University of Udine
Department of Mathematics, Computer Science and Physics

# MODELLING MULTI-AGENT EPISTEMIC PLANNING IN ASP

36th Italian Conference on Logic Programming

**Alessandro Burigana**, Francesco Fabiano,
Agostino Dovier and Enrico Pontelli

September 14, 2020

# Overview



1. Multi-Agent Epistemic Planning
2. Possibilities
3. The action language $m\mathcal{A}^\rho$
4. PLATO
5. Conclusions

Chapter 1

# Multi-Agent Epistemic Planning

# Introduction



## Epistemic Reasoning

Reasoning not only about agents' *perception of the world* but also about agents' *knowledge* and/or *beliefs* of her and others' beliefs.

# Introduction



## Epistemic Reasoning

Reasoning not only about agents' *perception of the world* but also about agents' *knowledge* and/or *beliefs* of her and others' beliefs.

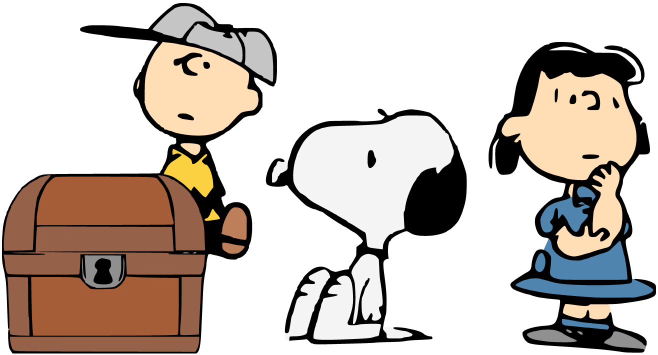
## Multi-agent Epistemic Planning Problem [BA11]

Finding *plans* where the goals can refer to:

- the state of the world
- the knowledge and/or the beliefs of the agents

## An Example



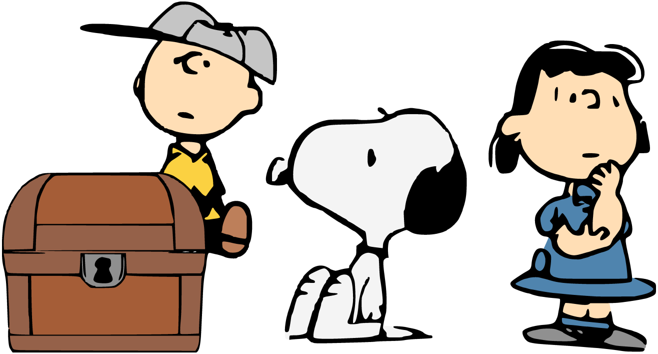


# An Example



## Initial State

- Snoopy and Charlie are looking while Lucy is ¬looking
- No one knows the coin position.

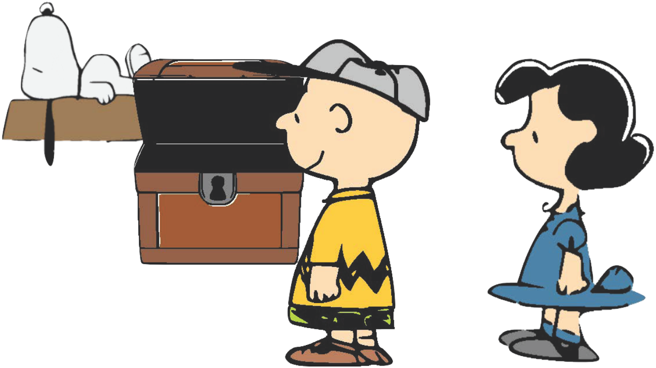


# An Example



## Goal State

- `Charlie` knows the `coin position`
- `Lucy` knows that `Charlie` knows the `coin position`
- `Snoopy` does not know anything about the plan execution



# Notations



Given a set of `agents` $\mathcal{AG}$

**Belief formulae** where ag $\in \mathcal{AG}$, $\alpha \subseteq \mathcal{AG}$

We use the operators $\mathbf{B}_{\mathrm{ag}}$ and $\mathbf{C}_{\alpha}$ to model the beliefs and the common knowledge of the `agents`.

# Notations



Given a set of `agents` $\mathcal{AG}$

## Belief formulae

where ag $\in \mathcal{AG}$, $\alpha \subseteq \mathcal{AG}$

We use the operators $\mathbf{B}_{ag}$ and $\mathbf{C}_{\alpha}$ to model the beliefs and the common knowledge of the `agents`.

## Properties of $\mathbf{B}_{ag}$

**KD45** and **S5** Axioms

Given the fluent formulae $\phi$, $\psi$ and the worlds $\mathtt{i}$, $\mathtt{j}$

| | | |
|---|---|---|
| D | $\neg\mathcal{R}_{\mathtt{i}}\bot$ | $\mathcal{B}$ $\mathcal{K}$ |
| K | $(\mathcal{R}_{\mathtt{i}}\varphi \wedge \mathcal{R}_{\mathtt{i}}(\varphi \implies \psi)) \implies \mathcal{R}_{\mathtt{i}}\psi$ | $\mathcal{B}$ $\mathcal{K}$ |
| T | $\mathcal{R}_{\mathtt{i}}\varphi \implies \varphi$ | $\mathcal{K}$ |
| 4 | $\mathcal{R}_{\mathtt{i}}\varphi \implies \mathcal{R}_{\mathtt{i}}\mathcal{R}_{\mathtt{i}}\varphi$ | $\mathcal{B}$ $\mathcal{K}$ |
| 5 | $\neg\mathcal{R}_{\mathtt{i}}\varphi \implies \mathcal{R}_{\mathtt{i}}\neg\mathcal{R}_{\mathtt{i}}\varphi$ | $\mathcal{B}$ $\mathcal{K}$ |

