A Simplicial Model for KB4: Epistemic Logic with Agents that May Die

Éric Goubault, Jérémy Ledent and Sergio Rajsbaum

Friday 18 March, 2022

MSP Group, University of Strathclyde

Introduction

Epistemic Logic: Syntax

Let Ag be a finite set of agents and Prop a set of atomic propositions.

Syntax:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_a \varphi \qquad p \in \text{Prop},\ a \in \text{Ag}$$

Example formula: $K_a \neg K_b \varphi$ where $a, b \in \text{Ag}$

"a knows that b does not know that the formula φ is true."

Epistemic Logic: Syntax

Let Ag be a finite set of agents and Prop a set of atomic propositions.

Syntax:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_a \varphi \qquad p \in \mathrm{Prop},\ a \in \mathrm{Ag}$$

Example formula: $K_a \neg K_b \varphi$ where $a, b \in \mathrm{Ag}$

"a knows that b does not know that the formula φ is true."

In distributed computing:

Agents	$\longleftrightarrow$	Processes
Atomic propositions	$\longleftrightarrow$	Facts about the system

Epistemic Logic : Semantics

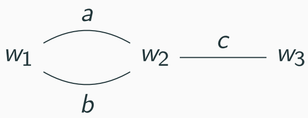
Kripke semantics : Based on Hintikka's idea of *"possible worlds"*.

Definition

An epistemic Kripke model $M = (W, \sim, L)$ is given by:

- a set W of possible worlds,
- for each $a \in \mathrm{Ag}$, an equivalence relation $\sim_a \subseteq W \times W$, called indistinguishability,
- a function $L : W \to \mathscr{P}(\mathrm{Prop})$ assigning atomic propositions to each world.

Example:



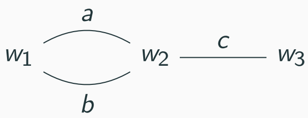
Kripke semantics : Based on Hintikka's idea of *"possible worlds"*.

Definition

An epistemic Kripke model $M = (W, \sim, L)$ is given by:

- a set W of possible worlds,
- for each $a \in \mathrm{Ag}$, an equivalence relation $\sim_a \subseteq W \times W$, called indistinguishability,
- a function $L : W \to \mathscr{P}(\mathrm{Prop})$ assigning atomic propositions to each world.

Example:



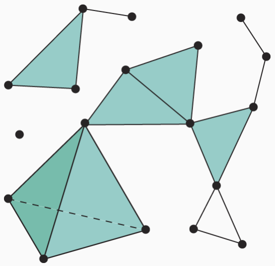
Satisfaction relation : $M, w \models K_a \varphi$ iff $M, w' \models \varphi$ for all w' such that $w \sim_a w'$.

Simplicial complexes

Definition

A simplicial complex is a pair (V, S) where:

- V is a set of vertices,
- S is a downward-closed family of subsets of V, called simplexes.



Simplicial complexes

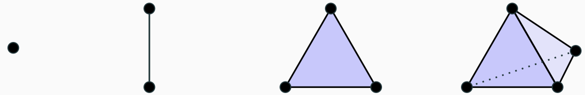
Definition

A simplicial complex is a pair (V, S) where:

- V is a set of vertices,
- S is a downward-closed family of subsets of V, called simplexes.

The dimension of a simplex $X \in S$ is $\dim(X) = |X| - 1$.

simplex



dimension	0	1	2	3	$\cdots$

Simplicial complexes

Definition

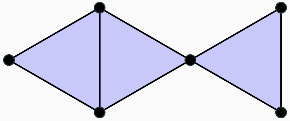
A simplicial complex is a pair (V, S) where:

- V is a set of vertices,
- S is a downward-closed family of subsets of V, called simplexes.

The dimension of a simplex $X \in S$ is $\dim(X) = |X| - 1$.

A facet is a simplex which is maximal w.r.t. inclusion.

A simplicial complex is pure if all facets have the same dimension.



