Modelling Multi-Agent Epistemic Planning in ASP
Supplementary Documentation

Alessandro Burigana[1], Francesco Fabiano[1], Agostino Dovier[1], Enrico Pontelli[2]

[1] DMIF Department, University of Udine, I-33100 Udine, Italy
burigana.alessandro@spes.uniud.it
{francesco.fabiano,agostino.dovier}@uniud.it
[2] Computer Science, New Mexico State University, Las Cruces, NM 88003, USA
epontell@cs.nmsu.edu

The following document provides supplementary information for the paper "Modelling Multi-Agent Epistemic Planning in ASP" submitted to the 36^{th} *International Conference on Logic Programming* (ICLP 2020).

A From Kripke Structures to Possibilities

In this section we try to briefly summarize the ideas behind the introduction of possibilities as epistemic state. The content of this section is mostly derived from Fabiano et al. (2019, 2020) that, in turn, were strongly influenced by Gerbrandy and Groeneveld (1997). For a more informative introduction the reader is addressed to Gerbrandy and Groeneveld (1997); Gerbrandy (1999); Aczel (1988).

A.1 Non-well-founded set theory fundamentals

Let us start by giving some fundamental definitions of non-well-founded set theory. First of all a *well-founded set* is described in Aczel (1988) as follows:

Definition 1 (well-founded set). *Let E be a set, E' one of its elements, E'' any element of E', and so on. A descent is the sequence of steps from E to E', E' to E'', etc. ... A set is* well-founded *(or* ordinary*) when it only gives rise to finite descents.*

Well-founded set theory states that all the sets in the sense of Definition 1 can be represented in the form of graphs, called *pictures*, (as shown in Figure 1). To formalize this concept of 'picture of a set' however it is necessary to introduce the concept of *decoration*:

Definition 2 (Decoration and Picture).

- *A* decoration *of a graph $\mathcal{G}=(V,E)$ is a function δ that assigns to each node $\mathsf{n} \in V$ a set δ_n in such a way that the elements of δ_n are exactly the sets assigned to successors of n,* i.e., *$\delta_\mathsf{n} = \{\delta_{\mathsf{n}'} \mid (\mathsf{n},\mathsf{n}') \in E\}$.*
- *If δ is a decoration of a pointed graph $(\mathcal{G},\mathsf{n})$, then $(\mathcal{G},\mathsf{n})$ is a* picture *of the set δ_n.*

Moreover, in well-founded set theory, it holds the Mostovski's lemma: "each well-founded graph[3] is a picture of exactly one set".

On the other hand in Aczel (1988) a *non-well-founded*, or *extraordinary set* in the sense of Mirimanoff, is a set that respects Definition 3.

[3] A well-founded graph is a graph that doesn't contain an infinite path $\mathsf{n} \to \mathsf{n}' \to \mathsf{n}'' \to \ldots$ of successors.

Definition 3 (Non-well-founded set). *A set is* non-well-founded *(or* extraordinary*) when among its descents there are some which are infinite.*

In fact, when the **Foundation Axiom**[4] is substituted by the **Anti-Foundation Axiom** (**AFA**), expressed by Aczel in Aczel (1988) as "*Every graph has a unique decoration*", the following consequences become true:

- Every graph is a picture of exactly one set (**AFA** as is formulated in Gerbrandy (1999));
- non-well-founded sets exist given that a non-well-founded pointed graph has to be a picture of a non-well-founded set.

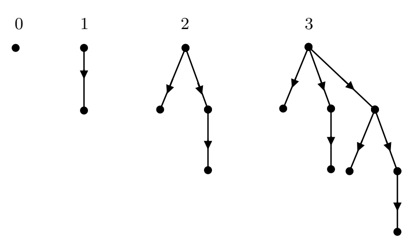
In Aczel (1988); Gerbrandy (1999) it is pointed out how non-well-founded sets can also be expressed through systems of equations. This concept will help us to formalize the notion of e-state in our action language.

A quick example of this representation can be derived by the set $\Omega = \{\Omega\}$ (Figure 2). We can, in fact, informally define this set by the (singleton) system of equations $x = \{x\}$. Systems of equations and their solutions are described more formally as follows in Gerbrandy (1999):

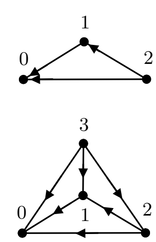
Definition 4 (System of equations). *For each class of atoms*[5] $\mathcal{X}$ *a* system of equation *in* $\mathcal{X}$ *is a class* τ *of equations* $\mathsf{x} = \mathtt{X}$*, where* $\mathsf{x} \in \mathcal{X}$ *and* $\mathtt{X} \subseteq \mathcal{X}$*, such that* τ *contains exactly one equation* $\mathsf{x} = \mathtt{X}$ *for each* $\mathsf{x} \in \mathcal{X}$*. A solution to a system of equations* τ *is a function* δ *that assigns to each* $\mathsf{x} \in \tau(\mathcal{X})$[6] *a set* δ_{x} *such that* $\delta_{\mathsf{x}} = \{\delta_{\mathsf{y}} \mid \mathsf{y} \in \mathtt{X}\}$*, where* $\mathsf{x} = \mathtt{X}$ *is an equation of* τ*. If* δ *is the solution to a system of equations* τ*, then the set* $\{\delta_{\mathsf{x}} \mid \mathsf{x} \in \tau(\mathcal{X})\}$ *is called the solution set of that system.*

Since both graphs and systems of equations are representations for non-well-founded sets, it is natural to investigate their relationships. In particular, it is interesting to point out how from a graph $\mathcal{G}{=}(V, E)$ it is possible to construct a system of equations τ and vice versa. The nodes in $\mathcal{G}$, in fact, can be the set of atoms $\tau(\mathcal{X})$ and, for each node $\mathsf{v} \in V$, an equation is represented by $\mathsf{v} = \{\mathsf{v}' \mid (\mathsf{v}, \mathsf{v}') \in E\}$. Since each graph has a unique decoration, each system of equations has a unique solution. This is also true when we consider bisimilar systems of equations. In fact we can collapse them into their minimal representation thanks to the concept of *maximum bisimulation*

[4] Expressed in Gerbrandy (1999) as "Only well-founded graphs have decorations".
[5] Objects that are not sets and have no further set-theoretic structure.
[6] $\tau(\mathcal{X})$ denotes the class of atoms $\mathcal{X}$ in which τ is described.



(a) Pictures of von Neumann ordinals where $0 = \emptyset$; $1 = \{\emptyset\}$; $2 = \{\emptyset, \{\emptyset\}\}$; $3 = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$.



(b) Alternative Pictures of von Neumann ordinals 2 and 3.

Figure 1: Well-founded sets represented through graphs Aczel (1988).

(a) Standard picture Ω. (b) Unfolding of the picture of Ω.

Figure 2: Representation of the non-well-founded set $\Omega = \{\Omega\}$ Aczel (1988).

as introduced in Dovier (2015). Bisimilar labeled graphs (or Kripke structures) have therefore a unique solution as well since we collapse their representations into the minimal one.

A.2 Possibilities

Let us introduce the notion of possibility, as in Gerbrandy and Groeneveld (1997):

Definition 5 (Possibilities). *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* $\mathsf{f} \in \mathcal{F}$ *a truth value* $\mathsf{u}(\mathsf{f}) \in \{0, 1\}$ *and to each agent* $\mathsf{ag} \in \mathcal{AG}$ *an information state* $\mathsf{u}(\mathsf{ag}) = \sigma$.
- *An* information state σ *is a set of possibilities.*

In PLATO this concept is used to describe an e-state of the planning problem. The intuition behind this idea is that a possibility u is a possible interpretation of the world and of the agents' beliefs; in fact $\mathsf{u}(\mathsf{f})$ specifies the truth value of the fluent f in u and $\mathsf{u}(\mathsf{A})$ is the set of all the interpretations the agent A considers possible in u.

Moreover a possibility can be pictured as a decoration of a labeled graph and therefore as a unique solution to a system of equations for possibilities (Definition 6). A possibility represents the solution to the minimal system of equations in which all bisimilar systems of equations are collapsed; that is the possibilities that represent decorations of bisimilar labeled graphs are bisimilar and can be represented by the minimal one. This shows that the class of bisimilar labeled graphs and, therefore, of bisimilar Kripke structures, used by $m\mathcal{A}^*$ as e-states, can be represented by a single possibility.

Definition 6 (Equations for possibilities). *Given a set of agents $\mathcal{AG}$ and a set of propositional variables $\mathcal{F}$, a* system of equations for possibilities *in a class of possibilities $\mathcal{X}$ is a set of equations such that for each $\mathsf{x} \in \mathcal{X}$ there exists exactly one equation of the form $\mathsf{x}(\mathsf{f}) = i$, where $i \in \{0, 1\}$, for each $\mathsf{f} \in \mathcal{F}$, and of the form $\mathsf{x}(\mathsf{ag}) = \mathtt{X}$, where $\mathtt{X} \subseteq \mathcal{X}$, for each $\mathsf{ag} \in \mathcal{AG}$.*
A solution to a system of equations for possibilities is a function δ that assigns to each atom x a possibility δ_{x} in such a way that if $\mathsf{x}(\mathsf{f}) = i$ is an equation then $\delta_{\mathsf{x}(\mathsf{f})} = i$, and if $\mathsf{x}(\mathsf{ag}) = \sigma$ is an equation, then $\delta_{\mathsf{x}(\mathsf{ag})} = \{\delta_{\mathsf{y}} \mid \mathsf{y} \in \sigma\}$.

