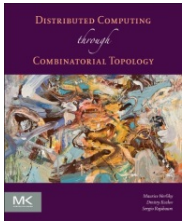
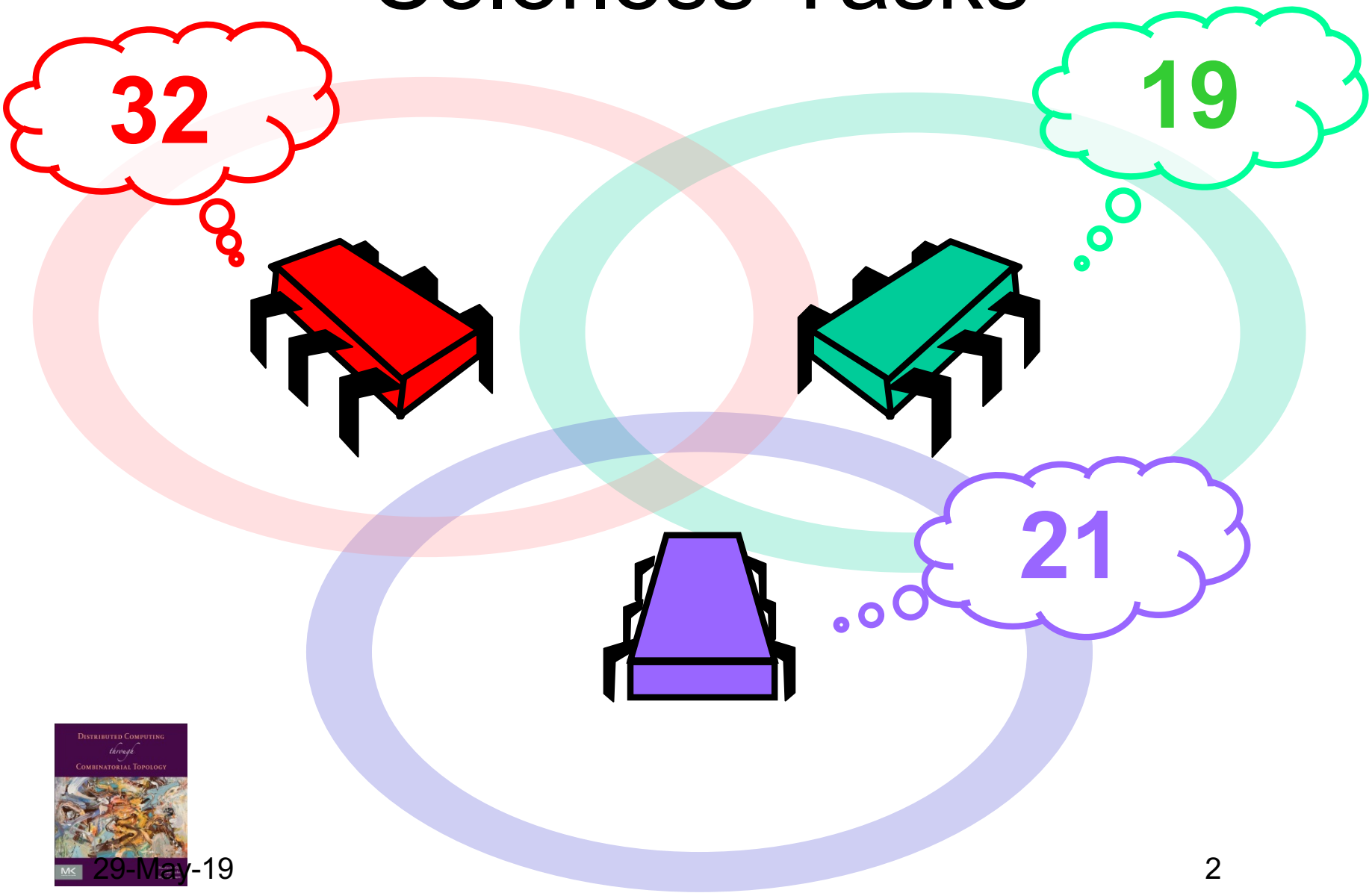
Colorless Tasks

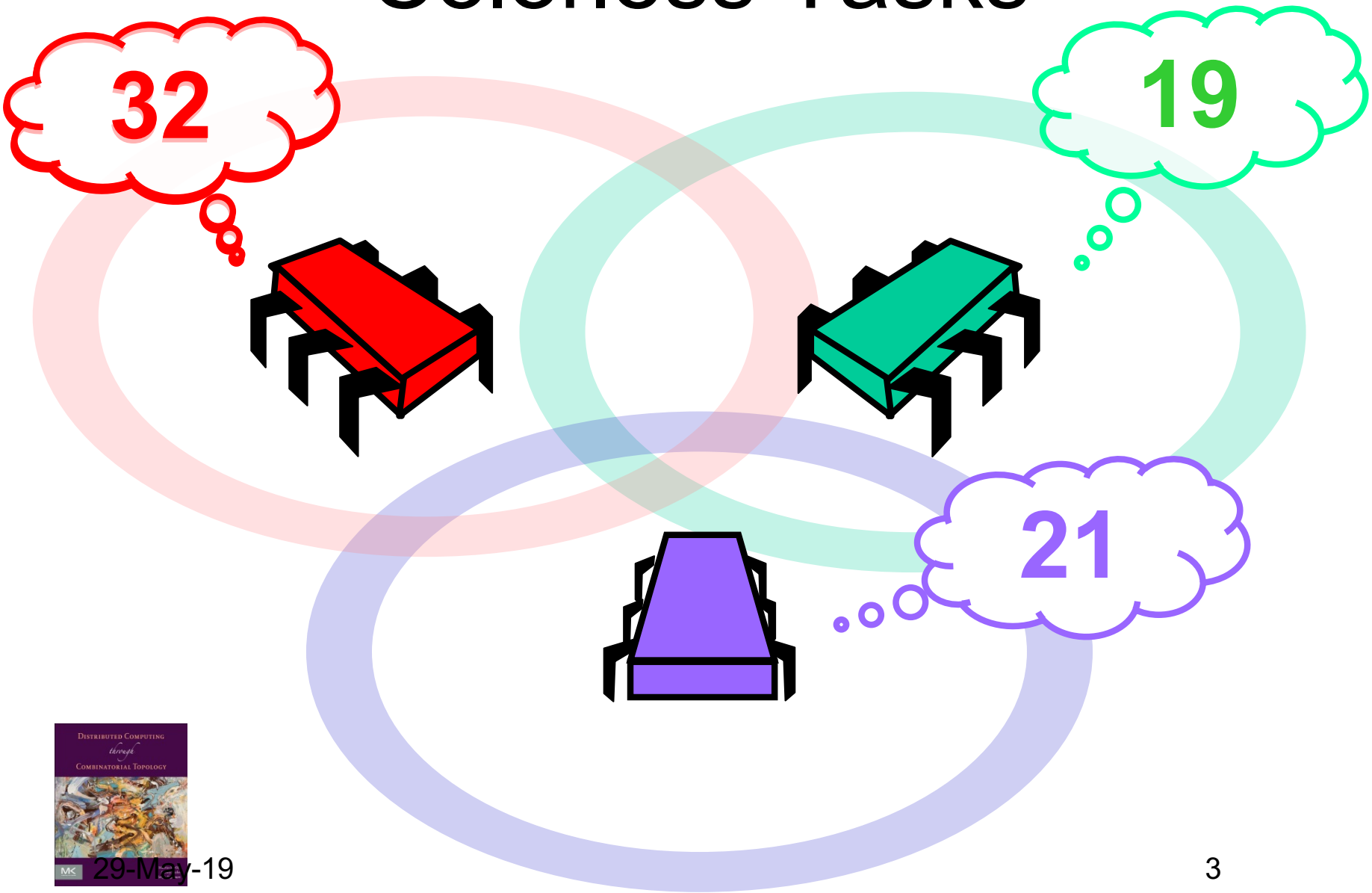
MITRO207, P4, 2019



Colorless Tasks



Colorless Tasks



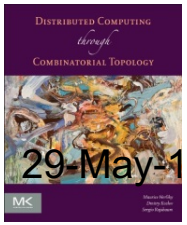
Colorless Tasks

The *set* of input values …

determines the *set* of output values.

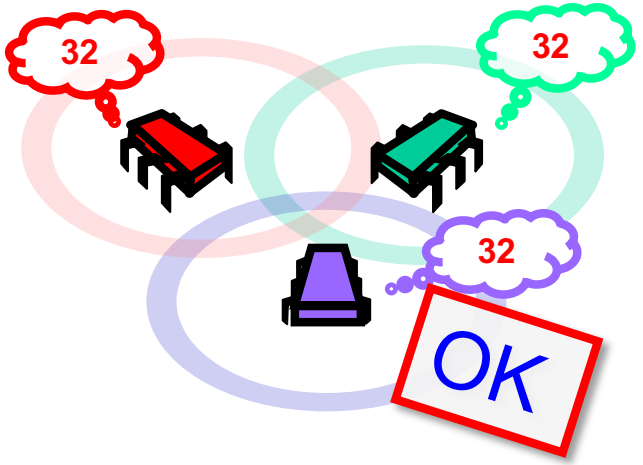
Number and identities irrelevant…

for both input and output values

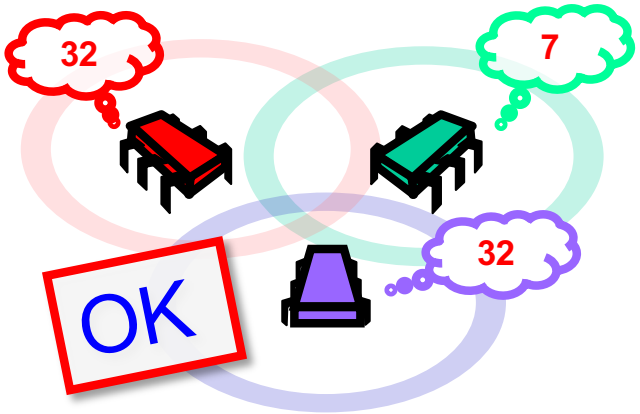


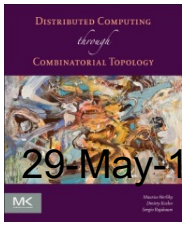
Examples

Consensus



k-set agreement



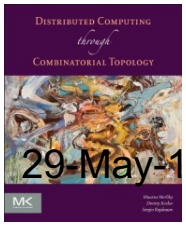


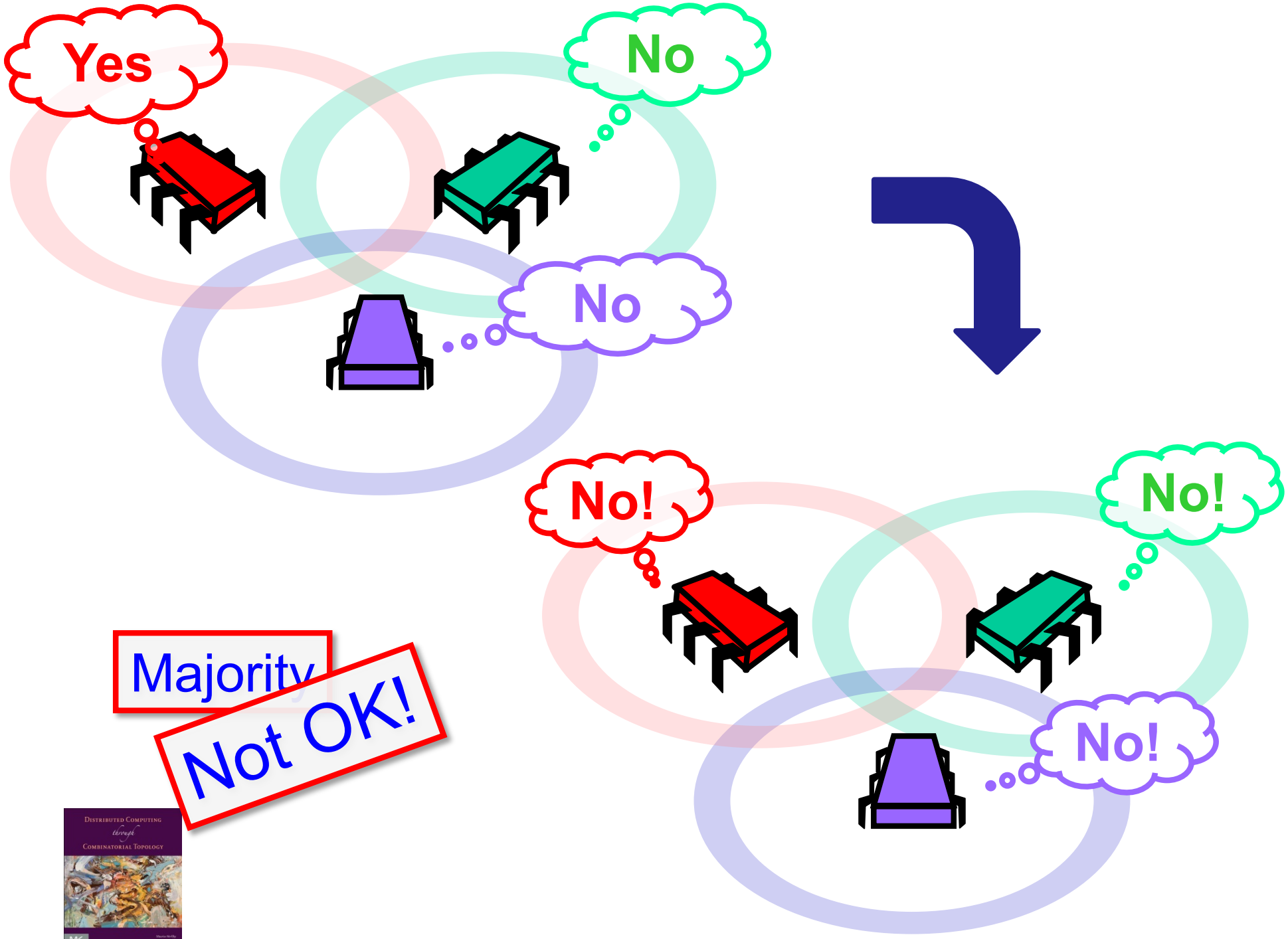
Non-Examples

Weak Symmetry-Breaking

When all participate …

At least one on group 0, at least one on group 1



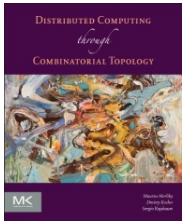

Yes
No
No
No!
No!
No!
Majority
Not OK!
Distributed Computing
through
Combinatorial Topology

Road Map

Operational Model

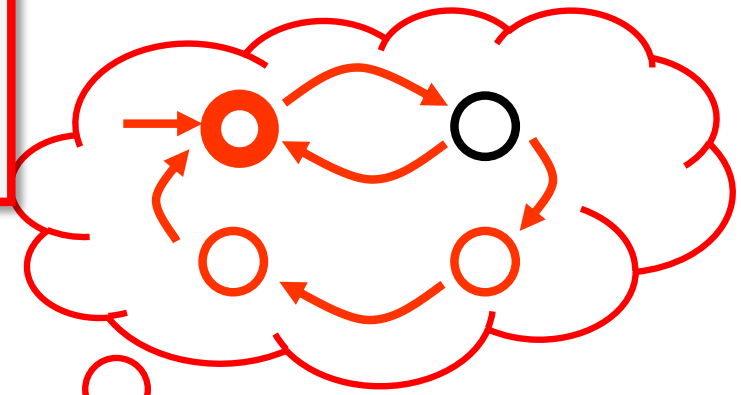
Combinatorial Model

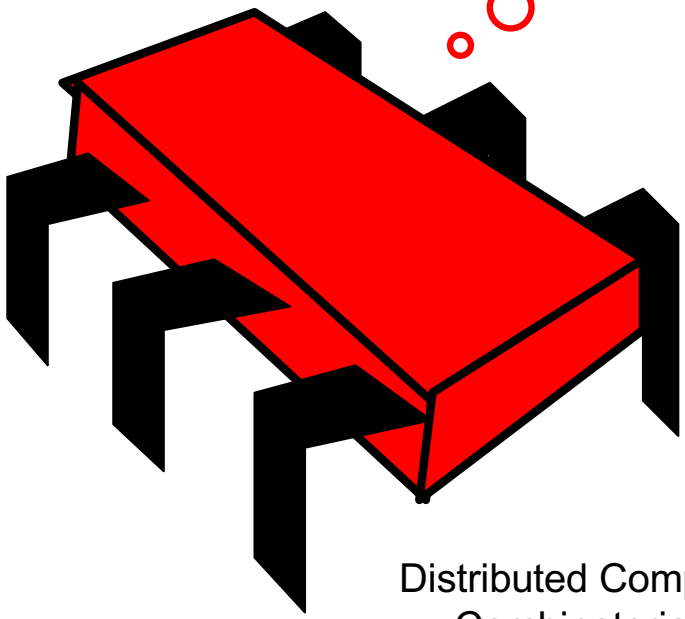
Main Theorem



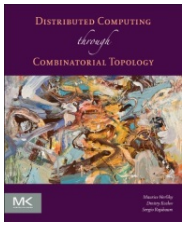
Processes

A *process* is a state machine



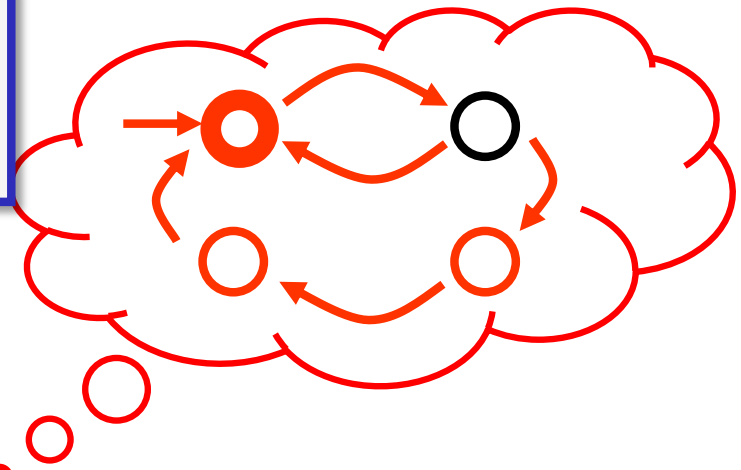


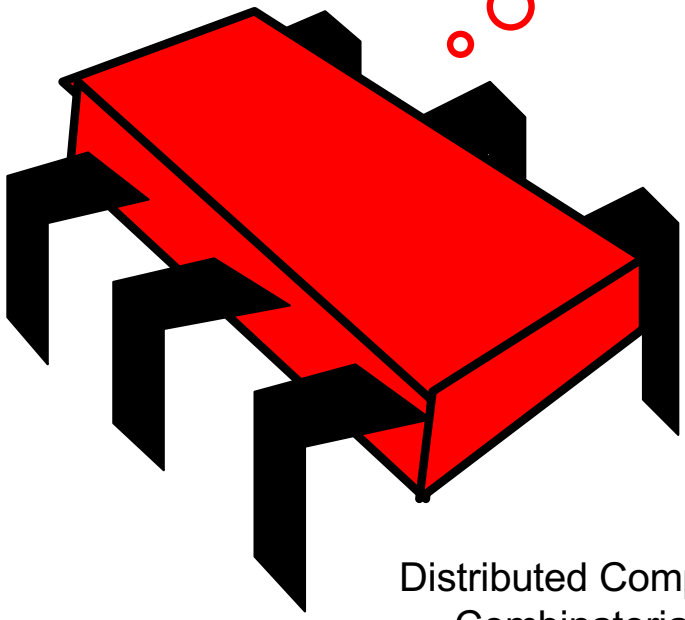
Could be Turing machines or more powerful



Processes

A process's state is called its *view*

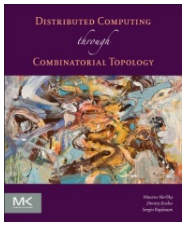




Process names taken from a domain Π

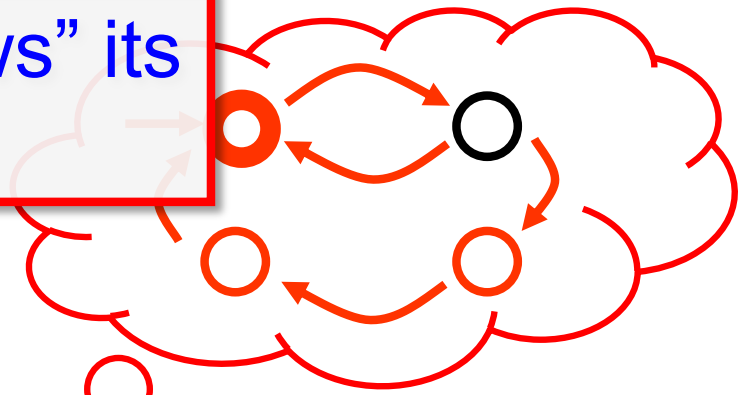
Each process has a unique *name* (color)

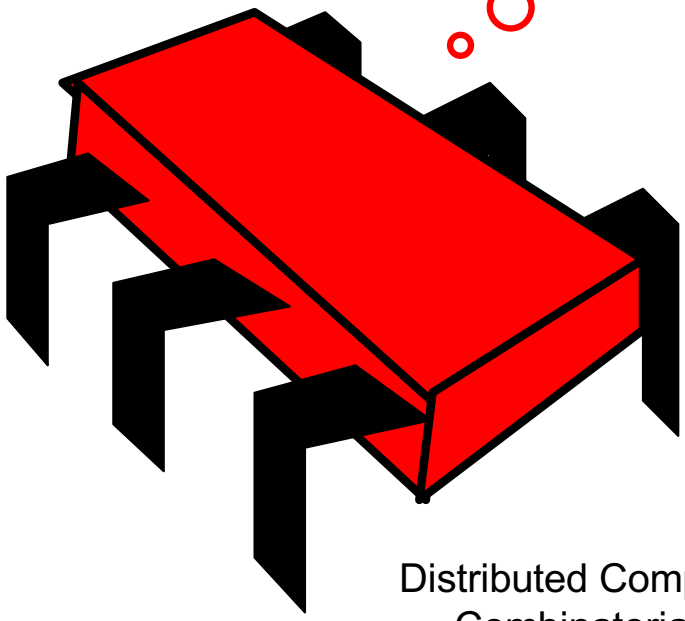
$P_i \in \Pi$



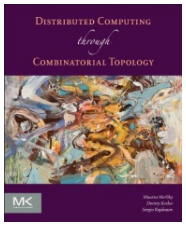
Processes

Each process “knows” its own name



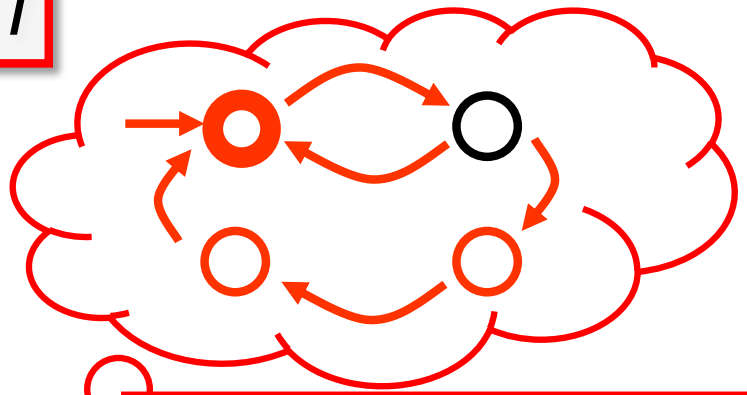


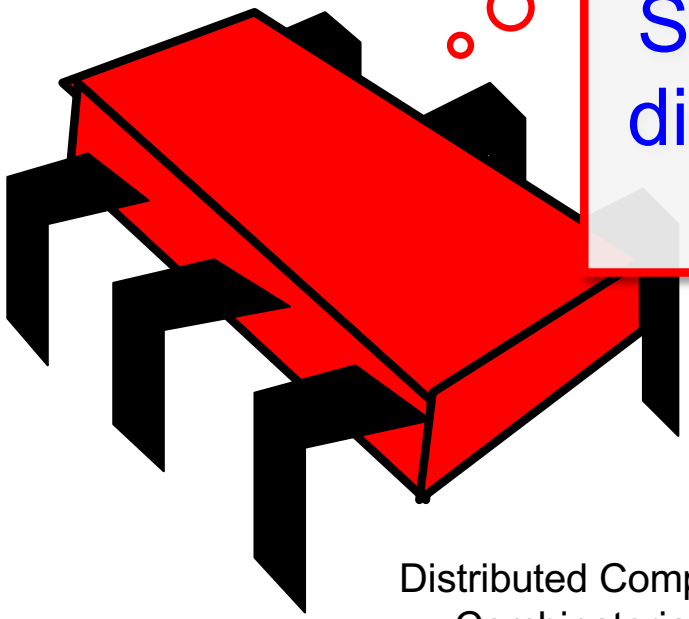
But not the names of the other processes



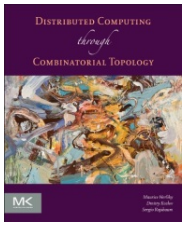
Processes

Often, P_i is just i

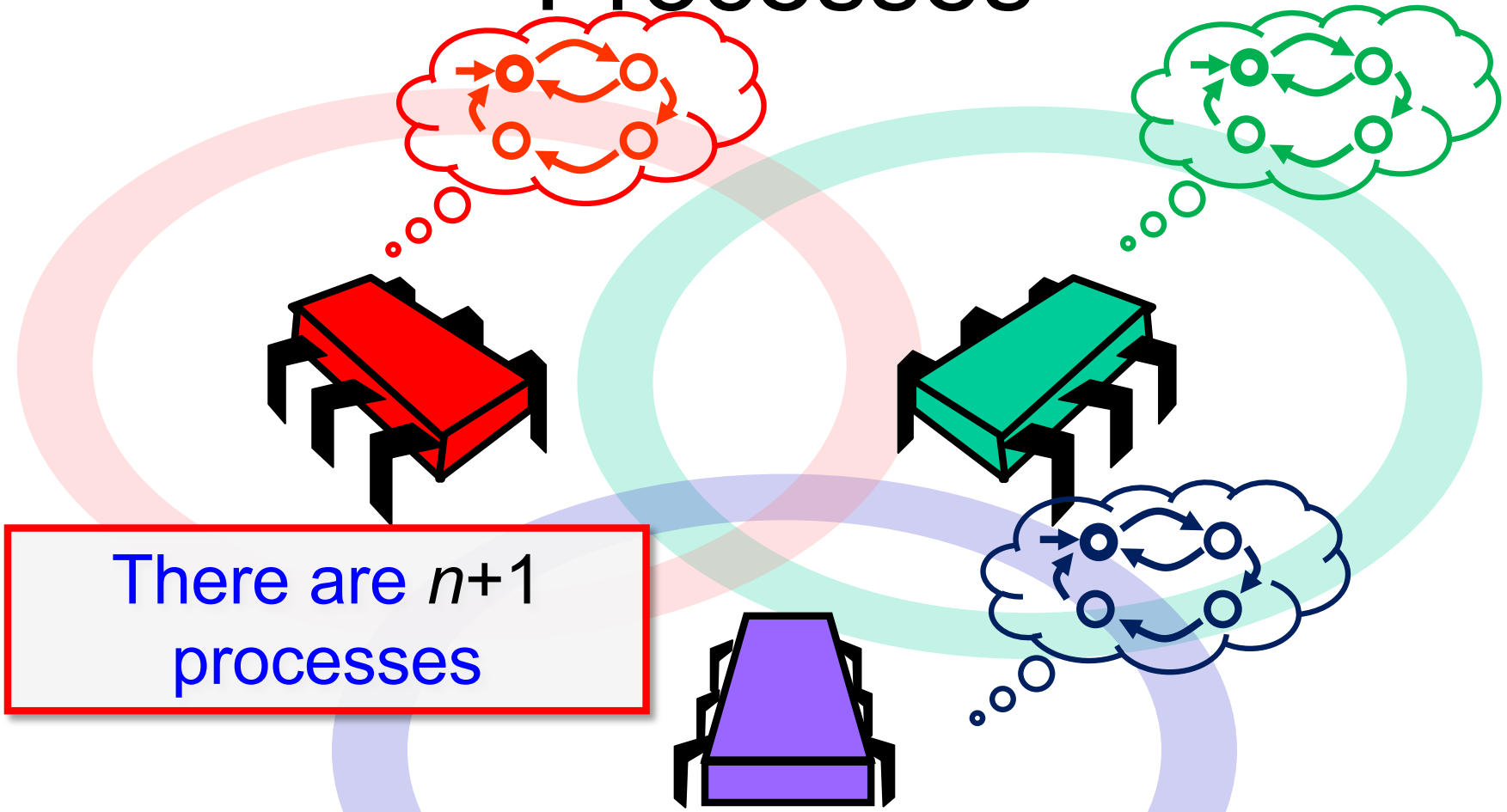




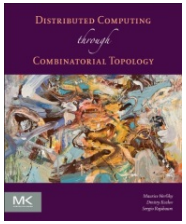
Sometimes P_i and i are distinct, and the process "knows" P_i but not i



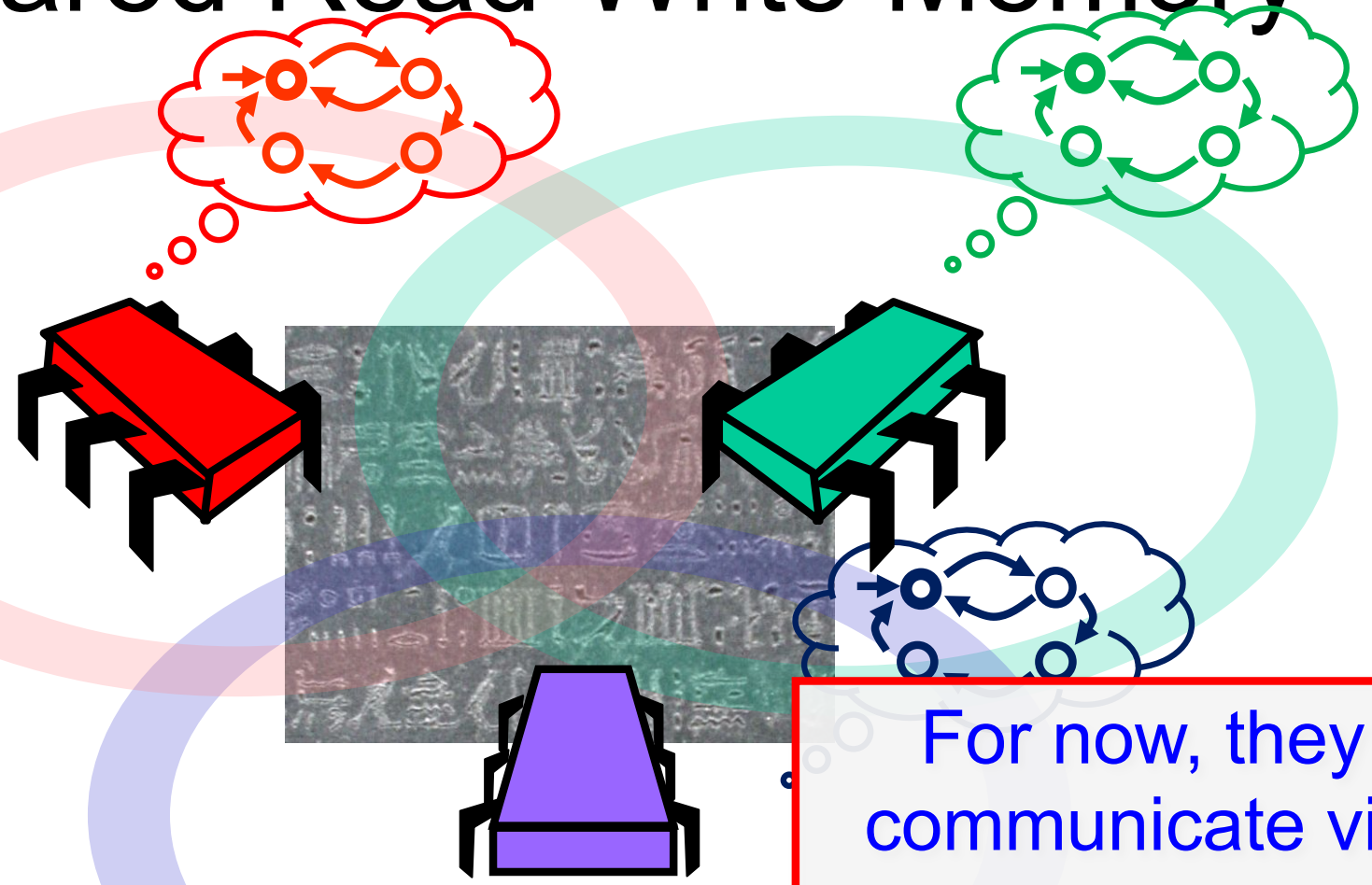
Processes



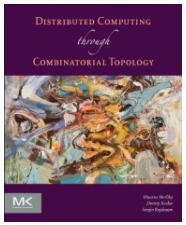
There are n+1 processes



Shared Read-Write Memory



For now, they communicate via reading & writing shared memory

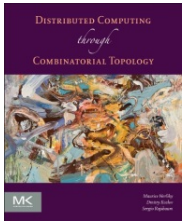


Immediate Snapshot

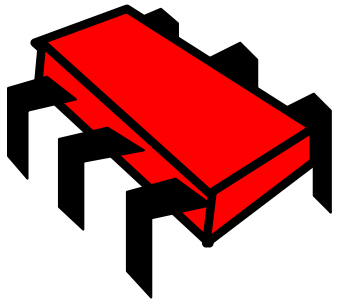
Individual reads & writes are too low-level …

A *snapshot* = atomic read of all memory

We will use *immediate snapshot* …



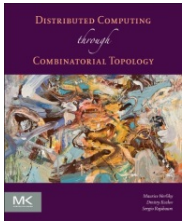
Immediate Snapshot



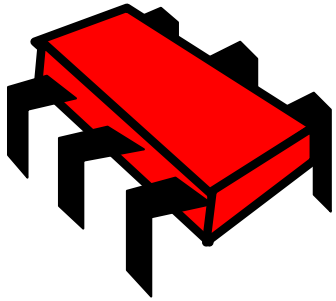
write view to memory

take snapshot

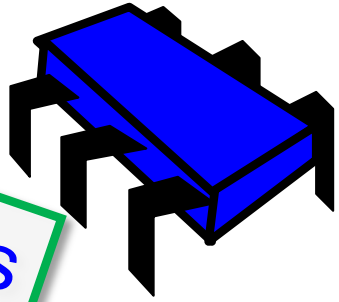
adjacent steps!



Immediate Snapshot



Concurrent steps

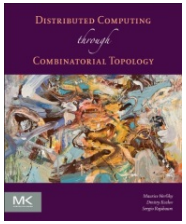


write view to memory

take snapshot

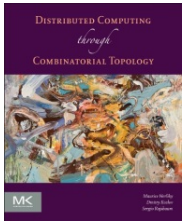
write view to memory

take snapshot

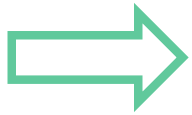


Immediate Snapshot

```
immediate
   mem[i] := view;
   snap := snapshot(mem[*])
```

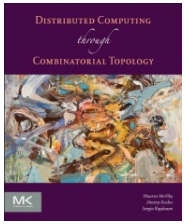


P	Q	R
write		
snap		
	write	
	snap	
		write
		snap
{p}	{p,q}	{p,q,r}



P	Q	R
write		
snap		
	write	write
	snap	snap
{p}	{p,q,r}	{p,q,r}

P	Q	R
write	write	write
snap	snap	snap
{p,q,r}	{p,q,r}	{p,q,r}



Realistic?

