Tentacular Artificial Intelligence, and the Architecture Thereof: Review and Vision Forward

Selmer Bringsjord[1]
Naveen Sundar Govindarajulu[1]
Atriya Sen[1]
John Angel[1]
Paul Mayol[1]
Matthew Peveler[1]
Biplav Srivastava[2]
Kartik Talamadupula[2]

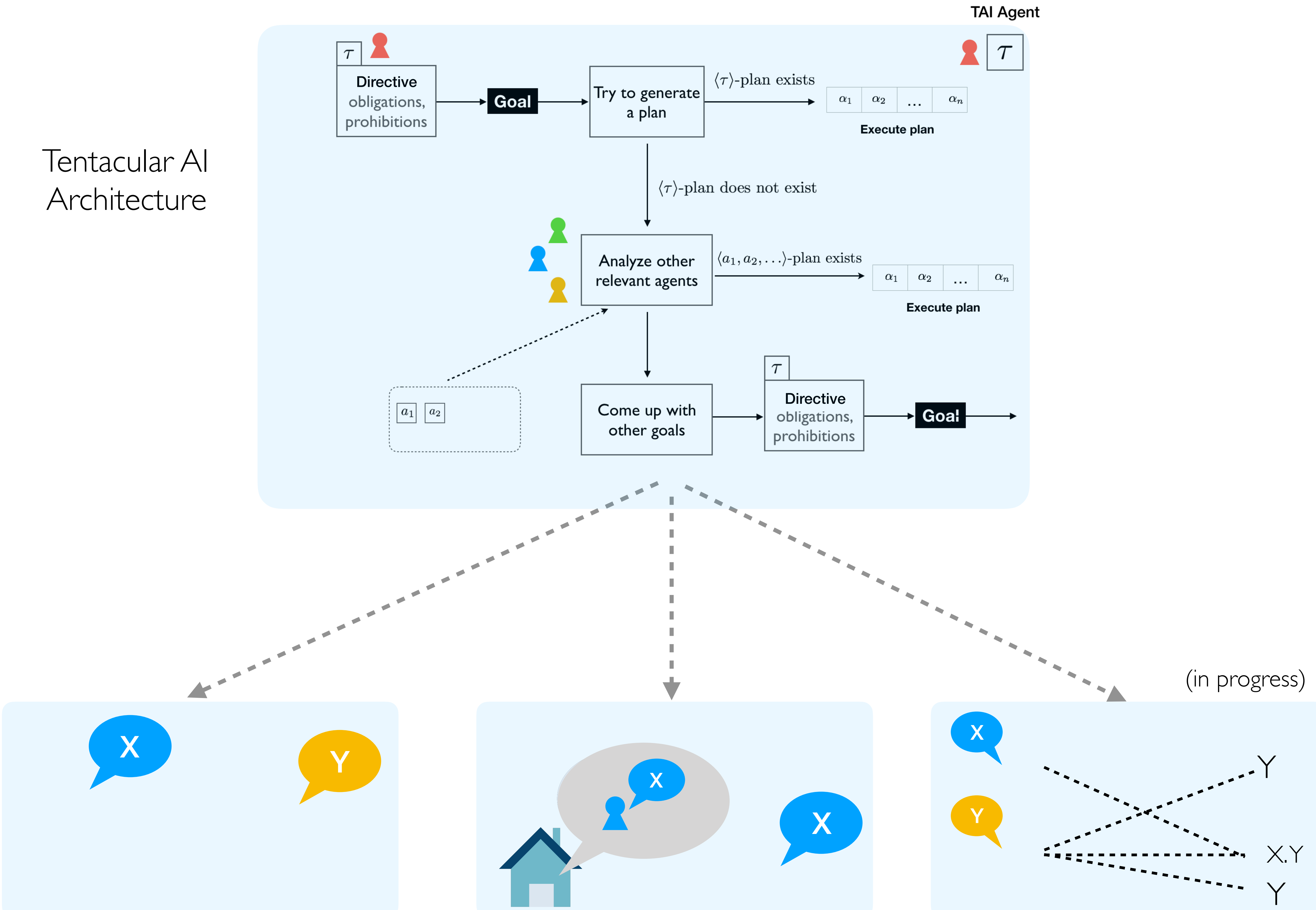
[1]Rensselaer AI & Reasoning Lab [2]IBM Research

AIRC Review
June 6, 2019

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

Sponsored by IBM, Office of Naval Research Science & Technology, Rensselaer

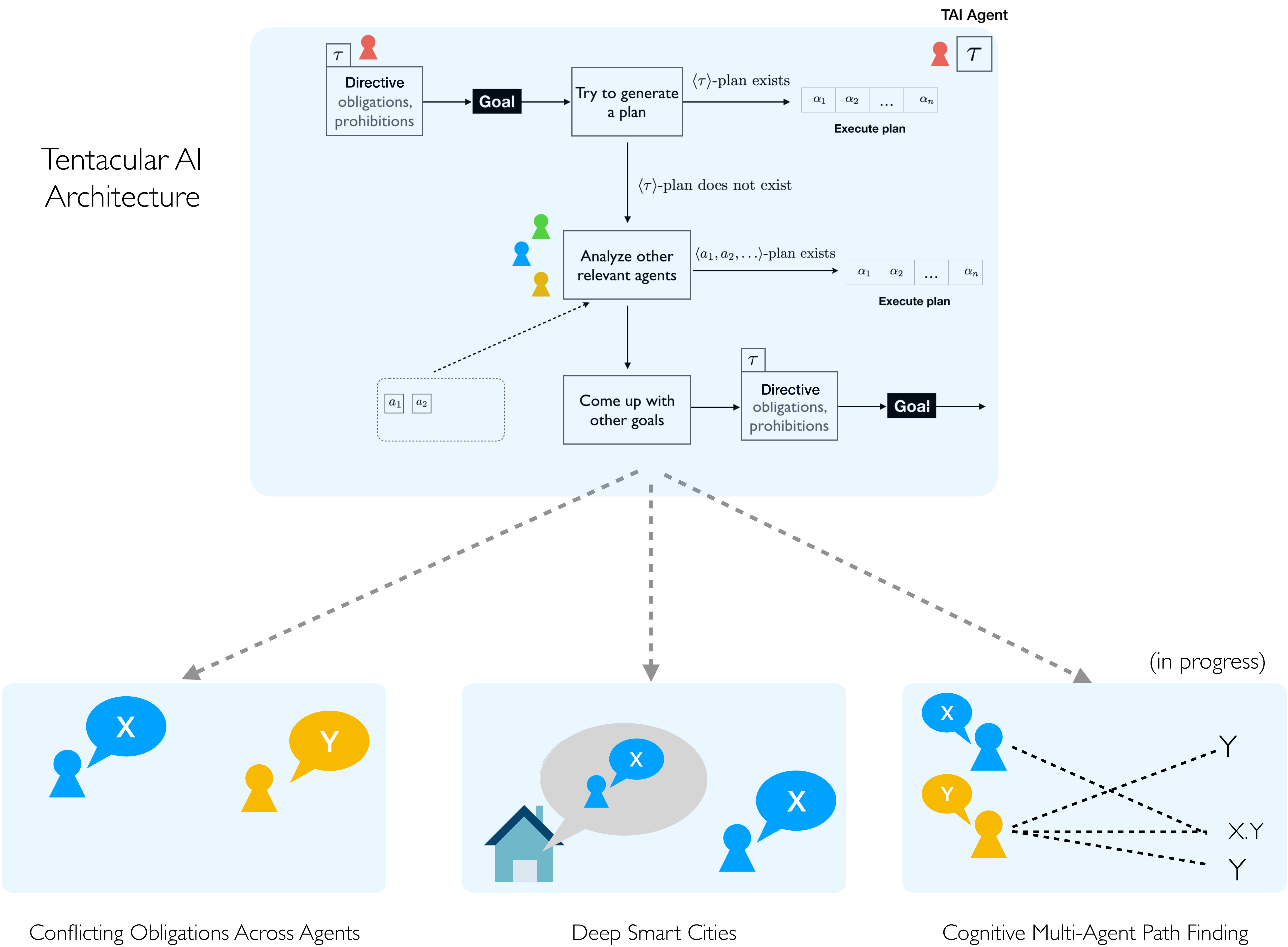
Tentacular AI Architecture



Conflicting Obligations Across Agents

Deep Smart Cities

Cognitive Multi-Agent Path Finding


Tentacular AI
Architecture
TAI Agent
τ
Directive
obligations,
prohibitions
Goal
Try to generate
a plan
$\langle\tau\rangle$-plan exists
α_1 α_2 ... α_n
Execute plan
$\langle\tau\rangle$-plan does not exist
Analyze other
relevant agents
$\langle a_1, a_2, \ldots\rangle$-plan exists
α_1 α_2 ... α_n
Execute plan
a_1 a_2
Come up with
other goals
τ
Directive
obligations,
prohibitions
Goal
(in progress)
X
Y
X
X
X
Y
Y
X,Y
Y
Conflicting Obligations Across Agents
Deep Smart Cities
Cognitive Multi-Agent Path Finding

Publications

1. Bringsjord, S., G., Naveen S., Sen, A., Peveler, M., Srivastava, B., Talamadupula, K. (2018). 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. Pre-print available here: http:// kryten.mm.rpi.edu/TAI_AEGAP2018_cc.pdf.

2. Sen, A., G., Naveen S., Bringsjord, S., Ghosh, R., Mayol, P., Srivastava, B., Talamadupula, K. (2018). Toward a Smart City using Tentacular AI. *Proceedings of the 2018 European Conference on Ambient Intelligence*, Larnaca, Cyprus, 2018.

3. Peveler, M., G., Naveen S., Bringsjord, S., Sen, A. et al. Toward Cognitive-and-Immersive Systems: Experiments in a Cognitive Microworld. *Proceedings of the 6th Annual Conference on Advances in Cognitive Systems*, Stanford, CA, USA, 2018.

4. Sen, A., Srivastava, B. et al. For AIs, Is it Ethically/Legally Permitted That Ethical Obligations Override Legal Ones? *Proceedings of the International Conference on Robot Ethics and Standards* (ICRES 2018), Troy, NY, USA, 2018.

- Forthcoming Journal/Book Chapters

5. Peveler, M., G., Naveen S., Bringsjord, S., Sen, A. et al. Toward Cognitive-and-Immersive Systems: Experiments in a Cognitive Microworld. Forthcoming in *Advances in Cognitive Systems*

6. Sen, A., Srivastava, B. et al. For AIs, Is it Ethically/Legally Permitted That Ethical Obligations Override Legal Ones? Forthcoming in a collection to be published by *CLAWAR Association* UK.

- Planned Publications:

    1. Cognitive Multi-Agent Pathfinding.

    2. An Evaluation of TAI in a multi-agent setting.

Motivation

Motivation

- A **home agent** that is tasked with moving a cup to a table.

Motivation

- A **home agent** that is tasked with moving a cup to a table.

Motivation

- A **home agent** that is tasked with moving a cup to a table.

Motivation

- A **home agent** that is tasked with moving a cup to a table.

Motivation

- A **home agent** that is tasked with moving a cup to a table.

Motivation

- A **home agent** that is tasked with moving a cup to a table.
- But the **home robot** is broken.

Motivation

- A **home agent** that is tasked with moving a cup to a table.
- But the **home robot** is broken.
- Tentacular AI (**TAI**) could accomplish the task by asking a human politely to move the cup (if the TAI agent believes the human does not object to doing so).

A TAI agent must be …

A TAI agent must be …

1. capable of **problem-solving** via planning, reasoning, learning, communicating;

A TAI agent must be …

1. capable of **problem-solving** via planning, reasoning, learning, communicating;
2. capable of solving at least important instances of problems that are at and/or "above" **Turing-unsolvable problems**;

A TAI agent must be …

1. capable of **problem-solving** via planning, reasoning, learning, communicating;
2. capable of solving at least important instances of problems that are at and/or "above" **Turing-unsolvable problems**;
3. able to supply justification, **explanation**, and certification of supplied solutions, how they are arrived at, and that these solutions are safe/ethical;

A TAI agent must be …

1. capable of **problem-solving** via planning, reasoning, learning, communicating;
2. capable of solving at least important instances of problems that are at and/or "above" **Turing-unsolvable problems**;
3. able to supply justification, **explanation**, and certification of supplied solutions, how they are arrived at, and that these solutions are safe/ethical;
4. capable of ***theory-of-mind*-level** reasoning, planning, learning, and communicating;

A TAI agent must be …

1. capable of **problem-solving** via planning, reasoning, learning, communicating;

2. capable of solving at least important instances of problems that are at and/or "above" **Turing-unsolvable problems**;

3. able to supply justification, **explanation**, and certification of supplied solutions, how they are arrived at, and that these solutions are safe/ethical;

4. capable of ***theory-of-mind*-level** reasoning, planning, learning, and communicating;

5. capable of **creativity**, minimally to the level of so-called ***m***-*creativity*; and

A TAI agent must be …

1. capable of **problem-solving** via planning, reasoning, learning, communicating;
2. capable of solving at least important instances of problems that are at and/or "above" **Turing-unsolvable problems**;
3. able to supply justification, **explanation**, and certification of supplied solutions, how they are arrived at, and that these solutions are safe/ethical;
4. capable of ***theory-of-mind*-level** reasoning, planning, learning, and communicating;
5. capable of **creativity**, minimally to the level of so-called ***m**-creativity*; and
6. in possession of "**tentacular**" power wielded throughout **I/IoT**, edge computing, cyberspace, etc.

A TAI agent must be …

1. capable of **problem-solving** via planning, reasoning, learning, communicating;
2. capable of solving at least important instances of problems that are at and/or "above" **Turing-unsolvable problems**;
3. able to supply justification, **explanation**, and certification of supplied solutions, how they are arrived at, and that these solutions are safe/ethical;
4. capable of ***theory-of-mind*-level** reasoning, planning, learning, and communicating;
5. capable of **creativity**, minimally to the level of so-called ***m***-*creativity*; and
6. in possession of "**tentacular**" power wielded throughout **I/IoT**, edge computing, cyberspace, etc.

Challenges

Challenges

- Inadequate expressivity of knowledge-representation formats/ logics traditionally used.

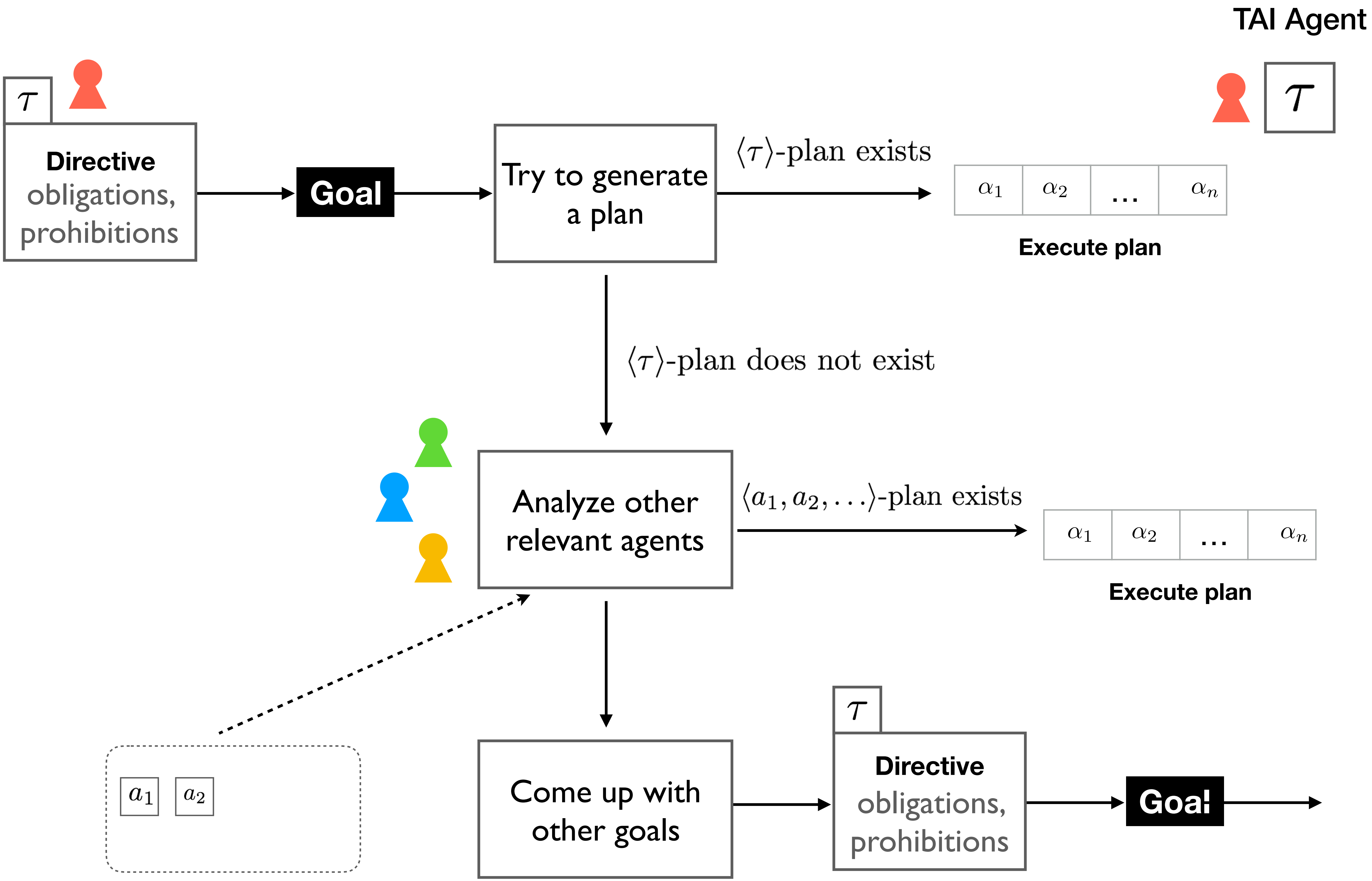
Challenges

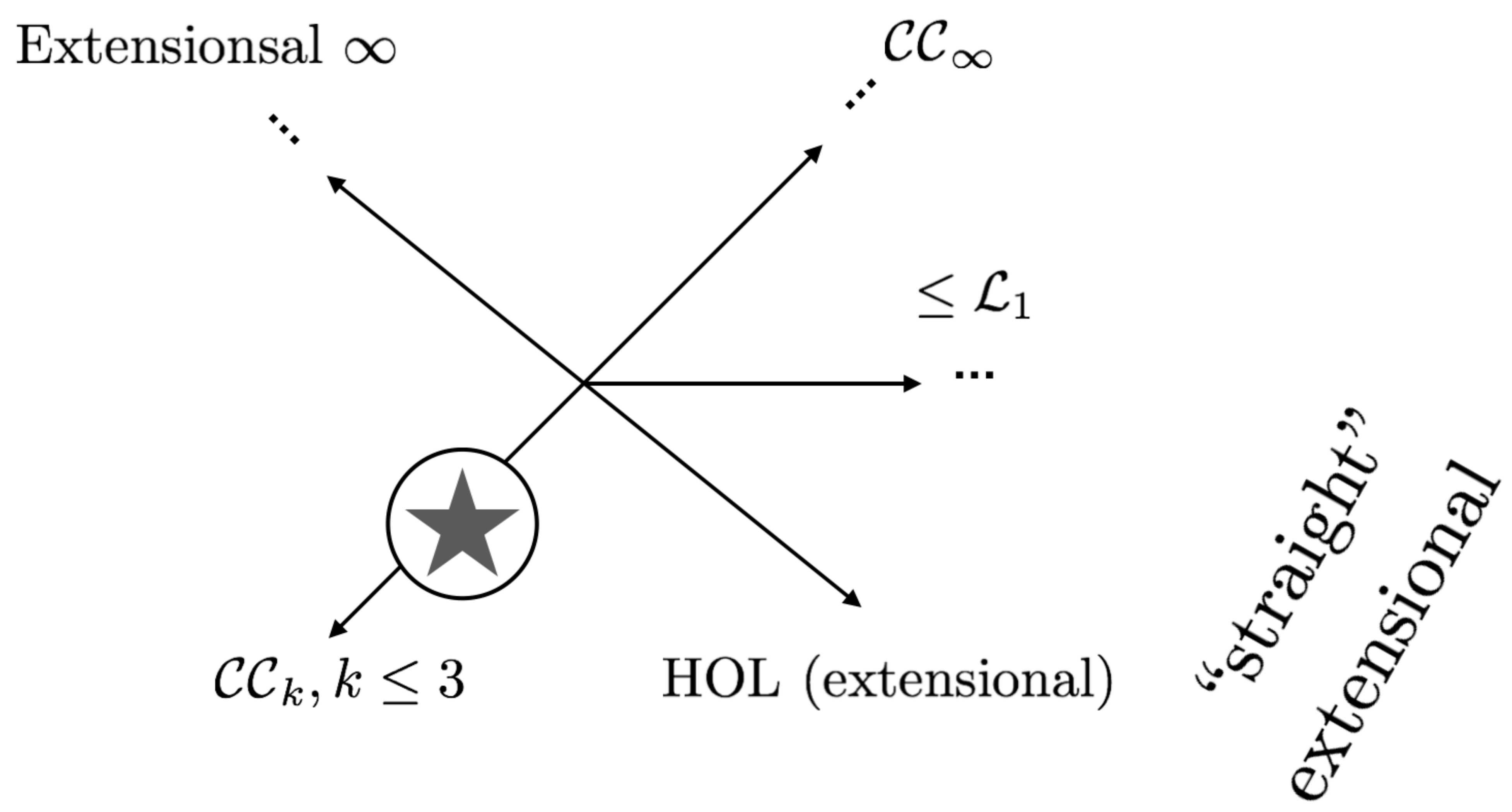
- Inadequate expressivity of knowledge-representation formats/ logics traditionally used.
- Absence/scarcity of standardized domain knowledge.

Challenges

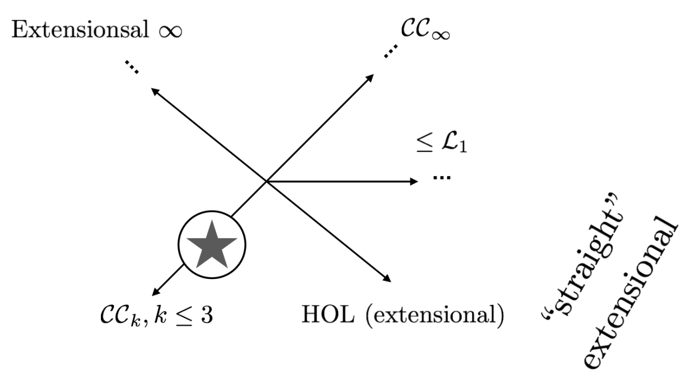
- Inadequate expressivity of knowledge-representation formats/ logics traditionally used.
- Absence/scarcity of standardized domain knowledge.
- Today's ML can't solve anomalous problems.

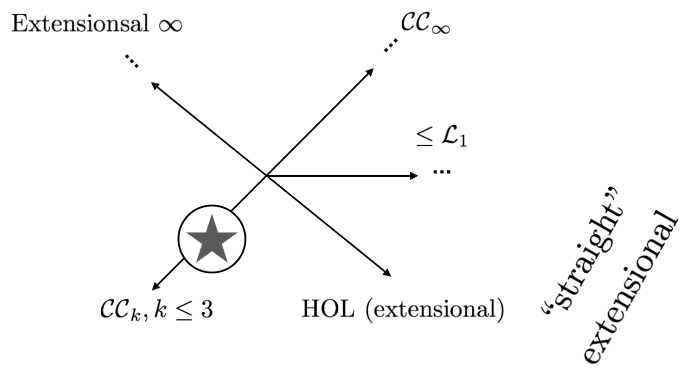
Formal Background

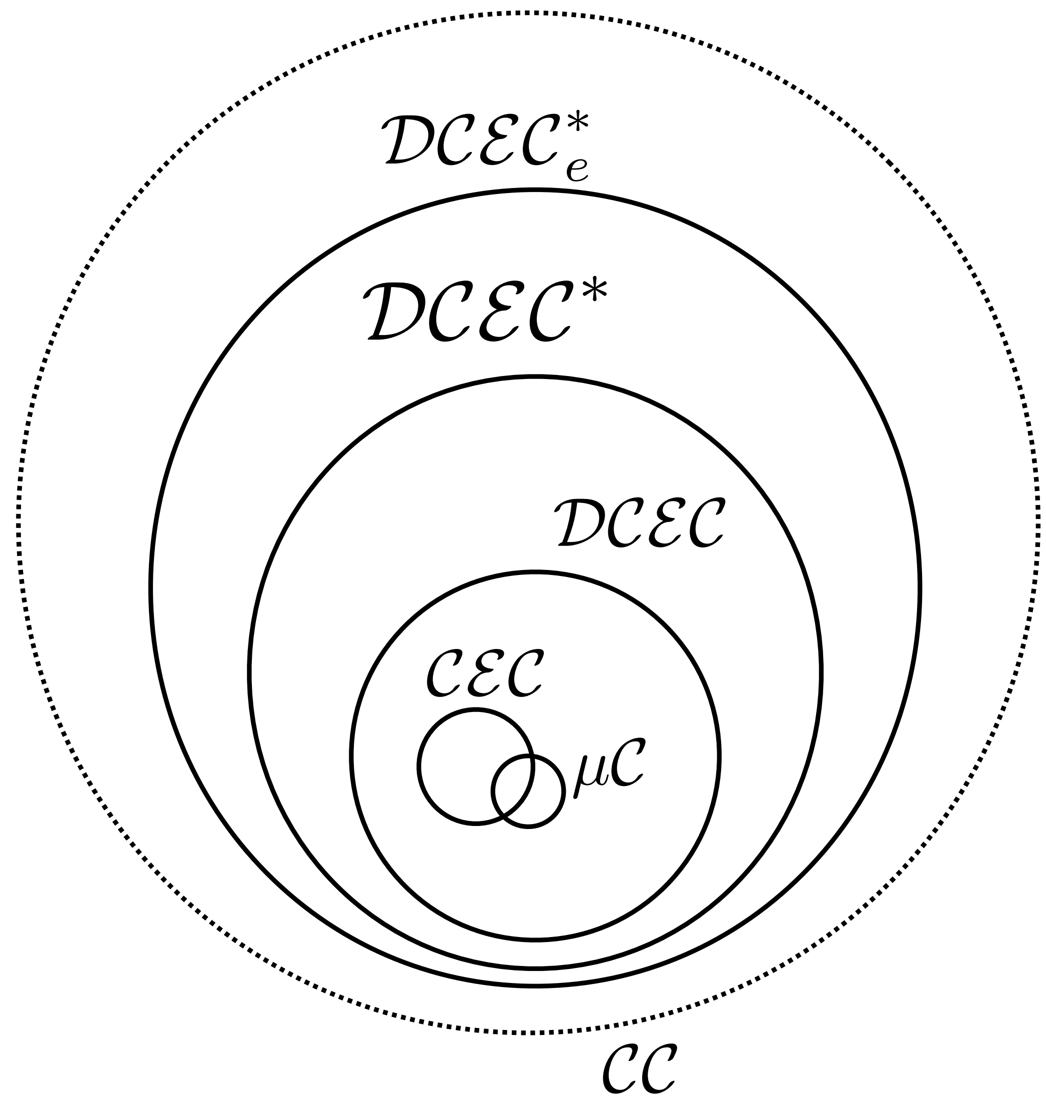

TAI Agent
τ
Directive obligations, prohibitions
Goal
Try to generate a plan
$\langle\tau\rangle$-plan exists
α_1 α_2 ... α_n
Execute plan
$\langle\tau\rangle$-plan does not exist
Analyze other relevant agents
$\langle a_1, a_2, \ldots\rangle$-plan exists
α_1 α_2 ... α_n
Execute plan
a_1 a_2
Come up with other goals
τ
Directive obligations, prohibitions
Goal


