Epistemic logics based on abilities

PhD in Computer Science

Andrés R. Saravia

LIIS group (Logics, Interaction and Intelligent Systems)
FaMAF, Universidad Nacional de Córdoba

26/XI/2024

Organigrama

Motivation

- Modal Logics: Modes of truth about expressions

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is cold in the city*

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*
  - *The students assist in the morning*

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*
  - *The students are obliged to assist in the morning*

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*
  - *The students are obliged to assist in the morning*
- Epistemic Logics: Modal Logics that reason about the knowledge of the agents

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*
  - *The students are obliged to assist in the morning*
- Epistemic Logics: Modal Logics that reason about the knowledge of the agents
- Usually describes the “knowing that” (Hintikka, 1962)

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*
  - *The students are obliged to assist in the morning*
- Epistemic Logics: Modal Logics that reason about the knowledge of the agents
- Usually describes the “knowing that” (Hintikka, 1962)
- Knowledge of the agents about propositional facts

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*
  - *The students are obliged to assist in the morning*
- Epistemic Logics: Modal Logics that reason about the knowledge of the agents
- Usually describes the “knowing that” (Hintikka, 1962)
- Knowledge of the agents about propositional facts
  - $K_i\varphi$: *“the agent i knows that φ holds”*

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*
  - *The students are obliged to assist in the morning*
- Epistemic Logics: Modal Logics that reason about the knowledge of the agents
- Usually describes the “knowing that” (Hintikka, 1962)
- Knowledge of the agents about propositional facts
  - $\mathrm{K}_i\varphi$: *“the agent i knows that* φ *holds”*
  - *John knows that the day is cloudy*

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*
  - *The students are obliged to assist in the morning*
- Epistemic Logics: Modal Logics that reason about the knowledge of the agents
- Usually describes the “knowing that” (Hintikka, 1962)
- Knowledge of the agents about propositional facts
  - $K_i\varphi$: *“the agent i knows that φ holds”*
  - *John knows that the day is cloudy*
  - *The robot knows that it is in the kitchen*

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*
  - *The students are obliged to assist in the morning*
- Epistemic Logics: Modal Logics that reason about the knowledge of the agents
- Usually describes the “knowing that” (Hintikka, 1962)
- Knowledge of the agents about propositional facts
  - $K_i\varphi$: *“the agent i knows that* φ *holds”*
  - *John knows that the day is cloudy*
  - *The robot knows that it is in the kitchen*
- Other patterns of knowledge

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*
  - *The students are obliged to assist in the morning*
- Epistemic Logics: Modal Logics that reason about the knowledge of the agents
- Usually describes the “knowing that” (Hintikka, 1962)
- Knowledge of the agents about propositional facts
  - $K_i\varphi$: *“the agent i knows that φ holds”*
  - *John knows that the day is cloudy*
  - *The robot knows that it is in the kitchen*
- Other patterns of knowledge
  - knowing why, knowing whether, knowing who, knowing the value

Motivation

- Modal Logics: Modes of truth about expressions
  - Necessity, possibility, obligations, and so on
  - *It is possible that is cold in the city*
  - *The students are obliged to assist in the morning*
- Epistemic Logics: Modal Logics that reason about the knowledge of the agents
- Usually describes the "knowing that" (Hintikka, 1962)
- Knowledge of the agents about propositional facts
  - $\mathrm{K}_i\varphi$: *"the agent i knows that φ holds"*
  - *John knows that the day is cloudy*
  - *The robot knows that it is in the kitchen*
- Other patterns of knowledge
  - knowing why, knowing whether, knowing who, knowing the value, knowing how

Knowing How

- Knowledge of the agents about their own *abilities* to reach goals

Knowing How

- Knowledge of the agents about their own *abilities* to reach goals
- Epistemic logic of Knowing How based on Labeled Transition Systems (LTSs) (Wang, 2015)

Knowing How

- Knowledge of the agents about their own *abilities* to reach goals
- Epistemic logic of Knowing How based on Labeled Transition Systems (LTSs) (Wang, 2015)
  - Kh(ψ, φ): *"when ψ holds, the agent knows how to achieve φ"*

Knowing How

- Knowledge of the agents about their own *abilities* to reach goals
- Epistemic logic of Knowing How based on Labeled Transition Systems (LTSs) (Wang, 2015)
  - Kh(ψ, φ): *"when ψ holds, the agent knows how to achieve φ"*
- Generalize this framework

Knowing How

- Knowledge of the agents about their own *abilities* to reach goals
- Epistemic logic of Knowing How based on Labeled Transition Systems (LTSs) (Wang, 2015)
  - Kh(ψ, φ): *“when ψ holds, the agent knows how to achieve φ”*
- Generalize this framework
  - Incorporate an uncertainty component between plans

Knowing How

- Knowledge of the agents about their own *abilities* to reach goals
- Epistemic logic of Knowing How based on Labeled Transition Systems (LTSs) (Wang, 2015)
  - Kh(ψ, φ): *"when ψ holds, the agent knows how to achieve φ"*
- Generalize this framework
  - Incorporate an uncertainty component between plans
  - Introduce multiple and less idealized agents

An example: Evacuation plan

EMERGENCY PROCEDURE

IN CASE OF FIRE

KEEP CALM

PULL FIRE ALARM

FROM A SAFE LOCATION

CALL 999 (FIRE BRIGADE)

Evacuation: use only stairs or ramps, avoid elevators.

If unsafe to evacuate: close door, block cracks, stay low near windows.

An example: Evacuation plan

Emergency procedure

IN CASE OF FIRE

KEEP CALM

PULL FIRE ALARM

FROM A SAFE LOCATION

CALL 999 (FIRE BRIGADE)

Evacuation: use only stairs or ramps, avoid elevators.

If unsafe to evacuate: close door, block cracks, stay low near windows.

An example: Evacuation plan

Emergency procedure

IN CASE OF FIRE

KEEP CALM

PULL FIRE ALARM

FROM A SAFE LOCATION

CALL 999 (FIRE BRIGADE)

Evacuation: use only stairs or ramps, avoid elevators.

If unsafe to evacuate: close door, block cracks, stay low near windows.

An example: Evacuation plan

EMERGENCY PROCEDURE

IN CASE OF FIRE

KEEP CALM

PULL FIRE ALARM

FROM A SAFE LOCATION

CALL 999 (FIRE BRIGADE)

Evacuation: use only stairs or ramps, avoid elevators.

If unsafe to evacuate: close door, block cracks, stay low near windows.

An example: Evacuation plan

EMERGENCY PROCEDURE

IN CASE OF FIRE

KEEP CALM

PULL FIRE ALARM

FROM A SAFE LOCATION

CALL 999 (FIRE BRIGADE)

Evacuation: use only stairs or ramps, avoid elevators.

If unsafe to evacuate: close door, block cracks, stay low near windows.

Consider two agents i and j:

An example: Evacuation plan

EMERGENCY PROCEDURE

IN CASE OF FIRE

KEEP CALM

PULL FIRE ALARM

FROM A SAFE LOCATION

CALL 999 (FIRE BRIGADE)

Evacuation: use only stairs or ramps, avoid elevators.

If unsafe to evacuate: close door, block cracks, stay low near windows.

Consider two agents *i* and *j*:

- *i* took a safety course and avoids using the elevator

An example: Evacuation plan

EMERGENCY PROCEDURE

IN CASE OF FIRE

KEEP CALM

PULL FIRE ALARM

FROM A SAFE LOCATION

CALL 999 (FIRE BRIGADE)

Evacuation: use only stairs or ramps, avoid elevators.

If unsafe to evacuate: close door, block cracks, stay low near windows.

