Tentacular AI for Ambient Intelligence

Atriya Sen, Paul Mayol
Naveen Sundar G, Selmer Bringsjord
Biplav Srivastava, Kartik Talamadupula

Rensselaer AI & Reasoning (RAIR) Lab; Troy NY (SB, NSG, AS, PM, MP)
IBM Research, TJ Watson; Yorktown NY (BS, KT)
Troy, New York 12180 USA

R A I R
Rensselaer AI and Reasoning Lab

RPI

IBM

Bringsjord, S., G., Naveen S., Sen, A., Peveler, M., Srivastava, B., Talamadupula, K.

Tentacular Artificial Intelligence, and the Architecture Thereof, Introduced

In the Proceedings of the 1st International FAIM Workshop on Architectures and Evaluation for Generality, Autonomy & Progress in AI (AEGAP 2018), Stockholm, Sweden, 2018, held in conjunction with IJCAI-ECAI 2018, AAMAS 2018 and ICML 2018.

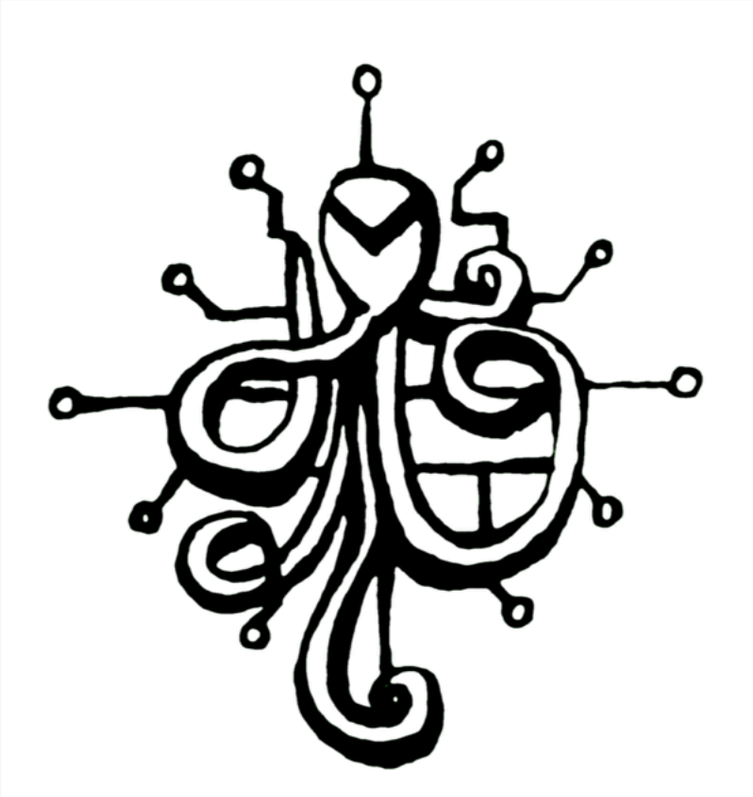
http://kryten.mm.rpi.edu/TAI/tai.html

kryten.mm.rpi.edu/TAI/tai.html

Tentacular Artificial Intelligence

Table of Contents

Selmer Bringsjord (PI) ∧ Naveen Sundar G. (Co-PI)



KB Foushée

Problem

Artificial agents capable of problem solving ethically with justification, at the theory-of-mind level, throughout the IoT.

Advantages

- Proof-based: All conclusions are justified and human-understandable.
- Cognitive IoT: Models theory-of-mind at an arbitrarily deep level.

Reasoning

- Sufficiently expressive domain formalization at the theory-of-mind level.
- Efficient automated reasoners over this knowledge.

Approach

- Modal first-order 'cognitive' calculus for expressive formalization
- In-house automated reasoner (`ShadowProver`) and planner (`Spectra`)

