Combinatorial Structures for Distributed Computing

Class 2: Asynchronous Computability Theorem

Petr Kuznetsov, Telecom ParisTech

Kyoto University, 2018

Distributed tasks

- T task, a one-shot distributed function (I,O,Δ):
  - ✓Set of input vectors I
  - ✓Set of output vectors O
  - ✓Task specification Δ: I ➔ 2^{O}
- A task T is read-write solvable if there is a read-write algorithm that ensures, for every input vector I in I:
  - ✓Every correct process eventually outputs a value (decides)
  - ✓The output vector O$\in\Delta$(I)

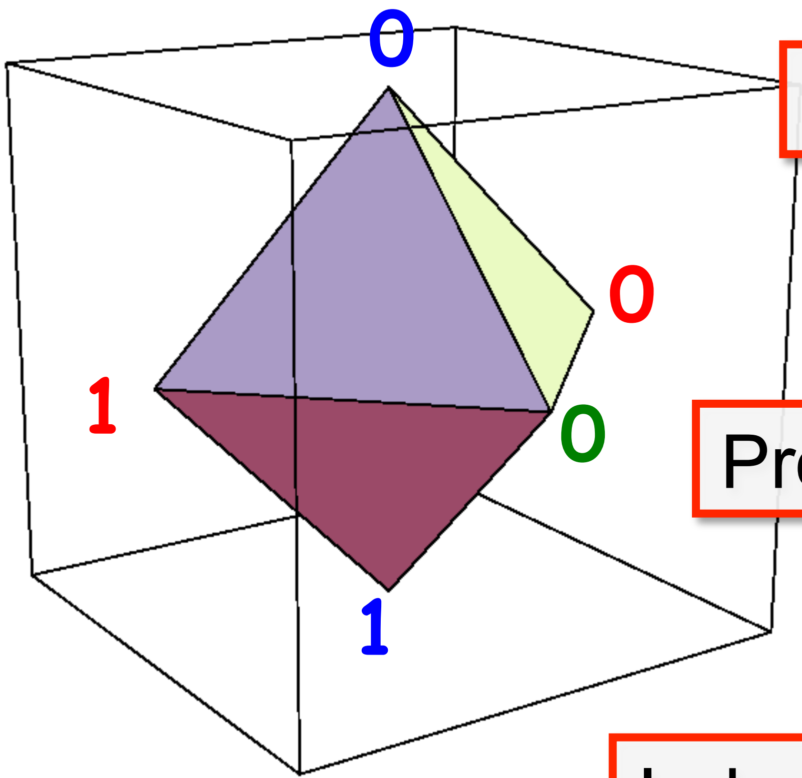
Asynchronous computability theorem [HS99,BG93]

A task (I,O,Δ) is read-write solvable if and only if there is a chromatic simplicial map from a subdivision $\chi^r(I)$ to O carried by Δ

Read-write model (RW) and IIS are equivalent [BG93,BG97,GR10]

