A Speedup Theorem for Asynchronous Computation

Ami Paz – Université Paris Saclay and CNRS

Joint work with:

Pierre Fraigniaud – Université Paris-Cité and CNRS

Sergio Rajsbaum – Instituto de Matemáticas, UNAM

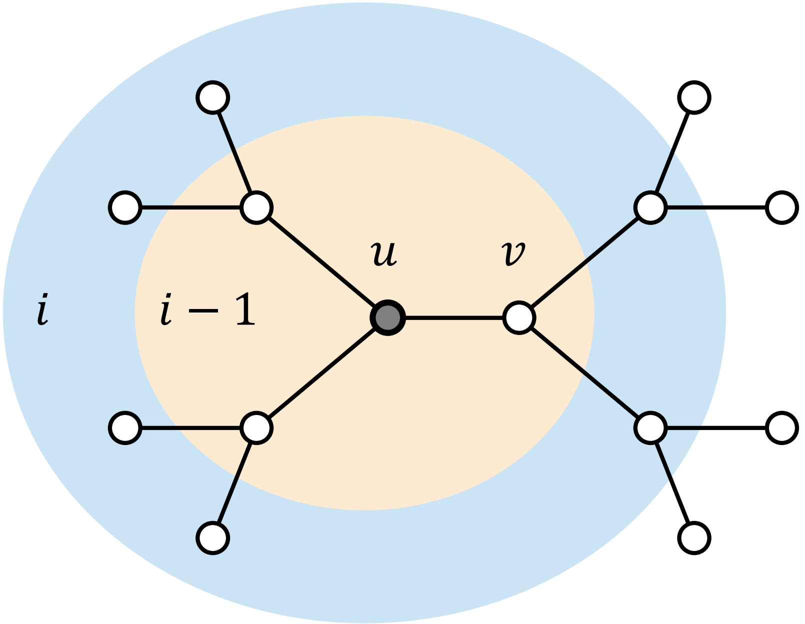
Speedup Theorem
(a.k.a. round elimination)

- Recent technique for LOCAL lower bounds
- Typical problems: Maximal Matching, Sinkless Orientation, Coloring
- We want to show that a problem $P = P_T$ has on T-round algorithm
- High-level idea:

Claim 1: i-round algorithm for problem $P_i \Longrightarrow$
$(i-1)$-round algorithm for another problem P_{i-1}

Claim 2: No 0-round algorithm for P_0

Technique: simulate distance-i view



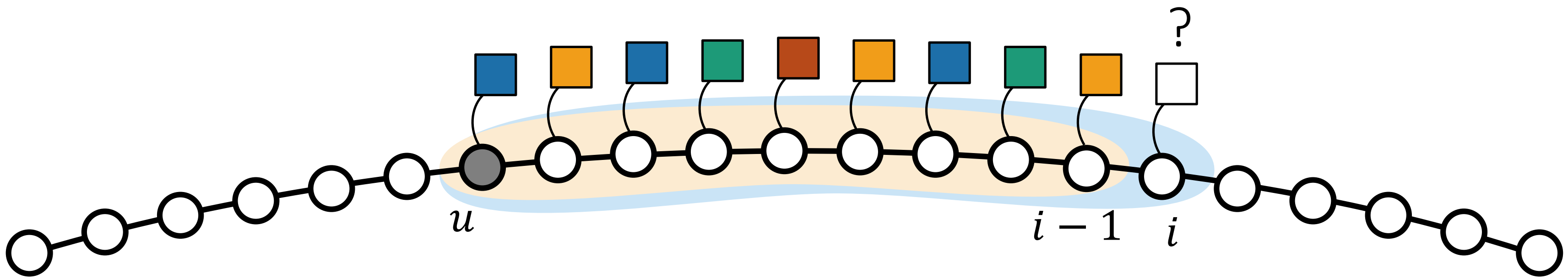
Speedup Theorem – Example

- Linial's lower bound for 3-coloring a ring
- High-level idea:

$\mathrm{col}_{i-1}(u) =$
$\{\mathrm{col}_i(u)$ | possible color of distance-i node$\}$

Claim 1: i-round algorithm for c-coloring $\Longrightarrow$ $(i-1)$-round algorithm for 2^c-coloring

Claim 2: No 0-round algorithm for $(n-1)$-coloring



Round Elimination in Async. Computation?

Shared memory, with iterated model

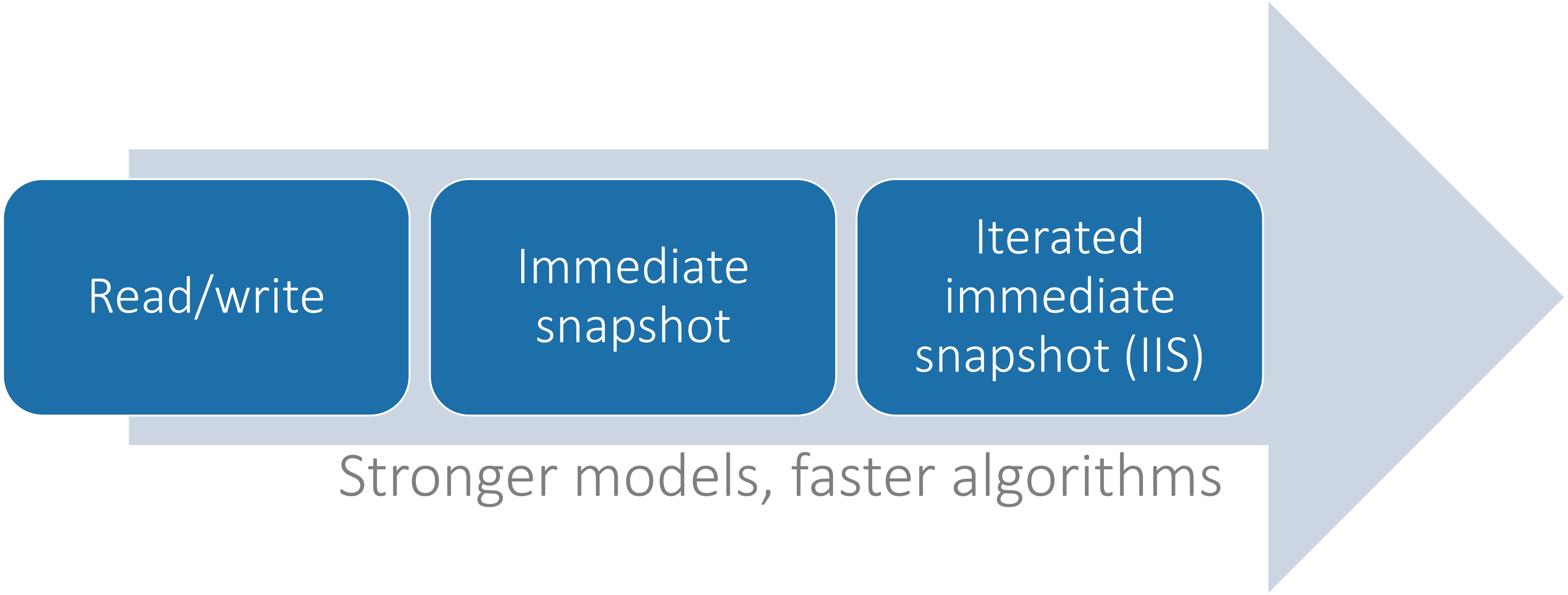
- distance-i is not defined
- But... we can still ask what can process u see in round i
- High-level idea – the same

Claim 1: i-round algorithm for task $\Pi_i \Longrightarrow$
$(i-1)$-round algorithm for another task Π_{i-1}

