Appendix for the Paper "State Transition in Multi-agent Epistemic Domains using Answer Set Programming"

Yusuf Izmirlioglu, Loc Pham, Tran Cao Son, and Enrico Pontelli

New Mexico State University, Las Cruces NM, USA,
yizmir,locpham@nmsu.edu, tson,epontell@cs.nmsu.edu

1 State Transition in the Example Scenarios

In this section, we give the necessary background information about the state transition for the example scenarios in the introduction of the main text. We follow the notation and terminology in the main text. We first explain the state transition function in Example 1, 2. To explain the event update model based semantics in Example 1, 2, we use the definitions in [1, 2] as below.

Let $D = \langle \mathcal{AG}, \mathcal{F} \rangle$ be a multi-agent domain, $\mathcal{L}$ be the set of fluent literals and $\mathcal{L}_{\mathcal{AG}}$ denote the set of belief formulae over $\langle \mathcal{AG}, \mathcal{F} \rangle$.

Definition 1 (Event Update Model). *An* event update model $\mathbf{\Sigma}$ *is a tuple* $\langle \Sigma, R_1, \ldots, R_n, \text{pre}, \text{eff} \rangle$ *where*

(i) Σ *is a set of* events*;*
(ii) each R_i *is a binary relation on* Σ*;*
(iii) $\text{pre} : \Sigma \to \mathcal{L}_{\mathcal{AG}}$ *is a function mapping each event* $e \in \Sigma$ *to a formula in* $\mathcal{L}_{\mathcal{AG}}$*; and*
(iv) $\text{eff} : \Sigma \to 2^{\mathcal{L}}$ *is a function mapping each event* $e \in \Sigma$ *to a set of fluent literals* $\varphi \subseteq \mathcal{L}$*.*

An event update instance ω *is a pair* $(\mathbf{\Sigma}, e)$ *where* $\mathbf{\Sigma}$ *is an event update model as defined above, and* $e \in \Sigma$ *is the designated or the actual event.*

Intuitively, an update model represents different views of an action occurrence depending on the observability of agents. Each view is represented by an event in Σ. The designated event is the one that agents who are aware of the action occurrence will observe. The relation R_i describes agent i's uncertainty on action execution—i.e., if $(\sigma, \tau) \in R_i$ and event σ is performed, then agent i may believe that event τ is executed instead. *pre* defines the precondition of the event and *eff* specifies the effect of the event.

Definition 2 (Updates by an Update Model). *Let* M *be a Kripke structure and* $\mathbf{\Sigma} = \langle \Sigma, R_1, \ldots, R_n, \text{pre}, \text{eff} \rangle$ *be an event update model. The* update *induced by* $\mathbf{\Sigma}$ *defines a Kripke structure* $M' = M \otimes \mathbf{\Sigma}$*, where:*

(i) $M'[S] = \{(u, \kappa) \mid u \in M[S], \kappa \in \Sigma, (M, u) \vDash \text{pre}(\kappa)\}$*;*
(ii) $((u, \kappa), (v, \rho)) \in M'[i]$ *iff* $(u, \kappa), (v, \rho) \in M'[S]$*,* $(u, v) \in M[i]$ *and* $(\kappa, \rho) \in R_i$*;*
(iii) *For all* $(u, \kappa) \in M'[S]$ *and* $\ell \in \mathcal{L}$*,* $M'[\pi]((u, \kappa)) \vDash \ell$ *iff* $(\ell \in \textit{eff}(\kappa))$ *or* $(M[\pi]((u)) \vDash \ell) \wedge (\bar{\ell} \notin \textit{eff}(\kappa))$*.*

The new Kripke structure M' is obtained from the component-wise cross-product of the old structure M and the event update model $\mathbf{\Sigma}$ for those worlds which satisfy precondition of the event. Valuation function of the worlds in the new structure is obtained by applying the effect of the event to the corresponding world in the original structure.

Definition 3 (State Update). *Let* (M, s) *be the initial state and* $(\mathbf{\Sigma}, e)$ *be an event update instance. The next state is the Kripke structure* (M', s') *where* $M' = M \otimes \mathbf{\Sigma}$ *and the new designated world is* $s' = (s, e)$*.*

Example 1. The initial Kripke structure in the first example, as depicted in Figure 1(a) is $(M, s) = \langle S, \pi, \mathcal{B}_A, \mathcal{B}_B \rangle$ where $S = \{s, u, v\}$, $\pi(s) = \{normal,\ sound\}$, $\pi(u) = \{normal,\ \neg sound\}$, $\pi(v) = \{\neg normal,\ \neg sound\}$, $\mathcal{B}_A = \{(s, u),\ (s, v)\}$, $\mathcal{B}_B = \{(s, s)\}$. The actual world is s.

Agent B takes the *check_voltage* action which senses the voltage level. The precondition of the action is $\psi = sound$ and the condition for full observability is $\top$. According to [2, 4], the event update instance of the *check_voltage* action for a sensing/announcement action is shown in Figure 1(a) in the main text. The σ event corresponds to sensed value being *normal* and the τ event corresponds to $\neg normal$. Formally the event update instance is $(\mathbf{\Sigma}, \sigma)$ where $\mathbf{\Sigma} = \langle \Sigma, R_A, R_B, pre, eff \rangle$, $\Sigma = \{\sigma,\ \tau\}$, $R_A = R_B = \{(\sigma, \sigma),\ (\tau, \tau)\}$, $pre(\sigma) = sound \wedge normal$, $pre(\tau) = sound \wedge \neg normal$, $eff(\sigma) = eff(\tau) = \emptyset$. The true event is σ.

The next state is (M', s') where $M' = M \otimes \mathbf{\Sigma}$ and $s' = (s, \sigma)$. Note that s satisfies precondition of σ event; but u, v do not satisfy precondition of σ or τ event. This is because u and v do not satisfy the precondition of the action. Hence the set of worlds at the next state is $M'[S] = \{(s, \sigma)\}$. The accessibility relations at the next state are $M'[A] = \emptyset$ and $M'[B] = \{((s, \sigma), (s, \sigma))\}$. The next state is shown in bottom-right of Figure 1(a) in the main text. Since u and v do not satisfy action precondition, the accessibility relation of agent A from s to u and s to v are removed. Therefore, at the next state agent A does not have any accessibility relation at the actual world (s, σ) and believes in every belief formula.

Example 2. In the second example, the initial state is $(M, s) = \langle S, \pi, \mathcal{B}_A, \mathcal{B}_B \rangle$ where $S = \{s, u, v\}$ where $\pi(s) = \{normal,\ sound\}$, $\pi(u) = \{\neg normal,\ sound\}$, $\mathcal{B}_A = \{(s, u),\ (u, u)\}$, $\mathcal{B}_B = \{(s, s),\ (u, u)\}$. The actual world is s.

Agent B performs the *check_voltage* action. Its precondition is $\psi = sound$ and the condition for full observability is $\top$. The initial state and the event update instance of the *check_voltage* action are shown in Figure 1(b) in the main text. The event update instance is $(\mathbf{\Sigma}, \sigma)$ is the same as in Example 1. Namely, $\mathbf{\Sigma} = \langle \Sigma, R_A, R_B, pre, eff \rangle$ where $\Sigma = \{\sigma,\ \tau\}$, $R_A = R_B = \{(\sigma, \sigma), (\tau, \tau)\}$, $pre(\sigma) = sound \wedge normal$, $pre(\tau) = sound \wedge \neg normal$, $eff(\sigma) = eff(\tau) = \emptyset$. The true event is σ.

The next state is (M', s') where $M' = M \otimes \mathbf{\Sigma}$ and $s' = (s, \sigma)$. Note that $(M, s) \vDash pre(\sigma)$, $(M, u) \vDash pre(\tau)$, so the set of worlds at the next state is $M'[S] = \{(s, \sigma),\ (u, \tau)\}$. The accessibility relations at the next state are $M'[A] = \emptyset$ and $M'[B] = \{((s, \sigma), (s, \sigma))\}$. $(s, s) \notin M[A]$ hence $((s, \sigma), (s, \sigma)) \notin M'[A]$. Besides, $((s, \sigma), (u, \tau)) \notin M'[A]$ since $(\sigma, \tau) \notin R_A$. Then, at the next state, agent A does not have any accessibility relation at the actual world (s, σ) and believes in every belief formula.

1.1 State Transition Function of [3]

[3] has constructed a model of state transition with two different operators $\mathbf{B}_i$, $\mathbf{K}_i$ for belief and knowledge, respectively. They assume that $\mathbf{B}_i$ satisfies $KD45_n$ and $\mathbf{K}_i$ satisfies $S5_n$ properties. In addition they assume both $\mathbf{B}_i$, $\mathbf{K}_i$ satisfy *KB1* and *KB2* properties. In their setting, a Kripke structure is defined as a tuple $\langle S, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n, \mathcal{K}_1, \ldots, \mathcal{K}_n \rangle$ and contains two different accessibility relations for each agent corresponding to the belief and knowledge operators. A pointed state is (M, s) where M is a Kripke structure as defined above and s is the designated (actual) world. Different from our setup, ontic actions are unconditional in [3]. Let ψ be the precondition and φ be the effect of an ontic action a. For a world $u \in M[S]$ which satisfies the precondition of a (i.e. $(M, u) \vDash \psi$), the new world $u^+ \in M'[S]$ is created at the next state by applying the effect of the action. The valuation of the new world is $M'[\pi](u^+) = (M[\pi](u) \setminus \overline{\varphi}) \cup \varphi$.

[3] defines full observer, partial observer and oblivious agents for ontic, sensing, announcement actions. For ontic actions partial observability is the same as full observability. Observability of the agents is computed at the actual world s and is fixed across all worlds. Hence for an ontic action a, the set F, O of full observer and oblivious agents are

$$F = \{i \in \mathcal{AG} \mid (M, s) \vDash \delta_{i,\mathsf{a}}\},$$

$$O = \{i \in \mathcal{AG} \mid (M, s) \nvDash \delta_{i,\mathsf{a}}\}.$$

In the setting of [3], full observer agents correct their beliefs about precondition of the action, but not about observability. To construct the next state, they define the temporary relations $\mathcal{R}_i$ representing agents beliefs after they have possibly been reset (updated with information from $\mathbf{K}_i$) due to observing an action whose precondition contradicts the agents belief. For an agent $i \in \mathcal{AG}$,

$$\mathcal{R}_i = \{(u, v) \mid (u, v) \in \mathcal{B}_i \ \vee \ ((u, v) \in \mathcal{K}_i \wedge (M, u) \vDash \mathbf{B}_i \neg\psi)\}$$

The accessibility relations in the next state (M', s') are computed as follows. Full observers correct their beliefs about action precondition and observe the effect of the action. Namely, the belief relation of a full observer agent i in M' is

$$\mathcal{B}'_i = \mathcal{B}_i \ \cup \ \{(u^+, v^+) \mid (M, u) \vDash \psi, \ (M, v) \vDash \psi, \ (u, v) \in \mathcal{R}_i\}$$

The accessibility relation for the knowledge operator of a full observer agent i at the next state becomes

$$\mathcal{K}'_i = \mathcal{K}_i \ \cup \ \{(u^+, v^+) \mid (M, u) \vDash \psi, \ (M, v) \vDash \psi, \ (u, v) \in \mathcal{K}_i\}$$

The belief relation of oblivious agents do not change, they continue to consider the old state. However an oblivious agent updates his knowledge relation such that he considers possible that the action might have happened (recall that knowledge relation satisfies $S5_n$ property).