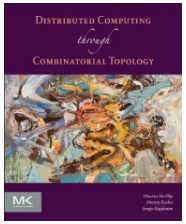
No!

My laptop reads only a few contiguous memory words at a time

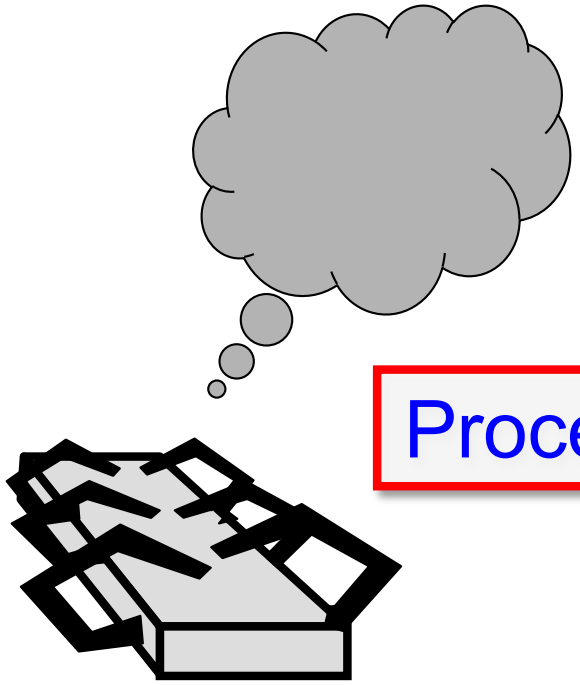
Yes!

Simpler lower bounds: if it's impossible with IS, it's impossible on your laptop.

Can implement IS from read-write

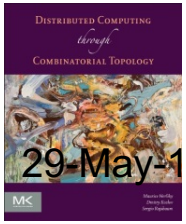


Crashes

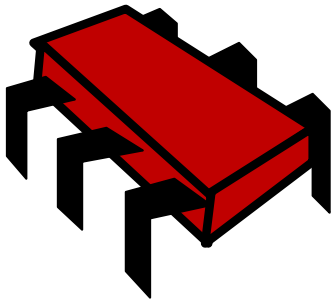


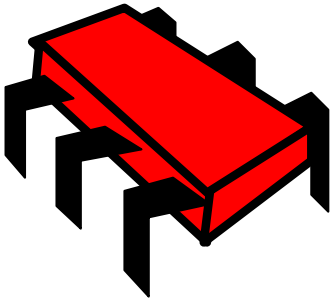
Processes may halt without warning

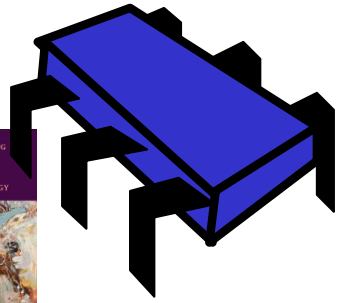
as many as n out of n+1

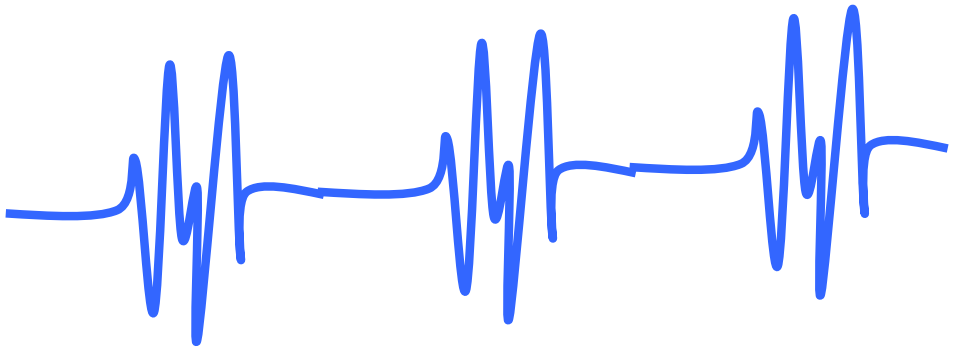


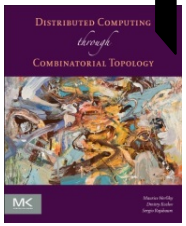
Asynchronous



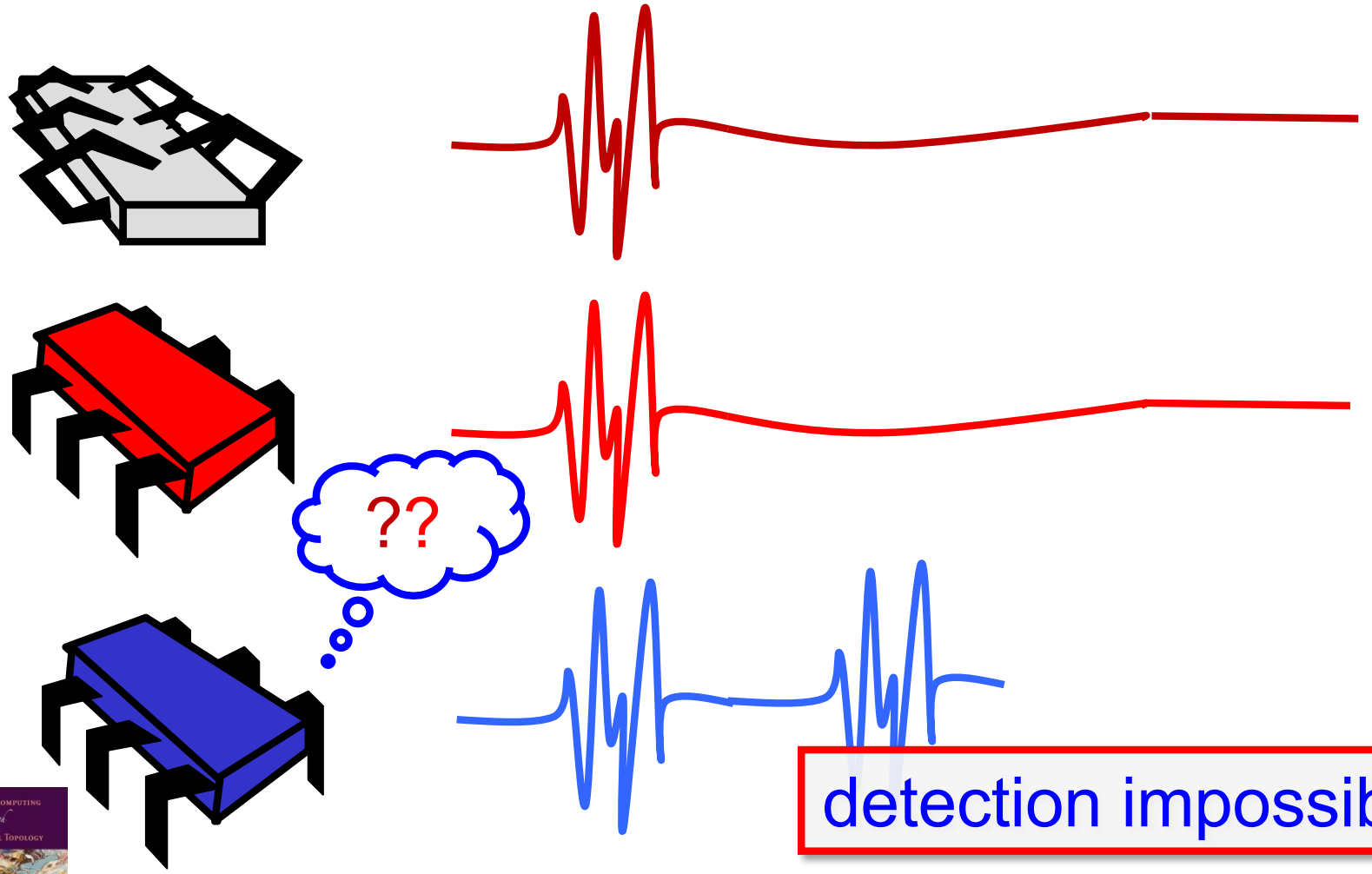


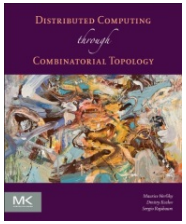






Asynchronous Failures





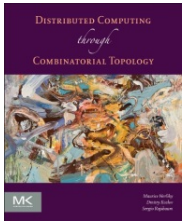
Configurations

$C = \{s_0, \ldots, s_n\}$

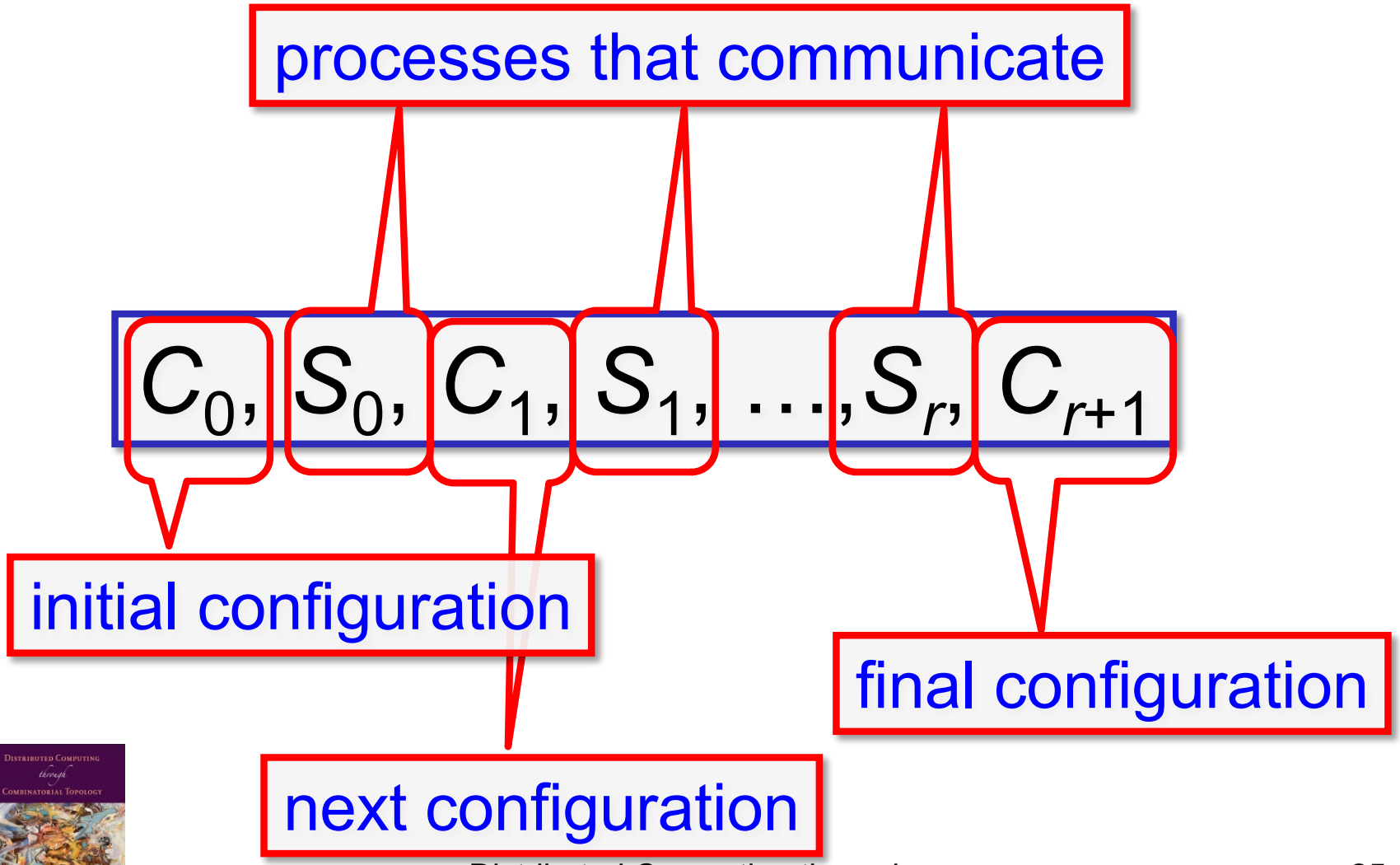
set of *simultaneous* process states

initial configurations

final configurations



Executions


processes that communicate
$C_0, S_0, C_1, S_1, \ldots, S_r, C_{r+1}$
initial configuration
next configuration
final configuration

Distributed Computing
through
Combinatorial Topology

Executions

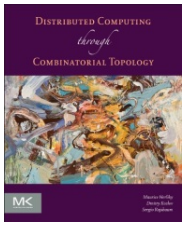
triple is a *concurrent step*

$C_0, S_0, C_1, S_1, \ldots, S_r, C_{r+1}$

Processes in S_0 said to *participate* in step

Only $P_i \in S_0$ can change between C_0 and C_1

state change - result of communication



Executions

finite

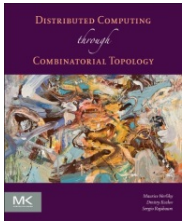
$C_0, S_0, C_1, S_1, \ldots, S_r, C_{r+1}$

infinite

$C_0, S_0, C_1, S_1, \ldots,$

partial

$C_0, S_0, C_1, S_1, \ldots, S_r, C_{r+1}$

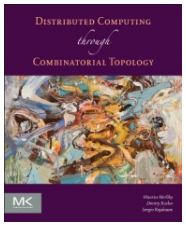


Crashes are Implicit

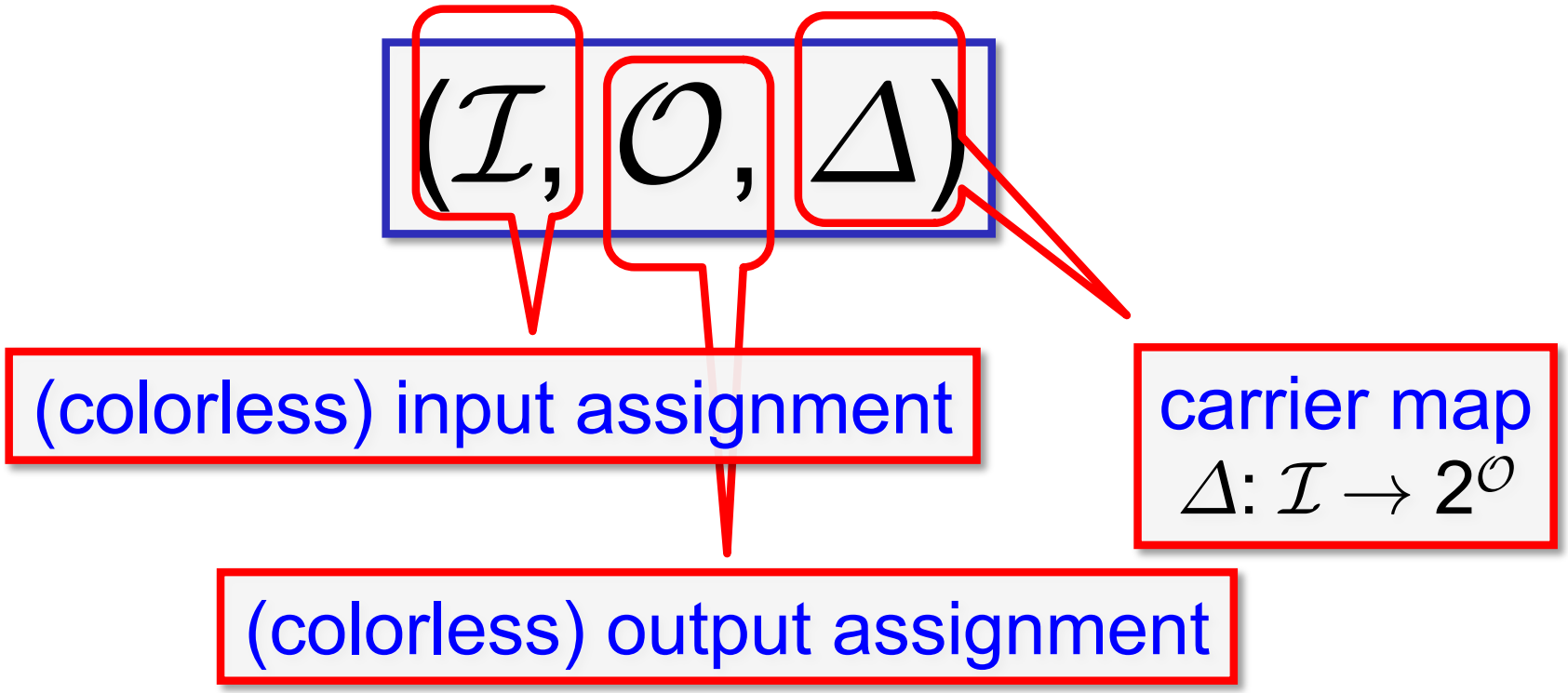
$C_0, S_0, C_1, S_1, \ldots, S_r, C_{r+1}$

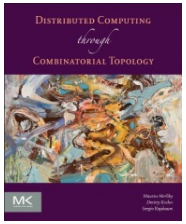
If P_i's state not final in the final configuration then P_i can be *crashed*.

crash cannot be detected in finite time



Colorless Tasks





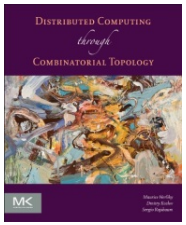
(Colored) Input Assignments

$$\{(P_j, v_j), \mid P_j \in \Pi, v_j \in V^{in}\}$$

name, value pair

domain of *process names*

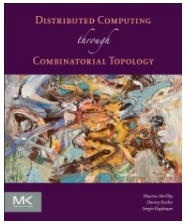
domain of *input values*



Colorless Input Assignments

$$\{(P_j, v_j), \mid P_j \in \Pi, v_j \in V^{in}\}$$

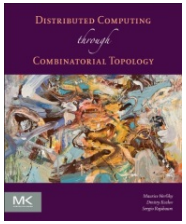
discard process names, keep values



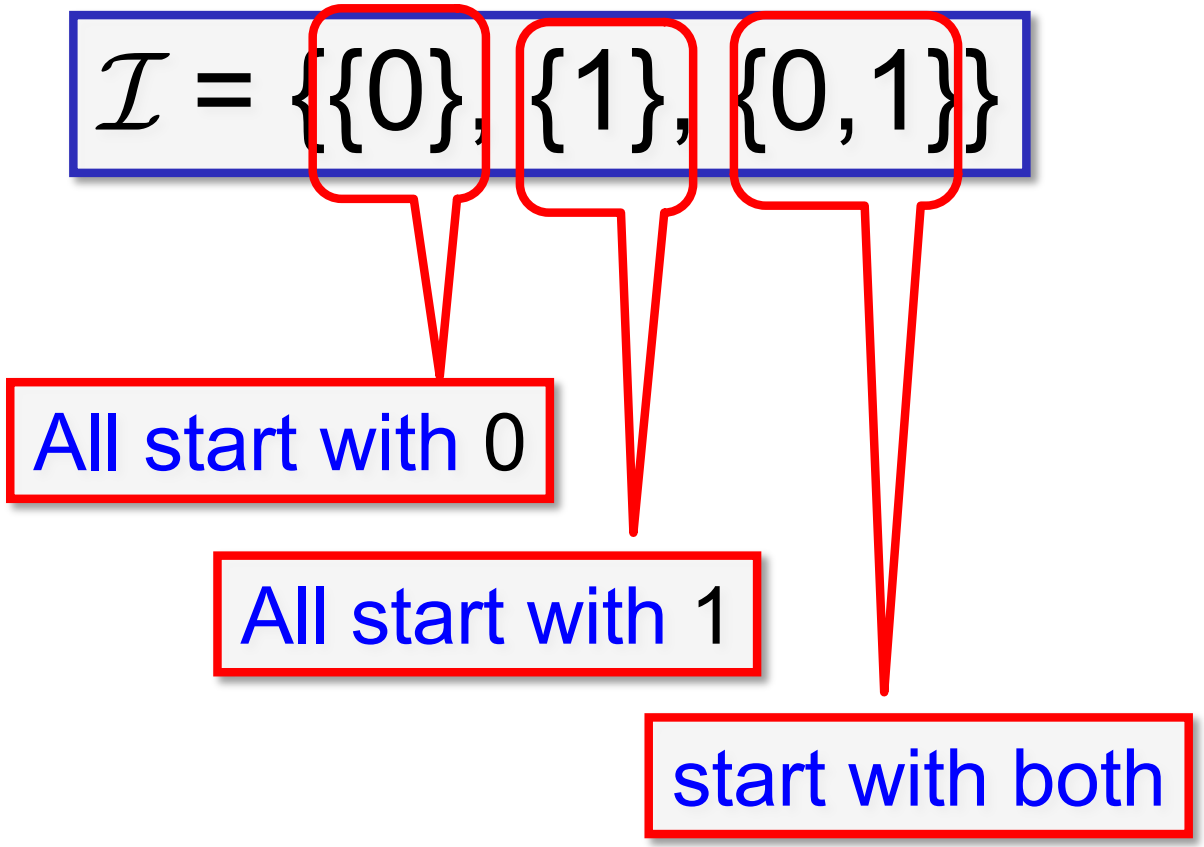
(Colorless) Output Assignments

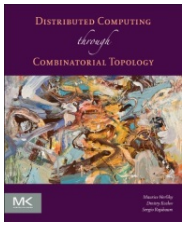
$$\{(P_j, v_j), \mid P_j \in \Pi, v_j \in V^{out}\}$$

$$\{(P_j, v_j), \mid P_j \in \Pi, v_j \in V^{out}\}$$



Example: Binary Consensus





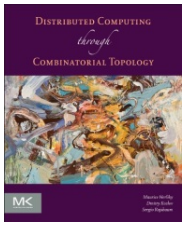
Example: Binary Consensus

$\mathcal{I} = \{\{0\}, \{1\}, \{0,1\}\}$

$\mathcal{O} = \{\{0\}, \{1\}\}$

All decide 0

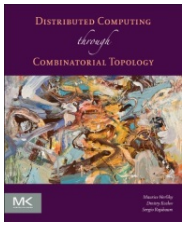
All decide 1



Example: Binary Consensus

$\Delta(\{0\}) = \{\{0\}\}$

All start with 0,
all decide 0

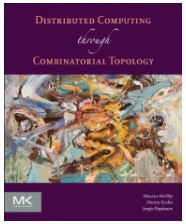


Example: Binary Consensus

$\Delta(\{0\}) = \{\{0\}\}$

$\Delta(\{1\}) = \{\{1\}\}$

All start with 1,
all decide 1



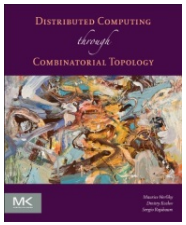
Example: Binary Consensus

$\Delta(\{0\}) = \{\{0\}\}$

$\Delta(\{1\}) = \{\{1\}\}$

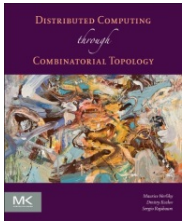
$\Delta(\{0,1\}) = \{\{0\},\{1\}\}$

with mixed inputs,
all decide 0,
or all decide 1



Colorless Layered Protocol

```
shared mem array 0..N-1,0..n of Value
view := input
for l := 0 to N-1 do
   immediate
      mem[l][i] := view;
      snap := snapshot(mem[l][*])
   view := set of values in snap
return δ(view)
```



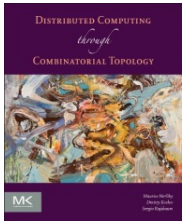
Colorless Layered Protocol

```
shared mem array 0..N-1,0..n of Value
view := input
for j := 0 to N-1 do
   immediate
     mem[j][i] := view;
     snap := snapshot(mem[j][ ]);
   view := set of values in snap
return δ(view)
```

2-dimensional memory array

row is clean *per-layer* memory

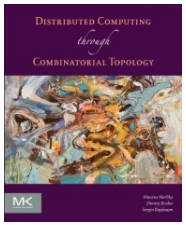
column is *per-process* word



Colorless Layered Protocol

```
shared mem array 0..N-1,0..n of Value
view := input
for j := 0 to N-1 do
   immediate
     mem[j][input] := input
     snap := snapshot(mem[j][*])
   view := set of values in snap
return δ(view)
```

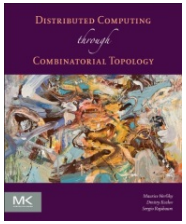
initial view is input value



Colorless Layered Protocol

```
shared mem array 0..N-1,0..n of Value
view := input
for l := 0 to N-1 do
   immediate
     mem[j][l] := view;
     snap := snapshot(mem[j][*])
   view := set of values in snap
return δ(view)
```

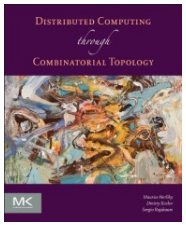
run for *N* layers



Colorless Layered Protocol

```
shared mem array 0..N-1,0.. of Value
view : Input
for j := 0 to N-1 do
  immediate
     mem[l][i] := view;
     snap := snapshot(mem[l][*])
   view := set of values in snap
return δ(view)
```

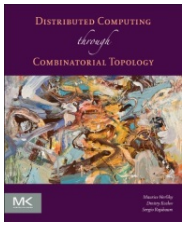
layer l : immediate write & snapshot of row l



Colorless Layered Protocol

```
shared mem array 0..N-1,0..n of Value
view := input
for j := 0 to N-1 do
   immediate
     mem[j][i] := view;
     snap := snapshot(mem[j][*])
   view := set of values in snap
return δ(view)
```

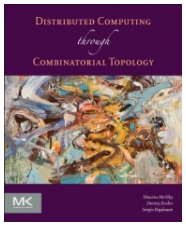
new view is set of values seen

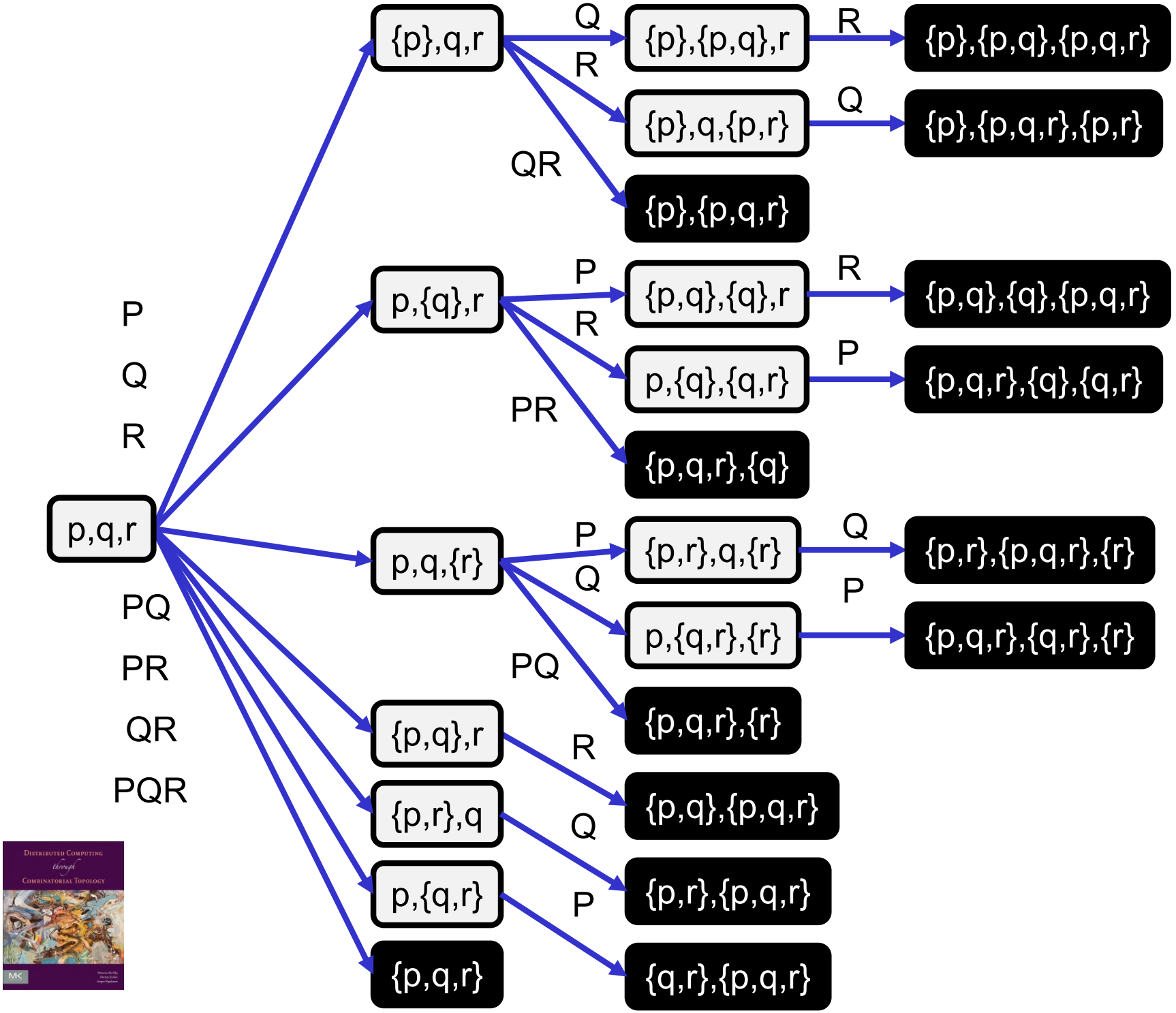


Colorless Layered Protocol

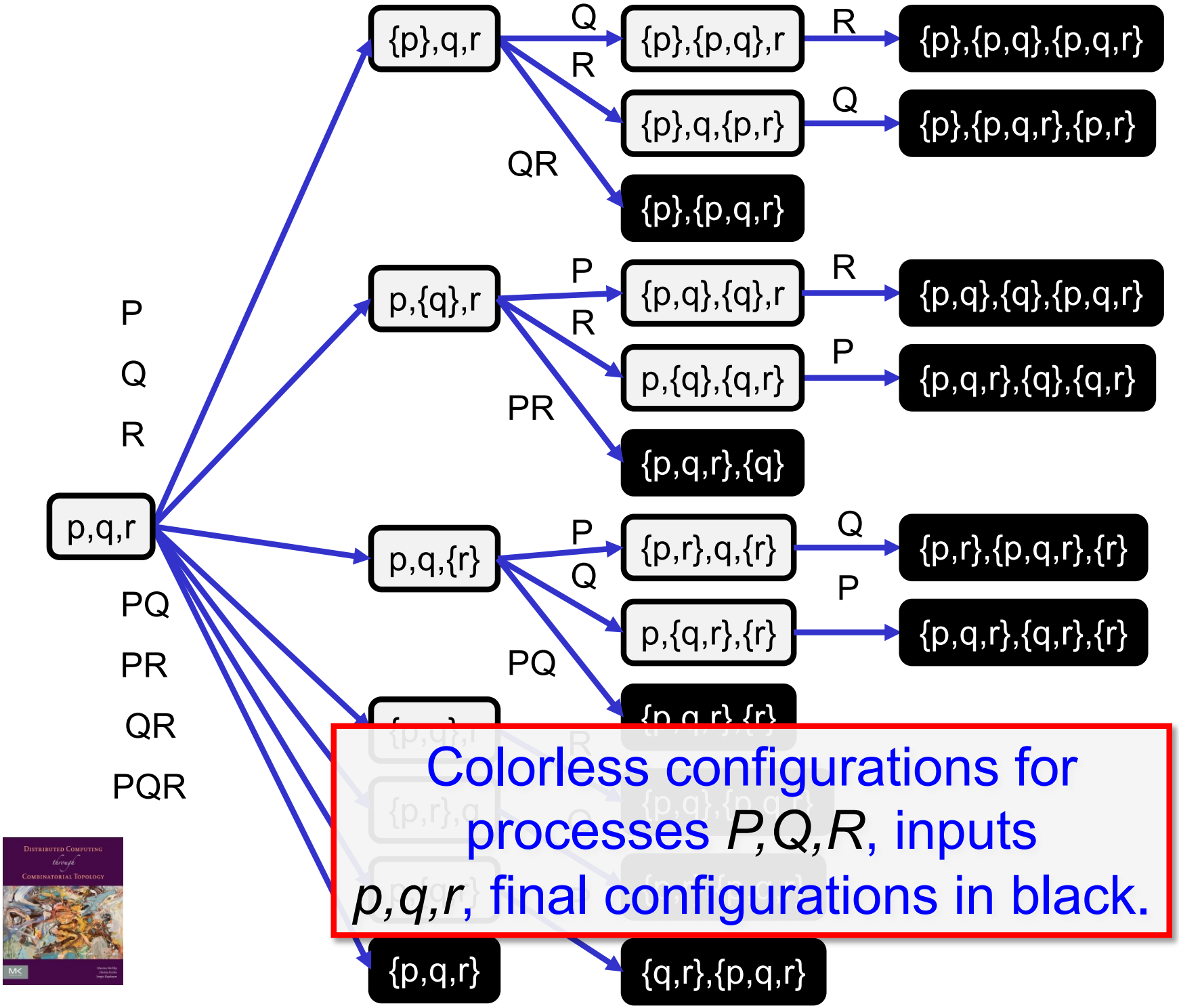
```
shared mem array 0..N-1,0..n of Value
view := input
for j := 0 to N-1 do
  immediate
    snap := snapshot(mem[j][*])
  view := set of values in snap
return δ(view)
```

finally apply decision value to final view




p,q,r
P
Q
R
PQ
PR
QR
PQR
{p},q,r
p,{q},r
p,q,{r}
{p,q},r
{p,r},q
p,{q,r}
{p,q,r}
Q
R
QR
{p},{p,q},r
{p},q,{p,r}
{p},{p,q,r}
R
Q
{p},{p,q},{p,q,r}
{p},{p,q,r},{p,r}
P
R
PR
{p,q},{q},r
p,{q},{q,r}
{p,q,r},{q}
R
P
{p,q},{q},{p,q,r}
{p,q,r},{q},{q,r}
P
Q
PQ
{p,r},q,{r}
p,{q,r},{r}
{p,q,r},{r}
Q
P
{p,r},{p,q,r},{r}
{p,q,r},{q,r},{r}
R
Q
P
{p,q},{p,q,r}
{p,r},{p,q,r}
{q,r},{p,q,r}

Distributed Computing
through
Combinatorial Topology


p,q,r
P
Q
R
PQ
PR
QR
PQR
{p},q,r
Q
R
QR
{p},{p,q},r
{p},q,{p,r}
{p},{p,q,r}
R
Q
{p},{p,q},{p,q,r}
{p},{p,q,r},{p,r}
p,{q},r
P
R
PR
p,{q},{q,r}
{p,q},{q},r
{p,q,r},{q}
R
P
{p,q},{q},{p,q,r}
{p,q,r},{q},{q,r}
p,q,{r}
P
Q
PQ
{p,r},q,{r}
p,{q,r},{r}
Q
P
{p,r},{p,q,r},{r}
{p,q,r},{q,r},{r}
{p,q,r}
{q,r},{p,q,r}
Colorless configurations for processes P,Q,R, inputs p,q,r, final configurations in black.