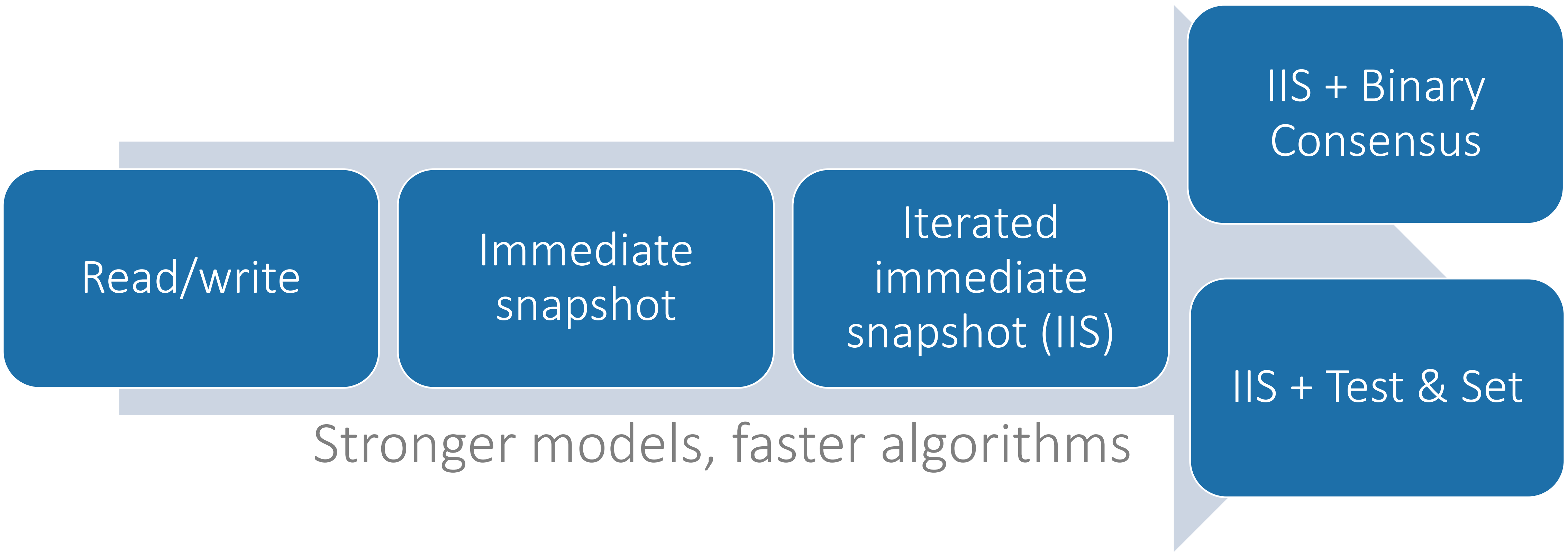
Claim 2: No 0-round algorithm for P_0

Technique: simulate possible views at round i

Shared Memory Models

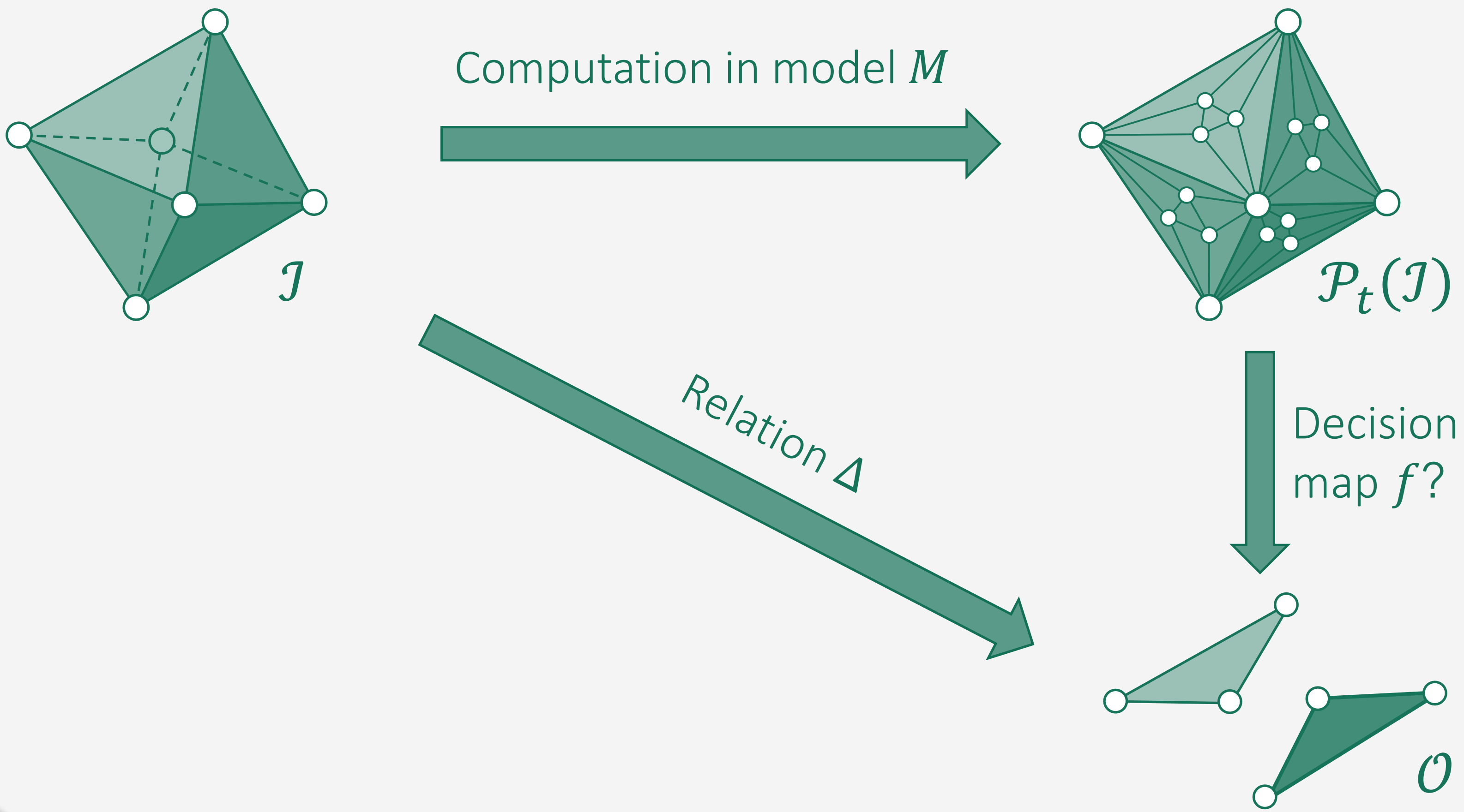


Shared Memory Models



Tools

Tasks and Protocols



Theorem [HS99]:

$(\mathcal{I}, \mathcal{O}, \Delta)$ is t-round solvable

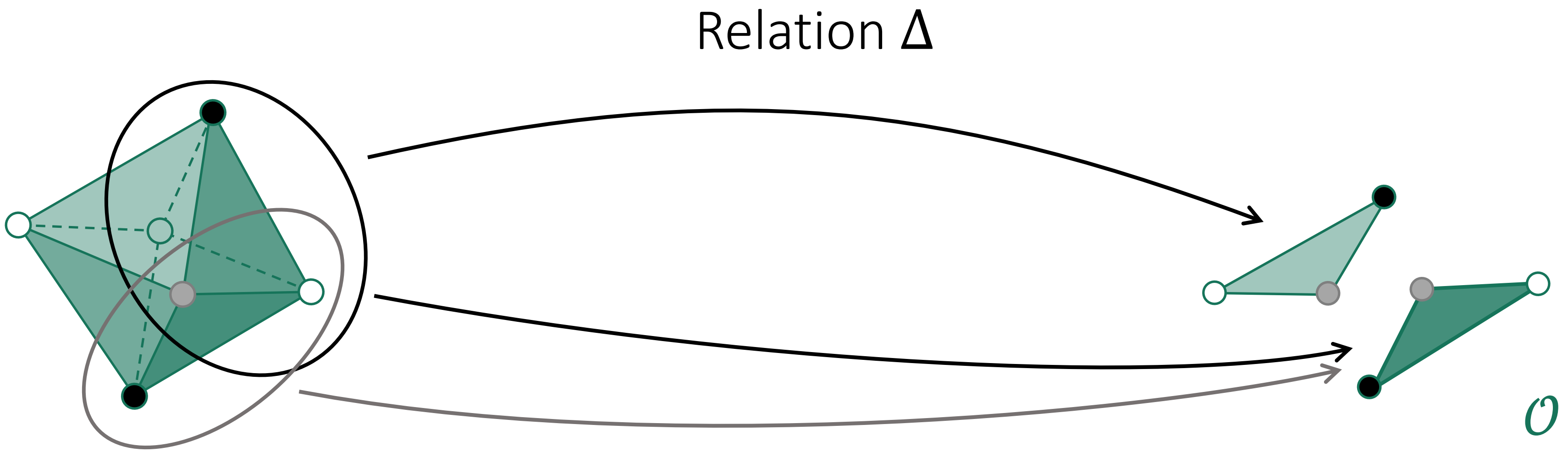
$\Leftrightarrow$

$\exists f : \mathcal{P}_t(\mathcal{I}) \to \mathcal{O}$ agreeing with Δ

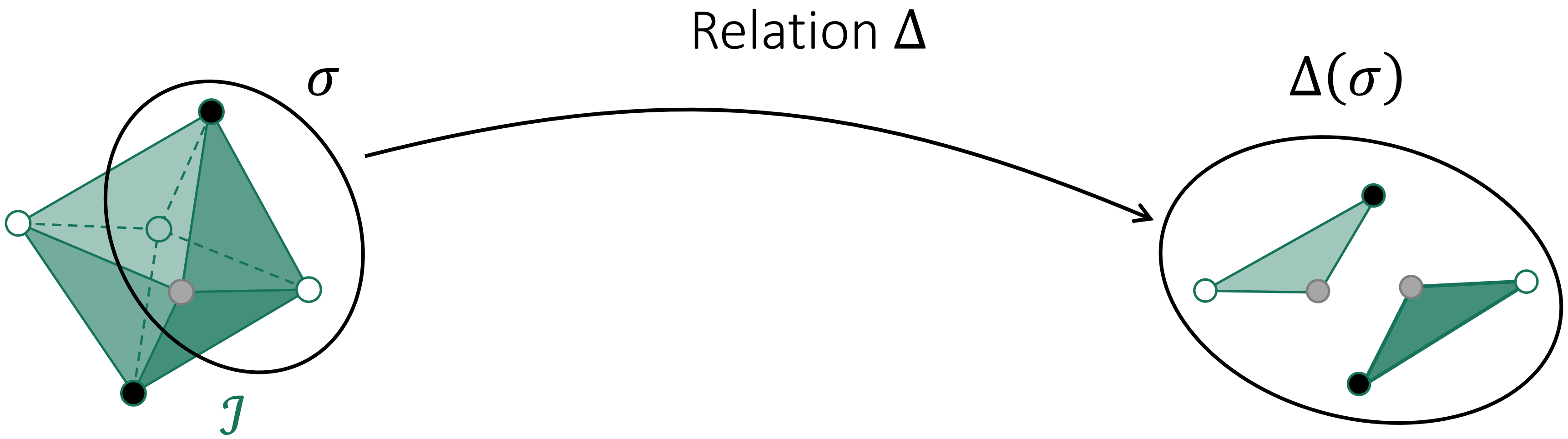
A Task

A task $\Pi = (\mathcal{I}, \mathcal{O}, \Delta)$

A Task

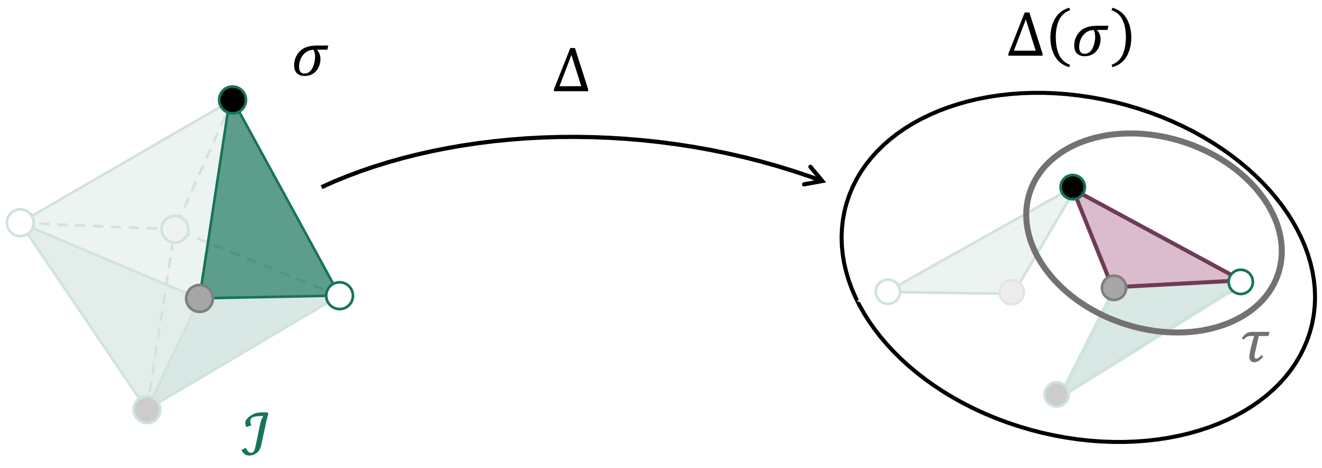


A Task



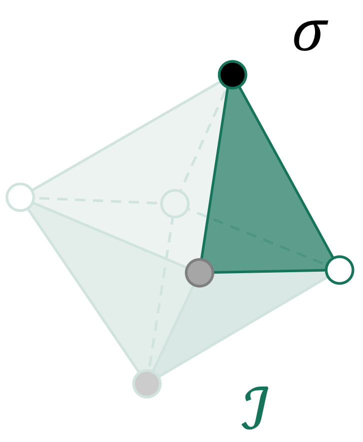
Definition: Local Task

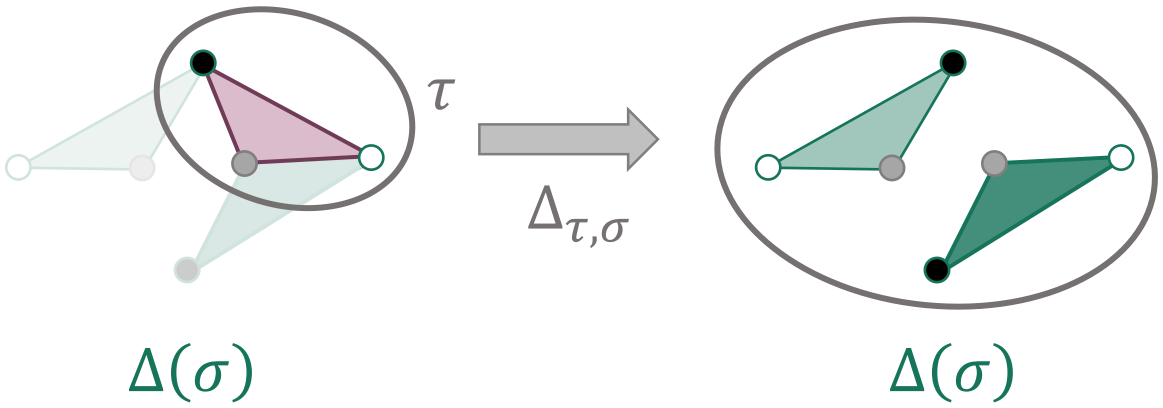
- Given $\sigma \in \mathcal{I}, \tau \subseteq V\big(\Delta(\sigma)\big)$