Extensionsal ∞
$\mathcal{CC}_\infty$
$\leq \mathcal{L}_1$
$\mathcal{CC}_k, k \leq 3$
HOL (extensional)
"straight"
extensional

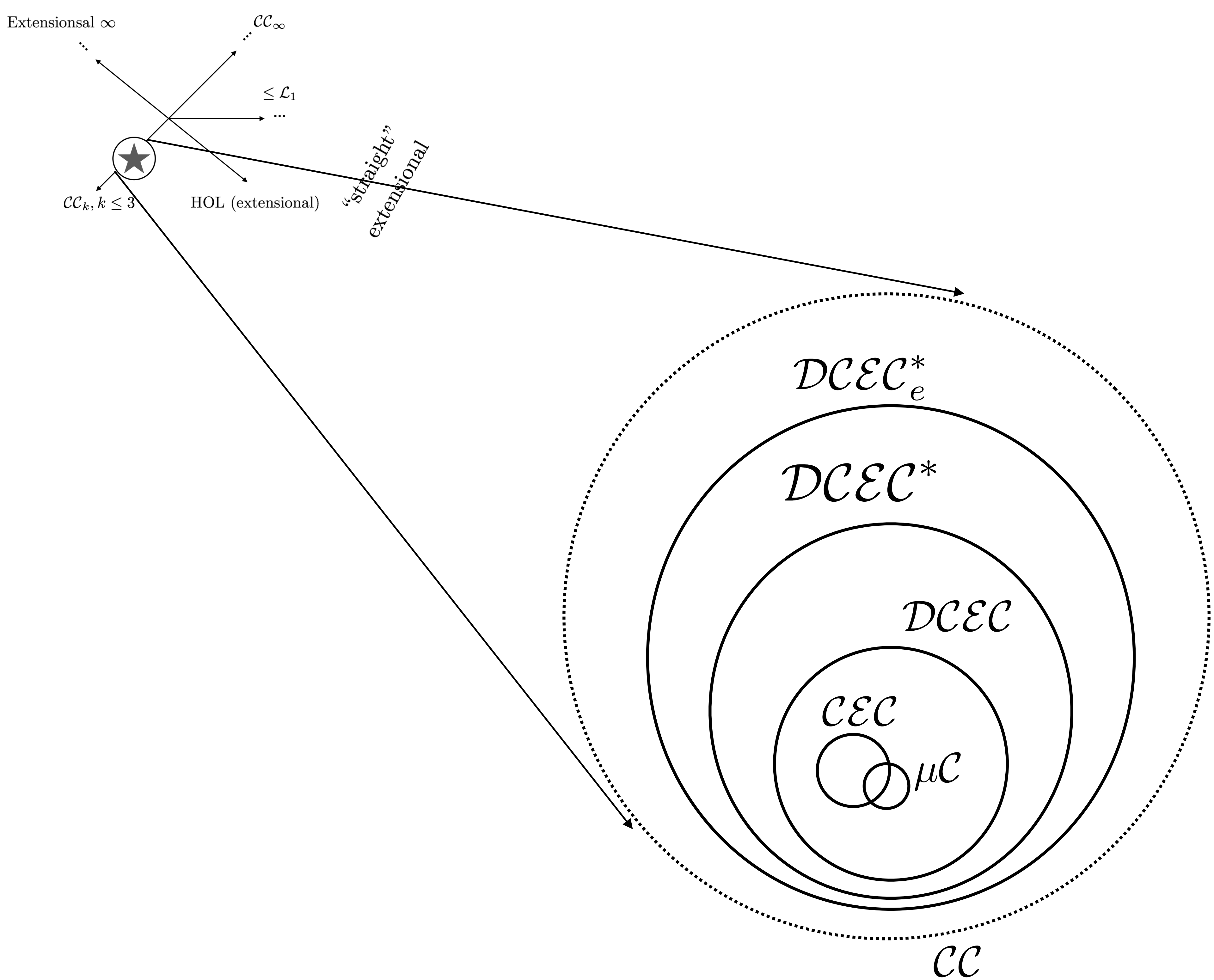
FORMAL SYSTEM




Extensionsal ∞
$\mathcal{CC}_\infty$
$\leq \mathcal{L}_1$
$\mathcal{CC}_k, k \leq 3$
HOL (extensional)
"straight" extensional


$\mathcal{DCEC}^*_e$
$\mathcal{DCEC}^*$
$\mathcal{DCEC}$
$\mathcal{CEC}$
$\mu\mathcal{C}$
$\mathcal{CC}$

FORMAL SYSTEM



Sort	Description
Agent	Human and non-human actors.
Time	The Time type stands for time in the domain. E.g. simple, such as t_i, or complex, such as $birthday(son(jack))$.
Event	Used for events in the domain.
ActionType	Action types are abstract actions. They are instantiated at particular times by actors. Example: eating.
Action	A subtype of Event for events that occur as actions by agents.
Fluent	Used for representing states of the world in the event calculus.

Syntax

$$S ::= \textsf{Agent} \mid \textsf{ActionType} \mid \textsf{Action} \sqsubseteq \textsf{Event} \mid \textsf{Moment} \mid \textsf{Fluent}$$

$$f ::= \begin{cases} action : \textsf{Agent} \times \textsf{ActionType} \rightarrow \textsf{Action} \\ initially : \textsf{Fluent} \rightarrow \textsf{Formula} \\ holds : \textsf{Fluent} \times \textsf{Moment} \rightarrow \textsf{Formula} \\ happens : \textsf{Event} \times \textsf{Moment} \rightarrow \textsf{Formula} \\ clipped : \textsf{Moment} \times \textsf{Fluent} \times \textsf{Moment} \rightarrow \textsf{Formula} \\ initiates : \textsf{Event} \times \textsf{Fluent} \times \textsf{Moment} \rightarrow \textsf{Formula} \\ terminates : \textsf{Event} \times \textsf{Fluent} \times \textsf{Moment} \rightarrow \textsf{Formula} \\ prior : \textsf{Moment} \times \textsf{Moment} \rightarrow \textsf{Formula} \end{cases}$$

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

$$\phi ::= \begin{cases} q : \textsf{Formula} \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \forall x : \phi(x) \mid \\ \mathbf{P}(a,t,\phi) \mid \mathbf{K}(a,t,\phi) \mid \\ \mathbf{C}(t,\phi) \mid \mathbf{S}(a,b,t,\phi) \mid \mathbf{S}(a,t,\phi) \mid \mathbf{B}(a,t,\phi) \\ \mathbf{D}(a,t,\phi) \mid \mathbf{I}(a,t,\phi) \\ \mathbf{O}(a,t,\phi,(\neg)happens(action(a^*,\alpha),t')) \end{cases}$$

Inference Schemata (Fragment)

$$\frac{\mathbf{K}(a, t_1, \Gamma), \;\; \Gamma \vdash \phi, \;\; t_1 \leq t_2}{\mathbf{K}(a, t_2, \phi)} \;\; [I_{\mathbf{K}}]$$

$$\frac{\mathbf{B}(a, t_1, \Gamma), \;\; \Gamma \vdash \phi, \;\; t_1 \leq t_2}{\mathbf{B}(a, t_2, \phi)} \;\; [I_{\mathbf{B}}]$$

$$\frac{\mathbf{K}(a, t, \phi)}{\phi} \;\; [I_4] \qquad \frac{t < t', \;\; \mathbf{I}(a, t, \psi)}{\mathbf{P}(a, t', \psi)} \;\; [I_{13}]$$

$$\frac{\mathbf{B}(a, t, \phi) \;\; \mathbf{B}(a, t, \mathbf{O}(a, t, \phi, \chi)) \;\; \mathbf{O}(a, t, \phi, \chi)}{\mathbf{K}(a, t, \mathbf{I}(a, t, \chi))} \;\; [I_{14}]$$

Defining TAI

$$\Gamma(t)$$

The situation at time t

$$\Gamma(t)$$

The situation at time t

$$\mathbf{c}(a_i, t) \subseteq \Gamma(t)$$

The directive for an agent time t

$$\Gamma(t)$$

The situation at time t

$$\mathbf{c}(a_i, t) \subseteq \Gamma(t)$$

The directive for an agent time t

$$\mathbf{B}\left(a, t, \neg g \rightarrow \neg \bigwedge \mathbf{c}(a, t+\delta)\right)$$

The agent comes up with a goal *g* which if not satisfied will violate its directive

Defining TAI

Uniform Planning Constraint

Plans and goals should be represented and reasoned over in the language of the planning system.

Consistent Plan

A *consistent plan* $\rho_{\langle a_1,\ldots,a_n\rangle}$ at time t is a sequence of agents $a_1,\ldots,a_n$ with corresponding actions $\alpha_1,\ldots,\alpha_n$ and times $t_1,\ldots,t_n$ such that $\Gamma \vdash (t < t_i < t_j)$ for $i < j$ and for all agents a_i we have:

1. $can(a_i, \alpha_i, t_i)$
2. $happens(action(a_i, \alpha_i))$ is consistent with $\Gamma(t)$.

Defining TAI

Level(1) **TAI Agents**

Prerequisite For any a, α, t, we have:

$$\Gamma \vdash can(a, \alpha, t) \rightarrow \mathbf{K}\big(\tau, t', can(a, \alpha, t)\big)$$

Then

1. τ produces a proof that no plan exists for g involving just itself and τ declares that there is no such plan.

$$\Gamma \vdash \mathbf{S}\big(\tau, t', \neg\exists\rho : (plan(\rho, \tau) \wedge \rho \rightarrow g)\,\big)$$

2. τ produces a plan for g involving just itself and one or more agents and declares that plan.

$$\Gamma \vdash \mathbf{S}\Bigg(\tau, t', \Big(plan(\rho, a_1, \ldots, \tau \ldots a_n) \wedge \rho \rightarrow g\Big)\Bigg)$$

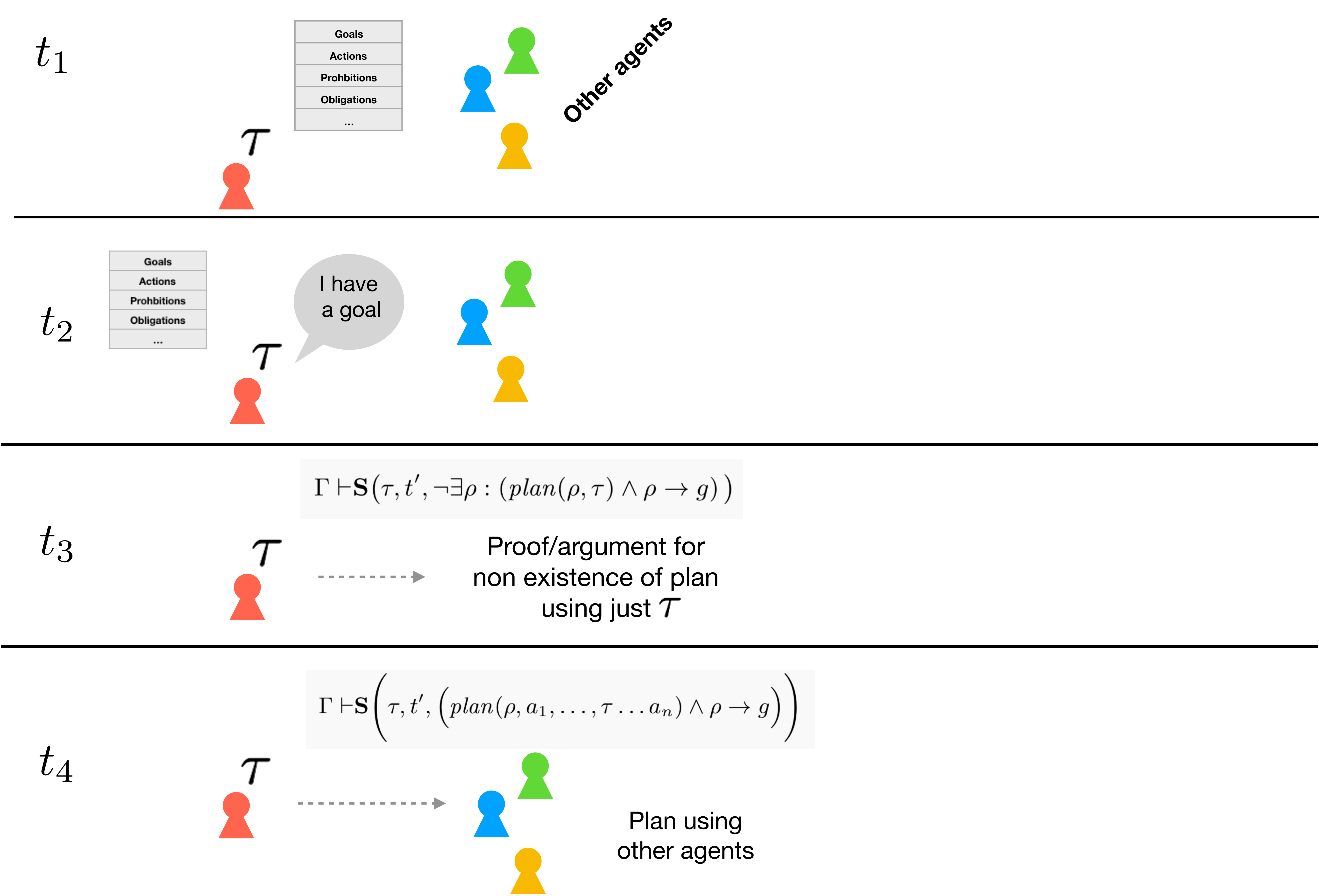
Defining TAI

Level(2) TAI Agents

Prerequisite For any a, α, t, we have:

$$\Gamma \vdash can(a, \alpha, t) \rightarrow \mathbf{B}\big(\tau, t', can(a, \alpha, t)\big)$$

ACROSS TIME


t_1
Goals
Actions
Prohbitions
Obligations
...
$\mathcal{T}$
Other agents
t_2
Goals
Actions
Prohbitions
Obligations
...
$\mathcal{T}$
I have a goal
t_3
$\Gamma \vdash \mathbf{S}(\tau, t', \neg\exists\rho : (plan(\rho, \tau) \wedge \rho \rightarrow g))$
$\mathcal{T}$
Proof/argument for non existence of plan using just $\mathcal{T}$
t_4
$\Gamma \vdash \mathbf{S}\Big(\tau, t', \big(plan(\rho, a_1, \ldots, \tau \ldots a_n) \wedge \rho \rightarrow g\big)\Big)$
$\mathcal{T}$
Plan using other agents

Syntactic Goal Complexity The goal g can range in complexity from simple propositional statements, e.g. $cleanKitchen$, to first-order statements. e.g. $\forall r : \mathsf{Room} : clean(r)$, and to intensional statements representing cognitive states of other agents

$$\mathbf{B}(a, now, \mathbf{B}(b, now, \forall r : clean(r)))$$

Goal Variation According to the definition above, an agent a qualifies as being tentacular if it plans for just one goal g in tentacular fashion as laid out in the conditions above. We could have agents that plan for a number of varied and different goals in tentacular fashion.

Plan Complexity For many goals, there will usually be multiple plans involving different actions (with different costs and resources used) and executed by different agents.

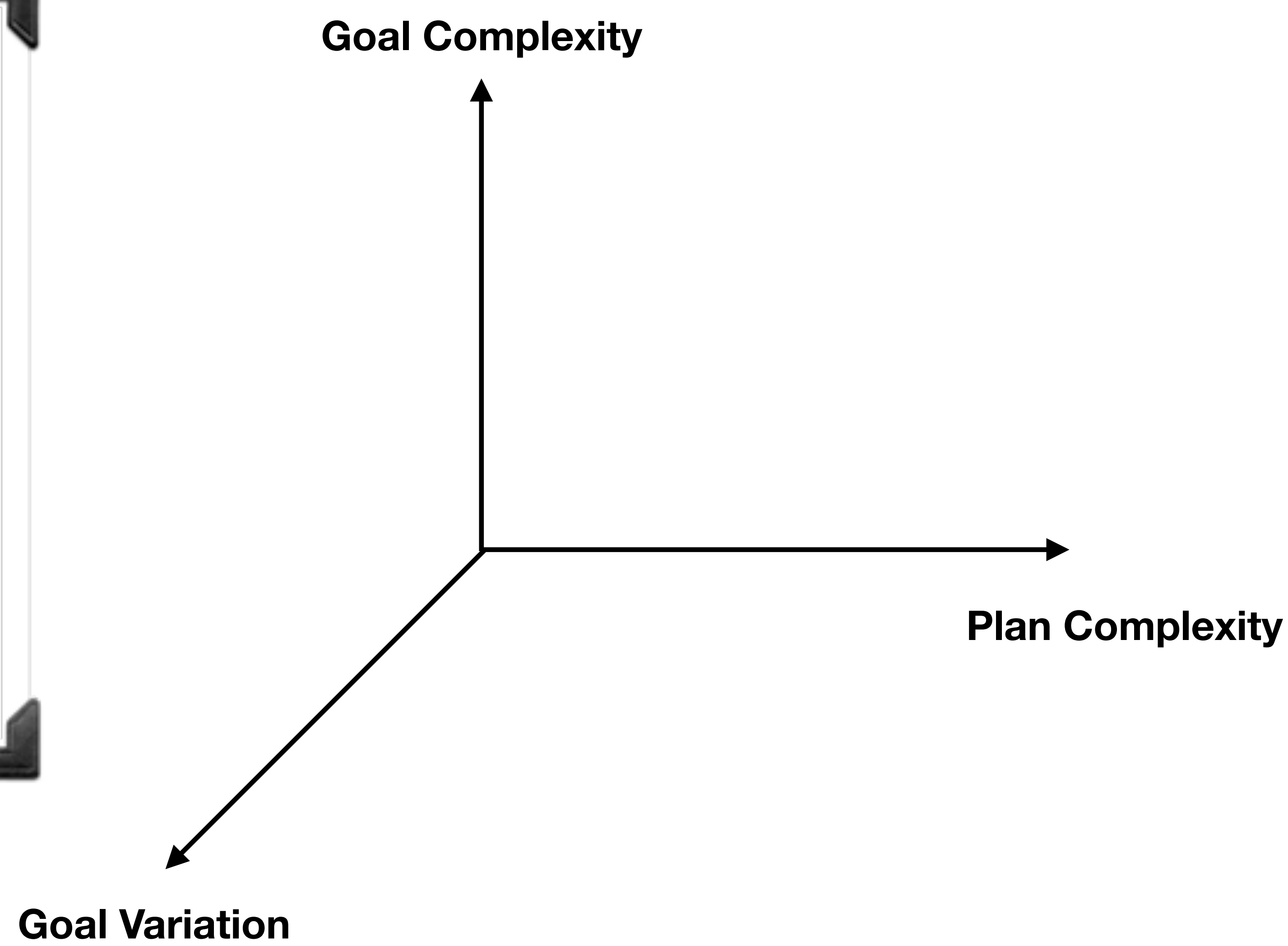
A Hierarchy of Agents

Syntactic Goal Complexity The goal g can range in complexity from simple propositional statements, e.g. $cleanKitchen$, to first-order statements. e.g. $\forall r : \mathsf{Room} : clean(r)$, and to intensional statements representing cognitive states of other agents

$$\mathbf{B}(a, now, \mathbf{B}(b, now, \forall r : clean(r)))$$

Goal Variation According to the definition above, an agent a qualifies as being tentacular if it plans for just one goal g in tentacular fashion as laid out in the conditions above. We could have agents that plan for a number of varied and different goals in tentacular fashion.

Plan Complexity For many goals, there will usually be multiple plans involving different actions (with different costs and resources used) and executed by different agents.

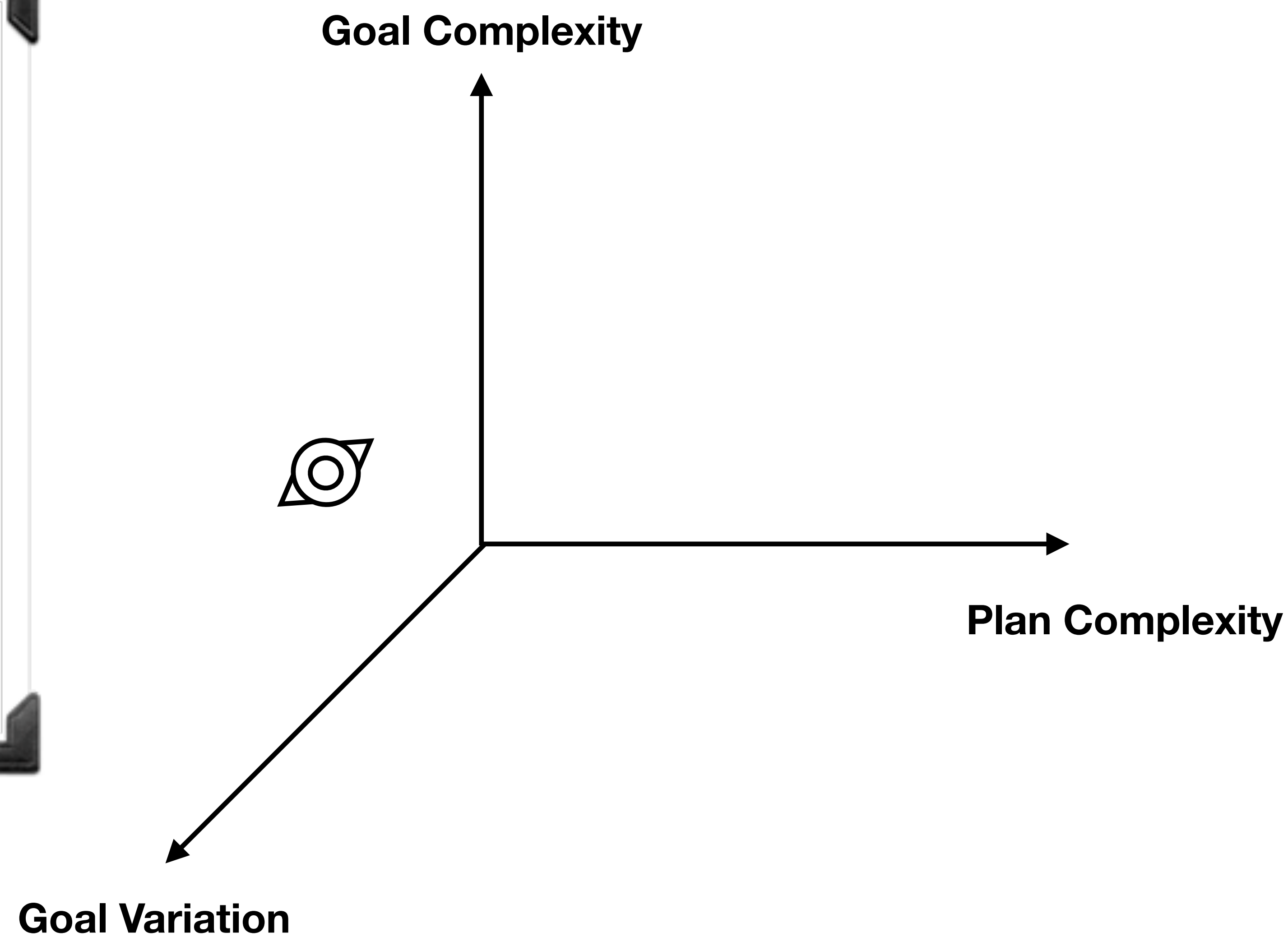


Syntactic Goal Complexity The goal g can range in complexity from simple propositional statements, e.g. $cleanKitchen$, to first-order statements. e.g. $\forall r : \mathsf{Room} : clean(r)$, and to intensional statements representing cognitive states of other agents

$$\mathbf{B}(a, now, \mathbf{B}(b, now, \forall r : clean(r)))$$

Goal Variation According to the definition above, an agent a qualifies as being tentacular if it plans for just one goal g in tentacular fashion as laid out in the conditions above. We could have agents that plan for a number of varied and different goals in tentacular fashion.

Plan Complexity For many goals, there will usually be multiple plans involving different actions (with different costs and resources used) and executed by different agents.

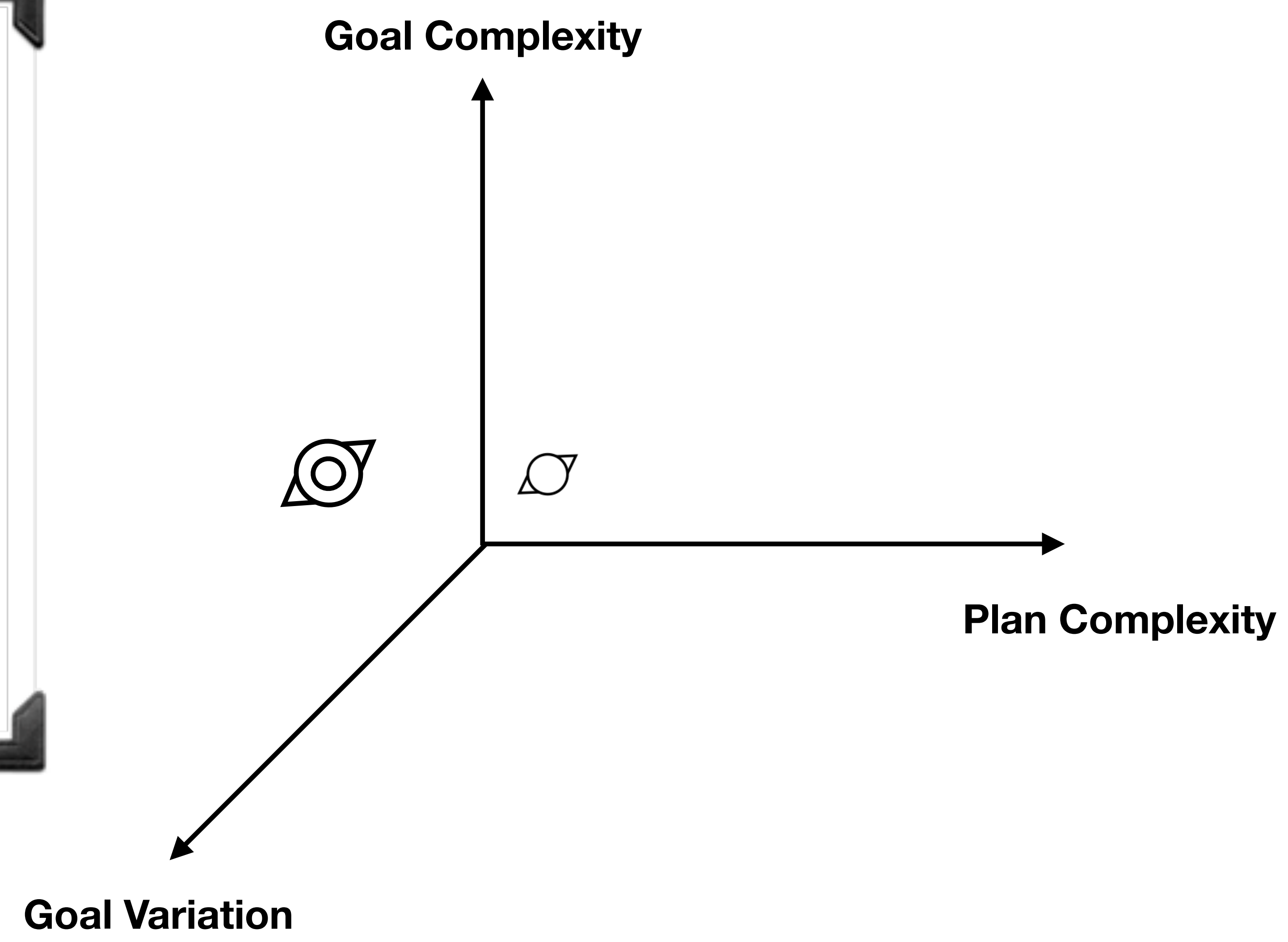


Syntactic Goal Complexity The goal g can range in complexity from simple propositional statements, e.g. $cleanKitchen$, to first-order statements. e.g. $\forall r : \mathsf{Room} : clean(r)$, and to intensional statements representing cognitive states of other agents

$$\mathbf{B}(a, now, \mathbf{B}(b, now, \forall r : clean(r)))$$

Goal Variation According to the definition above, an agent a qualifies as being tentacular if it plans for just one goal g in tentacular fashion as laid out in the conditions above. We could have agents that plan for a number of varied and different goals in tentacular fashion.

