Stone duality proofs for distributed computability theorems

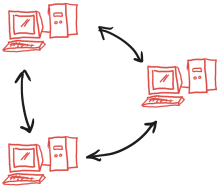
TACL '26

Cameron Calk
Emmanuel Godard

Aix-Marseille Université
Laboratoire d'Informatique et des Systèmes (LIS)

Overview

In distributed computing, several processes **communicate** in order to solve a **task** (consensus, renaming, . . .).

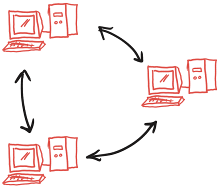


Overview

In distributed computing, several processes **communicate** in order to solve a **task** (consensus, renaming, . . .).

*Which tasks can be **solved** in a given **model** of communication?*

Lack of global information is the main obstruction to task-solvability.



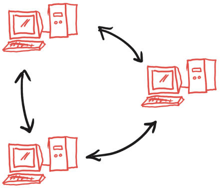
Overview

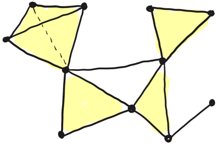
In distributed computing, several processes **communicate** in order to solve a **task** (consensus, renaming, ...).

Which tasks can be ***solved*** *in a given* ***model*** *of communication?*

Lack of global information is the main obstruction to task-solvability.

This question has been approached using **simplicial complexes**.





Overview

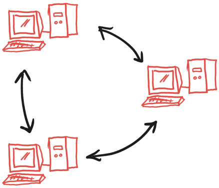
In distributed computing, several processes **communicate** in order to solve a **task** (consensus, renaming, ...).

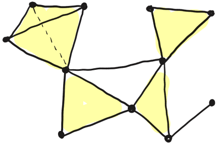
*Which tasks can be **solved** in a given **model** of communication?*

Lack of global information is the main obstruction to task-solvability.

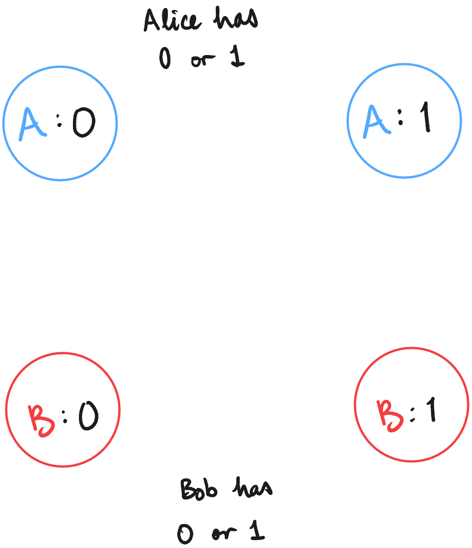
This question has been approached using **simplicial complexes**.

We extend this line of work using **spectral spaces**.

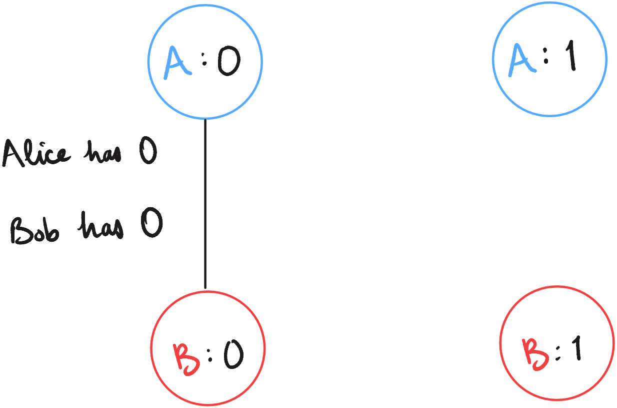




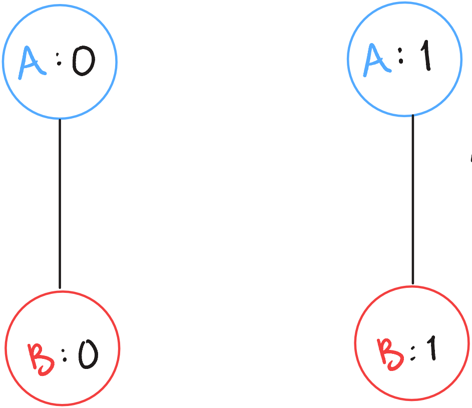
A toy example



A toy example



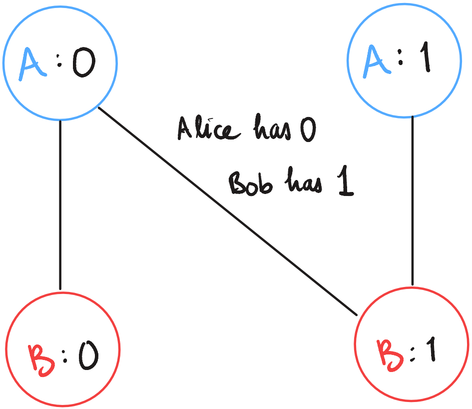
A toy example



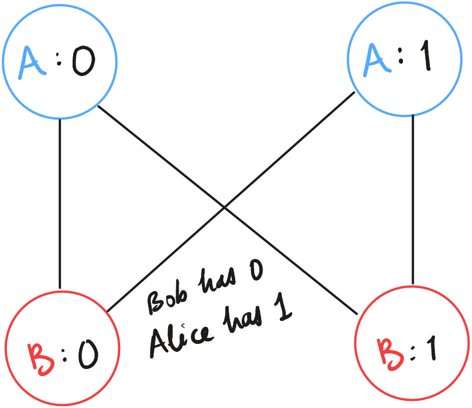
Alice has 1

Bob has 1

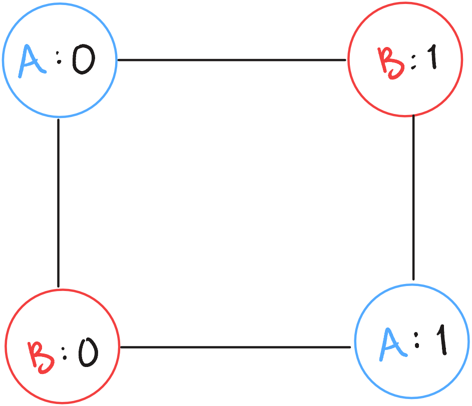
A toy example



A toy example

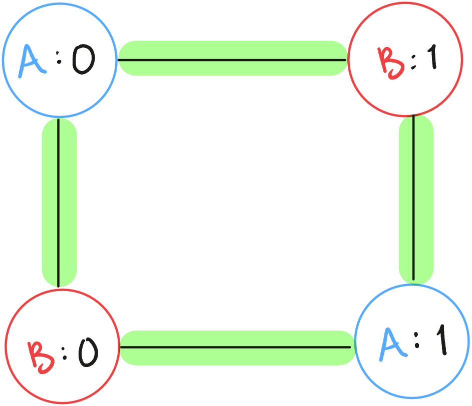


A toy example



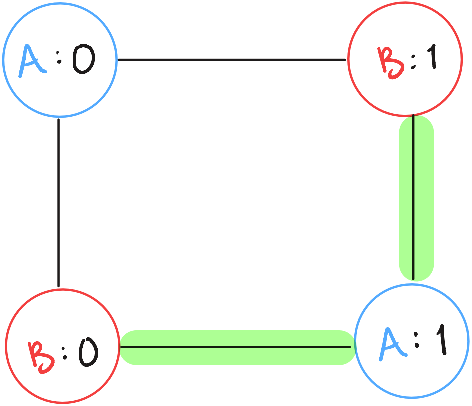
A toy example

Encodes all possible global states



A toy example

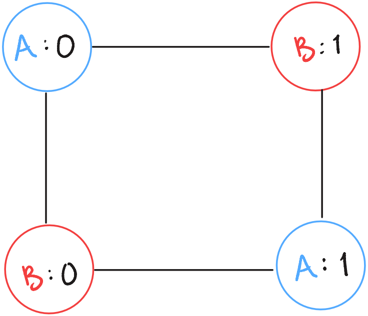
Encodes all possible global states



In this local state, Alice doesn't know which value Bob has

A toy example

With names and values
(coloured)

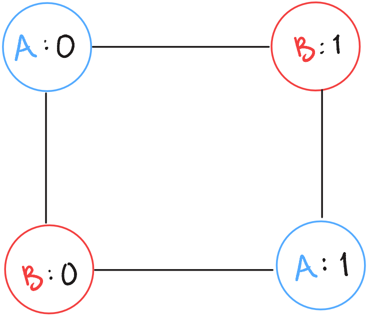


With only values
(colourless)

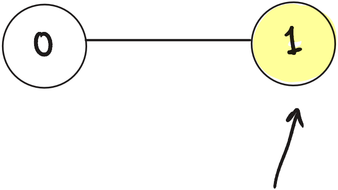
all processes
have 0

A toy example

With names and values
(coloured)



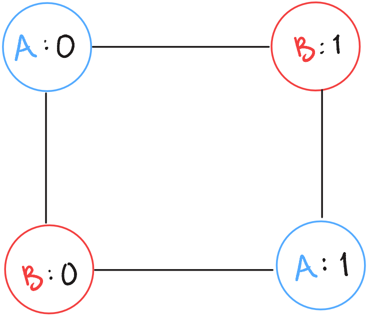
With only values
(colourless)



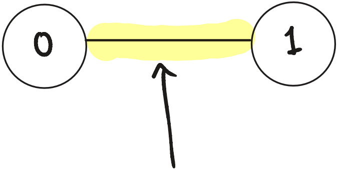
all processes
have 1

A toy example

With names and values (coloured)



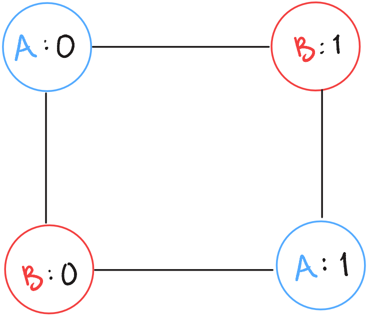
With only values (colourless)



There is at least one process with 0, and at least one with 1.

A toy example

Now, Alice and Bob communicate ...



There are three possibilities

- $A \longrightarrow B$

  Bob sees Alice's value, but she doesn't see his.