Definition: Local Task

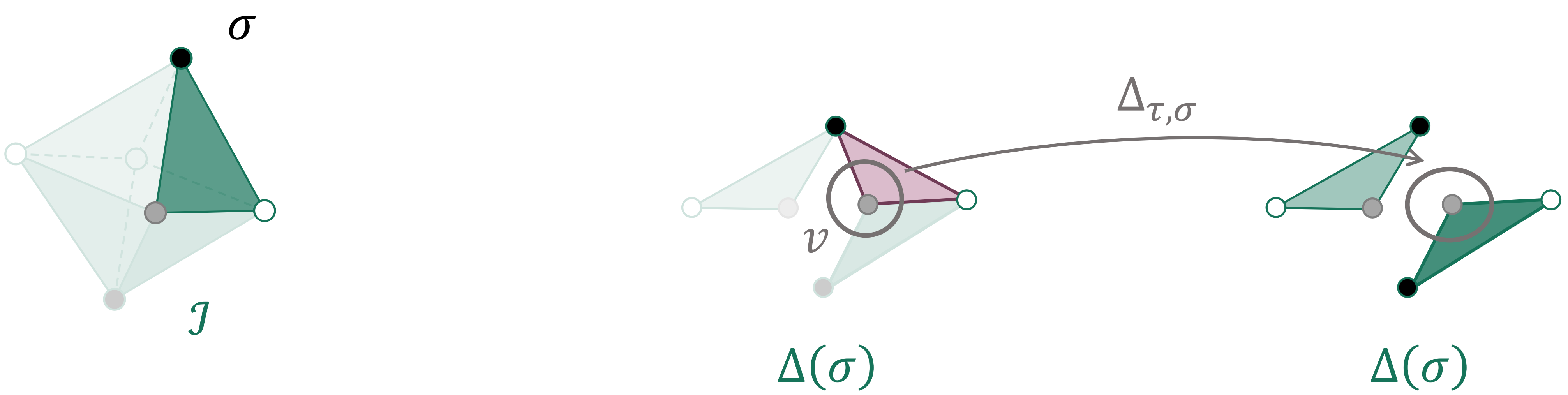
- Given $\sigma \in \mathcal{I}$, $\tau \subseteq V\big(\Delta(\sigma)\big)$, define local task $\Pi_{\tau,\sigma} = \big(\tau, \Delta(\sigma), \Delta_{\tau,\sigma}\big)$





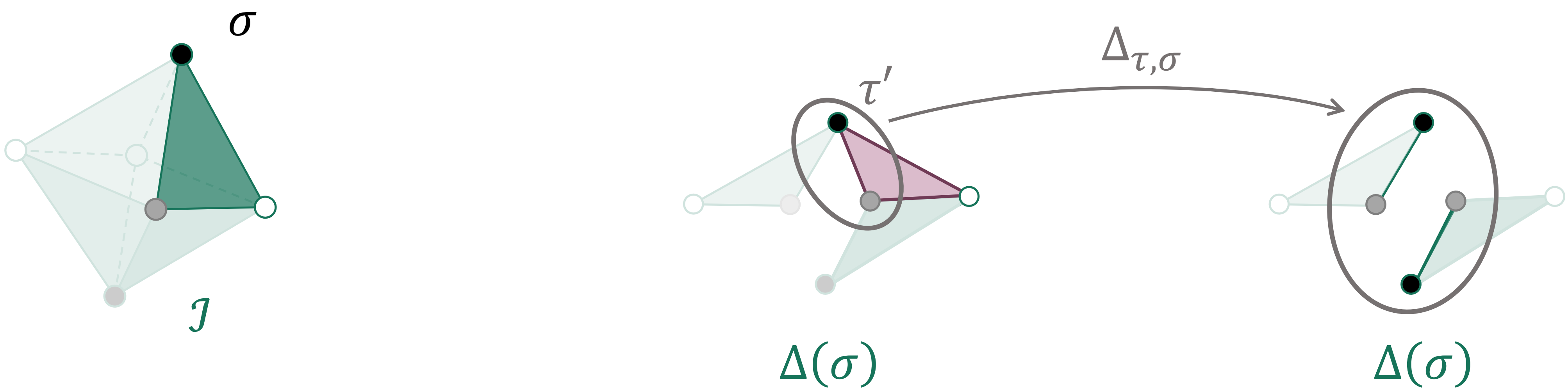
Definition: Local Task

- Given $\sigma \in \mathcal{I}$, $\tau \subseteq V\big(\Delta(\sigma)\big)$, define local task $\Pi_{\tau,\sigma} = \big(\tau, \Delta(\sigma), \Delta_{\tau,\sigma}\big)$, where $\Delta_{\tau,\sigma}$:
  - $\forall v \in \tau$: $\Delta_{\tau,\sigma}(v) = \{v\}$



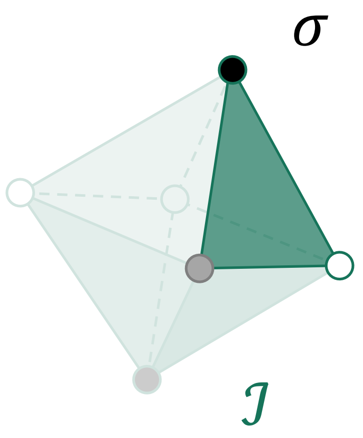
Definition: Local Task

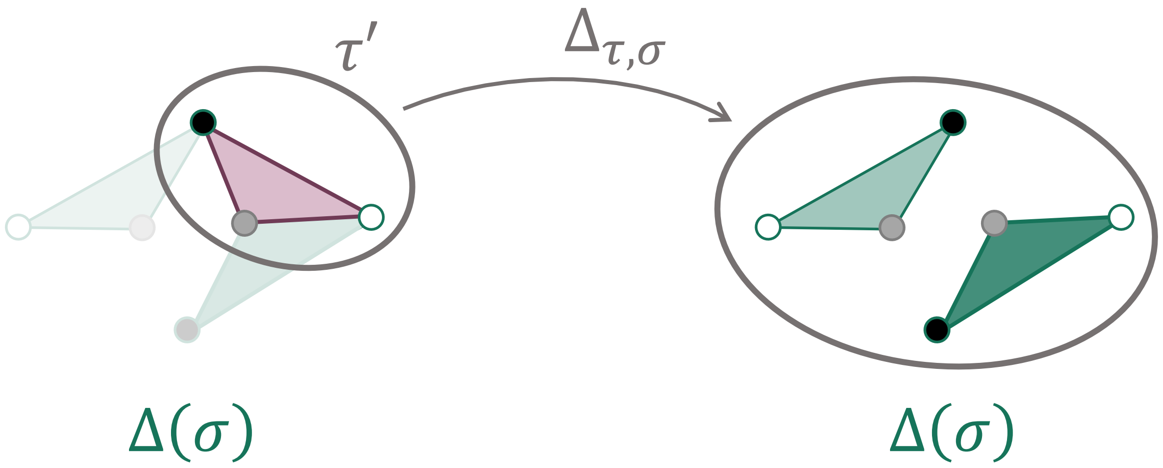
- Given $\sigma \in \mathcal{I}$, $\tau \subseteq V(\Delta(\sigma))$, define local task $\Pi_{\tau,\sigma} = (\tau, \Delta(\sigma), \Delta_{\tau,\sigma})$, where $\Delta_{\tau,\sigma}$:
  - $\forall v \in \tau$: $\Delta_{\tau,\sigma}(v) = \{v\}$
  - $\forall \tau' \subseteq \tau, |\tau'| > 1$: $\Delta_{\tau,\sigma}(\tau') = \text{proj}_{\text{id}(\tau')}\Delta(\sigma)$



Definition: Local Task

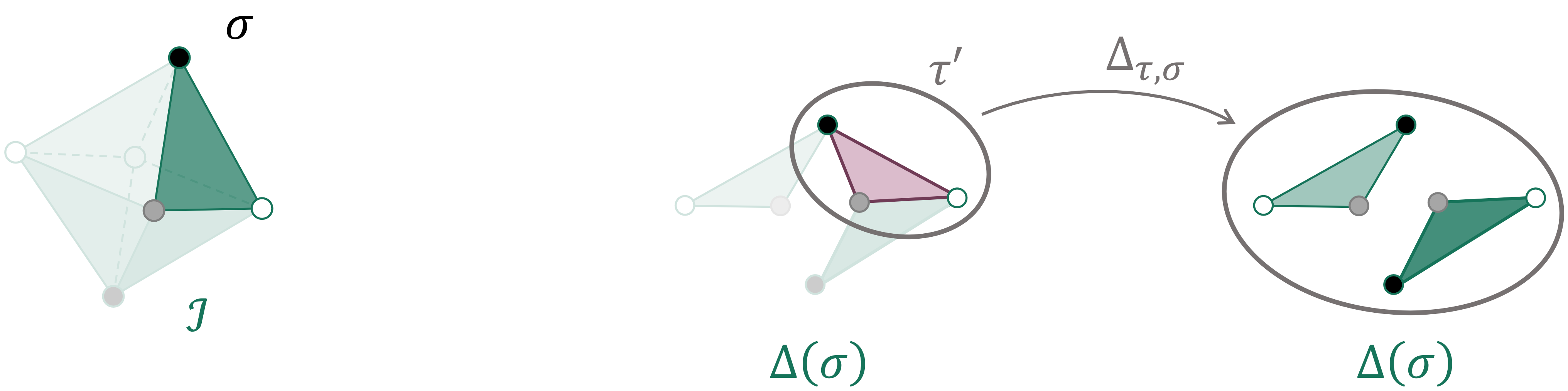
- Given $\sigma \in \mathcal{I}$, $\tau \subseteq V\big(\Delta(\sigma)\big)$, define local task $\Pi_{\tau,\sigma} = \big(\tau, \Delta(\sigma), \Delta_{\tau,\sigma}\big)$, where $\Delta_{\tau,\sigma}$:
  - $\forall v \in \tau$: $\Delta_{\tau,\sigma}(v) = \{v\}$
  - $\forall \tau' \subseteq \tau, |\tau'| > 1$: $\Delta_{\tau,\sigma}(\tau') = \mathrm{proj}_{\mathrm{id}(\tau')}\Delta(\sigma)$