- a task is solvable in IIS iff it is solvable in RW

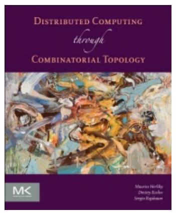
Input Complex for Binary Consensus



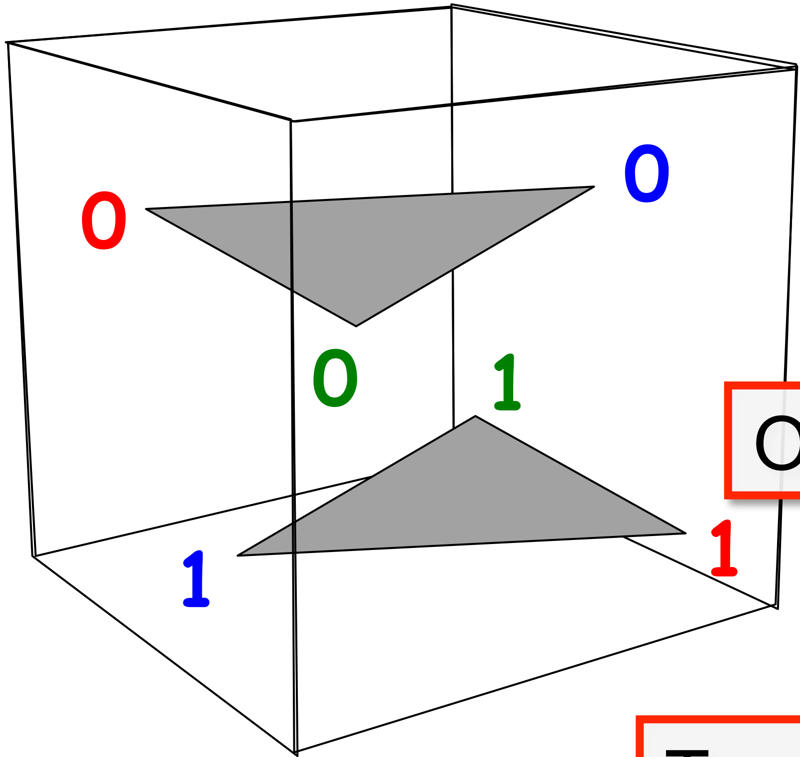
All possible initial states

Processes: red, green, blue

Independently assigned 0 or 1



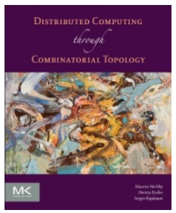
Output Complex for Binary Consensus



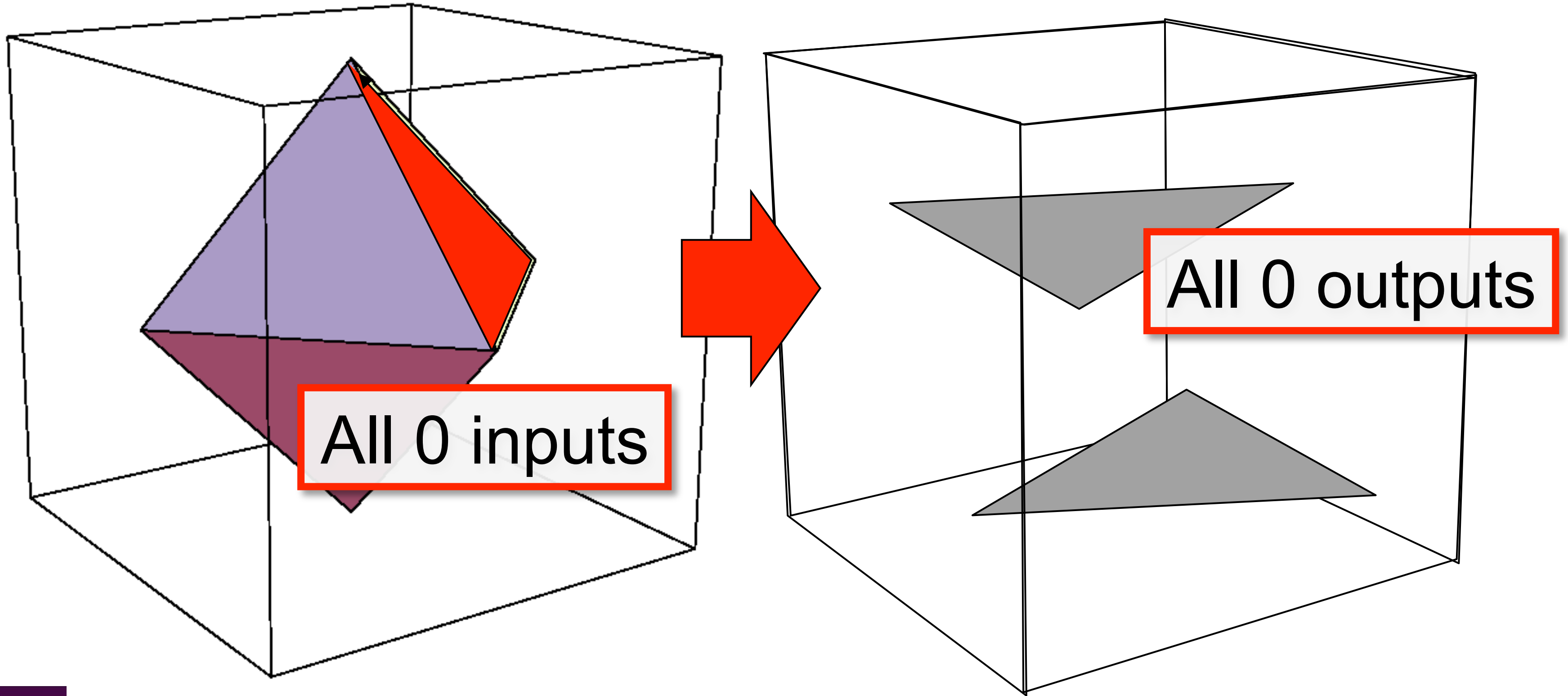
All possible final states

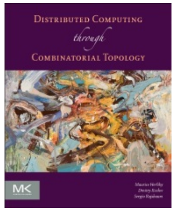
Output values all 0 or all 1

Two disconnected simplexes

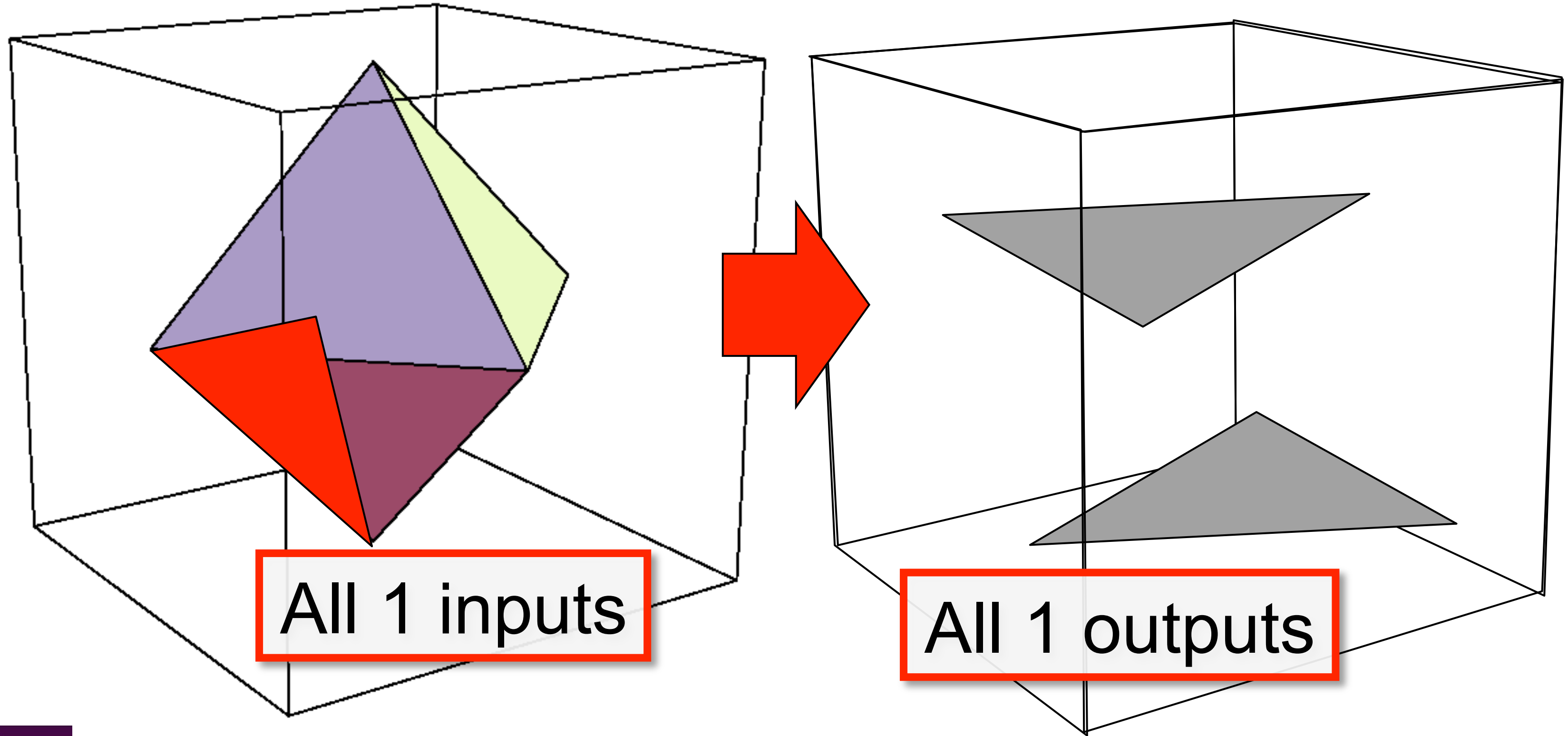


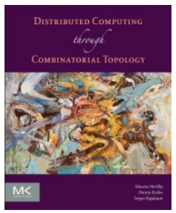
Carrier Map for Consensus



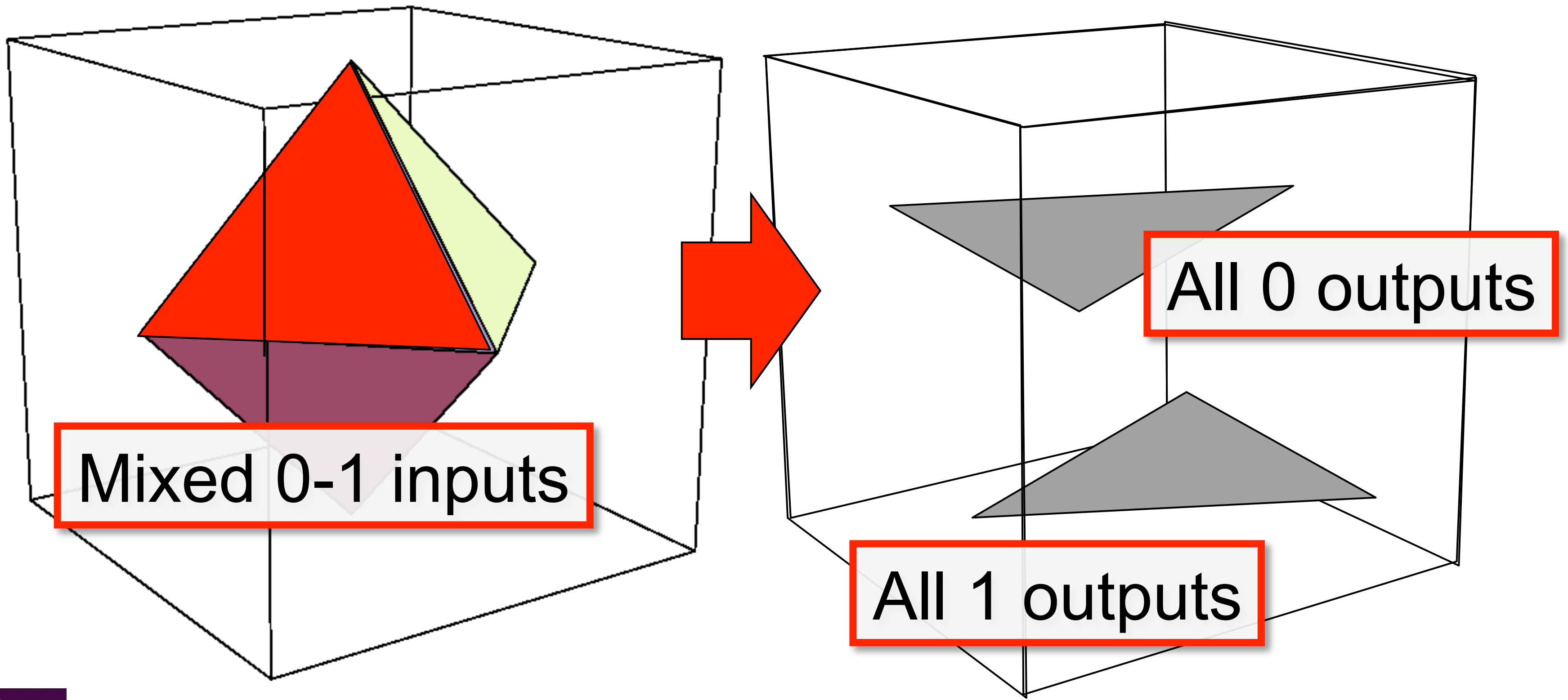


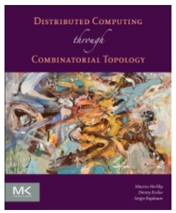
Carrier Map for Consensus





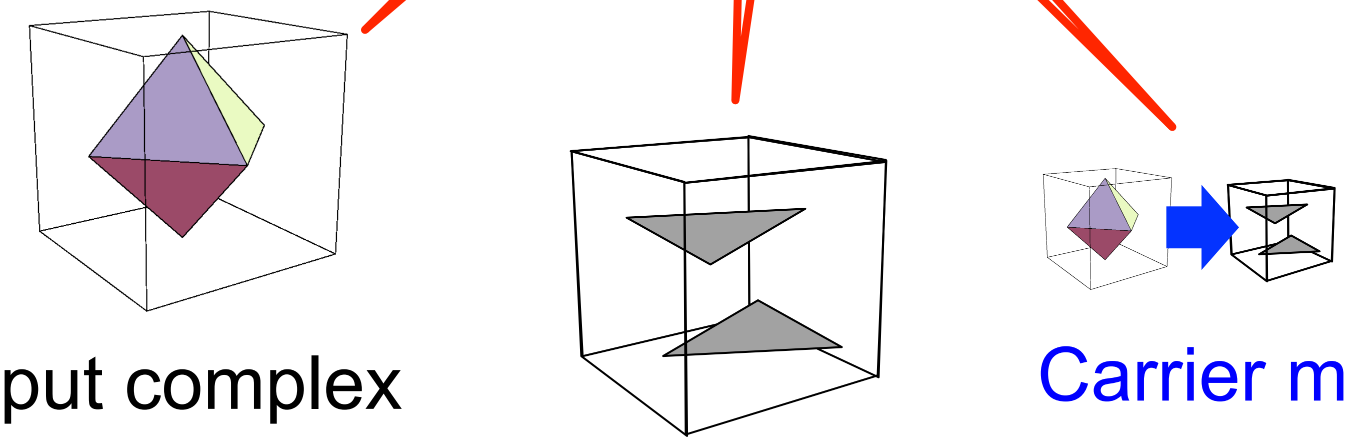
Carrier Map for Consensus





Task specification

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

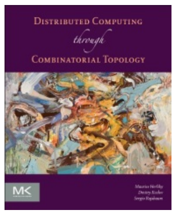


Input complex

Output complex

Carrier map

$\Delta\colon \mathcal{I} \to 2^{\mathcal{O}}$

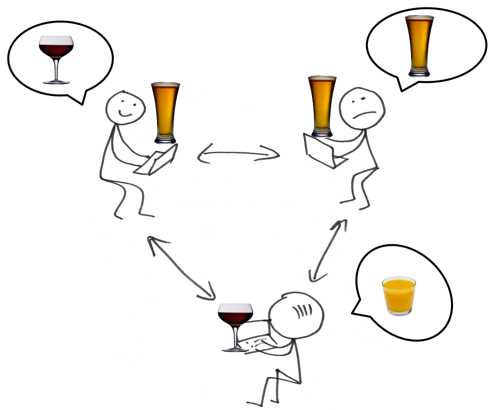


Colorless tasks

Correctness depends on inputs/outputs only, regardless of process identifiers

