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

Abstract. In this paper we develop a state transition function for partially observable multi-agent epistemic domains and implement it using Answer Set Programming (ASP). The transition function computes the next state upon an occurrence of a single action. Thus it can be used as a module in epistemic planners. Our transition function incorporates ontic, sensing and announcement actions and allows for arbitrary nested belief formulae and general common knowledge. A novel feature of our model is that upon an action occurrence, an observing agent corrects his (possibly wrong) initial beliefs about action precondition and his observability. By examples, we show that this step is necessary for robust state transition. We establish some properties of our state transition function regarding its soundness in updating beliefs of agents consistent with their observability.

Keywords: Answer Set Programming · Multi-agent systems · Epistemic planning · State transition

1 Introduction

Many Artificial Intelligence applications involve multiple autonomous agents interacting with each other and the environment. Agents can take actions that may change the physical state of the world as well as beliefs of agents. A typical problem in a multi-agent setting is how to update agents' beliefs in a sound manner upon an action occurrence, especially when some agents initially have incorrect or incomplete beliefs about the world and other agents. Another challenge is that not all agents might be able to fully observe the effect of the action. Some agents might only observe that the action takes place but not its effects (partial observer agents) and some agents might be completely unaware of the action occurrence (oblivious agents).

In this paper, we study the abovementioned problem of robust state transition upon an action occurrence in multi-agent epistemic settings. We use possible world semantics in the form of Kripke structure [11] to represent agents' beliefs and investigate action occurrences in possible world semantics. We classify actions into three categories: An *ontic* action changes the actual state of the world by changing the value of fluent(s). A *sensing* action allows an agent to learn the value of a set of fluent variable(s). An *announcement* action conveys

G. Gottlob et al. (Eds.): LPNMR 2022, LNAI 13416, pp. 273–286, 2022.
https://doi.org/10.1007/978-3-031-15707-3_21

the value of a set of fluent variable(s) to other agents. We develop a novel state transition function for ontic, sensing, announcement actions using Answer Set Programming (ASP), a popular logic programming paradigm. Our model allows for different levels of observability, uncertainity of the initial state, arbitrarily nested belief formulae and general common knowledge. Hence our ASP program can be imported as a module into single and multi agent epistemic planners to compute the next state. Our choice of ASP is due to its capability in writing compact and understandable rules in recursive form for state transition and entailment of belief formulae.

One important feature of our state transition is that when an action occurs, full and partial observer agents correct their initial (possibly wrong) beliefs about action precondition and their observability before the effect is realized. Namely, an observing agent realizes that the precondition of the action holds and he is not ignorant of the action. This correction step is vital for robust state transition because whether the effect of the action is applicable to a world in the Kripke structure depends on satisfaction of precondition and observability conditions at that world. We provide some examples to illustrate that without correcting beliefs, state transition is not robust[1] This is indeed the problem with the existing models of state transition.

One method to compute state transition is to employ *action models*, introduced in [1,2] and later extended to *event update models* in [6,10]. Event update models involve different events and agents' accessibility relations between events depending on their observability. The next state is computed by cross product of the initial Kripke structure with the event update model. However, Example 1 shows that the standard event update model [4,6], by itself, is not capable of correcting agents' beliefs and robust state transition.

Example 1. We examine a scenario with two agents A, B in a power plant. Agent B has a voltmeter device which senses the level of the voltage. At the actual state, the voltmeter is sound and the voltage level is normal. Agent A initially believes that the voltmeter is defective and he does not know the voltage level. This state is represented as a pointed Kripke structure, as in Fig. 1(a), top. Possible worlds are represented by circles. A double circle represents the true world. Links between worlds encode the belief accessibility relations of agents. Suppose that agent B takes the *check_voltage* action which senses the voltage level. Its precondition is the device being sound, i.e., *sound* and the condition for full observability is $\top$. Hence A and B are full observers at all worlds. The event update model for *check_voltage* is given in Fig. 1(a), bottom left. The σ event corresponds to sensed value being *normal* and the τ event corresponds to $\neg$*normal*. The result of applying this event model to the initial state is given in Fig. 1(a) bottom right. The action model removes the accessibility relation of agent A from world s to u and v because u, v do not satisfy action precondition. In the next state, A has no accessibility relation and believes in every formula.

[1] Details of state transition in these examples can be found in our online appendix at https://github.com/yizmirlioglu/Epistemic.

Intuitively, when A observes the action, he should realize that the meter is sound and learn the voltage level. Therefore, the event model in [4,6], by itself, is not capable of correcting agent's beliefs and robust state transition.