Definition: Local Task

- Given $\sigma \in \mathcal{I}, \tau \subseteq V(\Delta(\sigma))$, define local task $\Pi_{\tau,\sigma} = (\tau, \Delta(\sigma), \Delta_{\tau,\sigma})$, where $\Delta_{\tau,\sigma}$:
  - Vertex: itself
  - Not vertex: anywhere

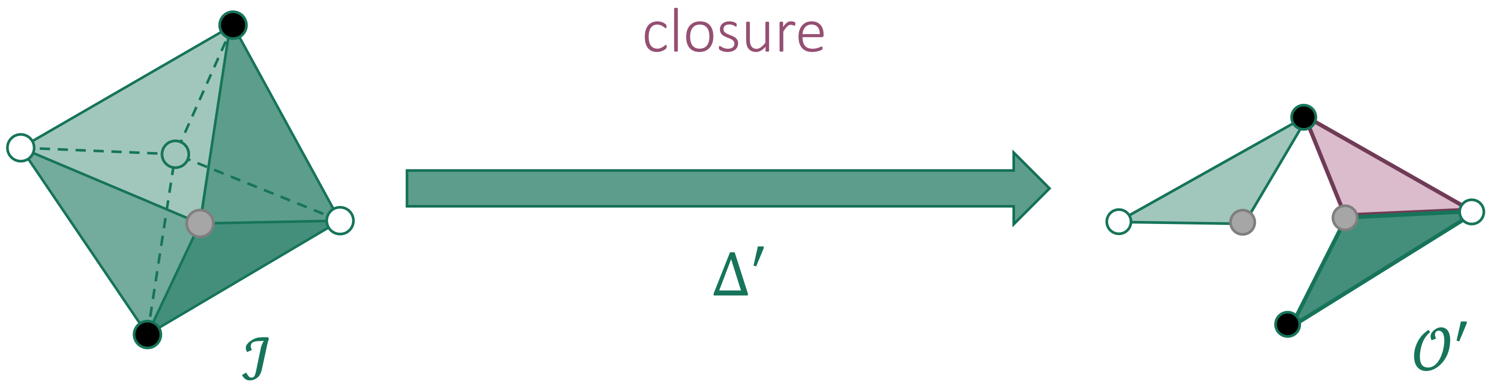


Definition: Closure Task

- The core of the technique: defining Π_{i-1}
- Given a task $\Pi = (\mathcal{I}, \mathcal{O}, \Delta)$ and a model M

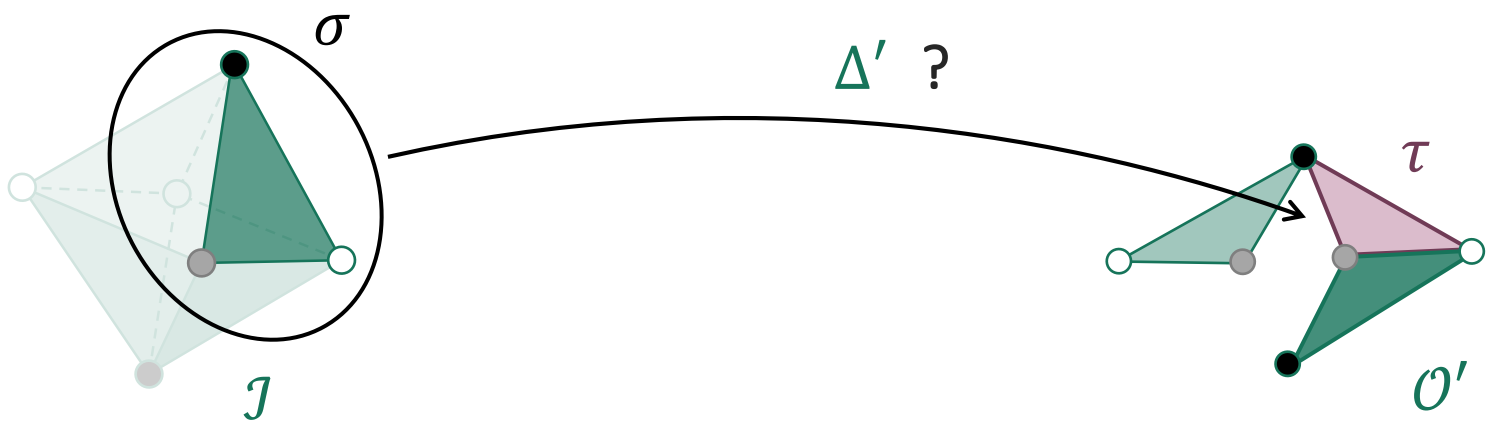
Definition: Closure Task

- The core of the technique: defining Π_{i-1}
- Given a task $\Pi = (\mathcal{I}, \mathcal{O}, \Delta)$ and a model M
  we define a closure task $\mathrm{cl}_M(\Pi) = (\mathcal{I}, \mathcal{O}', \Delta')$



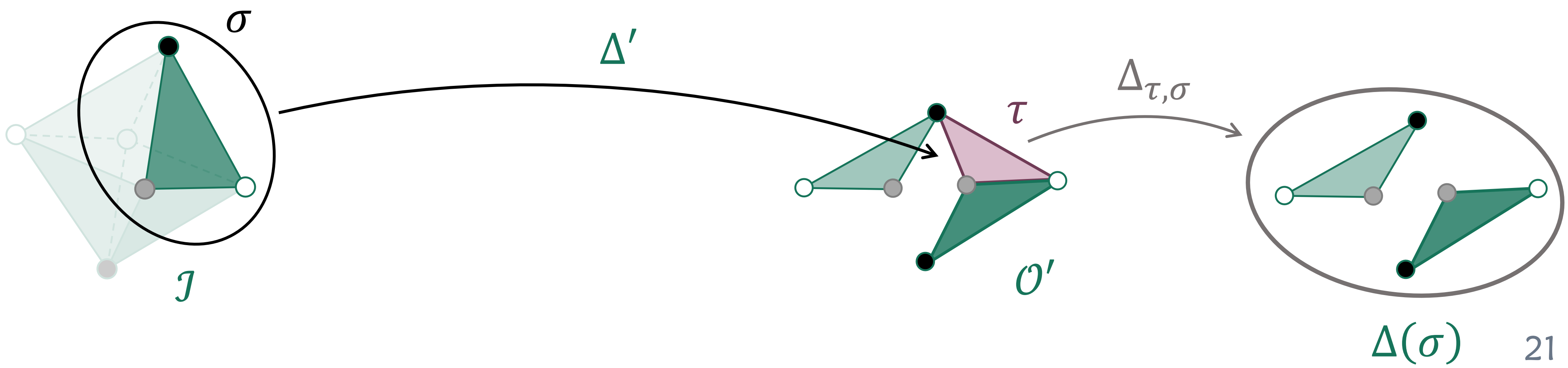
Definition: Closure Task

- The core of the technique: defining Π_{i-1}
- Given a task $\Pi = (\mathcal{I}, \mathcal{O}, \Delta)$ and a model M
  we define a closure task $\mathrm{cl}_M(\Pi) = (\mathcal{I}, \mathcal{O}', \Delta')$ by
  - $\forall \sigma \in \mathcal{I},\ \tau \subseteq V\big(\Delta(\sigma)\big)$: $\tau \in \Delta'(\sigma)$?



Definition: Closure Task

- The core of the technique: defining Π_{i-1}
- Given a task $\Pi = (\mathcal{I}, \mathcal{O}, \Delta)$ and a model M
  we define a closure task $\mathrm{cl}_M(\Pi) = (\mathcal{I}, \mathcal{O}', \Delta')$ by
  - $\forall \sigma \in \mathcal{I},\ \tau \subseteq V(\Delta(\sigma))$: $\tau \in \Delta'(\sigma) \Leftrightarrow$ the local task $\Pi_{\tau,\sigma}$ is 1-round solvable in M

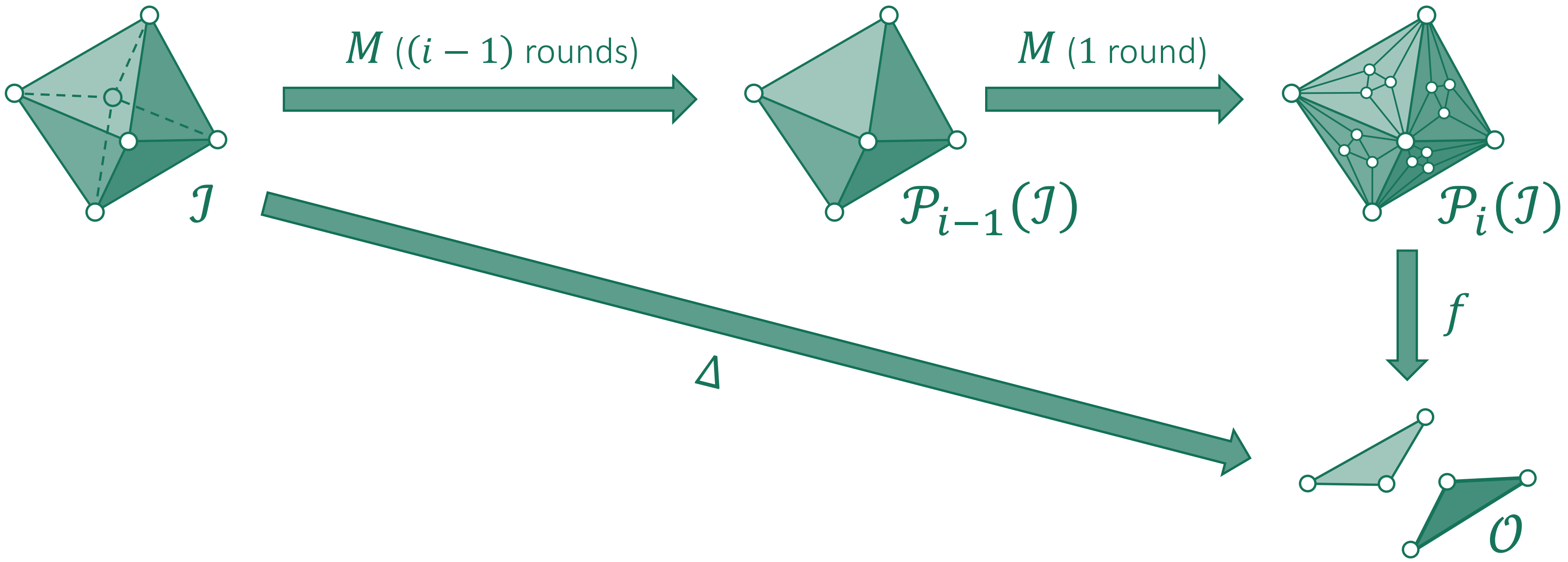


Main Theorem

Theorem: i-round algorithm for task Π in $M \Longrightarrow$ $(i-1)$-round algorithm for $\mathrm{cl}_M(\Pi)$ in M