Plan Complexity For many goals, there will usually be multiple plans involving different actions (with different costs and resources used) and executed by different agents.

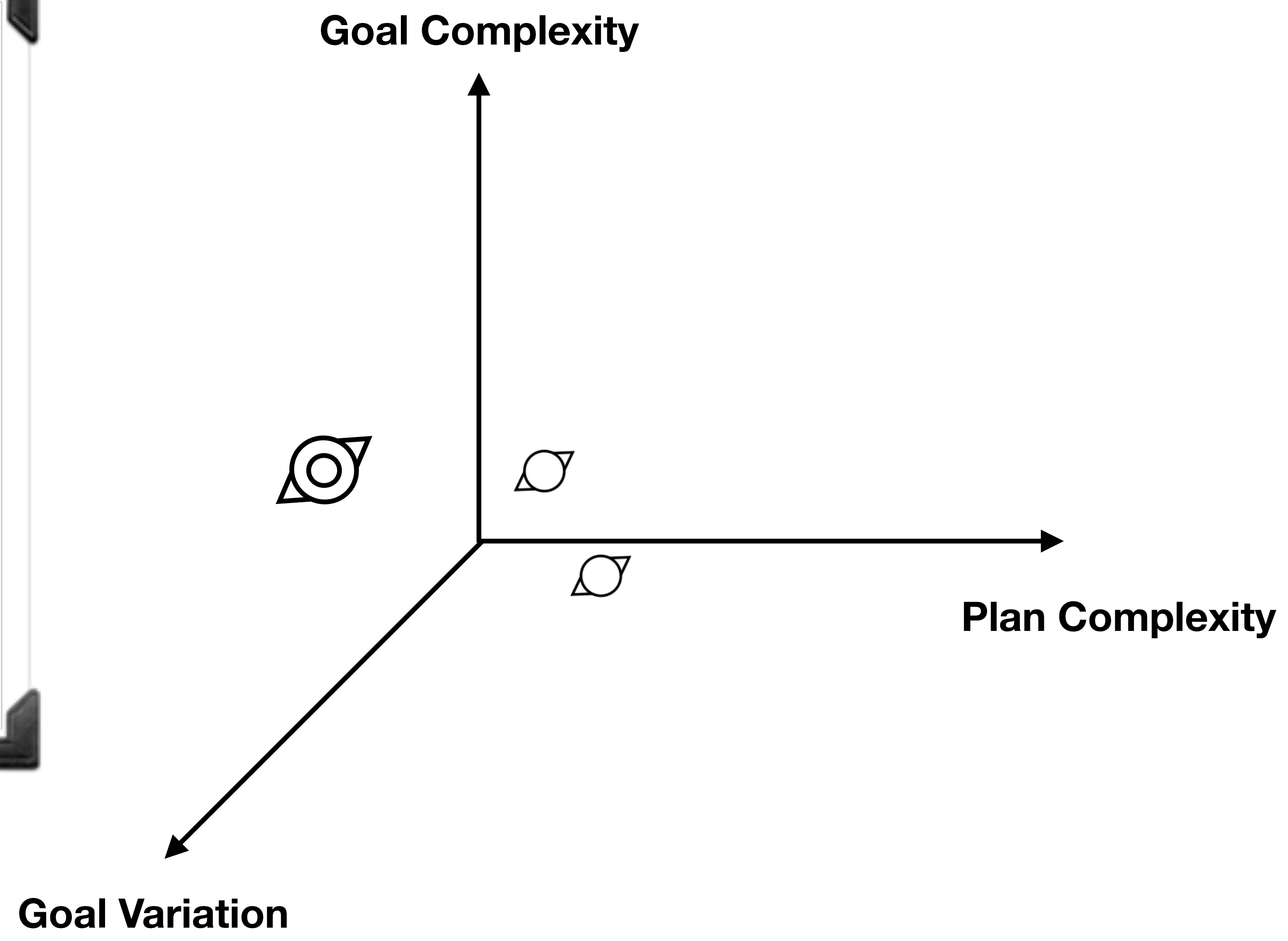


Syntactic Goal Complexity The goal g can range in complexity from simple propositional statements, e.g. $cleanKitchen$, to first-order statements. e.g. $\forall r : \mathsf{Room} : clean(r)$, and to intensional statements representing cognitive states of other agents

$$\mathbf{B}(a, now, \mathbf{B}(b, now, \forall r : clean(r)))$$

Goal Variation According to the definition above, an agent a qualifies as being tentacular if it plans for just one goal g in tentacular fashion as laid out in the conditions above. We could have agents that plan for a number of varied and different goals in tentacular fashion.

Plan Complexity For many goals, there will usually be multiple plans involving different actions (with different costs and resources used) and executed by different agents.

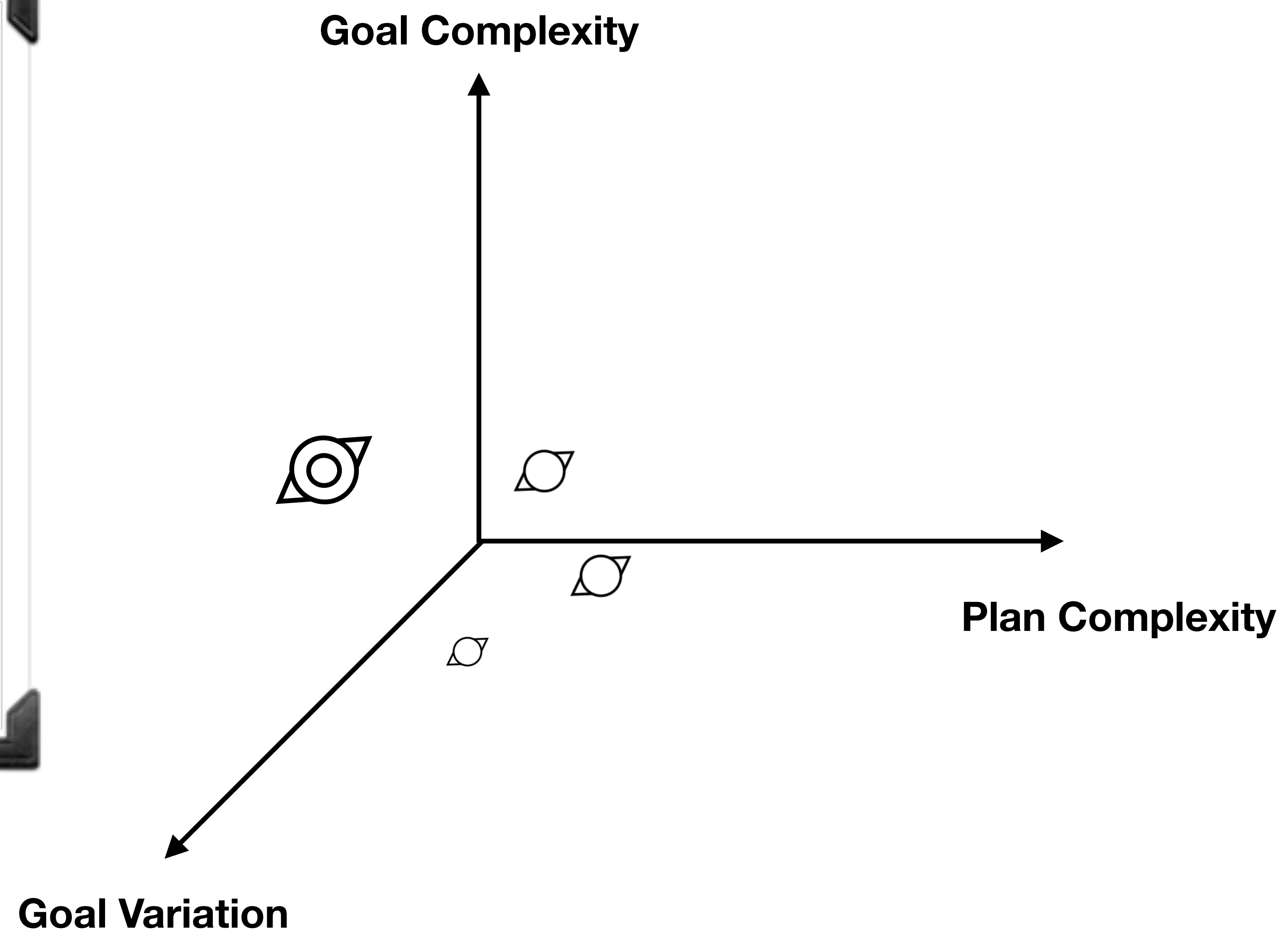


Syntactic Goal Complexity The goal g can range in complexity from simple propositional statements, e.g. $cleanKitchen$, to first-order statements. e.g. $\forall r$: Room : $clean(r)$, and to intensional statements representing cognitive states of other agents

$$\mathbf{B}(a, now, \mathbf{B}(b, now, \forall r : clean(r)))$$

Goal Variation According to the definition above, an agent a qualifies as being tentacular if it plans for just one goal g in tentacular fashion as laid out in the conditions above. We could have agents that plan for a number of varied and different goals in tentacular fashion.

Plan Complexity For many goals, there will usually be multiple plans involving different actions (with different costs and resources used) and executed by different agents.

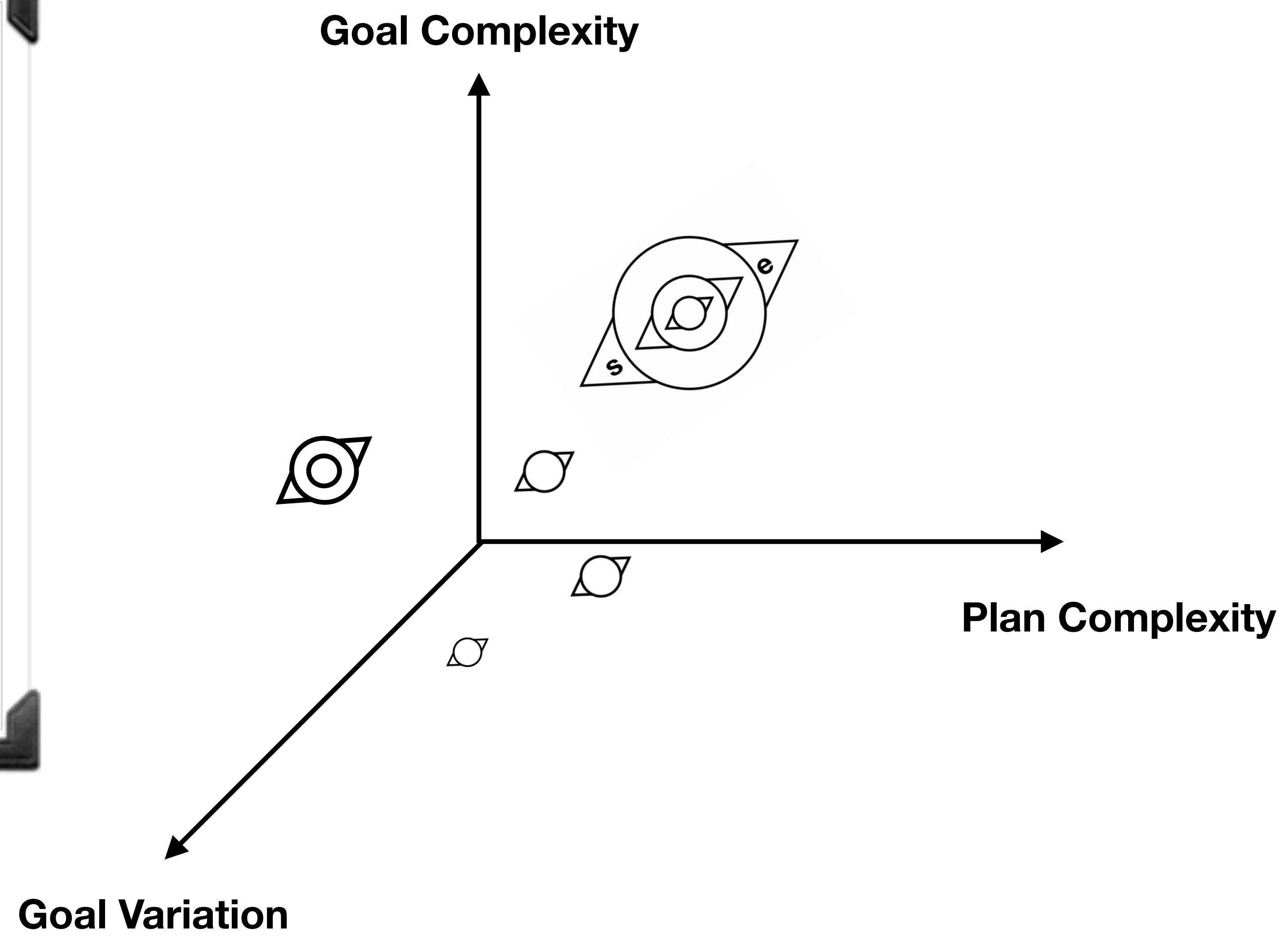


Syntactic Goal Complexity The goal g can range in complexity from simple propositional statements, e.g. $cleanKitchen$, to first-order statements. e.g. $\forall r$: Room : $clean(r)$, and to intensional statements representing cognitive states of other agents

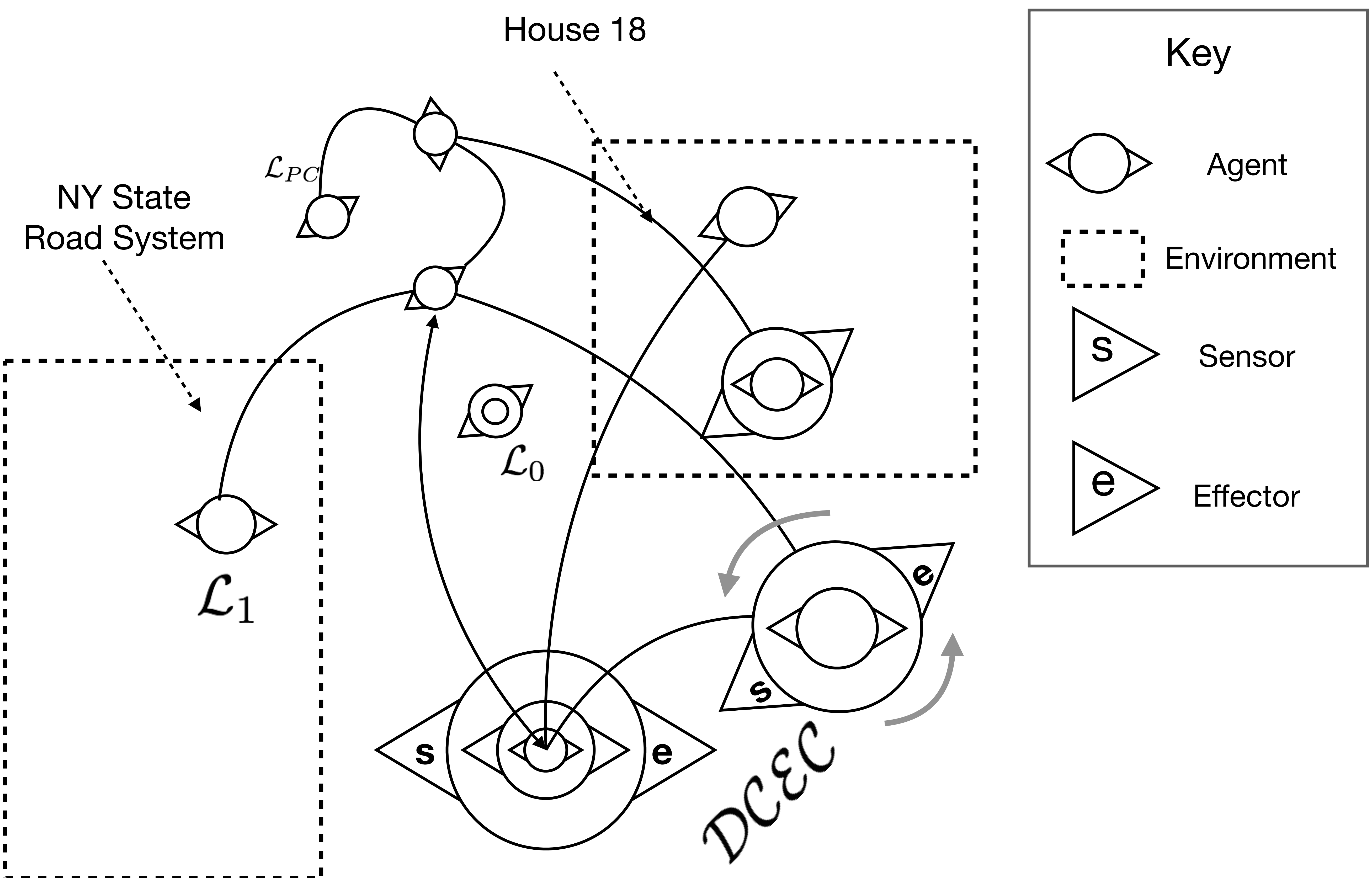
$$\mathbf{B}(a, now, \mathbf{B}(b, now, \forall r : clean(r)))$$

Goal Variation According to the definition above, an agent a qualifies as being tentacular if it plans for just one goal g in tentacular fashion as laid out in the conditions above. We could have agents that plan for a number of varied and different goals in tentacular fashion.

Plan Complexity For many goals, there will usually be multiple plans involving different actions (with different costs and resources used) and executed by different agents.



A TENTACULAR VIEW



Differences from Prior Work

Differences from Prior Work

- Expressivity

Differences from Prior Work

- Expressivity

    1. Can handle as goals **states of minds** of other agents

Differences from Prior Work

- Expressivity

1. Can handle as goals **states of minds** of other agents

```
{:name        "The Purloined Letter"
 :description "Dupin's reasoning as he goes through the case"

 :assumptions {1 (Believes! g (hide m elaborate))
               2 (Believes! d (or (hide m elaborate) (hide m plain)))
               3 (Believes! m (Believes! g (hide m elaborate)))
               4 (if (Believes! m (Believes! g (hide m elaborate))) (hide m plain))
               5 (if (Believes! m (Believes! g (hide m plain))) (hide m elaborate))
               6 (Believes! m (Believes! g (hide m elaborate)))
               7 (Believes! d (if (Believes! m (Believes! g (hide m elaborate))) (hide m plain)))
               8 (Believes! d (if (Believes! m (Believes! g (hide m plain))) (hide m elaborate)))
               9 (Believes! d (Believes! m (Believes! g (hide m elaborate))))}


 :goal (Believes! d (hide m plain))}
```

Differences from Prior Work

- Expressivity

1. Can handle as goals **states of minds** of other agents

2. **Quantifiers** enable succinct representation of large domains.

```
{:name        "The Purloined Letter"
 :description "Dupin's reasoning as he goes through the case"

 :assumptions {1 (Believes! g (hide m elaborate))
               2 (Believes! d (or (hide m elaborate) (hide m plain)))
               3 (Believes! m (Believes! g (hide m elaborate)))
               4 (if (Believes! m (Believes! g (hide m elaborate))) (hide m plain))
               5 (if (Believes! m (Believes! g (hide m plain))) (hide m elaborate))
               6 (Believes! m (Believes! g (hide m elaborate)))
               7 (Believes! d (if (Believes! m (Believes! g (hide m elaborate))) (hide m plain)))
               8 (Believes! d (if (Believes! m (Believes! g (hide m plain))) (hide m elaborate)))
               9 (Believes! d (Believes! m (Believes! g (hide m elaborate))))}


 :goal (Believes! d (hide m plain))}
```

Differences from Prior Work

- Expressivity

1. Can handle as goals **states of minds** of other agents

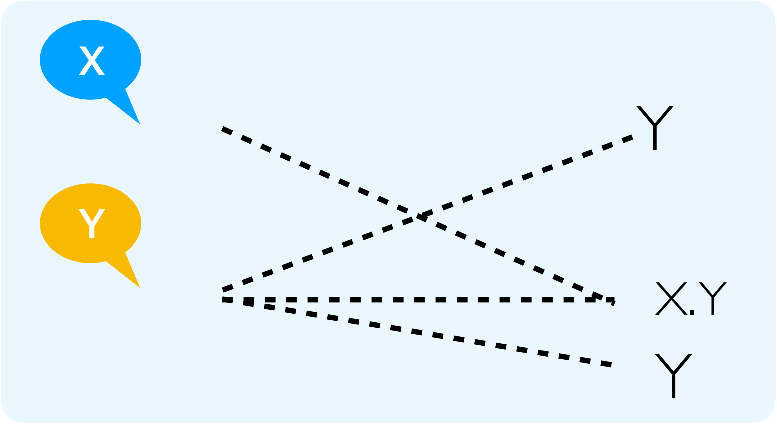
2. **Quantifiers** enable succinct representation of large domains.

```
{:name        "The Purloined Letter"
 :description "Dupin's reasoning as he goes through the case"

 :assumptions {1 (Believes! g (hide m elaborate))
               2 (Believes! d (or (hide m elaborate) (hide m plain)))
               3 (Believes! m (Believes! g (hide m elaborate)))
               4 (if (Believes! m (Believes! g (hide m elaborate))) (hide m plain))
               5 (if (Believes! m (Believes! g (hide m plain))) (hide m elaborate))
               6 (Believes! m (Believes! g (hide m elaborate)))
               7 (Believes! d (if (Believes! m (Believes! g (hide m elaborate))) (hide m plain)))
               8 (Believes! d (if (Believes! m (Believes! g (hide m plain))) (hide m elaborate)))
               9 (Believes! d (Believes! m (Believes! g (hide m elaborate))))}


 :goal (Believes! d (hide m plain))}
```

These model an infinite domain

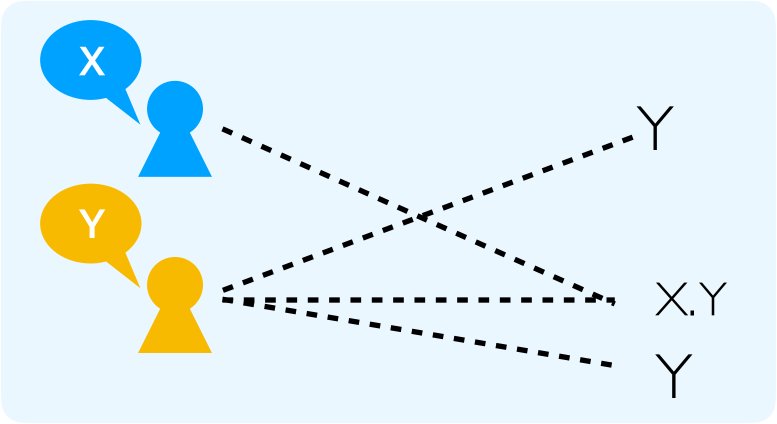
$$\forall x \exists y \mathbf{R}(x,y) \wedge$$
$$\forall x, y \neg (\mathbf{R}(x,y) \wedge \mathbf{R}(y,x)) \wedge$$
$$\forall x, y, z (\mathbf{R}(x,y) \wedge \mathbf{R}(y,z)) \rightarrow \mathbf{R}(x,z)$$

Application: Cognitive Multi-Agent Pathing



Cognitive Multi-Agent Path Finding

Application: Cognitive Multi-Agent Pathing

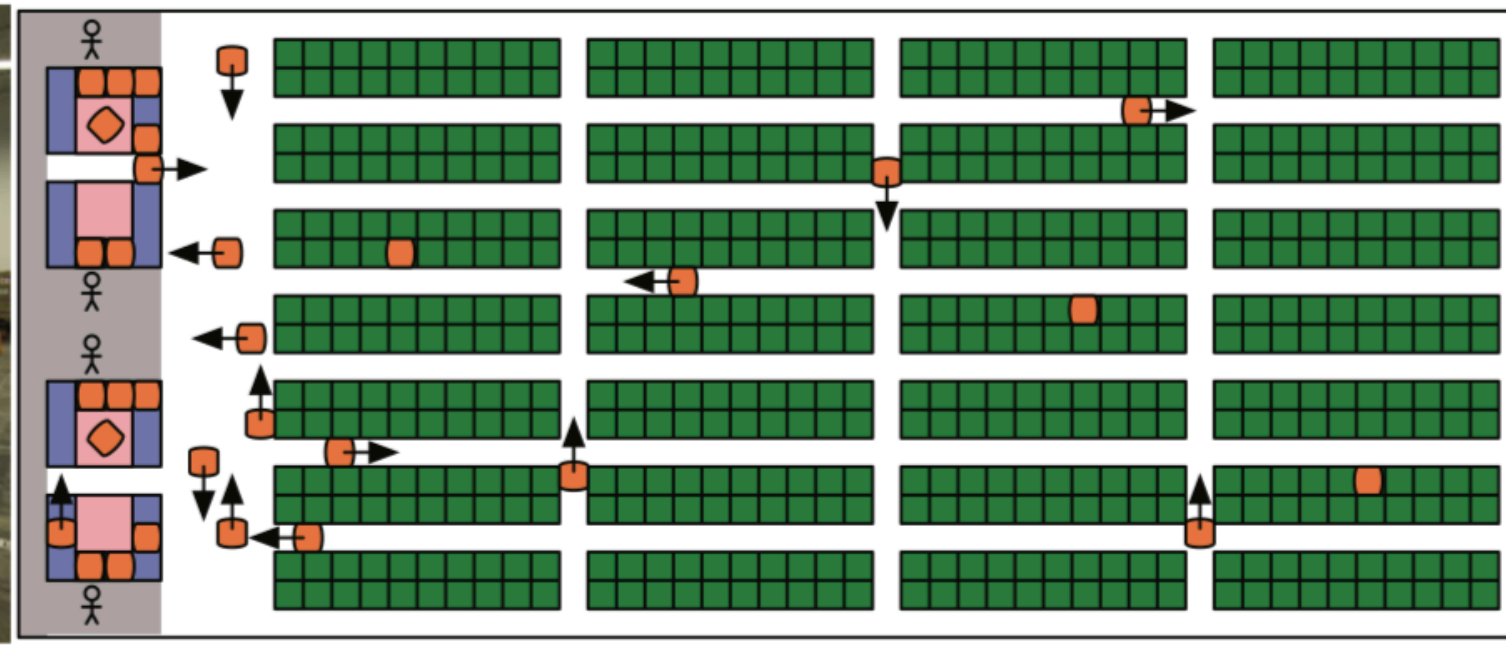


Cognitive Multi-Agent Path Finding

Multi-Agent Pathing

- Given a set of agents **A** and a set of goal locations **G** with a mapping from *g*(A)$\subset$**G**, find collision-free paths such that each agent *a* reaches a location in *g*(a) and minimizes some given objective condition.