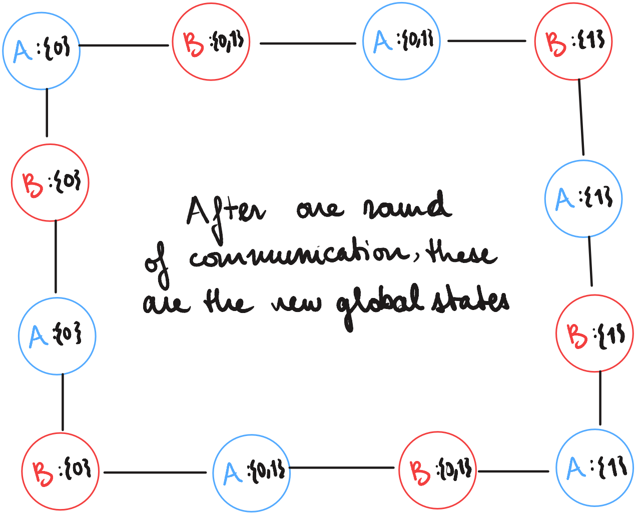
- $A \longleftrightarrow B$

  Alice and Bob see each other's values.

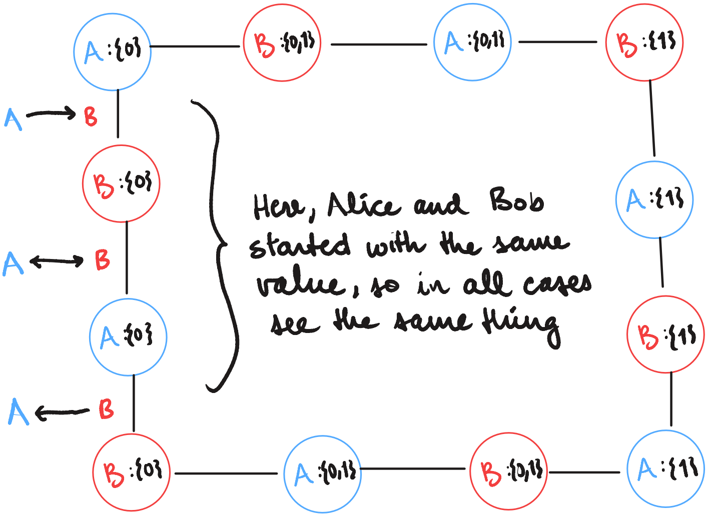
- $A \longleftarrow B$

  Alice sees Bob's value, but he doesn't see hers.

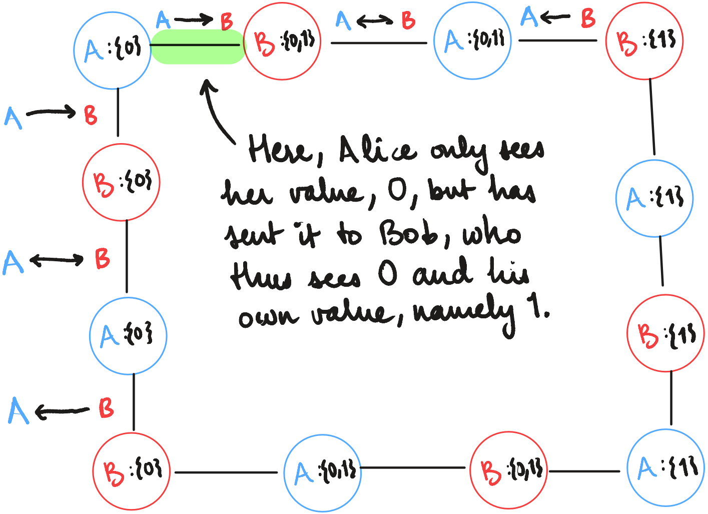
A toy example



A toy example

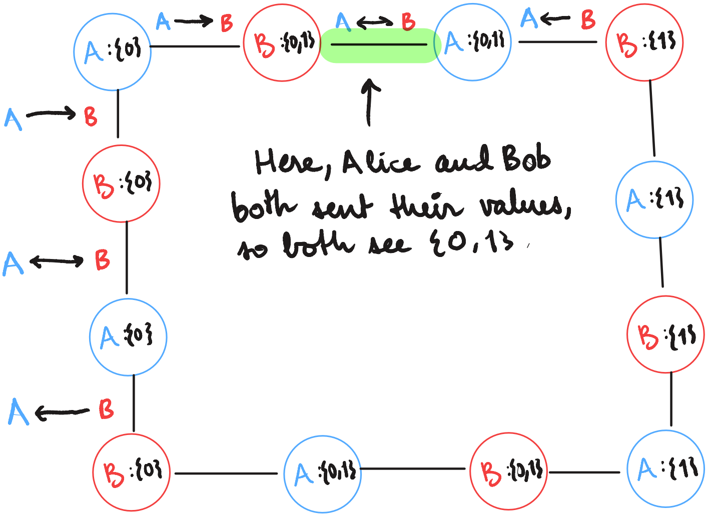


A toy example

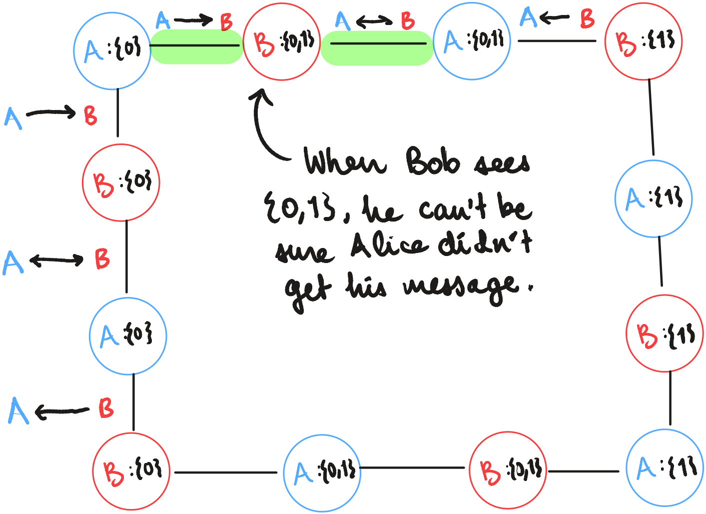


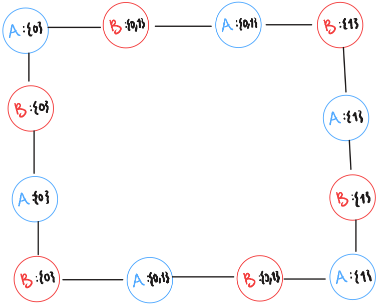
Here, Alice only sees her value, 0, but has sent it to Bob, who thus sees 0 and his own value, namely 1.

A toy example

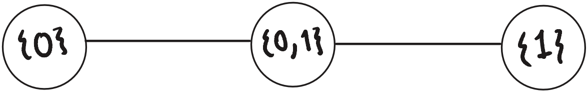


A toy example





Forgetting about names, we obtain this



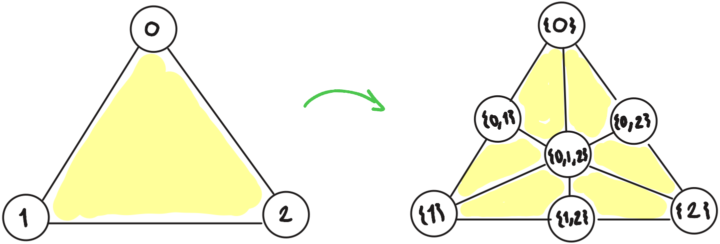
A toy example

So, communication transforms the initial configuration:

A toy example

So, communication transforms the initial configuration:

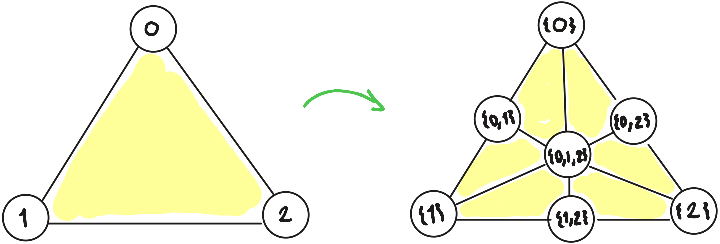
This generalises to any number of values:



A toy example

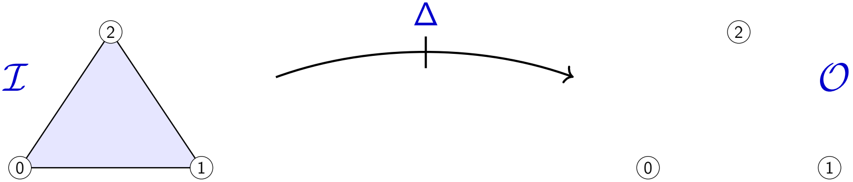
So, communication transforms the initial configuration:

This generalises to any number of values:



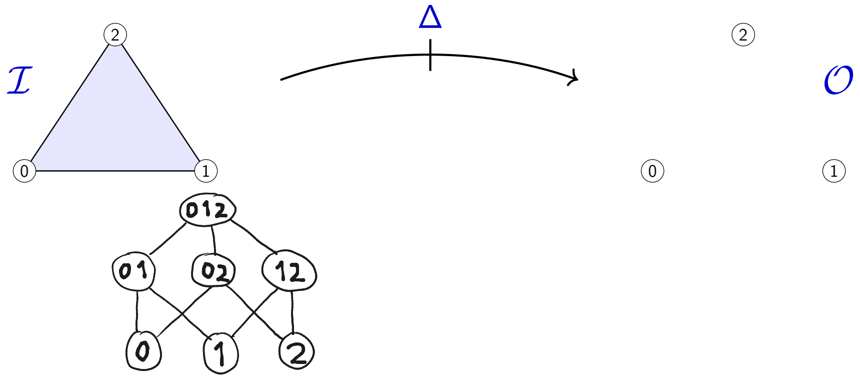
We have encoded distributed knowledge in simplicial complexes, and a communication protocol as a transformation thereof.

Simplicial semantics



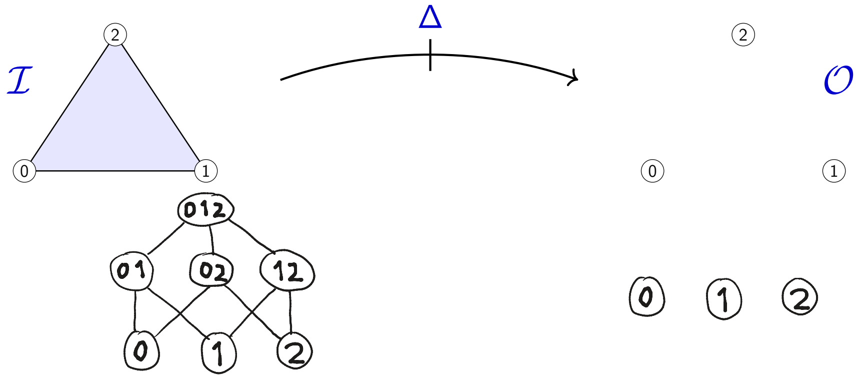
A distributed **task** (specification) is a relation between an input complex and an output complex.