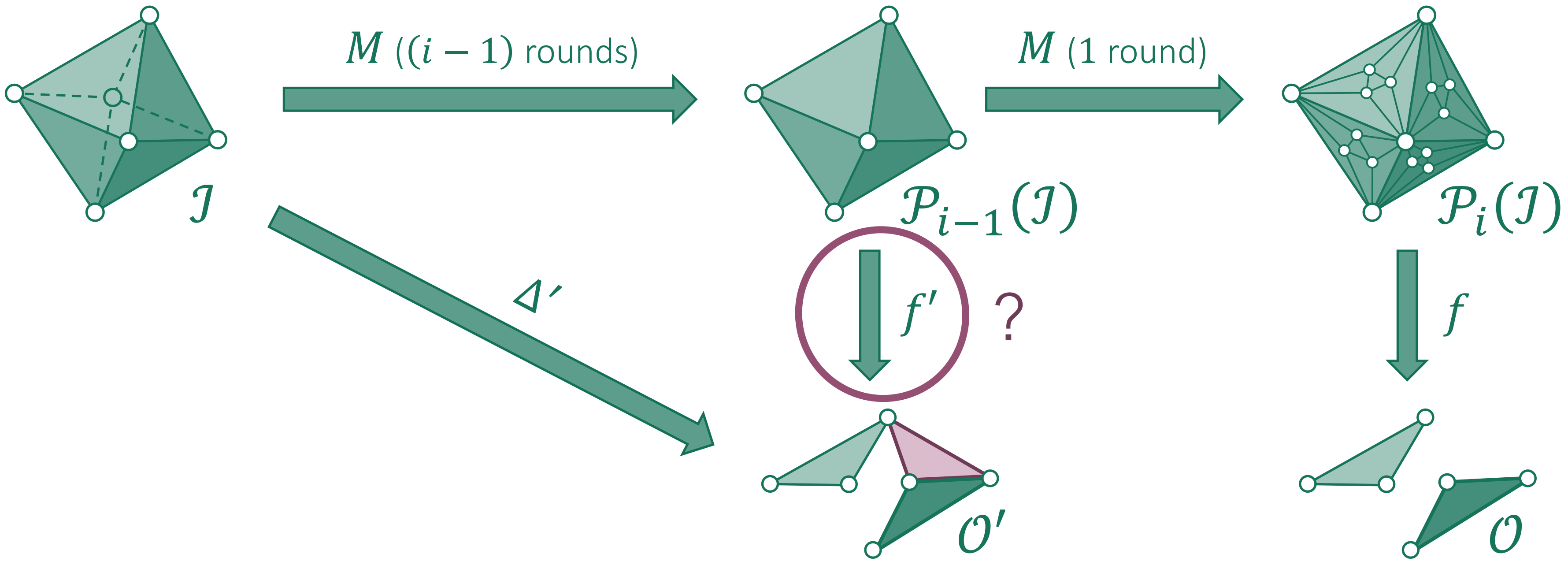
Main Theorem

Theorem: i-round algorithm for task Π in $M \implies$
$(i-1)$-round algorithm for $\text{cl}_M(\Pi)$ in M



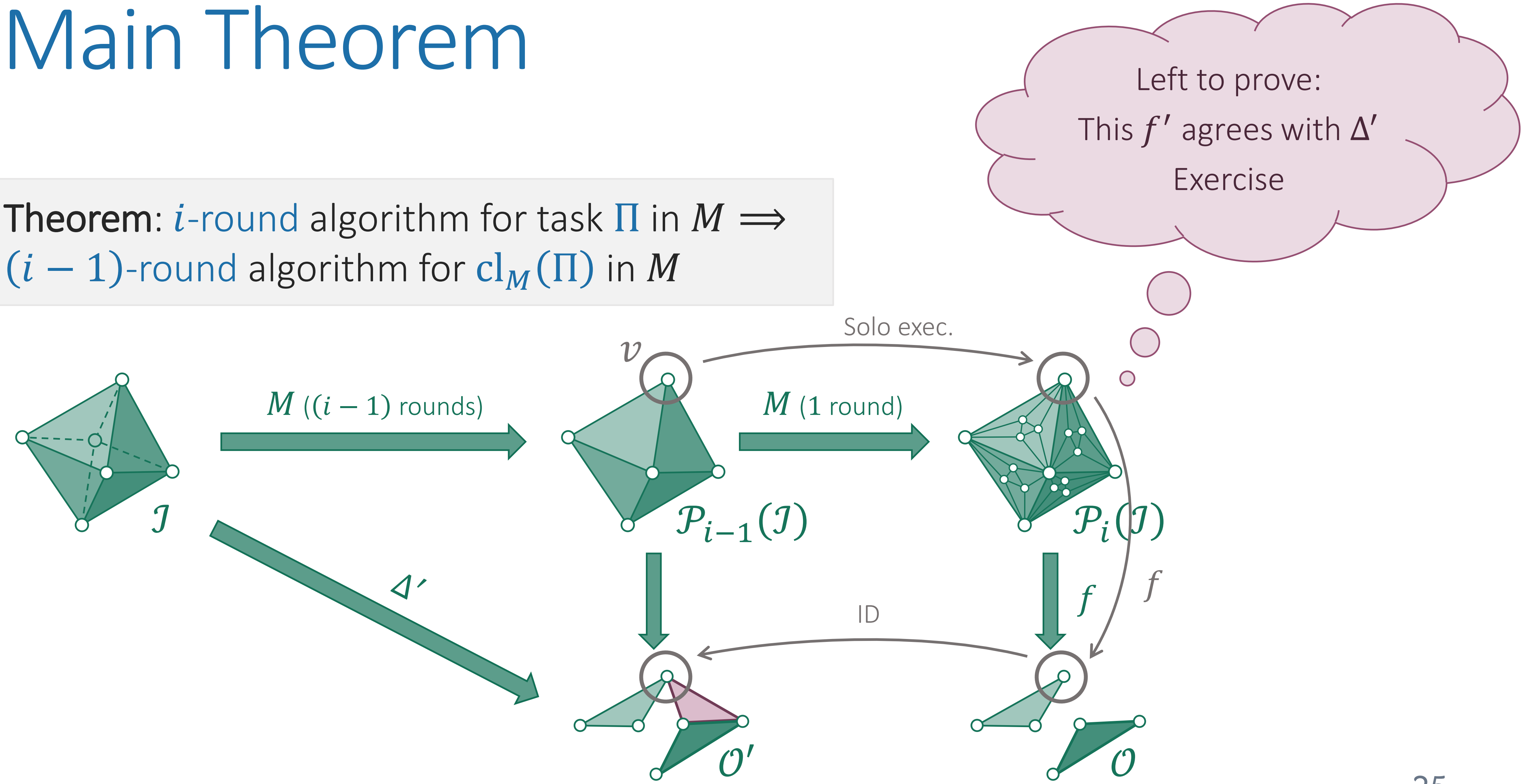
Main Theorem

Theorem: i-round algorithm for task Π in $M \Longrightarrow$
$(i-1)$-round algorithm for $\mathrm{cl}_M(\Pi)$ in M



Main Theorem

Theorem: i-round algorithm for task Π in $M \implies$ $(i-1)$-round algorithm for $\mathrm{cl}_M(\Pi)$ in M



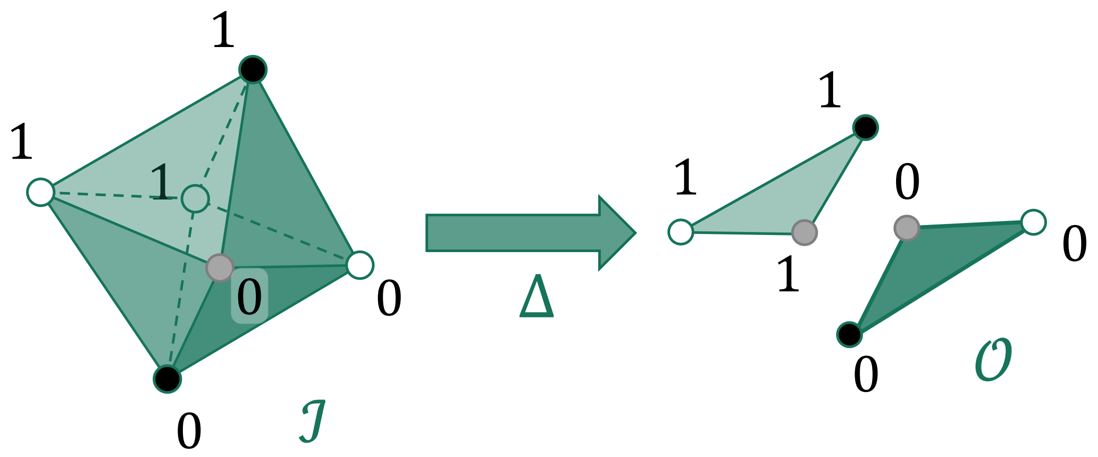
Main Theorem

Theorem: i-round algorithm for task Π in $M \Longrightarrow$ $(i-1)$-round algorithm for $\mathrm{cl}_M(\Pi)$ in M

Applications

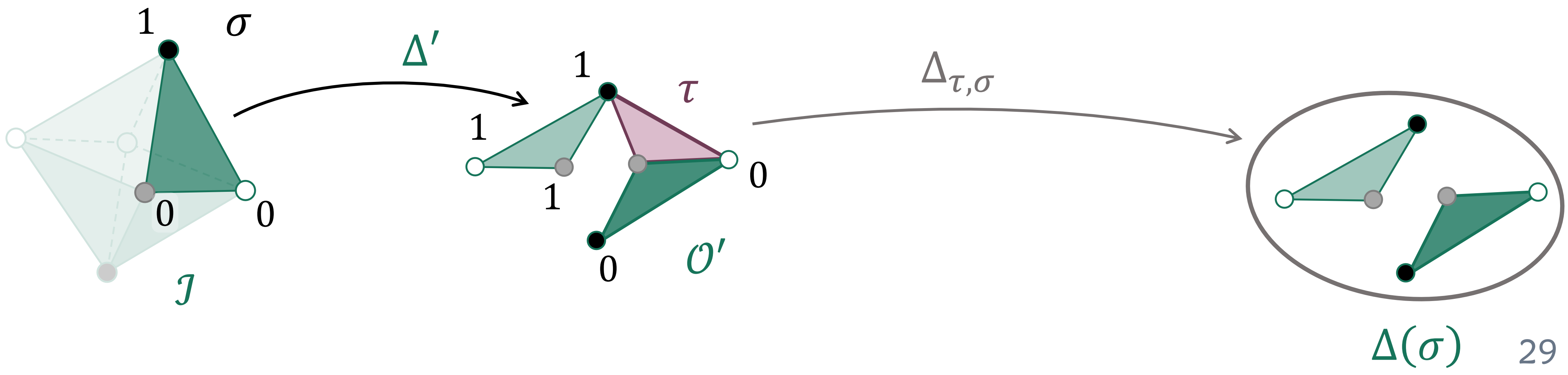
Impossibility of Consensus

- M = wait free iterated immediate snapshot
- $\Pi = (\mathcal{I}, \mathcal{O}, \Delta)$ = Consensus
- $\mathrm{cl}_M(\Pi)$ = ?