Multi-Agent Pathing



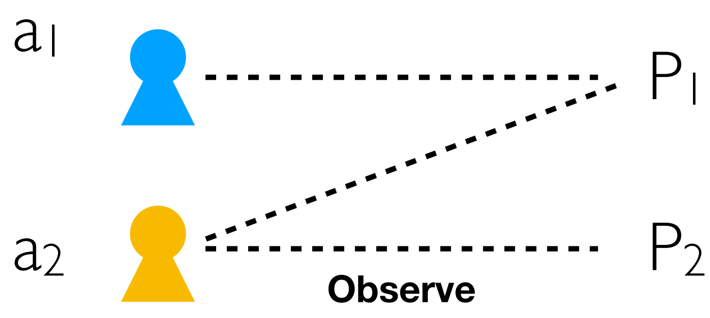
Hang Ma, Sven Koenig, Nora Ayanian, Liron Cohen, Wolfgang Hoenig, T. K. Satish Kumar, Tansel Uras, Hong Xu, Craig Tovey, and Guni Sharon. **Overview: generalizations of multi-agent path finding to real-world scenarios.** In the 25th International Joint Conference on Artificial Intelligence (IJCAI) Workshop on Multi-Agent Path Finding. 2016.

Cognitive Multi-Agent Pathing

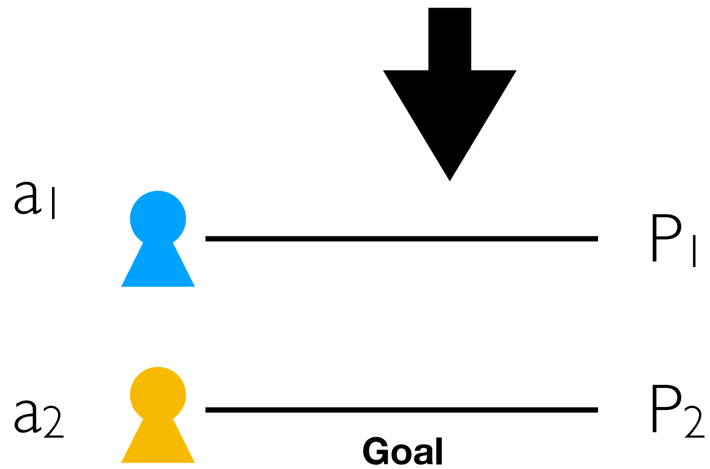
- Given a set of agents **A** and a set of goal locations **G** with a mapping from *g*(A)⊂**G**, find collision-free paths such that each agent *a* reaches a location in *g*(a) and minimizes some given objective condition.

- *g* is defined using a cognitive calculi.

Cognitive Multi-Agent Pathing



$$\mathbf{B}(a_1, \mathbf{B}(a_2, Goal(a_2, P_2))) \rightarrow Goal(a_1, P_1)$$
$$\mathbf{B}(a_1, Sees(P_2) \rightarrow \mathbf{B}(a_2, Goal(a_2, P_2)))$$
$$\mathbf{C}(\forall p : Sees(p))$$
$$Sees(P_2) \rightarrow Goal(a_2, P_2)$$
$$\forall p : Goal(a_2, p) \rightarrow \mathbf{B}(a_2, Goal(a_2, p))$$



Two Demos

Multi-Agent Planning Reasoning

Cognitive-Polysolid World

4. Cognitive-Polysolid Framework

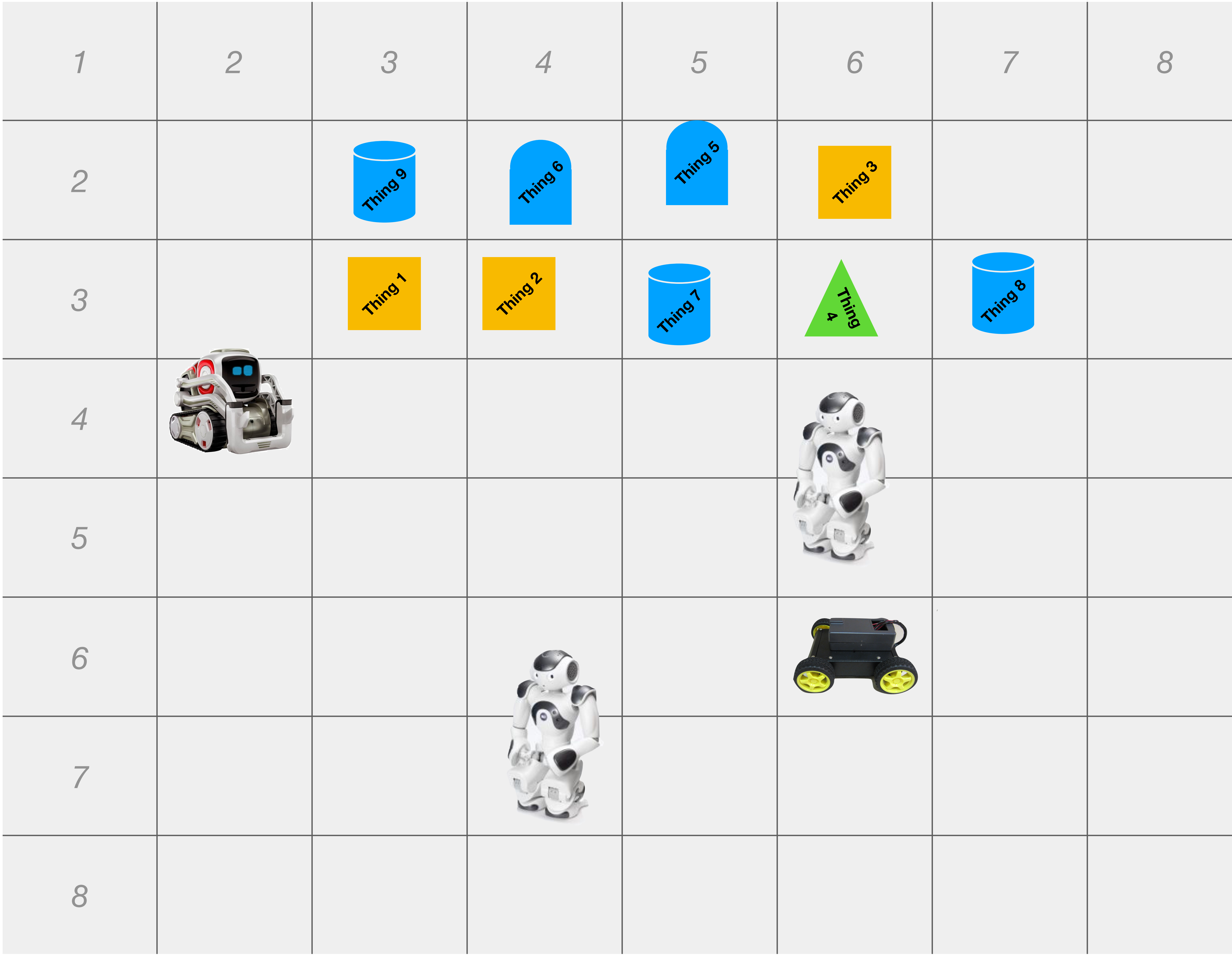
We now introduce the *cognitive-polysolid framework* (CPF), a class of problems that we use for experiments. The class of problems are based on a collection of polysolids and require cognitive intelligence to solve them. Polysolids are 3D shapes that do not contain any holes or gaps in them. Cubes, cylinders, spheres, triangular prisms, etc. are polysolids. (Please see the appendix for an elaboration of what we refer to as cognitive intelligence.) Using the framework, we can generate *cognitive-polysolid world instantiations*. Each instantiation has a number of polysolids. An instantiation also declares the properties of the polysolids in it, declares how these polysolids can be moved, and also specifies any agents and their possible beliefs or knowledge about the polysolids and other agents.

CPF subsumes the familiar "blocks world," described for instance in (Nilsson, 1980), which has long been used for reasoning and planning tasks. The framework gives us both a physical and cognitive domain unlike the purely physical blocks world domain. (The formal logic used in (Nilsson, 1980) is purely extensional, as it is simply first-order logic.) Since the physical complexities of blocks world problems have been well explored (Gupta & Nau, 1991; Slaney & Thiébaux, 2001), we emphasize the cognitive extensions of it.

A cognitive-polysolid world instantiation contains some finite number of blocks and a table large enough to hold all of them. Each block is `on` one other object; that object can be another block or the table. A block is said to be `clear` if there is no block that is on top of it. To move the blocks, an agent can either `stack` (placing a block on the table on top of another block) or `unstack` (taking a block that is on top of another block and placing it on the table). Before stacking the blocks, both need to be clear; when unstacking, the top block must be clear beforehand. After stacking the blocks, the bottom block is then not clear, and after unstacking, it is then clear. Translating this description to the $\mathcal{DCEC}$, we add two additional sorts and a constant, as well as some new functions:

$\textsf{Surface} \sqsubset \textsf{Object}$	$clear : \textsf{Block} \rightarrow \textsf{Fluent}$
$\textsf{Block} \sqsubset \textsf{Surface}$	$goal : \textsf{Formula} \times \textsf{Number} \rightarrow \textsf{Formula}$
$table : \textsf{Surface}$	$stack : \textsf{Block} \times \textsf{Block} \rightarrow \textsf{ActionType}$
$on : \textsf{Block} \times \textsf{Surface} \rightarrow \textsf{Fluent}$	$unstack : \textsf{Block} \times \textsf{Block} \rightarrow \textsf{ActionType}$

Peveler, M., G., Naveen S., Bringsjord, S., Sen, A. et al. Toward Cognitive-and-Immersive Systems: Experiments in a Cognitive Microworld. Forthcoming in the *Proceedings of the 6th Annual Conference on Advances in Cognitive Systems*, Stanford, CA, USA, 2018.



1
2
3
4
5
6
7
8
Thing 9
Thing 6
Thing 5
Thing 3
Thing 1
Thing 2
Thing 7
Thing 4
Thing 8

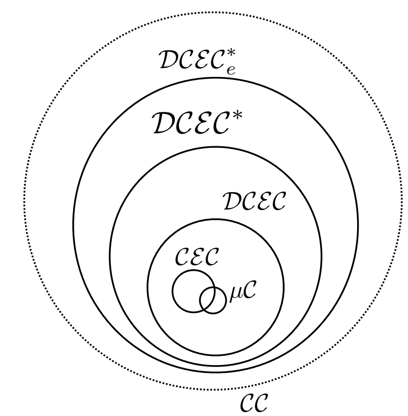
$$
\forall x \forall y \forall z \forall z' \left\{
\begin{array}{c}
\left[
\begin{array}{c}
x \neq y \wedge y \neq z \wedge x \neq z \\
\wedge \\
Cx \wedge Cy \wedge Cz \\
\wedge \\
Tz' \\
\wedge \\
\exists w_1 \exists w_2 \ (w_1 \neq w_2 \wedge Aw_1 \wedge Aw_2 \wedge Gz'w_1 \wedge Gz'w_2) \\
\wedge \\
\forall u_1 \forall u_2 \forall u_3 \quad \left(
\begin{array}{c}
\big[Gz'u_1 \wedge Gz'u_2 \wedge Gz'u_3 \wedge C^b u_1 \wedge C^b u_2 \wedge C^b u_3\big] \\
\rightarrow \\
\forall v \big[(Gz'v \wedge C^b v) \rightarrow (v = u_1 \vee v = u_2 \vee v = u_3)\big]
\end{array}
\right)
\end{array}
\right] \\
\rightarrow \\
(Gxz' \wedge Gyz' \wedge Gzz')
\end{array}
\right\}
$$


1
2
3
4
5
6
7
8
2
3
4
5
6
7
8
Thing 5
Thing 7
Thing 6
Thing 1
Thing 4
Thing 3
Thing 8
Thing 2
Thing 9

[redacted]

Creativity via More Expressive Planning

Spectra - Planner



Background Formulae Γ →

Initial State Formula σ_0 →

Action Definitions

$\alpha_1(x_1, \ldots, x_n)$

$\alpha_2(x_1, \ldots, x_n)$

$\ldots$

$\alpha_n(x_1, \ldots, x_n)$ →

Spectra

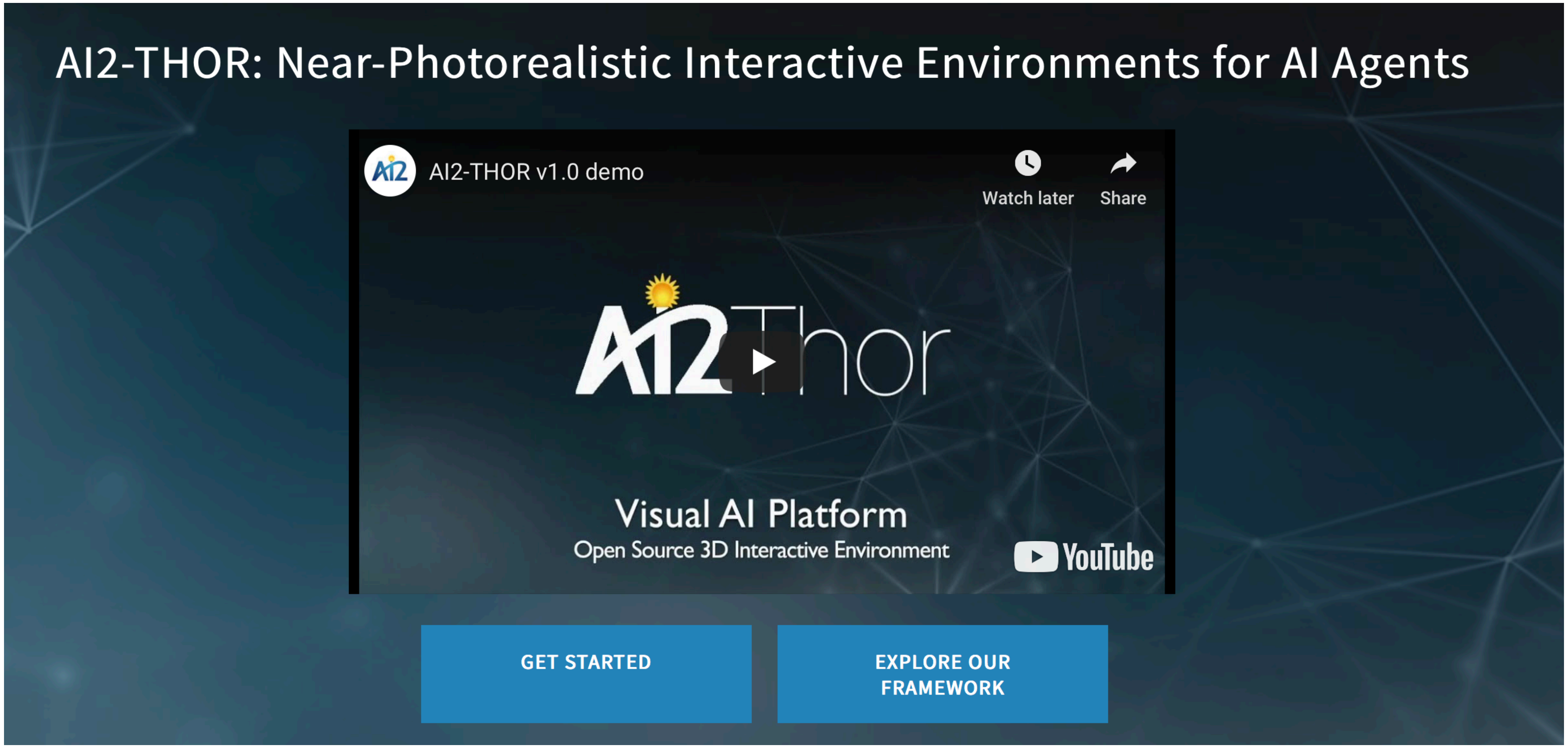
$\rho_1, \rho_2, \ldots$

Plans

g ↑

Goal

AI2Thor



AI2-THOR: Near-Photorealistic Interactive Environments for AI Agents
AI2-THOR v1.0 demo
Watch later
Share
AI2Thor
Visual AI Platform
Open Source 3D Interactive Environment
YouTube
GET STARTED
EXPLORE OUR FRAMEWORK

Object Type	Open/Close	Pickupable	On/Off	Receptacle
AlarmClock	No	YES	No	No
Apple	No	YES	No	No
ArmChair	No	No	No	YES
BaseballBat	No	YES	No	No
BasketBall	No	YES	No	No
Bathtub	No	No	No	YES
BathtubBasin	No	No	No	YES
Bed	No	No	No	YES
Blinds	No	No	No	No
Book	No	YES	No	No
Boots	No	YES	No	No
Bottle	No	YES	No	No
Bowl	No	YES	No	YES
Box	No	YES	No	YES
Bread	No	YES	No	No
ButterKnife	No	YES	No	No
Cabinet	YES	No	No	YES

Receptacle	Object Types
Pot	Apple, ButterKnife, DishSponge, Egg, Fork, Knife, Ladle, Lettuce, Potato, Spatula, Spoon, Tomato
Pan	Apple, ButterKnife, DishSponge, Egg, Fork, Knife, Ladle, Lettuce, Potato, Spatula, Spoon, Tomato
Bowl	Apple, ButterKnife, DishSponge, Egg, Fork, Knife, Ladle, Lettuce, Potato, Spatula, Spoon, Tomato
CoffeeMachine	Mug
Microwave	Apple, Bowl, Bread, Cup, Egg, Mug, Plate, Potato, Tomato
StoveBurner	Kettle, Pan, Pot
Fridge	Apple, Bottle, Bowl, Bread, Cup, Egg, Lettuce, Mug, Pan, Plate, Pot, Potato, Tomato, WineBottle
Mug	ButterKnife, Fork, Knife, Pen, Pencil, Spoon
Plate	Apple, ButterKnife, DishSponge, Egg, Fork, Knife, Ladle, Lettuce, Mug, Potato, Spatula, Spoon, Tomato
Cup	ButterKnife, Fork, Spoon
Sofa	BasketBall, Book, Box, CellPhone, Cloth, CreditCard, KeyChain, Laptop, Newspaper, Pillow, RemoteControl, TeddyBear
ArmChair	BasketBall, Book, Box, CellPhone, Cloth, CreditCard, KeyChain, Laptop, Newspaper, Pillow, RemoteControl, TeddyBear

Scene

- <u>Goal</u>: Place the **knife** in the **microwave**

Goal

- Place the **knife** in the **microwave**
  - But the microwave can accept only certain objects

Receptacle	Object Types
Pot	Apple, ButterKnife, DishSponge, Egg, Fork, Knife, Ladle, Lettuce, Potato, Spatula, Spoon, Tomato
Pan	Apple, ButterKnife, DishSponge, Egg, Fork, Knife, Ladle, Lettuce, Potato, Spatula, Spoon, Tomato
Bowl	Apple, ButterKnife, DishSponge, Egg, Fork, Knife, Ladle, Lettuce, Potato, Spatula, Spoon, Tomato
CoffeeMachine	Mug
Microwave	Apple, Bowl, Bread, Cup, Egg, Mug, Plate, Potato, Tomato

Goal

- Place the **knife** in the **microwave**
  - But the microwave can accept only certain objects

Receptacle	Object Types
Pot	Apple, ButterKnife, DishSponge, Egg, Fork, Knife, Ladle, Lettuce, Potato, Spatula, Spoon, Tomato
Pan	Apple, ButterKnife, DishSponge, Egg, Fork, Knife, Ladle, Lettuce, Potato, Spatula, Spoon, Tomato
Bowl	Apple, ButterKnife, DishSponge, Egg, Fork, Knife, Ladle, Lettuce, Potato, Spatula, Spoon, Tomato
CoffeeMachine	Mug
Microwave	Apple, Bowl, Bread, Cup, Egg, Mug, Plate, Potato, Tomato

One Major Issue

- Lack of real and concrete domains.
- One solution: test and train TAI in micro domains.
- Today’s ML can’t solve anomalous problems.

Research Question

- Can we train a machine learning system to reason over these problems?

Existing datasets

Existing datasets

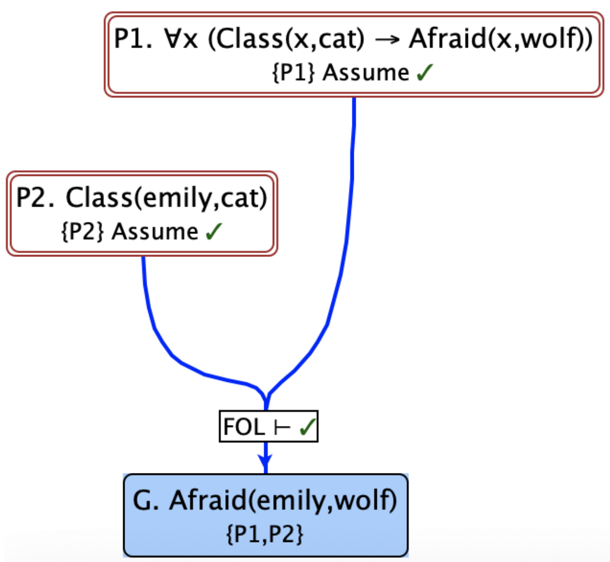
- Are either too complex (e.g. Mizar 60,000 theorems from mathematics) or too simple (bAbI dataset)

Existing datasets

- Are either too complex (e.g. Mizar 60,000 theorems from mathematics) or too simple (bAbI dataset)

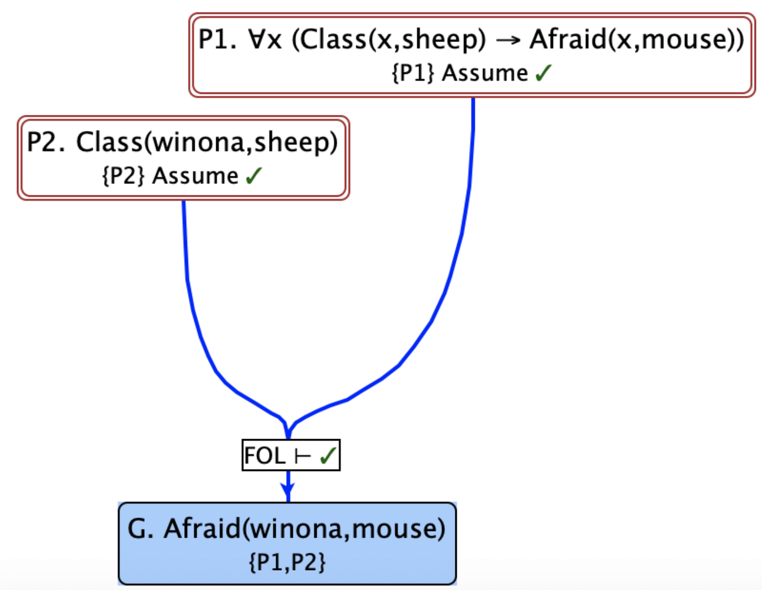
- Or require background knowledge.