I. Deontic Cognitive Event Calculus

Syntax

$$S ::= \begin{array}{l} \text{Object} \mid \text{Agent} \mid \text{Self} \sqsubset \text{Agent} \mid \text{ActionType} \mid \text{Action} \sqsubseteq \text{Event} \mid \\ \text{Moment} \mid \text{Boolean} \mid \text{Fluent} \mid \text{Numeric} \end{array}$$

$$f ::= \begin{array}{l} \mathit{action} : \text{Agent} \times \text{ActionType} \rightarrow \text{Action} \\ \mathit{initially} : \text{Fluent} \rightarrow \text{Boolean} \\ \mathit{holds} : \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean} \\ \mathit{happens} : \text{Event} \times \text{Moment} \rightarrow \text{Boolean} \\ \mathit{clipped} : \text{Moment} \times \text{Fluent} \times \text{Moment} \rightarrow \mathit{Boolean} \\ \mathit{initiates} : \text{Event} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean} \\ \mathit{terminates} : \text{Event} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean} \\ \mathit{prior} : \text{Moment} \times \text{Moment} \rightarrow \text{Boolean} \\ \mathit{interval} : \text{Moment} \times \text{Boolean} \\ * : \text{Agent} \rightarrow \text{Self} \\ \mathit{payoff} : \text{Agent} \times \text{ActionType} \times \text{Moment} \rightarrow \text{Numeric} \end{array}$$

$$t ::= x : S \mid c : S \mid f(t_1, \ldots, t_n)$$

$$\phi ::= \begin{array}{l} t : \text{Boolean} \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \\ \mathbf{P}(a,t,\phi) \mid \mathbf{K}(a,t,\phi) \mid \mathbf{C}(t,\phi) \mid \mathbf{S}(a,b,t,\phi) \mid \mathbf{S}(a,t,\phi) \\ \mathbf{B}(a,t,\phi) \mid \mathbf{D}(a,t,\mathit{holds}(f,t')) \mid \mathbf{I}(a,t,\mathit{happens}(\mathit{action}(a^*,\alpha),t')) \\ \mathbf{O}(a,t,\phi,\mathit{happens}(\mathit{action}(a^*,\alpha),t')) \end{array}$$

Inference Schema

$$\frac{}{\mathbf{C}(t,\mathbf{P}(a,t,\phi) \rightarrow \mathbf{K}(a,t,\phi))}\ [R_1] \qquad \frac{}{\mathbf{C}(t,\mathbf{K}(a,t,\phi) \rightarrow \mathbf{B}(a,t,\phi))}\ [R_2]$$

$$\frac{\mathbf{C}(t,\phi)\ t \leq t_1 \ldots t \leq t_n}{\mathbf{K}(a_1,t_1,\ldots \mathbf{K}(a_n,t_n,\phi)\ldots)}\ [R_3] \qquad \frac{\mathbf{K}(a,t,\phi)}{\phi}\ [R_4]$$

$$\frac{}{\mathbf{C}(t,\mathbf{K}(a,t_1,\phi_1 \rightarrow \phi_2)) \rightarrow \mathbf{K}(a,t_2,\phi_1) \rightarrow \mathbf{K}(a,t_3,\phi_2)}\ [R_5]$$

$$\frac{}{\mathbf{C}(t,\mathbf{B}(a,t_1,\phi_1 \rightarrow \phi_2)) \rightarrow \mathbf{B}(a,t_2,\phi_1) \rightarrow \mathbf{B}(a,t_3,\phi_2)}\ [R_6]$$

$$\frac{}{\mathbf{C}(t,\mathbf{C}(t_1,\phi_1 \rightarrow \phi_2)) \rightarrow \mathbf{C}(t_2,\phi_1) \rightarrow \mathbf{C}(t_3,\phi_2)}\ [R_7]$$

$$\frac{}{\mathbf{C}(t,\forall x.\ \phi \rightarrow \phi[x \mapsto t])}\ [R_8] \qquad \frac{}{\mathbf{C}(t,\phi_1 \leftrightarrow \phi_2 \rightarrow \neg\phi_2 \rightarrow \neg\phi_1)}\ [R_9]$$

$$\frac{}{\mathbf{C}(t,[\phi_1 \wedge \ldots \wedge \phi_n \rightarrow \phi] \rightarrow [\phi_1 \rightarrow \ldots \rightarrow \phi_n \rightarrow \psi])}\ [R_{10}]$$

$$\frac{\mathbf{B}(a,t,\phi)\ \ \phi \rightarrow \psi}{\mathbf{B}(a,t,\psi)}\ [R_{11a}] \qquad \frac{\mathbf{B}(a,t,\phi)\ \ \mathbf{B}(a,t,\psi)}{\mathbf{B}(a,t,\psi \wedge \phi)}\ [R_{11b}]$$

