Locally Solvable Tasks
and the Limitations of Valency Arguments

Hagit Attiya

Technion

Armando Castañeda

UNAM

Sergio Rajsbaum

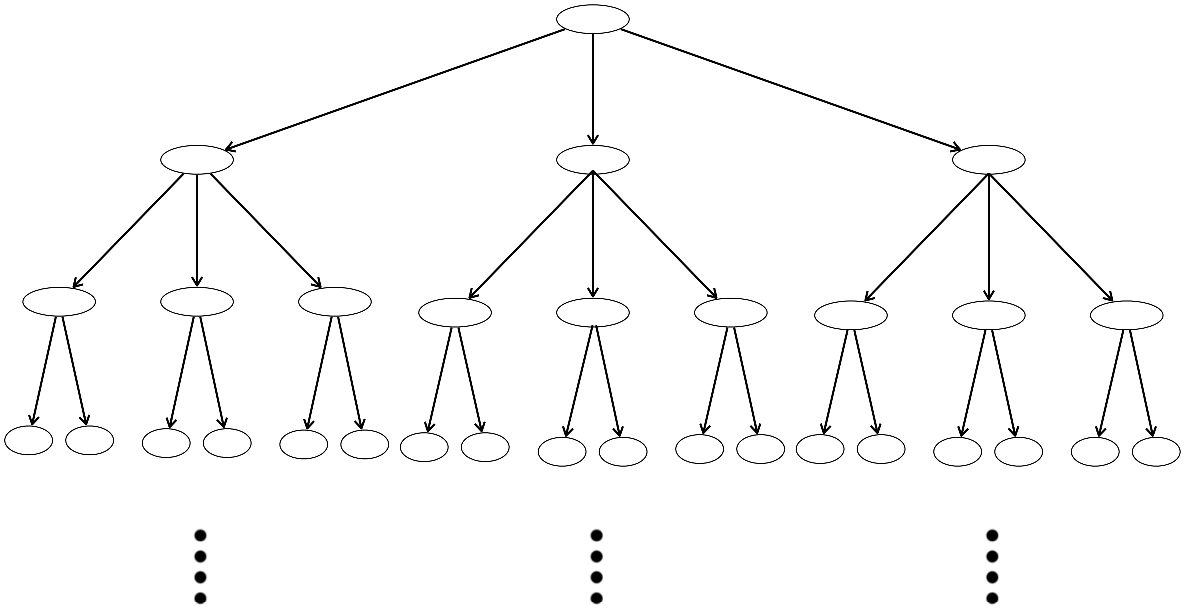
UNAM

Local Proof-Styles **Vs** **Global Proof-Styles**

(e.g. FLP85) (e.g. BG93, HS93, SZ93)

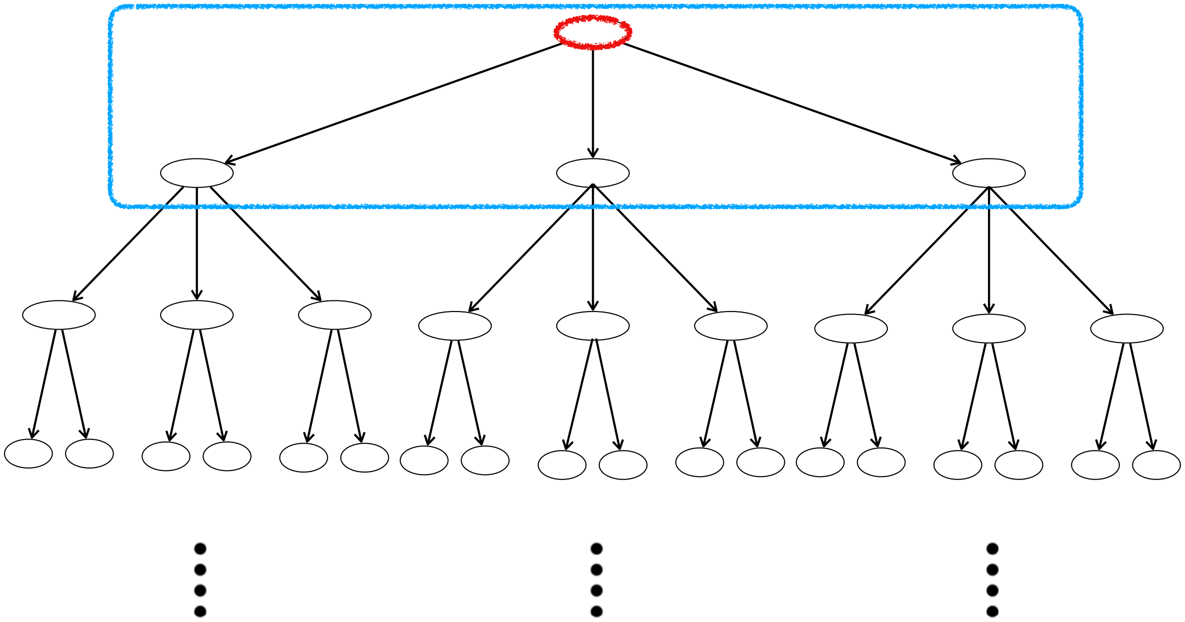
Local Proof-Style

- Each stage:
  - single configuration holding a property
  - indistinguishability analysis
  - some successors
  - pick a successor
- Essentially, it finds an invariant
- It **corners** the protocol
- FLP85's invariant: bivalent
- Liveness => no safety
- Safety => no liveness



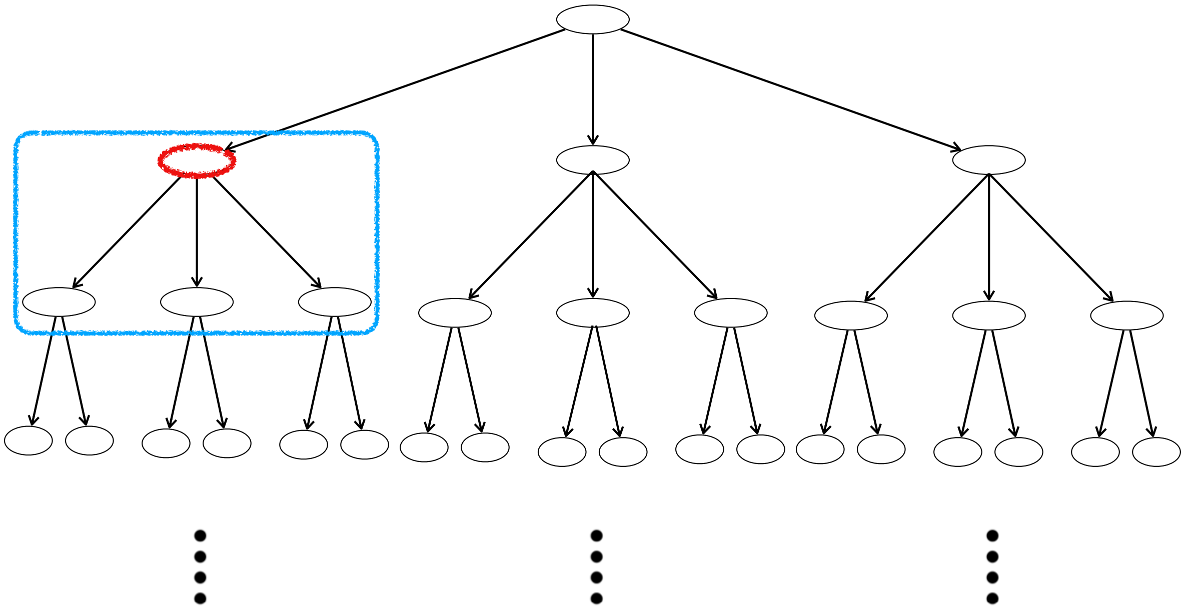
Local Proof-Style

- Each stage:
    - single configuration holding a property
    - indistinguishability analysis
    - some successors
    - pick a successor
- Essentially, it finds an invariant
- It **corners** the protocol
- FLP85's invariant: bivalent
- Liveness => no safety
- Safety => no liveness



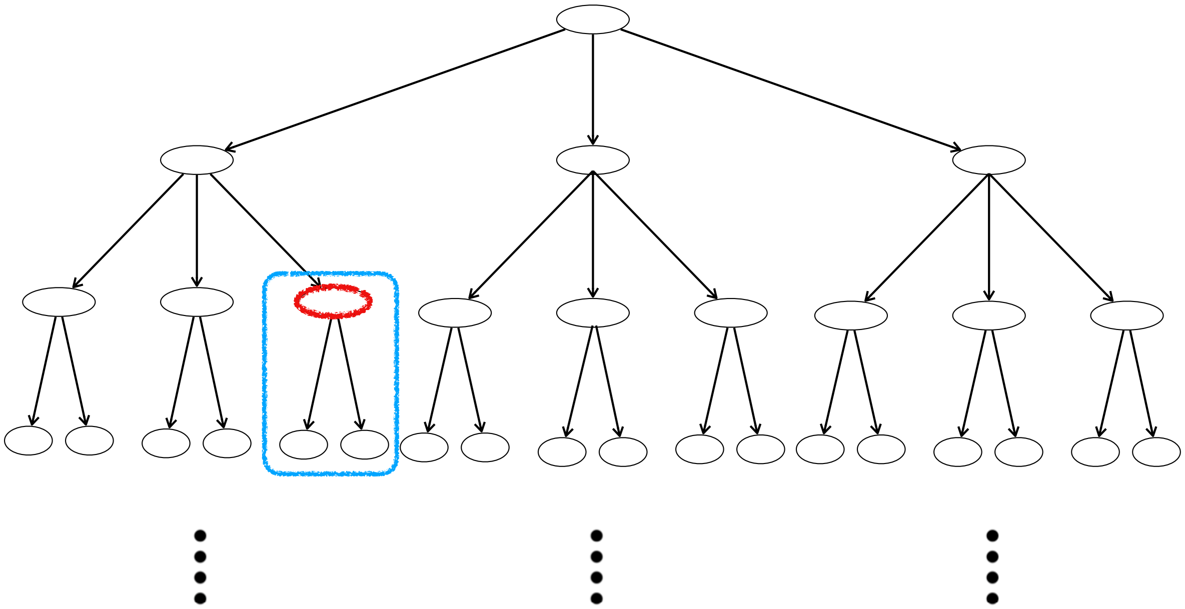
Local Proof-Style

- Each stage:
  - single configuration holding a property
  - indistinguishability analysis
  - some successors
  - pick a successor
- Essentially, it finds an invariant
- It **corners** the protocol
- FLP85's invariant: bivalent
- Liveness => no safety
- Safety => no liveness