Chapter 2

# Possibilities

# Overview



- Introduced by Gerbrandy and Groeneveld [GG97]
- Used to represent *multi-agent information change*
- Based on *non-well-founded sets*

# Formal Definition



## Possibility [GG97]

Let $\mathcal{AG}$ be a set of agents and $\mathcal{F}$ a set of propositional variables:

- A *possibility* u is a function that assigns to each propositional variable $\texttt{l} \in \mathcal{F}$ a truth value $\mathrm{u}(\texttt{l}) \in \{0, 1\}$ and to each agent $\texttt{ag} \in \mathcal{AG}$ a set of possibilities $\mathrm{u}(\texttt{ag}) = \sigma$ (*information state*).

Intuitively ...

- The possibility u is a possible interpretation of the world and of the agents' beliefs
- $\mathrm{u}(\texttt{l})$ specifies the truth value of the literal $\texttt{l}$
- $\mathrm{u}(\texttt{ag})$ is the set of all the interpretations the agent $\texttt{ag}$ considers possible in u
- Representable with graphs: we will use graph terminology

Chapter 3

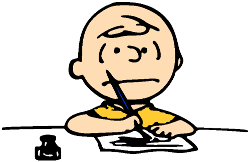
# The action language $m\mathcal{A}^{\rho}$

# Action types



We introduced the action language $m\mathcal{A}^\rho$ in [Fab+20]

- Used to describe MEP problems
- Uses possibilities as states
- Actions preconditions: belief formulae



# Action types



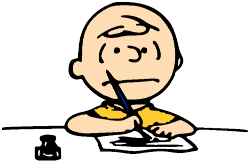
We introduced the action language $m\mathcal{A}^\rho$ in [Fab+20]

- Used to describe MEP problems
- Uses possibilities as states
- Actions preconditions: belief formulae

**Three** types of actions:

- *Ontic*: modifies some fluents of the world

  `Charlie` *`opens`* `the box`



# Action types



We introduced the action language $m\mathcal{A}^\rho$ in [Fab+20]

- Used to describe MEP problems
- Uses possibilities as states
- Actions preconditions: belief formulae

**Three** types of actions:

- Ontic: modifies some fluents of the world

  Charlie *opens* the box

- *Sensing*: senses the true value of a fluent

  `Charlie peeks inside the box`

# Action types



We introduced the action language $m\mathcal{A}^\rho$ in [Fab+20]

- Used to describe MEP problems
- Uses possibilities as states
- Actions preconditions: belief formulae

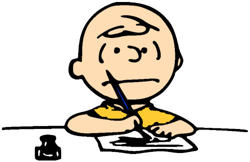
**Three** types of actions:

- Ontic: modifies some fluents of the world

  Charlie *opens* the box
- Sensing: senses the true value of a fluent

  Charlie *peeks* inside the box
- *Announcement*: announces the fluent to other agents

  Charlie *announces* the coin position



# Observability Relations



An *execution* of an action might change or not an agents' belief accordingly to her degree of awareness

| Action type | Full observers | Partial Observers | Oblivious |
|---|---|---|---|
| Ontic | ✓ | | ✓ |
| Sensing | ✓ | ✓ | ✓ |
| Announcement | ✓ | ✓ | ✓ |

Chapter 4

# PLATO

## Overview



PLATO, e**P**istemic mu**L**ti-agent **A**nswer se**T** programming s**O**lver:

- Declarative encoding in `ASP` of MEP
- Based on the language $m\mathcal{A}^\rho$
- Main components: *initial state generation*, *entailment*, *transition function*
- Exploits *clingo*'s multi-shot capabilities [Geb+19]
- Formal proof of correctness

# Encoding possibilities



Let u be a possibility.

### ASP encoding: possibilities

We encode u with the atom possible_world($T_u$, $R_u$, $P_u$), where:

- $T_u$ tells us when u was created
- $R_u$ is the *repetition* of u
- $P_u$ is the numerical index of u

# Encoding possibilities



Let u be a possibility.

### ASP encoding: possibilities

We encode u with the atom `possible_world`$(T_u, R_u, P_u)$, where:

- $T_u$ tells us when u was created
- $R_u$ is the *repetition* of u
- $P_u$ is the numerical index of u

### ASP encoding: pointed possibility

If u is the possibility that represents the *real configuration* of the world, we encode it with the atom `pointed`$(T_u, R_u, P_u)$.

When the context is clear we will use *only* $P_u$ instead of $(T_u, R_u, P_u)$.

# Encoding possibilities



Let u, v be two possibilities, let AG be an agent and let F be a fluent.

**ASP encoding: information states**

We encode $v \in u(\mathtt{AG})$ with the atom $\mathtt{believes}(\mathtt{P_u}, \mathtt{P_v}, \mathtt{AG})$.

# Encoding possibilities



Let u, v be two possibilities, let $\mathtt{AG}$ be an agent and let $\mathtt{F}$ be a fluent.

### ASP encoding: information states

We encode $v \in u(\mathtt{AG})$ with the atom $\mathtt{believes}(\mathtt{P_u}, \mathtt{P_v}, \mathtt{AG})$.