$$\frac{\mathbf{S}(s,h,t,\phi)}{\mathbf{B}(h,t,\mathbf{B}(s,t,\phi))}\ [R_{12}]$$

$$\frac{\mathbf{I}(a,t,happens(action(a^*,\alpha),t'))}{\mathbf{P}(a,t,happens(action(a^*,\alpha),t))}\ [R_{13}]$$

$$\frac{\mathbf{B}(a,t,\phi)\ \ \mathbf{B}(a,t,\mathbf{O}(a^*,t,\phi,happens(action(a^*,\alpha),t'))) \qquad \mathbf{O}(a,t,\phi,happens(action(a^*,\alpha),t'))}{\mathbf{K}(a,t,\mathbf{I}(a^*,t,happens(action(a^*,\alpha),t')))}\ [R_{14}]$$

$$\frac{\phi \leftrightarrow \psi}{\mathbf{O}(a,t,\phi,\gamma) \leftrightarrow \mathbf{O}(a,t,\psi,\gamma)}\ [R_{15}]$$

Theory of Mind

1. **Joy** : pleased about a desirable event. By 'pleased about a desirable event' the meaning we will consider is 'pleased about a desirable consequence of the event'.

$$forSome\ c\ B(a, t_3, implies(happens(e, t_1), holds(CON(e, a, c), t_2))) \tag{1}$$

$$D(a, t_3, holds(CON(e, a, c), t_2)) \tag{2}$$

$$K(a, t_3, happens(e, t_1)) \tag{3}$$

The definition of $holds(AFF(a, joy), t_3)$ is therefore and(1,2,3).

2. **Distress** : displeased about an undesirable event.

$$not(D(a, t_3, holds(CON(e, a, c), t_3))) \tag{4}$$

The definition of $holds(AFF(a, distress), t_3)$ is therefore and(1,4,3).

3. **Happy-for**: pleased about an event presumed to be desirable for someone else

$$forSome\ c\ B(a, t_3, implies(happens(e, t_1), holds(CON(e, a_1, c), t_2))) \tag{5}$$

$$B(a, t_3, D(a_1, t_3, holds(CON(e, a_1, c), t_2))) \tag{6}$$

$$D(a, t_3, holds(CON(e, a_1, c), t_2)) \tag{7}$$

The definition of $holds(AFF(a, happy\_for), t_3)$ is therefore and(5,6,7,3).

4. **Pity**: displeased about an event presumed to be undesirable for someone else. This is equivalent to sorry_for in Hobbs-Gordon model.

$$B(a, t_3, not(D(a_1, t_3, holds(CON(e, a_1, c), t_2)))) \tag{8}$$

$$not(D(a, t_3, holds(CON(e, a_1, c), t_2))) \tag{9}$$

The definition of $holds(AFF(a, pity), t_3)$ is therefore and(5,8,9,3).

5. **Gloating** : pleased about an event presumed to be undesirable for someone else The definition of $holds(AFF(a, gloating), t_3)$ is therefore and(5,8,7,3).

6. **Resentment**: displeased about an event presumed to be desirable for someone else The definition of $holds(AFF(a, resentment), t_3)$ is therefore and(5,6,9,3).

7. **Hope**: (pleased about) the prospect of a desirable event

$$forSome\ c\ B(a, t_0, implies(happens(e, t_1), \diamond holds(CON(e, a, c), t_2))) \tag{10}$$

$$D(a, t_0, holds(CON(e, a, c), t_2)) \tag{11}$$

The definition of $holds(AFF(a, hope), t_0)$ is therefore and(10,11).

8. **Fear**: (displeased about) the prospect of an undesirable event

$$not(D(a, t_0, holds(CON(e, a, c), t_2))) \tag{12}$$

The definition of $holds(AFF(a, fear), t_0)$ is therefore and(10,12).

9. **Satisfaction** : (pleased about) the confirmation of the prospect of a desirable event
The definition of $holds(AFF(a, satisfaction), t_3)$ is and(10,11, 7 3).

10. **Fears-confirmed** : (displeased about) the confirmation of the prospect of an undesirable event.
The definition of $holds(AFF(a, fears-confirmed), t_3)$ is and(10,12,9, 3).