Simplicial semantics of task-solvability



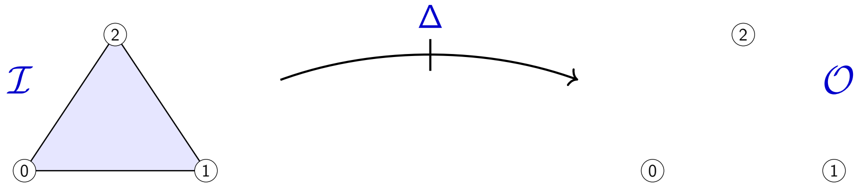
A distributed **task** (specification) is a relation between an input complex and an output complex.

Simplicial semantics of task-solvability



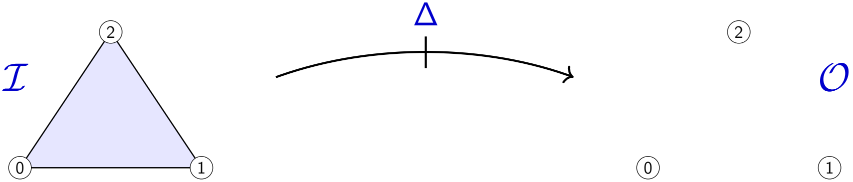
A distributed **task** (specification) is a relation between an input complex and an output complex.

Simplicial semantics of task-solvability



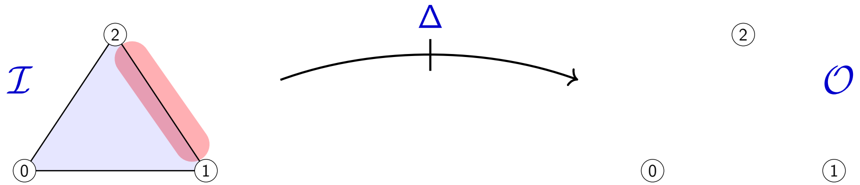
A distributed **task** (specification) is a relation between an input complex and an output complex.

Simplicial semantics of task-solvability



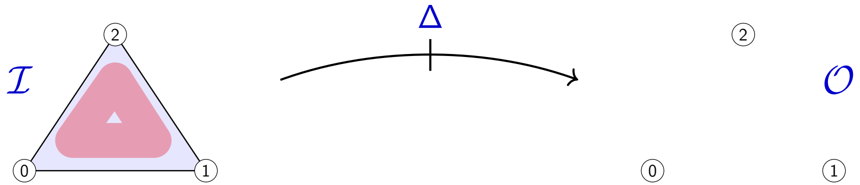
A distributed **task** (specification) is a relation between an input complex and an output complex.

Simplicial semantics of task-solvability



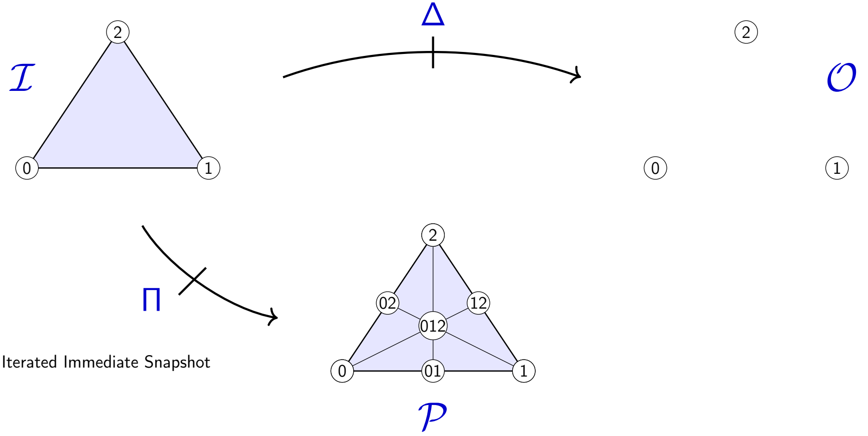
A distributed **task** (specification) is a relation between an input complex and an output complex.

Simplicial semantics of task-solvability



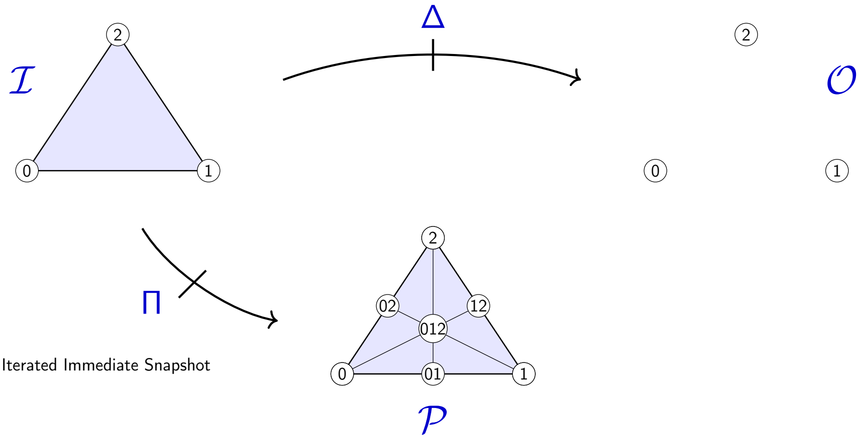
A distributed **task** (specification) is a relation between an input complex and an output complex.

Simplicial semantics of task-solvability



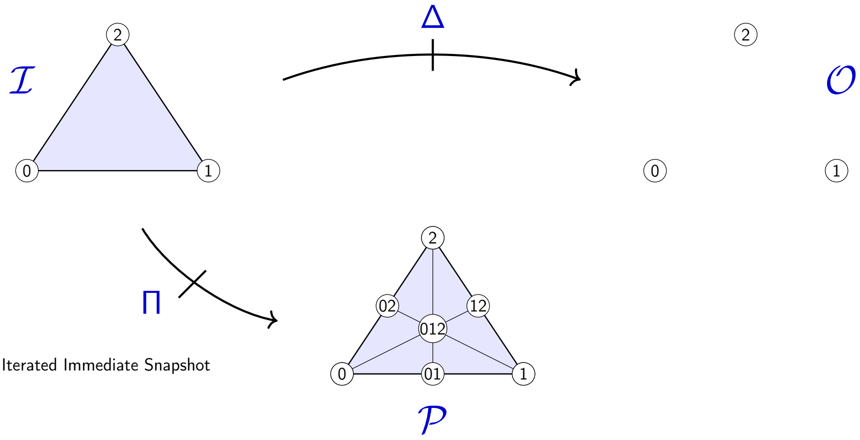
A communication **model** (protocol) relates inputs to reachable states.

Simplicial semantics of task-solvability



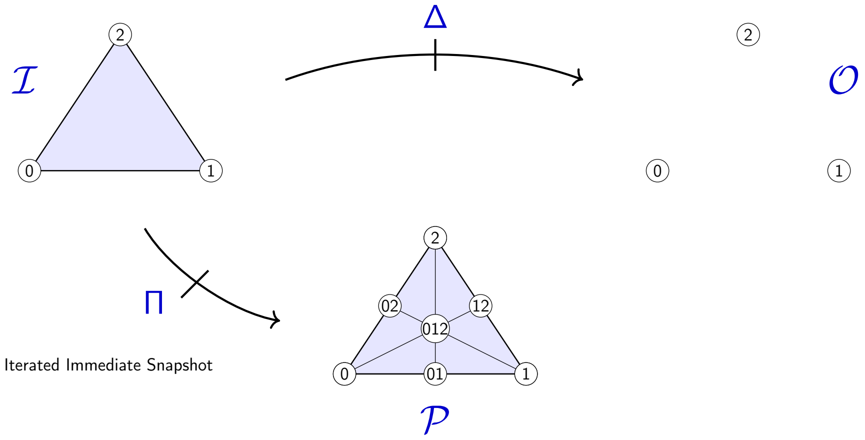
A communication **model** (protocol) relates inputs to reachable states.

Simplicial semantics of task-solvability



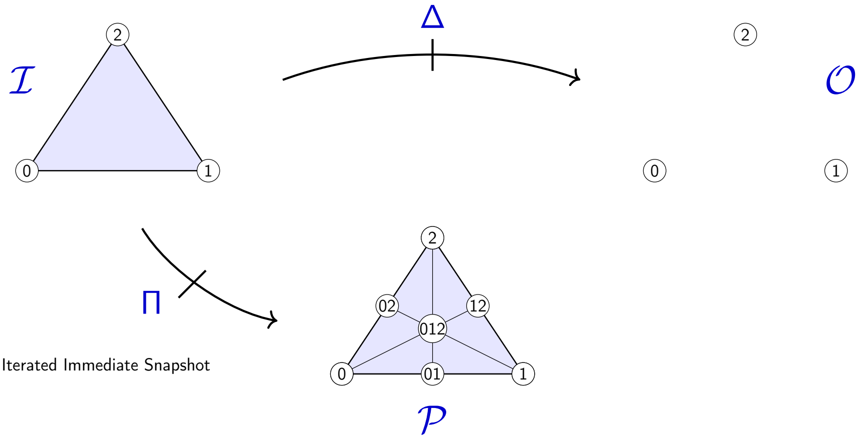
A communication **model** (protocol) relates inputs to reachable states.

Simplicial semantics of task-solvability



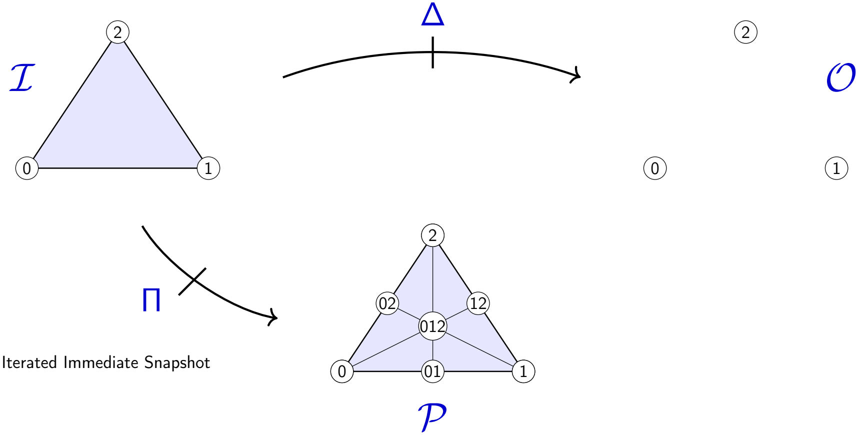
A communication **model** (protocol) relates inputs to reachable states.