- $T = (I, O, \Delta)$:
  - ✓ Set of input sets I
  - ✓ Set of output sets O
  - ✓ Task specification $\Delta: I \rightarrow 2^O$

- k-Set agreement
  - ✓ $I = O = s^N$
  - ✓ $\forall \sigma \in I: \Delta(\sigma) = skel^k \sigma$



Colorless Tasks

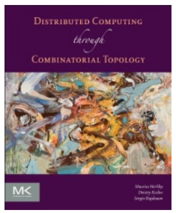
$$(\mathcal{I}, \mathcal{P}, \Xi)$$

(colorless) input complex

(colorless) protocol complex

strict carrier map

$\Xi \colon \mathcal{I} \to 2^{\mathcal{P}}$



(Colorless) Asynchronous Computability Theorem

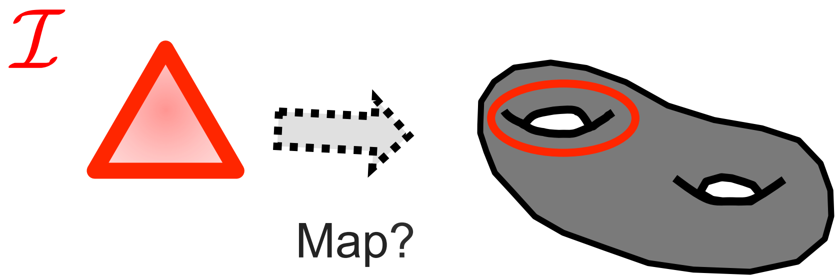
The colorless task $(\mathcal{I},\mathcal{O},\Delta)$ has a wait-free RW protocol …

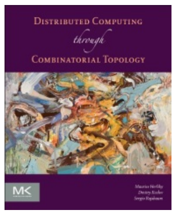
if and only if …

there is a continuous map

$f\colon |\mathcal{I}| \to |\mathcal{O}|$

carried by Δ.





(Colorless) t-resilient Asynchronous Computability Theorem

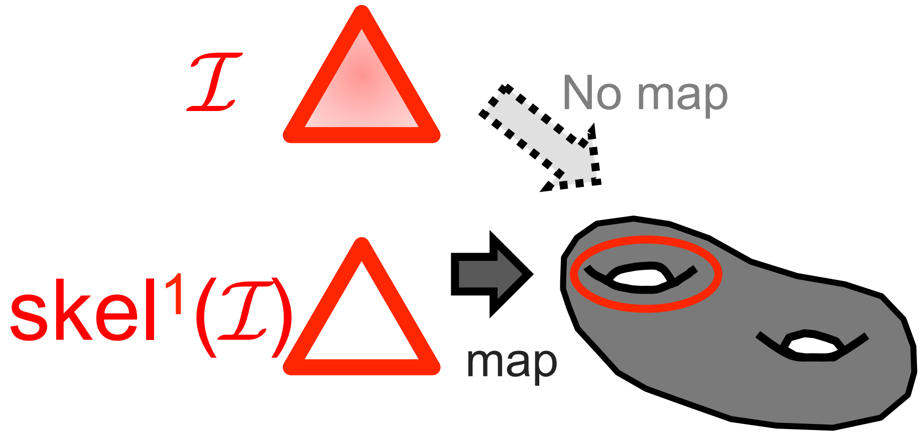
The colorless task $(\mathcal{I},\mathcal{O},\Delta)$ has a t-resilient RW protocol …

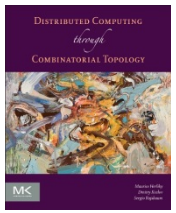
if and only if …

there is a continuous map

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

carried by Δ.