B PLATO correctness w.t.r. $m\mathcal{A}^\rho$

In what follows we will demonstrate that PLATO (e**P**istemic mu**L**ti-agent **A**nswer se**T** programming s**O**lver) is correct with respect to the semantic of the language $m\mathcal{A}^\rho$, introduced in Fabiano et al. (2020). In particular, we will prove the correctness of: i) the initial state construction (Proposition 2);

ii) the entailment (Proposition 1); and iii) the transition function (Propositions 3-5). For the sake of readability each type of action, namely *ontic*, *sensing* and *announcement*, will be treat separately. To improve the clarity we will only describe features of $m\mathcal{A}^\rho$ that are useful for the demonstrations.

Complete introductions to $m\mathcal{A}^\rho$ and to multi-agent epistemic planning can be found in Fabiano et al. (2020, 2019) and Le et al. (2018); Kominis and Geffner (2017, 2015); Baral et al. (2015); Muise et al. (2015); Fagin and Halpern (1994) respectively.

B.1 Preliminary concepts

First of all let us recall some concepts, introduced in Fabiano et al. (2020); Baral et al. (2015), that will ease the comprehension of the demonstrations.

Let us start with the definition of an *epistemic planning domain*. Intuitively, an epistemic planning domain contains all the necessary information to define a planning problem in a multi-agent epistemic scenario.

Definition 7 (Multi-agent epistemic planning domain). *We define a* multi-agent epistemic domain *as the tuple* $D = \langle \mathcal{F}, \mathcal{AG}, \mathcal{A}, \varphi_i, \varphi_g \rangle$ *where:*

- $\mathcal{F}$ *is the set of all the* fluents *of* D*;*
- $\mathcal{AG}$ *is the set of the* agents *of* D*;*
- $\mathcal{A}$ *represents the set of all the* actions *of* D*;*
- φ_i *is belief formula that describes the* initial conditions *of the planning process; and*
- φ_g *is belief formula that represents the* goal *conditions.*

Moreover, from now on, with the term *action instance* we will indicate an element of the set $\mathcal{AI} = \mathcal{A} \times \mathcal{AG}$. Intuitively, an action instance $\mathsf{a}\langle\mathsf{ag}\rangle$ identifies the execution of the action a by the agent ag.

Given a domain D we will refer to its components through the *parenthesis* operator. For instance, to access the elements $\mathcal{F}$ and $\mathcal{AG}$ of D we will use the more compact notation $D(\mathcal{F})$ and $D(\mathcal{AG})$, respectively. Allow us to make use of the compact notations: i) $D(\mathcal{BF})$; and ii) $D(\mathcal{S})$ to indicate: i) the set of belief formulae that can be built starting from $D(\mathcal{F})$ and $D(\mathcal{AG})$; and ii) the set of all the e-states reachable from $D(\varphi_i)$ with a finite sequence of elements of $D(\mathcal{AI})$, respectively.

As already mentioned the language $m\mathcal{A}^\rho$ distinguishes between three types of action.

Definition 8 (Action types in $m\mathcal{A}^\rho$). *Given an action* a*, a fluent* f *and a fluent formula* ϕ *the three types of action are:*

- Ontic *action, of the form "*a `causes` f*", used by an agent to modify certain properties of the world.*
- Sensing *action, of the form "*a `determines` f*", used by an agent to refine her beliefs about the world.*
- Announcement *action, of the form "*a `announces` ϕ*", used by an agent to affect the beliefs of other agents.*

Finally, another important concept of multi-agent epistemic planning is the *action observability*.

Definition 9 (Action Observability). *The execution of an action might change or not the beliefs of an agent depending on whether or not she is aware of the action's occurrence.* $m\mathcal{A}^\rho$ *identifies three levels of action observability given an action* a*, an agent* ag*:*

- fully observant *(denoted by* $\mathsf{ag} \in F_\mathsf{a}$*) if* ag *knows about the execution of* a *and about its effects on the world;*
- partially observant *(denoted by* $\mathsf{ag} \in P_\mathsf{a}$*) if* ag *knows about the execution of* a *but she does not know how* a *affected the world;*
- oblivious *(denoted by* $\mathsf{ag} \in O_\mathsf{a}$*) if* ag *does not know about the execution of* a*.*

Let us note that partial observability for world-altering actions is not admitted as, whenever an agent is aware of the execution of an ontic action, she must know its effects on the world as well.

Abbreviations. To avoid unnecessary clutter instead of using the predicate `possible_world`(T, R, P) to identify a generic possibility we will write `pos`(u) where the lowercase letter in `typewriter font` (generally u, v or p) identifies a generic triple (T, R, P). Whenever possible we will present a more "concrete" version of the ASP predicates by removing parts of the rule that are not necessary to capture its semantics. For example the rule for entailing a fluent f, that in ASP has the generic form:

`entails`(T, R, P, F) :- `time`(T), `holds`(T, R, P, F), `possible_world`(T, R, P), `fluent`(F).

will be rewritten as:

`entails`(u, f) :- `holds`(u, f), `fluent`(f).

Moreover let us make use of the notations Γ and Φ to identify PLATO's and $m\mathcal{A}^\rho$'s transition function respectively. In the following proofs we will use p' instead of $\Gamma(\mathsf{a}, \mathsf{p})$ or $\Phi(\mathsf{a}, \mathsf{p})$ when this does not cause ambiguity.

We are now ready to demonstrate the correctness of PLATO w.r.t. $m\mathcal{A}^\rho$.

B.2 Entailment correctness

As first step we need to demonstrate that the entailment in PLATO is correct w.r.t. the one introduced in Fabiano et al. (2020). To do that we will identify the predicates in PLATO that corresponds with an entailment rule in $m\mathcal{A}^\rho$ and prove their correctness. Let us begin by re-introducing the definition of entailment as defined in Fabiano et al. (2020).

Definition 10 (Entailment w.r.t. possibilities). *Let a domain* D*, the belief formulae* $\varphi, \varphi_1, \varphi_2 \in D(\mathcal{BF})$*, a fluent* $\mathsf{f} \in D(\mathcal{F})$*, an agent* $\mathsf{ag} \in D(\mathcal{AG})$*, a group of agents* $\alpha \subseteq D(\mathcal{AG})$*, and a possibility* $\mathsf{u} \in D(\mathcal{S})$ *be given. The entailment in* $m\mathcal{A}^\rho$ *is defined as follows:*

A. $\mathsf{u} \models \mathsf{f}$ *if* $\mathsf{u}(\mathsf{f}) = 1$*;*

B. $\mathsf{u} \models \mathbf{B}_\mathsf{ag}\varphi$ *if for each* $\mathsf{v} \in \mathsf{u}(\mathsf{ag})$, $\mathsf{v} \models \varphi$*;*

C. $\mathsf{u} \models \neg\varphi$ *if* $\mathsf{u} \not\models \varphi$*;*

D. $\mathsf{u} \models \varphi_1 \vee \varphi_2$ *if* $\mathsf{u} \models \varphi_1$ *or* $\mathsf{u} \models \varphi_2$*;*
E. $\mathsf{u} \models \varphi_1 \wedge \varphi_2$ *if* $\mathsf{u} \models \varphi_1$ *and* $\mathsf{u} \models \varphi_2$*;*

F. $\mathsf{u} \models \mathbf{E}_\alpha \varphi$ *if* $\mathsf{u} \models \mathbf{B}_{\mathsf{ag}} \varphi$ *for all* $\mathsf{ag} \in \alpha$*;*
G. $\mathsf{u} \models \mathbf{C}_\alpha \varphi$ *if* $\mathsf{u} \models \mathbf{E}^k_\alpha \varphi$ *for every* $k \geq 0$*, where* $\mathbf{E}^0_\alpha \varphi = \varphi$ *and* $\mathbf{E}^{k+1}_\alpha \varphi = \mathbf{E}_\alpha(\mathbf{E}^k_\alpha \varphi)$*.*

On the other hand, in PLATO, to describe the entailment we make use of the predicates `holds`(u, f) and `entails`(u, φ) where u represents a possibility, f a fluent and φ a belief formula. Intuitively, through the predicate `holds` we identify if a fluent is true or false in any given possibility u while, with the predicate `entails`, it is possible to verify whether a belief formula is derived starting from u.