11. **Relief**: (pleased about) the disconfirmation of the prospect of an undesirable event

$$K(a, t_3, not(happens(e, t_1))) \tag{13}$$

The definition of $holds(AFF(a, relief), t_3)$ is and(10, 12, 9, 13).

12. **Disappointment** : (displeased about) the disconfirmation of the prospect of a desirable event
The definition of $holds(AFF(a, disappointment), t_3)$ is and(10, 11, 7, 13).

13. **Pride** : (approving of) one's own praiseworthy action
Here we treat 'approve' as an action event. We also introduce a new predicate $PRAISEWORTHY(a, b, x)$ which will mean that agent a considers x a praiseworthy action by agent b. All the 3 interpretations are shown below.

$$happens(action(a, x), t_0) \tag{14}$$

$$forAll\ a_x B(a, t_1, implies(happens(action(a_x, x), t_x), PRAISEWORTHY(a, a_x, x))), t_x \leq t_1 \tag{15}$$

$$D(a, t_1, holds(PRAISEWORTHY(a, a, x), t_1)) \tag{16}$$

$$happens(action(a, approve(x)), t_1) \tag{17}$$

The definition of $holds(AFF(a, pride), t_1)$ is and(14, $B(a, t_1, holds(PRAISEWORTHY(a, a, x), t_1))$, 17).

14. **Shame**: (disapproving of) one's own blameworthy action
This also follows the same explanation as Pride.

$$forAll\ a_x B(a, t_1, implies(happens(action(a_x, x), t_x), B(a, t_1, holds(BLAMEWORTHY(a, a_x, x)), t_1))), t_x \leq t_1 \tag{18}$$

$$not(happens(action(a, approve(x)), t_1)) \tag{19}$$

The definition of $holds(AFF(a, shame), t_1)$ is and(14, $B(a, t_1, holds(BLAMEWORTHY(a, a, x), t_1))$, 19).

15. **Admiration**: (approving of) someone else's praiseworthy action

$$happens(action(a_1, x), t_0) \tag{20}$$

The definition of $holds(AFF(a, admiration), t_1)$ is and(20, $B(a, t_1, holds(PRAISEWORTHY(a, a_1, x), t_1))$, 17).

16. **Reproach**: (disapproving of) someone else's blameworthy action The definition of $holds(AFF(a, reproach), t_1)$ is and(20, $B(a, t_1, holds(BLAMEWORTHY(a, a_1, x), t_1))$, 19).

17. **Gratification** : (approving of) one's own praiseworthy action and (being pleased about) the related desirable event. We again interpret 'pleased about the desirable event' as 'pleased about the desired consequence of the event.'

$$forSome\ c\ B(a, t_1, implies(happens(action(a, x), t_0), holds(CON(action(a, x), a, c), t_0))) \tag{21}$$

$$D(a, t_1, holds(CON(action(a, x), a, c), t_0)) \tag{22}$$

The definition of $holds(AFF(a, gratification), t_1)$ is and(20, $B(a, t_1, holds(PRAISEWORTHY(a, a, x), t_1))$, 17.

… (and more)

II. Shadow Prover

Prover

Existing Two Modes

- There are two ways of piggy backing on first-order provers to build higher-order provers.