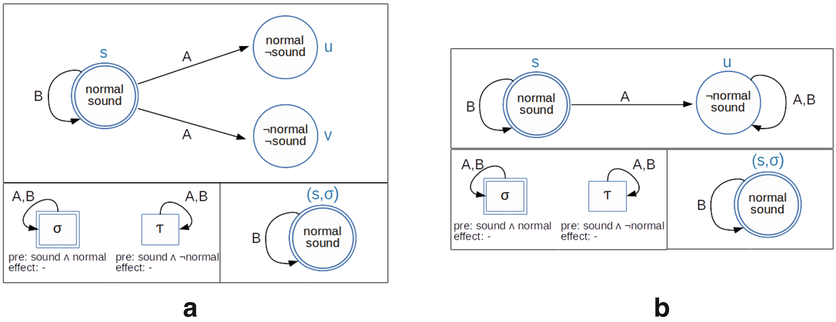

Fig. 1. (a) The first example (b) The second example

Example 2. Consider a variation of Example 1 depicted in Fig. 1 (b). Now agent A initially knows that the meter is sound, but he has incorrect belief that the voltage level is not normal. Agent B performs the *check_voltage* action as before. Applying the event model for the *check_voltage* action to the initial Kripke structure, we obtain the next state shown in Fig. 1(b) (bottom right). Again agent A ends up having no accessibility relation. Ideally A should change his belief and knows that the voltage level is normal. Hence the next state is counter-intuitive.

[7] has constructed a model of state transition where agents correct their beliefs about action precondition. However, their transition function does not involve belief correction for observability. Their framework requires two separate operators for belief and knowledge. As the next example suggests, correcting beliefs for an agent's own observability as well as his beliefs about other agents' observability are necessary for robust state transition.

Example 3. We examine another scenario in Fig. 2(left) with two agents A, B. The knowledge and beliefs of the agents are encoded with the knowledge and the belief accessibility relations. At the actual world, the door is closed and both agents are near to the door. Initially agent A believes that both A, B are near the door, however B believes that A is near but B is far from the door (wrong initial belief). Agent A performs the *open_door* action whose precondition is *haskey_a* and effect is *open*. The condition for full observability of agent A, B is *near_a*, *near_b* respectively. The next state according to transition function of [7] is shown in Fig. 2 (right). At the next state according to [7], agent B believes that the door is open but believes that he is far from the door. This is not a realistic outcome because B wouldn't observe opening the door if he were far from the door.

The above discussion inspires us to develop a robust state transition function for multi-agent domains. Using ASP, we compute state transition which corrects agents' beliefs about action precondition, observability and effect of the action. In

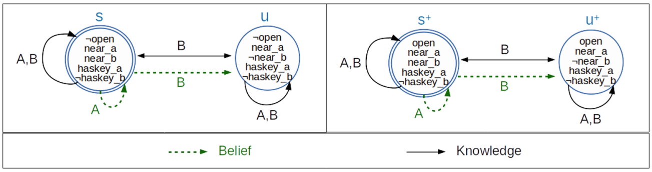

Fig. 2. Example 3

sensing/announcement actions, partial observer agents correct their beliefs about precondition and observability; full observer agents, in addition, also correct their beliefs about the sensing/announcement variables. We provide theorems about soundness of our state transition function in updating beliefs of the agents.

2 Preliminaries

Possible World Semantics: Let $\mathcal{AG}$ be a finite and non-empty set of agents and $\mathcal{F}$ be a set of fluents encoding the properties of the world. *Belief formulae* over $\langle \mathcal{AG}, \mathcal{F} \rangle$ are defined by the BNF:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \mathbf{B}_i \varphi$$

where $p \in \mathcal{F}$ is a fluent and $i \in \mathcal{AG}$. We refer to a belief formula which does not contain any occurrence of $\mathbf{B}_i$ as *a fluent formula*. In addition, for a formula γ and a non-empty set $\alpha \subseteq \mathcal{AG}$, $\mathbf{B}_\alpha \gamma$ and $\mathbf{C}_\alpha \gamma$ denote $\bigwedge_{i \in \alpha} \mathbf{B}_i \gamma$ and $\bigwedge_{k=0}^{\infty} \mathbf{B}_\alpha^k \gamma$, where $\mathbf{B}_\alpha^0 \gamma = \gamma$ and $\mathbf{B}_\alpha^{k+1} \gamma = \mathbf{B}_\alpha^k \mathbf{B}_\alpha \gamma$ for $k \geq 0$, respectively. Let $\mathcal{L}_{\mathcal{AG}}$ denote the set of belief formulae over $\langle \mathcal{AG}, \mathcal{F} \rangle$.

