The Topology of Asynchronous Byzantine Colorless Tasks

Joint with
Hammurabi Mendes
Christine Tasson

BROWN

Overview

Basic concepts of combinatorial topology

Duality between combinatorial & continuous mechanisms

First *characterization* of tasks that can be solved in the asynchronous Byzantine model

Road Map

Colorless Tasks

Operational Model

Combinatorial Model

Building Blocks

Crash Failure Solvability

Byzantine Failure Solvability

Road Map

Colorless Tasks

Operational Model

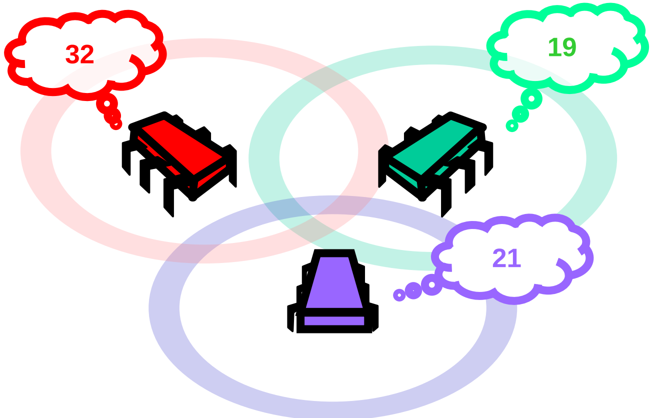
Combinatorial Model

Building Blocks