### ASP encoding: interpretations

We encode $u(\mathtt{F}) = 1$ with the atom $\mathtt{holds}(\mathtt{P_u}, \mathtt{F})$.

# Entailment



Given a possibility `P` and a belief formula `F`.

# Entailment



Given a possibility P and a belief formula F.

```
entails (P,          F)  :- holds(P,F), fluent(F).
entails (P,     neg(F))  :- not entails(P,F).
entails (P, and(F1,F2))  :- entails(P,F1), entails(P,F2).
entails (P,  or(F1,F2))  :- entails(P,F1).
entails (P,  or(F1,F2))  :- entails(P,F2).
```

# Entailment



Given a possibility P and a belief formula F.

```
entails     (P,             F)   :- holds(P,F), fluent(F).
entails     (P,        neg(F))   :- not entails(P,F).
entails     (P,   and(F1,F2))    :- entails(P,F1), entails(P,F2).
entails     (P,    or(F1,F2))    :- entails(P,F1).
entails     (P,    or(F1,F2))    :- entails(P,F2).
not_entails (P1,     b(AG,F))    :- not entails(P2,F), believes(P1,P2,AG).
entails     (P,      b(AG,F))    :- not not_entails(P,b(AG,F)).
```

# Entailment



Given a possibility P and a belief formula F.

```
entails      (P,             F)   :- holds(P,F), fluent(F).

entails      (P,        neg(F))   :- not entails(P,F).
entails      (P,   and(F1,F2))   :- entails(P,F1), entails(P,F2).
entails      (P,    or(F1,F2))   :- entails(P,F1).
entails      (P,    or(F1,F2))   :- entails(P,F2).

not_entails  (P1,      b(AG,F))   :- not entails(P2,F), believes(P1,P2,AG).
entails      (P,       b(AG,F))   :- not not_entails(P,b(AG,F)).

not_entails  (P1,     c(AGS,F))   :- not entails(P2,F), reaches(P1,P2,AGS).
entails      (P,      c(AGS,F))   :- not not_entails(P,c(AGS,F)).
```

# Ontic actions



Let *open* be an ontic action such that

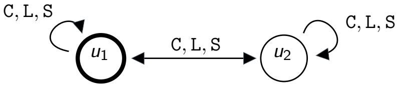
- It sets the fluent `opened` to true
- Only `Charlie` and `Snoopy` are attentive

# Ontic actions



Let *open* be an ontic action such that

- It sets the fluent `opened` to true
- Only `Charlie` and `Snoopy` are attentive




$u_1(\mathcal{F}) = \{\texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u_2(\mathcal{F}) = \{\texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

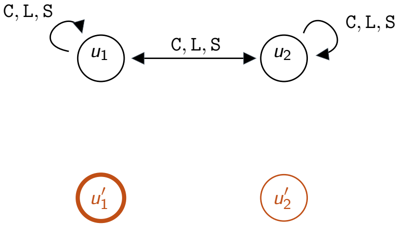
# Ontic actions



Let *open* be an ontic action such that

- It sets the fluent `opened` to true
- Only `Charlie` and `Snoopy` are attentive

$$\mathtt{pw}(\mathtt{u}'_{\mathtt{i}})\ \mathtt{:-}\ \mathtt{pointed}(\mathtt{u}_1), \mathtt{reaches}(\mathtt{u}_1, \mathtt{u}_{\mathtt{i}}, \mathtt{AGS}), \mathtt{fully\_obs}(\mathtt{AGS}). \quad (i \in \{1, 2\})$$




$u_1(\mathcal{F}) = \{\mathtt{head}, \mathtt{looking}_{\mathtt{Charlie}}, \mathtt{looking}_{\mathtt{Snoopy}}\}$

$u_2(\mathcal{F}) = \{\mathtt{looking}_{\mathtt{Charlie}}, \mathtt{looking}_{\mathtt{Snoopy}}\}$

$u'_1(\mathcal{F}) = \{\mathtt{opened}, \mathtt{head}, \mathtt{looking}_{\mathtt{Charlie}}, \mathtt{looking}_{\mathtt{Snoopy}}\}$

$u'_2(\mathcal{F}) = \{\mathtt{opened}, \mathtt{looking}_{\mathtt{Charlie}}, \mathtt{looking}_{\mathtt{Snoopy}}\}$

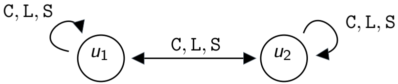
# Ontic actions



Let *open* be an ontic action such that

- It sets the fluent opened to true
- Only `Charlie` and `Snoopy` are attentive

$$\texttt{believes}(\texttt{u}'_\texttt{i}, \texttt{u}'_\texttt{j}, \texttt{AG}) \texttt{ :- } \texttt{pw}(\texttt{u}'_\texttt{i}), \texttt{pw}(\texttt{u}'_\texttt{j}), \texttt{fully\_obs}(\texttt{AG}), \texttt{believes}(\texttt{u}_\texttt{i}, \texttt{u}_\texttt{j}, \texttt{AG}), \texttt{pw}(\texttt{u}_\texttt{i}), \texttt{pw}(\texttt{u}_\texttt{j}). \qquad (i, j \in \{1, 2\})$$




$u_1(\mathcal{F}) = \{\texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u_2(\mathcal{F}) = \{\texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u'_1(\mathcal{F}) = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u'_2(\mathcal{F}) = \{\texttt{opened}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