Definition 11 (Entailment in PLATO). *Let a domain* D*, the belief formulae* $\varphi, \varphi_1, \varphi_2 \in D(\mathcal{BF})$*, a fluent* $\mathsf{f} \in D(\mathcal{F})$*, an agent* $\mathsf{ag} \in D(\mathcal{AG})$*, a group of agents* $\alpha \subseteq D(\mathcal{AG})$*, and a possibility* $\mathsf{u} \in D(\mathcal{S})$ *be given. The predicate* `entails` *in* PLATO *is defined as follows:*

1. $\mathtt{entails}(\mathsf{u}, \mathsf{f})$:- $\mathtt{holds}(\mathsf{u}, \mathsf{f}),\ \mathtt{fluent}(\mathsf{f})$.
2. $\mathtt{entails}(\mathsf{u}, \neg\mathsf{f})$:- $\mathtt{holds}(\mathsf{u}, \neg\mathsf{f}),\ \mathtt{fluent}(\mathsf{f})$.

3. $\mathtt{not\_entails}(\mathsf{u}, \mathtt{b}(\mathsf{ag}, \varphi))$:- $\mathtt{not\ entails}(\mathsf{v}, \varphi),\ \mathtt{believes}(\mathsf{u}, \mathsf{v}, \mathsf{ag})$.
4. $\mathtt{entails}(\mathsf{u}, \mathtt{b}(\mathsf{ag}, \varphi))$:- $\mathtt{not\ not\_entails}(\mathsf{u}, \mathtt{b}(\mathsf{ag}, \varphi))$.

5. $\mathtt{entails}(\mathsf{u}, \mathtt{neg}(\varphi))$:- $\mathtt{not\ entails}(\mathsf{u}, \varphi)$.
6. $\mathtt{entails}(\mathsf{u}, \mathtt{or}(\varphi_1, \varphi_2))$:- $\mathtt{entails}(\mathsf{u}, \varphi_1)$.
7. $\mathtt{entails}(\mathsf{u}, \mathtt{or}(\varphi_1, \varphi_2))$:- $\mathtt{entails}(\mathsf{u}, \varphi_2)$.
8. $\mathtt{entails}(\mathsf{u}, \mathtt{and}(\varphi_1, \varphi_2))$:- $\mathtt{entails}(\mathsf{u}, \varphi_1),\ \mathtt{entails}(\mathsf{u}, \varphi_2)$.

9. $\mathtt{not\_entails}(\mathsf{u}, \mathtt{c}(\alpha, \varphi))$:- $\mathtt{not\ entails}(\mathsf{v}, \varphi),\ \mathtt{reaches}(\mathsf{u}, \mathsf{v}, \mathsf{ag})$.
10. $\mathtt{entails}(\mathsf{u}, \mathtt{c}(\alpha, \varphi))$:- $\mathtt{not\ not\_entails}(\mathsf{u}, \mathtt{c}(\alpha, \varphi))$.

A possibility u reaches v if they satisfy the following rules:

11. $\mathtt{reaches}(\mathsf{u}, \mathsf{v}, \alpha)$:- $\mathtt{believes}(\mathsf{u}, \mathsf{v}, \mathsf{ag}), \mathtt{contains}(\alpha, \mathsf{ag})$.
12. $\mathtt{reaches}(\mathsf{u}, \mathsf{v}, \alpha)$:- $\mathtt{believes}(\mathsf{u}, \mathsf{p}, \mathsf{ag}), \mathtt{contains}(\alpha, \mathsf{ag}), \mathtt{reaches}(\mathsf{p}, \mathsf{v}, \alpha)$.

where `contains/2` is defined by a set of facts.

Proposition 1 (PLATO entailment correctness w.r.t. $m\mathcal{A}^\rho$**).** *Given a domain* D*, the set of its belief formulae* $D(\mathcal{BF})$*, and a possibility* $\mathsf{u} \in D(\mathcal{S})$ *we have that* $\mathsf{u} \models_{m\mathcal{A}^*} \psi$ *iff* $\mathsf{u} \models_{\mathit{PLATO}} \psi\ \forall \psi \in D(\mathcal{BF})$*.*

Proof.
To prove that the ASP encoding of the entailment is correct we will identify each rule of Definition 10 with a predicate of Definition 11.

- Rule A corresponds to Predicates *1* and *2*.

  A. $\mathsf{u} \models \mathsf{f}$ if $\mathsf{u}(\mathsf{f}) = 1$

  1. $\mathtt{entails}(\mathsf{u}, \mathsf{f})$:- $\mathtt{holds}(\mathsf{u}, \mathsf{f}),\ \mathtt{fluent}(\mathsf{f})$.

2. $\texttt{entails}(\mathsf{u}, \neg \mathsf{f})$:- $\texttt{holds}(\mathsf{u}, \neg\mathsf{f})$, $\texttt{fluent}(\mathsf{f})$.

Let us note that the predicate holds correctness is derived from Propositions 2, 3-5. In fact, being the construction of the initial state and the update function correct, it is straightforward to see that $\forall \mathsf{f} \in D(\mathcal{F})$ and $\forall \mathsf{u} \in D(\mathcal{S})$ the predicate $\texttt{holds}(\mathsf{u}, \mathsf{f})$ is true *iff* $\mathsf{u}(\mathsf{f}) = 1$ while $\texttt{holds}(\mathsf{u}, \neg\mathsf{f})$ is true *iff* $\mathsf{u}(\mathsf{f}) = 0$.

- Rule B corresponds to Predicate *4*.

  B. $\mathsf{u} \models \mathsf{f}$ if $\mathsf{u}(\mathsf{f}) = 1$

  3. $\texttt{not\_entails}(\mathsf{u}, \texttt{b}(\texttt{ag}, \varphi))$:- $\texttt{not entails}(\mathsf{v}, \varphi)$, $\texttt{believes}(\mathsf{u}, \mathsf{v}, \texttt{ag})$.
  4. $\texttt{entails}(\mathsf{u}, \texttt{b}(\texttt{ag}, \varphi))$:- $\texttt{not not\_entails}(\mathsf{u}, \texttt{b}(\texttt{ag}, \varphi))$.

  Similarly to the previous point, following Propositions 2, 3-5, we can derive the correctness of the predicate believes and consequently the correctness of reaches. Moreover, for this case we used an auxiliary predicate not_entails (Predicate *3*) that checks whether a given formula φ is not entailed by a possibility v. Namely we calculate the set $\mathcal{U}$ s.t. $\nexists \mathsf{u} \in \mathcal{U}, \mathsf{u} \not\models \varphi$. This can be rewritten as $\forall \mathsf{u} \in \mathcal{U}, \mathsf{u} \models \varphi$. Hence, for formulae of the type $\texttt{b}(\texttt{ag}, \varphi)$ we require that all of the possibilities believed by ag do entail φ as in Rule B.

- Rules C, D and E correspond to by Predicate *5*, Predicates *6*, *7* and Predicate *8* respectively.

  C. $\mathsf{u} \models \neg\varphi$ if $\mathsf{u} \not\models \varphi$;
  D. $\mathsf{u} \models \varphi_1 \vee \varphi_2$ if $\mathsf{u} \models \varphi_1$ or $\mathsf{u} \models \varphi_2$
  E. $\mathsf{u} \models \varphi_1 \wedge \varphi_2$ if $\mathsf{u} \models \varphi_1$ and $\mathsf{u} \models \varphi_2$

  5. $\texttt{entails}(\mathsf{u}, \texttt{neg}(\varphi))$:- $\texttt{not entails}(\mathsf{u}, \varphi)$.
  6. $\texttt{entails}(\mathsf{u}, \texttt{or}(\varphi_1, \varphi_2))$:- $\texttt{entails}(\mathsf{u}, \varphi_1)$.
  7. $\texttt{entails}(\mathsf{u}, \texttt{or}(\varphi_1, \varphi_2))$:- $\texttt{entails}(\mathsf{u}, \varphi_2)$.
  8. $\texttt{entails}(\mathsf{u}, \texttt{and}(\varphi_1, \varphi_2))$:- $\texttt{entails}(\mathsf{u}, \varphi_1)$, $\texttt{entails}(\mathsf{u}, \varphi_2)$.

  These Rules and Predicates represents the inductive steps of the entailment in $m\mathcal{A}^\rho$ and PLATO respectively, and it is straightforward to check their correspondence. The base cases are Rule A for $m\mathcal{A}^\rho$ and Predicates *1*, *2* for PLATO.

- Rule F is used to ease the writing of Rule G without adding any semantic to the entailment and was not necessary to transpose. The formula $\mathbf{E}_\alpha\varphi$ is, in fact, just a rewriting of $\bigwedge_{\mathsf{ag}\in\alpha} \mathbf{B}_{\mathsf{ag}}\varphi$.

- Rule G corresponds to Predicate *10*.