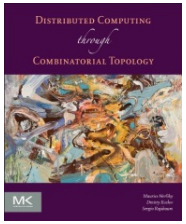
Distributed Computing
through
Combinatorial Topology

Road Map

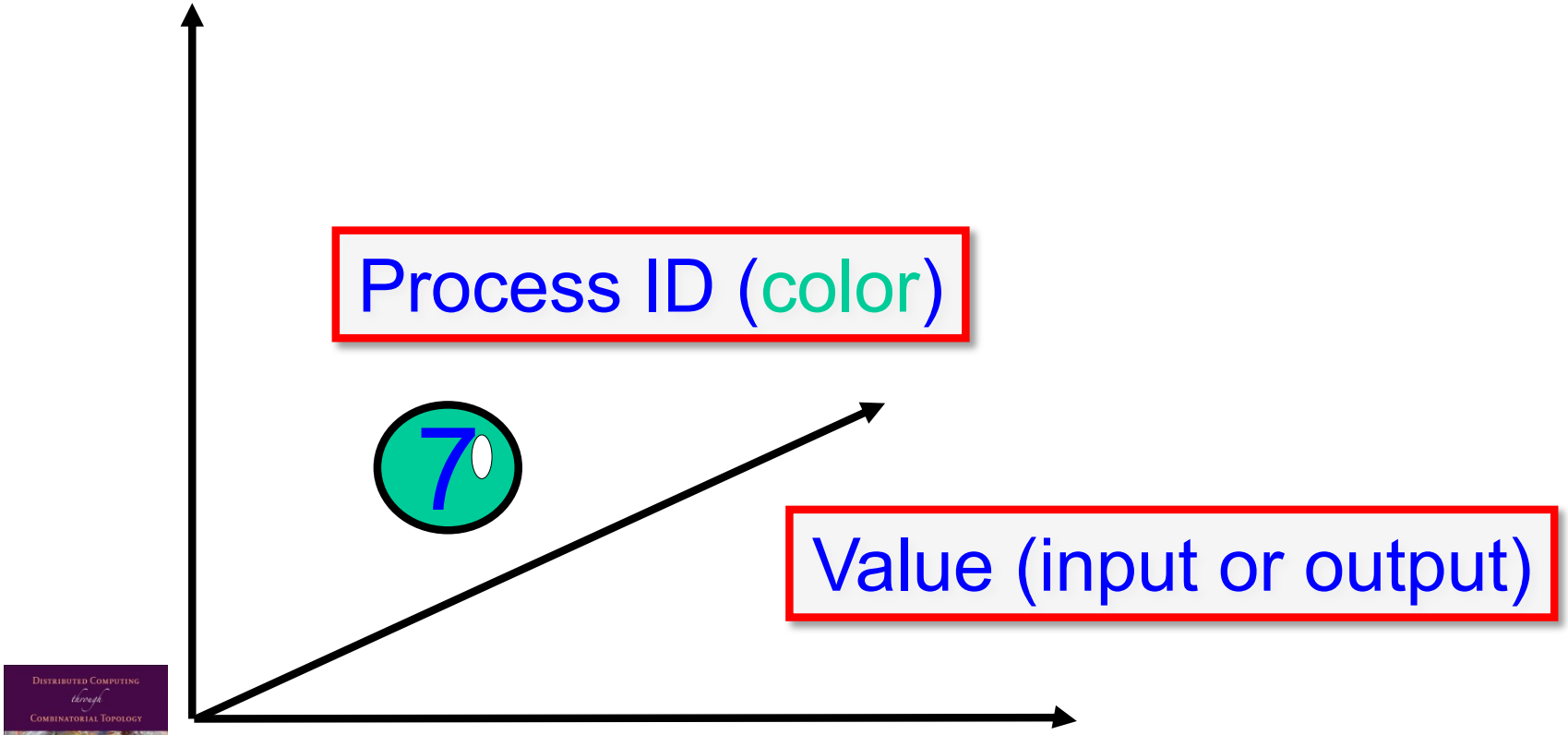
Operational Model

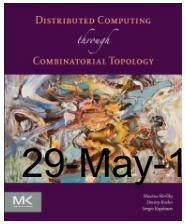
Combinatorial Model

Main Theorem

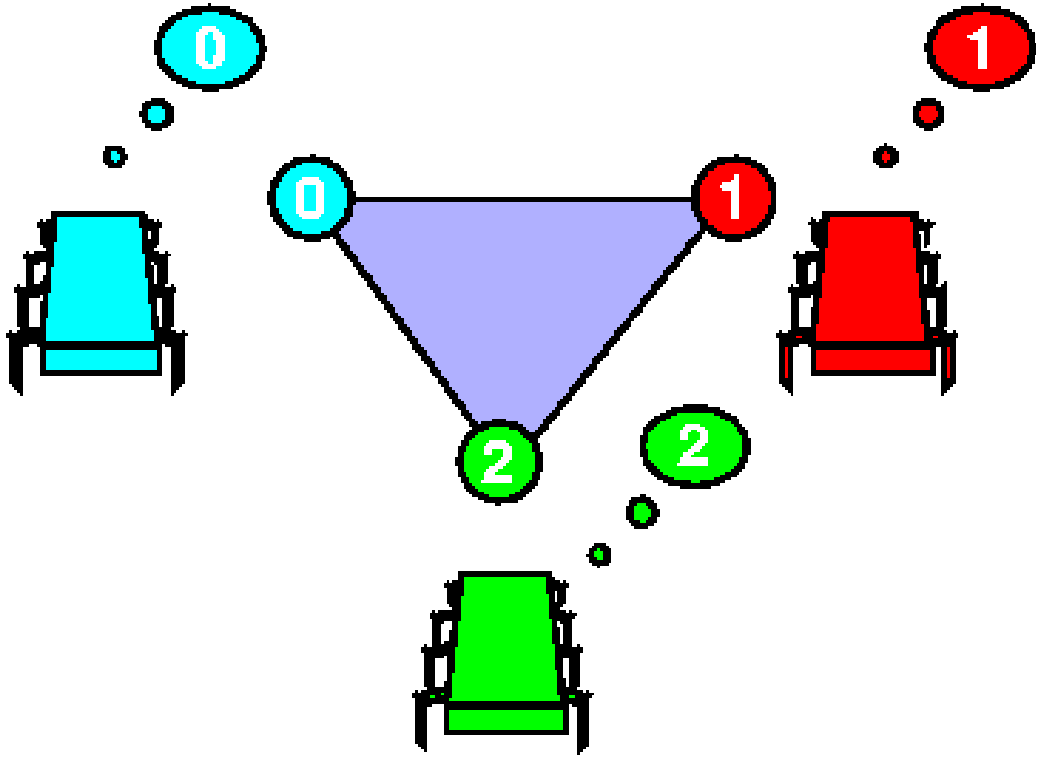


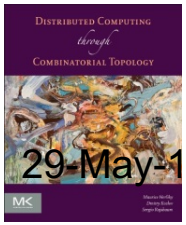
Vertex = Process State



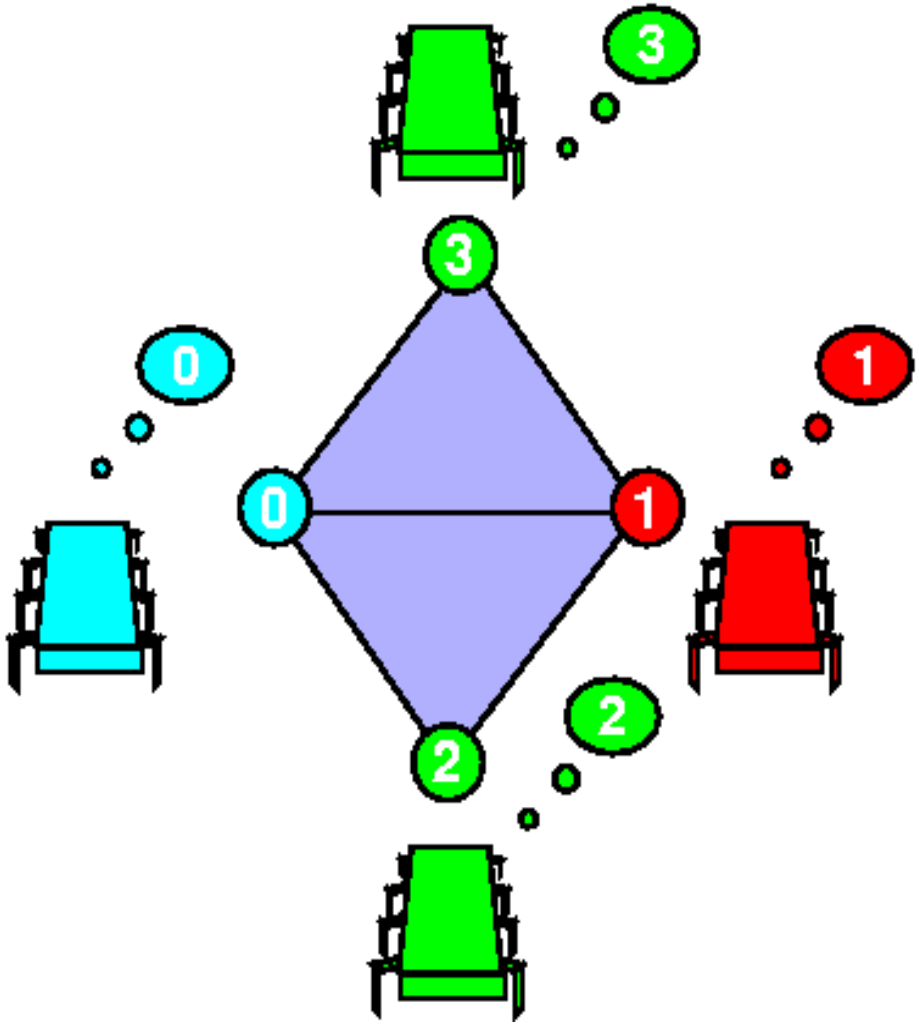


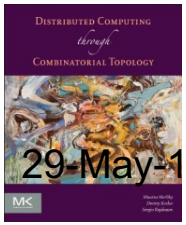
Simplex = Global State



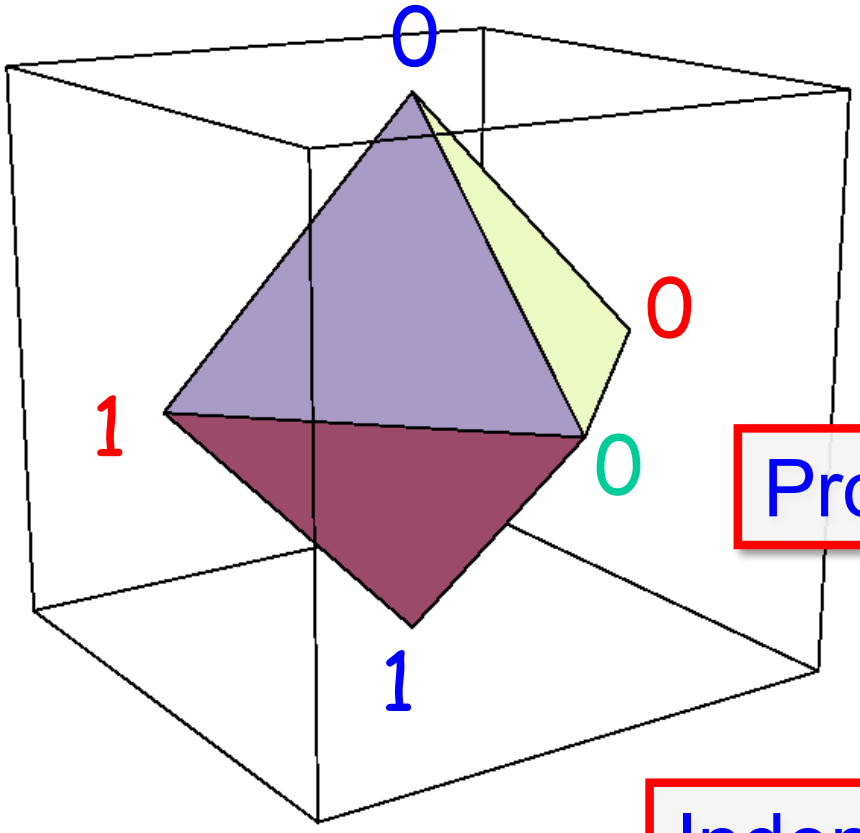


Complex = Global States





Input Complex for Binary Consensus

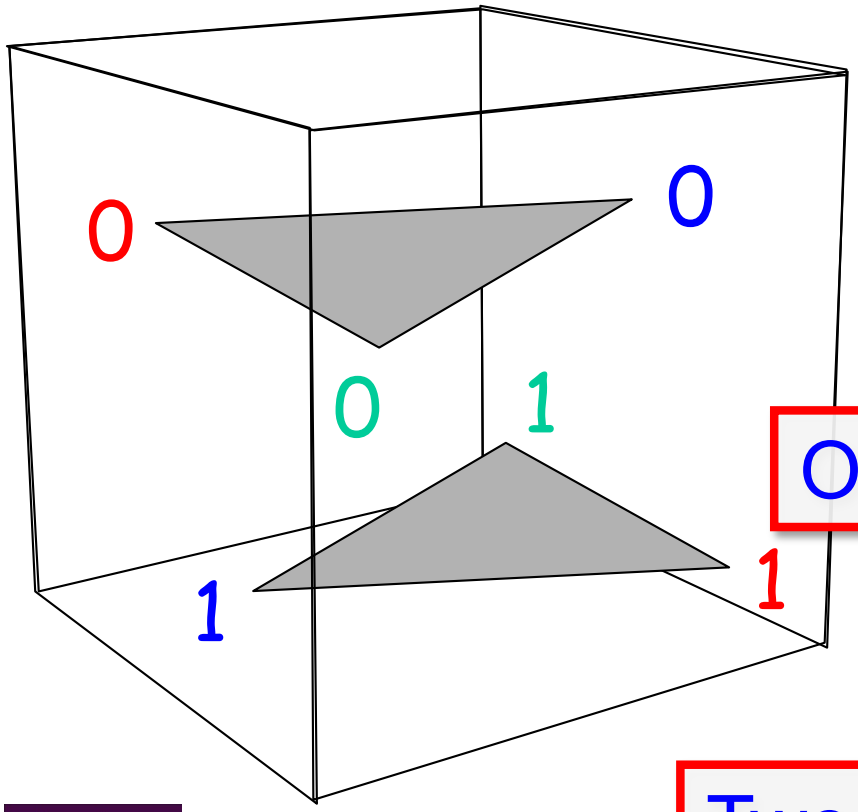


All possible initial states

Processes: red, green, blue

Independently assigned 0 or 1

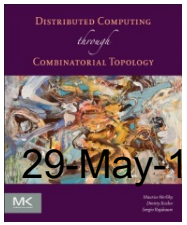
Output Complex for Binary Consensus



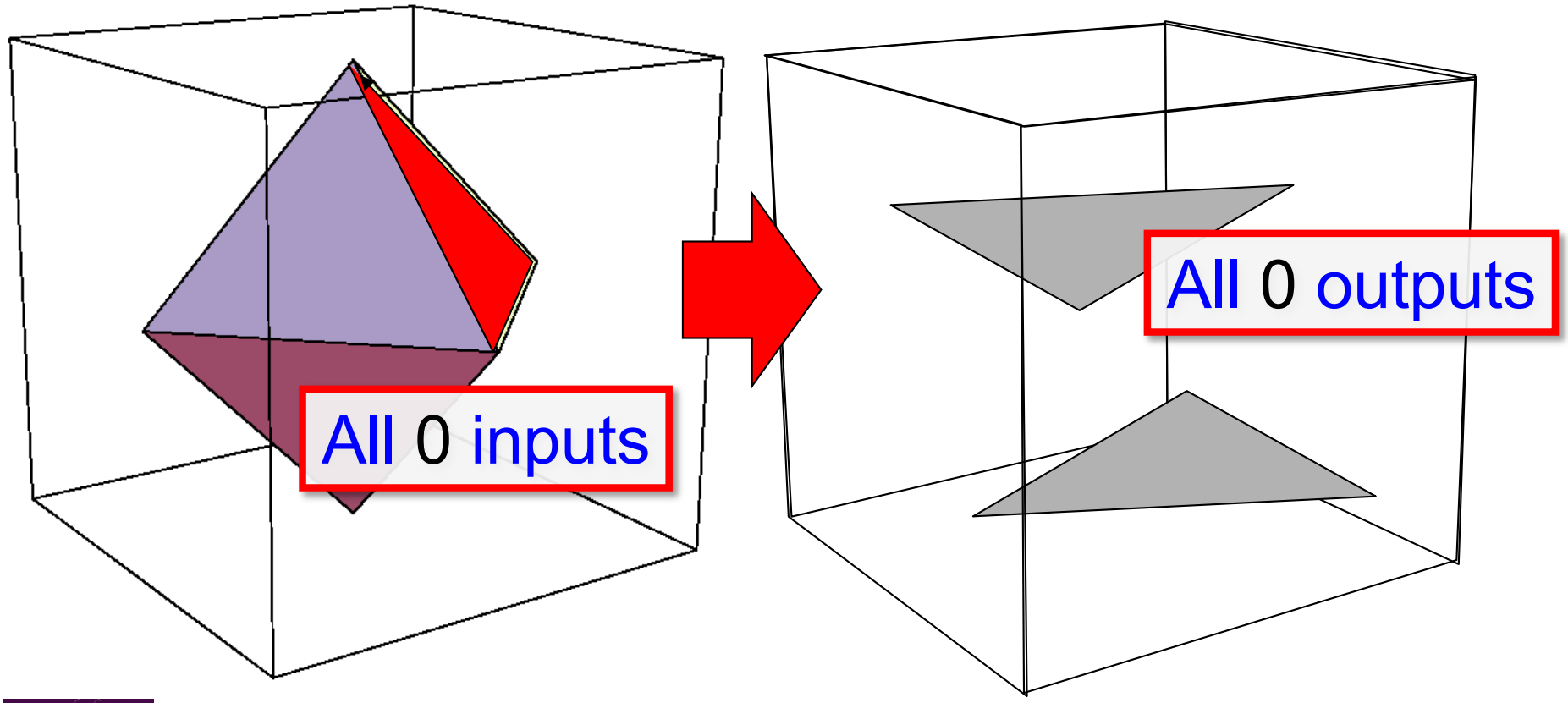
All possible final states

Output values all 0 or all 1

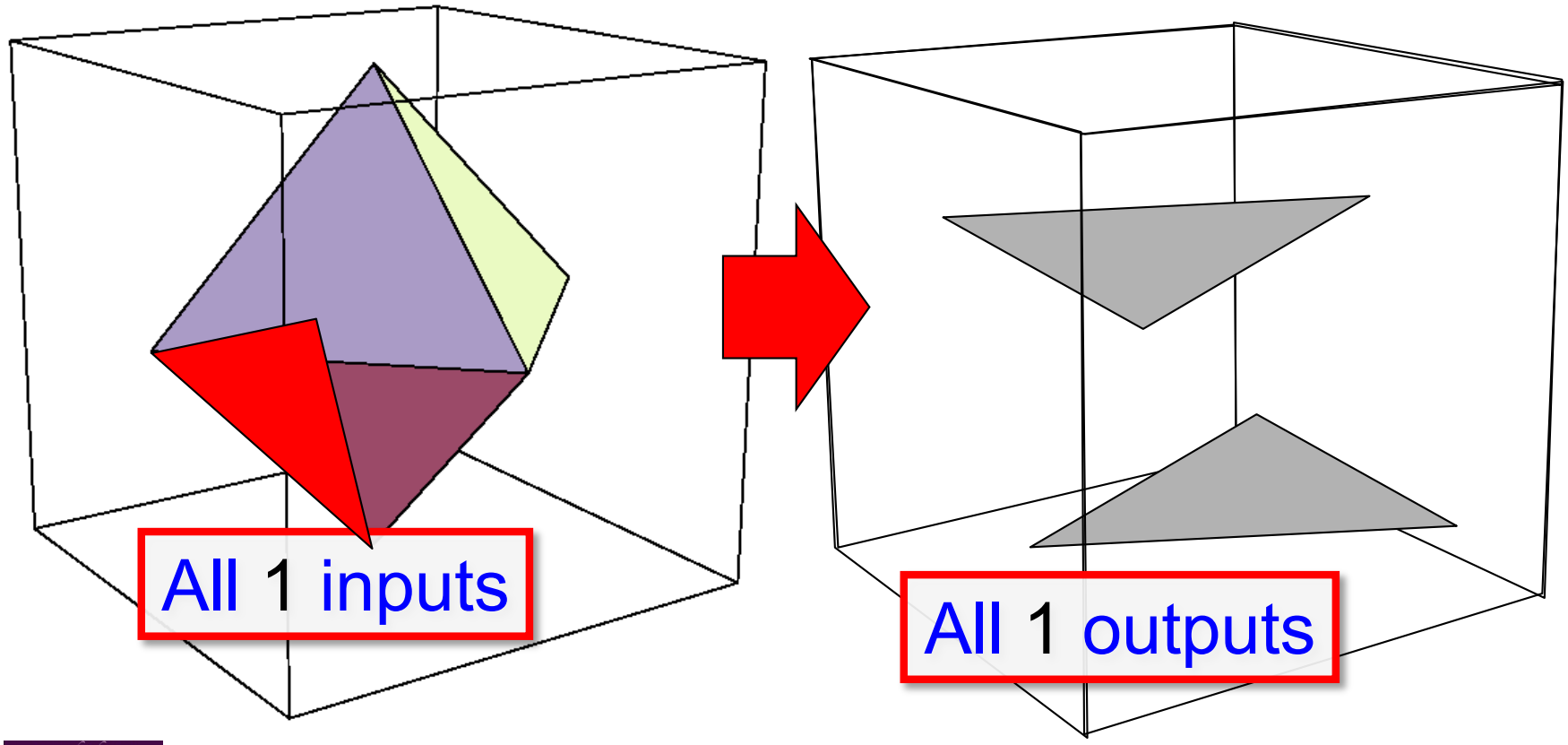
Two disconnected simplexes

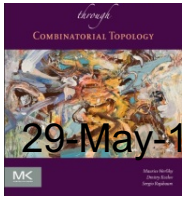


Carrier Map for Consensus

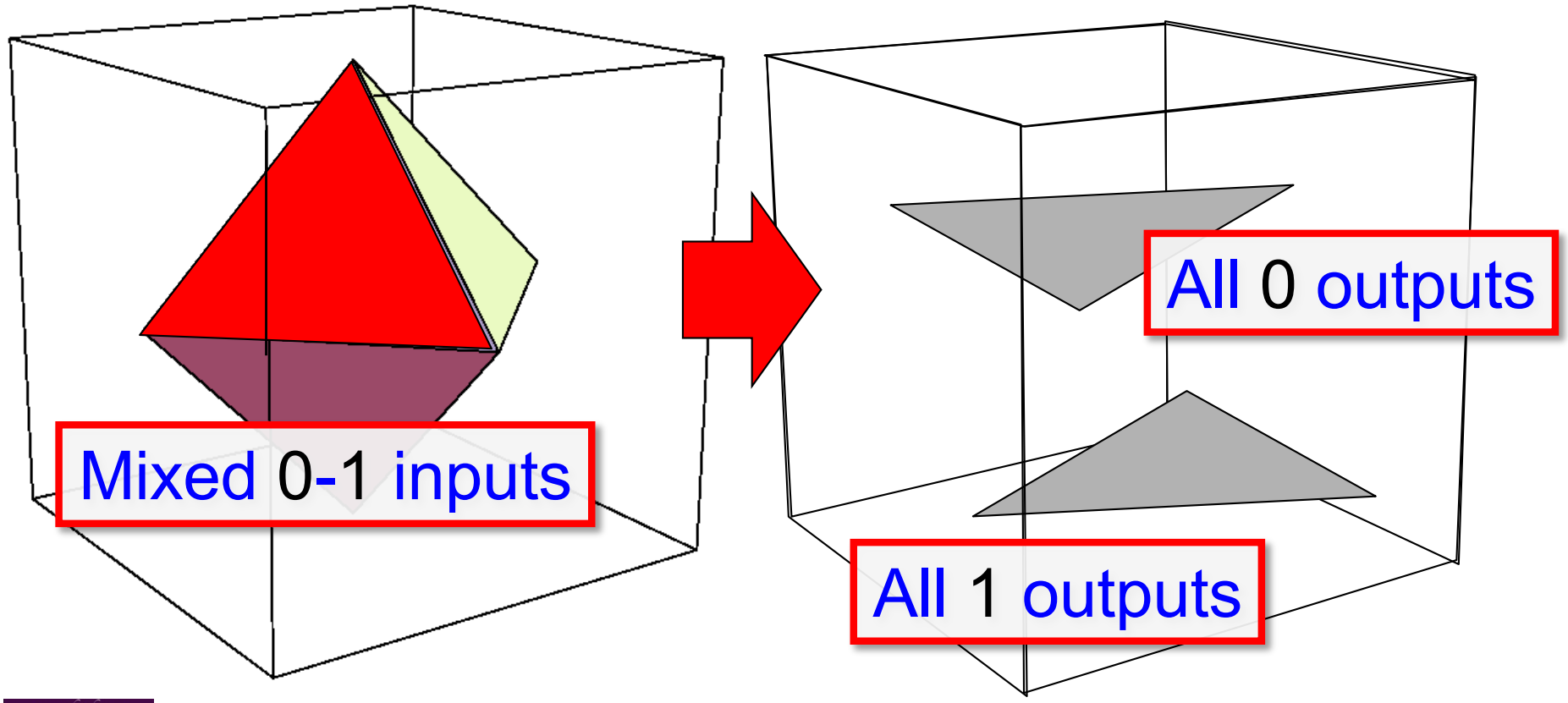


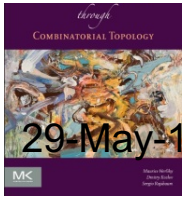
Carrier Map for Consensus





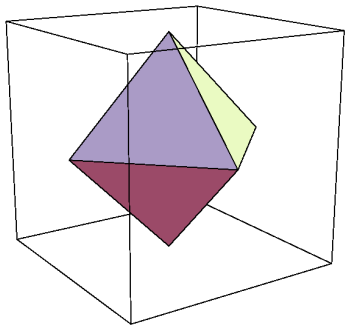
Carrier Map for Consensus



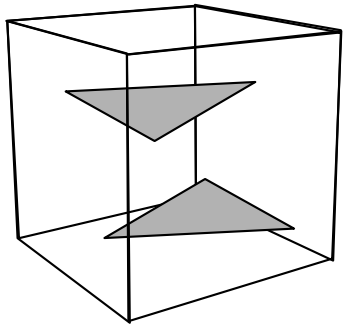


Task Specification

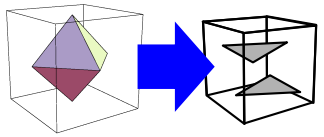
$(\mathcal{I}, \mathcal{O}, \Delta)$



Input complex

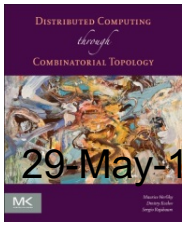


Output complex

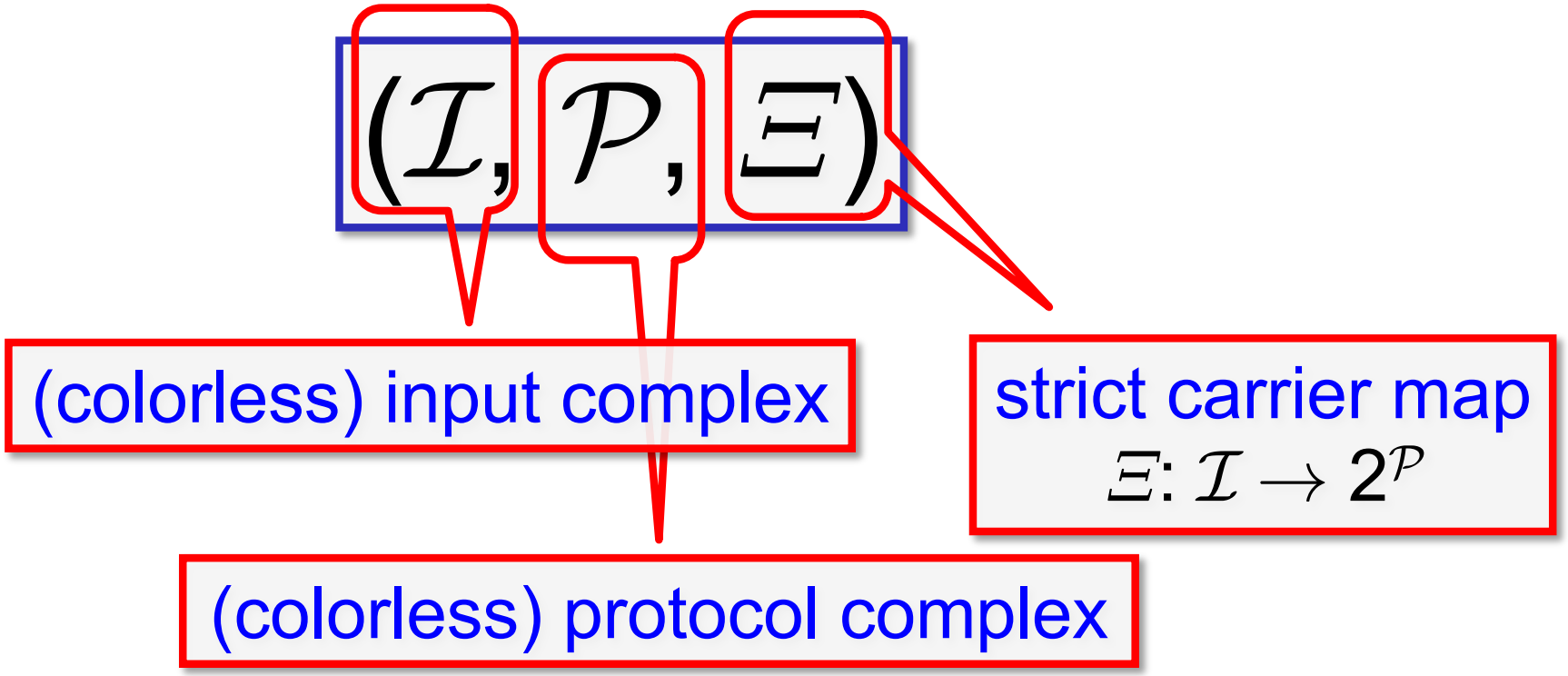


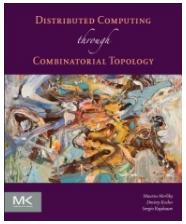
Carrier map

$\Delta: \mathcal{I} \rightarrow 2^{\mathcal{O}}$



Colorless Tasks





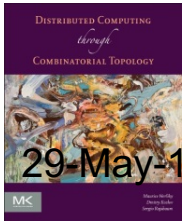
Protocol Complex