Consider two agents i and j:

- i took a safety course and avoids using the elevator
- j considers all evacuation options as valid

An example: Simplifying

Basic actions (Act)

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathrm{u_s}$)

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathrm{u_s}$)
- use ramp ($\mathrm{u_r}$)

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathtt{u_s}$)
- use ramp ($\mathtt{u_r}$)
- use elevator ($\mathtt{u_e}$)

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathtt{u_s}$)
- use ramp ($\mathtt{u_r}$)
- use elevator ($\mathtt{u_e}$)
- call fire brigade ($\mathtt{c_{fb}}$)

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathtt{u_s}$)
- use ramp ($\mathtt{u_r}$)
- use elevator ($\mathtt{u_e}$)
- call fire brigade ($\mathtt{c_{fb}}$)

Recommended plans: $\mathtt{u_s c_{fb}}$, $\mathtt{u_r c_{fb}}$

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathtt{u_s}$)
- use ramp ($\mathtt{u_r}$)
- use elevator ($\mathtt{u_e}$)
- call fire brigade ($\mathtt{c_{fb}}$)

Recommended plans: $\mathtt{u_s c_{fb}}$, $\mathtt{u_r c_{fb}}$
<u>Not</u> recommended plan: $\mathtt{u_e c_{fb}}$

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathtt{u_s}$)
- use ramp ($\mathtt{u_r}$)
- use elevator ($\mathtt{u_e}$)
- call fire brigade ($\mathtt{c_{fb}}$)

Recommended plans: $\mathtt{u_s c_{fb}}$, $\mathtt{u_r c_{fb}}$
Not recommended plan: $\mathtt{u_e c_{fb}}$
Agents (Agt)

- Agent *i* separates recommended ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}$) from not ($\{\mathtt{u_e c_{fb}}\}$)

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathtt{u_s}$)
- use ramp ($\mathtt{u_r}$)
- use elevator ($\mathtt{u_e}$)
- call fire brigade ($\mathtt{c_{fb}}$)

Recommended plans: $\mathtt{u_s c_{fb}}$, $\mathtt{u_r c_{fb}}$

Not recommended plan: $\mathtt{u_e c_{fb}}$

Agents (Agt)

- Agent *i* separates recommended ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}$) from not ($\{\mathtt{u_e c_{fb}}\}$)
- Agent *j* does not ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}$)

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathtt{u_s}$)
- use ramp ($\mathtt{u_r}$)
- use elevator ($\mathtt{u_e}$)
- call fire brigade ($\mathtt{c_{fb}}$)

Recommended plans: $\mathtt{u_s c_{fb}}$, $\mathtt{u_r c_{fb}}$
Not recommended plan: $\mathtt{u_e c_{fb}}$
Agents (Agt)

- Agent *i* separates recommended ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}$) from not ($\{\mathtt{u_e c_{fb}}\}$)
- Agent *j* does not ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}$)

States ($\mathbb{S}$) and propositional variables (Prop)

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathtt{u_s}$)
- use ramp ($\mathtt{u_r}$)
- use elevator ($\mathtt{u_e}$)
- call fire brigade ($\mathtt{c_{fb}}$)

Recommended plans: $\mathtt{u_s c_{fb}}$, $\mathtt{u_r c_{fb}}$
Not recommended plan: $\mathtt{u_e c_{fb}}$
Agents (Agt)

- Agent *i* separates recommended ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}$) from not ($\{\mathtt{u_e c_{fb}}\}$)
- Agent *j* does not ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}$)

States ($\mathbb{S}$) and propositional variables (Prop)

- w_1: there is a fire (f) and the protocol can be followed (c)

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathtt{u_s}$)
- use ramp ($\mathtt{u_r}$)
- use elevator ($\mathtt{u_e}$)
- call fire brigade ($\mathtt{c_{fb}}$)

Recommended plans: $\mathtt{u_s c_{fb}}$, $\mathtt{u_r c_{fb}}$
Not recommended plan: $\mathtt{u_e c_{fb}}$
Agents (Agt)

- Agent *i* separates recommended ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}$) from not ($\{\mathtt{u_e c_{fb}}\}$)
- Agent *j* does not ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}$)

States ($\mathbb{S}$) and propositional variables (Prop)

- w_1: there is a fire (f) and the protocol can be followed (c)
- w_2: it is a safe location (s)

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathtt{u_s}$)
- use ramp ($\mathtt{u_r}$)
- use elevator ($\mathtt{u_e}$)
- call fire brigade ($\mathtt{c_{fb}}$)

Recommended plans: $\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}$
Not recommended plan: $\mathtt{u_e c_{fb}}$
Agents (Agt)

- Agent *i* separates recommended ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}$) from not ($\{\mathtt{u_e c_{fb}}\}$)
- Agent *j* does not ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}$)

States (S) and propositional variables (Prop)

- w_1: there is a fire (f) and the protocol can be followed (c)
- w_2: it is a safe location (s)
- w_3: there is a fire (f), but the protocol can not be followed ($\neg c$)

An example: Simplifying

Basic actions (Act)

- use stairs ($\mathtt{u_s}$)
- use ramp ($\mathtt{u_r}$)
- use elevator ($\mathtt{u_e}$)
- call fire brigade ($\mathtt{c_{fb}}$)

Recommended plans: $\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}$
Not recommended plan: $\mathtt{u_e c_{fb}}$
Agents (Agt)

- Agent i separates recommended ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}$) from not ($\{\mathtt{u_e c_{fb}}\}$)
- Agent j does not ($\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}$)

States ($\mathbb{S}$) and propositional variables (Prop)

- w_1: there is a fire (f) and the protocol can be followed (c)
- w_2: it is a safe location (s)
- w_3: there is a fire (f), but the protocol can not be followed ($\neg c$)

Act $= \{\mathtt{u_s}, \mathtt{u_r}, \mathtt{u_e}, \mathtt{c_{fb}}\}$, Agt $= \{i, j\}$, $\mathbb{S} = \{w_1, w_2, w_3\}$ y Prop $= \{f, c, s\}$

Uncertainty-based LTSs (LTS^{U}s)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}, \mathrm{V} \rangle$:

Uncertainty-based LTSs (LTS^Us)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}, \mathrm{V} \rangle$:

- S is a non-empty set of states

Uncertainty-based LTSs (LTS^Us)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}, \mathrm{V} \rangle$:

- S is a non-empty set of states
- R_a a binary relation over S

Uncertainty-based LTSs (LTS^Us)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}, \mathrm{V} \rangle$:

- S is a non-empty set of states
- R_a a binary relation over S
- $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathsf{Prop})$ is a valuation function

Uncertainty-based LTSs (LTS^Us)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}, \mathrm{V} \rangle$:

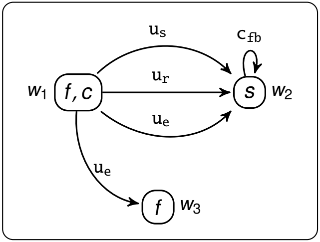
- S is a non-empty set of states
- R_a a binary relation over S
- $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathsf{Prop})$ is a valuation function
- $\mathrm{U}(i) \subseteq \mathcal{P}(\mathsf{Act}^*)$ s.t. if $\pi_1, \pi_2 \in \mathrm{U}(i)$ and $\pi_1 \neq \pi_2$, then $\pi_1 \cap \pi_2 = \emptyset$

Uncertainty-based LTSs (LTS^Us)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}, \mathrm{V} \rangle$:

- S is a non-empty set of states
- R_a a binary relation over S
- $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathsf{Prop})$ is a valuation function
- $\mathrm{U}(i) \subseteq \mathcal{P}(\mathsf{Act}^*)$ s.t. if $\pi_1, \pi_2 \in \mathrm{U}(i)$ and $\pi_1 \neq \pi_2$, then $\pi_1 \cap \pi_2 = \emptyset$

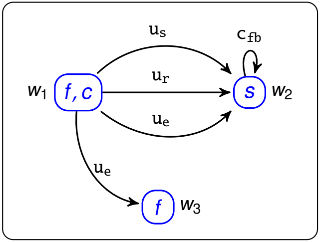


Uncertainty-based LTSs (LTS^Us)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathrm{Act}}, \{\mathrm{U}(i)\}_{i \in \mathrm{Agt}}, \mathrm{V} \rangle$:

- S is a non-empty set of states
- R_a a binary relation over S
- $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathrm{Prop})$ is a valuation function
- $\mathrm{U}(i) \subseteq \mathcal{P}(\mathrm{Act}^*)$ s.t. if $\pi_1, \pi_2 \in \mathrm{U}(i)$ and $\pi_1 \neq \pi_2$, then $\pi_1 \cap \pi_2 = \emptyset$



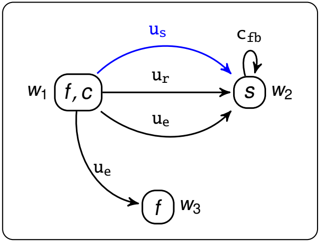
- $\mathrm{S} = \{w_1, w_2, w_3\}$

Uncertainty-based LTSs (LTS^Us)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathrm{Act}}, \{\mathrm{U}(i)\}_{i \in \mathrm{Agt}}, \mathrm{V} \rangle$:

- S is a non-empty set of states
- R_a a binary relation over S
- $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathrm{Prop})$ is a valuation function
- $\mathrm{U}(i) \subseteq \mathcal{P}(\mathrm{Act}^*)$ s.t. if $\pi_1, \pi_2 \in \mathrm{U}(i)$ and $\pi_1 \neq \pi_2$, then $\pi_1 \cap \pi_2 = \emptyset$



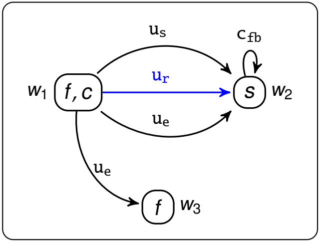
- $\mathrm{S} = \{w_1, w_2, w_3\}$
- $\mathrm{R}_{\mathrm{u_s}} = \mathrm{R}_{\mathrm{u_r}} = \{(w_1, w_2)\}$