Pure Simplicial Models

Definition (Goubault, Ledent, Rajsbaum (2018, 2021))

A pure simplicial model is given by $\mathscr{C} = (V, S, \chi, \ell)$ where:

- (V, S) is a pure simplicial complex.
- $\chi : V \to \text{Ag}$ is a colouring map, such that every simplex has vertices of different colour,
- $\ell : V \to \mathscr{P}(\text{Prop})$ assigns atomic propositions to the worlds.

Pure Simplicial Models

Definition (Goubault, Ledent, Rajsbaum (2018, 2021))

A pure simplicial model is given by $\mathscr{C} = (V, S, \chi, \ell)$ where:

- (V, S) is a pure simplicial complex.
- $\chi : V \to \mathrm{Ag}$ is a colouring map, such that every simplex has vertices of different colour,
- $\ell : V \to \mathscr{P}(\mathrm{Prop})$ assigns atomic propositions to the worlds.

Example: Consider a deck of four cards, $1, 2, 3, 4$, and three agents, a, b, c.
We deal one card to each agent, and the remaining card is kept hidden.

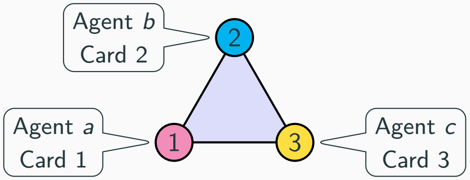
Pure Simplicial Models

Definition (Goubault, Ledent, Rajsbaum (2018, 2021))

A pure simplicial model is given by $\mathscr{C} = (V, S, \chi, \ell)$ where:

- (V, S) is a pure simplicial complex.
- $\chi : V \to \mathrm{Ag}$ is a colouring map, such that every simplex has vertices of different colour,
- $\ell : V \to \mathscr{P}(\mathrm{Prop})$ assigns atomic propositions to the worlds.

Example: Consider a deck of four cards, 1,2,3,4, and three agents, a, b, c.
We deal one card to each agent, and the remaining card is kept hidden.



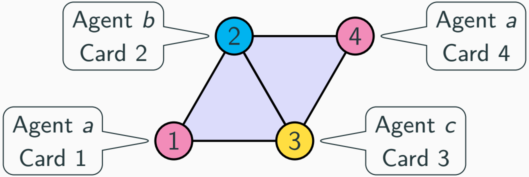
Pure Simplicial Models

Definition (Goubault, Ledent, Rajsbaum (2018, 2021))

A pure simplicial model is given by $\mathscr{C} = (V, S, \chi, \ell)$ where:

- (V, S) is a pure simplicial complex.
- $\chi : V \to \mathrm{Ag}$ is a colouring map, such that every simplex has vertices of different colour,
- $\ell : V \to \mathscr{P}(\mathrm{Prop})$ assigns atomic propositions to the worlds.

Example: Consider a deck of four cards, 1,2,3,4, and three agents, a, b, c.
We deal one card to each agent, and the remaining card is kept hidden.



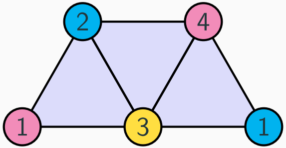
Pure Simplicial Models

Definition (Goubault, Ledent, Rajsbaum (2018, 2021))

A pure simplicial model is given by $\mathscr{C} = (V, S, \chi, \ell)$ where:

- (V, S) is a pure simplicial complex.
- $\chi : V \to \mathrm{Ag}$ is a colouring map, such that every simplex has vertices of different colour,
- $\ell : V \to \mathscr{P}(\mathrm{Prop})$ assigns atomic propositions to the worlds.

Example: Consider a deck of four cards, 1,2,3,4, and three agents, a, b, c.
We deal one card to each agent, and the remaining card is kept hidden.



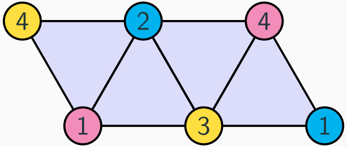
Pure Simplicial Models

Definition (Goubault, Ledent, Rajsbaum (2018, 2021))

A pure simplicial model is given by $\mathscr{C} = (V, S, \chi, \ell)$ where:

- (V, S) is a pure simplicial complex.
- $\chi : V \to \mathsf{Ag}$ is a colouring map, such that every simplex has vertices of different colour,
- $\ell : V \to \mathscr{P}(\mathsf{Prop})$ assigns atomic propositions to the worlds.