Crash Failure Solvability

Byzantine Failure Solvability

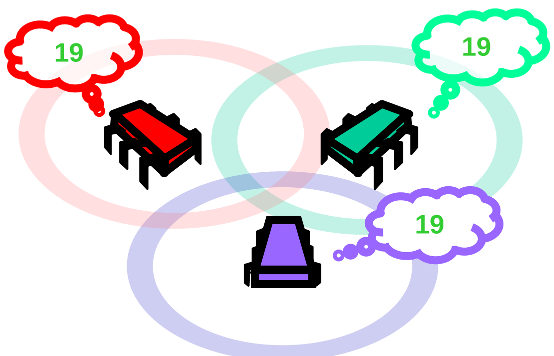
Tasks

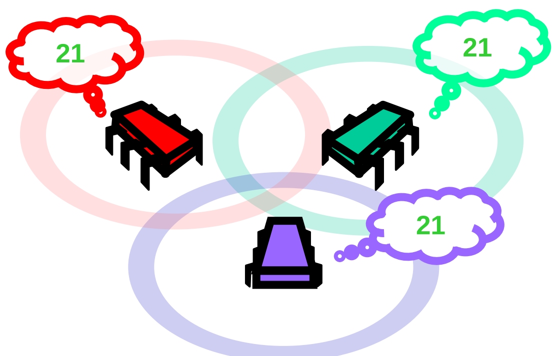


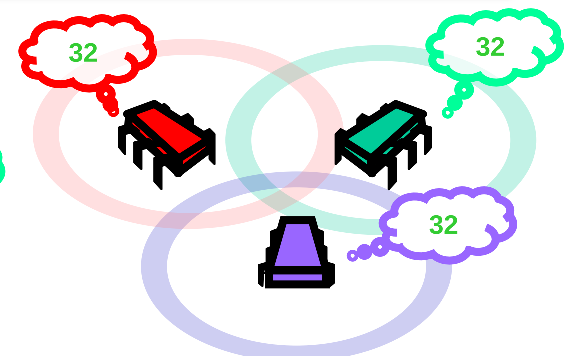
Possible set of *input* values

Finite computation

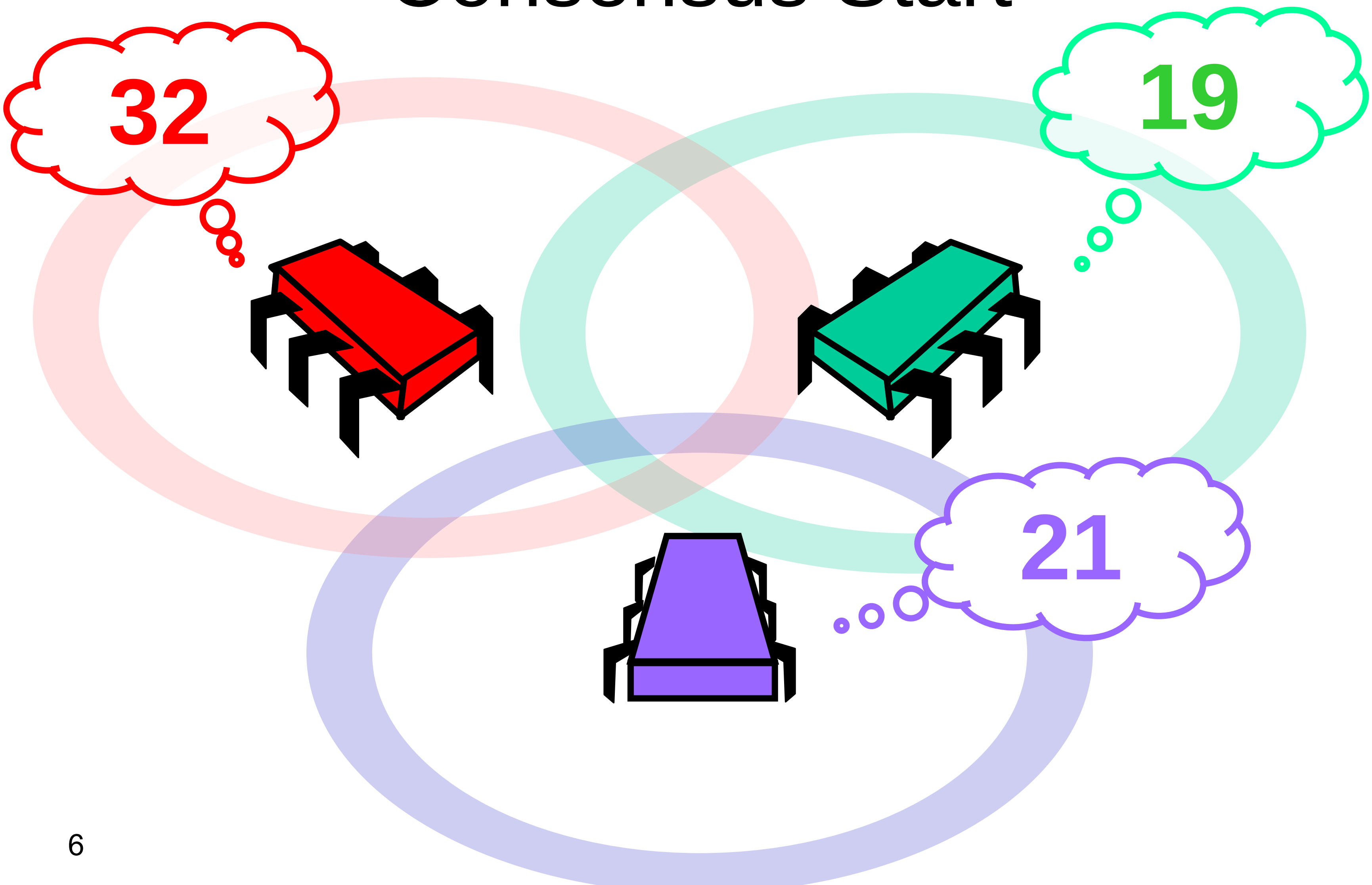
Possible set of *output* values



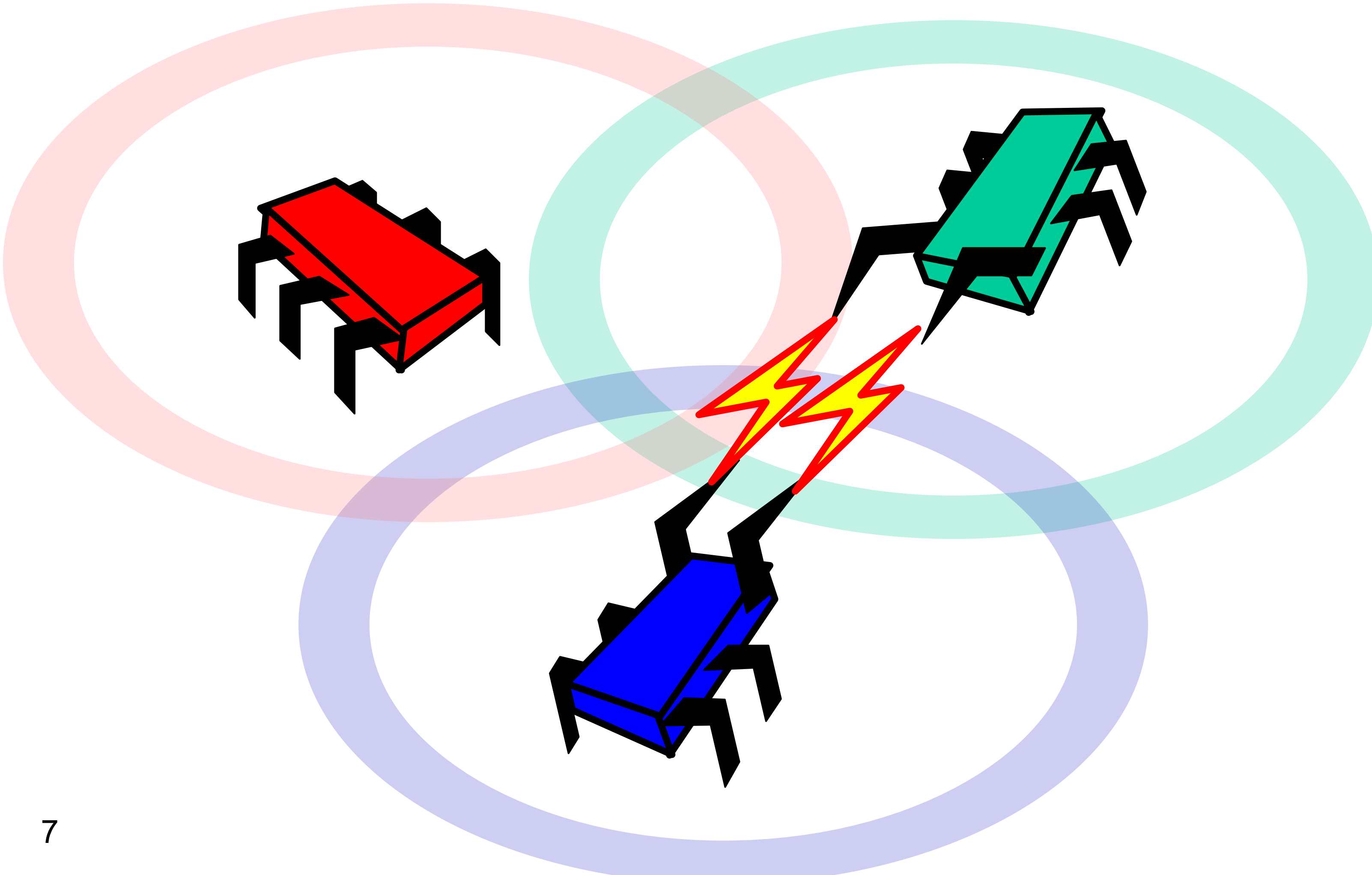




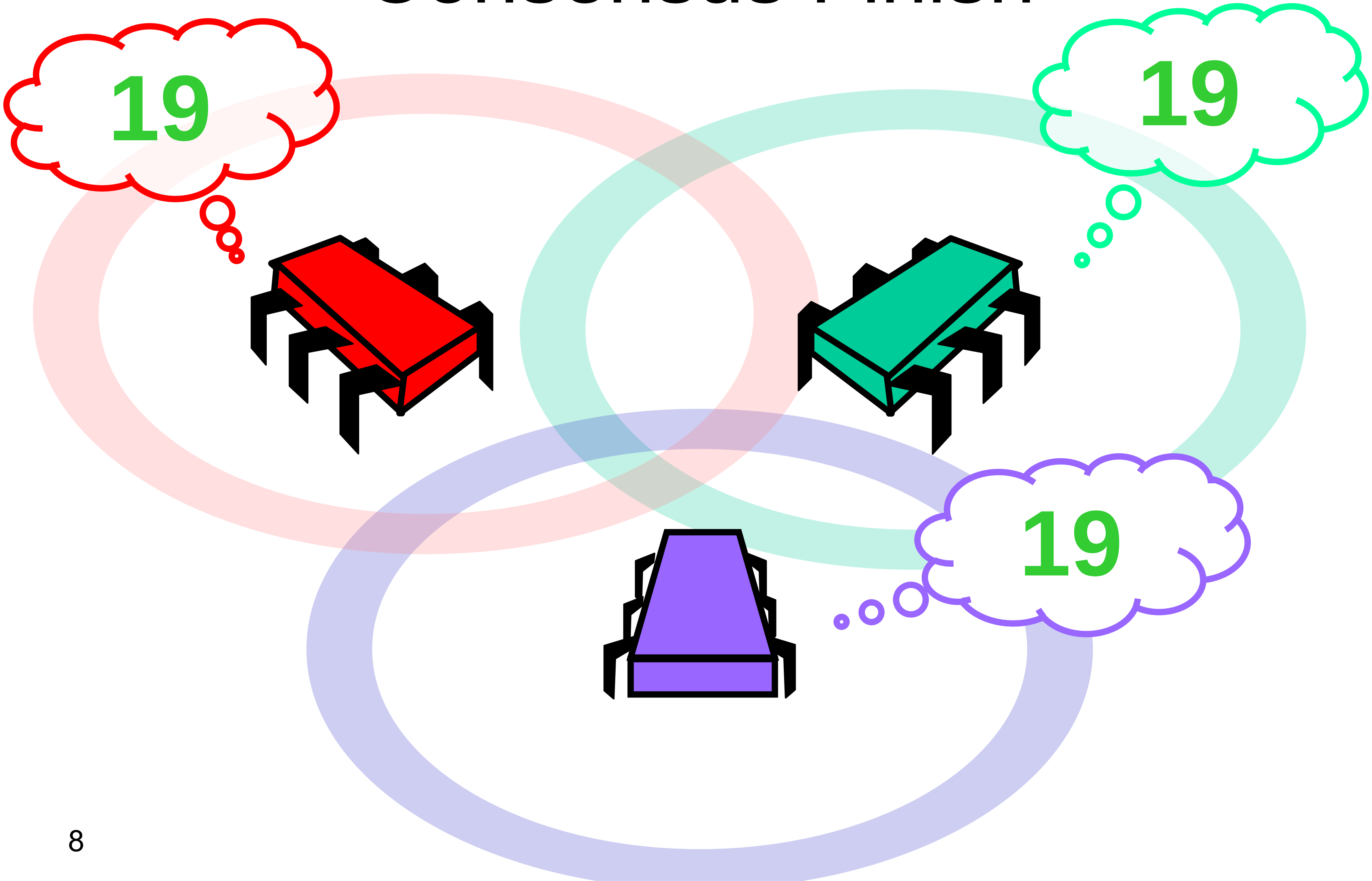
Consensus Start



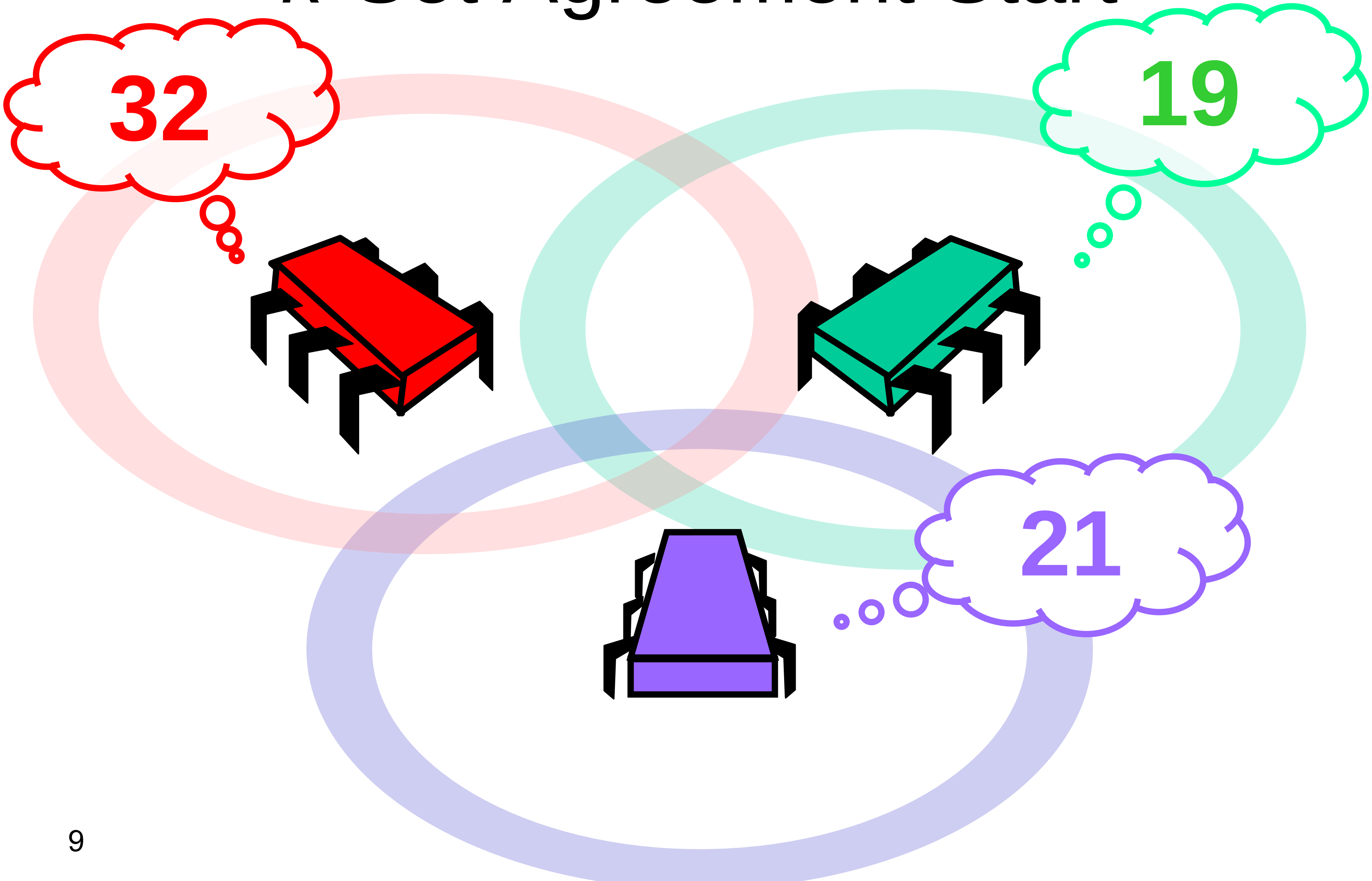
Communication



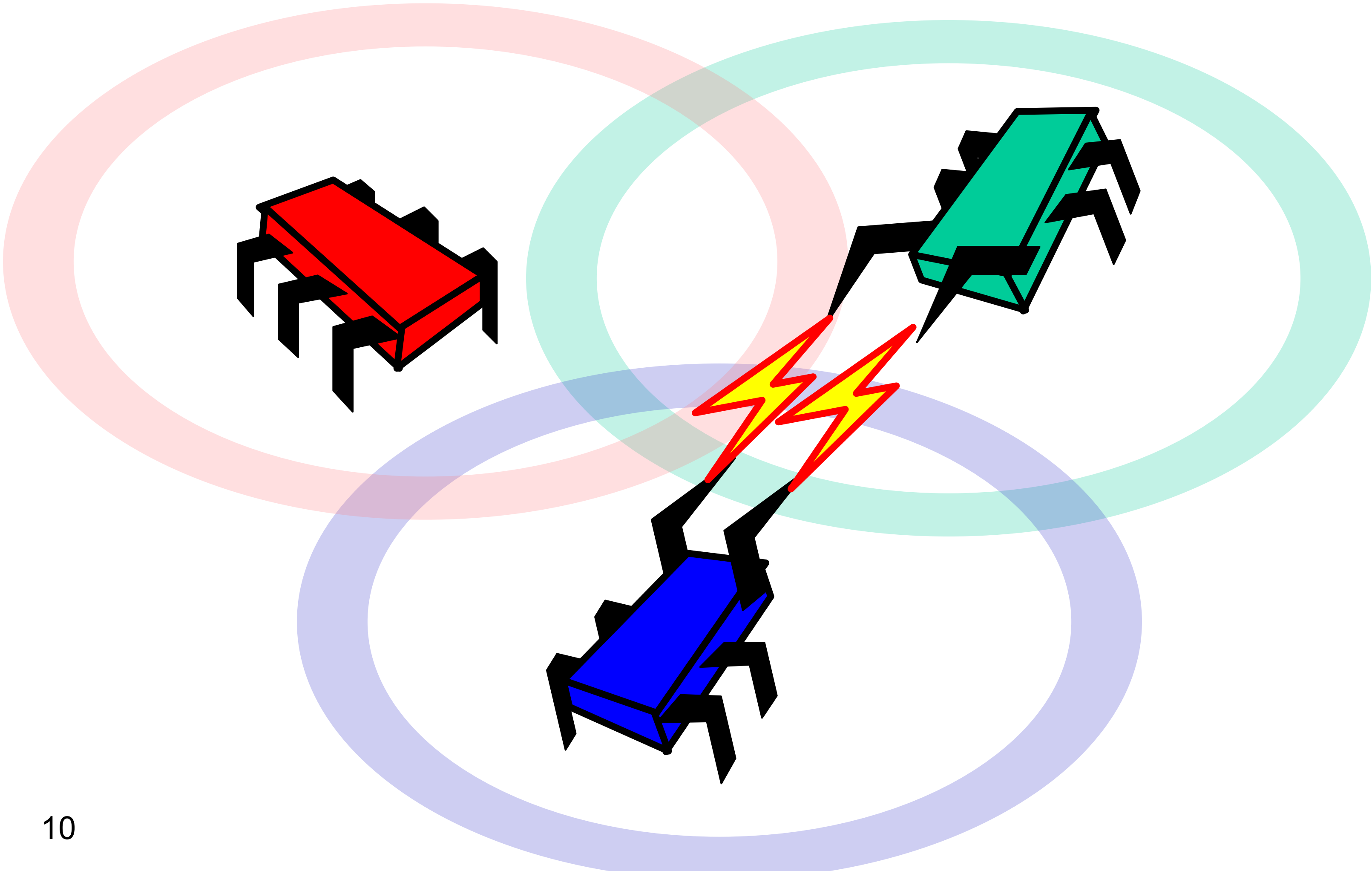
Consensus Finish



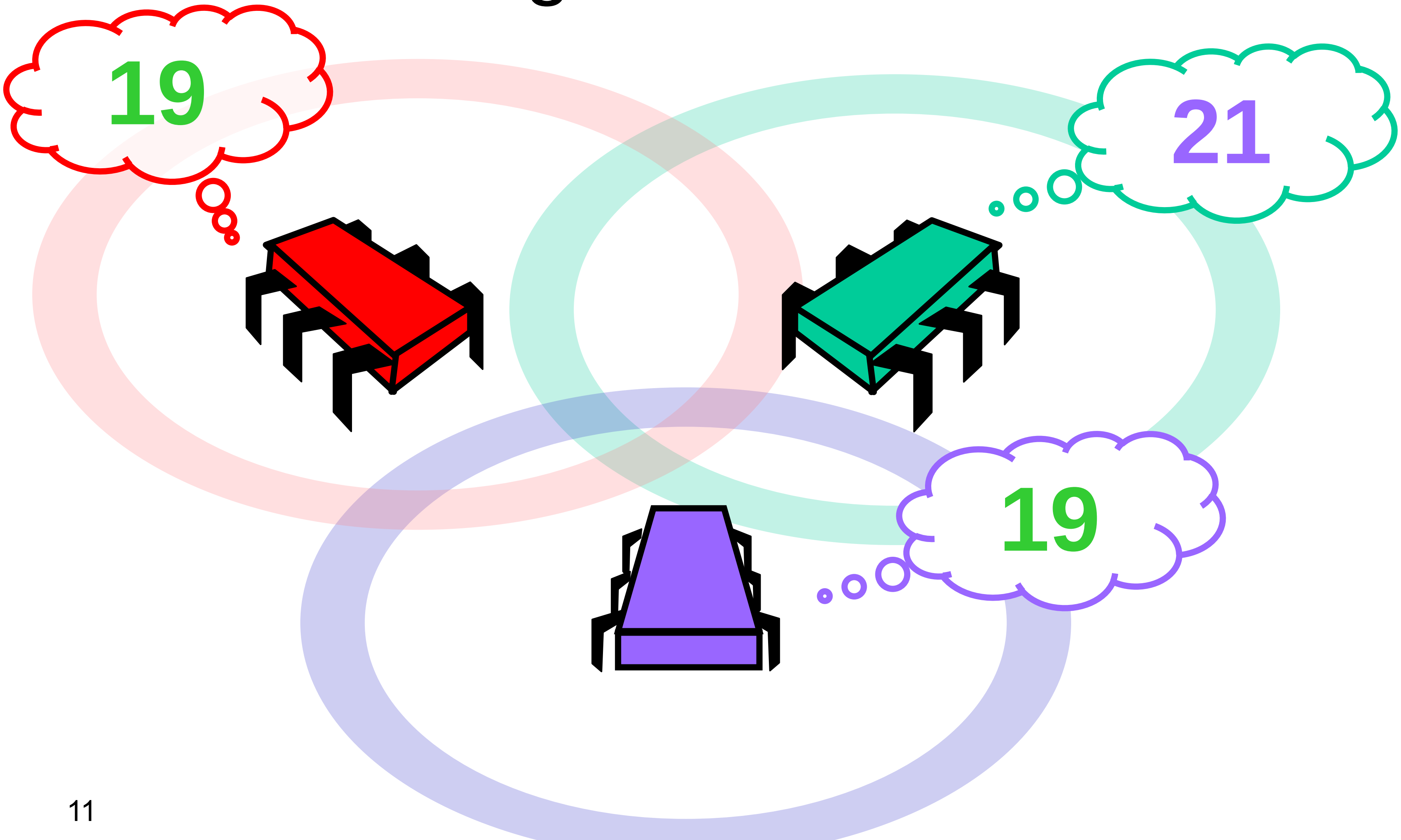