Uncertainty-based LTSs (LTS^Us)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathrm{Act}}, \{\mathrm{U}(i)\}_{i \in \mathrm{Agt}}, \mathrm{V} \rangle$:

- S is a non-empty set of states
- R_a a binary relation over S
- $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathrm{Prop})$ is a valuation function
- $\mathrm{U}(i) \subseteq \mathcal{P}(\mathrm{Act}^*)$ s.t. if $\pi_1, \pi_2 \in \mathrm{U}(i)$ and $\pi_1 \neq \pi_2$, then $\pi_1 \cap \pi_2 = \emptyset$

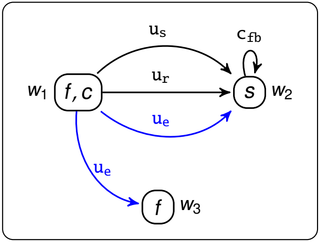


- $\mathrm{S} = \{w_1, w_2, w_3\}$
- $\mathrm{R}_{\mathtt{u_s}} = \mathrm{R}_{\mathtt{u_r}} = \{(w_1, w_2)\}$

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}, \mathrm{V} \rangle$:

- S is a non-empty set of states
- R_a a binary relation over S
- $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathsf{Prop})$ is a valuation function
- $\mathrm{U}(i) \subseteq \mathcal{P}(\mathsf{Act}^*)$ s.t. if $\pi_1, \pi_2 \in \mathrm{U}(i)$ and $\pi_1 \neq \pi_2$, then $\pi_1 \cap \pi_2 = \emptyset$

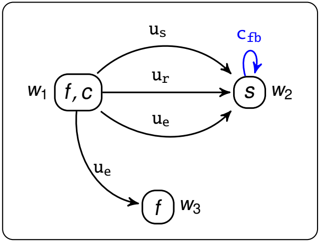


- $\mathrm{S} = \{w_1, w_2, w_3\}$
- $\mathrm{R}_{\mathtt{u_s}} = \mathrm{R}_{\mathtt{u_r}} = \{(w_1, w_2)\}$
- $\mathrm{R}_{\mathtt{u_e}} = \{(w_1, w_2), (w_1, w_3)\}$

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a\in\mathrm{Act}}, \{\mathrm{U}(i)\}_{i\in\mathrm{Agt}}, \mathrm{V}\rangle$:

- S is a non-empty set of states
- R_a a binary relation over S
- $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathrm{Prop})$ is a valuation function
- $\mathrm{U}(i) \subseteq \mathcal{P}(\mathrm{Act}^*)$ s.t. if $\pi_1, \pi_2 \in \mathrm{U}(i)$ and $\pi_1 \neq \pi_2$, then $\pi_1 \cap \pi_2 = \emptyset$



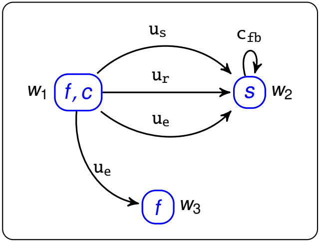
- $\mathrm{S} = \{w_1, w_2, w_3\}$
- $\mathrm{R}_{\mathtt{u_s}} = \mathrm{R}_{\mathtt{u_r}} = \{(w_1, w_2)\}$
- $\mathrm{R}_{\mathtt{u_e}} = \{(w_1, w_2), (w_1, w_3)\}$
- $\mathrm{R}_{\mathtt{c_{fb}}} = \{(w_2, w_2)\}$

Uncertainty-based LTSs (LTS^{U}s)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}, \mathrm{V} \rangle$:

- S is a non-empty set of states
- R_a a binary relation over S
- $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathsf{Prop})$ is a valuation function
- $\mathrm{U}(i) \subseteq \mathcal{P}(\mathsf{Act}^*)$ s.t. if $\pi_1, \pi_2 \in \mathrm{U}(i)$ and $\pi_1 \neq \pi_2$, then $\pi_1 \cap \pi_2 = \emptyset$



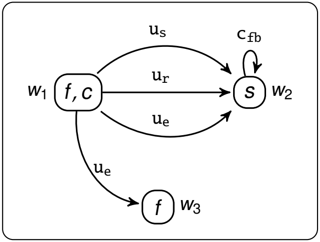
- $\mathrm{S} = \{w_1, w_2, w_3\}$
- $\mathrm{R}_{\mathtt{u_s}} = \mathrm{R}_{\mathtt{u_r}} = \{(w_1, w_2)\}$
- $\mathrm{R}_{\mathtt{u_e}} = \{(w_1, w_2), (w_1, w_3)\}$
- $\mathrm{R}_{\mathtt{c_{fb}}} = \{(w_2, w_2)\}$
- $\mathrm{V}(w_1) = \{f, c\}, \mathrm{V}(w_2) = \{s\}, \mathrm{V}(w_3) = \{f\}$

Uncertainty-based LTSs (LTS^{U}s)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}, \mathrm{V} \rangle$:

- S is a non-empty set of states
- R_a a binary relation over S
- $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathsf{Prop})$ is a valuation function
- $\mathrm{U}(i) \subseteq \mathcal{P}(\mathsf{Act}^*)$ s.t. if $\pi_1, \pi_2 \in \mathrm{U}(i)$ and $\pi_1 \neq \pi_2$, then $\pi_1 \cap \pi_2 = \emptyset$



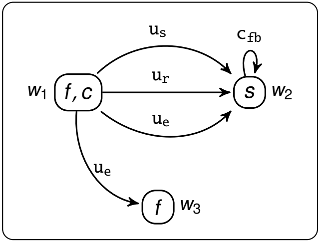
- $\mathrm{S} = \{w_1, w_2, w_3\}$
- $\mathrm{R}_{\mathtt{u_s}} = \mathrm{R}_{\mathtt{u_r}} = \{(w_1, w_2)\}$
- $\mathrm{R}_{\mathtt{u_e}} = \{(w_1, w_2), (w_1, w_3)\}$
- $\mathrm{R}_{\mathtt{c_{fb}}} = \{(w_2, w_2)\}$
- $\mathrm{V}(w_1) = \{f, c\}, \mathrm{V}(w_2) = \{s\}, \mathrm{V}(w_3) = \{f\}$
- $\mathrm{U}(i) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}, \{\mathtt{u_e c_{fb}}\}\}$

Uncertainty-based LTSs (LTS^Us)

Definition (LTS^U)

Given Act, Agt and Prop. $\mathcal{M} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a\in\mathsf{Act}}, \{\mathrm{U}(i)\}_{i\in\mathsf{Agt}}, \mathrm{V}\rangle$:

- S is a non-empty set of states
- R_a a binary relation over S
- $\mathrm{V} : \mathrm{S} \to \mathcal{P}(\mathsf{Prop})$ is a valuation function
- $\mathrm{U}(i) \subseteq \mathcal{P}(\mathsf{Act}^*)$ s.t. if $\pi_1, \pi_2 \in \mathrm{U}(i)$ and $\pi_1 \neq \pi_2$, then $\pi_1 \cap \pi_2 = \emptyset$



- $\mathrm{S} = \{w_1, w_2, w_3\}$
- $\mathrm{R}_{\mathtt{u_s}} = \mathrm{R}_{\mathtt{u_r}} = \{(w_1, w_2)\}$
- $\mathrm{R}_{\mathtt{u_e}} = \{(w_1, w_2), (w_1, w_3)\}$
- $\mathrm{R}_{\mathtt{c_{fb}}} = \{(w_2, w_2)\}$
- $\mathrm{V}(w_1) = \{f, c\}, \mathrm{V}(w_2) = \{s\}, \mathrm{V}(w_3) = \{f\}$
- $\mathrm{U}(i) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}, \{\mathtt{u_e c_{fb}}\}\}$
- $\mathrm{U}(j) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}\}$

Strong executability

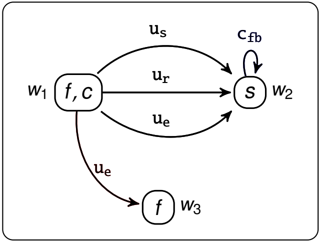
Fail-proof plans:

Strong executability

Fail-proof plans: Each partial execution has to be completed

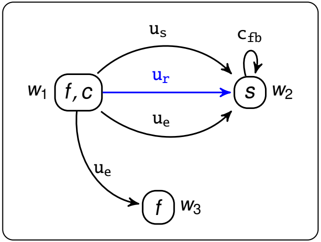
Strong executability

Fail-proof plans: Each partial execution has to be completed



Strong executability

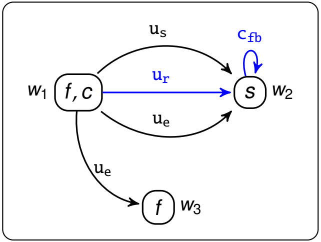
Fail-proof plans: Each partial execution has to be completed