Satisfaction of belief formulae is defined over *pointed Kripke structures* [11]. A Kripke structure M is a tuple $\langle S, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n \rangle$, where S is a set of worlds (denoted by $M[S]$), $\pi : S \mapsto 2^{\mathcal{F}}$ is a function that associates an interpretation of $\mathcal{F}$ to each element of S (denoted by $M[\pi]$), and for $i \in \mathcal{AG}$, $\mathcal{B}_i \subseteq S \times S$ is a binary relation over S (denoted by $M[i]$). For convenience, we will often draw a Kripke structure M as a directed labeled graph, whose set of labeled nodes represent S and whose set of labeled edges contains $s \xrightarrow{i} t$ iff $(s, t) \in \mathcal{B}_i$. The label of each node is its interpretation and the name of the world is written above the node. For $u \in S$ and a fluent formula φ, $M[\pi](u)$ and $M[\pi](u)(\varphi)$ denote the interpretation associated to u via π and the truth value of φ with respect to $M[\pi](u)$. For a world $u \in M[S]$, (M, u) is a *pointed Kripke structure*, also called *state* hereafter.

Given a belief formula φ and a state (M, u), $(M, u) \vDash \mathbf{B}_i \varphi$ if $(M, t) \vDash \varphi$ for every t such that $(u, t) \in \mathcal{B}_i$. $(M, u) \vDash \mathbf{C}_G \varphi$ if $(M, u) \vDash \varphi$ and $(M, t) \vDash \varphi$ for every t such that $(u, t) \in \mathbf{R}_G^*$ where $\mathbf{R}_G^*$ is the transitive closure of $\mathcal{B}_i$, $i \in G$.

For a fluent $f \in \mathcal{F}$, let $\overline{f} = \neg f$ and $\overline{\neg f} = f$; and for a set of fluent literals X, let $\overline{X} = \{\overline{\ell} \mid \ell \in X\}$. If $\chi = b_1 \wedge ... \wedge b_e$ and $\gamma = l_1 \wedge ... \wedge l_g$ are conjunctions of fluent literals, $\chi \cup \gamma$ denotes the set $\{b_1, ..., b_e, l_1, ..., l_g\}$.

Domain Description: Let $D = \langle \mathcal{AG}, \mathcal{F}, \mathcal{A} \rangle$ be a multi-agent epistemic domain. We assume that the precondition of an action is of the form $\psi = h_1 \wedge ... \wedge h_r \wedge \xi$ where ξ is a belief formula. In the multi-agent action language m$\mathcal{A}^*$ [4], the precondition of action a is encoded by the statement "**executable** a **if** ψ". We allow for conditional effects of an ontic action a. In m$\mathcal{A}^*$, effect of an ontic action a is described by "a **causes** β **if** μ", where μ is a fluent formula and β is a set of literals. Intuitively, if condition μ holds at a world u, the action replaces the relevant literals in the world with the ones in β. Let $Effects_{\mathsf{a}}$ be the set of (μ, β) pairs. We assume that if (μ, φ) and (μ', φ') are in $Effects_{\mathsf{a}}$ then $\mu \wedge \mu'$ is inconsistent. $M'[\pi](u') = \phi(\mathsf{a}, \pi(u))$ stands for interpretation of the resultant world u' upon applying the action a on the world u. Formally, if $(M, u) \vDash \mu$ and $(\mu, \beta) \in Effects_{\mathsf{a}}$, then $M'[\pi](u') = (\pi(u) \setminus \overline{\beta}) \cup \beta$. If $(M, u) \nvDash \mu$ for any $(\mu, \beta) \in Effects_{\mathsf{a}}$ then $M'[\pi](u') = \pi(u)$.

m$\mathcal{A}^*$ describes the effects of sensing and announcement actions by the statements "a **determines** φ" and "a **announces** φ" respectively. In the sensing actions, $\varphi = \{\rho_1, ..., \rho_o\}$ is the set of fluents that the agent senses, whereas in the announcement actions, φ is the set of fluents that the agent announces.

Full and partial observability conditions are encoded in m$\mathcal{A}^*$ as "i **observes** a **if** $\delta_{i,\mathsf{a}}$" and "i **aware_of** a **if** $\theta_{i,\mathsf{a}}$" respectively. We assume $\delta_{i,\mathsf{a}}$, $\theta_{i,\mathsf{a}}$ are conjuction of literals and they are pairwise disjoint. Note that observability depends on a world and it is defined over pointed Kripke structures. In case neither $\delta_{i,\mathsf{a}}$ nor $\theta_{i,\mathsf{a}}$ holds at (M, u), then agent i is oblivious at (M, u).

We say that a domain D is consistent if it satisfies the above conditions for action description and observability rules. We define the initial state as $T = (M, s)$ where s is the actual world.