k-Set Agreement Start



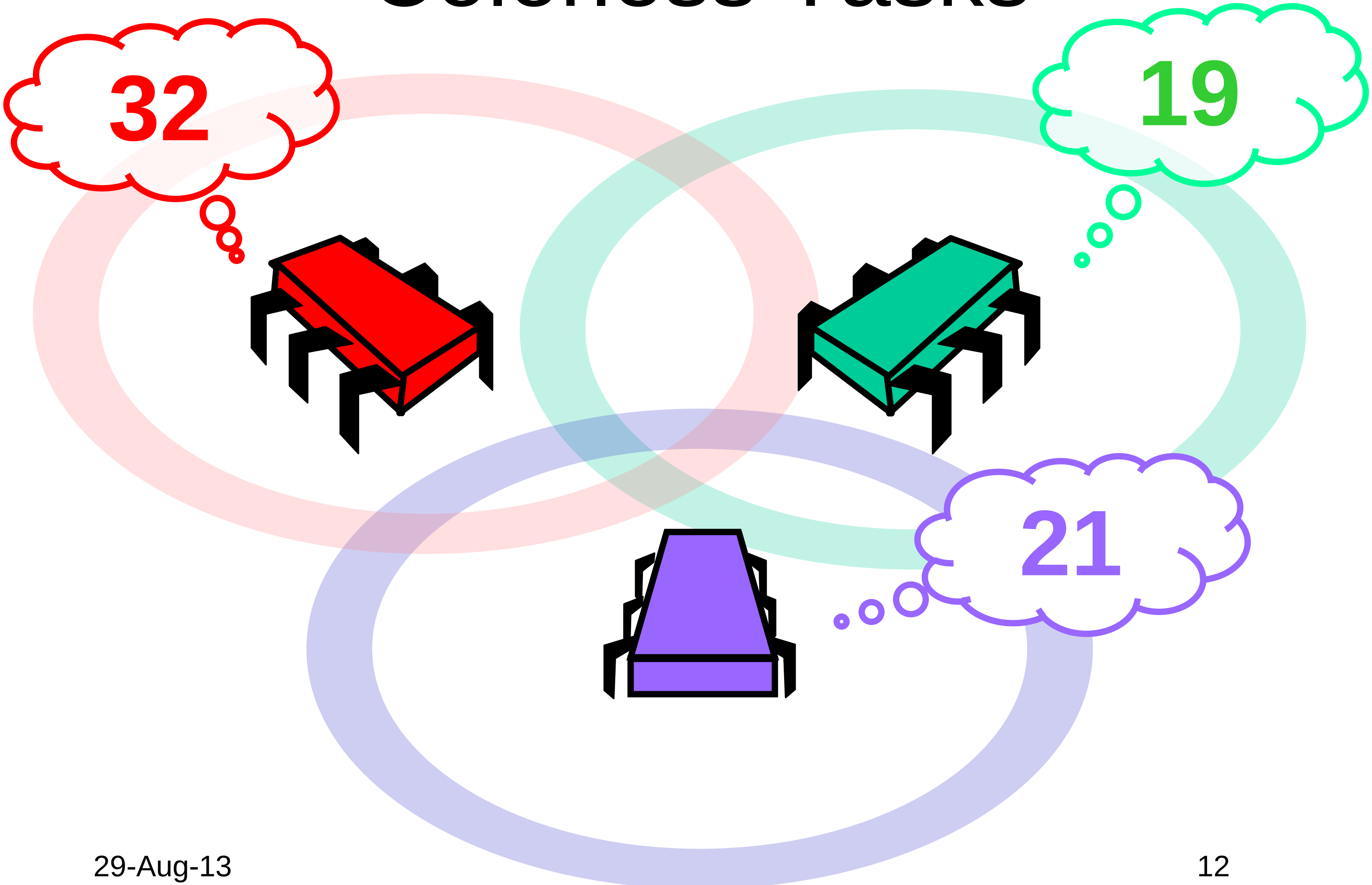
Communication



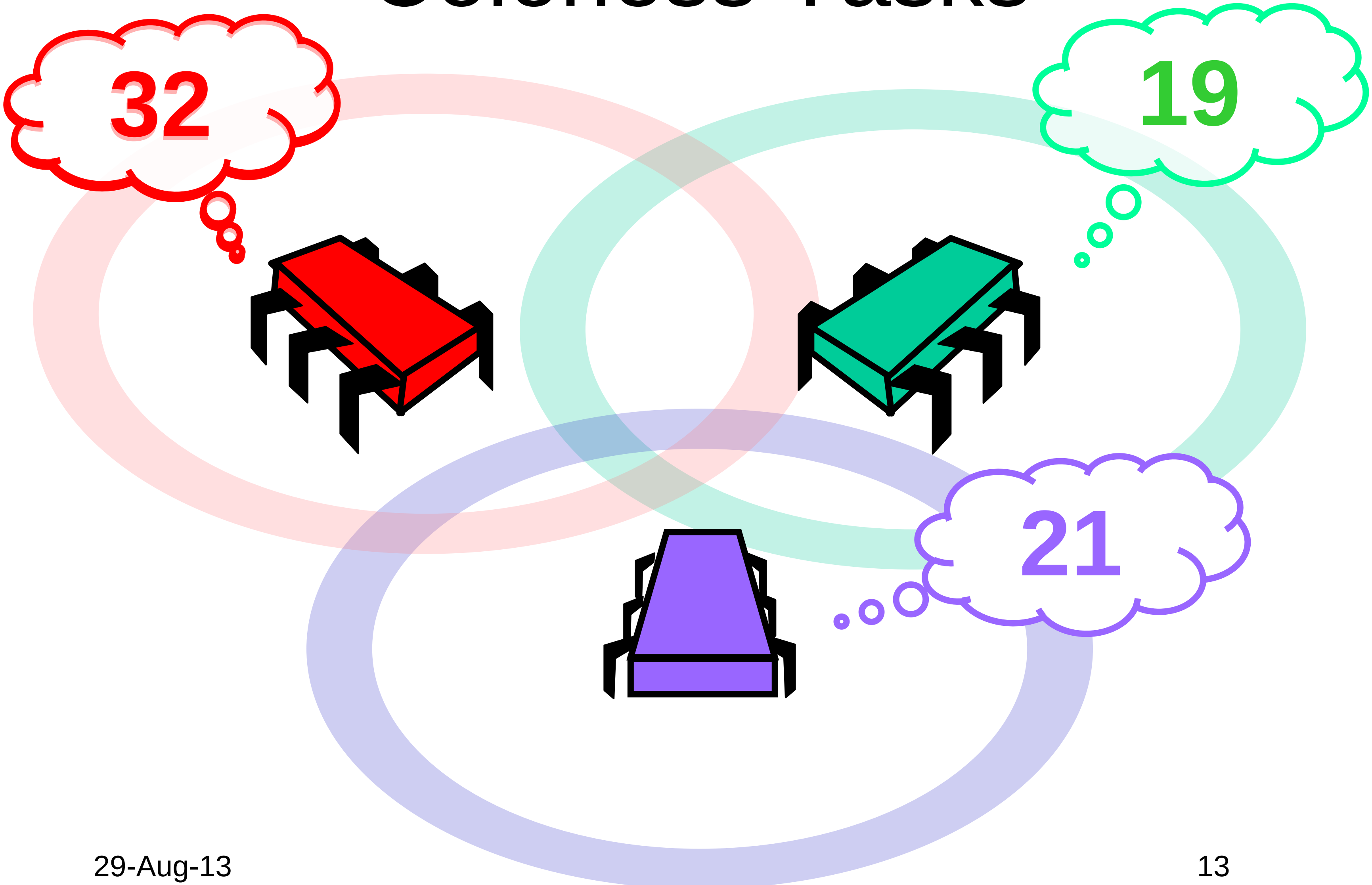
k-Set Agreement Finish



Colorless Tasks



Colorless Tasks



Colorless Tasks

The *set* of input values …

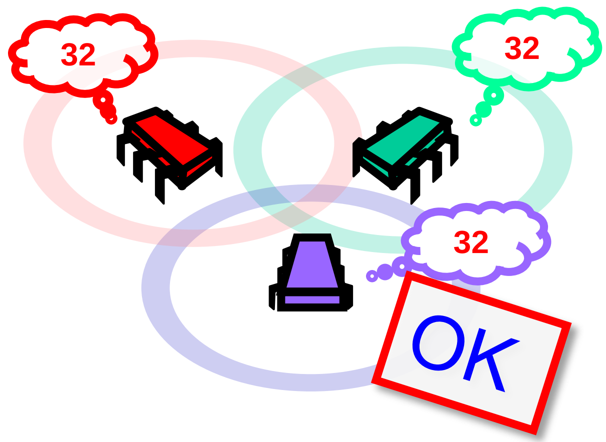
determines the *set* of output values.

Number and identities irrelevant…

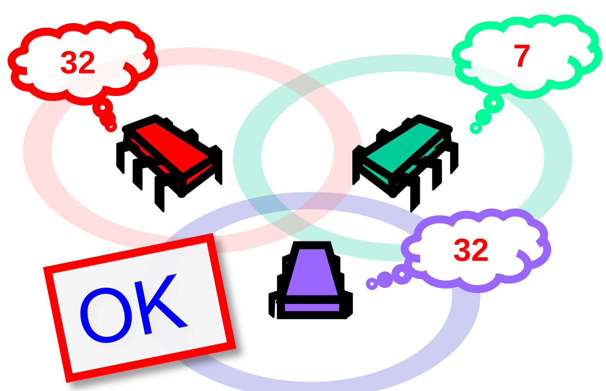
for both input and output values

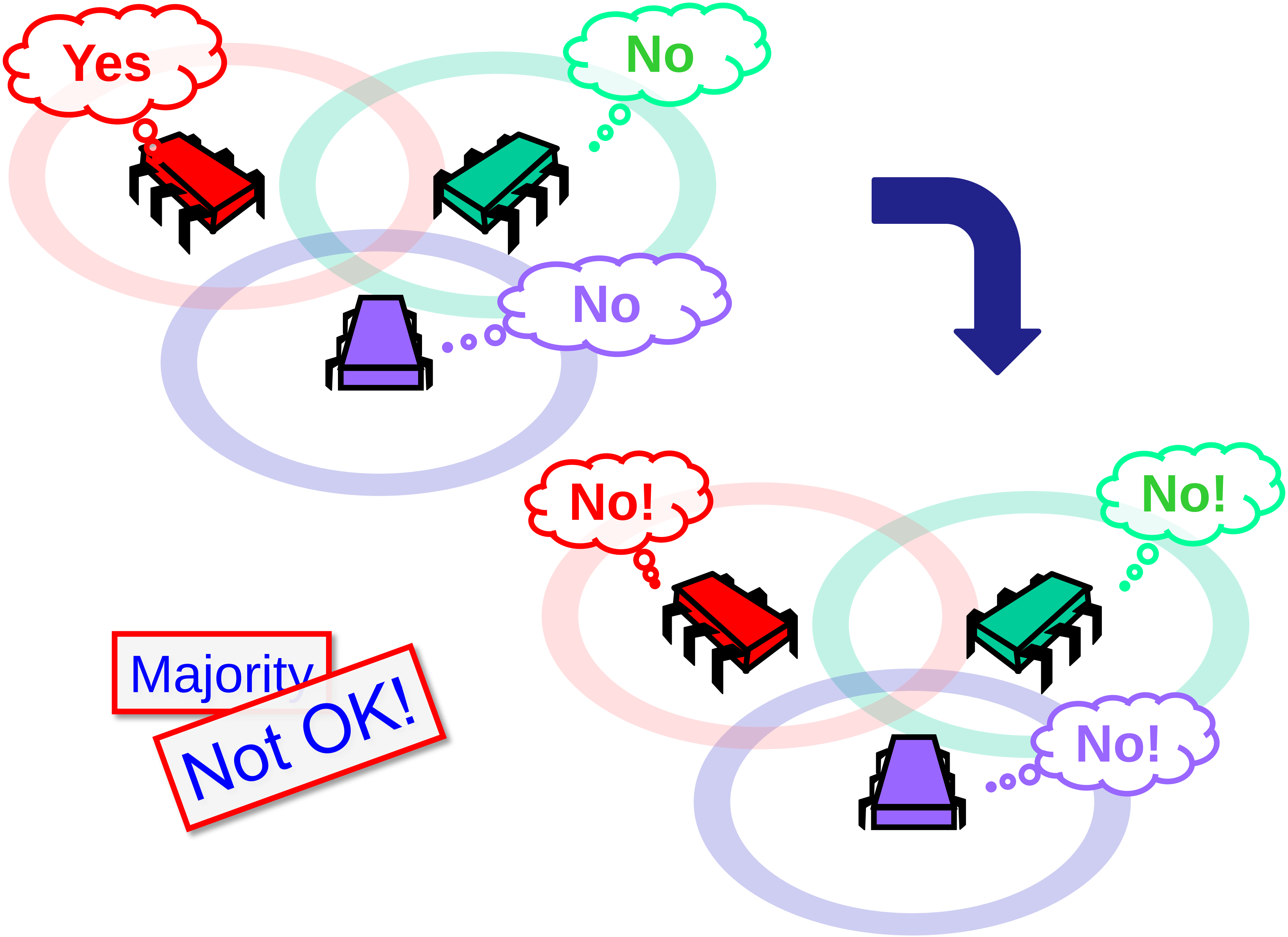
Examples

Consensus



k-set agreement





Yes
No
No
No!
No!
No!
Majority
Not OK!