Example: Consider a deck of four cards, 1,2,3,4, and three agents, a, b, c.
We deal one card to each agent, and the remaining card is kept hidden.



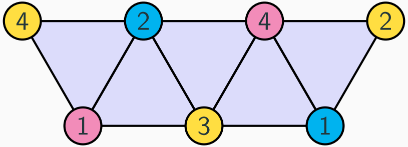
Pure Simplicial Models

Definition (Goubault, Ledent, Rajsbaum (2018, 2021))

A pure simplicial model is given by $\mathscr{C} = (V, S, \chi, \ell)$ where:

- (V, S) is a pure simplicial complex.
- $\chi : V \to \mathrm{Ag}$ is a colouring map, such that every simplex has vertices of different colour,
- $\ell : V \to \mathscr{P}(\mathrm{Prop})$ assigns atomic propositions to the worlds.

Example: Consider a deck of four cards, 1,2,3,4, and three agents, a, b, c.
We deal one card to each agent, and the remaining card is kept hidden.



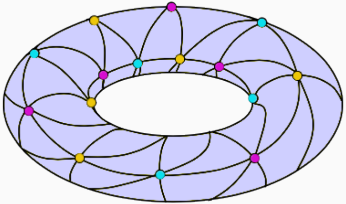
Pure Simplicial Models

Definition (Goubault, Ledent, Rajsbaum (2018, 2021))

A pure simplicial model is given by $\mathscr{C} = (V, S, \chi, \ell)$ where:

- (V, S) is a pure simplicial complex.
- $\chi : V \to \text{Ag}$ is a colouring map, such that every simplex has vertices of different colour,
- $\ell : V \to \mathscr{P}(\text{Prop})$ assigns atomic propositions to the worlds.

Example: Consider a deck of four cards, 1,2,3,4, and three agents, a, b, c.
We deal one card to each agent, and the remaining card is kept hidden.



Equivalence with Kripke models

Suppose the number of agents is $|\text{Ag}| = n + 1$.

Theorem (Goubault, Ledent, Rajsbaum (2018, 2021))

There is an equivalence of categories between the category of pure simplicial models of dimension n, and the category of proper and local Kripke models.

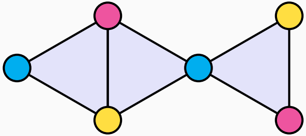
Equivalence with Kripke models

Suppose the number of agents is $|\mathrm{Ag}| = n+1$.

Theorem (Goubault, Ledent, Rajsbaum (2018, 2021))

There is an equivalence of categories between the category of pure simplicial models of dimension n, and the category of proper and local Kripke models.

Example: with three agents, $\mathrm{Ag} = \{ a, b, c \}$,



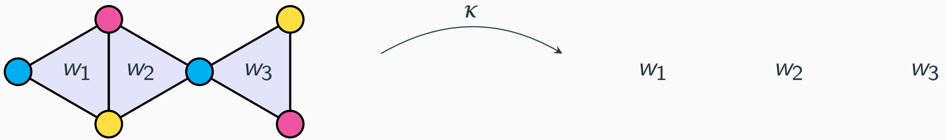
Equivalence with Kripke models

Suppose the number of agents is $|\mathrm{Ag}| = n+1$.

Theorem (Goubault, Ledent, Rajsbaum (2018, 2021))

There is an equivalence of categories between the category of pure simplicial models of dimension n, and the category of proper and local Kripke models.

Example: with three agents, $\mathrm{Ag} = \{$ a, b, c $\}$,



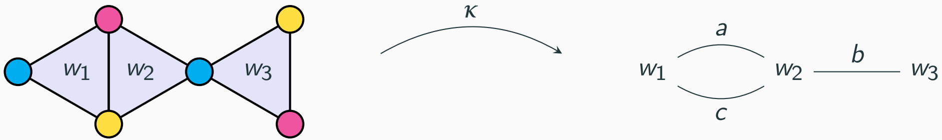
Equivalence with Kripke models

Suppose the number of agents is $|\mathrm{Ag}| = n+1$.