Answer Set Programming: Answer Set Programming (ASP) is a knowledge representation and reasoning paradigm [12,13] which provides a formal framework for declaratively solving problems. The idea of ASP is to model a problem by a set of logical formulas (called rules), so that its models (called answer sets) characterize the solutions of the problem. Our ASP formulation is based on stable model semantics [12]. ASP provides logical formulas, called rules, of the form

$$Head \leftarrow L_1, \ldots, L_k, not\ L_{k+1}, \ldots, not\ L_l \tag{1}$$

where $l \geq k \geq 0$, *Head* is a literal (i.e., an atom A or its negation $\neg A$) or $\bot$, and each L_i is a literal. A rule is called a *constraint* if *Head* is $\bot$, and a *fact* if $l = 0$. A set of rules is called a *program*. ASP provides special constructs to express nondeterministic choices, cardinality constraints, and aggregates. Programs using these constructs can be viewed as abbreviations for programs that consist of rules of the form (1). Further information about ASP can be found in [14].

3 State Transition Using ASP

Let $D = \langle \mathcal{AG}, \mathcal{F}, \mathcal{A} \rangle$ be a consistent multi-agent domain and (M, s) be the initial state. We study the problem of computing the next state $\Phi_D(\mathsf{a}, (M, s))$ given the initial state (M, s) and the occurrence of action a. Our state transition function

$\Phi_D(\mathsf{a},(M,s))$ works as follows: In ontic actions, we first correct full observer agents' beliefs about action precondition and observability, and then apply the effect of the action by modifying the relevant fluents. Namely, full observers observe the effects of the action and correct their beliefs, while oblivious agents remain in the old state.

Sensing and announcement actions do not alter the actual world, they only change beliefs of the agents. We assume that agents always announce truthfully and the listening agents always believe in the announced value of variables and update their beliefs accordingly. In sensing/announcement actions, a full observer agent i will correct his beliefs about precondition, his observability, and the sensing/announcement variables, i.e., he will correct the literals in ψ, $\delta_{i,\mathsf{a}}$ and φ. A partial observer agent i will correct his beliefs about only precondition and his observability, but not about the sensing/announcement variables. Beliefs of oblivious agents do not change. By construction, agents also correct their beliefs about belief of other agents, for all types of actions.

Below we build the ASP program $\Pi_{D,T,\mathsf{a}}$ which computes the next state $\Phi_D(\mathsf{a},T)$ given an initial state $T=(M,s)$ and occurrence of an action a. Due to limited space, we provide only the core ASP rules that illustrate the idea behind the formulation; for the full code we refer to our online repository[2].

Input: We represent agents and agent sets by $ag(I)$, $ag\_set(G)$ atoms. $formula(F)$ atom shows the belief formulae that appear in the domain D. Actions are described by $action(A)$, $type(A,Y)$, $exec(A,F)$, $causes(A,L,F)$, $determines(A,F)$, $announces(A,F)$ atoms. $observes(I,A,F)$ and $aware(I,A,F)$ atoms state the condition for full observability and partial observability of agent I, respectively. $pre\_lit(A,F)$ denote the literals $h_1,...,h_r$ in action precondition ψ, $full\_lit(I,A,F)$ denote the literals in $\delta_{i,\mathsf{a}}$ and $partial\_lit(I,A,F)$ 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 $world(U)$, $access(I,U,V)$, $val(U,F)$ atoms, respectively, where I denotes an agent, U and V are worlds, and F is a fluent. For efficiency, we state only the positive literals in the valuation of a world. $actual(S)$ stands for the actual world S. $occ(\mathsf{a})$ atom shows the action a that occurs. The next state is represented by $world\_n(U)$, $actual\_n(Z)$, $access\_n(I,U,V)$, $val\_n(U,F)$.

State Transition: We first 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. Some of the rules that compute entailment of belief formula are:

$$entails(U,F) \leftarrow world(U),\ val(U,F),\ fluent(F). \tag{2}$$

$$entails(U,\neg F) \leftarrow world(U),\ not\ val(U,F),\ fluent(F). \tag{3}$$

$$entails(U,F1\wedge F2) \leftarrow world(U),\ entails(U,F1),\ entails(U,F2),\ formula(F1\wedge F2). \tag{4}$$

$$\neg entails(U,B_I F) \leftarrow world(U),\ access(I,U,V),\ not\ entails(V,F),\ formula(B_I F). \tag{5}$$

[2] https://github.com/yizmirlioglu/Epistemic.

$$entails(U, B_I F) \leftarrow not\, \neg entails(U, B_I F),\ world(U),\ formula(B_I F). \tag{6}$$

Rule (5) 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 (6).