Road Map

Colorless Tasks

Operational Model

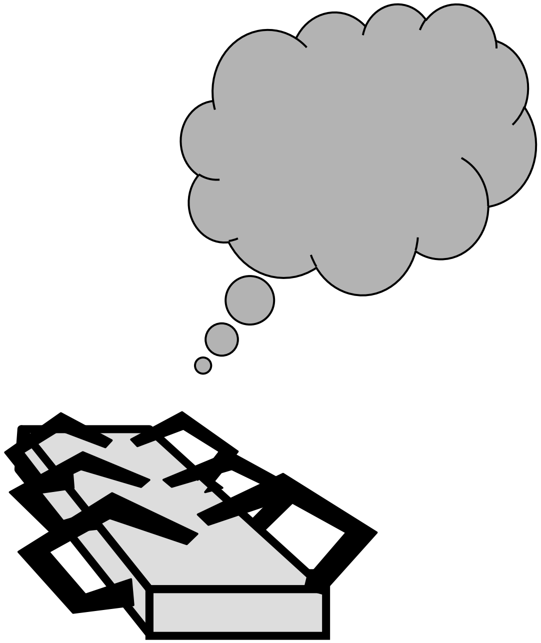
Combinatorial Model

Building Blocks

Crash Failure Solvability

Byzantine Failure Solvability

Failures



Crash failures: processes halt

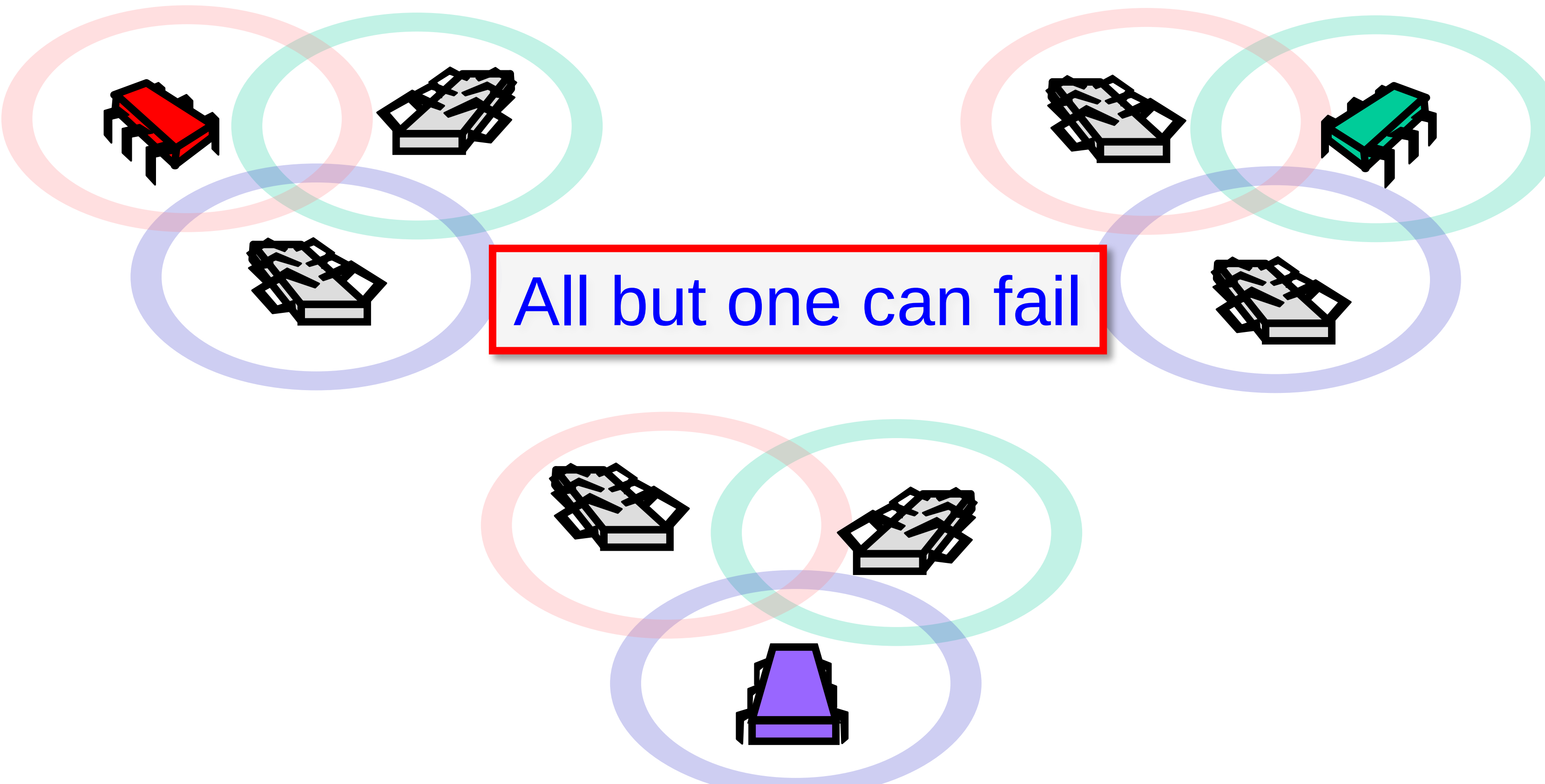
How many?

Failures

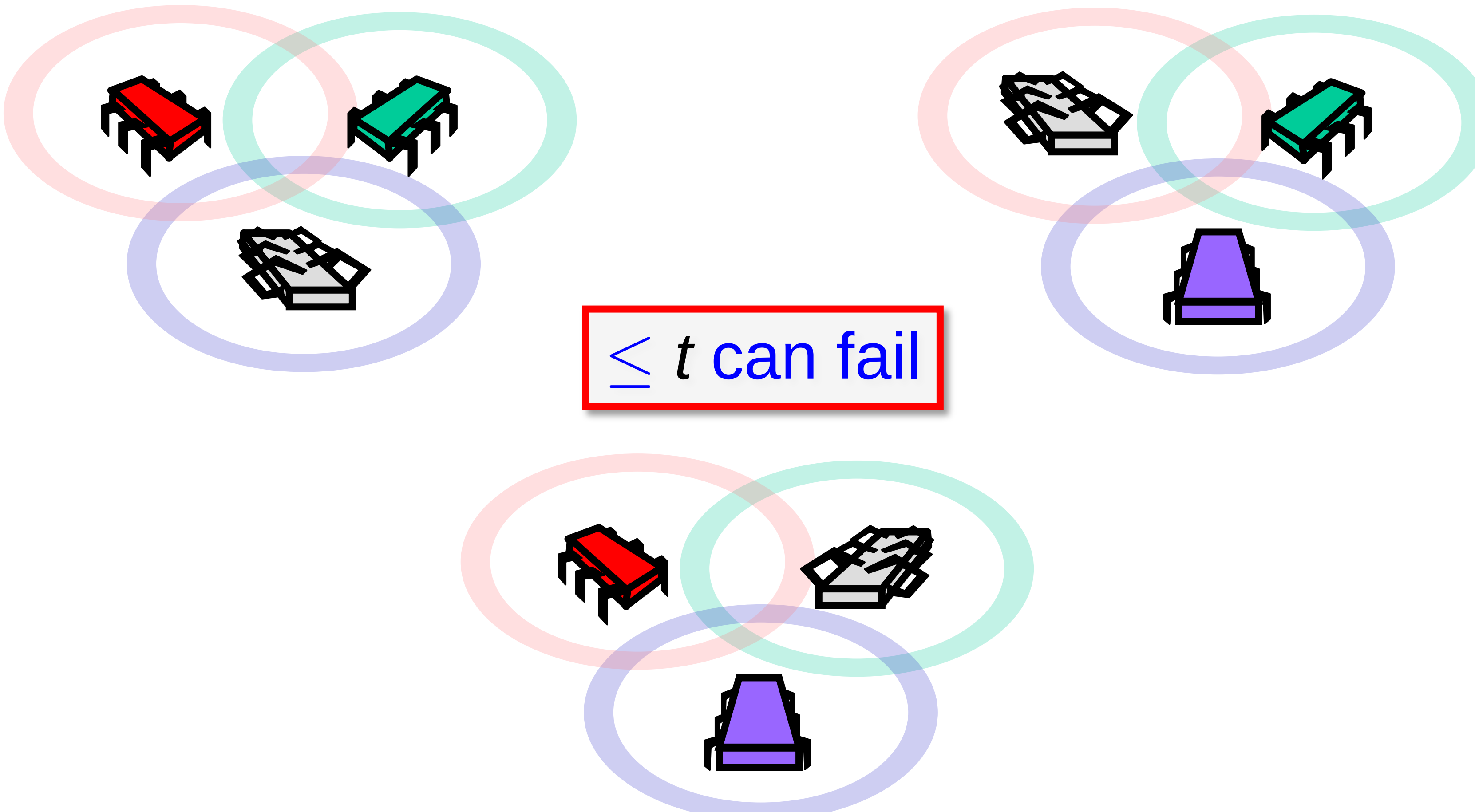
Crash failures: processes misbehave

How many?

Resilience: Wait-Free



Resilience: *t*-resilient



Road Map

Colorless Tasks

Operational Model

Combinatorial Model

Building Blocks

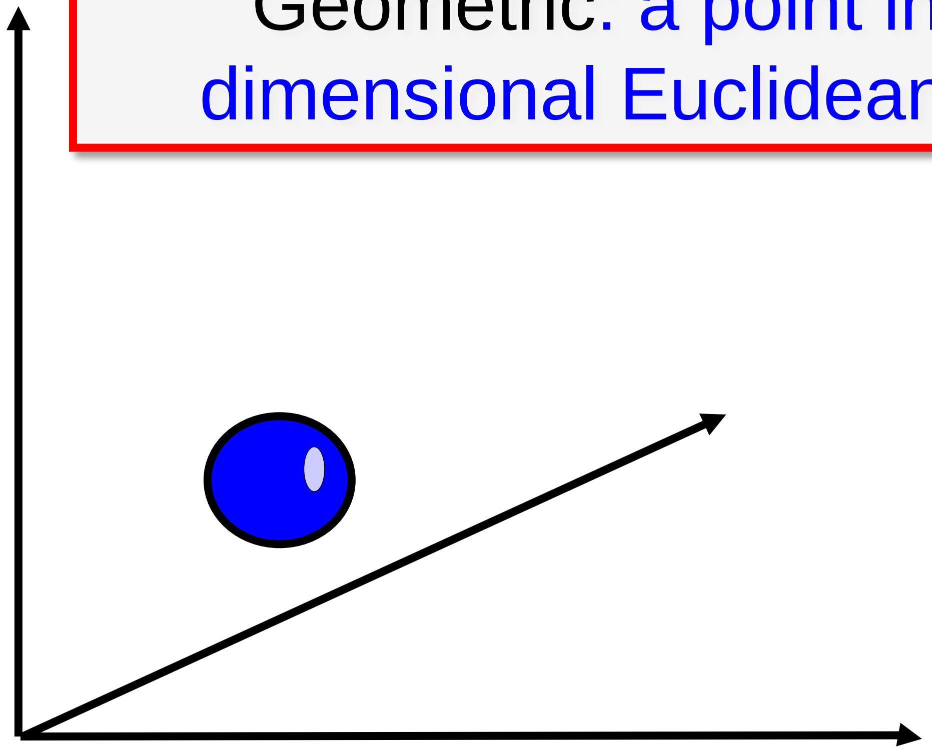
Crash Failure Solvability

Byzantine Failure Solvability

A Vertex

Combinatorial: an element of a set

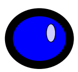
Geometric: a point in high-dimensional Euclidean Space



Simplexes

Combinatorial: a set of vertexes.