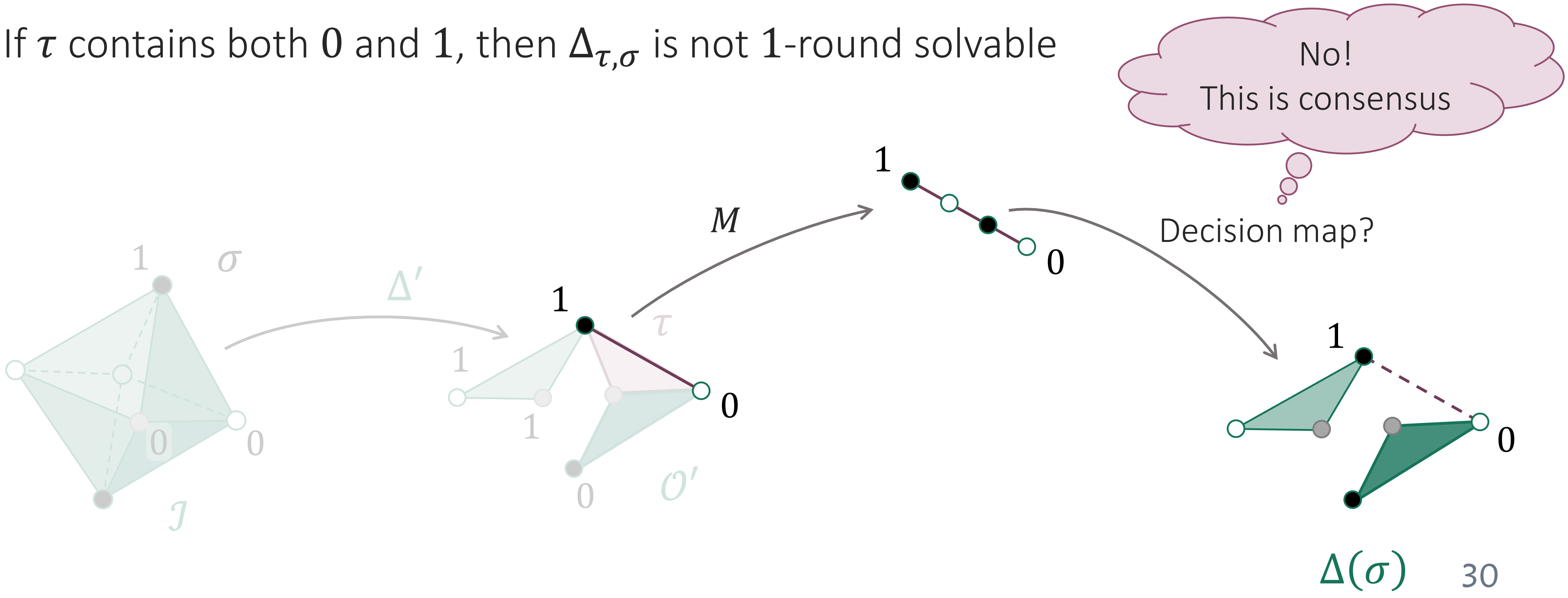
Impossibility of Consensus

- $\tau \in \Delta'(\sigma) \Leftrightarrow \Pi_{\tau,\sigma}$ is 1-round solvable in M
- If τ contains both 0 and 1, then $\Delta_{\tau,\sigma}$ is not 1-round solvable



Impossibility of Consensus

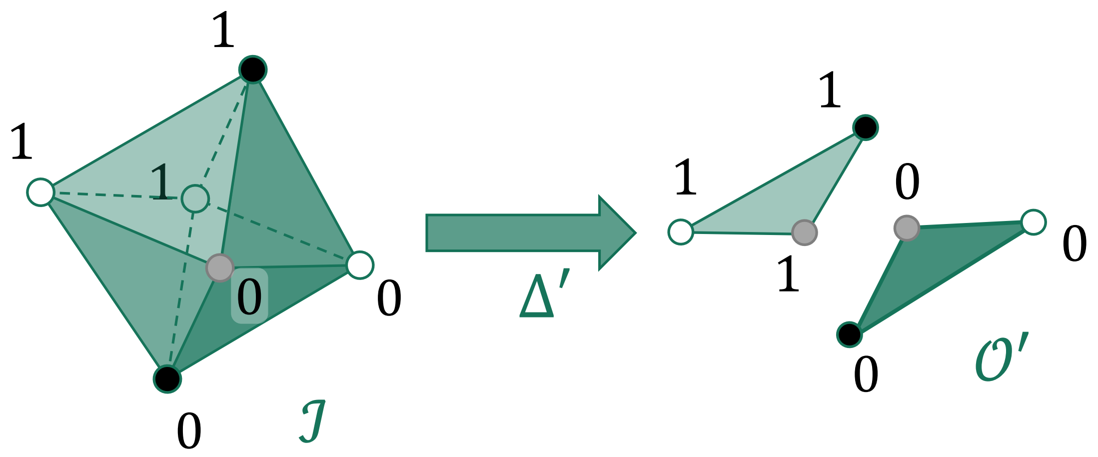
- $\tau \in \Delta'(\sigma) \Leftrightarrow \Pi_{\tau,\sigma}$ is 1-round solvable in M
- If τ contains both 0 and 1, then $\Delta_{\tau,\sigma}$ is not 1-round solvable



Impossibility of Consensus

- $\mathrm{cl}_M(\Pi) = (\mathcal{I}, \mathcal{O}', \Delta')$ has the same simplices as $\Pi = (\mathcal{I}, \mathcal{O}, \Delta)$
- So for consensus $\mathrm{cl}_M(\Pi) = \Pi$

Claim 1: i-round algorithm for consensus $\Longrightarrow$ $(i-1)$-round algorithm for consensus



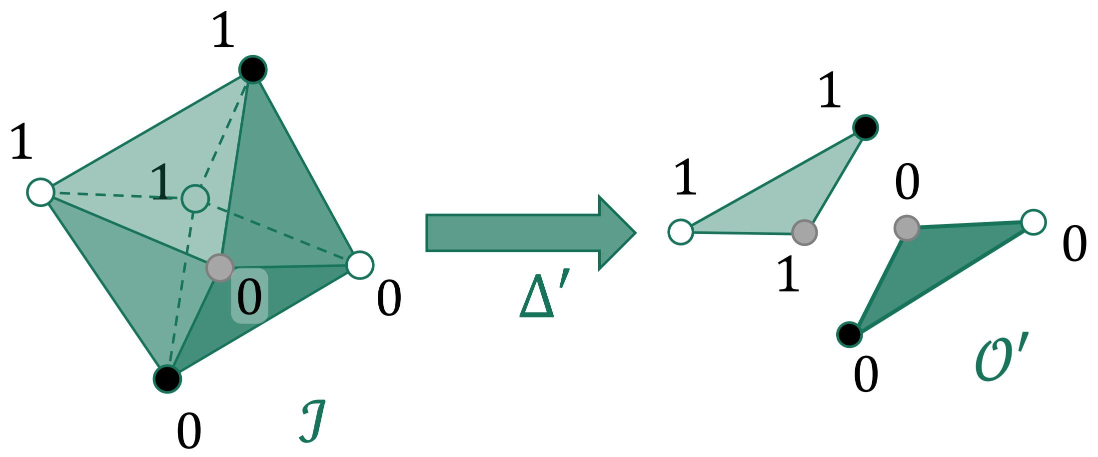
Impossibility of Consensus

Claim 1: i-round algorithm for consensus $\Longrightarrow$ $(i-1)$-round algorithm for consensus

Claim 2: No 0-round algorithm for consensus

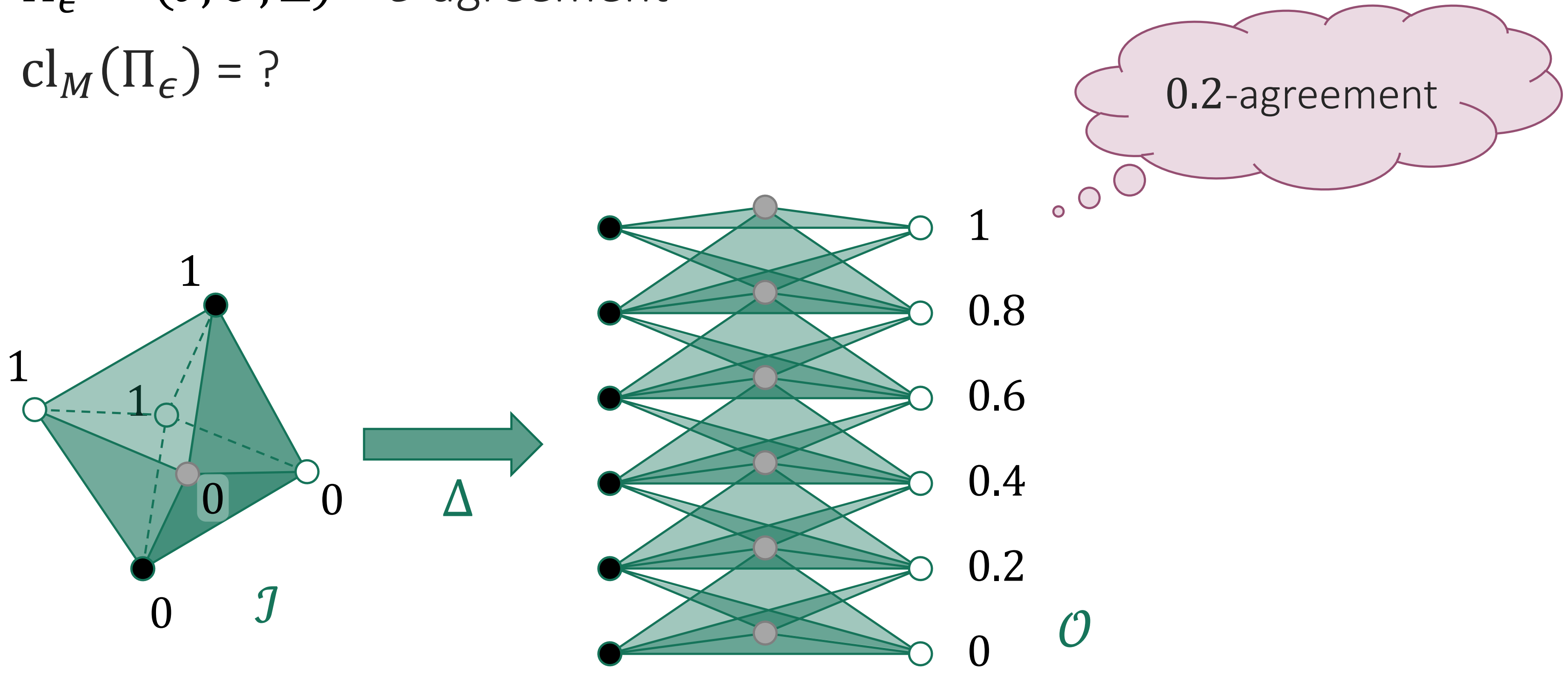
Conclusion: Impossibility of consensus in iterated immediate snapshot model

