



## EPISTEMIC LOGIC XIII

### TOWARDS A LOGIC OF KNOWING WHO

---

Yanjnig Wang  
Department of Philosophy  
Dec. 19th 2018

Background

Epistemic logic with assignments

## BACKGROUND

---

## IS IT NONSENSE?

BEIJING, May 7 (Xinhua) – Chinese Premier Li Keqiang harshly criticized the country’s excessive regulation on Wednesday, ridiculing that a citizen was even asked to prove “**your mother is your mother**” when obtaining a government permit.



“How ridiculous! The citizen only intended to go travelling abroad and take a vacation,” Li was quoted as saying Wednesday at a State Council executive meeting on the cabinet’s website.

# IN SOCIAL NETWORKS, IT IS SERIOUS TO PROVE YOU ARE YOU



How can I know the one who sent me the friend request is indeed Jeremy?

Yanjing: Tell me the name of the bar that we went to last time!

“Jeremy”: ....

In fact, there are several “popular” scams in China featuring “smart” use of misleading identities.

Moreover, one can have many names/user names online. Names can be considered as privacy due to search engines and automatic password crackers.

## IDENTITY IS NOT COMMON KNOWLEDGE

In standard epistemic logic, agents are identified with their names, and thus their identities are implicit common knowledge. Knowledge operator  $K_a$  is indexed by a name (rigid designator), which corresponds to a relation  $R_a$  in the models.

*One dark and stormy night, Adam was attacked and killed. His assailant, Bob, ran away, but was seen by a passer-by, Charles, who witnessed the crime from start to finish. This led quickly to Bob's arrest. Local news picked up the story, and that is how Dave heard it the next day from the radio, over breakfast.*

Now, in one sense we can say that both Charles and Dave know that Bob killed Adam. But clearly there is a difference in what they know about just this fact.

How do we tell the difference in epistemic logic?

## AN EXAMPLE BY GROVE (1995)

A broken robot (named  $a$ ) is calling for help from the maintenance robot (named  $b$ ) by sending requests. To plan further actions, the broken robot needs to know the maintenance robot knows that it needs help. Does  $K_a K_b H(a)$  express this, if the names  $a, b$  are not commonly known?

- (i)  $a$  knows that a robot named ' $b$ ', whatever it is, knows that a robot named ' $a$ ', whatever it is, needs help. (*de dicto*)
- (ii)  $a$  knows that the maintenance robot knows that a robot named ' $a$ ', whatever it is, needs help.
- (iii)  $a$ , the broken robot, knows that the maintenance robot knows that it, i.e. the broken robot, needs help. (*de re*)

## EXISTING WORK

- Grove (1995) proposed various semantics for epistemic logic based on model with world-agent pairs.
- Fitting, Thalmann, & Voronkov (2001) proposed *term modal logic* (constants are rigid). Even the propositional part is undecidable (Padmanabha & Ramanujam 2016). Decidable fragments are discussed in (Orlandelli & Corsi 2018) and (Padmanabha & Ramanujam 2018).
- Kooi (2007) allows non-rigid constants and introduces *dynamic term modal logic*. Again, it is highly undecidable.
- Corsi and Orlandelli (2013) proposed a first-order epistemic logic with  $|t : \frac{t_1 \dots t_n}{x_1 \dots x_n}|$  operators based on counterpart semantics.
- Holliday and Perry (2014) use of a version of FOIL with perspective changes to handle the multi-agent case.

Our approach is a **minimalistic one**: a small fragment of dynamic term modal logic by Kooi suffices, which can be understood by even strangers to those *de dicto /de re* discussions.

# EPISTEMIC LOGIC WITH ASSIGNMENTS

---

### Definition (Language ELAS)

Given a set of variables  $\mathbf{X}$ , a set of names  $\mathbf{N}$ , and a set of  $\mathbf{P}$  of predicate symbols:

$$t ::= x \mid a$$

$$\varphi ::= (t \approx t) \mid Pt \dots t \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_t\varphi \mid [x := t]\varphi$$

where  $x \in \mathbf{X}$ ,  $a \in \mathbf{N}$ ,  $P \in \mathbf{P}$ . We write  $\langle x := t \rangle\varphi$  as the abbreviation of  $\neg[x := t]\neg\varphi$ .

$[x := t]\varphi$ : “ $\varphi$  holds after assigning the **current value** of  $t$  to  $x$ .”

Names are **not** rigid in the epistemic setting but variables are, according to the semantics to be introduced later.

Definition (Epistemic model  $\mathcal{M}$  for ELAS)

A tuple  $\langle W, I, R, \rho, \eta \rangle$  where:

- $W$  is a non-empty set of possible worlds.
- $I$  is a non-empty set of agents (the constant domain)
- $R : I \rightarrow 2^{W \times W}$  and  $R_i$  is an equivalence relation over  $W$ .
- $\rho : \mathbf{P} \times W \rightarrow \bigcup_{n \in \omega} 2^{I^n}$  and  $\rho$  assigns each  $n$ -ary predicate on each world an  $n$ -ary relation on  $I$ .
- $\eta : \mathbf{N} \times W \rightarrow I$ .