$\mathcal{B}'_i = \mathcal{B}_i \ \cup \ \{(u^+, v) \mid (M, u) \vDash \psi, \ (u, v) \in \mathcal{B}_i\}$
$\mathcal{K}'_i = \mathcal{K}_i \ \cup \ \{(u^+, v^+) \mid (M, u) \vDash \psi, \ (M, v) \vDash \psi, \ (u, v) \in \mathcal{K}_i\} \ \cup \ \{(u^+, v), \ (v, u^+) \mid (M, u) \vDash \psi, \ (u, v) \in \mathcal{K}_i\}$

Example 3. As shown in Figure 2 in the main text, the initial Kripke structure in the third example is $(M, s) = \langle S, \pi, \mathcal{B}_A, \mathcal{B}_B, \mathcal{K}_A, \mathcal{K}_B \rangle$ where

$S = \{s, u, v\}$,
$\pi(s) = \{\neg open, \ near\_a, \ near\_b, \ haskey\_a, \ \neg haskey\_b\}$,
$\pi(u) = \{\neg open, \ near\_a, \ \neg near\_b, \ haskey\_a, \ \neg haskey\_b\}$,
$\mathcal{B}_A = \{(s, s)\}$, $\mathcal{B}_B = \{(s, u)\}$,
$\mathcal{K}_A = \{(s, s), \ (u, u)\}$, $\mathcal{K}_B = \{(s, s), \ (s, u), \ (u, u), \ (u, s)\}$.

The actual world is s. Observability of agents is computed at s and thus all agents are full observers. Namely, $F = \{A, B\}$ and $O = \emptyset$.

Agent A performs the *open_door* action whose precondition is $\psi = haskey\_a$ and its effect is $\varphi = \{open\}$. All worlds satisfy the precondition of the *open_door* action i.e. $(M, s) \vDash \psi$, $(M, u) \vDash \psi$. To compute the next state according to state transition function of [3], we first compute the temporary relation $\mathcal{R}_i$. In this example, the agents do not have false belief about the action precondition at any world, thus the relation $\mathcal{R}_i$ is identical to $\mathcal{B}_i$. Then the accessibility relations in the next state (M', s') are
$\mathcal{B}'_A = \{(s^+, s^+)\}$, $\mathcal{B}'_B = \{(s^+, u^+)\}$,
$\mathcal{K}'_A = \{(s^+, s^+), \ (u^+, u^+)\}$, $\mathcal{K}'_B = \{(s^+, s^+), \ (s^+, u^+), \ (u^+, u^+), \ (u^+, s^+)\}$.

The next state is shown in Figure 2 (right) in the main text. The actual world is $s' = s^+$. The state transition function of [3] handles correcting agents' beliefs about precondition, but not about observability. Actually, in this example correcting beliefs about precondition is not relevant as all agents have correct belief about precondition. However, correction for observability is necessary because the next state computed by [3] is counter-intuitive. At the next state, B believes that the door is open but B believes that he is oblivious i.e. B believes that he is not near to the door. Ideally, since B has observed the effect of the action, B must have realized that he is full observer of the action. Because B wouldn't observe opening the door if he were far from the door. Therefore the next state is not realistic.

2 ASP Rules for The State Transition

Let $D = \langle \mathcal{AG},\ \mathcal{F},\ \mathcal{A}\rangle$ be a consistent multi-agent epistemic domain and $T = (M, s)$ be the initial Kripke structure where s is the actual world. Note that the agents may not know the actual world in M. We develop a state stransition function using ASP for ontic, sensing, announcement actions. The domain description is given in m$\mathcal{A}^*$. In our framework, ontic actions have two types of observers (full observer and oblivious agents); sensing, announcement actions have three types of observers (full observer, partial observer and oblivious agents).

We study the problem of computing the next state $\Phi_D(a, (M, s))$ given an initial state (M, s) and the occurrence of action a. Below we build the ASP program $\Pi_{D,T,a}$ which computes the next state $\Phi_D(a, T)$ given an initial state $T = (M, s)$ and occurrence of an action a.

Input: We represent agents and agent sets by $\mathit{ag}(I)$, $\mathit{ag\_set}(G)$ atoms. $\mathit{formula}(F)$ atom shows the belief formulae (and subformulae) that appear in the domain D. In order to compute entailment at possible worlds, we need to identify the belief formulae corresponding to the given problem instance and domain description. Actions are described by $\mathit{action}(A)$, $\mathit{type}(A, Y)$, $\mathit{exec}(A, F)$, $\mathit{causes}(A, L, F)$, $\mathit{determines}(A, F)$, $\mathit{announces}(A, F)$ atoms. $\mathit{observes}(I, A, F)$ and $\mathit{aware}(I, A, F)$ atoms show the condition for full observability and partial observability of agent I, respectively. *fluent(F)* atoms denote the fluents, *pre_lit(A,F)* atoms denote the literals $h_1, ..., h_r$ in action precondition ψ, *full_lit(I,A,F)* atoms denote the literals in $\delta_{i,\mathsf{a}}$ and *partial_lit(I,A,F)* atoms denote the literals in $\theta_{i,\mathsf{a}}$. Sensing/announcement variables are identified by *varphi(A,F)* atoms.

The worlds, accessibility relations and the valuations at the initial state T are encoded by $\mathit{world}(U)$, $\mathit{access}(I, U, V)$, $\mathit{val}(U, F)$ atoms, respectively, where I denotes an agent, U and V are worlds, and F is a fluent. For efficiency, we state only positive literals in valuation of a world. $\mathit{actual}(S)$ stands for the actual world S. $\mathit{occ}(\mathsf{a})$ atom shows the action a that occurs. The next state is represented by $\mathit{world\_n}(U)$, $\mathit{actual\_n}(Z)$, $\mathit{access\_n}(I, U, V)$, $\mathit{val\_n}(U, F)$.

State Transition: We first identify the set of literals and belief formulae (and subformulae) that appear in the domain D. We show the fluent literals by the $\mathit{literal}(L)$ atoms and the belief formulae by the $\mathit{formula}(F)$ atoms. Given a problem instance, we consider the belief formulae (and subformulae) that appear in the domain description and initial state description i.e. ψ, μ, β, φ, $\delta_{i,\mathsf{a}}$, $\theta_{i,\mathsf{a}}$.

$$\mathit{literal}(F) \leftarrow \mathit{fluent}(F). \tag{1}$$
$$\mathit{literal}(\neg F) \leftarrow \mathit{fluent}(F). \tag{2}$$
$$\mathit{formula}(L) \leftarrow \mathit{literal}(L). \tag{3}$$

$$formula(\top). \tag{4}$$
$$formula(\bot). \tag{5}$$
$$formula(F) \leftarrow exec(A,F). \tag{6}$$
$$formula(F) \leftarrow observes(I,A,F). \tag{7}$$
$$formula(F) \leftarrow aware(I,A,F). \tag{8}$$
$$formula(F) \leftarrow causes(A,E,F). \tag{9}$$
$$formula(F) \leftarrow causes(A,E,F). \tag{10}$$
$$formula(F) \leftarrow goal(F). \tag{11}$$
$$formula(F) \leftarrow val(U,F). \tag{12}$$
$$formula(F) \leftarrow determines(A,F). \tag{13}$$
$$formula(F) \leftarrow announces(A,F). \tag{14}$$
$$formula(F_1) \leftarrow formula(F_1 \wedge F_2). \tag{15}$$
$$formula(F_2) \leftarrow formula(F_1 \wedge F_2). \tag{16}$$
$$formula(F_1) \leftarrow formula(F_1 \vee F_2). \tag{17}$$
$$formula(F_2) \leftarrow formula(F_1 \vee F_2). \tag{18}$$
$$formula(F) \leftarrow formula(\neg F). \tag{19}$$
$$formula(F) \leftarrow formula(B_I F). \tag{20}$$
$$formula(F) \leftarrow formula(C_G F). \tag{21}$$
$$formula(F) \leftarrow formula\_full(I,A,F). \tag{22}$$
$$formula(F) \leftarrow formula\_partial(I,A,F). \tag{23}$$

varphi(A,F) atoms denote the fluents corresponding to the sensing/announcement variables.

$$varphi(A,F) \leftarrow determines(A,F),\ occ(A). \tag{24}$$
$$varphi(A,F) \leftarrow announces(A,F),\ occ(A). \tag{25}$$

We compute entailment of belief formulae at the initial state T. *entails(U,F)* atom denotes that the world $U \in M[S]$ satisfies the belief formula F. The ASP rules that compute entailment of belief formula are:

$$entails(U,\top) \leftarrow world(U). \tag{26}$$
$$not\_entails(U,\bot) \leftarrow world(U). \tag{27}$$
$$entails(U,F) \leftarrow world(U),\ val(U,F),\ fluent(F). \tag{28}$$
$$entails(U,\neg F) \leftarrow world(U),\ not\ val(U,F),\ fluent(F). \tag{29}$$
$$entails(U,F1 \wedge F2) \leftarrow world(U),\ entails(U,F1),\ entails(U,F2),\ formula(F1 \wedge F2). \tag{30}$$
$$entails(U,F1 \vee F2) \leftarrow world(U),\ entails(U,F1),\ formula(F1 \vee F2). \tag{31}$$
$$entails(U,F1 \vee F2) \leftarrow world(U),\ entails(U,F2),\ formula(F1 \vee F2). \tag{32}$$
$$entails(S,U,\neg F) \leftarrow world(U),\ not\ entails(U,F),\ formula(\neg F). \tag{33}$$
$$\neg entails(U,B_I F) \leftarrow world(U),\ access(I,U,V),\ not\ entails(V,F),\ formula(B_I F). \tag{34}$$
$$entails(U,B_I F) \leftarrow not\ \neg entails(U,B_I F),\ world(U),\ formula(B_I F). \tag{35}$$

$$reach(G,U,V) \leftarrow access(I,U,V),\ ag\_set(G),\ contains(G,I),\ formula(C_G F). \tag{36}$$
$$reach(G,U,W) \leftarrow reach(G,U,V),\ access(I,V,W),\ ag\_set(G),\ contains(G,I),\ formula(C_G F). \tag{37}$$
$$\neg entails(U,C_G F) \leftarrow world(U),\ not\ entails(U,F),\ formula(C_G F). \tag{38}$$
$$\neg entails(U,C_G F) \leftarrow world(U),\ reach(G,U,V),\ not\ entails(V,F),\ formula(C_G F). \tag{39}$$
$$entails(U,C_G F) \leftarrow world(U),\ not\ \neg entails(U,C_G F),\ formula(C_G F). \tag{40}$$

Rule (34) states that the belief formula $B_I F$ is not entailed at world U if there is a world V (that agent I considers at U) and V does not satisfy F. If there is no such case, U entails $B_I F$ by the rule (35).