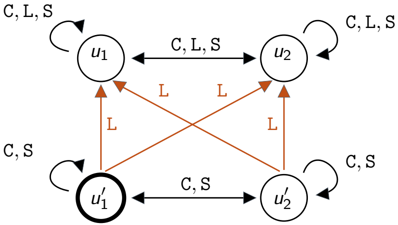
# Ontic actions



Let *open* be an ontic action such that

- It sets the fluent `opened` to true
- Only `Charlie` and `Snoopy` are attentive

$$\texttt{believes}(\texttt{u}'_\texttt{i}, \texttt{u}_\texttt{j}, \texttt{AG}) \texttt{ :- } \texttt{pw}(\texttt{u}'_\texttt{i}), \texttt{oblivious}(\texttt{AG}), \texttt{believes}(\texttt{u}_\texttt{i}, \texttt{u}_\texttt{j}, \texttt{AG}), \texttt{pw}(\texttt{u}_\texttt{i}), \texttt{pw}(\texttt{u}_\texttt{j}). \qquad (i, j \in \{1, 2\})$$




$u_1(\mathcal{F}) = \{\texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u_2(\mathcal{F}) = \{\texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u'_1(\mathcal{F}) = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u'_2(\mathcal{F}) = \{\texttt{opened}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

# Sensing/Announcement actions



Let *peek* be an ontic action such that

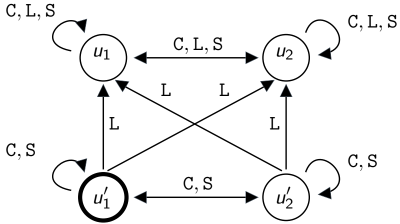
- `Charlie` senses the fluent `heads`
- Only `Charlie` and `Snoopy` are attentive

# Sensing/Announcement actions



Let *peek* be an ontic action such that

- `Charlie` senses the fluent `heads`
- Only `Charlie` and `Snoopy` are attentive




$u_1(\mathcal{F}) = \{\texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u_2(\mathcal{F}) = \{\texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u'_1(\mathcal{F}) = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u'_2(\mathcal{F}) = \{\texttt{opened}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

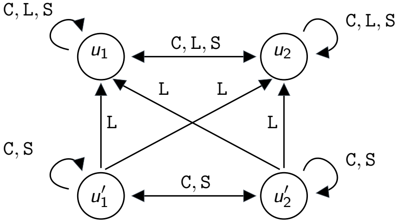
# Sensing/Announcement actions



Let *peek* be an ontic action such that

- `Charlie` senses the fluent `heads`
- Only `Charlie` and `Snoopy` are attentive

$$\mathtt{pw}(\mathtt{u}_1'') \ \mathtt{:-}\ \ \mathtt{pointed}(\mathtt{u}_1'), \mathtt{reaches}(\mathtt{u}_1', \mathtt{u}_1', \mathtt{AGS}), \mathtt{fully\_obs}(\mathtt{AGS}),$$
$$\mathtt{holds\_sensed}(\mathtt{u}_1, \mathtt{peek}).$$




$u_1''(\mathcal{F}) = \{\mathtt{opened}, \mathtt{head}, \mathtt{looking}_{\mathtt{Charlie}}, \mathtt{looking}_{\mathtt{Snoopy}}\}$

$u_1''$

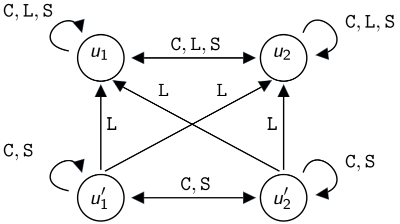
# Sensing/Announcement actions



Let *peek* be an ontic action such that

- `Charlie` senses the fluent **heads**
- Only `Charlie` and `Snoopy` are attentive

$$\mathtt{pw}(\mathtt{u}_2'') \; \texttt{:-} \; \mathtt{pointed}(\mathtt{u}_1'), \mathtt{reaches}(\mathtt{u}_1', \mathtt{u}_2', \mathtt{AGS}), \mathtt{not\_oblivious}(\mathtt{AGS}).$$




$u_1''(\mathcal{F}) = \{\mathtt{opened}, \mathtt{head}, \mathtt{looking}_{\mathtt{Charlie}}, \mathtt{looking}_{\mathtt{Snoopy}}\}$

$u_2''(\mathcal{F}) = \{\mathtt{opened}, \mathtt{looking}_{\mathtt{Charlie}}, \mathtt{looking}_{\mathtt{Snoopy}}\}$

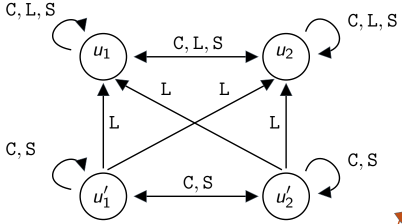
# Sensing/Announcement actions



Let *peek* be an ontic action such that

- `Charlie` senses the fluent `heads`
- Only `Charlie` and `Snoopy` are attentive

$$\texttt{believes}(\texttt{u}''_\texttt{i}, \texttt{u}''_\texttt{j}, \texttt{AG}) \texttt{ :- } \texttt{pw}(\texttt{u}''_\texttt{i}), \texttt{pw}(\texttt{u}''_\texttt{j}), \texttt{pw}(\texttt{u}'_\texttt{i}), \texttt{pw}(\texttt{u}'_\texttt{j}), \quad (i, j \in \{1, 2\})$$
$$\texttt{believes}(\texttt{u}'_\texttt{i}, \texttt{u}'_\texttt{j}, \texttt{AG}), \texttt{fully\_obs}(\texttt{AG}),$$
$$\texttt{holds\_sensed}(\texttt{u}_\texttt{i}, \texttt{peek}) = \texttt{holds\_sensed}(\texttt{u}_\texttt{j}, \texttt{peek}).$$