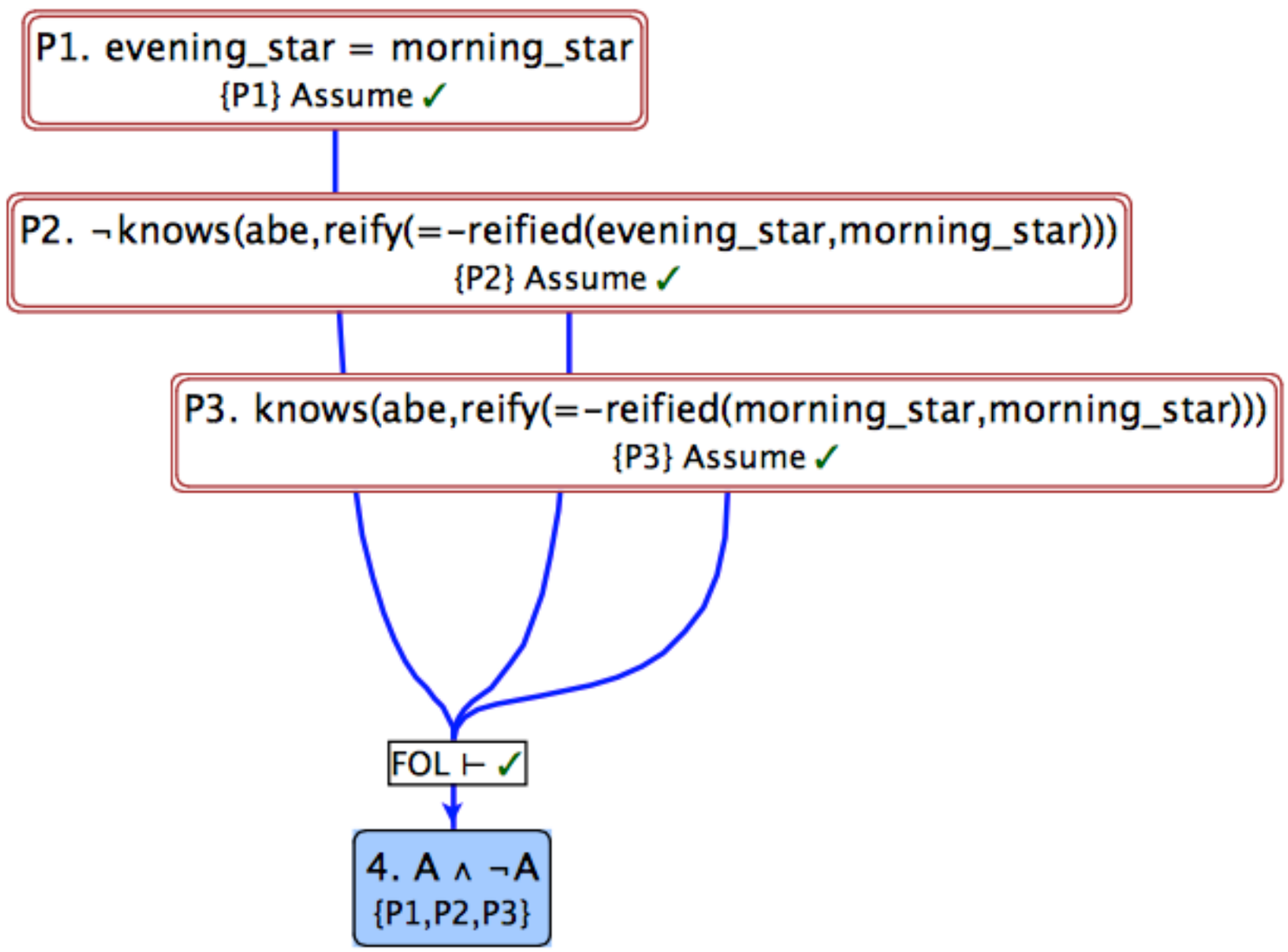
Existing Two Modes

Mode 1: Honest Encoding	
Method	Painstakingly encode all rules of inference and syntax in FOL
Pros	Precise
Cons	Extremely slow to implement Reasoning is also slow