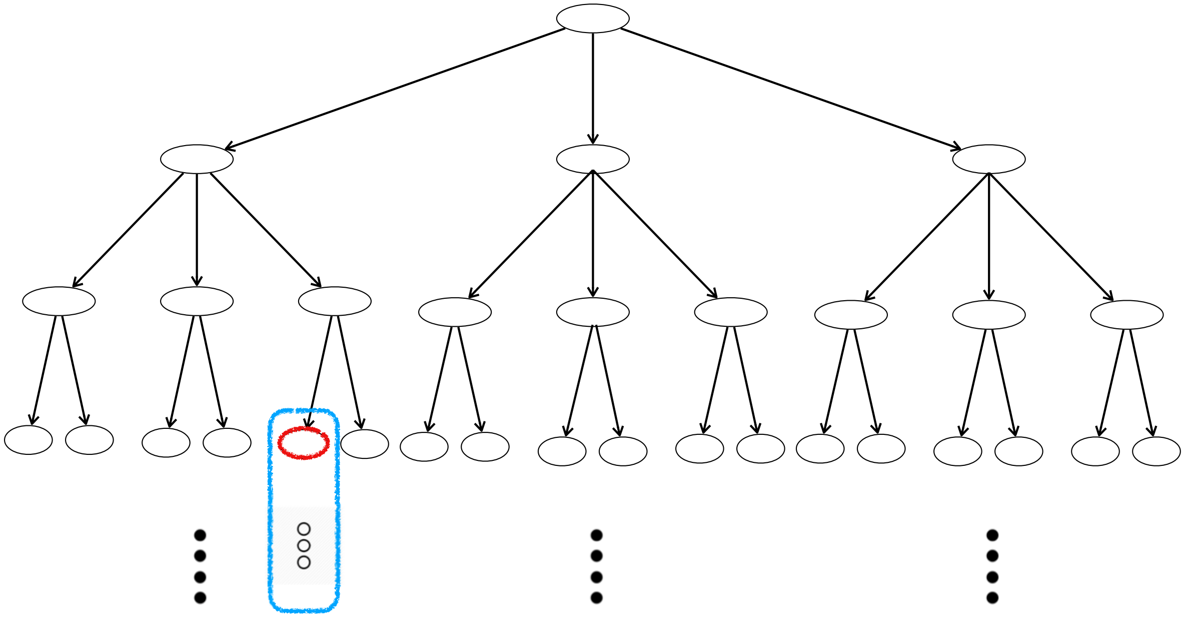
Local Proof-Style

- Each stage:
  - single configuration holding a property
  - indistinguishability analysis
  - some successors
  - pick a successor
- Essentially, it finds an invariant
- It **corners** the protocol
- FLP85's invariant: bivalent
- Liveness => no safety
- Safety => no liveness



Local Proof-Style

- Each stage:
  - single configuration holding a property
  - indistinguishability analysis
  - some successors
  - pick a successor
- Essentially, it finds an invariant
- It **corners** the protocol
- FLP85's invariant: bivalent
- Liveness => no safety
- Safety => no liveness



Local Proof-Style

- Started with FLP85: There is no 1-resilient message-passing protocol for consensus

- Consensus:
  - Termination: all correct processes decide
  - Validity: a decided value is a proposal
  - Agreement: correct processes decide the same value

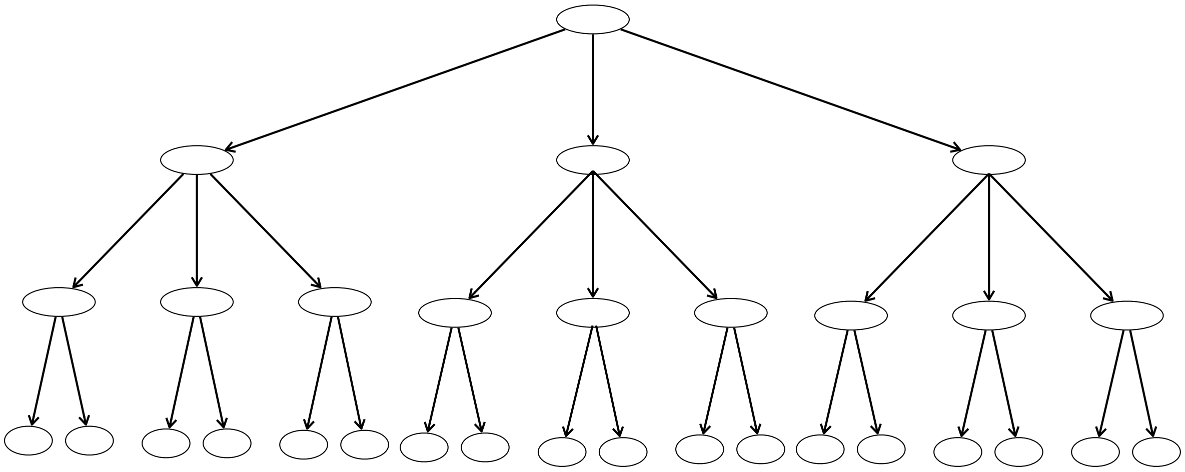
- Same proof-style for many tasks:
  - Approximate agreement
  - Randomized consensus
  - Concurrent data structures

- Simple and elegant approach

- Typically, only a few assumptions are needed (e.g. full-information)

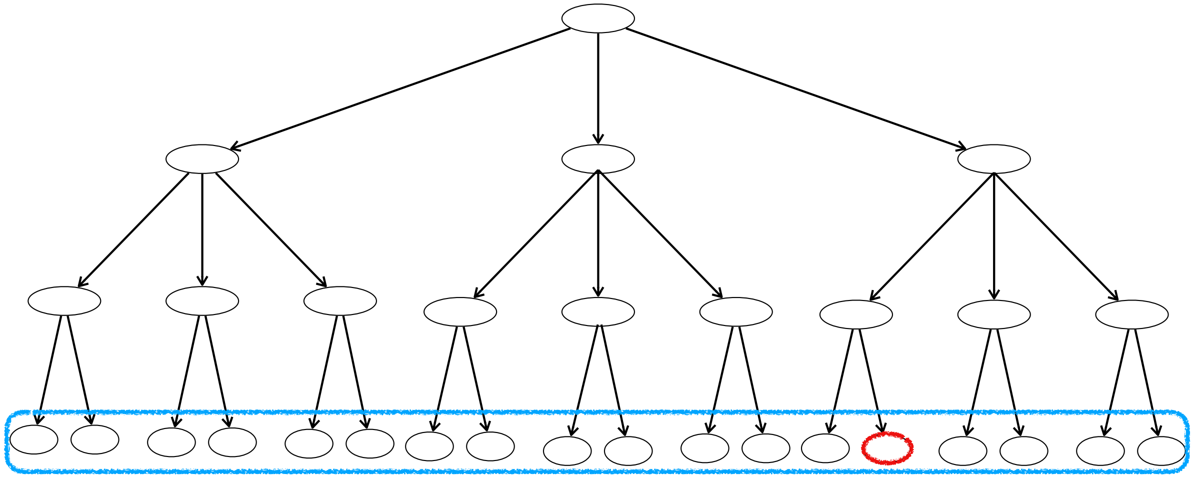
Global Proof-Style

- Only final configurations
- All in a combinatorial object
- Commutative properties in the object
- Analyze properties of the object
- There **exists** a mistake
- BG93, HS93, SZ93: Sperner's lemma
- Liveness => no safety



Global Proof-Style

- Only final configurations
- All in a combinatorial object
- Commutative properties in the object
- Analyze properties of the object
- There **exists** a mistake
- BG93, HS93, SZ93: Sperner's lemma
- Liveness => no safety



Global Proof-Style

- Started with BG93, HS93, SZ93: There is no wait-free read/write shared memory protocol for k-set agreement
- k-set agreement:
  - Termination: all correct processes decide
  - Validity: a decided value is a proposal
  - k-Agreement: correct processes decide at most k distinct values
- Same proof-style for other tasks, e.g. renaming, weak symmetry breaking
- Powerful tool: Solvability characterization of **any** task.
- Assumptions are needed. Some models are problematic (e.g. non-compact, no round-structure)
- Almost global proof. Ficher&Lynch82: t+1 round lower bound for synchronous consensus

Local vs Global

- Is there a local impossibility proof for set-agreement or renaming?
- Can a set agreement or renaming protocol be ‘cornered’?
- Is there always a local impossibility proof?
- Is the complexity of global style-proofs unavoidable?

Local vs Global

- Is there a local impossibility proof for set-agreement or renaming?
- Can a set agreement or renaming protocol be ‘cornered’?
- Is there always a local impossibility proof?
- Is the complexity of global style-proofs unavoidable?

