Epistemic Logic (XI)

Logics of (goal-directed) knowing how

Yanjing Wang
Department of Philosophy, Peking University
Nov. 29th, 2023

wangyanjing.com

Background

Relevance of a formal account of knowing how

- In epistemology:
  - Can knowledge-how be reduced to knowledge-that?
  - Anti-intellectualism: No, Knowledge-how is similar to ability (e.g., [Ryle 49])
  - Intellectualism: Yes, it is reducible based on linguistic formulation (e.g, [Stanley & Williamson 2001])
- In imperfect information games
  - Can a group of agents know how to win the game?
- In automated planning under uncertainty
  - Can an autonomous agent know how to achieve some goal?
- In non-classical logics
  - Some non-classical features are due to implicit epistemic modality (a different story)

Interesting examples in the philosophy literature

Is knowledge-how just ability? Can you say:

- ?I know how to digest.
- ?I know how to lift a 5kg bag.
- ?An infant knows how to ask for food.
- ?A dog knows how to catch a frisbee.
- ?A computer knows how to translate this sentence.
- ?A monkey played Chopin by luck. Does it knows how?
- ?What about a well-trained piano monkey?
- ?A skier escaped the avalanche, he knows how to do it?
- ?A broken-arm pianist knows how to play piano.

Interesting examples in the philosophy literature

Is the experience necessary in knowledge-how? Can you say:

- ?The trainer of an Olympic gym champion knows how to do the champion moves.
- ?You know the rules of Chess thus you know how to play.
- ?You know how to go to the central station even when you have never been there.
- ?A pilot knows how to fly a plane even if he was only trained in (extremely realistic) simulator.
- ?You can cook the right dish using wrong recipe and wrong ingredients (which happen to cancel each other's effects).

A particular type of "knowing how"

Clarifications:

- We do not focus on the philosophical debate between intellectualism and anti-intellectualism. See the collection of 200^{+} papers on the topic at `philpapers.org`.
- We focus on *goal-directed* "knowing how": knowing how to realize a goal, e.g., I know how to go to Beijing; I know how to know the answer; I know how to prove the theorem.
- We do not study "knowing how" in the following senses: I know how turtles reproduce; I know how happy she is; I know how to speak Chinese; I know how to behave at the dinner table....

By different verbs

Vendler's 4 categories of verbs denoting:
states, activities, accomplishments and achievements.

Dowty gives the following examples:

States	Activities	Accomplishments	Achievements
know	run	build	recognize
believe	walk	make a chair	find
have	swim	recover from illness	die

Activity directed, rule directed, goal directed, maintaining goal … see [Gochet 2013]

Towards a logic of knowing how

In AI, “knowing how” to achieve a goal is often treated as being able to (or can) reach a goal (Situation Calculus, ATL, STIT). See two excellent surveys: [Gochet 13] and [Ågotnes, Goranko, Jamroga, Wooldridge 15]. For true know-how, simply combining *know* and *can* does not work.

Two observations inspired by the discussions in philosophy:

- Knowing how to achieve a goal may not entail that you *can* realize the goal now: a chef knows how to make cakes even when there is no sugar. The chef can make a cake, given all the ingredients and equipments are there.
- Even when you can win a lottery by luckily buying the right ticket, it does not mean you know how to win the lottery, since you cannot knowingly guarantee the result.

Ideas from automated planning in AI

The goal of automated planning:

$$\texttt{Uncertain or false} \xrightarrow{a\ plan} \texttt{Certain and true}$$

Sources of uncertianty: initial states, observation power, non-deterministic actions

Uncertainty / Types	Init	Obs	Act	Probability
Classical	no	full	no	no
FOND	no	full	yes	no
MDP	no	full	yes	yes
Conformant	yes	none	yes	no
Contingent	yes	partial	yes	no
POMDP	yes	partial	yes	yes

Central idea from planning

Knowing how to achieve φ roughly means that there is a plan such that the agent knows that executing the plan can guarantee φ.

$$\textit{Roughly} : \exists \sigma K([\sigma]\varphi)$$

Similar to the intellectualism account of know-how by Stanley and Williamson.

Linear plan-based knowing how

Formal language

The language is defined as follows:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \mathsf{Kh}(\varphi, \varphi)$$

$\mathsf{Kh}(\psi, \varphi)$ reads *I know how to ensure that* φ *given* ψ. We define the universal modality $\mathsf{A}\varphi$ as $\mathsf{Kh}(\neg\varphi, \bot)$.

A model is simply a labeled transition system representing the (known) abilities of the agent: $(\mathcal{S}, \Sigma, \mathcal{R}, \mathcal{V})$ where:

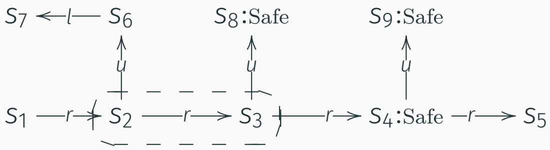
- $\mathcal{S}$ is a non-empty set of states;
- Σ is a non-empty set of actions (not in the language!);
- $\mathcal{R} : \Sigma \to 2^{\mathcal{S} \times \mathcal{S}}$ is a collection of transitions labelled by Σ;
- $\mathcal{V} : S \to 2^{\mathsf{P}}$ is a valuation function.

Lost with a map at hand

Lost with a map at hand

AI Conformant Planning: achieve certainty given uncertainty

A rookie spy sneaking in an enemy building was guided by his headquarters. The communication with the HQ was lost at some point. Now someone spotted him and pulled the alarm. In panic he got lost...



