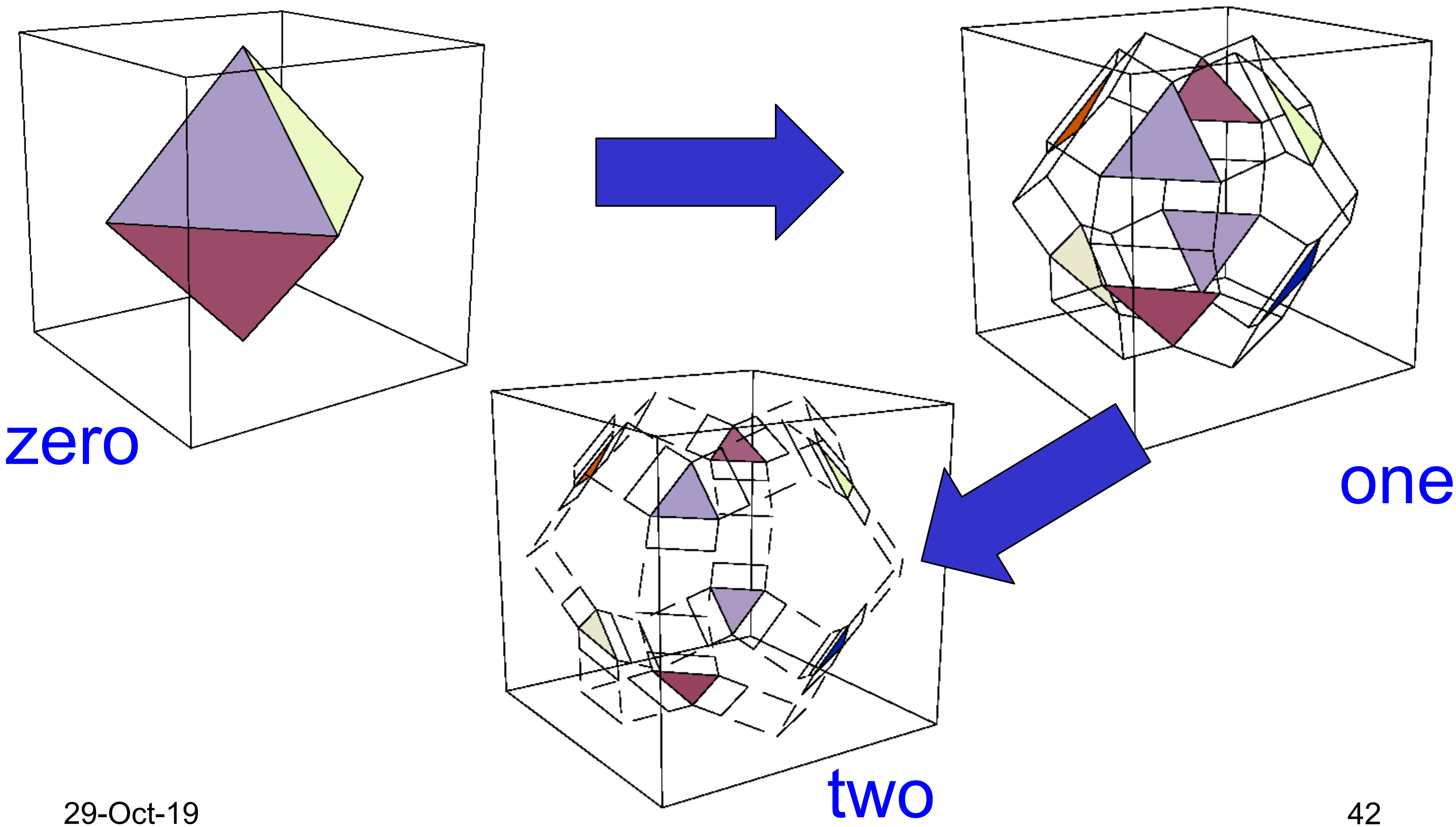
Protocol Complex: Round One



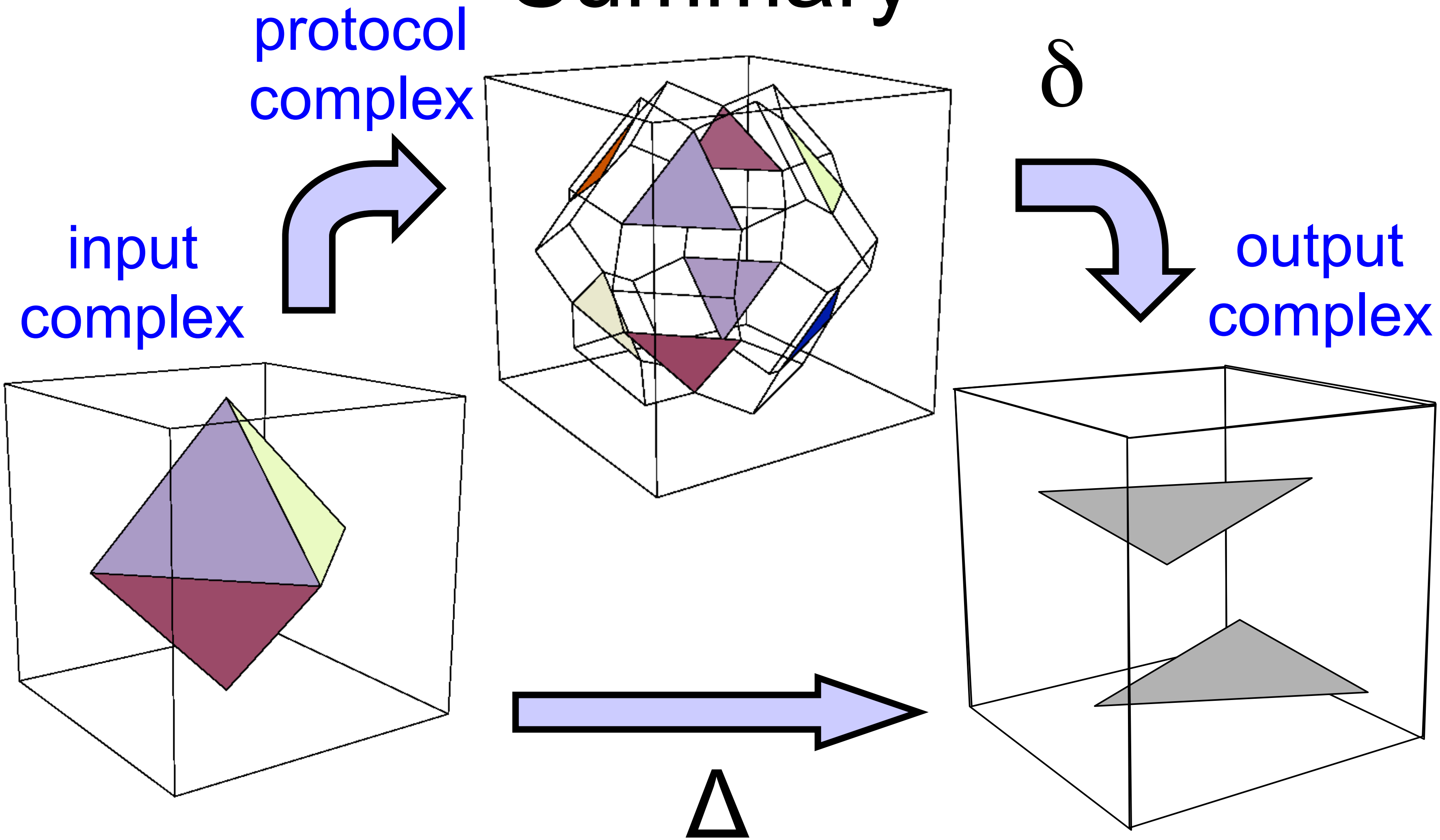
Protocol Complex: Round Two



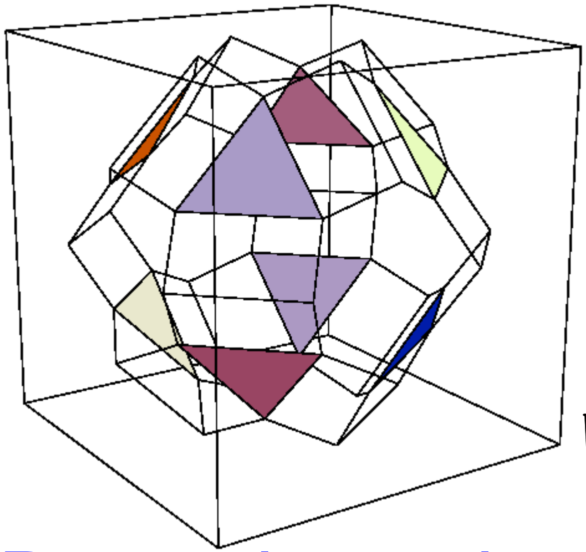
Protocol Complex Evolution



Summary