Then we compute observability of the agents at each world by

$$\textit{f\_obs(I,A,U)} \leftarrow \textit{observes(I,A,F), entails(U,F), world(U), occ(A).} \tag{41}$$

$$\textit{p\_obs(I,A,U)} \leftarrow \textit{aware(I,A,F), entails(U,F), world(U), occ(A).} \tag{42}$$

$$\textit{obliv(I,A,U)} \leftarrow \textit{not f\_obs(I,A,U), not p\_obs(I,A,U), world(U), ag(I), occ(A).} \tag{43}$$

The rule below checks whether the action a is executable i.e. the precondition of the action holds at the actual world (M, s). In this case, s' is the actual world at the next state.

$$\textit{pre\_hold(S)} \leftarrow \textit{actual(S), entails(S,F), exec(A,F), occ(A).} \tag{44}$$

$$\textit{actual\_n}(S') \leftarrow \textit{actual(S), pre\_hold(S), occ(A).} \tag{45}$$

We identify the possible worlds in the next state M' by the rules below. If the precondition of the action a holds at (M, s), then s' is a possible world at the next state. The worlds that are reachable from s' are also possible worlds in M'.

$$\textit{world\_n}(S') \leftarrow \textit{actual(S), pre\_hold(S), occ(A).} \tag{46}$$

$$\textit{world\_n(V)} \leftarrow \textit{actual\_n(Z), access\_n(I,Z,V).} \tag{47}$$

$$\textit{world\_n(V)} \leftarrow \textit{world\_n(U), access\_n(I,U,V).} \tag{48}$$

We construct the accessibility relations of full observers in the next state M' for an ontic action as below. Full observers correct their beliefs about action precondition and observability and observe the effect of the action. Suppose that $(M, U) \vDash \delta_{i,\mathsf{a}}$ and $(U, V) \in M[i]$. In the next state, we keep only the accessibility relations of agent i from U to the worlds V which satisfy action precondition and observability of i. In this case we apply the effect of the action to the world V, obtain $V' \in M'[S]$ and create the accessibility relation $(U', V') \in M'[i]$. However, if all the V worlds that agent i considers possible at U violate precondition and/or observability (indicated by the *ontic_cond(i,U)* atom), we cannot remove all the edges, thus we amend the worlds to obtain V_i and create relations from U' to V_i.

We first generate outgoing accessibility relations at the actual world s and then recursively generate for other worlds.

$$\textit{formula\_full}(I, A, F1 \land F2) \leftarrow \textit{exec(A,F1), observes(I,A,F2), ag(I).} \tag{49}$$

$$\neg\textit{ontic\_cond(I,U)} \leftarrow \textit{access(I,U,V), entails(V,F), formula\_full(I,A,F), occ(A), type(A,ontic).} \tag{50}$$

$$\begin{aligned}\textit{access\_n}(I, S', U') \leftarrow\ & \textit{actual(S), pre\_hold(S), access(I,S,U), f\_obs(I,A,S),}\\ & \textit{entails(U,F), formula\_full(I,A,F), occ(A), type(A,ontic).}\end{aligned} \tag{51}$$

$$\begin{aligned}\textit{access\_n}(I, S', U_I) \leftarrow\ & \textit{actual(S), pre\_hold(S), access(I,S,U), f\_obs(I,A,S),}\\ & \textit{not } \neg\textit{ontic\_cond(I,S), occ(A), type(A,ontic).}\end{aligned} \tag{52}$$

$$\begin{aligned}\textit{access\_n}(I, U', V') \leftarrow\ & \textit{world\_n}(U'), \textit{access(I,U,V), f\_obs(I,A,U),}\\ & \textit{entails(V,F), formula\_full(I,A,F), occ(A), type(A,ontic).}\end{aligned} \tag{53}$$

$$\begin{aligned}\textit{access\_n}(I, U', V_I) \leftarrow\ & \textit{world\_n}(U'), \textit{access(I,U,V), f\_obs(I,A,U),}\\ & \textit{not } \neg\textit{ontic\_cond(I,U), occ(A), type(A,ontic).}\end{aligned} \tag{54}$$

Note that if there is a world V that agent i considers possible at U and if V satisfies action precondition and observability, then $\neg$*ontic_cond(i,U)* atom is generated by (50). In this case, we do not amend the other V worlds that violate precondition and/or observability.

Note that agent i should also have beliefs at the world U_i in the next state. Thus we need to construct outgoing accessibility relations from the world $U_i \in M'[S]$, for each agent $j \in \mathcal{AG}$. The intuition is similar. First suppose that $j \neq i$ and $(U, V) \in M[j]$. If the world V satisfies action precondition and observability of j, then agent j will have an accessibility relation from U_i to V' at the next state. However, if all the V worlds that agent j considers possible at U violate precondition and/or observability (indicated by the *ontic_cond(j,U)* atom), we cannot remove all the edges, thus we amend the worlds to obtain V_j and create relations from U_i to V_j.

$$\begin{aligned}
&\textit{access_n}(J,U_I,V') \leftarrow J \neq I,\ \textit{world_n}(U_I),\ \textit{access(J,U,V)},\ \textit{f_obs(J,A,U)},\\
&\qquad \textit{entails(V,F)},\ \textit{formula_full(J,A,F)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. && (55)\\
&\textit{access_n}(J,U_I,V_J) \leftarrow J \neq I,\ \textit{world_n}(U_I),\ \textit{access(J,U,V)},\ \textit{f_obs(J,A,U)},\\
&\qquad \textit{not}\ \neg\textit{ontic_cond(J,U)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. && (56)
\end{aligned}$$

In the second case suppose that $j = i$ i.e. we construct accessibility relations of agent i at $U_i \in M'[S]$. Agent i should observe the effect of the action at the world U_i, regardless of his observability at $U \in M[S]$. At the next state, i keeps links from U_i to those V worlds which satisfy precondition and observability and removes links to V worlds which do not satisfy. However if all those V worlds that i considers possible at $U \in M[S]$ violate precondition and/or observability, agent i will create links from U_i to amended V_i worlds.

$$\begin{aligned}
&\textit{access_n}(I,U_I,V') \leftarrow \textit{world_n}(U_I),\ \textit{access(I,U,V)},\ \textit{entails(V,F)},\ \textit{formula_full(I,A,F)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. && (57)\\
&\textit{access_n}(I,U_I,V_I) \leftarrow \textit{world_n}(U_I),\ \textit{access(I,U,V)},\ \textit{not}\ \neg\textit{ontic_cond(I,U)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. && (58)
\end{aligned}$$

Oblivious agents remain at the old state and their beliefs do not change. We keep former accessibility relations of oblivious agents in M so that their beliefs remain the same. Namely accessibility relations of oblivious agents in the next state M' for an ontic action are constructed as

$$\begin{aligned}
&\textit{access_n(I,U,V)} \leftarrow \textit{world_n(U)},\ \textit{access(I,U,V)},\ \textit{occ(A)}. && (59)\\
&\textit{access_n}(I,U',V) \leftarrow \textit{world_n(U')},\ \textit{access(I,U,V)},\ \textit{obliv(I,A,U)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. && (60)\\
&\textit{access_n}(I,U_J,V) \leftarrow I \neq J,\ \textit{world_n}(U_J),\ \textit{access(I,U,V)},\ \textit{obliv(I,A,U)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. && (61)
\end{aligned}$$

For sensing/announcement actions, we need to check whether sensing/announcement variables are the same across two worlds $U, V \in M[S]$. *var_diff(U,V)* indicates that at least one variable differs across U and V.

$$\begin{aligned}
&\textit{var_diff(U,V)} \leftarrow \textit{access(I,U,V)},\ \textit{val(U,F)},\ \textit{not val(V,F)},\ \textit{varphi(A,F)},\ \textit{occ(A)}. && (62)\\
&\textit{var_diff(U,V)} \leftarrow \textit{access(I,U,V)},\ \textit{not val(U,F)},\ \textit{val(V,F)},\ \textit{varphi(A,F)},\ \textit{occ(A)}. && (63)\\
&\textit{var_diff(W,V)} \leftarrow \textit{world_n}(U^f_{I,W}),\ \textit{access(J,U,V)},\ \textit{val(W,F)},\ \textit{not val(V,F)},\ \textit{varphi(A,F)},\ \textit{occ(A)}. && (64)\\
&\textit{var_diff(W,V)} \leftarrow \textit{world_n}(U^f_{I,W}),\ \textit{access(J,U,V)},\ \textit{not val(W,F)},\ \textit{val(V,F)},\ \textit{varphi(A,F)},\ \textit{occ(A)}. && (65)
\end{aligned}$$

Now we create accessibility relations in the next state for a sensing/announcement action. We first consider full observers. Suppose that $(M,U) \vDash \delta_{i,\mathsf{a}}$ and $(U,V) \in M[i]$. In the next state agent i keeps links to those V worlds which satisfy precondition, observability of i and whose value of sensing/announcement variables are the same as U; and removes links to V worlds which do not satisfy such conditions. If all V worlds that agent i considers possible at U, violate precondition and/or observability and/or value of sensing/announcement variables (indicated by the *sa_f_cond(i,U)* atom), then i will amend all these V worlds and create link to amended $V^f_{i,U}$ worlds. The value of sensing/announcement variables at $V^f_{i,U}$ are taken from the world U due to belief correction.