$\mathtt{u_r}$ is strongly executable at w_1

Strong executability

Fail-proof plans: Each partial execution has to be completed

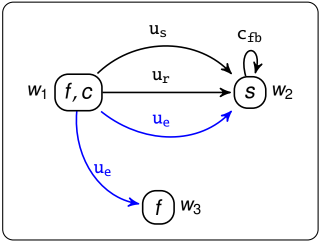


$\mathtt{u_r}$ is strongly executable at w_1

$\mathtt{u_r c_{fb}}$ is strongly executable at w_1

Strong executability

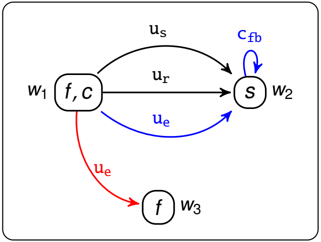
Fail-proof plans: Each partial execution has to be completed



$\mathtt{u_r}$ is strongly executable at w_1
$\mathtt{u_r c_{fb}}$ is strongly executable at w_1
$\mathtt{u_e}$ is strongly executable at w_1

Strong executability

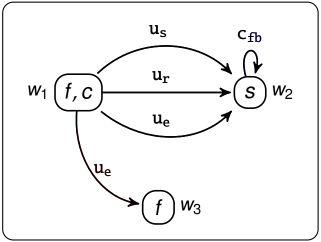
Fail-proof plans: Each partial execution has to be completed



$\mathtt{u_r}$ is strongly executable at w_1
$\mathtt{u_r c_{fb}}$ is strongly executable at w_1
$\mathtt{u_e}$ is strongly executable at w_1
$\mathtt{u_e c_{fb}}$ is not strongly executable at w_1

Strong executability

Fail-proof plans: Each partial execution has to be completed



$\mathtt{u_r}$ is strongly executable at w_1
$\mathtt{u_r c_{fb}}$ is strongly executable at w_1
$\mathtt{u_e}$ is strongly executable at w_1
$\mathtt{u_e c_{fb}}$ is not strongly executable at w_1

Definition (Strong executability for a plan)

$\sigma = b_1 \ldots b_n \in \text{Act}^*$ is *strongly executable* (SE) at $u \in \mathrm{S}$ iff, for each $k = 1, \ldots, n-1$,

$$v \in \mathrm{R}_{b_1 \ldots b_k}(u) \quad \text{implies} \quad \mathrm{R}_{b_{k+1}}(v) \neq \emptyset.$$

Definition

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{Kh}_i(\varphi,\varphi)$$

$\mathsf{Kh}_i(\psi,\varphi)$: *"when ψ holds, agent i knows how to reach φ"*.

Syntax and semantics of $\mathsf{L}_{\mathsf{Kh}_i}$ over LTS^Us

Definition

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{Kh}_i(\varphi, \varphi)$$

$\mathsf{Kh}_i(\psi, \varphi)$: *"when ψ holds, agent i knows how to reach φ"*.

$\mathcal{M}, w \models p$ iff $p \in \mathrm{V}(w)$

Definition

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{Kh}_i(\varphi, \varphi)$$

$\mathsf{Kh}_i(\psi, \varphi)$: *"when ψ holds, agent i knows how to reach φ".*

$\mathcal{M}, w \models p$	iff	$p \in \mathrm{V}(w)$
$\mathcal{M}, w \models \neg\varphi$	iff	$\mathcal{M}, w \not\models \varphi$

Syntax and semantics of $\mathsf{L}_{\mathsf{Kh}_i}$ over LTS^Us

Definition

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{Kh}_i(\varphi, \varphi)$$

$\mathsf{Kh}_i(\psi, \varphi)$: *"when ψ holds, agent i knows how to reach φ"*.

$\mathcal{M}, w \models p$	iff	$p \in \mathrm{V}(w)$
$\mathcal{M}, w \models \neg\varphi$	iff	$\mathcal{M}, w \not\models \varphi$
$\mathcal{M}, w \models \varphi \vee \psi$	iff	$\mathcal{M}, w \models \varphi$ or $\mathcal{M}, w \models \psi$

Definition

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{Kh}_i(\varphi, \varphi)$$

$\mathsf{Kh}_i(\psi, \varphi)$: *"when ψ holds, agent i knows how to reach φ"*.

$\mathcal{M}, w \models p$ iff $p \in \mathrm{V}(w)$

$\mathcal{M}, w \models \neg\varphi$ iff $\mathcal{M}, w \not\models \varphi$

$\mathcal{M}, w \models \varphi \vee \psi$ iff $\mathcal{M}, w \models \varphi$ or $\mathcal{M}, w \models \psi$

$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ iff there exists $\pi \in \mathrm{U}(i)$ s.t. for each $\sigma \in \pi$

Definition

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{Kh}_i(\varphi, \varphi)$$

$\mathsf{Kh}_i(\psi, \varphi)$: *“when ψ holds, agent i knows how to reach φ”*.

$\mathcal{M}, w \models p$ iff $p \in \mathrm{V}(w)$

$\mathcal{M}, w \models \neg\varphi$ iff $\mathcal{M}, w \not\models \varphi$

$\mathcal{M}, w \models \varphi \vee \psi$ iff $\mathcal{M}, w \models \varphi$ or $\mathcal{M}, w \models \psi$

$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ iff there exists $\pi \in \mathrm{U}(i)$ s.t. for each $\sigma \in \pi$

1. is SE at all ψ-states

Definition

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{Kh}_i(\varphi, \varphi)$$

$\mathsf{Kh}_i(\psi, \varphi)$: *"when ψ holds, agent i knows how to reach φ"*.

$\mathcal{M}, w \models p$ iff $p \in \mathrm{V}(w)$

$\mathcal{M}, w \models \neg\varphi$ iff $\mathcal{M}, w \not\models \varphi$

$\mathcal{M}, w \models \varphi \vee \psi$ iff $\mathcal{M}, w \models \varphi$ or $\mathcal{M}, w \models \psi$

$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ iff there exists $\pi \in \mathrm{U}(i)$ s.t. for each $\sigma \in \pi$

1. is SE at all ψ-states and
2. its executions from ψ-states always end in φ-states

Kh_i over LTS^Us

$\mathsf{Kh}_i(\psi, \varphi)$: *"when ψ holds, agent i knows how to reach φ"*.
$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ iff there exists $\pi \in \mathrm{U}(i)$ s.t. for each $\sigma \in \pi$

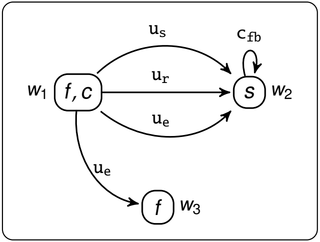
1. is SE at all ψ-states and
2. its executions from ψ-states always end in φ-states

$\mathrm{U}(i) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}, \{\mathtt{u_e c_{fb}}\}\}$
$\mathrm{U}(j) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}\}$

Kh$_i(\psi, \varphi)$: *“when ψ holds, agent i knows how to reach φ”.*
$\mathcal{M}, w \models \mathrm{Kh}_i(\psi, \varphi)$ iff there exists $\pi \in \mathrm{U}(i)$ s.t. for each $\sigma \in \pi$

1. is SE at all ψ-states and
2. its executions from ψ-states always end in φ-states

$\mathrm{U}(i) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}, \{\mathtt{u_e c_{fb}}\}\}$
$\mathrm{U}(j) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}\}$

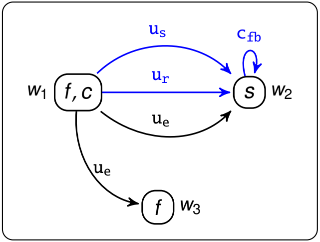


$\mathrm{Kh}_i(\psi, \varphi)$: *"when ψ holds, agent i knows how to reach φ".*

$\mathcal{M}, w \models \mathrm{Kh}_i(\psi, \varphi)$ iff there exists $\pi \in \mathrm{U}(i)$ s.t. for each $\sigma \in \pi$

1. is SE at all ψ-states and
2. its executions from ψ-states always end in φ-states

$\mathrm{U}(i) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}, \{\mathtt{u_e c_{fb}}\}\}$
$\mathrm{U}(j) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}\}$



$\models \mathrm{Kh}_i(f \wedge c, s)$

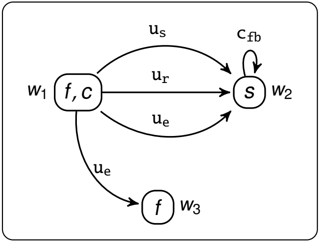
Kh_i over LTS^Us

$\mathsf{Kh}_i(\psi, \varphi)$: *"when ψ holds, agent i knows how to reach φ"*.