Simplicial semantics of task-solvability



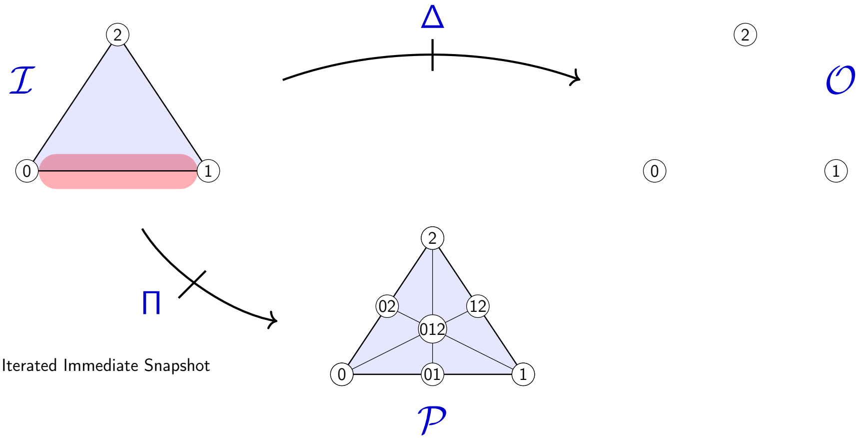
A communication **model** (protocol) relates inputs to reachable states.

Simplicial semantics of task-solvability



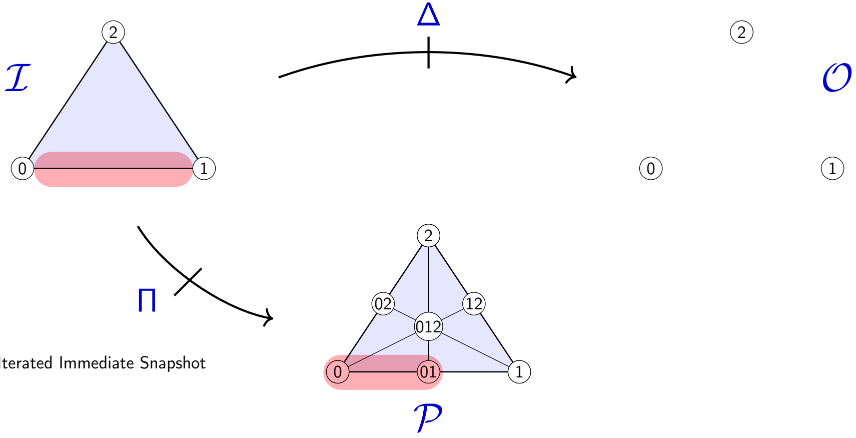
A communication **model** (protocol) relates inputs to reachable states.

Simplicial semantics of task-solvability



A communication **model** (protocol) relates inputs to reachable states.

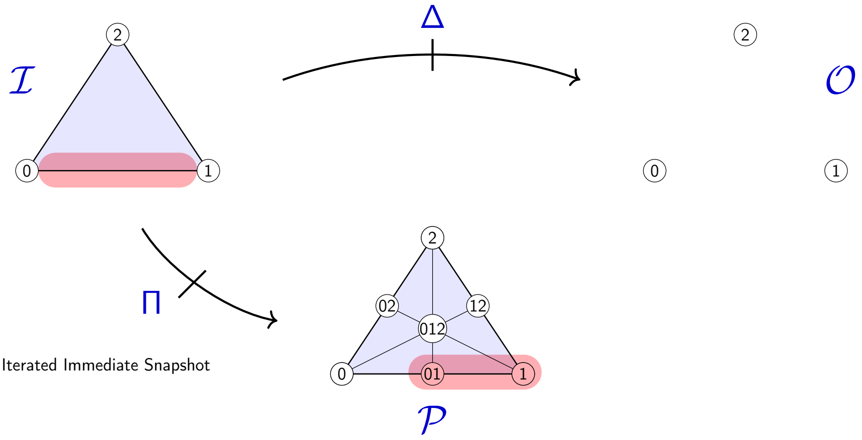
Simplicial semantics of task-solvability



Iterated Immediate Snapshot

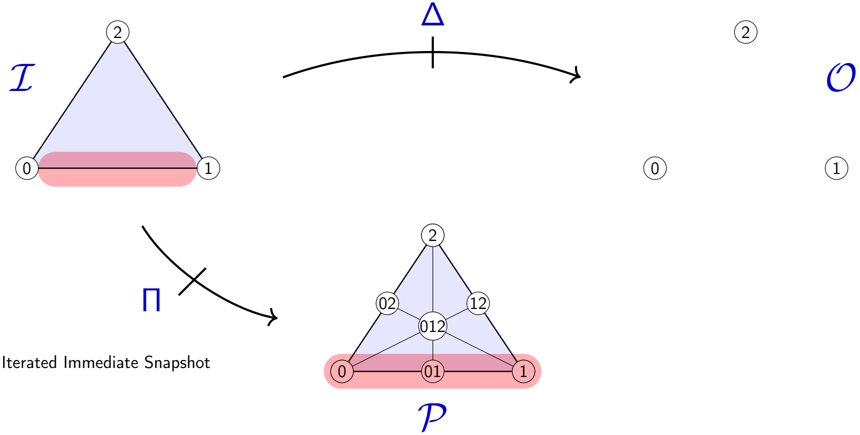
A communication **model** (protocol) relates inputs to reachable states.

Simplicial semantics of task-solvability



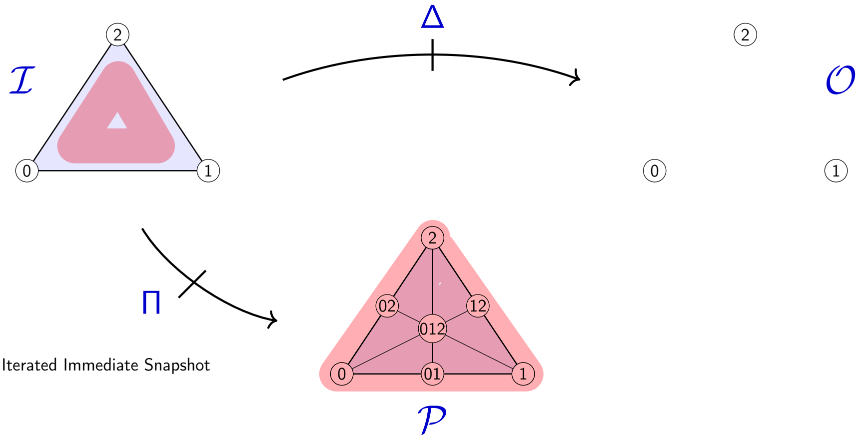
A communication **model** (protocol) relates inputs to reachable states.

Simplicial semantics of task-solvability



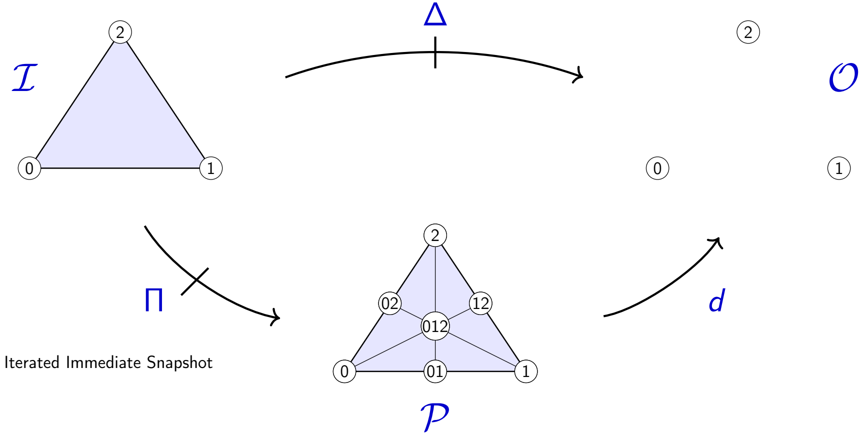
A communication **model** (protocol) relates inputs to reachable states.

Simplicial semantics of task-solvability



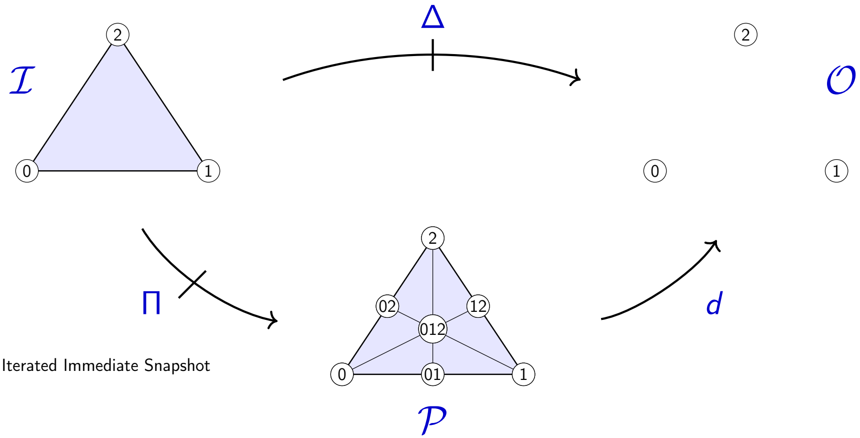
A communication **model** (protocol) relates inputs to reachable states.

Simplicial semantics of task-solvability



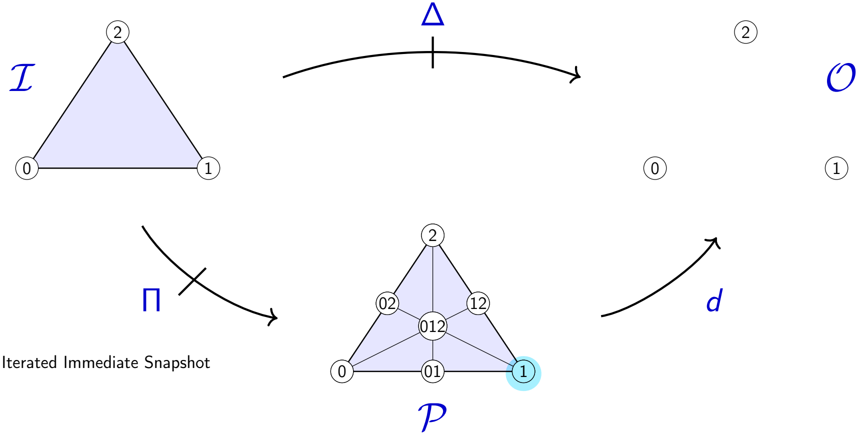
A **decision algorithm** is a simplicial map sending local states to valid outputs.