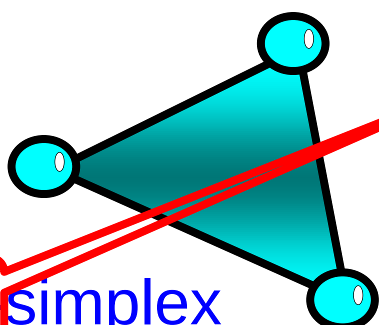
Geometric: convex hull of points in general position



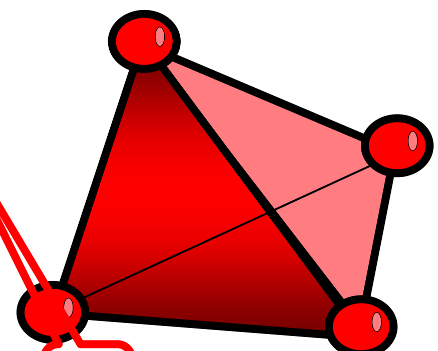
0-simplex

1-simplex

dimension

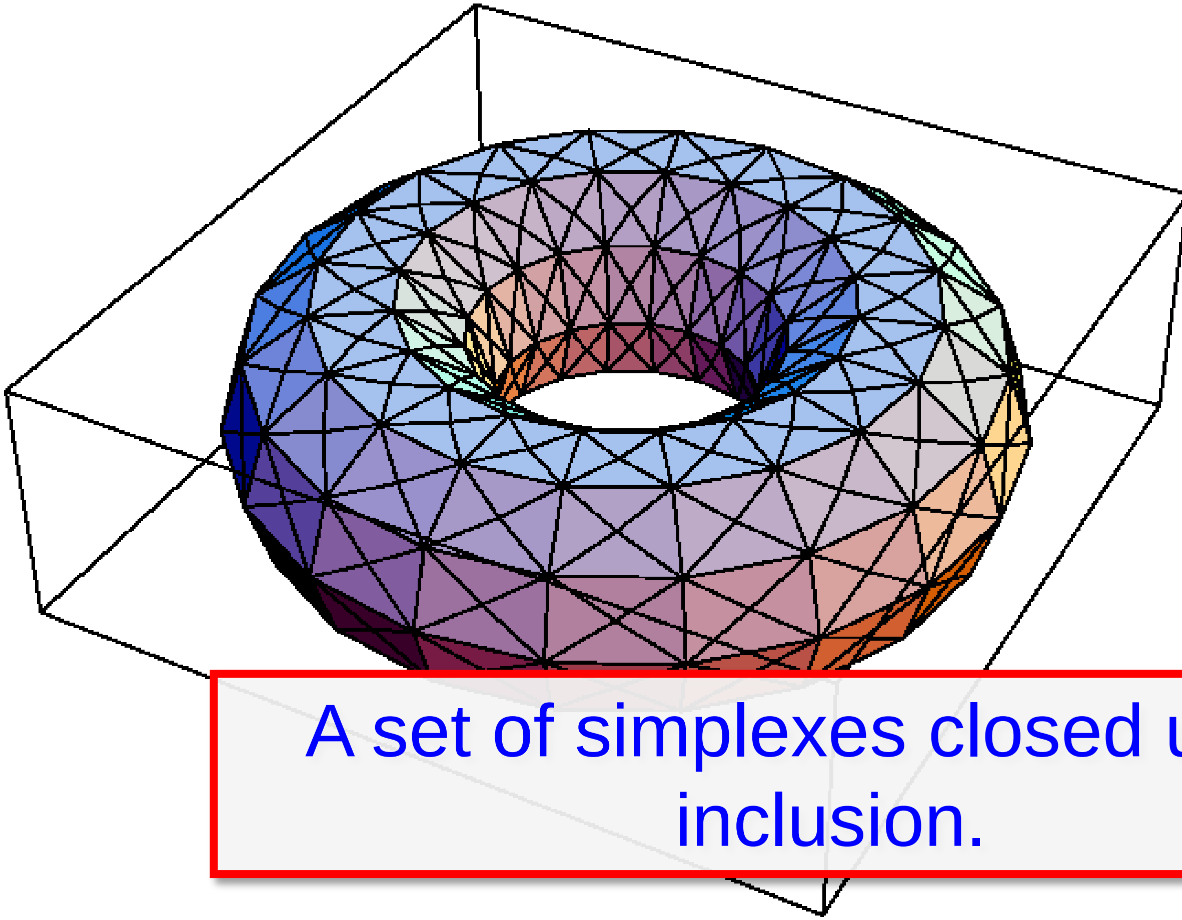


2-simplex

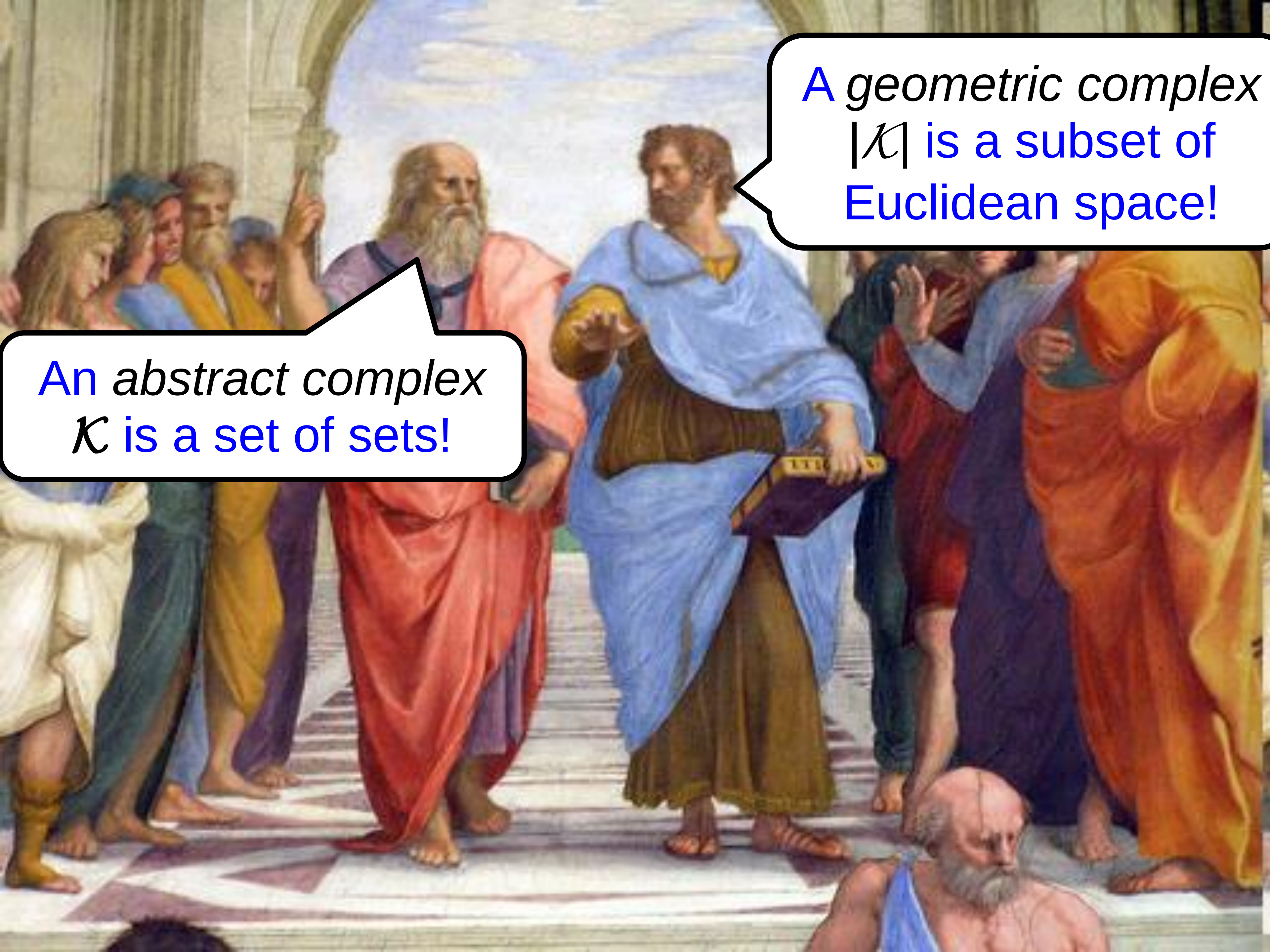


3-simplex

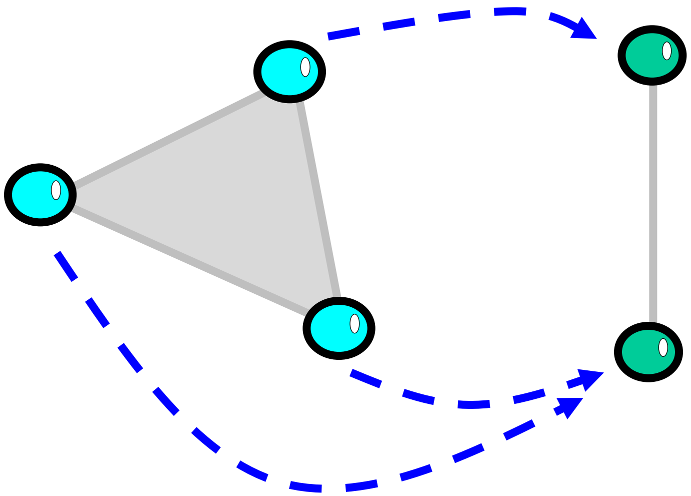
Simplicial Complex



A set of simplexes closed under inclusion.

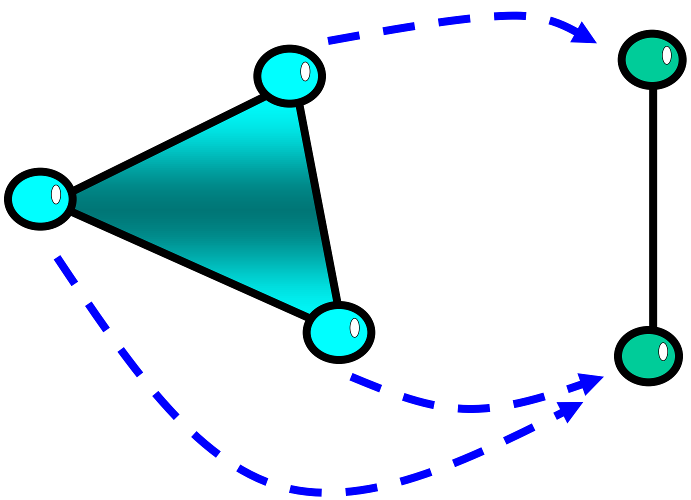

A *geometric complex* $|\mathcal{K}|$ is a subset of Euclidean space!
An *abstract complex* $\mathcal{K}$ is a set of sets!

Simplicial Maps



Vertex-to-vertex map …

Simplicial Map

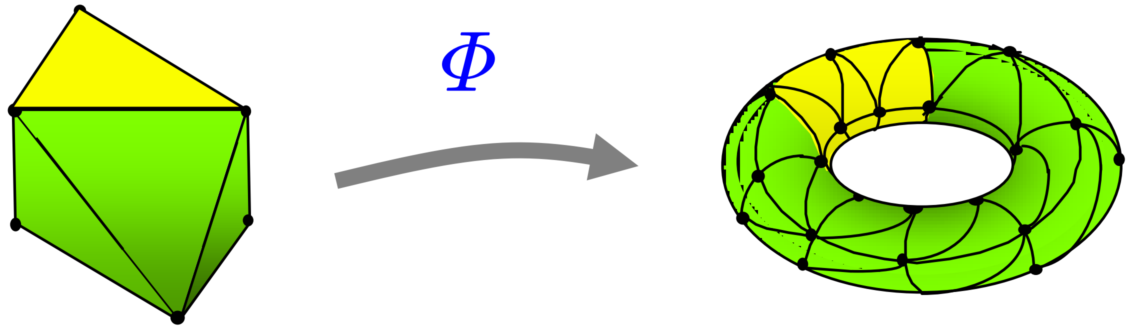


Vertex-to-vertex map …

that sends simplexes to simplexes

piece-wise linear map on geometric simplexes

Carrier Map



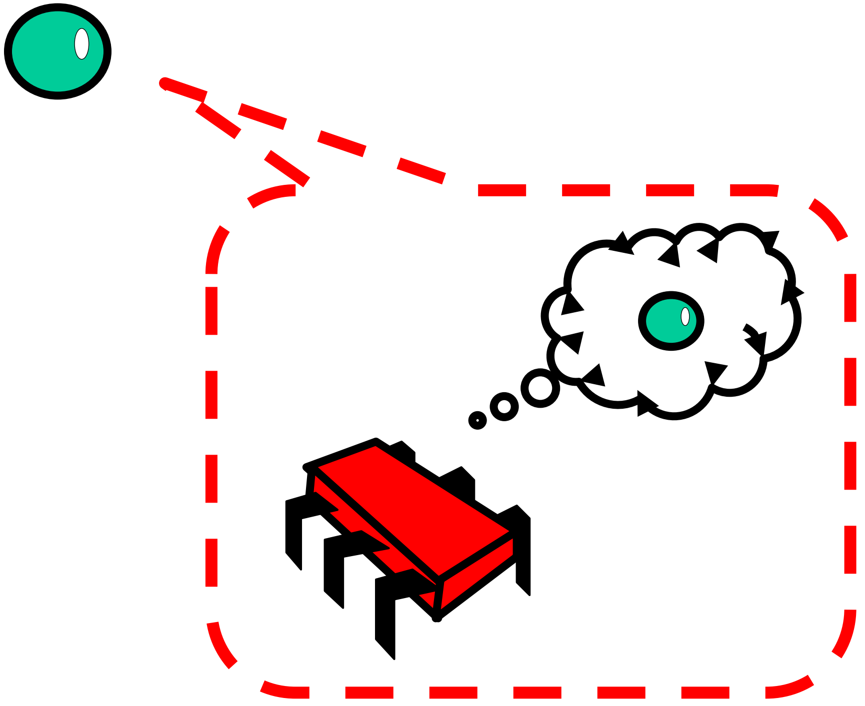
Maps simplex …

to subcomplex.