  B. $\mathsf{u} \models \mathbf{C}_\alpha\varphi$ if $\mathsf{u} \models \mathbf{E}^k_\alpha\varphi$ for every $k \geq 0$, where $\mathbf{E}^0_\alpha\varphi = \varphi$ and $\mathbf{E}^{k+1}_\alpha\varphi = \mathbf{E}_\alpha(\mathbf{E}^k_\alpha\varphi)$

  9. $\texttt{not\_entails}(\mathsf{u}, \texttt{c}(\alpha, \varphi))$:- $\texttt{not entails}(\mathsf{v}, \varphi)$, $\texttt{reaches}(\mathsf{u}, \mathsf{v}, \texttt{ag})$.
  10. $\texttt{entails}(\mathsf{u}, \texttt{c}(\alpha, \varphi))$:- $\texttt{not not\_entails}(\mathsf{u}, \texttt{c}(\alpha, \varphi))$.

  Similarly to Predicate *4* for formulae of the type $\texttt{c}(\alpha, \varphi)$ we require that all of the possibilities *reached* by α do entail φ. This is achieved through an auxiliary predicate not_entails (Predicate *9*) that checks whether a given formula φ is not entailed by a possibility v that is *reached* (Predicates *11* and *12*) by α.

B.3 Initial state construction correctness

As already mentioned in the paper the initial state description in $m\mathcal{A}^\rho$ must model a finitary **S5**-theory to ensure a finite number (up to bisimulation) of e-states that can satisfy the initial conditions (Son et al., 2014). For the sake of readability let us formally reintroduce the concept of Finitary **S5**-theory before proving the correctness of PLATO's initial state construction (w.r.t. $m\mathcal{A}^\rho$).

Definition 12 (Finitary S5-theory (Son et al., 2014)). *Let ϕ be a fluent formula and let $\mathsf{ag} \in \mathcal{AG}$ be an agent. A finitary **S5**-theory is a collection of formulae of the form (we use the short form $\boldsymbol{C}\ \phi$ instead of $\boldsymbol{C}_{\mathcal{AG}}\phi$):*

$$(i)\ \phi \qquad (ii)\ \boldsymbol{C}\ \phi \qquad (iii)\ \boldsymbol{C}\ (\boldsymbol{B}_{\mathsf{ag}}\phi \vee \boldsymbol{B}_{\mathsf{ag}}\neg\phi) \qquad (iv)\ \boldsymbol{C}\ (\neg\boldsymbol{B}_{\mathsf{ag}}\phi \wedge \neg\boldsymbol{B}_{\mathsf{ag}}\neg\phi)$$

Moreover, we require each fluent $\mathsf{f} \in \mathcal{F}$ to appear in at least one of the formulae (ii)–(iv).

Proposition 2 (PLATO initial state construction correctness w.r.t. $m\mathcal{A}^\rho$). *Given a domain D, the set of its belief formulae $D(\mathcal{BF})$, two possibilities $\mathsf{u}, \mathsf{v} \in D(\mathcal{S})$ such that u is the initial state in $m\mathcal{A}^\rho$ and v is the initial state in PLATO then $\mathsf{u} \models \psi$ iff $\mathsf{v} \models \psi$ $\forall\psi \in D(\mathcal{BF})$.*

Proof.
To prove that the initial state generated in PLATO is equal to the one derived in $m\mathcal{A}^\rho$ we will show that PLATO has the same behavior as $m\mathcal{A}^\rho$for each type of initial condition (formulae (i)–(iv)).

(ii) For a clearer demonstration let us start from the second type of condition, *i.e.*, $\mathbf{C}\phi$. This formulae are used to determine the set of possible worlds that are contained in the initial e-state. A fluent f is *initially known* if there exists a formula $\mathbf{C}(\mathsf{f})$ or $\mathbf{C}(\neg\mathsf{f})$. In the former case, all the initial possible world must derive that f is true, whereas in the latter that f is false. If there are no such formulae for f, then it is said to be *initially unknown*.
By definition $m\mathcal{A}^\rho$ initial e-state contains all the worlds s.t.: 1) are consistent in their fluents' truth value; 2) entail the correct truth value for each *initially known* fluent; and 3) generate all the different combinations of the *initially unknown* fluents. At the same manner PLATO determines the set of possible world (*i.e.*, `possible_world`) trough the following predicates:

13. `unknown_initially`(p) :- `not initially`$(\mathbf{C}(p))$, `fluent`(p).
14. `initial_dim(2**K)` :- `K = {fluent`(p): `unknown_initially`(p)`}`.
15. `possible_world(1..K)` :- `initial_dim(K)`.

16. `holds`(u, p) :- `initially`$(\mathbf{C}(p))$, `possible_world(u)`, `fluent`(p).
17. `K/2 {     holds(u, f) : possible_world(u)} K/2 :- unknown_initially(f), initial_dim(K)`.
18. `K/2 {not holds(u, f) : possible_world(u)} K/2 :- unknown_initially(f), initial_dim(K)`.

Where p can be either f or $\neg\mathsf{f}$ and the facts `initially`$(\mathbf{C}(p))$ are given.

(i) Formulae of type (i) are used to identify which possibility among the initial ones (determined by the previous step) identifies the pointed world. In particular, this type of condition is used to express the truth values of the fluents in the initial pointed world. That is, every formulae expressed through condition of this type must be true in the initial pointed world. In PLATO this type of condition is expressed as follows:

19. `pointed(u)` :- `initially`(p), `possible_world(u)`, `holds`(u, p), `fluent`(p).

Where p can be either f or $\neg\mathsf{f}$ and the facts `initially`(p) are given.

(*iii*) Formulae of the form $\mathbf{C}(\mathbf{B}_{\mathsf{ag}}\phi \vee \mathbf{B}_{\mathsf{ag}}\neg\phi)$ are used to filter out the edges of the initial state. In particular, during the initial state construction in $m\mathcal{A}^\rho$ formulae of this type remove the edges, labeled with ag, that link two possible worlds that "disagree" on the truth value of ϕ. This is also done in PLATO by means of the following predicates:

20. `not_b_init`(u, v, ag) :- `possible_world`({u; v}), `initially`($\mathbf{C}$(or(b(ag, ϕ), b(ag, $\neg\phi$)))).
21. `not_b_init`(u, v, ag) :- `possible_world`({u; v}), `initially`($\mathbf{C}$(or(b(ag, ϕ), b(ag, $\neg\phi$)))).
22. `believes`(u, v, ag) :- `possible_world`({u; v}), `not not_b_init`(u, v, ag).

(*iv*) Formulae of the type (*iv*) do not filter out any other edges. Since the construction of the initial state is achieved by removing the edges of a complete graph—*i.e.*, being $\mathcal{G}$ the set of initial possibilities, $\forall \mathsf{u} \in \mathcal{G}, \forall \mathsf{ag} \in \mathcal{AG}$ we have that $\mathsf{u}(\mathsf{ag}) = \mathcal{G}$. We can observe that this type of formulae do not contribute to this filtering, hence we do not consider them in the initial state generation in PLATO.

Let us note that, being formulae (*i*)–(*iv*) the only ones allowed, PLATO constructs the initial state only by means of Predicates *13*–*22*.

B.4 Transition function correctness

Allow us to use the compact notation $\mathsf{u}(\mathcal{F}) = \{\mathsf{f} \mid \mathsf{f} \in D(\mathcal{F}) \wedge \mathsf{u} \models \mathsf{f}\} \cup \{\neg\mathsf{f} \mid \mathsf{f} \in D(\mathcal{F}) \wedge \mathsf{u} \not\models \mathsf{f}\}$ and to rewrite the predicate `possible_world` as `poss` for the sake of readability.

B.4.1 Ontic actions correctness

Let a domain D, its set of action instances $D(\mathcal{AI})$, and the set $D(\mathcal{S})$ of all the e-states reachable from $D(\varphi_i)$ with a finite sequence of action instances be given. The transition function $\Phi : D(\mathcal{AI}) \times D(\mathcal{S}) \to D(\mathcal{S}) \cup \{\emptyset\}$ for ontic actions is defined as follows:

Definition 13 ($m\mathcal{A}^\rho$ ontic actions transition function). *Let an ontic action instance* $\mathsf{a} \in D(\mathcal{AI})$*, a possibility* $\mathsf{u} \in D(\mathcal{S})$ *and an agent* $\mathsf{ag} \in D(\mathcal{AG})$ *be given.*

If a *is not executable in* u*, then* $\Phi(\mathsf{a},\mathsf{u}) = \emptyset$ *otherwise* $\Phi(\mathsf{a},\mathsf{u}) = \mathsf{u}'$*, where:*

$$e(\mathsf{a},\mathsf{u}) = \{\ell \mid (\mathsf{a}\ \textit{causes}\ \ell) \in D\};\ \textit{and}$$
$$\overline{e(\mathsf{a},\mathsf{u})} = \{\neg\ell \mid \ell \in e(\mathsf{a},\mathsf{u})\}\ \textit{where}\ \neg\neg\ell\ \textit{is replaced by}\ \ell.$$