Protocol Implies Map

May assume protocol complex is $\mathcal{P} = X^N \text{skel}^t \mathcal{I}$.

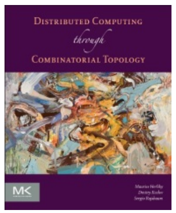
decision map

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

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

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

carried by Δ.



Simplicial Approximation Theorem

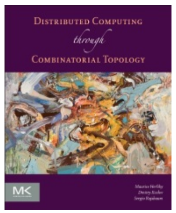
- Given a continuous map

$$f\colon |\mathcal{A}| \to |\mathcal{B}|$$

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

$$\phi\colon \mathrm{X}^{N}\, \mathcal{A} \to \mathcal{B}$$

Holds for most any mesh-shrinking subdivision



Map Implies Protocol

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

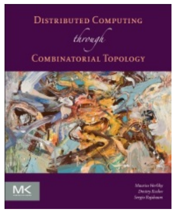
$\phi: X^N\ \text{skel}^t\, \mathcal{I} \to \mathcal{O}$

carried by Δ.

Solve using …

barycentric agreement

t-set agreement



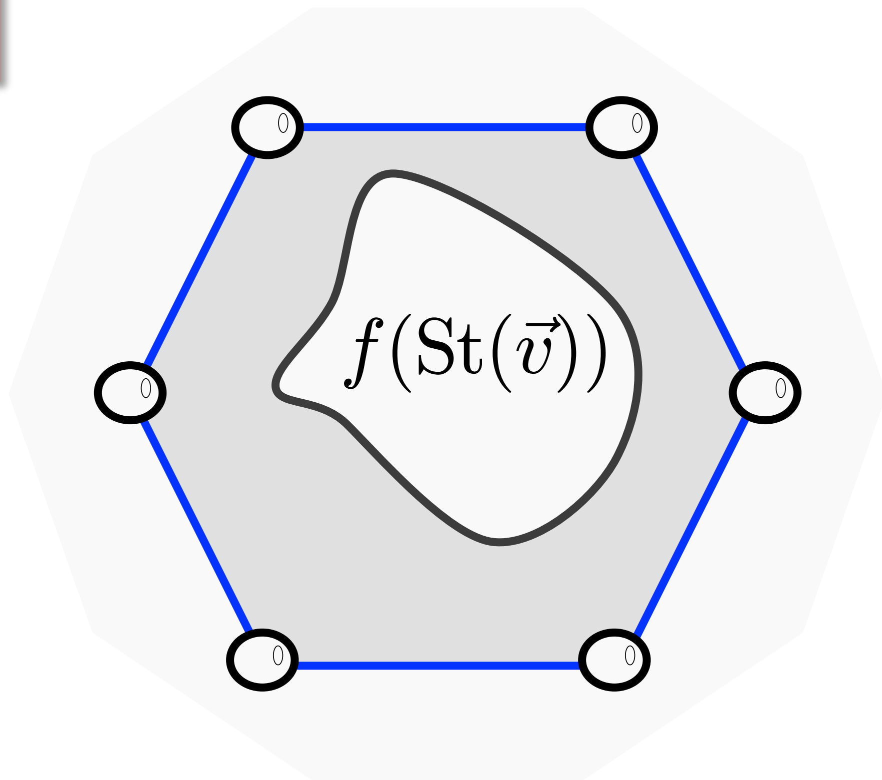
Simplicial Approximation

ϕ: X ^{N} $\mathcal{A} \rightarrow \mathcal{B}$

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

for every v in $\mathcal{A}$ $\cdots$

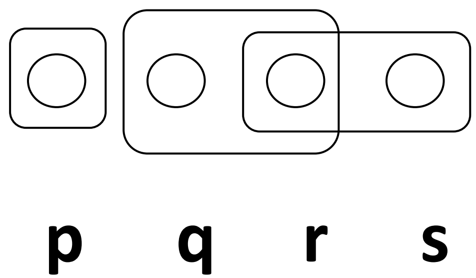
$$f(St(\vec{v})) \subseteq St(\phi(\vec{v}))$$



B

What about…

- Generic sub-models of RW
  - ✓Many problems (e.g., consensus) cannot be solved wait-free
  - ✓So restrictions (sub-models) of RW were considered
- Adversarial models specifying the possible correct sets [DFGT,2009]
  - Non-uniform/correlated faults
  - For colorless tasks, a superset-closed adversaries is characterized by its core size



Combinatorial Structures for Distributed Computing

Affine tasks

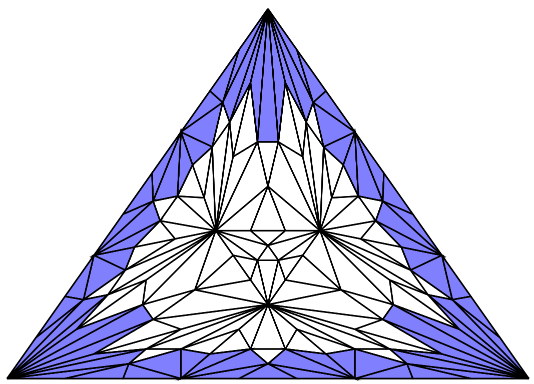
Kyoto University, 2018

Model as a task [KR16,KR18]

- A (long-lived, non-compact) model can be matched by a (one-shot, compact) task

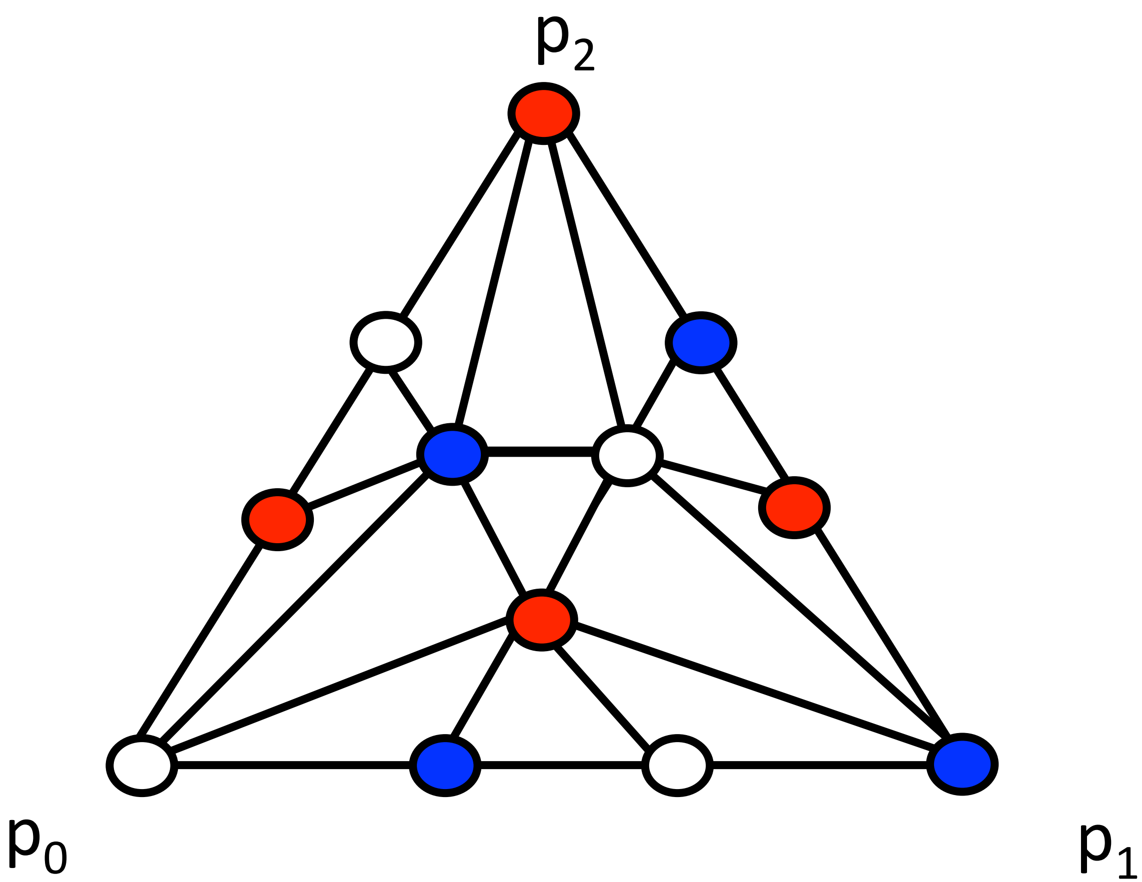
- Any fair adversary has a matching task
  - ✓ also holds for adversaries
  - ✓“natural” models