Our result: There is no local impossibility proof for (n-1)-set agreement or (2n-2)-renaming in Iterated Immediate Snapshot (IIS) model

Local vs Global

- Is there a local impossibility proof for set-agreement or renaming?
- Can a set agreement or renaming protocol be ‘cornered’?
- Is there always a local impossibility proof?
- Is the complexity of global style-proofs unavoidable?

Our result: There is no local impossibility proof for (n-1)-set agreement or (2n-2)-renaming in Iterated Immediate Snapshot (IIS) model

This talk about set agreement

Previous Work

- Line of research recently started by Alistarh, Aspnes, Ellen, Gelashvili and Zhu in 2019

- Defined extension-based proofs in Non-uniform IIS (NIIS)

- Their result: No extension-based proof for k-set agreement in NIIS

- Our approach is different

- More about this later

Iterated Immediate Snapshot (IIS)

- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure
- Full-information: each process writes all it knows
- Infinite bidimensional shared memory: M[1 …][1 … n]
- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
    - Sequence of concurrency classes
    - Processes in a concurrency class
      write together then snapshot together

Iterated Immediate Snapshot (IIS)

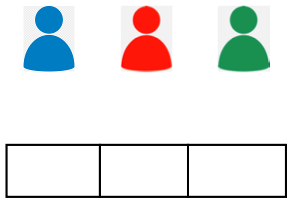
- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure
- Full-information: each process writes all it knows
- Infinite bidimensional shared memory: $M[\overset{\overset{\text{Round}}{\uparrow}}{1 \ldots}][\overset{\overset{\text{Proc ID}}{\uparrow}}{1 \ldots n}]$
- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
    - Sequence of concurrency classes
    - Processes in a concurrency class
      write together then snapshot together

Iterated Immediate Snapshot (IIS)

- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure
- Full-information: each process writes all it knows

Round Proc ID

- Infinite bidimensional shared memory: M[1 …][1 … n]
- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
    - Sequence of concurrency classes
    - Processes in a concurrency class write together then snapshot together



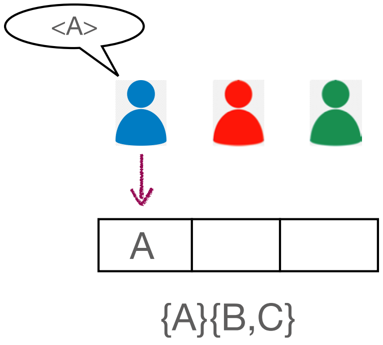
{A}{B,C}

Iterated Immediate Snapshot (IIS)

- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure
- Full-information: each process writes all it knows

Round   Proc ID

- Infinite bidimensional shared memory: M[1 …][1 … n]
- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
    - Sequence of concurrency classes
    - Processes in a concurrency class write together then snapshot together

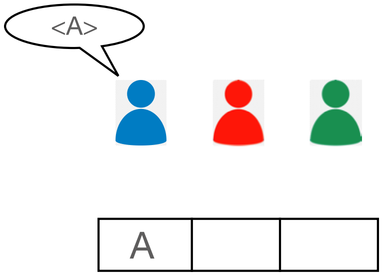


Iterated Immediate Snapshot (IIS)

- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure
- Full-information: each process writes all it knows
- Infinite bidimensional shared memory: M[1 …][1 … n]

Round Proc ID

- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
  - Sequence of concurrency classes
  - Processes in a concurrency class write together then snapshot together



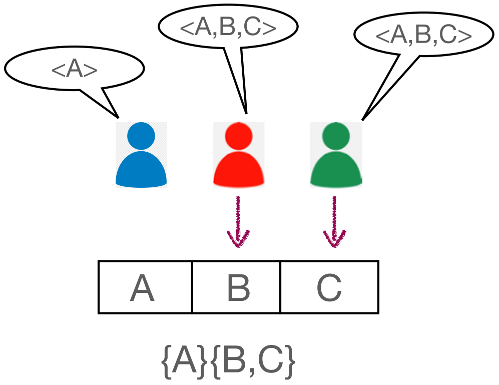
{A}{B,C}

Iterated Immediate Snapshot (IIS)

- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure
- Full-information: each process writes all it knows

Round Proc ID

- Infinite bidimensional shared memory: M[1 …][1 … n]
- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
    - Sequence of concurrency classes
    - Processes in a concurrency class write together then snapshot together

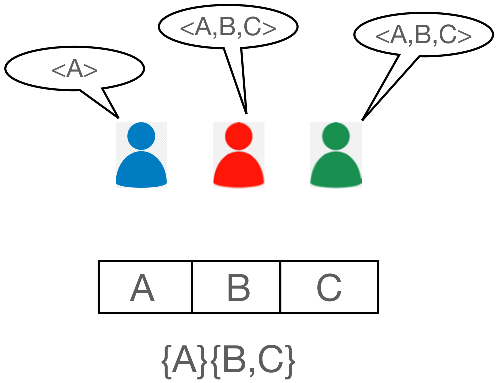


Iterated Immediate Snapshot (IIS)

- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure
- Full-information: each process writes all it knows
- Infinite bidimensional shared memory: M[1 …][1 … n]

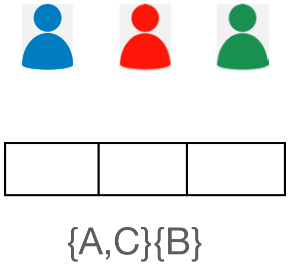
Round Proc ID

- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
  - Sequence of concurrency classes
  - Processes in a concurrency class write together then snapshot together



Iterated Immediate Snapshot (IIS)

- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure
- Full-information: each process writes all it knows
- Infinite bidimensional shared memory: $M[\overset{\text{Round}}{\uparrow}1 \ldots][1 \ldots \overset{\text{Proc ID}}{\uparrow} n]$
- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
    - Sequence of concurrency classes
    - Processes in a concurrency class write together then snapshot together

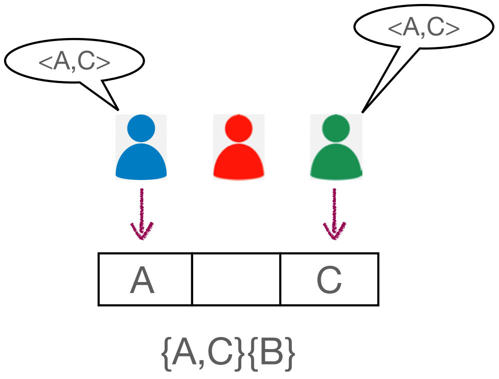


Iterated Immediate Snapshot (IIS)

- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure

Round Proc ID

- Full-information: each process writes all it knows
- Infinite bidimensional shared memory: M[1 …][1 … n]
- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
  - Sequence of concurrency classes
  - Processes in a concurrency class write together then snapshot together

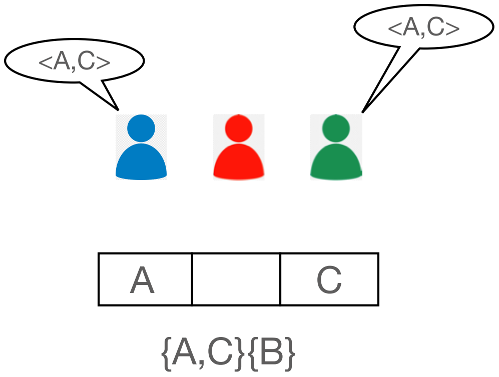


Iterated Immediate Snapshot (IIS)

- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure
- Full-information: each process writes all it knows
- Infinite bidimensional shared memory: M[1 …][1 … n]

Round Proc ID

- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
    - Sequence of concurrency classes
    - Processes in a concurrency class write together then snapshot together

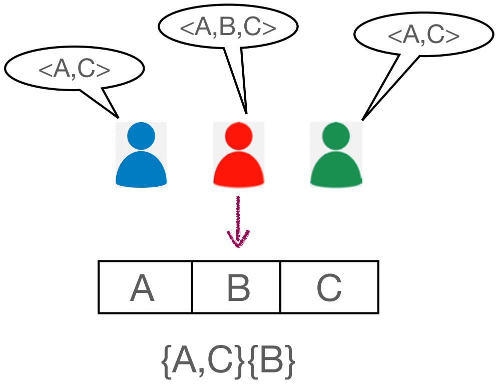


Iterated Immediate Snapshot (IIS)

- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure
- Full-information: each process writes all it knows

Round Proc ID

- Infinite bidimensional shared memory: M[1 …][1 … n]
- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
  - Sequence of concurrency classes
  - Processes in a concurrency class write together then snapshot together

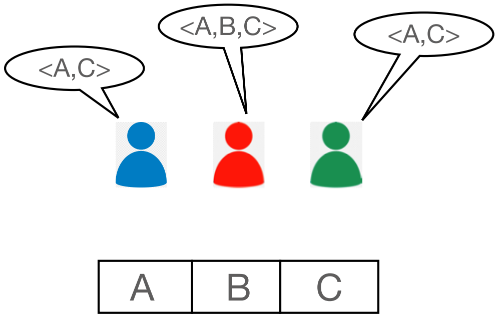


Iterated Immediate Snapshot (IIS)

- n asynchronous processes
- Wait-free: at most n-1 crash failures
- Round-based structure
- Full-information: each process writes all it knows
- Infinite bidimensional shared memory: M[1 …][1 … n]

Round   Proc ID

- Round r: processes do **immediate snapshot** in M[r]
- Immediate snapshot:
  - Sequence of concurrency classes
  - Processes in a concurrency class write together then snapshot together