Theorem (Goubault, Ledent, Rajsbaum (2018, 2021))

There is an equivalence of categories between the category of pure simplicial models of dimension n, and the category of proper and local Kripke models.

Example: with three agents, $\mathrm{Ag} = \{ a, b, c \}$,

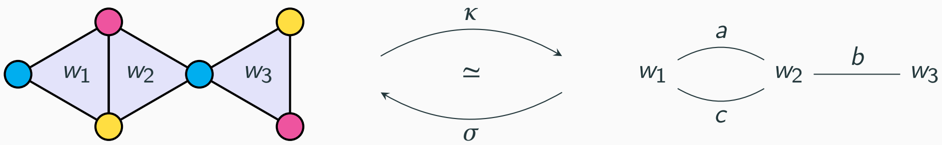


Suppose the number of agents is $|\mathrm{Ag}| = n + 1$.

Theorem (Goubault, Ledent, Rajsbaum (2018, 2021))

There is an equivalence of categories between the category of pure simplicial models of dimension n, and the category of proper and local Kripke models.

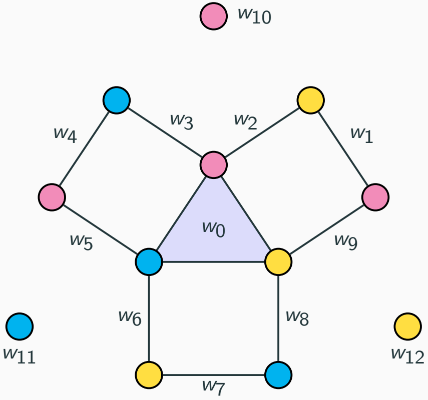
Example: with three agents, $\mathrm{Ag} = \{$ a, b, c $\}$,



Contribution

What about impure simplicial models?

- Previous paper: pure simplicial models, i.e., all worlds must have the same dimension.
- This paper: what happens if we lift this restriction?

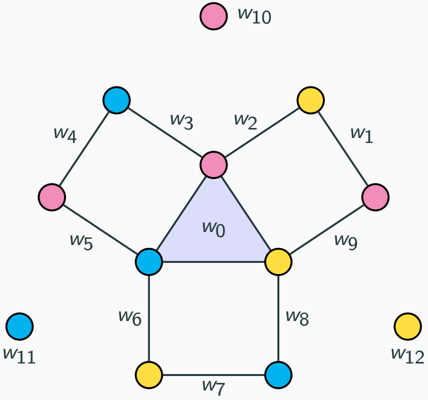


What about impure simplicial models?

- Previous paper: pure simplicial models, i.e., all worlds must have the same dimension.
- This paper: what happens if we lift this restriction?

Contributions:

- Find an equivalent class of Kripke models.
- Axiomatise the logic.

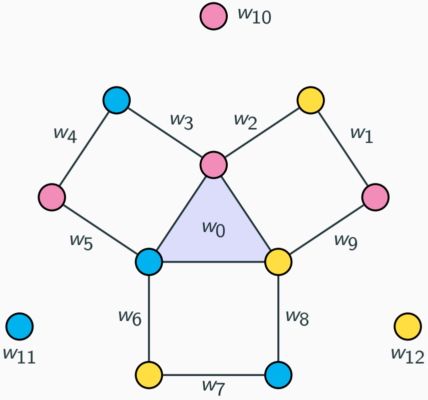


What about impure simplicial models?

- Previous paper: pure simplicial models, i.e., all worlds must have the same dimension.
- This paper: what happens if we lift this restriction?

Contributions:

- Find an equivalent class of Kripke models.
- Axiomatise the logic.