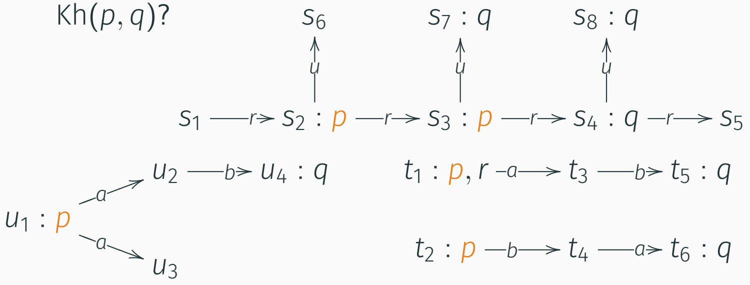
Suppose he has the above map but does not know whether he is at s_2 or s_3 (the *bubble*). Does he *know how* to be safe? Yes! *ru* will make sure his safety eventually.

$\mathrm{Kh}(\psi, \varphi)$ is true iff there is a plan σ (sequence of actions) such that you know that, given ψ, σ is always fully executable and it can get you to some φ world in the end.

First semantics [Wang LORI15, Synthese18]

$\mathcal{M}, s \vDash \mathrm{Kh}(\psi, \varphi) \quad \Leftrightarrow \quad$ *there exists* a $\sigma \in \Sigma^*$ such that *for all* $\mathcal{M}, s' \vDash \psi$:
(1) σ is strongly executable at s', and
(2) *for all* t if $s' \xrightarrow{\sigma} t$ then $\mathcal{M}, t \vDash \varphi$

$\mathrm{Kh}(p, q)$?



$s_1 \vDash \mathrm{Kh}(p, q) \quad u_1 \vDash \neg\mathrm{Kh}(p, q) \quad t_1 \vDash \neg\mathrm{Kh}(p, q)$

$\mathcal{M}, s \vDash \mathrm{A}\varphi \quad \Leftrightarrow \quad \mathrm{Kh}(\neg\varphi, \bot) \quad \Leftrightarrow \quad$ for all $t \in \mathcal{S}, \mathcal{M}, t \vDash \varphi$

Other semantics

Achieving while maintaining [Li & Wang ICLA17]: $\mathrm{Khm}(\psi, \chi, \varphi)$ means knowing how to achieve φ given ψ by only passing χ-states in-between.

$$\begin{array}{lcl} \mathcal{M}, s \vDash \mathrm{Khm}(\psi, \chi, \varphi) & \Leftrightarrow & \textit{there exists a } \sigma \in \Sigma^* \text{ s.t. } \textit{for all } \mathcal{M}, s' \vDash \psi : \\ & & (1)\ \sigma \text{ is } \textit{strongly } \chi\textit{-executable} \text{ at } s', \text{ and} \\ & & (2)\ \textit{for all } t \text{ if } s' \xrightarrow{\sigma} t \text{ then } \mathcal{M}, t \vDash \varphi \end{array}$$

Stopping means achieving [Li Studies in Logic 17]: $\mathrm{Khw}(\psi, \varphi)$ means knowing how to achieve φ when the execution stops.

$$\begin{array}{lcl} \mathcal{M}, s \vDash \mathrm{Khw}(\psi, \varphi) & \Leftrightarrow & \textit{there exists a } \sigma \in \Sigma^* \text{ s.t. } \textit{for all } \mathcal{M}, s' \vDash \psi : \\ & & \textit{for all } t \text{ if } s' \xrightarrow{\sigma}_W t \text{ then } \mathcal{M}, t \vDash \varphi \end{array}$$

where $s \xrightarrow{\sigma}_W t$ means that the execution of σ from s may terminate at t. E.g., $\mathrm{Khw}(p, q)$ is true below.

$$t : q \xleftarrow{a} s : p \xrightarrow{a} w : \xrightarrow{a} u : q$$

Differences of the three Know-how operators

$$
\begin{tikzcd}
 & s_6 & s_7 : q & s_8 : q & \\
s_1 \arrow[r, "r"] & s_2 : {\color{red}p} \arrow[u, "u"] \arrow[r, "r"] & s_3 : {\color{red}p} \arrow[u, "u"] \arrow[r, "r"] & s_4 : q \arrow[u, "u"] \arrow[r, "r"] & s_5
\end{tikzcd}
$$

- $\mathrm{Kh}(p, q)$ holds: *ru* is the only strongly executable witness.
- $\mathrm{Khm}(p, p, q)$ fails: *ru* not only passes p states.
- $\mathrm{Khw}(p, \neg p \wedge \neg q)$ holds, as witnessed by *rrr*. However, $\mathrm{Kh}(p, \neg p \wedge \neg q)$ fails since *rrr* is not strongly executable.

Clearly $\mathrm{Kh}(p, q)$ can be defined by $\mathrm{Khm}(p, \top, q)$.

Proof System for the first semantics [Wang LORI15]

Axioms		Rules	
TAUT	all axioms of propositional logic	MP	$\dfrac{\varphi, \varphi \to \psi}{\psi}$
DISTU	$Ap \land A(p \to q) \to Aq$	NECU	$\dfrac{\varphi}{A\varphi}$
COMPKh	$Kh(p, r) \land Kh(r, q) \to Kh(p, q)$	SUB	$\dfrac{\varphi(p)}{\varphi[\psi/p]}$
EMP	$A(p \to q) \to Kh(p, q)$		
TU	$Ap \to p$		
4KU	$Kh(p, q) \to AKh(p, q)$		
5KU	$\neg Kh(p, q) \to A\neg Kh(p, q)$		

Provable:

PREKh: $Kh(Kh(p, q) \land p, q)$, POSTKh: $Kh(r, Kh(p, q) \land p) \to Kh(r, q)$

MONO: from $\vdash \varphi \to \psi$ infer $\vdash Kh(\chi, \varphi) \to Kh(\chi, \psi)$.