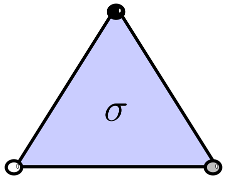


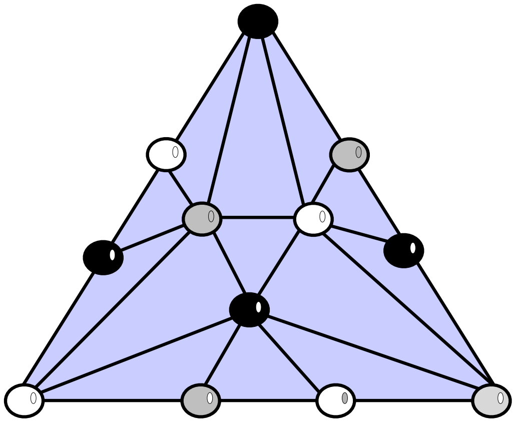
{A,C}{B}

Topological Interpretation of IIS

- View = vertex
- Configuration = set of views = (combinatorial) simplex
- Partial configuration = subset of a configuration = simplex
- Initial configurations = input simplexes
- A bunch of simplexes make a (combinatorial) simplicial complex
- Like a graph in higher dimensions
- Commutativity of operations in a single object

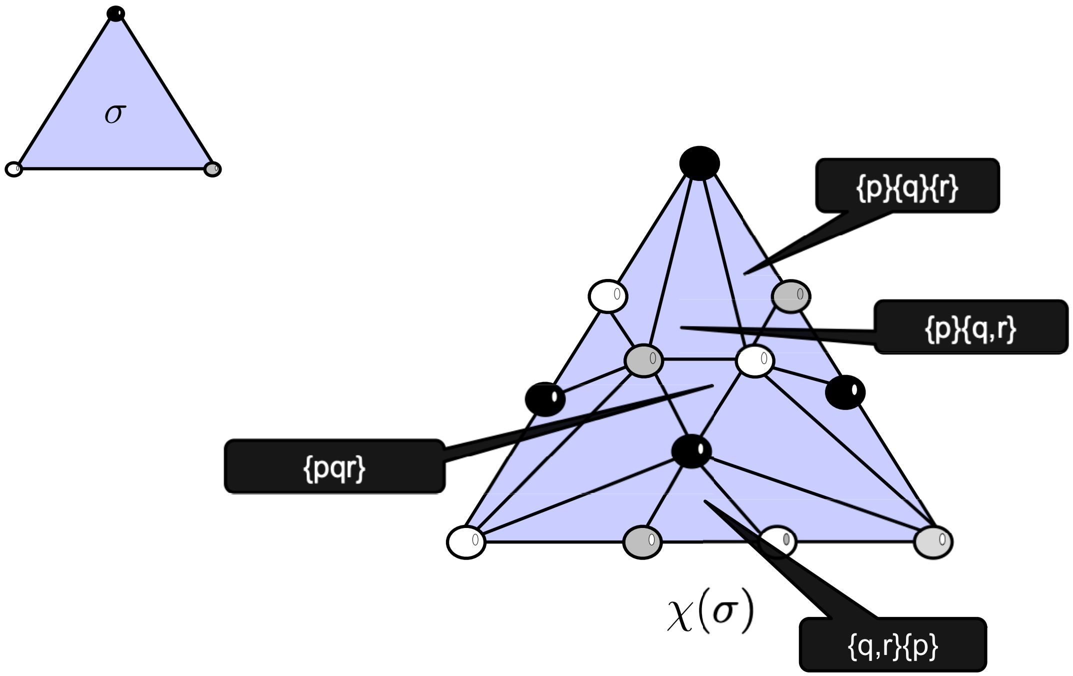
Topological Interpretation of IIS





$\chi(\sigma)$

Topological Interpretation of IIS



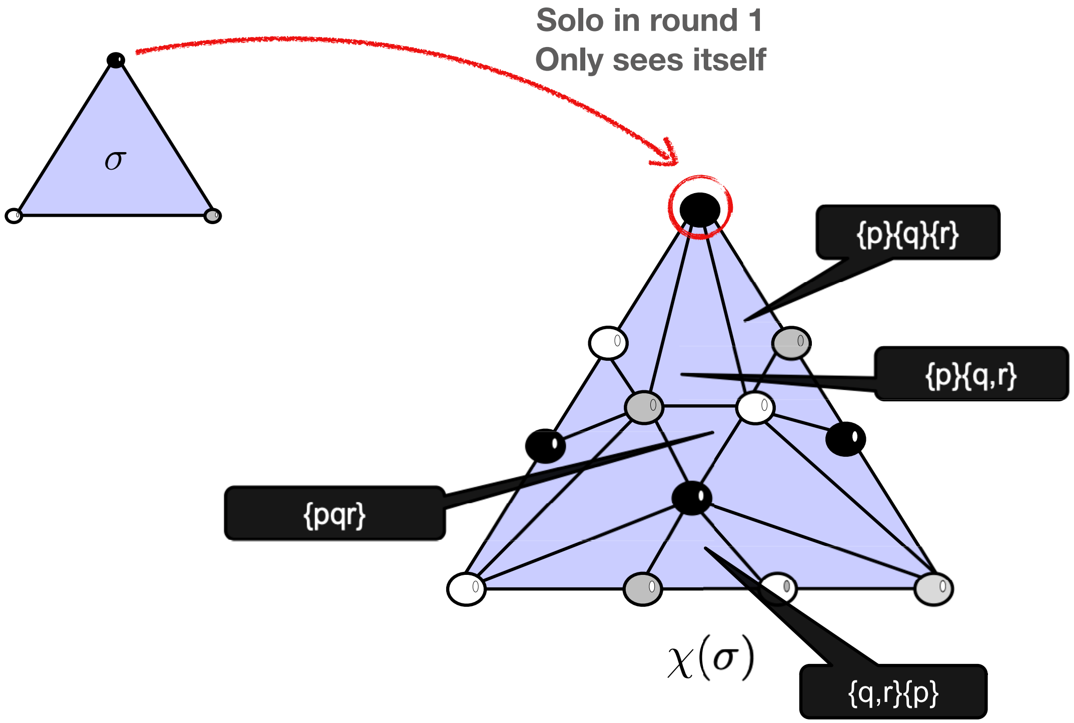
{p}{q}{r}

{p}{q,r}

{pqr}

{q,r}{p}

Topological Interpretation of IIS



Solo in round 1
Only sees itself

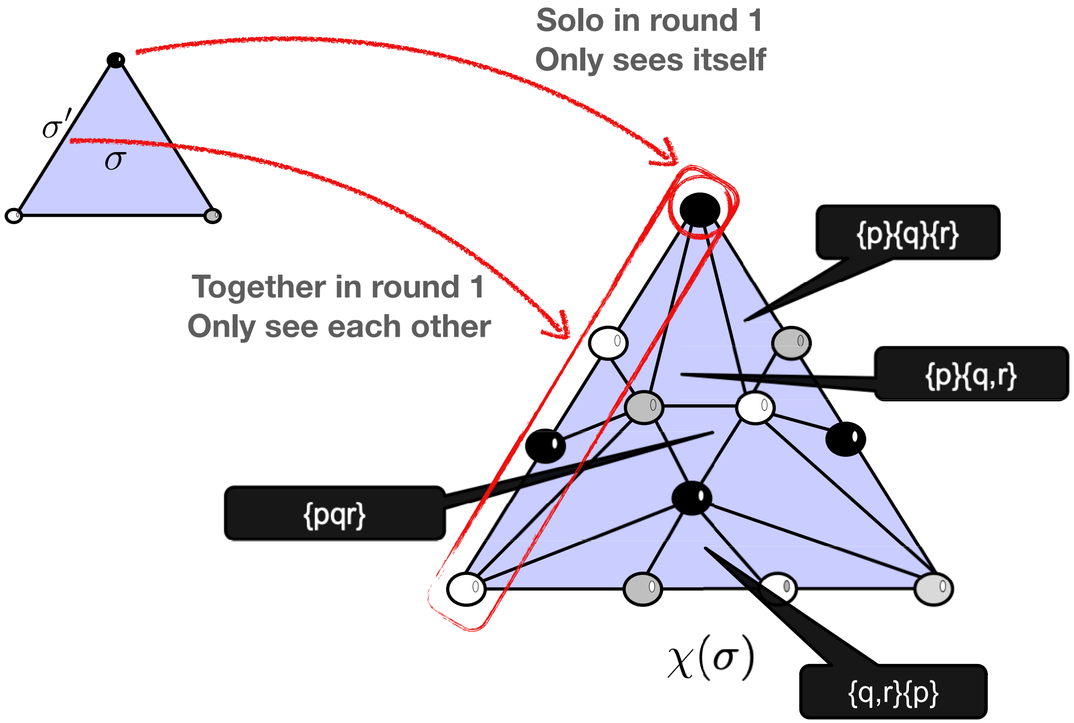
{p}{q}{r}

{p}{q,r}

{pqr}

{q,r}{p}

Topological Interpretation of IIS



Solo in round 1
Only sees itself

Together in round 1
Only see each other

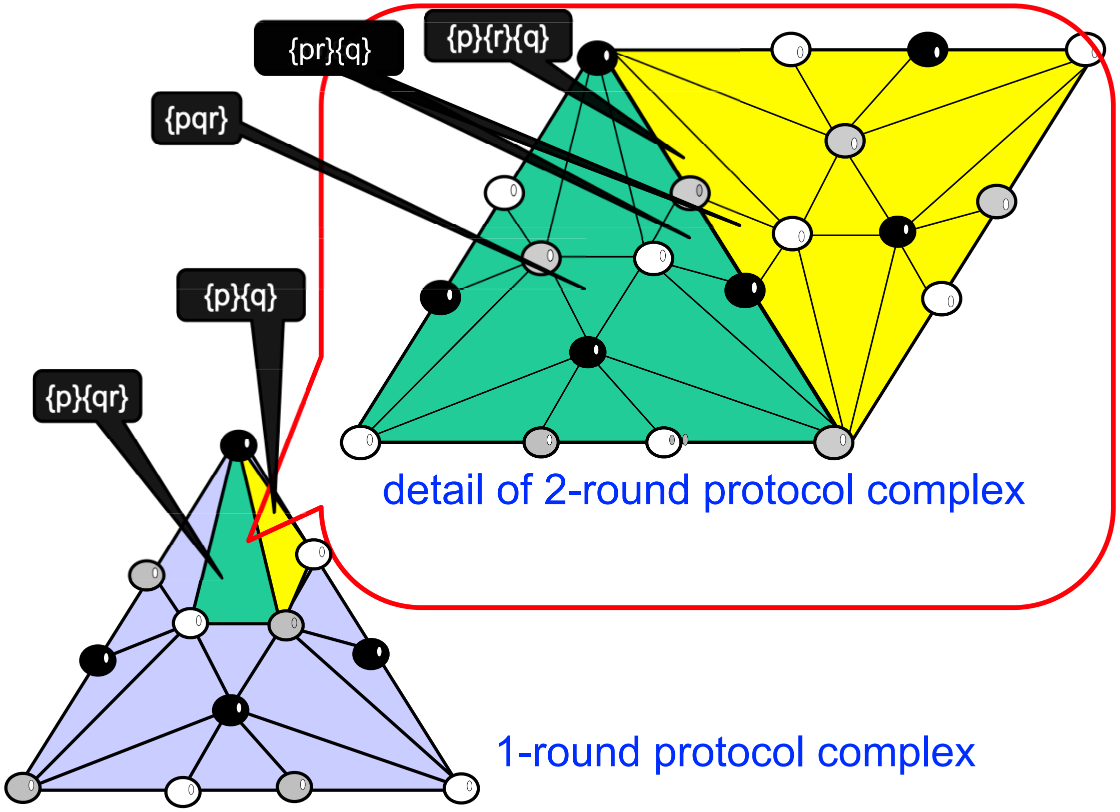
{p}{q}{r}

{p}{q,r}

{pqr}

{q,r}{p}

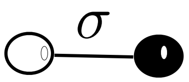
Topological Interpretation of IIS



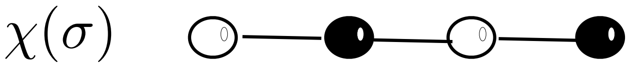
detail of 2-round protocol complex

1-round protocol complex

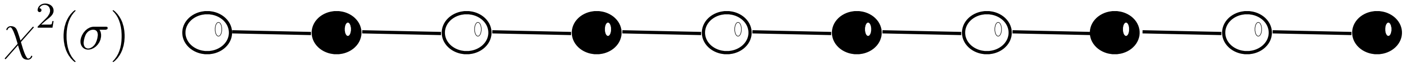