- E.g., k-concurrency:



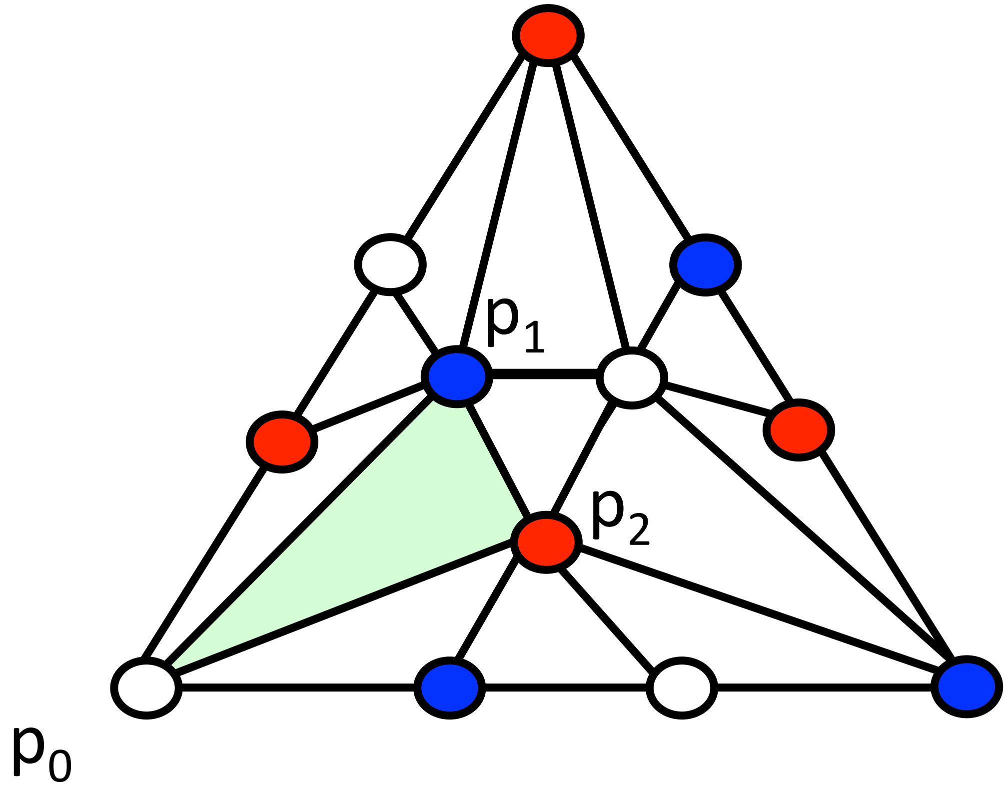
IS as a task $(s^N, \mathcal{X}(s^N), \mathcal{X})$

A process starts at its corner…



IS as a task (s^N, $\mathcal{X}(s^N)$, $\mathcal{X}$)

and outputs a vertex of it color (carrier-preserving)



Chromatic simplex agreement on χ(I)

IS - the task for wait-freedom

Read-write model (RW) and IIS are equivalent [BG93,BG97,GR10]

- a task is solvable in IIS iff it is solvable in RW

Asynchronous computability theorem[HS93]:

A task $(\mathcal{I},O,\Delta)$ is wait-free read-write solvable if and only if there is a chromatic simplicial map from a subdivision $\chi^r(\mathcal{I})$ to O carried by Δ

Model as a task?

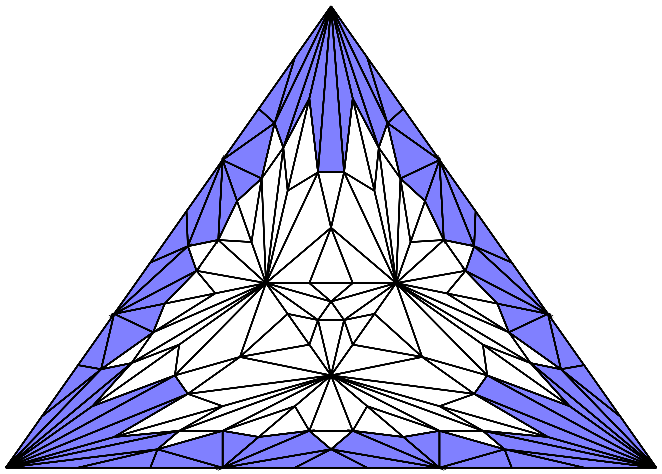
- M model, a set of (infinite) runs
  - ✓Alternating writes and snapshots
- T task, a one-shot distributed function (I,O,Δ):
  - ✓Set of input vectors I (input complex)
  - ✓Set of output vectors O (output complex)
  - ✓Task specification Δ: I ➔ 2^O (carrier map)
- T*, iterations of T, have the same task computability as M

(Solving a task in M is equivalent to solving T)

Affine tasks

(s^N, L, Δ):

- s^N – N-dimensional simplex
- $L \subseteq \mathcal{X}^k(s^N)$
- $\Delta(\sigma)=\mathcal{X}^k(\sigma) \cap L$



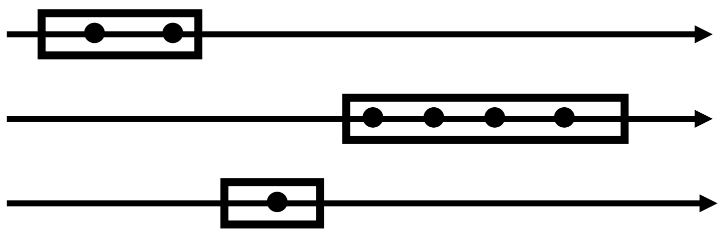
$L=\mathcal{X}^k(s^N)$: IS

Model as a task

- IS is the matching affine task for wait-free runs
  - ✓ What about restrictions of wait-free?

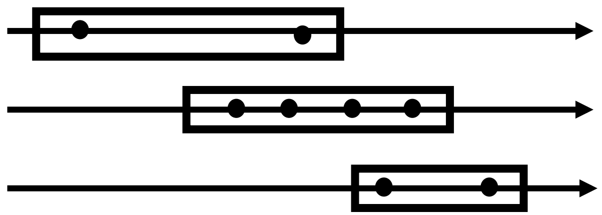
- k-concurrency?
  - ✓ a subset of RW runs where at most k process are concurrently active

Concurrency levels [Gaf09]



1-concurrent: at most one process makes progress at a time (global lock)

k-concurrent: at most k processes make progress concurrently (k-resource semaphore)



n-concurrency = wait-freedom ≅ IS

A matching affine task for k-concurrency (0<k<n)?

