First Steps in Updating Knowing How

Carlos Areces[a], Raul Fervari[a,b], Andrés R. Saravia[a] and
Fernando R. Velázquez-Quesada[c]

[a]CONICET and Universidad Nacional de Córdoba, Argentina
[b]Guangdong Technion Israel Institute of Technology, China
[c]University of Bergen, Norway

31/VIII/2022

Overview of the talk

- Background
- The Knowing How logic
- Dynamic modalities: Ontic & epistemic updates
- Conclusions and future work

An Epistemic Logic of Knowing How

- Knowing how: epistemic notion related to *the abilities of an agent has to achieve a goal*.

An Epistemic Logic of Knowing How

- Knowing how: epistemic notion related to *the abilities of an agent has to achieve a goal*.
- Wang [2015,2018]: a framework for knowing how logics.

An Epistemic Logic of Knowing How

- Knowing how: epistemic notion related to *the abilities of an agent has to achieve a goal*.
- Wang [2015,2018]: a framework for knowing how logics.
- Areces et al. [2021]: a generalized version by introducing epistemic indistinguishability.

An Epistemic Logic of Knowing How

- Knowing how: epistemic notion related to *the abilities of an agent has to achieve a goal*.
- Wang [2015,2018]: a framework for knowing how logics.
- Areces et al. [2021]: a generalized version by introducing epistemic indistinguishability.
    - Makes a distinction between ontic/factual information, and epistemic information.

An Epistemic Logic of Knowing How

- Knowing how: epistemic notion related to *the abilities of an agent has to achieve a goal*.
- Wang [2015,2018]: a framework for knowing how logics.
- Areces et al. [2021]: a generalized version by introducing epistemic indistinguishability.
  - Makes a distinction between ontic/factual information, and epistemic information.
- This work: a dynamic epistemic approach of knowing how.

An Epistemic Logic of Knowing How

- Knowing how: epistemic notion related to *the abilities of an agent has to achieve a goal*.
- Wang [2015,2018]: a framework for knowing how logics.
- Areces et al. [2021]: a generalized version by introducing epistemic indistinguishability.
  - Makes a distinction between ontic/factual information, and epistemic information.
- This work: a dynamic epistemic approach of knowing how.
  - Actions updating different kinds of information.

Dynamic Operators over Knowing How

- We introduce dynamic modalities of two types:

Dynamic Operators over Knowing How

- We introduce dynamic modalities of two types:
  1. Ontic updates

Dynamic Operators over Knowing How

- We introduce dynamic modalities of two types:
  1. Ontic updates: modify the ontic information of the models (announcements and arrow updates)

Dynamic Operators over Knowing How

- We introduce dynamic modalities of two types:
  1. Ontic updates: modify the ontic information of the models (announcements and arrow updates)
  2. Epistemic updates

Dynamic Operators over Knowing How

- We introduce dynamic modalities of two types:
  1. Ontic updates: modify the ontic information of the models (announcements and arrow updates)
  2. Epistemic updates: modify the perception of the agent about her own abilities (refinements, learning how)

Knowing How: Models

Definition (Uncertainty-based LTS)

An LTS^{U} is a tuple $\mathcal{M} = \langle \mathsf{W}, \{\mathsf{R}_a\}_{a \in \mathsf{Act}}, \{\mathbb{S}_i\}_{i \in \mathsf{Agt}}, \mathsf{V} \rangle$ where:

Knowing How: Models

Definition (Uncertainty-based LTS)

An LTS^{U} is a tuple $\mathcal{M} = \langle \text{W}, \{\text{R}_a\}_{a \in \text{Act}}, \{\mathbb{S}_i\}_{i \in \text{Agt}}, \text{V} \rangle$ where:

- $\langle \text{W}, \{\text{R}_a\}_{a \in \text{Act}}, \text{V} \rangle$ is an LTS

Knowing How: Models

Definition (Uncertainty-based LTS)

An LTS^{U} is a tuple $\mathcal{M} = \langle \text{W}, \{\text{R}_a\}_{a \in \text{Act}}, \{\mathbb{S}_i\}_{i \in \text{Agt}}, \text{V} \rangle$ where:

- $\langle \text{W}, \{\text{R}_a\}_{a \in \text{Act}}, \text{V} \rangle$ is an LTS and
- $\mathbb{S}_i \subseteq 2^{\text{Act}^*} \setminus \{\emptyset\}$ s.t.

Knowing How: Models

Definition (Uncertainty-based LTS)

An LTS^{U} is a tuple $\mathcal{M} = \langle \mathsf{W}, \{\mathsf{R}_a\}_{a \in \mathsf{Act}}, \{\mathbb{S}_i\}_{i \in \mathsf{Agt}}, \mathsf{V} \rangle$ where:

- $\langle \mathsf{W}, \{\mathsf{R}_a\}_{a \in \mathsf{Act}}, \mathsf{V} \rangle$ is an LTS and
- $\mathbb{S}_i \subseteq 2^{\mathsf{Act}^*} \setminus \{\emptyset\}$ s.t.
  - $\emptyset \notin \mathbb{S}_i$

Knowing How: Models

Definition (Uncertainty-based LTS)

An LTS^{U} is a tuple $\mathcal{M} = \langle \text{W}, \{\text{R}_a\}_{a \in \text{Act}}, \{\mathbb{S}_i\}_{i \in \text{Agt}}, \text{V} \rangle$ where:

- $\langle \text{W}, \{\text{R}_a\}_{a \in \text{Act}}, \text{V} \rangle$ is an LTS and
- $\mathbb{S}_i \subseteq 2^{\text{Act}^*} \setminus \{\emptyset\}$ s.t.
  - $\emptyset \notin \mathbb{S}_i$
  - $\pi_1, \pi_2 \in \mathbb{S}_i$ with $\pi_1 \neq \pi_2$ implies $\pi_1 \cap \pi_2 = \emptyset$

Knowing How: Models

Definition (Uncertainty-based LTS)

An LTS^{U} is a tuple $\mathcal{M} = \langle W, \{R_a\}_{a \in \text{Act}}, \{\mathbb{S}_i\}_{i \in \text{Agt}}, V \rangle$ where:

- $\langle W, \{R_a\}_{a \in \text{Act}}, V \rangle$ is an LTS and
- $\mathbb{S}_i \subseteq 2^{\text{Act}^*} \setminus \{\emptyset\}$ s.t.
  - $\emptyset \notin \mathbb{S}_i$
  - $\pi_1, \pi_2 \in \mathbb{S}_i$ with $\pi_1 \neq \pi_2$ implies $\pi_1 \cap \pi_2 = \emptyset$

$\mathbb{S}_i$ *represents the sets of plans agent* i *cannot distinguish between each other.*

Baking a good cake (a simplified scenario)

Baking a good cake (a simplified scenario)

- Two agents attempt to produce a good cake (a goal g), provided they have all the ingredientes (h).