We first generate outgoing accessibility relations at the actual world s and then recursively generate for other worlds.

$$\begin{aligned}
&\neg \mathit{sa_f_cond(I,U)} \leftarrow \mathit{access(I,U,V)},\ \mathit{entails}(V,F),\ \mathit{formula_full(I,A,F)},\\
&\quad \mathit{not\ var_diff(U,V)},\ \mathit{occ}(A),\ \mathit{type(A,sa)}. && (66)\\
&\mathit{access_n}(I,S',U') \leftarrow \mathit{actual(S)},\ \mathit{pre_hold(S)},\ \mathit{access(I,S,U)},\ \mathit{f_obs(I,A,S)},\ \mathit{entails(U,F)},\\
&\quad \mathit{formula_full(I,A,F)},\ \mathit{not\ var_diff(S,U)},\ \mathit{occ}(A),\ \mathit{type(A,sa)}. && (67)\\
&\mathit{access_n}(I,S',U^f_{I,S}) \leftarrow \mathit{actual(S)},\ \mathit{pre_hold(S)},\ \mathit{access(I,S,U)},\ \mathit{f_obs(I,A,S)},\\
&\quad \mathit{not}\ \neg \mathit{sa_f_cond(I,S)},\ \mathit{occ}(A),\ \mathit{type(A,sa)}. && (68)\\
&\mathit{access_n}(I,U',V') \leftarrow \mathit{world_n}(U'),\ \mathit{access(I,U,V)},\ \mathit{f_obs(I,A,U)},\ \mathit{entails(V,F)},\\
&\quad \mathit{formula_full(I,A,F)},\ \mathit{not\ var_diff(U,V)},\ \mathit{occ}(A),\ \mathit{type(A,sa)}. && (69)\\
&\mathit{access_n}(I,U',V^f_{I,U}) \leftarrow \mathit{world_n}(U'),\ \mathit{access(I,U,V)},\ \mathit{f_obs(I,A,U)},\\
&\quad \mathit{not}\ \neg \mathit{sa_f_cond(I,U)},\ \mathit{occ}(A),\ \mathit{type(A,sa)}. && (70)
\end{aligned}$$

Note that if there is a world V (that agent i considers possible at U) and if V satisfies action precondition, full observability of i and has the same value of sensing/announcement variables as U, then $\neg$*sa_f_cond(i,U)* atom is generated by (66). In this case, we do not amend the other V worlds that violate such conditions.

Partial observers correct for only the precondition and observability, but not for the sensing/announcement variables. Suppose that $(M,U) \vDash \theta_{i,\mathsf{a}}$ and $(U,V) \in M[i]$. In the next state, agent i keeps links to those V worlds which satisfy precondition and observability of i; and remove links to V worlds which do not satisfy precondition and observability. However, if all V worlds that agent i considers possible at U violate precondition and/or observability (indicated by the *sa_p_cond(i,U)* atom), then i will amend all these V worlds and create links to amended V^p_i worlds.

$$\begin{aligned}
&\mathit{formula_partial}(I,A,F1 \wedge F2) \leftarrow \mathit{exec(A,F1)},\ \mathit{aware(I,A,F2)},\ \mathit{ag(I)}. && (71)\\
&\neg \mathit{sa_p_cond(I,U)} \leftarrow \mathit{access(I,U,V)},\ \mathit{entails(V,F)},\ \mathit{formula_partial(I,A,F)},\ \mathit{occ}(A),\ \mathit{type(A,sa)}. && (72)\\
&\mathit{access_n}(I,S',U') \leftarrow \mathit{actual(S)},\ \mathit{pre_hold(S)},\ \mathit{access(I,S,U)},\ \mathit{p_obs(I,A,S)},\ \mathit{entails(U,F)},\\
&\quad \mathit{formula_partial(I,A,F)},\ \mathit{occ}(A),\ \mathit{type(A,sa)}. && (73)\\
&\mathit{access_n}(I,S',U^p_I) \leftarrow \mathit{actual(S)},\ \mathit{pre_hold(S)},\ \mathit{access(I,S,U)},\ \mathit{p_obs(I,A,S)},\\
&\quad \mathit{not}\ \neg \mathit{sa_p_cond(I,S)},\ \mathit{occ}(A),\ \mathit{type(A,sa)}. && (74)\\
&\mathit{access_n}(I,U',V') \leftarrow \mathit{world_n}(U'),\ \mathit{access(I,U,V)},\ \mathit{p_obs(I,A,U)},\ \mathit{entails(V,F)},\\
&\quad \mathit{formula_partial(I,A,F)},\ \mathit{occ}(A),\ \mathit{type(A,sa)}. && (75)\\
&\mathit{access_n}(I,U',V^p_I) \leftarrow \mathit{world_n}(U'),\ \mathit{access(I,U,V)},\ \mathit{p_obs(I,A,U)},\\
&\quad \mathit{not}\ \neg \mathit{sa_p_cond(I,U)},\ \mathit{occ}(A),\ \mathit{type(A,sa)}. && (76)
\end{aligned}$$

If there is a world V (that agent i considers possible at U) and if V satisfies action precondition and partial observability of i, then $\neg$*sa_p_cond(i,U)* atom is generated by (72). In this case, we do not amend the other V worlds that violate precondition and/or partial observability.

Note that agent i should also have beliefs at the world $U^f_{i,W}$ and U^p_i in the next state. Thus we need to construct outgoing accessibility relations from the world $U^f_{i,W}$ and U^p_i, for each agent $j \in \mathcal{AG}$. First consider the case $j \neq i$ and $(U,V) \in M[j]$. Accessibility relations of agent j at $U^f_{i,W}$ and U^p_i are created depending on whether j is a full observer or partial observer at world $U \in M[S]$ using similar intuition as above. Here the *sa_f_cond(j,U,W)* atom indicates that all the worlds that agent j considers possible at U violate action precondition and/or full observability of j and/or value of a sensing/announcement variable is different from world $W \in M[S]$.

$$\neg sa\_f\_cond(J,U,W) \leftarrow J \neq I,\ world\_n(U^f_{I,W}),\ access(J,U,V),\ entails(V,F),\ formula\_full(J,A,F),\ not\ var\_diff(W,V),\ occ(A),\ type(A,sa). \quad (77)$$

$$access\_n(J, U^f_{I,W}, V') \leftarrow J \neq I,\ world\_n(U^f_{I,W}),\ access(J,U,V),\ f\_obs(J,A,U),\ entails(V,F),\ formula\_full(J,A,F),\ not\ var\_diff(W,V),\ occ(A),\ type(A,sa). \quad (78)$$

$$access\_n(J, U^f_{I,W}, V^f_{J,W}) \leftarrow J \neq I,\ world\_n(U^f_{I,W}),\ access(J,U,V),\ f\_obs(J,A,U),\ not\ \neg sa\_f\_cond(J,U,W),\ occ(A),\ type(A,sa). \quad (79)$$

$$access\_n(J, U^f_{I,W}, V') \leftarrow J \neq I,\ world\_n(U^f_{I,W}),\ access(J,U,V),\ p\_obs(J,A,U),\ entails(V,F),\ formula\_partial(J,A,F),\ occ(A),\ type(A,sa). \quad (80)$$

$$access\_n(J, U^f_{I,W}, V^p_J) \leftarrow J \neq I,\ world\_n(U^f_{I,W}),\ access(J,U,V),\ p\_obs(J,A,U),\ not\ \neg sa\_p\_cond(J,U),\ occ(A),\ type(A,sa). \quad (81)$$

$$access\_n(J, U^p_I, V') \leftarrow J \neq I,\ world\_n(U^p_I),\ access(J,U,V),\ f\_obs(J,A,U),\ entails(V,F),\ formula\_full(J,A,F),\ not\ var\_diff(U,V),\ occ(A),\ type(A,sa). \quad (82)$$

$$access\_n(J, U^p_I, V^f_{J,U}) \leftarrow J \neq I,\ world\_n(U^p_I),\ access(J,U,V),\ f\_obs(J,A,U),\ not\ \neg sa\_f\_cond(J,U),\ occ(A),\ type(A,sa). \quad (83)$$

$$access\_n(J, U^p_I, V') \leftarrow J \neq I,\ world\_n(U^p_I),\ access(J,U,V),\ p\_obs(J,A,U),\ entails(V,F),\ formula\_partial(J,A,F),\ occ(A),\ type(A,sa). \quad (84)$$

$$access\_n(J, U^p_I, V^p_J) \leftarrow J \neq I,\ world\_n(U^p_I),\ access(J,U,V),\ p\_obs(J,A,U),\ not\ \neg sa\_p\_cond(J,U),\ occ(A),\ type(A,sa). \quad (85)$$

Consider the second case $j=i$ i.e. we construct accessibility relations of agent i at worlds $U^f_{i,W}$ and U^p_i of the next state. Agent i should behave as a full observer at $U^f_{i,W}$ and behave as a partial observer at U^p_i, regardless of his observability at $U \in M[S]$.

$$access\_n(I, U^f_{I,W}, V') \leftarrow world\_n(U^f_{I,W}),\ access(I,U,V),\ entails(V,F),\ formula\_full(I,A,F),\ not\ var\_diff(W,V),\ occ(A),\ type(A,sa). \quad (86)$$

$$access\_n(I, U^f_{I,W}, V^f_{I,W}) \leftarrow world\_n(U^f_{I,W}),\ access(I,U,V),\ not\ \neg sa\_f\_cond(I,U,W),\ occ(A),\ type(A,sa). \quad (87)$$

$$access\_n(I, U^p_I, V') \leftarrow world\_n(U^p_I),\ access(I,U,V),\ entails(V,F),\ formula\_partial(I,A,F),\ occ(A),\ type(A,sa). \quad (88)$$

$$access\_n(I, U^p_I, V^p_I) \leftarrow world\_n(U^p_I),\ access(I,U,V),\ not\ \neg sa\_p\_cond(I,U),\ occ(A),\ type(A,sa). \quad (89)$$

Accessibility relations of oblivious agents are constructed in a similar manner to the ontic actions. Oblivious agents remain at the old state and their beliefs do not change.

$$access\_n(I,U',V) \leftarrow world\_n(U'),\ access(I,U,V),\ obliv(I,A,U),\ occ(A),\ type(A,sa). \quad (90)$$

$$access\_n(I, U^p_J, V) \leftarrow I \neq J,\ world\_n(U^p_J),\ access(I,U,V),\ obliv(I,A,U),\ occ(A),\ type(A,sa). \quad (91)$$

$$access\_n(I, U^f_{J,W}, V) \leftarrow I \neq J,\ world\_n(U^f_{J,W}),\ access(I,U,V),\ obliv(I,A,U),\ occ(A),\ type(A,sa). \quad (92)$$

We also need to compute the valuation function $M'[\pi]$ at the next state M'. We first consider ontic actions. Valuation of old worlds $U \in M[S]$ remain the same in M' so that beliefs of oblivious agents do not change. Valuation of U' worlds are computed by applying the conditional effect of the ontic action. Namely, $M'[\pi](U') = \phi(a, \pi(U))$. If $\pi(U)$ satisfies μ, then the literals in β are placed into the valuation of U to obtain the valuation of U'. *cond(A,F)* atom denotes the μ conditions in the conditional effects ($Effects_A$) of the action A. Note that the μ conditions are disjoint but may not be exhaustive. *one_cond_satis(U)* atom indicates that the world $U \in M[S]$ satisfies one condition μ. If U does not satisfy any μ condition, then the valuation of U' is the same as U.