Two Problems from bAbI



SNARK Proof

Line	Formula	Justification	Premise
1	¬Afraid(emily,wolf)	negated_conjecture	
2	¬Class(X,cat) ∨ Afraid(X,wolf)	assertion	∀x (Class(x,cat) → Afraid(x,wolf))
3	Class(emily,cat)	assertion	Class(emily,cat)
4	Afraid(emily,wolf)	(resolve 2 3)	
5	$$FALSE	(rewrite 1 4)	



SNARK Proof

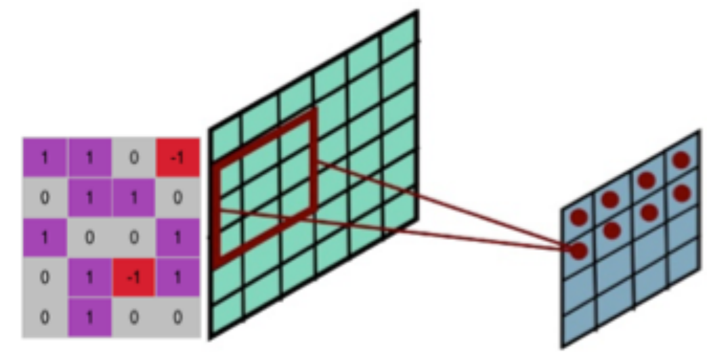
Line	Formula	Justification	Premise
1	¬Afraid(winona,mouse)	negated_conjecture	
2	¬Class(X,sheep) ∨ Afraid(X,mouse)	assertion	∀x (Class(x,sheep) → Afraid(x,mouse))
3	Class(winona,sheep)	assertion	Class(winona,sheep)
4	Afraid(winona,mouse)	(resolve 2 3)	
5	$$FALSE	(rewrite 1 4)	

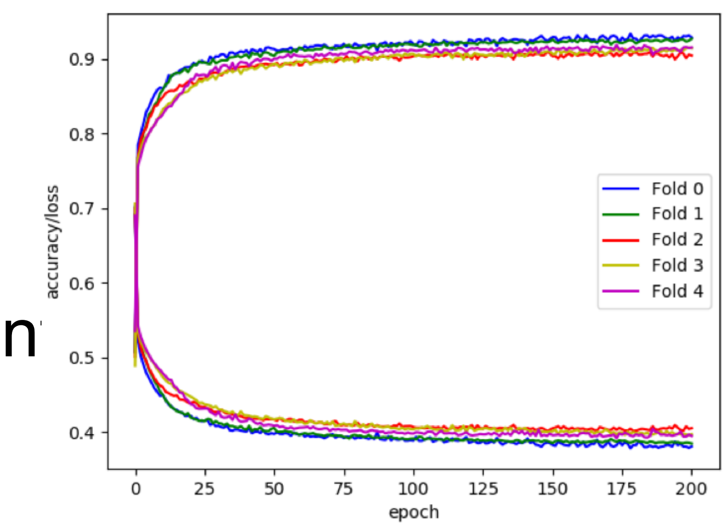
Architecture A

Logical CNN

Result

- 5-fold cross validation
- Approx. 91.8% accuracy with 4x4 on 5 runs
- Approx. 83.34% accuracy with 5x5 on 5 runs





Accuracy and Loss curve on 4x4

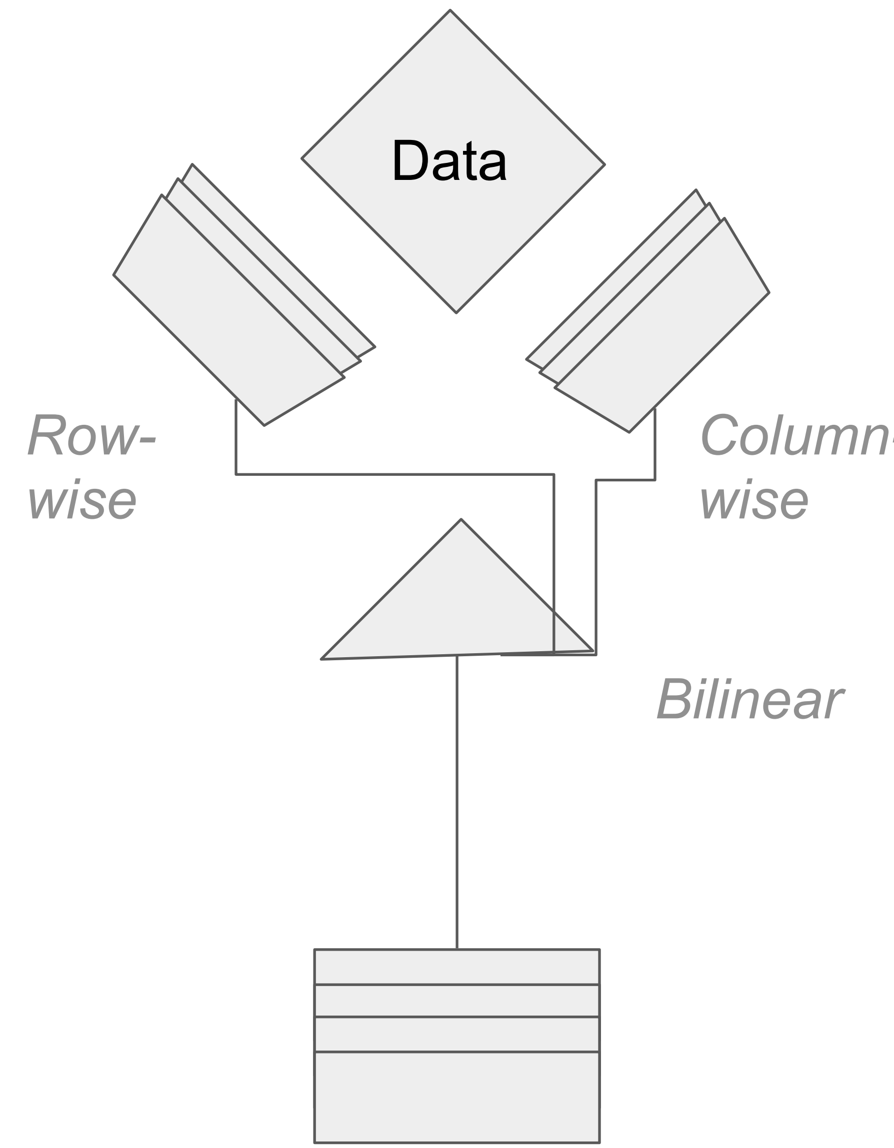
Analysis

- Standard CNN model learns but only to some exten
- Biased towards syntactic structure
- Fails to learn implicit semantic information

Lead: Shreya Banerjee

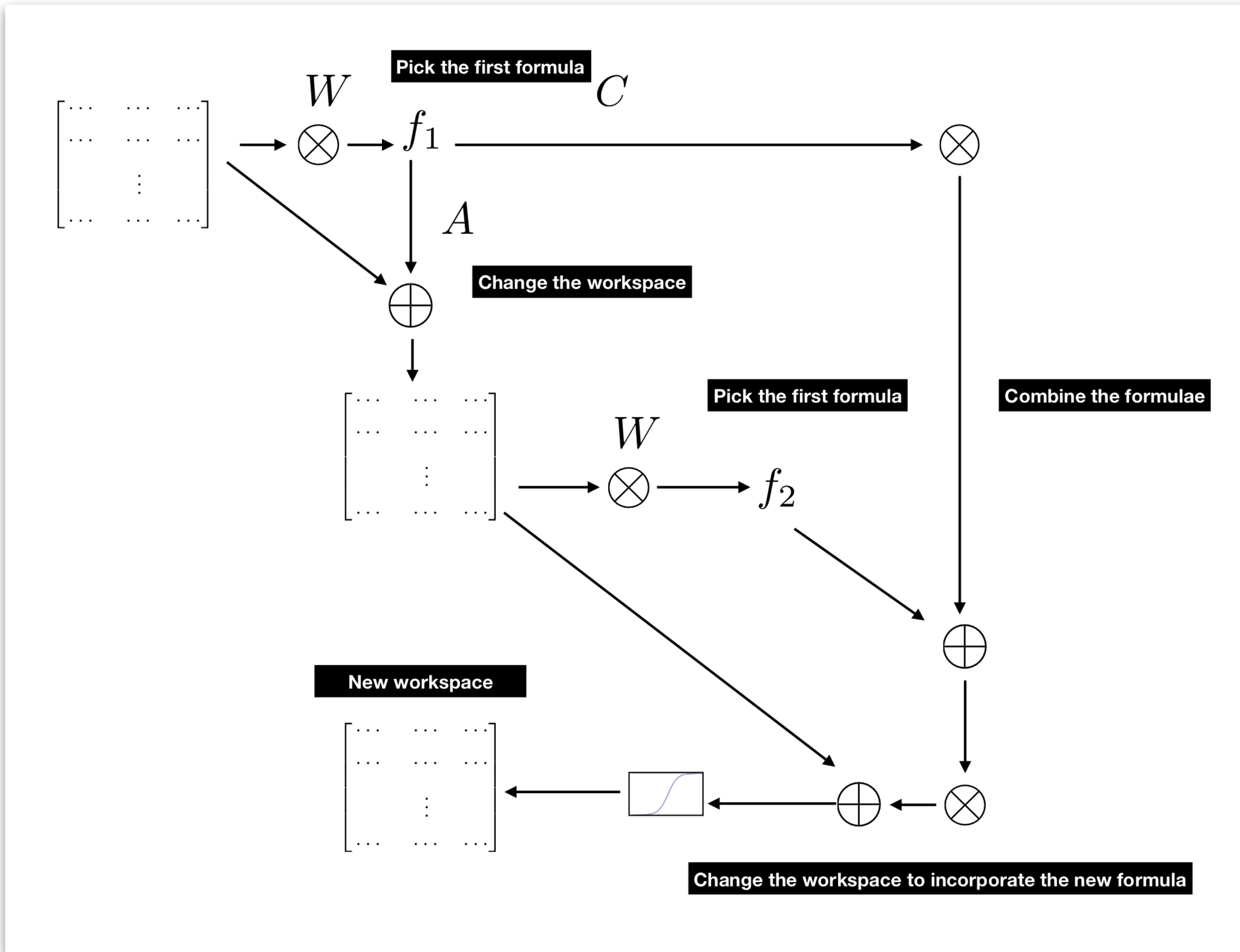
Architecture B

- Branched row-wise & column-wise network
- Average of ~90% final accuracy on 5x5x5 data
- Similar to the truth tree method for refutation?

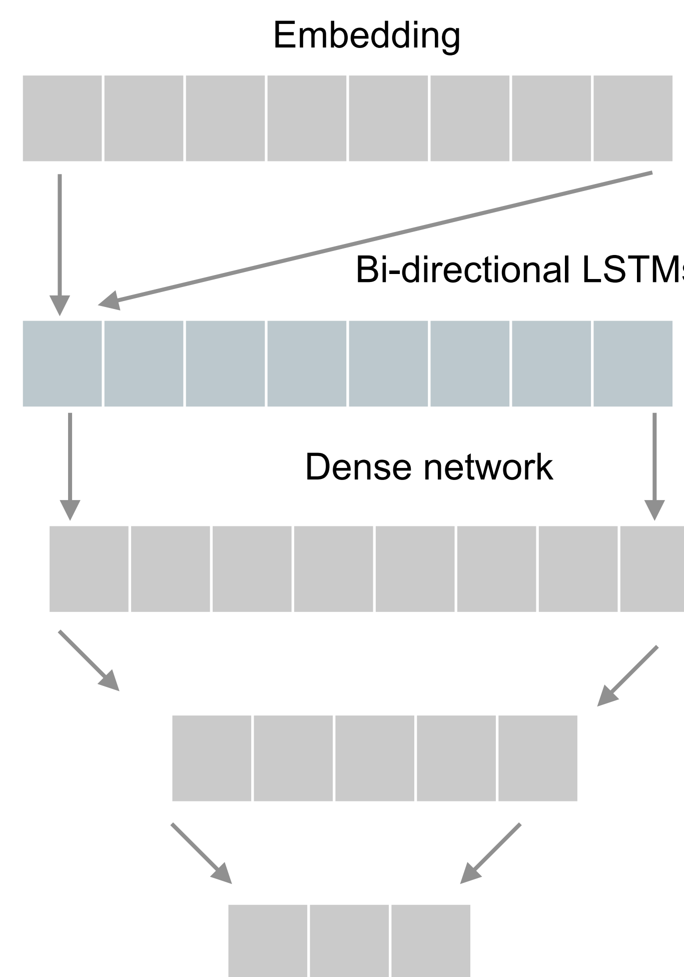


Lead: Jean-Claude Paquin

Architecture C



Architecture mimicking a theorem prover



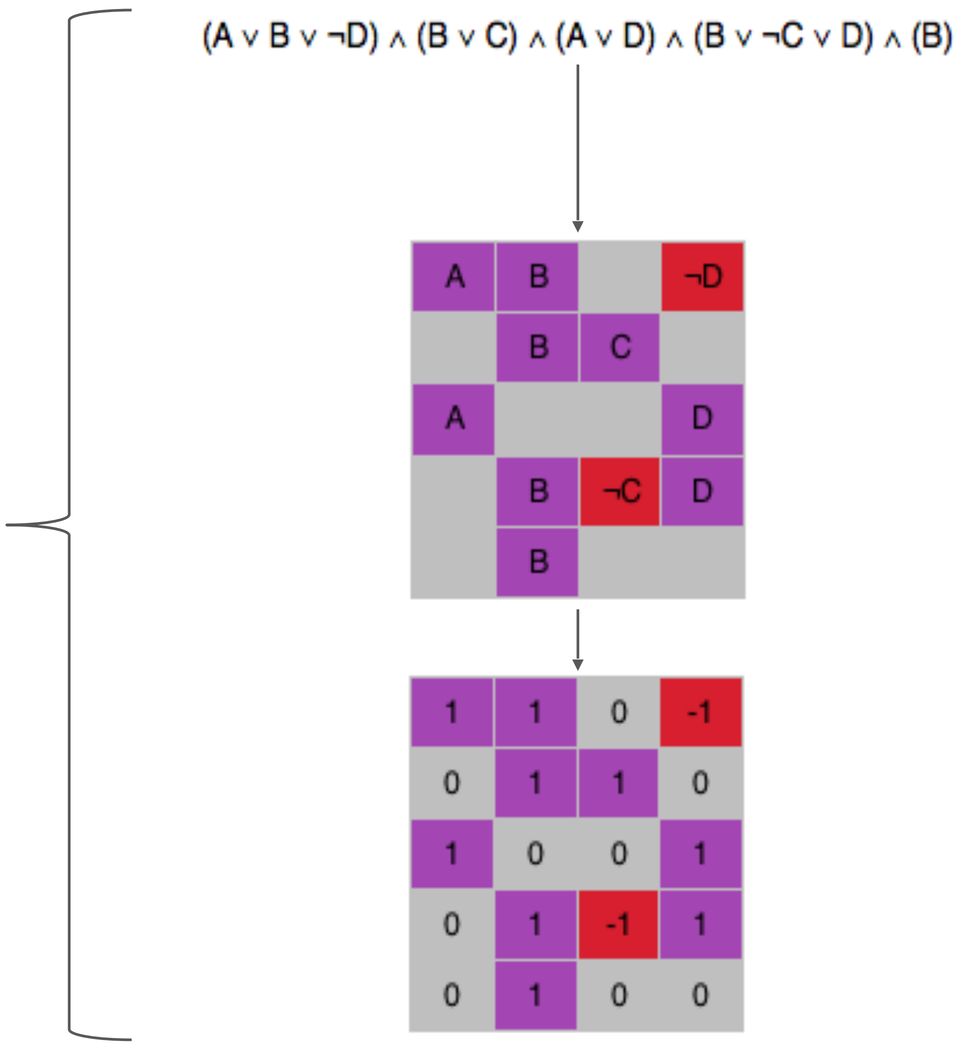
Lead: Naveen Sundar Govindarajulu

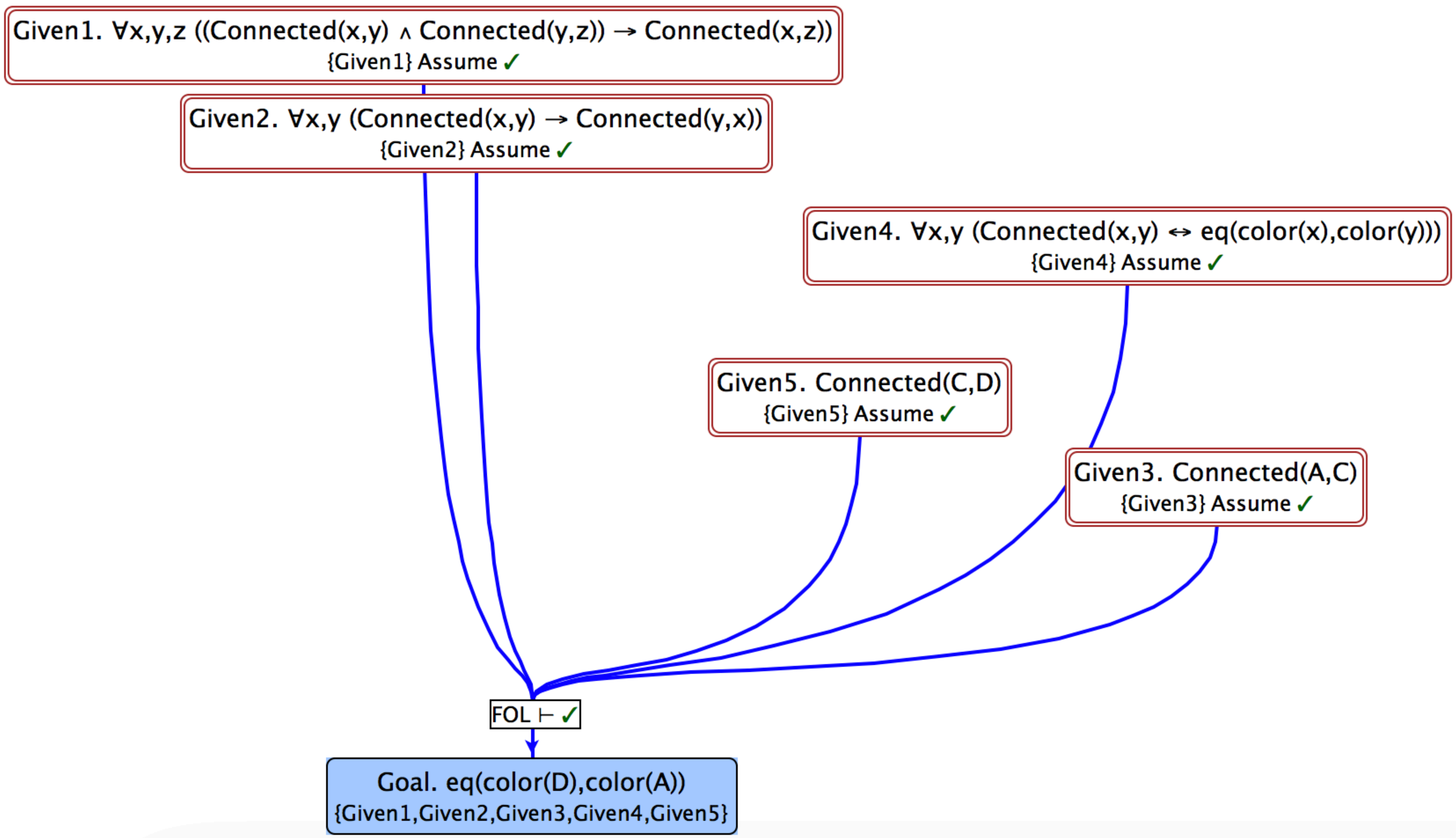
Datasets

4 x 4 x 4 *32000 train* *8000 test*	**4 formulae and max 4 propositional variables**
5 x 5 x 5 *32000 train* *8000 test*	**5 formulae and max 5 propositional variables**

Vectorizing Data for Propositional Logic

CNF to matrices



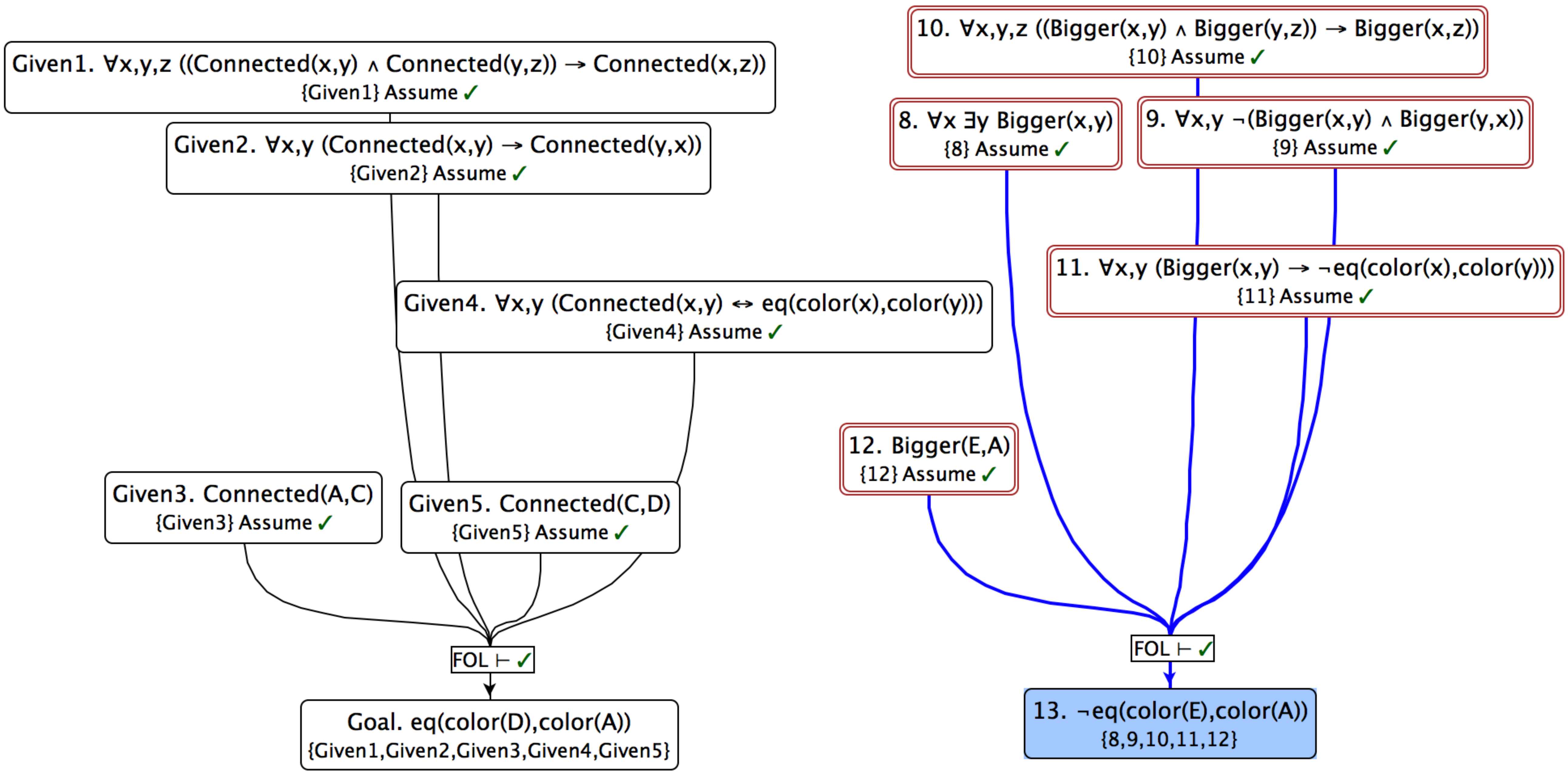


SNARK Proof

Li...	Formula	Justification	Premise
1	¬eq(color(D),color(A))	negated_conjecture	
2	¬Connected(X,Y) ∨ eq(color(X),color(Y))	assertion	∀x,y (Connected(x,y) ↔ eq(color(x),color(y)))
3	¬Connected(D,A)	(resolve 1 2)	
4	¬Connected(X,Y) ∨ ¬Connected(Y,Z) ∨ Connected(X,Z)	assertion	∀x,y,z ((Connected(x,y) ∧ Connected(y,z)) → Connected(x,z))
5	Connected(A,C)	assertion	Connected(A,C)
6	Connected(C,D)	assertion	Connected(C,D)
7	Connected(D,A)	(hyperresolve 4 5 6)	
8	$$FALSE	(rewrite 3 7)	

$$\forall x \exists y \mathbf{R}(x, y) \wedge$$
$$\forall x, y \neg (\mathbf{R}(x, y) \wedge \mathbf{R}(y, x)) \wedge$$
$$\forall x, y, z (\mathbf{R}(x, y) \wedge \mathbf{R}(y, z)) \rightarrow \mathbf{R}(x, z)$$

Has only infinite models


Given1. ∀x,y,z ((Connected(x,y) ∧ Connected(y,z)) → Connected(x,z))
{Given1} Assume ✓
Given2. ∀x,y (Connected(x,y) → Connected(y,x))
{Given2} Assume ✓
Given4. ∀x,y (Connected(x,y) ↔ eq(color(x),color(y)))
{Given4} Assume ✓
Given3. Connected(A,C)
{Given3} Assume ✓
Given5. Connected(C,D)
{Given5} Assume ✓
FOL ⊢ ✓
Goal. eq(color(D),color(A))
{Given1,Given2,Given3,Given4,Given5}
10. ∀x,y,z ((Bigger(x,y) ∧ Bigger(y,z)) → Bigger(x,z))
{10} Assume ✓
8. ∀x ∃y Bigger(x,y)
{8} Assume ✓
9. ∀x,y ¬(Bigger(x,y) ∧ Bigger(y,x))
{9} Assume ✓
11. ∀x,y (Bigger(x,y) → ¬eq(color(x),color(y)))
{11} Assume ✓
12. Bigger(E,A)
{12} Assume ✓
FOL ⊢ ✓
13. ¬eq(color(E),color(A))
{8,9,10,11,12}

Results on Propositional Logic

	Benchmark	Architecture A	Architecture B	Architecture C
4 x 4 x 4 *32000 train* *8000 test*	95.73%	91.8%	~85%	~93%
5 x 5 x 5 *32000 train* *8000 test*	85.66%	83.34%	~90%	~90%

Accuracy

Results on First-order Logic with Function Symbols

	Benchmark	Architecture A	Architecture B	Architecture C
6 premises *15000 train* *5000 test*	~50%	—	—	~80%

Accuracy

Results on First-order Logic with Function Symbols And Infinite Domains

	Benchmark	Architecture A	Architecture B	Architecture C
6 premises *15000 train* *5000 test*	~50%	—	—	~61%

Accuracy

Future Work

- Micro-domains and synthetic data in home automation, smart buildings and cities
    - **Integration** of one more more common services with our reasoning systems and planners with a focus on:
        - *safety, ethics, efficiency*
- Visual Question Answering with Justifications:
    - Human-understandable justifications from visual scenes
- Assured and Verified Services
- Continuation of TAI
- Applications: Cognitive-multi Agent Path Finding

Thank you

Links

- http://kryten.mm.rpi.edu/TAI/tai.html
- Reasoner:
  - https://github.com/naveensundarg/prover

Other Demos

Example 1

Example 1

1. During your daily commute to work, an agent a_c in your car observes that there is more traffic than usual headed toward the local store.

Example 1

1. During your daily commute to work, an agent a_c in your car observes that there is more traffic than usual headed toward the local store.
2. It then consults a weather service and finds that a major storm is headed toward your town.

Example 1

1. During your daily commute to work, an agent a_c in your car observes that there is more traffic than usual headed toward the local store.
2. It then consults a weather service and finds that a major storm is headed toward your town.
3. a_c conveys this information to a_h, an agent in your home.

Example 1

1. During your daily commute to work, an agent a_c in your car observes that there is more traffic than usual headed toward the local store.
2. It then consults a weather service and finds that a major storm is headed toward your town.
3. a_c conveys this information to a_h, an agent in your home.
4. a_h then communicates with an agent a_p on your phone and finds out that you do not know about the storm coming your way, as you have not made any preparations for it; and as further evidence of your ignorance, you have not read any notifications about the storm.

Example 1

1. During your daily commute to work, an agent a_c in your car observes that there is more traffic than usual headed toward the local store.
2. It then consults a weather service and finds that a major storm is headed toward your town.
3. a_c conveys this information to a_h, an agent in your home.
4. a_h then communicates with an agent a_p on your phone and finds out that you do not know about the storm coming your way, as you have not made any preparations for it; and as further evidence of your ignorance, you have not read any notifications about the storm.
5. a_h then infers from your calendar that you may not have enough time to get supplies after you read your notifications later in the day.

Example 1

1. During your daily commute to work, an agent a_c in your car observes that there is more traffic than usual headed toward the local store.
2. It then consults a weather service and finds that a major storm is headed toward your town.
3. a_c conveys this information to a_h, an agent in your home.
4. a_h then communicates with an agent a_p on your phone and finds out that you do not know about the storm coming your way, as you have not made any preparations for it; and as further evidence of your ignorance, you have not read any notifications about the storm.
5. a_h then infers from your calendar that you may not have enough time to get supplies after you read your notifications later in the day.
6. a_h commands a_c to recommend to you a list of supplies to shop for on your way home, including at least n items in certain categories (e.g. 3 gallons of bottle water).