No matter in how many rounds



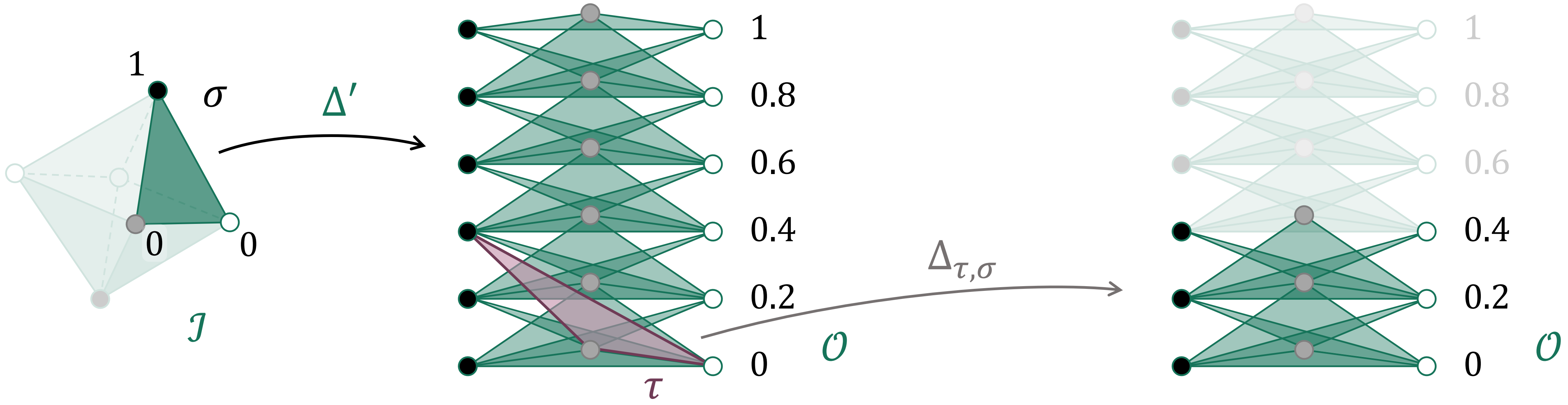
Approximate Agreement

- $M =$ wait free iterated immediate snapshot
- $\Pi_\epsilon = (\mathcal{I}, \mathcal{O}, \Delta)$ = ϵ-agreement
- $\text{cl}_M(\Pi_\epsilon)$ = ?



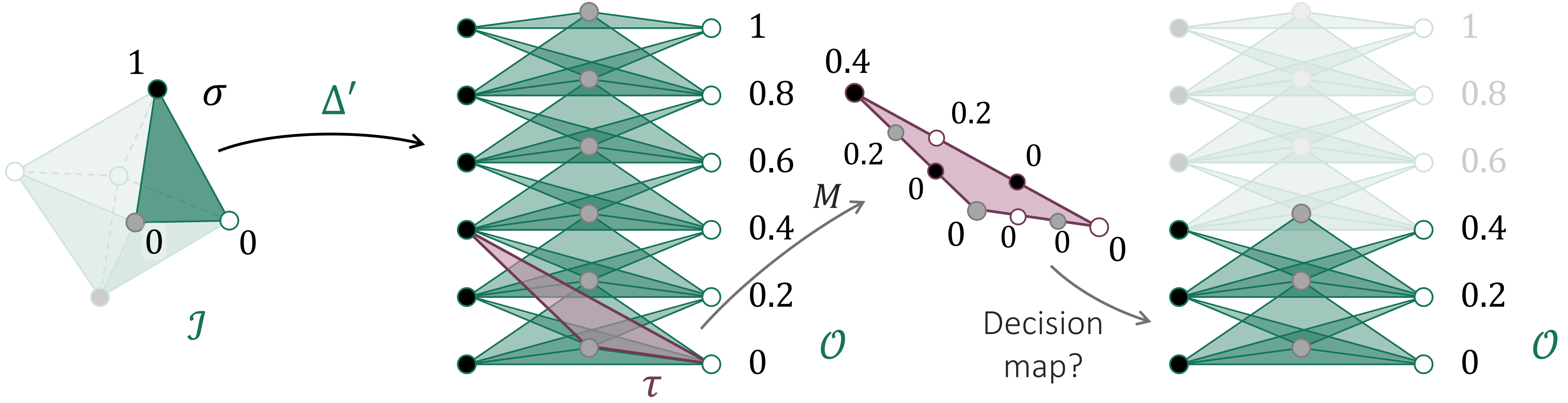
Approximate Agreement

- $\tau \in \Delta'(\sigma) \Leftrightarrow \Pi_{\tau,\sigma}$ is 1-round solvable in M



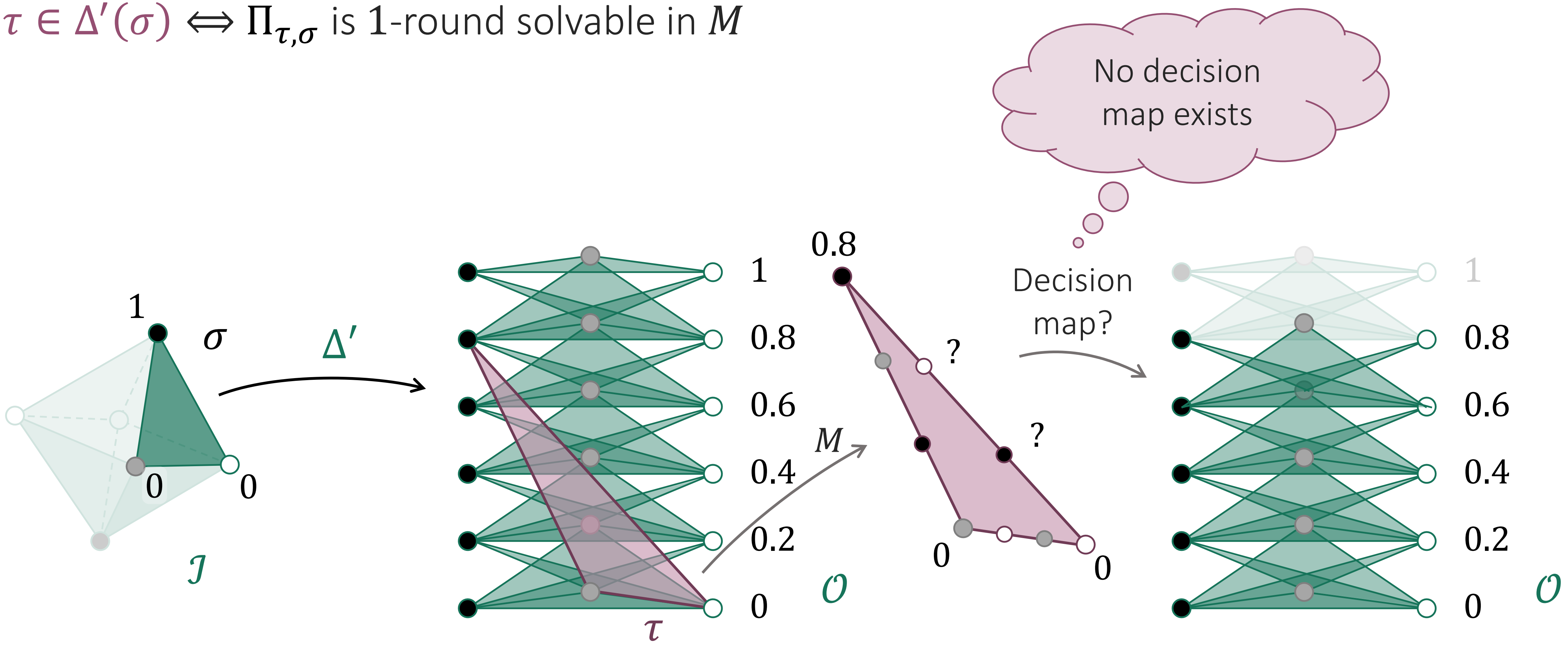
Approximate Agreement

- $\tau \in \Delta'(\sigma) \Leftrightarrow \Pi_{\tau,\sigma}$ is 1-round solvable in M



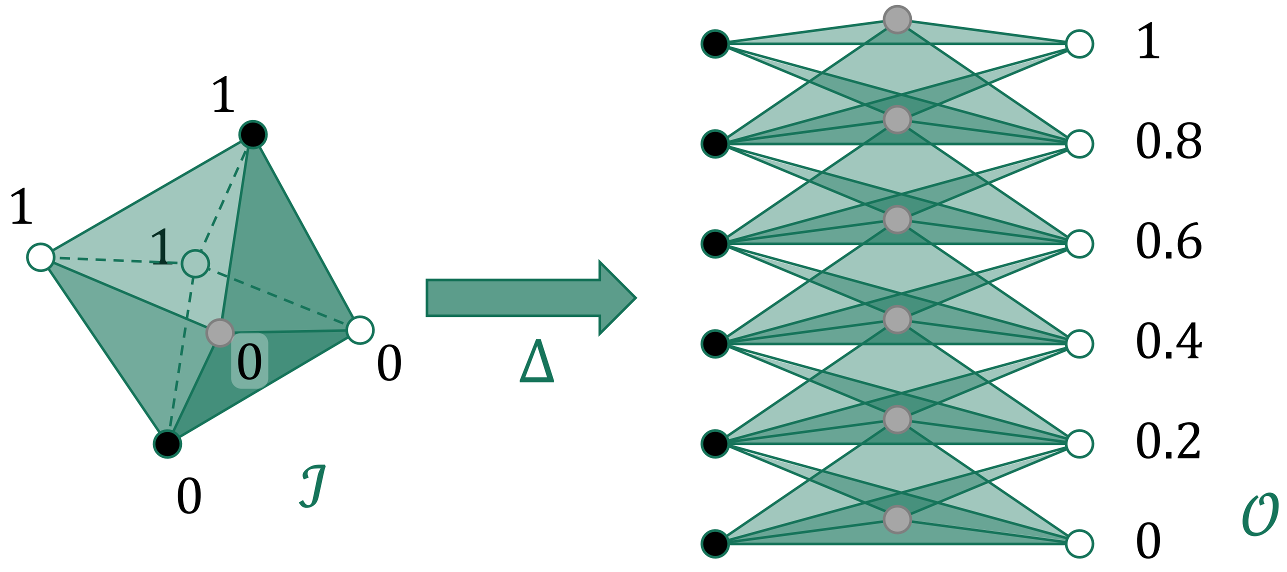
Approximate Agreement

- $\tau \in \Delta'(\sigma) \Leftrightarrow \Pi_{\tau,\sigma}$ is 1-round solvable in M