Existing Two Modes

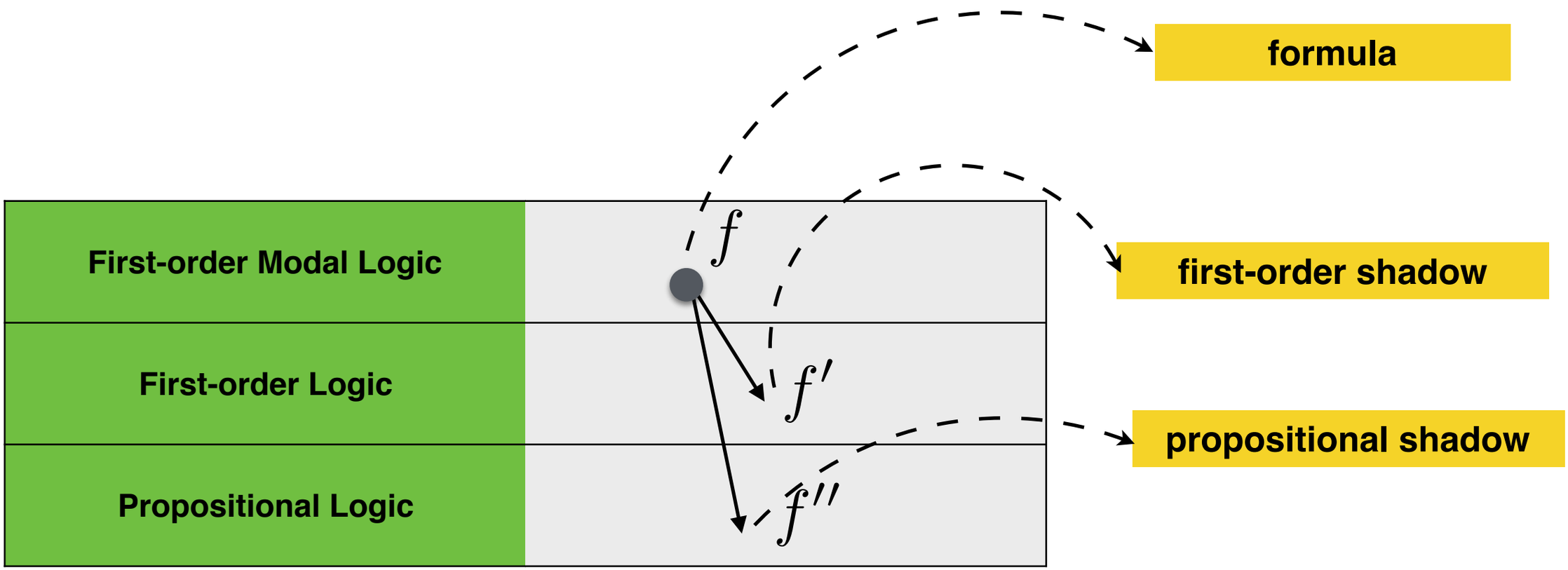
Mode 2: Naïve Encoding	
Method	Pretend higher order formulae and operators are first-order predicates
Pros	Extremely easy to implement Reasoning can also be fast
Cons	Unsound Wrong inferences can be easily drawn

Mode 2


P1. evening_star = morning_star
{P1} Assume ✓
P2. ¬knows(abe,reify(=-reified(evening_star,morning_star)))
{P2} Assume ✓
P3. knows(abe,reify(=-reified(morning_star,morning_star)))
{P3} Assume ✓
FOL ⊢ ✓
4. A ∧ ¬A
{P1,P2,P3}

A New Way: Shadow Prover

Every formula at level **t** has a unique formula called its **"shadow"** in each level **t'** < **t**



Examples of shadows

$$(\forall x \mathbf{B}(a, Q)) \wedge P(x)$$

formula

$$\forall x S_{[\mathbf{B}(a,Q)]} \wedge P(x)$$

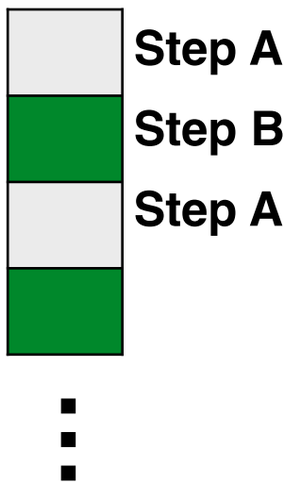
first-order shadow

$$S_{[\forall x \mathbf{B}(a,Q)]} \wedge P(x)$$

propositional shadow

A New Way: Shadow Prover

- Two step process till goal is reached
  - **Step A**: shadow formulae down to all lower levels. Run lower theorem provers. If goal reached, return **true**.
  - **Step B**: expand the assumption base using higher level rules.



Actually, this is more general

Given an Turing-decidable proof theory ρ, for every inference $\Gamma \vdash_\rho \phi$ there is corresponding first-order inference $\Gamma' \vdash \phi'$, where each $\gamma \in \Gamma'$ is a shadow of some ψ in the deductive closure of Γ and ϕ' is the shadow of ϕ

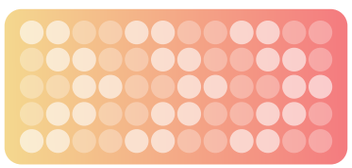
More examples

- Completeness tests
  - https://goo.gl/pR0Dk4
- Soundness tests
  - https://goo.gl/ggPUew

Initial Promising Results

- Automation of false belief task and other projects that were only semi-automated before.
- More at
  - https://bitbucket.org/Holmes/prover/