Proof System for the second semantics [Li Wang ICLA17]

Axioms

TAUT	all axioms of propositional logic
DISTU	$\mathsf{A}p \wedge \mathsf{A}(p \to q) \to \mathsf{A}q$
COMPKh	$\mathsf{Kh}(p,o,r) \wedge \mathsf{Kh}(r,o,q) \wedge \mathsf{A}(r \to o) \to \mathsf{Kh}(p,o,q)$
EMP	$\mathsf{A}(p \to q) \to \mathsf{Kh}(p,\bot,q)$
TU	$\mathsf{A}p \to p$
4KU	$\mathsf{Kh}(p,o,q) \to \mathsf{A}\mathsf{Kh}(p,o,q)$
5KU	$\neg\mathsf{Kh}(p,o,q) \to \mathsf{A}\neg\mathsf{Kh}(p,o,q)$
UKhm	$\mathsf{A}(p' \to p) \wedge \mathsf{A}(o \to o') \wedge \mathsf{A}(q \to q') \wedge \mathsf{Kh}(p,o,q) \to \mathsf{Kh}(p',o',q')$
OneKhm	$\mathsf{Kh}(p,o,q) \wedge \neg\mathsf{Kh}(p,\bot,q) \to \mathsf{Kh}(p,\bot,o)$

Rules are MP, NECU, SUB as before.

Proof System for the third semantics [Li SL17]

Axioms		Rules
TAUT	all axioms of propositional logic	MP
DISTU	$Ap \wedge A(p \to q) \to Aq$	NECU
AKh	$A(p' \to p) \wedge A(q \to q') \wedge \mathrm{Khw}(p, q) \to \mathrm{Khw}(p', q')$	SUB
EMP	$A(p \to q) \to \mathrm{Khw}(p, q)$	
TU	$Ap \to p$	
4KU	$\mathrm{Khw}(p, q) \to A\mathrm{Khw}(p, q)$	
5KU	$\neg\mathrm{Khw}(p, q) \to A\neg\mathrm{Khw}(p, q)$	

$s_1 : p \xrightarrow{a} s_3 : r \xrightarrow{b} s_5 : q$ $\mathrm{Khw}(p, r) \wedge \mathrm{Khw}(r, q) \not\to \mathrm{Khw}(p, q)$

$s_2 : p, r \xrightarrow{b} s_4 : q$

See [Xun Wang LORI19] for the fix by a *skipping* semantics.

Example: Canonical model for MCS Γ (for the first semantics)

A single canonical model does not work!

Given a maximal consistent set Γ w.r.t. $\mathbb{SKH}$, let $\Sigma_\Gamma = \{\langle\psi,\varphi\rangle \mid \mathsf{Kh}(\psi,\varphi) \in \Gamma\}$, the canonical model for Γ is $\mathcal{M}^c_\Gamma = \langle \mathcal{S}^c_\Gamma, \Sigma_\Gamma, \mathcal{R}^c, \mathcal{V}^c\rangle$ where:

- $\mathcal{S}^c_\Gamma = \{\Delta \mid \Delta \text{ is a MCS w.r.t. } \mathbb{SKH} \text{ and } \Gamma|_{\mathsf{Kh}} = \Delta|_{\mathsf{Kh}}\}$;
- $\Delta \xrightarrow{\langle\psi,\varphi\rangle}_c \Theta$ iff $\mathsf{Kh}(\psi,\varphi) \in \Gamma$,$\psi \in \Delta$, and $\varphi \in \Theta$;
- $p \in V^c(\Delta)$ iff $p \in \Delta$.

Clearly Γ is a state in $\mathcal{M}^c_\Gamma$.

Handy result: If $\varphi \in \Delta$ for all $\Delta \in S^c_\Gamma$ then $A\varphi \in \Delta$ for all $\Delta \in S^c_\Gamma$

.

Completeness for the first proof system

Lemma (Truth lemma)

For any $\varphi \in \Gamma : \mathcal{M}^c_\Gamma, \Delta \vDash \varphi \iff \varphi \in \Delta$

$\implies$: We do not prove the contrapositive for the Kh case. It requires induction over the length of the witness sequence σ for the truth of $\mathsf{Kh}(\psi, \varphi)$, where `COMPKh` plays an important role.

Theorem

The proof systems are strongly complete w.r.t. the class of all models w.r.t the corresponding semantics.

See Yanjun Li's PhD thesis for decidability of these logics via finite canonical models.

Bisimulations

- The $\exists\forall$-schema in the semantics is similar to the neighborhood semantics for modal logic.
- We are inspired by *monotonic bisimulation* studied by Marc Pauly about Game Logic and Helle Hvid Hansen about monotonic neighborhood modal logic.
- In monotonic bis, wZv implies:
  For any $X \in \nu_{\mathcal{M}}(w)$, there is $X' \in \nu_{\mathcal{N}}(v)$ such that for any $x' \in X'$ there is $x \in X$ such that xZx'.
- The “neighborhood” can be viewed as the collection of the sets that the agent can ensure to achieve by some plans.
- The extra complications are due to the fact that $\mathrm{Kh}(\psi, \varphi)$ is global.

Ideas

We write $U \xrightarrow{\sigma} V$ whenever σ is strongly executable for all $u \in U$, and V is the set of states reachable from U after executing σ.

We write $U \to V$ whenever there is a $\sigma \in \Sigma^*$ such that $U \xrightarrow{\sigma} V$.

$$\begin{tikzcd} s : p \arrow[r, "a"'] \arrow[rr, bend left, "b"] & s_1 : p \arrow[r, "b"'] & s_2 : q \end{tikzcd} \qquad \begin{tikzcd} t : p \arrow[r, "c"'] \arrow[rr, bend left, "c"] & t_1 : q & t_2 : q \arrow[l, "c"] \end{tikzcd}$$

- $X \xrightarrow{\epsilon} X$ thus $X \to X$ for all subsets of the state space.
- $\{s, s_1\} \xrightarrow{b} \{s_2\}$ thus $\{s, s_1\} \to \{s_2\}$.
- $\{t\} \xrightarrow{c} \{t_1, t_2\}$ thus $\{t\} \to \{t_1, t_2\}$.
- The above two models satisfy exactly the same Kh-formulas.