Approximate Agreement

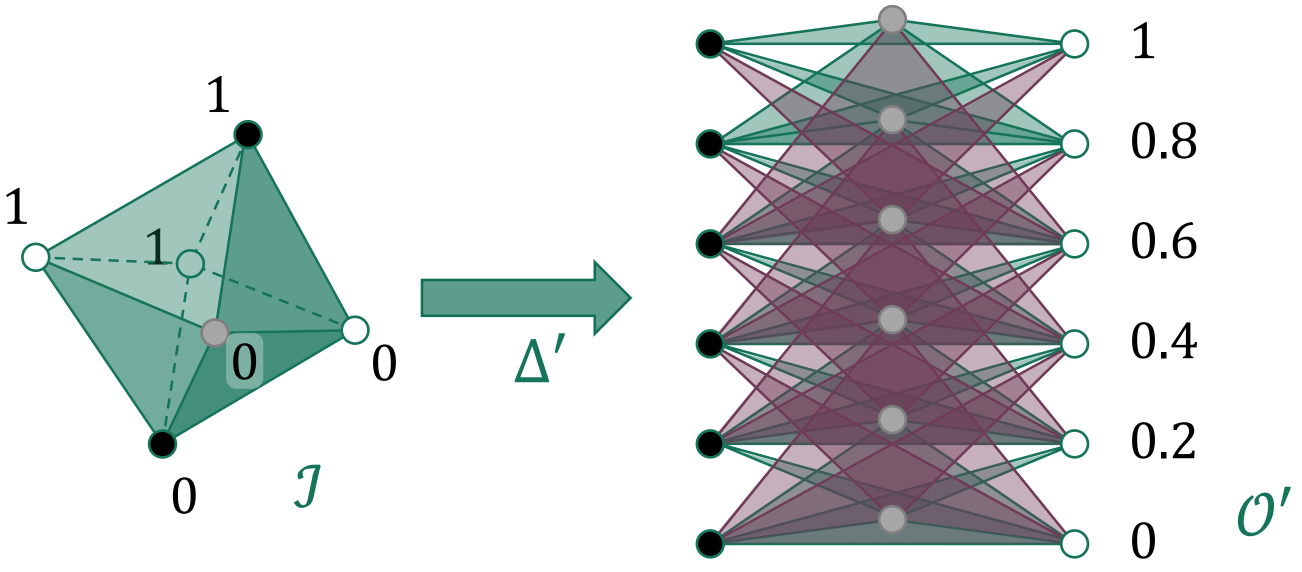
- $\mathrm{cl}_M(\Pi_\epsilon)$ = ?
  - Simplices of δ-agreement are in Δ', but only for small δ
  - We show: $\mathrm{cl}_M(\Pi_\epsilon)$ = 2ϵ-agreement



Approximate Agreement

- $\mathrm{cl}_M(\Pi_\epsilon) = \mathrm{cl}_M(\Pi_{2\epsilon})$

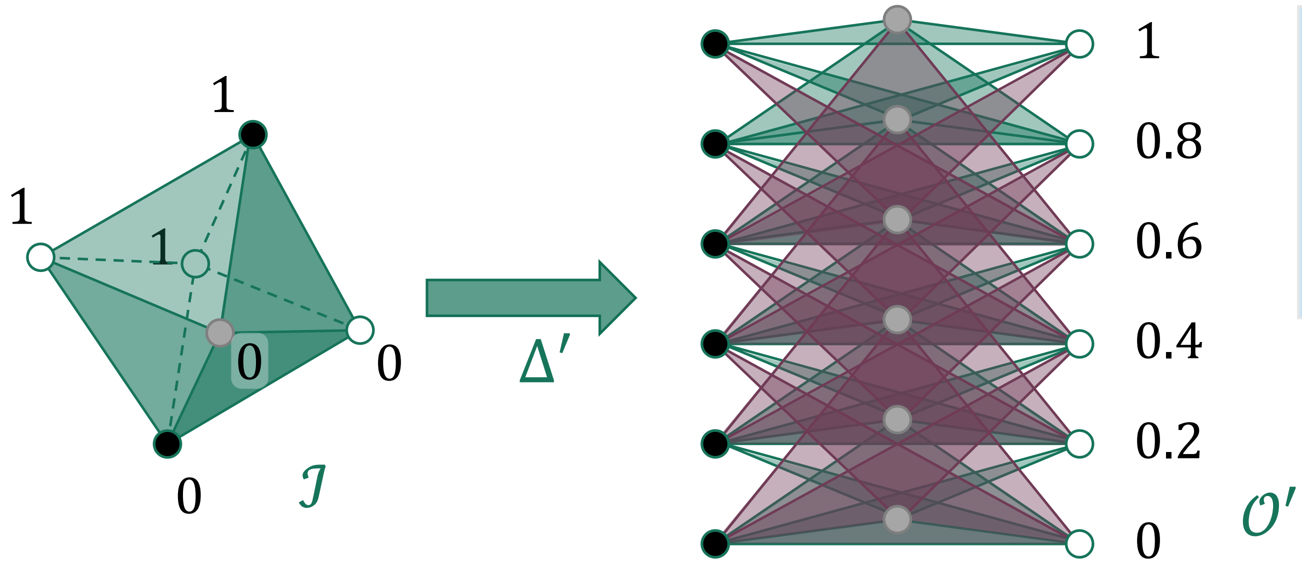
Claim 1: i-round algorithm ϵ-agreement $\Longrightarrow$ $(i-1)$-round algorithm for 2ϵ-agreement



Approximate Agreement

Claim 1: i-round algorithm ϵ-agreement $\Longrightarrow$ $(i-1)$-round algorithm for 2ϵ-agreement

Claim 2: No 0-round alg. for δ-agreement, $\delta < 1$



If ϵ-agreement solvable in t rounds,

2ϵ-agreement solvable in $t-1$ rounds

…

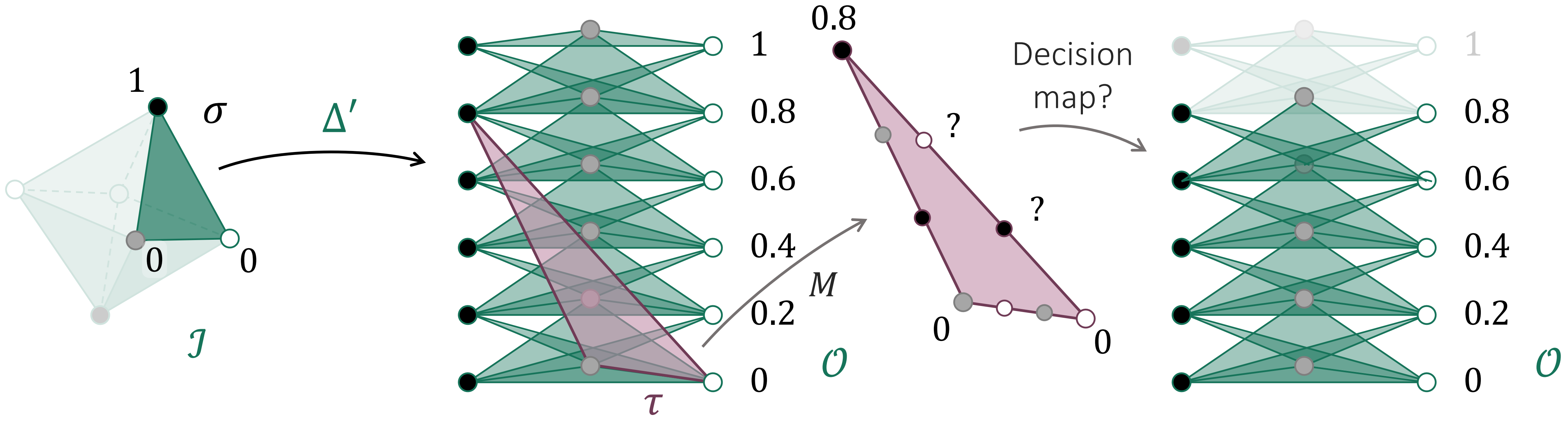
$2^t\epsilon$-agreement solvable in 0 rounds

Conclusion:
ϵ-agreement requires $\log_2 1/\epsilon$ rounds

With $n \geq 3$ processes

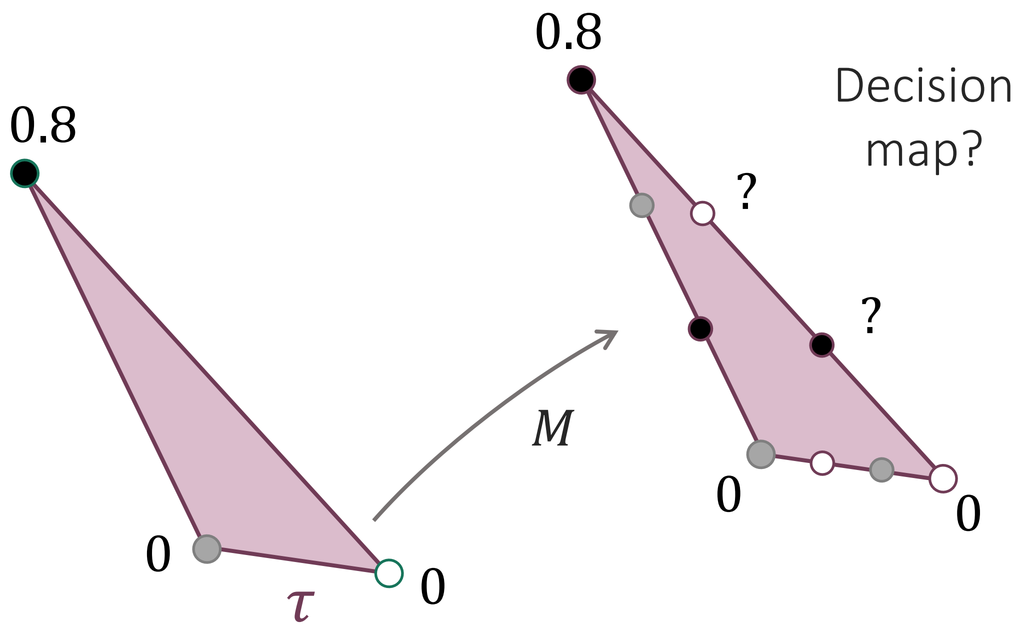
The Power of Simplicity

- Core of the proofs: What happens in a single round?



The Power of Simplicity

- Core of the proofs: What happens in a single round?
- Makes the proofs easy to extend



One Technique to Prove them All

Impossibility of consensus

Model
IIS
IIS + T&S

Consensus time lower bounds

Model	Time for $n = 2$	Time for $n \geq 3$
IIS	$\log_3 \epsilon$	$\log_2 \epsilon$
IIS + T&S	$\log_3 \epsilon$	$\log_2 \epsilon$
IIS + Bin. Consensus		$\min(\log_2 1/\epsilon, \log n - 1)$

With limitations