Satisfaction relation

- Define $\mathscr{C}, w \models \varphi$, where w is a facet of $\mathscr{C}$:

$$
\begin{array}{lll}
\mathscr{C}, w \models p & \text{iff} & p \in \ell(w) \\
\mathscr{C}, w \models \neg\varphi & \text{iff} & \mathscr{C}, w \not\models \varphi \\
\mathscr{C}, w \models \varphi \wedge \psi & \text{iff} & \mathscr{C}, w \models \varphi \text{ and } \mathscr{C}, w \models \psi \\
\mathscr{C}, w \models K_a \varphi & \text{iff} & \mathscr{C}, w' \models \varphi \text{ for all } w' \in \text{Facet}(\mathscr{C}) \text{ such that } a \in \chi(w \cap w')
\end{array}
$$

Satisfaction relation

- Define $\mathscr{C}, w \models \varphi$, where w is a facet of $\mathscr{C}$:

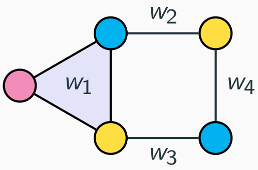
$$
\begin{array}{lll}
\mathscr{C}, w \models p & \text{iff} & p \in \ell(w) \\
\mathscr{C}, w \models \neg\varphi & \text{iff} & \mathscr{C}, w \not\models \varphi \\
\mathscr{C}, w \models \varphi \wedge \psi & \text{iff} & \mathscr{C}, w \models \varphi \text{ and } \mathscr{C}, w \models \psi \\
\mathscr{C}, w \models K_a \varphi & \text{iff} & \mathscr{C}, w' \models \varphi \text{ for all } w' \in \mathrm{Facet}(\mathscr{C}) \text{ such that } a \in \chi(w \cap w')
\end{array}
$$

Satisfaction relation

- Define $\mathscr{C}, w \models \varphi$, where w is a facet of $\mathscr{C}$:

$$
\begin{array}{lll}
\mathscr{C}, w \models p & \text{iff} & p \in \ell(w) \\
\mathscr{C}, w \models \neg\varphi & \text{iff} & \mathscr{C}, w \not\models \varphi \\
\mathscr{C}, w \models \varphi \wedge \psi & \text{iff} & \mathscr{C}, w \models \varphi \text{ and } \mathscr{C}, w \models \psi \\
\mathscr{C}, w \models K_a \varphi & \text{iff} & \mathscr{C}, w' \models \varphi \text{ for all } w' \in \mathrm{Facet}(\mathscr{C}) \text{ such that } a \in \chi(w \cap w')
\end{array}
$$

Example: with $\mathrm{Ag} = \{$ a , b , c $\}$ and $\mathrm{Prop} = \{p\}$.

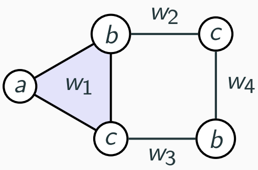


p is true in w_1 only.

- Define $\mathscr{C}, w \models \varphi$, where w is a facet of $\mathscr{C}$:

$$
\begin{array}{lll}
\mathscr{C}, w \models p & \text{iff} & p \in \ell(w) \\
\mathscr{C}, w \models \neg\varphi & \text{iff} & \mathscr{C}, w \not\models \varphi \\
\mathscr{C}, w \models \varphi \land \psi & \text{iff} & \mathscr{C}, w \models \varphi \text{ and } \mathscr{C}, w \models \psi \\
\mathscr{C}, w \models K_a \varphi & \text{iff} & \mathscr{C}, w' \models \varphi \text{ for all } w' \in \mathrm{Facet}(\mathscr{C}) \text{ such that } a \in \chi(w \cap w')
\end{array}
$$

Example: with $\mathrm{Ag} = \{a, b, c\}$ and $\mathrm{Prop} = \{p\}$.



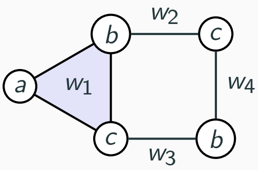
p is true in w_1 only.