$$\textit{val\_n(U,F)} \leftarrow \textit{world(U)},\ \textit{world\_n(U)},\ \textit{val(U,F)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{93}$$

$$\textit{one\_cond\_satis(U)} \leftarrow \textit{world(U)},\ \textit{entails(U,F)},\ \textit{cond(A,F)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{94}$$

$$\textit{val\_n}(U', E) \leftarrow \textit{world\_n}(U'),\ \textit{entails(U,F)},\ \textit{causes(A,E,F)},\ \textit{fluent(E)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{95}$$

$$\textit{val\_n}(U', H) \leftarrow \textit{world\_n}(U'),\ \textit{val(U,H)},\ \textit{entails(U,F)},\ \textit{cond}(A, F),\ \textit{not causes}(A, \neg H, F),\ \textit{fluent(H)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{96}$$

$$\textit{val\_n}(U', H) \leftarrow \textit{world\_n}(U'),\ \textit{val(U,H)},\ \textit{not one\_cond\_satis(U)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{97}$$

Now we compute the valuation of U_i worlds in the next state for an ontic action. Note that μ or β for an ontic action may include common fluent(s) with precondition and/or observability formula. For robust state transition, the observing agent should first correct for precondition and observability, and then apply the effect of the action. Let $\lambda(U_i) = (\pi(U) \setminus (\overline{\psi \cup \delta_{i,a}})) \cup (\psi \cup \delta_{i,a})$ be an interpretation such that agent i corrects his beliefs at world $U \in M[S]$ about precondition and his observability. We compute $\lambda(U_i)$ by the rules

$$\textit{lambda}(U_I, H) \leftarrow \textit{world\_n}(U_I),\ \textit{pre\_lit(A,H)},\ \textit{fluent(H)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{98}$$

$$\textit{lambda}(U_I, H) \leftarrow \textit{world\_n}(U_I),\ \textit{full\_lit(i,A,H)},\ \textit{fluent(H)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{99}$$

$$\textit{lambda}(U_I, H) \leftarrow \textit{world\_n}(U_I),\ \textit{val(U,H)},\ \textit{not pre\_lit}(A, \neg H),\ \textit{not full\_lit}(I, A, \neg H),\ \textit{fluent(H)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{100}$$

To realize the effect of the action, we need to determine whether the interpretation $\lambda(U_i)$ satisfies any μ condition among the conditional effects. Recall that the conditions μ are fluent formulae. We parse literals and subformulas of μ by the rules

$$\textit{cond\_formula}(A, F) \leftarrow \textit{causes(A,E,F)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{101}$$

$$\textit{cond\_formula}(A, F1) \leftarrow \textit{cond\_formula}(A, F1 \wedge F2),\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{102}$$

$$\textit{cond\_formula}(A, F2) \leftarrow \textit{cond\_formula}(A, F1 \wedge F2),\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{103}$$

$$\textit{cond\_formula}(A, F1) \leftarrow \textit{cond\_formula}(A, F1 \vee F2),\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{104}$$

$$\textit{cond\_formula}(A, F2) \leftarrow \textit{cond\_formula}(A, F1 \vee F2),\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{105}$$

$$\textit{cond\_formula}(A, F) \leftarrow \textit{cond\_formula}(A, \neg F),\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{106}$$

Then we compute whether the interpretation $\lambda(U_i)$ satisfies a condition μ in $\textit{Effects}_a$. Let $\textit{entails\_lambda}(U_i, F)$ atom denotes that $\lambda(U_i)$ satisfies a formula F.

$$\textit{entails\_lambda}(U_I, H) \leftarrow \textit{world\_n}(U_I),\ \textit{lambda}(U_I, H). \tag{107}$$

$$\textit{entails\_lambda}(U_I, \neg H) \leftarrow \textit{world\_n}(U_I),\ \textit{not lambda}(U_I, H),\ \textit{fluent(H)}. \tag{108}$$

$$\textit{entails\_lambda}(U_I, \neg F) \leftarrow \textit{world\_n}(S', U_I),\ \textit{not entails\_lambda}(U_I, F),\ \textit{cond\_formula}(A, \neg F),\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{109}$$

$$\textit{entails\_lambda}(U_I, F1 \wedge F2) \leftarrow \textit{entails\_lambda}(U_I, F1),\ \textit{entails\_lambda}(U_I, F2),\ \textit{cond\_formula}(A, F1 \wedge F2),\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{110}$$

$$\textit{entails\_lambda}(U_I, F1 \vee F2) \leftarrow \textit{entails\_lambda}(U_I, F1),\ \textit{cond\_formula}(A, F1 \vee F2),\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{111}$$

$$\textit{entails\_lambda}(U_I, F1 \vee F2) \leftarrow \textit{entails\_lambda}(U_I, F2),\ \textit{cond\_formula}(A, F1 \vee F2),\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{112}$$

Valuation of $U_i \in M'[S]$ is computed by $M'[\pi](U_i) = \phi(a, \lambda(U_i))$. Namely, if $\lambda(U_i)$ satisfies μ, then the literals in β are placed into the valuation of U_i. If $\lambda(U_i)$ does not satisfy any μ (denoted by $\textit{one\_cond\_lambda\_satis}(U_i)$ atom), then the valuation of U_i is the same as $\lambda(U_i)$.

$$\textit{one\_cond\_lambda\_satis}(U_I) \leftarrow \textit{entails\_lambda}(U_I, F),\ \textit{cond(A,F)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{113}$$

$$\textit{val\_n}(U_I, E) \leftarrow \textit{entails\_lambda}(U_I, F),\ \textit{causes(A,E,F)},\ \textit{fluent(E)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{114}$$

$$\textit{val\_n}(U_I, H) \leftarrow \textit{lambda}(U_I, H),\ \textit{entails\_lambda}(U_I, F),\ \textit{cond}(A, F),\ \textit{not causes}(A, \neg H, F),\ \textit{fluent(H)},\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{115}$$

$$\textit{val\_n}(U_I, H) \leftarrow \textit{lambda}(U_I, H),\ \textit{not one\_cond\_lambda\_satis}(U_I),\ \textit{occ(A)},\ \textit{type(A,ontic)}. \tag{116}$$

Last, we compute the valuation of worlds at the next state for a sensing/announcement action. The valuation of the world U and U' in M' are the same as valuation of $U \in M[S]$. Valuation of U_i^p and $V_{i,U}^f$ worlds may be different from $\pi(U)$. Recall that U_i^p is created for partial observer agent i where he corrects for action precondition and observability; and $V_{i,U}^f$ is created for full observer agent i where he corrects for precondition, observability and sensing/announcement variables (with respect to $U \in M[S]$).

$$\textit{val\_n(U,F)} \leftarrow \textit{world(U)},\ \textit{world\_n(U)},\ \textit{val(U,F)},\ \textit{occ(A)},\ \textit{type(A,sa)}. \tag{117}$$
$$\textit{val\_n}(U', F) \leftarrow \textit{world(U)},\ \textit{world\_n}(U'),\ \textit{val(U,F)},\ \textit{occ(A)},\ \textit{type(A,sa)}. \tag{118}$$
$$\textit{val\_n}(U_I^p, H) \leftarrow \textit{world\_n}(U_I^p),\ \textit{pre\_lit(A,H)},\ \textit{fluent(H)},\ \textit{occ(A)},\ \textit{type(A,sa)}. \tag{119}$$
$$\textit{val\_n}(U_I^p, H) \leftarrow \textit{world\_n}(U_I^p),\ \textit{partial\_lit(I,A,H)},\ \textit{fluent(H)},\ \textit{occ(A)},\ \textit{type(A,sa)}. \tag{120}$$
$$\begin{aligned}\textit{val\_n}(U_I^p, H) \leftarrow\ & \textit{world\_n}(U_I^p),\ \textit{val(U,H)},\ \textit{not pre\_lit}(A, \neg H),\ \textit{not partial\_lit}(I, A, \neg H),\\ & \textit{fluent(H)},\ \textit{occ(A)},\ \textit{type(A,sa)}.\end{aligned} \tag{121}$$
$$\textit{val\_n}(V_{I,U}^f, H) \leftarrow \textit{world\_n}(V_{I,U}^f),\ \textit{pre\_lit(A,H)},\ \textit{fluent(H)},\ \textit{occ(A)},\ \textit{type(A,sa)}. \tag{122}$$
$$\textit{val\_n}(V_{I,U}^f, H) \leftarrow \textit{world\_n}(V_{I,U}^f),\ \textit{full\_lit(I,A,H)},\ \textit{fluent(H)},\ \textit{occ(A)},\ \textit{type(A,sa)}. \tag{123}$$
$$\textit{val\_n}(V_{I,U}^f, F) \leftarrow \textit{world\_n}(V_{I,U}^f),\ \textit{varphi(A,F)},\ \textit{val(U,F)},\ \textit{occ(A)},\ \textit{type(A,sa)}. \tag{124}$$
$$\begin{aligned}\textit{val\_n}(V_{I,U}^f, h) \leftarrow\ & \textit{world\_n}(V_{I,U}^f),\ \textit{val(V,H)},\ \textit{not pre\_lit}(A, \neg H),\\ & \textit{not full\_lit}(I, A, \neg H),\ \textit{not varphi(A,H)},\ \textit{fluent(H)},\ \textit{occ(A)},\ \textit{type(A,sa)}.\end{aligned} \tag{125}$$

We compute entailment of belief formulae at the next state by adding rules that are analogous to the rules (26)–(40).