$\sigma : \mathbf{X} \rightarrow I$  is the assignment for variables. We will consider pointed models with variable assignment:  $\mathcal{M}, w, \sigma$ .

## Definition (Semantics)

$$\begin{aligned} \mathcal{M}, w, \sigma \models K_t \varphi &\Leftrightarrow \mathcal{M}, v, \sigma \models \varphi \text{ for all } v \text{ s.t. } wR_{\sigma_w(t)}v \\ \mathcal{M}, w, \sigma \models [x := t]\varphi &\Leftrightarrow \mathcal{M}, w, \sigma[x \mapsto \sigma_w(t)] \models \varphi \end{aligned}$$

$$\text{where } \sigma_w(t) = \begin{cases} \sigma(t) & t \in X \\ \eta(t, w) & t \in \mathbf{N} \end{cases}.$$

$K_t$  formulas are evaluated based on what  $t$  refers to on the **current world**.

An **ELAS** formula is *valid* (over epistemic models) if it holds on all the (epistemic) models with assignments  $\mathcal{M}, s, \sigma$ .

## UNDERSTANDING $[x := t]$ BY TRANSLATION

We translate **ELAS** into a (2-sorted) first-order language with a **ternary relation symbol**  $R$  for the accessibility relation, a **function symbol**  $f_a$  for each name  $a$ , and an  $n + 1$ -ary relation symbol  $Q^P$  for each predicate symbol  $P$ .

$$\text{Tr}_w(x) = x \quad \text{Tr}_w(a) = f_a(w)$$

$$\text{Tr}_w(t \approx t') = \text{Tr}_w(t) \approx \text{Tr}_w(t') \quad \text{Tr}_w(P\bar{t}) = Q_P(w, \text{Tr}_w(\bar{t}))$$

$$\text{Tr}_w(\neg\psi) = \neg\text{Tr}_w(\psi) \quad \text{Tr}_w(\varphi \wedge \psi) = \text{Tr}_w(\varphi) \wedge \text{Tr}_w(\psi).$$

$$\text{Tr}_w(K_t\psi) = \forall v(R(w, \text{Tr}_w(t), v) \rightarrow \text{Tr}_v(\psi))$$

$$\text{Tr}_w([x := t]\psi) = \exists x(x \approx \text{Tr}_w(t) \wedge \text{Tr}_w(\psi)) = \forall x(x \approx \text{Tr}_w(t) \rightarrow \text{Tr}_w(\psi))$$

(Given  $x \neq t$ )

We can define free and bound occurrences of variables accordingly to the first-order translation.

## WHAT CAN WE EXPRESS?

- $[x := b]K_a x \approx b$ : *a* knows who *b* is. We abbreviate it as  $K_a b$ .
- $\neg K_a a$ : *a* does not know he is called *a* (e.g. the most foolish person may not know that he is the most foolish person).
- $b \approx c \wedge K_a b \wedge \neg K_a c$ : *a* knows who *b* is but does not know who *c* is, although they are two names of the same person.
- $[x := a][y := b](K_c M(y, x) \wedge \neg K_c (a \approx x \wedge b \approx y))$ : Charles knows who killed whom that night but does not know the names of the murderer and the victim.
- $K_d M(b, a) \wedge \neg K_d a \wedge \neg K_d b$ : Dave knows that a person named Bob killed a person named Adam without knowing who they are.

## WHAT CAN WE EXPRESS?

- (i)  $a$ , the broken robot, knows that the robot named ' $b$ ' knows that the robot named ' $a$ ' needs help.  $K_a K_b H(a)$
- (ii)  $a$  knows that the maintenance robot knows that the robot named ' $a$ ', whatever it is, needs help.  $[y := b] K_a K_y H(a)$
- (iii)  $a$  knows that the maintenance robot knows that it, i.e. the broken robot, needs help.  $[x := a] [y := b] K_x K_y H(x)$