Vertex: process name, view

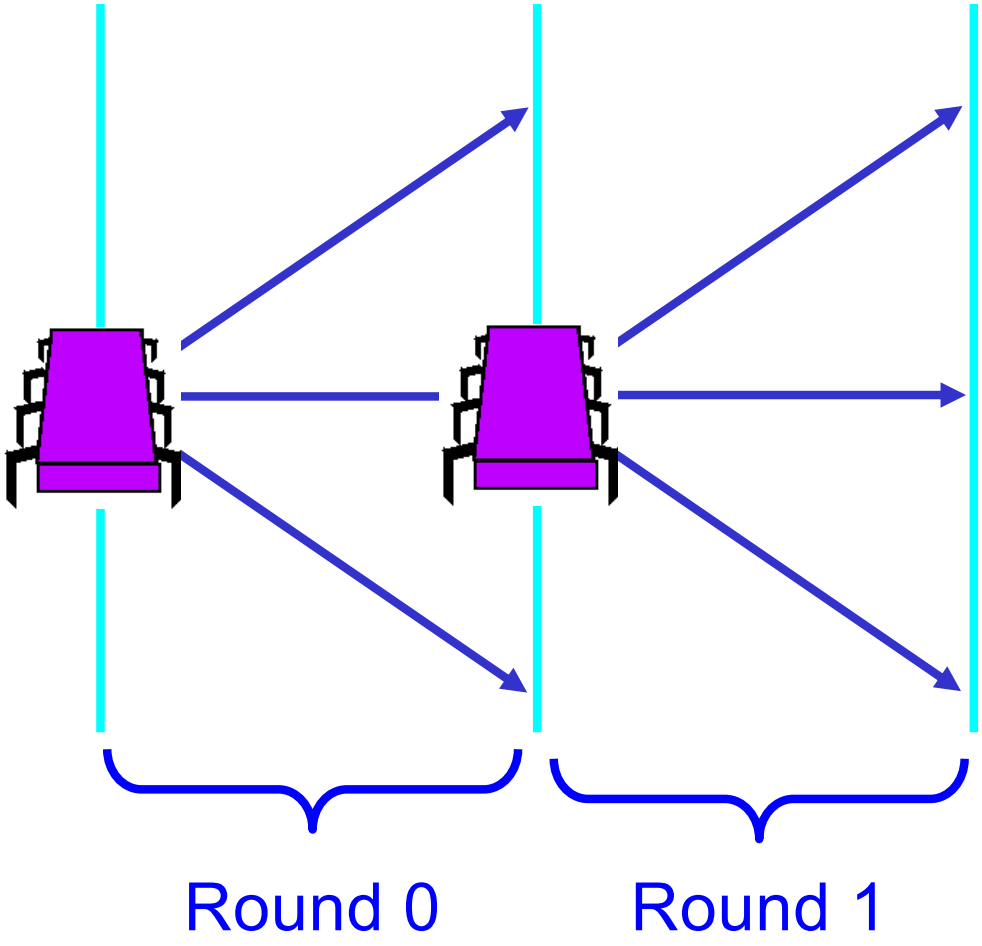
all values read and written

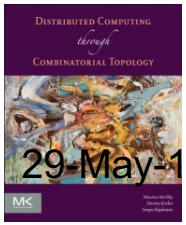
Simplex: compatible set of views

Each execution defines a simplex

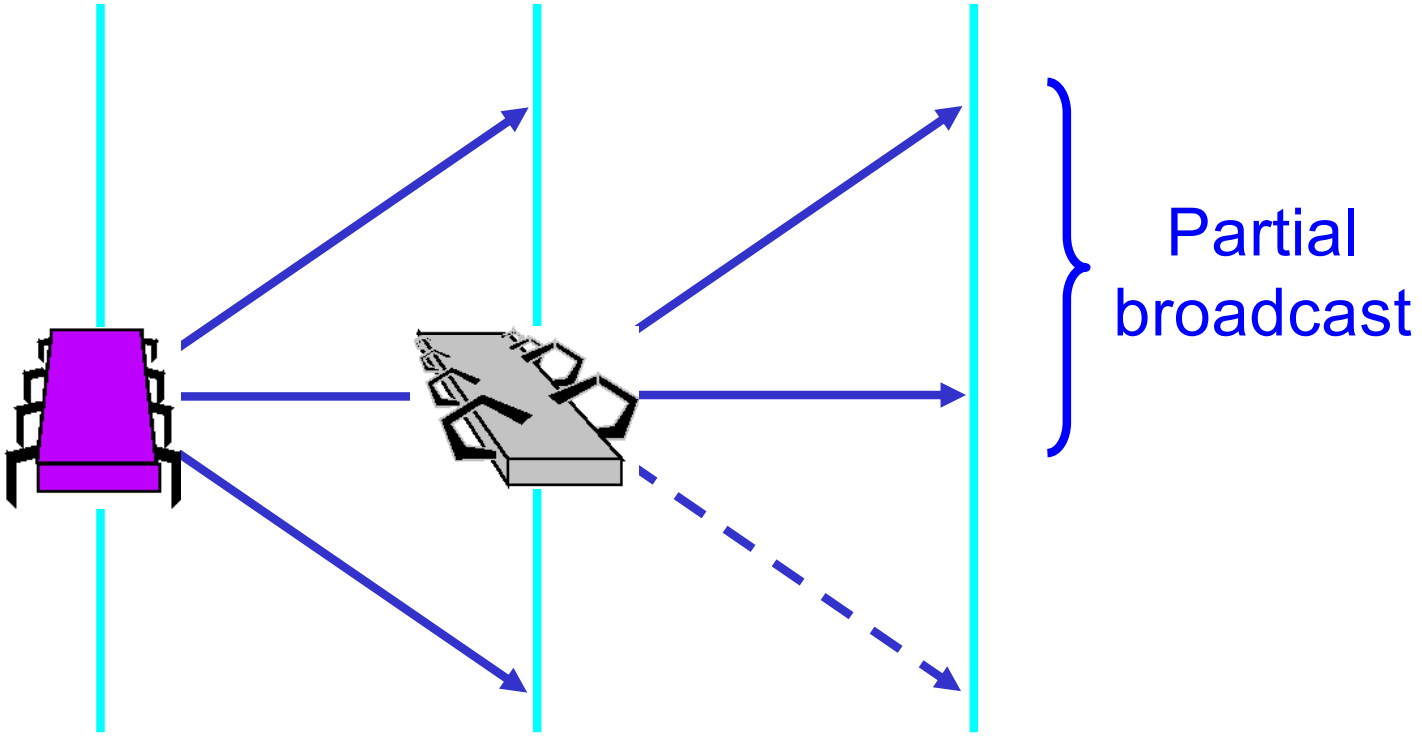


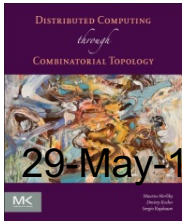
Example: Synchronous Message-Passing



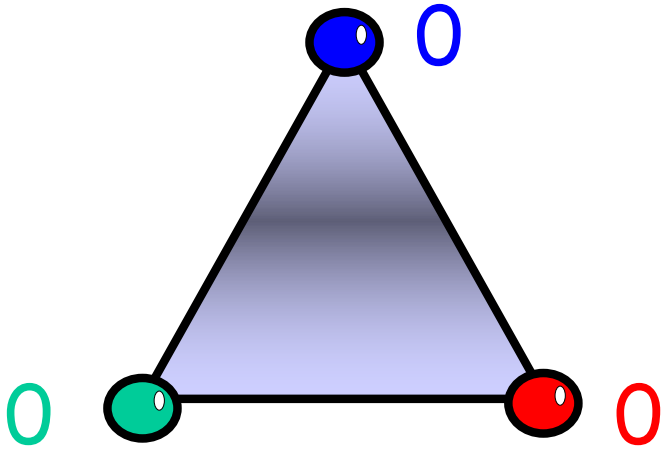


Failures: Fail-Stop





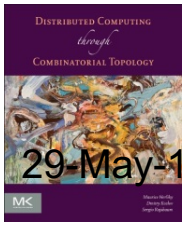
Single Input: Round Zero



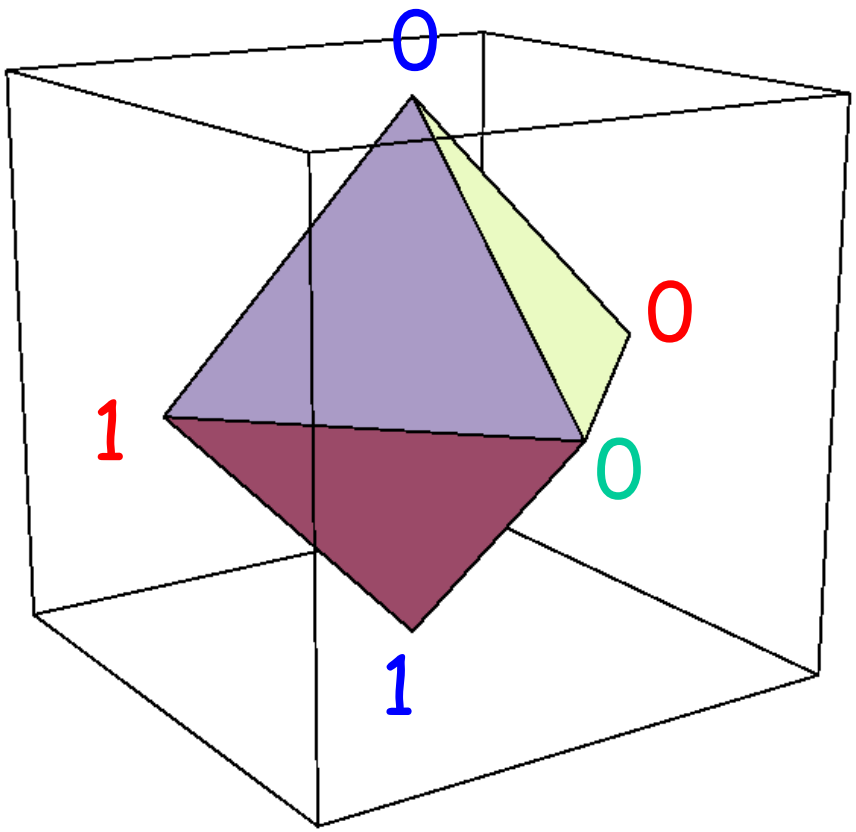
No messages sent

View is input value

Same as input simplex



Round Zero Protocol Complex

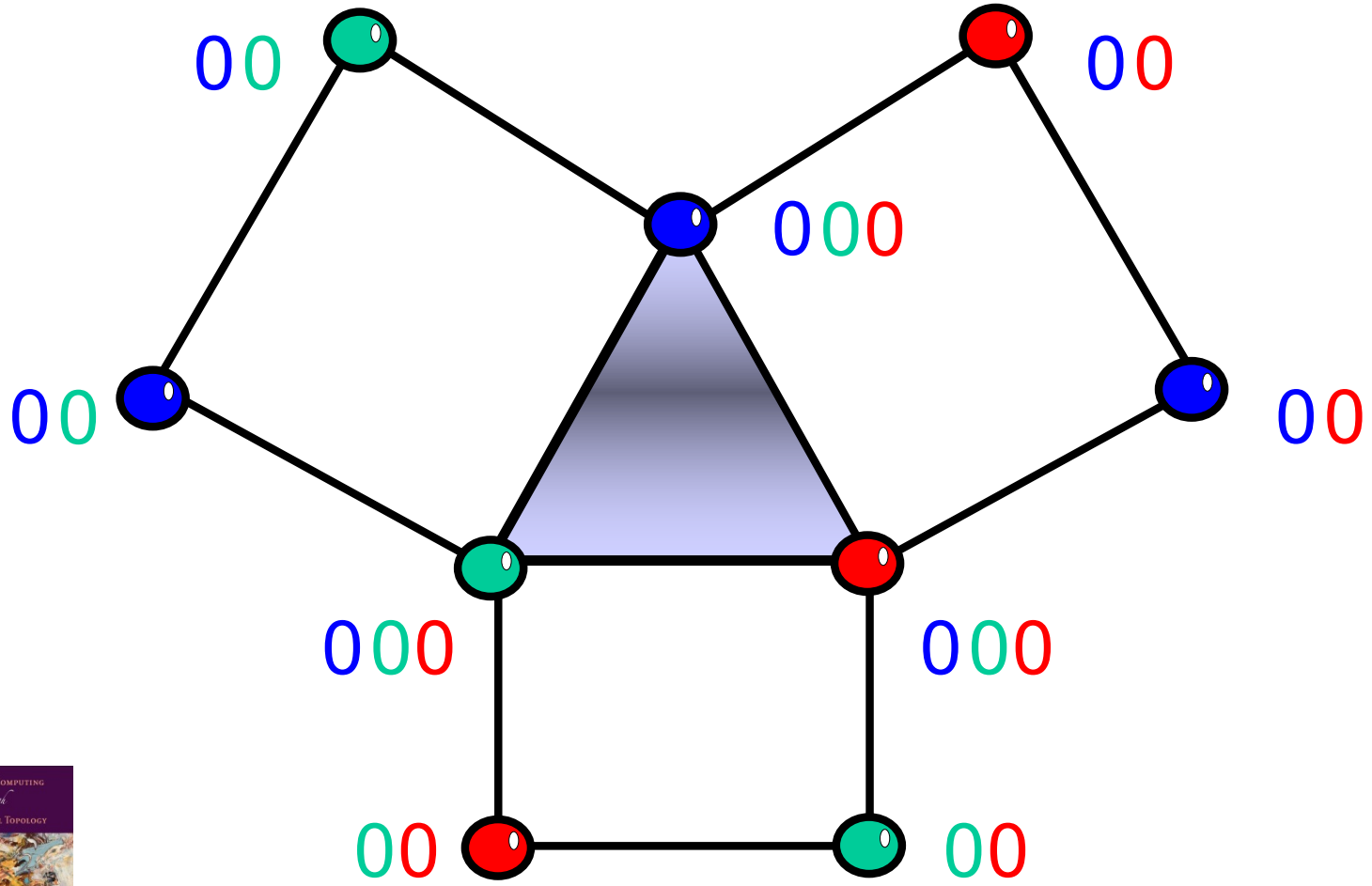


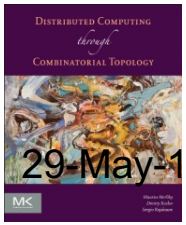
No messages sent

View is input value

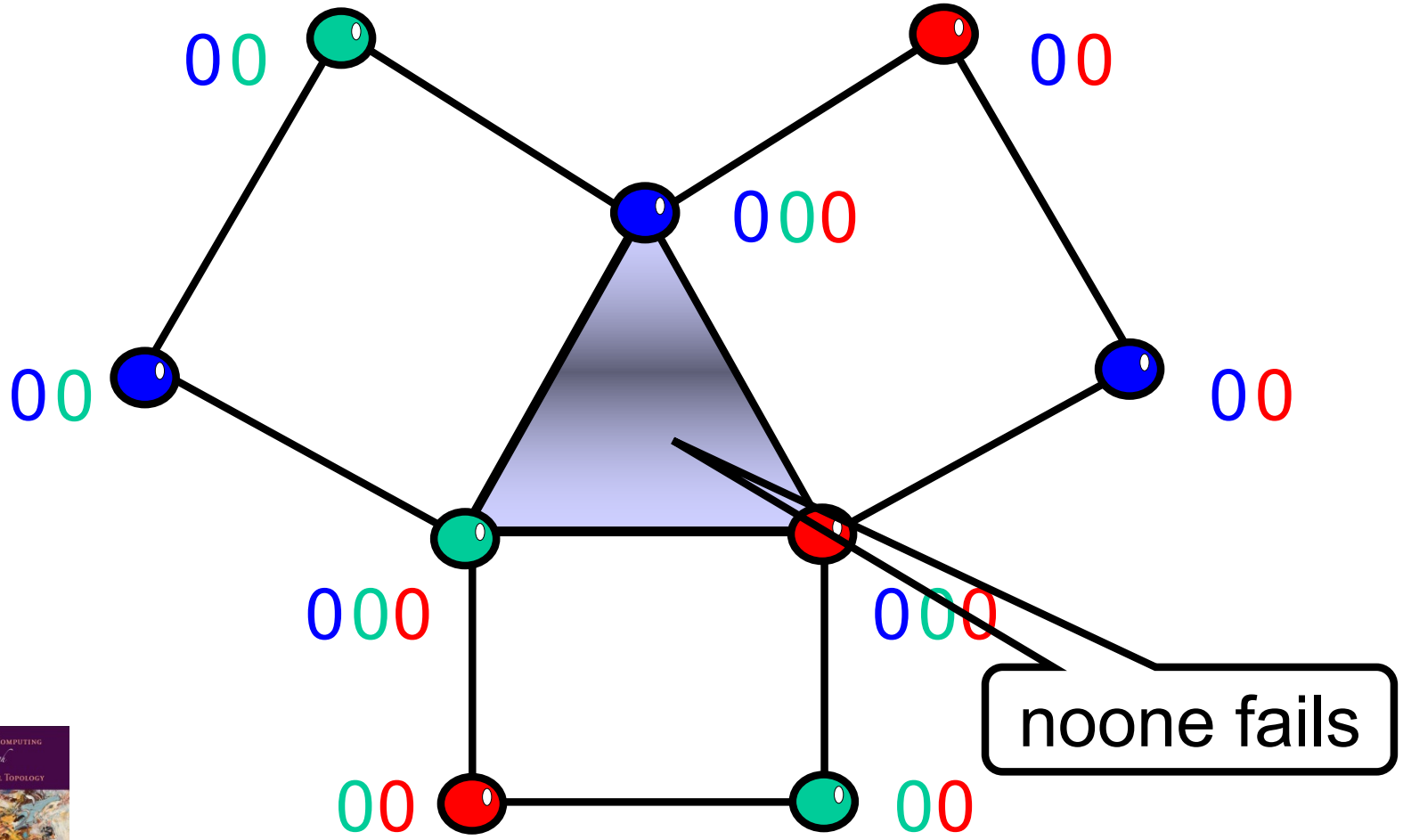
Same as input complex

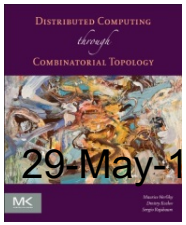
Single Input: Round One



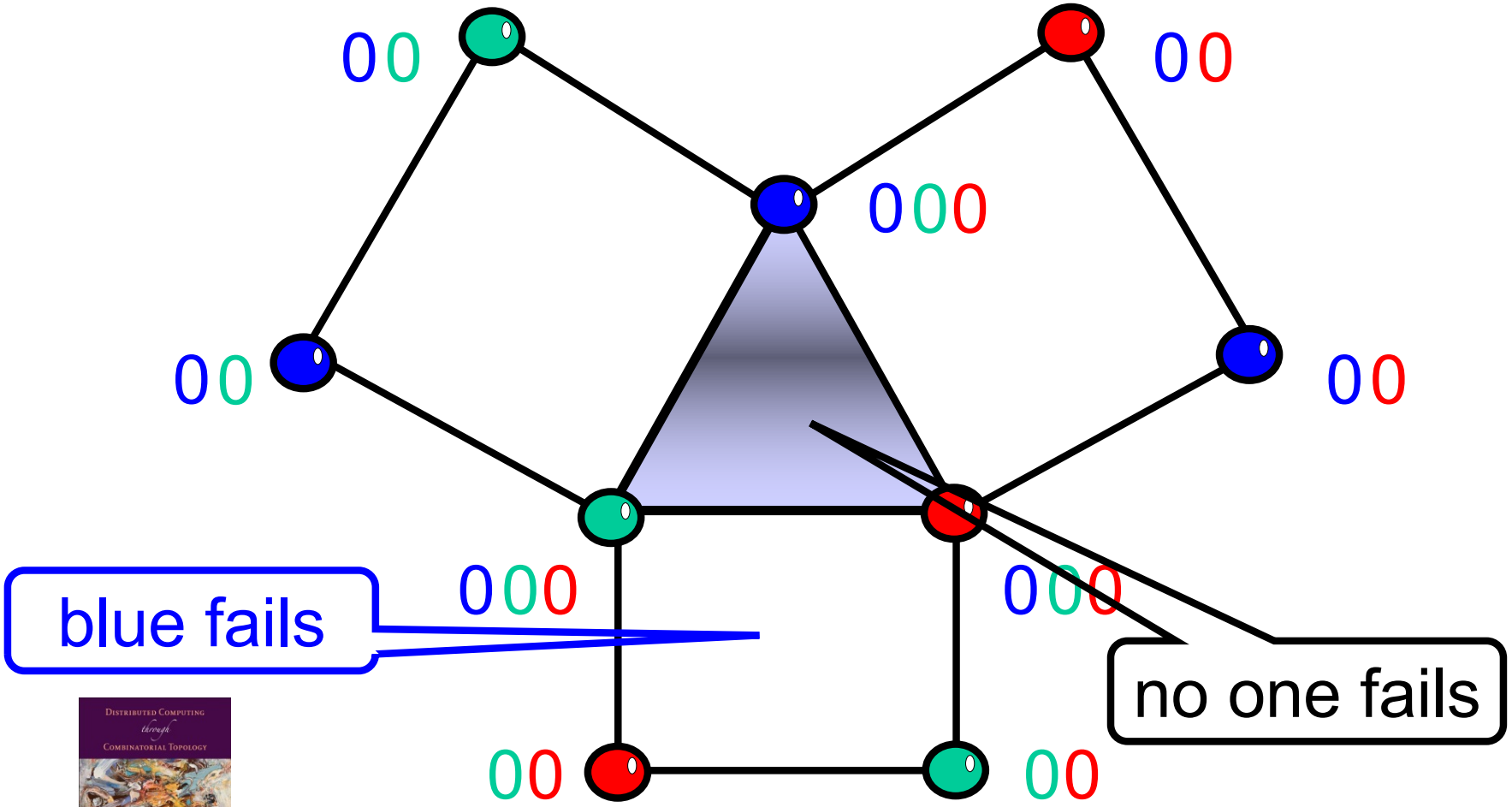


Single Input: Round One

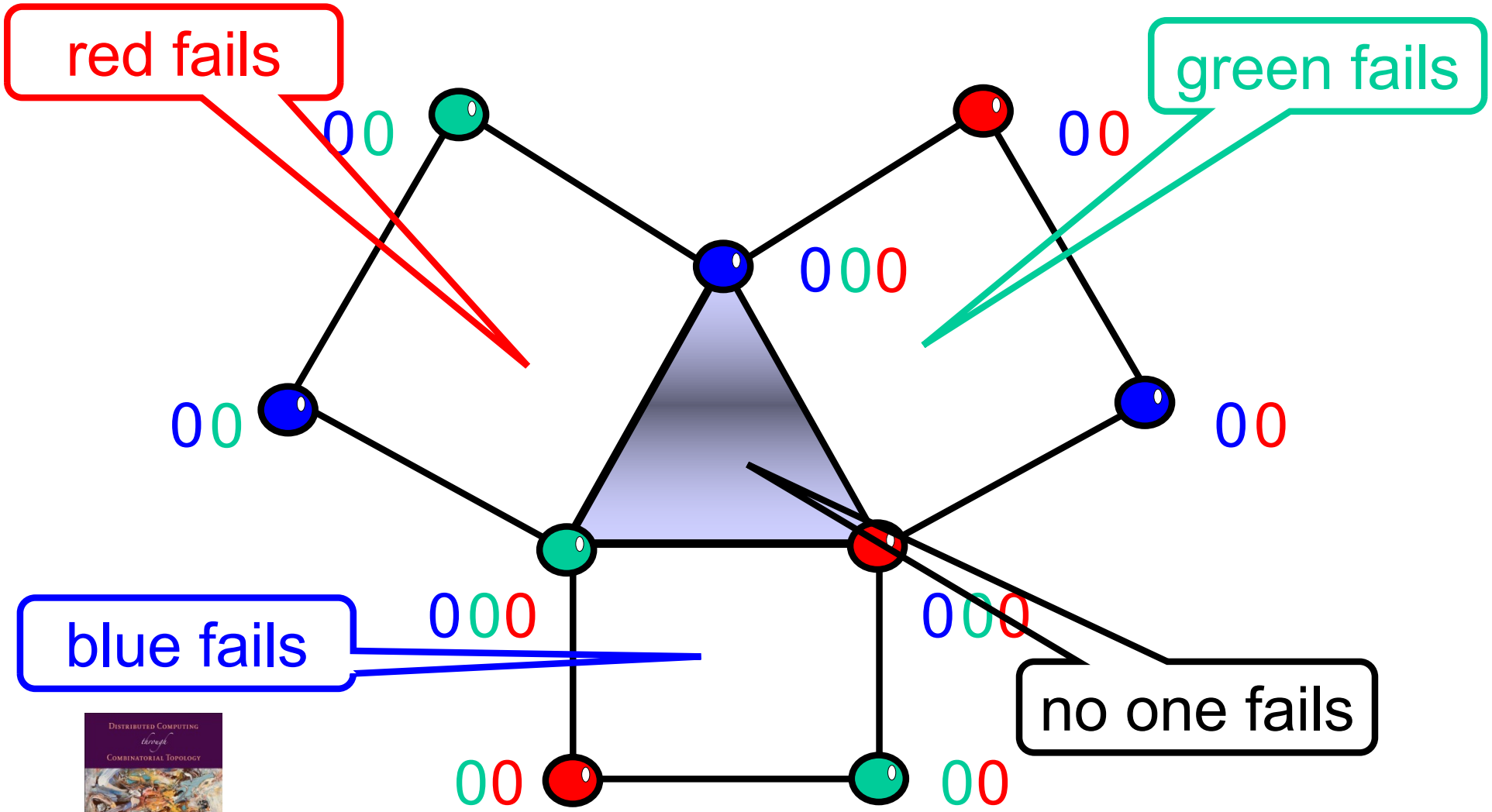




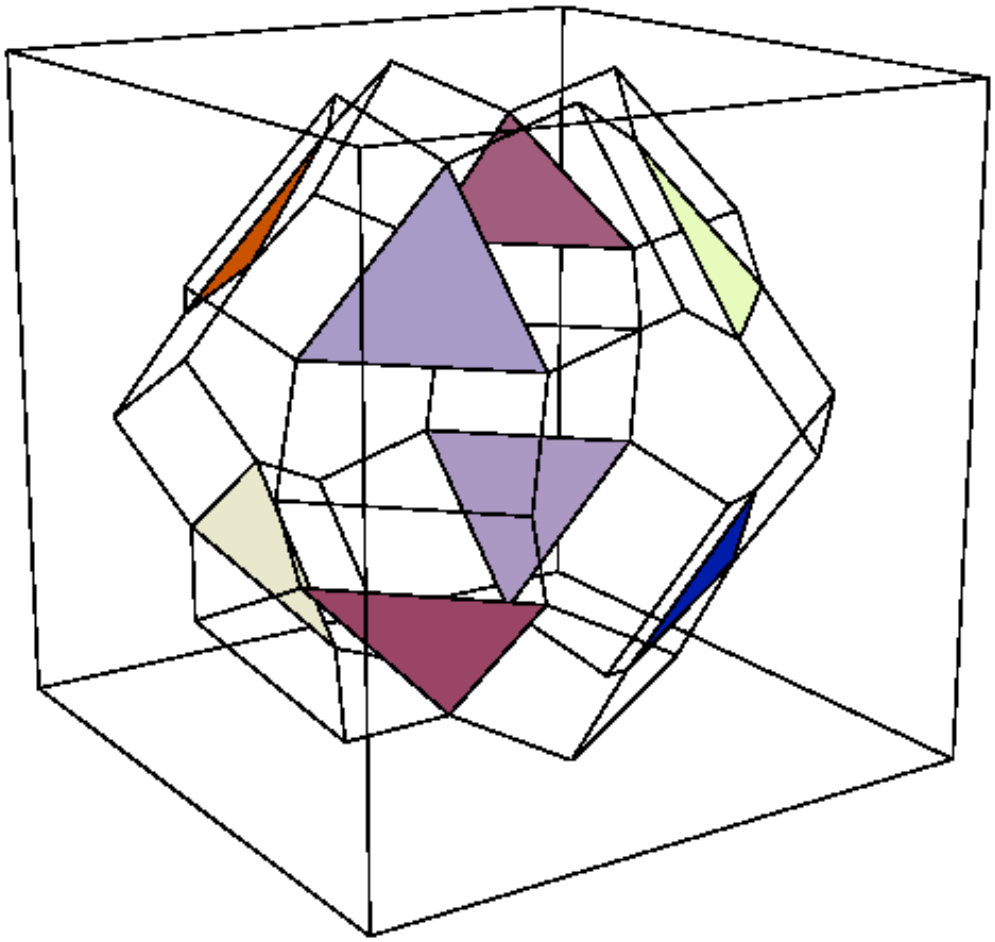
Single Input: Round One

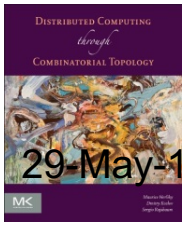


Single Input: Round One

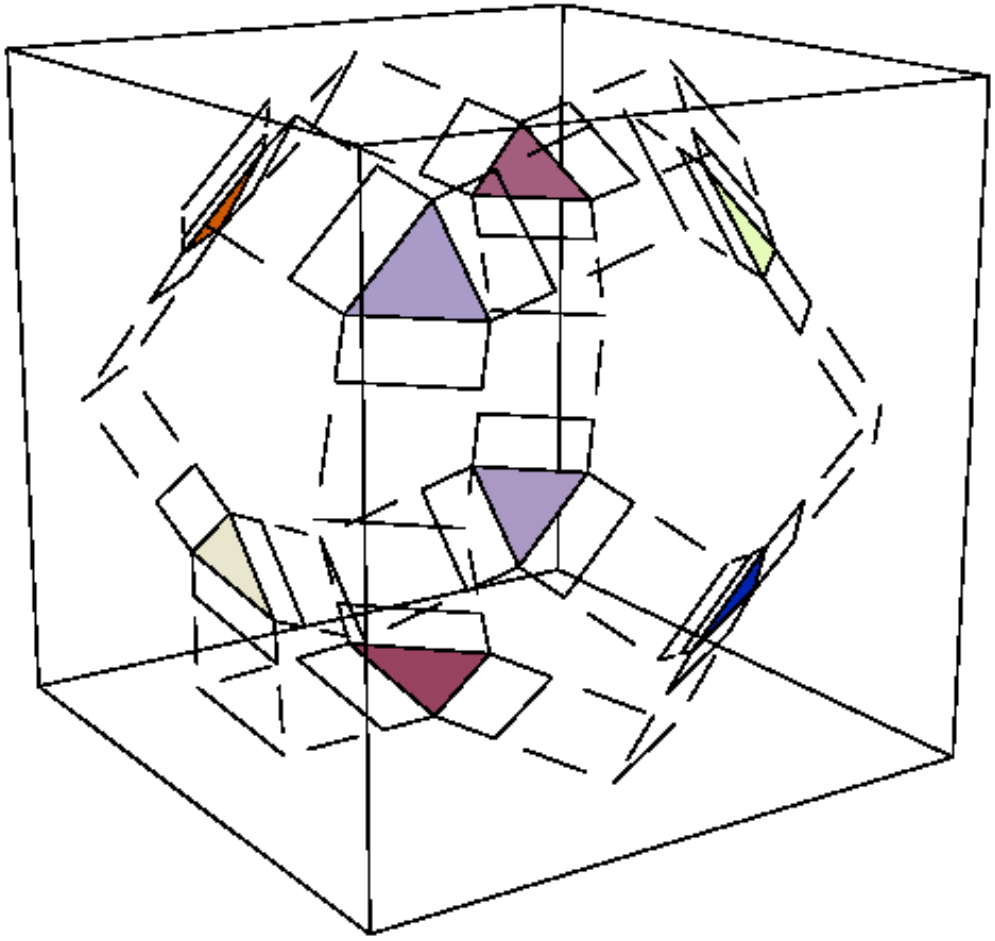


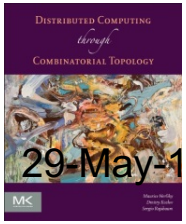
Protocol Complex: Round One



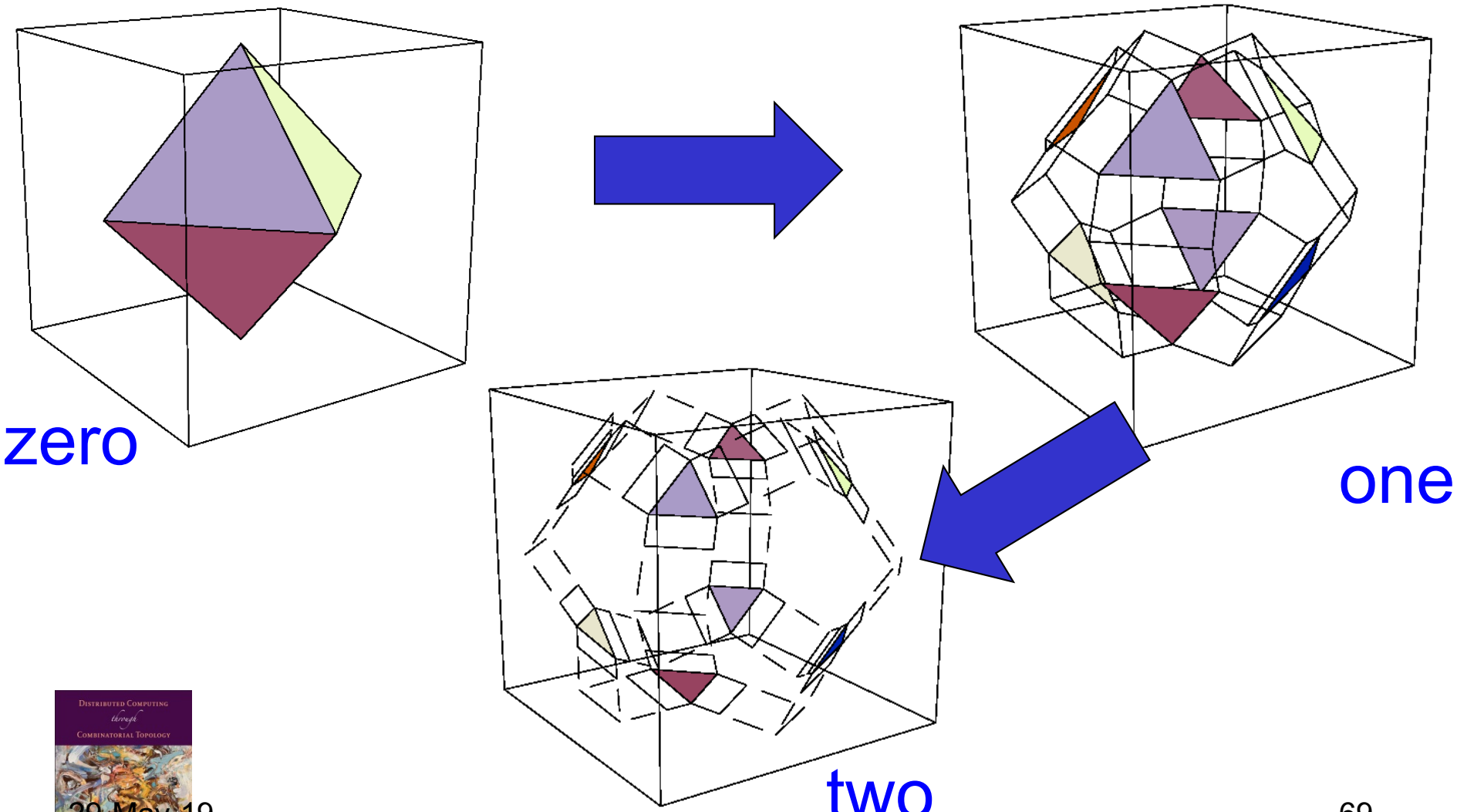


Protocol Complex: Round Two

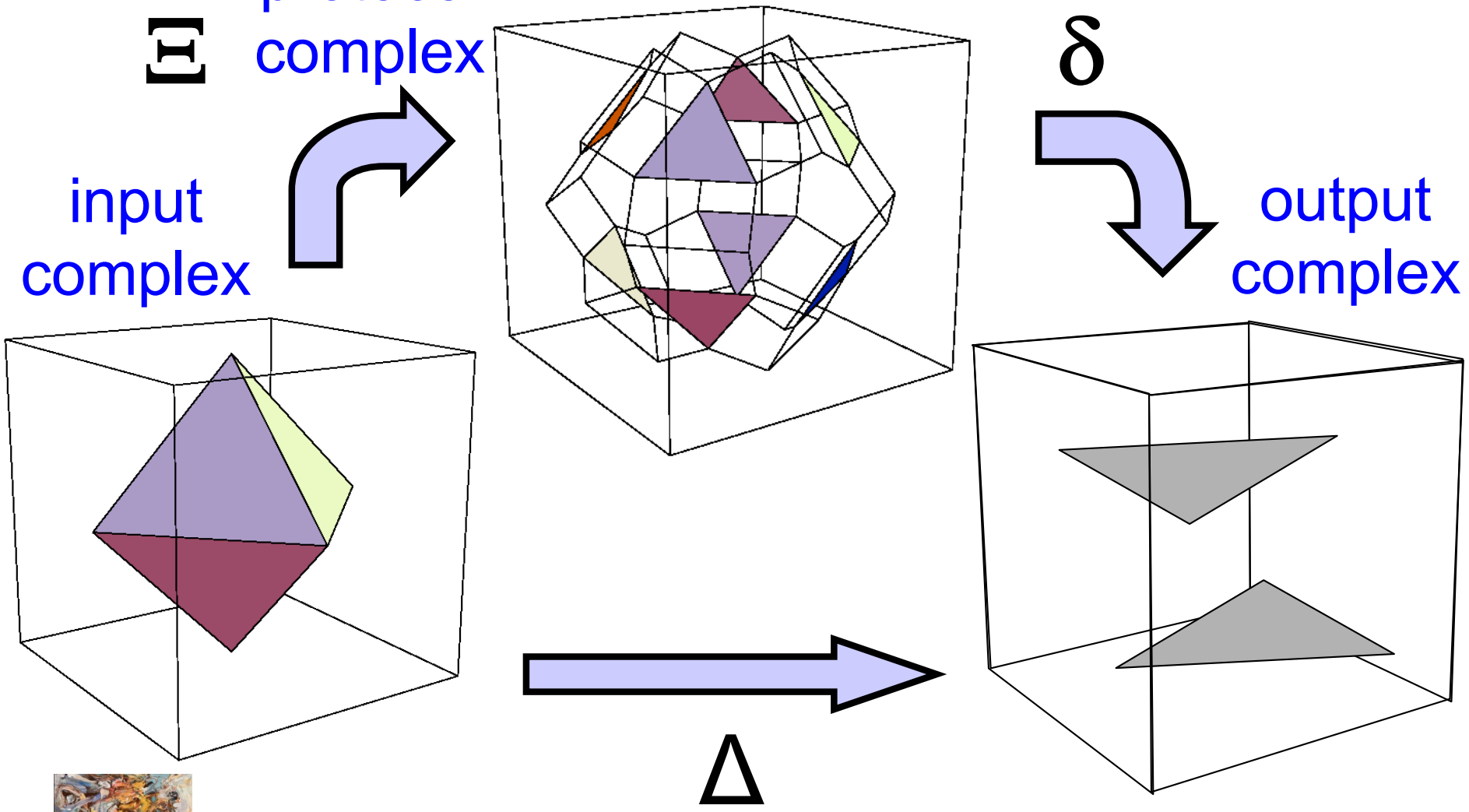




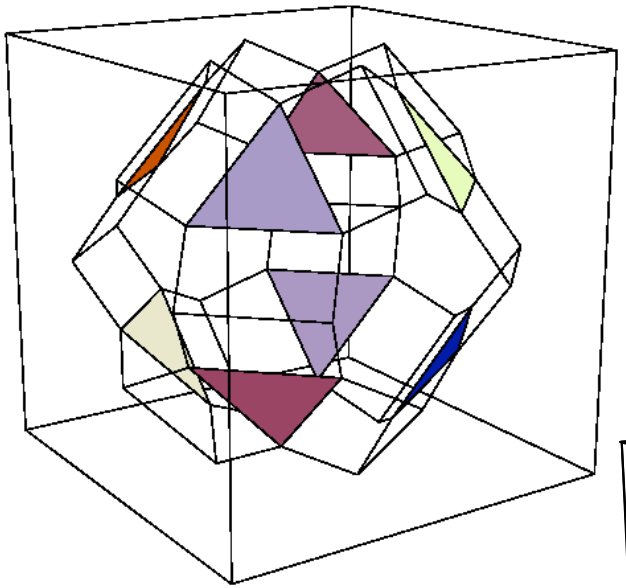
Protocol Complex Evolution



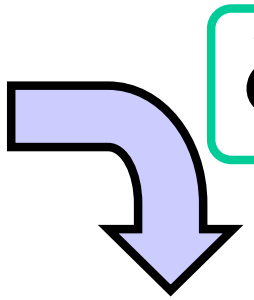
Summary



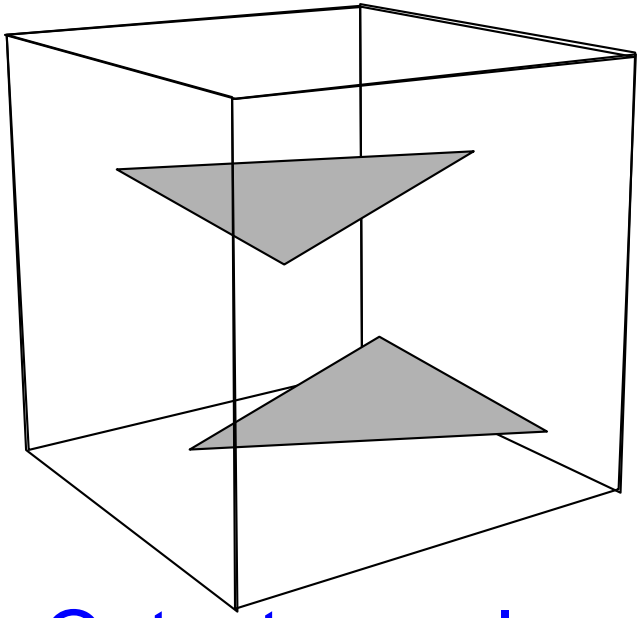
Decision Map



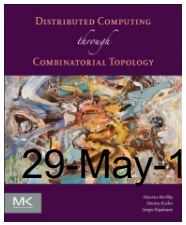
Protocol complex



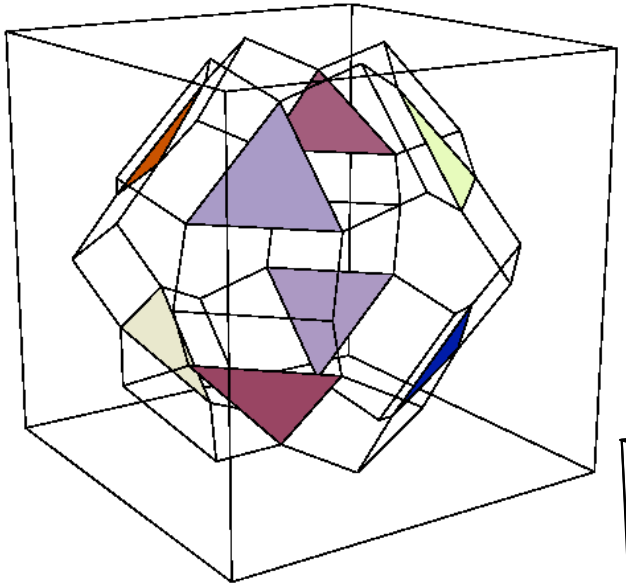
Simplicial map, sending simplexes to simplexes