$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ iff there exists $\pi \in \mathrm{U}(i)$ s.t. for each $\sigma \in \pi$

1. is SE at all ψ-states and
2. its executions from ψ-states always end in φ-states

$\mathrm{U}(i) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}, \{\mathtt{u_e c_{fb}}\}\}$
$\mathrm{U}(j) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}\}$



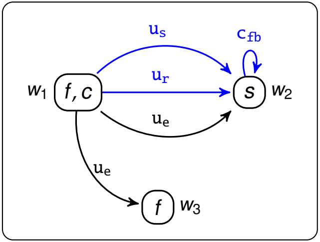
$\models \mathsf{Kh}_i(f \wedge c, s)$
$\not\models \mathsf{Kh}_j(f \wedge c, s)$

Kh_i over LTS^Us

$\mathsf{Kh}_i(\psi, \varphi)$: *"when ψ holds, agent i knows how to reach φ"*.
$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ iff there exists $\pi \in \mathrm{U}(i)$ s.t. for each $\sigma \in \pi$

1. is SE at all ψ-states and
2. its executions from ψ-states always end in φ-states

$\mathrm{U}(i) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}, \{\mathtt{u_e c_{fb}}\}\}$
$\mathrm{U}(j) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}\}$

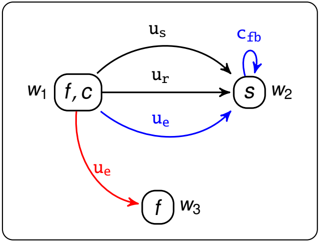


$\models \mathsf{Kh}_i(f \wedge c, s)$
$\not\models \mathsf{Kh}_j(f \wedge c, s)$
- $\mathtt{u_s c_{fb}}$ and $\mathtt{u_r c_{fb}}$ take the agent from all $(f \wedge c)$-states and reaches only s-states;

$\mathsf{Kh}_i(\psi,\varphi)$: *"when ψ holds, agent i knows how to reach φ".*
$\mathcal{M}, w \models \mathsf{Kh}_i(\psi,\varphi)$ iff there exists $\pi \in \mathrm{U}(i)$ s.t. for each $\sigma \in \pi$

1. is SE at all ψ-states and
2. its executions from ψ-states always end in φ-states

$\mathrm{U}(i) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}, \{\mathtt{u_e c_{fb}}\}\}$
$\mathrm{U}(j) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}\}$

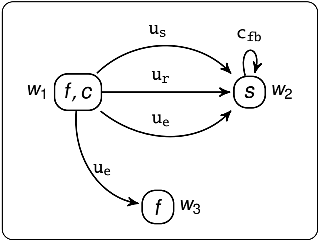


$\models \mathsf{Kh}_i(f \wedge c, s)$
$\not\models \mathsf{Kh}_j(f \wedge c, s)$
- $\mathtt{u_s c_{fb}}$ and $\mathtt{u_r c_{fb}}$ take the agent from all $(f \wedge c)$-states and reaches only s-states;
- plan $\mathtt{u_e c_{fb}}$ does not complete at w_3;

$\text{Kh}_i(\psi, \varphi)$: *"when ψ holds, agent i knows how to reach φ"*.
$\mathcal{M}, w \models \text{Kh}_i(\psi, \varphi)$ iff there exists $\pi \in \text{U}(i)$ s.t. for each $\sigma \in \pi$

1. is SE at all ψ-states and
2. its executions from ψ-states always end in φ-states

$\text{U}(i) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}\}, \{\mathtt{u_e c_{fb}}\}\}$
$\text{U}(j) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}\}$



$\models \text{Kh}_i(f \wedge c, s)$
$\not\models \text{Kh}_j(f \wedge c, s)$
- $\mathtt{u_s c_{fb}}$ and $\mathtt{u_r c_{fb}}$ take the agent from all $(f \wedge c)$-states and reaches only s-states;
- plan $\mathtt{u_e c_{fb}}$ does not complete at w_3;
$\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}$ is not SE at w_1

Universal modality A

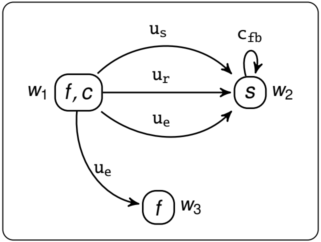
$\mathsf{A}\varphi := \bigvee_{i \in \mathsf{Agt}} \mathsf{Kh}_i(\neg\varphi, \bot)$: *"at all states φ holds"*

$\mathcal{M}, w \models \mathsf{A}\varphi$ iff for all $u \in \mathrm{S}$, $\mathcal{M}, u \models \varphi$

Universal modality A

$$\mathsf{A}\varphi := \bigvee_{i\in\mathsf{Agt}} \mathsf{Kh}_i(\neg\varphi, \bot)\text{: \textit{“at all states } }\varphi\text{\textit{ holds”}}$$

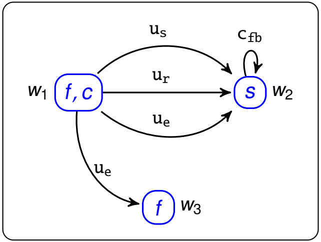
$$\mathcal{M}, w \models \mathsf{A}\varphi \text{ iff for all } u \in \mathrm{S},\ \mathcal{M}, u \models \varphi$$



Universal modality A

$$\mathsf{A}\varphi := \bigvee_{i \in \mathsf{Agt}} \mathsf{Kh}_i(\neg\varphi, \bot)\text{: “\textit{at all states} } \varphi \text{ \textit{holds}”}$$

$$\mathcal{M}, w \models \mathsf{A}\varphi \text{ iff for all } u \in \mathrm{S},\ \mathcal{M}, u \models \varphi$$

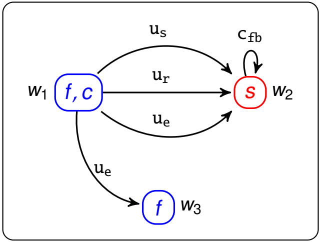


$\models \mathsf{A}(f \vee s)$

Universal modality A

$\mathsf{A}\varphi := \bigvee_{i \in \mathsf{Agt}} \mathsf{Kh}_i(\neg\varphi, \bot)$: *"at all states φ holds"*

$\mathcal{M}, w \models \mathsf{A}\varphi$ iff for all $u \in \mathsf{S}$, $\mathcal{M}, u \models \varphi$



$\models \mathsf{A}(f \vee s)$

$\not\models \mathsf{A}(f \vee c)$

Axiomatization $\mathcal{L}_{\mathsf{Kh}_i}^{\mathrm{LTS}^U}$ de $\mathsf{L}_{\mathsf{Kh}_i}$

$\underline{\mathcal{L}}$:	TAUT	$\vdash \varphi$ for propositional tautologies
	DISTA	$\vdash \mathsf{A}(\varphi \to \psi) \to (\mathsf{A}\varphi \to \mathsf{A}\psi)$
	TA	$\vdash \mathsf{A}\varphi \to \varphi$
	4KhA	$\vdash \mathsf{Kh}_i(\psi, \varphi) \to \mathsf{A}\mathsf{Kh}_i(\psi, \varphi)$
	5KhA	$\vdash \neg\mathsf{Kh}_i(\psi, \varphi) \to \mathsf{A}\neg\mathsf{Kh}_i(\psi, \varphi)$
$\underline{\mathcal{L}_{\mathrm{LTS}^U}}$:	KhA	$\vdash (\mathsf{A}(\chi \to \psi) \land \mathsf{Kh}_i(\psi, \varphi) \land \mathsf{A}(\varphi \to \theta)) \to \mathsf{Kh}_i(\chi, \theta)$
<u>Rules:</u>		$\dfrac{\varphi \quad (\varphi \to \psi)}{\psi}$ MP $\qquad \dfrac{\varphi}{\mathsf{A}\varphi}$ NECA

Representation benefits

- Improvement from a previous proposal ($\mathsf{L_{Kh}}$) that used LTSs without uncertainty ($\mathcal{L} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \mathrm{V} \rangle$)

  $\mathcal{L}, w \models \mathsf{Kh}(\psi, \varphi)$ iff there exists $\sigma \in \mathsf{Act}^*$ s.t.

  1. is SE at all ψ-states and
  2. its executions from ψ-states always end in φ-states

Reasons for not knowing how that are not represented.

Representation benefits

- Improvement from a previous proposal ($\mathsf{L_{Kh}}$) that used LTSs without uncertainty ($\mathcal{L} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \mathrm{V} \rangle$)

  $\mathcal{L}, w \models \mathsf{Kh}(\psi, \varphi)$ iff there exists $\sigma \in \mathsf{Act}^*$ s.t.

  1. is SE at all ψ-states and
  2. its executions from ψ-states always end in φ-states

Reasons for not knowing how that are not represented.

$\{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}$ solved two major issues:

Representation benefits

- Improvement from a previous proposal ($\mathsf{L_{Kh}}$) that used LTSs without uncertainty ($\mathcal{L} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \mathrm{V} \rangle$)

  $\mathcal{L}, w \models \mathsf{Kh}(\psi, \varphi)$ iff there exists $\sigma \in \mathsf{Act}^*$ s.t.