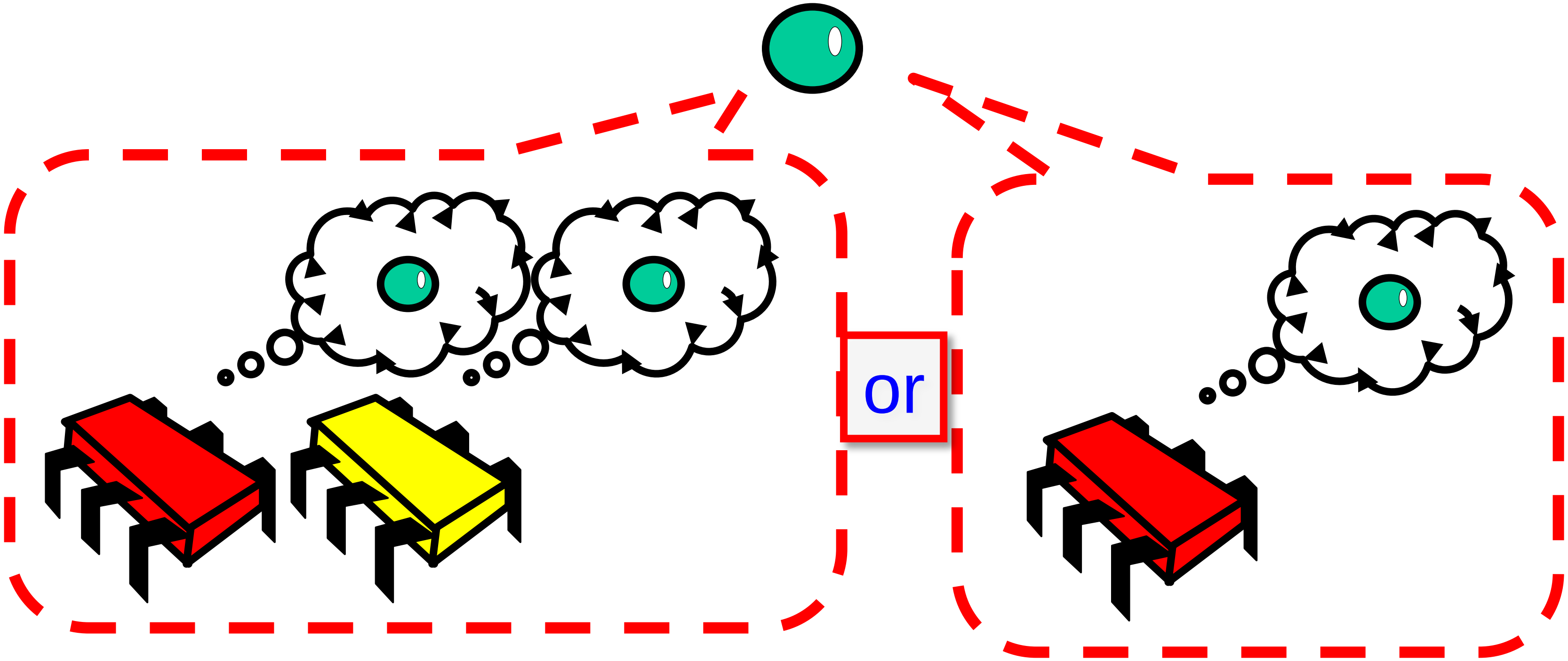
Monotonic: if $\sigma \subset \tau$ then $\Phi(\sigma) \subset \Phi(\tau)$