H.
$$\mathsf{u}'(\mathsf{f}) = \begin{cases} 1 & \textit{if}\ \mathsf{f} \in (\mathsf{u}(\mathcal{F}) \setminus \overline{e(\mathsf{a},\mathsf{u})}) \cup e(\mathsf{a},\mathsf{u}) \\ 0 & \textit{if}\ \neg\mathsf{f} \in (\mathsf{u}(\mathcal{F}) \setminus \overline{e(\mathsf{a},\mathsf{u})}) \cup e(\mathsf{a},\mathsf{u}) \end{cases}$$

I.
$$\mathsf{u}'(\mathsf{ag}) = \begin{cases} \mathsf{u}(\mathsf{ag}) & \textit{if}\ \mathsf{ag} \in O_{\mathsf{a}} \\ \bigcup\limits_{\mathsf{w} \in \mathsf{u}(\mathsf{ag})} \Phi(\mathsf{a},\mathsf{w}) & \textit{if}\ \mathsf{ag} \in F_{\mathsf{a}} \end{cases}$$

Definition 14 (PLATO ontic actions transition function). *Let an executable ontic action instance* $\mathsf{a} \in D(\mathcal{AI})$ *s.t.* a *causes* ℓ*, the possibilities* $\mathsf{u},\mathsf{v} \in D(\mathcal{S})$*, the possibility* $\mathsf{pointed}_{\mathsf{u}} \in D(\mathcal{S})$ *s.t.* $\mathsf{pointed}_{\mathsf{u}}$ *represents the* pointed *possibility before calculating* $\Gamma(\mathsf{a},\mathsf{u})$ *and an agent* $\mathsf{ag} \in D(\mathcal{AG})$ *be given. The transition function for* a *in PLATO is defined as follows:*

23. :- plan(T, a), not executable(T, a).

24. poss(u′) :- poss(u), reach_fully(pointed$_\mathsf{u}$, u), plan(T, a).

25. holds(u′, ℓ) :- causes(a, ℓ), poss(u), poss(u′), plan(T, a).
26. holds(u′, p) :- not causes(a, p), holds(u, p), poss(u), poss(u′), plan(T, a).

27. believes(u′, v, ag) :- believes(u, v, ag), oblivious(ag, a), poss($\{\mathsf{u}, \mathsf{u}', \mathsf{v}\}$).
28. believes(u′, v′, ag) :- believes(u, v, ag), fully_obs(ag, a), poss($\{\mathsf{u}, \mathsf{u}', \mathsf{v}, \mathsf{v}'\}$).

Where p can be either f *or* ¬f*.*

Proposition 3 (PLATO ontic actions correctness w.r.t. $m\mathcal{A}^\rho$). *Given a domain D, the set of its belief formulae $D(\mathcal{BF})$, an ontic action $\mathsf{a} \in D(\mathcal{A})$ and two possibilities $\mathsf{u}, \mathsf{v} \in D(\mathcal{S})$ such that $\mathsf{u} \models \psi$ iff $\mathsf{v} \models \psi$ $\forall \psi \in D(\mathcal{BF})$ then $\Phi(\mathsf{a}, \mathsf{u}) \models \psi$ iff $\Gamma(\mathsf{a}, \mathsf{v}) \models \psi$ $\forall \psi \in D(\mathcal{BF})$ (where Φ and Γ represents tha transition function of $m\mathcal{A}^\rho$ and PLATO respectively).*

Proof.
To prove that two possibilities generated from two different transition functions, starting from equal possibilities, entail the same formulae we need to demonstrate that the updated possibilities have the same structural properties. To show this we will identify each Rule of Definition 13 with Predicates of Definition 14.

- Rule H corresponds to Predicates *25* and *26*.

  $$H.\ \mathsf{u}'(\mathsf{f}) = \begin{cases} 1 & \text{if } \mathsf{f} \in (\mathsf{u}(\mathcal{F}) \setminus \overline{e(\mathsf{a}, \mathsf{u})}) \cup e(\mathsf{a}, \mathsf{u}) \\ 0 & \text{if } \neg\mathsf{f} \in (\mathsf{u}(\mathcal{F}) \setminus \overline{e(\mathsf{a}, \mathsf{u})}) \cup e(\mathsf{a}, \mathsf{u}) \end{cases}$$

  25. holds(u′, ℓ) :- causes(a, ℓ), poss(u), poss(u′), plan(T, a).
  26. holds(u′, p) :- not causes(a, p), holds(u, p), poss(u), poss(u′), plan(T, a).

  Let us start by showing that the updated possibilities u′ and v′, generated from $\Phi(\mathsf{a}, \mathsf{u})$ and $\Gamma(\mathsf{a}, \mathsf{v})$ respectively, are equal w.r.t. the fluents truth value. Let us consider the case when the action a causes f; in this scenario u′(f) is equal to 1 (Equation H) and holds(v′, f) is valid (Predicate *25*) meaning that both u′ and v′ consider f to be true.
  Similarly when the action a causes ¬f we will have that u′(f) is equal to 0 (Equation H) while the predicate holds(v′, ¬f) is true (Predicate *25*) causing f to be false in u′ and v′.
  Finally we need to show that the fluents that are not modified by the action have the same truth value both in u′ and v′. This is easily derived in $m\mathcal{A}^\rho$ as in Equation H the fluents modified are only the ones that belongs to the set $e(\mathsf{a}, \mathsf{u})$—namely the effects of a—while the other are preserved from $\mathsf{u}(\mathcal{F})$. On the other hand, in PLATO, this is accomplished with Predicate *26* that explicitly sets every fluent that is not an effect of a as it was in v. Given that we assumed u and v to entail the same formulae, and therefore to have the same truth value for fluents, we can conclude that also the fluents not directly modified by a have the same value in u′ and v′.

- After the fluents truth value we need to demonstrate that the beliefs update is the same in both $m\mathcal{A}^\rho$ and PLATO.

– Let us start with the beliefs related to the oblivious agents. The first case of Rule I (Rule I_1) corresponds to Predicate *27*.

  I_1. $\mathsf{u}'(\mathsf{ag}) = \mathsf{u}(\mathsf{ag})$ if $\mathsf{ag} \in O_\mathsf{a}$

  27. `believes`$(\mathsf{u}', \mathsf{v}, \mathsf{ag})$:- `believes`$(\mathsf{u}, \mathsf{v}, \mathsf{ag})$, `oblivious`$(\mathsf{ag}, \mathsf{a})$, `poss`$(\{\mathsf{u}, \mathsf{u}', \mathsf{v}\})$.

  As described in Rule I_1 an oblivious agent ag, from u', believes the same set of possibilities $\mathcal{U}_\mathsf{ag}$ that she believed in u. In PLATO the behavior of an oblivious agent ag is described by Predicate *27* that creates a predicate `believes` from v' to each possibility that belongs to the set $\mathcal{V}_\mathsf{ag}$ of possibilities believed by ag in v. Given that, by definition, u and v must entail the same formulae we have that the sets of possibilities believed by an agent starting from u and v must be equals. In particular this means that the sets $\mathcal{U}_\mathsf{ag}$ and $\mathcal{V}_\mathsf{ag}$ are the same set and, therefore, an oblivious agent's beliefs are the same starting from u' or v'.

– Next we will demonstrate that the beliefs of fully observant agents are equals in u' and v'. The second case of Rule I (Rule I_2) corresponds to Predicate *28*.

  I_2. $\mathsf{u}'(\mathsf{ag}) = \bigcup_{\mathsf{w} \in \mathsf{u}(\mathsf{ag})} \Phi(\mathsf{a}, \mathsf{w})$ if $\mathsf{ag} \in F_\mathsf{a}$

  28. `believes`$(\mathsf{u}', \mathsf{v}', \mathsf{ag})$:- `believes`$(\mathsf{u}, \mathsf{v}, \mathsf{ag})$, `fully_obs`$(\mathsf{ag}, \mathsf{a})$, `poss`$(\{\mathsf{u}, \mathsf{u}', \mathsf{v}, \mathsf{v}'\})$.

  This scenario for $m\mathcal{A}^\rho$ is described in Rule I_2 where it is shown how a fully observant agent ag, starting from u', believes the updated version of the possibilities that she believed starting from u. The same holds for PLATO where Predicate *28* creates a predicate `believes` from v' to every updated version of the possibility belived by ag in v. This means that a fully observant agent, that necessarily believes the same set $\mathcal{P}_\mathsf{ag}$ of possibilities starting from u and v, believes the updated version of $\mathcal{P}_\mathsf{ag}$ starting from u' and v'. As shown in the other points the result of both the transition functions on a possibility p is the same possibility p' and therefore the updated version of $\mathcal{P}_\mathsf{ag}$ is equal in both $m\mathcal{A}^\rho$ and PLATO.