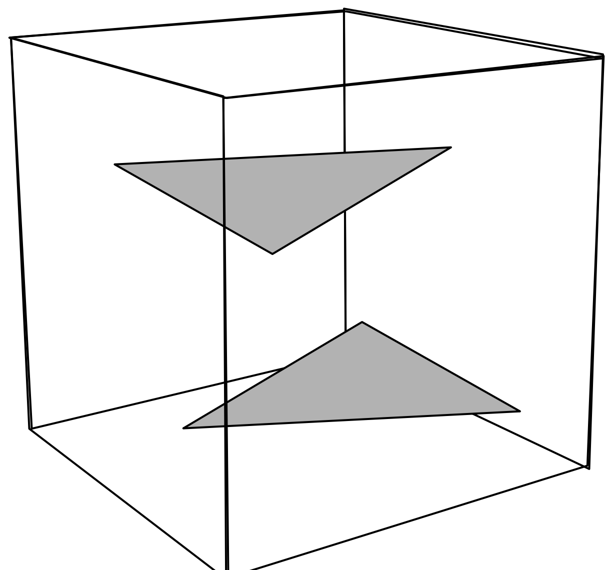
Decision Map



Protocol complex

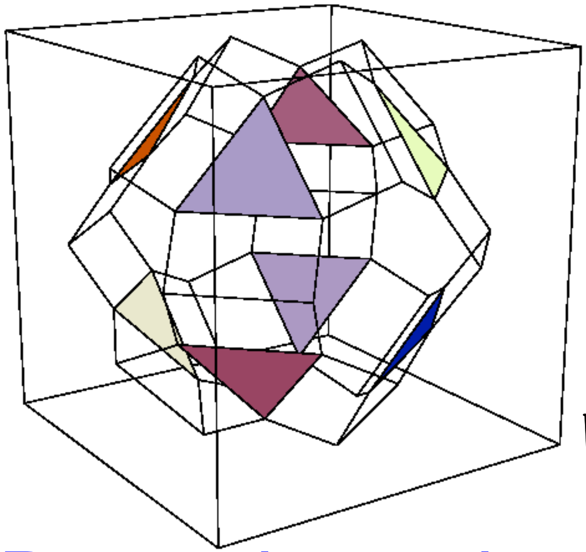
δ

Simplicial map, sending simplexes to simplexes

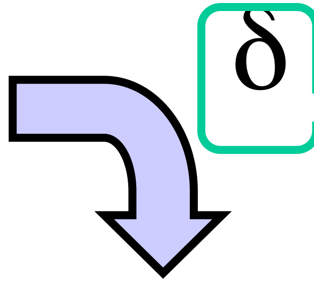


Output complex