$$\textit{entails\_n}(U, \top) \leftarrow \textit{world\_n}(U). \tag{126}$$
$$\neg\textit{entails\_n}(U, \bot) \leftarrow \textit{world\_n}(U). \tag{127}$$
$$\textit{entails\_n(U,F)} \leftarrow \textit{world\_n}(U),\ \textit{val\_n(U,F)},\ \textit{fluent}(F). \tag{128}$$
$$\textit{entails\_n}(U, \neg F) \leftarrow \textit{world\_n}(U),\ \textit{not val\_n(U,F)},\ \textit{fluent}(F). \tag{129}$$
$$\textit{entails\_n}(U, F1 \wedge F2) \leftarrow \textit{world\_n}(U),\ \textit{entails\_n(U,F1)},\ \textit{entails\_n(U,F2)},\ \textit{formula}(F1 \wedge F2). \tag{130}$$
$$\textit{entails\_n}(U, F1 \vee F2) \leftarrow \textit{world\_n}(U),\ \textit{entails\_n(U,F1)},\ \textit{formula}(F1 \vee F2). \tag{131}$$
$$\textit{entails\_n}(U, F1 \vee F2) \leftarrow \textit{world\_n}(U),\ \textit{entails\_n(U,F2)},\ \textit{formula}(F1 \vee F2). \tag{132}$$
$$\textit{entails\_n}(S, U, \neg F) \leftarrow \textit{world\_n}(U),\ \textit{not entails\_n(U,F)},\ \textit{formula}(\neg F). \tag{133}$$
$$\neg\textit{entails\_n}(U, B_I F) \leftarrow \textit{world\_n}(U),\ \textit{access\_n(I,U,V)},\ \textit{not entails\_n(V,F)},\ \textit{formula}(B_I F). \tag{134}$$
$$\textit{entails\_n}(U, B_I F) \leftarrow \textit{not}\ \neg\textit{entails\_n}(U, B_I F),\ \textit{world\_n}(U),\ \textit{formula}(B_I F). \tag{135}$$

$$\textit{reach\_n(G,U,V)} \leftarrow \textit{access\_n(I,U,V)},\ \textit{ag\_set(G)},\ \textit{contains(G,I)},\ \textit{formula}(C_G F). \tag{136}$$
$$\textit{reach\_n(G,U,W)} \leftarrow \textit{reach\_n(G,U,V)},\ \textit{access\_n(I,V,W)},\ \textit{ag\_set(G)},\ \textit{contains(G,I)},\ \textit{formula}(C_G F). \tag{137}$$
$$\neg\textit{entails\_n}(U, C_G F) \leftarrow \textit{world\_n}(U),\ \textit{not entails\_n(U,F)},\ \textit{formula}(C_G F). \tag{138}$$
$$\neg\textit{entails\_n}(U, C_G F) \leftarrow \textit{world\_n}(U),\ \textit{reach\_n(G,U,V)},\ \textit{not entails\_n(V,F)},\ \textit{formula}(C_G F). \tag{139}$$
$$\textit{entails\_n}(U, C_G F) \leftarrow \textit{world\_n}(U),\ \textit{not}\ \neg\textit{entails\_n}(U, C_G F),\ \textit{formula}(C_G F). \tag{140}$$

If the user has specified goal condition(s), we check whether these goal conditions are satisfied at the next state. *achieved(F)* atom denotes that the goal F is satisfied at the next state (M', s'). If all goal conditions are accomplished, then *allachieved* atom is generated.

$$\textit{achieved(F)} \leftarrow \textit{actual\_n(Z)},\ \textit{entails\_n(Z,F)},\ \textit{goal(F)}. \tag{141}$$
$$\neg\textit{all\_achieved} \leftarrow \textit{not achieved(F)},\ \textit{goal(F)}. \tag{142}$$
$$\textit{allachieved} \leftarrow \textit{not}\ \neg\textit{all\_achieved}. \tag{143}$$

3 Proof of The Theorems

In order to establish soundness of our planner, we provide results on our ASP-based state transition in updating the state and beliefs of agents in a robust way. Let $D = \langle \mathcal{AG},\ \mathcal{F},\ \mathcal{A} \rangle$ be a multi-agent epistemic domain and $T = (M, s)$ be the initial state where s is the actual world.

Theorem 1. *The ASP program $\Pi_{D,T,a}$ has an answer set provided that D is a consistent domain.*

Proof of Theorem 1:
The proof is by the splitting sequence theorem. We create the ASP subprograms as below:

$P^0_{M,s}$ is the subprogram that consists of the facts in the input representation,
$P^1_{M,s}$ is the subprogram that consists of the rules (26)-(40),
$P^2_{M,s}$ is the subprogram that consists of the rules (41)-(44),
$P^3_{M,s}$ is the subprogram that consists of the rules (62)-(65),
$P^4_{M,s}$ is the subprogram that consists of the rule (45)-(61) and (66)-(92),
$P^5_{M,s}$ is the subprogram that consists of the rules (98)-(100),
$P^6_{M,s}$ is the subprogram that consists of the rules (101)-(106),
$P^7_{M,s}$ is the subprogram that consists of the rules (107)-(112),
$P^8_{M,s}$ is the subprogram that consists of the rules (93)-(97) and (113)-(125),
$P^9_{M,s}$ is the subprogram that consists of the rules (126)-(140),
$P^{10}_{M,s}$ is the subprogram that consists of the rules (141)-(143).

We define the union of subprograms as $\Pi^k_{M,s} = \cup^k_{i=0} P^i_{M,s}$. We construct the splitting sequence for $\Pi_{M,s} = \Pi^{10}_{M,s}$ as $U = \langle U_0, ..., U_9 \rangle$ where U_i consists of all atoms in the subprograms $P^0_{M,s}$ up to $P^i_{M,s}$. Namely, U_0 is the set of facts in the input representation and U_i includes all atoms in the program $\Pi^i_{M,s}$, $1 \leq i \leq 9$. Observe that the bottom part of $\Pi^{k+1}_{M,s}$ with respect to U_k is $\Pi^k_{M,s}$, and the top part is $P^{k+1}_{M,s}$.

According to the splitting sequence theorem, a solution to $\Pi_{M,s}$ with respect to U is a sequence of literals $\langle X_0, ..., X_{10} \rangle$ such that

- X_0 is an answer set for $b_{U_0}(\Pi_{M,s})$,
- X_{i+1} is an answer set for $e_{U_i}(b_{U_{i+1}}(\Pi_{M,s}) \setminus b_{U_i}(\Pi_{M,s}),\ \cup^i_{j=0} X_j)$, that is, X_{i+1} is an answer set for $e_{U_i}(P^{i+1}_{M,s},\ \cup^i_{j=0} X_j)$.

Note that the atoms in the head of the rules in $\Pi^i_{M,s}$ do not appear in the rules of subprograms $P^0_{M,s}$ to $P^{i-1}_{M,s}$. Therefore answer set X_{i+1} exists if X_i exists. We know that X_0 exists which describes the initial state and the domain. Thus answer set for each of the subprograms $X_0, ..., X_{10}$ exists. Then, by the splitting sequence theorem, $Z = \cup^{10}_{i=0}\ X_i$ is an answer set for $\Pi_{M,s}$. Hence $\Pi_{M,s}$ has an answer set.

Theorem 2. *Suppose that a is an ontic action, $(\mu, \beta) \in Effects_a$, Z is an answer set of the ASP program $\Pi_{D,T,a}$ and $occ(a)$, $pre\_hold(s) \in Z$.*

1. *For $i \in \mathcal{AG}$, if $\mathsf{entails}(s, \delta_{i,a})$, $\mathsf{entails}(s, \mathbf{B}_i\, \mu) \in Z$ then $\mathsf{entails\_n}(s', \mathbf{B}_i\, \ell) \in Z$ for $\ell \in \beta$.*
2. *Suppose that $\mathsf{entails}(s, \neg\delta_{i,a}) \in Z$. For a belief formula η, $\mathsf{entails\_n}(s', \mathbf{B}_i\, \eta) \in Z$ if and only if $\mathsf{entails}(s, \mathbf{B}_i\, \eta) \in Z$.*
3. *Suppose that $\mathsf{entails}(s, \delta_{i,a})$, $\mathsf{entails}(s, \mathbf{B}_i\, \delta_{j,a}) \in Z$ where $i \neq j$, $i, j \in \mathcal{AG}$. If $\mathsf{entails}(s, \mathbf{B}_i\, \mathbf{B}_j\, \mu) \in Z$ then $\mathsf{entails\_n}(s', \mathbf{B}_i\, \mathbf{B}_j\, \ell) \in Z$ holds, for $\ell \in \beta$.*
4. *Suppose that $\mathsf{entails}(s, \mathbf{B}_i\, \neg\delta_{j,a}) \in Z$ holds where $i \neq j$, $i, j \in \mathcal{AG}$. For a belief formula η, if $\mathsf{entails}(s, \mathbf{B}_i\, \mathbf{B}_j\, \eta) \in Z$ then $\mathsf{entails\_n}(s', \mathbf{B}_i\, \mathbf{B}_j\, \eta) \in Z$.*

Proof of Theorem 2:

1) Let (M,s) be the initial state, (M',s') be the next state and Z be an answer set of the ASP program $\Pi_{D,T,a}$. By assumption, *pre_hold(s)*, $entails(s,\delta_{i,a})$, $entails(s,\mathbf{B}_i\,\mu)$ atoms belong to Z. Since $entails(s,\delta_{i,a}) \in Z$ then *f_obs(i,a,s)* $\in Z$ by the rule (41).

We examine the worlds $u \in M[S]$ such that $(s,u) \in M[i]$, i.e. *access(i,s,u)* atom is in the input. Since $(M,s) \models \mathbf{B}_i\,\mu$, we know $(M,u) \models \mu$ for every $u \in M[S]$ such that $(s,u) \in M[i]$. This implies that $entails(u,\mu) \in Z$. There are two cases to consider:

(i) There exists $u \in M[S]$ such that $(s,u) \in M[i]$ and $(M,u) \models \psi \wedge \delta_{i,a}$. Namely, Z includes $entails(u,\psi\wedge\delta_{i,a})$. In this case, $\neg$*ontic_cond(i,s)* atom is in Z, by the rule (50). Then according to the rule (51) and rule (47), *access_n(i,s',u')* and *world_n(u')* atom are generated respectively. That is, $(s',u') \in M'[i]$. The rules (95)–(97) computes the valuation of u' world for an ontic action i.e. $M'[\pi](u') = \phi(a,\pi(u))$. Since $entails(u,\mu)$ is in Z, the rule (95) ensures that $val\_n(u',\ell)$ atom is in the answer set Z where ℓ is a literal in β. This means that if $(s',u') \in M'[i]$ then $(M',u') \models \ell$ holds for $\ell \in \beta$. Therefore, $Z \models \mathbf{B}_i\ell$ holds, $\ell \in \beta$.

(ii) $(M,u) \models \neg(\psi \wedge \delta_{i,a})$ for all $u \in M[S]$ such that $(s,u) \in M[i]$. Namely, there is no $u \in M[S]$ such that *access(i,s,u)* and $entails(u,\psi \wedge \delta_{i,a})$ are in Z. In this case, $\neg ontic\_cond(i,s)$ atom is not in Z. Then the rule (52) generates $access\_n(i,s',u_i)$ atom and the rule (47) generates $world\_n(u_i)$ atom. Hence $(s',u_i) \in M'[i]$. The rules (98), (99), (100) define the interpretation $\lambda(u_i) = (\pi(u) \setminus (\overline{\psi \cup \delta_{i,a}})) \cup (\psi \cup \delta_{i,a})$. Then the set of rules (107)–(109) compute whether $\lambda(u_i)$ entails a fluent formula μ. The valuation $M'[\pi](u_i) = \phi(a,\lambda(u_i))$ is computed by the rules (113)–(116). Since $(M,u) \models \mu$, we have $\lambda(u_i) \models \mu$. Then the rule (114) ensures that $val\_n(u_i,\ell)$ atom is in the answer set Z where ℓ is a literal in β. Thus, if $(s',u_i) \in M'[i]$ then $(M',u_i) \models \ell$ holds for $\ell \in \beta$. Consequently, $Z \models \mathbf{B}_i\ell$ holds, $\ell \in \beta$.