Definition (Kh-Bis [Fervari, Velázquez-Quesada, Wang RSL21])

Let $\mathcal{M} = \langle W, \mathcal{R}, \mathcal{V} \rangle$ and $\mathcal{M}' = \langle W', \mathcal{R}', \mathcal{V}' \rangle$, a non-empty relation $Z \subseteq W \times W'$ is called an Kh-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ if and only if wZw' implies:

Atom: $\mathcal{V}(w) = \mathcal{V}'(w')$.

Kh-**Zig:** for any propositionally definable $U \subseteq W$, if $U \to V$ for some $V \subseteq W$, then there is $V' \subseteq W'$ such that
(i) $Z[U] \to V'$ and
(ii) for each $v' \in V'$ there is a $v \in V$ such that vZv'.

Kh-**Zag:** Symmetric

A-**Zig:** for any v in W there is a v' in W' such that vZv'.

A-**Zag:** Symmetric

We do need the A-Zig and A-Zag in the definition, although A is definable by Kh.

Results

Theorem (Invariance)

Let $\mathcal{M}, w$ and $\mathcal{M}', w'$ be two pointed models, with $\mathcal{M} = \langle W, \mathcal{R}, \mathcal{V} \rangle$ and $\mathcal{M}' = \langle W', \mathcal{R}', \mathcal{V}' \rangle$. If $\mathcal{M}, w \leftrightarroweq_{\mathsf{Kh}} \mathcal{M}', w'$, then $\mathcal{M}, w \equiv_{\mathsf{Kh}} \mathcal{M}', w'$.

Theorem (Hennessy–Milner)

Let $\mathcal{M} = \langle W, \mathcal{R}, \mathcal{V} \rangle$, $\mathcal{M}' = \langle W', \mathcal{R}', \mathcal{V}' \rangle$ be two finite models, $w \in W$ and $w' \in W'$. $\mathcal{M}, w \equiv_{\mathsf{Kh}} \mathcal{M}', w'$ iff $\mathcal{M}, w \leftrightarroweq_{\mathsf{Kh}} \mathcal{M}', w'$.

Khm-bisimulation

Definition

Let $\mathcal{M} = \langle W, \mathcal{R}, \mathcal{V} \rangle$ and $\mathcal{M}' = \langle W', \mathcal{R}', \mathcal{V}' \rangle$ be two relational models. A non-empty relation $Z \subseteq (W \times W')$ is called an Khm-bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ if and only if wZw' implies:

Atom, U-Zig and U-Zag as before.

Khm-**Zig:** for any propositional definable $U \subseteq W$, if $U \xrightarrow{X} V$ for some $X, V \subseteq W$, then there are $X', V' \subseteq W'$ such that *(i)* $Z[U] \xrightarrow{X'} V'$, *(ii)* for each $x' \in X'$ there is a $x \in X$ such that xZx', *(iii)* for each $v' \in V'$ there is a $v \in V$ such that vZv'.

Khm-**Zag:** Symmetric

Khw-bisimulation

Definition

A non-empty relation $Z \subseteq (W \times W')$ is called an Khw-bisimulation if wZw' implies:

Atom, U-Zig and U-Zag as in before.

Khm-**Zig** for any propositional definable $U \subseteq W$, if $U \rightarrow_W V$ for some $V \subseteq W$, then
(i) there is $V' \subseteq W'$ such that $Z[U] \rightarrow_W V'$ and,
(ii) for each $v' \in V'$ there is a $v \in V$ such that vZv'.

Khm-**Zag** Symmetric.

The corresponding invariance results and Hennessy–Milner-like theorems hold.

The logic of Khm is strictly more expressive than the logic of Kh but incomparable with the logic of Khw.

A brief summary for now

Features of the logics of *knowing how* so far:

- Global knowledge
- Conditional modality
- No explicit *knowing that* operator
- Based on linear plans
- No observation during plan executions

What about a local notion with explicit know-that operator?

Strategy-based knowing how

Language and Models [Fervari, Herzig, Li, Wang IJCAI17]

$$\varphi := p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \mathsf{K}\varphi \mid \mathsf{Kh}\varphi$$

Note that we have the explicit know-that operator K in the language.

A model is a labeled transition system with an epistemic relation: $\langle \mathcal{S}, \Sigma, \mathcal{R}, \sim, \mathcal{V} \rangle$ where:

- $\langle \mathcal{S}, \Sigma, \mathcal{R}, \mathcal{V} \rangle$ is a labelled transition system as before.
- $\sim \subseteq S \times S$ is an equivalence relation (bubbles everywhere).

Example (reflexive arrows are omitted)

$$\begin{tikzcd}
s : \neg lightOn \arrow[r, "\texttt{observe}"] \arrow[d, dotted, no head] & t : \neg lightOn \arrow[d, leftrightarrow, "\texttt{flip}"] \\
u : lightOn \arrow[r, "\texttt{observe}"] & v : lightOn
\end{tikzcd}$$

Uniformly executable strategy

- The agent's *epistemic state* at s: $[s] = \{t \ : \ s \sim t\}$
- A *strategy* is a partial function σ from epistemic states to actions.
- σ is *uniformly executable* if $\sigma([s])$ executable at every $s' \in [s]$. Empty strategy is always uniformly executable.

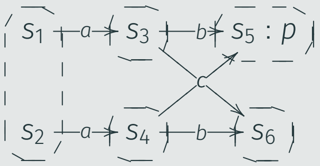
Example

$$
\begin{tikzcd}
s : \neg lightOn \arrow[r, "\texttt{observe}"] \arrow[d, dash, dotted] & t : \neg lightOn \arrow[d, <->, "\texttt{flip}"] \\
u : lightOn \arrow[r, "\texttt{observe}"] & v : lightOn
\end{tikzcd}
$$

$\sigma' = \{\{s, u\} \mapsto \texttt{observe}, \{t\} \mapsto \texttt{flip}\}$ is uniformly executable.