Topological Interpretation of IIS



input simplex (initial configuration)

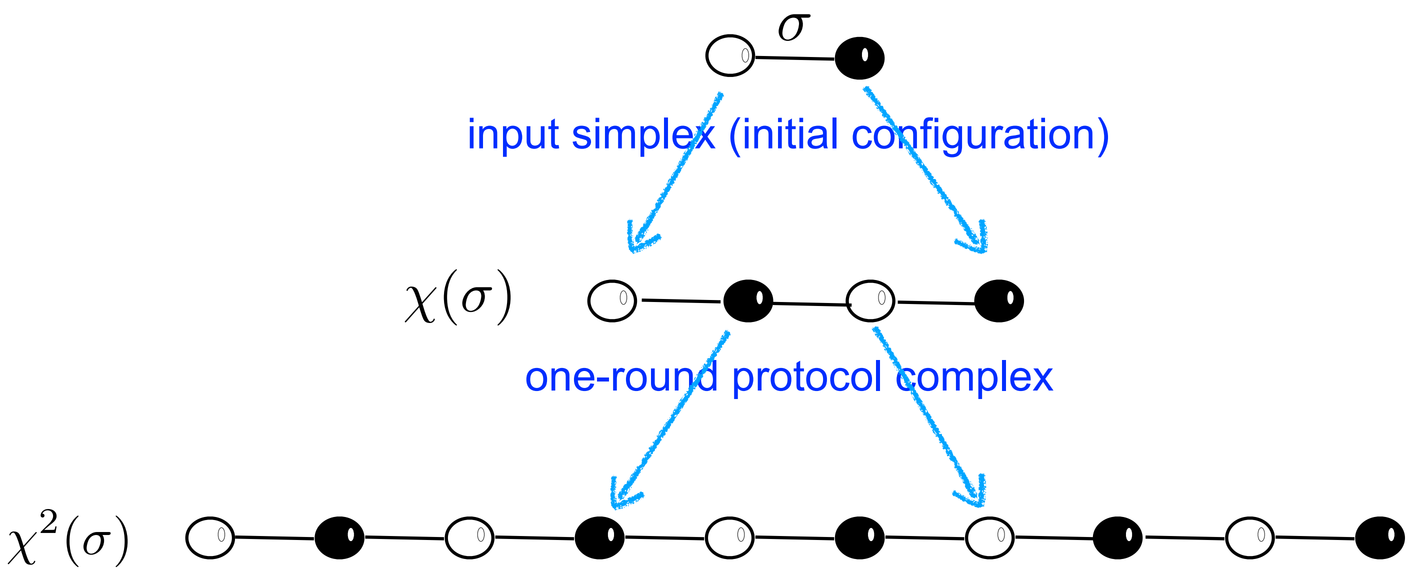


one-round protocol complex



two-round protocol complex

Topological Interpretation of IIS

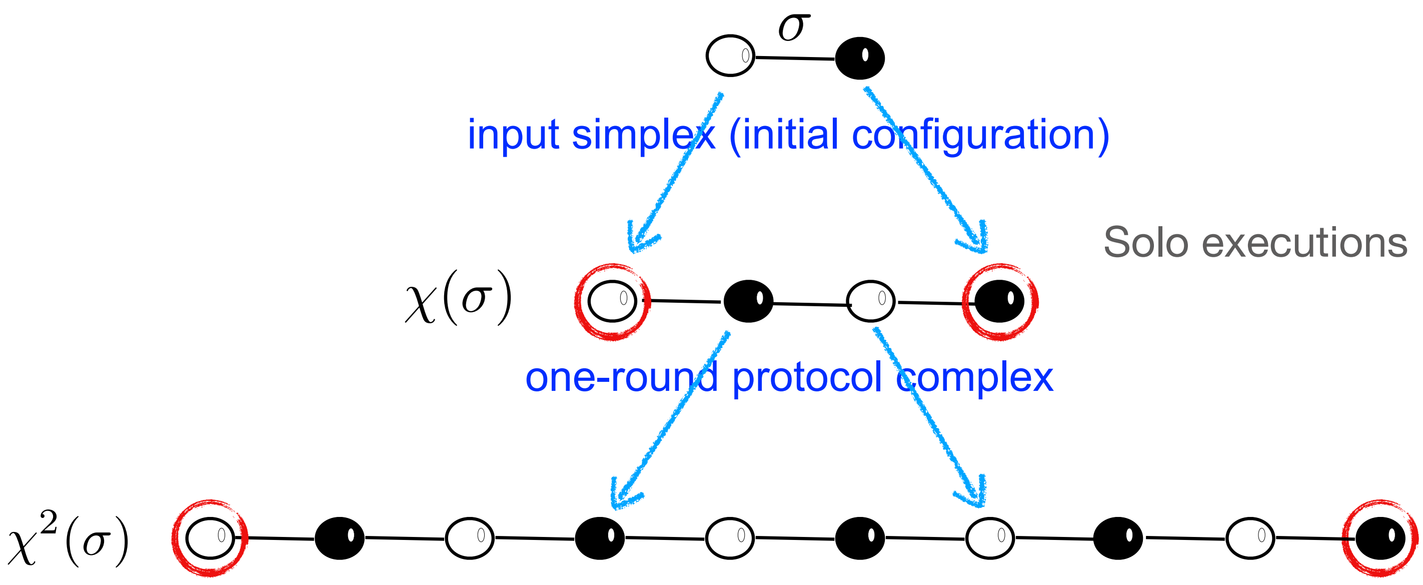


input simplex (initial configuration)

one-round protocol complex

two-round protocol complex

Topological Interpretation of IIS



input simplex (initial configuration)

Solo executions

one-round protocol complex

two-round protocol complex

Bounded Termination

- Task: Input/Output relation (consensus, set agreement, renaming)
- Task with **finite number** of input configurations => **Bounded termination**
- Processes decide/terminate after R rounds; R is unknown a priori

```
Protocol Generic(input: v_i)
  view_i = v_i
  for r = 1 up to R do
    view_i = IS(M[r], view_i)
  endfor
  decide dec(view_i)
endProtocol
```

Task Solvability

- $\chi^m(\sigma)$ = complex with all configuration after m IIS rounds starting at configuration σ
- Protocol = function from vertices of $\chi^R(\sigma)$ (R-round views) to decisions

The protocol solves a task T $\Leftrightarrow$ decisions in simplexes of $\chi^R(\sigma)$ satisfy T's specification, for every input simplex σ