B.4.2 Sensing actions correctness

Let a domain D, its set of action instances $D(\mathcal{AI})$, and the set $D(\mathcal{S})$ of all the e-states reachable from $D(\varphi_i)$ with a finite sequence of action instances be given. The transition function $\Phi : D(\mathcal{AI}) \times D(\mathcal{S}) \to D(\mathcal{S}) \cup \{\emptyset\}$ for sensing actions is defined as follows:

Definition 15 ($m\mathcal{A}^\rho$ sensing actions transition function). *Let a sensing action instance* a $\in D(\mathcal{AI})$ *used to determine the fluent* f*, a possibility* $\mathsf{u} \in D(\mathcal{S})$ *and an agent* $\mathsf{ag} \in D(\mathcal{AG})$ *be given. If* a *is not executable in* u*, then* $\Phi(\mathsf{a}, \mathsf{u}) = \emptyset$ *otherwise* $\Phi(\mathsf{a}, \mathsf{u}) = \mathsf{u}'$*, where:*

$$\begin{aligned} e(\mathsf{a}, \mathsf{u}) =& \{\mathsf{f} \mid (\mathsf{a}\ \textbf{\textit{determines}}\ \mathsf{f}) \in D \land \mathsf{u} \models \mathsf{f}\} \\ & \cup \{\neg \mathsf{f} \mid (\mathsf{a}\ \textbf{\textit{determines}}\ \mathsf{f}) \in D \land \mathsf{u} \not\models \mathsf{f}\} \end{aligned}$$

J. $\mathsf{u}'(\mathcal{F}) = \mathsf{u}(\mathcal{F})$

K. $$\mathsf{u}'(\mathsf{ag}) = \begin{cases} \mathsf{u}(\mathsf{ag}) & \textit{if } \mathsf{ag} \in O_\mathsf{a} \\ \bigcup_{\mathsf{w} \in \mathsf{u}(\mathsf{ag})} \Phi(\mathsf{a}, \mathsf{w}) & \textit{if } \mathsf{ag} \in P_\mathsf{a} \\ \bigcup_{\mathsf{w} \in \mathsf{u}(\mathsf{ag}):\ e(\mathsf{a}, \mathsf{w}) = e(\mathsf{a}, \mathsf{u})} \Phi(\mathsf{a}, \mathsf{w}) & \textit{if } \mathsf{ag} \in F_\mathsf{a} \end{cases}$$

Definition 16 (PLATO sensing actions transition function). *Let an executable sensing action instance* $\mathsf{a} \in D(\mathcal{AI})$ *s.t.* a ***determines*** f*, the possibilities* $\mathsf{u}, \mathsf{v} \in D(\mathcal{S})$*, the possibility* $\mathsf{pointed_u} \in D(\mathcal{S})$ *s.t.* $\mathsf{pointed_u}$ *represents the* pointed *possibility before calculating* $\Gamma(\mathsf{a},\mathsf{u})$ *and an agent* $\mathsf{ag} \in D(\mathcal{AG})$ *be given. The transition function for* a *in PLATO is defined as follows:*

23. :- plan(T, a), not executable(T, a).

29. poss(u′) :- plan(T, a), poss(u), reach_fully(pointed$_u$, u), entails(u, p), entails(pointed$_u$, p).
30. poss(u′) :- plan(T, a), poss(u), believes(pointed$_u$, u, ag), partial_obs(ag, a).
31. poss(u′) :- plan(T, a), poss({u,v}), believes(pointed$_u$, v, ag), partial_obs(ag, a), reach_not_oblivious(v, u).

32. holds(u′, p) :- plan(T, a), poss(u), poss(u′), holds(u, p).

33. believes(u′, v, ag) :- believes(u, v, ag), oblivious(ag, a), poss({u, u′, v}).
34. believes(u′, v′, ag) :- believes(u, v, ag), partial_obs(ag, a), poss({u, u′, v, v′}).
35. believes(u′, v′, ag) :- believes(u, v, ag), fully_obs(ag, a), holds({u, v}, p), poss({u, u′, v, v′}).

Where p *can be either* f *or* ¬f.

Proposition 4 (PLATO sensing actions correctness w.r.t. $m\mathcal{A}^\rho$**).** *Given a domain* D*, the set of its belief formulae* $D(\mathcal{BF})$*, an sensing action* $\mathsf{a} \in D(\mathcal{A})$ *and two possibilities* $\mathsf{u}, \mathsf{v} \in D(\mathcal{S})$ *such that* $\mathsf{u} \models \psi$ *iff* $\mathsf{v} \models \psi$ $\forall \psi \in D(\mathcal{BF})$ *then* $\Phi(\mathsf{a},\mathsf{u}) \models \psi$ *iff* $\Gamma(\mathsf{a},\mathsf{v}) \models \psi$ $\forall \psi \in D(\mathcal{BF})$ *(where* Φ *and* Γ *represents tha transition function of* $m\mathcal{A}^\rho$ *and PLATO respectively).*

Proof.
To prove that two possibilities generated from two different transition functions, starting from equal possibilities, entail the same formulae we need to demonstrate that the updated possibilities have the same structural properties. To show this we will identify each Rule of Definition 15 with Predicates of Definition 16.

- Rule J corresponds to Predicates *32*.

  J. $\mathsf{u}'(\mathcal{F}) = \mathsf{u}(\mathcal{F})$

  32. holds(u′, p) :- plan(T, a), poss(u), poss(u′), holds(u, p).

  Let us start by showing that the updated possibilities u' and v', generated from $\Phi(\mathsf{a},\mathsf{u})$ and $\Gamma(\mathsf{a},\mathsf{v})$ respectively, are equal w.r.t. the fluents truth value. This is easily derived: in fact in $m\mathcal{A}^\rho$ (Equation J) the fluents interpretation in u' is the equal to the fluents interpretation of u and in PLATO the predicates holds are valid on the same fluents interpretation in both v and v' (Predicate *32*). Given that we assumed u and v to entail the same formulae, and therefore to have the same truth value for fluents, we can conclude that also the fluents have the same value in u' and v'.

- After the fluents truth value we need to demonstrate that the beliefs update is the same in both $m\mathcal{A}^\rho$ and PLATO.
  - Let us start with the beliefs related to the oblivious agents. The first case of Rule K (Rule K_1) corresponds to Predicate *33*.

K_1. $\mathsf{u}'(\mathsf{ag}) = \mathsf{u}(\mathsf{ag})$ if $\mathsf{ag} \in O_\mathsf{a}$

33. believes(u′, v, ag) :- believes(u, v, ag), oblivious(ag, a), poss({u, u′, v}).

As for the ontic actions an oblivious agent ag, from u′, believes the same set of possibilities $\mathcal{U}_\mathsf{ag}$ that she believed in u (Rule K_1) and in PLATO ag believes, from v′, the set $\mathcal{V}_\mathsf{ag}$ of possibilities believed by ag in v (Predicate *33*). Given that, by definition, u and v must entail the same formulae we have that the sets of possibilities believed by an agent starting from u and v must be equals. In particular this means that the sets $\mathcal{U}_\mathsf{ag}$ and $\mathcal{V}_\mathsf{ag}$ are the same set and, therefore, an oblivious agent's beliefs are the same starting from u′ or v′.

– Next we need to show that the partially observant agents' beliefs are equals in u′ and v′. The second case of Rule K (Rule K_2) corresponds to Predicate *34*.

K_2. $\mathsf{u}'(\mathsf{ag}) = \bigcup_{\mathsf{w} \in \mathsf{u}(\mathsf{ag})} \Phi(\mathsf{a}, \mathsf{w})$ if $\mathsf{ag} \in P_\mathsf{a}$

34. believes(u′, v′, ag) :- believes(u, v, ag), partial_obs(ag, a), poss({u, u′, v, v′}).

This scenario for $m\mathcal{A}^\rho$ is described by Rule K_2 where it is shown how a partially observant agent ag, starting from u′, beliefs the updated version of the possibilities that she believed starting from u. The same holds for PLATO where Predicate *34* creates a predicate believes from v′ to every updated version of the possibility belived by ag in v. This means that a partially observant agent, that necessarily believes the same set $\mathcal{P}_\mathsf{ag}$ of possibilities starting from u and v, believes the updated version of $\mathcal{P}_\mathsf{ag}$ starting from u′ and v′. As shown in the other points the result of both the transition functions on a possibility p is the same possibility p′ and therefore the updated version of $\mathcal{P}_\mathsf{ag}$ is equal in both $m\mathcal{A}^\rho$ and PLATO.

– Finally we need to demonstrate that also the beliefs of the fully observant agents are equals in u′ and v′. The third case of Rule K (Rule K_3) corresponds to Predicate *35*.

K_3. $\mathsf{u}'(\mathsf{ag}) = \bigcup_{\mathsf{w} \in \mathsf{u}(\mathsf{ag}):\ e(\mathsf{a},\mathsf{w}) = e(\mathsf{a},\mathsf{u})} \Phi(\mathsf{a}, \mathsf{w})$ if $\mathsf{ag} \in F_\mathsf{a}$

29. poss(u′) :- plan(T, a), poss(u), reach_fully(pointed$_\mathsf{u}$, u), entails(u, ϕ), entails(pointed$_\mathsf{u}$, ϕ).
30. poss(u′) :- plan(T, a), poss(u), believes(pointed$_\mathsf{u}$, u, ag), partial_obs(ag, a).
31. poss(u′) :- plan(T, a), poss({u,v}), believes(pointed$_\mathsf{u}$, v, ag), partial_obs(ag, a),
reach_not_oblivious(v, u).