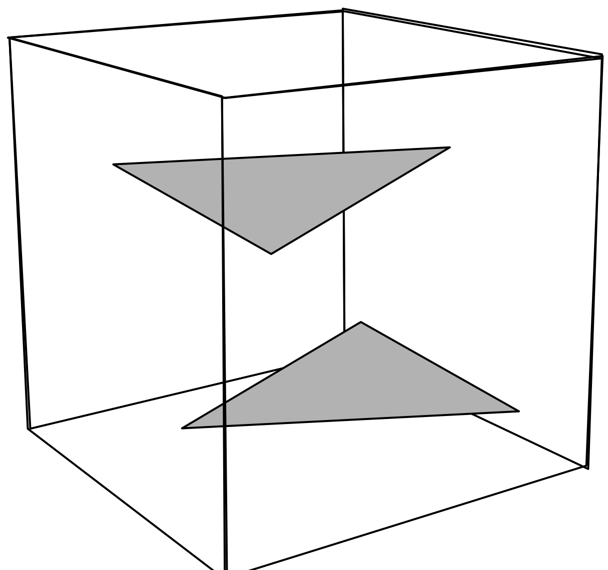
Lower Bound Strategy



Protocol complex

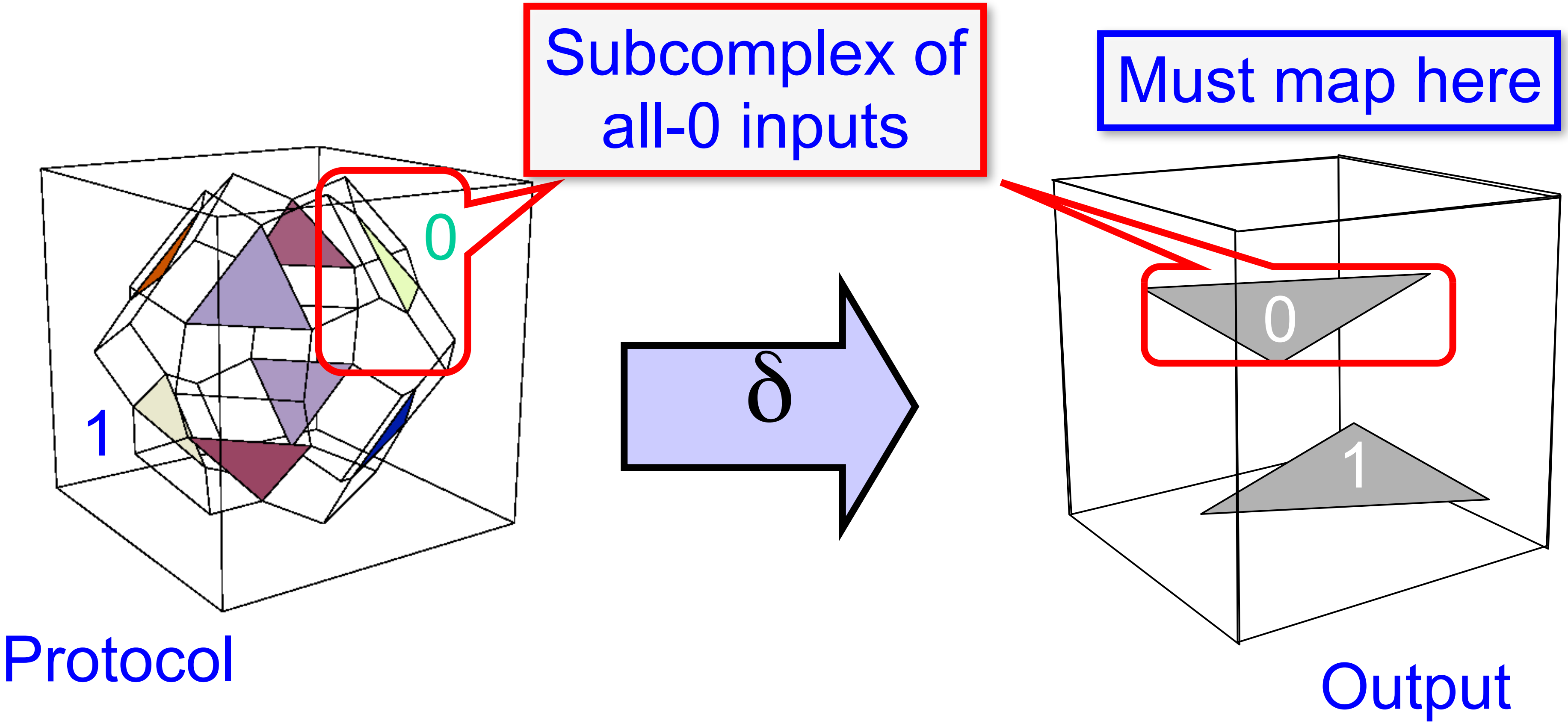


Find topological "obstruction" to this simplicial map

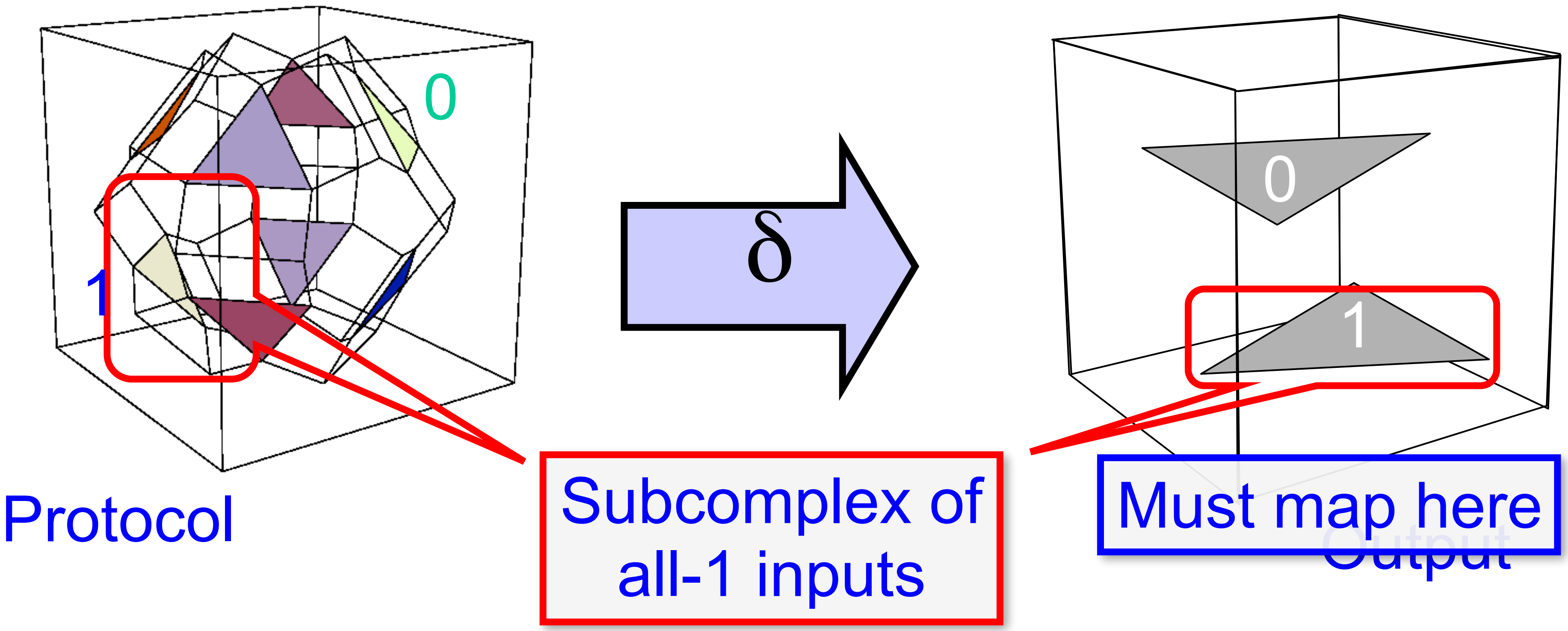