Example

You are not sure about the exact cause of some symptom, thus you have some uncertainty about your current state represented by the bubble over s_1 and s_2 below. After taking the test a, you will know the cause thus the current state. Taking the treatments b or c may have different effects depending on your actual state: it may cure your symptom (p) or not ($\neg p$). Now given the initial uncertainty about s_1 and s_2, do you know how to make sure p?



There is no sensible uniform *linear plan* in this scenario

Semantics

- $M, s \models \mathrm{K}\varphi$ if $M, t \models \varphi$ for every t such that $s \sim t$

- $M, s \models \mathrm{Kh}\varphi$ if there exists a *uniformly executable strategy* σ such that:
  1. all complete executions starting from s terminate
  2. for every final epistemic state $[t]$ after executing σ, all $t' \in [t]$ satisfies φ.

Example (M is depicted as follows)

$$\begin{tikzcd} s : \neg p \arrow[loop above, "a"] \arrow[r, "a"] & t : p \end{tikzcd}$$

$\sigma = \{\{s\} \mapsto a\}$ is uniformly executable, but there is an infinite execution of σ starting from s. $\mathcal{M}, s \not\models \mathrm{Kh}p$.

A complete axiomatization

TAUT	all axioms of propositional logic	MP	$\dfrac{\varphi, \varphi \to \psi}{\psi}$
DISTK	$\mathsf{K}p \land \mathsf{K}(p \to q) \to \mathsf{K}q$	NECK	$\dfrac{\varphi}{\mathsf{K}\varphi}$
T	$\mathsf{K}p \to p$	MonoKh	$\dfrac{\varphi \to \psi}{\mathsf{Kh}\varphi \to \mathsf{Kh}\psi}$
4	$\mathsf{K}p \to \mathsf{K}\mathsf{K}p$	SUB	$\dfrac{\varphi(p)}{\varphi[\psi/p]}$
5	$\neg\mathsf{K}p \to \mathsf{K}\neg\mathsf{K}p$		
AxKtoKh	$\mathsf{K}p \to \mathsf{Kh}p$		
AxKhtoKhK	$\mathsf{Kh}p \to \mathsf{Kh}\mathsf{K}p$		
AxKhtoKKh	$\mathsf{Kh}p \to \mathsf{K}\mathsf{Kh}p$		
AxKhKh	$\mathsf{Kh}\mathsf{Kh}p \to \mathsf{Kh}p$		
AxKhbot	$\neg\mathsf{Kh}\bot$		

Properties

- Kh is not normal

$$\not\models \mathsf{Kh}p \wedge \mathsf{Kh}(p \to q) \to \mathsf{Kh}q$$

- negative introspection provable:

$$\models \neg\mathsf{Kh}p \to \mathsf{K}\neg\mathsf{Kh}p$$

- sequences of modal operators reduce:

$$\models \mathsf{KK}p \leftrightarrow \mathsf{K}p$$

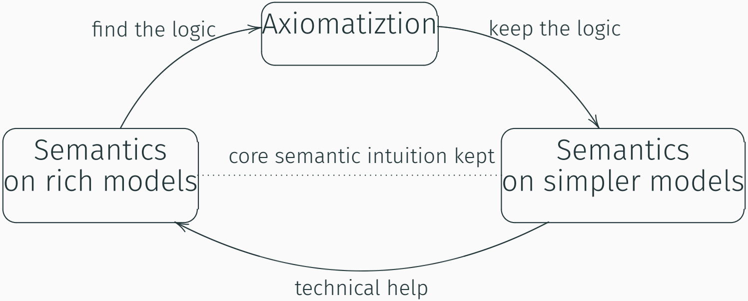
$$\models \mathsf{KhKh}p \leftrightarrow \mathsf{Kh}p$$

$$\models \mathsf{KhK}p \leftrightarrow \mathsf{Kh}p$$

$$\models \mathsf{KKh}p \leftrightarrow \mathsf{Kh}p$$

- sound and complete: soundness of $\mathsf{KhKh}p \to \mathsf{Kh}p$ is highly non-trivial!
- decidable: we can construct a finite canonical model.

Again, we are seeking for a simplified semantics



Alternative neighbourhood semantics [Li& Wang Synthese 21]

The model class $\mathcal{C}$ consists of all *mixed* epistemic models $\langle S, \sim, N, V\rangle$ satisfying the following conditions.

- For all $s \in W$, any $X, Y \subseteq W, X \in N(s)$ implies $Y \in N(s)$ ($\mathtt{MonoKh}$).
- For all $s \in W$, $\emptyset \notin N(s)$ ($\mathtt{AxKhbot}$)
- For any $s, t \in W$, $s \sim t$ implies $N(s) = N(t)$ ($\mathtt{AxKhtoKKh}$).
- For all $s \in W$, $[s] \in N(s)$ ($\mathtt{AxKtoKh}$).
- For all $s \in W$ and $X \subseteq W$, if $X \in N(s)$ then $Y = \{t \mid [t] \subseteq X\} \in N(s)$ ($\mathtt{AxKhtoKhK}$)
- For all $s \in W$ and $X, Y \subseteq W$, if $X \in N(s)$, and $Y \in N(t)$ for all $t \in X$, we will have $Y \in N(s)$. ($\mathtt{AxKhKh}$).

The neighborhoods are forcible sets by strategies .

A unified framework

A general framework [Li & Wang AIJ21]

How to unify those different notions of planning based Kh?