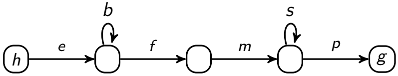
Baking a good cake (a simplified scenario)

- Two agents attempt to produce a good cake (a goal g), provided they have all the ingredientes (h).
- g is achieved via the following actions: adding eggs (e), beating the eggs (b), adding flour (f), adding milk (m), stir (s) and bake the preparation (p) (the plan *ebfmsp*).

Baking a good cake (a simplified scenario)

- Two agents attempt to produce a good cake (a goal g), provided they have all the ingredientes (h).
- g is achieved via the following actions: adding eggs (e), beating the eggs (b), adding flour (f), adding milk (m), stir (s) and bake the preparation (p) (the plan *ebfmsp*).

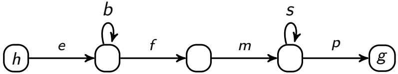
$\mathcal{M}$:



Baking a good cake (a simplified scenario)

- Two agents attempt to produce a good cake (a goal g), provided they have all the ingredientes (h).
- g is achieved via the following actions: adding eggs (e), beating the eggs (b), adding flour (f), adding milk (m), stir (s) and bake the preparation (p) (the plan *ebfmsp*).
- Agent i is aware of that is the way to get a good cake.

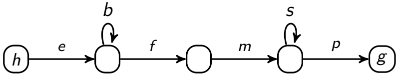
$\mathcal{M}$:



Baking a good cake (a simplified scenario)

- Two agents attempt to produce a good cake (a goal g), provided they have all the ingredientes (h).
- g is achieved via the following actions: adding eggs (e), beating the eggs (b), adding flour (f), adding milk (m), stir (s) and bake the preparation (p) (the plan *ebfmsp*).
- Agent i is aware of that is the way to get a good cake.

$\mathcal{M}$:

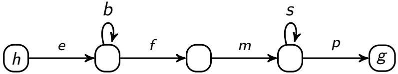


$$\mathbb{S}_i = \{\{ebfmsp\}\},$$

Baking a good cake (a simplified scenario)

- Two agents attempt to produce a good cake (a goal g), provided they have all the ingredientes (h).
- g is achieved via the following actions: adding eggs (e), beating the eggs (b), adding flour (f), adding milk (m), stir (s) and bake the preparation (p) (the plan *ebfmsp*).
- Agent i is aware of that is the way to get a good cake.
- Agent j considers that the order in the instructions do not matter (e.g., *ebfmsp* and *ebmfsp* are indistinguishable).

$\mathcal{M}$:

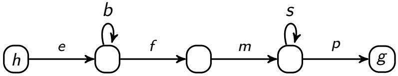


$$\mathbb{S}_i = \{\{ebfmsp\}\},$$

Baking a good cake (a simplified scenario)

- Two agents attempt to produce a good cake (a goal g), provided they have all the ingredientes (h).
- g is achieved via the following actions: adding eggs (e), beating the eggs (b), adding flour (f), adding milk (m), stir (s) and bake the preparation (p) (the plan *ebfmsp*).
- Agent i is aware of that is the way to get a good cake.
- Agent j considers that the order in the instructions do not matter (e.g., *ebfmsp* and *ebmfsp* are indistinguishable).

$\mathcal{M}$:



$$\mathbb{S}_i = \{\{ebfmsp\}\},$$
$$\mathbb{S}_j = \{\{ebfmsp, ebmfsp\}\}$$

Strong executability (SE)

Strong executability (SE)

A plan should be fail proof:

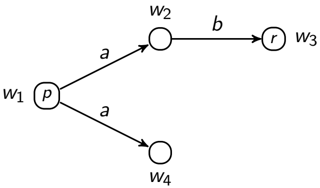
Every partial execution should be completed.

Strong executability (SE)

A plan should be fail proof:

Every partial execution should be completed.

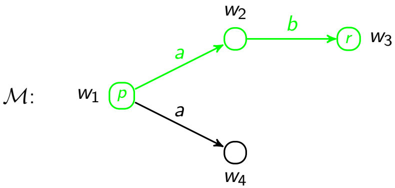
$\mathcal{M}$:



Strong executability (SE)

A plan should be fail proof:

Every partial execution should be completed.

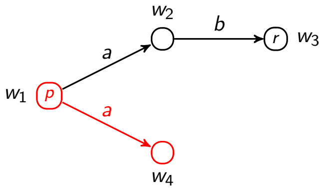


Strong executability (SE)

A plan should be fail proof:

Every partial execution should be completed.

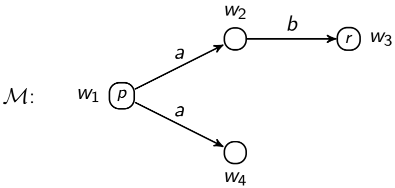
$\mathcal{M}$:



Strong executability (SE)

A plan should be fail proof:

Every partial execution should be completed.

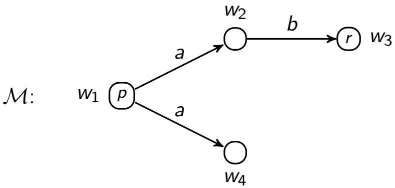


ab is not strongly executable at w_1

Strong executability (SE)

A plan should be fail proof:

Every partial execution should be completed.



ab is not strongly executable at w_1

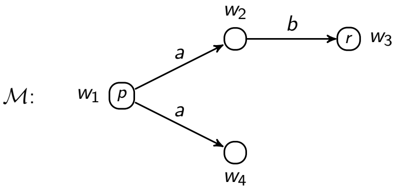
- $\sigma \in \text{Act}^*$ is SE at a state u iff

  every partial execution of σ from u can be completed.