Satisfaction relation

- Define $\mathscr{C}, w \models \varphi$, where w is a facet of $\mathscr{C}$:

$$
\begin{array}{lll}
\mathscr{C}, w \models p & \text{iff} & p \in \ell(w) \\
\mathscr{C}, w \models \neg\varphi & \text{iff} & \mathscr{C}, w \not\models \varphi \\
\mathscr{C}, w \models \varphi \wedge \psi & \text{iff} & \mathscr{C}, w \models \varphi \text{ and } \mathscr{C}, w \models \psi \\
\mathscr{C}, w \models K_a \varphi & \text{iff} & \mathscr{C}, w' \models \varphi \text{ for all } w' \in \mathrm{Facet}(\mathscr{C}) \text{ such that } a \in \chi(w \cap w')
\end{array}
$$

Example: with $\mathrm{Ag} = \{a, b, c\}$ and $\mathrm{Prop} = \{p\}$.



p is true in w_1 only.

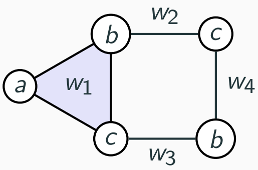
- $\mathscr{C}, w_1 \models K_a p$
- $\mathscr{C}, w_1 \models \neg K_b p$

Satisfaction relation

- Define $\mathscr{C}, w \models \varphi$, where w is a facet of $\mathscr{C}$:

$$\begin{array}{lll} \mathscr{C}, w \models p & \text{iff} & p \in \ell(w) \\ \mathscr{C}, w \models \neg\varphi & \text{iff} & \mathscr{C}, w \not\models \varphi \\ \mathscr{C}, w \models \varphi \wedge \psi & \text{iff} & \mathscr{C}, w \models \varphi \text{ and } \mathscr{C}, w \models \psi \\ \mathscr{C}, w \models K_a \varphi & \text{iff} & \mathscr{C}, w' \models \varphi \text{ for all } w' \in \text{Facet}(\mathscr{C}) \text{ such that } a \in \chi(w \cap w') \end{array}$$

Example: with $\text{Ag} = \{a, b, c\}$ and $\text{Prop} = \{p\}$.



p is true in w_1 only.

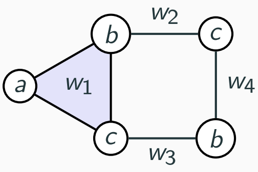
- $\mathscr{C}, w_1 \models K_a p$
- $\mathscr{C}, w_1 \models \neg K_b p$
- $\mathscr{C}, w_4 \models (K_b \neg p) \wedge (K_c \neg p)$

Satisfaction relation

- Define $\mathscr{C}, w \models \varphi$, where w is a facet of $\mathscr{C}$:

$$
\begin{array}{lll}
\mathscr{C}, w \models p & \text{iff} & p \in \ell(w) \\
\mathscr{C}, w \models \neg\varphi & \text{iff} & \mathscr{C}, w \not\models \varphi \\
\mathscr{C}, w \models \varphi \land \psi & \text{iff} & \mathscr{C}, w \models \varphi \text{ and } \mathscr{C}, w \models \psi \\
\mathscr{C}, w \models K_a\varphi & \text{iff} & \mathscr{C}, w' \models \varphi \text{ for all } w' \in \text{Facet}(\mathscr{C}) \text{ such that } a \in \chi(w \cap w')
\end{array}
$$

Example: with $\text{Ag} = \{a, b, c\}$ and $\text{Prop} = \{p\}$.



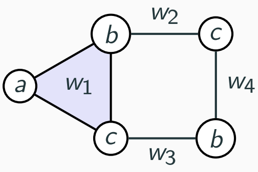
p is true in w_1 only.