Defining R_k

Contention sets: all the processes that share a carrier (≈ see each other):

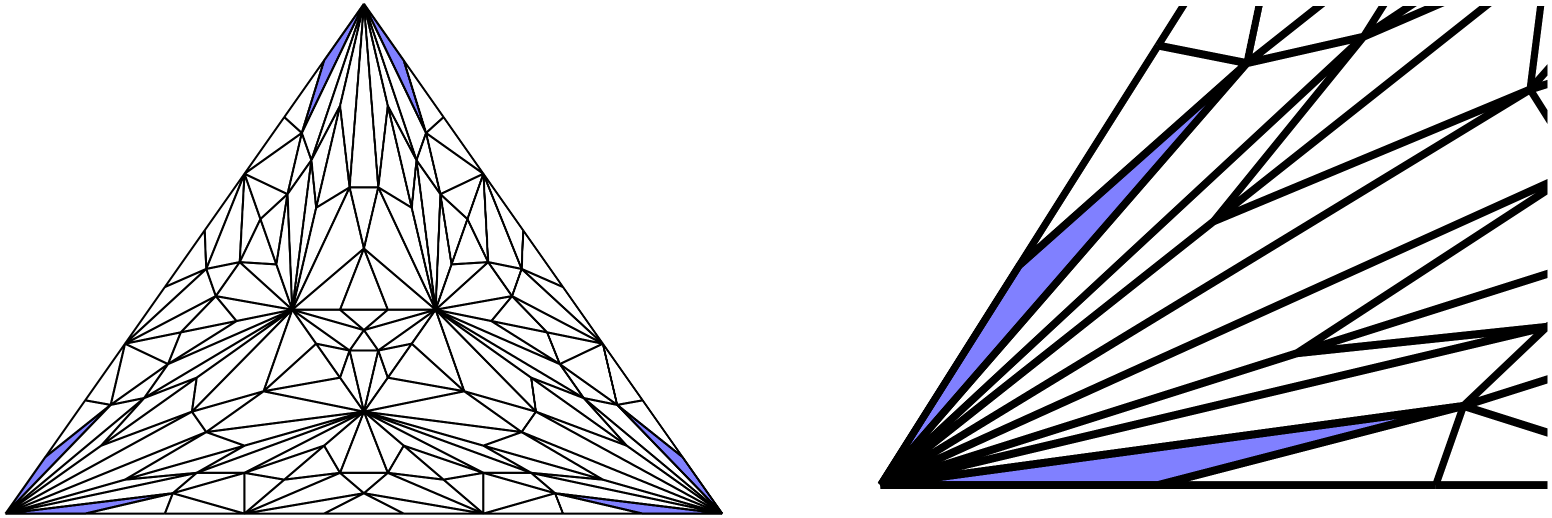
$$Cont(\sigma) = \{S \subseteq \Pi, \forall p, p' \in S, carrier(p, \sigma) = carrier(p', \sigma)\}$$

Include all simplices in $\mathcal{X}^2(s^N)$ of contention k or less

$$\mathcal{R}_k = \{\sigma \in \mathrm{Chr}^2 \mathbf{s}, \forall S \in Cont(\sigma), |S| \leq k\}$$

R_1

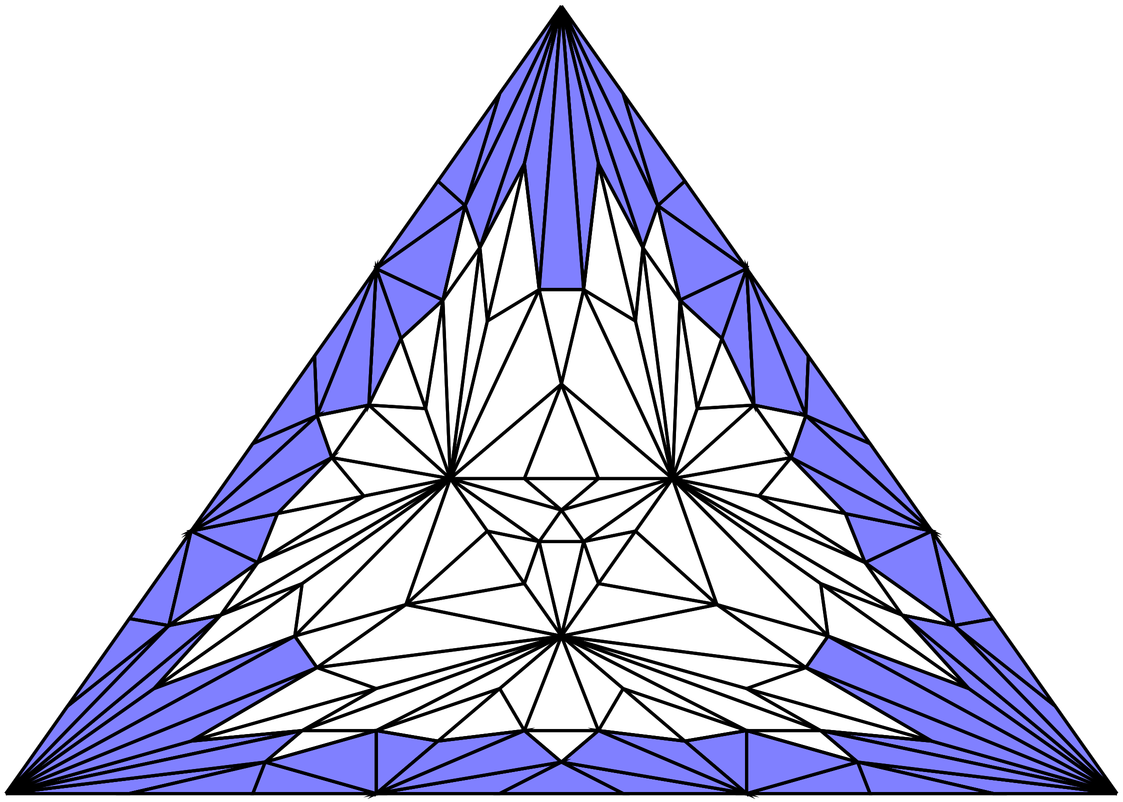
Process proceed in the same total order in two IS rounds:



L_{ord}: total order task for s^2

R_2

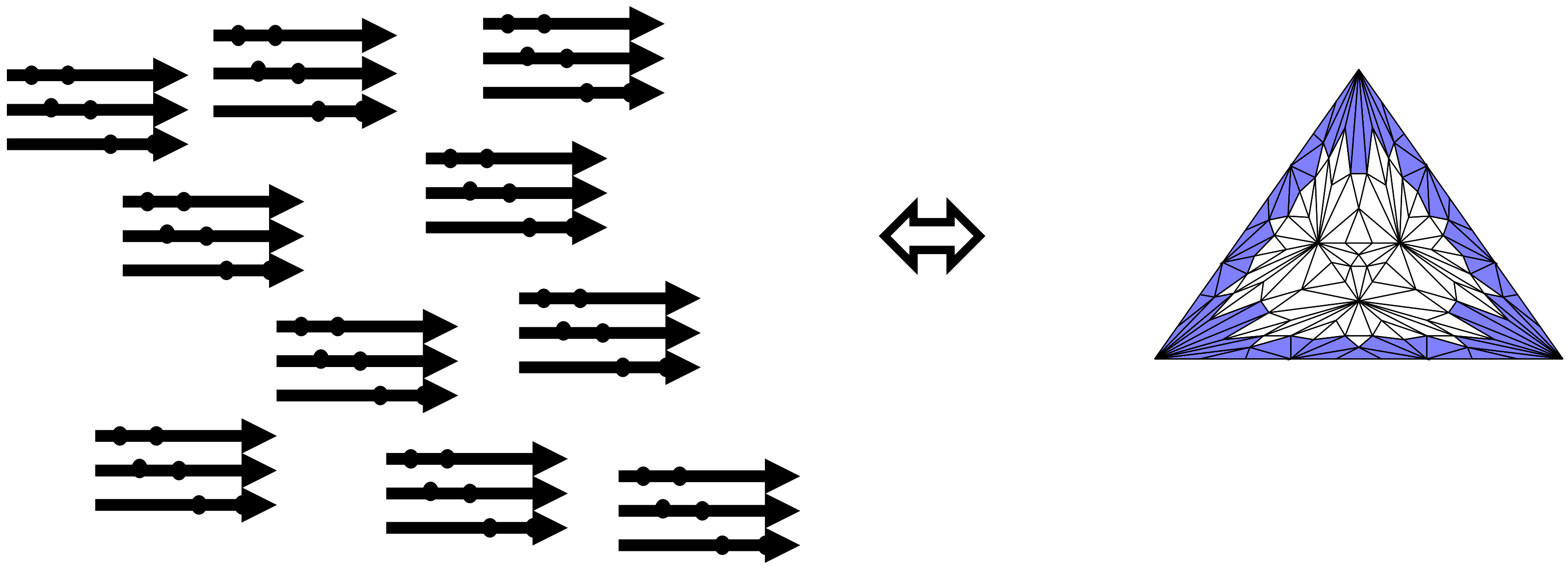
All simplices that touch 1-dimenional faces