Always OK to discard inputs

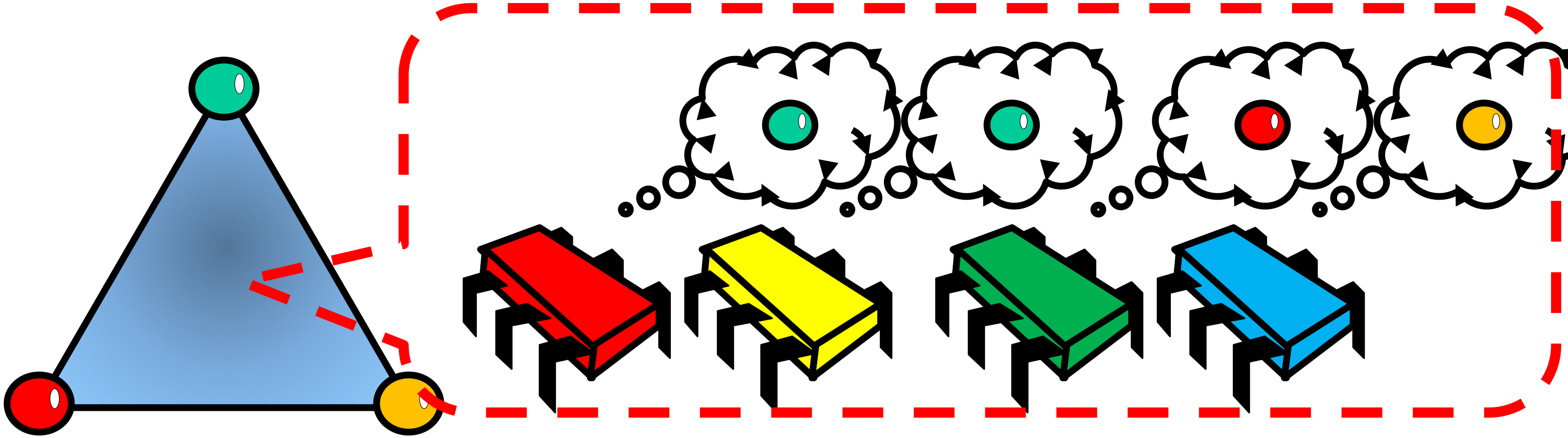
Vertex = Input or Output Value



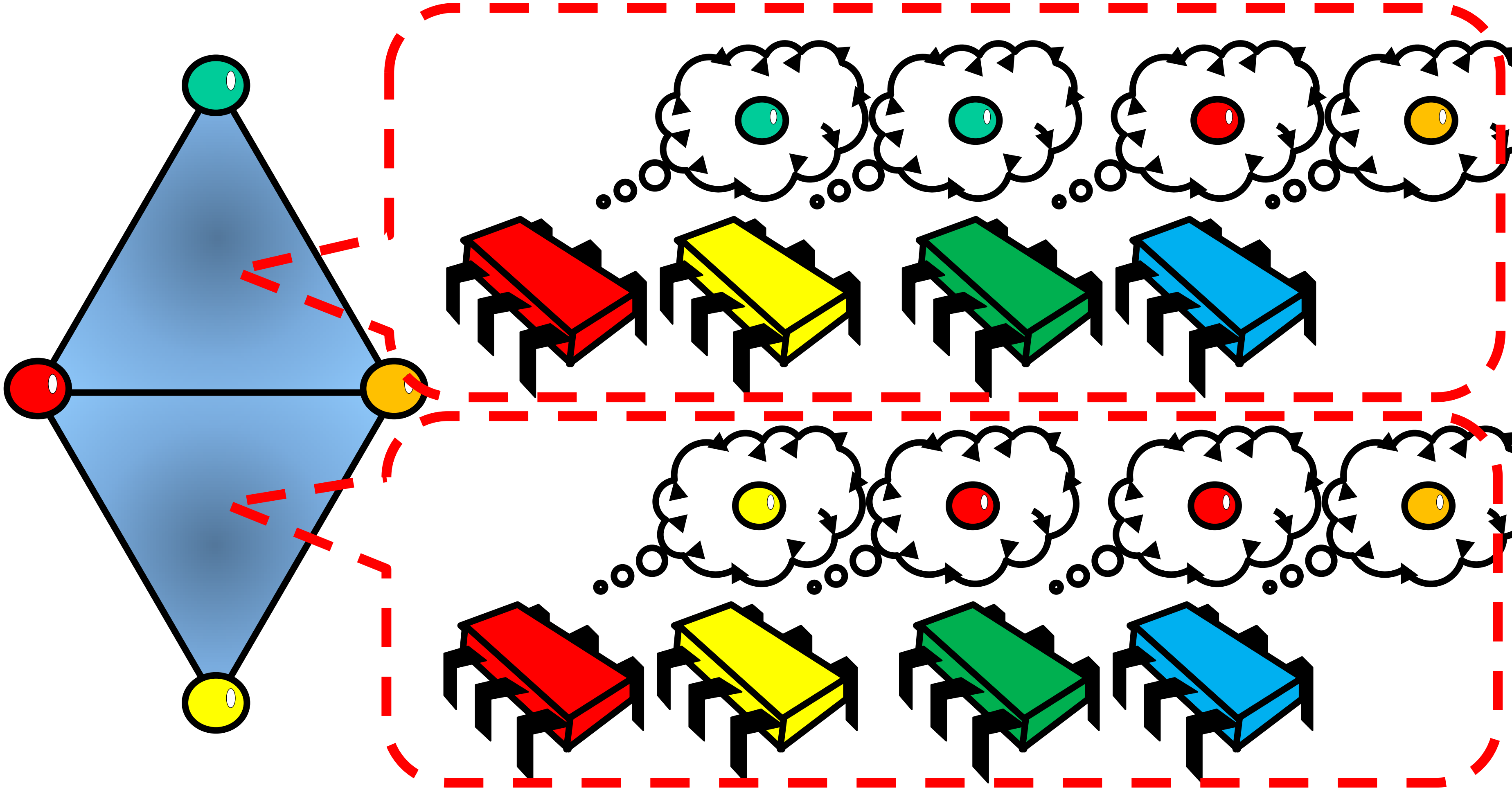
Vertex = Input or Output Value



Simplex = Compatible Values

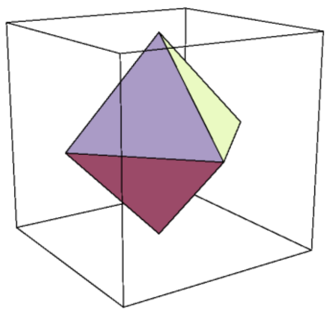


Simplex = Compatible Values

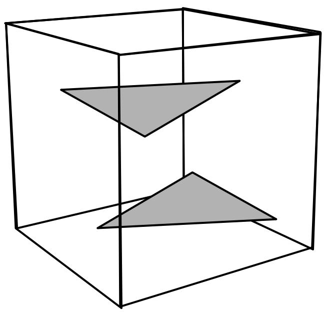


Task Specification

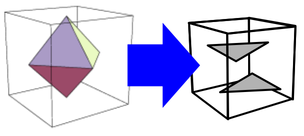
$$(\mathcal{I}, \mathcal{O}, \Delta)$$



Input complex

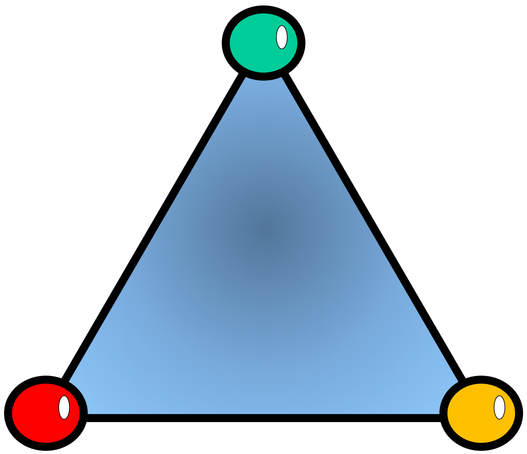


Output complex

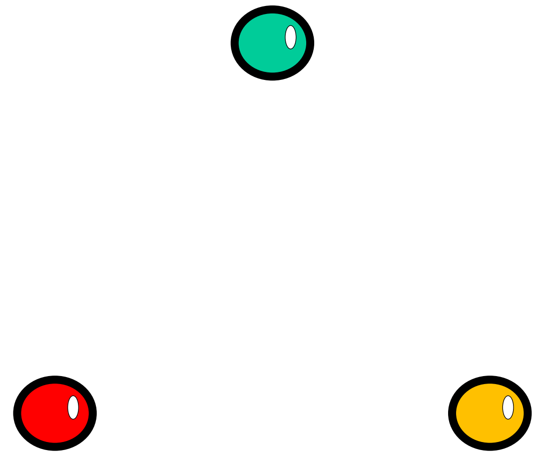


Carrier map

Consensus

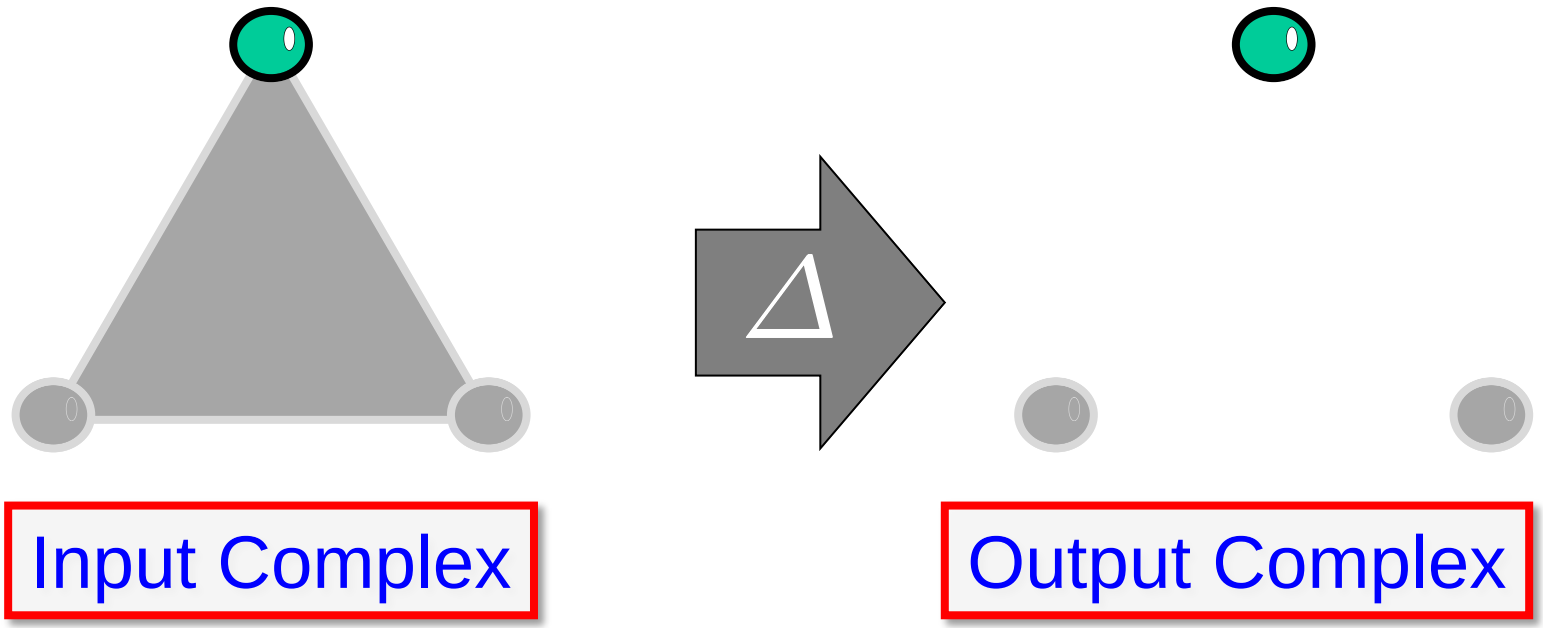


Input Complex

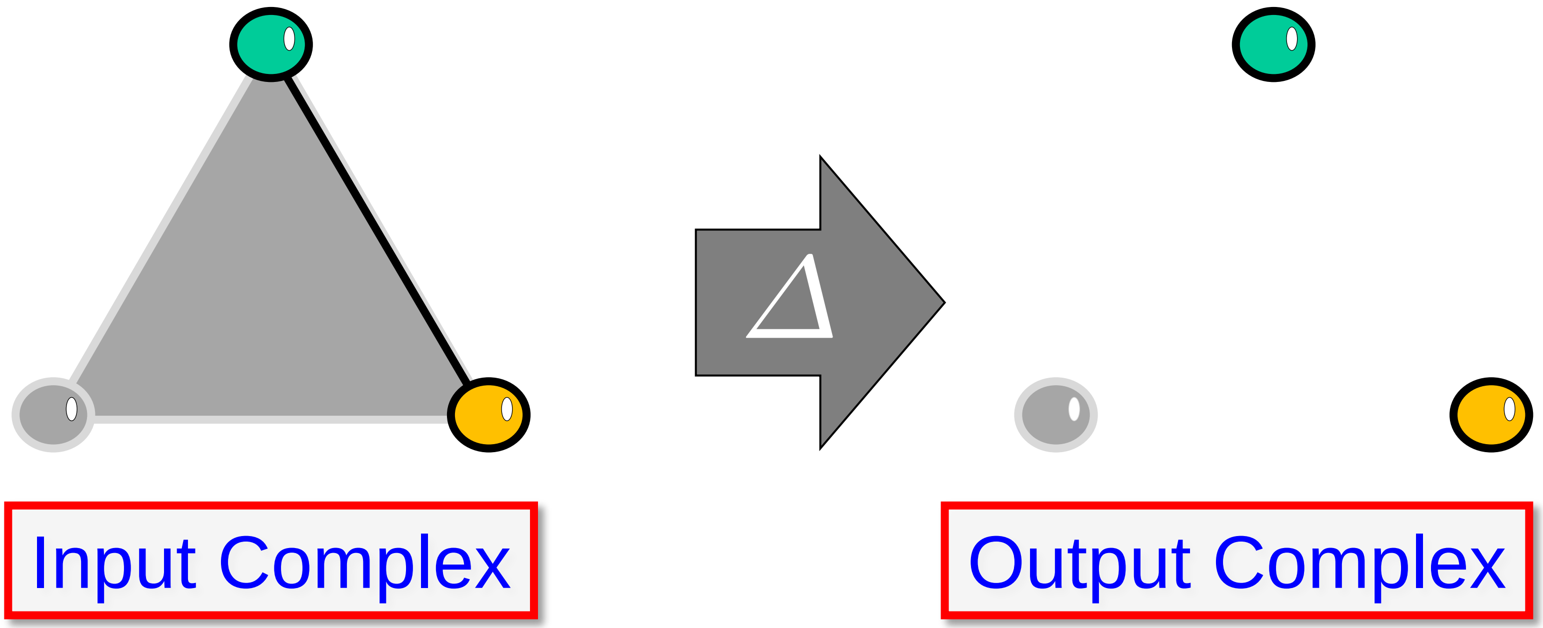


Output Complex

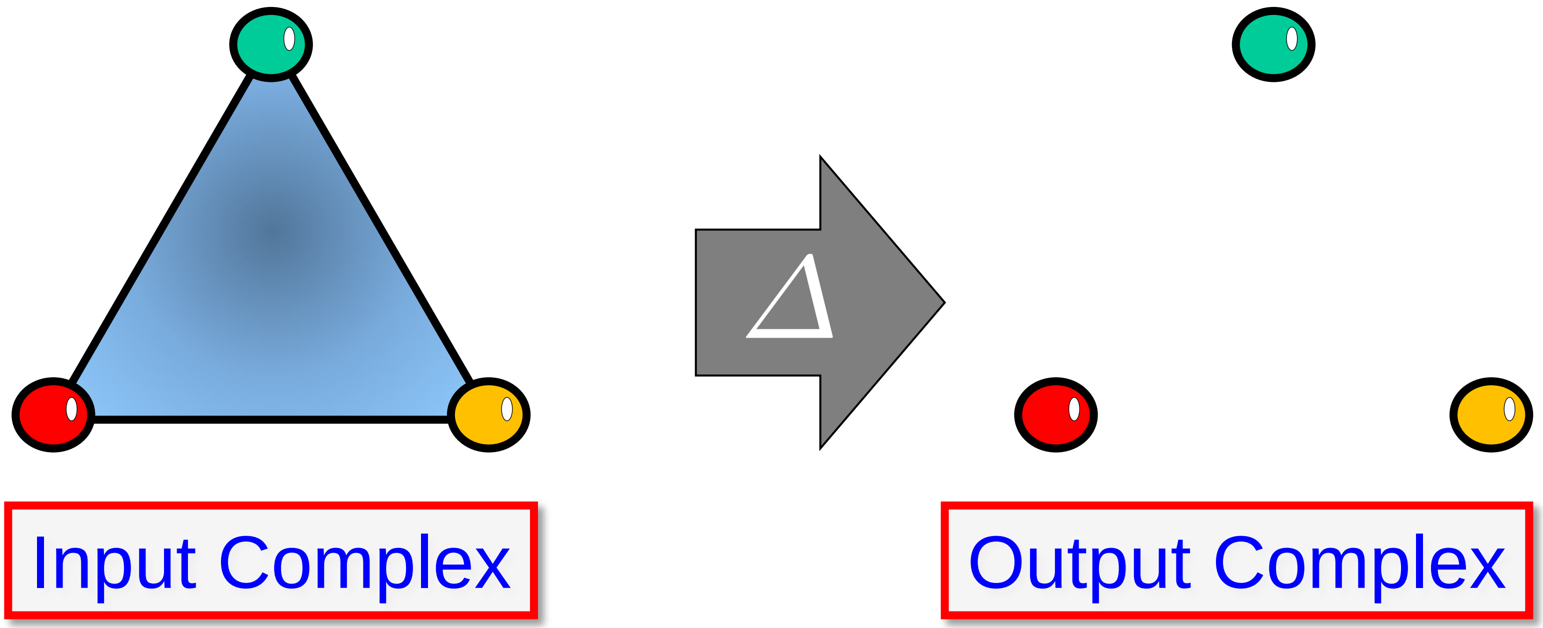
Carrier Map



Consensus



Consensus



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 += messages received;
    }
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: possible view

Full information: messages sent & received

Simplex: compatible set of views

Each execution defines a simplex

Road Map

Colorless Tasks

Operational Model

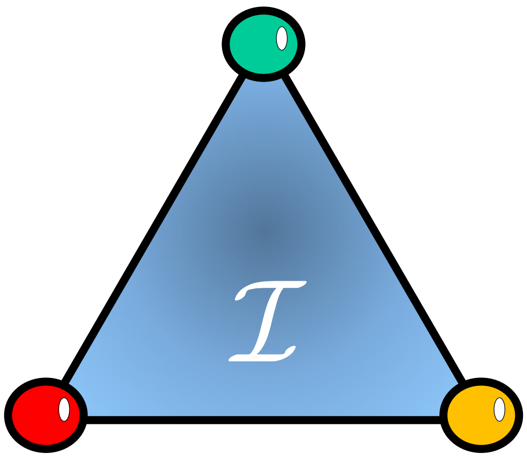
Combinatorial Model

Building Blocks

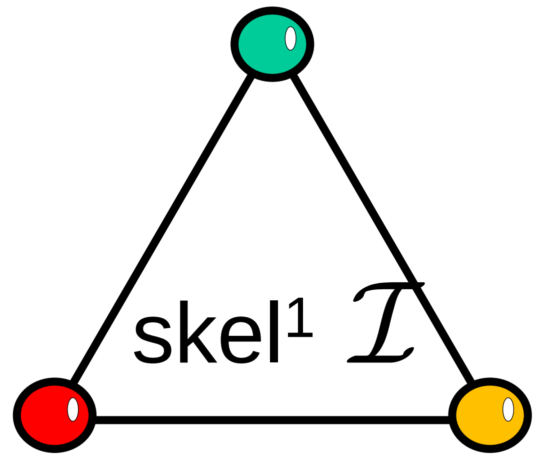
Crash Failure Solvability

Byzantine Failure Solvability

k-Set Agreement



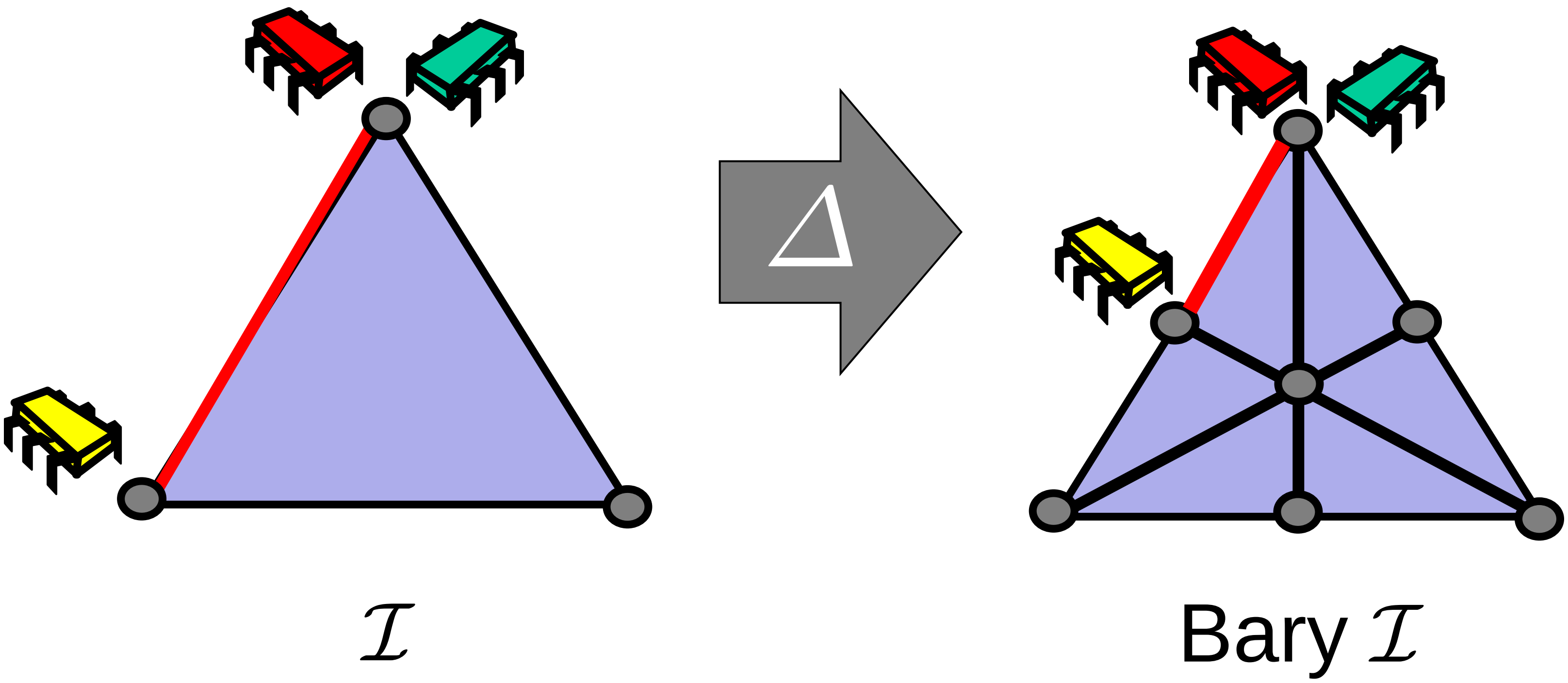
Input Complex



Output Complex

Δ is *k*-skeleton operator

Barycentric Agreement



Δ is *barycentric subdivsion* operator

Closely related to *snapshot*

k-Set Agreement in Crash Failure Model

Theorem

There exists a t-resilient (t+1) -set agreement protocol ($\mathcal{I}$, skel^t $\mathcal{I}$, skel$^t(\cdot)$)

Proof

Broadcast value, wait for all but t values, decide least one.

k-Set Agreement in Byzantine Failure Model

Theorem

There exists a t-resilient $(t+1)$-set agreement protocol

$(\mathcal{I}, \mathrm{skel}^t\, \mathcal{I}, \mathrm{skel}^t(\cdot))$

Iff $n+1 > t \cdot (\dim \mathcal{I} + 2)$

new!