Simplicial semantics of task-solvability



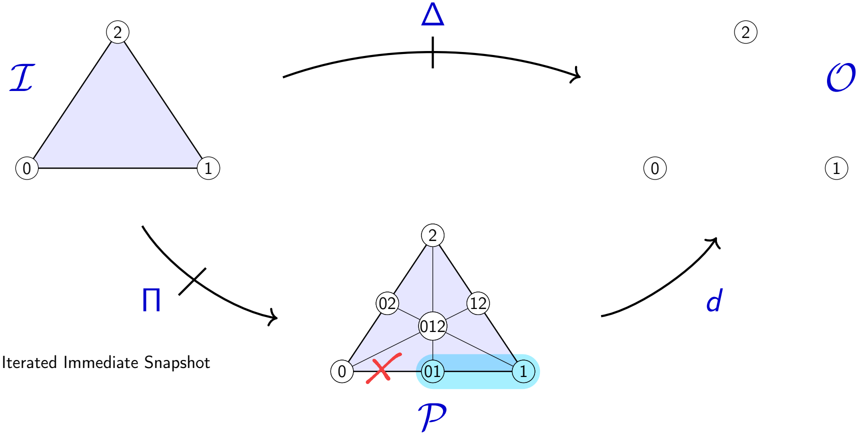
A **decision algorithm** is a simplicial map sending local states to valid outputs.

Simplicial semantics of task-solvability



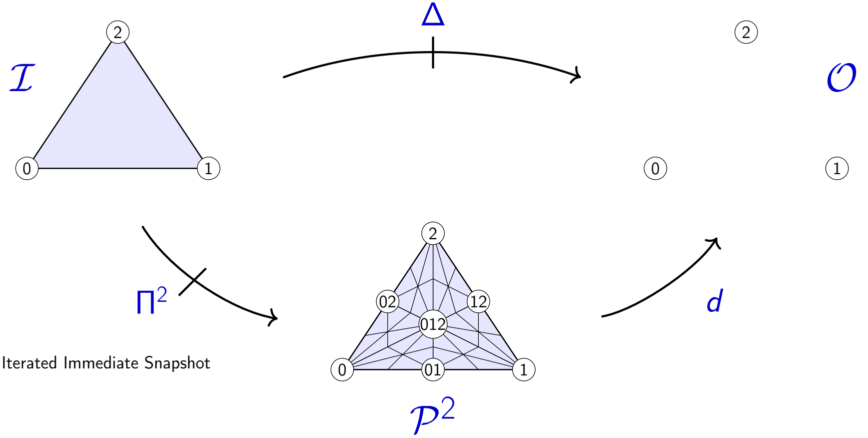
A **decision algorithm** is a simplicial map sending local states to valid outputs.

Simplicial semantics of task-solvability



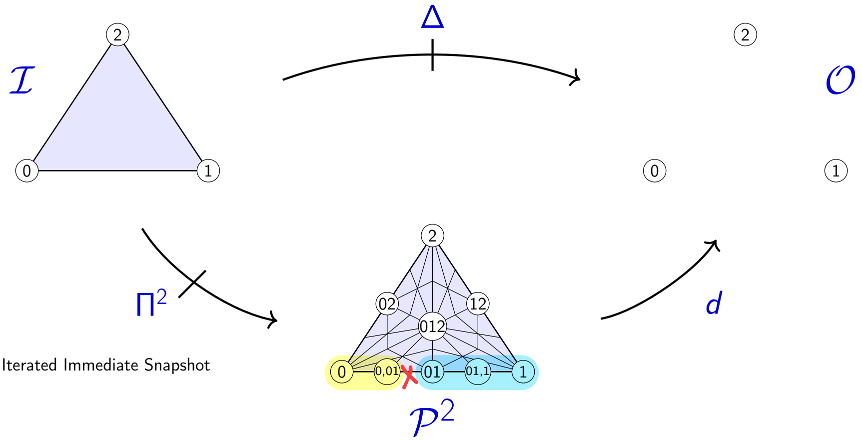
A **decision algorithm** is a simplicial map sending local states to valid outputs.

Simplicial semantics of task-solvability



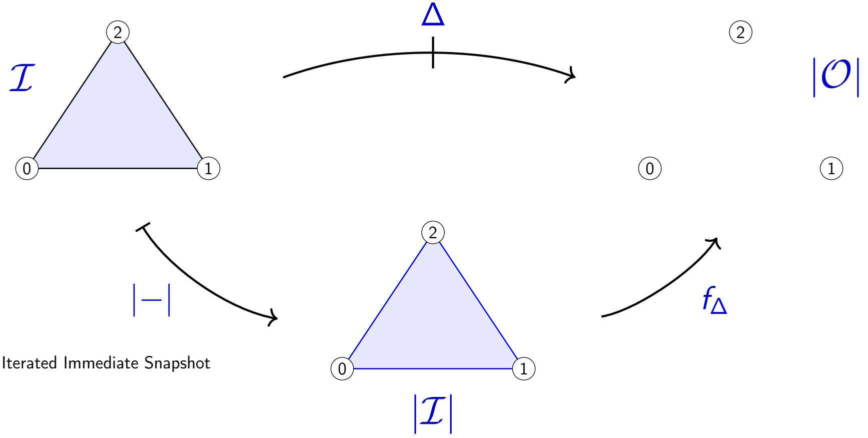
A **decision algorithm** is a simplicial map sending local states to valid outputs.

Simplicial semantics of task-solvability



A **decision algorithm** is a simplicial map sending local states to valid outputs.

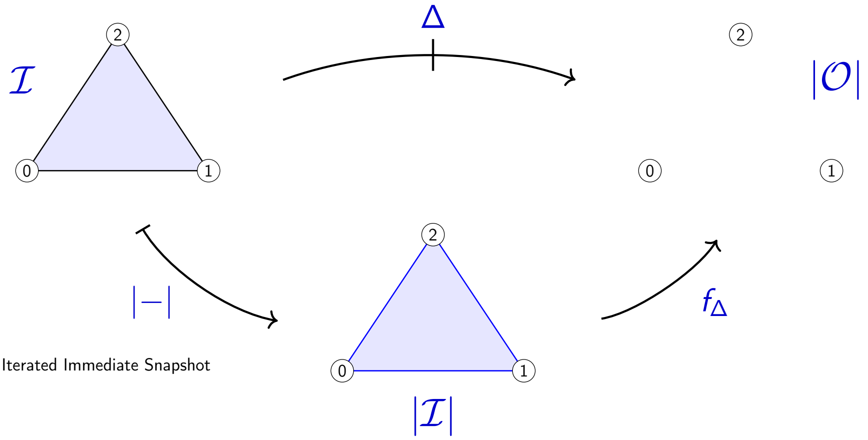
Simplicial semantics of task-solvability



Theorem ([Herlihy, Shavit])

A colorless task $(\mathcal{I}, \mathcal{O}, \Delta)$ is solved by the IIS protocol $\iff$ *There exists a continuous map $f_\Delta : |\mathcal{I}| \to |\mathcal{O}|$.*

Simplicial semantics of task-solvability



Theorem ([Herlihy, Shavit], [Saks, Zarahoglou], Gödel prize '04)

A colorless task $(\mathcal{I}, \mathcal{O}, \Delta)$ *is solved by a wait-free protocol* $\iff$ *There exists a continuous map* $f_\Delta : |\mathcal{I}| \to |\mathcal{O}|$.

Duality intervenes

Duality

The relations Δ and Π can be seen as maps from **finite posets** to **down-sets of finite posets**.

$$\Gamma \subseteq \mathcal{C} \times \mathcal{C}' \qquad \leftrightarrow \qquad \gamma : \mathcal{C} \longrightarrow \mathcal{D}(\mathcal{C}')$$

Duality

The relations Δ and Π can be seen as maps from **finite posets** to **down-sets of finite posets**.

$$\Gamma \subseteq \mathcal{C} \times \mathcal{C}' \qquad \leftrightarrow \qquad \gamma : \mathcal{C} \longrightarrow \mathcal{D}(\mathcal{C}')$$

Birkhoff duality expresses a **contravariant equivalence** between their respective categories:

$$\begin{tikzcd} P \arrow[d, "f"'] \\ Q \end{tikzcd} \qquad \begin{tikzcd} \mathsf{Pos}_f \arrow[r, bend left, "\mathcal{D}(-)"] & \mathsf{DL}_f \arrow[l, bend left, "\mathcal{J}(-)"] \end{tikzcd} \qquad \begin{tikzcd} \mathcal{D}(P) \\ \mathcal{D}(Q) \arrow[u, "f^{-1}"'] \end{tikzcd}$$

Duality

The relations Δ and Π can be seen as maps from **finite posets** to **down-sets of finite posets**.

$$\Gamma \subseteq \mathcal{C} \times \mathcal{C}' \qquad \leftrightarrow \qquad \gamma : \mathcal{C} \longrightarrow \mathcal{D}(\mathcal{C}')$$

Birkhoff duality expresses a **contravariant equivalence** between their respective categories:

$$\begin{tikzcd} P \arrow[d, "f"'] \\ Q \end{tikzcd} \qquad \begin{tikzcd} \mathsf{Pos}_f \arrow[r, bend left, "\mathcal{D}(-)"] & \mathsf{DL}_f \arrow[l, bend left, "\mathcal{J}(-)"] \end{tikzcd} \qquad \begin{tikzcd} \mathcal{D}(P) \\ \mathcal{D}(Q) \arrow[u, "f^{-1}"'] \end{tikzcd}$$

Stone duality generalises this to the infinite case:

Spectral spaces $\qquad \begin{tikzcd} \mathsf{Spec} \arrow[r, bend left, "\mathcal{KO}(-)"] & \mathsf{DL} \arrow[l, bend left, "St(-)"] \end{tikzcd} \qquad$ Distributive lattices

Duality

The relations Δ and Π can be seen as maps from **finite posets** to **down-sets of finite posets**.

$$\Gamma \subseteq \mathcal{C} \times \mathcal{C}' \qquad \leftrightarrow \qquad \gamma : \mathcal{C} \longrightarrow \mathcal{D}(\mathcal{C}')$$

Birkhoff duality expresses a **contravariant equivalence** between their respective categories:

$$\begin{tikzcd} P \arrow[d, "f"'] \\ Q \end{tikzcd} \qquad \begin{tikzcd} \mathsf{Spec}_f \arrow[r, bend left, "\mathcal{D}(-)"] & \mathsf{DL}_f \arrow[l, bend left, "\mathcal{J}(-)"] \end{tikzcd} \qquad \begin{tikzcd} \mathcal{D}(P) \\ \mathcal{D}(Q) \arrow[u, "f^{-1}"'] \end{tikzcd}$$

Stone duality generalises this to the infinite case:

Spectral spaces $\qquad \begin{tikzcd} \mathsf{Spec} \arrow[r, bend left, "\mathcal{KO}(-)"] & \mathsf{DL} \arrow[l, bend left, "St(-)"] \end{tikzcd} \qquad$ Distributive lattices

A homomorphism appears!