k-concurrency = R_k*

T is solvable in R_k* iff T is solvable k-concurrently:

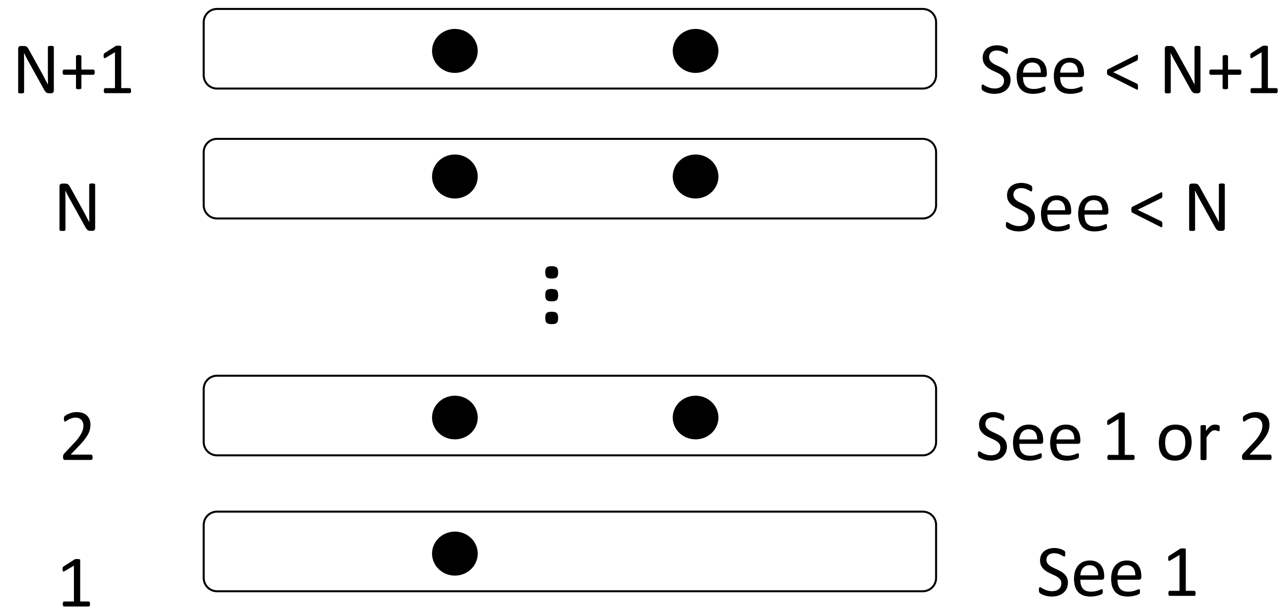
1. k-concurrency simulates R_k*
2. R_k* simulates k-concurrency



1. From k-concurrency to R_k*

R_k can be solved k-concurrently:

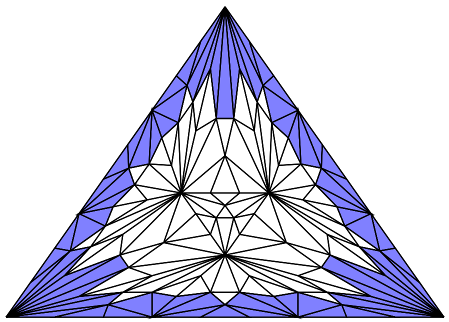
k-concurrent chromatic simplex agreement on R_k



Two rounds of k-concurrent IS implementation [BG93] give R^k

2. From to R_k* *to* k-concurrency

- R^k can be used to solve k-set agreement:
  - ✓Decide on the value of (up to k) "leaders" processes (chosen by the size of IS^1 output)
- IIS (and thus R_k*) can simulate RW [BG97,GR10]

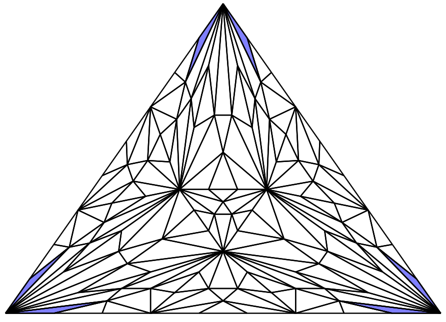


Simulate a protocol that uses read-write and k-set consensus objects?

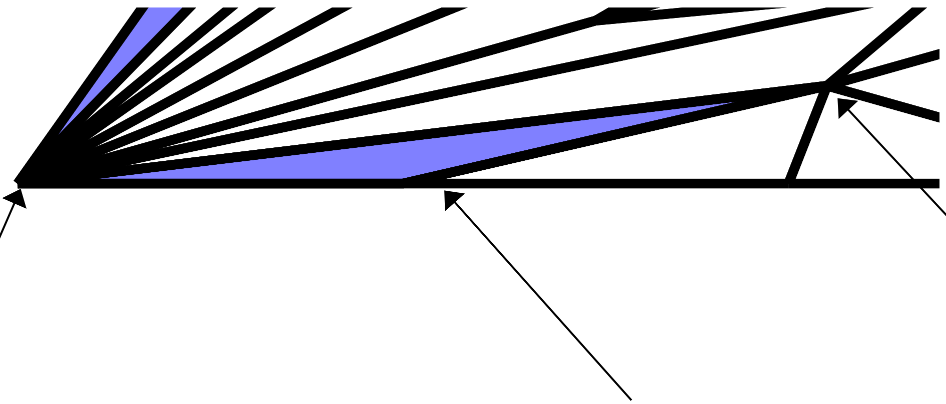
Not that simple: how to combine simulating RW with solving k-SA?