## VALID AND INVALID FORMULAS

valid  $x \approx y \rightarrow K_t x \approx y, x \not\approx y \rightarrow K_t x \not\approx y.$

invalid  $x \approx a \rightarrow K_t x \approx a, x \not\approx a \rightarrow K_t x \not\approx a, x \approx a \rightarrow K_a x \approx a$

valid  $K_x \varphi \rightarrow K_x K_x \varphi, \neg K_x \varphi \rightarrow K_x \neg K_x \varphi, K_t \varphi \rightarrow \varphi.$

invalid  $K_t \varphi \rightarrow K_t K_t \varphi, \neg K_t \varphi \rightarrow K_t \neg K_t \varphi$

valid  $[x := y] \varphi \rightarrow \varphi[y/x]$  ( $\varphi[y/x]$  is admissible)

invalid  $[x := a] \varphi \rightarrow \varphi[a/x]$

valid  $x \approx a \rightarrow (K_x \varphi \rightarrow K_a \varphi)$

invalid  $[x := b] K_a \varphi \rightarrow K_a [x := b] \varphi$

valid  $[x := y] K_a \varphi \rightarrow K_a [x := y] \varphi$

# SYSTEM SELAS

<b>TAUT</b>	all axioms of PL	<b>ID</b>	$t \approx t$
<b>DISTK</b>	$K_t(\varphi \rightarrow \psi) \rightarrow (K_t\varphi \rightarrow K_t\psi)$		
<b>Tx</b>	$K_x\varphi \rightarrow \varphi$	<b>SUBP</b>	$\bar{t} \approx \bar{t}' \rightarrow (P\bar{t} \leftrightarrow P\bar{t}')$ ( $P$ can be $\approx$ )
<b>4x</b>	$K_x\varphi \rightarrow K_xK_x\varphi$	<b>SUBK</b>	$t \approx t' \rightarrow (K_t\varphi \leftrightarrow K_{t'}\varphi)$
<b>5x</b>	$\neg K_x\varphi \rightarrow K_x\neg K_x\varphi$	<b>SUBAS</b>	$t \approx t' \rightarrow ([x := t]\varphi \leftrightarrow [x := t']\varphi)$
<b>RIGIDP</b>	$x \approx y \rightarrow K_t x \approx y$	<b>RIGIDN</b>	$x \not\approx y \rightarrow K_t x \not\approx y$
<b>DETA</b>	$\langle x := t \rangle \varphi \rightarrow [x := t]\varphi$	<b>DAS</b>	$\langle x := t \rangle \top$
<b>KAS</b>	$[x := t](\varphi \rightarrow \psi) \rightarrow ([x := t]\varphi \rightarrow [x := t]\psi)$		
<b>EFAS</b>	$[x := t]x \approx t$		
<b>SUB2AS</b>	$\varphi[y/x] \rightarrow [x := y]\varphi$ ( $\varphi[y/x]$ is admissible)		

Rules:

<b>MP</b>	$\frac{\varphi, \varphi \rightarrow \psi}{\psi}$	<b>NECK</b>	$\frac{\vdash \varphi}{\vdash K_t \varphi}$	<b>NECAS</b>	$\frac{\vdash \varphi \rightarrow \psi}{\vdash \varphi \rightarrow [x := t]\psi}$ ( $x \notin Fv(\varphi)$ )
-----------	--	-------------	---	--------------	--

No explicit quantifiers! The **Barcan trick** does not work.

- Extend the language with countably many new variables.
- Build a pseudo canonical frame using maximal consistent sets for various sublanguages of the extended language, with witnesses for the names.
- Given a maximal consistent set, cut out its generated subframe from the pseudo frame, and build a constant-domain canonical model, by taking certain equivalence classes of variables as the domain.
- Show that the truth lemma holds for the canonical model.
- Take the reflexive symmetric transitive closure of the relations in pseudo model and show that the truth of the formulas in the original language are preserved.
- Extending each consistent set of the original model to a maximal consistent set with witnesses.

### Proposition

*$[x := t]$  cannot be eliminated qua expressivity.*

### Theorem

*SELAS is sound and complete w.r.t. S5 models.*

### Theorem

*SELAS is decidable over arbitrary models or reflexive models.  
SELAS is not decidable over S5 models.*

A special case can be expressed by  $[x := a]K_b(x \approx a)$ .

But there are different interpretations of “knowing who”:

- I know who can help Alice. (mention-some)
  - $\exists x K_b(H(x, a))$ .
- I know who came to the party yesterday. (mention-all)
  - Strongly exhaustive reading:  $\forall x (K_a \text{come}(x) \vee K_a \neg \text{come}(x))$
  - Weakly exhaustive reading:  $\forall x (\text{come}(x) \rightarrow K_a \text{come}(x))$
- You know who gave the talk today, because you know the speaker is:
  - that guy over there (by identification)
  - Yanjing Wang (by name)
  - A guy from Peking University (by affiliation)
  - ...

The meaning of *knowing who* based on *conceptual covers* (Aloni 2001).

- Model theoretical issues of **ELAS**.
- Extension with increasing and varying domain models.
- Extension with function symbols.
- Extension with a (termed) common knowledge operator.
- Extension with limited quantifications over agents, e.g., all the agents know or some agent knows.
- What if multiple people have the same name?

In general, see what happens if we **replace standard epistemic logic** with **ELAS** in existing work of epistemic logic.

Agents are not just indexes...