  1. is SE at all ψ-states and
  2. its executions from ψ-states always end in φ-states

Reasons for not knowing how that are not represented.

$\{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}$ solved two major issues:

- Issue 1: The agent had all the plans to choose one

Representation benefits

- Improvement from a previous proposal ($\mathsf{L_{Kh}}$) that used LTSs without uncertainty ($\mathcal{L} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \mathrm{V} \rangle$)

  $\mathcal{L}, w \models \mathsf{Kh}(\psi, \varphi)$ iff there exists $\sigma \in \mathsf{Act}^*$ s.t.

  1. is SE at all ψ-states and
  2. its executions from ψ-states always end in φ-states

Reasons for not knowing how that are not represented.

$\{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}$ solved two major issues:

- Issue 1: The agent had all the plans to choose one

- Issue 2: The agent distinguishes each plan from all the others

Representation benefits

- Improvement from a previous proposal ($\mathsf{L_{Kh}}$) that used LTSs without uncertainty ($\mathcal{L} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \mathrm{V} \rangle$)

  $\mathcal{L}, w \models \mathsf{Kh}(\psi, \varphi)$ iff there exists $\sigma \in \mathsf{Act}^*$ s.t.

  1. is SE at all ψ-states and
  2. its executions from ψ-states always end in φ-states

Reasons for not knowing how that are not represented.

$\{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}$ solved two major issues:

- Issue 1: The agent had all the plans to choose one
  - The agent migth not be conscious that certain plans exist

- Issue 2: The agent distinguishes each plan from all the others

Representation benefits

- Improvement from a previous proposal ($\mathsf{L_{Kh}}$) that used LTSs without uncertainty ($\mathcal{L} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \mathrm{V} \rangle$)

  $\mathcal{L}, w \models \mathsf{Kh}(\psi, \varphi)$ iff there exists $\sigma \in \mathsf{Act}^*$ s.t.

  1. is SE at all ψ-states and
  2. its executions from ψ-states always end in φ-states

Reasons for not knowing how that are not represented.

$\{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}$ solved two major issues:

- Issue 1: The agent had all the plans to choose one
  - The agent migth not be conscious that certain plans exist
  - being able of $\neq$ being conscious of being able ($\bigcup_{\pi \in \mathrm{U}(i)} \pi \subsetneq \mathsf{Act}^*$)
- Issue 2: The agent distinguishes each plan from all the others

Representation benefits

- Improvement from a previous proposal ($\mathsf{L_{Kh}}$) that used LTSs without uncertainty ($\mathcal{L} = \langle \mathrm{S}, \{\mathrm{R}_a\}_{a \in \mathsf{Act}}, \mathrm{V} \rangle$)

  $\mathcal{L}, w \models \mathsf{Kh}(\psi, \varphi)$ iff there exists $\sigma \in \mathsf{Act}^*$ s.t.

  1. is SE at all ψ-states and
  2. its executions from ψ-states always end in φ-states

Reasons for not knowing how that are not represented.

$\{\mathrm{U}(i)\}_{i \in \mathsf{Agt}}$ solved two major issues:

- Issue 1: The agent had all the plans to choose one
  - The agent migth not be conscious that certain plans exist
  - being able of $\neq$ being conscious of being able ($\bigcup_{\pi \in \mathrm{U}(i)} \pi \subsetneq \mathsf{Act}^*$)
- Issue 2: The agent distinguishes each plan from all the others
  - For the agent the effects of two plans can be indistinct ($\mathrm{U}(j) = \{\{\mathtt{u_s c_{fb}}, \mathtt{u_r c_{fb}}, \mathtt{u_e c_{fb}}\}\}$)

General Benefits

- Multiple agents can share the same LTS

General Benefits

- Multiple agents can share the same LTS
- Model more imperfect and less idealized agents

General Benefits

- Multiple agents can share the same LTS
- Model more imperfect and less idealized agents
- $\mathsf{L}_{\mathsf{Kh}_i}$ generalizes L_{Kh}

General Benefits

- Multiple agents can share the same LTS
- Model more imperfect and less idealized agents
- $\mathsf{L}_{\mathsf{Kh}_i}$ generalizes L_{Kh}

  every LTS has an equivalent LTS^U with $\mathrm{U}(i) := \{\{\sigma\} \mid \sigma \in \mathsf{Act}^*\}$

General Benefits

- Multiple agents can share the same LTS
- Model more imperfect and less idealized agents
- $\mathsf{L}_{\mathsf{Kh}_i}$ generalizes L_{Kh}

  every LTS has an equivalent LTS^U with $\mathrm{U}(i) := \{\{\sigma\} \mid \sigma \in \mathsf{Act}^*\}$
- Lower computational complexity

General Benefits

- Multiple agents can share the same LTS
- Model more imperfect and less idealized agents
- $\mathsf{L}_{\mathsf{Kh}_i}$ generalizes L_{Kh}

  every LTS has an equivalent LTS^U with $\mathrm{U}(i) := \{\{\sigma\} \mid \sigma \in \mathsf{Act}^*\}$
- Lower computational complexity
  - Model checking: P (vs PSpace for L_{Kh})

General Benefits

- Multiple agents can share the same LTS
- Model more imperfect and less idealized agents
- $\mathsf{L_{Kh_i}}$ generalizes $\mathsf{L_{Kh}}$

  every LTS has an equivalent LTS^U with $\mathrm{U}(i) := \{\{\sigma\} \mid \sigma \in \mathsf{Act}^*\}$
- Lower computational complexity
  - Model checking: P (vs PSpace for $\mathsf{L_{Kh}}$)
  - SAT: NP-complete (vs $\mathrm{NP}^{\mathrm{NP}}$ upperbound for $\mathsf{L_{Kh}}$)

General Benefits

- Multiple agents can share the same LTS
- Model more imperfect and less idealized agents
- $\mathsf{L}_{\mathsf{Kh}_i}$ generalizes L_{Kh}

  every LTS has an equivalent LTS^U with $\mathrm{U}(i) := \{\{\sigma\} \mid \sigma \in \mathsf{Act}^*\}$
- Lower computational complexity
  - Model checking: P (vs PSpace for L_{Kh})
  - SAT: NP-complete (vs $\mathrm{NP}^{\mathrm{NP}}$ upperbound for L_{Kh})
- A clear distinction between
- ontic information, available abilities (LTS part),

General Benefits

- Multiple agents can share the same LTS
- Model more imperfect and less idealized agents
- $\mathsf{L}_{\mathsf{Kh}_i}$ generalizes L_{Kh}

  every LTS has an equivalent LTS^U with $\mathrm{U}(i) := \{\{\sigma\} \mid \sigma \in \mathsf{Act}^*\}$
- Lower computational complexity
  - Model checking: P (vs PSpace for L_{Kh})
  - SAT: NP-complete (vs $\mathrm{NP}^{\mathrm{NP}}$ upperbound for L_{Kh})
- A clear distinction between
  - ontic information, available abilities (LTS part), and
  - epistemic information, perceived-as-possible abilities (U)

General Benefits

- Multiple agents can share the same LTS
- Model more imperfect and less idealized agents
- $\mathsf{L}_{\mathsf{Kh}_i}$ generalizes L_{Kh}

  every LTS has an equivalent LTS^U with $\mathrm{U}(i) := \{\{\sigma\} \mid \sigma \in \mathsf{Act}^*\}$
- Lower computational complexity
  - Model checking: P (vs PSpace for L_{Kh})
  - SAT: NP-complete (vs $\mathsf{NP}^{\mathsf{NP}}$ upperbound for L_{Kh})
- A clear distinction between
  - ontic information, available abilities (LTS part), and
  - epistemic information, perceived-as-possible abilities (U)
- We can define ontic and epistemic dynamic operators that modifiy each type of information

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

- $[!\chi]\varphi$: *"after announcing χ, φ holds"*

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

- $[!\chi]\varphi$: *"after announcing χ, φ holds"*
- Deletes states ($\mathbb{S}$) and the relations between these ($\mathbb{R}$)

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

- $[!\chi]\varphi$: *"after announcing χ, φ holds"*
- Deletes states (S) and the relations between these (R)
- More expressive than $\mathsf{L}_{\mathsf{Kh}_i}$ over arbitrary LTS^Us

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

- $[!\chi]\varphi$: *"after announcing χ, φ holds"*
- Deletes states (S) and the relations between these (R)
- More expressive than $\mathsf{L}_{\mathsf{Kh}_i}$ over arbitrary LTS^Us
- SAT problem is decidable only over a restricted class of models

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

- $[!\chi]\varphi$: *"after announcing χ, φ holds"*
- Deletes states ($\mathbb{S}$) and the relations between these ($\mathbb{R}$)
- More expressive than $\mathsf{L}_{\mathsf{Kh}_i}$ over arbitrary LTS^Us
- SAT problem is decidable only over a restricted class of models