Output complex

Consensus Example

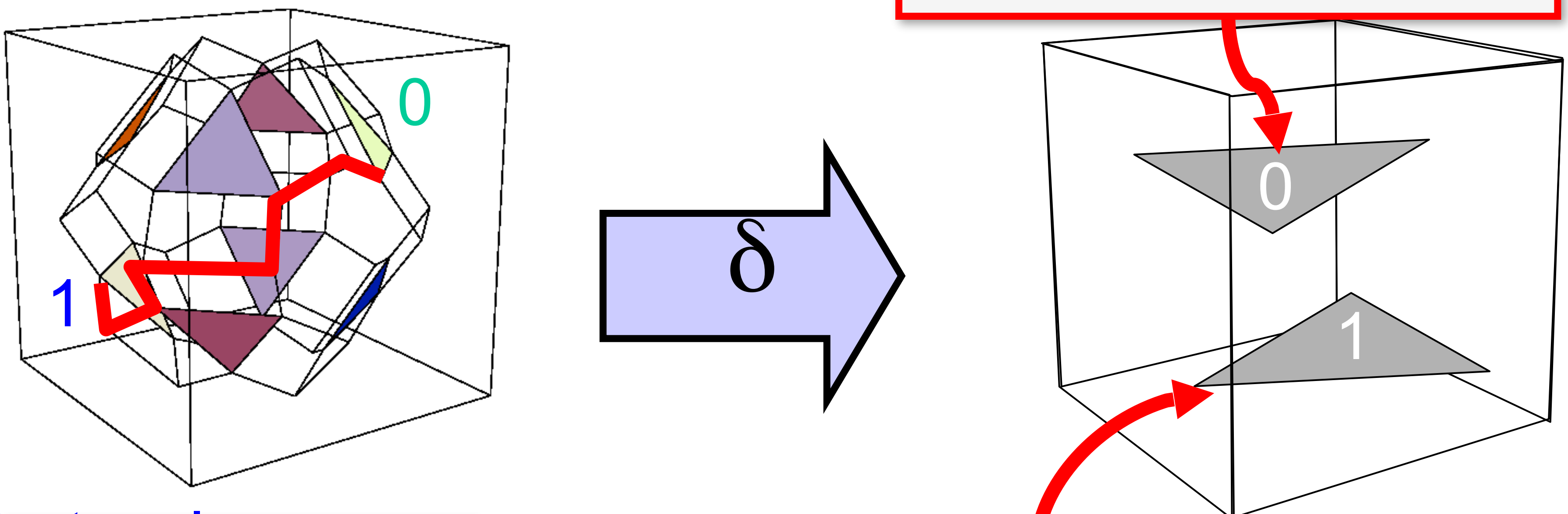


Consensus Example



Consensus Example

Image under δ must start here ..



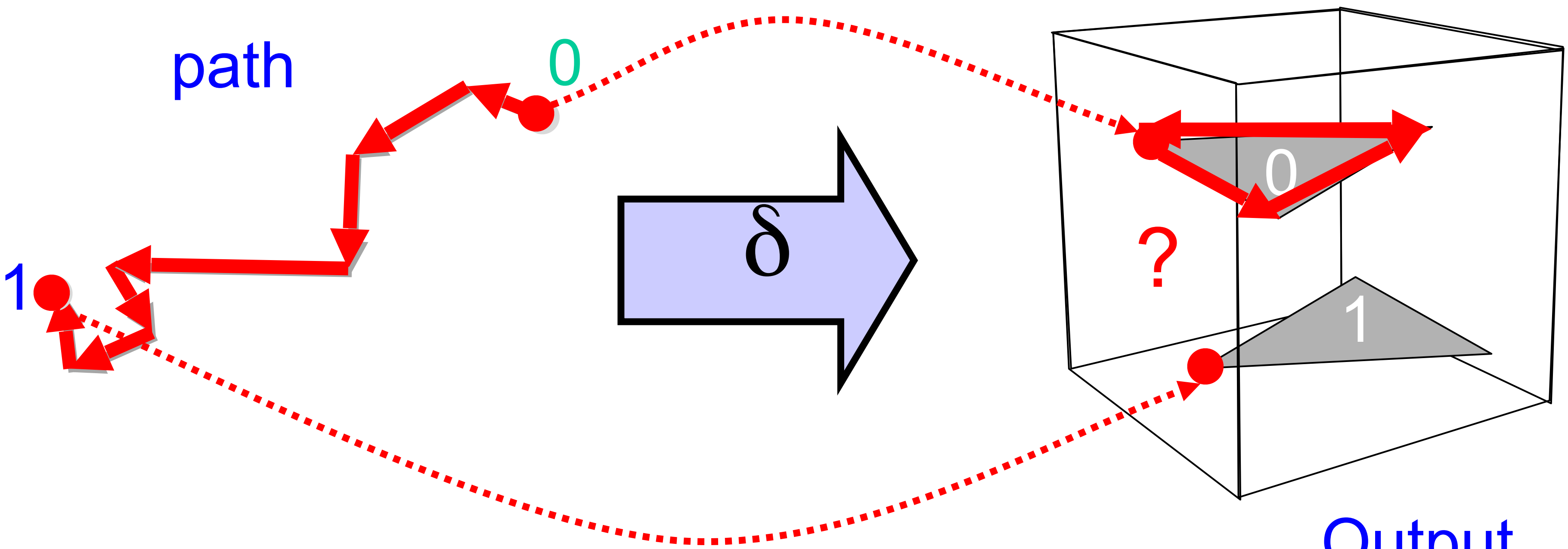
Protocol

Path from “all-0” to “all-1”