Example: total order (k=1)

Solution of any task (I,O,Δ) in just one iteration of L_{ord}



{p0},{p1},{p2} | {p0},{p1},{p2}



Decide on
a (deterministically chosen) value v_0 in Δ({p0})

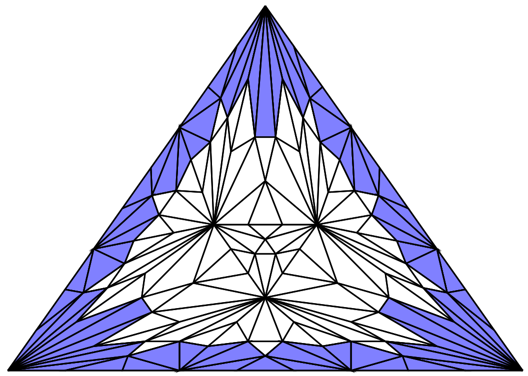
Decide on
a value v_1 Δ({p0,p1})
In the link of v_0

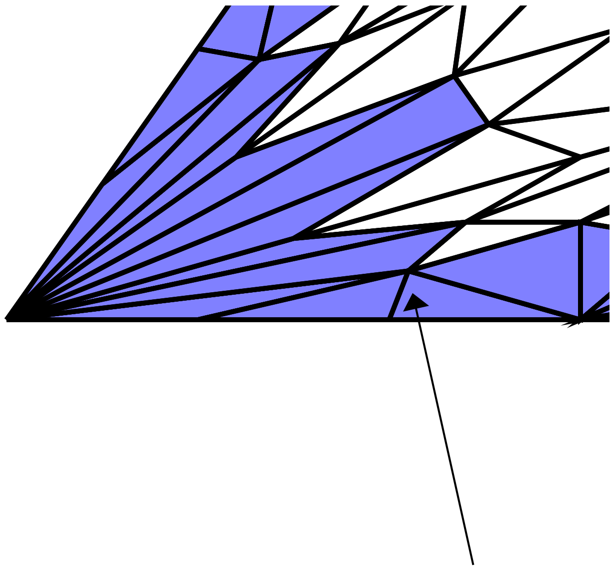
Decide on
a value v_2 in Δ({p0,p1,p2})
In the link of $\{v_0, v_1\}$

Example: R_2

More iterations might be needed

{p0},{p1},{p2} | {p0},{p1},{p2}





Who are the leaders?

Simulating k-concurrency

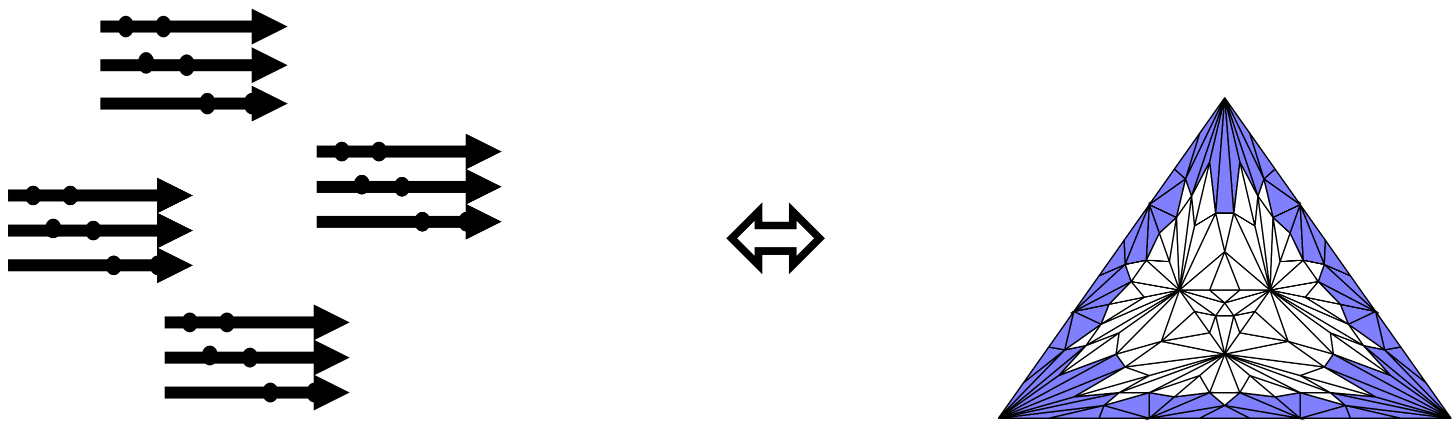
- Adaptive k-set consensus
  - ✓k-commit-adopt: commit (decide) if among k “fastest” non-terminated processes, adopt otherwise
- RW + (adaptive) k-set consensus => k state machines
  - ✓Generalized universality [GG11]
  - ✓m active simulators: machines *1..min(m,k)* are active
  - ✓Any RW protocol on up to k state machines can be simulated
- k processes simulate a k-concurrent system
  - ✓Extended BG simulation [Gaf09]
  - ✓Let state machines be (EBG) simulators

RW + k-set agreement simulate k-concurrency

k-concurrency = R_k*

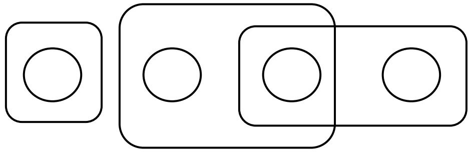
T is solvable in R_k* iff T is solvable k-concurrently:

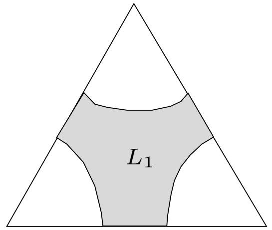
1. k-concurrency simulates R_k*
2. R_k* simulates k-concurrency



Other models?

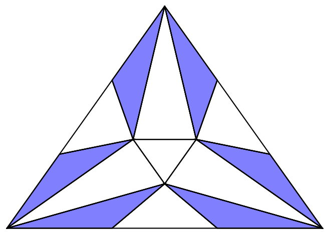
- Adversarial models [DFGT09]
  - ✓Non-uniform/correlated faults
  - ✓[SHG16]: t-resilience





- Set-consensus collections [DFGK16]
  - ✓RW + set-consensus objects in $\{(s_1,t_1),\ldots,(s_m,t_m)\}$
  - ✓k-concurrency ≅ k-set consensus

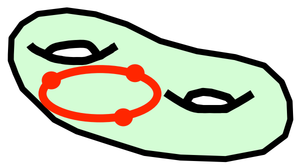
- Affine tasks are in $\mathcal{X}^2(s^N)$
  - ✓Sometimes even in $\mathcal{X}^1(s^N)$



2-consensus (TAS)

What is good about it?

- Compact representation of non-compact models
- Conjecture: possible for all "natural models"
  - ✓ Captured by computing artifacts
  - ✓ Not 0-1-exclusion, WSB, Möbius etc.
- Conjecture: relations between models (affine tasks) are decidable
  - ✓ Reduces to maps between bounded sub-complexes of $\mathcal{X}^2(s^N)$
  - ✓ 3-process, read-write wait-free solvability of (colorless) tasks are undecidable [GK95,HR97]

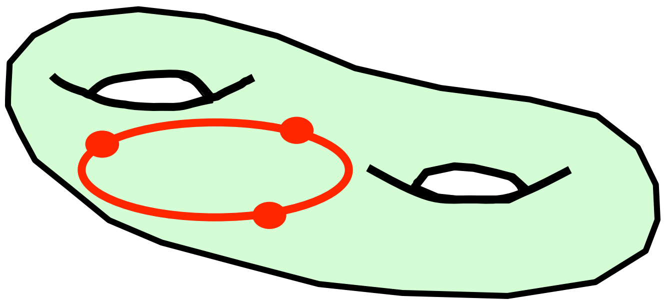


Decidability of tasks

- Given a task T and a model M …
- Is it decidable that T can be solved in M?

in general, no

- 3-process, read-write wait-free solvability of (colorless) tasks is undecidable [GK95,HR97]
  - ✓ Loop agreement task is reducible [HS93] to loop contractibility ≅ word problem
  - ✓ Extends to 2-resilient solvability



Concluding

- Computability can be captured by the analysis of the corresponding simplicial complex
  - ✓For tasks and (some) adversarial models
- Open problems
  - ✓Long-lived abstractions (queues, hash tables, TMs…)
  - ✓Byzantine adversary: a faulty process deviates arbitrarily
  - ✓Anonymous systems?
  - ✓Partial synchrony
- Mathematics induced by DC?