Cost of Byzantine failures!

The Necessary Part

Byzantine processes cannot influence decisions!

Non-Faulty processes cannot “believe” value with < t+1 witnesses

If $n+1 > t \cdot (\dim \mathcal{I} + 2)$ then some value has at least t+1 witnesses

The Sufficient Part

Variation of *reliable broadcast* protocol of [Bracha 87] and [Shrikanth &Toueg 87]

Non-Faulty processes agree on values sent by others, even faulty processes.

If one non-faulty process receives a message, so do the others (liveness)

Barycentric Agreement in Crash Failure Model

Theorem

There is a *t*-resilient barycentric agreement protocol
$(\mathcal{I}, \mathrm{Bary}^N \mathcal{I}, \mathrm{Bary}^N(\cdot))$

Proof

Variation of *stable vectors* algorithm of
[Attiya et al. 90]

Barycentric Agreement in Byzantine Failure Model

Theorem

There is a *t*-resilient barycentric agreement protocol

$(\mathcal{I}, \mathrm{bary}\mathcal{I}^t\, \mathcal{I}, \mathrm{bary}\mathcal{I}^t(\cdot))$

Iff $n+1 > t \cdot (\dim \mathcal{I} + 2)$

new!

Cost of Byzantine failures!

The Necessary Part

Byzantine processes cannot influence decisions!

Non-Faulty processes cannot “believe” value with < t+1 witnesses

If $n+1 > t \cdot (\dim \mathcal{I} + 2)$ then some value has at least t+1 witnesses

The Sufficient Part

Byzantine variation of *stable vectors* algorithm of [Attiya et al. 90]

Use reliable broadcast to spread values

Ignore values with fewer than t+1 witnesses …

Road Map

Colorless Tasks

Operational Model

Combinatorial Model

Building Blocks

Crash Failure Solvability

Byzantine Failure Solvability

Solvability for Crash Failures

Theorem

There is a t-resilient protocol for task

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

Iff there is a continuous map

$f\colon |\mathrm{skel}^t\, \mathcal{I}| \to |\mathcal{O}|$

carried by Δ

$f(|\sigma|) \subseteq \Delta(\sigma)$

The Necessary Part

We can model the set of process views after a full-information protocol as a *protocol complex* $\mathcal{P}$

There is a simplicial decision map

δ: $\mathcal{P} \to \mathcal{O}$ carried by Δ

WLOG, $\mathcal{P}$ is isomorphic to $\text{bary}^N \text{skel}^t \mathcal{I}$

δ: $\text{bary}^N \text{skel}^t \mathcal{I} \to \mathcal{O}$

induces a piece-wise linear map

$|\delta|: |\text{bary}^N \text{skel}^t \mathcal{I}| \to |\mathcal{O}|$

$|\delta|: |\text{skel}^t \mathcal{I}| \to |\mathcal{O}|$

carried by Δ

The Sufficient Part

f: $|\text{skel}^t\, \mathcal{I}| \to |\mathcal{O}|$
has a *simplicial approximation* for some N > 0
ϕ: $\text{bary}^N\, \text{skel}^t\, \mathcal{I} \to \mathcal{O}$

Step 1: use *t*-set agreement protocol to go from vertex of $\mathcal{I}$ to vertex of $\text{skel}^t\, \mathcal{I}$

Step 2: use repeated barycentric agreement to go from vertex of $\text{skel}^t\, \mathcal{I}$ to vertex of : $\text{bary}^N\, \text{skel}^t\, \mathcal{I}$

Step 3: from vertex $v \in$: $\text{bary}^N\, \text{skel}^t\, \mathcal{I}$, decide $\phi(v)$

Road Map

Colorless Tasks

Operational Model

Combinatorial Model

Building Blocks

Crash Failure Solvability

Byzantine Failure Solvability

Solvability for Byzantine Failures

Theorem

There is a t-resilient protocol for task $(\mathcal{I}, \mathcal{O}, \Delta)$

Iff there is a continuous map

$f\colon |\mathrm{skel}^t\, \mathcal{I}| \to |\mathcal{O}|$

~~carried by Δ~~

Iff $n+1 > t \cdot (\dim \mathcal{I} + 2)$

new!

Cost of Byzantine failures!

The Necessary Part

We can model the set of process views after a full-information protocol as a *protocol complex* $\mathcal{P}$

There is a simplicial decision map
δ: $\mathcal{P} \to \mathcal{O}$ carried by Δ

WLOG, $\mathcal{P}$ is isomorphic to $\text{bary}^N \text{skel}^t \mathcal{I}$

δ: $\text{bary}^N \text{skel}^t \mathcal{I} \to \mathcal{O}$

induces a piece-wise linear map

$|\delta|: |\text{bary}^N \text{skel}^t \mathcal{I}| \to |\mathcal{O}|$

$|\delta|: |\text{skel}^t \mathcal{I}| \to |\mathcal{O}|$

carried by Δ

The Necessary Part

We can model the set ... f process views after a full-information pro... ...s a *protocol complex* $\mathcal{P}$

Ther... ...mplicial decision map
...δ: $\mathcal{P} \to \mathcal{O}$ carried by Δ

...OG, $\mathcal{P}$ is is...rphic to skel^t bary^N $\mathcal{I}$

δ: ... $\mathcal{I} \to \mathcal{O}$
induces a piece-wi...
$|\delta|$: $|\text{skel}^t \text{ bary}^N \mathcal{I}| \to |\mathcal{O}|$
$|\delta|$: $|\text{skel}^t \mathcal{I}| \to |\mathcal{O}|$
carried by Δ

Basic Idea is the same

Technical details more involved

The Sufficient Part

f: $|\mathrm{skel}^t \mathcal{I}| \to |\mathcal{O}|$
has a *simplicial approximation* for some N > 0
ϕ: $\mathrm{bary}^N\ \mathrm{skel}^t\ \mathcal{I} \to \mathcal{O}$

Step 1: use *t*-set agreement protocol to go from vertex of $\mathcal{I}$ to vertex of $\mathrm{skel}^t\ \mathcal{I}$

Step 2: use repeated barycentric agreement to go from vertex of $\mathrm{skel}^t\ \mathcal{I}$ to vertex of : $\mathrm{bary}^N\ \mathrm{skel}^t\ \mathcal{I}$

Step 3: from vertex $v \in$: $\mathrm{bary}^N\ \mathrm{skel}^t\ \mathcal{I}$, decide $\phi(v)$

The Sufficient Part

f: $|\text{skel}^t$ ... $| \to |\mathcal{O}|$
has a *simplicial* ... *ation* for some N > 0
... $\text{bary}^N \mathcal{I} \to \mathcal{O}$

S... use t-set agreement protocol to
go from vert... of $\mathcal{I}$ to vertex of $\text{skel}^t \mathcal{I}$

Step 2: use repea... agreement
to go from vertex of ...
to vertex of : $\text{skel}^t\ \text{bary}^N \mathcal{I}$

Step 3: from vertex $v \in$: $\text{skel}^t\ \text{bary}^N \mathcal{I}$,
decide $\phi(v)$

Basic Idea is the same

Technical details more involved

Conclusions

Many have looked at specific tasks …

consensus

approximate agreement

k-set agreement

Often with much weaker validity!

First to look at general (colorless) tasks …

First to characterize what can and can't be solved

Conclusions

The *language* of combinatorial topology (*vertex, simplex, skeleton, simplicial map* …) allows us to *state* and *prove* such results succinctly

Important to exploit the duality of combinatorial and continuous model (such as *simplicial approximation*)

Here, we did not need “advanced” concepts like connectivity, but they are needed elsewhere, such as the synchronous model …

Open Problems

Colored tasks?

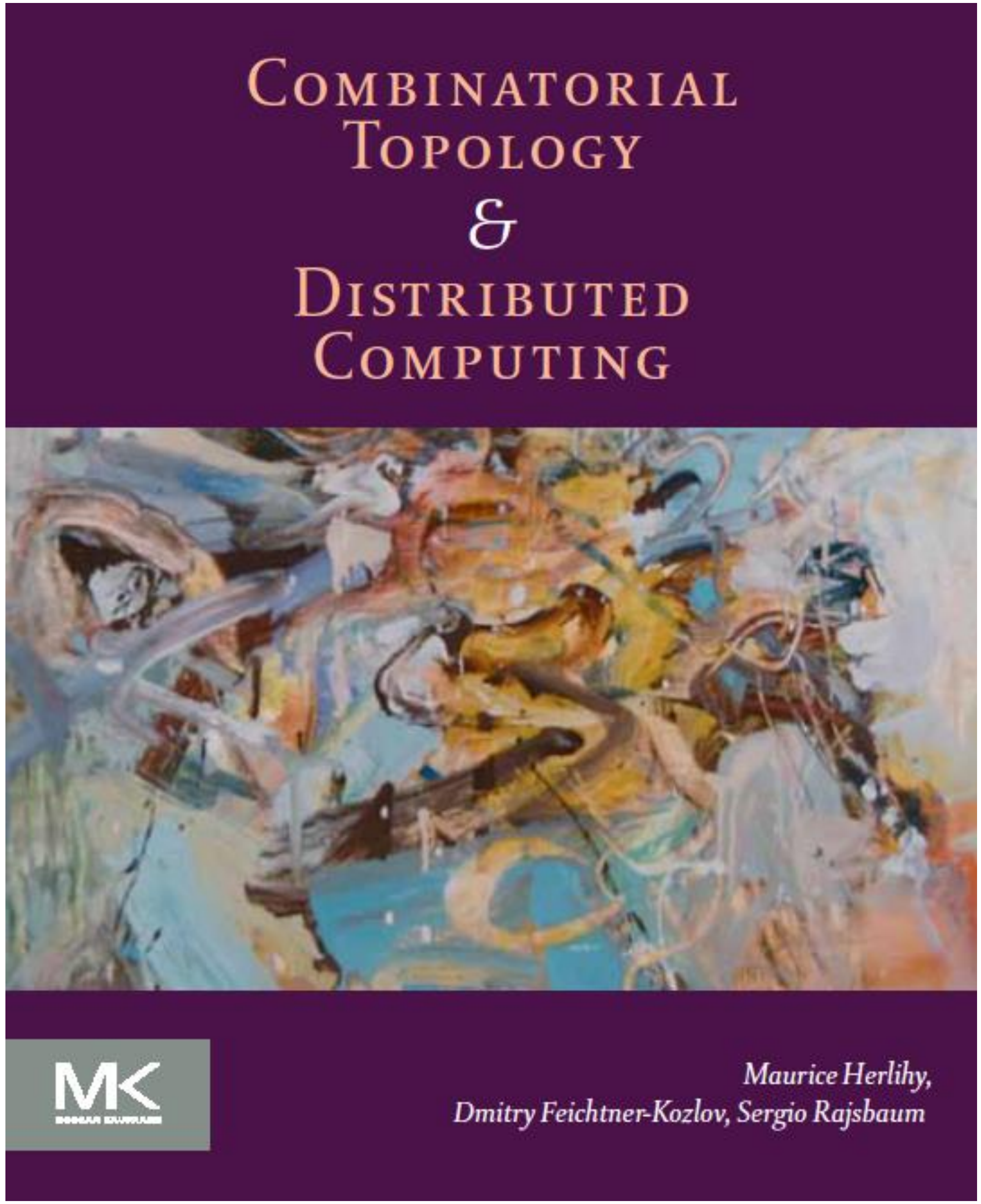
Complexity?

"Rational" adversaries?

Mechanism design?

Long-lived computations?

Randomized?


COMBINATORIAL TOPOLOGY
&
DISTRIBUTED COMPUTING
Maurice Herlihy,
Dmitry Feichtner-Kozlov, Sergio Rajsbaum