Epistemic: Refinement ($\mathsf{L}_{\mathsf{Ref}} = \mathsf{L}_{\mathsf{Kh}_i} + \langle \sigma_1 \nsim \sigma_2 \rangle$)

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

- $[!\chi]\varphi$: *"after announcing χ, φ holds"*
- Deletes states (S) and the relations between these (R)
- More expressive than $\mathsf{L}_{\mathsf{Kh}_i}$ over arbitrary LTS^{U}s
- SAT problem is decidable only over a restricted class of models

Epistemic: Refinement ($\mathsf{L}_{\mathsf{Ref}} = \mathsf{L}_{\mathsf{Kh}_i} + \langle \sigma_1 \nsim \sigma_2 \rangle$)

- Given a set of plans, divide it in two to be distinguishable

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

- $[!\chi]\varphi$: *"after announcing χ, φ holds"*
- Deletes states ($\mathbb{S}$) and the relations between these ($\mathbb{R}$)
- More expressive than $\mathsf{L}_{\mathsf{Kh}_i}$ over arbitrary LTS^Us
- SAT problem is decidable only over a restricted class of models

Epistemic: Refinement ($\mathsf{L}_{\mathsf{Ref}} = \mathsf{L}_{\mathsf{Kh}_i} + \langle\sigma_1 \nsim \sigma_2\rangle$)

- Given a set of plans, divide it in two to be distinguishable
- $\langle\sigma_1 \nsim \sigma_2\rangle\varphi$: *"there exists a way of separating σ_1 y σ_2 s.t. φ holds"*

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

- $[!\chi]\varphi$: *"after announcing χ, φ holds"*
- Deletes states ($\mathbb{S}$) and the relations between these ($\mathbb{R}$)
- More expressive than $\mathsf{L}_{\mathsf{Kh}_i}$ over arbitrary LTS^Us
- SAT problem is decidable only over a restricted class of models

Epistemic: Refinement ($\mathsf{L}_{\mathsf{Ref}} = \mathsf{L}_{\mathsf{Kh}_i} + \langle\sigma_1 \nsim \sigma_2\rangle$)

- Given a set of plans, divide it in two to be distinguishable
- $\langle\sigma_1 \nsim \sigma_2\rangle\varphi$: *"there exists a way of separating σ_1 y σ_2 s.t. φ holds"*
- $[\sigma_1 \nsim \sigma_2]\varphi$: *"for each way of separating σ_1 and σ_2, φ holds"*

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

- $[!\chi]\varphi$: *"after announcing χ, φ holds"*
- Deletes states ($\mathbb{S}$) and the relations between these ($\mathbb{R}$)
- More expressive than $\mathsf{L}_{\mathsf{Kh}_i}$ over arbitrary LTS^Us
- SAT problem is decidable only over a restricted class of models

Epistemic: Refinement ($\mathsf{L}_{\mathsf{Ref}} = \mathsf{L}_{\mathsf{Kh}_i} + \langle\sigma_1 \nsim \sigma_2\rangle$)

- Given a set of plans, divide it in two to be distinguishable
- $\langle\sigma_1 \nsim \sigma_2\rangle\varphi$: *"there exists a way of separating σ_1 y σ_2 s.t. φ holds"*
- $[\sigma_1 \nsim \sigma_2]\varphi$: *"for each way of separating σ_1 and σ_2, φ holds"*
- $\not\models \mathsf{Kh}_j(f \wedge c, s)$

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

- $[!\chi]\varphi$: *"after announcing χ, φ holds"*
- Deletes states ($\mathbb{S}$) and the relations between these ($\mathbb{R}$)
- More expressive than $\mathsf{L}_{\mathsf{Kh}_i}$ over arbitrary LTS^Us
- SAT problem is decidable only over a restricted class of models

Epistemic: Refinement ($\mathsf{L}_{\mathsf{Ref}} = \mathsf{L}_{\mathsf{Kh}_i} + \langle \sigma_1 \nsim \sigma_2 \rangle$)

- Given a set of plans, divide it in two to be distinguishable
- $\langle \sigma_1 \nsim \sigma_2 \rangle\varphi$: *"there exists a way of separating σ_1 y σ_2 s.t. φ holds"*
- $[\sigma_1 \nsim \sigma_2]\varphi$: *"for each way of separating σ_1 and σ_2, φ holds"*
- $\not\models \mathsf{Kh}_j(f \wedge c, s) \models \langle \mathtt{u_s c_{fb}} \nsim \mathtt{u_e c_{fb}} \rangle \mathsf{Kh}_j(f \wedge c, s)$

Dynamic operators

Ontic: Public Announcement ($\mathsf{PAL}_{\mathsf{Kh}_i} = \mathsf{L}_{\mathsf{Kh}_i} + [!\chi]$)

- $[!\chi]\varphi$: *"after announcing χ, φ holds"*
- Deletes states (S) and the relations between these (R)
- More expressive than $\mathsf{L}_{\mathsf{Kh}_i}$ over arbitrary LTS^Us
- SAT problem is decidable only over a restricted class of models

Epistemic: Refinement ($\mathsf{L}_{\mathsf{Ref}} = \mathsf{L}_{\mathsf{Kh}_i} + \langle \sigma_1 \nsim \sigma_2 \rangle$)

- Given a set of plans, divide it in two to be distinguishable
- $\langle \sigma_1 \nsim \sigma_2 \rangle\varphi$: *"there exists a way of separating σ_1 y σ_2 s.t. φ holds"*
- $[\sigma_1 \nsim \sigma_2]\varphi$: *"for each way of separating σ_1 and σ_2, φ holds"*
- $\not\models \mathsf{Kh}_j(f \wedge c, s) \models \langle \mathtt{u_s c_{fb}} \nsim \mathtt{u_e c_{fb}} \rangle \mathsf{Kh}_j(f \wedge c, s)$
- More expressive than $\mathsf{L}_{\mathsf{Kh}_i}$ over arbitrary LTS^Us

Issues with these operators

- Ontic updates: limited axiomatizations via reductions axioms

Issues with these operators

- Ontic updates: limited axiomatizations via reductions axioms
- Epistemic updates: fails uniform substitution, no axiomatizations

Issues with these operators

- Ontic updates: limited axiomatizations via reductions axioms
- Epistemic updates: fails uniform substitution, no axiomatizations
- $\mathsf{L}_{\mathsf{Kh}_i}$ can not capture the effect of these modalities as Kh_i describe plans implicitly

Issues with these operators

- Ontic updates: limited axiomatizations via reductions axioms
- Epistemic updates: fails uniform substitution, no axiomatizations
- $\mathsf{L}_{\mathsf{Kh}_i}$ can not capture the effect of these modalities as Kh_i describe plans implicitly
- Extend the static language with basic modalities ($[b]$, $b \in \mathsf{Act}$) for executions ($\mathsf{L}_{\mathsf{Kh}_i,\Box} = \mathsf{L}_{\mathsf{Kh}_i} + [b]$)

Issues with these operators

- Ontic updates: limited axiomatizations via reductions axioms
- Epistemic updates: fails uniform substitution, no axiomatizations
- $\mathsf{L}_{\mathsf{Kh}_i}$ can not capture the effect of these modalities as Kh_i describe plans implicitly
- Extend the static language with basic modalities ($[b]$, $b \in \mathsf{Act}$) for executions ($\mathsf{L}_{\mathsf{Kh}_i,\Box} = \mathsf{L}_{\mathsf{Kh}_i} + [b]$)
  - $[b]\varphi$: *"each execution of b reaches φ-states"*

Issues with these operators

- Ontic updates: limited axiomatizations via reductions axioms
- Epistemic updates: fails uniform substitution, no axiomatizations
- $\mathsf{L}_{\mathsf{Kh}_i}$ can not capture the effect of these modalities as Kh_i describe plans implicitly
- Extend the static language with basic modalities ($[b]$, $b \in \mathsf{Act}$) for executions ($\mathsf{L}_{\mathsf{Kh}_i,\Box} = \mathsf{L}_{\mathsf{Kh}_i} + [b]$)
  - $[b]\varphi$: *"each execution of b reaches φ-states"*
  - $\langle b\rangle\varphi$: *"there is an execution of b that reaches a φ-state"*

Issues with these operators

- Ontic updates: limited axiomatizations via reductions axioms
- Epistemic updates: fails uniform substitution, no axiomatizations
- $\mathsf{L}_{\mathsf{Kh}_i}$ can not capture the effect of these modalities as Kh_i describe plans implicitly
- Extend the static language with basic modalities ($[b]$, $b \in \mathsf{Act}$) for executions ($\mathsf{L}_{\mathsf{Kh}_i,\Box} = \mathsf{L}_{\mathsf{Kh}_i} + [b]$)
  - $[b]\varphi$: *"each execution of b reaches φ-states"*
  - $\langle b\rangle\varphi$: *"there is an execution of b that reaches a φ-state"*
- Define a dynamic modality ($[!b]$) that distinguishes an action from others ($\mathsf{L}_{\mathsf{Kh}_i,\Box,[!b]} = \mathsf{L}_{\mathsf{Kh}_i,\Box} + [!b]$)