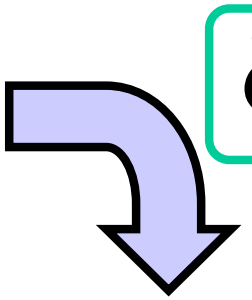
Output complex



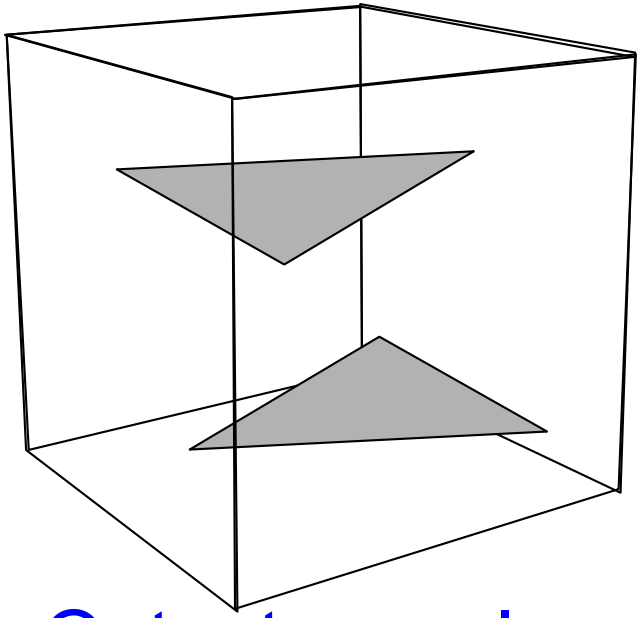
Strategy for Lower Bounds/Impossibility Results



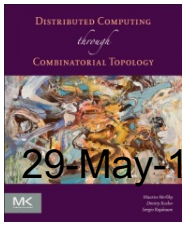
Protocol complex



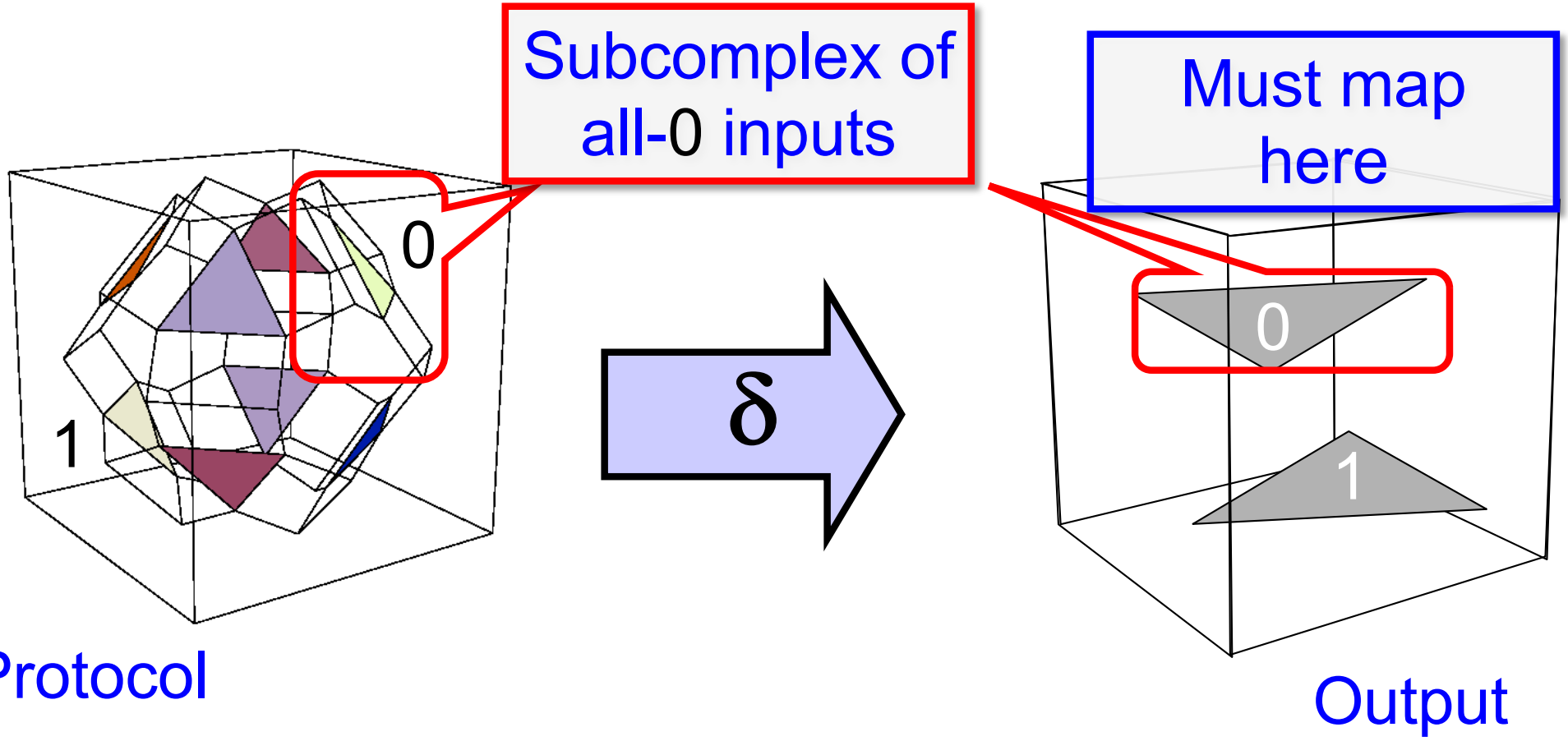
Find topological "obstruction" to this simplicial map

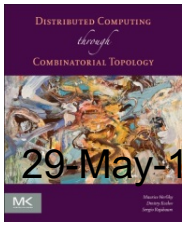


Output complex

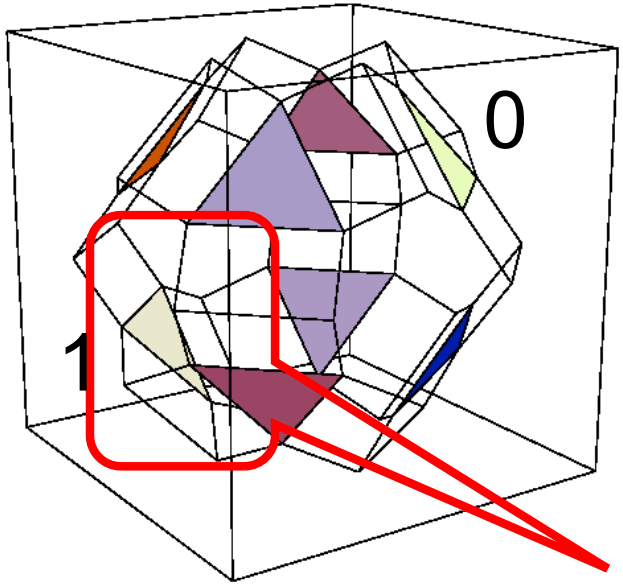


Consensus Example

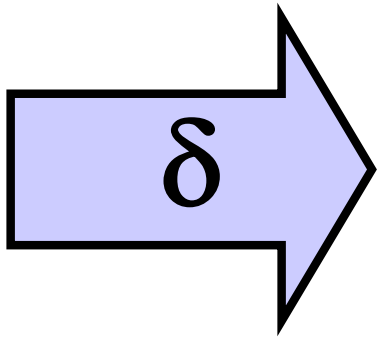


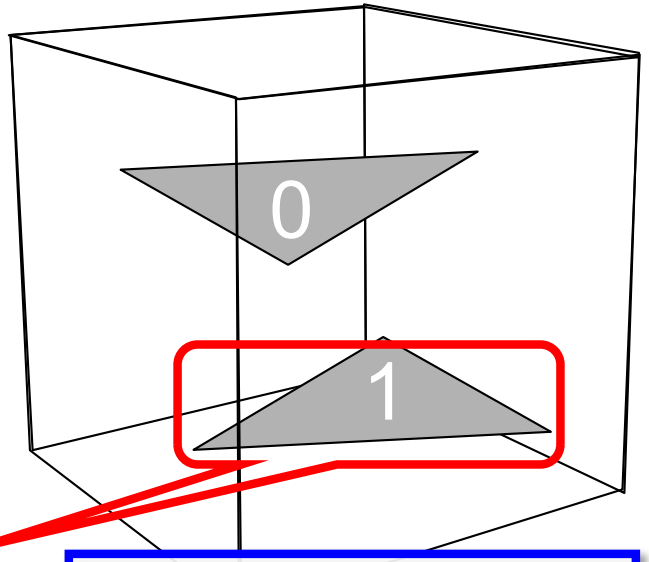


Consensus Example



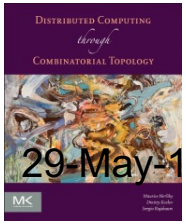
Protocol



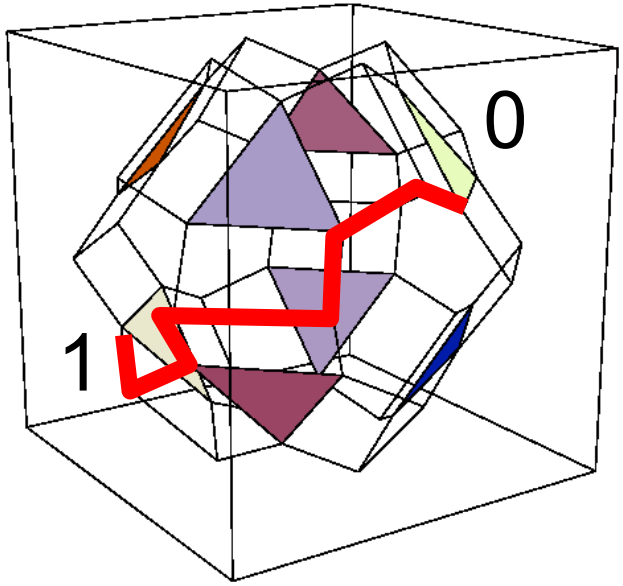


Subcomplex of all-1 inputs

Must map here



Consensus Example



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

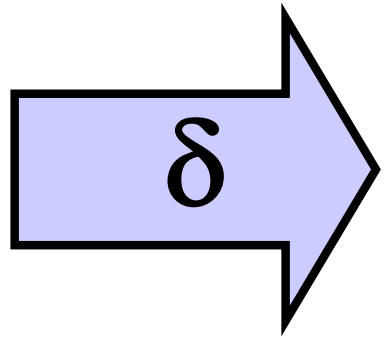
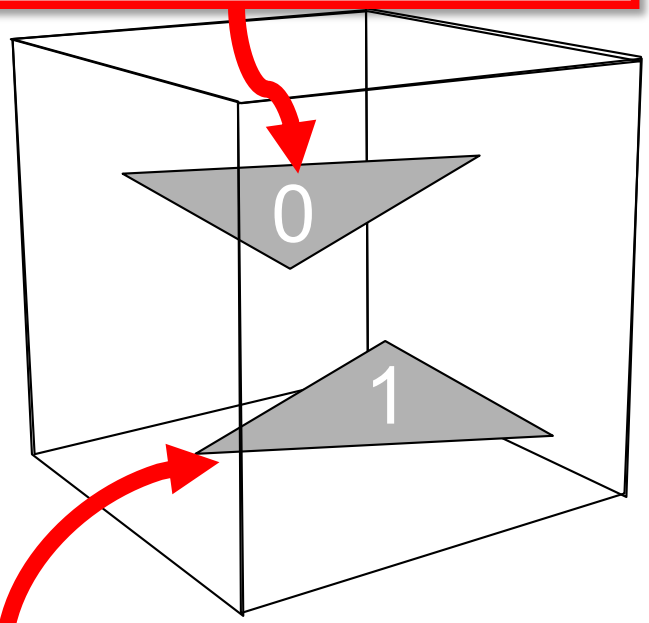


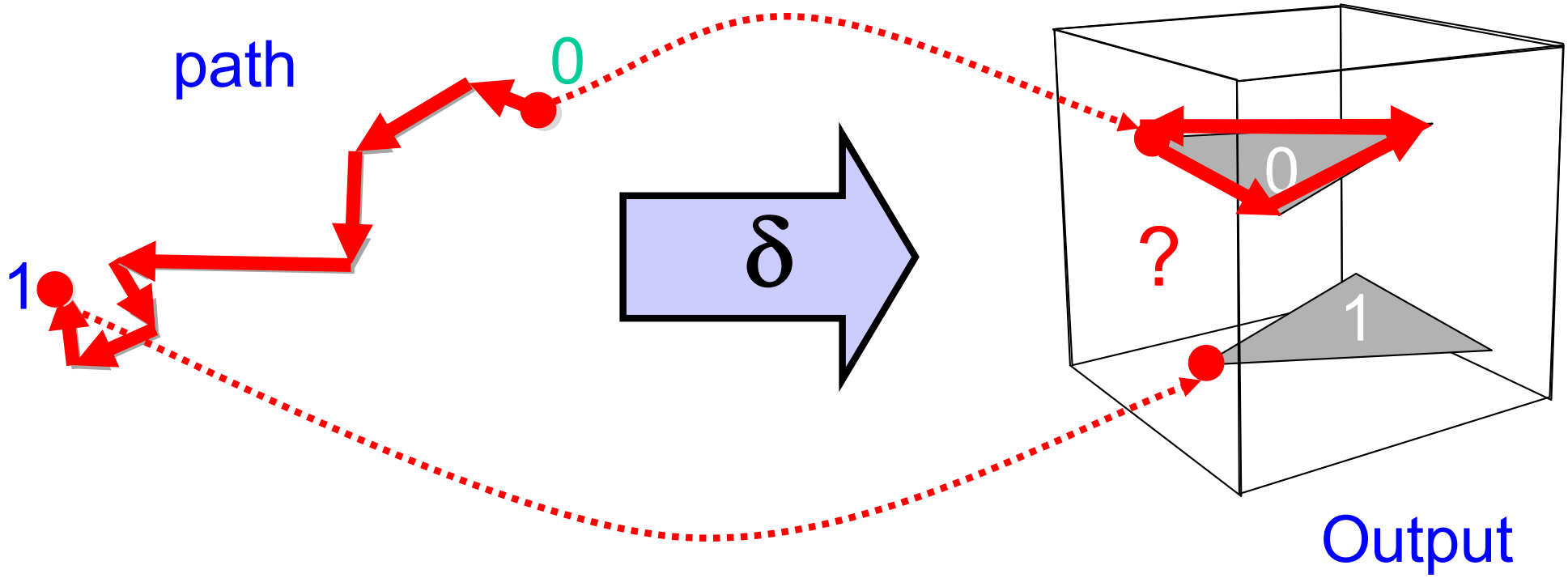
Image under δ must start here ..

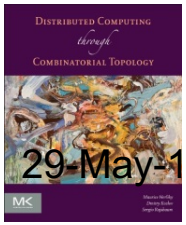


and end here

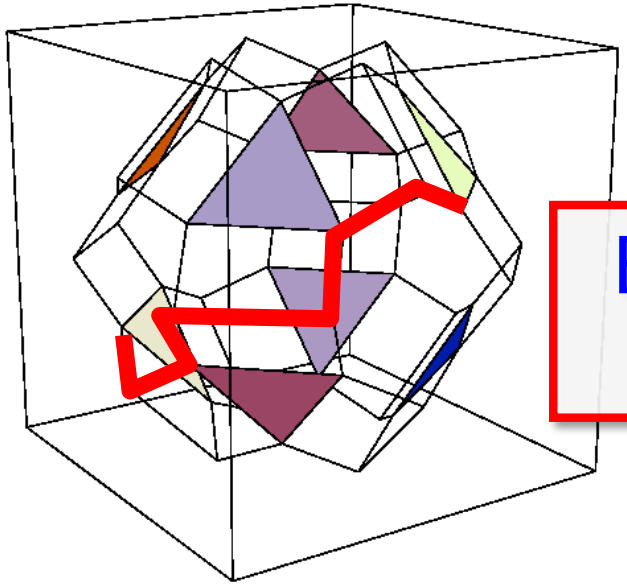
Output

Consensus Example





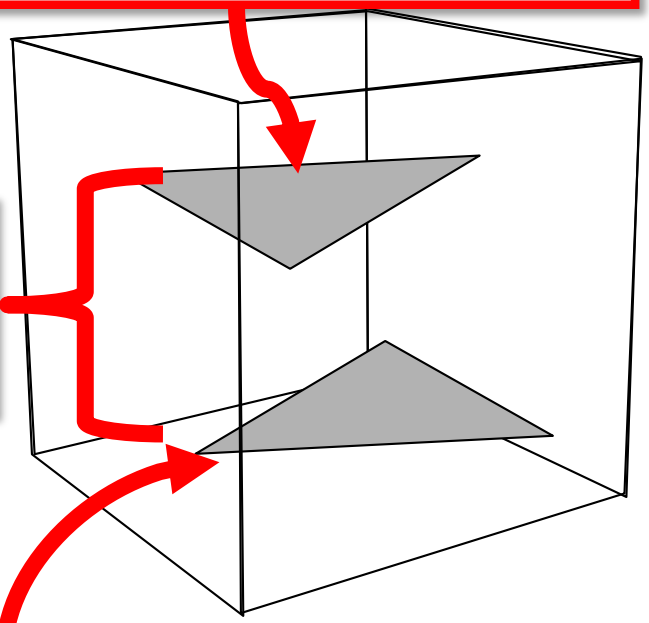
Consensus Example



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

But this "hole" is an obstruction

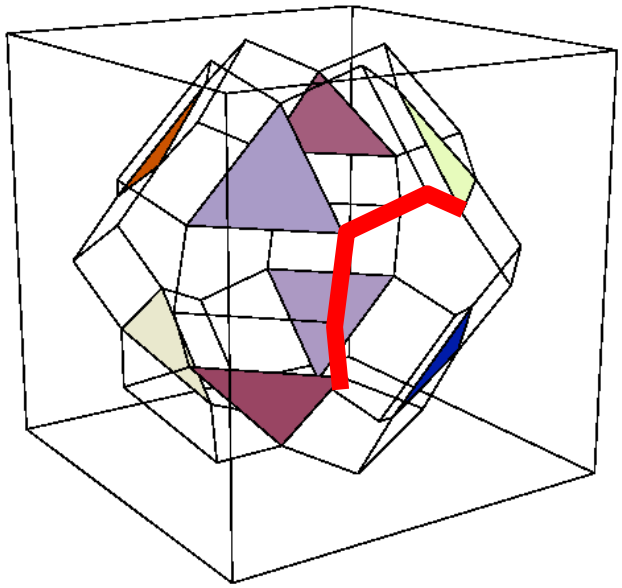
Image under δ must start here ..



and end here

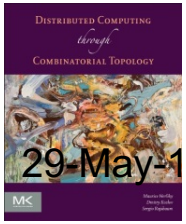
Output

Conjecture



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

Model-independent!