35. believes(u′, v′, ag) :- believes(u, v, ag), fully_obs(ag, a), holds({u, v}, p),
poss({u, u′, v, v′}).

Given that $\Phi(\mathsf{a}, \mathsf{u})$ is assumed to be applied starting from the pointed world we have that a fully observant agent, starting from the pointed possibility, only believes possibilities where f has the same truth value that has in the pointed one. This case is matched exactly in PLATO by the combination of Predicates *29* and *35*. On the other hand, if a world is reached by a fully observant agent not directly from the pointed world—*i.e.*, it is reached by a fully observant through a path of partially and fully observant agents that starts with a partially observant one—its updated version will only have fully observant edges to the updated possibilities with the same interpretation of f. This is because the Rule K_2 is firstly applied and finally (possibly after other applications of Rule K) Rule K_3 is used. In fact, by applying Rule K_2 first, Φ is recursively applied on both possibilities that have and do not have the same interpretation of f w.r.t. the pointed world. it is straightforward to see that this rule is transposed in PLATO trough the combination of Predicates *31* and *35*.

B.4.3 Announcements actions correctness

Let a domain D, its set of action instances $D(\mathcal{AI})$, and the set $D(\mathcal{S})$ of all the e-states reachable from $D(\varphi_i)$ with a finite sequence of action instances be given. The transition function $\Phi : D(\mathcal{AI}) \times D(\mathcal{S}) \rightarrow D(\mathcal{S}) \cup \{\emptyset\}$ for announcement actions is defined as follows.

Definition 17 ($m\mathcal{A}^\rho$ announcement actions transition function). *Let an announcement action instance $\mathsf{a} \in D(\mathcal{AI})$ used to announce the fluent formula ϕ, a possibility $\mathsf{u} \in D(\mathcal{S})$ and an agent $\mathsf{ag} \in D(\mathcal{AG})$ be given.*
If a is not executable in u, then $\Phi(\mathsf{a},\mathsf{u}) = \emptyset$ otherwise $\Phi(\mathsf{a},\mathsf{u}) = \mathsf{u}'$, where:

$$e(\mathsf{a},\mathsf{u}) = \begin{cases} 0 & \textit{if } \mathsf{u} \models \phi \\ 1 & \textit{if } \mathsf{u} \models \neg\phi \end{cases}$$

L. $$\mathsf{u}'(\mathcal{F}) = \mathsf{u}(\mathcal{F})$$

M. $$\mathsf{u}'(\mathsf{ag}) = \begin{cases} \mathsf{u}(\mathsf{ag}) & \textit{if } \mathsf{ag} \in O_\mathsf{a} \\ \bigcup_{\mathsf{w} \in \mathsf{u}(\mathsf{ag})} \Phi(\mathsf{a},\mathsf{w}) & \textit{if } \mathsf{ag} \in P_\mathsf{a} \\ \bigcup_{\mathsf{w} \in \mathsf{u}(\mathsf{ag}):\ e(\mathsf{a},\mathsf{w}) = e(\mathsf{a},\mathsf{u})} \Phi(\mathsf{a},\mathsf{w}) & \textit{if } \mathsf{ag} \in F_\mathsf{a} \end{cases}$$

Definition 18 (PLATO announcement actions transition function). *Let an executable announcement action instance $\mathsf{a} \in D(\mathcal{AI})$ s.t. a **announces** ϕ, the possibilities u,$\mathsf{v} \in D(\mathcal{S})$, the possibility $\mathsf{pointed_u} \in D(\mathcal{S})$ s.t. $\mathsf{pointed_u}$ represents the* pointed *possibility before calculating $\Gamma(\mathsf{a},\mathsf{u})$ and an agent $\mathsf{ag} \in D(\mathcal{AG})$ be given. The transition function for a in PLATO is defined as follows:*

23. :- plan(T, a), not executable(T, a).

36. poss(u′) :- plan(T, a), poss(u), reach_fully(pointed$_u$, u), entails(u, ϕ), entails(pointed$_u$, ϕ).
37. poss(u′) :- plan(T, a), poss(u), believes(pointed$_u$, u, ag), partial_obs(ag, a).
38. poss(u′) :- plan(T, a), poss({u,v}), believes(pointed$_u$, v, ag), partial_obs(ag, a), reach_not_oblivious(v, u).

39. holds(u′, p) :- plan(T, a), poss(u), poss(u′), holds(u, p).

40. believes(u′, v, ag) :- believes(u, v, ag), oblivious(ag, a), poss({u, u′, v}).
41. believes(u′, v′, ag) :- believes(u, v, ag), partial_obs(ag, a), poss({u, u′, v, v′}).
42. believes(u′, v′, ag) :- believes(u, v, ag), fully_obs(ag, a), holds({u, v}, p), poss({u, u′, v, v′}).

Where p can be either f or $\neg\mathsf{f}$.

Proposition 5 (PLATO announcement actions correctness w.r.t. $m\mathcal{A}^\rho$). *Given a domain D, the set of its belief formulae $D(\mathcal{BF})$, an announcement action $\mathsf{a} \in D(\mathcal{A})$ and two possibilities $\mathsf{u},\mathsf{v} \in D(\mathcal{S})$ such that $\mathsf{u} \models \psi$ iff $\mathsf{v} \models \psi$ $\forall\psi \in D(\mathcal{BF})$ then $\Phi(\mathsf{a},\mathsf{u}) \models \psi$ iff $\Gamma(\mathsf{a},\mathsf{v}) \models \psi$ $\forall\psi \in D(\mathcal{BF})$ (where Φ and Γ represents tha transition function of $m\mathcal{A}^\rho$ and PLATO respectively).*

Proof.
To prove that two possibilities generated from two different transition functions, starting from equal possibilities, entail the same formulae we need to demonstrate that the updated possibilities have the same structural properties. To show this we will identify each Rule of Definition 17 with Predicates of Definition 18.

- Rule L corresponds to Predicates *39*.

  L. $\mathsf{u}'(\mathcal{F}) = \mathsf{u}(\mathcal{F})$

  39. `holds`(u', p) :- `plan`(T, a), `poss`(u), `poss`(u'), `holds`(u, p).

  Let us start by showing that the updated possibilities u' and v', generated from $\Phi(\mathsf{a}, \mathsf{u})$ and $\Gamma(\mathsf{a}, \mathsf{v})$ respectively, are equal w.r.t. the fluents truth value. This is easily derived: in fact in $m\mathcal{A}^\rho$ (Equation L) the fluents interpretation in u' is the equal to the fluents interpretation of u and in PLATO the predicates `holds` are valid on the same fluents interpretation in both v and v' (Predicate *39*). Given that we assumed u and v to entail the same formulae, and therefore to have the same truth value for fluents, we can conclude that also the fluents have the same value in u' and v'.

- After the fluents truth value we need to demonstrate that the beliefs update is the same in both $m\mathcal{A}^\rho$ and PLATO.
  - Let us start with the beliefs related to the oblivious agents. The first case of Rule M (Rule M_1) corresponds to Predicate *40*.

    M_1. $\mathsf{u}'(\mathsf{ag}) = \mathsf{u}(\mathsf{ag})$ if $\mathsf{ag} \in O_{\mathsf{a}}$

    40. `believes`$(\mathsf{u}', \mathsf{v}, \mathsf{ag})$:- `believes`(u, v, ag), `oblivious`(ag, a), `poss`$(\{\mathsf{u}, \mathsf{u}', \mathsf{v}\})$.

    As for the ontic actions an oblivious agent ag, from u', believes the same set of possibilities $\mathcal{U}_{\mathsf{ag}}$ that she believed in u (Rule M_1) and in PLATO ag `believes`, from v', the set $\mathcal{V}_{\mathsf{ag}}$ of possibilities believed by ag in v (Predicate *40*). Given that, by definition, u and v must entail the same formulae we have that the sets of possibilities believed by an agent starting from u and v must be equals. In particular this means that the sets $\mathcal{U}_{\mathsf{ag}}$ and $\mathcal{V}_{\mathsf{ag}}$ are the same set and, therefore, an oblivious agent's beliefs are the same starting from u' or v'.

  - Next we need to show that the partially observant agents' beliefs are equals in u' and v'. The second case of Rule M (Rule M_2) corresponds to Predicate *41*.

    M_2. $\mathsf{u}'(\mathsf{ag}) = \bigcup_{\mathsf{w} \in \mathsf{u}(\mathsf{ag})} \Phi(\mathsf{a}, \mathsf{w})$ if $\mathsf{ag} \in P_{\mathsf{a}}$

    41. `believes`$(\mathsf{u}', \mathsf{v}', \mathsf{ag})$:- `believes`(u, v, ag), `partial_obs`(ag, a), `poss`$(\{\mathsf{u}, \mathsf{u}', \mathsf{v}, \mathsf{v}'\})$.