In both cases we have shown that $(M',s') \models \mathbf{B}_i\ell$, for $\ell \in \varphi$ hence the result is established.

2) Let (M,s) be the initial state, (M',s') be the next state and Z be an answer set of the ASP program $\Pi_{D,T,a}$. Assume that *pre_hold(s)*, $entails(s,\neg\delta_{i,a})$ atoms belong to Z. Then according to the rules (41), (42) *f_obs(i,a,s)* $\notin Z$ and *p_obs(i,a,s)* $\notin Z$. Hence *obliv(i,a,s)* $\in Z$ by the rule (60).

Suppose that $entails(s,\mathbf{B}_i\eta) \in Z$ for a belief formula η. Then $(M,u) \models \eta$ for $u \in M[S]$ such that $(s,u) \in M[i]$. Since *obliv(i,a,s)* $\in Z$, according to the rule (60), *access_n(i,s',u)* $\in Z$ if and only if *access(i,s,u)* $\in Z$. Namely, $(s',u) \in M'[i]$ if and only if $(s,u) \in M[i]$. Note that if *access_n(i,s',u)* $\in Z$ then the rule (47) ensures that *world_n(u)* $\in Z$. Then the rule (59) imposes that if *access(j,u,v)* $\in Z$ then *access_n(j,u,v)* $\in Z$ for $j \in \mathcal{AG}$. Hence the accessibility relations at $u \in M'[S]$ are the same as $u \in M[S]$. Therefore $entails\_n(u,\eta) \in Z$ if and only if $entails(u,\eta) \in Z$. Consequently, we obtain $entails\_n(s',\mathbf{B}_i\,\eta) \in Z$ if and only if $entails(s,\mathbf{B}_i\,\eta) \in Z$.

3) Let (M,s) be the initial state, (M',s') be the next state and Z be an answer set of the ASP program $\Pi_{D,T,a}$. Assume that *pre_hold(s)*, $entails(s,\delta_{i,a})$, $entails(s,\mathbf{B}_i\,\delta_{j,a})$, $entails(s,\mathbf{B}_i\,\mathbf{B}_j\,\mu)$ atoms belong to Z. Since $entails(s,\delta_{i,a}) \in Z$ then *f_obs(i,a,s)* $\in Z$ by the rule (41). By assumption, if *access(i,s,u)* $\in Z$ and *access(j,u,v)* $\in Z$, then $entails(u,\delta_{j,a}) \in Z$ and $entails(s,v,\mu) \in Z$.

Since *f_obs(i,a,s)* $\in Z$, according to the rules (51), (52), there are two cases to consider:

(i) $entails(u,f_1)$, $formula\_full(i,a,f_1) \in Z$. Then *access_n(i,s',u')* $\in Z$ by the rule (51) and $\neg$*ontic_cond(i,s)* $\in Z$ by the rule (50). Since $entails(u,\delta_{j,a}) \in Z$, there are two subcases: (a) $entails(v,f_2)$, $formula\_full(j,a,f_2) \in Z$. In this subcase *access_n(j,u',v')* $\in Z$ by the rule (53) and $\neg$*ontic_cond(j,u)* $\in Z$ by the rule (50). By assumption $entails(v,\mu) \in Z$ hence the rule (95) imposes $val\_n(v',\ell)$ atom is in the answer set Z, for $\ell \in \beta$. Then by the rules (34), (35) we have $entails\_n(u',\mathbf{B}_j\ell) \in Z$. (b) $\neg$*ontic_cond(j,u)* $\notin Z$. In this subcase $access\_n(j,u',v_j) \in Z$ by the rule (54). By assumption $entails(v,\mu) \in Z$ hence $lambda(v_j,\mu) \in Z$ by the rule (100). Since $lambda(v_j,\mu) \in Z$, the rule (107) ensures that $entails\_lambda(v_j,\mu) \in Z$. Then the rule (114) imposes $val\_n(v_j,\ell)$ atom is in the answer set Z, for $\ell \in \beta$. According to the rules (34), (35) we have $entails\_n(u',\mathbf{B}_j\ell) \in Z$. In each subcase, we have shown that $entails\_n(u',\mathbf{B}_j\ell) \in Z$. Therefore, (34), (35) imply $entails\_n(s',\mathbf{B}_i\mathbf{B}_j\ell) \in Z$.

(ii) $\neg$*ontic_cond(i,s)* $\notin Z$. In this subcase $access\_n(i,s',u_i) \in Z$ by the rule (52). Since $entails(u,\delta_{j,a}) \in Z$, there are two subcases: (a) *entails(v,f)*, *formula_full(j,a,f)* $\in Z$. In this subcase $access\_n(j,u_i,v') \in Z$ by the rule (55) and $\neg$*ontic_cond(j,u)* $\in Z$ by the rule (50). By assumption $entails(v,\mu) \in Z$ hence the rule

(95) imposes $val\_n(v', \ell)$ atom is in the answer set Z, for $\ell \in \beta$. Then by the rules (34), (35) we have $entails\_n(u_i, \mathbf{B}_j \ell) \in Z$. (b) $\neg ontic\_cond(j, u) \notin Z$. In this subcase $access\_n(j, u_i, v_j) \in Z$ by the rule (56). By assumption $entails(v, \mu) \in Z$ hence $lambda(v_j, \mu) \in Z$ by the rule (100). Since $lambda(v_j, \mu) \in Z$, the rule (107) ensures that $entails\_lambda(v_j, \mu) \in Z$. Then the rule (114) imposes $val\_n(v_j, \ell)$ atom is in the answer set Z, for $\ell \in \beta$. According to the rules (34), (35) we have $entails\_n(u_i, \mathbf{B}_j \ell) \in Z$. In each subcase, we have shown that $entails\_n(u_i, \mathbf{B}_j \ell) \in Z$. Therefore, (34), (35) imply $entails\_n(s', \mathbf{B}_i \mathbf{B}_j \ell) \in Z$, for $\ell \in \beta$.

4) Let (M, s) be the initial state, (M', s') be the next state and Z be an answer set of the ASP program $\Pi_{D,T,a}$. Assume that *pre_hold(s)*, $entails(s, \mathbf{B}_i \neg\delta_{j,a})$, $entails(s, \mathbf{B}_i \mathbf{B}_j \eta)$ atoms belong to Z. Suppose that *access(i,s,u)* $\in Z$ and *access(j,u,v)* $\in Z$. Then $entails(u, \neg\delta_{j,a}) \in Z$ and $entails(v, \eta) \in Z$. Note that the valuation of the world v remains the same between M and M' due to rule (93). Hence $entails\_n(v, \eta) \in Z$. By the rules (41), (42) *f_obs(j,a,u)* $\notin Z$ and *p_obs(j,a,u)* $\notin Z$. Hence *obliv(j,a,u)* $\in Z$ by the rule (60). There exist two cases, agent i is either full observer or oblivious at s:
(i) $entails(s, \delta_{i,a}) \in Z$ i.e. $(M, s) \models \delta_{i,a}$. Then *f_obs(i,a,s)* $\in Z$ by the rule (41). At world $u \in M[S]$, there are two subcases in (M', s'): (a) *access_n(i,s',u')* $\in Z$. Since *obliv(j,a,u)* $\in Z$, the rule (60) implies that if *access(j,u,v)* $\in Z$ then *access_n(j,u',v)* $\in Z$. Since $entails\_n(v, \eta) \in Z$, we have $entails\_n(u', \mathbf{B}_j \eta) \in Z$ for every $u' \in M'[S]$ such that *access_n(i,s',u')* $\in Z$.
(b) $access\_n(i, s', u_i) \in Z$. Since *obliv(j,a,u)* $\in Z$ and $j \neq i$, the rule (61) implies that if *access(j,u,v)* $\in Z$ then $access\_n(j, u_i, v) \in Z$. Since $entails\_n(v, \eta) \in Z$, we have $entails\_n(u_i, \mathbf{B}_j \eta) \in Z$ for every $u_i \in M'[S]$ such that $access\_n(i, s', u_i) \in Z$. In both subcases, we conclude that if *access_n(i,s',t)* $\in Z$ then $entails\_n(t, \mathbf{B}_j \eta) \in Z$. Therefore $entails\_n(s', \mathbf{B}_i \mathbf{B}_j \eta) \in Z$.
(ii) $entails(s, \neg\delta_{i,a}) \in Z$ i.e. $(M, s) \models \neg\delta_{i,a}$. Then *obliv(i,a,s)* $\in Z$ by the rule (43). In this case *access_n(i,s',u)* $\in Z$ by the rule (60). Moreover *world_n(u)* $\in Z$ by the rule (47). Since *obliv(j,a,u)* $\in Z$, the rule (60) implies that if *access(j,u,v)* $\in Z$ then *access_n(j,u,v)* $\in Z$. Since $entails\_n(v, \eta) \in Z$, we have $entails\_n(u, \mathbf{B}_j \eta) \in Z$ for every $u \in M'[S]$ such that *access_n(i,s',u)* $\in Z$. Therefore $entails\_n(s', \mathbf{B}_i \mathbf{B}_j \eta) \in Z$.