$u''_1(\mathcal{F}) = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u''_2(\mathcal{F}) = \{\texttt{opened}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

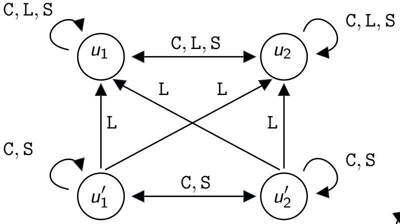
# Sensing/Announcement actions



Let *peek* be an ontic action such that

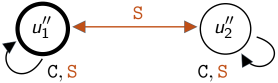
- `Charlie` senses the fluent `heads`
- Only `Charlie` and `Snoopy` are attentive

$$\texttt{believes}(\texttt{u}''_\texttt{i}, \texttt{u}''_\texttt{j}, \texttt{AG}) \;\texttt{:-}\; \texttt{pw}(\texttt{u}''_\texttt{i}), \texttt{pw}(\texttt{u}''_\texttt{j}), \texttt{pw}(\texttt{u}'_\texttt{i}), \texttt{pw}(\texttt{u}'_\texttt{j}), \texttt{believes}(\texttt{u}'_\texttt{i}, \texttt{u}'_\texttt{j}, \texttt{AG}), \texttt{partially\_obs}(\texttt{AG}). \qquad (i, j \in \{1, 2\})$$




$u''_1(\mathcal{F}) = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u''_2(\mathcal{F}) = \{\texttt{opened}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$



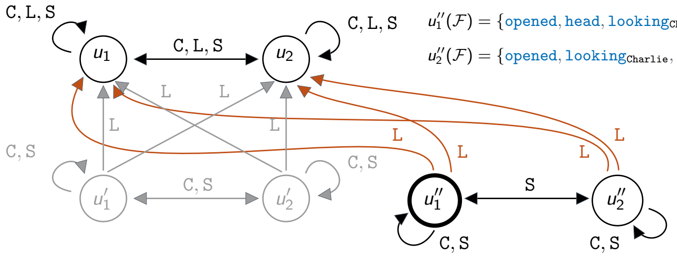
# Sensing/Announcement actions



Let *peek* be an ontic action such that

- Charlie senses the fluent heads
- Only Charlie and Snoopy are attentive

$$\texttt{believes}(\texttt{u}''_\texttt{i}, \texttt{u}_\texttt{j}, \texttt{AG}) \texttt{ :- } \texttt{pw}(\texttt{u}''_\texttt{i}), \texttt{pw}(\texttt{u}'_\texttt{i}), \texttt{pw}(\texttt{u}_\texttt{j}), \texttt{believes}(\texttt{u}'_\texttt{i}, \texttt{u}_\texttt{j}, \texttt{AG}), \texttt{oblivious}(\texttt{AG}). \qquad (i, j \in \{1, 2\})$$




$u''_1(\mathcal{F}) = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

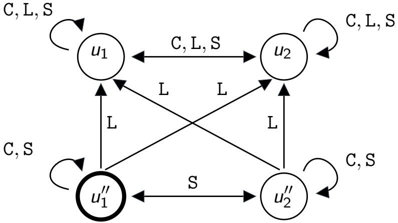
$u''_2(\mathcal{F}) = \{\texttt{opened}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

# Sensing/Announcement actions



Let *peek* be an ontic action such that

- `Charlie` senses the fluent **heads**
- Only `Charlie` and `Snoopy` are attentive




$u_1(\mathcal{F}) = \{\texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u_2(\mathcal{F}) = \{\texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u_1''(\mathcal{F}) = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u_2''(\mathcal{F}) = \{\texttt{opened}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

# Correctness w.r.t. $m\mathcal{A}^\rho$



Let u, v be two possibilities and $\psi$ be a belief formula.

### Entailment correctness

For each u, we have that $\forall\ \psi\quad u \models_\Phi \psi \text{ iff } u \models_\Gamma \psi$ .

### Initial state generation correctness

For each u, v such that u is the initial state in $m\mathcal{A}^\rho$ and v is the initial state in PLATO then $\forall\ \psi\quad u \models_\Phi \psi \text{ iff } v \models_\Gamma \psi$ .

### Transition function correctness

Let a be an action instance. For each u, v such that $\forall\ \psi\quad u \models_\Phi \psi \text{ iff } v \models_\Gamma \psi$ , then $\forall\ \psi\quad \Phi(a, u) \models_\Phi \psi \text{ iff } \Gamma(a, v) \models_\Gamma \psi$ .