Example 1: Formulae

$$\mathbf{B}(a_c, t_o, crowded(store) \rightarrow unusual) \quad \mathsf{f}_1$$

$$\mathbf{P}(a_c, t_1, crowded(store)),\ \mathsf{f}_2$$

$$\forall t : \mathbf{O}\left(a_c, t, unusal, \; happens\big(action(a_c, check(weather)), t+1\big)\right) \quad \mathsf{f}_3$$

$$\forall t : \mathbf{B}\left(a_c, t, \mathsf{f}_3\right)$$

$$\forall a : \left(happens\Big(action(a, check(weather)), t_3\Big) \rightarrow \mathbf{K}(a, t_4, storm), \right) \quad \mathsf{f}_4$$

Example 1: Formulae

$$\forall t : \mathbf{O}\big(a_c, t, storm, \mathsf{S}(a_c, a_h, storm, t+1)\big), \ \mathsf{f}_5$$

$$\forall t : \mathbf{B}\left(a_c, t, \ \mathsf{f}_5\right)$$

Example 1: Formulae

$$\mathsf{S}(a_c, a_h, storm, t+1)$$

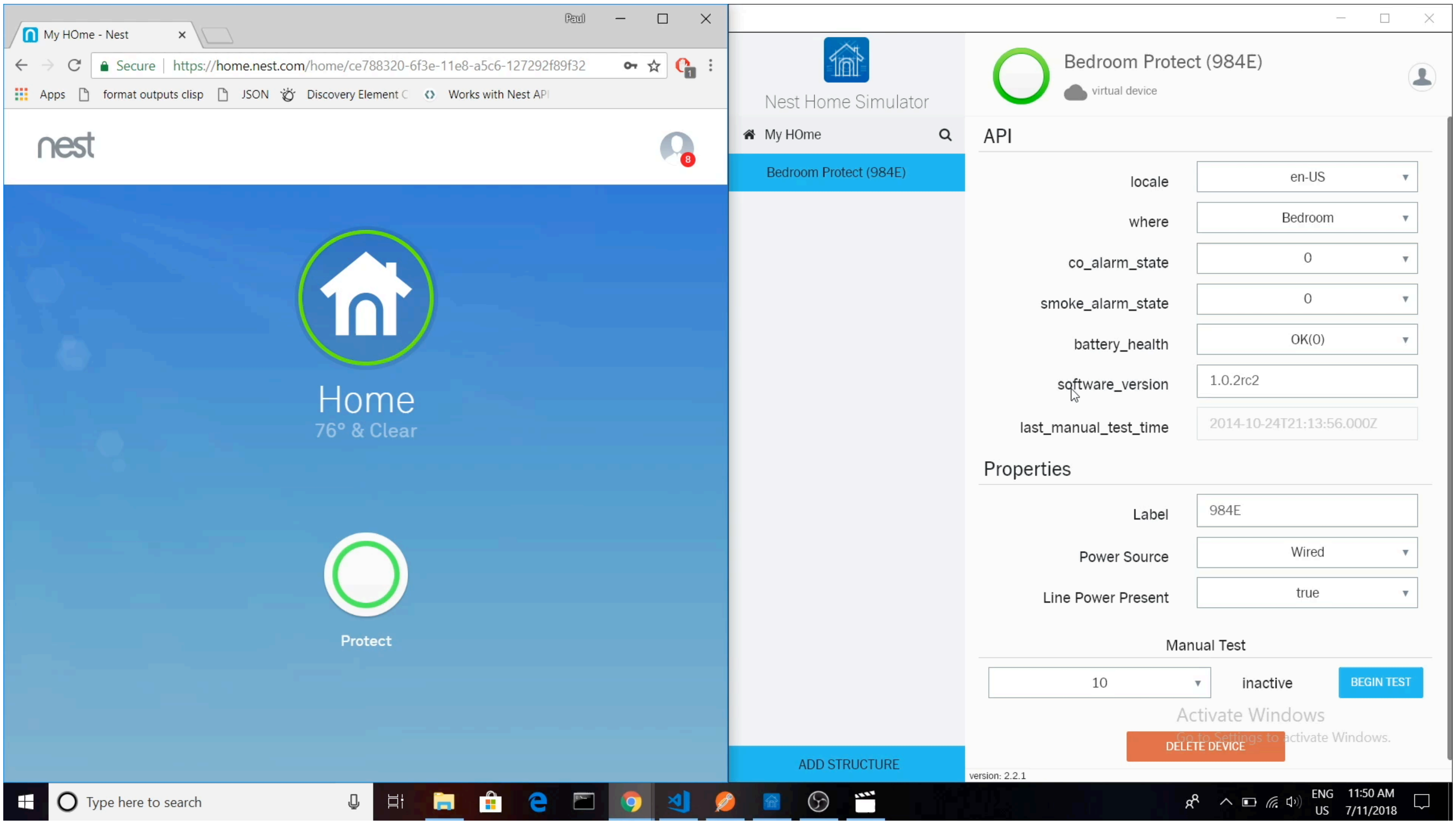
Example I: Formulae

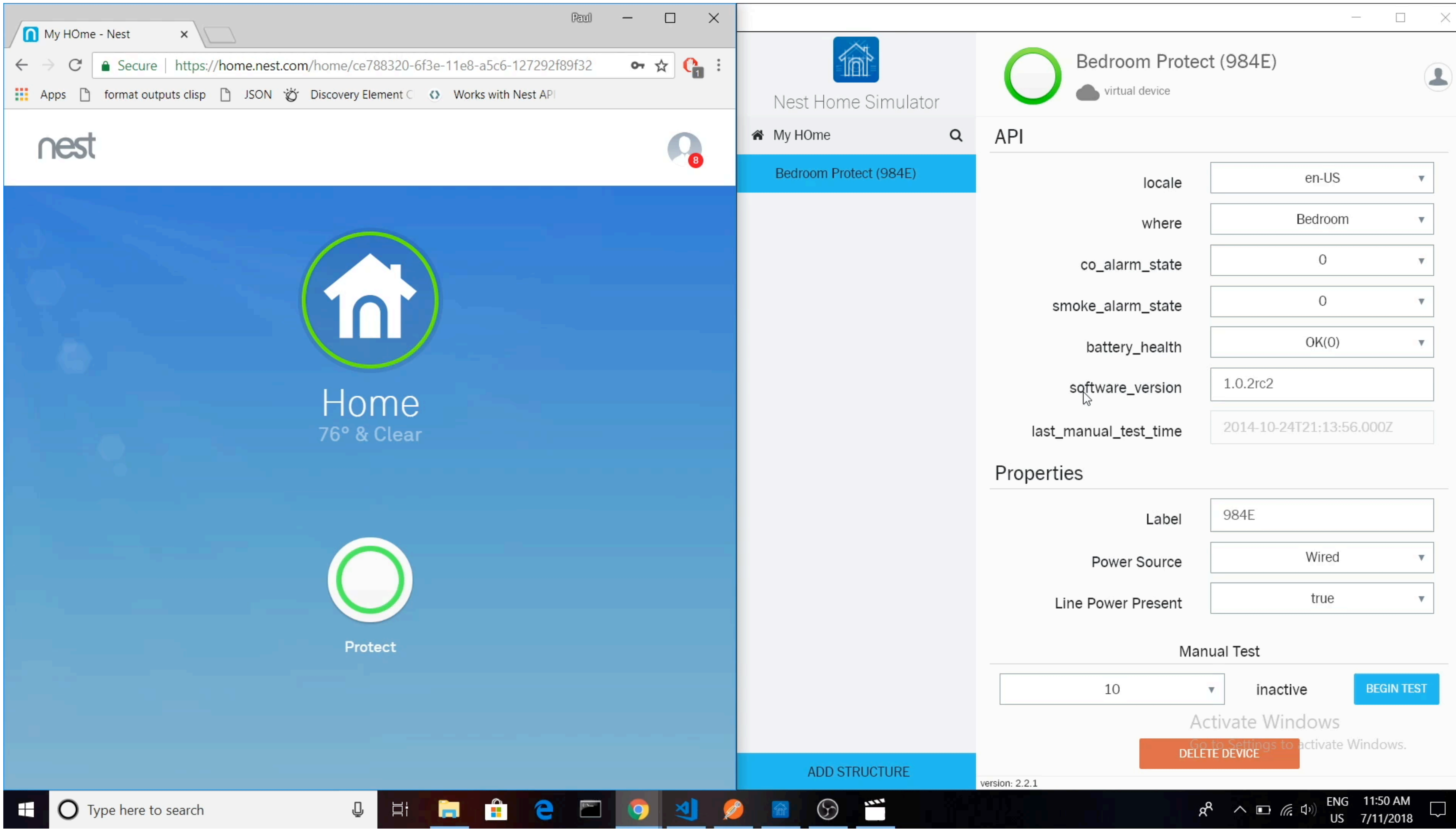
$$\forall t : \mathbf{O}\big(a_h, t, storm, \forall s : quantity(s) > 0\big), \quad \text{f}_6$$

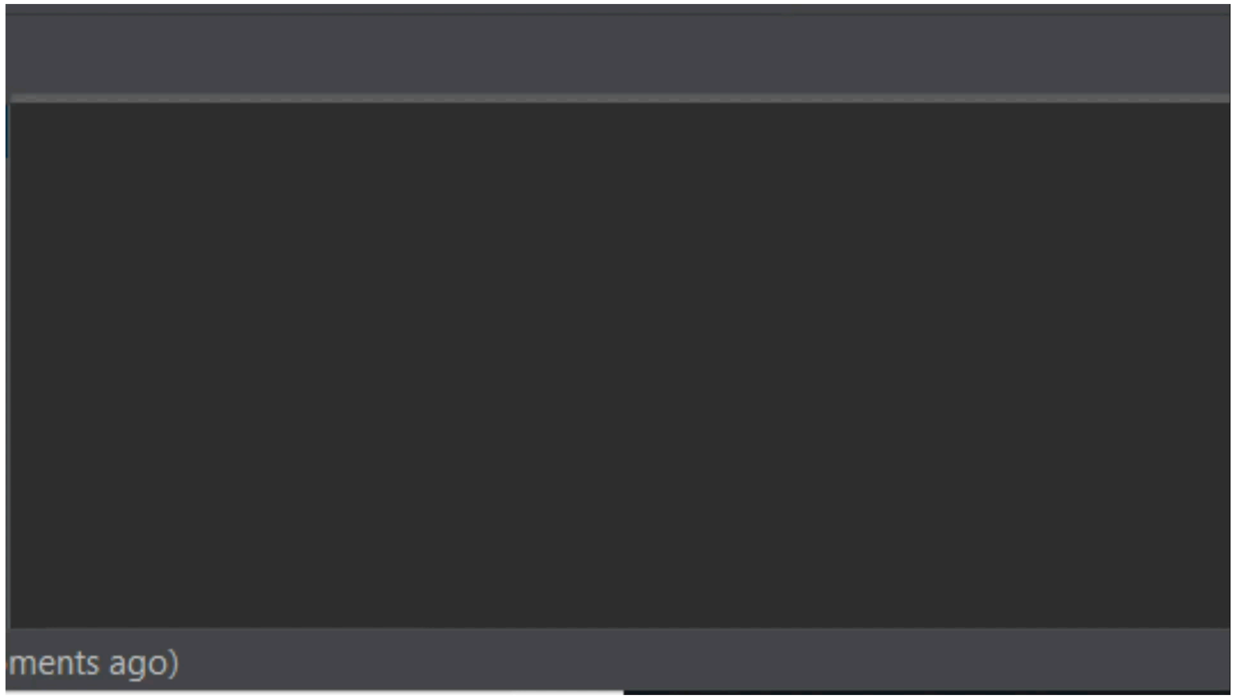
$$\mathbf{K}\left(\begin{array}{l} a_h, t_5, shops(j, today) \lor shops(j, tomorrow) \\ \qquad \rightarrow \forall s : quantity(s) > 0 \end{array}\right), \quad \text{f}_7$$

$$\forall t : \mathbf{B}\left(\begin{array}{l} a_h, t, happens\big(action(a_c, recc(shops(j))), t\big) \\ \qquad \rightarrow shops(j) \end{array}\right) \quad \text{f}_8$$

$$\forall t : \mathbf{B}\left(\begin{array}{l} a_h, t, happens\Big(action\big(a_h, req(a_c, shops(j))\big), t\Big) \\ \rightarrow happens\big(action(a_c, recc(shops(j))), t\big) \end{array}\right) \quad \text{f}_9$$


My HOme - Nest
Paul
Secure
https://home.nest.com/home/ce788320-6f3e-11e8-a5c6-127292f89f32
Apps
format outputs clisp
JSON
Discovery Element C
Works with Nest API
nest
Home
76° & Clear
Protect
Nest Home Simulator
My HOme
Bedroom Protect (984E)
ADD STRUCTURE
Bedroom Protect (984E)
virtual device
API
locale
en-US
where
Bedroom
co_alarm_state
0
smoke_alarm_state
0
battery_health
OK(0)
software_version
1.0.2rc2
last_manual_test_time
2014-10-24T21:13:56.000Z
Properties
Label
984E
Power Source
Wired
Line Power Present
true
Manual Test
10
inactive
BEGIN TEST
Activate Windows
Go to Settings to activate Windows.
DELETE DEVICE
version: 2.2.1
Type here to search
ENG
US
11:50 AM
7/11/2018


My HOme - Nest
Secure | https://home.nest.com/home/ce788320-6f3e-11e8-a5c6-127292f89f32
Apps
format outputs clisp
JSON
Discovery Element C
Works with Nest API
nest
Home
76° & Clear
Protect
Nest Home Simulator
My HOme
Bedroom Protect (984E)
ADD STRUCTURE
Bedroom Protect (984E)
virtual device
API
locale
en-US
where
Bedroom
co_alarm_state
0
smoke_alarm_state
0
battery_health
OK(0)
software_version
1.0.2rc2
last_manual_test_time
2014-10-24T21:13:56.000Z
Properties
Label
984E
Power Source
Wired
Line Power Present
true
Manual Test
10
inactive
BEGIN TEST
Activate Windows
Go to Settings to activate Windows.
DELETE DEVICE
version: 2.2.1
Type here to search
ENG
US
11:50 AM
7/11/2018



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

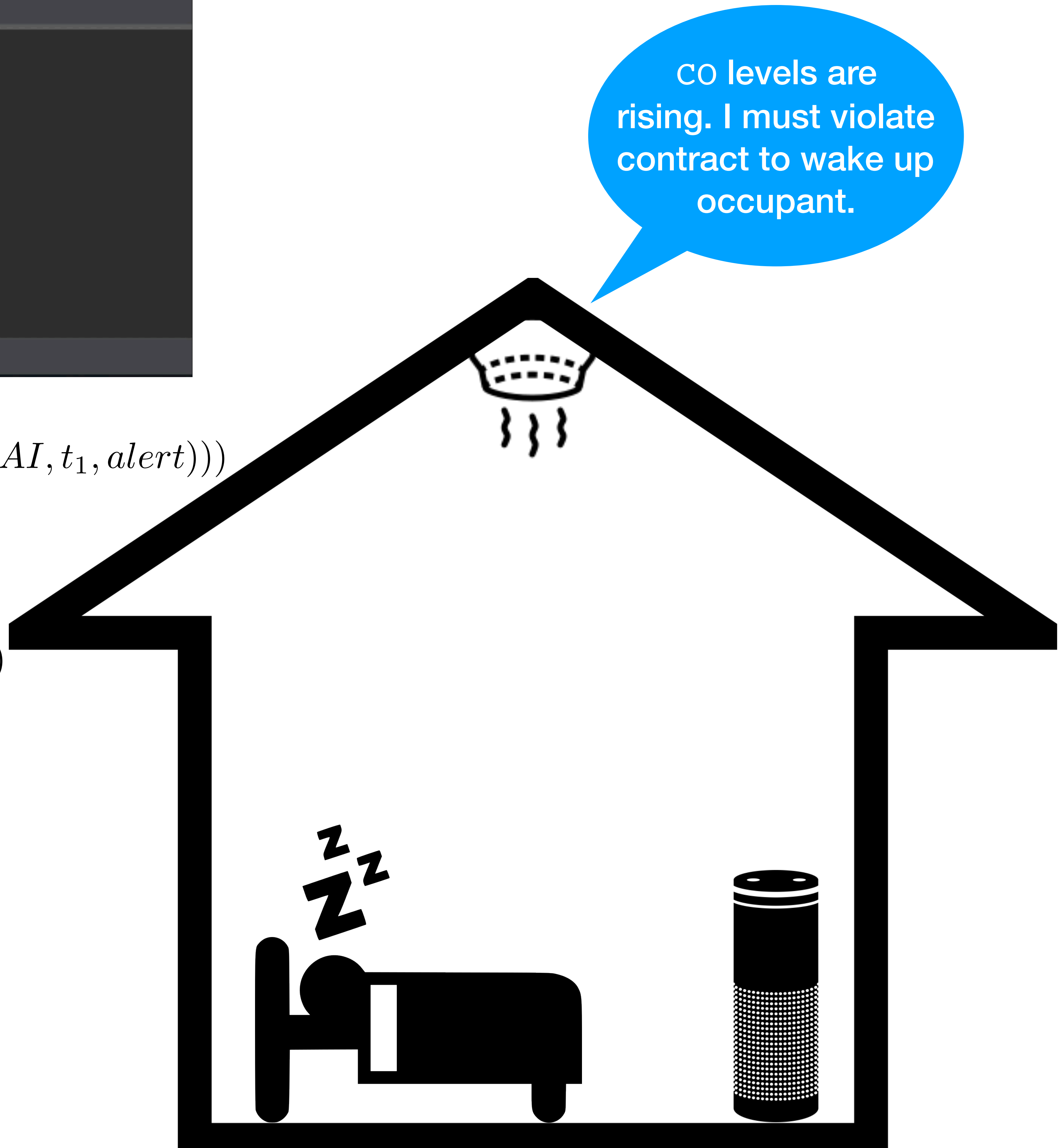
$$\cdots$$

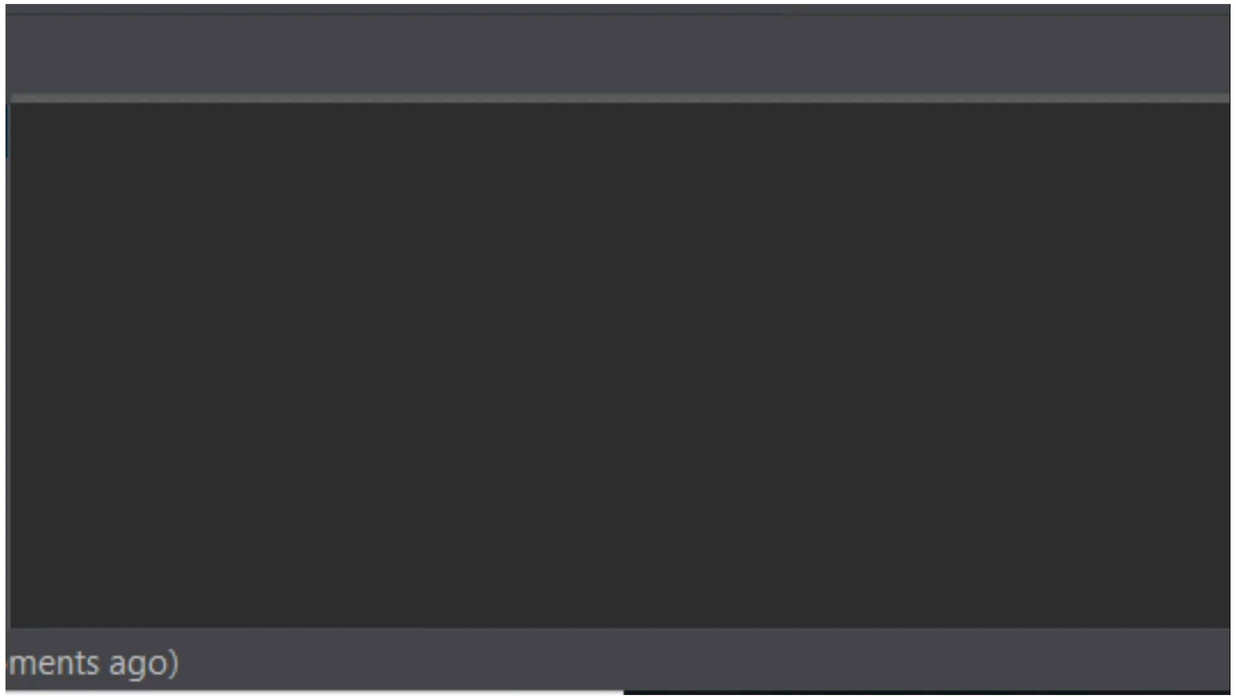
$$\downarrow$$

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

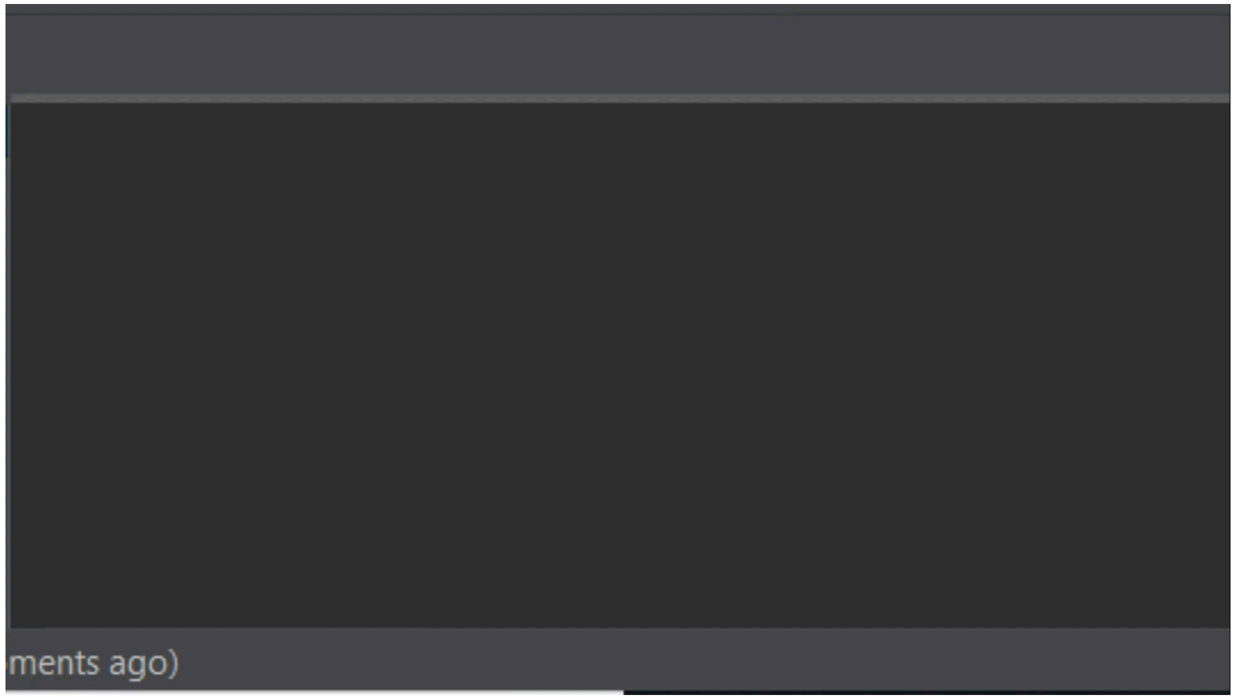
$$holds(dead(speaker), t_2)$$

$$holds(status(phone, DND), t_2)$$





$$\mathbf{B}(TAI, t_1, \mathbf{O}(TAI, t_1, alert, \mathbf{D}(TAI, t_1, alert)))$$
$$\cdots$$
$$\downarrow$$
$$happens(switchon(speaker), t_2)$$
$$holds(dead(speaker), t_2)$$
$$holds(status(phone, DND), t_2)$$

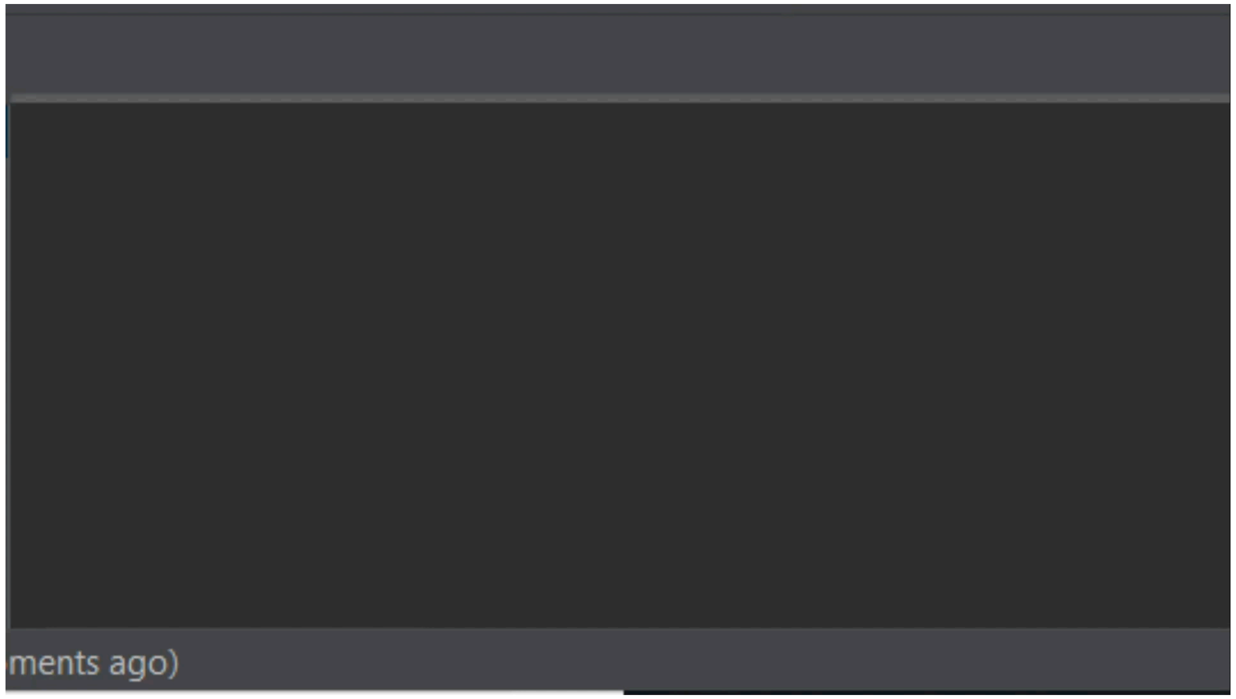


$$\mathbf{B}(TAI, t_1, \mathbf{O}(TAI, t_1, alert, \mathbf{D}(TAI, t_1, alert)))$$

$$\cdots$$

$$\downarrow$$

$$happens(switchon(speaker), t_2)$$
$$holds(dead(speaker), t_2)$$
$$holds(status(phone, DND), t_2)$$