    This scenario for $m\mathcal{A}^\rho$ is described by Rule M_2 where it is shown how a partially observant agent ag, starting from u', beliefs the updated version of the possibilities that she believed starting from u. The same holds for PLATO where Predicate *41* creates a predicate `believes` from v' to every updated version of the possibility belived by ag in v. This means that a partially observant agent, that necessarily believes the same set $\mathcal{P}_{\mathsf{ag}}$ of possibilities starting from u and v, believes the updated version of $\mathcal{P}_{\mathsf{ag}}$ starting from u' and v'. As shown in the other points the result of both the transition functions on a possibility p is the same possibility p' and therefore the updated version of $\mathcal{P}_{\mathsf{ag}}$ is equal in both $m\mathcal{A}^\rho$ and PLATO.

– Finally we need to demonstrate that also the beliefs of the fully observant agents are equals in u' and v'. The third case of Rule M (Rule M_3) corresponds to Predicate *42*.

$$M_3.\ \mathsf{u}'(\mathsf{ag}) = \bigcup_{\mathsf{w}\in\mathsf{u}(\mathsf{ag}):\ e(\mathsf{a},\mathsf{w})=e(\mathsf{a},\mathsf{u})} \Phi(\mathsf{a},\mathsf{w}) \text{ if } \mathsf{ag} \in F_\mathsf{a}$$

36. poss(u′) :- plan(T, a), poss(u), reach_fully(pointed$_\mathsf{u}$, u), entails(u, ϕ), entails(pointed$_\mathsf{u}$, ϕ).
37. poss(u′) :- plan(T, a), poss(u), believes(pointed$_\mathsf{u}$, u, ag), partial_obs(ag, a).
38. poss(u′) :- plan(T, a), poss({u,v}), believes(pointed$_\mathsf{u}$, v, ag), partial_obs(ag, a), reach_not_oblivious(v, u).

42. believes(u′, v′, ag) :- believes(u, v, ag), fully_obs(ag, a), holds({u, v}, p), poss({u, u′, v, v′}).

Given that $\Phi(\mathsf{a},\mathsf{u})$ is assumed to be applied starting from the pointed world we have that a fully observant agent, starting from the pointed possibility, only believes possibilities where ϕ has the same truth value that has in the pointed one. This case is matched exactly in PLATO by the combination of Predicates *36* and *42*. On the other hand, if a world is reached by a fully observant agent not directly from the pointed world—*i.e.*, it is reached by a fully observant through a path of partially and fully observant agents that starts with a partially observant one—its updated version will only have fully observant edges to the updated possibilities with the same interpretation of ϕ. This is because the Rule M_2 is firstly applied and finally (possibly after other applications of Rule M) Rule M_3 is used. In fact, by applying Rule M_2 first, Φ is recursively applied on both possibilities that have and do not have the same interpretation of ϕ w.r.t. the pointed world. It is straightforward to see that this rule is transposed in PLATO trough the combination of Predicates *38* and *42*.

C Domains Description

In this section we provide an in detail description of the benchmarks collected from the literature (Kominis and Geffner, 2015; Huang et al., 2017; Le et al., 2018; Fabiano et al., 2020) used to test PLATO.

- *Assembly Line* (**AL**). In this problem there are two agents, each responsible for processing a different part of a product. Each agent can fail in processing her part and can inform the other agent of the status of her task (action `tell`). Two agents decide to *assemble* (action `act_assemble`) the product or *restart* (action `act_res`), depending on their knowledge about the product status. The goal in this domain is fixed, *i.e.*, the agents must assemble the product, but what varies is the *depth* of the belief formulae used as executability conditions.

- *Coin in the Box* (**CB**). $n \geq 3$ agents are in a room where in the middle there is a box containing a coin. None of the agents know whether the coin lies heads or tails up and the box is locked. One agent has the key to open the box (action `open`). Only *attentive* agents may be aware of the execution of an action. If an agent is attentive, she may look inside the box (action `peek`) to sense the state of the coin. An agent may also share the result (action `shout`). The goals usually consist in some agents knowing whether the coin lies heads or tails up while other agents know that she knows or are ignorant about this.

- *Collaboration and Communication* (**CC**). In this domains $n \geq 2$ agents move along a corridor with $k \geq 2$ rooms in which $m \geq 1$ boxes can be located. Whenever an agent enters a room, she can determine if a certain box is in the room. Moreover, agents can communicate information about the boxes' position to the another *attentive* agents. Initially we place the all the agents inside room 2. The position of the boxes in initially unknown to each agent. An agent ag may move only to adjacent rooms (actions $\mathsf{left}\langle\mathsf{ag}\rangle$ and $\mathsf{right}\langle\mathsf{ag}\rangle$). To verify the presence of a box b and to communicate it to other agents, an agent can perform the actions $\mathsf{check}\langle\mathsf{ag}\rangle(\mathsf{b})$ and $\mathsf{tell}\langle\mathsf{ag}\rangle(\mathsf{b}, \mathsf{ag}_2)$, respectively.

- *Grapevine* (**Gr**). $n \geq 2$ agents are located in $k \geq 2$ rooms. Each agent ag knows a "secret", represented by the fluent s_ag. An agent can move freely to an adjacent room (actions $\mathsf{left}\langle\mathsf{ag}\rangle$ and $\mathsf{right}\langle\mathsf{ag}\rangle$) and she can share a secret with the agents (action $\mathsf{share}\langle\mathsf{ag}\rangle(\mathsf{s})$) that are in the room with her. This domain supports different goals, from sharing secrets with other agents to having misconceptions about agents' beliefs.

- *Selective Communication* (**SC**). **SC** has $n \geq 2$ agents that start in one of the $k \geq 2$ rooms in a corridor. Every agent is free to move from one room to its adjacent (actions `left` and `right`). In only one of the rooms, an agent may acquire some information, represented by the fluent `q`, by performing the action `sense`. Once an agent acquired such information, she may communicate it to others with the action `shout`. Depending on the room in which this action is performed, different agents *observe* the action. The goals usually require some agents to know certain properties while other agents ignore them.

Bibliography

Aczel, P. (1988). Non-well-founded sets. CSLI Lecture Notes, 14.

Baral, C., Gelfond, G., Pontelli, E., and Son, T. C. (2015). An action language for multi-agent domains: Foundations. *CoRR*, abs/1511.01960.

Barwise, J. and Etchemendy, J. (1987). *The liar: An essay on truth and circularity.* Oxford University Press.

Dovier, A. (2015). Logic programming and bisimulation. In Vos, M. D., Eiter, T., Lierler, Y., and Toni, F., editors, *ICLP*, volume 1433 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Fabiano, F., Burigana, A., Dovier, A., and Pontelli, E. (2020). EFP 2.0: A multi-agent epistemic solver with multiple e-state representations. In Beck, J. C., Buffet, O., Hoffmann, J., Karpas, E., and Sohrabi, S., editors, *Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling, Nancy, France, October 26-30, 2020*, pages 101–109. AAAI Press.

Fabiano, F., Riouak, I., Dovier, A., and Pontelli, E. (2019). Non-well-founded set based multi-agent epistemic action language. In *Proceedings of the 34th Italian Conference on Computational Logic*, volume 2396 of *CEUR Workshop Proceedings*, pages 242–259, Trieste, Italy.

Fagin, R. and Halpern, J. Y. (1994). Reasoning about knowledge and probability. *Journal of the ACM (JACM)*, 41(2):340–367.

Gerbrandy, J. (1999). *Bisimulations on planet Kripke.* Inst. for Logic, Language and Computation, Univ. van Amsterdam.

Gerbrandy, J. and Groeneveld, W. (1997). Reasoning about information change. *Journal of Logic, Language and Information*, 6(2):147–169.

Huang, X., Fang, B., Wan, H., and Liu, Y. (2017). A general multi-agent epistemic planner based on higher-order belief change. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 1093–1101.

Kominis, F. and Geffner, H. (2015). Beliefs in multiagent planning: From one agent to many. In *Proceedings of the International Conference on Automated Planning and Scheduling, ICAPS*, pages 147–155.

Kominis, F. and Geffner, H. (2017). Multiagent online planning with nested beliefs and dialogue. In *Proceedings of the International Conference on Automated Planning and Scheduling, ICAPS*, pages 186–194, Pittsburgh, Pennsylvania, USA.

Le, T., Fabiano, F., Son, T. C., and Pontelli, E. (2018). EFP and PG-EFP: Epistemic forward search planners in multi-agent domains. In *Proceedings of the Twenty-Eighth International Conference on Automated Planning and Scheduling*, pages 161–170, Delft, The Netherlands. AAAI Press.

Muise, C. J., Belle, V., Felli, P., McIlraith, S. A., Miller, T., Pearce, A. R., and Sonenberg, L. (2015). Planning over multi-agent epistemic states: A classical planning approach. In *Proc. of AAAI*, pages 3327–3334.

Son, T. C., Pontelli, E., Baral, C., and Gelfond, G. (2014). Finitary s5-theories. In *European Workshop on Logics in Artificial Intelligence*, pages 239–252. Springer.