Combinatorial Topology and Distributed Computing

Maurice Herlihy
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

\( \kappa \)-set agreement
Agree on · $k$ Inputs

$k$-set agreement
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

0-simplex

1-simplex

2-simplex

3-simplex

dimension
Simplicial Complex

Combinatorial: a set of simplexes closed under inclusion.

Geometric: simplexes “glued together” along faces …
Simplicial Maps

Vertex-to-vertex map...

Diagram showing simplicial maps with vertices and arrows connecting them.
Simplicial Map

Vertex-to-vertex map …
that sends simplexes to simplexes
piece-wise linear map on geometric simplexes
Carrier Map

Maps simplex \ldots to subcomplex.

Preserves intersections: \( M (\frac{3}{4} \, \hat{\zeta}) = M (\frac{3}{4}) \, \hat{\zeta} \, M (\hat{\zeta}) \)
Vertex = Process State

Process ID (color)

Value (input or output)
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

All 0 inputs

All 0 outputs
Carrier Map for Consensus

All 1 inputs

All 1 outputs
Carrier Map for Consensus

Mixed 0-1 inputs

All 0 outputs

All 1 outputs
Task Specification

(\(I, O, \mathcal{C}\))

Input complex

Output complex

Carrier map
view = my input value;
for (i = 0; i < r; i++) {
    broadcast view;
    view += messages received;
}
return \delta(\text{view})
Protocol

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

Start with input value
view = my input value;
for (i = 0; i < r; i++) {
broadcast view;
view += messages received;
}
return \( \delta(\text{view}) \)

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 $\delta(view)$

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

no one fails
Single Input: Round One

blue fails

no one fails
Single Input: Round One

- red fails
- green fails
- blue fails

no one fails
Protocol Complex: Round One
Protocol Complex: Round Two
Protocol Complex Evolution

zero

one

two
Summary

protocol complex

input complex

Δ

δ

output complex
Decision Map

Simplicial map, sending simplexes to simplexes

Protocol complex

Output complex
Lower Bound Strategy

Find topological "obstruction" to this simplicial map

Protocol complex

Output complex
Consensus Example

Subcomplex of all-0 inputs

Protocol

Output

\( \delta \)

Must map here
Consensus Example

Protocol

Subcomplex of all-1 inputs

Must map here

Output

δ
Consensus Example

Path from “all-0” to “all-1”

Image under $\delta$ must start here.. and end here

Protocol

Output
Consensus Example

![Diagram of consensus example with path and output]

- Path: From 1 to 0 to ?
- Output: 1
Consensus Example

Path from "all-0" to "all-1"

But this "hole" is an obstruction

Image under $\delta$ must start here..

and end here

Output
A protocol cannot solve consensus if its complex is \textit{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

Only P runs

Sperner coloring
protocol complex

Only Q and R run
Impossibility of 2-Set Agreement

Contradiction: at most 2 can be chosen
Thank You!
This work is licensed under a Creative Commons Attribution-Noncommercial 3.0 Unported License.