If Adversary keeps Protocol Complex path-connected …

Forever …

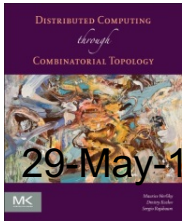
Consensus is ***impossible***

For *r* rounds …

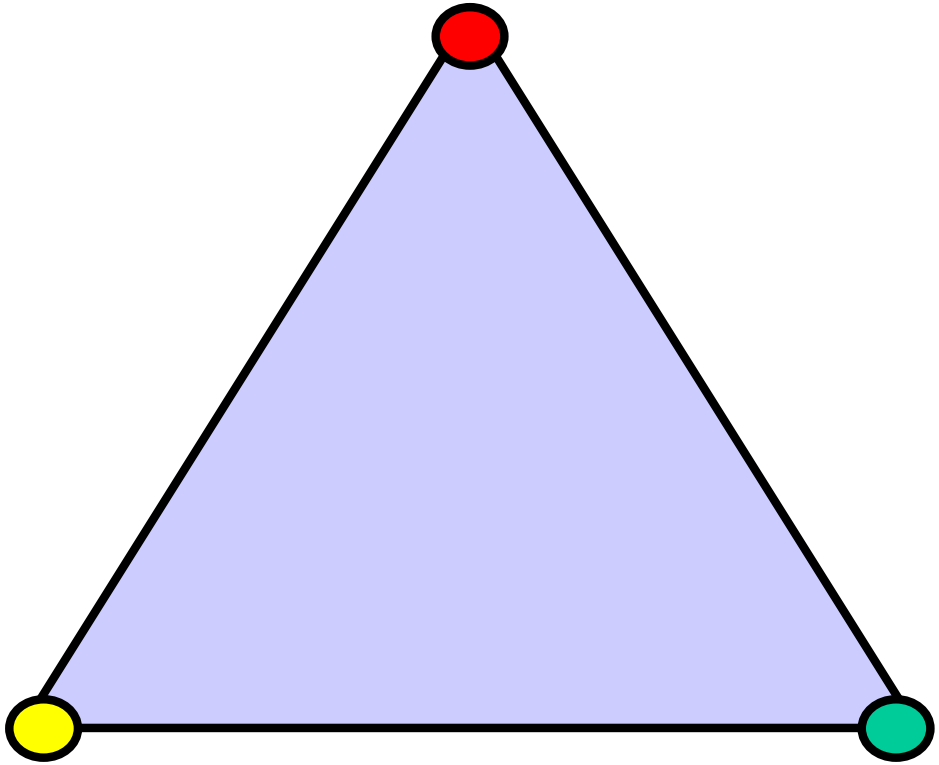
A ***round-complexity*** lower bound

For time *t* …

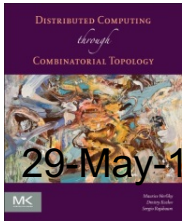
A ***time-complexity*** lower bound



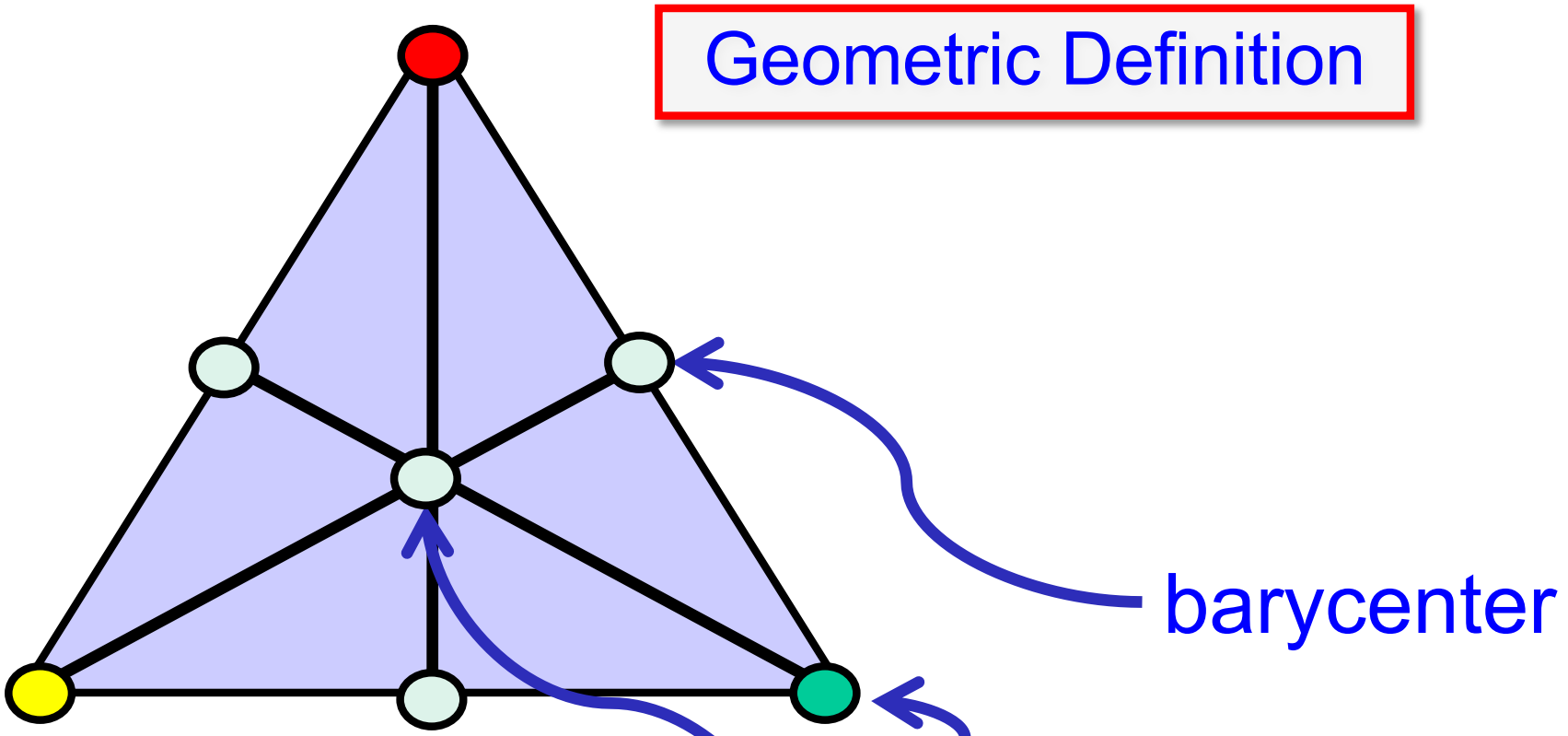
Barycentric Subdivision

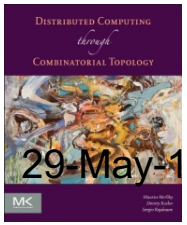


σ

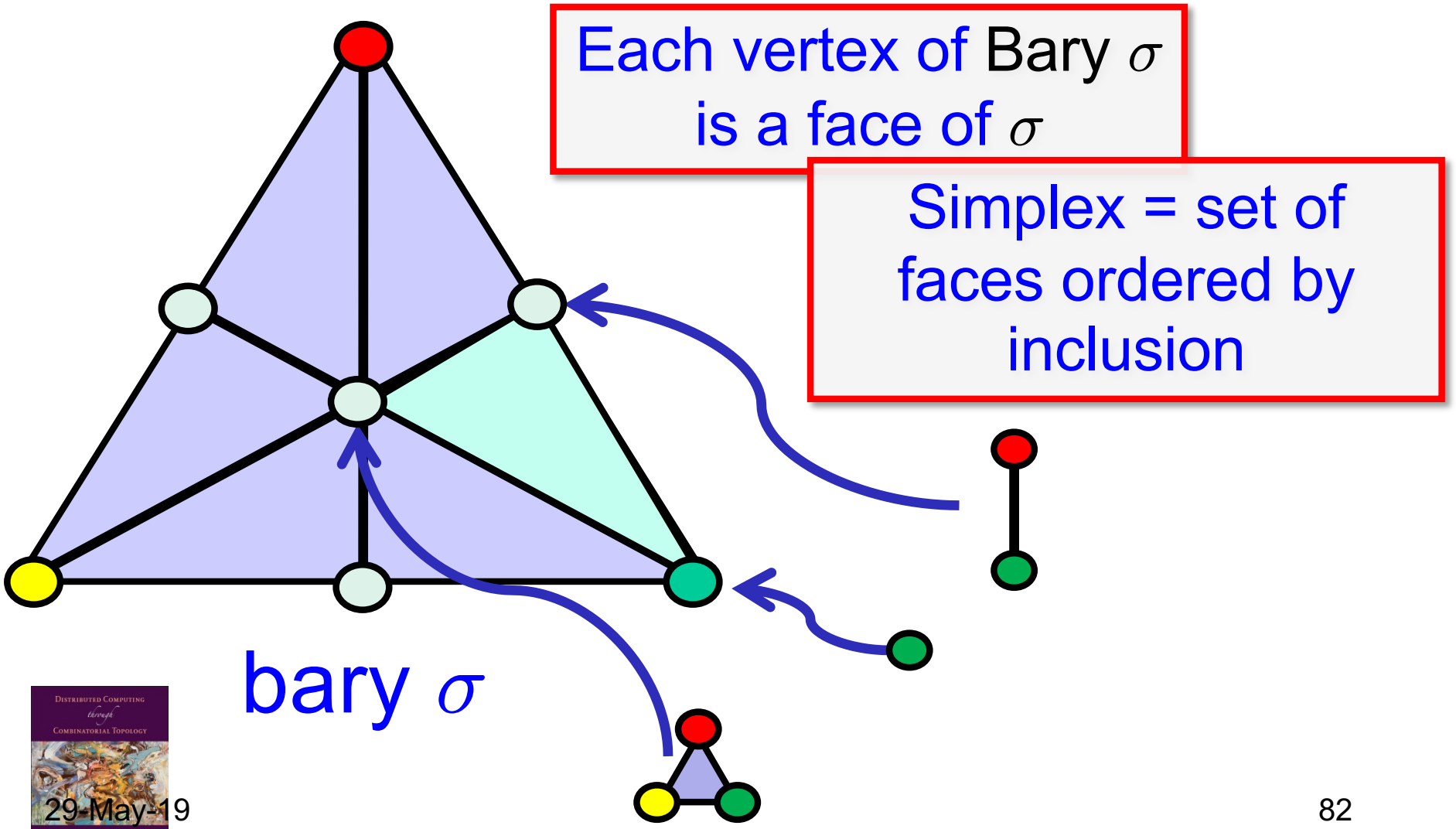


Barycentric Subdivision

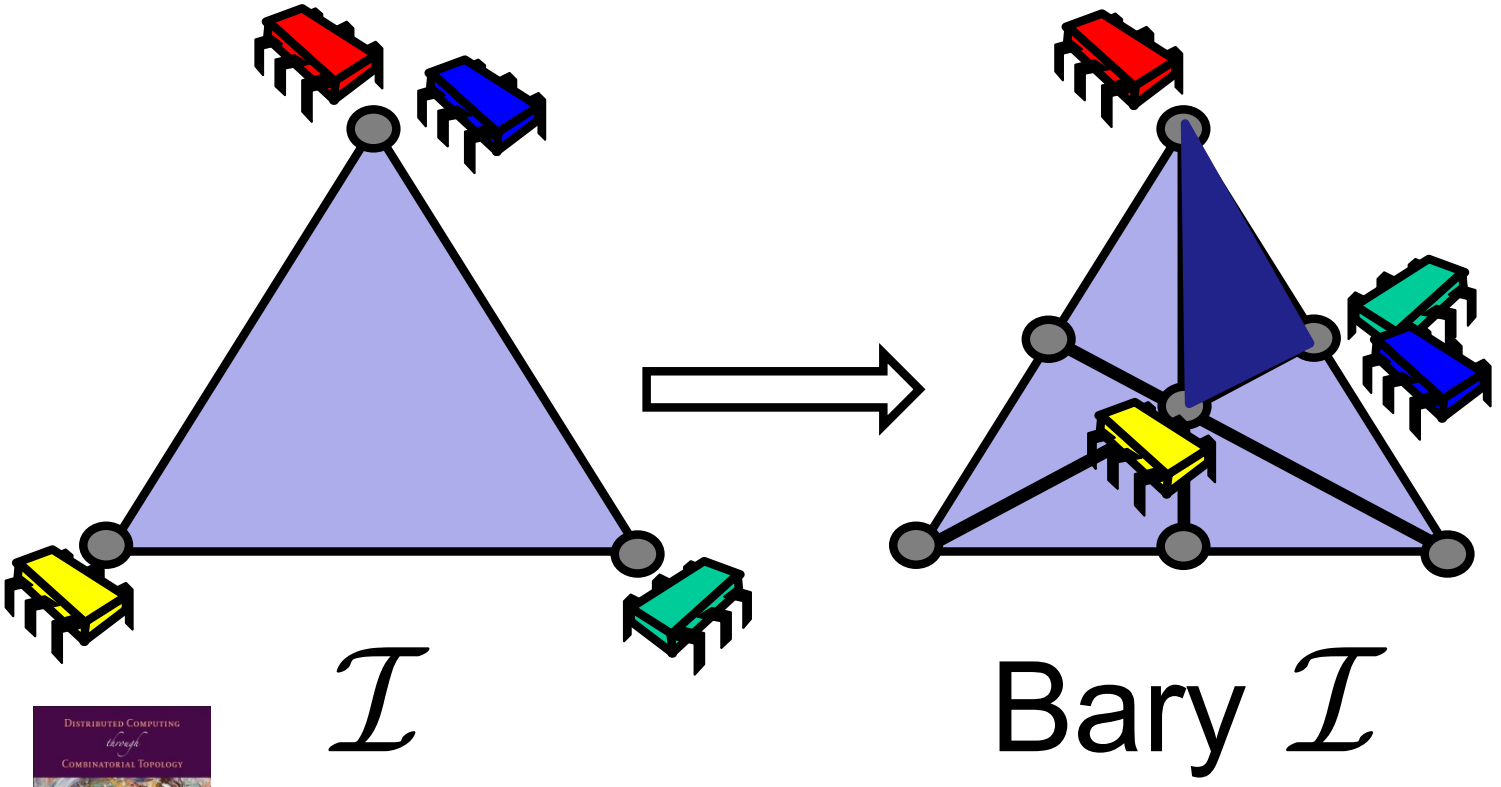


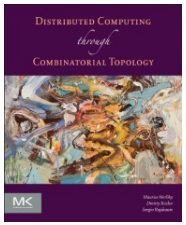


Barycentric Subdivision

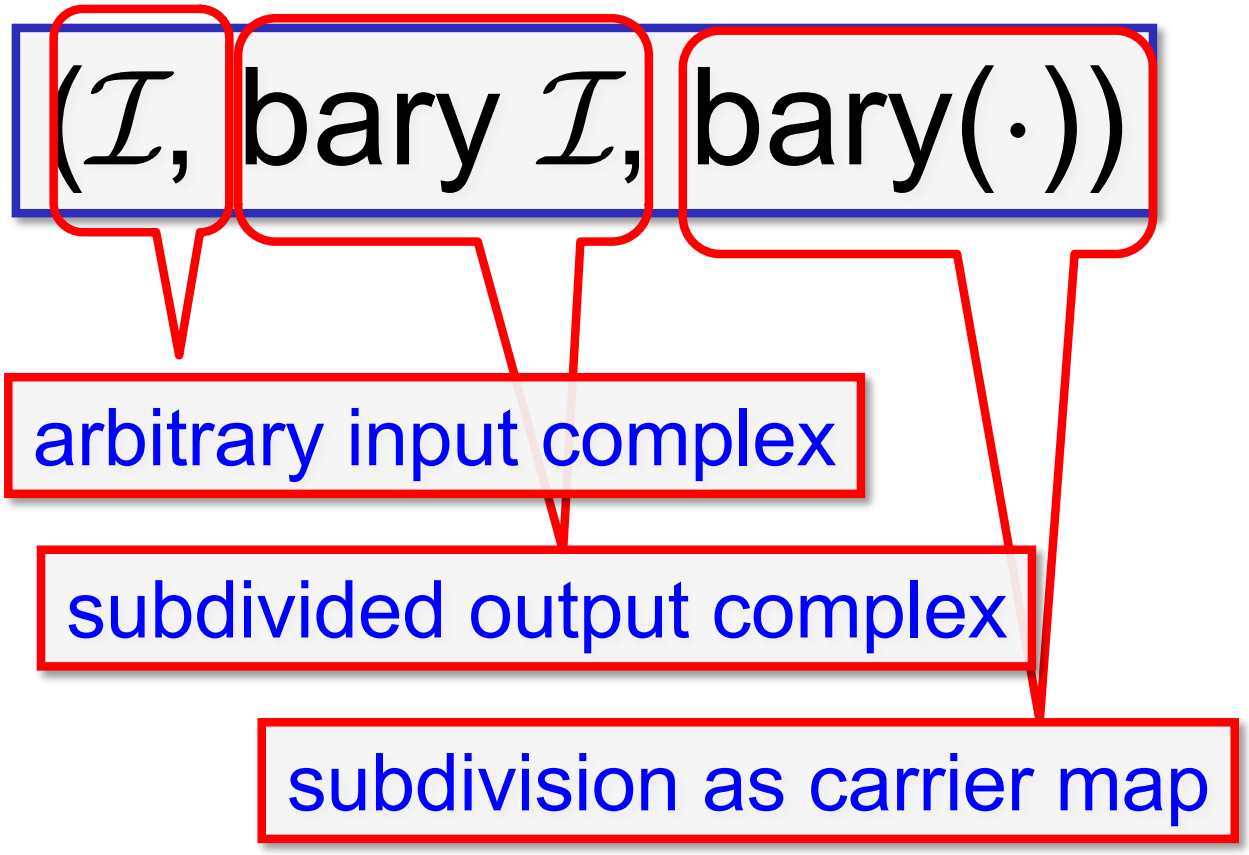


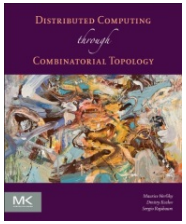
Barycentric Agreement





Barycentric Agreement

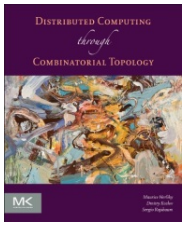




If There are n+1 Processes

$(\mathcal{I}, \text{bary skel}^n \mathcal{I}, \text{bary skel}^n)$

Inputs only from n-skeleton of input complex



Theorem

A one-layer immediate snapshot protocol solves the (*n+1*)-process barycentric agreement task
($\mathcal{I}$, bary skel^n $\mathcal{I}$, bary skel^n)

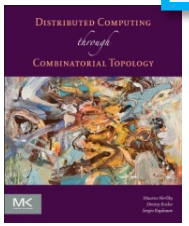
Proof

All input simplices belong to skel^n $\mathcal{I}$

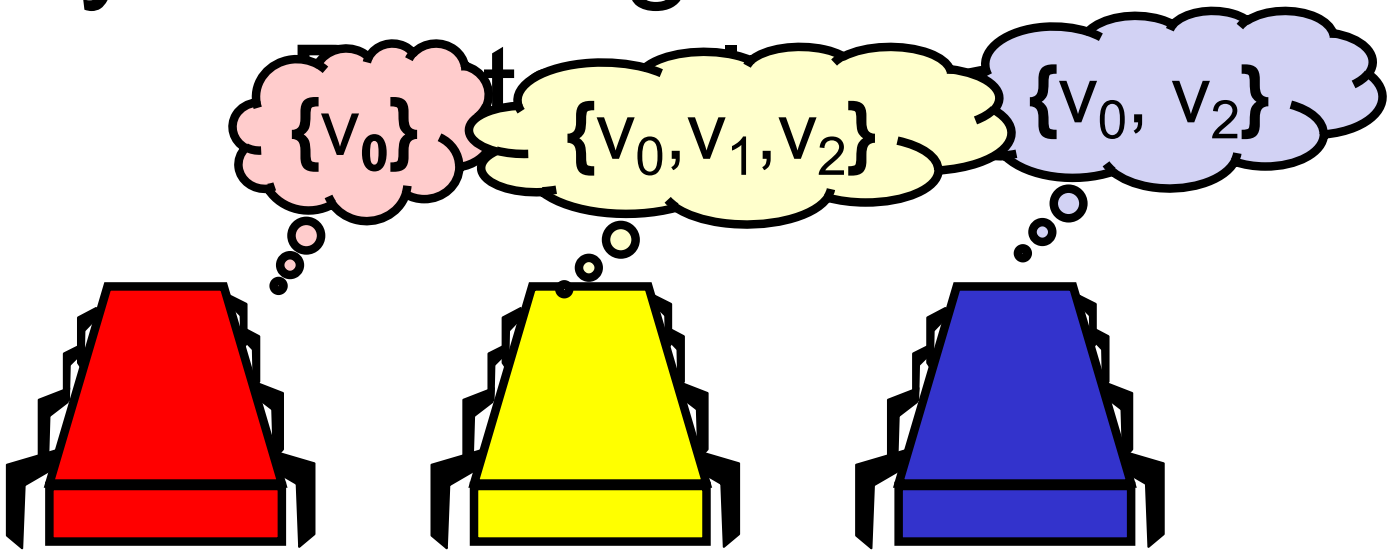
Immediate snapshot returns

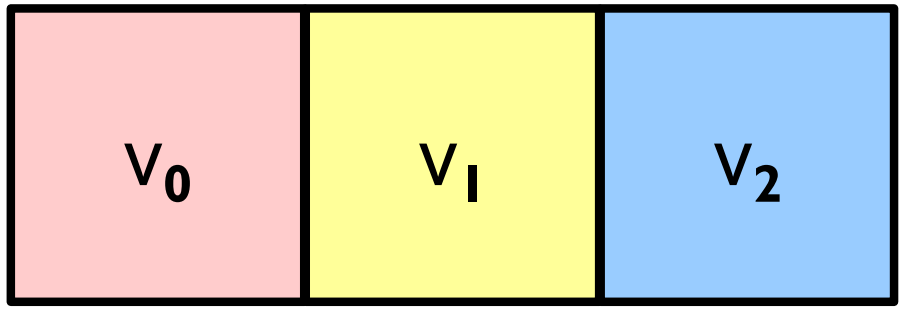
Set of input vertices (face of input simplex)

Faces ordered wrt one another

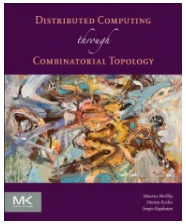


Barycentric Agreement

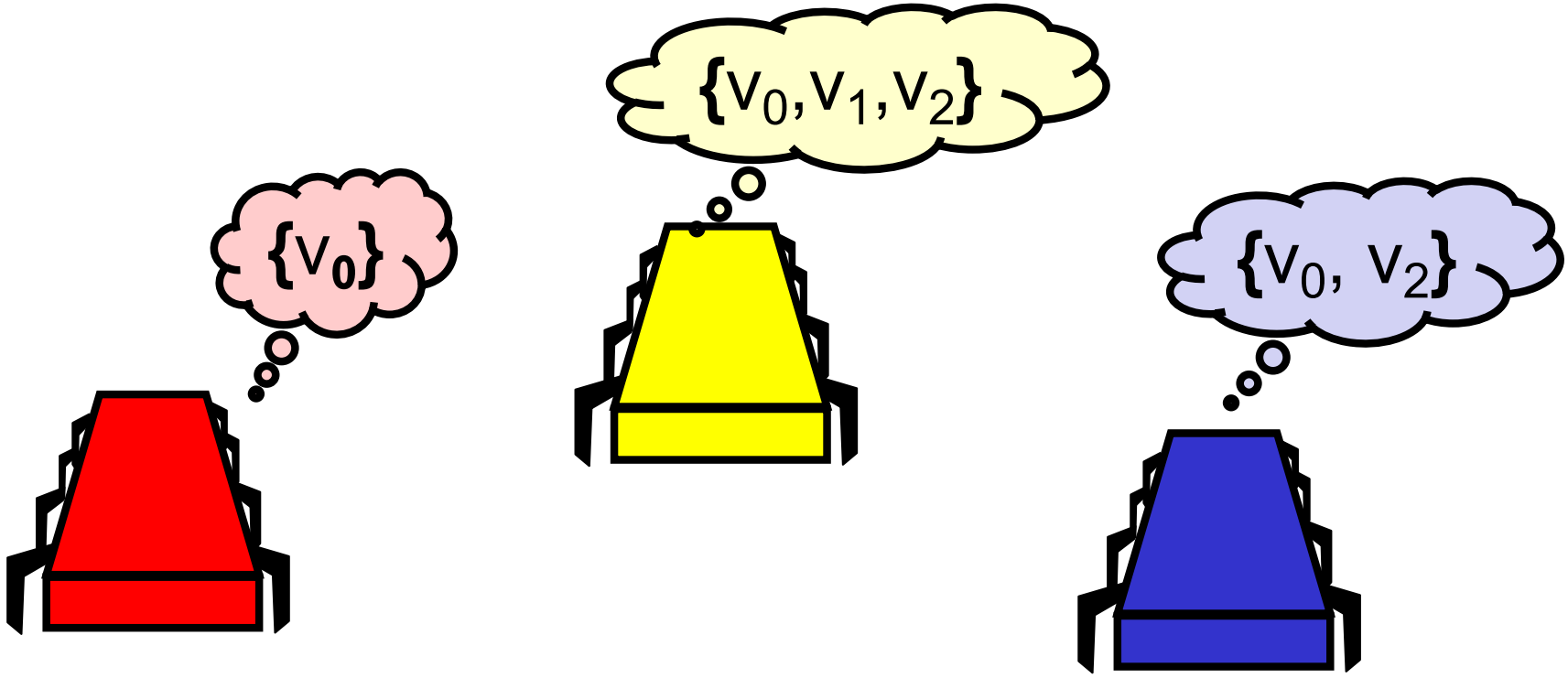


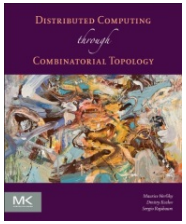


Snapshots are ordered



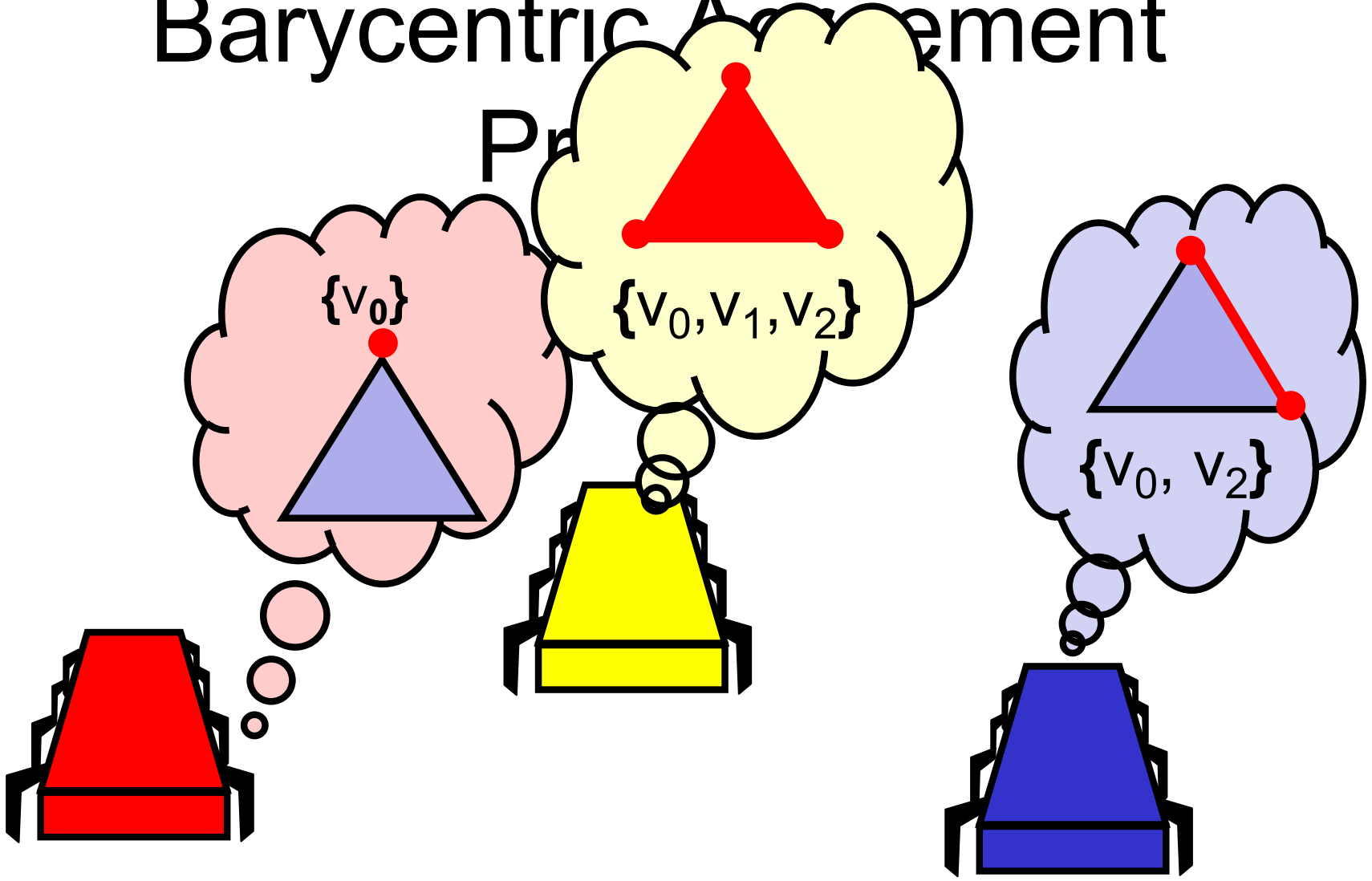
Barycentric Agreement Protocol

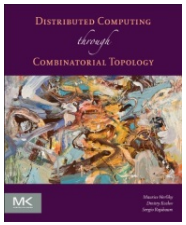




Each view is a face of σ

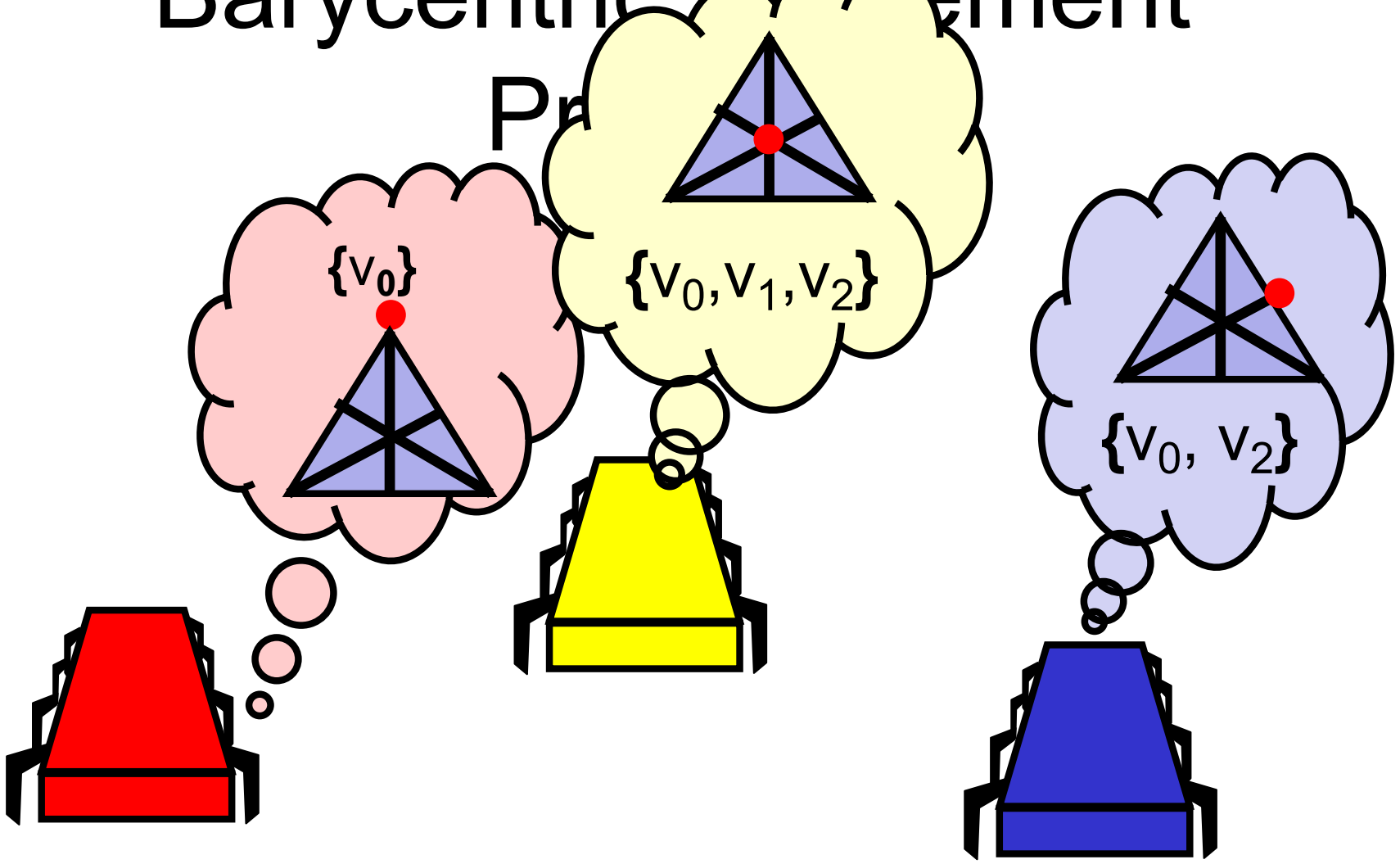
Barycentric Agreement

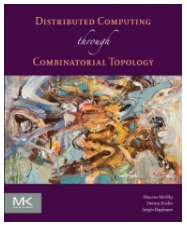




Each view is a face of σ

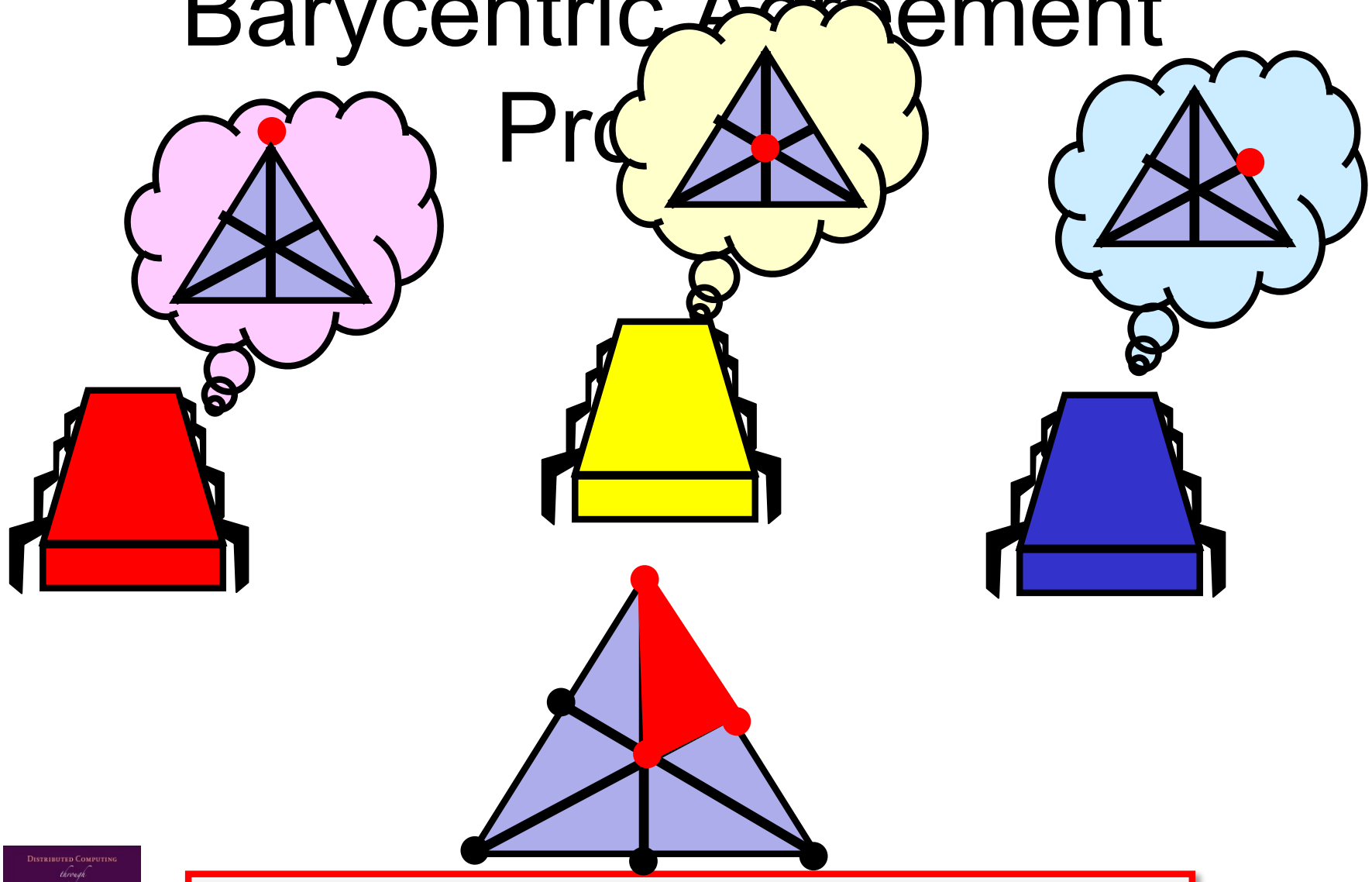
Barycentric Agreement Pr

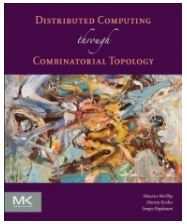




Each face of σ is a vertex of Bary σ

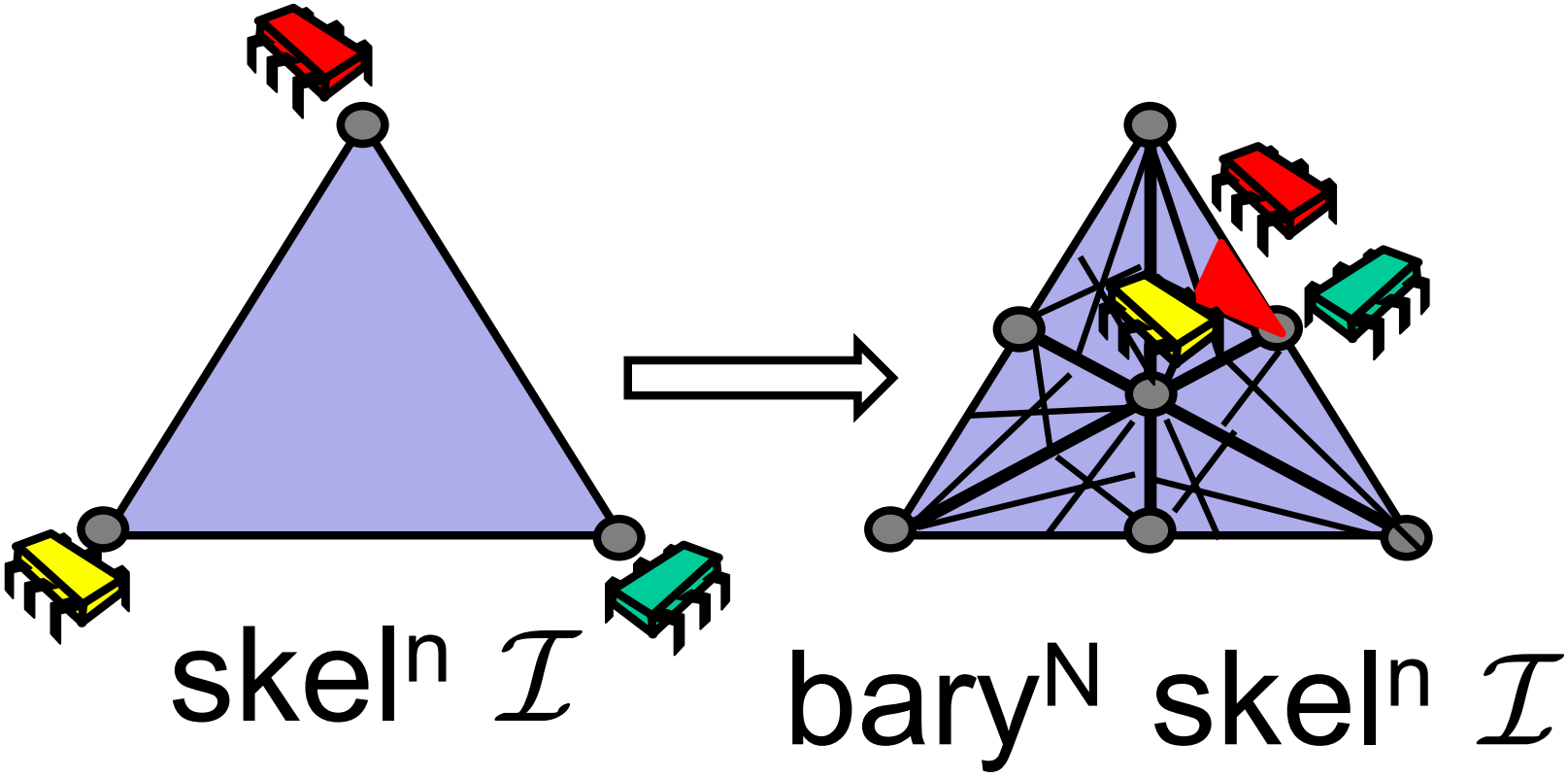
Barycentric Agreement

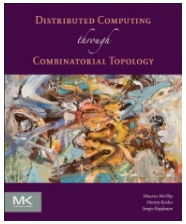




Ordered faces $\Rightarrow$ simplex of Bary σ

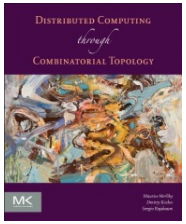
Iterated Barycentric Agreement



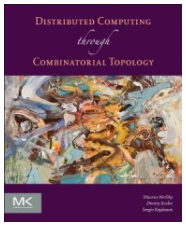


Iterated Barycentric Agreement

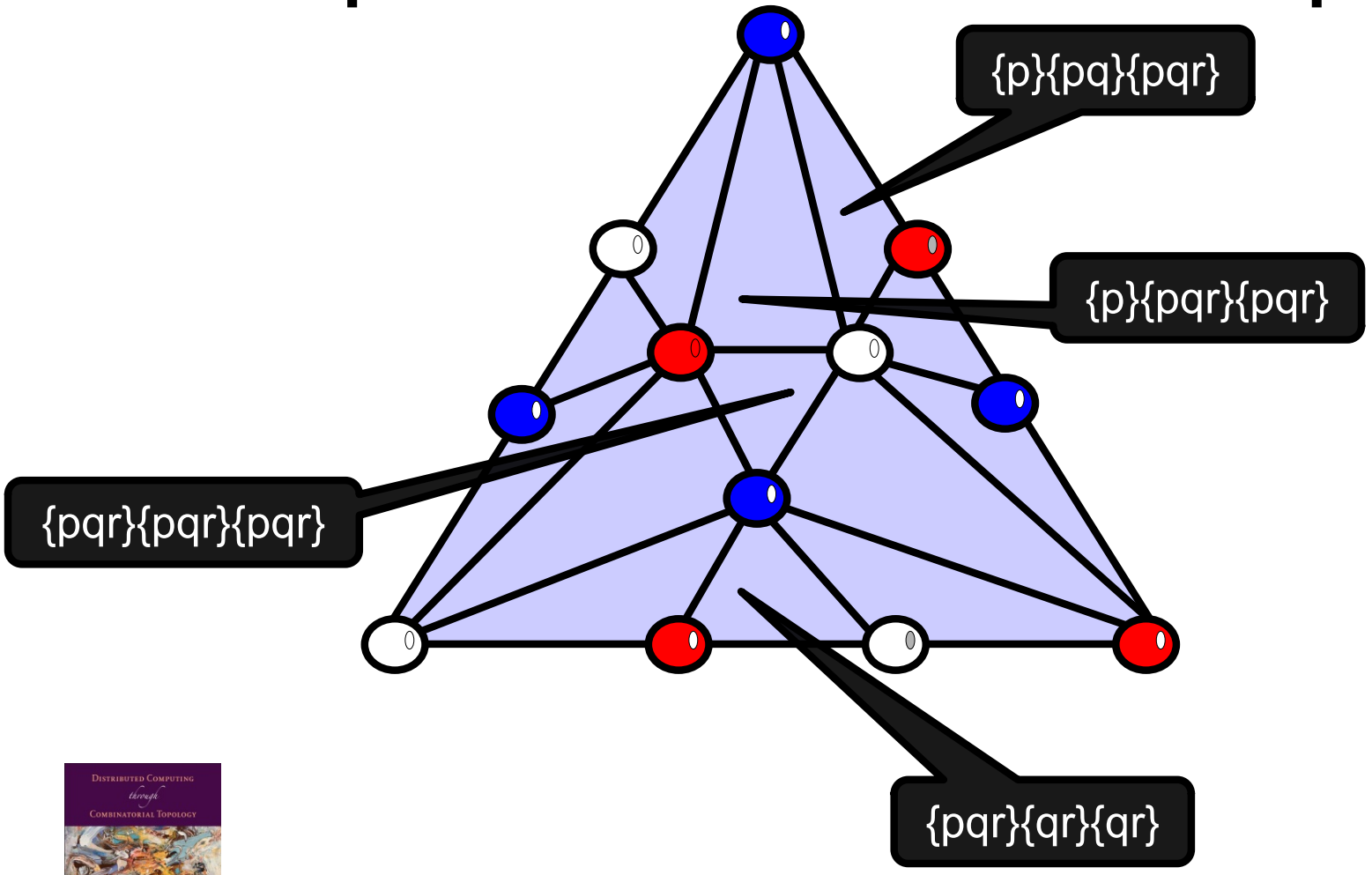
$$(\mathcal{I},\ \mathrm{bary}^{N}\ \mathrm{skel}^{n}\ \mathcal{I},\ \mathrm{bary}^{N}\ \mathrm{skel}^{n})$$