Then we compute observability of the agents at each world by

$$f\_obs(I, A, U) \leftarrow observes(I, A, F),\ entails(U, F),\ world(U),\ occ(A). \tag{7}$$

$$p\_obs(I, A, U) \leftarrow aware(I, A, F),\ entails(U, F),\ world(U),\ occ(A). \tag{8}$$

$$obliv(I, A, U) \leftarrow not\, f\_obs(I, A, U),\ not\, p\_obs(I, A, U),\ world(U),\ ag(I),\ occ(A). \tag{9}$$

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

$$pre\_hold(S) \leftarrow actual(S),\ entails(S, F),\ exec(A, F),\ occ(A). \tag{10}$$

$$actual\_n(S') \leftarrow actual(S),\ pre\_hold(S),\ occ(A). \tag{11}$$

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

$$world\_n(S') \leftarrow actual(S),\ pre\_hold(S),\ occ(A). \tag{12}$$

$$world\_n(V) \leftarrow actual\_n(Z),\ access\_n(I, Z, V). \tag{13}$$

$$world\_n(V) \leftarrow world\_n(U),\ access\_n(I, U, V). \tag{14}$$

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 $(M, U) \models \delta_{i,\mathsf{a}}$ 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 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.

$$formula\_full(I, A, F1 \wedge F2) \leftarrow exec(A, F1),\ observes(I, A, F2),\ ag(I). \tag{15}$$

$$\neg ontic\_cond(I, U) \leftarrow access(I, U, V),\ entails(V, F),\ formula\_full(I, A, F), \\ occ(A),\ type(A, ontic). \tag{16}$$

$$access\_n(I, U', V') \leftarrow world\_n(U'),\ access(I, U, V),\ f\_obs(I, A, U), \\ entails(V, F),\ formula\_full(I, A, F),\ occ(A),\ type(A, ontic). \tag{17}$$

$$access\_n(I, U', V_I) \leftarrow world\_n(U'),\ access(I, U, V),\ f\_obs(I, A, U), \\ not\, \neg ontic\_cond(I, U),\ occ(A),\ type(A, ontic). \tag{18}$$

Oblivious agents remain at the old state and their beliefs do not change. We keep all accessibility relations in M so that beliefs of oblivious agents remain the same, namely $M[i] \subseteq M'[i]$ for all $i \in \mathcal{AG}$.

$$access\_n(I,U,V) \leftarrow world\_n(U),\ access(I,U,V),\ occ(A). \tag{19}$$

$$access\_n(I,U',V) \leftarrow world\_n(U'),\ access(I,U,V),\ obliv(I,A,U),\ occ(A),\ type(A,ontic). \tag{20}$$

$$access\_n(I,U_J,V) \leftarrow I \neq J,\ world\_n(U_J),\ access(I,U,V),\ obliv(I,A,U),\ occ(A),\ type(A,ontic). \tag{21}$$

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.

$$var\_diff(U,V) \leftarrow access(I,U,V),\ val(U,F),\ not\, val(V,F),\ varphi(A,F),\ occ(A). \tag{22}$$

$$var\_diff(U,V) \leftarrow access(I,U,V),\ not\, val(U,F),\ val(V,F),\ varphi(A,F),\ occ(A). \tag{23}$$

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.

$$\neg sa\_f\_cond(I,U) \leftarrow access(I,U,V),\ entails(V,F),\ formula\_full(I,A,F),\ not\, var\_diff(U,V),\ occ(A),\ type(A,sa). \tag{24}$$

$$access\_n(I,U',V') \leftarrow world\_n(U'),\ access(I,U,V),\ f\_obs(I,A,U),\ entails(V,F),\ formula\_full(I,A,F),\ not\, var\_diff(U,V),\ occ(A),\ type(A,sa). \tag{25}$$

$$access\_n(I,U',V^f_{I,U}) \leftarrow world\_n(U'),\ access(I,U,V),\ f\_obs(I,A,U),\ not\, \neg sa\_f\_cond(I,U),\ occ(A),\ type(A,sa). \tag{26}$$

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.

$$formula\_partial(I,A,F1 \wedge F2) \leftarrow exec(A,F1),\ aware(I,A,F2),\ ag(I). \tag{27}$$

$$\neg sa\_p\_cond(I,U) \leftarrow access(I,U,V),\ entails(V,F),\ formula\_partial(I,A,F),\ occ(A),\ type(A,sa). \tag{28}$$

$$access\_n(I,U',V') \leftarrow world\_n(U'),\ access(I,U,V),\ p\_obs(I,A,U),$$
$$entails(V,F),\ formula\_partial(I,A,F),\ occ(A),\ type(A,sa). \tag{29}$$
$$access\_n(I,U',V_I^p) \leftarrow world\_n(U'),\ access(I,U,V),\ p\_obs(I,A,U),$$
$$not\ \neg sa\_p\_cond(I,U),\ occ(A),\ type(A,sa). \tag{30}$$