In the topological approach to distributed computing, carrier maps associated to protocols must satisfy **extra conditions**.

Indeed, a **protocol map** $\pi : \mathcal{I} \to \mathcal{D}(\mathcal{P})$ is such that

$\pi(\sigma \cap \tau) = \pi(\sigma) \cap \pi(\tau)$.

$\bigcup_{\sigma \in \mathcal{I}} \pi(\sigma) = \mathcal{P}$.

A homomorphism appears!

In the topological approach to distributed computing, carrier maps associated to protocols must satisfy **extra conditions**.

Indeed, a **protocol map** $\pi : \mathcal{I} \to \mathcal{D}(\mathcal{P})$ is such that

$\pi(\sigma \cap \tau) = \pi(\sigma) \cap \pi(\tau)$.

$\bigcup_{\sigma \in \mathcal{I}} \pi(\sigma) = \mathcal{P}$.

These conditions are precisely what are needed to ensure that its **join-lift** $\pi^{\vee}$ is a homomorphism of lattices!

A homomorphism appears!

In the topological approach to distributed computing, carrier maps associated to protocols must satisfy **extra conditions**.

Indeed, a **protocol map** $\pi : \mathcal{I} \to \mathcal{D}(\mathcal{P})$ is such that

$\pi(\sigma \cap \tau) = \pi(\sigma) \cap \pi(\tau)$.

$\bigcup_{\sigma \in \mathcal{I}} \pi(\sigma) = \mathcal{P}$.

These conditions are precisely what are needed to ensure that its **join-lift** $\pi^{\vee}$ is a homomorphism of lattices!

$\pi^{\vee} : \mathcal{D}(\mathcal{I}) \to \mathcal{D}(\mathcal{P})$ sends a subcomplex $J \subseteq \mathcal{I}$ to $\bigcup_{\sigma \in J} \pi(\sigma)$.

This map preserves unions but in general not intersections.

We thus obtain a **homomorphism** of lattices $\pi^{\vee} : \mathcal{D}(\mathcal{I}) \to \mathcal{D}(\mathcal{P})$.

A homomorphism appears!

In the topological approach to distributed computing, carrier maps associated to protocols must satisfy **extra conditions**.

Indeed, a **protocol map** $\pi : \mathcal{I} \to \mathcal{D}(\mathcal{P})$ is such that

$\pi(\sigma \cap \tau) = \pi(\sigma) \cap \pi(\tau)$.

$\bigcup_{\sigma \in \mathcal{I}} \pi(\sigma) = \mathcal{P}$.

These conditions are precisely what are needed to ensure that its **join-lift** $\pi^\vee$ is a homomorphism of lattices!

$\pi^\vee : \mathcal{D}(\mathcal{I}) \to \mathcal{D}(\mathcal{P})$ sends a subcomplex $J \subseteq \mathcal{I}$ to $\bigcup_{\sigma \in J} \pi(\sigma)$.

This map <u>preserves unions</u> but in general <u>not intersections</u>.

We thus obtain a **homomorphism** of lattices $\pi^\vee : \mathcal{D}(\mathcal{I}) \to \mathcal{D}(\mathcal{P})$.

Dualising, we obtain a **monotone map** $\rho : \mathcal{P} \to \mathcal{I}$.

Spectral semantics

Protocols, categorically

In the context of round-based, full-information, oblivious protocols...

The **protocol complex** is functorially related to the input complex $\mathcal{I}$:

$$\begin{tikzcd} \mathcal{I} \arrow[d, "\phi"'] \\ \mathcal{I}' \end{tikzcd} \qquad \mathsf{sComp} \xrightarrow{\mathcal{P}(-)} \mathsf{sComp} \qquad \begin{tikzcd} \mathcal{P}(\mathcal{I}) \arrow[d, "\mathcal{P}\phi"] \\ \mathcal{P}(\mathcal{I}'), \end{tikzcd}$$

Protocols, categorically

In the context of round-based, full-information, oblivious protocols. . .

The **protocol complex** is functorially related to the input complex $\mathcal{I}$:

$$\begin{tikzcd} \mathcal{I} \arrow[d, "\phi"'] \\ \mathcal{I}' \end{tikzcd} \qquad \mathsf{sComp} \xrightarrow{\mathcal{P}(-)} \mathsf{sComp} \qquad \begin{tikzcd} \mathcal{P}(\mathcal{I}) \arrow[d, "\mathcal{P}\phi"] \\ \mathcal{P}(\mathcal{I}'), \end{tikzcd}$$

and the **protocol map** is natural in $\mathcal{I}$:

$$\begin{tikzcd} \mathcal{I} \arrow[r, "\pi_{\mathcal{I}}"] \arrow[d, "\phi"'] & \mathcal{D}(\mathcal{P}(\mathcal{I})) \arrow[d, "\mathcal{D}(\mathcal{P}\phi)"] \\ \mathcal{I}' \arrow[r, "\pi_{\mathcal{I}'}"'] & \mathcal{D}(\mathcal{P}(\mathcal{I}')). \end{tikzcd}$$

Protocols, categorically

In the context of round-based, full-information, oblivious protocols. . .

The **protocol complex** is functorially related to the input complex $\mathcal{I}$:

$$\begin{tikzcd} \mathcal{I} \arrow[d, "\phi"'] \\ \mathcal{I}' \end{tikzcd} \qquad \mathsf{sComp} \xrightarrow{\mathcal{P}(-)} \mathsf{sComp} \qquad \begin{tikzcd} \mathcal{P}(\mathcal{I}) \arrow[d, "\mathcal{P}\phi"] \\ \mathcal{P}(\mathcal{I}'), \end{tikzcd}$$

and the **protocol map** is natural in $\mathcal{I}$:

$$\begin{tikzcd} \mathcal{D}(\mathcal{I}) \arrow[r, "\pi^{\vee}_{\mathcal{I}}"] \arrow[d, "\phi_*"'] & \mathcal{D}(\mathcal{P}(\mathcal{I})) \arrow[d, "\mathcal{D}(\mathcal{P}\phi)"] \\ \mathcal{D}(\mathcal{I})' \arrow[r, "\pi^{\vee}_{\mathcal{I}'}"'] & \mathcal{D}(\mathcal{P}(\mathcal{I}')). \end{tikzcd}$$

A directed limit

In particular, **iterating** a protocol $(\mathcal{P}, \pi)$ yields a sequence

$$\mathcal{D}(\mathcal{I}) \xrightarrow{\pi_{\mathcal{I}}^{\vee}} \mathcal{D}(\mathcal{P}(\mathcal{I})) \xrightarrow{\pi_{\mathcal{P}(\mathcal{I})}^{\vee}} \cdots \xrightarrow{\pi_{\mathcal{P}^2(\mathcal{I})}^{\vee}} \mathcal{D}(\mathcal{P}^n(\mathcal{I})) \xrightarrow{\pi_{\mathcal{P}^n(\mathcal{I})}^{\vee}} \cdots ,$$

A directed limit

In particular, **iterating** a protocol $(\mathcal{P}, \pi)$ yields a sequence

$$\mathcal{D}(\mathcal{I}) \xrightarrow{\pi^\vee_{\mathcal{I}}} \mathcal{D}(\mathcal{P}(\mathcal{I})) \xrightarrow{\pi^\vee_{\mathcal{P}(\mathcal{I})}} \cdots \xrightarrow{\pi^\vee_{\mathcal{P}^2(\mathcal{I})}} \mathcal{D}(\mathcal{P}^n(\mathcal{I})) \xrightarrow{\pi^\vee_{\mathcal{P}^n(\mathcal{I})}} \cdots,$$

Dualising, we obtain a sequence

$$\mathcal{I} \xleftarrow{\rho_0} \mathcal{P}(\mathcal{I}) \xleftarrow{\rho_1} \mathcal{P}^2(\mathcal{I}) \xleftarrow{\rho_2} \cdots \xleftarrow{\rho_{n-1}} \mathcal{P}^n(\mathcal{I}) \xleftarrow{\rho_n} \cdots$$

A directed limit

In particular, **iterating** a protocol $(\mathcal{P}, \pi)$ yields a sequence

$$\mathcal{D}(\mathcal{I}) \xrightarrow{\pi^{\vee}_{\mathcal{I}}} \mathcal{D}(\mathcal{P}(\mathcal{I})) \xrightarrow{\pi^{\vee}_{\mathcal{P}(\mathcal{I})}} \cdots \xrightarrow{\pi^{\vee}_{\mathcal{P}^2(\mathcal{I})}} \mathcal{D}(\mathcal{P}^n(\mathcal{I})) \xrightarrow{\pi^{\vee}_{\mathcal{P}^n(\mathcal{I})}} \cdots,$$

Dualising, we obtain a sequence

$$\mathcal{I} \xleftarrow{\rho_0} \mathcal{P}(\mathcal{I}) \xleftarrow{\rho_1} \mathcal{P}^2(\mathcal{I}) \xleftarrow{\rho_2} \cdots \xleftarrow{\rho_{n-1}} \mathcal{P}^n(\mathcal{I}) \xleftarrow{\rho_n} \cdots$$

The limit of this diagram is a **spectral space** denoted by $\mathcal{P}^{\infty}(\mathcal{I})$.

A directed limit

In particular, **iterating** a protocol $(\mathcal{P}, \pi)$ yields a sequence

$$\mathcal{D}(\mathcal{I}) \xrightarrow{\pi_{\mathcal{I}}^{\vee}} \mathcal{D}(\mathcal{P}(\mathcal{I})) \xrightarrow{\pi_{\mathcal{P}(\mathcal{I})}^{\vee}} \cdots \xrightarrow{\pi_{\mathcal{P}^2(\mathcal{I})}^{\vee}} \mathcal{D}(\mathcal{P}^n(\mathcal{I})) \xrightarrow{\pi_{\mathcal{P}^n(\mathcal{I})}^{\vee}} \cdots ,$$

Dualising, we obtain a sequence

$$\mathcal{I} \xleftarrow{\rho_0} \mathcal{P}(\mathcal{I}) \xleftarrow{\rho_1} \mathcal{P}^2(\mathcal{I}) \xleftarrow{\rho_2} \cdots \xleftarrow{\rho_{n-1}} \mathcal{P}^n(\mathcal{I}) \xleftarrow{\rho_n} \cdots$$

The limit of this diagram is a **spectral space** denoted by $\mathcal{P}^\infty(\mathcal{I})$.