Strong executability (SE)

A plan should be fail proof:

Every partial execution should be completed.



ab is not strongly executable at w_1

- $\sigma \in \text{Act}^*$ is SE at a state u iff

  every partial execution of σ from u can be completed.
- $\pi \subseteq \text{Act}^*$ is SE at a state u iff for *all* $\sigma \in \pi$, σ is SE at u.

Knowing How: Formulas and semantics

Definition (L_{Kh_i}-formulas)

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

Knowing How: Formulas and semantics

Definition (L_{Kh_i}-formulas)

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

$\mathsf{Kh}_i(\psi, \varphi)$: *"The agent i knows how to achieve φ given ψ."*

Knowing How: Formulas and semantics

Definition (L_{Kh_i}-formulas)

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

$\mathsf{Kh}_i(\psi, \varphi)$: *"The agent i knows how to achieve φ given ψ."*

Definition

$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ *iff$_{def}$* there is $\pi \in \mathbb{S}_i$ s.t.

Knowing How: Formulas and semantics

Definition (L_{Kh_i}-formulas)

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

$\mathsf{Kh}_i(\psi, \varphi)$: *"The agent i knows how to achieve φ given ψ."*

Definition

$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ *iff*$_{def}$ there is $\pi \in \mathbb{S}_i$ s.t.

1. π is SE at all ψ-states

Knowing How: Formulas and semantics

Definition (L_{Kh_i}-formulas)

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

$\mathsf{Kh}_i(\psi, \varphi)$: *"The agent i knows how to achieve φ given ψ."*

Definition

$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ *iff*$_{def}$ there is $\pi \in \mathbb{S}_i$ s.t.

1. π is SE at all ψ-states, and
2. from ψ-states π reaches only to φ-states.

Knowing How: Example

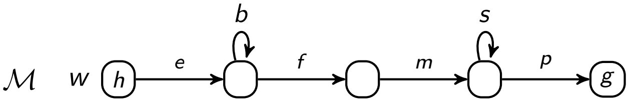
$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ *iff*$_{def}$ there is $\pi \in \mathbb{S}_i$ s.t.

1. π is SE at all ψ-states, and
2. from ψ-states π reaches only to φ-states.

Knowing How: Example

$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ *iff*$_{def}$ there is $\pi \in \mathbb{S}_i$ s.t.

1. π is SE at all ψ-states, and
2. from ψ-states π reaches only to φ-states.

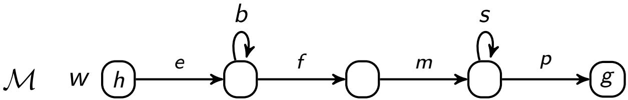


- $\mathbb{S}_i = \{\{ebfmsp\}\}$, $\mathbb{S}_j = \{\{ebfmsp, ebmfsp\}\}$.

Knowing How: Example

$\mathcal{M}, w \models \mathsf{Kh}_i(\psi, \varphi)$ *iff$_{def}$* there is $\pi \in \mathbb{S}_i$ s.t.

1. π is SE at all ψ-states, and
2. from ψ-states π reaches only to φ-states.



- $\mathbb{S}_i = \{\{ebfmsp\}\}, \qquad \mathbb{S}_j = \{\{ebfmsp, ebmfsp\}\}.$
- $\mathcal{M}, w \models \mathsf{Kh}_i(h, g) \wedge \neg\mathsf{Kh}_j(h, g)$

Ontic vs. Epistemic Information

Two distinct types of information in an LTS^U:

Ontic vs. Epistemic Information

Two distinct types of information in an $\mathsf{LTS}^{\mathsf{U}}$:

- Ontic information: provided by the graph part

Ontic vs. Epistemic Information

Two distinct types of information in an LTS^{U}:

- Ontic information: provided by the graph part
  - the available states, the accessibility relations, etc.

Ontic vs. Epistemic Information

Two distinct types of information in an $\mathsf{LTS}^{\mathsf{U}}$:

- Ontic information: provided by the graph part
  - the available states, the accessibility relations, etc.
- Epistemic information: given by the indistinguishability sets $\mathbb{S}_i$

Ontic vs. Epistemic Information

Two distinct types of information in an $\mathsf{LTS}^{\mathsf{U}}$:

- Ontic information: provided by the graph part
  - the available states, the accessibility relations, etc.
- Epistemic information: given by the indistinguishability sets $\mathbb{S}_i$
  - the perception of each agent about her own abilities.

Ontic vs. Epistemic Information

Two distinct types of information in an LTS^{U}:

- Ontic information: provided by the graph part
  - the available states, the accessibility relations, etc.
- Epistemic information: given by the indistinguishability sets $\mathbb{S}_i$
  - the perception of each agent about her own abilities.

This enables us to define ways of updating these two types of information.

Ontic updates: Announcement

Definition ($\mathsf{PAL}_{\mathsf{Kh}_i}$ formulas)

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

Ontic updates: Announcement

Definition ($\mathsf{PAL}_{\mathsf{Kh}_i}$ formulas)

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

$[!\chi]\varphi$: *"After announcing χ, φ holds."*

Ontic updates: Announcement

Definition ($\mathsf{PAL}_{\mathsf{Kh}_i}$ formulas)

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

$[!\chi]\varphi$: *"After announcing χ, φ holds."*

Definition ($\mathcal{M}_{!\chi}$)

$\mathcal{M}, w \models [!\chi]\varphi$ *iff* $\mathcal{M}, w \models \chi$ implies $\mathcal{M}_{!\chi}, w \models \varphi$

Ontic updates: Announcement

Definition ($\mathsf{PAL}_{\mathsf{Kh}_i}$ formulas)

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

$[!\chi]\varphi$: *"After announcing χ, φ holds."*

Definition ($\mathcal{M}_{!\chi}$)

$\mathcal{M}, w \models [!\chi]\varphi$ *iff* $\mathcal{M}, w \models \chi$ implies $\mathcal{M}_{!\chi}, w \models \varphi$; where
$\mathcal{M}_{!\chi} = \langle \mathsf{W}_{!\chi}, \mathsf{R}_{!\chi}, \mathbb{S}, \mathsf{V}_{!\chi} \rangle$:

Ontic updates: Announcement

Definition ($\mathsf{PAL}_{\mathsf{Kh}_i}$ formulas)

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

$[!\chi]\varphi$: *"After announcing χ, φ holds."*

Definition ($\mathcal{M}_{!\chi}$)

$\mathcal{M}, w \models [!\chi]\varphi$ *iff* $\mathcal{M}, w \models \chi$ implies $\mathcal{M}_{!\chi}, w \models \varphi$; where $\mathcal{M}_{!\chi} = \langle \mathsf{W}_{!\chi}, \mathsf{R}_{!\chi}, \mathbb{S}, \mathsf{V}_{!\chi} \rangle$:

- $\mathsf{W}_{!\chi} = [\![\chi]\!]^{\mathcal{M}}$

Ontic updates: Announcement

Definition ($\mathsf{PAL}_{\mathsf{Kh}_i}$ formulas)

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

$[!\chi]\varphi$: *"After announcing χ, φ holds."*

Definition ($\mathcal{M}_{!\chi}$)

$\mathcal{M}, w \models [!\chi]\varphi$ *iff* $\mathcal{M}, w \models \chi$ implies $\mathcal{M}_{!\chi}, w \models \varphi$; where $\mathcal{M}_{!\chi} = \langle \mathsf{W}_{!\chi}, \mathsf{R}_{!\chi}, \mathbb{S}, \mathsf{V}_{!\chi} \rangle$:

- $\mathsf{W}_{!\chi} = [\![\chi]\!]^{\mathcal{M}}$,
- $(\mathsf{R}_{!\chi})_a = \{(w, v) \in \mathsf{R}_a \mid w \in [\![\chi]\!]^{\mathcal{M}}, \mathsf{R}_a(w) \subseteq [\![\chi]\!]^{\mathcal{M}}\}$

Ontic updates: Announcement

Definition ($\mathsf{PAL}_{\mathsf{Kh}_i}$ formulas)

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

$[!\chi]\varphi$: *"After announcing χ, φ holds."*

Definition ($\mathcal{M}_{!\chi}$)

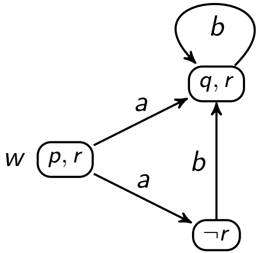
$\mathcal{M}, w \models [!\chi]\varphi$ *iff* $\mathcal{M}, w \models \chi$ implies $\mathcal{M}_{!\chi}, w \models \varphi$; where $\mathcal{M}_{!\chi} = \langle \mathsf{W}_{!\chi}, \mathsf{R}_{!\chi}, \mathbb{S}, \mathsf{V}_{!\chi} \rangle$:

- $\mathsf{W}_{!\chi} = [\![\chi]\!]^{\mathcal{M}}$,
- $(\mathsf{R}_{!\chi})_a = \{(w, v) \in \mathsf{R}_a \mid w \in [\![\chi]\!]^{\mathcal{M}}, \mathsf{R}_a(w) \subseteq [\![\chi]\!]^{\mathcal{M}}\}$, and
- $\mathsf{V}_{!\chi}(w) = \mathsf{V}(w)$.

Announcement: Example

$$\mathcal{M}, w \models \mathsf{Kh}_i(p, q),\ \mathbb{S}_i = \{\{ab\}\}$$

$\mathcal{M}$:

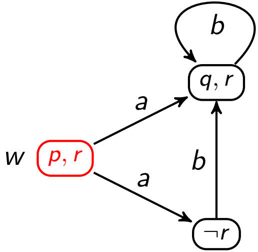


$$\mathcal{M}, w \models [!r]\mathsf{Kh}_i(p, q)?$$

Announcement: Example

$$\mathcal{M}, w \models \mathsf{Kh}_i(p, q),\ \mathbb{S}_i = \{\{ab\}\}$$

$\mathcal{M}$:

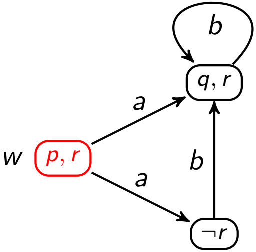


$$\mathcal{M}, w \models [!r]\mathsf{Kh}_i(p, q)$$

Announcement: Example

$$\mathcal{M}, w \models \mathsf{Kh}_i(p, q),\ \mathbb{S}_i = \{\{ab\}\}$$

$\mathcal{M}$:

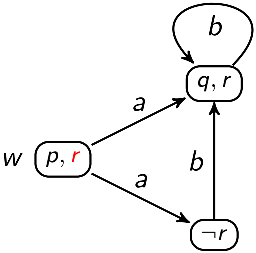


$$\mathcal{M}, w \models r \rightarrow \mathcal{M}_{!r}, w \models \mathsf{Kh}_i(p, q)$$

Announcement: Example

$$\mathcal{M}, w \models \mathsf{Kh}_i(p, q),\ \mathbb{S}_i = \{\{ab\}\}$$

$\mathcal{M}$:

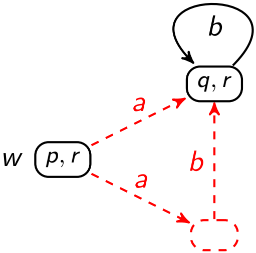


$$\mathcal{M}, w \models r \rightarrow \mathcal{M}_{!r}, w \models \mathsf{Kh}_i(p, q)$$

Announcement: Example

$$\mathcal{M}, w \models \mathsf{Kh}_i(p, q),\ \mathbb{S}_i = \{\{ab\}\}$$

$\mathcal{M}$:

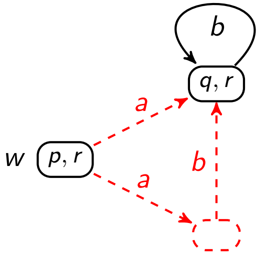


$$\mathcal{M}, w \models r \to \mathcal{M}_{!r}, w \models \mathsf{Kh}_i(p, q)$$

Announcement: Example

$$\mathcal{M}, w \models \mathsf{Kh}_i(p, q),\ \mathbb{S}_i = \{\{ab\}\}$$

$\mathcal{M}$:

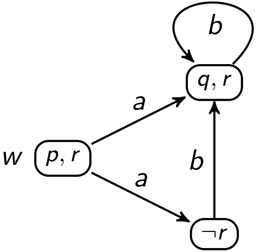


$$\mathcal{M}, w \models r \rightarrow \mathcal{M}_{!r}, w \not\models \mathsf{Kh}_i(p, q)$$

Announcement: Example

$$\mathcal{M}, w \models \mathsf{Kh}_i(p, q),\ \mathbb{S}_i = \{\{ab\}\}$$

$\mathcal{M}$:



$$\mathcal{M}, w \not\models [!r]\mathsf{Kh}_i(p, q)$$

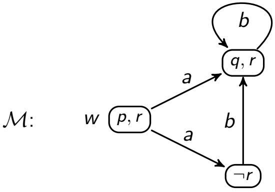
Theorem

$\mathsf{PAL}_{\mathsf{Kh}_i}$ *is more expresive than* $\mathsf{L}_{\mathsf{Kh}_i}$ *over arbitrary* $\mathsf{LTS}^{\mathsf{U}}$*s.*

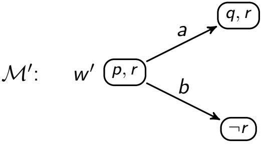
Theorem

$\mathsf{PAL}_{\mathsf{Kh}_i}$ *is more expresive than* $\mathsf{L}_{\mathsf{Kh}_i}$ *over arbitrary* $\mathsf{LTS}^{\mathsf{U}}$*s.*

Let $\mathcal{M}$ and $\mathcal{M}'$ two indistinguishable $\mathsf{LTS}^{\mathsf{U}}$s for $\mathsf{L}_{\mathsf{Kh}_i}$:



$\mathcal{M}, w \not\models [!r]\mathsf{Kh}_i(p, q)$
$\mathbb{S}_i := \{\{ab\}\}$

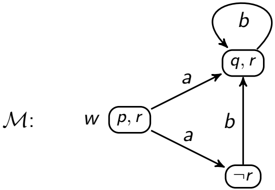


$\mathcal{M}', w' \models [!r]\mathsf{Kh}_i(p, q)$?
$\mathbb{S}'_i := \{\{a\}\}$

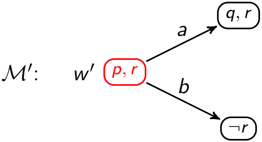
Theorem

$\mathsf{PAL}_{\mathsf{Kh}_i}$ *is more expresive than* $\mathsf{L}_{\mathsf{Kh}_i}$ *over arbitrary* $\mathsf{LTS^U}$*s.*

Let $\mathcal{M}$ and $\mathcal{M}'$ two indistinguishable $\mathsf{LTS^U}$s for $\mathsf{L}_{\mathsf{Kh}_i}$:



$\mathcal{M}, w \not\models [!r]\mathsf{Kh}_i(p, q)$
$\mathbb{S}_i := \{\{ab\}\}$

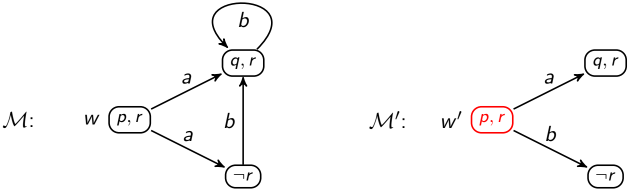


$\mathcal{M}', w' \models [!r]\mathsf{Kh}_i(p, q)$
$\mathbb{S}_i' := \{\{a\}\}$

Theorem

$\mathrm{PAL}_{\mathrm{Kh}_i}$ *is more expresive than* $\mathrm{L}_{\mathrm{Kh}_i}$ *over arbitrary* $\mathrm{LTS^U}$*s.*

Let $\mathcal{M}$ and $\mathcal{M}'$ two indistinguishable $\mathrm{LTS^U}$s for $\mathrm{L}_{\mathrm{Kh}_i}$:



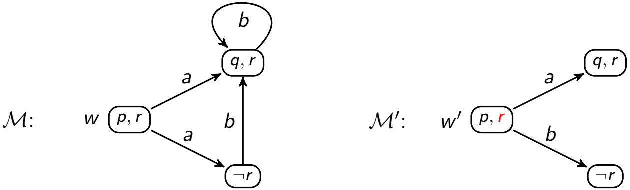
$\mathcal{M}, w \not\models [!r]\mathrm{Kh}_i(p, q)$
$\mathbb{S}_i := \{\{ab\}\}$

$\mathcal{M}', w' \models r \rightarrow \mathcal{M}_{!r}, w' \models \mathrm{Kh}_i(p, q)$
$\mathbb{S}'_i := \{\{a\}\}$

Theorem

$\mathsf{PAL}_{\mathsf{Kh}_i}$ *is more expresive than* $\mathsf{L}_{\mathsf{Kh}_i}$ *over arbitrary* $\mathsf{LTS}^{\mathsf{U}}$*s.*

Let $\mathcal{M}$ and $\mathcal{M}'$ two indistinguishable $\mathsf{LTS}^{\mathsf{U}}$s for $\mathsf{L}_{\mathsf{Kh}_i}$:



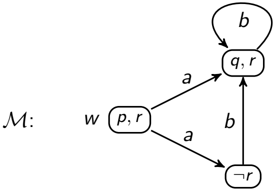
$\mathcal{M}, w \not\models [!r]\mathsf{Kh}_i(p, q)$ $\quad$ $\mathcal{M}', w' \models r \rightarrow \mathcal{M}_{!r}, w' \models \mathsf{Kh}_i(p, q)$

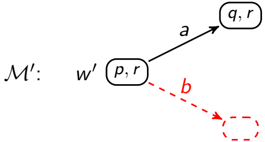
$\mathbb{S}_i := \{\{ab\}\}$ $\quad$ $\mathbb{S}'_i := \{\{a\}\}$

Theorem

$\mathsf{PAL}_{\mathsf{Kh}_i}$ *is more expresive than* $\mathsf{L}_{\mathsf{Kh}_i}$ *over arbitrary* $\mathsf{LTS}^{\mathsf{U}}$*s.*

Let $\mathcal{M}$ and $\mathcal{M}'$ two indistinguishable $\mathsf{LTS}^{\mathsf{U}}$s for $\mathsf{L}_{\mathsf{Kh}_i}$:





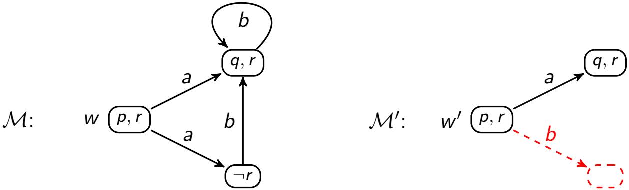
$\mathcal{M}, w \not\models [!r]\mathsf{Kh}_i(p, q)$
$\mathbb{S}_i := \{\{ab\}\}$

$\mathcal{M}', w' \models r \rightarrow \mathcal{M}_{!r}, w' \models \mathsf{Kh}_i(p, q)$
$\mathbb{S}'_i := \{\{a\}\}$

Theorem

$\mathsf{PAL}_{\mathsf{Kh}_i}$ *is more expresive than* $\mathsf{L}_{\mathsf{Kh}_i}$ *over arbitrary* $\mathsf{LTS}^{\mathsf{U}}$*s.*

Let $\mathcal{M}$ and $\mathcal{M}'$ two indistinguishable $\mathsf{LTS}^{\mathsf{U}}$s for $\mathsf{L}_{\mathsf{Kh}_i}$:



$\mathcal{M}, w \not\models [!r]\mathsf{Kh}_i(p, q)$

$\mathbb{S}_i := \{\{ab\}\}$

$\mathcal{M}', w' \models r \rightarrow \mathcal{M}_{!r}, w' \models \mathsf{Kh}_i(p, q)$

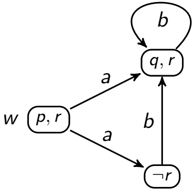
$\mathbb{S}'_i := \{\{a\}\}$

Theorem

$\mathsf{PAL}_{\mathsf{Kh}_i}$ *is more expresive than* $\mathsf{L}_{\mathsf{Kh}_i}$ *over arbitrary* $\mathsf{LTS}^{\mathsf{U}}$*s.*

Let $\mathcal{M}$ and $\mathcal{M}'$ two indistinguishable $\mathsf{LTS}^{\mathsf{U}}$s for $\mathsf{L}_{\mathsf{Kh}_i}$:

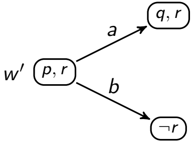
$\mathcal{M}$:



$\mathcal{M}, w \not\models [!r]\mathsf{Kh}_i(p, q)$

$\mathbb{S}_i := \{\{ab\}\}$

$\mathcal{M}'$:



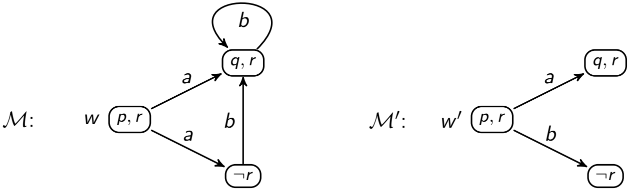
$\mathcal{M}', w' \models [!r]\mathsf{Kh}_i(p, q)$

$\mathbb{S}'_i := \{\{a\}\}$

Theorem

$\mathsf{PAL}_{\mathsf{Kh}_i}$ *is more expresive than* $\mathsf{L}_{\mathsf{Kh}_i}$ *over arbitrary* $\mathsf{LTS}^{\mathsf{U}}$*s.*

Let $\mathcal{M}$ and $\mathcal{M}'$ two indistinguishable $\mathsf{LTS}^{\mathsf{U}}$s for $\mathsf{L}_{\mathsf{Kh}_i}$:



$\mathcal{M}, w \not\models [!r]\mathsf{Kh}_i(p, q)$
$\mathbb{S}_i := \{\{ab\}\}$

$\mathcal{M}', w' \models [!r]\mathsf{Kh}_i(p, q)$
$\mathbb{S}'_i := \{\{a\}\}$

$\mathsf{PAL}_{\mathsf{Kh}_i}$ can distinguish between the *class of arbitrary models* and the class of models s.t. for all $\pi \in \mathbb{S}_i$, $\pi \subseteq \mathsf{Act}$.

Reduction axioms

$\mathsf{PAL}_{\mathsf{Kh}_i}$ can distinguish between the *class of arbitrary models* and the class of models s.t. for all $\pi \in \mathbb{S}_i$, $\pi \subseteq \mathsf{Act}$.

Reduction axioms

$\mathsf{PAL}_{\mathsf{Kh}_i}$ can distinguish between the *class of arbitrary models* and the class of models s.t. for all $\pi \in \mathbb{S}_i$, $\pi \subseteq \mathsf{Act}$.

- This cannot be done in $\mathsf{L}_{\mathsf{Kh}_i}$ [Areces et al. (TARK 2021)].

Reduction axioms

$\mathsf{PAL}_{\mathsf{Kh}_i}$ can distinguish between the *class of arbitrary models* and the class of models s.t. for all $\pi \in \mathbb{S}_i$, $\pi \subseteq \mathsf{Act}$.

- This cannot be done in $\mathsf{L}_{\mathsf{Kh}_i}$ [Areces et al. (TARK 2021)].
- In these models (where $\pi \subseteq \mathsf{Act}$), every $\mathsf{PAL}_{\mathsf{Kh}_i}$ formula can be reduced to a $\mathsf{L}_{\mathsf{Kh}_i}$ formula:

Reduction axioms

$\mathsf{PAL}_{\mathsf{Kh}_i}$ can distinguish between the *class of arbitrary models* and the class of models s.t. for all $\pi \in \mathbb{S}_i$, $\pi \subseteq \mathsf{Act}$.

- This cannot be done in $\mathsf{L}_{\mathsf{Kh}_i}$ [Areces et al. (TARK 2021)].
- In these models (where $\pi \subseteq \mathsf{Act}$), every $\mathsf{PAL}_{\mathsf{Kh}_i}$ formula can be reduced to a $\mathsf{L}_{\mathsf{Kh}_i}$ formula:

$$[!\chi]\mathsf{Kh}_i(\varphi, \psi) \leftrightarrow (\chi \rightarrow \mathsf{Kh}_i(\chi \wedge [!\chi]\varphi, \chi \wedge [!\chi]\psi)).$$

Other kinds of updates

- $\mathsf{PAL}_{\mathsf{Kh}_i}$ is not the only way of updating ontic information.