# Experimental evaluation



| **SC**: $\|\mathcal{AG}\| = 9$, $\|\mathcal{F}\| = 12$, $\|\mathcal{A}\| = 14$ | | | | |
|---|---|---|---|---|
| $L$ | `many` | `frumpy` | K-BIS | P-MAR |
| 4 | .24 | .24 | .03 | **.007** |
| 6 | 2.56 | 2.49 | .16 | **.04** |
| 8 | 36.79 | 38.34 | 4.23 | **.30** |
| 9 | 204.52 | 146.343 | 5.79 | **.83** |
| 10 | `TO` | 839.27 | 7.36 | **1.78** |

| **Gr**: $\|\mathcal{AG}\| = 3$, $\|\mathcal{F}\| = 9$, $\|\mathcal{A}\| = 24$ | | | | |
|---|---|---|---|---|
| $L$ | Total | Ground | Solve | Atoms |
| 3 | .97 | .60 | .36 | 28'615 |
| 4 | 4.25 | 2.24 | 2.01 | 42'022 |
| 5 | 32.83 | 2.52 | 30.31 | 71'482 |
| 6 | 211.69 | 5.27 | 206.41 | 140'305 |
| 7 | 1066.80 | 16.94 | 1049.86 | 302'623 |

# Experimental evaluation



| **SC**: $\|\mathcal{AG}\| = 9,\ \|\mathcal{F}\| = 12,\ \|\mathcal{A}\| = 14$ | | | | |
|---|---|---|---|---|
| $L$ | `many` | `frumpy` | K-BIS | P-MAR |
| 4 | .24 | .24 | .03 | **.007** |
| 6 | 2.56 | 2.49 | .16 | **.04** |
| 8 | 36.79 | 38.34 | 4.23 | **.30** |
| 9 | 204.52 | 146.343 | 5.79 | **.83** |
| 10 | `TO` | 839.27 | 7.36 | **1.78** |

| **Gr**: $\|\mathcal{AG}\| = 3,\ \|\mathcal{F}\| = 9,\ \|\mathcal{A}\| = 24$ | | | | |
|---|---|---|---|---|
| $L$ | Total | Ground | Solve | Atoms |
| 3 | .97 | .60 | .36 | 28'615 |
| 4 | 4.25 | 2.24 | 2.01 | 42'022 |
| 5 | 32.83 | 2.52 | 30.31 | 71'482 |
| 6 | 211.69 | 5.27 | 206.41 | 140'305 |
| 7 | 1066.80 | 16.94 | 1049.86 | 302'623 |

| | **CC_1**: $\|\mathcal{AG}\| = 2,\ \|\mathcal{F}\| = 10,\ \|\mathcal{A}\| = 16$ | | | | **CC_2**: $\|\mathcal{AG}\| = 3,\ \|\mathcal{F}\| = 13,\ \|\mathcal{A}\| = 24$ | | | |
|---|---|---|---|---|---|---|---|---|
| $L$ | `single` | `multi` | K-BIS | P-MAR | `single` | `multi` | K-BIS | P-MAR |
| 3 | 48.74 | 6.52 | .08 | **.02** | 153.76 | 14.07 | .13 | **.03** |
| 4 | 188.32 | 8.74 | .16 | **.03** | `TO` | 28.02 | .54 | **.10** |
| 5 | `TO` | 7.68 | 1.14 | **.16** | `TO` | 16.13 | 4.89 | **.60** |
| 6 | 1222.67 | 10.83 | 4.42 | **0.64** | `TO` | 14.84 | 12.66 | **1.71** |
| 7 | `TO` | 30.08 | 16.06 | **2.61** | `TO` | 56.48 | 142.06 | **12.37** |

Chapter 5

# Conclusions

# Conclusions



- Exploited a *declarative* approach to implement Multi-Agent Epistemic Planning
- Improved readability and code maintenance
- Straightforward semantical adaptations
- Results comparable to the imperative approach
- Formal *proof* of correctness

# Future works



- Enhancement of the entailment rules
- Implementation of heuristics
- Formal proof of equivalence between $m\mathcal{A}^*$ and $m\mathcal{A}^\rho$
- We are using PLATO to implement novel concepts in MEP, such as *trust*, *lies* and *misconceptions*

# The end




**Thank You**
**for the attention**

# References I


[BA11] Thomas Bolander and Mikkel Birkegaard Andersen. "Epistemic planning for single-and multi-agent systems". In: *Journal of Applied Non-Classical Logics* 21.1 (2011), pp. 9–34.

[Fab+20] Francesco Fabiano et al. "EFP 2.0: A Multi-Agent Epistemic Solver with Multiple E-State Representations". In: *Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling, Nancy, France, October 26-30, 2020*. AAAI Press, 2020, pp. 101–109. URL: `https://aaai.org/ojs/index.php/ICAPS/article/view/6650`.

# References II


[Geb+19] Martin Gebser et al. "Multi-shot ASP solving with clingo". In: *Theory and Practice of Logic Programming* 19 (2019), pp. 27–82. DOI: 10.1017/S1471068418000054.

[GG97] J. Gerbrandy and W. Groeneveld. "Reasoning about information change". In: *Journal of Logic, Language and Information* 6.2 (1997), pp. 147–169. DOI: 10.1023/A:1008222603071.

[Hua+17] X. Huang et al. "A general multi-agent epistemic planner based on higher-order belief change". In: 2017, pp. 1093–1101.

[Mui+15] Christian J. Muise et al. "Planning Over Multi-Agent Epistemic States: A Classical Planning Approach". In: *Proc. of AAAI*. 2015, pp. 3327–3334.

# References III

[Son+14] Tran Cao Son et al. "Finitary S5-theories". In: *European Workshop on Logics in Artificial Intelligence*. Springer. 2014, pp. 239–252.

# Ohter Planners

To the best of our knowledge `EFP 2.0` is the only *comprehensive* epistemic multi-agent planner.

Other planners with the best results in the literature are:

- `RP-MEP` [Mui+15]: translates into classical planning. Only deals with a finite level of nested beliefs.
- `MEPK` [Hua+17]: does not support dynamic common knowledge.