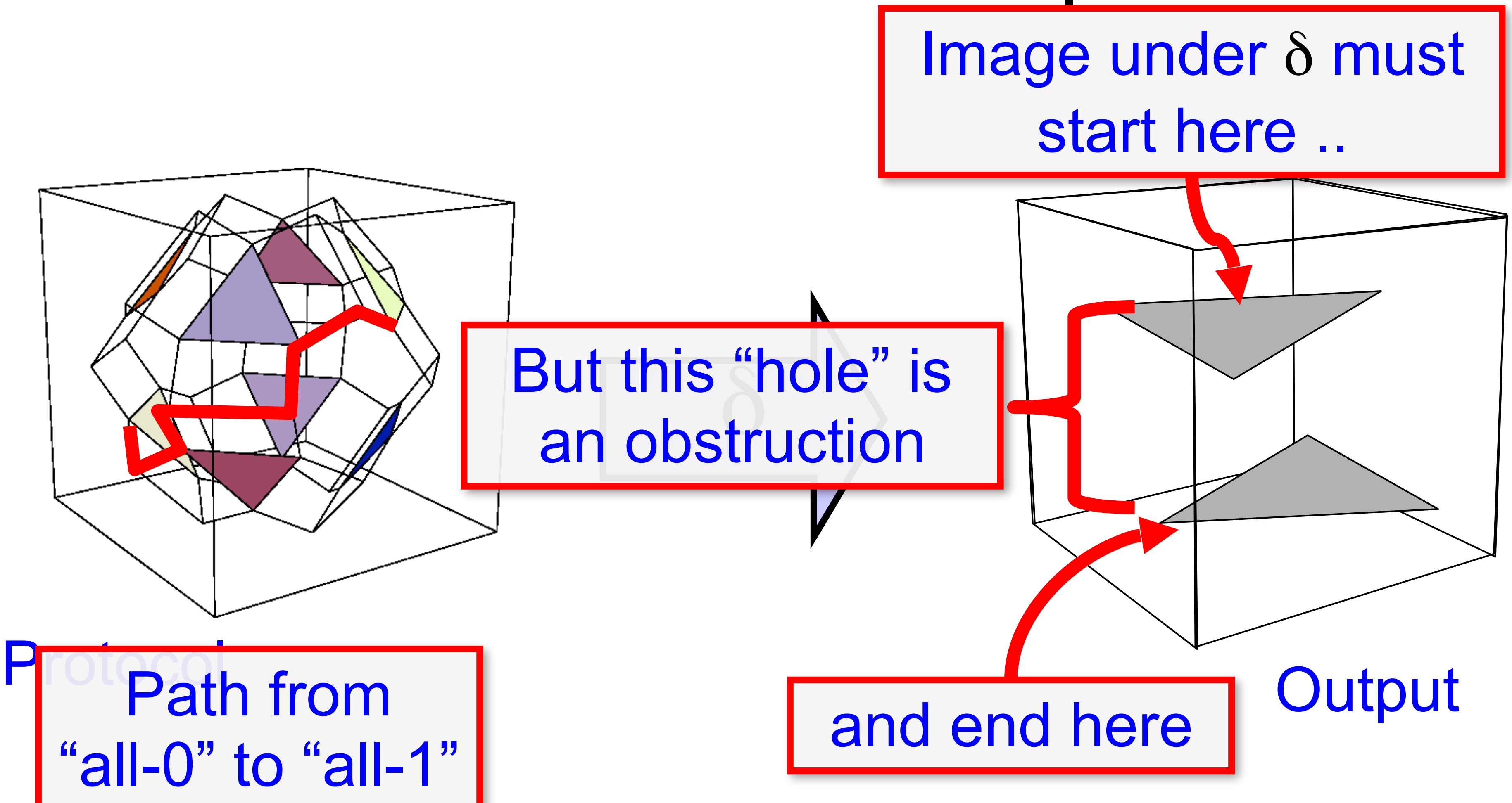
and end here

Output

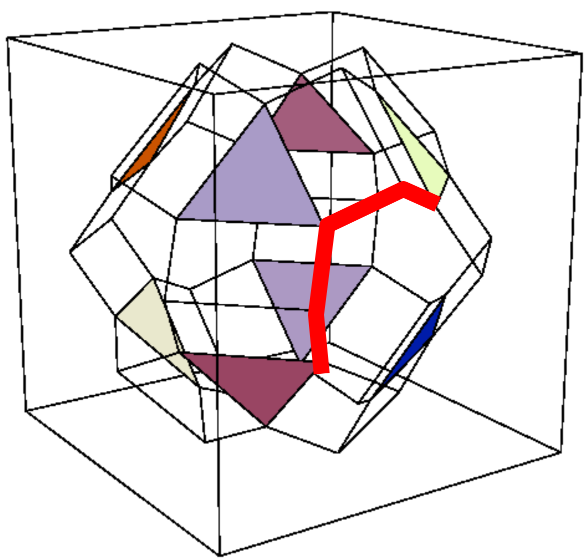
Consensus Example



Consensus Example



Theorem



A protocol cannot solve consensus if its complex is ***path-connected***

Model-independent!