The idea: use a programming language to specify all kinds of plans:

$$\pi \quad ::= \quad \epsilon \mid a \mid (\pi;\pi) \mid \texttt{if } \varphi \texttt{ then } \pi \texttt{ else } \pi \mid \texttt{while } \varphi \texttt{ do } \pi$$

For example, you may have an unbounded plan to go out of a grand maze: always turn left (l) if you cannot go forward (f) anymore, until you know you are out of the maze:

$$\texttt{while } \neg \mathsf{K}\, out \texttt{ do } (\texttt{while } \langle f \rangle \top \texttt{ do } f); l).$$

Different types of plans

- Simple plan [Naumov&Tao2018]:

$$\sigma ::= \epsilon \mid a$$

- Linear plans [Wang2015]:

$$\sigma ::= \epsilon \mid a \mid \sigma ; \sigma$$

- Skipable linear plans [Wang2019], i.e. linear plan with executability tests to skip:

$$\sigma ::= \epsilon \mid \texttt{if } \langle a \rangle \top \texttt{ then } a \texttt{ else } \epsilon \mid \sigma ; \sigma$$

- Stoppable linear plans [Li16], i.e. linear plan with executability tests to stop:

$$\sigma ::= \epsilon \mid \texttt{if } \langle a \rangle \top \texttt{ then } a ; \sigma \texttt{ else } \epsilon$$

Different types of plans

- Unbounded linear plans [Levesque2005], i.e., linear plans with while-loops:

$$\sigma ::= \epsilon \mid a \mid \sigma;\sigma \mid \texttt{while}\ \varphi\ \texttt{do}\ \sigma$$

- Conditional plans, i.e., linear plan with conditionals:

$$\sigma ::= \epsilon \mid a \mid \sigma;\sigma \mid \texttt{if}\ \varphi\ \texttt{then}\ \sigma\ \texttt{else}\ \sigma$$

- Knowledge-based plans [Lang12], i.e. any program with epistemic conditions:

$$\sigma ::= \epsilon \mid a \mid \sigma;\sigma \mid \texttt{if}\ \mathsf{K}\varphi\ \texttt{then}\ \sigma\ \texttt{else}\ \sigma \mid \texttt{while}\ \mathsf{K}\varphi\ \texttt{do}\ \sigma$$

where φ is an EAL formula (with both $[a]$ and K). We have a structural operational semantics for these programs to compute the execution sequences.

Definition 10 (Structural operational semantics (SOS) for $\texttt{Prg}$). *Given a model $\mathcal{M}$, the semantics of programs with respect to $\mathcal{M}$ is defined by the following rules about the transition relation in the shape of $\langle \pi, s\rangle \overset{z}{\Rightarrow} \gamma$, where $\pi \in \texttt{Prg}$, $s \in S_{\mathcal{M}}$, $z \in \mathbf{A} \cup \{\epsilon\}$, and γ is either in the form of $\langle \pi', s'\rangle$ or s', where $\pi' \in \texttt{Prg}$ and $s' \in S_{\mathcal{M}}$. We also call the rules without the premises* axioms.

$$\textit{Atom}\ \frac{}{\langle a, s\rangle \overset{a}{\Rightarrow} t}\ (t \in Q(s)(a)) \qquad \textit{Skip}\ \frac{}{\langle \epsilon, s\rangle \overset{\epsilon}{\Rightarrow} s}$$

$$\textit{CompoL}\ \frac{\langle \pi_1, s\rangle \overset{z}{\Rightarrow} \langle \pi_1', s'\rangle}{\langle \pi_1; \pi_2, s\rangle \overset{z}{\Rightarrow} \langle \pi_1'; \pi_2, s'\rangle} \qquad \textit{CompoR}\ \frac{\langle \pi_1, s\rangle \overset{z}{\Rightarrow} s'}{\langle \pi_1; \pi_2, s\rangle \overset{z}{\Rightarrow} \langle \pi_2, s'\rangle}$$

$$\textit{IfthenP}\ \frac{}{\langle \texttt{if}\ \varphi\ \texttt{then}\ \pi_1\ \texttt{else}\ \pi_2, s\rangle \overset{\epsilon}{\Rightarrow} \langle \pi_1, s\rangle}\ (\textit{if } \mathcal{M}, s \vDash \varphi)$$

$$\textit{IfthenN}\ \frac{}{\langle \texttt{if}\ \varphi\ \texttt{then}\ \pi_1\ \texttt{else}\ \pi_2, s\rangle \overset{\epsilon}{\Rightarrow} \langle \pi_2, s\rangle}\ (\textit{if } \mathcal{M}, s \nvDash \varphi)$$

$$\textit{While}\ \frac{}{\langle \texttt{while}\ \varphi\ \texttt{do}\ \pi, s\rangle \overset{\epsilon}{\Rightarrow} \langle \texttt{if}\ \varphi\ \texttt{then}\ (\pi; \texttt{while}\ \varphi\ \texttt{do}\ \pi)\ \texttt{else}\ \epsilon, s\rangle}$$

We say γ is terminal if $\gamma = t$ for some state t, we say $\gamma = \langle \pi, s\rangle$ is stuck if there is no γ' and z such that $\gamma \overset{z}{\Rightarrow} \gamma'$ according to the SOS with respect to the model.

Definition (ELKh Language)

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \mathsf{K}\varphi \mid \mathsf{Kh}\varphi$$

A model is as in the previous section:$\langle \mathcal{S}, \Sigma, \mathcal{R}, \mathcal{V} \rangle$ and the semantics is given by:

$$\boxed{\begin{array}{lcl} \mathcal{M}, s \vDash_x \mathsf{Kh}\varphi & \iff & \text{there is an x-plan } \sigma \text{ such that for all } s' \sim s \\ & & \text{1. } \sigma \text{ is strongly executable on } s'; \\ & & \text{2. } \mathcal{M}, t \vDash_x \varphi \text{ for each } s' \xrightarrow{\sigma} t. \end{array}}$$

The *strong executability* here oughly means that the plan is always executable and it will definitely terminate.

Main Result [Li & Wang AIJ 21]

The following proof system is sound and complete for all the semantics based on ten different notions of plans.