Other kinds of updates

- $\mathsf{PAL}_{\mathsf{Kh}_i}$ is not the only way of updating ontic information.
- We applied similar ideas using an Arrow Update Logic [Kooi and Renne (RSL 2011)].

Other kinds of updates

- PAL_{Kh_i} is not the only way of updating ontic information.
- We applied similar ideas using an Arrow Update Logic [Kooi and Renne (RSL 2011)].
- We can also perform epistemic updates (affecting directly the "knowing how").

Other kinds of updates

- $\mathsf{PAL}_{\mathsf{Kh}_i}$ is not the only way of updating ontic information.
- We applied similar ideas using an Arrow Update Logic [Kooi and Renne (RSL 2011)].
- We can also perform epistemic updates (affecting directly the "knowing how").
- Proposal: refining the indistinguishability between plans, i.e., making plans distinguishable for the agent.

Other kinds of updates

- $\mathsf{PAL}_{\mathsf{Kh}_i}$ is not the only way of updating ontic information.
- We applied similar ideas using an Arrow Update Logic [Kooi and Renne (RSL 2011)].
- We can also perform epistemic updates (affecting directly the "knowing how").
- Proposal: refining the indistinguishability between plans, i.e., making plans distinguishable for the agent.
  1. Explicit refinement for two given plans.

Other kinds of updates

- $\mathsf{PAL}_{\mathsf{Kh}_i}$ is not the only way of updating ontic information.
- We applied similar ideas using an Arrow Update Logic [Kooi and Renne (RSL 2011)].
- We can also perform epistemic updates (affecting directly the "knowing how").
- Proposal: refining the indistinguishability between plans, i.e., making plans distinguishable for the agent.
  1. Explicit refinement for two given plans.
  2. Arbitrary refinements.

Other kinds of updates

- PAL_{Kh_i} is not the only way of updating ontic information.
- We applied similar ideas using an Arrow Update Logic [Kooi and Renne (RSL 2011)].
- We can also perform epistemic updates (affecting directly the "knowing how").
- Proposal: refining the indistinguishability between plans, i.e., making plans distinguishable for the agent.
  1. Explicit refinement for two given plans.
  2. Arbitrary refinements.
  3. "Learning how".

Epistemic updates: Refinement (L_{Ref})

Definition (L_{Ref} formulas)

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{Kh}_i(\varphi,\varphi) \mid \langle \sigma_1 \not\rightsquigarrow \sigma_2 \rangle \varphi$$

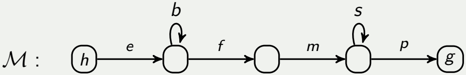
Epistemic updates: Refinement (L_{Ref})

Definition (L_{Ref} formulas)

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{Kh}_i(\varphi, \varphi) \mid \langle \sigma_1 \not\sim \sigma_2 \rangle \varphi$$

$\langle \sigma_1 \not\sim \sigma_2 \rangle \varphi$: *"After it is stated that plans σ_1 and σ_2 are distinguishable, φ holds."*

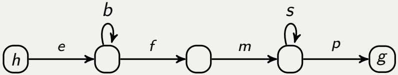
Epistemic updates: Refinement (L_{Ref}) (cont.)



$$\mathbb{S}_i = \{\{ebfmsp\}\}, \qquad \mathbb{S}_j = \{\{ebfmsp, ebmfsp\}\}$$

- $\mathcal{M}, w \not\models \mathsf{Kh}_j(h, g)$

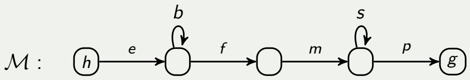
Epistemic updates: Refinement (L_{Ref}) (cont.)

$\mathcal{M}$: 

$\mathbb{S}_i = \{\{ebfmsp\}\}$, $\mathbb{S}_j = \{\{ebfmsp\}, \{ebmfsp\}\}$

- $\mathcal{M}, w \not\models \mathrm{Kh}_j(h, g)$ but $\mathcal{M}, w \models \langle ebfmsp \not\sim ebmfsp \rangle \mathrm{Kh}_j(h, g)$;

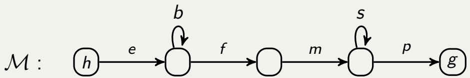
Epistemic updates: Refinement ($\mathsf{L}_{\mathsf{Ref}}$) (cont.)



$\mathbb{S}_i = \{\{ebfmsp\}\}$, $\mathbb{S}_j = \{\{ebfmsp\}, \{ebmfsp\}\}$

- $\mathcal{M}, w \not\models \mathsf{Kh}_j(h, g)$ but $\mathcal{M}, w \models \langle ebfmsp \not\sim ebmfsp \rangle \mathsf{Kh}_j(h, g)$;
  - **generates new knowledge**

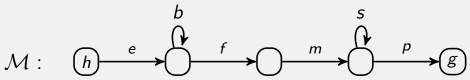
Epistemic updates: Refinement (L_{Ref}) (cont.)



$\mathbb{S}_i = \{\{ebfmsp\}\}$, $\mathbb{S}_j = \{\{ebfmsp\}, \{ebmfsp\}\}$

- $\mathcal{M}, w \not\models \mathsf{Kh}_j(h, g)$ but $\mathcal{M}, w \models \langle ebfmsp \not\sim ebmfsp \rangle \mathsf{Kh}_j(h, g)$;
  - **generates new knowledge**
- $\mathcal{M}, w \models \mathsf{Kh}_i(h, g)$ and $\mathcal{M}, w \models \langle ebfmsp \not\sim ebmfsp \rangle \mathsf{Kh}_i(h, g)$.

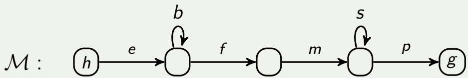
Epistemic updates: Refinement (L_{Ref}) (cont.)