If Adversary keeps Protocol Complex path-connected …

Forever …

Consensus is ***impossible***

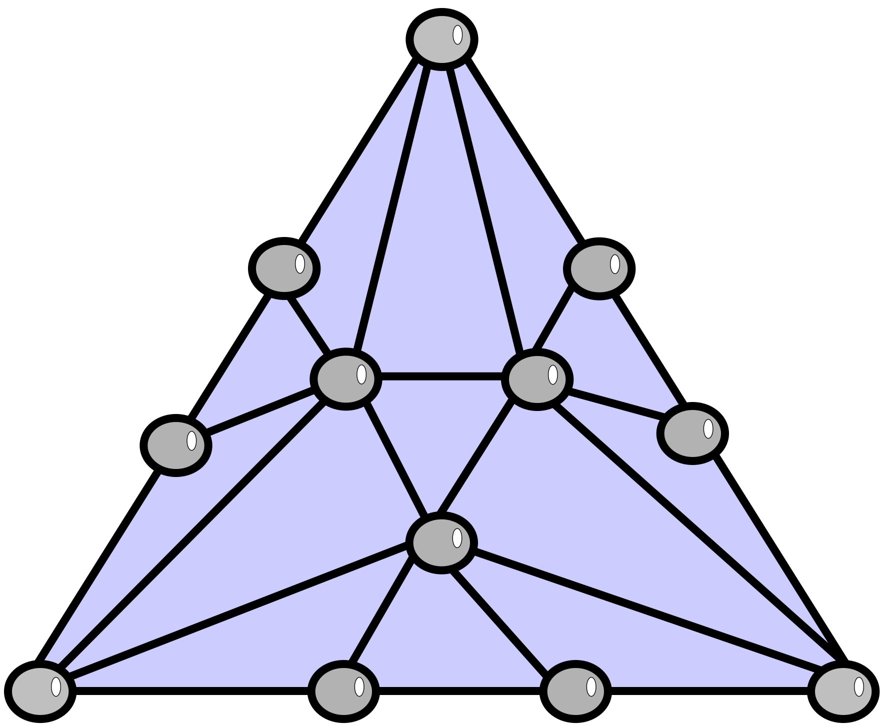
For *r* rounds …

A ***round-complexity*** lower bound

For time *t* …

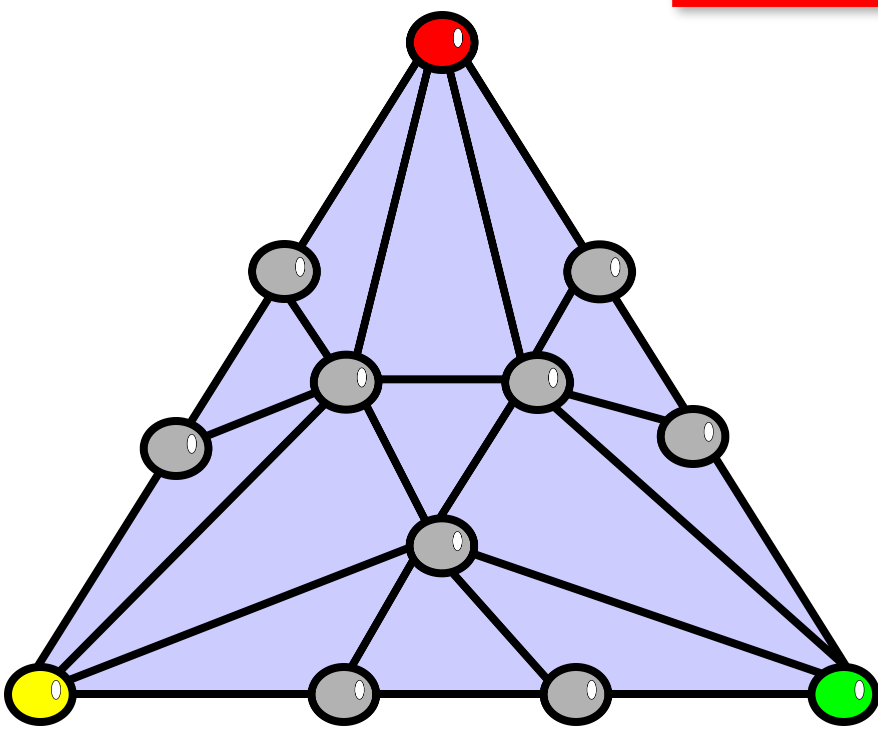
A ***time-complexity*** lower bound

Sperner Coloring



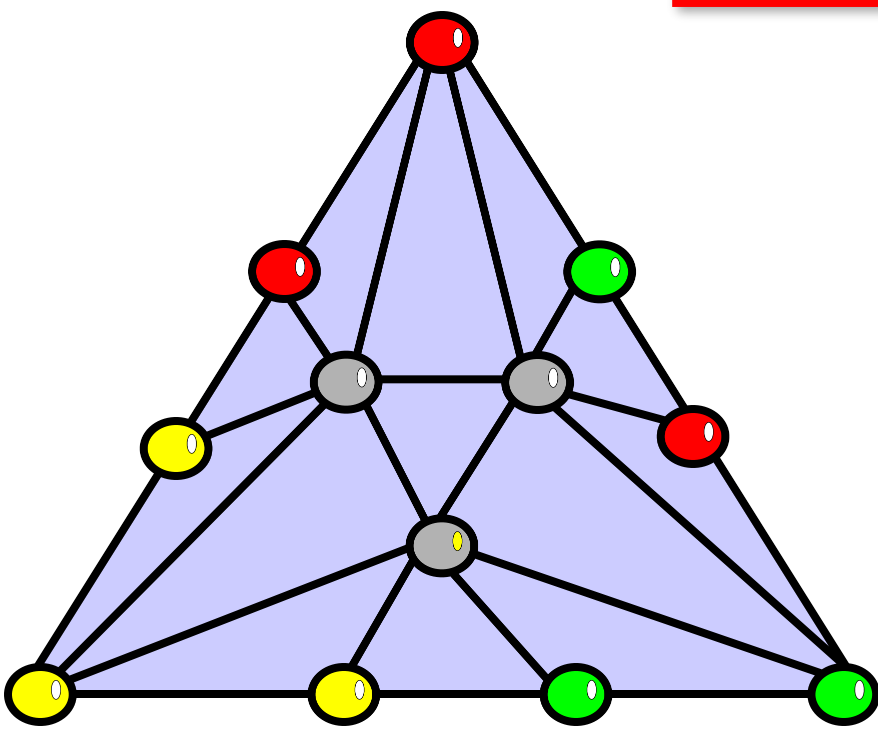
Sperner Coloring

"Corners" have distinct colors