TAUT	all axioms of propositional logic	MP	$\dfrac{\varphi, \varphi \to \psi}{\psi}$
DISTK	$\mathsf{K}p \land \mathsf{K}(p \to q) \to \mathsf{K}q$	NECK	$\dfrac{\varphi}{\mathsf{K}\varphi}$
T	$\mathsf{K}p \to p$	MonoKh	$\dfrac{\varphi \to \psi}{\mathsf{Kh}\varphi \to \mathsf{Kh}\psi}$
4	$\mathsf{K}p \to \mathsf{KK}p$	SUB	$\dfrac{\varphi(p)}{\varphi[\psi/p]}$
5	$\neg\mathsf{K}p \to \mathsf{K}\neg\mathsf{K}p$		
AxKtoKh	$\mathsf{K}p \to \mathsf{Kh}p$		
AxKhtoKKh	$\mathsf{Kh}p \to \mathsf{KKh}p$		
AxKhbot	$\neg\mathsf{Kh}\bot$		

Where are the $\mathsf{KhKh}\varphi \to \mathsf{Kh}\varphi$ and $\mathsf{Kh}\varphi \to \mathsf{KhK}\varphi$? The first is back when restricted to finite models (and knowledge-based plans) and the latter depends on the property of perfect recall.

Know-how based planning [Li & Wang TARK21]

So far, we focused on planning-based know-how.

epistemic planning $\Longleftarrow$ knowing how

planning with a goal φ = model checking $\mathsf{Kh}\varphi$

We can do much more

epistemic planning

$\neq$

planning with know-that goals e.g., $\mathsf{K}_i(\varphi \land \neg \mathsf{K}_j \varphi)$.

Higher-order epistemic planning

The notion of planning (know-how) is in the object language rather than the meta language.

(secret of success of modal logic):

- $Kh_1(\varphi \land Kh_1 \neg\varphi)$
- $Kh_1(\varphi \land \neg Kh_1 \neg\varphi)$
- $Kh_1(\varphi \land \neg Kh_2 \neg\varphi \land K_2 Kh_1 \neg\varphi)$
- $\neg Kh_1 \varphi \land \neg Kh_2 \varphi \land Kh_1 Kh_2 \varphi$
- $Kh_1 Kh_2 Kh_1 Kh_2 \varphi$

Meta-level Epistemic Planning

In the literature of planning, people mainly focused on planning in concrete scenarios.

We would also like to handle the following meta-level planning scenarios (i.e., planning at some abstract level):

Example (Collaboration)

I know that once you know the lemma is true, then you know how to prove the theorem, and I indeed know how to prove the lemma, thus I know how to let you know how to prove the theorem.

As the planner, I just need to know who knows how to do what.

For all these, we do need a more powerful logic

Features to be added

For the purpose of know-how-based planning:

- Multi-agent
- Local + Global knowledge-how
- Perfect recall
- Model checking algorithm: for concrete higher-order planning
- Theorem proving /satisfiability checking algorithm: for meta-level planning

One kind of shrimps, two ways to cook

image source: `m.ecook.cn/caipu/259740580`

Model checking and satisfiability checking

Ongoing work: multiple agents and a temporal modality

Definition (ELKhB Language)

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid \mathsf{K}_i\varphi \mid \mathsf{Kh}_i\varphi \mid \Box\varphi$$

$\Box\varphi$ reads "forever φ in the future", e.g., $\mathsf{K}_j\Box(\mathsf{K}_i p \to \mathsf{Kh}_i q)$ says agent j knows that whenever i knows p, it knows how to achieve q. A model is a multi-agent *epistemic transition system* with perfect recall: $\langle W, \{\sim_i \mid i \in \mathbf{I}\}, \{\Sigma_i \mid i \in \mathbf{I}\}, \{Q(a) \mid a \in \Sigma = \bigcup_{i\in\mathbf{I}} \Sigma_i\}, V\rangle$.

$\mathcal{M}, s \vDash \mathsf{Kh}_i\varphi$	$\iff$	there is a π (knowledge-based plan for i)
		such that for all $s' \sim_i s$:
		1. π is strongly executable on s';
		2. $\mathcal{M}, t \vDash \varphi$ for each $t \in Q(\pi)(s')$
$\mathcal{M}, s \vDash \Box\varphi$	$\iff$	for each $\sigma \in \Sigma^*$, $\mathcal{M}, t \vDash \varphi$ for each $t \in Q(\sigma)(s)$.

where knowledge-based plans for i ($\mathtt{Prg}(i)$) is defined as:

$$\pi \quad ::= \quad \epsilon \mid a \mid (\pi;\pi) \mid \mathbf{if}\ \mathsf{K}_i\varphi\ \mathbf{then}\ \pi\ \mathbf{else}\ \pi$$

Example goal formulas

For agent 1, model checking $\mathrm{Kh}_1 \mathit{Goal}$, where *Goal* can be:

- $\varphi \wedge \mathrm{Kh}_1 \neg\varphi$
- $\varphi \wedge \neg\mathrm{Kh}_1 \neg\varphi$
- $\varphi \wedge \neg\mathrm{Kh}_2 \neg\varphi \wedge \mathrm{K}_2\mathrm{Kh}_1 \neg\varphi$
- $\neg\mathrm{Kh}_1 \varphi \wedge \neg\mathrm{Kh}_2 \varphi \wedge \mathrm{Kh}_1\mathrm{Kh}_2 \varphi$
- $\mathrm{Kh}_2\mathrm{Kh}_1\mathrm{Kh}_2 \varphi$
- $p \wedge \hat{K}_1 \mathrm{Kh}_2 p \wedge \hat{K}_1 \mathrm{Kh}_3 p$
- $\Box\Diamond\mathrm{Kh}_2 \Box p$
- $\Box(\neg\mathrm{Kh}_2 p \rightarrow \mathrm{Kh}_1\mathrm{Kh}_2 p)$: always ready to help

Moreover, we do not need to only check Kh-formulas, we can check any ELKhB formula!