This construction defines **a functor**

$$\begin{aligned} \mathcal{P}^\infty : \mathsf{sComp} &\longrightarrow \mathsf{Spec} \\ \mathcal{I} &\longmapsto \mathcal{P}^\infty(\mathcal{I}). \end{aligned}$$

Characterising task-solvability

Our contribution : spectral semantics

$$\mathcal{I} \xrightarrow{\pi} \mathcal{D}(\mathcal{P})$$

Our contribution : spectral semantics

$$\mathcal{D}(\mathcal{I}) \xrightarrow{\pi^{\vee}} \mathcal{D}(\mathcal{P})$$

Our contribution : spectral semantics

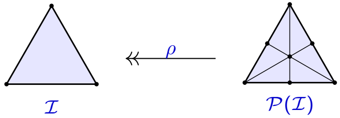
Duality

$$\mathcal{D}(\mathcal{I}) \xrightarrow{\pi^{\vee}} \mathcal{D}(\mathcal{P}) \longleftrightarrow \mathcal{I} \overset{\rho}{\twoheadleftarrow} \mathcal{P}$$

Our contribution : spectral semantics

Duality

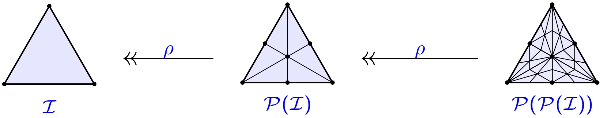
$\mathcal{D}(\mathcal{I}) \xrightarrow{\pi^\vee} \mathcal{D}(\mathcal{P})$ $\longleftrightarrow$ $\mathcal{I} \overset{\rho}{\twoheadleftarrow} \mathcal{P}$



Our contribution : spectral semantics

Duality

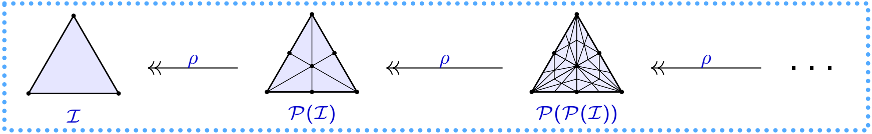
$\mathcal{D}(\mathcal{I}) \xrightarrow{\pi^{\vee}} \mathcal{D}(\mathcal{P})$ $\longleftrightarrow$ $\mathcal{I} \overset{\rho}{\twoheadleftarrow} \mathcal{P}$



Our contribution : spectral semantics

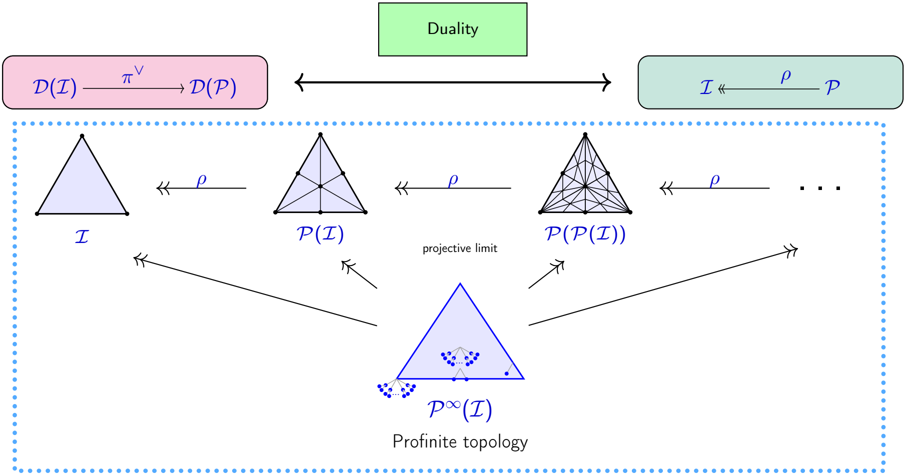
Duality

$\mathcal{D}(\mathcal{I}) \xrightarrow{\pi^\vee} \mathcal{D}(\mathcal{P})$ $\longleftrightarrow$ $\mathcal{I} \overset{\rho}{\twoheadleftarrow} \mathcal{P}$



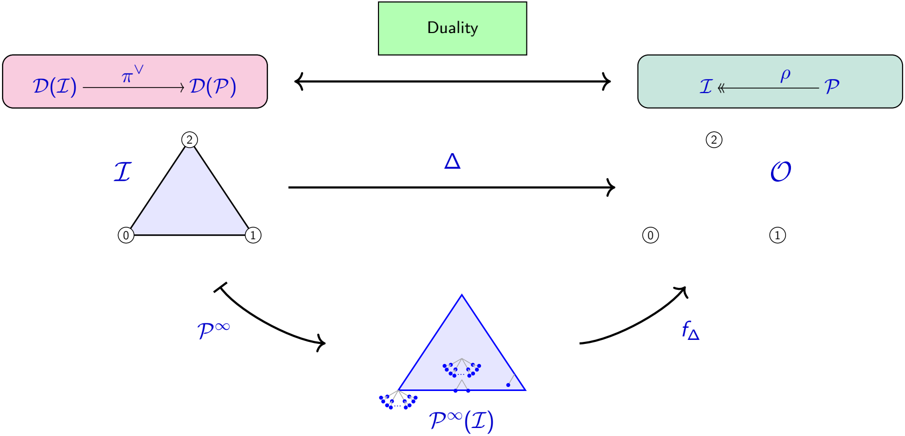
Pos_f

Our contribution : spectral semantics



Spec

Our contribution : spectral semantics

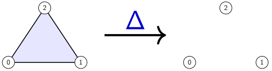


Theorem (**C.**, Godard)

A colorless task $(\mathcal{I}, \mathcal{O}, \Delta)$ *is solved by a protocol* $(\mathcal{P}, \pi)$ $\iff$ *There exists a spectral map* $f_\Delta : \mathcal{P}^\infty(\mathcal{I}) \to \mathcal{O}$.

Conclusion

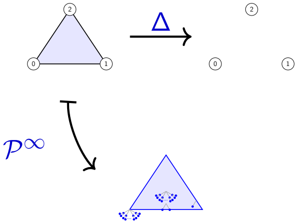
We provide a topological characterisation of task-solvability for a large class of iterated protocols.



Conclusion

We provide a topological characterisation of task-solvability for a large class of iterated protocols.

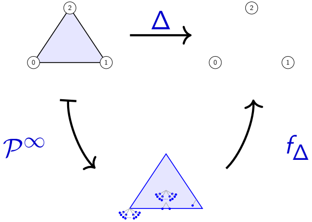
Protocols $\longleftrightarrow$ **Spectral spaces**



We provide a topological characterisation of task-solvability for a large class of iterated protocols.

Protocols $\longleftrightarrow$ **Spectral spaces**

Algorithms $\longleftrightarrow$ **Spectral maps**

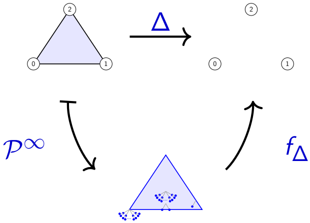


Conclusion

We provide a topological characterisation of task-solvability for a large class of iterated protocols.

Protocols $\longleftrightarrow$ **Spectral spaces**

Algorithms $\longleftrightarrow$ **Spectral maps**



The use of Stone duality and categorical methods allows us to encode

- distributed knowledge using T_0 topology,
- epistemic dynamics as a projective limit,

while generalising the original approach.

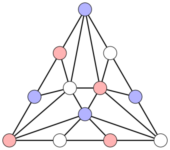
Future work

Distributed computability theorems

Models/tasks with identifiers

Study of labelled spectral spaces

Correspondence anonymous $\leftrightarrow$ identifiers.



Future work

Distributed computability theorems

Models/tasks with identifiers

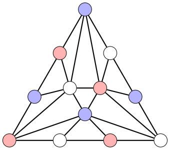
Study of labelled spectral spaces

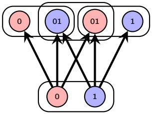
Correspondence anonymous $\leftrightarrow$ identifiers.

Asynchronous and probabilistic models

Event structures as protocols.

Spectral space associated to an event structure.





Future work

Distributed computability theorems

Models/tasks with identifiers

Study of labelled spectral spaces
Correspondence anonymous $\leftrightarrow$ identifiers.

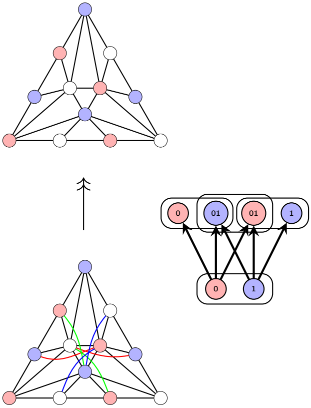
Asynchronous and probabilistic models

Event structures as protocols.
Spectral space associated to an event structure.

Equivalence of models

By topological methods

Bisimulation by retract.
Sub-spaces and sub-models.



Future work

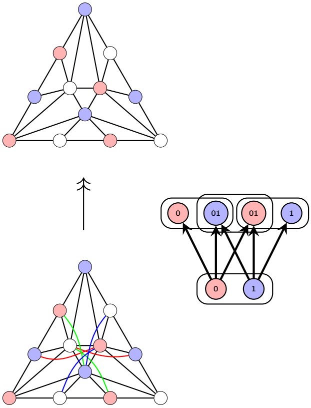
Distributed computability theorems

Models/tasks with identifiers

Study of labelled spectral spaces
Correspondence anonymous $\leftrightarrow$ identifiers.

Asynchronous and probabilistic models

Event structures as protocols.
Spectral space associated to an event structure.



Equivalence of models

By topological methods

Bisimulation by retract.
Sub-spaces and sub-models.

By algebraic and logical methods

Open maps and (co)algebras.
Epistemic-temporal logic via duality.

$$(D_U(\phi \Rightarrow \psi) \Rightarrow D_U\phi \Rightarrow D_U\psi)$$

Future work

Distributed computability theorems

Models/tasks with identifiers

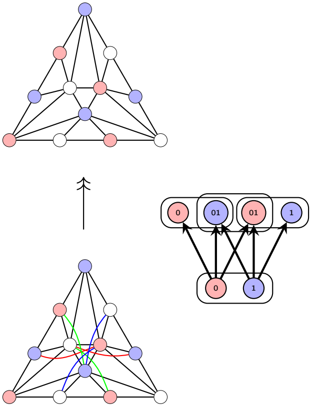
Study of labelled spectral spaces

Correspondence anonymous $\leftrightarrow$ identifiers.

Asynchronous and probabilistic models

Event structures as protocols.

Spectral space associated to an event structure.



Equivalence of models

By topological methods

Bisimulation by retract.

Sub-spaces and sub-models.

By algebraic and logical methods

Open maps and (co)algebras.

Epistemic-temporal logic via duality.

$$(D_U(\phi \Rightarrow \psi) \Rightarrow D_U\phi \Rightarrow D_U\psi)$$

Formalisation in a proof assistant

ROCQ

Thank you!

The spectral space of a protocol

We can describe the space $\mathcal{P}^\infty(\mathcal{I})$ concretely.