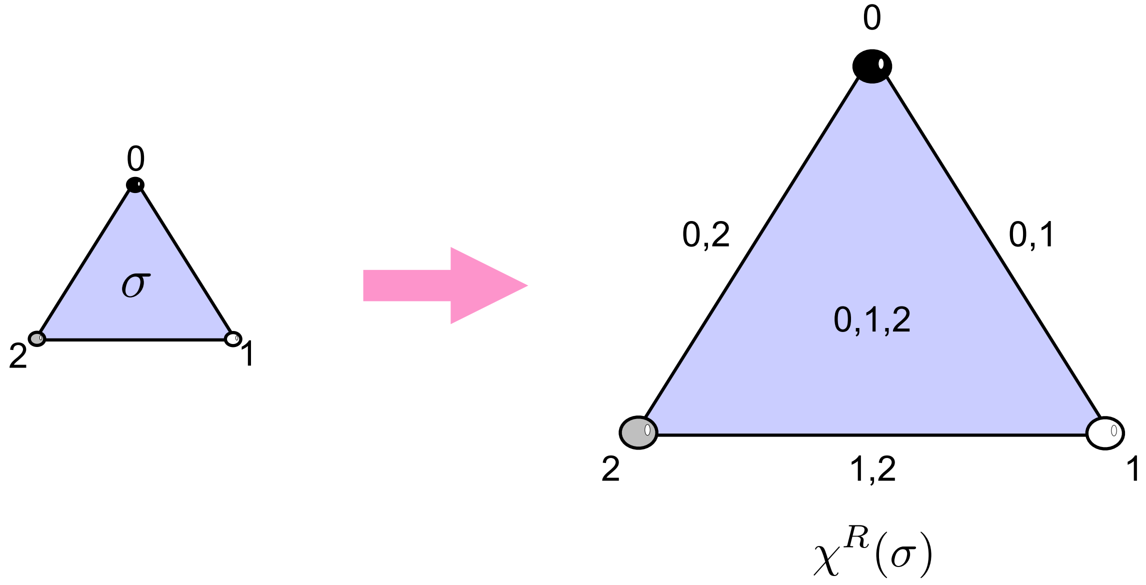
Task Solvability

- $\chi^m(\sigma)$ = complex with all configuration after m IIS rounds starting at configuration σ
- Protocol = function from vertices of $\chi^R(\sigma)$ (R-round views) to decisions

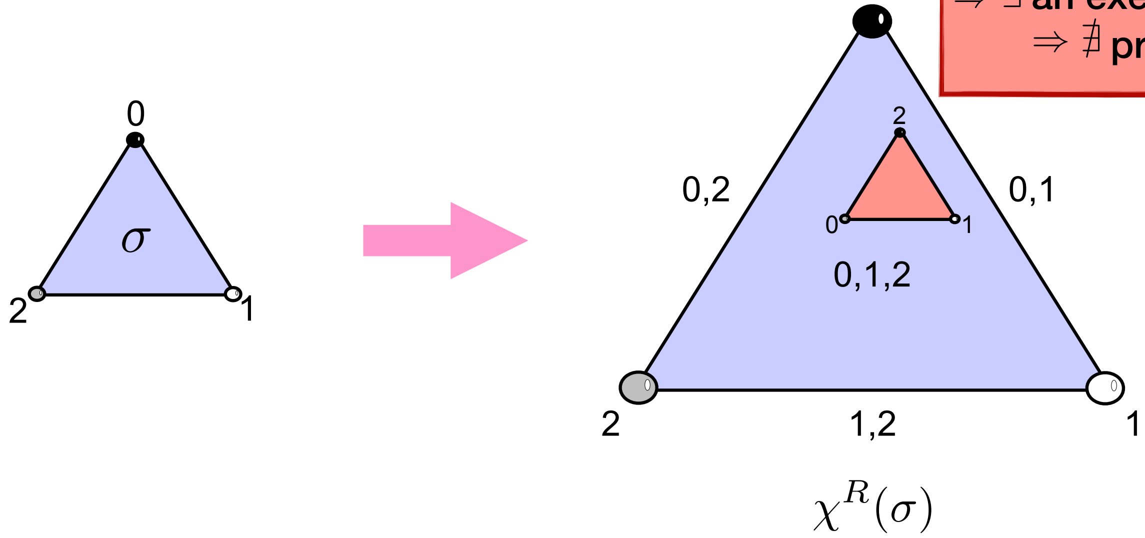
The protocol solves a task T $\Leftrightarrow$ decisions in simplexes of $\chi^R(\sigma)$ satisfy T's specification, for every input simplex σ

Task T is solvable in IIS $\Leftrightarrow$ T is solvable in standard wait-free read/write shared memory model

Global Impossibility Proof for 2-Set Agreement



Global Impossibility Proof for 2-Set Agreement



<u>Sperner's lemma:</u> there is a simplex with all colors

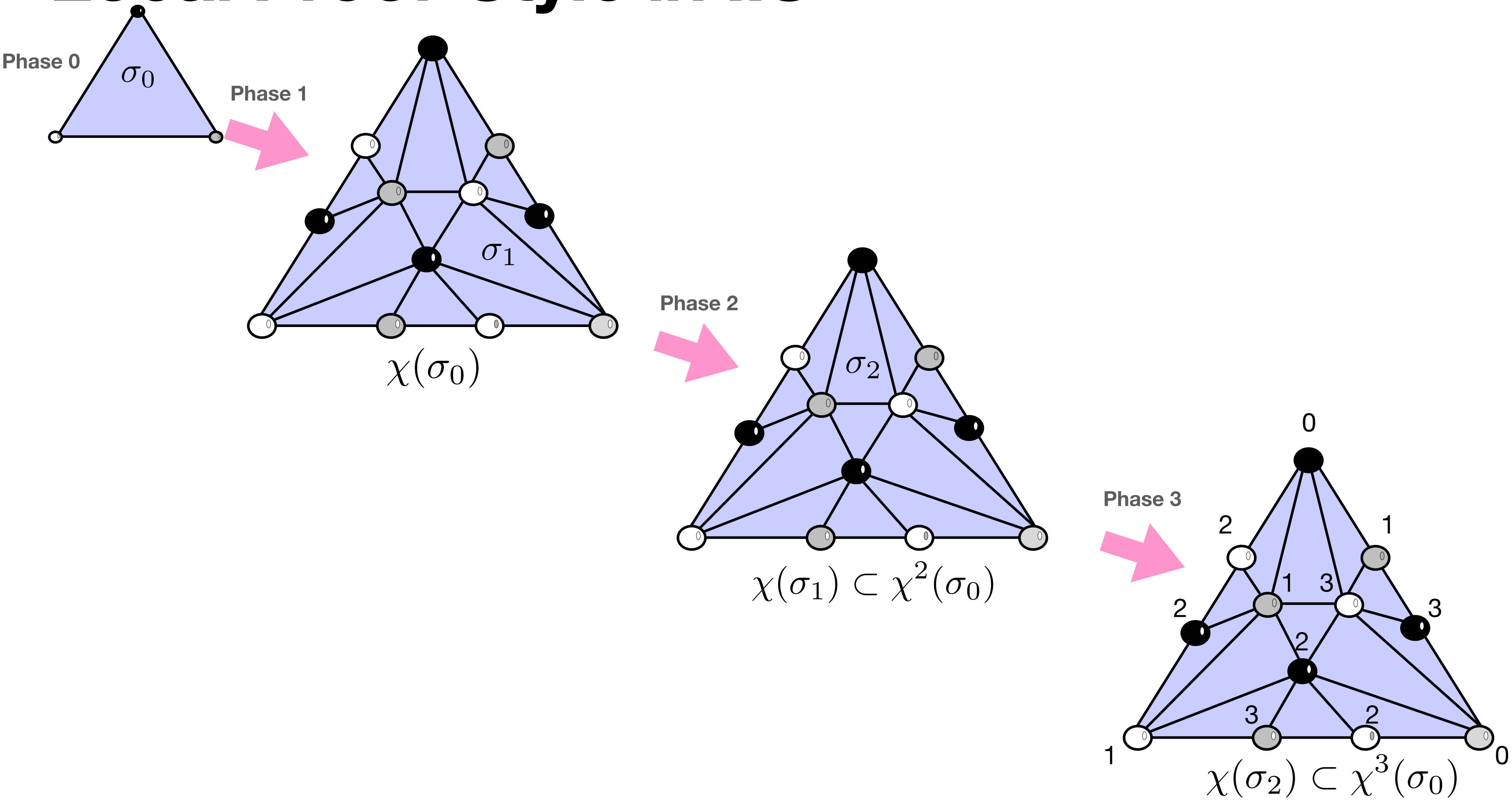
$\Rightarrow \exists$ an execution with 3 distinct decisions

$\Rightarrow \nexists$ protocol for 2-set agreement

Local Proof-Style in IIS

- For j-round simplex (configuration) $\sigma' \in \chi^j(\sigma)$,
  $\chi^{R-j}(\sigma')$ = R-round simplexes at the end of σ'-only extensions with R-j rounds
- Valency of σ': set with all decisions in $\chi^{R-j}(\sigma')$
- Phase i > 0:
  - Starts with a (i-1)-round simplex $\sigma_{i-1} \in \chi^{i-1}(\sigma_0)$
  - All successors after one round in $\chi(\sigma_{i-1}) \subset \chi^i(\sigma_0)$ (i-round simplexes)
  - The hypothetical protocol gives all valencies in $\chi(\sigma_{i-1})$
  - Pick a simplex σ_i in $\chi(\sigma_{i-1})$
- Phase 0: Pick σ_0 using all valencies of input simplexes (initial configurations)
- When i = R, the protocol **must reveal all decisions** in $\chi(\sigma_{R-1}) \subset \chi^R(\sigma_0)$
- The protocol **does not exist** if valencies or decisions are inconsistent