Perfect recall

Given a model $\mathcal{M}$, if $(s_1, s_2) \in Q(a)$ for some $a \in \Sigma$ and $s_2 \sim_i s_4$ then there must exist a state s_3 and an action $b \in \Sigma$ such that

- $s_1 \sim_i s_3$;
- $(s_3, s_4) \in Q(b)$;
- if $a \in \Sigma_i$ then $a = b$.

$$
\begin{tikzcd}
s_1 \arrow[d, "a"] & \\
s_2 \arrow[r, dash, dotted, "i"] & s_4
\end{tikzcd}
\quad \xrightarrow{\texttt{PR}(a)} \quad
\begin{tikzcd}
s_1 \arrow[d, "a"] \arrow[r, dash, dotted, "i"] & s_3 \arrow[d, "b"] \\
s_2 \arrow[r, dash, dotted, "i"] & s_4
\end{tikzcd}
$$

Proof system $\mathbb{SELKHB}$

Axioms

TAUT	all axioms of propositional logic
DISTK	$\mathsf{K}_i(p \to q) \to (\mathsf{K}_i p \to \mathsf{K}_i q)$
T	$\mathsf{K}_i p \to p$
4	$\mathsf{K}_i p \to \mathsf{K}_i \mathsf{K}_i p$
5	$\neg \mathsf{K}_i p \to \mathsf{K}_i \neg \mathsf{K}_i p$
AxKtoKh	$\mathsf{K}_i p \to \mathsf{Kh}_i p$
AxKhtoKhK	$\mathsf{Kh}_i p \to \mathsf{Kh}_i \mathsf{K}_i p$
AxKhtoKKh	$\mathsf{Kh}_i p \to \mathsf{K}_i \mathsf{Kh}_i p$
DIST$\Box$	$\Box(p \to q) \to (\Box p \to \Box q)$
T$\Box$	$\Box p \to p$
4$\Box$	$\Box p \to \Box\Box p$
Kh$\Diamond$	$\mathsf{Kh}_i \mathsf{K}_i p \to \Diamond \mathsf{K}_i p$
PR	$\mathsf{K}_i \Box \varphi \to \Box \mathsf{K}_i \varphi$
K$\Box$	$\mathsf{K}_i \Box(\mathsf{K}_i p \to \mathsf{Kh}_i q) \to (\mathsf{Kh}_i p \to \mathsf{Kh}_i q)$

Rules

MP	$\dfrac{\varphi, \varphi \to \psi}{\psi}$
SUB	$\dfrac{\varphi}{\varphi[\psi/p]}$
NECK	$\dfrac{\varphi}{\mathsf{K}_i \varphi}$
NEC$\Box$	$\dfrac{\varphi}{\Box \varphi}$

Example of meta-planning

Recall the collaboration story.

Given $\mathsf{Kh}_1\mathsf{K}_2 p$ and $\mathsf{K}_1\Box(\mathsf{K}_2 p \to \mathsf{Kh}_2 q)$, we can derive $\mathsf{Kh}_1\mathsf{Kh}_2 q$.

Assume $\mathsf{K}_1\Box(\mathsf{K}_2 p \to \mathsf{Kh}_2 q)$

We have $\mathsf{K}_1\mathsf{K}_1\Box(\mathsf{K}_2 p \to \mathsf{Kh}_2 q)$ by Axiom `4`.

Then by `PR`, we have $\mathsf{K}_1\Box\mathsf{K}_1(\mathsf{K}_2 p \to \mathsf{Kh}_2 q)$.

Thus $\mathsf{K}_1\Box(\mathsf{K}_1\mathsf{K}_2 p \to \mathsf{K}_1\mathsf{Kh}_2 q)$ by normality of K_1.

By `AxKtoKh`, we have $\mathsf{K}_1\Box(\mathsf{K}_1\mathsf{K}_2 p \to \mathsf{Kh}_1\mathsf{Kh}_2 q)$.

By `K`$\Box$, we have $\mathsf{K}_1\Box(\mathsf{K}_1\mathsf{K}_2 p \to \mathsf{Kh}_1\mathsf{Kh}_2 q) \wedge \mathsf{Kh}_1\mathsf{K}_2 p \to \mathsf{Kh}_1\mathsf{Kh}_2 p$.

Finally, by the assumption $\mathsf{Kh}_1\mathsf{K}_2 p$ we have $\mathsf{Kh}_1\mathsf{Kh}_2 q$.

Technical results

- PTIME model checking algorithm for the full language over explicit models (*more for free*).
- Complete axiomatization of the logic over finite models with perfect recall w.r.t. knowledge-based plans.
- The logic is decidable.

Approach of group know-how

Starting from Naumov and Tao [AIJ18] with also lots of variants.

- Coalition-based know-how: $H_C\varphi$
- Based on distributed knowledge $K_C\varphi$
- Based on one-step joint action
- Crucially rely on the cooperation axiom: $H_C\varphi \wedge H_D\psi \to H_{C\cup D}(\varphi \wedge \psi)$ (given C and D are disjoint).

The cooperation axiom does not work if the plan is no longer one step. We are working on it together (with Yanjun and Bin)!

Further directions

Conclusions and Future directions

We present some planning-based semantic notions of single-agent knowing how and use the bundle modality Kh without breaking it into quantifiers and usual modalities. We also discuss the logics. This is only the beginning of an interesting story.

- DEL based knowing how. [Li & Wang LORI19]
- Activity based know-how [with Yichen]
- Update of knowledge-how [Areces et al 23]
- Commonly knowing how
- Model theoretical questions
- Probabilistic knowing how
- Logical omniscience of knowing how
- Philosophical questions regarding the axioms
- Modelling Lingdao...