- $\mathscr{C}, w_1 \models K_a p$
- $\mathscr{C}, w_1 \models \neg K_b p$
- $\mathscr{C}, w_4 \models (K_b \neg p) \land (K_c \neg p)$
- $\mathscr{C}, w_2 \models K_a p$

- Define $\mathscr{C}, w \models \varphi$, where w is a facet of $\mathscr{C}$:

$$
\begin{array}{lll}
\mathscr{C}, w \models p & \text{iff} & p \in \ell(w) \\
\mathscr{C}, w \models \neg\varphi & \text{iff} & \mathscr{C}, w \not\models \varphi \\
\mathscr{C}, w \models \varphi \wedge \psi & \text{iff} & \mathscr{C}, w \models \varphi \text{ and } \mathscr{C}, w \models \psi \\
\mathscr{C}, w \models K_a \varphi & \text{iff} & \mathscr{C}, w' \models \varphi \text{ for all } w' \in \mathrm{Facet}(\mathscr{C}) \text{ such that } a \in \chi(w \cap w')
\end{array}
$$

Example: with $\mathrm{Ag} = \{a, b, c\}$ and $\mathrm{Prop} = \{p\}$.



p is true in w_1 only.

- $\mathscr{C}, w_1 \models K_a p$
- $\mathscr{C}, w_1 \models \neg K_b p$
- $\mathscr{C}, w_4 \models (K_b \neg p) \wedge (K_c \neg p)$
- $\mathscr{C}, w_2 \models K_a p$
- $\mathscr{C}, w_1 \models K_b K_a p$

Dead or Alive?

Define the following formulas:

$$\mathrm{dead}(a) = K_a\,\mathrm{false} \qquad\qquad \mathrm{alive}(a) = \neg\,\mathrm{dead}(a)$$

One can check that:

$$\mathscr{C}, w \models \mathrm{alive}(a) \quad \text{iff} \quad a \in \chi(w)$$

Dead or Alive?

Define the following formulas:

$$\mathrm{dead}(a) = K_a\,\mathrm{false} \qquad\qquad \mathrm{alive}(a) = \neg\,\mathrm{dead}(a)$$

One can check that:

$$\mathscr{C}, w \models \mathrm{alive}(a) \quad \text{iff} \quad a \in \chi(w)$$

In every simplicial model:

- Dead agents know everything: $\models \mathrm{dead}(a) \implies K_a\varphi$.
- Alive agents satisfy Axiom **T**: $\models \mathrm{alive}(a) \implies (K_a\varphi \Rightarrow \varphi)$.
- Alive agents know they are alive: $\models \mathrm{alive}(a) \implies (K_a\,\mathrm{alive}(a))$.

Axiomatisation

KB4: the following axioms are valid in all simplicial models.

$$\mathbf{K} : K_a\varphi \land K_a(\varphi \Rightarrow \psi) \implies K_a\psi \qquad \mathbf{B} : \varphi \implies K_a\neg K_a\neg\varphi \qquad \mathbf{4} : K_a\varphi \implies K_aK_a\varphi$$

Axiomatisation

KB4: the following axioms are valid in all simplicial models.

$$\mathbf{K} : K_a\varphi \wedge K_a(\varphi \Rightarrow \psi) \implies K_a\psi \qquad \mathbf{B} : \varphi \implies K_a\neg K_a\neg\varphi \qquad \mathbf{4} : K_a\varphi \implies K_aK_a\varphi$$

Two extra axioms: not provable in **KB4**, but valid in all simplicial models.

- **NE**: there is at least one alive agent.
- **SA**: if an agent alone, then this agent knows that it is alone.

$$\mathbf{NE} : \bigvee_{a\in \mathrm{Ag}} \mathrm{alive}(a) \qquad \mathbf{SA}_a : \left(\mathrm{alive}(a) \wedge \bigwedge_{b\in \mathrm{Ag}\setminus\{a\}} \mathrm{dead}(b)\right) \implies K_a \bigwedge_{b\in \mathrm{Ag}\setminus\{a\}} \mathrm{dead}(b)$$

Axiomatisation

KB4: the following axioms are valid in all simplicial models.

$$\mathbf{K} : K_a\varphi \wedge K_a(\varphi \Rightarrow \psi) \implies K_a\psi \qquad \mathbf{B} : \varphi \implies K_a\neg K_a\neg\varphi \qquad \mathbf{4} : K_a\varphi \implies K_aK_a\varphi$$

Two extra axioms: not provable in **KB4**, but valid in all simplicial models.