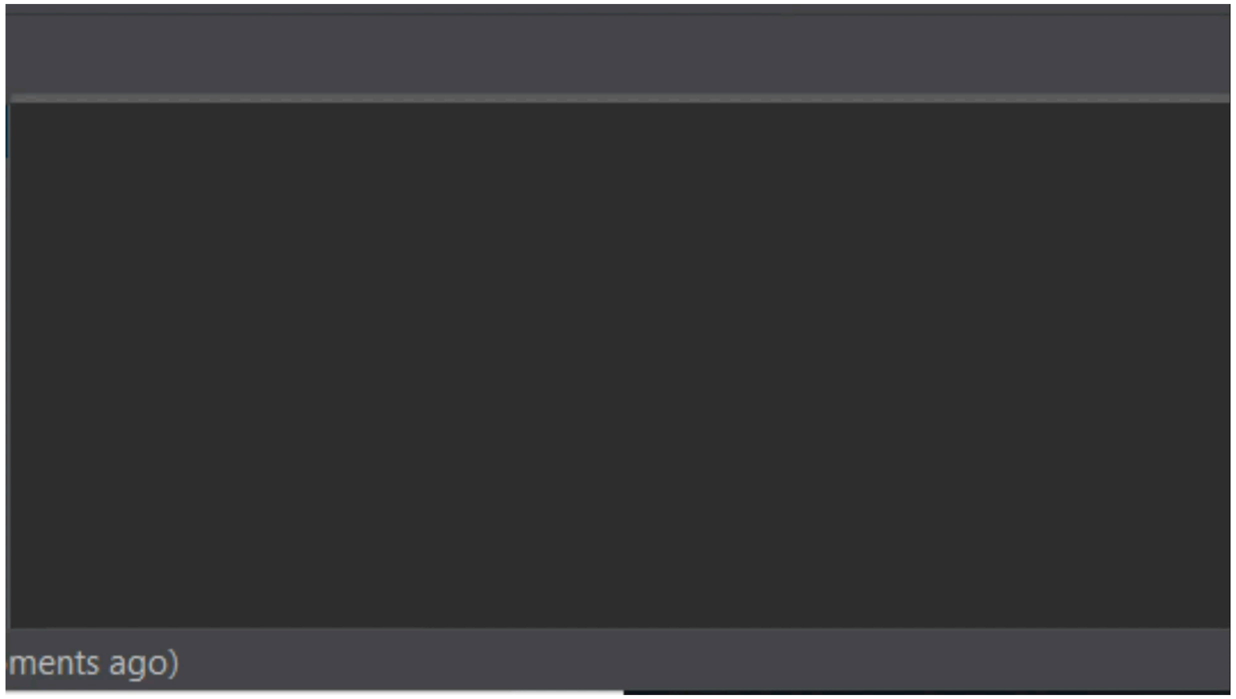
$$\mathbf{B}(TAI, t_1, \mathbf{O}(TAI, t_1, alert, \mathbf{D}(TAI, t_1, alert)))$$

$$\cdots$$

$$\downarrow$$

$$happens(switchon(speaker), t_2)$$
$$holds(dead(speaker), t_2)$$
$$holds(status(phone, DND), t_2)$$

$$\downarrow$$



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

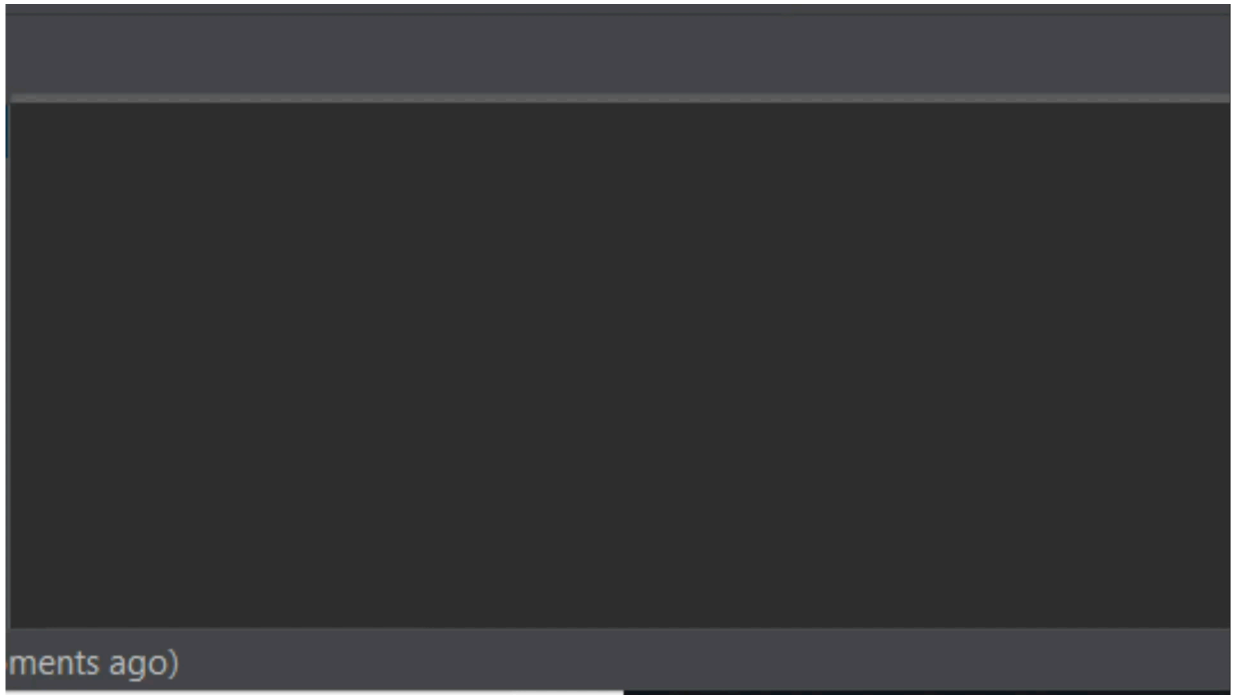
$$\cdots$$

$$\downarrow$$

$$happens(switchon(speaker), t_2)$$
$$holds(dead(speaker), t_2)$$
$$holds(status(phone, DND), t_2)$$

$$\downarrow$$

$$happens(ring(phone), t_3)$$



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

$$\cdots$$

$$\downarrow$$

$$happens(switchon(speaker), t_2)$$
$$holds(dead(speaker), t_2)$$
$$holds(status(phone, DND), t_2)$$

$$\downarrow$$

$$happens(ring(phone), t_3)$$

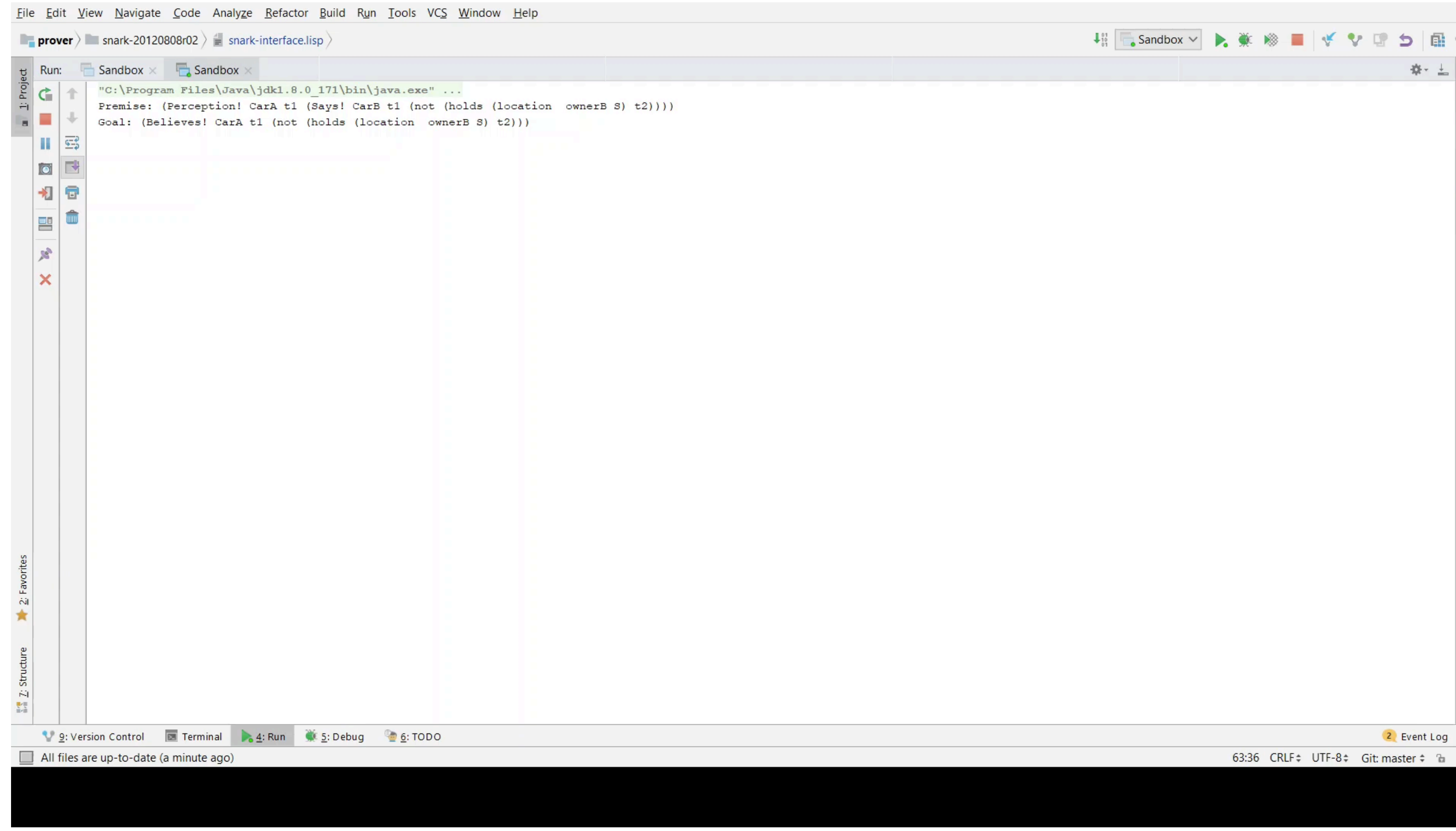
- Novel applications to the *energy regulation* of smart buildings, a new collaboration with Elena Markosa (*University of Southern Denmark)*.

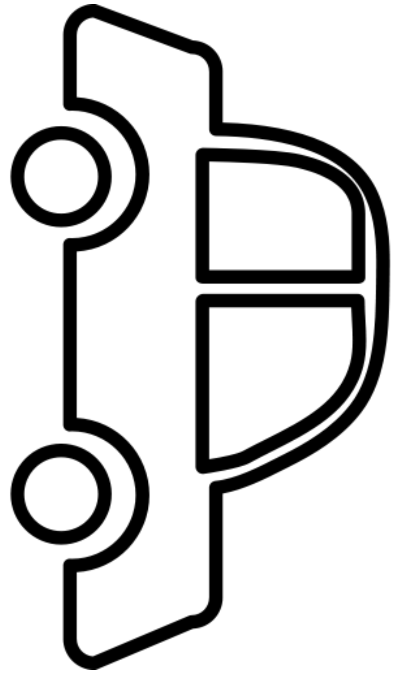
- Potential access to smart building with electricity meters that measure energy consumption from lighting, ventilation units, and heating. Every room instrumented for carbon monoxide, temperature, etc.

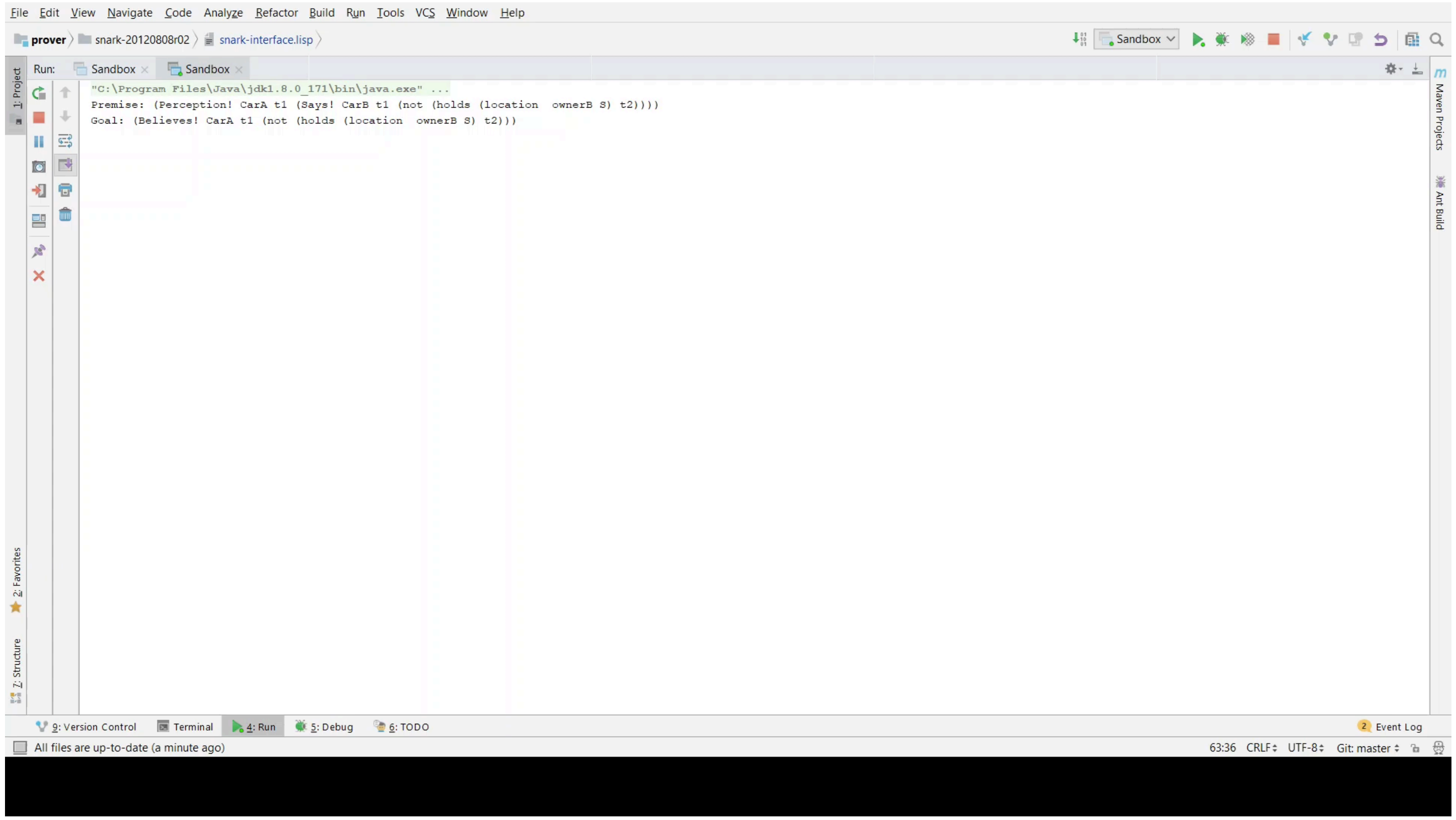
- Potential access to real data from the *Danish Building Research Institute, Siemens and Schneider Electric.*

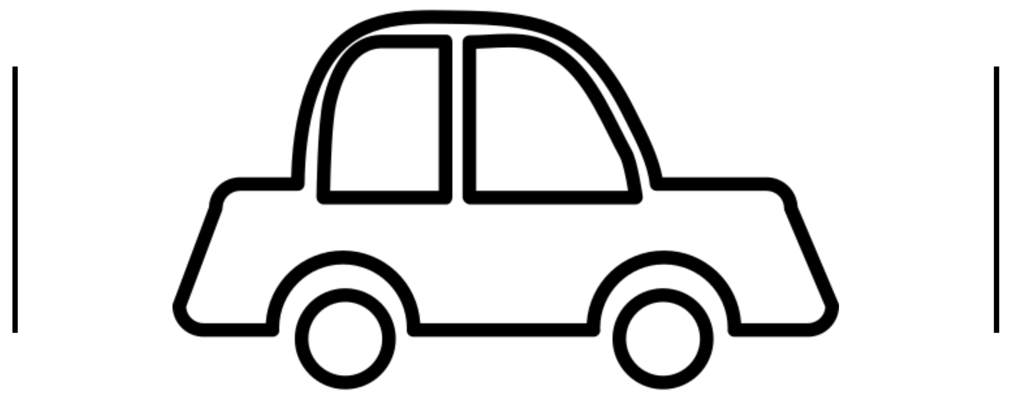
- From smart sensors, TAI agents learn *traits* in occupants' behavior by *anti-unification*, resulting in *beliefs* about their states of mind.

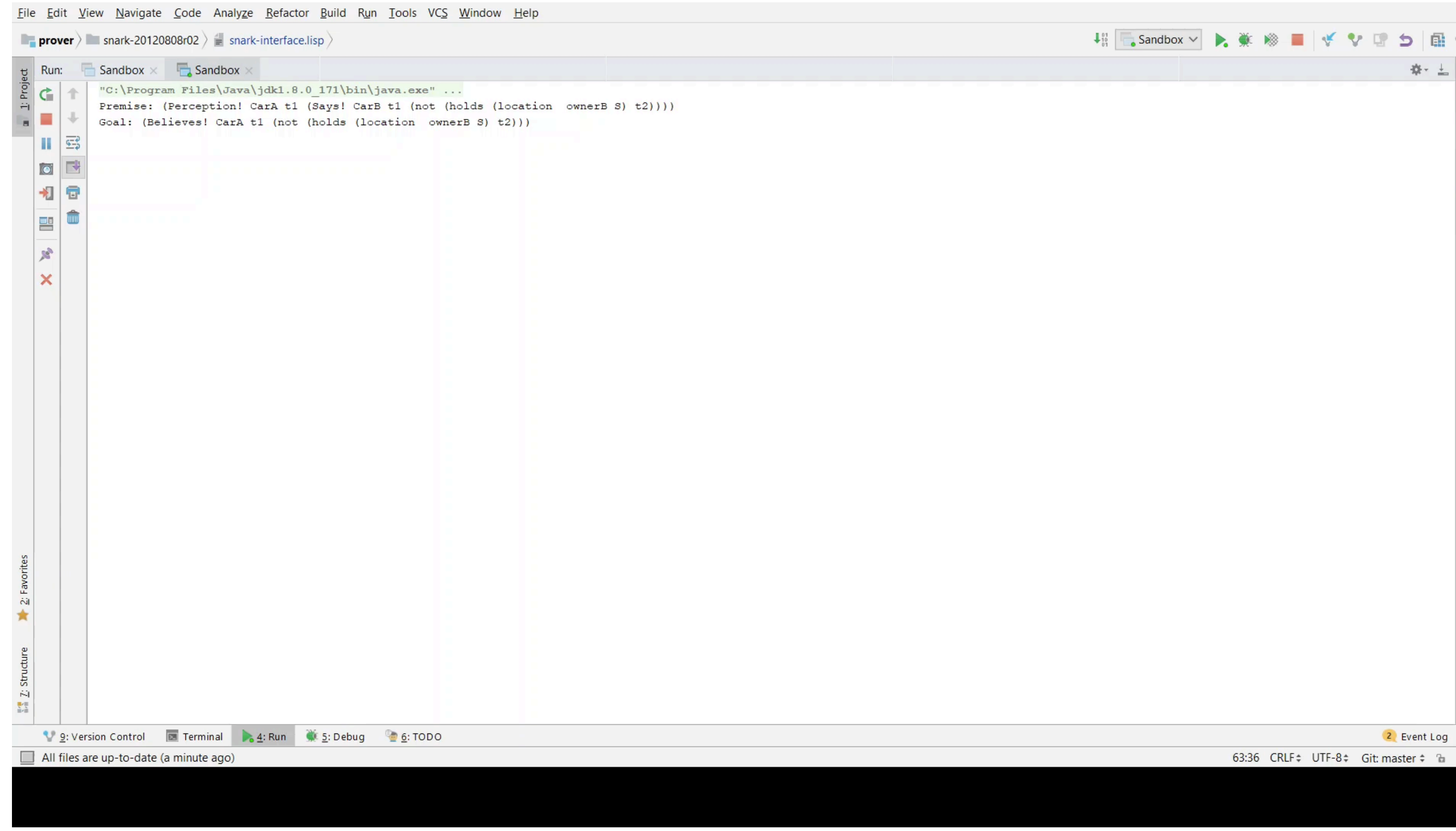
- These beliefs, conjoined with *perceived* actions, are used to regulate room parameters for energy efficiency and convenience.

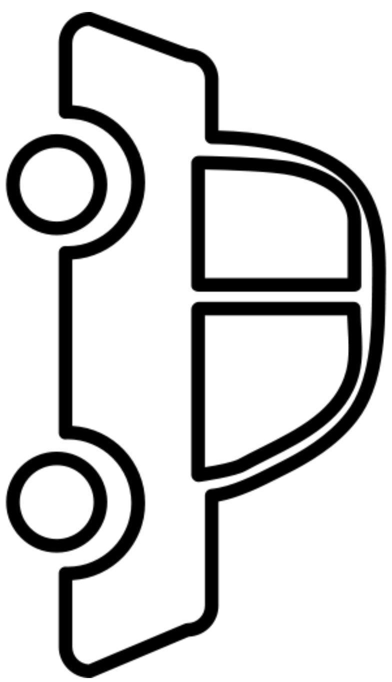

"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
Premise: (Perception! CarA t1 (Says! CarB t1 (not (holds (location ownerB S) t2))))
Goal: (Believes! CarA t1 (not (holds (location ownerB S) t2)))
All files are up-to-date (a minute ago)

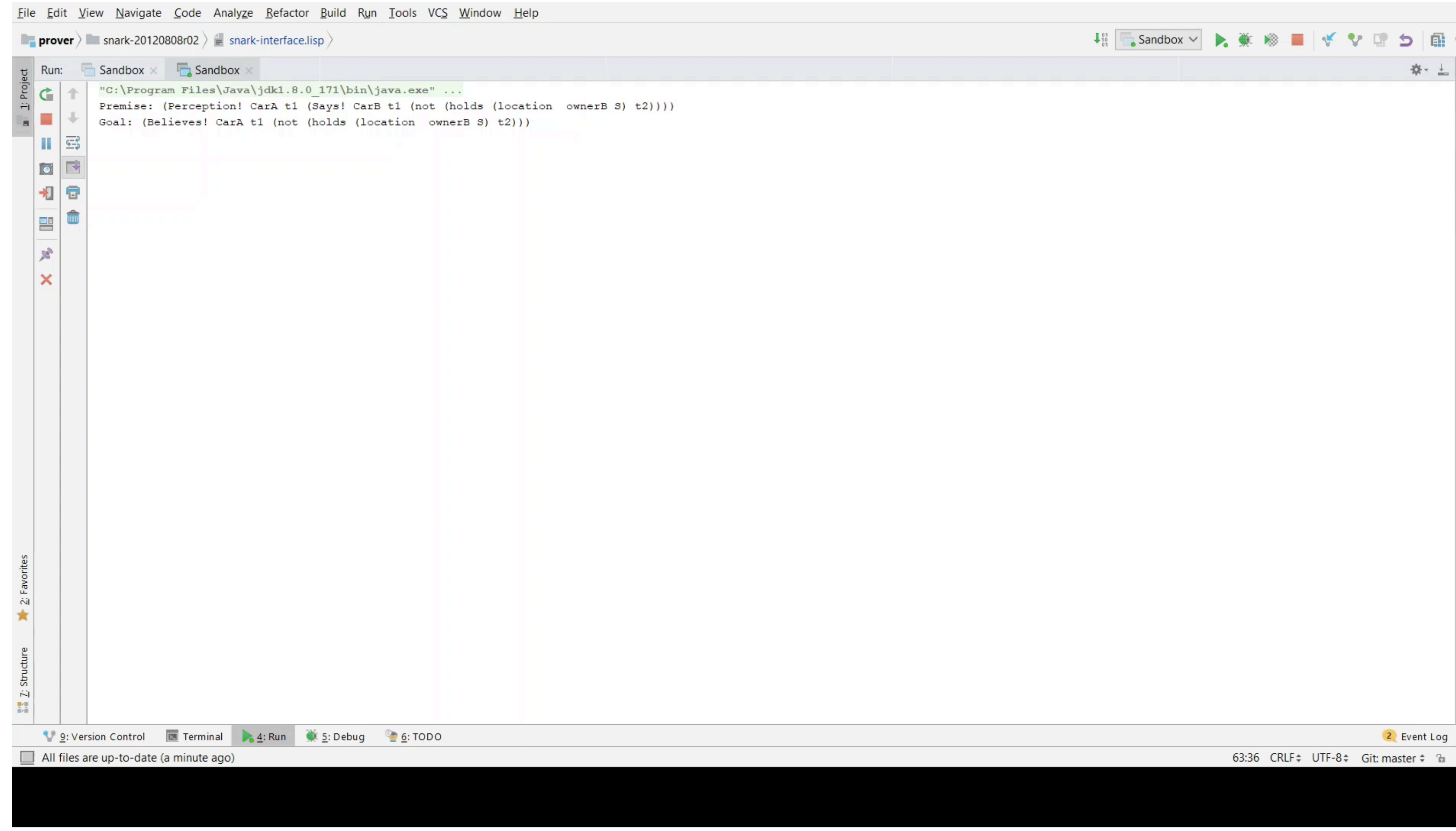


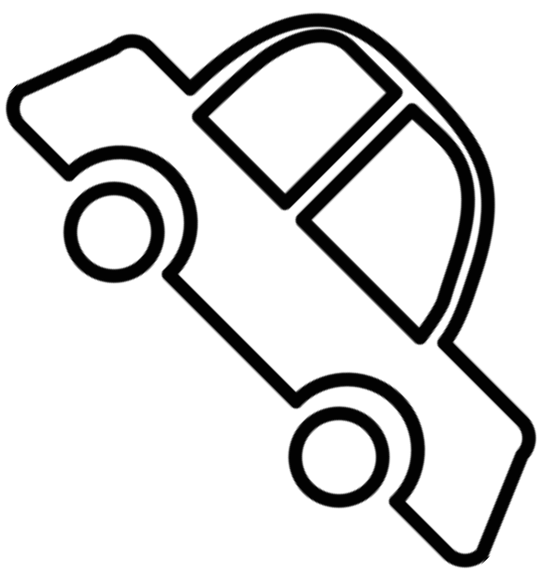

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
prover snark-20120808r02 snark-interface.lisp
Run: Sandbox Sandbox
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
Premise: (Perception! CarA t1 (Says! CarB t1 (not (holds (location ownerB S) t2))))
Goal: (Believes! CarA t1 (not (holds (location ownerB S) t2)))
9: Version Control Terminal 4: Run 5: Debug 6: TODO Event Log
All files are up-to-date (a minute ago)
63:36 CRLF UTF-8 Git: master