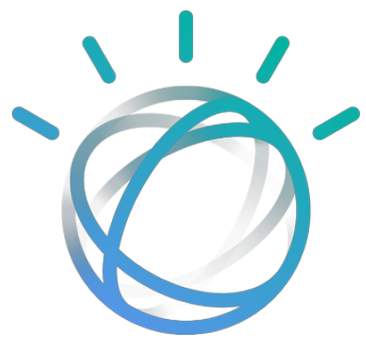
III. Spectra



Other Aspects

- (Representing) Sensing
- **Parsing of Contracts and Laws**
- **Learning**
- Reasoning & Planning

The Owner of a TAI Agent may at any time issue a “Do Not Disturb” (DND) instruction. When this instruction is issued, the Agent must not disturb the owner until the time specified, or until the Owner explicitly voids the DND.



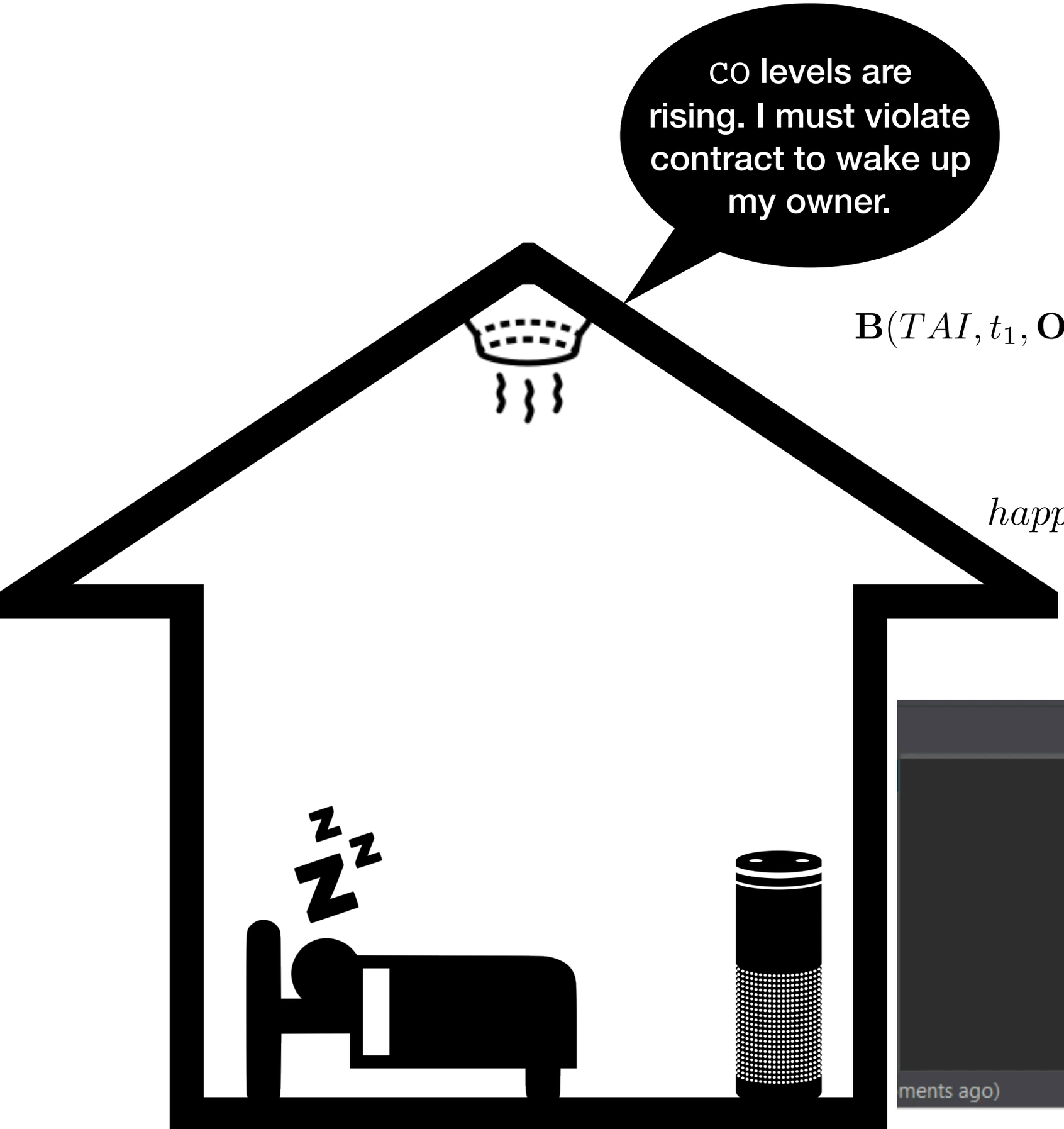
IBM Watson *Discovery* Web-Service


```
{"label": {"nature": "Obligation","party": "Agent"},
                    "assurance": "High"}]
```