Accessibility relations of oblivious agents are constructed in a similar manner to the ontic actions. We also need to compute the valuation function at the next state M'. We first consider ontic actions. Note that μ, 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 $\lambda(U_i) = (\pi(U) \setminus (\overline{\psi \cup \delta_{i,\mathsf{a}}})) \cup (\psi \cup \delta_{i,\mathsf{a}})$ 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

$$lambda(U_I,H) \leftarrow world\_n(U_I),\ pre\_lit(A,H),\ fluent(H),$$
$$occ(A),\ type(A,ontic). \tag{31}$$
$$lambda(U_I,H) \leftarrow world\_n(U_I),\ full\_lit(I,A,H),\ fluent(H),$$
$$occ(A),\ type(A,ontic). \tag{32}$$
$$lambda(U_I,H) \leftarrow world\_n(U_I),\ val(U,H),\ not\ pre\_lit(A,\neg H),$$
$$not\ full\_lit(I,A,\neg H),\ fluent(H),\ occ(A),\ type(A,ontic). \tag{33}$$

Whether the interpretation $\lambda(U_i)$ satisfies a belief formula is denoted by $entails\_lambda(U_i,F)$ atom and can be computed by the ASP rules similar to (2)–(6). Valuation of U', $U_i \in M'[S]$ are computed by $M'[\pi](U') = \phi(\mathsf{a}, \pi(U))$ and $M'[\pi](U_i) = \phi(\mathsf{a}, \lambda(U_i))$ respectively. Namely, if $\pi(U)$ (resp. $\lambda(U_i)$) satisfies μ, then the literals in β are placed into the valuation of U' (resp. U_i).

$$val\_n(U',E) \leftarrow world\_n(U'),\ entails(U,F),\ causes(A,E,F),\ fluent(E),$$
$$occ(A),\ type(A,ontic). \tag{34}$$
$$val\_n(U',H) \leftarrow world\_n(U'),\ val(U,H),\ entails(U,F),\ not\ causes(A,\neg H,F),$$
$$fluent(H),\ occ(A),\ type(A,ontic). \tag{35}$$
$$val\_n(U_I,E) \leftarrow entails\_lambda(U_I,F),\ causes(A,E,F),\ fluent(E),$$
$$occ(A),\ type(A,ontic). \tag{36}$$
$$val\_n(U_I,H) \leftarrow lambda(U_I,H),\ entails\_lambda(U_I,F),\ not\ causes(A,\neg H,F),$$
$$fluent(H),\ occ(A),\ type(A,ontic). \tag{37}$$

Last, we compute the valuation of worlds at the next state for a sensing/announcement action. The valuation of the world $U' \in M'[S]$ is 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 $U \in M[S]$).

$$val\_n(U_I^p, H) \leftarrow world\_n(U_I^p),\ pre\_lit(A, H),\ fluent(H),\ occ(A),\ type(A, sa). \quad (38)$$

$$val\_n(U_I^p, H) \leftarrow world\_n(U_I^p),\ partial\_lit(I, A, H),\ fluent(H),$$
$$occ(A),\ type(A, sa). \quad (39)$$

$$val\_n(U_I^p, H) \leftarrow world\_n(U_I^p),\ val(U, H),\ not\ pre\_lit(A, \neg H),$$
$$not\ partial\_lit(I, A, \neg H),\ fluent(H),\ occ(A),\ type(A, sa). \quad (40)$$

$$val\_n(V_{I,U}^f, H) \leftarrow world\_n(V_{I,U}^f),\ pre\_lit(A, H),\ fluent(H),\ occ(A),\ type(A, sa). \quad (41)$$

$$val\_n(V_{I,U}^f, H) \leftarrow world\_n(V_{I,U}^f),\ full\_lit(I, A, H),\ fluent(H),$$
$$occ(A),\ type(A, sa). \quad (42)$$

$$val\_n(V_{I,U}^f, F) \leftarrow world\_n(V_{I,U}^f),\ varphi(A, F),\ val(U, F),\ occ(A),\ type(A, sa). \quad (43)$$

$$val\_n(V_{I,U}^f, h) \leftarrow world\_n(V_{I,U}^f),\ val(V, H),\ not\ pre\_lit(A, \neg H),$$
$$not\ full\_lit(I, A, \neg H),\ not\ varphi(A, H),\ fluent(H),\ occ(A),\ type(A, sa). \quad (44)$$

To compute the entailment of belief formulae at the next state, we add rules that are analogous to the rules (2)–(6) by replacing $entails(U, F)$, $world(U)$, $access(I, U, V)$, $val(U, F)$ atoms with $entails\_n(U, F)$, $world\_n(U)$, $access\_n(I, U, V)$, $val\_n(U, F)$ respectively.