Theorem 3. *Suppose that a is a sensing/announcement action, Z is an answer set of the ASP program $\Pi_{D,T,a}$ and $occ(a)$, $pre\_hold(s) \in Z$.*

1. *For $i \in \mathcal{AG}$, $\ell \in \varphi$, if $entails(s, \delta_{i,a})$, $entails(s, \ell) \in Z$ then $entails\_n(s', \mathbf{B}_i \ell) \in Z$.*
2. *For $i \in \mathcal{AG}$, $\ell \in \varphi$, if $entails(s, \delta_{i,a})$, $entails(s, \neg\ell) \in Z$ then $entails\_n(s', \mathbf{B}_i \neg\ell) \in Z$.*
3. *Suppose that $entails(s, \theta_{i,a})$, $entails(s, \mathbf{B}_i \delta_{j,a}) \in Z$ where $i \neq j$, $i, j \in \mathcal{AG}$. Then $entails\_n(s', \mathbf{B}_i (\mathbf{B}_j \ell \vee \mathbf{B}_j \bar{\ell})) \in Z$ for $\ell \in \varphi$.*
4. *Suppose that $obliv(i, a, s) \in Z$. For a belief formula η, $entails\_n(s', \mathbf{B}_i \eta) \in Z$ if and only if $entails(s, \mathbf{B}_i \eta) \in Z$.*

Proof of Theorem 3:

1) Let (M, s) be the initial state, (M', s') be the next state and Z be an answer set of the ASP program $\Pi_{D,T,a}$. Assume that *pre_hold(s)*, $entails(s, \delta_{i,a})$, $entails(s, \ell)$ atoms belong to Z. Since $entails(s, \delta_{i,a}) \in Z$ then *f_obs(i,a,s)* $\in Z$ by the rule (41). We examine the worlds $u \in M[S]$ such that $(s, u) \in M[i]$, i.e. *access(i,s,u)* atom is in the input. For the world u, there are two cases to consider:
(i) $entails(u, \psi \wedge \delta_{i,\mathsf{a}}) \in Z$ and *var_diff(s,u)* $\notin Z$. In this case, $\neg$*sa_f_cond(i,s)* atom is in Z, by the rule (66). Since *var_diff(s,u)* $\notin Z$, it must be that $entails(u, \ell) \in Z$ for $\ell \in \varphi$ by the rules (62), (63). As *pre_hold(s)* and $entails(u, \psi \wedge \delta_{i,\mathsf{a}})$ atoms are in Z, *access_n(i,s',u')* $\in Z$ by the rule (67) and consequently *world_n(u')* $\in Z$ by the rule (47). That is, $(s', u') \in M'[i]$. According to the rule (118), valuation of u' is the same as u i.e. $M'[\pi](u') = M[\pi](u)$. Therefore $val\_n(u', \ell)$ and $entails\_n(u', \ell)$ atoms are in Z.
(ii) $\neg$*sa_f_cond(i,s)* $\notin Z$. In this case, $access\_n(i, s', u^f_{i,s}) \in Z$ by the rule (68) and consequently $world\_n(u^f_{i,s}) \in Z$ by the rule (47). According to the rule (124), valuation of sensing/announcement variables φ are the same across s and $u^f_{i,s}$. Hence $val\_n(u^f_{i,s}, \ell)$ and $entails\_n(u^f_{i,s}, \ell)$ atoms are in Z.
In both cases, we have obtained that if *access_n(i,s',t)* $\in Z$ then $entails\_n(t, \ell) \in Z$. Therefore $entails\_n(s', \mathbf{B}_i \ell) \in Z$.

2) The proof for part (2) is similar to part (1), with the only exception that $(M, s) \models \overline{\ell}$. Hence by replacing ℓ with $\overline{\ell}$ in the proof of part (1), we obtain the result in part (2).

3) Let (M, s) be the initial state, (M', s') be the next state and Z be an answer set of the ASP program $\Pi_{D,T,a}$. Assume that *pre_hold(s)*, *entails*$(s, \theta_{i,a})$, *entails*$(s, \mathbf{B}_i\, \delta_{j,a})$ atoms belong to Z where $i \neq j$, $i, j \in \mathcal{AG}$. Since *entails*$(s, \theta_{i,a}) \in Z$ then *p_obs(i,a,s)* $\in Z$ by the rule (42). Suppose that *access(i,s,u)*, *access(j,u,v)* atoms are in the input i.e. $(s, u) \in M[i]$, $(u, v) \in M[j]$. By assumption *entails*$(u, \delta_{j,a}) \in Z$. Since *entails*$(s, \theta_{i,a})$, there are two cases for agent i:
(i) *entails*$(u, \psi \wedge \theta_{j,a}) \in Z$. Then *access_n(i,s',u')* $\in Z$ by the rule (73). In this case, $\neg$*sa_p_cond(i,s)* atom is in Z, by the rule (72). Since *entails*$(u, \delta_{j,a}) \in Z$, there are two subcases: (a) *entails*$(v, \psi \wedge \delta_{j,\mathsf{a}}) \in Z$ and *var_diff(u,v)* $\notin Z$. In this case, $\neg$*sa_f_cond(j,u)* atom is in Z, by the rule (66). Then *access_n*$(j, u', v') \in Z$ by the rule (69) and consequently *world_n*$(v') \in Z$ by the rule (48). Since *var_diff(u,v)* $\notin Z$, the value of sensing/announcement variables in φ must be the same across u and v according to the rules (62), (63). This means that if *entails*$(u, \ell) \in Z$ then if *entails_n*$(v', \ell) \in Z$, or if *entails*$(u, \neg\ell) \in Z$ then if *entails_n*$(v', \neg\ell) \in Z$ for $\ell \in \varphi$. Hence *entails_n*$(u', \mathbf{B}_j\, \ell \vee \mathbf{B}_j\, \neg\ell) \in Z$ holds and consequently *entails_n*$(s', \mathbf{B}_i\, (\mathbf{B}_j\, \ell \vee \mathbf{B}_j\, \neg\ell)) \in Z$ for $\ell \in \varphi$.
(b) $\neg$*sa_f_cond(j,u)* $\notin Z$. In this case, *access_n*$(j, u', v^f_{j,u}) \in Z$ by the rule (70) and consequently *world_n*$(v^f_{j,u}) \in Z$ by the rule (48). According to the rule (124), valuation of sensing/announcement variables φ are the same across u and $v^f_{j,u}$. This means that if *entails*$(u, \ell) \in Z$ then if *entails_n*$(v^f_{j,u}, \ell) \in Z$, or if *entails*$(u, \neg\ell) \in Z$ then if *entails_n*$(v^f_{j,u}, \neg\ell) \in Z$ for $\ell \in \varphi$. Hence *entails_n*$(u', \mathbf{B}_j\, \ell \vee \mathbf{B}_j\, \neg\ell) \in Z$ holds and consequently *entails_n*$(s', \mathbf{B}_i\, (\mathbf{B}_j\, \ell \vee \mathbf{B}_j\, \neg\ell)) \in Z$ for $\ell \in \varphi$.
(ii) $\neg$*sa_p_cond(i,s)* $\notin Z$. Then *access_n*$(i, s', u^p_i) \in Z$ by the rule (74). Since *entails*$(u, \delta_{j,a}) \in Z$, there are two subcases: (a) *entails*$(v, \psi \wedge \delta_{j,\mathsf{a}}) \in Z$ and *var_diff(u,v)* $\notin Z$. In this case, $\neg$*sa_f_cond(j,u)* atom is in Z, by the rule (66). Then *access_n*$(j, u^p_i, v') \in Z$ by the rule (82) and consequently *world_n*$(v') \in Z$ by the rule (48). Since *var_diff(u,v)* $\notin Z$, the value of sensing/announcement variables in φ must be the same across u and v according to the rules (62), (63). This means that if *entails*$(u, \ell) \in Z$ then if *entails_n*$(v', \ell) \in Z$, or if *entails*$(u, \neg\ell) \in Z$ then if *entails_n*$(v', \neg\ell) \in Z$ for $\ell \in \varphi$. Hence *entails_n*$(u^p_i, \mathbf{B}_j\, \ell \vee \mathbf{B}_j\, \neg\ell) \in Z$ holds and consequently *entails_n*$(s', \mathbf{B}_i\, (\mathbf{B}_j\, \ell \vee \mathbf{B}_j\, \neg\ell)) \in Z$ for $\ell \in \varphi$.
(b) $\neg$*sa_f_cond(j,u)* $\notin Z$. In this case, *access_n*$(j, u^p_i, v^f_{j,u}) \in Z$ by the rule (83) and consequently *world_n*$(v^f_{j,u}) \in Z$ by the rule (48). According to the rule (124), valuation of sensing/announcement variables φ are the same across u and $v^f_{j,u}$. This means that if *entails*$(u, \ell) \in Z$ then if *entails_n*$(v^f_{j,u}, \ell) \in Z$, or if *entails*$(u, \neg\ell) \in Z$ then if *entails_n*$(v^f_{j,u}, \neg\ell) \in Z$ for $\ell \in \varphi$. Hence *entails_n*$(u^p_i, \mathbf{B}_j\, \ell \vee \mathbf{B}_j\, \neg\ell) \in Z$ holds and consequently *entails_n*$(s', \mathbf{B}_i\, (\mathbf{B}_j\, \ell \vee \mathbf{B}_j\, \neg\ell)) \in Z$ for $\ell \in \varphi$.
In all cases, we obtain *entails_n*$(s', \mathbf{B}_i\, (\mathbf{B}_j\, \ell \vee \mathbf{B}_j\, \neg\ell)) \in Z$ for $\ell \in \varphi$ thus the result holds.

4) Let (M, s) be the initial state, (M', s') be the next state and Z be an answer set of the ASP program $\Pi_{D,T,a}$. Assume that *pre_hold(s)*, *obliv*(i, a, s) atoms belong to Z.

Suppose that *entails*$(s, \mathbf{B}_i \eta) \in Z$ for a belief formula η. Then $(M, u) \models \eta$ for $u \in M[S]$ such that $(s, u) \in M[i]$. Since *obliv(i,a,s)* $\in Z$, according to the rule (90), *access_n(i,s',u)* $\in Z$ if and only if *access(i,s,u)* $\in Z$. Namely, $(s', u) \in M'[i]$ if and only if $(s, u) \in M[i]$. Note that if *access_n(i,s',u)* $\in Z$ then the rule (47) ensures that *world_n(u)* $\in Z$. Then the rule (59) imposes that if *access(j,u,v)* $\in Z$ then *access_n(j,u,v)* $\in Z$ for $j \in \mathcal{AG}$. Hence the accessibility relations at $u \in M'[S]$ are the same as $u \in M[S]$. Therefore *entails_n*$(u, \eta) \in Z$ if and only if *entails*$(u, \eta) \in Z$. Consequently, we obtain *entails_n*$(s', \mathbf{B}_i\, \eta) \in Z$ if and only if *entails*$(s, \mathbf{B}_i\, \eta) \in Z$.

References

1. Baral, C., Gelfond, G., Pontelli, E., Son, T.C.: An action language for multi-agent domains: Foundations. arXiv.org p. https://arxiv.org/abs/1511.01960v3 (2020)
2. Baral, C., Gelfond, G., Pontelli, E., Son, T.C.: An action language for multi-agent domains. Artificial Intelligence **302**, 103601 (2022)

3. Buckingham, D., Kasenberg, D., Scheutz, M.: Simultaneous representation of knowledge and belief for epistemic planning with belief revision pp. 172–181 (2020). https://doi.org/10.24963/kr.2020/18, https://doi.org/10.24963/kr.2020/18
4. Ditmarsch, H.v., van der Hoek, W., Kooi, B.: Dynamic epistemic logic. Springer Publishing Company, Incorporated, 1st edn. (2007)