$\mathbb{S}_i = \{\{ebfmsp\}\}$, $\mathbb{S}_j = \{\{ebfmsp\}, \{ebmfsp\}\}$

- $\mathcal{M}, w \not\models \mathsf{Kh}_j(h, g)$ but $\mathcal{M}, w \models \langle ebfmsp \not\sim ebmfsp \rangle \mathsf{Kh}_j(h, g)$;
  - **generates new knowledge**
- $\mathcal{M}, w \models \mathsf{Kh}_i(h, g)$ and $\mathcal{M}, w \models \langle ebfmsp \not\sim ebmfsp \rangle \mathsf{Kh}_i(h, g)$.
  - **preserves knowledge**

Epistemic updates: Refinement ($\mathsf{L}_{\mathsf{Ref}}$) (cont.)



$\mathbb{S}_i = \{\{ebfmsp\}\}$, $\mathbb{S}_j = \{\{ebfmsp\}, \{ebmfsp\}\}$

- $\mathcal{M}, w \not\models \mathsf{Kh}_j(h, g)$ but $\mathcal{M}, w \models \langle ebfmsp \nsim ebmfsp \rangle \mathsf{Kh}_j(h, g)$;
  - **generates new knowledge**
- $\mathcal{M}, w \models \mathsf{Kh}_i(h, g)$ and $\mathcal{M}, w \models \langle ebfmsp \nsim ebmfsp \rangle \mathsf{Kh}_i(h, g)$.
  - **preserves knowledge**

Property:

$\mathsf{L}_{\mathsf{Ref}}$ is more expressive than $\mathsf{L}_{\mathsf{Kh}_i}$.

Arbitrary Refinement (L_{ARef})

Arbitrary Refinement (L_{ARef})

Definition (L_{ARef})

$\mathcal{M}, w \models \langle \not\succ \rangle \varphi$ *iff*$_{def}$
there are $\sigma_1, \sigma_2 \in \text{Act}^*$ s.t. $\mathcal{M}, w \models \langle \sigma_1 \not\succ \sigma_2 \rangle \varphi$.

Arbitrary Refinement ($\mathsf{L}_{\mathsf{ARef}}$)

Definition ($\mathsf{L}_{\mathsf{ARef}}$)

$\mathcal{M}, w \models \langle \not\sim \rangle \varphi$ *iff*$_{def}$
there are $\sigma_1, \sigma_2 \in \mathsf{Act}^*$ s.t. $\mathcal{M}, w \models \langle \sigma_1 \not\sim \sigma_2 \rangle \varphi$.

$\langle \not\sim \rangle \varphi$: *"After it is stated that some pair of plans are distinguishable, φ holds."*

Arbitrary Refinement (L_{ARef})

Definition (L_{ARef})

$\mathcal{M}, w \models \langle \not\sim \rangle \varphi$ *iff*$_{def}$
there are $\sigma_1, \sigma_2 \in \mathsf{Act}^*$ s.t. $\mathcal{M}, w \models \langle \sigma_1 \not\sim \sigma_2 \rangle \varphi$.

$\langle \not\sim \rangle \varphi$: *"After it is stated that some pair of plans are distinguishable, φ holds."*

Property:

L_{ARef} is more expressive than L_{Kh_i}.

Learning How ($\mathsf{L_{Lh}}$)

These new modalities enable us to define a goal-oriented learning modality:

$$\langle \psi, \varphi \rangle_i \chi := \langle \not\sim \rangle (\mathsf{Kh}_i(\psi, \varphi) \wedge \chi)$$

Learning How (L_{Lh})

These new modalities enable us to define a goal-oriented learning modality:

$$\langle \psi, \varphi \rangle_i \chi := \langle \not\sim \rangle (\mathsf{Kh}_i(\psi, \varphi) \wedge \chi)$$

"The agent i can learn how to achieve φ given ψ and after this learning operation takes place, χ holds."

Learning How (L_{Lh})

These new modalities enable us to define a goal-oriented learning modality:

$$\langle \psi, \varphi \rangle_i \chi := \langle \not\sim \rangle (\mathsf{Kh}_i(\psi, \varphi) \wedge \chi)$$

"The agent i can learn how to achieve φ given ψ and after this learning operation takes place, χ holds."

$\mathsf{L}_i(\psi, \varphi) := \langle \psi, \varphi \rangle_i \top$: **learnability test**

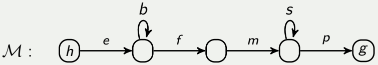
Learning How (L_{Lh})

These new modalities enable us to define a goal-oriented learning modality:

$$\langle \psi, \varphi \rangle_i \chi := \langle \nsim \rangle (\mathsf{Kh}_i(\psi, \varphi) \wedge \chi)$$

"The agent i can learn how to achieve φ given ψ and after this learning operation takes place, χ holds."

$\mathsf{L}_i(\psi, \varphi) := \langle \psi, \varphi \rangle_i \top$: **learnability test**



$\mathbb{S}_i = \{\{ebfmsp\}\},$
$\mathbb{S}_j = \{\{ebfmsp, ebmfsp\}\}$

$$\mathcal{M}, w \models \mathsf{L}_j(h, g)$$

Learning How (L_{Lh})

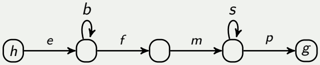
These new modalities enable us to define a goal-oriented learning modality:

$$\langle \psi, \varphi \rangle_i \chi := \langle \not\sim \rangle (\mathsf{Kh}_i(\psi, \varphi) \wedge \chi)$$

"The agent i can learn how to achieve φ given ψ and after this learning operation takes place, χ holds."

$\mathsf{L}_i(\psi, \varphi) := \langle \psi, \varphi \rangle_i \top$: **learnability test**

$\mathcal{M}$:



$$\mathbb{S}_i = \{\{ebfmsp\}\},$$
$$\mathbb{S}_j = \{\{ebfmsp, ebmfsp\}\}$$

$$\mathcal{M}, w \models \mathsf{L}_j(h, g)$$

Property:

L_{Lh} is more expressive than L_{Kh_i}.

Conclusions

Dynamic modalities in the context of *knowing how* logics.

- Ontic updates:
  - Announcement-like and arrow-update-like modalities
  - Axiomatizations over a particular class of models
- Epistemic updates:
  - Refining the perception of an agent regarding her own abilities.
  - Preliminary thoughts and some semantic properties.

Future work

- Study other dynamic operators in this context.
- Explore alternative techniques for obtaining proof systems without a general rule of substitution.
- Find fragments that are axiomatizable via reduction axioms by studying the operators' expressivity.