4 Properties of the State Transition Function

We now provide results that our ASP formulation updates the state and beliefs of agents in a robust way. The proof of the theorems can be found in the appendix, available online[3]. Throughout the section, we assume $D = \langle \mathcal{AG}, \mathcal{F}, \mathcal{A} \rangle$ is a multi-agent epistemic domain and $T = (M, s)$ is the initial state where s is the actual world. We first ensure that the ASP program $\Pi_{D,T,\mathsf{a}}$ yields an answer set.

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

Theorem 2 describes how beliefs of full observer and oblivious agents change due to the occurrence of an ontic action. Full observers observe the effect of the action and update their beliefs accordingly. Beliefs of oblivious agents do not change. Moreover a full observer agent knows that another full observer agent has updated his beliefs and beliefs of oblivious agents stay the same.

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

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

[3] https://github.com/yizmirlioglu/Epistemic.

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

Theorem 3 states that full observers learn the value of the sensing/announcement variables $\ell \in \varphi$ while partial observers know that full observers know the value of sensing variables; belief of oblivious agents stays the same.

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

1. *For* $i{\in}\mathcal{AG}$, $\ell{\in}\varphi$, *if* $entails(s, \delta_{i,\mathsf{a}})$, $entails(s, \ell){\in}Z$ *then* $entails\_n(s', \mathbf{B}_i\ell){\in}Z$.
2. *For* $i{\in}\mathcal{AG}$, $\ell{\in}\varphi$, *if* $entails(s, \delta_{i,\mathsf{a}})$, $entails(s, \neg\ell){\in}Z$ *then* $entails\_n(s', \mathbf{B}_i\neg\ell){\in}Z$.
3. *Suppose that* $entails(s, \theta_{i,a})$, $entails(s, \mathbf{B}_i\, \delta_{j,\mathsf{a}}) \in Z$ *where* $i \neq j$, $i, j \in \mathcal{AG}$. *Then* $entails\_n(s', \mathbf{B}_i\, (\mathbf{B}_j\, \ell \vee \mathbf{B}_j\, \overline{\ell})) \in Z$ *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* $entails(s, \mathbf{B}_i\, \eta) \in Z$.

5 Example Scenarios

This section demonstrates our state transition function by applying it to the example scenarios in the introduction. We consider the belief operator in the Kripke structures at Fig. 1, 2. The ASP encoding of input and output for these scenarios can be found in our online repository. For instance, the initial state and the computed next state of the first scenario are

$actual(s).\ world(s).\ world(u).\ world(v).$

$val(s, normal).\ val(s, sound).\ val(u, normal).\ access(a, s, u).\ access(a, s, v).\ access(b, s, s).$

$actual\_n(prime(s)).\ world\_n(prime(s)).\ world\_n(subf(u, a, s)).\ world\_n(subf(v, a, s)).$

$val\_n(prime(s), normal).\ val\_n(prime(s), sound).\ val\_n(subf(u, a, s), normal).$

$val\_n(subf(u, a, s), sound).\ val\_n(subf(v, a, s), normal).\ val\_n(subf(v, a, s), sound).$

$access\_n(a, prime(s), subf(u, a, s)).\ access\_n(a, prime(s), subf(v, a, s)).\ access\_n(b, prime(s), prime(s)).$

The next state of each scenario is depicted in Fig. 3 according to the ASP output. Now the next state is intuitive: In the first scenario, agent A has corrected his beliefs at world u, v and he believes that the meter is sound and the voltage level is normal. In the second scenario, the transition function was able to revert agent A's initial incorrect belief about the sensing variable *normal*. After the sensing action, A believes that the voltage level is normal as expected. In the third scenario, at the next state B knows that the door is open and he has realized that he is full observer (*near_b*). Besides agent A believes that B is full observer and A believes that B believes that B is full observer.

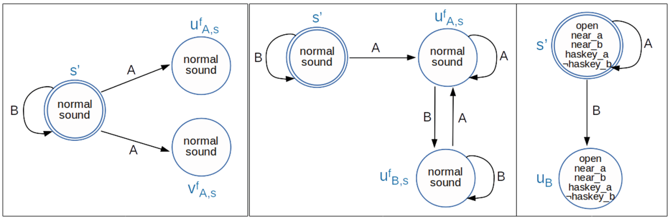

Fig. 3. Solution of example scenarios in the introduction

6 Related Literature

In dynamic epistemic logic literature, state transition in possible world semantics have been studied by [3,4,7,15]. Event update models have also been employed for state transition [1,2,6]. For multi-agent contexts, [4,16] have developed action languages that describe the domain, actions and observability of agents. [4] utilizes a simple belief correction mechanism for sensing/announcement actions where the full observer agents "directly learn the actual state of the world".