Local Proof-Style in IIS



Phase 0

σ_0

Phase 1

σ_1

$\chi(\sigma_0)$

Phase 2

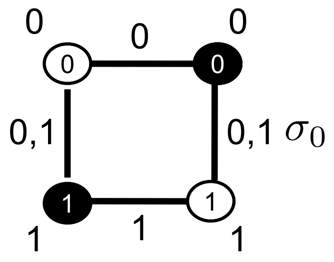
σ_2

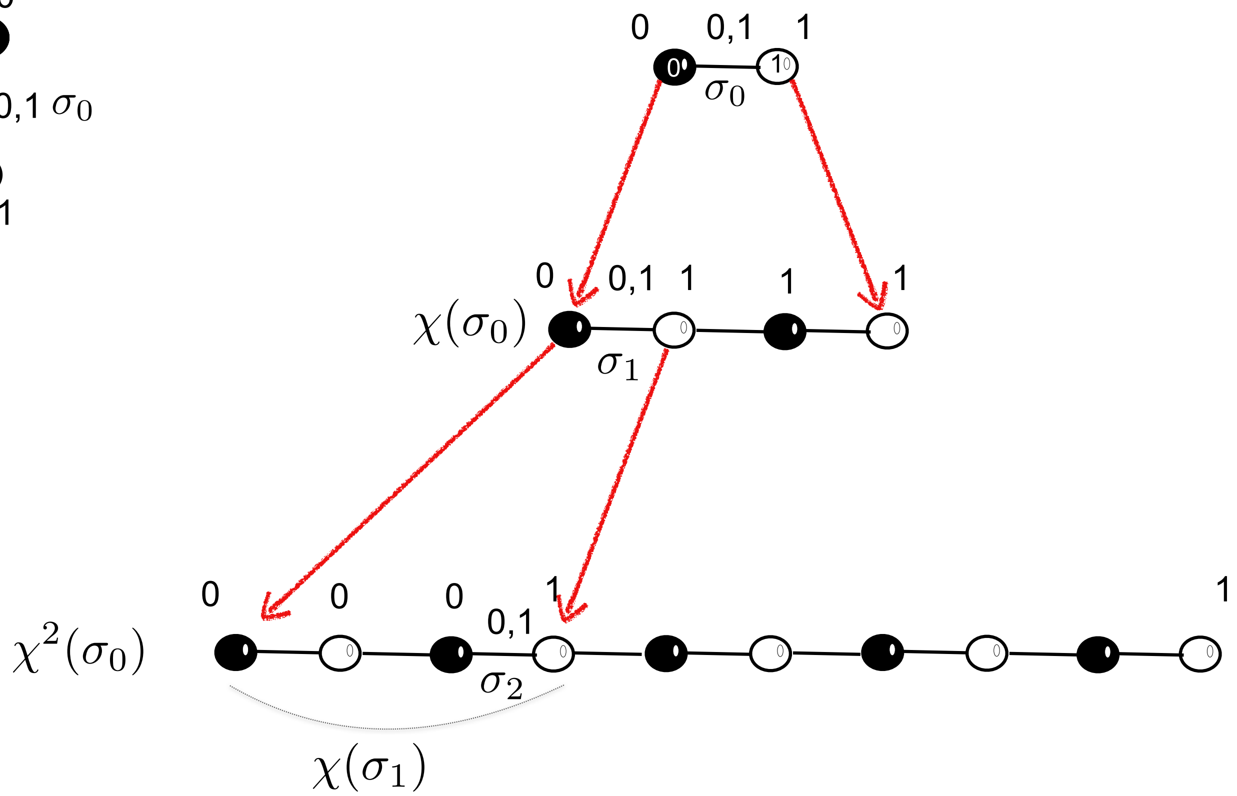
$\chi(\sigma_1) \subset \chi^2(\sigma_0)$

Phase 3

$\chi(\sigma_2) \subset \chi^3(\sigma_0)$

Local Impossibility Proof for Consensus





Local Proof for Set Agreement?

- Set agreement is impossible so **there must be mistake**s, i.e. simplexes with more than k distinct decisions

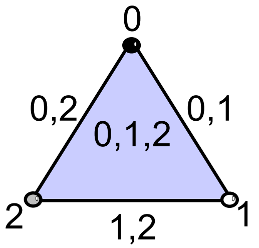
- Can a protocol **hide** its unavoidable mistakes?

- **How** to hide your mistakes?

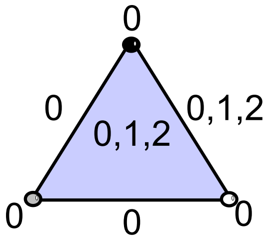
- What needs to be **avoided**?

Local Proof for Set Agreement?

- Fully-valent. Equivalent of bivalent for set agreement.
  Sperner's lemma => there is a mistake in $\chi(\sigma_{R-1})$



- There are more cases.
  Sort of Sperner's lemma => there is a mistake in $\chi(\sigma_{R-1})$

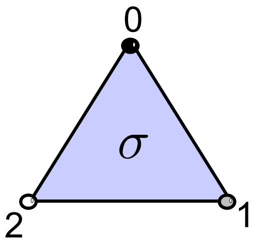


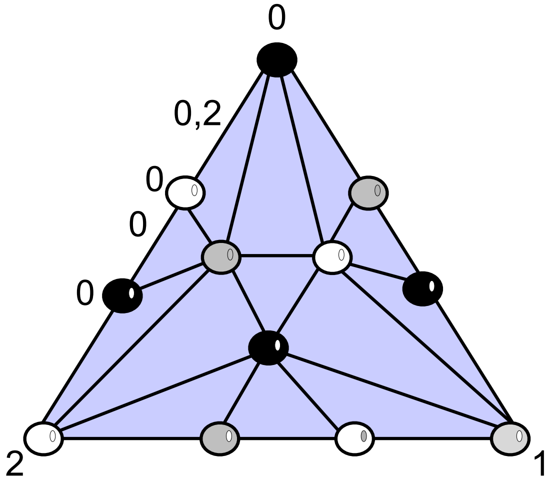
Local Proof for Set Agreement?

- Key observation: distinct protocols induce same valencies
- The hypothetical protocol can be **more than just one protocol**
- Each protocol has unavoidable mistakes ‘in different places’
- Strategy: pick the decision of a protocol with **no local mistakes** in $\chi(\sigma_{R-1})$
- Our formalization: **Valency tasks** and **local solvability**

Valency Tasks for Set Agreement

- A task like consensus, set agreement or renaming
- Input simplexes = simplexes in $\chi^{R-1}(\sigma)$ for a set agreement input simplex σ, $R > 1$
- Each simplex has a valency satisfying validity, i.e. valency is a subset of proposals

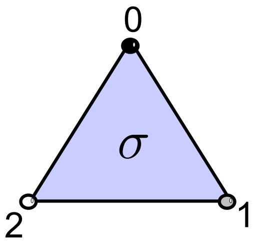


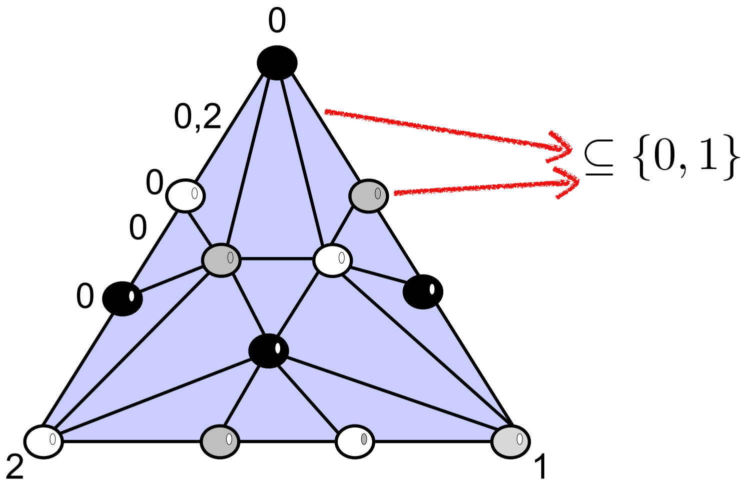


$\chi(\sigma)$

Valency Tasks for Set Agreement

- A task like consensus, set agreement or renaming
- Input simplexes = simplexes in $\chi^{R-1}(\sigma)$ for a set agreement input simplex σ, $R > 1$
- Each simplex has a valency satisfying validity, i.e. valency is a subset of proposals