Points x_σ are **protocol sequences**, sequences (σ) of simplices with

- $\sigma_n \in \mathcal{P}^n(\mathcal{I})$ and,
- $\rho_n(\sigma_{n+1}) = \sigma_n$,

The **open topology** is generated by the sets

$$U_{n,\sigma} := \{(\tau) \in \mathcal{P}^\infty(\mathcal{I}) \mid \tau_n \geq \sigma\},$$

for $\sigma \in \mathcal{P}^n(\mathcal{I})$ *i.e.* the generating open sets in each $\mathcal{P}^n(\mathcal{I})$.

The **specialization order** $\leq$ on $\mathcal{P}^\infty(\mathcal{I})$ is defined by

$$x_\sigma \leq x_\tau \iff \quad \forall n \in \mathbb{N} \quad \sigma_n \geq \tau_n \quad \text{in } \mathcal{P}^n(\mathcal{I}),$$

the opposite order of the point-wise order on protocol sequences.

The spectral space of Iterated Immediate Snapshot

Recall that the IIS protocol corresponds to the **barycentric subdivision** functor $\mathcal{B} : \mathsf{sComp} \longrightarrow \mathsf{sComp}$.

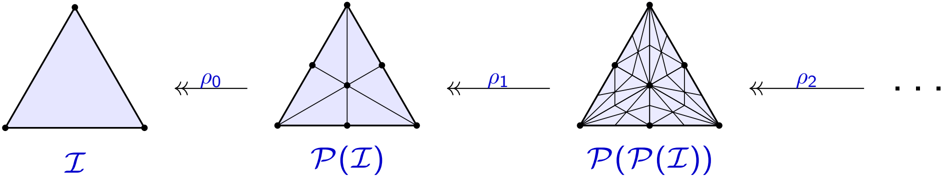
Protocol sequences consist of "**smaller and smaller**" simplices.

Thus, for every sequence (σ), $\exists! x \in |\mathcal{I}|$ such that $\{x\} = \bigcap |\sigma|$.

However, many sequences result in the same point in $|\mathcal{I}|$

For every $x \in |\mathcal{I}|$, there exists a unique **maximal** point (σ^x) such that $x \in \bigcap |\sigma^x|$.

This means that $|\mathcal{I}|$ is homeomorphic to the subspace of $\mathcal{B}^\infty(\mathcal{I})$ consisting of **maximal points**.



Revisiting the original result

We saw that for **barycentric subdivision**, the maximal points of $\mathcal{B}^\infty(\mathcal{I})$ contains $|\mathcal{I}|$ as its subspace of maximal points.

This falls into the context of **Wallman spaces**.

> Given a compact **normal space** *i.e.* $|\mathcal{I}|$,
> and the choice of a **normal** basis for its topology, *i.e.* $(|\sigma_n|)_{(\sigma)\in\mathcal{B}^\infty(\mathcal{I})}$,
> we obtain a spectral space *i.e.* $\mathcal{B}^\infty(\mathcal{I})$ whose **maximal points** are isomorphic to the normal space we started with.

A **mesh-shrinking** subdivision protocol defines a normal basis for $|\mathcal{I}|$.

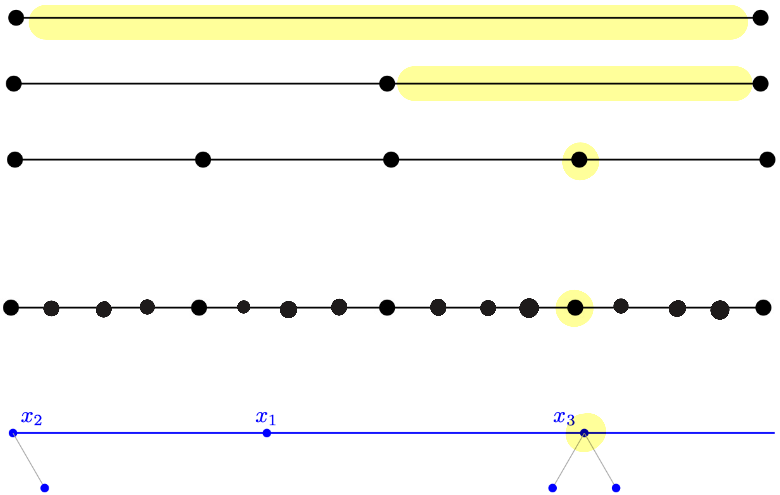
Thus, we recover the original theorem via the following lemma:

Lemma

For a mesh-shrinking protocol $(\mathcal{P},\pi)$, and a task $(\mathcal{I},\mathcal{O},\Delta)$, there exists a continuous map $|\mathcal{I}|\to|\mathcal{O}|$ respecting Δ if, and only if, there exists a spectral map $\mathcal{P}^\infty(\mathcal{I})\to\mathcal{O}$ respecting Δ.

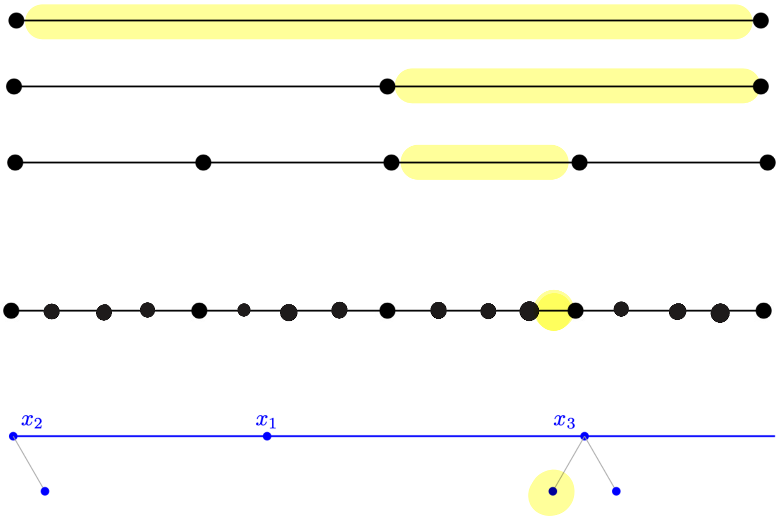
This is essentially **simplicial approximation** via spectral spaces.

The spectral space of Iterated Immediate Snapshot

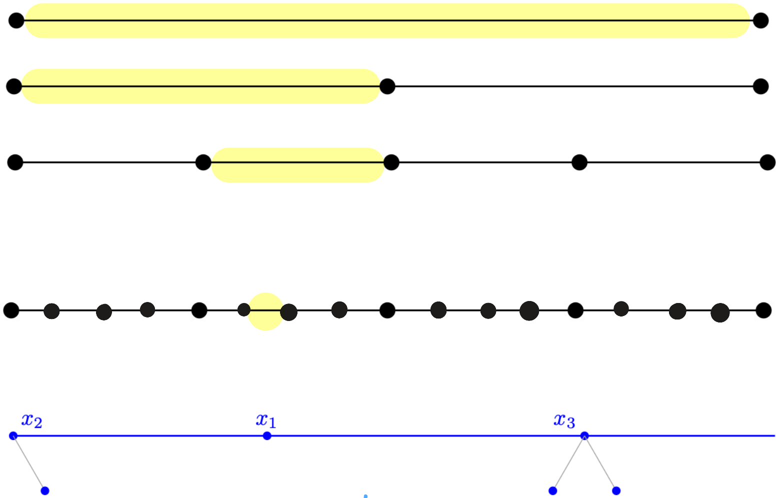


$x_3 = \frac{k}{2^n}$

The spectral space of Iterated Immediate Snapshot

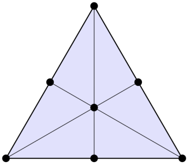


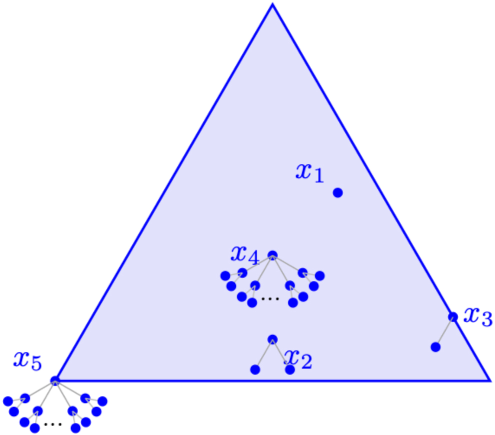
The spectral space of Iterated Immediate Snapshot



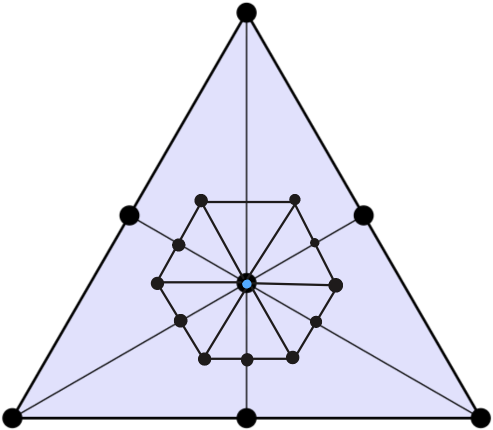
$x_1 \neq \frac{k}{2^n}$

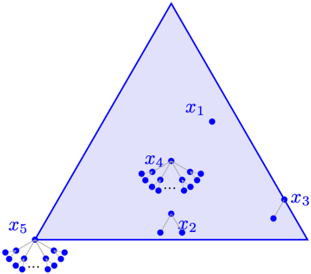
The spectral space of Iterated Immediate Snapshot



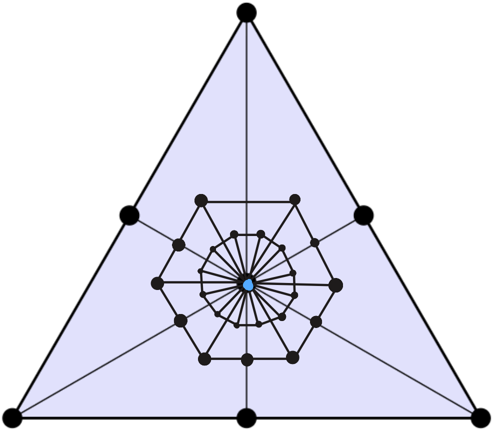


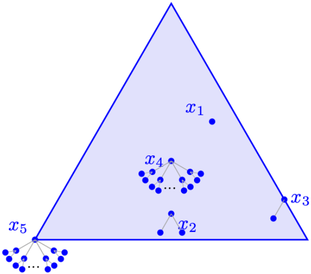
The spectral space of Iterated Immediate Snapshot





The spectral space of Iterated Immediate Snapshot





The spectral space of Iterated Immediate Snapshot