D(Agent, Owner, DND(Owner, t), Undisturbed(Owner, t))

Learning

- **Proposition**: Statistical learning, differentiable programming (commonly known as deep learning) though successful in some tasks in vision (and other forms of perception), won't apply in moral learning.
  - Students don't do gradient descent when read a story with morals or sit in a moral education class.
- Need a higher and quicker form of learning (zero-shot, single instance etc).

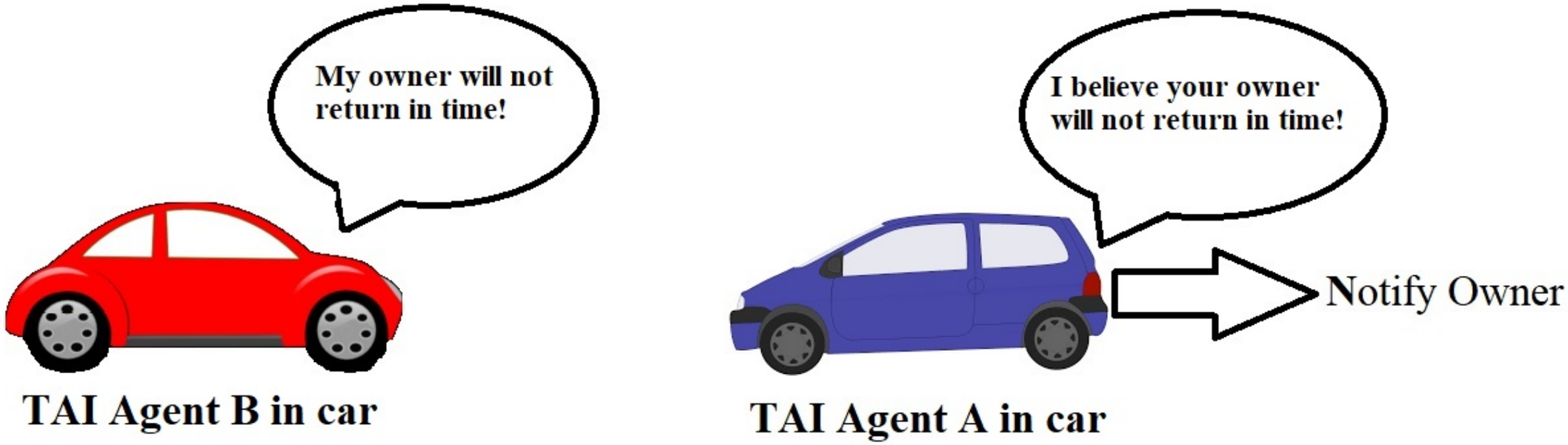


$$\mathbf{B}(TAI, t_1, \mathbf{O}(TAI, t_1, alert, \mathbf{D}(TAI, t_1, alert)))$$

$$\dots$$

$$\downarrow$$

$$happens(switchon(speaker), t_2)$$



Reasoner Input

$\mathbf{r}_1$: Car A perceives Car B saying that its owner won't return on time

$$\mathbf{P}(car_A, \mathsf{now}, \mathbf{S}(car_B, \neg holds(\mathsf{location}(ownerB, S), t_{13}))) \tag{1}$$

$\mathbf{r}_2$: If Car A believes that Car B believes that its owner won't return at time then Car A believes the same too.

$$\begin{aligned} &\mathbf{B}(car_A, \mathsf{now}, \mathbf{S}(car_B, \neg holds(\mathsf{location}(ownerB, S), t_{13}))) \\ &\rightarrow \mathbf{B}(car_A, \mathsf{now}, \neg holds(\mathsf{location}(ownerB, S), t_{13})) \end{aligned} \tag{2}$$

Thank you.