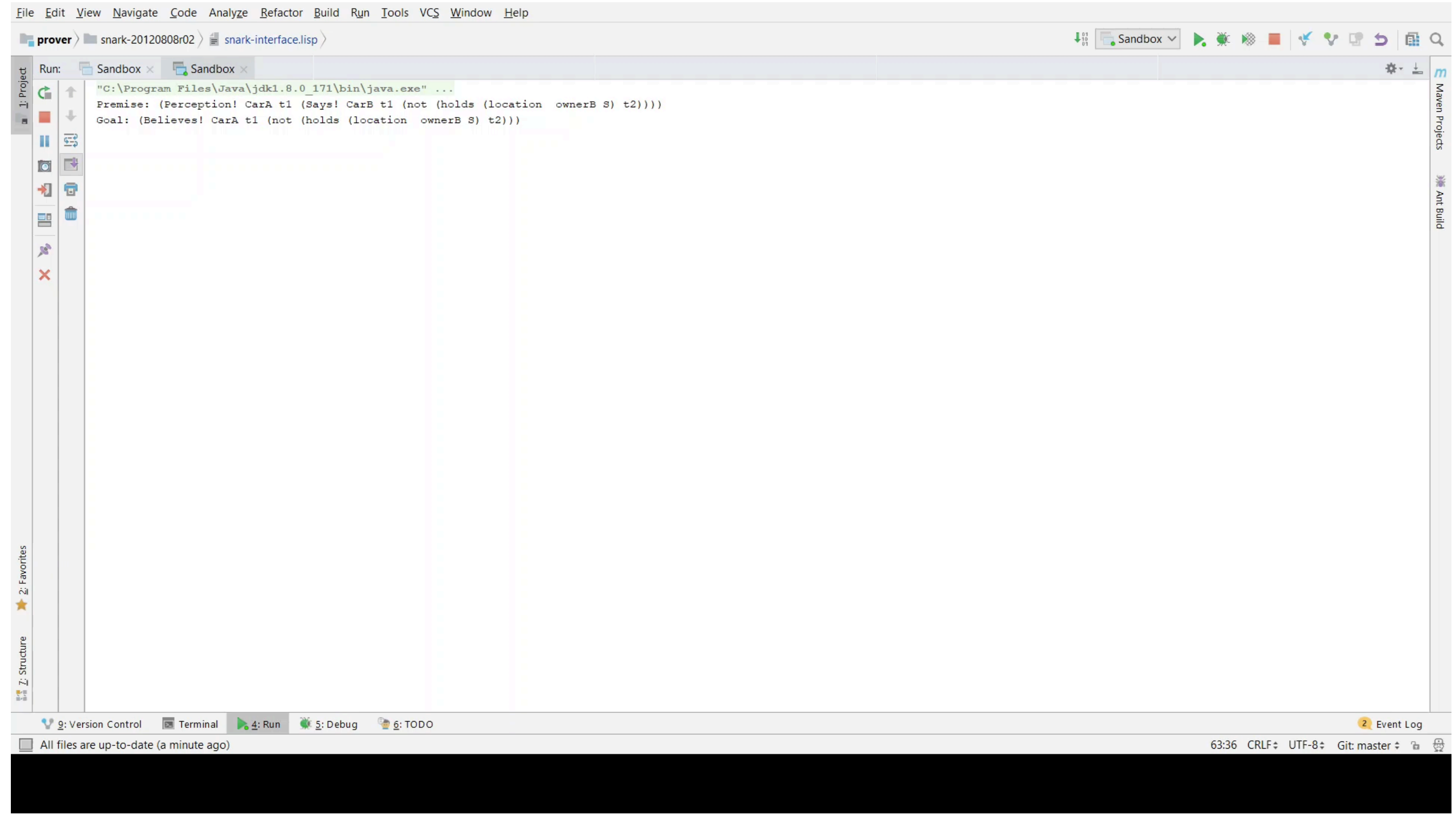


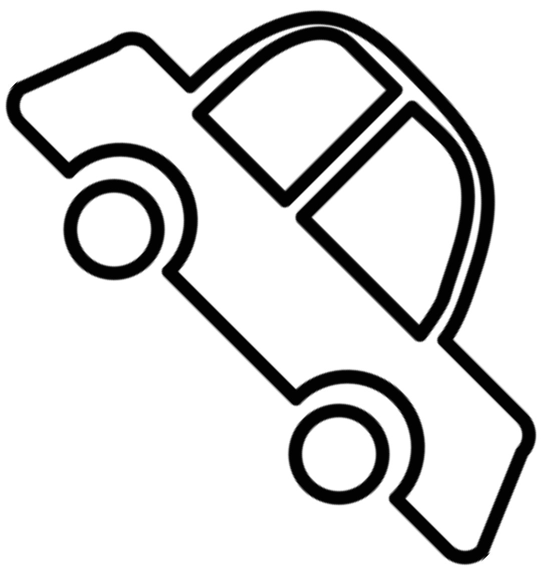

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
prover snark-20120808r02 snark-interface.lisp
Sandbox
Run: Sandbox Sandbox
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
Premise: (Perception! CarA t1 (Says! CarB t1 (not (holds (location ownerB S) t2))))
Goal: (Believes! CarA t1 (not (holds (location ownerB S) t2)))
9: Version Control Terminal 4: Run 5: Debug 6: TODO Event Log
All files are up-to-date (a minute ago)
63:36 CRLF UTF-8 Git: master

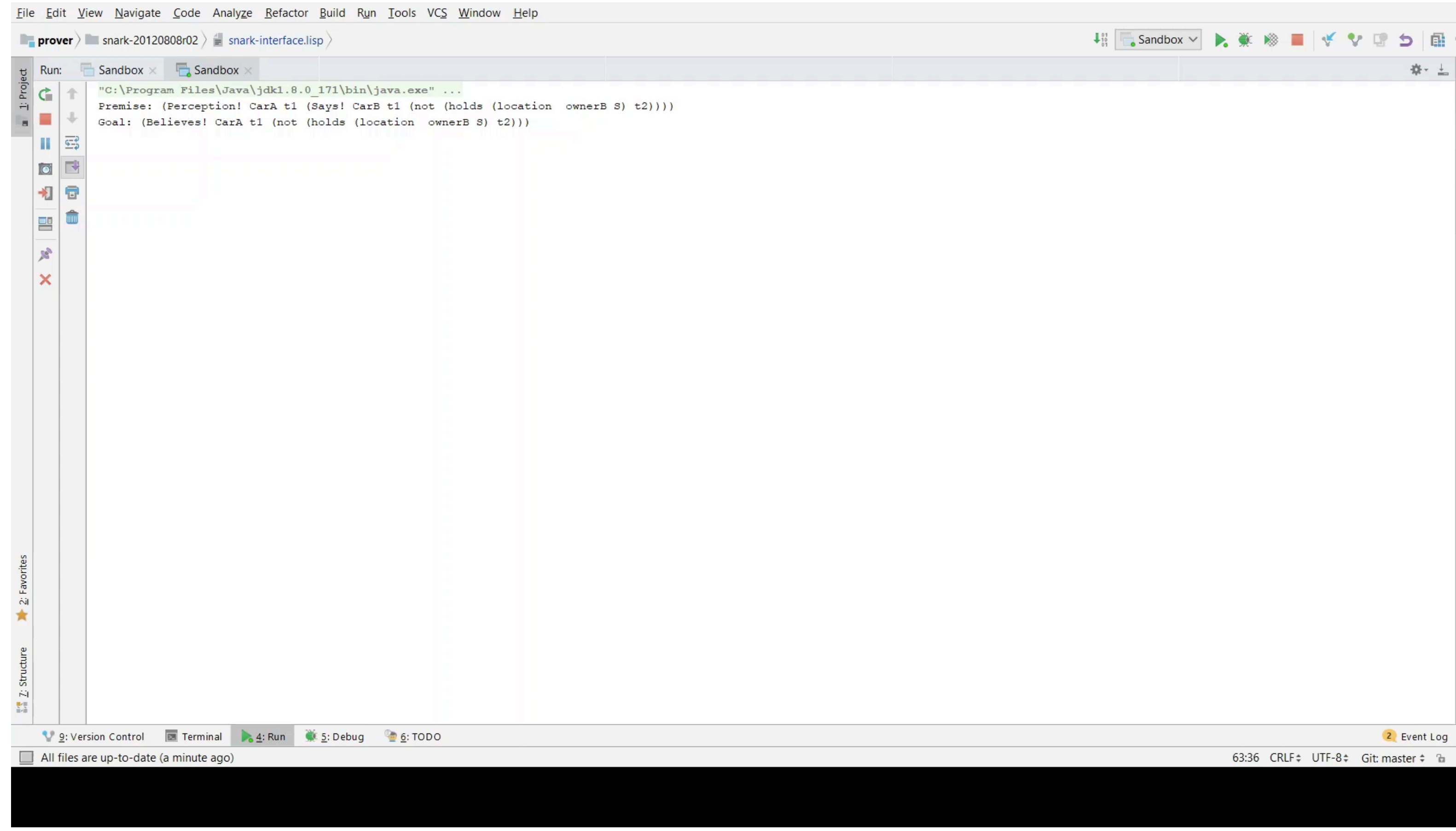



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
prover snark-20120808r02 snark-interface.lisp
Sandbox
Run: Sandbox Sandbox
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
Premise: (Perception! CarA t1 (Says! CarB t1 (not (holds (location ownerB S) t2))))
Goal: (Believes! CarA t1 (not (holds (location ownerB S) t2)))
9: Version Control Terminal 4: Run 5: Debug 6: TODO
Event Log
All files are up-to-date (a minute ago)
63:36 CRLF UTF-8 Git: master

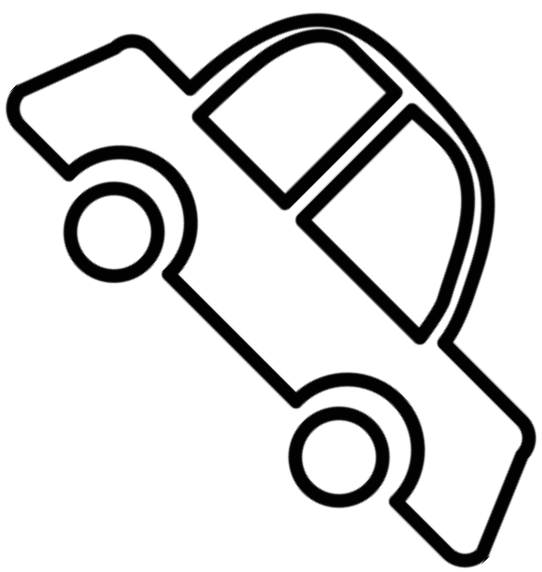



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
prover
snark-20120808r02
snark-interface.lisp
Sandbox
Run:
Sandbox
Sandbox
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
Premise: (Perception! CarA t1 (Says! CarB t1 (not (holds (location ownerB S) t2))))
Goal: (Believes! CarA t1 (not (holds (location ownerB S) t2)))
9: Version Control
Terminal
4: Run
5: Debug
6: TODO
Event Log
All files are up-to-date (a minute ago)
63:36 CRLF UTF-8 Git: master

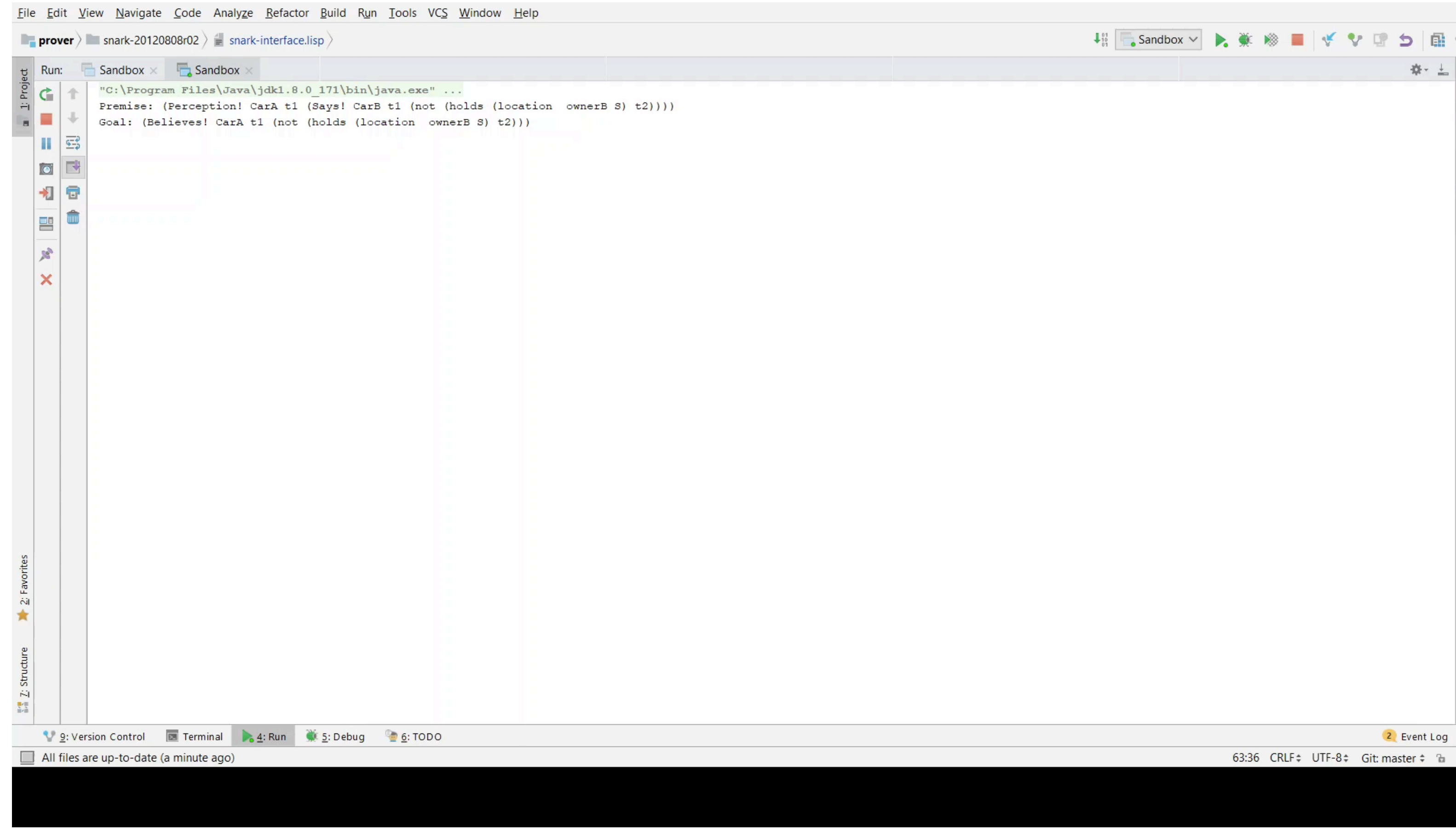




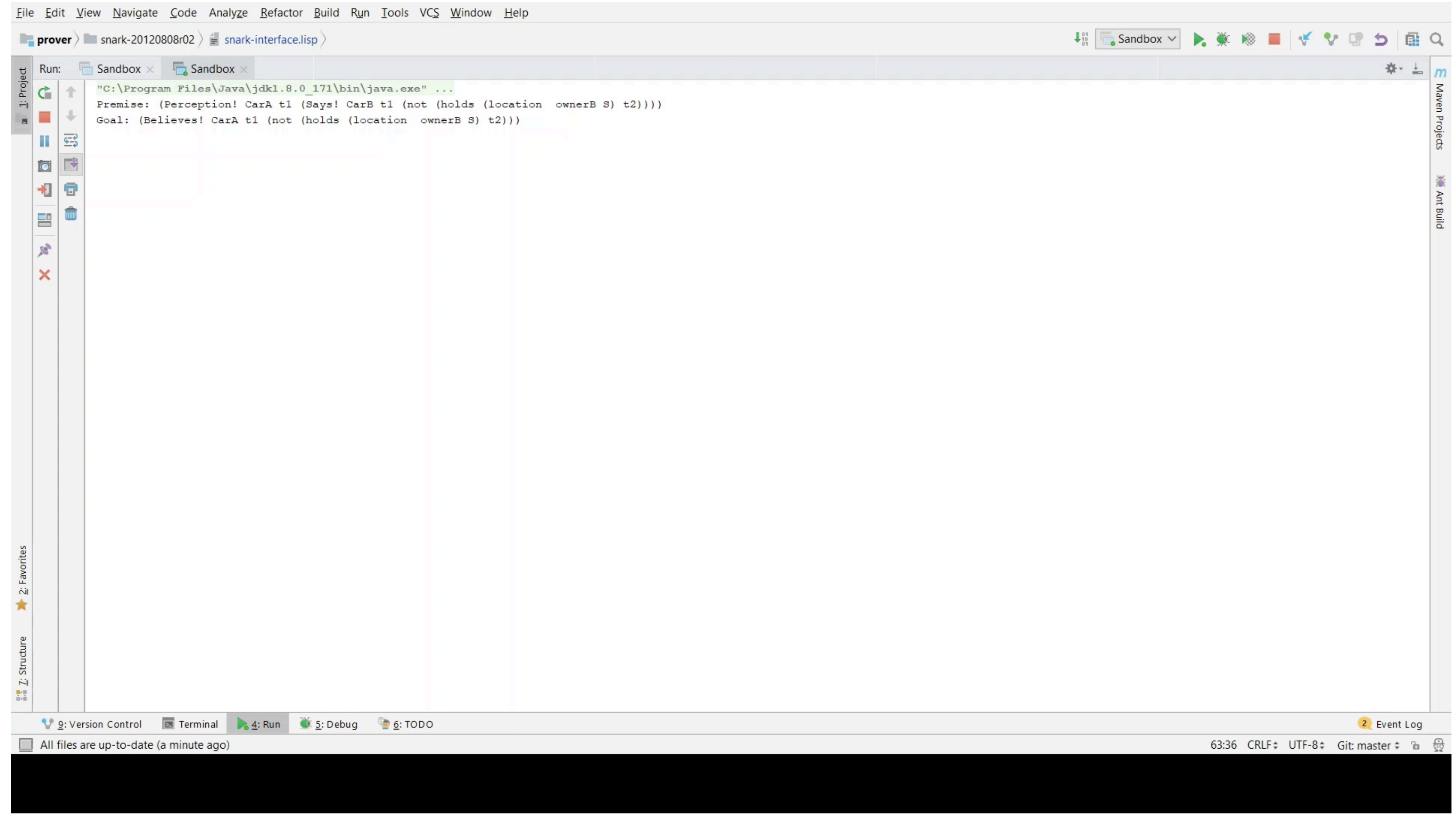
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
Premise: (Perception! CarA t1 (Says! CarB t1 (not (holds (location ownerB S) t2))))
Goal: (Believes! CarA t1 (not (holds (location ownerB S) t2)))

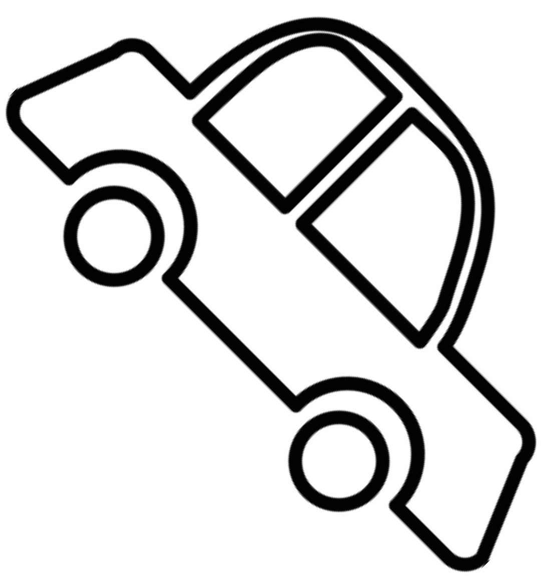


I believe my owner will not return soon

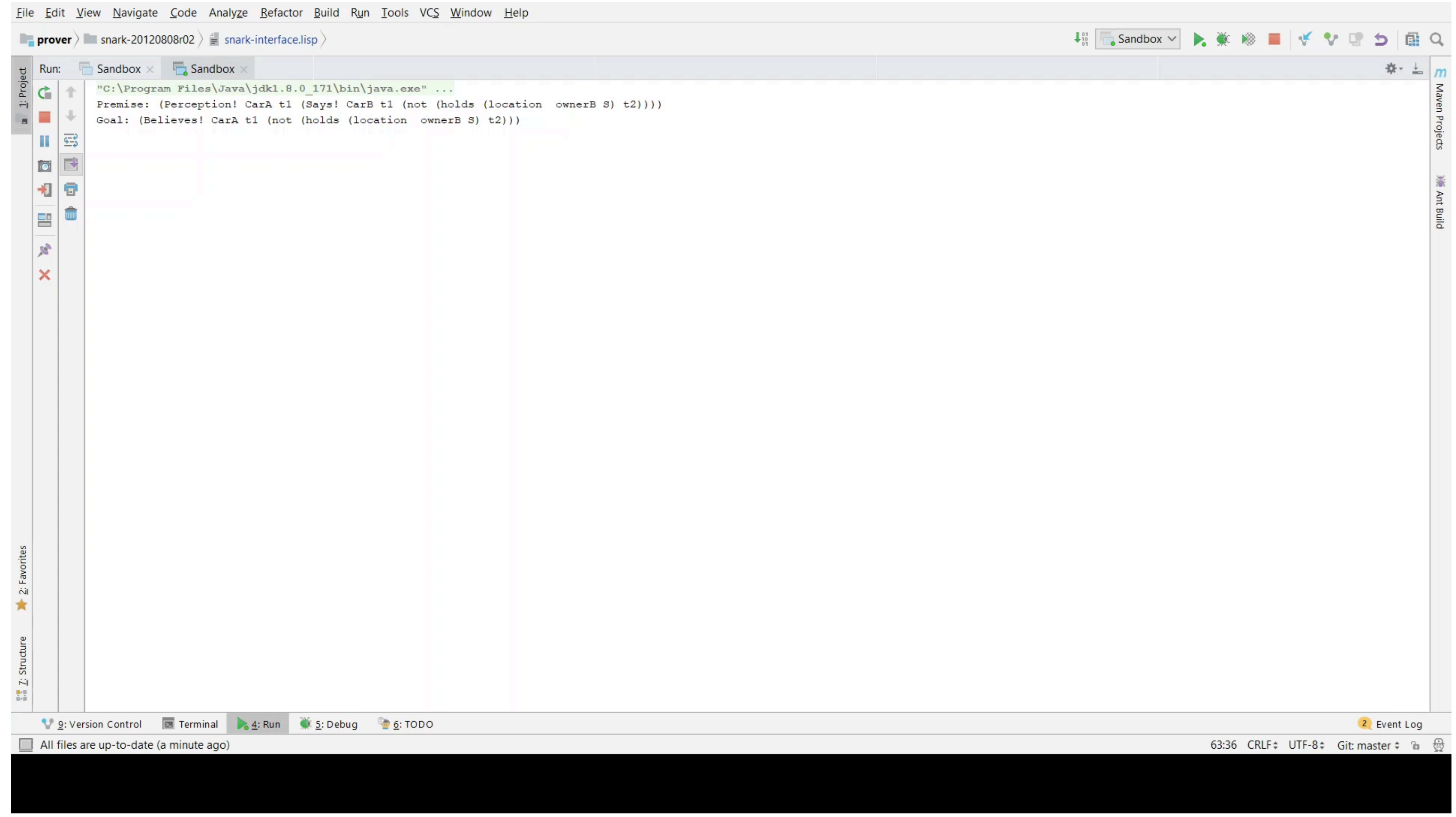

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
prover snark-20120808r02 snark-interface.lisp
Sandbox
Run: Sandbox Sandbox
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
Premise: (Perception! CarA t1 (Says! CarB t1 (not (holds (location ownerB S) t2))))
Goal: (Believes! CarA t1 (not (holds (location ownerB S) t2)))
9: Version Control Terminal 4: Run 5: Debug 6: TODO Event Log
All files are up-to-date (a minute ago)
63:36 CRLF UTF-8 Git: master


I reason that I cannot park here
I believe my owner will not return soon


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
prover snark-20120808r02 snark-interface.lisp
Run: Sandbox Sandbox
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
Premise: (Perception! CarA t1 (Says! CarB t1 (not (holds (location ownerB S) t2))))
Goal: (Believes! CarA t1 (not (holds (location ownerB S) t2)))
All files are up-to-date (a minute ago)

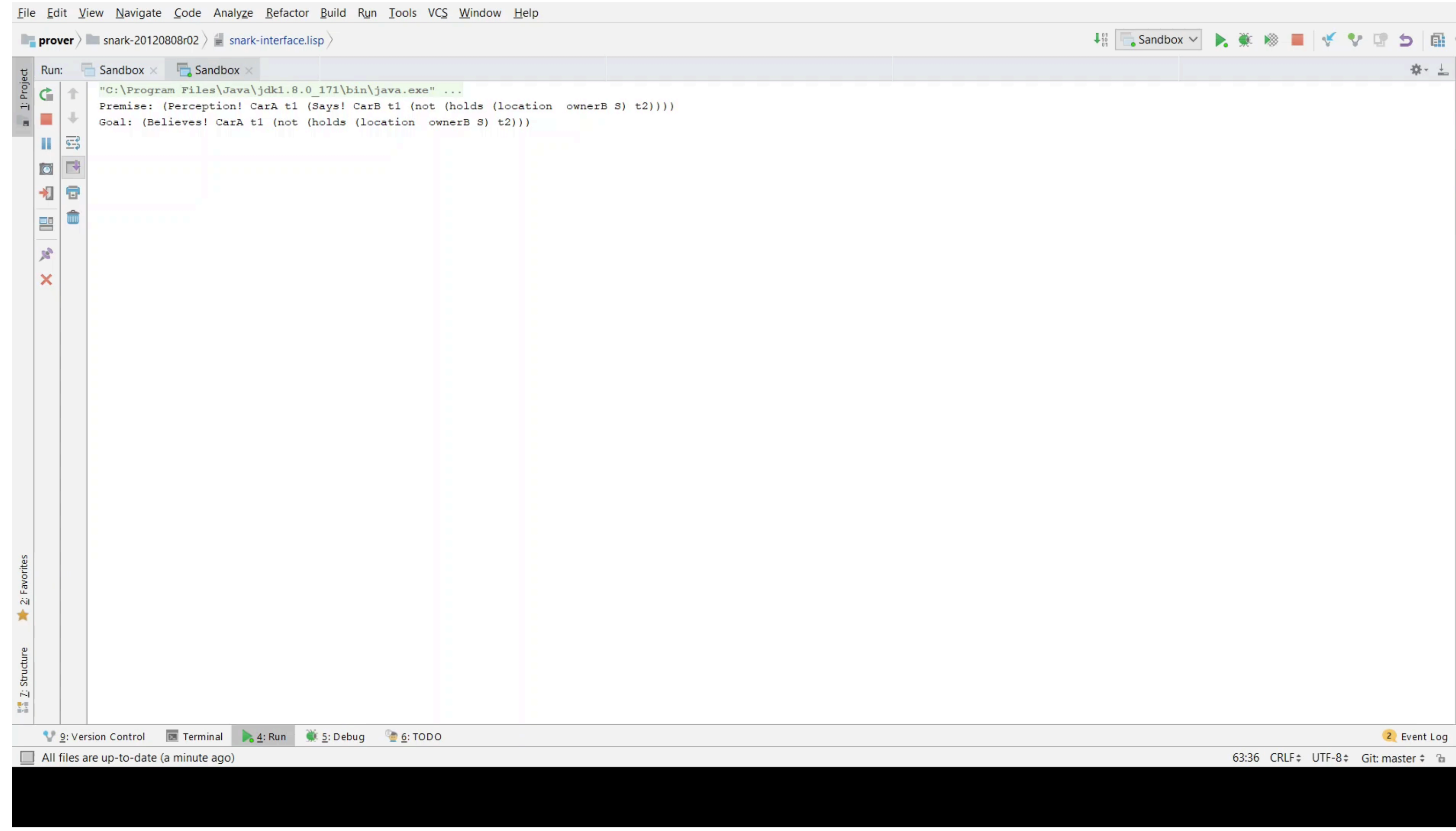



I believe my owner will not return soon



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
prover snark-20120808r02 snark-interface.lisp
Run: Sandbox Sandbox
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
Premise: (Perception! CarA t1 (Says! CarB t1 (not (holds (location ownerB S) t2))))
Goal: (Believes! CarA t1 (not (holds (location ownerB S) t2)))
All files are up-to-date (a minute ago)

I believe my owner will not return soon


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
prover snark-20120808r02 snark-interface.lisp
Run: Sandbox Sandbox
"C:\Program Files\Java\jdk1.8.0_171\bin\java.exe" ...
Premise: (Perception! CarA t1 (Says! CarB t1 (not (holds (location ownerB S) t2))))
Goal: (Believes! CarA t1 (not (holds (location ownerB S) t2)))
9: Version Control Terminal 4: Run 5: Debug 6: TODO Event Log
All files are up-to-date (a minute ago)
63:36 CRLF UTF-8 Git: master


I believe my owner will not return soon