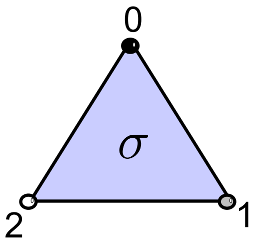


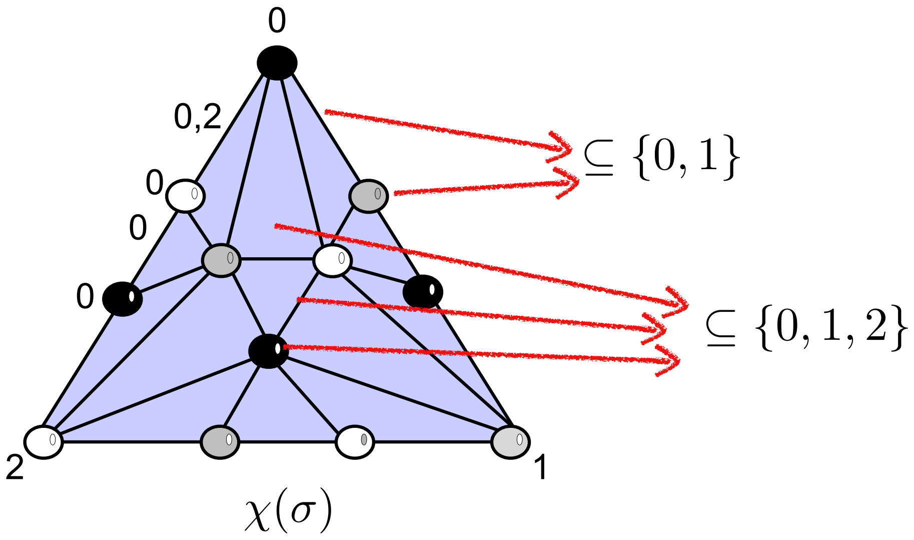


$\chi(\sigma)$

Valency Tasks for Set Agreement

- A task like consensus, set agreement or renaming
- Input simplexes = simplexes in $\chi^{R-1}(\sigma)$ for a set agreement input simplex σ, $R > 1$
- Each simplex has a valency satisfying validity, i.e. valency is a subset of proposals





k-Local Solvability for Set Agreement

- Valency task $\langle \sigma, \chi^{R-1}(\sigma), val \rangle$

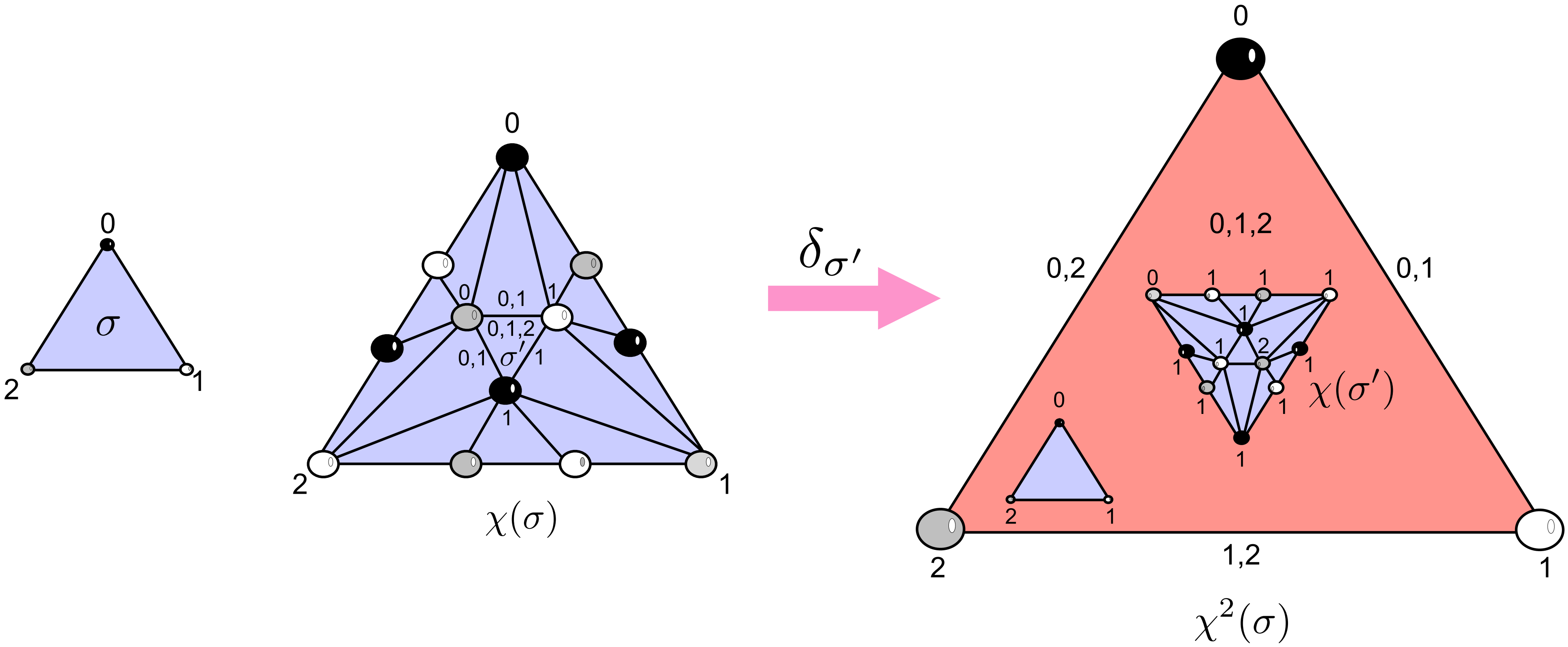
- It is <u>k-locally solvable</u> if $\forall \sigma' \in \chi^{R-1}(\sigma)$, there is a R-round protocol $\delta_{\sigma'} : V(\chi^R(\sigma)) \rightarrow in(\sigma)$ such that:
    - Valency-validity: decisions satisfy valencies specified by val
    - k-Local agreement: no more than k decisions in every simplex in $\chi(\sigma') \subset \chi^R(\sigma)$

- <u>Rough idea:</u> a bunch of protocols ‘solve each part’ of $\chi^R(\sigma)$

- val satisfies validity $\Rightarrow$ $\delta_{\sigma'}$ is a Sperner coloring

- Sperner’s lemma $\Rightarrow$ $\delta_{\sigma'}$ has mistakes <u>somewhere</u> in $\chi^R(\sigma)$ but <u>not in</u> $\chi(\sigma')$

2-Local Solvability for Set Agreement



Main Result

$\forall R > 1$, there are valency tasks $\langle \sigma, \chi^{R-1}(\sigma), val \rangle$ for set agreement that are (n-1)-locally solvable, for every input simplex σ

$\forall R > 1$, there is no valency task $\langle \sigma, \chi^{R-1}(\sigma), val \rangle$ for consensus that is 1-locally solvable, whenever σ has distinct inputs

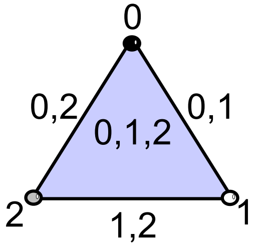
No Local Style-Proofs for Set Agreement - Valencies

- For simplicity, every process starts with its ID (inputless version)

- There is one input simplex $\sigma = \{(P_0, 0), (P_1, 1), \ldots, (P_n, n)\}$

- $\forall \sigma' \subset \sigma$, valency of σ' = inputs in σ'

- σ is a fully-valent configuration



- Pick any $R > 1$ and consider an (n-1)-locally solvable valency task $\langle \sigma, \chi^{R-1}(\sigma), val \rangle$

- For each $i \in \{1, \ldots, R-2\}$, set valencies to the simplexes in $\chi^i(\sigma)$ that are <u>compatible</u> with val (not trivial, not super hard)

No Local Style-Proofs for Set Agreement - Strategy

- Strategy:
  - In phase $i = 0$, $\sigma_0 = \sigma$
  - In phase $i \in \{1, \ldots, R-1\}$, reply the valencies in $\chi(\sigma_{i-1}) \subset \chi^i(\sigma)$
  - In phase $i = R$, reply the decisions in $\chi(\sigma^{R-1}) \subset \chi^R(\sigma)$ by protocol $\delta_{\sigma_{R-1}}$
- Existence of $\delta_{\sigma_{R-1}}$ due to local solvability $\langle \sigma, \chi^{R-1}(\sigma), val \rangle$
- No more than n-1 distinct decisions in $\chi(\sigma^{R-1}) \subset \chi^R(\sigma)$
- No local impossibility proof for set agreement QED
- R and the valencies can be revealed in advance $\Rightarrow$ no adaptiveness is needed

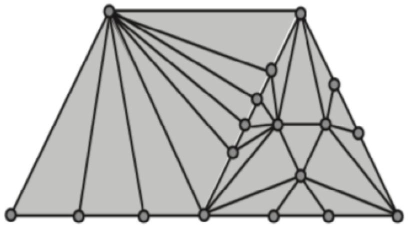
Variants of Local Proof-Style in IIS

- R does not need to be unknown
- Valencies do not need to be unknown
- Pick more than one simplex in each phase (but not a lot)
- Successors after several rounds in the future instead of just one
- Even go all the way up to one round before decision

Differences with Alistarh, Aspnes, Ellen, Gelashvili and Zhu

- Interaction between a **protocol** and a **prover**
- Each phase starts with a finite execution E
- The prover asks **decision or valency queries** to the protocol
- After finitely many queries, the prover **commits** on a finite extension of E
- The **prover wins** if it finds a contradiction or performs infinitely many phases
- Otherwise the **protocol wins**
- There is **no impossibility extension based-proof** if there is a protocol that wins *against any prover*

Differences with Alistarh, Aspnes, Ellen, Gelashvili and Zhu



- Processes can decide at distinct rounds
- **Non-uniform IIS** (NIIS) model.
  Complexes: non-uniform subdivisions
- Their result: there is no extension-based proof for the impossibility of k-set agreement in the NIIS model
- They **do not allow** bounded termination
- Otherwise, prover performs exhaustive search, constructs simplicial complex (non-uniform subdivision) and applies Sperner's lemma
- Not in the spirit of local style-proofs but it is allowed (if bounded termination is assumed)

Wrapping Up

- Simple formalization of local style-proofs in IIS
- Valency tasks and local-solvability
- There are locally solvable valency tasks for set agreement
- $\Rightarrow$ No local impossibility proof for set agreement
- The result holds for unbounded and bounded termination
- (2n-2)-Renaming. Studied through weak symmetry breaking
- Same approach taking care of symmetries of decisions

Future Work

- Variants of local style-proofs
- Other tasks (e.g. approximate agreement)
- Other wait-free shared memory models
- Non-compact models (e.g. t-resilient); bounded termination is an issue
- Models with no round-structure