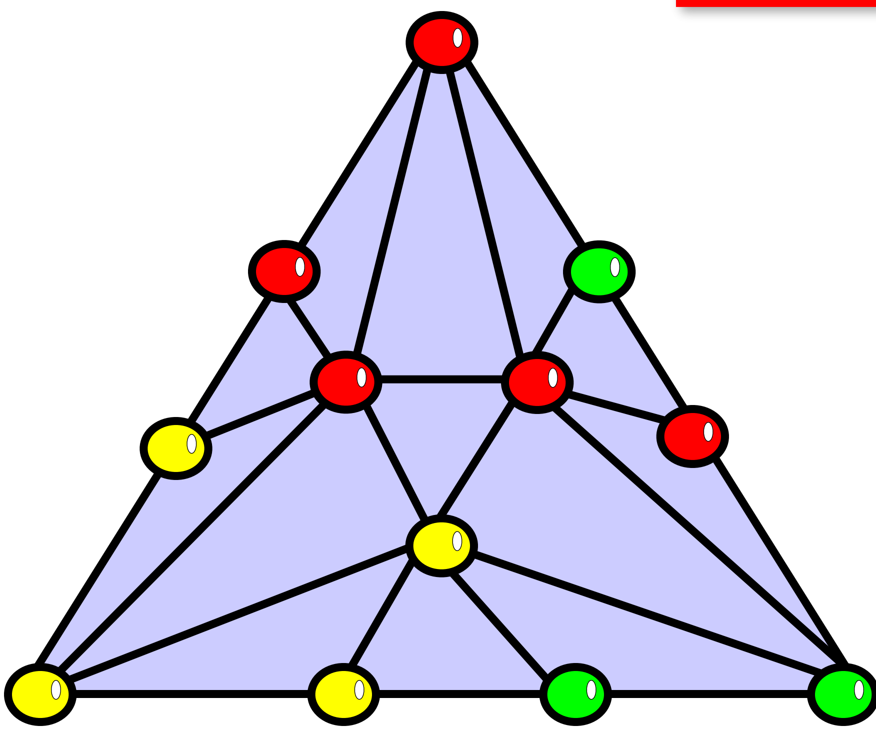
Sperner Coloring

"Corners" have distinct colors



Edge vertexes have corner colors

Sperner Coloring

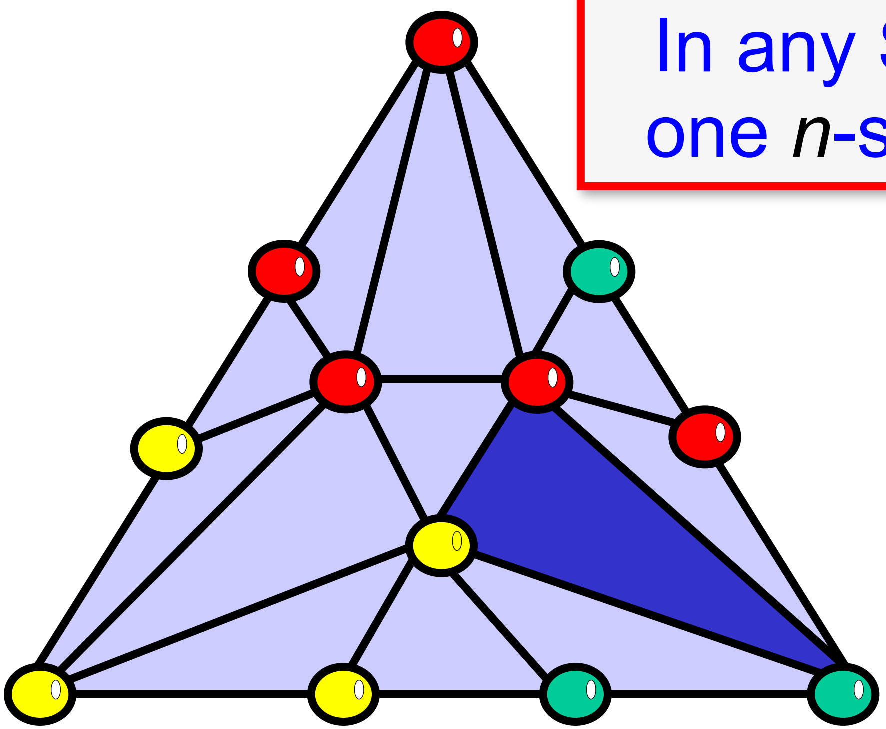


“Corners” have distinct colors

Edge vertexes have corner colors

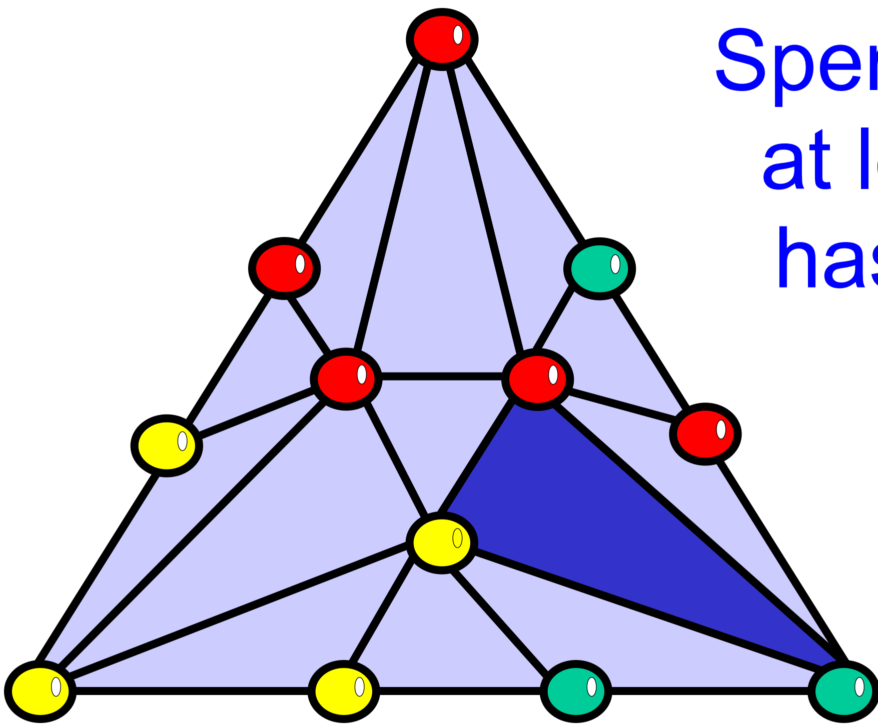
Every vertex has face boundary colors

Sperner's Lemma



In any Sperner coloring, at least one *n*-simplex has all *n+1* colors

Sperner's Lemma