One-round IS executions

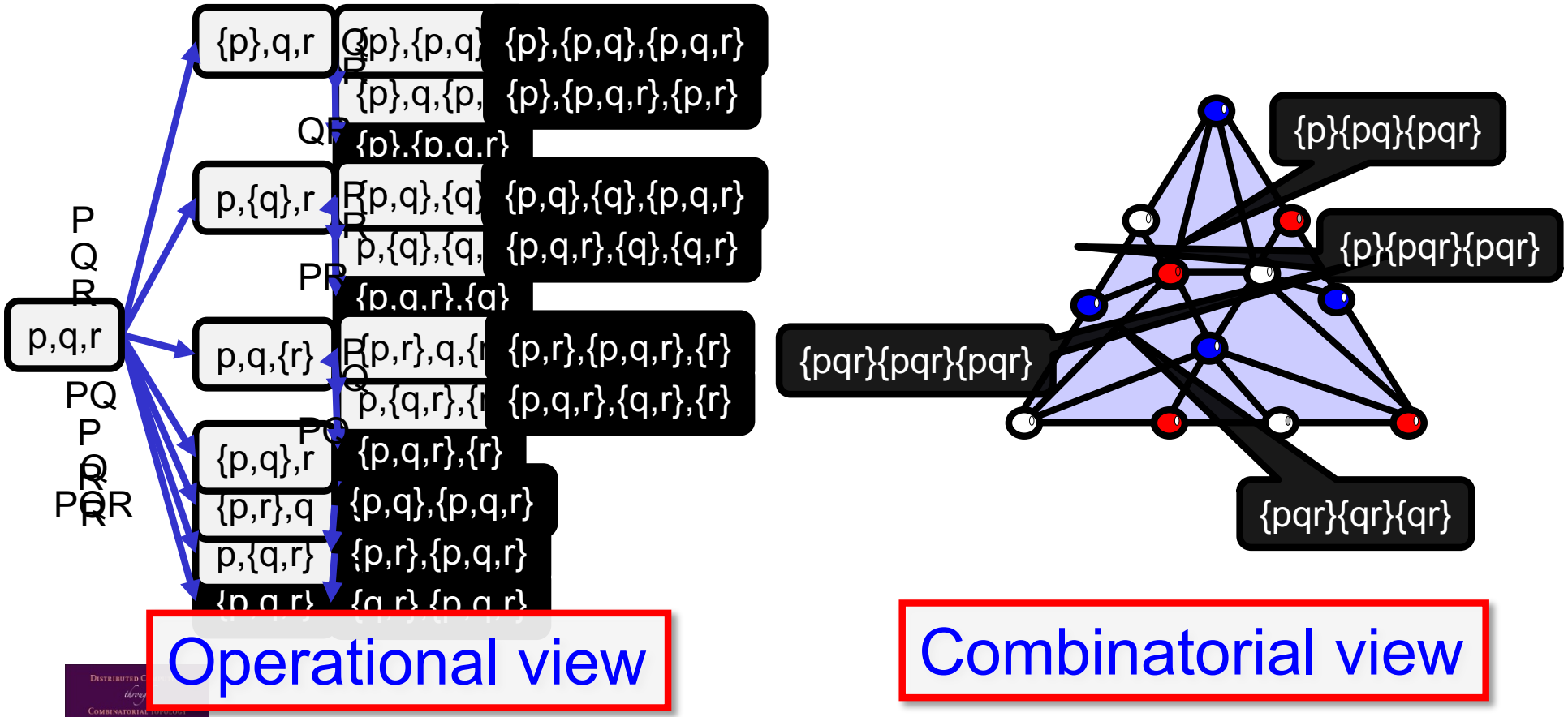


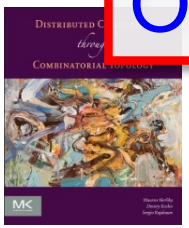
One-Layer Immediate Snapshot Protocol Complex



Distributed Computing through
Combinatorial Topology

Compare Views





Compositions

Given protocols

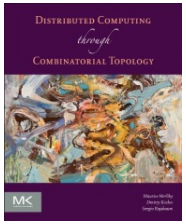
$(\mathcal{I},\mathcal{P},\Xi)$ $(\mathcal{I}',\mathcal{P}',\Xi')$

where $\mathcal{P} \subseteq \mathcal{I}'$

their *composition* is

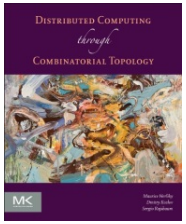
$(\mathcal{I},\mathcal{P}'',\Xi'')$

where $\Xi'' = \Xi' \circ \Xi$ and $\mathcal{P}'' = \Xi''(\mathcal{I})$



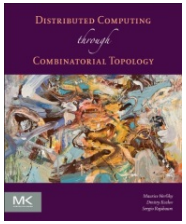
Theorem

The protocol complex for a single-layer colorless IS protocol $(\mathcal{I},\mathcal{P},\Xi)$ is Bary $\mathcal{I}$



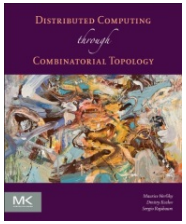
Theorem

The protocol complex for an N-layer IS protocol $(\mathcal{I},\mathcal{P},\Xi)$ is $\text{Bary}^N\,\mathcal{I}$



Corollary

The protocol complex for an *N*-layer IS protocol $(\mathcal{I},\mathcal{P},\Xi)$ is *n*-connected

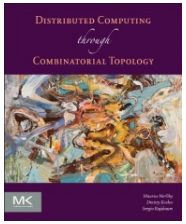


Road Map

Operational Model

Combinatorial Model

Main Theorem



Fundamental Theorem

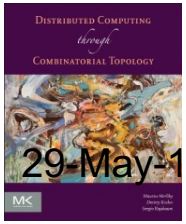
Theorem

Colorless task $(\mathcal{I},\mathcal{O},\Delta)$ has a wait-free $(n+1)$-process layered IS protocol iff there is a continuous map

$f: |\text{skel}^n \mathcal{I}| \to |\mathcal{O}|$...

carried by Δ

$\forall \sigma \in \text{skel}^n \mathcal{I}, f(|\sigma|) \subseteq |\Delta(\sigma)|$



Map $\Rightarrow$ Protocol

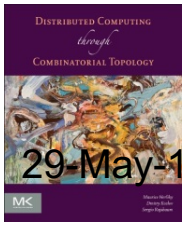
Lemma

If …

there is a continuous
f: $|\text{skel}^n \mathcal{I}| \to |\mathcal{O}|$ carried by Δ

then …

there is a WF layered protocol for $(\mathcal{I},\mathcal{O},\Delta)$



Recall: Simplicial Approximation

$\phi : \mathcal{A} \to \mathcal{B}$ is a simplicial approximation of $f : |\mathcal{A}| \to |\mathcal{B}|$ if …

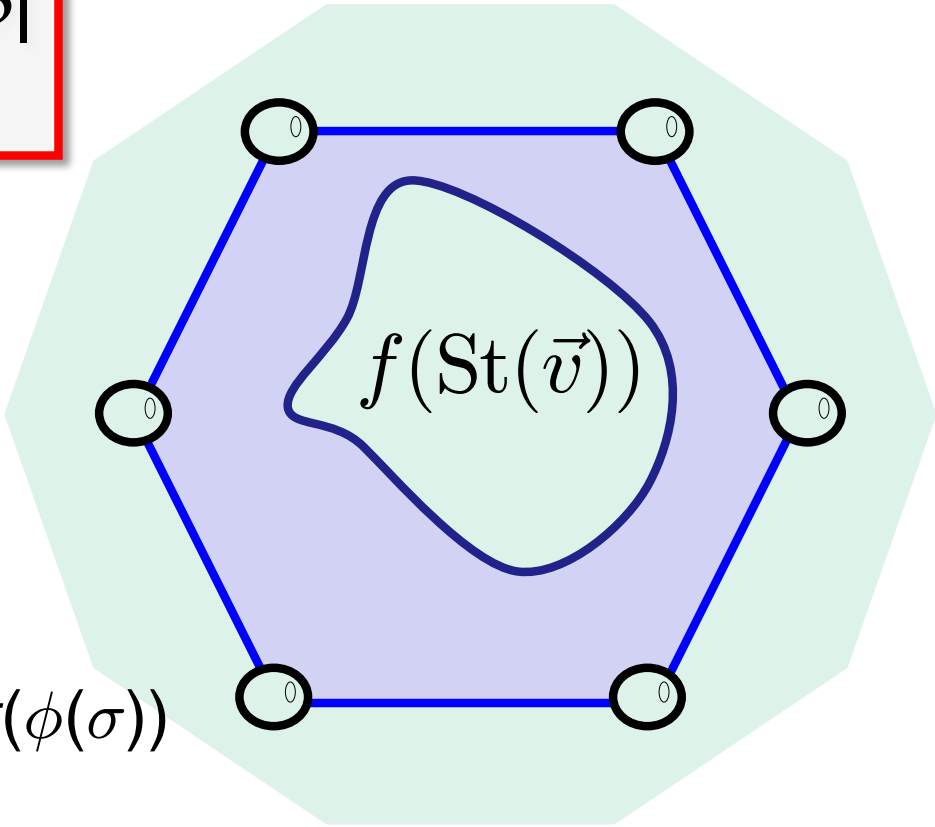
for every v in $\mathcal{A}$ …

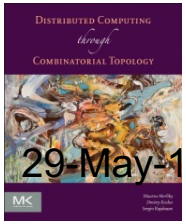
$$f(St(\vec{v})) \subseteq St(\phi(\vec{v}))$$

Alternatively:

$\forall\ \sigma \in \mathcal{A}$, $f(|\sigma|) \subseteq \bigcap_{v \in \sigma} St(\phi(v)) = St(\phi(\sigma))$

(open stars)





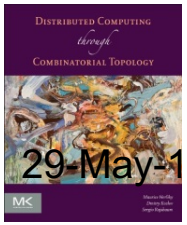
Recall: Simplicial Approximation Theorem

- Given a continuous map

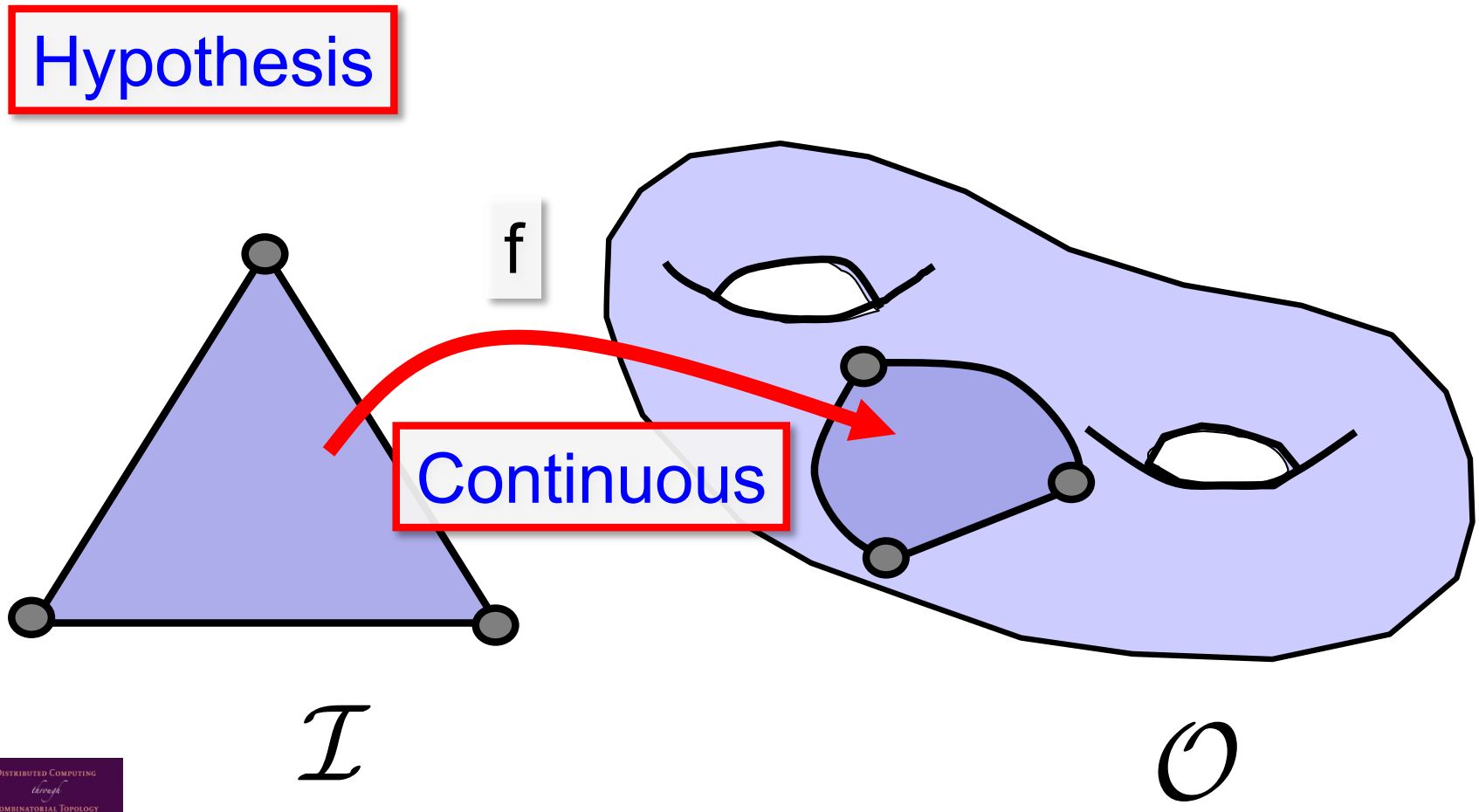
$$f : |A| \to |B|$$

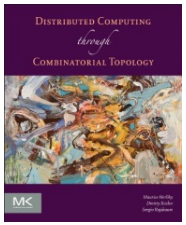
- there is an N such that f has a simplicial approximation

$$\phi : Bary^N A \to B$$

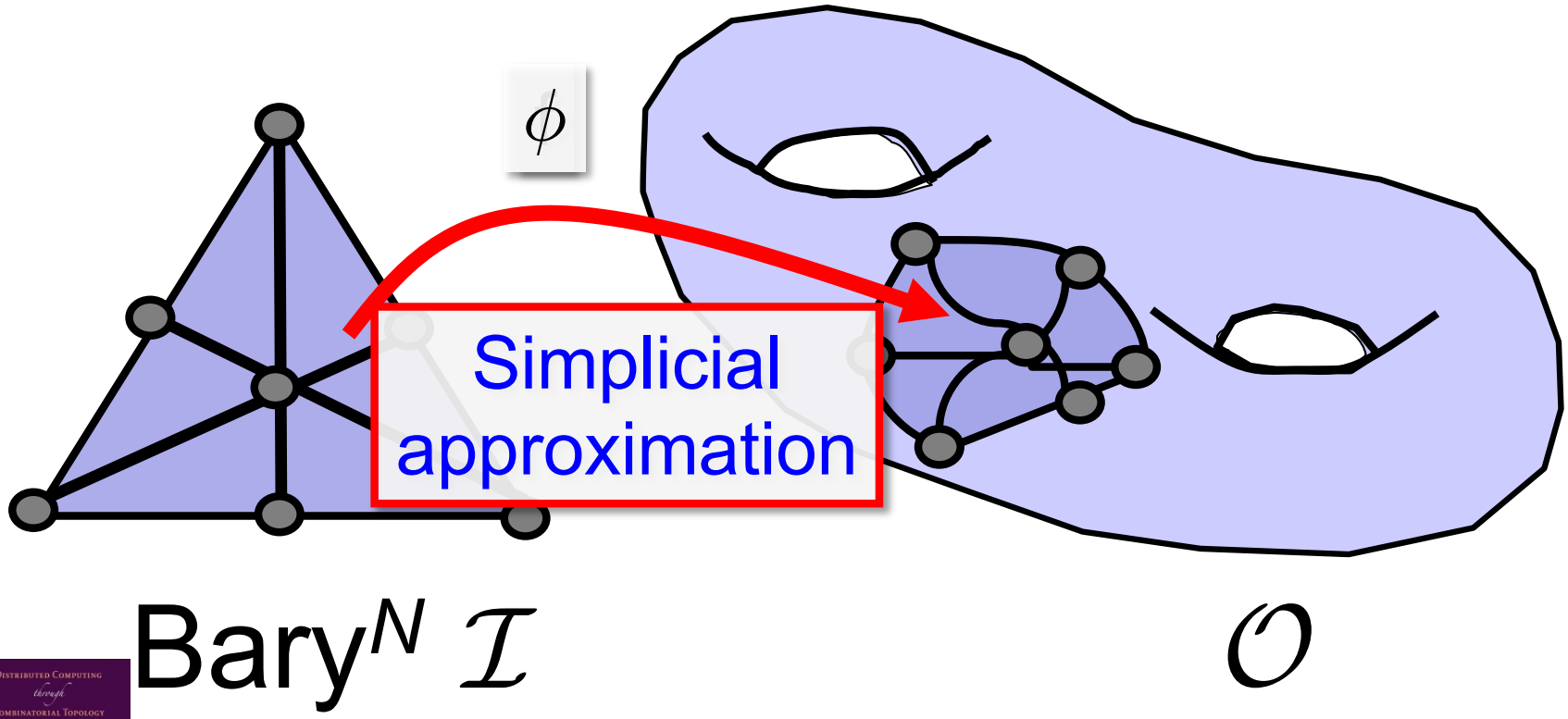


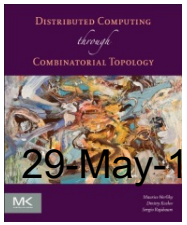
Map ⇒ Protocol



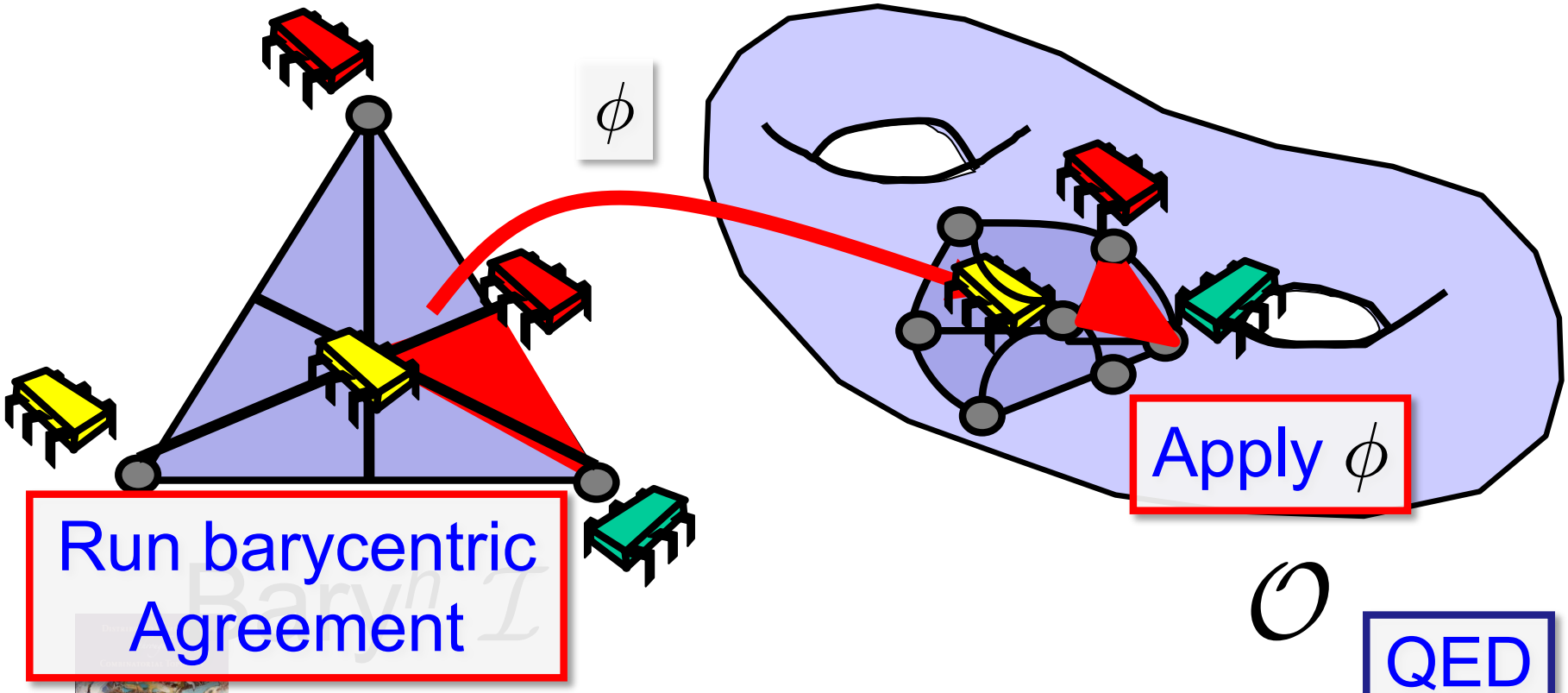


Map $\Rightarrow$ Protocol





Map $\Rightarrow$ Protocol



Protocol $\Rightarrow$ Map

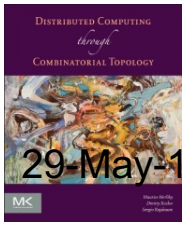
Lemma

If …

there is a WF-RW protocol for $(\mathcal{I},\mathcal{O},\Delta)$ …

then …

there is a continuous f: $|\text{skel}^n \mathcal{I}| \rightarrow |\mathcal{O}|$ carried by Δ.

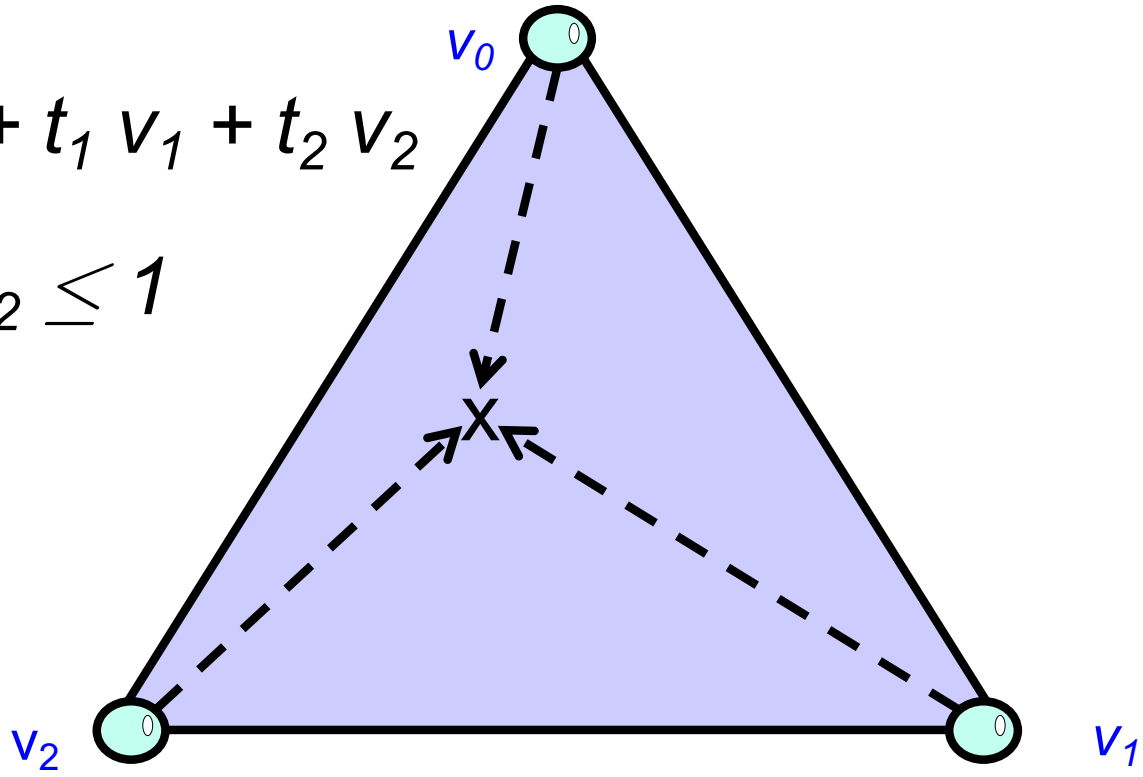


Recall: Barycentric Coordinates

$x = t_0 v_0 + t_1 v_1 + t_2 v_2$

$0 \leq t_0, t_1, t_2 \leq 1$

$\sum_i t_i = 1$



Given a complex $\mathcal{C}$, very point of $|\mathcal{C}|$ has a unique representation using barycentric coordinates

Protocol $\Rightarrow$ Map

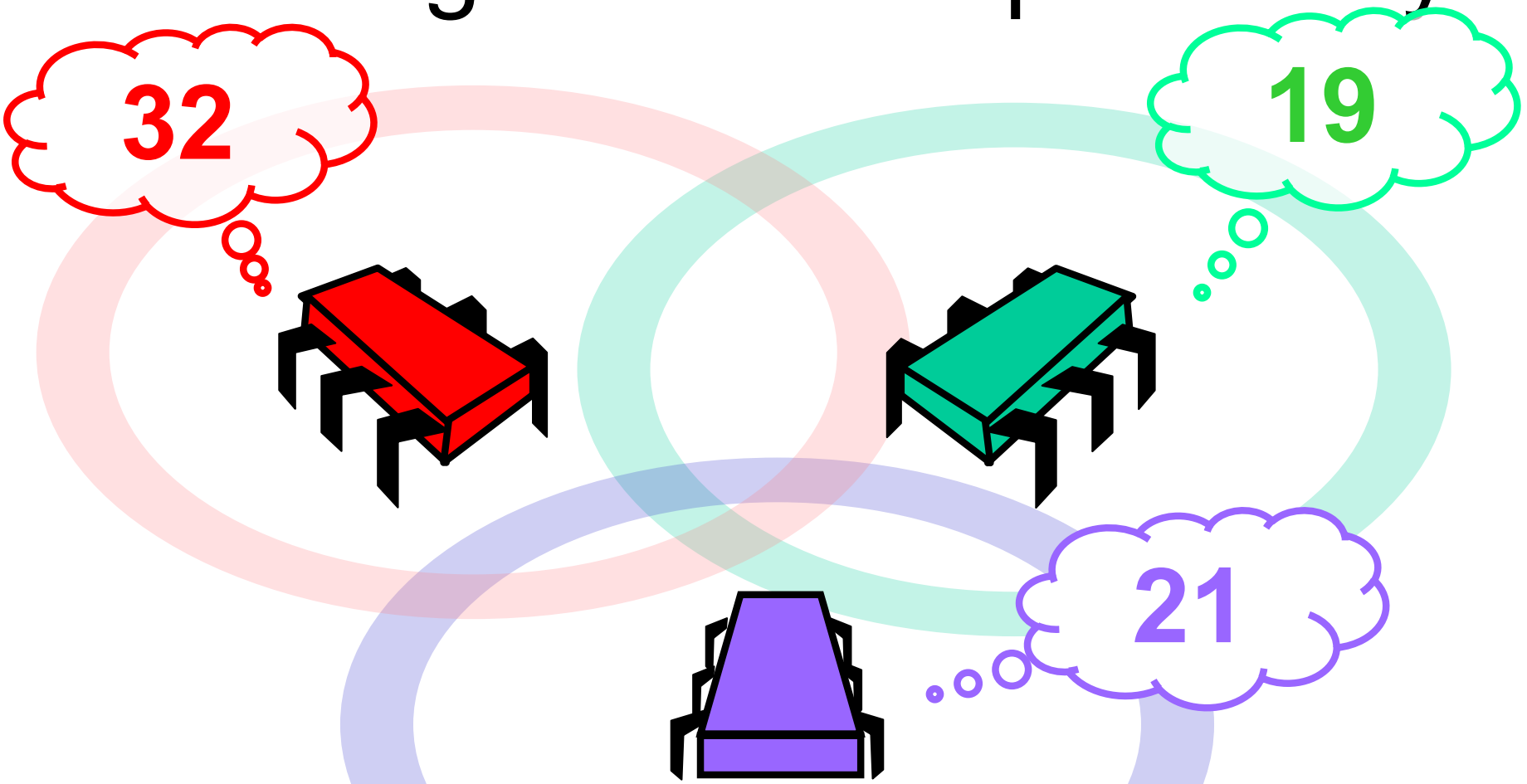
- Bary^N - the protocol map
- δ: $\text{Bary}^N\ \mathcal{I} \rightarrow$ O - the decision map

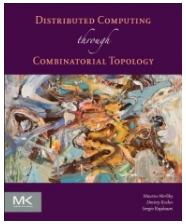
$\delta \circ \text{Bary}^N$ is carried by Δ:
$\forall \sigma \in \mathcal{I}$, $\delta \circ \text{Bary}^N(\sigma) \subseteq \Delta(\sigma)$

Take $\phi = |\delta|$: $|\text{Bary}^N\ \mathcal{I}| \rightarrow |O|$
(Barycentric extension of δ to $|\text{Bary}^N(\mathcal{I})| = |\mathcal{I}|$)is carried by Δ:
$\forall \sigma \in \mathcal{I}$, $\phi(|\sigma|) \subseteq |\delta \circ \text{Bary}^N(\sigma)| \subseteq |\Delta(\sigma)|$

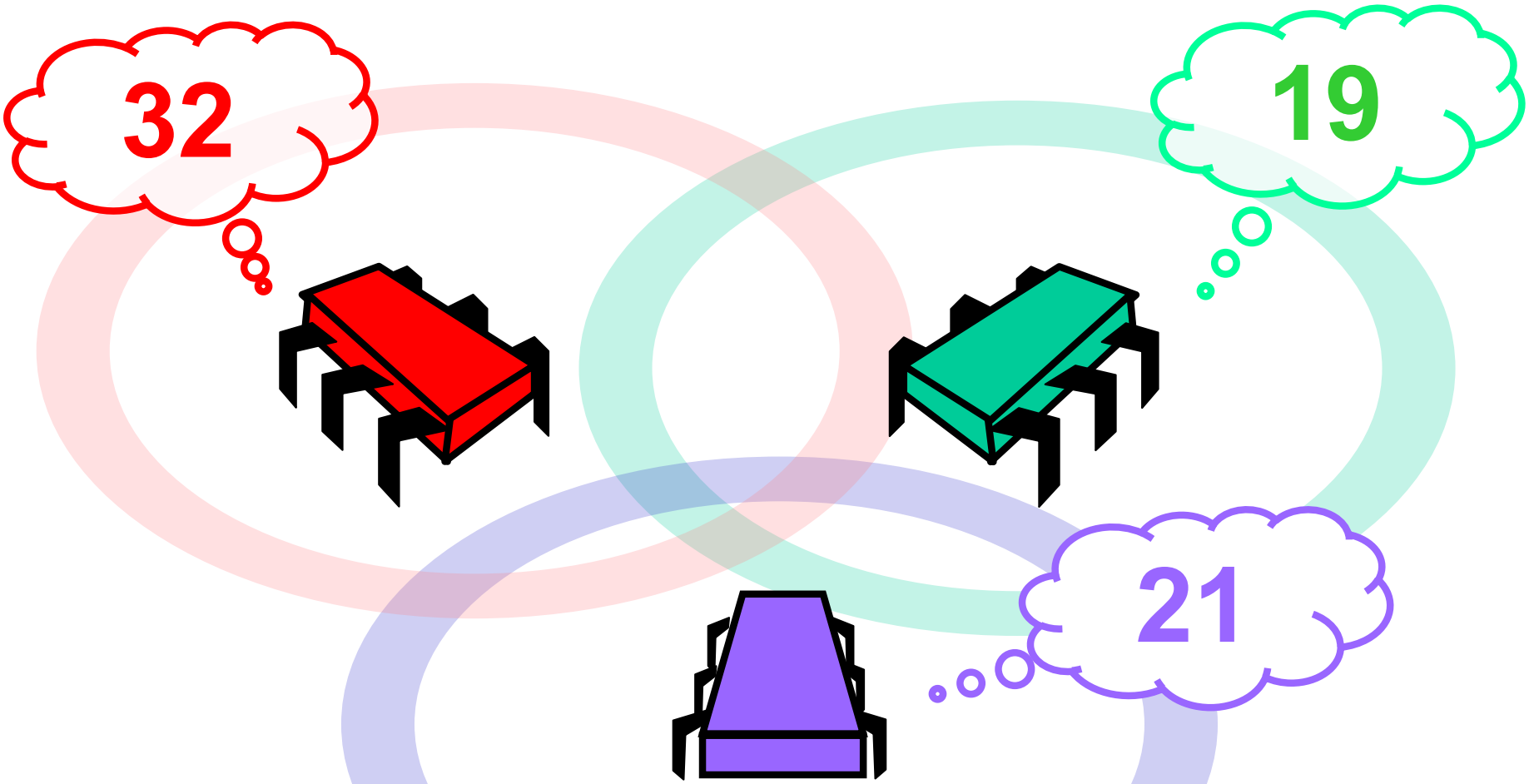
QED

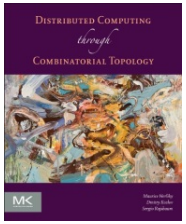
Application: *k*-set Agreement Impossibility