[7] proposed an alternative state transition function, where full and partial observers correct their beliefs about action precondition, but not about observability. Observability of agents is computed at the actual world and assumed to be fixed across worlds. Thus an agent corrects his beliefs even in those worlds where he is not a full or partial observer. Conditional effects are not allowed for an ontic action. The authors do not examine how an agent's beliefs about other agents change during state transition. In our model, the knowledge operator is not required and the belief operator is sufficient for belief correction. Besides, we do not assume a fixed observability across all worlds. By construction, our state transition function corrects an agent's first order beliefs and beliefs about other agents (higher order beliefs).

Our work also contributes to the field on applications of Answer Set Programming. ASP has been utilized in epistemic reasoning literature by [5,8,9,17]. [5] have used ASP to encode Kripke structures and showed that epistemic problems such as "Muddy child", "Sum and Product" can be solved in this setting. [9,17] have developed conditional epistemic planners for single agent setting. A multi-agent planner have been implemented using ASP by [8].

7 Conclusion

We have developed an ASP-based state transition function for ontic, sensing and announcement actions for partially observable multi-agent epistemic domains. One novel feature of our transition function is that agents correct their belief about precondition, observability and sensing/announcement variables upon an action occurrence. By examples, we have shown that this step is crucial for observing the effect of the action, and thus for robust state transition.

Answer Set Programming enables us to write state transition in terms of simple, understandable logical rules in recursive form. We establish some properties of our planner regarding its robustness in updating beliefs of agents consistent with their level of observability. For future work, we aim to implement a planner using this ASP formulation. Our transition function can also be used in existing conformant and conditional epistemic planning systems as a module to compute the next state.

Acknowledgment. The authors have been partially supported by NSF grants 2151254, 1914635 and 1757207. Tran Cao Son was also partially supported by NSF grant 1812628.

References

1. Baltag, A., Moss, L.: Logics for epistemic programs. Synthese (2004)
2. Baltag, A., Moss, L., Solecki, S.: The logic of public announcements, common knowledge, and private suspicions. In: 7th TARK, pp. 43–56 (1998)
3. Baral, C., Gelfond, G., Pontelli, E., Son, T.C.: An action language for reasoning about beliefs in multi-agent domains. In: Proceedings of NMR (2012)
4. Baral, C., Gelfond, G., Pontelli, E., Son, T.C.: An action language for multi-agent domains. Artif. Intell. **302**, 103601 (2022)
5. Baral, C., Gelfond, G., Son, T.C., Pontelli, E.: Using answer set programming to model multi-agent scenarios involving agents' knowledge about other's knowledge. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 259–266 (2010)
6. Benthem, J.V., Eijck, J.V., Kooi, B.P.: Logics of communication and change. Inf. Comput. **204**(11), 1620–1662 (2006)
7. Buckingham, D., Kasenberg, D., Scheutz, M.: Simultaneous representation of knowledge and belief for epistemic planning with belief revision, 172–181 (2020). https://doi.org/10.24963/kr.2020/18
8. Burigana, A., Fabiano, F., Dovier, A., Pontelli, E.: Modelling multi-agent epistemic planning in asp. Theory Pract. Logic Program. **20**(5), 593–608 (2020)
9. Cabalar, P., Fandinno, J., del Cerro, L.F.: Dynamic epistemic logic with asp updates: application to conditional planning. arXiv preprint arXiv:1905.10621 (2019)
10. Ditmarsch, H.V., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic, 1st edn. Springer (2007)
11. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning About Knowledge. MIT Press (1995)
12. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R., Bowen, K. (eds.) Logic Programming: Proceedings of the Fifth International Conference and Symposium, pp. 1070–1080 (1988)
13. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Gen. Comput. 365–387 (1991)
14. Lifschitz, V.: Answer Set Programming. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24658-7

15. Pontelli, E., Son, T.C., Baral, C., Gelfond, G.: Answer set programming and planning with knowledge and world-altering actions in multiple agent domains. In: Erdem, E., Lee, J., Lierler, Y., Pearce, D. (eds.) Correct Reasoning. LNCS, vol. 7265, pp. 509–526. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30743-0_35
16. Rajaratnam, D., Thielscher, M.: Representing and reasoning with event models for epistemic planning. In: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, vol. 18, pp. 519–528 (2021)
17. Tu, P.H., Son, T.C., Baral, C.: Reasoning and planning with sensing actions, incomplete information, and static causal laws using answer set programming. Theory Pract. Logic Program. **7**(4), 377–450 (2007)