# Domains I

- *Assembly Line* (**AL**): two agents are responsible for processing a different part of a product. They can fail in processing their part and inform the other of the status of her task. The agents decide to *assemble* the product or *restart*. Goal: the agents must assemble the product. We change the *depth* of the belief formulae.
- *Coin in the Box* (**CB**). $n \geq 3$ agents are in a room. There is a closed box containing a coin. None of the agents know the coin position. One agent has the key. An agent may look inside the box to sense the state of the coin and also share the result.

# Domains II

- *Collaboration and Communication* (**CC**). $n \geq 2$ agents move along a corridor with $k \geq 2$ rooms in which $m \geq 1$ boxes can be located. Agents can determine if a certain box is in the room they are in. They can communicate information about the boxes' position. Agents may move only to adjacent rooms.
- *Grapevine*. $n \geq 2$ agents are located in $k \geq 2$ rooms. Each agent ag knows a "secret" (`s_ag`). Agents can move to an adjacent room and share their secret within the same room.
- *Selective Communication* (**SC**). $n \geq 2$ agents within one of the $k \geq 2$ rooms in a corridor. Agents can move to an adjacent room. In only one of the rooms, agents may acquire some information `q` and may communicate it to others.

# Finitary S5 Theories

## Finitary **S5**-theory [Son+14]

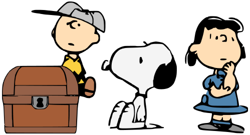
Let $\phi$ be a fluent formula and let $\mathsf{i} \in \mathcal{AG}$ be an agent. A *finitary* **S5**-*theory* is a collection of formulae of the form:

$$(i)\ \phi \quad (ii)\ C\ \phi \quad (iii)\ C\ (B_{\mathsf{i}}\phi \vee B_{\mathsf{i}}\neg\phi) \quad (iv)\ C\ (\neg B_{\mathsf{i}}\phi \wedge \neg B_{\mathsf{i}}\neg\phi)$$

Each fluent $\mathsf{f} \in \mathcal{F}$ must appear in at least one of the formulae (*ii*)–(*iv*) (for at least one agent $\mathsf{i} \in \mathcal{AG}$).

A finitary **S5**-theory has *finitely many* S5-models up to equivalence.

# Initial state generation



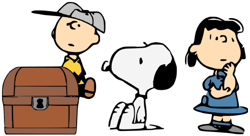
Given

- $\mathcal{AG} = \{\mathtt{Snoopy}, \mathtt{Charlie}, \mathtt{Lucy}\}$
- $\mathcal{F} = \{\mathtt{opened}, \mathtt{head}, \mathtt{looking}_{\mathtt{ag}}\}$ $\mathtt{ag} \in \mathcal{AG}$

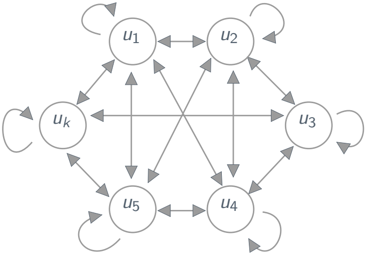
# Initial state generation

Given

- $\mathcal{AG} = \{\texttt{Snoopy}, \texttt{Charlie}, \texttt{Lucy}\}$
- $\mathcal{F} = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{ag}}\}$ $\texttt{ag} \in \mathcal{AG}$



Consider a formula of a finitary **S5** theory.

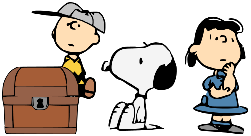



Formula type:

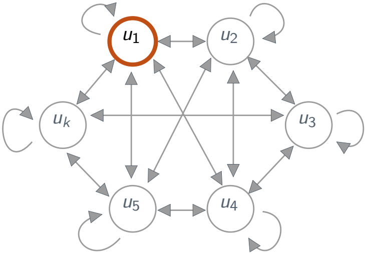
# Initial state generation

Given

- $\mathcal{AG} = \{\texttt{Snoopy}, \texttt{Charlie}, \texttt{Lucy}\}$
- $\mathcal{F} \quad = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{ag}}\}$ $\texttt{ag} \in \mathcal{AG}$



Consider a formula of a finitary **S5** theory.



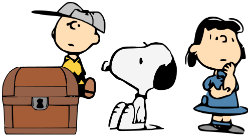

Formula type:

(i) $\phi$

# Initial state generation

Given

- $\mathcal{AG} = \{\texttt{Snoopy}, \texttt{Charlie}, \texttt{Lucy}\}$
- $\mathcal{F} = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{ag}}\}$ $\texttt{ag} \in \mathcal{AG}$



Consider a formula of a finitary **S5** theory.


$u_1(\mathcal{F}) = \{\texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u_2(\mathcal{F}) = \{\texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

Formula type:

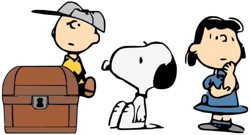
(i) $\phi$

(ii) $C\ \phi$

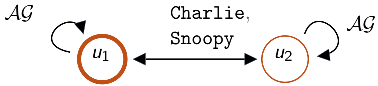
# Initial state generation

Given

- $\mathcal{AG} = \{\texttt{Snoopy}, \texttt{Charlie}, \texttt{Lucy}\}$
- $\mathcal{F} \quad = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{ag}}\}$ ag $\in \mathcal{AG}$



Consider a formula of a finitary **S5** theory.