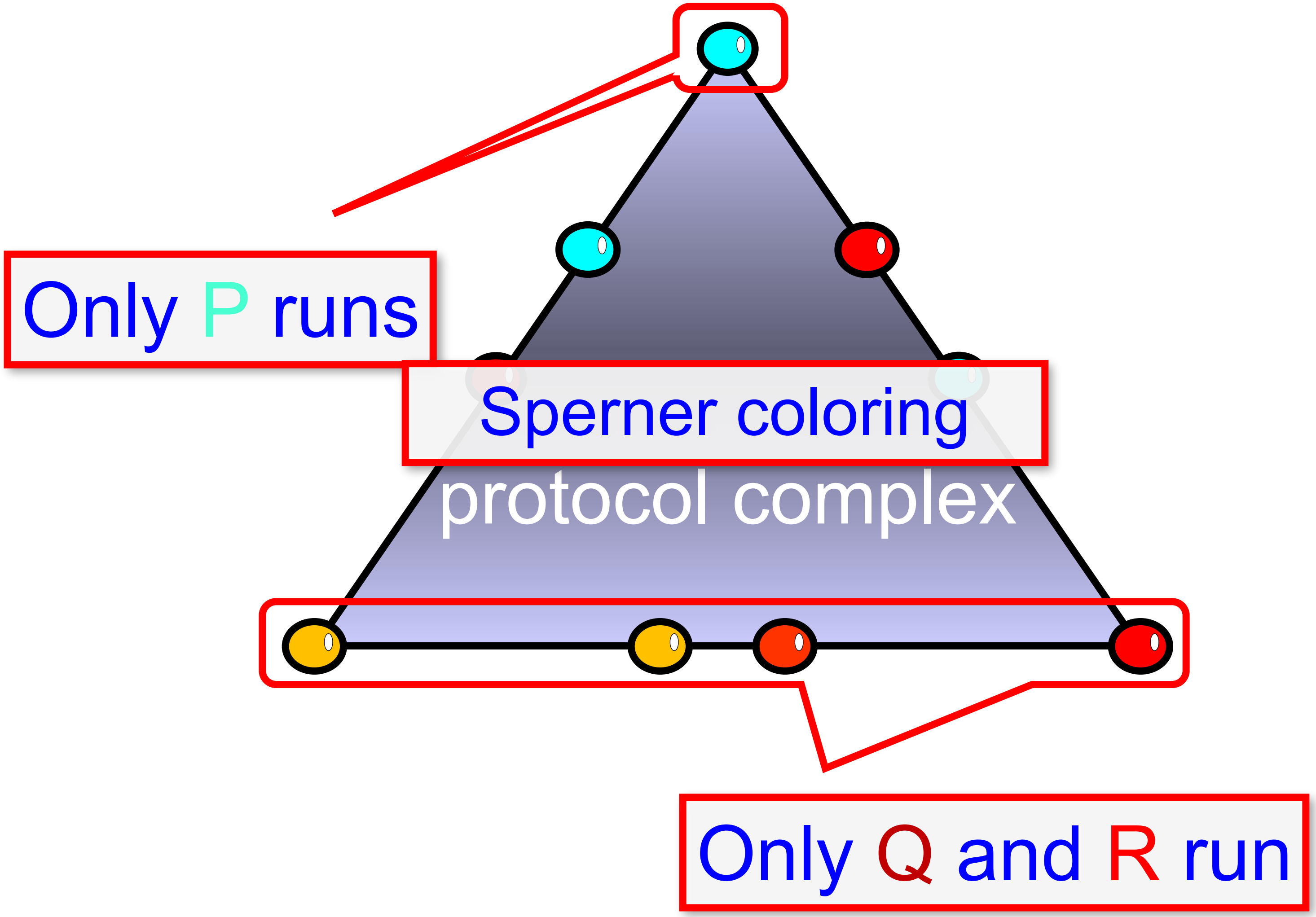


If the boundary has a Sperner coloring, then at least one triangle has all three colors

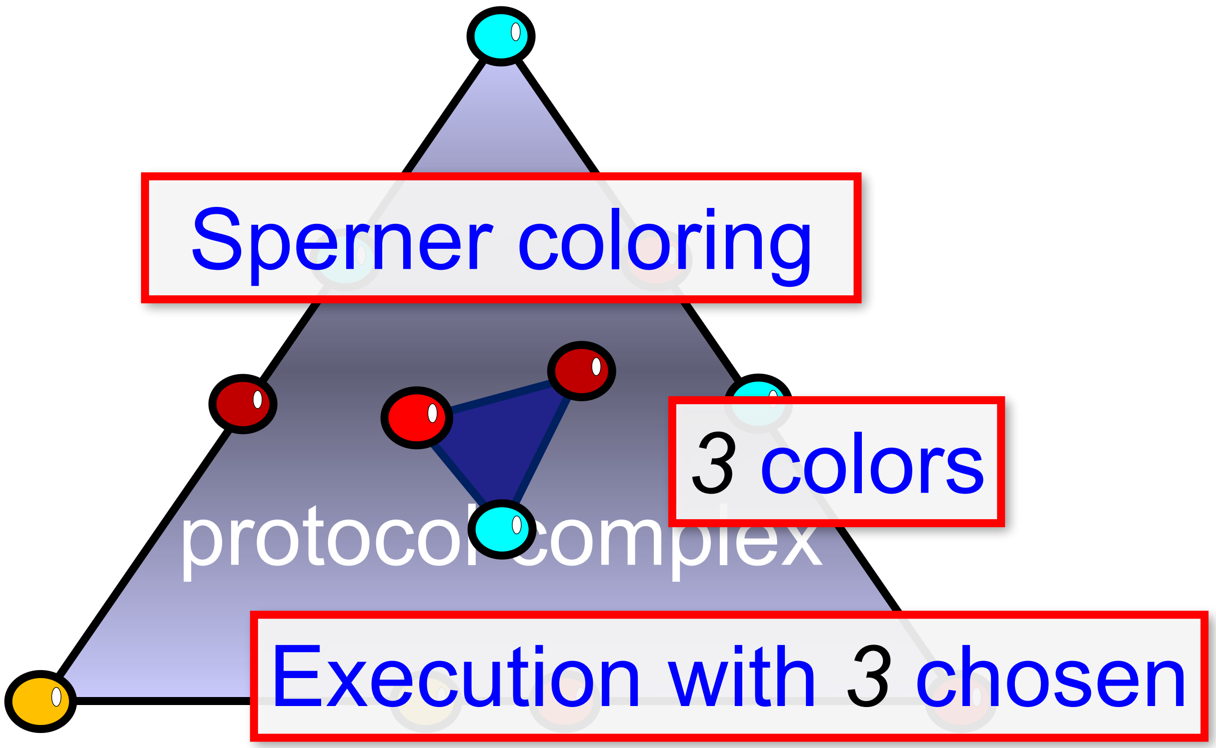
Asynchronous *k*-Set Agreement is Impossible

3-process asynchronous read-write protocol complex is a subdivided triangle
(trust me)

2-Set Agreement



Impossibility of 2-Set Agreement



Contradiction: at most 2 can be chosen

Thank You!