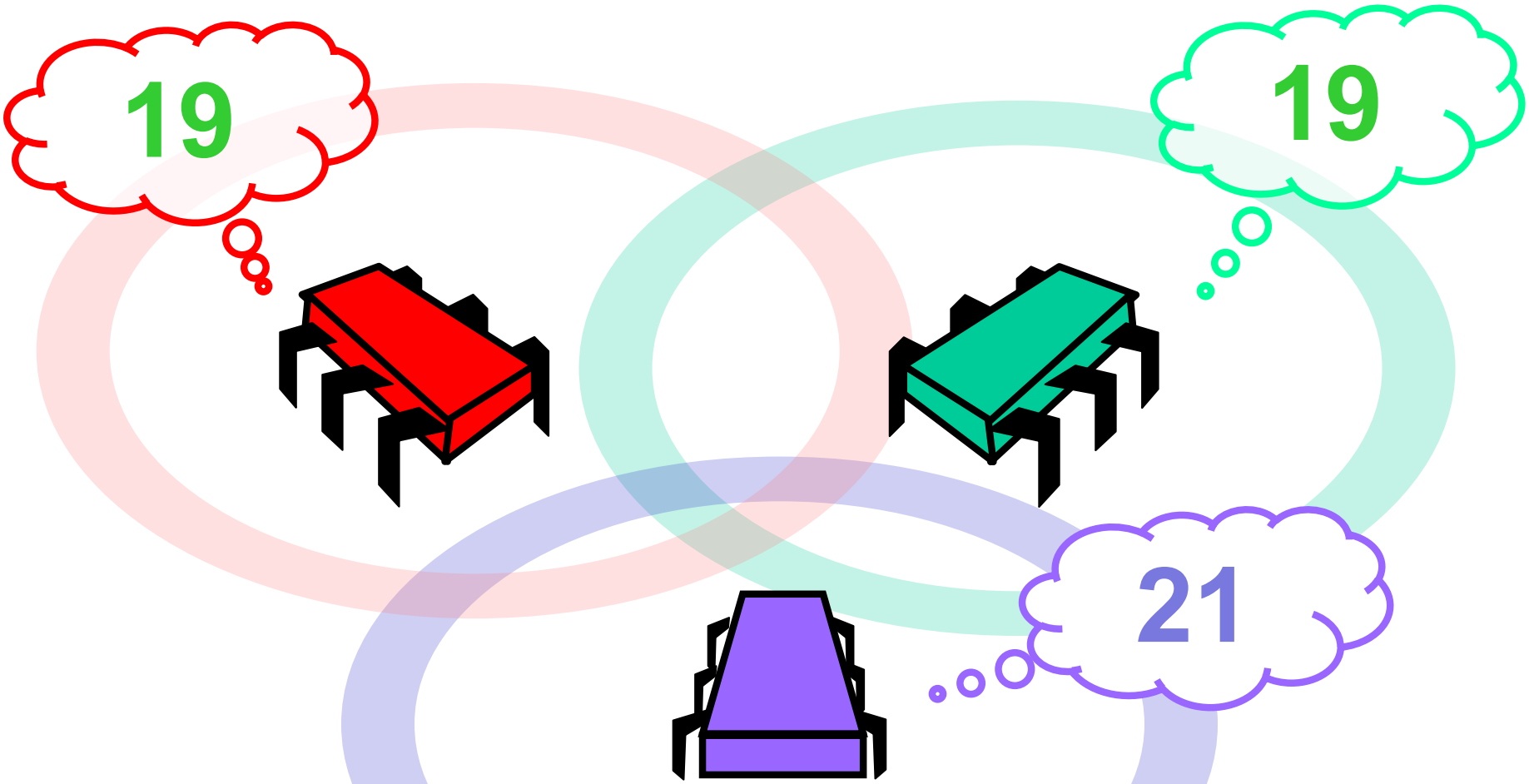


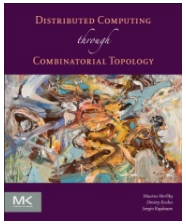
k-set Agreement: before





k-set Agreement: after

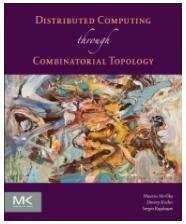




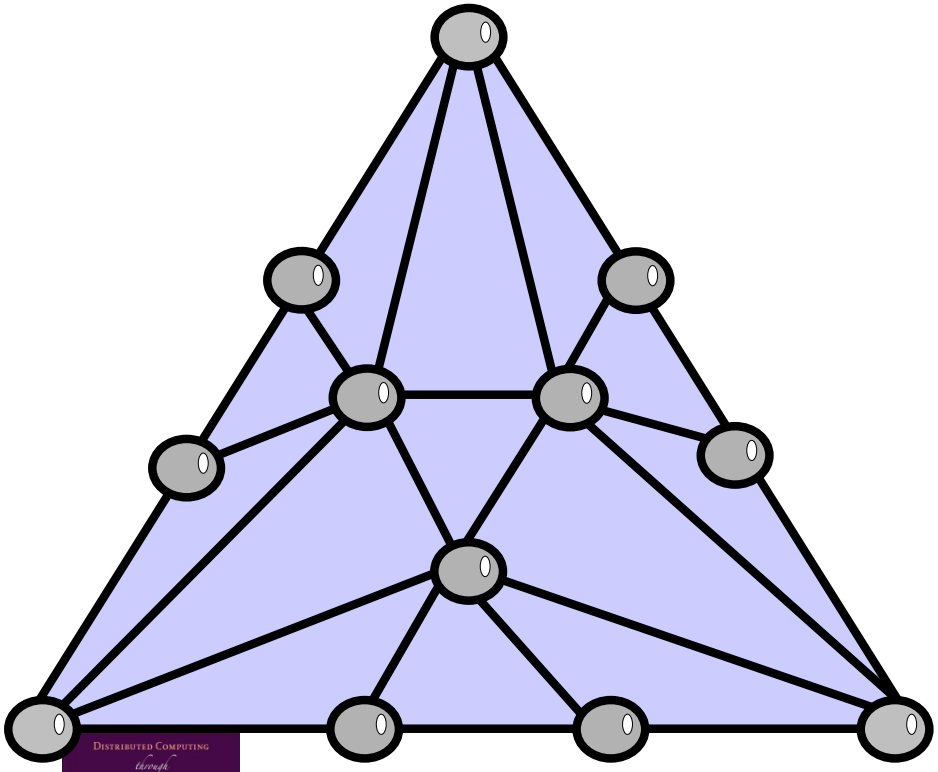
$k = 2$

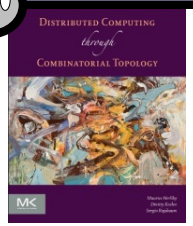
Theorem

No layered (n+1)-process IS protocol can solve *n*-set agreement



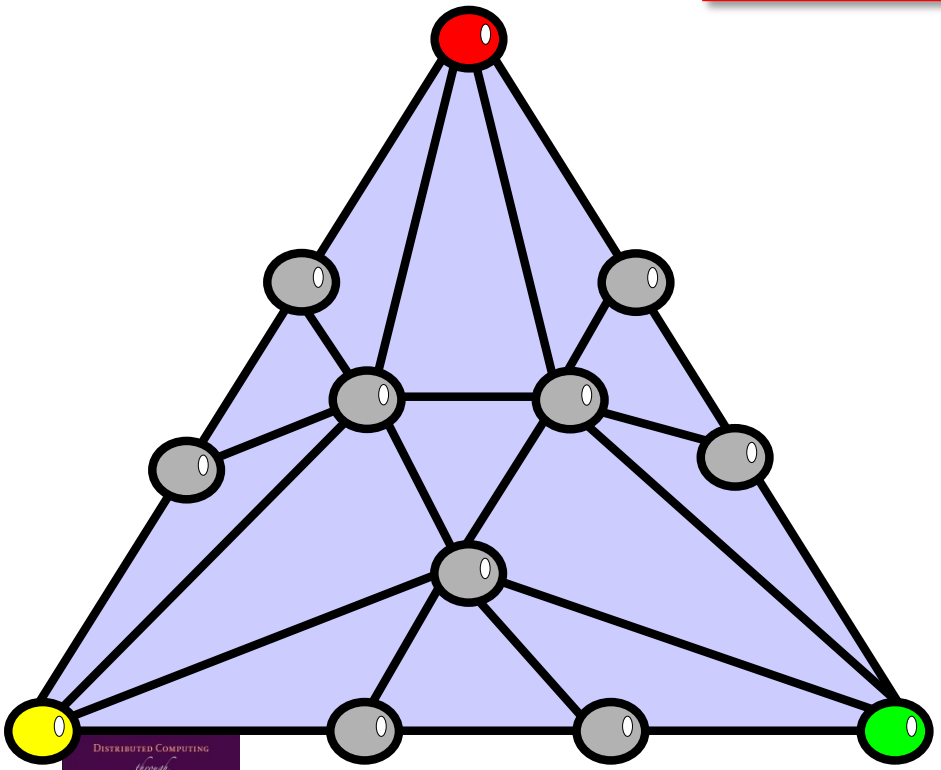
Sperner Coloring

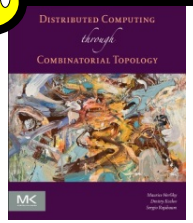




Sperner Coloring

"Corners" have distinct colors

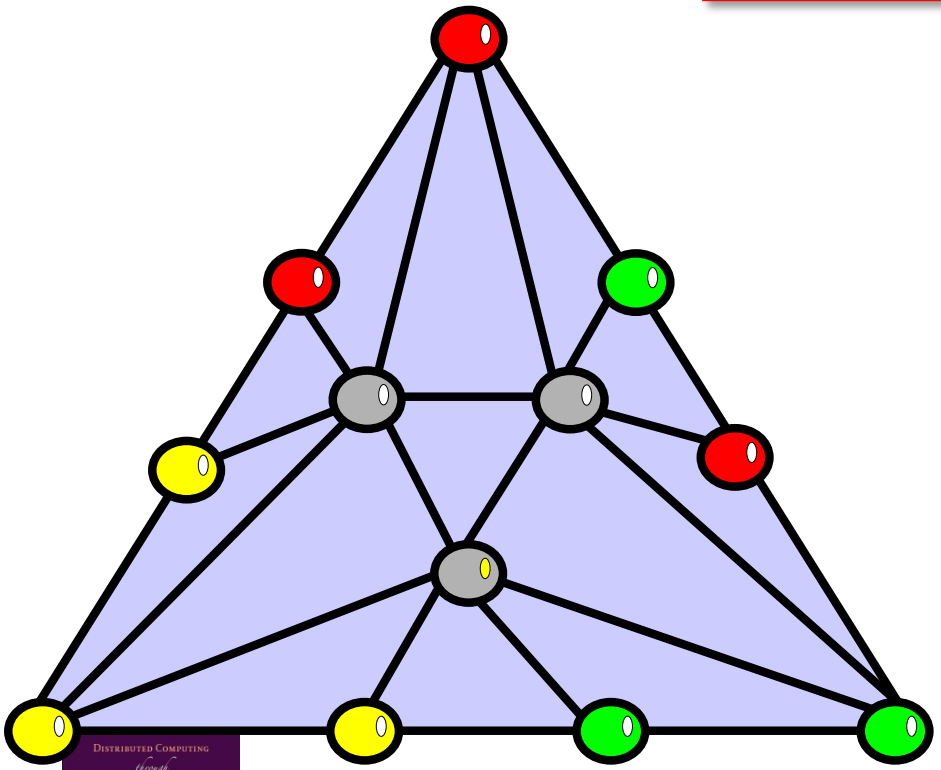


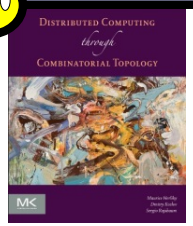


Sperner Coloring

"Corners" have distinct colors

Edge vertexes have corner colors



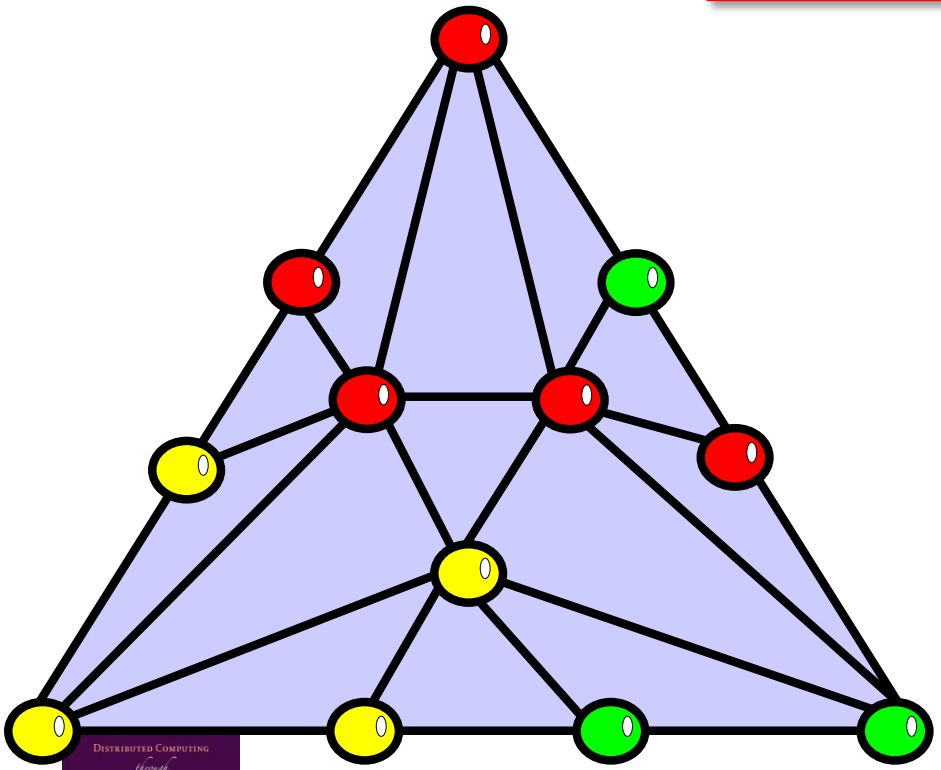


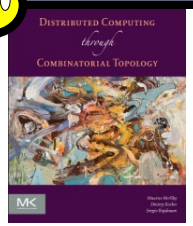
Sperner Coloring

"Corners" have distinct colors

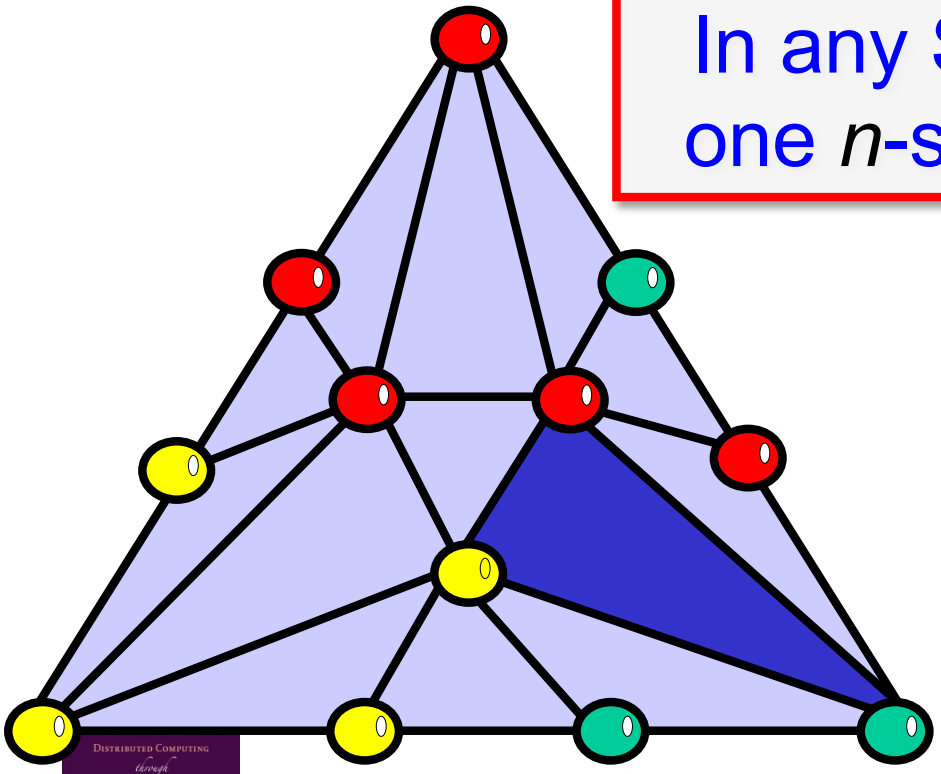
Edge vertexes have corner colors

Every vertex has face boundary colors

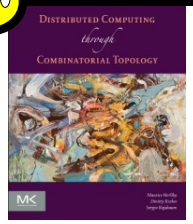




Sperner's Lemma



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

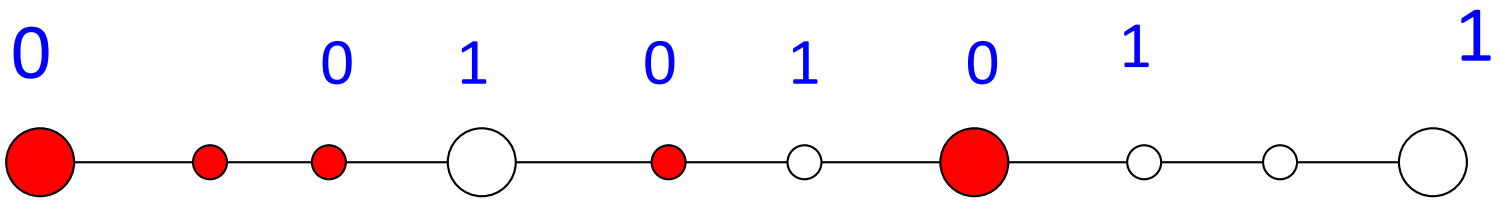


Sperner's lemma: inductive step

Claim: for each k=0,…,N, face 0,…,k contains an odd number of k-dimensional simplexes colored 0,…,k

By induction: k=0 - trivial (exactly one)

k=1, simple counting



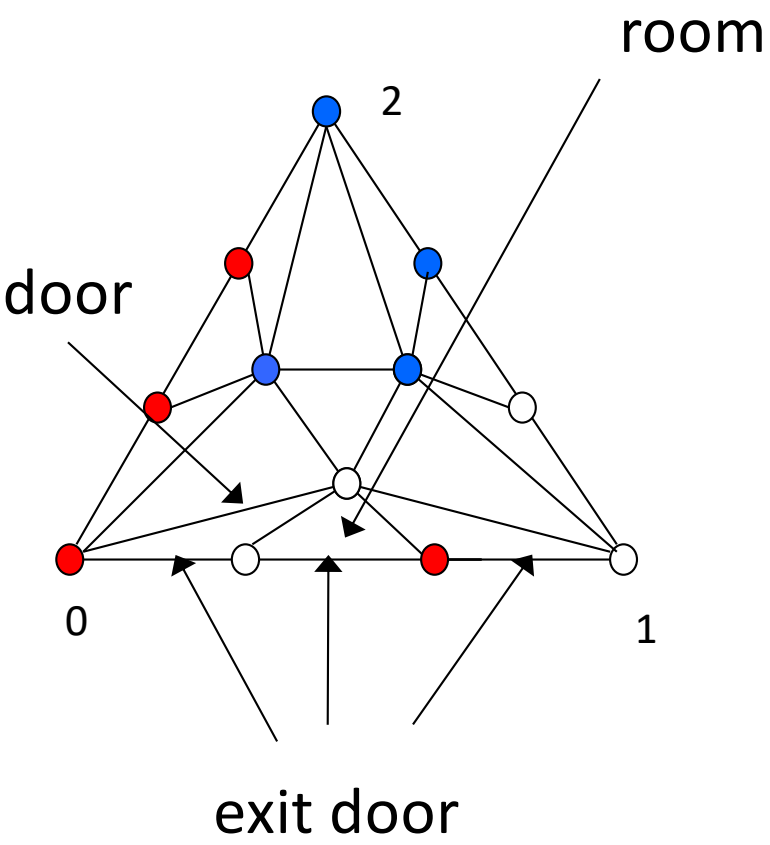
Suppose the claim holds for k=N-1 and consider the face 0,…,N

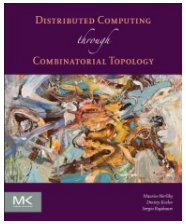
Sperner: rooms and doors

Each N-simplex is a room

An (N-1)-dimensional face
(a subset of N-1 vertices)
of a room colored in
0,…,N-1 is a door

A door is an exit if it is
contained in the face
0,…,N-1

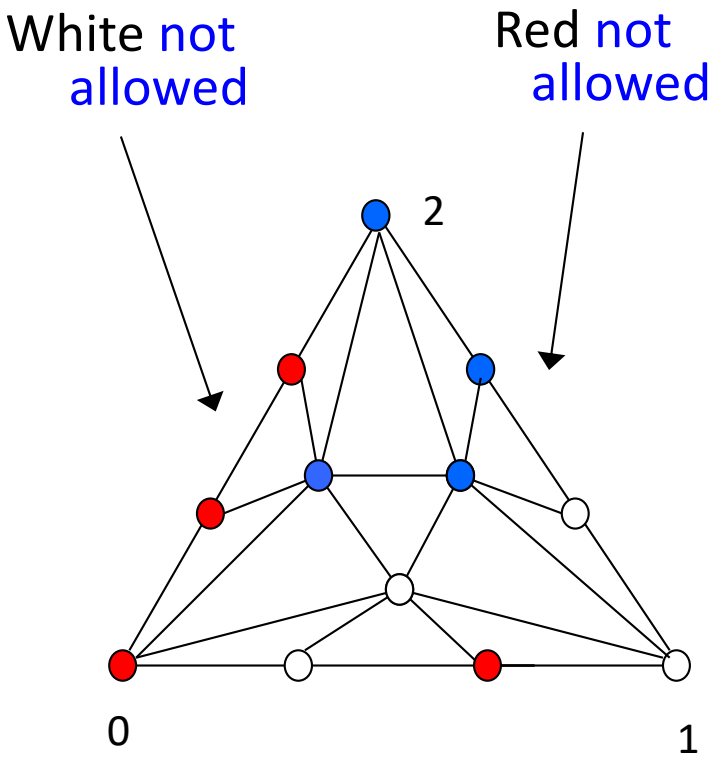


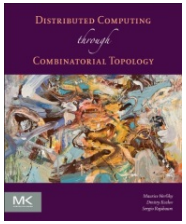


Sperner: exit doors

There is an odd number of exits!

- No face other than 0,…,N-1 can contain simplexes colored 0,…,N-1
- Exits may only be contained in 0,…,N-1

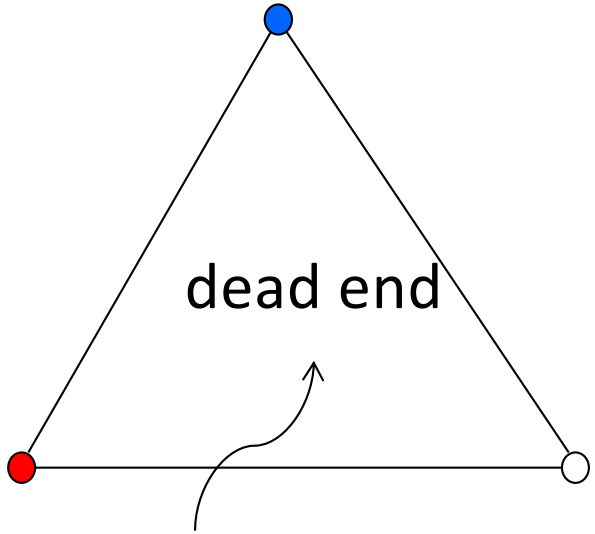


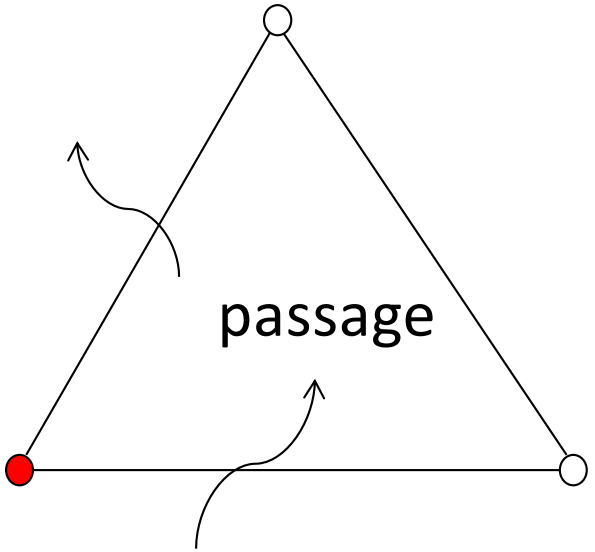


Sperner: passages and dead ends

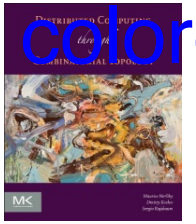
A room with a door is either:

- A passage (has two doors), or
- A dead end (has no doors)





There must be an odd number of dead ends (fully colored simplexes)



Sperner's: counting fully colored rooms

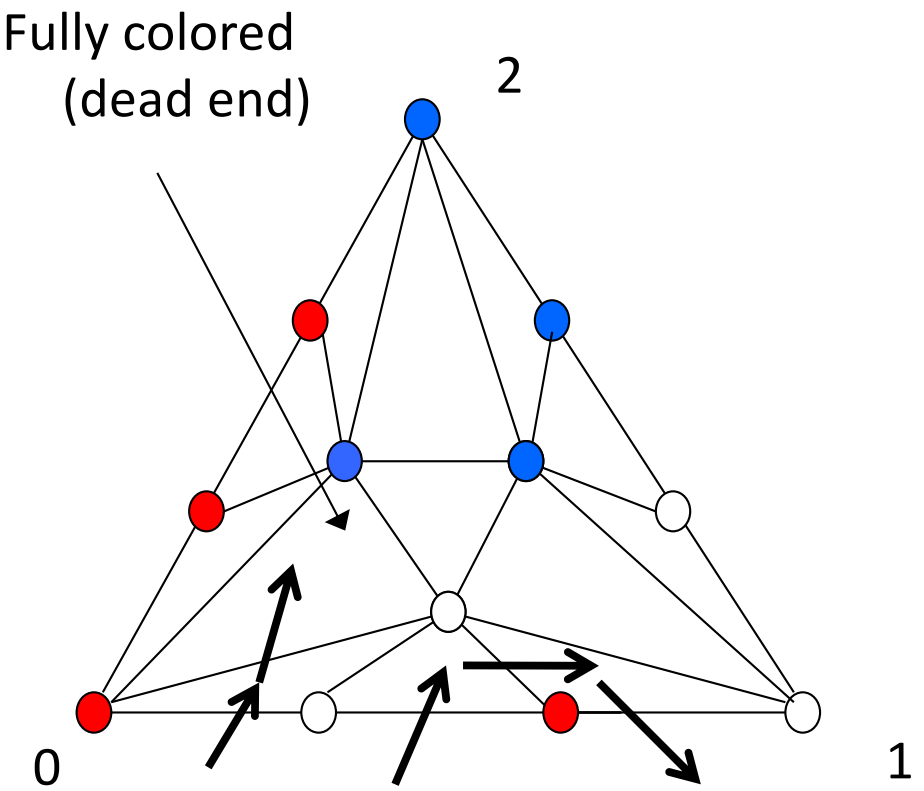
Start with an exit and walk through the doors

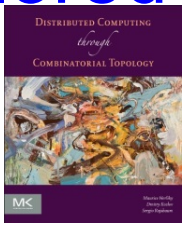
Two cases are possible:

- Stop in a dead end
- Reach another exit

The number of exit doors is odd =>

The total number of fully colored rooms is odd





No Layered IS Protocol can solve n-Set Agreement

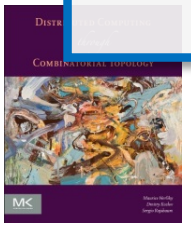
Assume protocol exists:

Run layered IS protocol

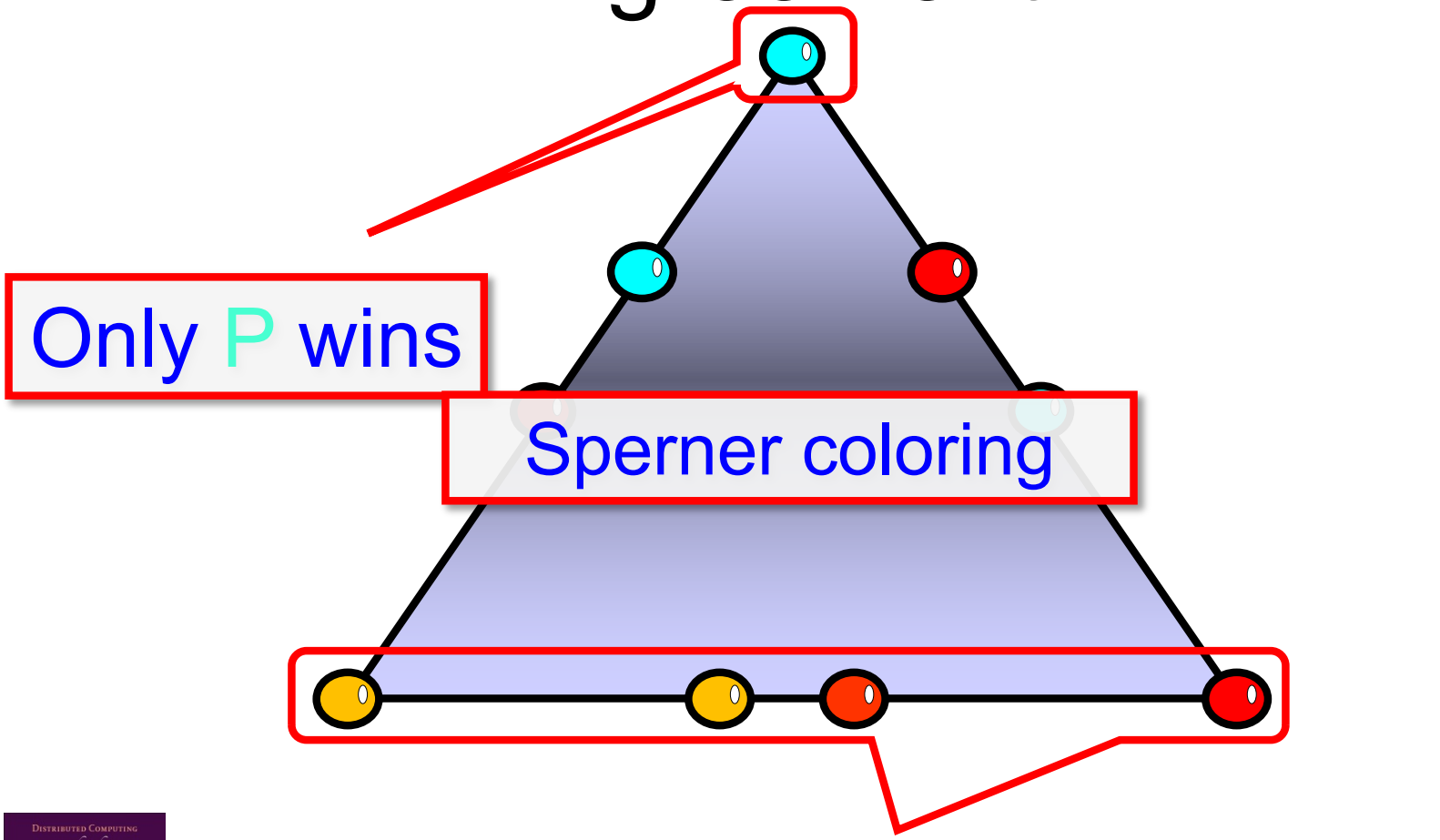
Choose value based on vertex

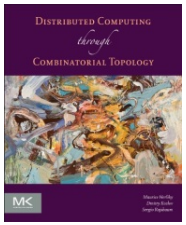
Idea:

Color vertex with “winning” process name …

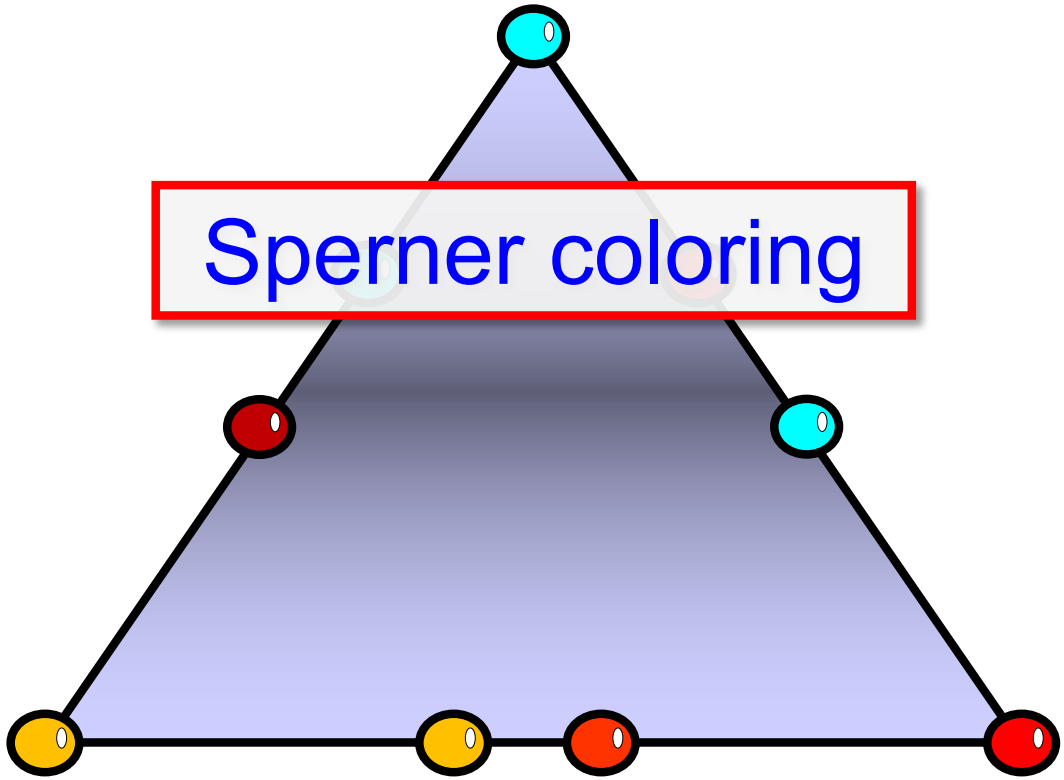


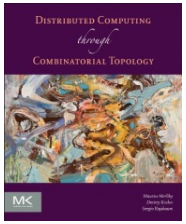
Layered IS protocol for *n*-Set Agreement



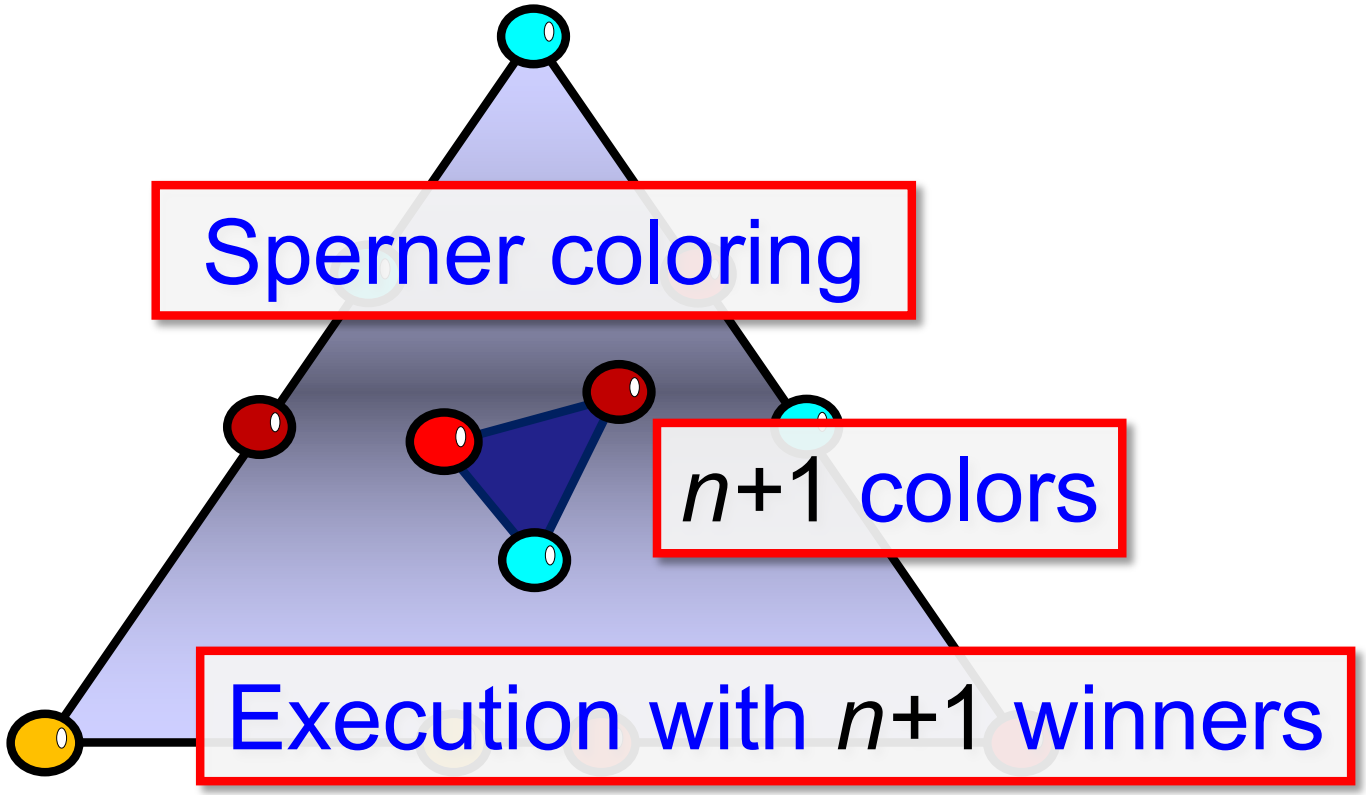


Layered IS protocol for *n*-Set Agreement





Layered IS protocol for *n*-Set Agreement



Contradiction: at most *n* can win