$u_1(\mathcal{F}) = \{\texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u_2(\mathcal{F}) = \{\texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

Formula type:

(i) $\phi$

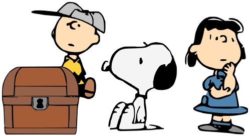
(ii) $C\ \phi$

(iii) $C\ (B_{\mathrm{i}}\phi \vee B_{\mathrm{i}}\neg\phi)$

Formula:

$C\ (B_{\texttt{Lucy}}\texttt{head} \vee B_{\texttt{Lucy}}\neg\texttt{head})$

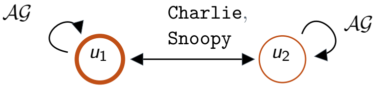
# Initial state generation



Given

- $\mathcal{AG} = \{\texttt{Snoopy}, \texttt{Charlie}, \texttt{Lucy}\}$
- $\mathcal{F} \quad = \{\texttt{opened}, \texttt{head}, \texttt{looking}_{\texttt{ag}}\}$ $\texttt{ag} \in \mathcal{AG}$

Consider a formula of a finitary **S5** theory.




$u_1(\mathcal{F}) = \{\texttt{head}, \texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

$u_2(\mathcal{F}) = \{\texttt{looking}_{\texttt{Charlie}}, \texttt{looking}_{\texttt{Snoopy}}\}$

Formula type:

(i) $\phi$

(ii) $C\ \phi$

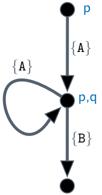
(iii) $C\ (B_i\phi \vee B_i\neg\phi)$

(iv) $C\ (\neg B_i\phi \wedge \neg B_i\neg\phi)$

# From Possibilities to Kripke Structures

Considering a possibility

A possibility

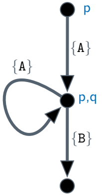


# From Possibilities to Kripke Structures

Considering a possibility

Can be expressed as a *system of equations*

A possibility




Its system of equation

$$\begin{cases} w(p) = 1 & w(q) = 0 \\ v(p) = 1 & v(q) = 1 \\ u(p) = 0 & u(q) = 0 \\ w(\mathtt{A}) = \{v\} & w(\mathtt{B}) = \{\emptyset\} \\ v(\mathtt{A}) = \{v\} & v(\mathtt{B}) = \{u\} \\ u(\mathtt{A}) = \{\emptyset\} & u(\mathtt{B}) = \{\emptyset\} \end{cases}$$

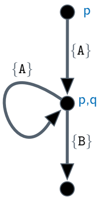
# From Possibilities to Kripke Structures

Considering a possibility

Can be expressed as a *system of equations*

Systems of equations have unique solutions

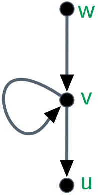
A possibility




Its system of equation

$$\begin{cases} w(p) = 1 & w(q) = 0 \\ v(p) = 1 & v(q) = 1 \\ u(p) = 0 & u(q) = 0 \\ w(\mathtt{A}) = \{v\} & w(\mathtt{B}) = \{\emptyset\} \\ v(\mathtt{A}) = \{v\} & v(\mathtt{B}) = \{u\} \\ u(\mathtt{A}) = \{\emptyset\} & u(\mathtt{B}) = \{\emptyset\} \end{cases}$$

The solution



# From Possibilities to Kripke Structures

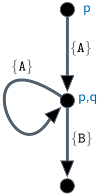
Considering a possibility

Can be expressed as a *system of equations*

Systems of equations have unique solutions

The solution decorates a Kripke structure

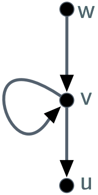
A possibility



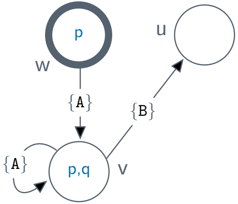

Its system of equation

$$\begin{cases} w(p) = 1 & w(q) = 0 \\ v(p) = 1 & v(q) = 1 \\ u(p) = 0 & u(q) = 0 \\ w(\mathtt{A}) = \{v\} & w(\mathtt{B}) = \{\emptyset\} \\ v(\mathtt{A}) = \{v\} & v(\mathtt{B}) = \{u\} \\ u(\mathtt{A}) = \{\emptyset\} & u(\mathtt{B}) = \{\emptyset\} \end{cases}$$

The solution




Relative Kripke Structure

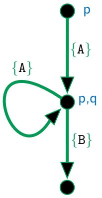


# From Possibilities to Kripke Structures

Considering a possibility
Can be expressed as a *system of equations*
Systems of equations have unique solutions
The solution decorates a Kripke structure

| A possibility | Its system of equation | The solution | Relative Kripke Structure |
|---|---|---|---|

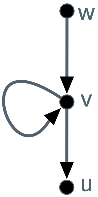
**A possibility**




**Its system of equation**

$$\begin{cases} \mathrm{w}(p) = 1 & \mathrm{w}(q) = 0 \\ \mathrm{v}(p) = 1 & \mathrm{v}(q) = 1 \\ \mathrm{u}(p) = 0 & \mathrm{u}(q) = 0 \\ \mathrm{w}(\mathtt{A}) = \{\mathrm{v}\} & \mathrm{w}(\mathtt{B}) = \{\emptyset\} \\ \mathrm{v}(\mathtt{A}) = \{\mathrm{v}\} & \mathrm{v}(\mathtt{B}) = \{\mathrm{u}\} \\ \mathrm{u}(\mathtt{A}) = \{\emptyset\} & \mathrm{u}(\mathtt{B}) = \{\emptyset\} \end{cases}$$

**The solution**




**Relative Kripke Structure**