Issues with these operators

- Ontic updates: limited axiomatizations via reductions axioms
- Epistemic updates: fails uniform substitution, no axiomatizations
- $\mathsf{L}_{\mathsf{Kh}_i}$ can not capture the effect of these modalities as Kh_i describe plans implicitly
- Extend the static language with basic modalities ($[b]$, $b \in \mathsf{Act}$) for executions ($\mathsf{L}_{\mathsf{Kh}_i,\Box} = \mathsf{L}_{\mathsf{Kh}_i} + [b]$)
  - $[b]\varphi$: *"each execution of b reaches φ-states"*
  - $\langle b\rangle\varphi$: *"there is an execution of b that reaches a φ-state"*
- Define a dynamic modality ($[!b]$) that distinguishes an action from others ($\mathsf{L}_{\mathsf{Kh}_i,\Box,[!b]} = \mathsf{L}_{\mathsf{Kh}_i,\Box} + [!b]$)
  - $[!b]\varphi$: *"after announcing b is distinguishable, φ holds".*

Issues with these operators

- Ontic updates: limited axiomatizations via reductions axioms
- Epistemic updates: fails uniform substitution, no axiomatizations
- $\mathsf{L}_{\mathsf{Kh}_i}$ can not capture the effect of these modalities as Kh_i describe plans implicitly
- Extend the static language with basic modalities ($[b]$, $b \in \mathsf{Act}$) for executions ($\mathsf{L}_{\mathsf{Kh}_i,\Box} = \mathsf{L}_{\mathsf{Kh}_i} + [b]$)
  - $[b]\varphi$: *"each execution of b reaches φ-states"*
  - $\langle b\rangle\varphi$: *"there is an execution of b that reaches a φ-state"*
- Define a dynamic modality ($[!b]$) that distinguishes an action from others ($\mathsf{L}_{\mathsf{Kh}_i,\Box,[!b]} = \mathsf{L}_{\mathsf{Kh}_i,\Box} + [!b]$)
  - $[!b]\varphi$: *"after announcing b is distinguishable, φ holds".*
- Sound and complete axiomatization via reduction axioms

Issues with these operators

- Ontic updates: limited axiomatizations via reductions axioms
- Epistemic updates: fails uniform substitution, no axiomatizations
- $\mathsf{L}_{\mathsf{Kh}_i}$ can not capture the effect of these modalities as Kh_i describe plans implicitly
- Extend the static language with basic modalities ($[b]$, $b \in \mathsf{Act}$) for executions ($\mathsf{L}_{\mathsf{Kh}_i,\Box} = \mathsf{L}_{\mathsf{Kh}_i} + [b]$)
  - $[b]\varphi$: *"each execution of b reaches φ-states"*
  - $\langle b\rangle\varphi$: *"there is an execution of b that reaches a φ-state"*
- Define a dynamic modality ($[!b]$) that distinguishes an action from others ($\mathsf{L}_{\mathsf{Kh}_i,\Box,[!b]} = \mathsf{L}_{\mathsf{Kh}_i,\Box} + [!b]$)
  - $[!b]\varphi$: *"after announcing b is distinguishable, φ holds".*
- Sound and complete axiomatization via reduction axioms
- $\mathsf{L}_{\mathsf{Kh}_i,\Box}$ and $\mathsf{L}_{\mathsf{Kh}_i,\Box,[!b]}$ are equally expressive and decidable.

In short...

$\mathsf{L}_{\mathsf{Kh}_i}$ logic [Ch. 4]

In short...

L_{Kh_i} logic [Ch. 4]

- Distiction between ontic and epistemic information

In short...

L_{Kh_i} logic [Ch. 4]

- Distiction between ontic and epistemic information
- Represents more reasons of knowing how

In short...

$\mathsf{L}_{\mathsf{Kh}_i}$ logic [Ch. 4]

- Distiction between ontic and epistemic information
- Represents more reasons of knowing how
- Sound and complete axiomatization [Ch. 5]

In short...

$\mathsf{L}_{\mathsf{Kh}_i}$ logic [Ch. 4]

- Distiction between ontic and epistemic information
- Represents more reasons of knowing how
- Sound and complete axiomatization [Ch. 5]
- Bisimulations results and formula equivalence [Ch. 6]

In short...

L_{Kh_i} logic [Ch. 4]

- Distiction between ontic and epistemic information
- Represents more reasons of knowing how
- Sound and complete axiomatization [Ch. 5]
- Bisimulations results and formula equivalence [Ch. 6]
- Generalizes the semantics based on LTSs [Ch. 6]

In short...

L_{Kh_i} logic [Ch. 4]

- Distiction between ontic and epistemic information
- Represents more reasons of knowing how
- Sound and complete axiomatization [Ch. 5]
- Bisimulations results and formula equivalence [Ch. 6]
- Generalizes the semantics based on LTSs [Ch. 6]
- Lower computational complexity [Ch. 7]

In short...

$\mathsf{L}_{\mathsf{Kh}_i}$ logic [Ch. 4]

- Distiction between ontic and epistemic information
- Represents more reasons of knowing how
- Sound and complete axiomatization [Ch. 5]
- Bisimulations results and formula equivalence [Ch. 6]
- Generalizes the semantics based on LTSs [Ch. 6]
- Lower computational complexity [Ch. 7]

Dynamic operators [Ch. 8]

In short...

$\mathsf{L}_{\mathsf{Kh}_i}$ logic [Ch. 4]

- Distiction between ontic and epistemic information
- Represents more reasons of knowing how
- Sound and complete axiomatization [Ch. 5]
- Bisimulations results and formula equivalence [Ch. 6]
- Generalizes the semantics based on LTSs [Ch. 6]
- Lower computational complexity [Ch. 7]

Dynamic operators [Ch. 8]

- Public announcements ($[!\chi]$) and arrow updates ($[E]$) [Ch. 8.1]

In short...

$\mathsf{L}_{\mathsf{Kh}_i}$ logic [Ch. 4]

- Distiction between ontic and epistemic information
- Represents more reasons of knowing how
- Sound and complete axiomatization [Ch. 5]
- Bisimulations results and formula equivalence [Ch. 6]
- Generalizes the semantics based on LTSs [Ch. 6]
- Lower computational complexity [Ch. 7]

Dynamic operators [Ch. 8]

- Public announcements ($[!\chi]$) and arrow updates ($[E]$) [Ch. 8.1]
- Specific refinement $\langle \sigma_1 \nrightarrow \sigma_2 \rangle$, arbitrary refinement $\langle \nrightarrow \rangle$, public refinement of a plan $[!\sigma]$ and its private version $[!\sigma, i]$ [Ch. 8.2 and 9]

In short...

$\mathsf{L}_{\mathsf{Kh}_i}$ logic [Ch. 4]

- Distiction between ontic and epistemic information
- Represents more reasons of knowing how
- Sound and complete axiomatization [Ch. 5]
- Bisimulations results and formula equivalence [Ch. 6]
- Generalizes the semantics based on LTSs [Ch. 6]
- Lower computational complexity [Ch. 7]

Dynamic operators [Ch. 8]

- Public announcements ($[!\chi]$) and arrow updates ($[E]$) [Ch. 8.1]
- Specific refinement $\langle \sigma_1 \nsim \sigma_2 \rangle$, arbitrary refinement $\langle \nsim \rangle$, public refinement of a plan $[!\sigma]$ and its private version $[!\sigma, i]$ [Ch. 8.2 and 9]

Deontic operators (abilities, norms and compliance) [Ch. 10]

Future work

- Consider alternatives to strong executability (weak executability, traces of partial executions, and so on)

Future work

- Consider alternatives to strong executability (weak executability, traces of partial executions, and so on)
- Caracterize the exact complexity class of the dynamic and deontic logics

Future work

- Consider alternatives to strong executability (weak executability, traces of partial executions, and so on)
- Caracterize the exact complexity class of the dynamic and deontic logics
- Analysis of possible class of models, expressivity results, axiomatizations and extensions of the base language

Future work

- Consider alternatives to strong executability (weak executability, traces of partial executions, and so on)
- Caracterize the exact complexity class of the dynamic and deontic logics
- Analysis of possible class of models, expressivity results, axiomatizations and extensions of the base language
- Consider other dynamic operators such as “learning how” and “forgetting how”

Referencias

Hintikka, J. (1962). *Knowledge and Belief*. Ithaca, NY: Cornell University Press.

Wang, Y. (2015). "A Logic of Knowing How". En: *Logic, Rationality, and Interaction – 5th International Workshop, LORI 2015*, págs. 392-405. DOI: 10.1007/978-3-662-48561-3_32.