- **NE**: there is at least one alive agent.
- **SA**: if an agent alone, then this agent knows that it is alone.

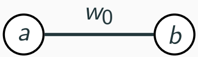
$$\mathbf{NE} : \bigvee_{a\in \mathrm{Ag}} \mathrm{alive}(a) \qquad \mathbf{SA}_a : \left(\mathrm{alive}(a) \wedge \bigwedge_{b\in \mathrm{Ag}\setminus\{a\}} \mathrm{dead}(b)\right) \implies K_a \bigwedge_{b\in \mathrm{Ag}\setminus\{a\}} \mathrm{dead}(b)$$

Theorem (Completeness)

The axiom system ***KB4*** *+* ***NE*** *+* ***SA*** *is sound and complete w.r.t. (impure) simplicial models.*

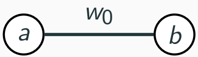
Extensions (Journal version)

Worlds are facets vs. Worlds are simplexes.



$\Rightarrow$ Axiom **SA** is false

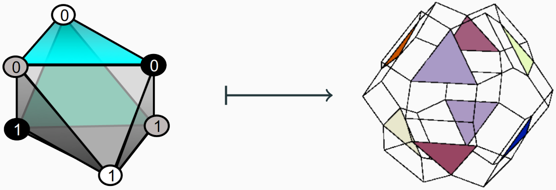
Worlds are facets vs. Worlds are simplexes.



$\Rightarrow$ Axiom **SA** is false

Dynamic Epistemic Logic.

Synchronous message-passing model with crashes



Conclusions

Related work.

Hans van Ditmarsch (WoLLIC 2021):
Wanted Dead or Alive: Epistemic Logic for Impure Simplicial Complexes.

- Dead agents know nothing.
- Axiom **T** is true, Axiom **K** is false.
- Complete axiomatization is an open question.

Conclusions

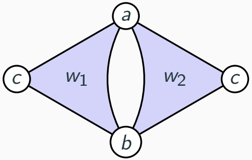
Related work.

Hans van Ditmarsch (WoLLIC 2021):
Wanted Dead or Alive: Epistemic Logic for Impure Simplicial Complexes.

- Dead agents know nothing.
- Axiom **T** is true, Axiom **K** is false.
- Complete axiomatization is an open question.

Future work.

- Many other variants are possible.



Conclusions

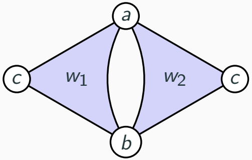
Related work.

Hans van Ditmarsch (WoLLIC 2021):
Wanted Dead or Alive: Epistemic Logic for Impure Simplicial Complexes.

- Dead agents know nothing.
- Axiom **T** is true, Axiom **K** is false.
- Complete axiomatization is an open question.

Future work.

- Many other variants are possible.
- The importance of topology is well-established in distributed computing.



Conclusions

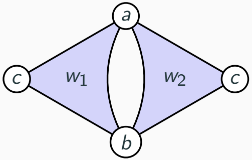
Related work.

Hans van Ditmarsch (WoLLIC 2021):
Wanted Dead or Alive: Epistemic Logic for Impure Simplicial Complexes.

- Dead agents know nothing.
- Axiom **T** is true, Axiom **K** is false.
- Complete axiomatization is an open question.

Future work.

- Many other variants are possible.
- The importance of topology is well-established in distributed computing.
- Can we make use of it for other applications?



Conclusions

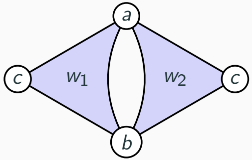
Related work.

Hans van Ditmarsch (WoLLIC 2021):
Wanted Dead or Alive: Epistemic Logic for Impure Simplicial Complexes.

- Dead agents know nothing.
- Axiom **T** is true, Axiom **K** is false.
- Complete axiomatization is an open question.

Future work.

- Many other variants are possible.
- The importance of topology is well-established in distributed computing.
- Can we make use of it for other applications?



Thanks for listening!